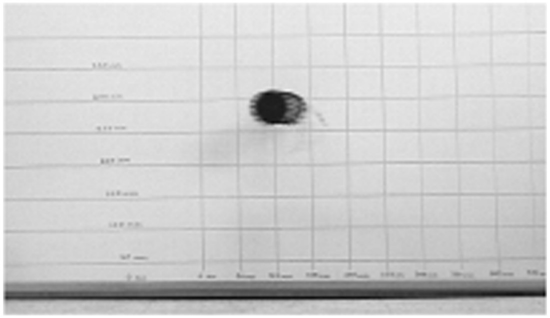

Fig. 14. Reached peak for a jumping Scout with a weak spring.

The peak of the jump occurs at the time of zero vertical speed, therefore $t_{\rm peak} = (v_0 \sin\lambda/g)$. Using (8), the coordinates of the peak become

$$\mathbf{p}_{v_0,\lambda}(t_{\rm peak}) = \begin{pmatrix} 0 \\ \frac{v_0^2 \sin\lambda \cos\lambda}{g} \\ \frac{(v_0 \sin\lambda)^2}{2g} \end{pmatrix}. \quad (10)$$

As the origin is set to the Scout's center, the radius of the wheels must be subtracted from this number to ensure a successful leap over the obstacle. This assumes that the foot will not interfere.

The launch angle can be selected at runtime, but is typically set to about 60°. The governing factor in determining the launch speed is the strength of the spring foot, and it can be expressed as a function of the peak height $z_{\rm peak}$ yielding $v_0 = \sqrt{2gz_{\rm peak}}/\sin\lambda$. Jump heights were extracted from video sequences of jumping Scouts, as shown in Fig. 14. The measured height of 57 cm for a strong spring corresponds to a launch speed of 3.6 m/s. The weak spring gave 30 cm and 2.6 m/s.

IV. Conclusion

Models for the various modi of locomotion for a Scout robot were developed. They are governed primarily by the position and shape of the foot which, in turn, depends on the length of the winch cable.

The models evince that despite its small size, a Scout is capable of a surprisingly wide range of motions, including jumping over obstacles several times its own height.

Acknowledgment

The authors thank the anonymous reviewers for their thoughtful comments.

References

[1] M. H. Raibert, H. B. Brown, Jr., and M. Chepponis, "Experiments in balance with a 3D one-legged hopping machine," *Int. J. Robot. Res.*, vol. 3, no. 2, pp. 75–92, 1984.

[2] P. Fiorini and J. Burdick, "The development of hopping capabilities for small robots," *Auton. Robots*, vol. 14, no. 2–3, pp. 239–254, Mar./May 2003.

[3] Sandia National Lab., 2000, Hop to it: Sandia Hoppers Leapfrog Conventional Wisdom About Robot Mobility [Online]. Available: http://www.sandia.gov/LabNews/LN10-20-00/hop_story.html

[4] P. E. Rybski, N. P. Papanikolopoulos, S. A. Stoeter, D. G. Krantz, K. B. Yesin, M. Gini, R. Voyles, D. F. Hougen, B. Nelson, and M. D. Erickson, "Enlisting Rangers and Scouts for reconnaissance and surveillance," *IEEE Robot. Autom. Mag.*, vol. 7, no. 4, pp. 14–24, Dec. 2000.

[5] MTS Systems Corp.. Eden Prairie, MN, DARPA Scout Assy Jun. 2000.

[6] L. D. Landau and E. M. Lifshitz, *Theory of Elasticity*, ser. Course on Theoretical Physics, 2nd English ed. New York: Pergamon, 1970, vol. 7.

[7] S. P. Timoshenko and J. M. Gere, *Theory of Elastic Stability*, 2nd ed. New York: McGraw-Hill, 1961.

[8] G. Dudek and M. Jenkin, *Computational Principles of Mobile Robots*. Cambridge, U.K.: Cambridge Univ. Press, 2000.

Cooperative Hunting by Distributed Mobile Robots Based on Local Interaction

Zhiqiang Cao, Min Tan, Lei Li, Nong Gu, and Shuo Wang

***Abstract*—This paper proposes a distributed control approach called *local interactions with local coordinate systems* (LILCS) to multirobot hunting tasks in unknown environments, where a team of mobile robots hunts a target called evader, which will actively try to escape with a safety *strategy*. This robust approach can cope with accumulative errors of wheels and imperfect communication networks. Computer simulations show the validity of the proposed approach.**

***Index Terms*—Distributed control, hunting, local interaction, mobile robots.**

I. Introduction

Multiple mobile robot systems have been extensively studied [1]. As pointed out in [2], multiple robots executing a given task have to face some challenges: accumulative errors due to wheels slippage; imperfect communication networks (e.g., packet loss, transmission delay); etc. Therefore, an elaborate design for a multirobot system should be required. In this paper, we address specific issues related to the problem of multirobot hunting.

