Robotica (2008) volume 26, pp. 357–370. © 2008 Cambridge University Press
doi:10.1017/S0263574707004080 Printed in the United Kingdom

Distributed gradient and particle swarm optimization for multi-robot motion planning

Gerasimos G. Rigatos*

Unit of Industrial Automation, Industrial Systems Institute, 26504 Rion Patras, Greece.

(Received in Final Form: December 17, 2007. First published online: February 1, 2008)

SUMMARY

Two distributed stochastic search algorithms are proposed for motion planning of multi-robot systems: (i) distributed gradient, (ii) swarm intelligence theory. Distributed gradient consists of multiple stochastic search algorithms that start from different points in the solutions space and interact with each other while moving toward the goal position. Swarm intelligence theory is a derivative-free approach to the problem of multi-robot cooperation which works by searching iteratively in regions defined by each robot's best previous move and the best previous move of its neighbors. The performance of both approaches is evaluated through simulation tests.

KEYWORDS: Multi-robot system; Motion planning; Stochastic search algorithms; Distributed gradient; Lyapunov stability; Swarm intelligence.

1. Introduction

In recent years there has been growing interest in multi-robot systems since swarms of cooperating robots can perform complicated tasks that a single robot can not carry out. As the cost of robotic vehicles goes down and their size becomes more compact the number of military and industrial applications of multi-robot systems increases. Possible industrial applications of multi-robot systems include hazardous inspection, underwater or space exploration, assembling and transportation, search and rescue, and underground exploitation of energy resources.[1] Some examples of military applications are guarding, escorting, patrolling (surface surveillance), and strategic behaviors, such as stalking and attacking.

Control of cooperating robotic vehicles has been extensively studied in both the behavior-based and the system-theoretic approach. Behavior-based approaches for multi-robot systems have the advantage of being flexible, easy to implement and update, while they require no explicit models of the vehicle/robot and its environment.[2] These approaches are well suited to domains in which mathematical representation of tasks are difficult to obtain, and models are not available, too complex for computation, or time-varying.[3] On the other hand, system-theoretic approaches have provable performance and are applicable in cases where tasks can be parameterized, but require models of the vehicles and their environment. In the system-theoretic approach the stability of cooperative motion can been analyzed using Lyapunov theory through which suitable control functions for the steering of the individual mobile agents can be also found.[4] In the latter case, distributed control laws for multi-robot systems have been derived and have made possible motion planning through obstacles and convergence of mobile agents to targeted regions.[5,6] To implement this cooperative behavior issues related to distributed sensing, measurements fusion, and communication between the individual robots have also to be taken into account.[7,8]

This paper studies multi-robot swarms following principally the system-theoretic point of view, i.e., it is assumed that an explicit mathematical model of the robots and their interaction with the environment is available. The objective is to succeed motion planning of the multi-robot system in a workspace that contains obstacles. A usual approach for doing this is the *potential fields theory*, in which the individual robots are steered toward an equilibrium by the gradient of an harmonic potential.[9–12] Variances of this method use nonlinear anisotropic harmonic potential fields which introduce to the robots' motion directional and regional avoidance constraints.[11]

The novelty introduced is the so-called *distributed gradient* algorithm. There are M robots which emanate from arbitrary positions in the 2D space and the potential of each robot consists of two terms: (i) the cost V^i due to the distance of the i-th robot from the goal state, (ii) the cost due to the interaction with the other $M-1$ robots. Moreover, a repulsive field, generated by the proximity to obstacles, is taken into account. The gradient of the aggregate potential provides the kinematic model for each robot, and defines a path toward the equilibrium. Thus, it is proved that the update of the position of each robot is described by a gradient algorithm which contains an interaction term with the gradient algorithms defining the motion of the rest $M-1$ robots. Distributed gradient assures simultaneous convergence of the individual robots toward the equilibrium, and this convergence is analytically proved with the use of Lyapunov stability theory and LaSalle's theorem. It is shown that the mean position of the multi-robot system reaches precisely the goal state x^* while each robot stays in a bounded area close to x^*. The distributed gradient algorithm is an original result for the area of stochastic approximations and adaptive systems and can have several engineering applications. Moreover, it is of interest for the field of nanorobotics since it approximates

* Corresponding author. E-mail: grigat@isi.gr

the Brownian motion and simulates the diffusion of nano-particles.[13]

An alternative solution to multi-robot motion planning proposed by this paper is based on *swarm intelligence*. Previous applications of the particle swarm optimization algorithm for the steering toward desirable final positions and the simulation-solution of diffusion systems can be found in.[14,15] This method works by searching iteratively in regions defined by each robot's best previous move and the best previous move of its neighbors. Swarm intelligence is evident in biological systems and has been also studied in statistical physics, where collective behavior of self-propelled particles has been observed.[16] The method is useful for the avoidance of local minima. A swarm, which is a collection of robots, can converge to a wide range of distributions, while no individual robot is aware of the distribution it is working to realize. The dynamic behavior of the robots under the particle swarm algorithm can be analyzed with the use of ordinary differential equations.[17] It can be shown that appropriate tuning of the differential equation's coefficients can prevent explosion, i.e. the robots velocity is kept within certain bounds.

The structure of the paper is as follows: In Section 2 elements of stochastic search algorithms are summarized and distributed gradient is proposed for multi-robot motion planning. Stability analysis of the distributed gradient algorithm is performed with the use of Lyapunov theory. In Section 3 particle swarm theory is proposed for multi-robot motion planning. The dynamic behavior of the robots is analyzed with the use of ordinary differential equations. In Section 4 the performance of the distributed gradient algorithm and of the particle swarm theory in the problem of multi-robot motion planning is tested through simulation tests. Finally, in Section 5 concluding remarks are stated.

2. Distributed Stochastic Search for Multi-Robot Motion Planning

Motion planning of multi-robot systems can be solved with the use of *distributed stochastic search* algorithms. These can be multiple gradient algorithms that start from different points in the solutions space and interact with each other while moving toward the goal position. Distributed gradient algorithms stem from stochastic search algorithms which retreated in[18,19] if an interaction term is added:

$$x^i(t+1) = x^i(t) + \gamma^i(t)[h(x^i(t)) + e^i(t)] + \sum_{j=1, j\neq i}^{M} g(x^i - x^j),\ i = 1, 2, \ldots, M. \quad (1)$$

The term $h(x(t)^i) = -\nabla_{x^i} V^i(x^i)$ indicates a local gradient algorithm, i.e., motion in the direction of decrease of the cost function $V^i(x^i) = \frac{1}{2} e^i(t)^T e^i(t)$. The term $\gamma^i(t)$ is the algorithms step while the stochastic disturbance $e^i(t)$ enables the algorithm to escape from local minima. The term $\sum_{j=1, j\neq i}^{M} g(x^i - x^j)$ describes the interaction between the i-th and the rest $M-1$ stochastic search algorithms. Convergence analysis based on the Lyapunov stability theory can be stated in the case of distributed gradient algorithms. This is important for the problem of multi-robot motion planning.

2.1. Kinematic model of the multi-robot system

The objective is to lead a swarm of M mobile robots, with different initial positions on the 2-D plane, to a desirable final position. The position of each robot in the 2-D space is described by the vector $x^i \in R^2$. The motion of the robots is synchronous, without time delays, and it is assumed that at every time instant each robot i is aware about the position and the velocity of the other $M-1$ robots. The cost function that describes the motion of the i-th robot toward the goal state is denoted as $V(x^i): R^n \to R$. The value of $V(x^i)$ is high on hills, small in valleys, while it holds $\nabla_{x^i} V(x^i) = 0$ at the goal position and at local optima. The following conditions must hold:

(i) The cohesion of the swarm should be maintained, i.e. the norm $\|x^i - x^j\|$ should remain upper bounded $\|x^i - x^j\| < \epsilon^h$,
(ii) Collisions between the robots should be avoided, i.e. $\|x^i - x^j\| > \epsilon^l$,
(iii) Convergence to the goal state should be succeeded for each robot through the negative definiteness of the associated Lyapunov function $\dot{V}^i(x^i) = \dot{e}^i(t)^T e^i(t) < 0$.[18]

The interaction between the i-th and the j-th robot is

$$g(x^i - x^j) = -(x^i - x^j)[g_a(\|x^i - x^j\|) - g_r(\|x^i - x^j\|)], \quad (2)$$

where $g_a()$ denotes the attraction term and is dominant for large values of $\|x^i - x^j\|$, while $g_r()$ denotes the repulsion term and is dominant for small values of $\|x^i - x^j\|$. Function $g_a()$ can be associated with an attraction potential, i.e. $\nabla_{x_i} V_a(\|x^i - x^j\|) = (x^i - x^j) g_a(\|x^i - x^j\|)$. Function $g_r()$ can be associated with a repulsion potential, i.e. $\nabla_{x_i} V_r(\|x^i - x^j\|) = (x^i - x^j) g_r(\|x^i - x^j\|)$. A suitable function $g()$ that describes the interaction between the robots is given by[20]

$$g(x^i - x^j) = -(x^i - x^j)\left(a - b e^{\frac{\|x^i - x^j\|^2}{\sigma^2}}\right), \quad (3)$$

where the parameters a, b, and c are suitably tuned. It holds that $g_a(x^i - x^j) = -a$, i.e. attraction has a linear behavior (spring-mass system) $\|x^i - x^j\| g_a(x^i - x^j)$. Moreover, $g_r(x^i - x^j) = b e^{\frac{-\|x^i - x^j\|^2}{\sigma^2}}$ which means that $g_r(x^i - x^j)\|x^i - x^j\| \leq b$ is bounded. Applying Newton's laws to the i-th robot yields

$$\dot{x}^i = v^i, \quad m^i \dot{v}^i = U^i, \quad (4)$$

where the aggregate force is $U^i = f^i + F^i$. The term $f^i = -K_v v^i$ denotes friction, while the term F^i is the propulsion. Assuming zero acceleration $\dot{v}^i = 0$ one gets $F^i = K_v v^i$, which for $K_v = 1$ and $m^i = 1$ gives $F^i = v^i$. Thus an approximate kinematic model is

$$\dot{x}^i = F^i. \quad (5)$$

According to the Euler-Langrange principle, the propulsion F^i is equal to the derivative of the total potential of each robot, i.e.

