Cooperative Hunting by Multiple Mobile Robots Based on Local Interaction

Zhi-Qiang Cao, Min Tan, Saeid Nahavandi & Nong Gu

1. Introduction

The problem of multi-robot coordination and cooperation has drawn great interest in recent years [1]. Generally speaking, for a given task, utilizing more than one robot may enhance the quality of the solution. Furthermore, many inherently distributed tasks must require a distributed solution. However, if the robots are not properly organized, the interference among them will block the task. Many challenging issues should be considered carefully. In this paper, we conduct our research in the context of multi-robot hunting.

The hunting task concerning mobile robots and its target to be hunted is a particular challenge due to the nature of unknown and irregular motion of the target. In order to coordinate the motion of multiple mobile robots to capture or enclose a target, a novel feedback-control law [2], linear autonomous system [3] and Multiple Objective Behavior Coordination [4] have been used. Other related works including pursuit game [5]- [7], whose environments are usually modeled in grid. In this chapter, we choose non-grid environments where each robot with a limited visual field moves in any collision-free direction.

This chapter considers a typical hunting task where multiple mobile robots (pursuers) cooperatively hunt an invader with certain intelligence in unknown environments. After the invader is found, its position information is broadcasted. If each robot knows where other robots are, it may take action from the group's perspective. However, this will generate a communication burden that is too heavy to be practical. An approach where no positions of the robots are exchanged must be pursued. With less information, what action should the individual robot take? In this paper, an approach called *Cooperative Local Interaction* (CLI) is proposed to achieve cooperative hunting.

This chapter is organized as follows. Section 2 describes CLI approach where the hunting task is divided into four states and the detailed design in each state is given. In section III, the motion strategies for the invader are designed. Simulations are conducted in section IV and section V concludes the chapter.

2. CLI Approach

Assume that a multi-robot team is used to perform a hunting task. We label each robot $R_i (i = 1,2,...,N) \in \aleph$. In order to avoid possible collisions, a range sensor model S_{range} is used to perceive the environment. We assume that the robots and the invader can see each other

Source: Cutting Edge Robotics, ISBN 3-86611-038-3, pp. 784, ARS/plV, Germany, July 2005 Edited by: Kordic, V.; Lazinica, A. & Merdan, M.

with the same range as that of s_{range}. Each robot may recognize other teammates and also can recognize the invader.
The task is divided into four states, which are as follows:

search state — each robot will search the environment until the invader is found.
pursue state — when a robot knows the position of the invader either by perception or by communication and it thinks that the occasion to catch the invader hasn’t come yet, it keeps pursuing the invader.
catch state — when a robot thinks that the condition to catch the invader has been met, it will catch the invader within a period of time.
predict state — after a robot loses track of the invader, it will predict the invader within a period of time.

The central idea of CLI approach can be stated as follows: each robot determines its current state, makes decisions based on the information of the invader and other robots near it. The task is executed through local interaction among the robots. In the following, the detailed design in each state is introduced.

2.1 Search State

The robot endeavours to find the invader in the search state. The individual robot keeps away from other robots to utilize the resources better when it sees them (Boolean variable b1=1), and in other cases (b1=0) it moves randomly. Regardless of initial distribution of robots and the environment’s shape, an effective search can be accomplished. The robot’s moving direction $(x_r, y_r)^T$ is as follows.

$$\begin{bmatrix} x_r \\ y_r \end{bmatrix} = \begin{cases} \begin{bmatrix} p_{cx} - p_{nex} \\ p_{cy} - p_{ney} \end{bmatrix} / \sqrt{(p_{cx} - p_{nex})^2 + (p_{cy} - p_{ney})^2} & b_1 = 1 \\ \begin{bmatrix} \cos(-\tau \cdot sig\rho) & -\sin(-\tau \cdot sig\rho) \\ \sin(-\tau \cdot sig\rho) & \cos(-\tau \cdot sig\rho) \end{bmatrix} \cdot \begin{bmatrix} x_d \\ y_d \end{bmatrix} & b_1 = 0 \end{cases} \tag{1}$$

where $(p_{cx}, p_{cy})^T$, $(p_{nex}, p_{ney})^T$ are coordinates of the robot and another one nearest to it, respectively; $(x_d, y_d)^T$ refers to the robot’s heading; τ is an angle randomly rotated; $\rho(\rho \in [0,1])$ is a random number and $sig\rho = \begin{cases} -1 & 0.5 > \rho \geq 0 \\ 1 & 1 \geq \rho \geq 0.5 \end{cases}$.