The hunting task, conducted by mobile robots and its target evader, is a particular challenge due to the nature of the unknown irregular motion of the evader. Its achievement is significant, especially in the military field. Yamaguchi has proposed a feedback-control law for coordinating the motion of multiple robots to capture/enclose a target by making a troop formation [3]. The linear autonomous system method has been used to generate the shape of a multirobot group to capture a target [4]. In [5], multiple objective behavior coordination is used to provide mechanisms for distributed command fusion across a group of robots to complete the task of cooperative target acquisition. In other related works, a pursuit game discusses two kinds of agents: predator (hunter or pursuer) and prey (evader or invader) [6]–[14]. It has been a testbed of multiagent cooperation. In [8] and [9], reinforcement learning has

Manuscript received February 1, 2005. This paper was recommended for publication by Associate Editor F. Y. Wang and Editor S. Hutchinson upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grant 50475179, and in part by the National Basic Research Program of China under Grant 2002CB312200.

Z. Cao, M. Tan, and S. Wang are with the Laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences, Beijing 100080, China (e-mail: zqcao@compsys.ia.ac.cn; tan@compsys.ia.ac.cn; swang@compsys.ia.ac.cn).

L. Li is with the Institute of Software, Chinese Academy of Sciences, Beijing 100080, China (e-mail: l.li@ia.ac.cn).

N. Gu is with the School of Engineering and Technology, Deakin University, Geelong VIC 3217, Australia (e-mail: ng@deakin.edu.au).

Digital Object Identifier 10.1109/TRO.2006.862495

been used. The pursuit game and map-building problems have been combined in a probabilistic framework in [10]. In [11]–[14], the case of supervisory agents (such as a helicopter) that can estimate the evader's position but not capture it has been investigated.

In this paper, multiple mobile robots with limited sensing capabilities are required to hunt a moving target evader with certain intelligence. The robots do not have a global coordinate system, and the influence of wheel slippage is eliminated. Also, no explicit communication is employed. Taking the evader as the bond to connect the robots, a distributed approach called *local interactions with local coordinate systems* (LILCS) is proposed. The cooperation may emerge by local interactions among the robots.

The paper is organized as follows. Section II gives the proposed LILCS approach in detail. Section III mainly presents an active escaping strategy for the evader. Computer simulations are conducted in Section IV, and Section V concludes the paper.

II. LILCS Approach

In this paper, the robots are labeled as R_i $(i = 1, 2, \ldots, N)$. Each robot is considered an omnidirectional mobile robot with 360° visual capability, and the robot has only its local coordinate system (LCS). In order to perceive the environment for collision avoidance, a range sensor model S_{range} is adopted. The robots may recognize each other, and each robot can identify the evader. Each robot may acquire the positions of the evader and the robots near it. In addition, we assume that the visual ranges of the robots and the evader are the same as that of S_{range}. Compared with the robot's radius, its maximum sensing range is much greater.

In order to achieve the hunting task in unknown environments with some robots having only LCSs, a flexible LILCS approach is proposed, and the central idea is as follows: each robot makes its own decisions based on its ambient circumstance, and the cooperation may emerge through local interactions among the robots, which is beneficial to the task. For a robot R_i, when it does not know where the evader is, it is in *search* state and should find the evader first. Once R_i sees the evader, its pursuit begins, and the robot will take actions based on the detected evader and its neighboring robots. If the condition to catch the evader is not satisfied, the robot is in *pursue* state and keeps pursuing the evader, or else, the robot thinks that it is the time to catch the evader. This is termed as the robot being in *catch* state. Considering R_i may lose track of the evader, it should be capable of predicting the evader, and the robot is in *predict* state. Once R_i thinks the prediction is useless, it has to re-search the environment. The process is repeated until the task is completed.

In the following, the detailed designs for the four distinct states mentioned above are given respectively.

A. Search State

Each robot in this state endeavors to find the evader by randomly turning left or right at an angle τ, which is within $(0, (\pi/4)]$ in order to make the robot explore the environment more efficiently.