$$F^i = -\nabla_{x^i}\{V^i(x^i) + \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1,j\neq i}^{M}[V_a(\|x^i - x^j\| + V_r(\|x^i - x^j\|)]\} \Rightarrow$$

$$F^i = -\nabla_{x^i}\{V^i(x^i)\} + \sum_{j=1,j\neq i}^{M}[\nabla_{x^i}V_a(\|x^i - x^j\|) - \nabla_{x^i}V_r(\|x^i - x^j\|)] \Rightarrow$$

$$F^i = -\nabla_{x^i}\{V^i(x^i)\} + \sum_{j=1,j\neq i}^{M}[-(x^i - x^j)g_a(\|x^i - x^j\|) - (x^i - x^j)g_r(\|x^i - x^j\|)] \Rightarrow$$

$$F^i = -\nabla_{x^i}\{V^i(x^i)\} - \sum_{j=1,j\neq i}^{M} g(x^i - x^j).$$

Substituting in Eq. (5) one gets Eq. (1), i.e. $x^i(t+1) = x^i(t) + \gamma^i(t)[-\nabla_{x^i}V^i(x^i) + e^i(t+1)] - \sum_{j=1,j\neq i}^{M} g(x^i - x^j)$, $i = 1, 2, \ldots, M$, with $\gamma^i(t) = 1$, which verifies that the kinematic model of a multi-robot system is equivalent to a distributed gradient search algorithm.

2.2. *Stability of the multi-robot system*

The behavior of the multi-robot system is determined by the behavior of its center (mean of the vectors x^i) and of the position of each robot with respect to this center. The center of the multi-robot system is given by

$$\bar{x} = E(x^i) = \frac{1}{M}\sum_{i=1}^{M} x^i \Rightarrow \dot{\bar{x}} = \frac{1}{M}\sum_{i=1}^{M}\dot{x}^i \Rightarrow$$

$$\dot{\bar{x}} = \frac{1}{M}\sum_{i=1}^{M}\left[-\nabla_{x^i}V^i(x^i) - \sum_{j=1,j\neq i}^{M}(g(x^i - x^j))\right]. \quad (6)$$

From Eq. (3) it can be seen that $g(x^i - x^j) = -g(x^j - x^i)$, i.e. $g()$ is an odd function. Therefore, it holds that $\frac{1}{M}(\sum_{j=1,j\neq i}^{M} g(x^i - x^j)) = 0$, and

$$\dot{\bar{x}} = \frac{1}{M}\sum_{i=1}^{M}[-\nabla_{x^i}V^i(x^i)]. \quad (7)$$

Denoting the goal position by x^*, and the distance between the i-th robot and the mean position of the multi-robot system by $e^i(t) = x^i(t) - \bar{x}$ the objective of distributed gradient for robot motion planning can be summarized as follows:

(i) $lim_{t\to\infty}\bar{x} = x^*$, i.e. the center of the multi-robot system converges to the goal position,
(ii) $lim_{t\to\infty}x^i = \bar{x}$, i.e. the i-th robot converges to the center of the multi-robot system,
(iii) $lim_{t\to\infty}\dot{\bar{x}} = 0$, i.e. the center of the multi-robot system stabilizes at the goal position.

If conditions (i) and (ii) hold then $lim_{t\to\infty}x^i = x^*$. Furthermore, if condition (iii) also holds then all robots will stabilize close to the goal position.

It is known that the stability of local gradient algorithms can be proved with the use of Lyapunov theory[19]. A similar approach can be followed in the case of the distributed gradient algorithms given by Eq. (1). The following simple Lyapunov function is considered for each gradient algorithm[20]:

$$V_i = \frac{1}{2}{e^i}^T e^i \Rightarrow V_i = \frac{1}{2}\|e_i\|^2. \quad (8)$$

Thus, one gets

$$\dot{V}^i = {e^i}^T\dot{e}^i \Rightarrow \dot{V}^i = (\dot{x}^i - \dot{\bar{x}})e^i$$

$$\Rightarrow \dot{V}^i = \left[-\nabla_{x^i}V^i(x^i) - \sum_{j=1,j\neq i}^{M} g(x^i - x^j) + \frac{1}{M}\sum_{j=1}^{M}\nabla_{x^j}V^j(x^j)\right]e^i.$$

Substituting $g(x^i - x^j)$ from Eq. (3) yields

$$\dot{V}_i = \left[-\nabla_{x^i}V^i(x^i) - \sum_{j=1,j\neq i}^{M}(x^i - x^j)a + \sum_{j=1,j\neq i}^{M}(x^i - x^j) \times g_r(\|x^i - x^j\|) + \frac{1}{M}\sum_{j=1}^{M}\nabla_{x^j}V^j(x^j)\right]e^i,$$

which gives,

$$\dot{V}_i = -a\left[\sum_{j=1,j\neq i}^{M}(x^i - x^j)\right]e^i + \sum_{j=1,j\neq i}^{M} g_r(\|x^i - x^j\|) \times (x^i - x^j)^T e^i - \left[\nabla_{x^i}V^i(x^i) - \frac{1}{M}\sum_{j=1}^{M}\nabla_{x^j}V^j(x^j)\right]^T e^i.$$

It holds that $\sum_{j=1}^{M}(x^i - x^j) = Mx^i - M\frac{1}{M}\sum_{j=1}^{M}x^j = Mx^i - M\bar{x} = M(x^i - \bar{x}) = Me^i$, therefore

$$\dot{V}_i = -aM\|e^i\|^2 + \sum_{j=1,j\neq i}^{M} g_r(\|x^i - x^j\|)(x^i - x^j)^T e^i - \left[\nabla_{x^i}V^i(x^i) - \frac{1}{M}\sum_{j=1}^{M}\nabla_{x^j}V^j(x^j)\right]^T e^i. \quad (9)$$

It assumed that for all x^i there is a constant $\bar{\sigma}$ such that

$$\|\nabla_{x^i}V^i(x^i)\| \leq \bar{\sigma}. \quad (10)$$

https://doi.org/10.1017/S0263574707004080 Published online by Cambridge University Press

Eq. (10) is reasonable since for a robot moving on a 2-D plane, the gradient of the cost function $\nabla_{x^i} V^i(x^i)$ is expected to be bounded. Moreover it is known that the following inequality holds:

$$\sum_{j=1,j\neq i}^{M} g_r(x^i - x^j)^T e^i \leq \sum_{j=1,j\neq i}^{M} be^i \leq \sum_{j=1,j\neq i}^{M} b\|e^i\|.$$

Thus the application of Eq. (9) gives:

$$\begin{aligned}\dot{V}^i &\leq aM\|e^i\|^2 + \sum_{j=1,j\neq i}^{M} g_r(\|x^i - x^j\|)\|x^i - x^j\| \cdot \|e^i\| \\ &\quad + \|\nabla_{x^i} V^i(x^i) - \frac{1}{M}\sum_{j=1}^{M}\nabla_{x^j} V^j(x^j)\|\|e^i\| \\ &\Rightarrow \dot{V}^i \leq aM\|e^i\|^2 + b(M-1)\|e^i\| + 2\bar{\sigma}\|e^i\|,\end{aligned}$$

where it has been taken into account that

$$\sum_{j=1,j\neq i}^{M} g_r(\|x^i - x^j\|)^T\|e^i\| \leq \sum_{j=1,j\neq i}^{M} b\|e^i\| = b(M-1)\|e^i\|,$$

and from Eq. (10),

$$\begin{aligned}&\left\|\nabla_{x^i} V^i(x^i) - \frac{1}{M}\sum_{j=1}^{M}\nabla_{x^i} V^j(x^j)\| \leq \|\nabla_{x^i} V^i(x^i)\right\| \\ &\quad + \frac{1}{M}\left\|\sum_{j=1}^{M}\nabla_{x^i} V^j(x^j)\right\| \leq \bar{\sigma} + \frac{1}{M}M\bar{\sigma} \leq 2\bar{\sigma}.\end{aligned}$$

Thus, one gets

$$\dot{V}^i \leq aM\|e^i\| \cdot \left[\|e^i\| - \frac{b(M-1)}{aM} - 2\frac{\bar{\sigma}}{aM}\right]. \tag{11}$$

The following bound ϵ is defined:

$$\epsilon = \frac{b(M-1)}{aM} + \frac{2\bar{\sigma}}{aM} = \frac{1}{aM}(b(M-1) + 2\bar{\sigma}). \tag{12}$$

Thus, when $\|e^i\| > \epsilon$, $\dot{V}_i$ will become negative and consequently the error $e^i = x^i - \bar{x}$ will decrease. Therefore the error e^i will remain in an area of radius ϵ i.e. the position x^i of the i-th robot will stay in the cycle with center $\bar{x}$ and radius ϵ.

