Decision and Finite-Time Cooperative Control of Multi-Robot Systems for Odour Source Localisation

Qiang Lu

Dissertation for Ph.D. Degree

Centre for Intelligent and Networked Systems

Central Queensland University

July 2013

Approved for Submission by Principal Supervisor Prof. Qing-Long Han

Declaration

The work contained in this dissertation has not been previously submitted either in whole or in part for a degree at Central Queensland University or any other tertiary institution. To the best of my knowledge and belief, the material presented in this dissertation is original except where due reference is made in text.

Qiang Lu

10 July 2013

Abstract

In the last two decades, the problem of odour source localisation has been widely studied by using a single robot due to its practical significance for human security, such as searching for the source of toxic gas leakage and locating the origin of a fire at its initial stage. Recently, using a multi-robot system to locate the odour source has received increasing interest from researchers because of several major benefits over a single robot such as a wider detection range and multiple detection information. In this dissertation, two aspects on the odour source localisation problem are studied; one is to learn from data detected and collected by the multi-robot system to make a decision on the position of the odour source while the other is to coordinate and control the multi-robot system to locate the source of odour in terms of the decision results. In particular, a distributed coordination control architecture including two levels: a decision level and a control level, is designed. In the decision level, a new distributed decision algorithm, which can make a decision on the position of the odour source, is formulated. In the control level, a particle swarm optimization based finite-time motion control algorithm, a consensus-based finite-time motion control algorithm, and a potential-based finite-time motion control algorithm, are then developed to control the robot group to locate the odour source, respectively. Finally, the effectiveness of the proposed solutions consisting of the architecture, the decision algorithm, and the control algorithms for odour source localisation is illustrated through simulations.

Acknowledgements

First and foremost, I would like to thank my Principal Supervisor, Professor Qing-Long Han, for his assistance and guidance over the past four years. He is not only a great mentor but also a good friend to me. His patience and encouragement help me to finish this dissertation. And his suggestions and insights help me solve challenging research problems. I also wish to thank my Associate Supervisor, Professor Shirong Liu, for his helpful suggestions in the process of the dissertation writing. He is always willing to give me suggestions and support.

In terms of technical guidance and advice, I would like to thank Dr. Chai Li, Dr. Xie-Fu Jiang, Dr. Chen Peng. I am also very grateful to the brave souls who provide valuable feedback: I would like to thank Dr. Dennis Jarvis, Dr. Xianming Zhang, Dr. Ke Ding, Dr. Baolin Zhang, and Dr. Yulong Wang. I am particularly grateful to Dr. Dawei Zhang, Mrs. Jia Wang, Mr. Yufeng Lin, Mr. Yanpeng Guan, Mr. Xiaohua Ge, Dr. Gang Liu, and Mr. Shu Chen for their friendship.

Last but not least, I would like to thank my wife for her love and support. Without her support for the family, I cannot complete this dissertation.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In nature, there exist some biological entities such as moths and lobsters, that use their olfactory sense to detect and react to information in their environment [90, 16, 139, 153, 152]. Odour source localisation originates from the behaviors of such biological entities, and can be seen in the foraging by lobsters and mate-seeking by moths. Due to its practical significance for human security such as searching for the source of toxic gas leakage and locating the origin of a fire at its initial stage, the problem of odour source localisation has received much attention from researchers and engineers [4, 49, 37, 43, 88, 137].

In the last two decades, the problem of odour source localisation has been widely studied by using *a single robot system.* Due to several major benefits over a single robot such as a wider detection range and multiple detection information, using *a multi-robot system* to locate the odour source has received increasing interest from researchers. In this dissertation, two aspects on the problem of odour source localisation are studied; one is to learn from data detected and collected by the multi-robot system to **make a decision** on the position of the odour source while the other is to **coordinate and control the multi-robot system** to locate the source of odour in terms of the decision results.

1.1 Background

1.1.1 Problem Definition

Odour source localisation is a type of ill-posed and dynamical optimization problem, which can be stated as

Problem 1 (Odour Source Localisation)**.** *An odour source localisation problem consists of*

- *A set $\mathcal{N}$ of N mobile robots or vehicles,*
- *A set $\mathcal{X}$ of positions in a two-dimensional search space $\mathbb{R}^2$,*
- *A set $\mathcal{M} \subseteq \mathcal{N} \times \mathcal{X}$ of possible pairs, and*
- *A map $f : \mathcal{X} \times [0, \infty] \to \mathbb{R}$ giving a time-varying concentration value for each position.*

A feasible (suboptimal) solution is a set of pairs $\mathcal{S} \subseteq \mathcal{M}$ such that there exists at least a position x_i ($i \in \mathcal{N}, x_i \in \mathcal{X}$, $(i, x_i) \in \mathcal{S}$), which satisfies that position x_i lies in the vicinity of the odour source, i.e. $\|x_i - x_s\|_2 < \epsilon$ (the symbol $\| \cdot \|_2$ refers to an Euclidean vector norm; $x_s \in \mathcal{X}$ denotes the position of the odour source; and the choice of ϵ depends on applications). A feasible solution S is optimal if the position x_i is the position of the odour source, i.e. $\|x_i - x_s\| = 0$.