B. Pursue State

Once the robot R_i sees the evader, its pursuit begins. R_i establishes a local polar coordinate system $\sum_i$, whose pole and polar axis direction are its center and heading, respectively. Based on $\sum_i$, R_i acquires the coordinates of the evader and the robots in Γ_i, which is the set of R_i's neighboring robots who also see the evader from the viewpoint of R_i. According to the distance to the evader, three zones $(0, x_{\text{near}}]$, $(x_{\text{near}}, x_{\text{far}}]$, and $(\chi_{\text{far}}, +\infty)$ are generated, where χ_{near} and $\chi_{\text{far}} (\chi_{\text{far}} > \chi_{\text{near}})$ both are parameters determined by R_i's maximum sensing range $S_{i\_\max}$ and its radius r_i. We define $\chi_{\text{far}} = S_{i\_\max} - k_1 r_i$ and $\chi_{\text{near}} = S_{i\_\max} - k_2 r_i$, where k_1 and k_2 are chosen to make $\chi_{\text{far}} \in ((S_{i\_\max}/2), (2S_{i\_\max}/3))$, $\chi_{\text{near}} \in ((S_{i\_\max}/3), (S_{i\_\max}/2))$. When the distance from R_i to the evader is not greater than χ_{near}, it is for R_i to judge whether it is time to catch the evader. The judging process is described as follows.

Step 1) We denote with $N_{\Gamma i}$ the number of elements in Γ_i and $N_{\Gamma i} \geq 0$. If $N_{\Gamma i} \leq 1, R_i$ thinks that it has to catch the evader, or else it calculates the number $n_{\Pi i}$ of elements in Π_i, which is the set of R_i and the robots whose distances to the evader are within $(0, x_{\text{near}}]$ in Γ_i.

Step 2) If $n_{\Pi i} > 2$, R_i randomly chooses one from Π_i, which is denoted R^i_{base}. When $n_{\Pi i}$ is an even number, R_i obtains the robot R^i_{near} that makes the angles between the ray from the evader to R^i_{base} and the rays from the evader to other robots in Π_i minimal. Then R_i establishes a pole coordinate system $\sum^{\text{eveni}}_{\text{base}}$, whose pole and polar axis direction are the evader's center and the interior bisector direction of the angle between the ray from the evader to R^i_{base} and the ray from the evader to R^i_{near}, respectively. If $n_{\Pi i}$ is an odd number, R_i establishes a polar coordinate system $\sum^{\text{oddi}}_{\text{base}}$, whose pole is the evader's center with the polar axis direction of the ray from the evader to R^i_{base}.

Step 3) Calculate the coordinates of all robots in Π_i in $\sum^{\text{eveni}}_{\text{base}}$ (or $\sum^{\text{oddi}}_{\text{base}}$). All polar angles are within $[0, 2\pi)$, and they consist of a set Λ_{base}.

Step 4) When there exists a robot R^i_{base} such that one element in Λ_{base} is within $[(\pi/2), \pi)$, and another one is within $[\pi, (3\pi/2))$, R_i thinks that it may catch the evader.

Based on the above judging process, when R_i thinks that the condition to catch the evader is not satisfied, it is in *pursue* state and has to pursue the evader first. The local decision making is as follows.

Step 1) If the set Γ_i is empty, R_i makes its decision without considering other robots (go to Step 4), or else it obtains the robot $R_i^{\min}$, which makes the angles formed by R_i, evader, and the robots in Γ_i minimal, and the minimal angle is expressed by $\varphi_{\min}$.

Step 2) If $\varphi_{\min}$ is less than ς, where $\varsigma = (2\pi/3)(N_{\Gamma i} \leq 2)$ or $\varsigma = (2\pi/(N_{\Gamma i} + 1))(N_{\Gamma i} > 2)$, R_i and $R_i^{\min}$ should cooperate with each other (go to Step 3), or else, R_i may take action based on its own intention (go to Step 4).

Step 3) If the robot R_i considers $R_i^{\min}$, it should move toward an ideal position, which is located in the ray from the evader to $R_i^{\min}$ with a rotation of angle ς, and whose distance to the evader d_{i-e} depends on d_{c-e}, which describes how far R_i is away from the evader. d_{i-e} is χ_{far} when $d_{c-e} > \chi_{\text{far}}$, and $\chi_{\text{near}} - 2\Delta$ (Δ is a margin) in other cases.

Step 4) When no other robots are considered, R_i will move toward the position which is located in the ray from the evader to R_i, and whose distance to the evader is d_{i-e}.

