Distributed Algorithms for Carrying a Ladder by Omnidirectional Robots in Near Optimal Time*

Yuichi Asahiro[1], Hajime Asama[2], Satoshi Fujita[3], Ichiro Suzuki[4], and Masafumi Yamashita[1]

[1] Kyushu University, Fukuoka, Fukuoka 812-8581, Japan
{asahiro,mak}@csce.kyushu-u.ac.jp
[2] The Institute of Physical and Chemical Research (RIKEN)
Wako, Saitama 351-01, Japan
asama@cel.riken.go.jp
[3] Hiroshima University, Kagamiyama, Higashi-Hiroshima 739-8511, Japan
fujita@se.hiroshima-u.ac.jp
[4] University of Wisconsin – Milwaukee, Milwaukee, WI 53201, USA
suzuki@cs.uwm.edu

Abstract. Consider two omnidirectional robots carrying a ladder, one at each end, in the plane without obstacles. Given start and goal positions of the ladder, what is a time-optimal motion of the robots subject to given constraints on their kinematics such as maximum acceleration and velocity? Using optimal control theory, Chen, Suzuki and Yamashita solved this problem under a kinematic constraint that the speed of each robot must be either 0 or a given constant v at any moment during the motion. Their solution, which requires complicated calculation, is centralized and off-line. The objective of this paper is to demonstrate that even without the complicated calculation, a motion that is sufficiently close to time-optimal can be obtained using a simple distributed algorithm in which each robot decides its motion individually based on the current and goal positions of the ladder.

1 Introduction

Recently distributed autonomous robot systems have attracted the attention of many researchers as a new approach for designing intelligent and robust systems [6]. A distributed autonomous robot system is a group of robots that individually and autonomously decide their motion. In particular, in such a system no robot is allowed to act as a leader that controls the other robots in the group.

* This paper has been prepared as a summary of a presentation given at the 1998 Dagstuhl Workshop on the authors' on-going research prőject. This work was supported in part by a Scientific Research Grant-in-Aid from the Ministry of Education, Science and Culture of Japan under Grants, the National Science Foundation under Grant IRI-9307506, the Office of Naval Research under Grant N00014-94-1-0284, and an endowed chair supported by Hitachi Ltd. at Faculty of Engineering Science, Osaka University.

Christensen et al. (Eds.): Sensor Based Intelligent Robots, LNAI 1724, pp. 240–254, 1999.

A crucial issue in designing such a system is to have the robots autonomously coordinate their motion so that the system achieves a certain goal, since in many applications,

1. the interest of an individual robot can conflict with that of the entire system, and
2. each robot may be unaware of the global system state.

Typically there are two kinds of coordination problems; conflict resolution and cooperation. Roughly speaking, conflict resolution attempts to avoid certain (undesirable) situations such as collision of robots. Cooperation on the other hand aims to eventually generate a certain (favorable) situation.

The problem we discuss in this paper is a motion coordination problem for producing a time optimal behavior of the system.

Centralized and distributed approaches are two major paradigms for designing motion coordination in multi-robot systems [9], each with its own merits and demerits. It is not our objective here to discuss them in detail, and we only mention that roughly speaking a centralized approach tends to yield efficient solutions, while solutions obtained by a distributed approach can be more robust against failures. Most of the work in the literature, however, takes either a centralized or a leader-follower approach [2,8,12,13]. The work on distributed approachs often assumes the existence of some navigation devices [15,7], and only a few take a fully distributed approach [1,11,14,16,17]. Among these, Miyata, et al. [11] and Ahmadabadi and Nakano [1] discuss how a group of robots may carry and handle an object, which is the subject of this paper. They investigate how the robots can coordinate their motion to carry an object, while in this paper we investigate how quickly the robots can carry an object.

Recently a time optimal motion of robots has been investigated by some researchers. In [4], Chen, et al. discuss the problem of computing a time-optimal motion for two omnidirectional robots carrying a ladder from an initial position to a final position in a plane without obstacles, and calculate an optimal path using a method based on variational calculus, a branch of functional analysis [5], under the assumption that the speed of each robot must be either 0 or a given constant v at any moment during the motion. In [10] Mediavilla, et al. describe a path planning method for three robots for obtaining collision free time-optimal trajectories using a mathematical programming method. Both of these papers have adopted an off-line (and therefore centralized) setting: a motion is computed and given to each robot in advance.

The objective of this paper is to demonstrate that even without the complicated calculation of [4], a motion that is sufficiently close to time-optimal can be obtained for two omnidirectional robots carrying a ladder, using a simple *distributed algorithm* in which each robot decides its motion individually based on the current and goal positions of the ladder. The algorithms we propose are based on the following simple idea:

> Basically we let each robot pursue its individual interest (e.g., mainly moving toward the goal) when deciding its motion. Since the robots are

carrying a ladder, however, their intended motions may be "incompatible," making it impossible for the robots to move as intended. Now, suppose that there is a (virtual) "coordinator" that resolves the conflict in a "fair" manner for both, allowing the robots to continue to move. Then the resulting motion may be sufficiently close to time-optimal.