2.3. *Stability in the case of a quadratic cost function*

The case of a convex quadratic cost function is examined, for instance

$$V^i(x^i) = \frac{A}{2}\|x^i - x^*\|^2 = \frac{A}{2}(x^i - x^*)^T(x^i - x^*), \tag{13}$$

where $x^* = [0, 0]$ is a minimum point $V^i(x^i = x^*) = 0$. The distributed gradient algorithm is expected to converge to x^*. The robotic vehicles will follow different different trajectories on the 2-D plane and will end at the goal position.

Using Eq. (13) yields $\nabla_{x^i} V^i(x^i) = A(x^i - x^*)$. Moreover, the assumption $\nabla_{x^i} V^i(x^i) \leq \bar{\sigma}$ can be used, since the gradient of the cost function remains bounded. The robotic vehicles will concentrate round $\bar{x}$ and will stay in a radius ϵ given by Eq. (12). The motion of the mean position $\bar{x}$ of the vehicles is

$$\begin{aligned}\dot{\bar{x}} &= -\frac{1}{M}\sum_{i=1}^{M}\nabla_{x^i} V^i(x^i) \Rightarrow \dot{\bar{x}} = -\frac{A}{M}(x^i - x^*) \\ &\Rightarrow \dot{\bar{x}} - \dot{x}^* = -\frac{A}{M}x^i + \frac{A}{M}x^* \Rightarrow \dot{\bar{x}} - \dot{x}^* = -A(\bar{x} - x^*).\end{aligned} \tag{14}$$

The variable $e_\sigma = \bar{x} - x^*$ is defined, and consequently

$$\dot{e}_\sigma = -Ae_\sigma \Rightarrow \epsilon_\sigma(t) = c_1 e^{-At} + c_2, \tag{15}$$

with $c_1 + c_2 = e_\sigma(0)$. Eq. (15) is an homogeneous differential equation, which for $A > 0$ results into $lim_{t\to\infty} e_\sigma(t) = 0$, thus $lim_{t\to\infty}\bar{x}(t) = x^*$. It is left to make more precise the position to which each robot converges.

2.4. *Convergence analysis using La Salle's theorem*

It has been shown that $lim_{t\to\infty}\bar{x}(t) = x^*$ and from Eq. (11) that each robot will stay in a cycle C of center $\bar{x}$ and radius ϵ given by Eq. (12). The Lyapunov function given by Eq. (8) is negative semi-definite, therefore asymptotic stability cannot be guaranteed. It remains to make precise the area of convergence of each robot in the cycle C of center $\bar{x}$ and radius ϵ. To this end, La Salle's theorem can be employed.[20,21]

La Salle's Theorem: Assume the autonomous system $\dot{x} = f(x)$ where $f : D \to R^n$. Assume $C \subset D$, a compact set which is positively invariant with respect to $\dot{x} = f(x)$, i.e. if $x(0) \in C \Rightarrow x(t) \in C\ \forall\ t$. Assume that $V(x) : D \to R$ is a continuous and differentiable Lyapunov function such that $\dot{V}(x) \leq 0$ for $x \in C$, i.e. $V(x)$ is negative semi-definite in C. Denote by E the set of all points in C such that $\dot{V}(x) = 0$. Denote by M the largest invariant set in E and its boundary by L^+, i.e. for $x(t) \in E : lim_{t\to\infty} x(t) = L^+$, or in other words L^+ is the positive limit set of E. Then every solution $x(t) \in C$ will converge to M as $t \to \infty$. (See Fig. 1).

La Salle's theorem in applicable in the case of the multi-robot system and helps to describe more precisely the area round $\bar{x}$ to which the robot trajectories x^i will converge. A generalized Lyapunov function is introduced which is expected to verify the stability analysis based on Eq. (11). It holds that

$$\begin{aligned}V(x) &= \sum_{i=1}^{M} V^i(x^i) \\ &\quad + \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1,j\neq i}^{M}\{V_a(\|x^i - x^j\| - V_r(\|x^i - x^j\|)\}\end{aligned}$$

https://doi.org/10.1017/S0263574707004080 Published online by Cambridge University Press

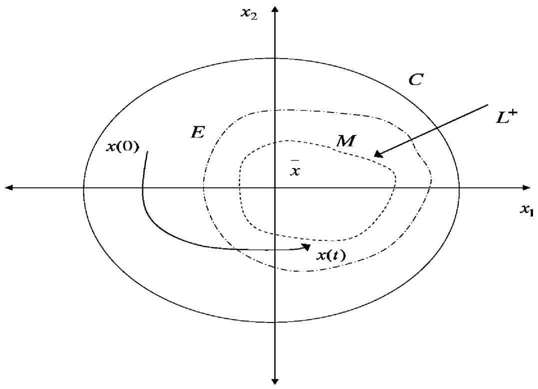


Fig. 1. LaSalle's theorem: C: invariant set, $E \subset C$: invariant set which satisfies $\dot{V}(x) = 0$, $M \subset E$: invariant set, which satisfies $\dot{V}(x) = 0$, and which contains the limit points of $x(t) \in E$, L^+ the set of limit points of $x(t) \in E$.

$$\Rightarrow V(x) = \sum_{i=1}^{M} V^i(x^i) + \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1, j\neq i}^{M} \{a\|x^i - x^j\| - V_r(\|x^i - x^j\|)$$

and

$$\nabla_{x^i} V(x) = \left[\sum_{i=1}^{M} \nabla_{x^i} V^i(x^i)\right] + \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1, j\neq i}^{M} \nabla_{x^i} \times \{a\|x^i - x^j\| - V_r(\|x^i - x^j\|)\}$$
$$\Rightarrow \nabla_{x^i} V(x) = \left[\sum_{i=1}^{M} \nabla_{x^i} V^i(x^i)\right] + \sum_{j=1, j\neq i}^{M} (x^i - x^j) \times \{g_a(\|x^i - x^j\|) - g_r(\|x^i - x^j\|)\}$$
$$\Rightarrow \nabla_{x^i} V(x) = \left[\sum_{i=1}^{M} \nabla_{x^i} V^i(x^i)\right] + \sum_{j=1, j\neq i}^{M} (x^i - x^j) \times \{a - g_r(\|x^i - x^j\|)\}$$

and using Eq. (1) with $\gamma^i(t) = 1$ yields $\nabla_{x^i} V(x) = -\dot{x}^i$, and

$$\dot{V}(x) = \nabla_x V(x)^T \dot{x} = \sum_{i=1}^{M} \nabla_{x^i} V(x)^T \dot{x}^i$$
$$\Rightarrow \dot{V}(x) = -\sum_{i=1}^{M} \|\dot{x}^i\|^2 \leq 0. \quad (16)$$

Therefore, in the case of a quadratic cost function it holds $V(x) > 0$ and $\dot{V}(x) \leq 0$ and the set $C = \{x : V(x(t)) \leq V(x(0))\}$ is compact and positively invariant. Thus, by applying La Salle's theorem one can show the convergence of $x(t)$ to the set $M \subset C$, $M = \{x : \dot{V}(x) = 0\} \Rightarrow M = \{x : \dot{x} = 0\}$.

3. Particle Swarm Theory for Multi-Robot Motion Planning

3.1. The particle swarm theory

It has been shown that the distributed gradient algorithm can have satisfactory performance for the motion planning problem of multi-robot systems in the case of quadratic cost functions. An alternative method of distributed search for the goal position is the particle swarm algorithm which belongs to derivative-free optimization techniques[17].

The similarity between the particle models and the distributed gradient algorithms is noteworthy. Particle models consist of M particles with mass m^i, position x^i, and velocity v^i. Each particle has a self-propelling force F^i. To prevent the particles from reaching large speeds, a friction force with coefficient K_v is introduced. In addition, each particle is subject to an attractive force which is affected by the proximity σ to other particles. This force is responsible for swarming. To prevent particle collisions a shorter-range repulsive force is introduced. In analogy to Eq. (3), the potential of the particles is given by

$$V_a - V_r = \sum_{j=1, j\neq i}^{M} a e^{-\frac{(\|x^i - x^j\|)^2}{\sigma^2}} - \sum_{j=1, j\neq i}^{M} b e^{-\frac{(\|x^i - x^j\|)^2}{\sigma^2}},$$

where a and b determine the strength of the attractive and the repulsive force respectively. Thus, the motion equations for each particle are[16]:

$$m_i \frac{\partial}{\partial t} v^i = F^i - K_v v^i - \nabla(V_a - V_r).$$
$$\frac{\partial}{\partial t} x^i = v^i \quad (17)$$

The particle swarm algorithm evolves in the search space by modifying the trajectories of the independent vectors $x^i(t)$ which are called *particles*. Considering each robot as a particle, the new position of each robot $x^i(t+1)$ is selected taking into account the moves of the robot from its current position $x^i(t)$ and the best moves of the rest $M-1$ robots from their positions at time instant t, i.e. $x^j(t)$ $j = 1, \ldots, M \vee j \neq i$.

Assume a set of M robots which is initialized at random positions $x^i(0)$ and which have initial velocities $v^i(0)$. The cost function of the i-th robot is denoted again by $V^i(x^i)$. The following parameters are defined (Fig. 2):

(i) $x^i(t)$ is the position vector of the i-th robot at time instant t,
(ii) $p^i(t)$ is the best position (according to V^i) to which the i-th robot can move, starting from its current position $x^i(t)$,
(iii) $p^g(t)$ is the best position (according to V^i) to which the neighbors of the i-th robot can move, starting from their current positions $x^j(t)$ $j = 1, \ldots, M \vee j \neq i$.