2.2 Pursue State

Let P_T, C_T denote the current position of the invader and its center, respectively. $\wp(\chi_1, \chi_2] = \left\{ P \middle| \chi_1 < \left| \overrightarrow{PP_T} \right| \leq \chi_2 \right\}$ is defined $(\chi_1, \chi_2]$ zone around the invader, that is, the distance from position $P(P \in \wp(\chi_1, \chi_2])$ to the invader is within $(\chi_1, \chi_2]$. For robot $R_q \in \aleph$, when its position $P_r(q) \in \wp(\chi_1, \chi_2]$, the robot is called a $(\chi_1, \chi_2]$ robot.
Once a robot acquires the information of the invader, according to the distance to the invader, three zones can be generated and they are $\wp(0, x_{near}]$, $\wp(x_{near}, x_{far}]$ and $\wp(\chi_{far}, +\infty)$, where χ_{near} and χ_{far} ($x_{far} > x_{near}$) both are parameters determined by the robot’s maximum sensing range and its radius. Only when a robot is a $(0, x_{near}]$ robot is it meaningful to judge whether the condition to catch the invader is satisfied.

For a $(0, x_{near}]$ robot R_i whose position is expressed by $P_r(i)$, let $P_r^i(m_{nb})(m_{nb} = 1,2,\ldots,N_i)(N_i \geq 1)$ denote the positions of all $(0, x_{near}]$ robots near it and itself. The robots involved consist of a set $\aleph_i$. The process judging whether the condition to catch the invader is satisfied is described in the following.

Step_1: if $N_i \geq 3$, the robot chooses a m_{nb}.

Step_2: when N_i is an even number, the robot obtains $P_r^i(\min_{m_{nb}})$, which is $P_r^i(m)$ that makes the angle between $\overrightarrow{P_T p_r^i(m_{nb})}$ and $\overrightarrow{P_T P_r^i(m)}(m = 1,2,\ldots,N_i, m \neq m_{nb})$ minimal and establishes a pole coordinate system $\Sigma_{eveni}^{m_{nb}}$ whose pole, polar axis direction are C_T and $\dfrac{\overrightarrow{P_T p_r^i(m_{nb})} + \overrightarrow{P_T P_r^i(\min_{m_{nb}})}}{2}$, respectively. If N_i is an odd number, the robot establishes a polar coordinate system $\Sigma_{oddi}^{m_{nb}}$ whose pole is C_T with the polar axis direction of $\overrightarrow{P_T P_r^i(m_{nb})}$.

Step_3: calculate the coordinates $P_s^{m_{nb}i}(\lambda_s, \gamma_s)(s = 1,2,..,N_i)$ in $\Sigma_{eveni}^{m_{nb}}$ (or $\Sigma_{oddi}^{m_{nb}}$) for all robots in $\aleph_i$, where $\gamma_s \in [0,2\pi)$.

Step_4: When $\exists m_{nb}$ such that $\exists \gamma_k \in \left[\frac{\pi}{2}, \pi\right) \cap \exists \gamma_j \in \left[\pi, \frac{3\pi}{2}\right)(k, j = 1,2,\ldots,N_i)$, the condition to catch the invader is considered to be satisfied.

When a robot thinks that the condition to catch the invader is not satisfied, it is in the pursue state and has to pursue the invader first. Considering any robot R_i, the local decision-making process is as follows.

Step_1: acquire the positions of all other robots near R_i $P_{rn}^i(m_n)(m_n = 1,2,\ldots N_n)$, where N_n may be null and it means there is not a robot near R_i (go to Step_6). If $N_n > 0$, it establishes a polar coordinate system Σ_{ri} whose pole, polar axis direction are C_T and $\overrightarrow{P_T P_r(i)}$, respectively.

Step_2: calculate the coordinates $P_l^{neari}(\mu_l, \varphi_l)(l = 1,\ldots,N_n)$ for all neighboring robots in Σ_{ri}, where $\varphi_l \in (-\pi, \pi]$.

Step_3: get the angle $\varphi_{\min}$ that is the minimum of $|\varphi_l|$ and the corresponding robot's position $P_{rn}^i(\min)$.

Step_4: if $\varphi_{\min}$ is less than ς, R_i should endeavour to reduce or eliminate the interference with other robots (go to Step_5), otherwise, there is no need to consider other robots (go to Step_6), where $\varsigma = \frac{2\pi}{3}(N_n \leq 2)$ or $\varsigma = \frac{2\pi}{N_n + 1}(N_n > 2)$.