C. Catch State

When the robot R_i thinks that it may catch the evader, if $N_{\Gamma i} \leq 1$, two cases are considered. When no other robot is considered, R_i hopes to move toward the evader directly, or else it will consider another robot, and the distance between its ideal position and the evader is $r_c \cdot r_c$ is defined as $k_3 r_i$, where k_3 is chosen to make $r_c < \chi_{\text{near}} - 2\Delta$. When $n_{\Pi i} > 2$, R_i will move toward the evader, and the distance from its ideal position to the evader is r_c. In this case, the decision making will last within certain steps $(\text{step}_{\text{catch}})$ without examining the condition to catch the evader.

D. Predict State

For R_i chasing the evader, if suddenly the evader is invisible, it should predict the evader within certain steps $(\text{step}_{\text{predict}})$, based on its motion information and the recorded position of the evader in its previous LCS $\sum_{\rm pre}^{i}$. If the evader escapes along the direction away from R_i and moves in R_i's maximum step size, the suppositional escape position of the evader in $\sum_{\rm pre}^{i}$ can be calculated. Then R_i obtains the suppositional coordinates of the evader in $\sum_i$, which determines the expected direction of R_i.

E. Motion Strategy

The decisions of all four states mentioned above are made regardless of the obstacles in the environment. In order to ensure each robot moves safely, a motion strategy is proposed to combine its ideal motion with readings from sensors.

The robot adopts $S_{\rm range}$ to perceive the world. The nine sensors are arranged evenly, and the detecting zone of each sensor is considered a sector. These sensors are labeled S_t $(t = 0, 1, \ldots, 8)$, starting anticlockwise from the reverse direction of the robot's heading. The robot may know the presence or absence of other objects in each sector zone, as well as the nearest distance to them. For the robot R_i, we denote with $P_{\rm rr}(\rho_r, \theta_r)$ the coordinates of its ideal position in Σ_i. The coordinates of the detecting border of each sensor in Σ_i are described as $P_s^t(\rho_t, \theta_t)$, where ρ_t is the maximum sensing range when no obstacle is detected. Otherwise, reading from S_t after the evader is considered, and $\theta_t \in [-\pi + (2\pi/9)t, -\pi + (2\pi/9)(t+1)]$. We denote with $P_a(\rho_a, \theta)$ the coordinates of R_i's next position, where ρ_a is the step size determined by ρ_r and the maximum step size, and θ is the angle rotated by R_i. The goal is to seek θ within $[-\zeta_{r\max}, \zeta_{r\max}]$ of the robot's heading with predetermined ρ_a, such that the robot moves in a collision-free direction having the least angle with the ideal direction.

On the basis of sensory information, the distances from P_a to the detecting border of each sensor $P_a P_s^t$ $(t = 0, 1, \ldots, 8)$ should be greater than or equal to a safety distance $D_{\rm safe}$ determined by the robot's velocity and its radius, namely

$$P_a P_s^t \geq D_{\rm safe}(t = 0, 1, \ldots, 8). \tag{1}$$

The final value of θ should satisfy (1) and make $|\theta - \theta_r|$ a minimum. Considering the tth sensor, we have

$$\sqrt{(\rho_a \cos\theta(t) - \rho_t \cos\theta_t)^2 + (\rho_a \sin\theta(t) - \rho_t \sin\theta_t)^2} \geq D_{\rm safe} \tag{2}$$

where $\theta(t)$ are the values of θ satisfying the condition of the tth sensor in (1). From (2), we get the following.

When $|\rho_a - \rho_t| \geq D_{\rm safe}$ is satisfied, $\theta(t) \in [-\zeta_{r\max}, \zeta_{r\max}]$.

When $\rho_a + \rho_t < D_{\rm safe}$ is satisfied, $\theta(t) \in \Phi$, the empty set.

When $\rho_a + \rho_t \geq D_{\rm safe} \cap |\rho_a - \rho_t| < D_{\rm safe}$ is satisfied

$$\theta(t) - \theta_t \in [-2\pi + \arccos N_v, -\arccos N_v] \cup [\arccos N_v, 2\pi - \arccos N_v] \tag{3}$$

where $N_v = ((\rho_a^2 + \rho_t^2 - D_{\rm safe}^2)/2\rho_a\rho_t)$.

Any value within the range of θ_t should be suitable for (3), therefore

$$\begin{aligned}\theta(t) \in & \left\{ \left[\arccos N_v - \frac{25}{9}\pi + \frac{2\pi}{9}t, -\arccos N_v - \pi + \frac{2\pi}{9}t \right] \right. \\ & \left. \cup \left[\arccos N_v - \frac{7}{9}\pi + \frac{2\pi}{9}t, -\arccos N_v + \pi + \frac{2\pi}{9}t \right] \right\} \\ & \cap [-\zeta_{r\max}, \zeta_{r\max}]\end{aligned} \tag{4}$$

when $\arccos N_v \leq (8\pi/9)$ is satisfied, and $\theta(t) \in \Phi$, when $\arccos N_v > (8\pi/9)$ is satisfied.