Figure 2 describes the Von Neumann region round each mobile robot. The 2D-plane is divided into a grid of square cells and at the time instant k, the robot is assumed to be cell at $c_{i,j}$. Then at time instant $k+1$, the robot can make

https://doi.org/10.1017/S0263574707004080 Published online by Cambridge University Press

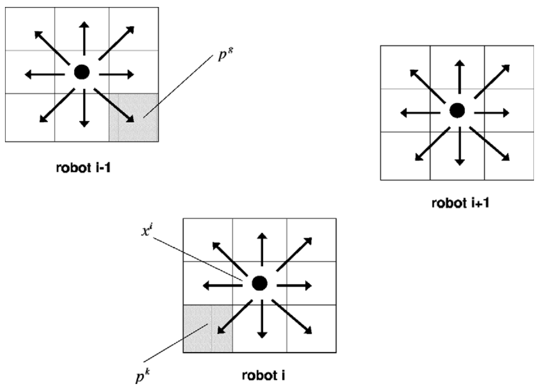


Fig. 2. Von Neumann region round each robot in the particle swarm algorithm.

one of the following moves: $c_{i+1,j}$, $c_{i+1,j-1}$,$c_{i+1,j+1}$, $c_{i-1,j}$, $c_{i-1,j-1}$, $c_{i-1,j+1}$, $c_{i,j-1}$, and $c_{i,j+1}$.

3.2. *Stability of the particle swarm algorithm*

The primary concerns of the particle swarm theory are: (i) Each particle i should move in the direction of cost function decrease (negative gradient), taking into account the directions already examined by the neighboring particles, (ii) The velocity of each particle should approach 0 as time goes to infinity. To this end, the dynamic behavior of the particle swarm can be studied with the use of ordinary differential equations, following the analysis given in[17]. The position and the velocity update of the i-th particle is:

$$v^i(t+1) = v^i(t) + \phi_1(p^k - x^i) + \phi_2(p^g - x^i), \quad (18)$$
$$x^i(t+1) = x^i(t) + v^i(t+1). \quad (19)$$

Variable p^k denotes the best possible move of the k-th individual robot, taking into account that the Von-Neumann motion pattern of the robot permits movement to eight neighboring cells. p^g is the best possible move of the neighboring robots, i.e. the movement that results in the largest decrease of the cost function. It holds that

$$v^i(t+1) = v^i(t) + \phi_1(p^k - x^i) + \phi_2(p^g - x^i)$$
$$\Rightarrow v^i(t+1) = v^i(t) + \frac{\phi_1 p^i + \phi_2 p^g}{\phi_1 + \phi_2}(\phi_1 + \phi_2) - x^i(\phi_1 + \phi_2)$$
$$\Rightarrow v^i(t+1) = v^i(t) + \phi\bar{p} - \phi x^i \Rightarrow v^i(t+1)$$
$$= v^i(t) + \phi(\bar{p} - x^i).$$

The parameter ϕ_1 determines the contribution to the update of the position of the i-th robot of the term $p^k - x^i$ which denotes the distance between robot's position x^i and the position reached after its best possible move p^k. The parameter ϕ_2 determines the contribution to the update of the position of the i-th robot of the term $p^g - x^i$, which denotes the distance between the robot's position x^i and the position reached after the best possible move of its neighbors p^g. Through parameters ϕ_1 and ϕ_2 the following parameters are defined: $\phi = \phi_1 + \phi_2$ and $\bar{p}^i = \frac{\phi_1 p^i + \phi_2 p^g}{\phi_1 + \phi_2}$. Thus, the following simplified equations can be derived

$$v^i(t+2) = v^i(t+1) + \phi(\bar{p} - x^i(t+1)), \quad (20)$$
$$x^i(t+1) = x^i(t) + v^i(t+1). \quad (21)$$

To refine the search in the solutions space, tuning through a constriction coefficient $\chi = \frac{\kappa}{\rho_2}$, $\kappa \in (0, 1)$ is introduced in Eq. (19) and Eq. (18). In that case the particle swarm algorithm takes the following form:

$$v^i(t+1) = \chi(v^i(t) + \phi(\bar{p}^i - x^i(t)),$$
$$x^i(t+1) = \chi(x^i(t) + v^i(t+1)). \quad (22)$$

It holds that

$$v^i(t+2) = v^i(t+1)(1-\phi) + \phi(\bar{p} - x^i(t)). \quad (23)$$

Subtracting Eq. (20) from Eq. (23) yields

$$v^i(t+2) + (\phi - 2)v^i(t+1) + v^i(t) = 0.$$

Using the z-transform a frequency space expression of the above difference equation is $z^2 + (\phi - 2)z + 1 = 0$. Thus, the dynamic behavior of the particle depends on the roots of the polynomial $z^2 + (\phi - 2)z + 1$ which are $\rho_1 = 1 - \frac{\phi}{2} + \frac{\sqrt{\phi^2 - 4\phi}}{2}$ and $\rho_2 = 1 - \frac{\phi}{2} - \frac{\sqrt{\phi^2 - 4\phi}}{2}$. The general solution of the differential equation is

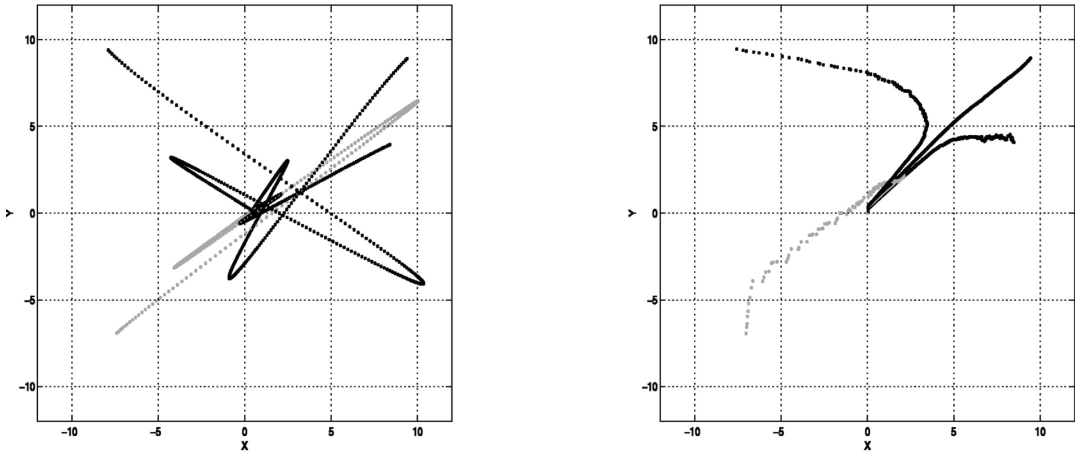
$$v^i(t) = c_1^i e^{\rho_1 t} + c_2^i e^{\rho_2 t}. \quad (24)$$

The parameters c_1^i and c_2^i are random. In Eq. (24) the stability condition $lim_{t\to\infty} v^i(t) = 0$ is assured if $\phi \geq 4$[17]. The pseudocode of the particle swarm algorithm is summarized as follows:

Initialize the robots population randomly: $x^i(0)$, $v^i(0)$, $i = 1, 2, \ldots, M$.
Do (until convergence to x^*)
{
For ($i = 1; i < M; i++$)
For all possible moves p^i, $i = 1, 2, \ldots, N$ from x^i
$p^k = arg\ min_{p^i}\{V^i(p^i)\}$
For all particles x^j, $j = 1, 2, \cdots, G$ in area g
$p^g = arg\ min_{p^j}\{V^i(p^g)\}$;
If ($V^i(p^k) < V^i(x^i)$)
$v^i(t+1) = v^i(t) + \phi_1(p^k - x^i) + \phi_2(p^g - x^i)$;
$v^i(t+1) = sgn\{v^i(t+1) min[v^i(t+1), v_{max}]\}$;
$x^i(t+1) = x^i(t) + v^i(t+1)$;
}.

Parameters ϕ_1 and ϕ_2 are selected from a uniform distribution, taking into account the above mentioned convergence conditions. The robots velocity is bounded in the interval $\pm v_{max}$. Random weighting with the use of the parameters ϕ_1 and ϕ_2 helps to avoid local minima but can lead to explosion. It can be observed that:

- The term $\phi_1(p^k - x^i)$ stands for $\nabla_{x^i} V^i(x^i)$ of Eq. (1).
- The term $\phi_2(p^g - x^i)$ stands for the term $\sum_{j=1, j\neq i}^{M} g(x^i - x^j)$. It is assumed that in the neighborhood of the i-th particle the rest $M - 1$ particles are contained.



(a) **(b)**

Fig. 3. (a) Distributed gradient and (b) Particle swarm with robots interaction in an obstacles-free environment, considering a quadratic cost function.

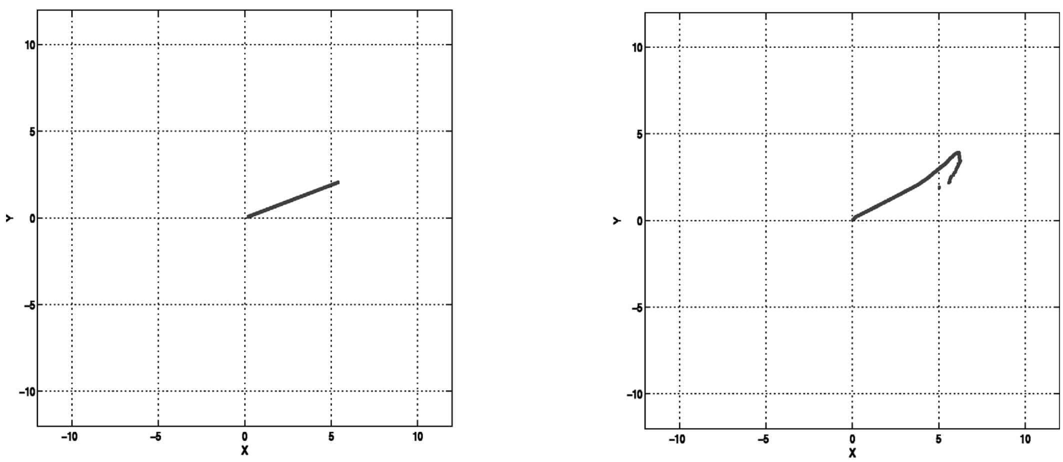
- The condition $V^i(p^k) < V^i(x^i)$ denotes movement in the direction of the negative gradient of the cost function $V^i(x^i)$.