Step_5: acquire the coordinate $P_{\min}^i(\varpi_i, \phi_i)$ for R_i in a polar coordinate system $\Sigma_{\min}^i$ whose pole and polar axis direction are C_T and $\overrightarrow{P_T P_{rn}^i(\min)}$, respectively. Naturally, we have $|\phi_i| = \varphi_{\min}$. If $\phi_i > 0$, the coordinate of the robot's ideal position in $\Sigma_{\min}^i$ should be $P_d^i(r_{dis}, \varsigma)$, otherwise, it should be $P_d^i(r_{dis}, -\varsigma)$, where r_{dis} is χ_{far} (when R_i is a $(\chi_{far}, +\infty)$ robot) or $\chi_{near} - 2\Delta$ (in other cases). Δ is a margin.

Step_6: the ideal motion position $P_r^d(i)$ for R_i should locate in $\overrightarrow{P_T P_r(i)}$ and $\left|\overrightarrow{P_T P_r^d(i)}\right| = r_{dis}$.

2.3 Catch State

For the robot R_i, once the condition to catch the invader is satisfied, it is in the catch state and will catch the invader by moving to the ideal position $P_r^{cd}(i)$, which locate in $\overrightarrow{P_T P_r(i)}$ and $\left|\overrightarrow{P_T P_r^{cd}(i)}\right| = r_c$ $(r_c < \chi_{near} - 2\Delta)$, where r_c is so small that the invader cannot move any more when most of the robots involved arrive at their ideal positions. The decision-making process is adopted within certain steps $step_{catch}$ without considering the condition to catch the invader. If the step number the robot moves reaches $step_{catch}$, it means that the invader maybe breaks away from being caught. In this case, the relevant robots need to re-analyze the situation.

2.4 Predict State

Because of the complexity of the task and its environment, the robots possibly lose track of the invader. For R_i, when it loses track of the invader, it is in the predict state and has to predict the invader within certain steps $step_{predict}$ based on previous invader position P_{pre}^T and its motion information to find the invader again. We denote with P_{pre}^i the previous position of R_i. If the invader escapes along $\overrightarrow{P_{pre}^i P_{pre}^T}$ and moves in the maximum step size of R_i, the suppositional escaping position of the invader $P_{\sup}^T$ can be calculated. Thus the robot hopes to move along $\overrightarrow{P_r(i)P_{\sup}^T}$. If the step number R_i in the predict state moves reaches $step_{predict}$, it shows that it is difficult to find the invader by prediction. This requires R_i to re-search the environment.

2.5 Motion Strategy

The above decision-making of each state generates an output (a moving direction or a motion position) without considering the obstacles. Therefore, an effective motion strategy is indispensable to make each robot move effectively and safely. It combines the corresponding output with readings from sensors to control the robot.

The strategy proposed here may enable each robot to obtain its safe moving direction that has the least angle with its ideal direction on the premise of predetermined step size.

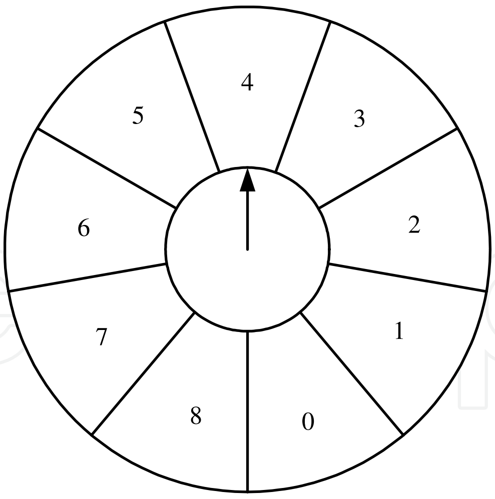


Figure 1. The layout of sensors

The robot adopts a range sensor model S_{range} to perceive the environment. The detecting zone of each sensor is a sector. Fig. 1 shows the layout of sensors whose numbers are assigned from 0 to 8 as starting from the reverse direction of the robot's heading, which is shown in arrow. The robot can know the presence or absence of other objects in each sector as well as the distance to them. For any robot R_i, it establishes a polar coordinate system Σ_{Ri} whose pole and polar axis direction are its center and current moving direction, respectively. Denote $P_{rr}(\rho_r, \theta_r)$ as the coordinates of the ideal motion position of R_i in Σ_{Ri}.