The set of θ satisfying (1) is defined as Ω, which is the intersection of $\theta(t)$ $(t = 0, 1, \ldots, 8)$. When Ω is not empty, the most preferred value of θ can be obtained to make $|\theta - \theta_r|$ a minimum, or else, the proper θ cannot be found. In this case, the robot will only turn at a right angle $\zeta_{r\max}$.

III. Strategies for the Evader

Assume that the evader adopts the same sensor model as that of an individual robot. When the evader does not see any robot or static obstacle, it moves randomly; otherwise, it should move safely, and thus a safety strategy is required.

The evader establishes a local polar coordinate system Σ_e, whose pole is its center and polar axis direction is its heading. We denote with $P_e^i(\rho_i, \theta_i)$ the coordinates of the detecting border of sensors S_e^i $(i = 0, 1, \ldots, 8)$ in Σ_e, where ρ_i is reading from S_e^i when it senses any object, or else the sensor is ignored, and ρ_i is chosen at a far greater value than the maximum sensing range of the evader for calculation, and $\theta_i \in [-\pi + (2\pi/9)i, -\pi + (2\pi/9)(i+1)]$. Based on the evader's heading, Q (a multiple of four) directions are generated, and their set $\Im$ is depicted as follows:

$$\Im = \left\{ \zeta_q | \zeta_q = -\pi + \frac{2q\pi}{Q}(q = 0, 1, \ldots, Q-1) \right\}. \tag{5}$$

The evader may move to the position $P_n^q(V_e, \zeta_q)$ with predetermined step size V_e without collisions, when the distances from this position to the detecting border of each sensor are greater than or equal to a safety distance $L_{\rm safe}$, which is determined by the evader's velocity and radius; that is

$$d_i(\zeta_q) = \min\left(P_n^q P_e^i\right) \geq L_{\rm safe}(i = 0, 1, \ldots, 8). \tag{6}$$

When $\exists \zeta_q \in \Im$ satisfying (6), the evader is still capable of moving, otherwise, the evader cannot find a feasible direction, which shows that it has been captured.

We label $\Psi = \{\zeta_q | \zeta_q = -\pi + (2q\pi/Q)(q = (Q/4), (Q/4) + 1, \ldots, (3Q/4) - 1)\}$ as the set of directions within $[-(\pi/2), (\pi/2))$ of the evader's heading. When it may still move, the safety strategy is used to select the best one from all ζ_q satisfying (6) in Ψ, and the best value ζ should make ${\rm dis}(\zeta_q)$ a maximum, thus

$$\begin{aligned}{\rm dis}(\zeta) &= \max_{\zeta_q} {\rm dis}(\zeta_q) \\ &= \max_{\zeta_q} \min\left(d_0(\zeta_q), d_1(\zeta_q), \ldots, d_8(\zeta_q)\right).\end{aligned} \tag{7}$$

If ζ is found, the evader will rotate ζ with V_e, or else, it turns a right angle $\zeta_{e\max}$ without any change in position.

IV. Simulations

A team of robots of ID 1,2,... adopting the LILCS approach is used to hunt an evader T, which is regarded as a special round robot. They have the same physical parameters: the radius, maximum step size, and maximum sensing range are 0.2, 0.1, and 3.0, respectively. The parameters in the LILCS approach and the evader's safety strategy are shown in Table I.

Simulation 1 adopts four robots executing the task in an environment shown in Fig. 1. Fig. 2 describes the variations of each robot m_state. When m_state is 1, 2, 3, or 4, the robot is in *search*, *pursue*, *catch*, or *predict* states, respectively. Initially, only the robot of ID 4 does not see the evader. At time step $ts1$, the robot of ID 3 loses track of the

TABLE I
VALUES OF SOME PARAMETERS

Parameter	Value	Parameter	Value
τ	$\frac{\pi}{18}$	k_1	6.5
k_2	8.5	Δ	0.15
k_3	2.5	$step_{catch}$	15
$step_{predict}$	15	D_{safe}	0.3
$\zeta_{r\max}$	$\frac{\pi}{2}$	L_{safe}	0.39
Q	72	$\zeta_{e\max}$	$\frac{\pi}{2}$