4. Simulation Tests

4.1. Convergence toward the equilibrium

In the conducted simulation tests the multi-robot system consisted of 10 robots which were randomly initialized in the 2-D field. Theoretically, there is no constraint in the number of robots that constitute the robotic swarm. Of course, the number of robots can be increased and if it is sufficiently large then the obtained measurements will be also statistically significant. Two cases were distinguished: (i) motion in an obstacle-free environment (Fig. 3–Fig. 4) and (ii) motion in an environment with obstacles (Fig. 5–Fig. 6). The objective was to lead the robot swarm to the origin $[x_1, x_2] = [0, 0]$. To avoid obstacles, apart from the motion equations given in Sections 2 and 3 repulsive forces between the obstacles and the robots had to be taken into account.

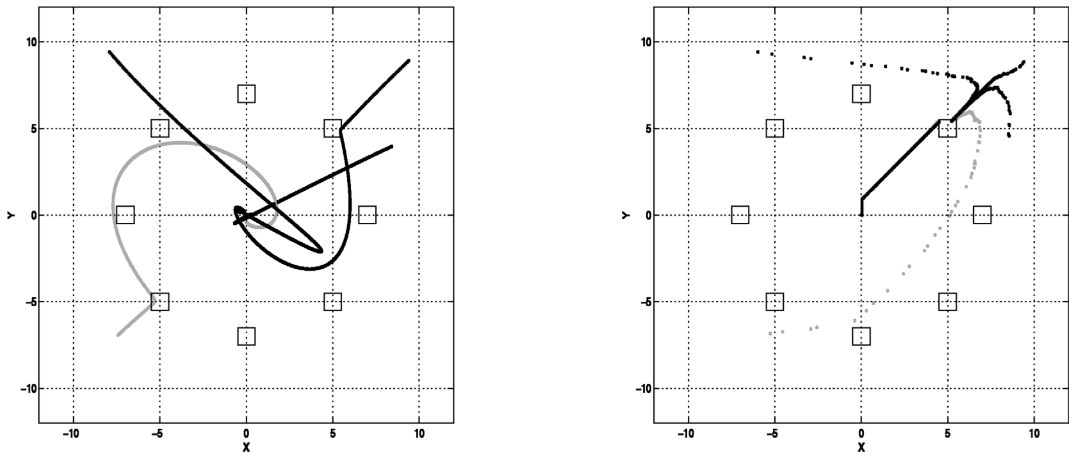
Results about the motion of the robots in an obstacle-free 2D-plane were obtained. Figure 3(a) describes the motion of the individual robots toward the goal state, in an obstacle-free environment, when the distributed gradient algorithm is applied, while Fig. 3(b) shows the motion of the robots in the same environment when particle swarm optimization is used to steer the robots. Figure 4(a) demonstrates how the mean position of the multi-robot formation approaches the goal state when the motion takes place in an obstacle-free environment and the distributed-gradient algorithm is used to steer the robots. Figure 4(b) shows the convergence of the average position of the robotic swarm to the equilibrium $[x^*, y^*] = [0, 0]$, in a 2D-plane without obstacles, when the steering of the robots is the result of particle swarm optimization. Next, motion of the robots in a 2D-plane that contains obstacles is studied. Figure 5(a) demonstrates the motion of the individual robots toward the goal state, in an



(a) **(b)**

Fig. 4. (a) Distributed gradient and (b) Particle swarm with robots interaction in an obstacles-free environment: trajectory of the mean of the multi-robot system.

https://doi.org/10.1017/S0263574707004080 Published online by Cambridge University Press



(a) **(b)**

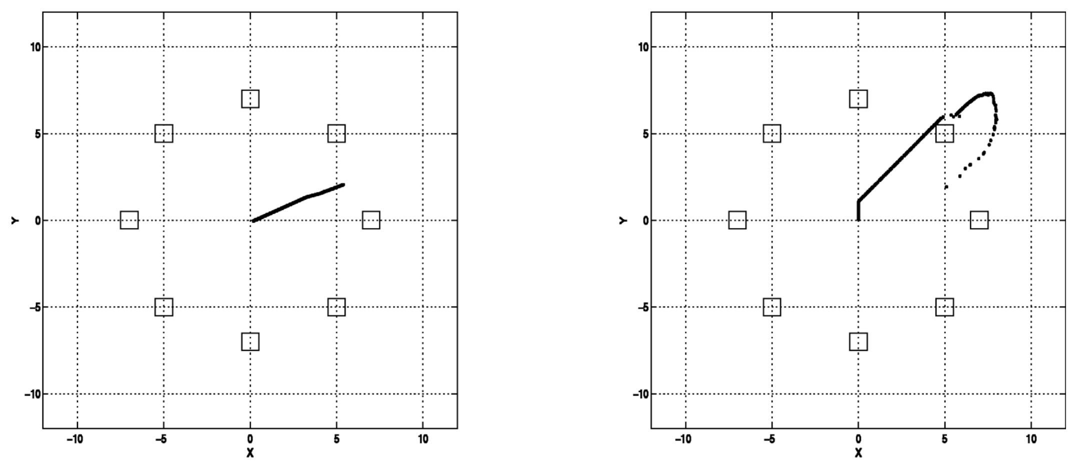
Fig. 5. (a) Distributed gradient and (b) Particle swarm with robots interaction in an environment with obstacles, considering a quadratic cost function.

environment with obstacles, when the steering of the robots is the result of the distributed gradient algorithm. The robots are now also subject to attractive and repulsive forces due to the obstacles. Figure 5(b) presents the convergence of the individual robots to the attractor $[x^*, y^*] = [0, 0]$, when the motion takes place again in an environment with obstacles and when the robots' steering is the result of the particle swarm optimization. Figure 6(a) shows how the average position of the multi-robot formation reaches the equilibrium when the motion is performed in a 2D-plane which contains obstacles and when the robots' trajectories are generated by the distributed gradient algorithm. Finally, in Fig. 6(b) the motion of the mean of the multi-robot system toward the goal state is given, when the robots move again in an environment that contains obstacles, while their paths are generated by the particle swarm optimization algorithm.

When the multi-robot system evolved in an environment with obstacles, the interaction between the individual robots (attractive and repulsive forces) had to be loose, so as to give priority to obstacles avoidance. Therefore coefficients a and b in Eq. (3) were set to small values. The repulsive potential due to the obstacles was calculated by a relation similar to Eq. (3) after substituting x^j with x_o^j, where x_o^j was the center of the j-th obstacle.

In the case of distributed gradient the relative values of the parameters a and b that appear in the attractive and repulsive potential respectively, affected the performance of the algorithm. For $a > b$ the cohesion of the robotic swarm was maintained and abrupt displacements of the individual robots were avoided. In the particle swarm algorithm the area of possible moves round each robot was a Von Neumann one (Fig. 2). It was observed that the ratio $\lambda = \frac{\phi_1}{\phi_2}$ affected the performance of the algorithm. It was observed that large λ resulted in excessive wandering of the robots, while small λ led to the early formation of a robot cluster.

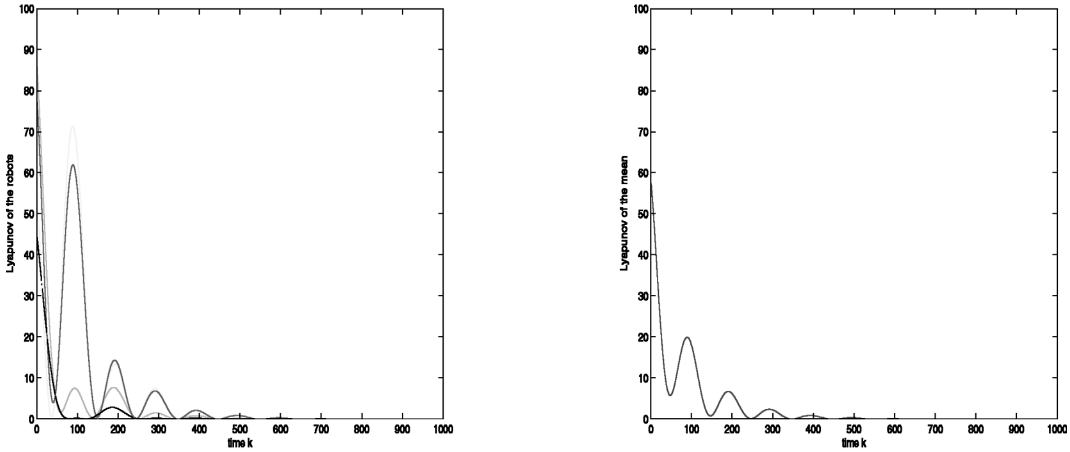
Figure 7 presents the variation of the Lyapunov functions of the robots when the motion takes place in an environments without obstacles, and the distributed gradient approach is



(a) **(b)**

Fig. 6. (a) Distributed gradient and (b) Particle swarm with robots interaction in an obstacles-free environment: trajectory of the mean of the multi-robot system.