We will present two such algorithms. The first one is obtained by a naive application of this idea and works well for many, but not all, instances. The second one requires a slightly more complicated calculation, and works well for many of the instances for which the first algorithm does not.

The paper is organized as follows: Section 2 gives an outline of the method of [4] for computing an exact optimal motion under some (rather strong) assumptions. Section 3 presents the two distributed algorithms we propose. Section 4 describes the physical robots we plan to use and reports the results of some preliminary experiments using them. Section 5 presents the computer simulation results. We then conclude the paper by giving some remarks in Section 6.

2 Time-Optimal Motion for Carrying a Ladder

Chen, et al. discussed the problem of computing a time-optimal motion for two omnidirectional robots carrying a ladder from an initial position to a final position in a plane without obstacles [4]. In order to explain a formidable nature of the problem, suppose that robots A and B must move to A' and B', respectively, as is shown in Fig. 1, carrying a ladder. (We use "A" and "B" to refer to the robots as well as their initial positions.) A time optimal motion is shown in Fig. 2. You can observe that the path of robots B is not a straight line segment, indicating that B yields to give enough time for A to complete the necessary rotation. Thus this time optimal motion is not attainable if one of them insists on its individual interest of reaching its goal as quick as possible.

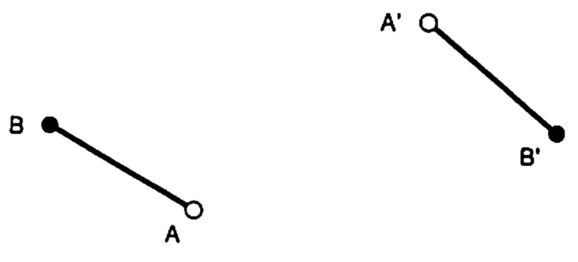

Fig. 1. Robots A (empty circle) and B (filled circle) move to A' and B', respectively, carrying a ladder.

Let v be the maximum speed of a robot. In [4] it is assumed that the speed of a robot is either 0 or v at any moment. (Computing a time-optimal motion in the

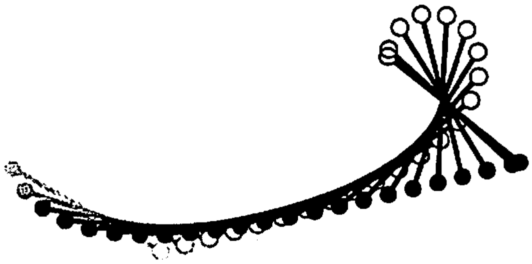

Fig. 2. Time-optimal motion for the instance shown in Fig. 1. The figure consists of a series of snapshots. In the first few shots the segment increases in grayscale as time increases.

general case is still open.) Without loss of generality assume that $|BB'| \geq |AA'|$. For convenience, we take AB to be of unit length ($|AB| = 1$), and let $L = |BB'|$ be the distance between B and B'. We set up a Cartesian coordinate system as is shown in Fig. 3, where B and B' are at the origin $(0,0)$ and $(L,0)$, respectively. Let α and β, respectively, be the angles that AB and $A'B'$ make with the x-axis.

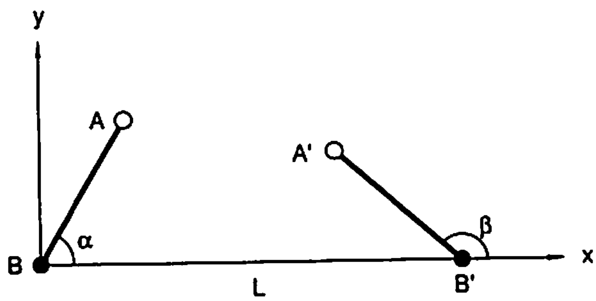

Fig. 3. The setup of the Cartesian coordinate system with reference to the initial and final positions of the robots.

Since the robots can move only at speed v or 0 at any moment, a lower bound on the time it takes to move AB to $A'B'$ is L/v. Intuitively, this lower bound is achievable if A can complete the necessary rotation around B within time L/v while B moves straight to B' at speed v. The following theorem characterizes the case.

Theorem 1. *AB can be moved to $A'B'$ in optimal time L/v if and only if either*

1. $\alpha = \beta$, *or*
2. $0° < \alpha < \beta < 180°$ *and* $\tan(\beta/2) \leq \tan(\alpha/2)e^{2L}$,

where L, α and β are as defined above.

Now we consider the case in which the lower bound L/v is not attainable. Let (x, y) be the coordinates of B during a motion, and ϕ the angle between the x-axis and the direction of the velocity of B. See Fig. 4.