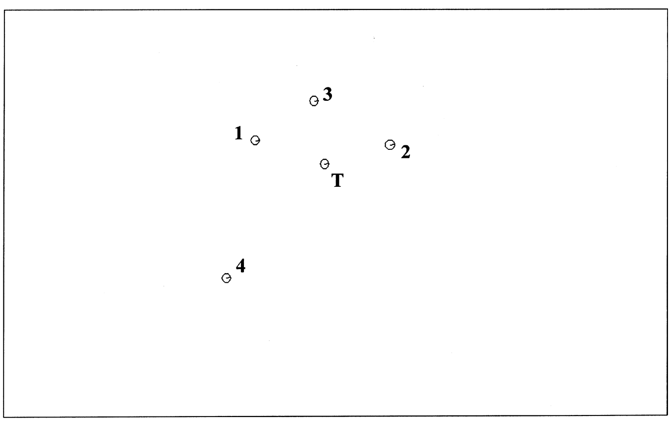

Fig. 1. Initial environment of Simulation 1.

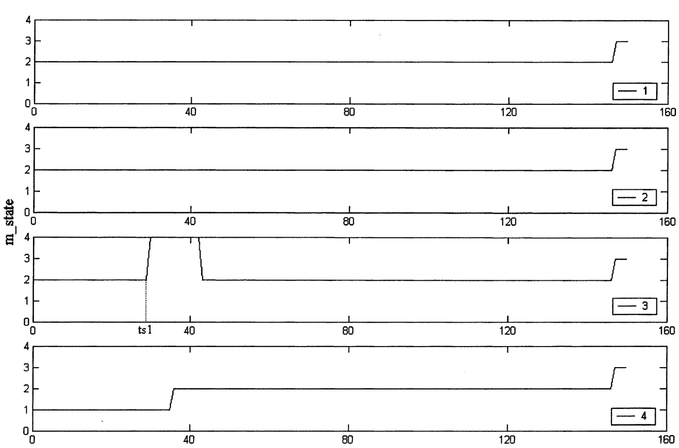

Fig. 2. State variations of each robot in Simulation 1.

evader. By prediction, it finds the evader again. At this moment, all robots pursue the evader. Ultimately, the evader is captured.

Simulation 2 is used to illustrate the robustness of the LILCS approach. The motion trajectories of all robots and the evader are shown in Fig. 3, where S_1, S_2, S_3, S_T are initial positions of the robots of ID 1, 2, 3, and the evader, respectively. After the robot of ID 2 becomes dysfunctional and stops motion in P_1, the robot of ID 1 has to chase the evader by itself until the robot of ID 3 also finds the evader. By the efforts of these two robots, the evader is captured.

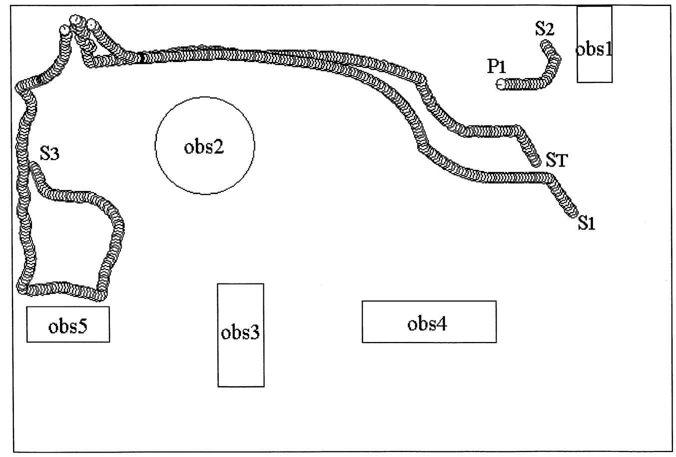

Fig. 3. Trajectories of the robots and the evader in Simulation 2.

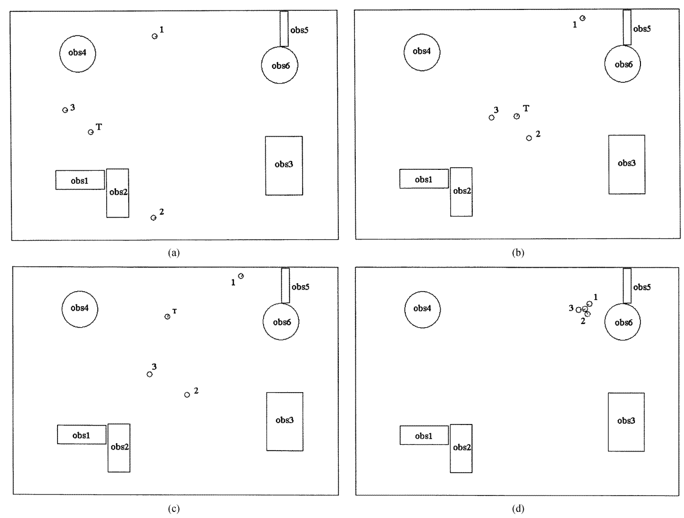

Fig. 4. Selected images for the hunting process of Simulation 3.