The coordinates of the detecting border of sensors $s_t(t=0,1,\ldots,8)$ in Σ_{Ri} are $P_s^t(\rho_t,\theta_t)$, where ρ_t is the maximum sensing range when no obstacle is detected, otherwise, reading from s_t after the invader is considered, and $\theta_t \in \left[-\pi+\frac{2\pi}{9}t, -\pi+\frac{2\pi}{9}(t+1)\right]$. We denote with $P_a(\rho_a,\theta)$ the coordinates of next position of R_i in Σ_{Ri}, where ρ_a is the step size determined by the robot's current position, ideal position and maximum step size; θ is the angle it rotated. The goal is to seek θ within $[-\zeta_{r\max},\zeta_{r\max}]$ of current moving direction on the premise of predetermined ρ_a such that the robot moves along the collision-free direction that has the least angle with the ideal direction. Base on sensory information, the distances $P_aP_s^t(t=0,1,\ldots,8)$ from P_a to the detecting border of each sensor should be greater than or equal to a safety distance D_{safe} determined by the robot's velocity and its radius, namely,

$$P_aP_s^{\,t} \ge D_{safe}(t=0,1,\ldots,8) \tag{2}$$

The final value of θ should satisfy eq. (2) and make $|\theta-\theta_r|$ a minimum. Considering the t^{th} sensor, we have

$$\sqrt{(\rho_a\cos\theta(t)-\rho_t\cos\theta_t)^2+(\rho_a\sin\theta(t)-\rho_t\sin\theta_t)^2} \ge D_{safe} \tag{3}$$

where $\theta(t)$ are the values of θ satisfying the condition of the t^{th} sensor in eq. (2). From eq. (3), it can be obtained that

$$\cos(\theta(t)-\theta_t) \le \frac{\rho_a^{\,2}+\rho_t^{\,2}-D_{safe}^{\,2}}{2\rho_a\rho_t} = D \tag{4}$$

When $|\rho_a-\rho_t| \ge D_{safe}$ is satisfied, $\theta(t)\in[-\zeta_{r\max},\zeta_{r\max}]$.
When $\rho_a+\rho_t < D_{safe}$ is satisfied, $\theta(t)\in\Phi$, the empty set.
When $\rho_a+\rho_t \ge D_{safe} \cap |\rho_a-\rho_t| < D_{safe}$ is satisfied, we have

$$\theta(t)-\theta_t \in [-2\pi+\arccos D, -\arccos D] \cup [\arccos D, 2\pi-\arccos D] \tag{5}$$

Any value within the range of θ_t should be suitable for the above equation, therefore,

$$\theta(t)\in\left\{\left[\arccos D-\frac{25}{9}\pi+\frac{2\pi}{9}t, -\arccos D-\pi+\frac{2\pi}{9}t\right] \cup \left[\arccos D-\frac{7}{9}\pi+\frac{2\pi}{9}t, -\arccos D+\pi+\frac{2\pi}{9}t\right]\right\} \cap [-\zeta_{r\max},\zeta_{r\max}] \tag{6}$$

when $\arccos D \le \frac{8\pi}{9}$ is satisfied, and $\theta(t)\in\Phi$, when $\arccos D > \frac{8\pi}{9}$ is satisfied.
The set of the values of θ satisfying the conditions of all sensors 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 turn right angle $\zeta_{r\max}$ without any change in its position.

3. Strategies for the Invader

Assume that the invader adopts the same model as that of the individual robot. While the invader does not see any robot or static obstacle, it moves randomly; otherwise, it will move along a safety direction determined by the safety-motion strategy. The invader establishes a polar coordinate system Σ_e whose pole is its center and the polar axis direction is its heading. Denote $P_e^i(\rho_i,\theta_i)$ as the coordinates of the detecting border of sensors $S_e^i(i=0,1,...,8)$ in Σ_e, where ρ_i is the reading from S_e^i when it senses any object, or else, the sensor is ignored and for convenience ρ_i is far greater than the maximum sensing range of the invader; $\theta_i \in \left[-\pi+\frac{2\pi}{9}i,-\pi+\frac{2\pi}{9}(i+1)\right]$. Based on the invader's current direction, Q (a multiple of 4) directions are generated and their set $\Im$ is depicted as follows.

$$\Im=\left\{\zeta_q \middle| \zeta_q=-\pi+\frac{2q\pi}{Q}(q=0,1,...,Q-1)\right\} \tag{7}$$

The invader may move to the position $P_n^q(V_e,\zeta_q)$ on the premise of the predetermined step size V_e without any collisions when the distance from the position to the detecting border of each sensor should be greater than or equal to a safety distance L_{safe} influenced by the invader's velocity and its radius, that is,

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

When $\exists\zeta_q\in\Im$ satisfying eq. (8), the invader is still capable of moving, otherwise, no feasible moving direction is available and the invader is captured.