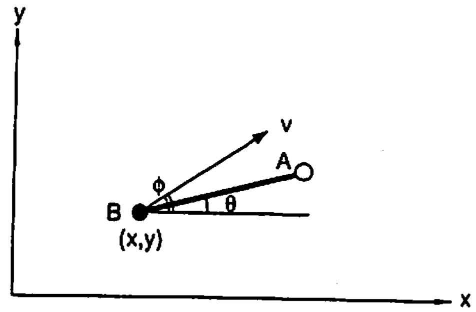

Fig. 4. The definitions of θ and ϕ. The arrow represents the velocity of robot B.

Since $|AB| = 1$, robot A rotates around robot B with angular speed $\omega = \dot{\theta}$, where the angle between the velocity of B and the velocity of A due to rotation around B is $90° + \theta - \phi$. Therefore V, the resultant speed of A, is given by

$$V^2 = v^2 + \omega^2 - 2\omega v \sin(\theta - \phi). \tag{1}$$

Since $V = v$ and $\omega = \dot{\theta}$, Eq. 1 can be rewritten as

$$\dot{\theta} - 2\dot{x}\sin\theta + 2\dot{y}\cos\theta = 0, \tag{2}$$

which describes the constraint on the optimal motion.

Now our task is to minimize the integral

$$F = \int_0^t \sqrt{\dot{x}^2 + \dot{y}^2}dt \tag{3}$$

subject to the constraint

$$\dot{\theta} - 2\dot{x}\sin\theta + 2\dot{y}\cos\theta = 0 \tag{4}$$

and the boundary conditions

$$\begin{cases} x(0) = 0 \\ y(0) = 0 \end{cases} \tag{5}$$

$$\begin{cases} x(t) = L \\ y(t) = 0 \end{cases} \tag{6}$$

By applying calculus of variations we can find the differential equations that an optimal trajectory obeys. In the next theorem $F(\phi, k)$ and $E(\phi, k)$ denote the Legendre elliptic integrals of the first and second kind, respectively, defined by

$$F(\phi, k) = \int_0^\phi \frac{d\theta}{\sqrt{1 - k^2 \sin^2\theta}} \tag{7}$$

$$E(\phi, k) = \int_0^{\phi} \sqrt{1 - k^2 \sin^2 \theta} d\theta. \quad (8)$$

Theorem 2. *Suppose that the lower bound L/v is not attainable. An optimal motion in which segment AB rotates about B counterclockwise can be obtained from*

$$\dot{x} = \frac{\dot{\theta}}{2} \sin\theta + cv \cos\theta \sin(\theta + \delta) \quad (9)$$

$$\dot{y} = -\frac{\dot{\theta}}{2} \cos\theta + cv \sin\theta \sin(\theta + \delta) \quad (10)$$

$$\dot{\theta} = 2v\sqrt{1 - c^2 \sin^2(\theta + \delta)}, \quad (11)$$

were constants c and δ are numerically calculated from

$$\frac{1}{2}(\cos\alpha - \cos\beta) + \frac{\cos\delta}{2c}\left[\sqrt{1 - c^2 \sin^2(\alpha + \delta)} - \sqrt{1 - c^2 \sin^2(\beta + \delta)}\right]$$
$$+\frac{\sin\delta}{2c}\left[F(\beta + \delta, c) - F(\alpha + \delta, c) - E(\beta + \delta, c) + E(\alpha + \delta, c)\right] = L \quad (12)$$

and

$$\frac{1}{2}(\sin\alpha - \sin\beta) - \frac{\sin\delta}{2c}\left[\sqrt{1 - c^2 \sin^2(\alpha + \delta)} - \sqrt{1 - c^2 \sin^2(\beta + \delta)}\right]$$
$$+\frac{\cos\delta}{2c}\left[F(\beta + \delta, c) - F(\alpha + \delta, c) - E(\beta + \delta, c) + E(\alpha + \delta, c)\right] = 0. \quad (13)$$

The time needed to execute the motion is obtained from

$$t = \frac{1}{2v}\left[F(\beta + \delta, c) - F(\alpha + \delta, c)\right]. \quad (14)$$

Figs. 2 and 2 show two optimal motions computed by the method.

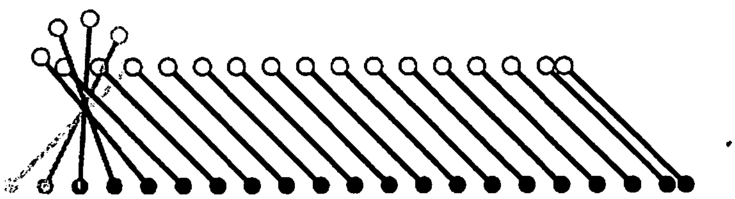

Fig. 5. Optimal motion in which the lower bound L/v is attained.