Simulation 3 is conducted to test the robotic system's adaptability. Fig. 4 gives several selected images, and the initial environment is shown in Fig. (4a). Fig. 5 depicts the variations of each robot m_state. From Figs. 4 and 5, the hunting process may be described as follows. At the beginning, only the robot of ID 3 sees the evader. After a period of time, the robot of ID 2 also finds the evader. The cooperation between these two robots arises. When the robots arrive in the positions shown in Fig. (4b), the evader T is transferred manually to the position described in Fig. (4c). For these two robots, a sudden losing of the evader happens. They have to predict the evader, which is no use. So the robots of ID 2 and 3 are back to *search* state. Shortly, the robot of ID 3 sees the evader again and restarts to pursue it. Then the other two robots find the evader successively, and finally, the evader is captured by the efforts of all robots [see Fig. (4d)].

Because there is no communication among the robots with only LCSs, not all the robots participate in the hunting activity. From all simulations we conducted, the LILCS approach is considered a robust one.

V. CONCLUSION

This paper mainly focuses on hunting by distributed mobile robots in unknown environments. In order to eliminate the effects of accumula-

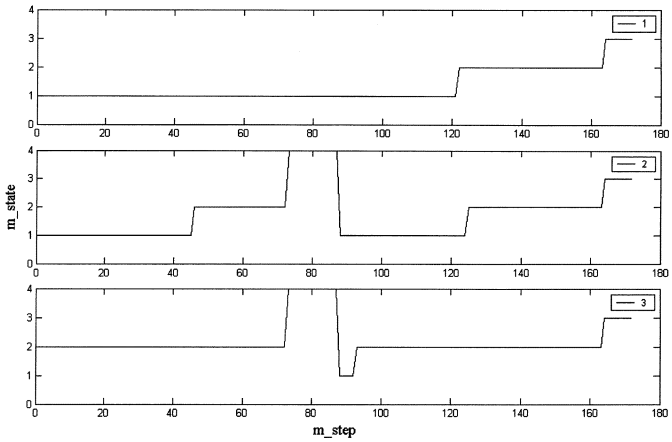

Fig. 5. State variations of each robot in Simulation 3.

tive errors rising from wheels slippage and imperfect communication, in this paper, the robots do not have a global coordinate system and no information is allowed to exchange. A distributed approach called LILCS is proposed. The approach is independent of the environment, and can cope with unexpected events. Each robot makes its own decisions by analyzing its ambient situation. The cooperation may emerge by local interactions among the robots. Simulations show the effectiveness of the LILCS approach.

REFERENCES

[1] Y. U. Cao, A. S. Fukunaga, and A. B. Kahng, "Cooperative mobile robotics: Antecedents and directions," *Auton. Robots*, vol. 4, no. 1, pp. 7–27, 1997.

[2] M. J. Mataric, "Coordination and learning in multirobot systems," *IEEE Intell. Syst. Mag.*, vol. 13, no. 2, pp. 6–8, 1998.

[3] H. Yamaguchi, "A cooperative hunting behavior by mobile-robot troops," *Int. J. Robot. Res.*, vol. 18, no. 8, pp. 931–940, 1999.

[4] H. Yamaguchi and T. Arai, "Distributed and autonomous control method for generating shape of multiple mobile robot group," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Munich, Germany, 1994, pp. 800–807.

[5] P. Pirjanian and M. J. Mataric, "Multi-robot target acquisition using multiple objective behavior coordination," in *Proc. IEEE Int. Conf. Robot. Autom.*, San Francisco, CA, 2000, pp. 2696–2702.

[6] E. Osawa, "A metalevel coordination strategy for reactive cooperative planning," in *Proc. 1st Int. Conf. Multi-Agent Syst.*, 1995, pp. 297–303.

[7] J. Denzinger and M. Fuchs, "Experiments in learning prototypical situations for variants of the pursuit game," in *Proc. Int. Conf. Multi-Agent Syst.*, 1996, pp. 48–55.