We label Ψ as the set of all directions within $\left[-\frac{\pi}{2},\frac{\pi}{2}\right)$ of the current direction and $\Psi=\left\{\zeta_q \middle| \zeta_q=-\pi+\frac{2q\pi}{Q}\left(q=\frac{Q}{4},\frac{Q}{4}+1,...,\frac{3Q}{4}-1\right)\right\}$. When the invader can still move, the safety-motion strategy is used to select the best one ζ from all ζ_q satisfying eq. (8) in the set Ψ, and the best value should make $dis(\zeta_q)$ maximum, thus,

$$dis(\zeta)=\max_{\zeta_q} dis(\zeta_q)=\max_{\zeta_q}\min\left(d_0(\zeta_q),d_1(\zeta_q),...,d_8(\zeta_q)\right) \tag{9}$$

If ζ is found, the invader will rotate ζ with the step size V_e, or else, it only turns right $\frac{\pi}{2}$.

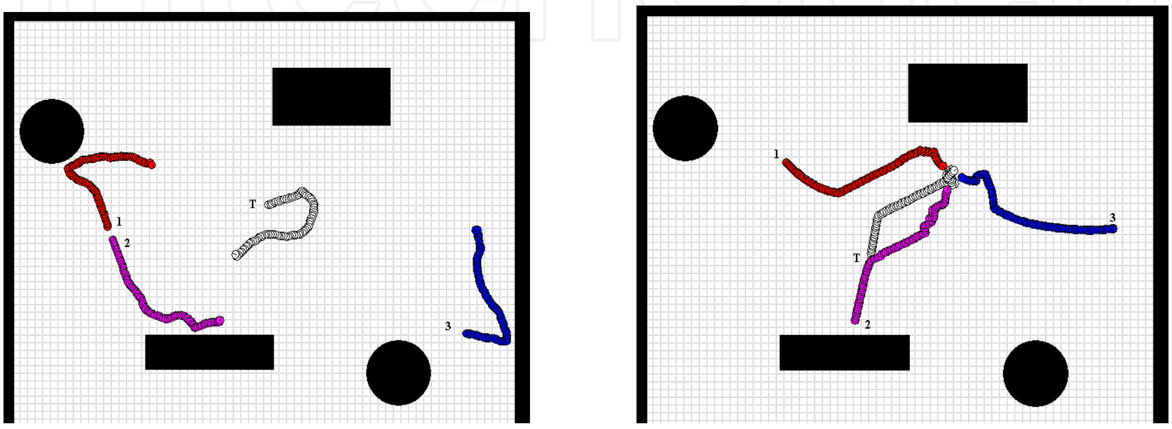
4. Simulations

In the following simulations, a random noise D_d with a mean μ and a variance σ^2 is introduced into the individual robot's sensing system in the form of $D_m=D_a(1+0.1\cdot D_d)$, where D_m and D_a are the simulated measured value and accurate value, respectively. In addition, considering the communication transfer among the robots may be failed occasionally, we assume that the robot cannot acquire the necessary information by communication with a probability of $prob_c$.

A team of robots of ID 1,2, … is required to hunt a tricky invader T. The invader is regarded as one special case of a round robot. They have the same parameters: the radius, maximum step size and maximum sensing range are 0.2, 0.1 and 3.0, respectively. μ, σ^2 in the random noise are 0 and 0.33, respectively. $prob_c$ is 0.02. The parameters in CLI approach and safety-motion strategy are shown in Table I.

Parameter	Value	Parameter	Value
τ	$\frac{\pi}{18}$	χ_{near}	1.3
χ_{far}	1.7	Δ	0.15
r_c	0.5	$step_{catch}$	15
$step_{predict}$	15	D_{safe}	0.3
$\zeta_{r\max}$	$\frac{\pi}{2}$	L_{safe}	0.39
Q	72	——	

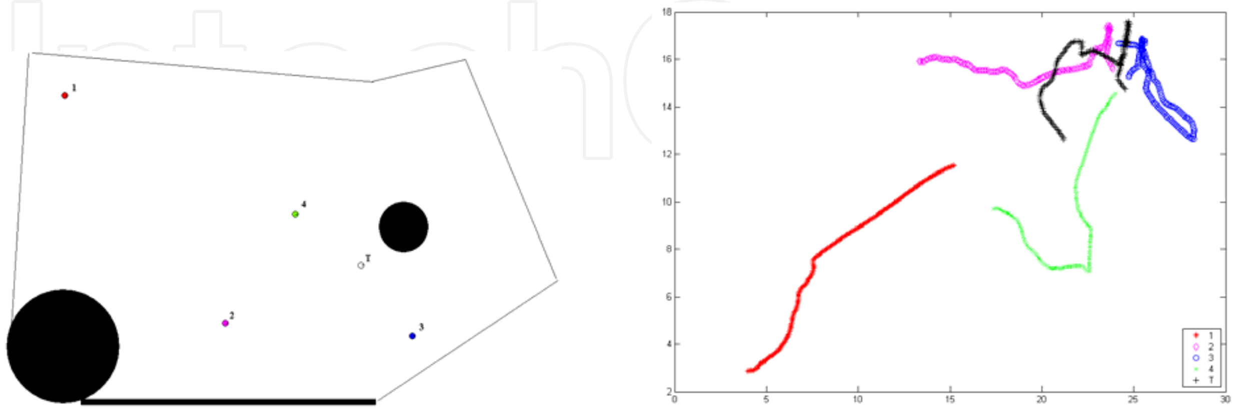
Table 1. The values of parameters concerned