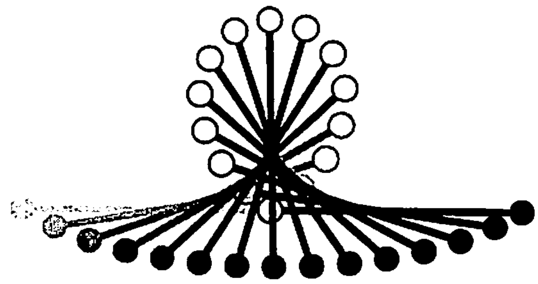

Fig. 6. Optimal motion in which the ladder turns around by nearly 180°.

3 Distributed Algorithms for Carrying a Ladder

In this section we describe two distributed algorithms for two omnidirectional robots carrying a ladder. They are designed based on the following idea, which is a restatement of the idea mentioned in Section 1.

> At any moment both robots intend to move toward their respective goal positions, and in case of a conflict they both yield equally by adjusting the directions of motion so that the distance between them is always equal to the length of the ladder.

Consider two robots A and B carring a ladder of length ℓ (see Fig. 3).[1] They both execute the same algorithm. Let A and B, and A' and B' be the start and goal positions, respectively, where $|\overline{AB}| = |\overline{A'B'}| = \ell$. Let $|\overline{AA'}| = L'$ and $|\overline{BB'}| = L$, where we assume without loss of generality that $L' \leq L$. Angles α and β are as defined in Fig. 3.

We view an algorithm for robot R as a mechanism that takes as input the current and goal positions of the ladder and produces as output a force vector $\mathbf{f}_R$ by which we attempt to drive R. Since R is connected to another robot by a rigid ladder, however, most likely it cannot move in the direction given by $\mathbf{f}_R$. Specifically, if R is about to move closer to (or away from) the other robot, then the ladder pushes (or pulls) R by some force $\mathbf{h}_R$. So in this paper we assume that R's actual motion is determined by force $\mathbf{f}_R + \mathbf{h}_R$. Note that the term $\mathbf{h}_R$ in $\mathbf{f}_R + \mathbf{h}_R$ effectively forces the robots to "yield equally" in case of a conflict. The assumption that an algorithm's output $\mathbf{f}_R$ does not depend on $\mathbf{h}_R$ somewhat simplifies the task of designing algorithms. (In contrast, physical robots are likely to have a force sensor for detecting the motion of the ladder, and the sensor output will be used explicitly in a feedback control scheme that drives the robot.)

Both algorithms ALG1 and ALG2 we present below are memoryless in the sense that their output is a function of the current input (and is independent of the motions in the past). It is therefore sufficient to view A and B of Fig. 3 as

[1] In Section 2, we assumed that $\ell = 1$.

the robots' current positions and specify the output for input A, B, A' and B'. As we will see shortly ALG2 is an extension of ALG1.

As we mentioned earlier, an algorithm's output is a force vector $\mathbf{f}_R$. In the following, however, we describe ALG1 and ALG2 in terms of *the target velocity vector* that the force they compute should allow the robot to achieve, if force $\mathbf{h}_R$ were not present.

Algorithm ALG1: Both A and B move toward their respective goals A' and B' at speeds v_A and v_B, respectively, where $v_A = v(L'/L)^s$ and $v_B = v$. Note that v is the robots' maximum speed and $s \geq 0$ is a parameter of ALG1.

Algorithm ALG2: Let $\mathbf{v}_A$ and $\mathbf{v}_B$ be the velocity vectors for robots A and B that ALG1 computes. ALG2 uses auxiliary velocity vectors $\mathbf{r}_A$ and $\mathbf{r}_B$ for robots A and B, respectively, that rotate the ladder couterclockwise. The direction of $\mathbf{r}_A$ (or $\mathbf{r}_B$) is $\alpha + \pi/2$ (or $\alpha - \pi/2$), i.e., perpendicular to the ladder $\overline{AB}$, and $||\mathbf{r}_A|| = ||\mathbf{r}_B|| = c(\beta - \alpha)$, where constant $c \geq 0$ is a parameter of ALG2. Then the velocity vectors that ALG2 computes for A and B are respectively $v(\mathbf{v}_A + \mathbf{r}_A)/a$ and $v(\mathbf{v}_B + \mathbf{r}_B)/a$, where $a = max\{||\mathbf{v}_A+\mathbf{r}_A||, ||\mathbf{v}_B+\mathbf{r}_B||\}$. Observe that ALG2 coincides with ALG1 when $c = 0$.

4 The Physical Robot System

For experiments we use a physical robot system involving two omnidirectional robots developed at RIKEN [3]. Fig. 7 shows the configuration of used in a preliminary experiment where the two robots are connected to a ladder via a force sensor placed on top through which each can sense the motion of the other. The connection via the force sensor is rather rigid, but it at least allows a human operator to hold one end of the ladder attached to a robot and gently push and pull the ladder so that the robot smoothly follows the motion of the ladder.