[8] M. Tan, "Multi-agent reinforcement learning: independent vs. cooperative agents," in *Proc. 10th Int. Conf. Mach. Learning*, 1993, pp. 330–337.

[9] N. Ono, K. Fukumoto, and O. Ikeda, "Collective behavior by modular reinforcement-learning animats," in *Proc. 4th Int. Conf. Simul. Adapt. Behav.*, 1996, pp. 618–624.

[10] J. Hespanha, H. Kim, and S. Sastry, "Multiple-agent probabilistic pursuit-evasion games," in *Proc. 38th IEEE CDC*, 1999, pp. 2432–2437.

[11] S. Rashid, "Design and Implementation of Multi-Agent Control: Pursuit and Map Building," Master's thesis, Univ. California, Berkeley, CA, 2000.

[12] S. Rashid and J. Kim, Multiple-Agent Probabilistic Pursuit-Evasion Games in 2.5D Univ. California, Berkeley, CA, Tech. Rep. UCB/ERL M99/34, 1999.

[13] R. Vidal, S. Rashid, C. Sharp, O. Shakernia, J. Kim, and S. Sastry, "Pursuit-evasion games with unmanned ground and aerial vehicles," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2001, pp. 2948–2955.

[14] R. Vidal, O. Shakernia, H. J. Kim, D. H. Shim, and S. Sastry, "Probabilistic pursuit-evasion games: theory, implementation, and experimental evaluation," *IEEE Trans. Robot. Autom.*, vol. 18, no. 5, pp. 662–669, Oct. 2002.

Homography-Based Visual Servo Tracking Control of a Wheeled Mobile Robot

Jian Chen, Warren E. Dixon, Darren M. Dawson, and Michael McIntyre

***Abstract*—A visual servo tracking controller is developed in this paper for a monocular camera system mounted on an underactuated wheeled mobile robot (WMR) subject to nonholonomic motion constraints (i.e., the camera-in-hand problem). A prerecorded image sequence (e.g., a video) of three target points is used to define a desired trajectory for the WMR. By comparing the target points from a stationary reference image with the corresponding target points in the live image and the prerecorded sequence of images, projective geometric relationships are exploited to construct Euclidean homographies. The information obtained by decomposing the Euclidean homography is used to develop a kinematic controller. A Lyapunov-based analysis is used to develop an adaptive update law to actively compensate for the lack of depth information required for the translation error system. Experimental results are provided to demonstrate the control design.**

***Index Terms*—Lyapunov methods, mobile robot, nonholonomic, visual servo control.**

I. INTRODUCTION

Wheeled mobile robots (WMRs) are often required to execute tasks in environments that are unstructured. Due to the uncertainty in the environment, numerous researchers have investigated different sensing methodologies as a means to enable improved autonomous response by the system. Given this motivation, researchers initially targeted the use of a variety of sonar- and laser-based sensors. Some initial work also targeted the use of a fusion of various sensors to build a map of the environment for WMR navigation (see [19], [22], [34], [36], [38], and the references within; other early innovative mobile robot control research is given in [20]). While this is still an active area of research, various shortcomings associated with these technologies and recent advances in image extraction/interpretation technology and advances in control theory have motivated researchers to investigate the sole use of camera-based vision systems for autonomous navigation. For example, using consecutive image frames and an object database, the authors of [21] recently proposed a monocular visual servo tracking controller for WMRs based on a linearized system of equations and extended Kalman filtering (EKF) techniques. Also, using EKF techniques on the linearized kinematic model, the authors of [8] used feedback from a monocular omnidirectional camera system (similar to [1]) to enable

Manuscript received February 9, 2004; revised June 14, 2004. This paper was recommended for publication by Associate Editor F. Chaumette and Editor S. Hutchinson upon evaluation of the reviewers' comments. This work was supported in part by the National Science Foundation under Grant DMI-9457967, in part by the Office of Naval Research under Grant N00014-99-1-0589, in part by a DOC Grant, in part by an ARO Automotive Center Grant at Clemson University, and in part by the Air Force Office of Scientific Research under Contract F49620-03-1-0381 at the University of Florida. This paper was presented in part at the IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, NV, October 2003.

J. Chen is with the Department of Mechanical and Aeronautical Engineering, Clarkson University, Potsdam, NY 13699-5725 USA (e-mail: jian.chen@ieee.org).

W. E. Dixon is with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611 USA.

D. M. Dawson and M. McIntyre are with the Department of Electrical and Computer Engineering, Clemson University, Clemson, SC 29634 USA.

Digital Object Identifier 10.1109/TRO.2006.862476