(a) Before the invader is found (b) After the robotic system detects the invader

Figure 2. The trajectories of the robots and the invader in simulation 1

Fig. 2(a)~(b) show the trajectories of the robots and the invader before/after the invader is found in simulation 1, respectively. Three robots are chosen to execute the task. From Fig. 2, it is seen that at the initial stage of motion, the robot of ID 1 and 2 will keep away from each other to enlarge their visual fields. When the robot of ID 2 detects the invader T, it informs other robots and the pursuit begins. Each robot decides its own movement. By local interaction among the robots, finally the invader is captured.

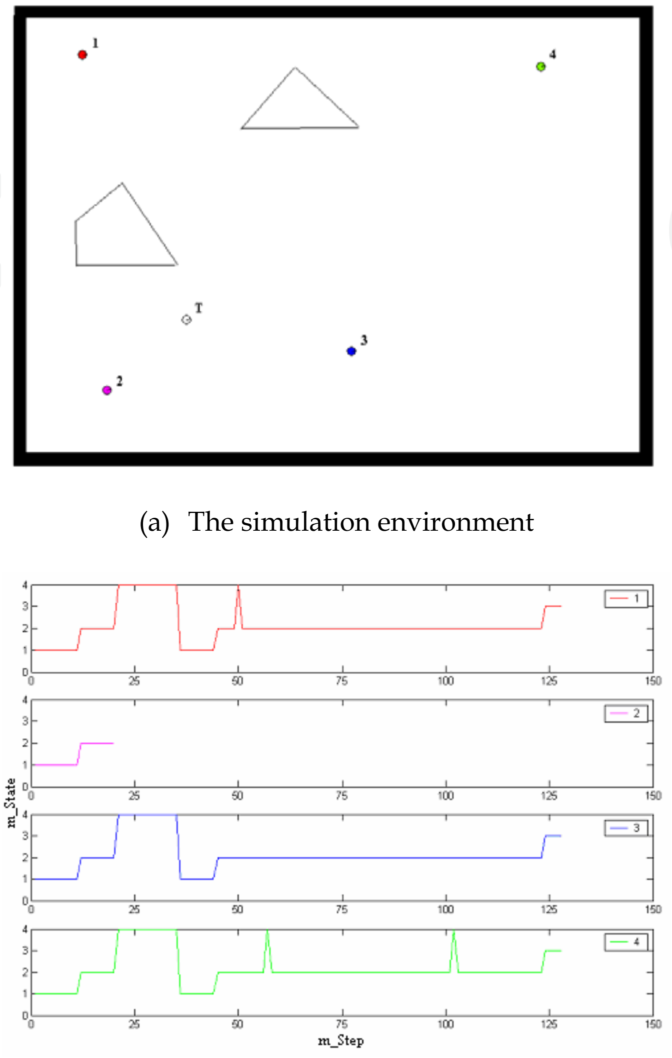


(a) The initial environment (b) The coordinates of robots and the invader

Figure 3. The simulation 2

In simulation 2, four robots are adopted and the environment is depicted in Fig. 3(a). Fig. 3(b) shows the coordinates of robots and the invader. We cans see that when the robot of

ID 1 is trying to move closer to the invader, the other three robots have already captured it. Although perhaps there are many robots (4 in this simulation) executing the task, it may be completed by local interaction among three robots. In general, more robots' participation within certain range may shorten the task time.



(a) The simulation environment

(b) The variations of the robots' states

Figure 4. The simulation 3

Simulation 3 considers the case where a robot is suddenly abnormal. The initial environment is shown in Fig. 4(a). We denote with m_State the robot's state. When m_State is chosen 1, 2, 3, 4, the robot is in the search, pursue, catch, and predict states, respectively. The variations of m_State for each robot as the task progresses are plotted in Fig. 4(b).

The process may be described as follows: The robot of ID 2 sees the invader after a short period of searching, and it broadcasts the invader's information to others. The other robots begin to move intentionally. However, several steps later, the robot of ID 2 suddenly becomes dysfunctional. The only source to provide the invader's information is cut off. Thereupon other robots try to predict the invader, and it does not work. They have to re-search the environment. Later, the robotic system finds the invader again and ultimately captures it.