The results of the experiment using two robots in this configuration, however, has indicated that a more flexible link between the robots and the ladder are required. The relatively rigid connection between the robots and the ladder requires the robots to react to the motion of the other much more quickly than they actually can, resulting in an undesirable motion that is not smooth. A promising solution for this problem is to place a flexible multi-link mechanism that acts as a buffer between the ladder and the force sensors. We have designed such a mechanism consisting of springs and dumpers, and are now installing them.

5 Performance Evaluation by Simulation

5.1 Simulation Model

We conducted computer simulation to analyze the behavior of the robot system introduced in the last section executing the algorithms proposed in Section 3.

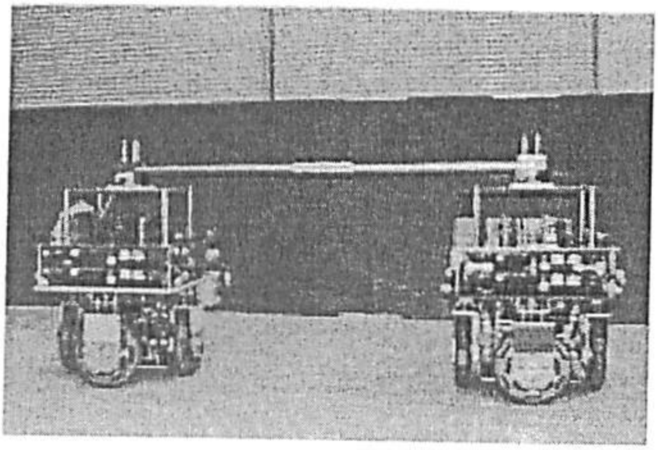

Fig. 7. Two RIKEN robots connected by a ladder. The ladder is fixed to each robot via a force sensor placed at the top.

We model the robots by a disk of radius 10 with a maximum speed of 1 per unit time. The length of ladder is 100. The multi-link mechanism is modeled by an ideal spring.

We use discrete time $0, 1, \ldots$ and denote by $\mathbf{p}_R(t)$ and $\mathbf{v}_R(t)$ the position of R and the target velocity vector of R that an algorithm specifies at time t, respectively. Also, we let $\mathbf{x}_R(t)$ denote the vector from the center of disk R to the position of the corresponding endpoint of the ladder at t. To simplify the simulation we assume:

1. Force $\mathbf{h}_R(t)$ that robot R receives from the ladder at time t is given by $k\mathbf{x}_R(t)$, where k is a spring constant.
2. The velocity of R at time t is $\mathbf{w}_R(t) = \mathbf{v}_R(t) + k\mathbf{x}_R(t)$. (We ignore the kinematic constraints of R and assume that R can attain any target velocity instantaneously.) If the length of $\mathbf{w}_A(t)$ or $\mathbf{w}_B(t)$ exceeds 1 (the robots' maximum speed), then we normalize them by dividing both by $\max\{||\mathbf{w}_A(t)||, ||\mathbf{w}_B(t)||\}$.
3. At time $t+1$, R is at position $\mathbf{p}_R(t+1) = \mathbf{p}_R(t) + \mathbf{w}_R(t)$.

The constant k is a parameter that controls the stiffness of the link between a robot and the ladder and therefore the amount of freedom of a robot's motion. When $k = 0$, for example, the robots can move freely irrespective of the position of the ladder. In this paper we used $k = 10.0$ which we found to be sufficiently large to keep the endpoints of the ladder within the disks of radius 10 representing the two robots (i.e., $||\mathbf{x}_R(t)|| \leq 10$).

5.2 Algorithm ALG1

To investigate the effect of parameter s on the performance of Algorithm ALG1, we examine the number of steps N necessary for the robots to reach their goals using the setup of Fig. 3 for $L = 200, 400, \ldots, 1600$, $0° < \alpha \leq 90°$ and $s = 0.0, 0.5, \ldots, 4.0$. To reduce the number of instances to examine, however, we consider only those cases in which $\alpha = 180° - \beta$ and (by symmetry) $\alpha \leq 90°$. Note that $N \geq L$ holds since the robots' maximum speed is 1.

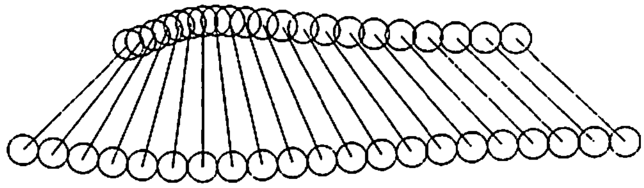

Fig. 8. The motion of the robots obeying ALG1 for $s = 3.0, L = 400$ and $\alpha = 45°$.

Fig. 8 shows a motion obtained by ALG1 for $s = 3.0, L = 400$ and $\alpha = 45°$.