https://doi.org/10.1017/S0263574707004080 Published online by Cambridge University Press



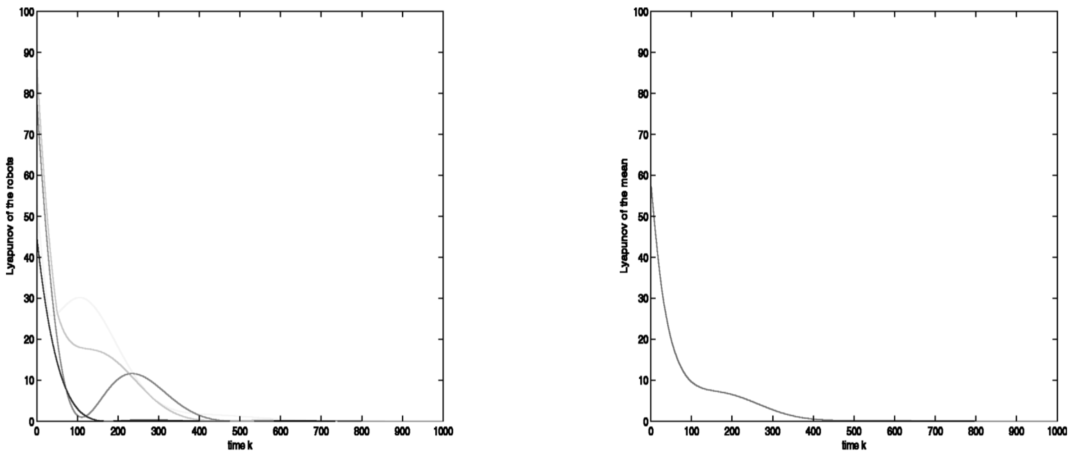
(a) **(b)**

Fig. 7. Distributed gradient approach in an obstacles-free environment: (a) Lyapunov function of the individual robots and (b) Lyapunov function of the mean.

applied to steer the robots toward the attractor. The Lyapunov function is given both in the case of the individual robots and in the case of the mean position of the multi-robot formation. It can be observed that the Lyapunov functions are not monotonous, i.e. and this change of the sign of the Lyapunov function's derivative is due to the fact that the robots' path encircle the target position several times before finally stabilizing at $[x^*, y^*] = [0, 0]$. This means that robots which were initially approaching fast the goal position had to make circles round the attractor in order to wait for those robots which had delayed. This maintained the cohesion of the multi-robot swarm.

Figure 8 presents the variation of the Lyapunov function of the robots when the motion takes place in an environment with obstacles, and the distributed gradient approach is applied to steer the robots toward the attractor. The Lyapunov function is given again both in the case of the individual robots and in the case of the mean position of the multi-robot formation. In that case the Lyapunov functions tend to become monotonous, i.e. continuously decreasing, and this is due to the fact that the interaction forces between the robots have been made weaker after suitable tuning of the coefficients α and β. This enables the robots to approach to the goal state following an almost linear trajectory, since the interaction forces that caused curving of the robots path and encircling of the attractor have now been diminished.

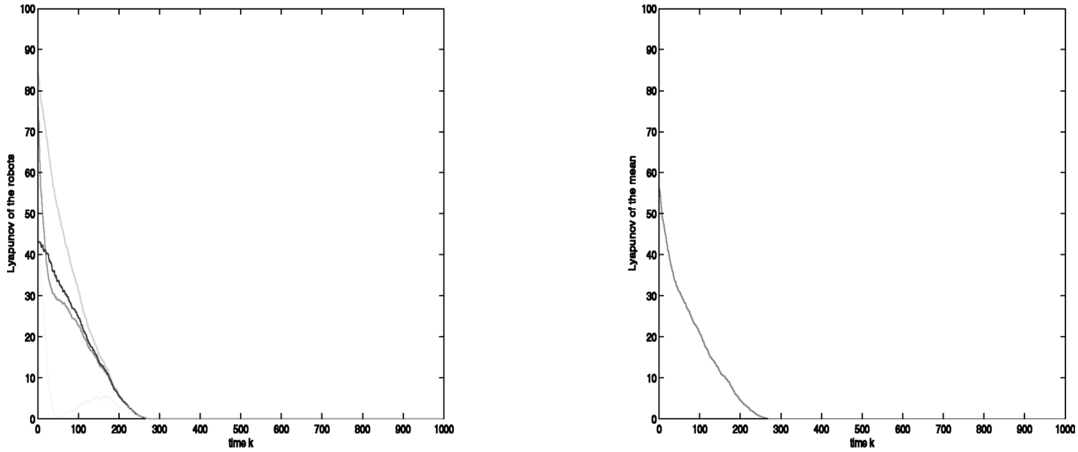
Figure 9 shows the variation of the Lyapunov function of the robots and the Lyapunov function of the mean position of the multi-robot formation, when particle swarm optimization is applied to steer the robots toward the attractor and no obstacles are present in the 2D plane. The Lyapunov functions are monotonically decreasing which is due to the tuning of the interaction forces between the robots. Setting the interaction between the robots at low values enables almost linear convergence toward the goal state, which means that the quadratic error function keeps on decreasing as the motion of the robots continues.



(a) **(b)**

Fig. 8. Distributed gradient approach in an environment with obstacles: (a) Lyapunov function of the individual robots and (b) Lyapunov function of the mean.

https://doi.org/10.1017/S0263574707004080 Published online by Cambridge University Press



(a) **(b)**

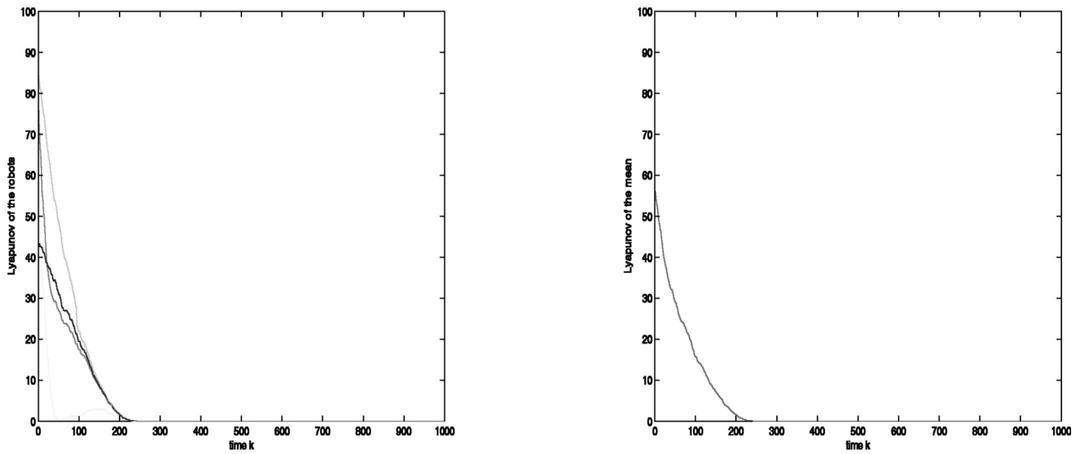
Fig. 9. Particle swarm approach in an obstacles-free environment: (a) Lyapunov function of the individual robots and (b) Lyapunov function of the mean.

Figure 10 shows the variation of the Lyapunov function of the robots and of the Lyapunov function of the average position of the multi-robot system when particle swarm optimization is applied to steer the robots toward the attractor in the presence of obstacles. It can be observed again that the Lyapunov functions decrease monotonically until they become zero, and comparing to the motion in an obstacle-free environment the rate of approach toward the steady state $[e = 0, \dot{e} = 0]$ is larger. This is due to the fact that the interaction between the robots has become looser, thus the robots are enabled to approach rapidly the goal state without waiting for convergence of the rest of the swarm.

Regarding the significance of the mean and the variance of the multi-robot system for evaluating the behavior of the robotics swarm, the following can be stated: although the average position of the multi-robot system is not always meaningful, for instance in case that the individual bypass an obstacle with equal probability from its left or right side, it is a useful parameter that helps to monitor this many-body system. The mean and the variance of the multi-robot formation becomes particularly significant when the motion takes place at nanoscale. In the latter case, all information about the behavior of the nanoparticles swarm is contained in the mean position of the swarm and its variance.

Finally, simulation results about the motion of the multi-robot swarm in a workspace with polyhedric obstacles have been given. Thus additional evaluation for the performance of the proposed motion planning algorithms is obtained. The obstacles considered in the simulation experiments are not points but polyhedra with cover certain regions in the 2D plane. Therefore the attractive and repulsive forces generated between the robots and obstacles affect the robots' trajectories and may also result in local minima. These results are depicted in Fig. 11–Fig. 14.