To further confirm the validity of CLI approach, it is also compared with individual action (IA), an approach where each robot takes individual action regardless of other robots. Not any local interaction among the robots adopting IA approach exists. A series of simulations (simulation 4) are conducted with the distance *d* increasing (see Fig. 5). For each *d*, 20 runs were performed and the results are shown in Fig. 6, which describes the relationship of average step numbers of the robots adopting different approaches versus *d*. It can be seen that although the robots adopting IA approach sometimes may shorten the completion time than those adopting CLI approach, in many cases, the selfish behaviour of the robots adopting IA approach will lead to a delay of completion time. From all simulations conducted, CLI approach is considered an effective one.

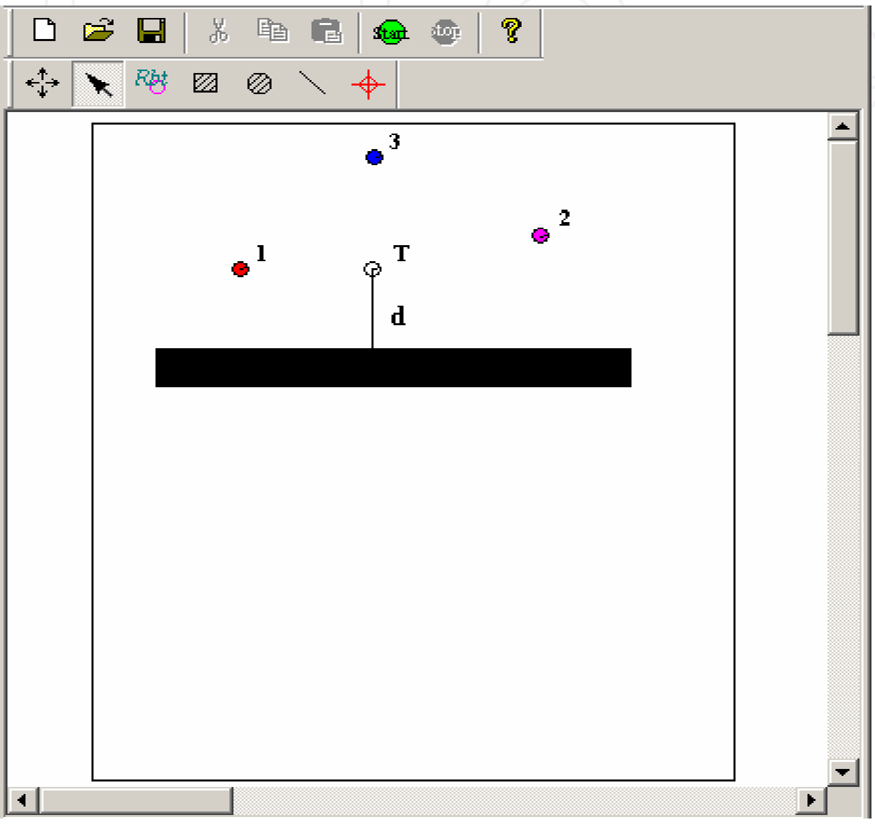


Figure 5. The test environment of simulation 4

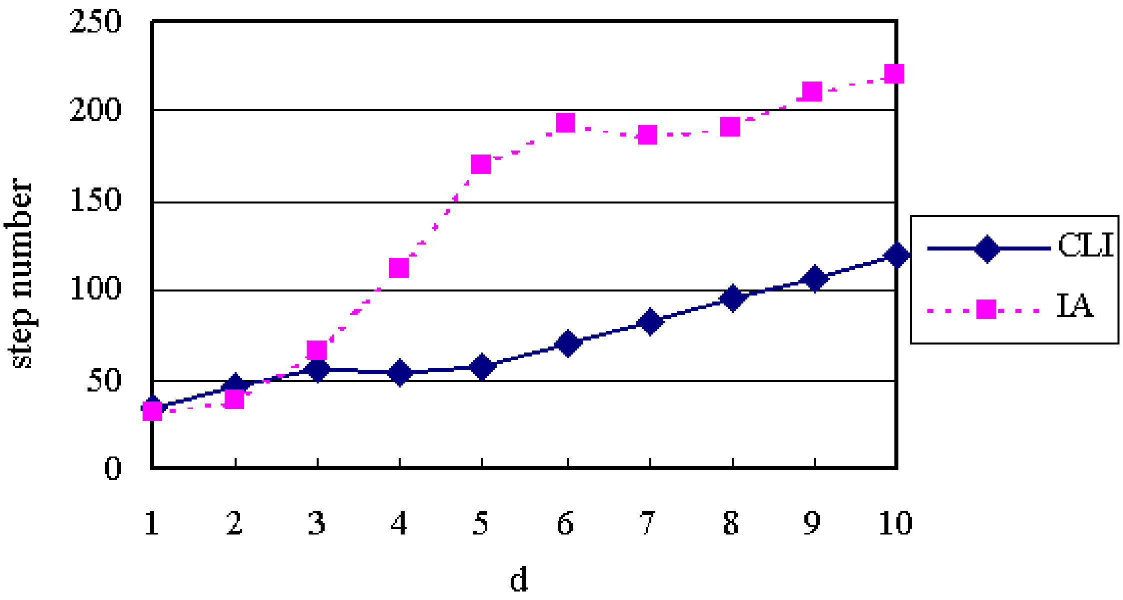


Figure 6. The comparison of different approaches for simulation 4

5. Conclusions

This chapter has mainly focused on the problem of cooperative hunting by multiple mobile robots in unknown environments. Because the positions of the robots are not exchanged among them in order to reduce the communication burden, it is hard for each robot to make a global decision. A better idea is to complete the task by local interaction among the robots.

In this chapter, an effective approach called Cooperative Local Interaction (CLI) has been proposed. The approach is robust and independent of the environments. As the invader actively tries to escape by adopting the safety-motion strategy, the difficulty of hunting is increased. The validity of CLI approach is supported by simulations.

6. References

[1] Y. U. Cao, A. S. Fukunaga, A. B. Kahng, “Cooperative mobile robotics: antecedents and directions,” Autonomous Robots, 1997, 4(1), pp.7-27

[2] H. Yamaguchi, “A cooperative hunting behavior by mobile-robot troops,” International Journal of Robotics Research, 1999, 18(8), pp.931-940

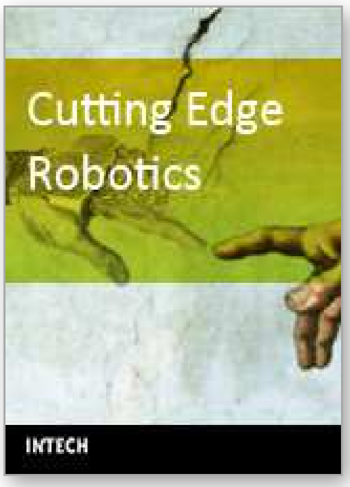
[3] H. Yamaguchi, T. Arai, “Distributed and autonomous control method for generating shape of multiple mobile robot group,” in Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS’94, Munich, Germany, 800-807