Note that the robot further from its destination moves nearly straight, as in an optimal motion shown in Fig. 2 for the same instance. In fact, ALG1 is expected to perform well when L and α are large since the robots need not rotate the ladder too quickly, and our simulation results confirm this. We present only the results for $L = 200$ in Fig. 9, where we observe the following:

1. The performance of ALG1 is not very sensitive to the value of s. Setting $s = 3$ seems to work particularly well for most cases.
2. ALG1 shows sufficiently good performance for $\alpha \geq 20°$. For $\alpha \geq 65°$ it always achieves the lower bound of $N = L = 200$.
3. The performance of ALG1 is noticeably worse when $\alpha \leq 10°$ and rapidly degrades as α approaches 0.

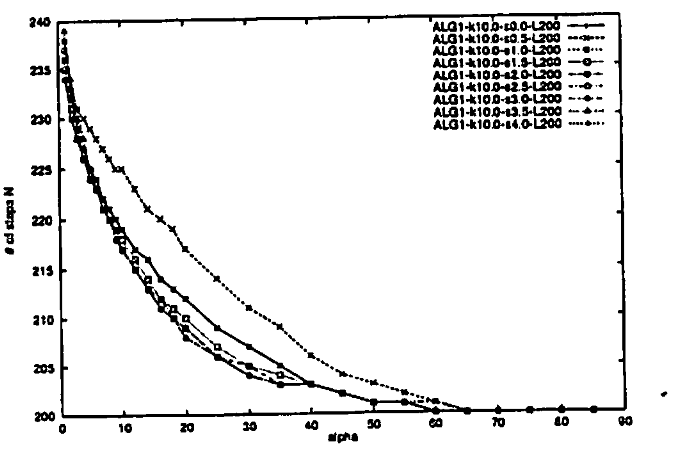

Fig. 9. The number N of steps needed by ALG1 for $L = 200$ and various values of α.

Comparing the motion generated by ALG1 for $s = 3.0, L = 200$ and $\alpha = 1°$ shown in Fig. 10 with an optimal motion shown in Fig. 2 computed by the method of Section 2, we notice that in the former the ladder starts to rotate only toward the end of the motion, while in the latter rotation starts as soon as the ladder starts moving. This observation has motivated us to introduce in ALG2 auxiliary velocity vectors $\mathbf{r}_A$ and $\mathbf{r}_B$ that help rotate the ladder during motion.

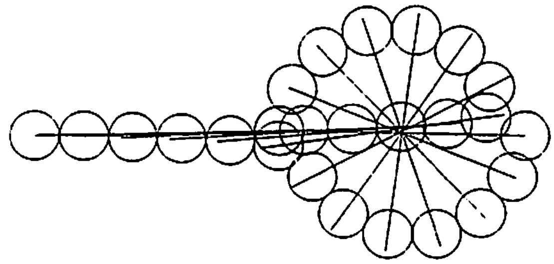

Fig. 10. The motion generated by ALG1 for $s = 3.0, L = 200$ and $\alpha = 1°$.

5.3 Algorithm ALG2

Recall that ALG2 has a parameter c and reduces to ALG1 when $c = 0$. Using the same set of instances as in Subsection 5.2 and for various values of c, we again measure the number N of steps necessary for the robots to reach the goal. Throughout this section we use $s = 3.0$ which was found to work well in ALG1.

Fig. 11 shows the results for $L = 200$ and $0° < \alpha \leq 90°$, for several values of c. We observe the following:

1. ALG1, i.e., ALG2 with $c = 0$, performs better than ALG2 with $c > 0$ for all $\alpha \geq 20°$.
2. When $\alpha < 20°$ ALG2 with $c = 0.5$ shows the best performance.

Figs. 12 and 13 show the motions generated by ALG2 with $c = 0.5$ for the instances examined for ALG1 in Figs. 8 and 10, respectively.

Again, in Fig. 12 the robot further from its destination moves nearly straight, as in an optimal motion shown in Fig. 2 for the same instance. The motion shown in Fig. 13 is fairly similar to an optimal motion shown in Fig. 2 and takes much less time than the motion in Fig. 10 generated by ALG1.

Based on the results shown in Fig. 11 we conclude that small effort to rotate the ladder is sufficient to avoid undesired motion such as the one shown in Fig. 10. The results also indicate that larger values of c tend to increase N. We verified this through additional simulation, and present only the results for $L \geq 400$ in

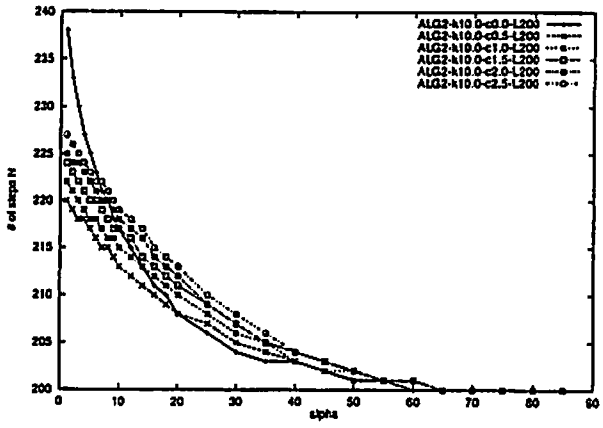

Fig. 11. The number N of steps needed by ALG2 for $L = 200$ and various values of α.