It can be observed that in the case of motion in a 2D-plane with arbitrarily positioned polyhedric obstacles the distributed gradient algorithm results in smoother trajectories



(a) **(b)**

Fig. 10. Particle swarm approach in an environment with obstacles: (a) Lyapunov function of the individual robots and (b) Lyapunov function of the mean.

https://doi.org/10.1017/S0263574707004080 Published online by Cambridge University Press

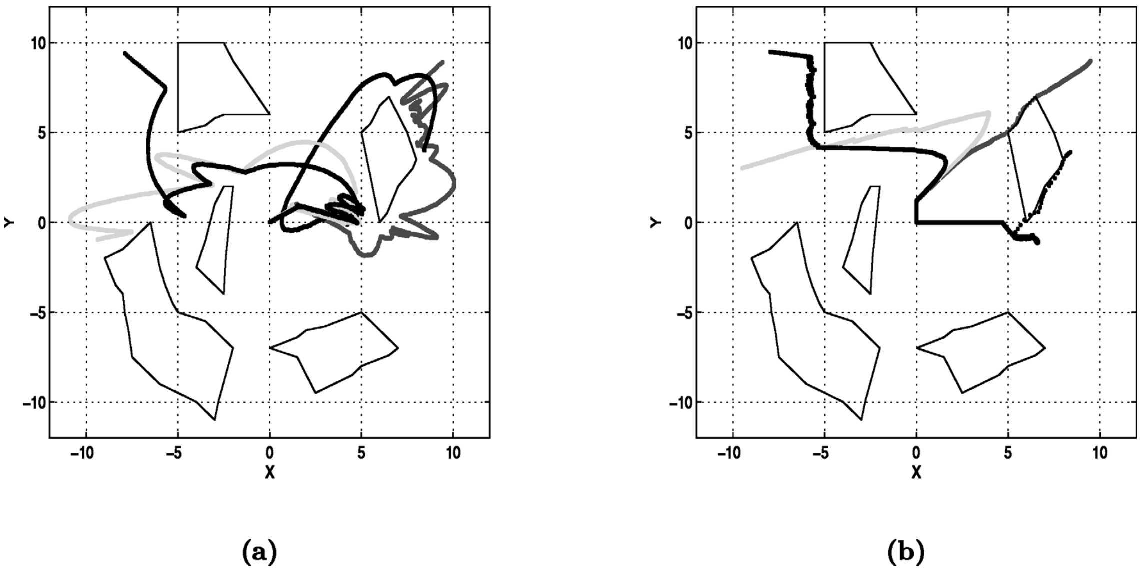


Fig. 11. (a) Distributed gradient and, (b) Particle swarm optimization for motion planning of the multi robot system in a 2D-plane with polyhedric obstacles.

than the particle swarm optimization method. For even though the obstacles are not symmetrically placed in the 2D-plane and thus the aggregate force exerted on the robots in not zero, the multi-robot system in both cases converges to the origin $[x^* = 0, y^* = 0]$. The cost function is is no longer a convex one. Local minima can be generated due to the proximity to obstacles and the attractive forces between the individual robots. For instance, local minima can be found in narrow passages between the obstacles. The inclusion of a stochastic term in the equations that describe the position update of the robots, i.e. in Eq. (1) and Eq. (19) may enable escape from minima minimum. Moreover, suitable tuning of the attractive and repulsive forces that exist between the robots may also permit the robots to move away from local minima.

4.2. Tuning issues and performance of the stochastic search algorithms

Regarding the tuning of the coefficients α and b which appear in Eq. (3) and which affect the trajectories of the individual robots, the following should be noted: coefficient α describes the influence that the attractive potential $V_\alpha(x)$ has one the i-th robot (particle), and coefficient b describes the effect of the repulsive potential $V_r(x)$. These potential terms are respectively given by

$$V_\alpha(x) = \frac{1}{2}\alpha(x^i - x^j)^2, \quad V_r(x) = \frac{1}{2}\sigma^2 b e^{\|\frac{(x^i - x^j)^2}{\sigma^2}\|}. \quad (25)$$

The derivation of $V_\alpha(x)$ with respect to x^i generates an attractive spring force $F_\alpha(x)$ while the derivation of $V_r(x)$

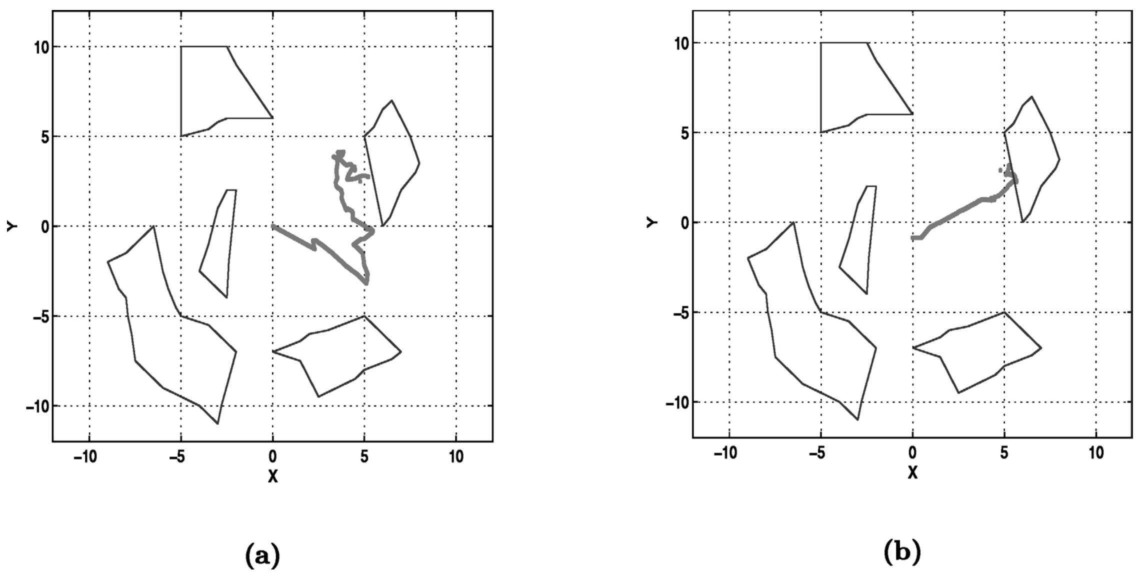


Fig. 12. (a) Distributed gradient, and (ii) Particle swarm optimization in a 2D-plane with polyhedric obstacles: trajectories of the mean of the multi-robot system.

https://doi.org/10.1017/S0263574707004080 Published online by Cambridge University Press

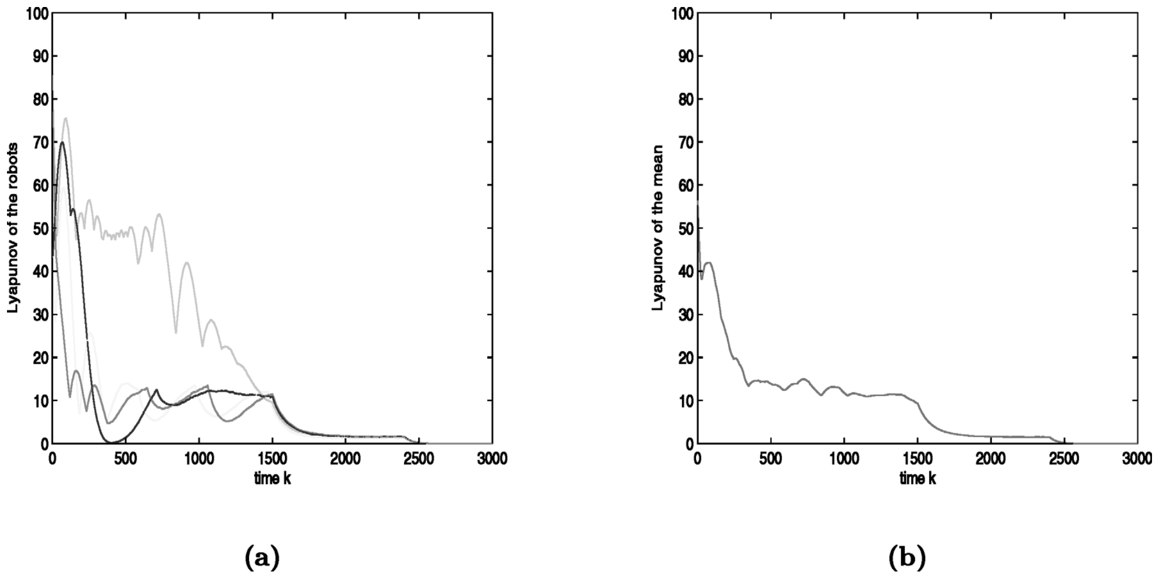


Fig. 13. Distributed gradient in a 2D-plane with polyhedric obstacles: (i) Lyapunov functions of the robots, (ii) Lyapunov function of the mean of the multi-robot formation.

with respect to x^i results in a repulsive force that contains a Gaussian term. These forces are explicitly given by

$$F_\alpha(x) = \nabla V_\alpha(x) = \alpha(x^i - x^j),$$
$$F_r(x) = \nabla V_b(x) = (x^i - x^j) b e^{\| \frac{(x^i - x^j)^2}{\sigma^2} \|}.$$

Similarly a repulsive force can describe the interaction between the i-th particle and the j-th obstacle, where the latter is assumed to be located at position x_o^j. Setting $\alpha \leq b$, or in general choosing the ratio α/b to be much smaller than 1 means that attractive forces between the individual robots are weaker than the repulsive forces. This may delay the convergence of the individual robots to the goal state $[x^*, y^*] = [0, 0]$. One may also notice that the robots' paths become curved and encircle the goal state several times, and in that way the robots remain in sufficient distance.

Regarding the tuning of coefficient $\lambda = \phi_1/\phi_2$ which appears in the particle swarm optimization algorithm, this is performed *ad hoc*. Large values of λ resulted in excessive wandering of the robots, i.e. the robots encircled many times the goal state before finally converging to it. On the other hand small values of λ resulted in an early formation of a robot's cluster, which may delay convergence to the goal state in a motion-plane with obstacles.