[4] Paolo Pirjanian, M. J. Mataric, “Multi-robot target acquisition using multiple objective behavior coordination,” in Proceedings of the 2000 IEEE International Conference on Robotics and Automation, San Francisco, CA, 2000: 2696-2702

[5] J. Denzinger, M. Fuchs, “Experiments in learning prototypical situations for variants of the pursuit game,” in Proceedings 2nd International Conference on Multi-agent Systems, 1996, pp.48-55

[6] Ono, N., Fukumoto, K. and Ikeda, O, “Collective Behavior by Modular Reinforcement-Learning Animats,” in Proceedings of the 4th International Conference on the Simulation of Adaptive Behavior, 1997: 618-624

[7] R. Vidal, S. Rashid, C. Sharp, O. Shakernia, J. Kim, S. Sastry, “Pursuit-evasion games with unmanned ground and aerial vehicles,” in Proceedings IEEE International Conference on Robotics and Automation, 2001, pp.2948-2955



Cutting Edge Robotics

Edited by Vedran Kordic, Aleksandar Lazinica and Munir Merdan

ISBN 3-86611-038-3
Hard cover, 784 pages
Publisher Pro Literatur Verlag, Germany
Published online 01, July, 2005
Published in print edition July, 2005

This book is the result of inspirations and contributions from many researchers worldwide. It presents a collection of wide range research results of robotics scientific community. Various aspects of current research in robotics area are explored and discussed. The book begins with researches in robot modelling & design, in which different approaches in kinematical, dynamical and other design issues of mobile robots are discussed. Second chapter deals with various sensor systems, but the major part of the chapter is devoted to robotic vision systems. Chapter III is devoted to robot navigation and presents different navigation architectures. The chapter IV is devoted to research on adaptive and learning systems in mobile robots area. The chapter V speaks about different application areas of multi-robot systems. Other emerging field is discussed in chapter VI - the human- robot interaction. Chapter VII gives a great tutorial on legged robot systems and one research overview on design of a humanoid robot.The different examples of service robots are showed in chapter VIII. Chapter IX is oriented to industrial robots, i.e. robot manipulators. Different mechatronic systems oriented on robotics are explored in the last chapter of the book.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821