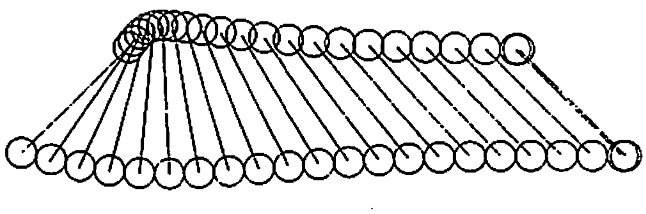

Fig. 12. The motion generated by ALG2 with $c = 0.5$, for $L = 400$ and $\alpha = 45°$.

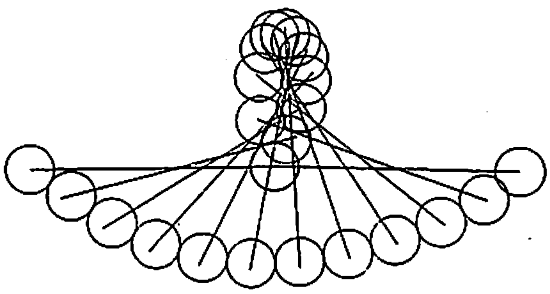

Fig. 13. The motion generated by ALG2 with $c = 0.5$, for $L = 200$ and $\alpha = 1°$.

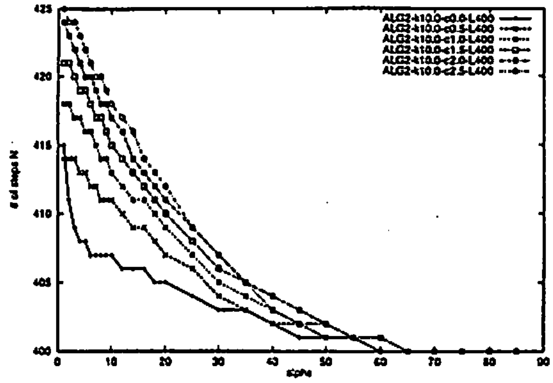

Fig. 14. The number N of steps needed by ALG2 for $L = 400$ and various values of α.

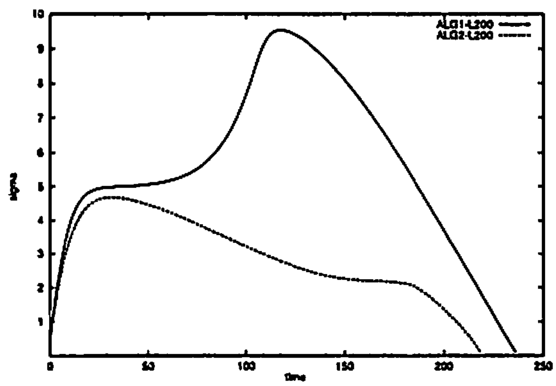

Fig. 15. The change of σ during the motions of Figs. 10 and 13 for $L = 200$ and $\alpha = 1°$.

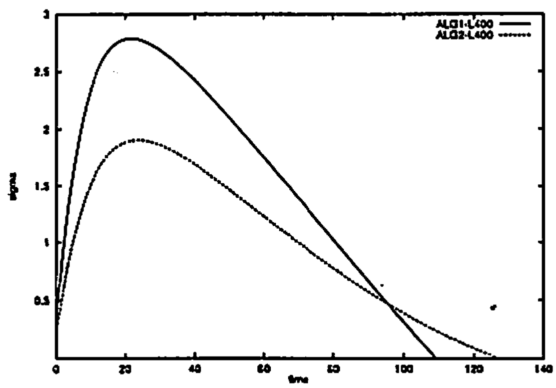

Fig. 16. The change of σ during the motions of Figs. 8 and 12 for $L = 400$ and $\alpha = 45°$.

Fig. 14, which shows that ALG1 performs in general better than ALG2 with $c > 0$.

Finally, we compare ALG1 and ALG2 with $c = 0.5$ from a different point of view. The measure is $||\mathbf{x}_R(t)||$, i.e., the distance between the center of robot R and the corresponding endpoint of the ladder at time t. If an algorithm produces a motion with small $||\mathbf{x}_R(t)||$, then it might be said that the actions of the two robots are coordinated well, and consequently, physical robots using the algorithm can be expected to move smoothly. Figs. 15 and 16 show the ratio $\sigma = 100 \times (||\mathbf{x}_R(t)||/\ell)(\%)$ taking time as the parameter, where $\ell = 100$ is the length of ladder. Fig. 15 shows the case $L = 200$ and $\alpha = 1°$ (motions in Figs. 10 and 13), and Fig. 16 the case $L = 400$ and $\alpha = 45°$ (motions in Figs. 8 and 12). Clearly ALG2 performs better than ALG1 in both cases.