Regarding the convergence of the distributed gradient algorithm to the global minimum this can be sure only in the case of a convex cost function. The distributed gradient algorithm risks to be trapped to local minima, however the fact that the search in the solutions space is distributed and the existence of random terms to the

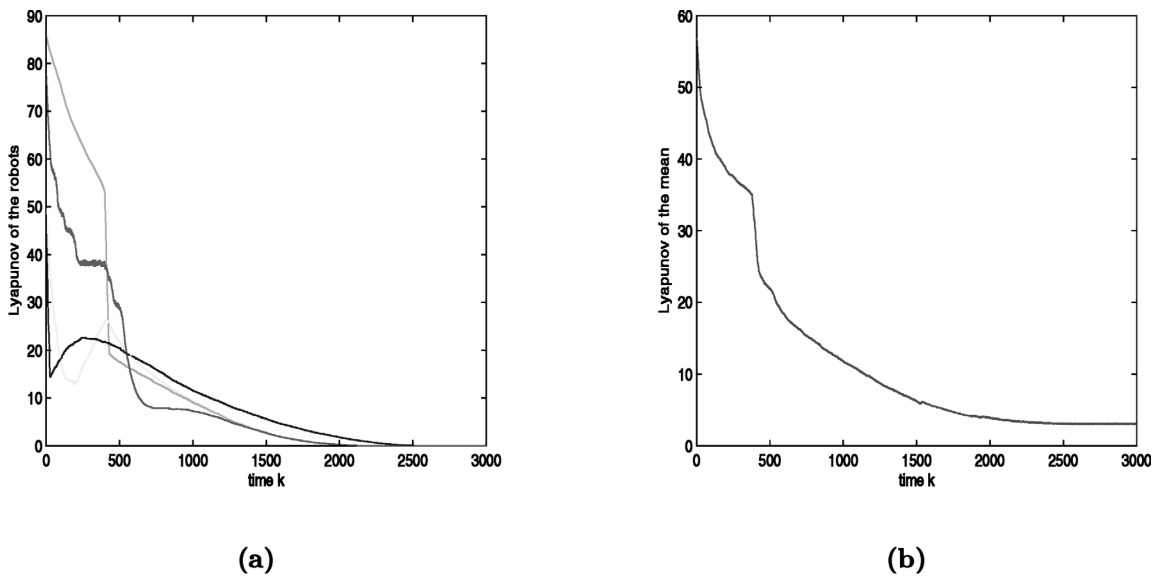


Fig. 14. Particle swarm optimization in a 2D-plane with polyhedric obstacles: (i) Lyapunov functions of the robots, (ii) Lyapunov function of the mean of the multi-robot formation.

https://doi.org/10.1017/S0263574707004080 Published online by Cambridge University Press

individual gradient algorithms make possible the escape from these local minima. The tuning of the coefficients of the distributed gradient algorithm is performed *ad-hoc* and affects the paths in which the individual robots approach the goal state, as well as the shape of the trajectories the robots follow trying to deviate obstacles.

Finally regarding the performance of the proposed motion planning algorithms, distributed gradient appears to be superior than particle swarm optimization for the following reasons: (i) distributed gradient stands for the mathematical formulation of a physical phenomenon (Brownian motion and interaction between diffusing particles) while particle swarm optimization has no direct relation to physics laws, (ii) in distributed gradient convergence to attractors is proved based on Lyapunov stability analysis. It is shown that the mean of the variables updated through the distributed gradient algorithm converges exactly to the equilibrium $[x^*, y^*] = [0, 0]$, while the individual variables stay at a circle of radius ϵ round the attractor, as predicted by LaSalle's theorem. On the other hand there is no strict mathematical proof of the convergence of particle swarm optimization approach, (iii) the fact that the particle swarm optimization method is a derivative-free optimization technique (while distributed gradient required the explicit calculation of derivatives) is moderated by the fact that particle-swarm optimization needs heuristic tuning of several parameters to converge to a fixed point, (iv) Regarding the capability to succeed global optimization, the distributed gradient algorithm is as powerful as the particle swarm optimization approach, since it can contain stochastic terms that enable escape from local minima.

5. Conclusions

In this paper the problem of distributed multi-robot motion planning was studied. A M-robot swarm was considered and the objective was to lead the swarm to a goal position. The kinematic model of the robots was derived using elements of the potential fields theory. The potential of each robot consisted of two terms: (i) the cost V^i due to the distance of the i-th robot from the goal state, (ii) the cost due to the interaction with the other $M - 1$ robots. The differentiation of the potential provided the kinematic model for each robot which was shown to be equivalent to a *distributed gradient* algorithm. The convergence to the goal state was studied with the use of Lyapunov stability theory and LaSalle's theorem. It was proved that in the case of a quadratic cost function V^i the mean position of the multi-robot system converges to the goal state x^* while each robot stays in a bounded area close to x^*.

Moreover, the paper considered a derivative-free technique capable of solving the problem of multi-robot motion planning. The multi-robot system was viewed as a swarm of M particles and the update of the position and velocity of each robot was carried out with the use of particle swarm theory. In that case there was no explicit calculation of the potential function's gradient. The dynamic behavior of the particle swarm was studied with the use of ordinary differential equations. Appropriate tuning of the differential equation's coefficients can assure that the particles velocity will converge asymptotically to zero.

Distributed gradient and particle swarm theory for multi-robot motion planning were evaluated through simulation tests. It was observed that when the multi-robot system was evolving in an environment with obstacles, the interaction between the individual robots (attractive and repulsive forces) had to be loose, so as to give priority to obstacles avoidance. Both methods succeeded cooperative behavior of the robots without requirement for explicit coordination or communication. The performance of both methods was satisfactory, however distributed gradient was evaluated to have advantages over the particle swarm optimization mainly due to its sound convergence proof and the smooth motion patterns it produced in various environments.

Acknowledgments

The author would like to thank Université Paris XI – Institut d' Électronique Fondamentale, for research cooperation in 2007.

References

1. R. Groß, M. Bonani, F. Mondada and M. Dorigo, "Autonomous self-assembly in swarm-bots," *IEEE Trans. Robot.* **22**(6), 1115–1130 (2006).
2. G. G. Rigatos, S. G. Tzafestas and G. J. Evangelidis, "Reactive Parking Control of a non-holonomic vehicle via a fuzzy learning automaton," *IEE Procceedings on Control Theory and Applications* **148**(2), 169–180 (2001).
3. B. E. Bishop, "On the use of redundant manipulator techniques for control of platoons of cooperative robotic vehicles," *IEEE Trans. syst. Man Cybern.–Part A*, **33**(5), 608–615 (2003).
4. Y. Hong, L. Gao, D. Cheng and J. Hu, "Luapunov-based approach to multi-agent systems with switching jointly connected interconnection," *IEEE Trans. Autom. Control* **52**(5), 943–948 (2007).
5. Y. Guo and L. E. Parker, A distributed and optimal motion planning approach for multiple mobile robots, *In: Procceedings 2002 IEEE International Conference on Robotics and Automation*, Washington DC, (May 2002), pp. 2612–2619.
6. A. Sinha and D. Ghose, "Generalization of linear cyclic pursuit with application to rendezvous of multiple autonomous agents," *IEEE Trans. Autom. Control* **51**(11), 1819–1824 (2006).
7. E. Pagello, A. D' Angelo and E. Menegatti, "Cooperation issues and distributed sensing for multi-robot systems," *Proceedings of the IEEE* **94**(7), 1–12 (2006) 1370–1383.
8. R. Sepulchre, D. A. Paley and N. E. Leonard, "Stabilization of planar collective motion: all to all communication," *IEEE Trans. Autom. Control* **52**(5), 811–824 (2007).
9. O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.* **5**(1) 90–99 (1986).
10. E. Rimon and D. E. Koditscheck, "Exact robot navigation using artifical potential functions," *IEEE Trans. Robot. Autom.* **8**, 501–518 (1991).
11. S. A. Masoud and A. A. Masoud, "Motion planning in the presence of directional and regional avoidance constraints using nonlinear, nisotropic, harmonic potential fields: a physical metaphor," *IEEE Trans. Syst. Man Cybern.—Part A* **32**(6), 705–723 (2002).
12. J. H. Reif and H. Wang, "Social potential fields: A distributed behavioral control for autonomous robots," Robot. Auton. Syst., *Elsevier*, **27**, 171–194 (1999).
13. F. Comets and T. Meyre, "Calcul stochastique et modèles de diffusions," *Dunod*, Paris, 2006.
14. G. G. Rigatos, A. P. Tzes and S. G. Tzafestas, "Distributed Stochastic Search for Multi-robot Cooperative Behavior,

IMACS 2005 International Conference, Paris, France (July 2005).

15. W. Jatmitko, K. Sekiyama and T. Fukuda, “A PSO-based mobile robot for odour source localization in dynamic advection-diffusion with obstacles environment,” *IEEE Comput. Intell. Mag.* **2**(2), 37–42 (2007).
16. H. Levine and W. J. Rappel, “Self-organization in systems of self-propelled particles,” *Phys. Rev. E* **63** (2000).
17. M. Clerk and J. Kennedy, “The Particle Swarm-Explosion, Stability, and Convergence in a Multidimensional Complex Space,” *IEEE Trans. Evol. Comput.* **6**(1), 58–73, (2002).
18. M. Duflo, Algorithmes stochastiques, Mathématiques et Applications *Springer* (1996).
19. A. Benveniste, M. Métivier and P. Priouret, “Adaptive algorithms and stochastic approximations,” *Springer* (1990).
20. V. Gazi and K. Passino, “Stability analysis of social foraging swarms,” IEEE Trans. Sys. Man Cybern.—Part B, *Cybernetics*, **34**(1), 539–557 (2004).
21. H. Khalil, *Nonlinear Systems* (*Prentice Hall*), 1996.

https://doi.org/10.1017/S0263574707004080 Published online by Cambridge University Press