6 Conclusions

Time-optimal motion of two omnidirectional robots carrying a ladder can be computed using optimal control theory in a off-line and centralized manner. This paper has demonstrated that even without the complicated calculation of such an approach, a motion that is sufficiently close to time-optimal can be obtained using a simple distributed algorithm in which each robot decides its motion individually based on the current and goal positions of the ladder.

We presented two such algorithms, ALG1 and ALG2, and using computer simulation demonstrated that the former performs sufficiently well for large L and α (two of the parameters describing a given instance), while for small L and α the latter algorithm performs better. We also observed that the smoothness of motion is another advantage of ALG2.

This paper has reported only the preliminary results of the authors' ongoing project on multi-robot coordination. Issues to be investigated in the future include the following:

1. Implementing ALG1 and ALG2 on the RIKEN robots.
2. Coordination of robots having different capabilities.
3. Obstacle avoidance while carrying a ladder.
4. Extension to the problem of carrying an n-gon.

References

1. M. N. Ahmadabadi and E. Nakano, "Constrain and move: a new concept to develop distributed transferring protocols," *Proc. IEEE Int. Conf. on Robotics and Automation*, 2318–2325 (1997).
2. R. Alami, S. Fleury, M. Herrb, F. Ingrand, and S.Qutub, "Operating a large fleet of mobile robots using the plan-merging paradigm," *Proc. IEEE Int. Conf. on Robotics and Automation*, 2312–2317 (1997).
3. H. Asama, M. Sato, L. Bogoni, H. Kaetsu, A. Matsumoto, and I. Endo, "Development of an Omni-directional mobile robot with 3 DOF decoupling drive mechanism," *Proc. IEEE Int. Conf. on Robotics and Automation*, 1925–1930 (1995).

4. Z, Chen, I. Suzuki, and M. Yamashita, "Time optimal motion of two robots carrying a ladder under a velocity constraint," *IEEE Trans. Robotics and Automation 13*, 5, 721–729 (1997).
5. R. Courant and D. Hilbert, *Methods of Mathematical Physics*, Interscience Publishers, New York, 1953.
6. Proceedings of the International Symposium on Distributed Autonomous Robot Systems (DARS) in 1992, 1994, 1996, and 1998.
7. B.R. Donald, "Information Invariants in Robotics: Part I – State, Communication, and Side-Effects," *Proc. IEEE Int. Conf. on Robotics and Automation*, 276–283 (1993).
8. K. Kosuge and T. Oosumi, "Decentralized control of multiple robots handling and objects," *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 556-561 (1996).
9. J-C. Latombe, "Robot Motion Planning," Kluwer Academic Publishers, Boston, 1991.
10. M. Mediav̄illa, J. C. Fraile, J. R. Perán, and G. I. Dodds, "Optimization of collision free trajectories in multi-robot systems," *Proc. IEEE Int. Conf. on Robotics and Automation*, 2910–2915 (1998).
11. N. Miyata, J. Ota, Y. Aiyama, J. Sasaki, and T. Arai, "Cooperative transport system with regrasping car-like mobile robots," *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1754–1761 (1997).
12. Y. Nakamura, K. Nagai, and T. Yoshikawa, "Dynamics and stability in coordination of multiple robotics mechanisms," *The Int. J. of Robotics Research*, 8,2, 44–60 (1989).
13. D. J. Stilwell and J. S. Bay, "Toward the development of a material transport system using swarms of ant-like robots," *Proc. IEEE Int. Conf. on Robotics and Automation*, 766–771 (1995).
14. I. Suzuki and M. Yamashita: "A Theory of distributed anonymous mobile robots –formation and agreement problems," *SIAM J. Computing* (to appear).
15. L.L. Whitcomb, D.E. Koditschek, and J.B.D. Cabrera, "Toward the Automatic Control of Robot Assembly Tasks via Potential Functions: The Case of 2-D Sphere Assemblies," *Proc. IEEE Int. Conf. on Robotics and Automation*, 2186–2191 (1992).
16. H. Yamaguchi, "Adaptive formation control for distributed autonomous mobile robot groups," *Proc. IEEE Int. Conf. on Robotics and Automation*, 2300–2305 (1997).
17. H. Yamaguchi, "A cooperative hunting behavior by mobile robot troops," *Proc. IEEE Int. Conf. on Robotics and Automation*, 3204–3209 (1998).

Lecture Notes in Artificial Intelligence 1724
Subseries of Lecture Notes in Computer Science

Henrik I. Christensen Horst Bunke
Hartmut Noltemeier (Eds.)

Sensor Based Intelligent Robots

International Workshop
Dagstuhl Castle, Germany, September/October 1998
Selected Papers

Springer