The problem of odour source localisation $f(x_i, t)$ ($x_i \in \mathcal{X}$, $t \in [0, \infty]$) is illustrated in Figure 1.1 where the odour source marked by the yellow circle releases a great deal of filaments (a group of odour molecules) that form a plume shaped by wind (the arrow denotes the direction of wind and the length of the arrow denotes the speed of wind). The position of the odour source is located at the (10m,0m) and the search area is limited at 100m × 100m. From this figure, one can see that, compared with a single robot [76, 29, 111], a multi-robot system obviously provides

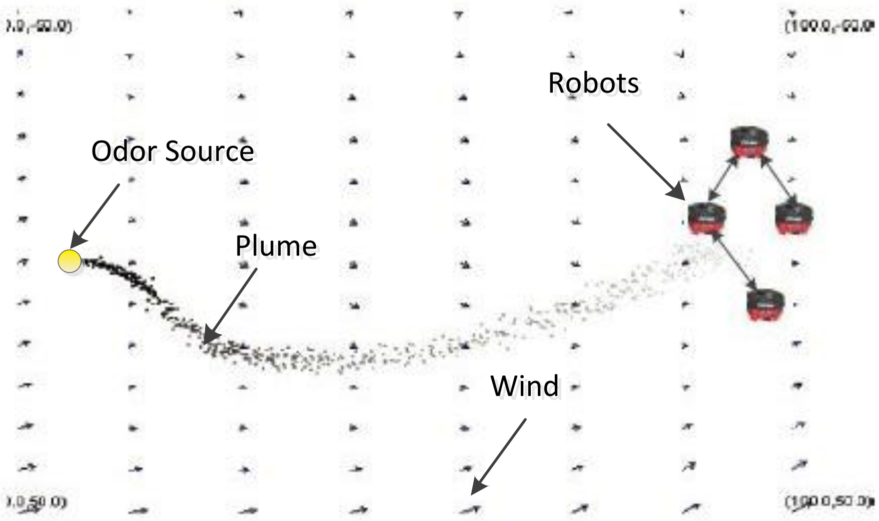


Figure 1.1: An instantaneous plume where black dots denote the filaments that form a plume and arrows denote the wind speed and direction.

a wider detection region to capture the time-varying plume. Moreover, four main characteristics of the problem of odour source localisation are summarized by

1. There exist a global concentration maximum, which is located in the vicinity of the position of the odour source, and multiple local concentration maxima along the plume;
2. The positions with local concentration maxima are time-varying due to the influence of wind and dispersion of odour molecules;
3. Odour concentration occurs within a narrow region at each time, that is, at a given time t, the odour dispersion region $\mathcal{X}$ can be partitioned into two parts: $\mathcal{X}_1$ and $\mathcal{X}_2$ ($\mathcal{X} = \mathcal{X}_1 \bigcup \mathcal{X}_2$) where $f(x_i, t) > 0$, if $x_i \in \mathcal{X}_1$, and $f(x_i, t) = 0$, if $x_i \in \mathcal{X}_2$.
4. Wind direction and speed are time-varying.

Therefore, the characteristics of the problem will necessarily influence the design of the decision algorithms and controllers of the multi-robot systems. Consequently, we will seek the new theories, methods, and technologies to deal with the problem of odour source localisation. In order to validate the effectiveness of the proposed solutions for odour source localisation, we need to build a simulation environment whose kernel is an odour dispersion model.

1.1.2 Odour Dispersion Model

In terms of the odour dispersion studies [90, 16, 139, 153, 152], Farrell *et al.* (2002) [28] modeled the movement of filaments released by the odour source (see [28] for details), which is given by

$$\dot{p}_i = v_a + v_i \tag{1.1}$$

where $p_i = (x, y)$ is the ith filament location; $v_i = (v_{ix}, v_{iy})$ presents the dispersion velocity of the ith filament relative to the centerline and can be modeled by using a white noise process with a given spectral density $(\sigma_{ix},\ \sigma_{iy})$; and $v_a = (v_{ax}, v_{ay})$ denotes the advection velocity, which is calculated by

$$\frac{\partial v_{ax}}{\partial t} = -v_{ax}\frac{\partial v_{ax}}{\partial x} - v_{ay}\frac{\partial v_{ax}}{\partial y} + \frac{1}{2}K_x\frac{\partial^2 v_{ax}}{\partial x^2} + \frac{1}{2}K_y\frac{\partial^2 v_{ax}}{\partial y^2} \tag{1.2}$$

$$\frac{\partial v_{ay}}{\partial t} = -v_{ax}\frac{\partial v_{ay}}{\partial x} - v_{ay}\frac{\partial v_{ay}}{\partial y} + \frac{1}{2}K_x\frac{\partial^2 v_{ay}}{\partial x^2} + \frac{1}{2}K_y\frac{\partial^2 v_{ay}}{\partial y^2} \tag{1.3}$$

where K_x and K_y represent diffusivity and are assumed to be the constant.

Moreover, the size of the ith filament varies over time. Hence, the growth rate of the ith filament is described by

$$\frac{\mathrm{d}R_i}{\mathrm{d}t} = \frac{3}{2}\gamma R_i(0)^{\frac{1}{3}} \tag{1.4}$$

where $R_i(0)$ is the ith filament's size at time $t = 0$; and γ is a parameter.

The instantaneous chemical concentration at the position $z = (x, y)$ can be computed by

$$C(z,t) = \sum_{i=1}^{N} C_i(z,t) \tag{1.5}$$

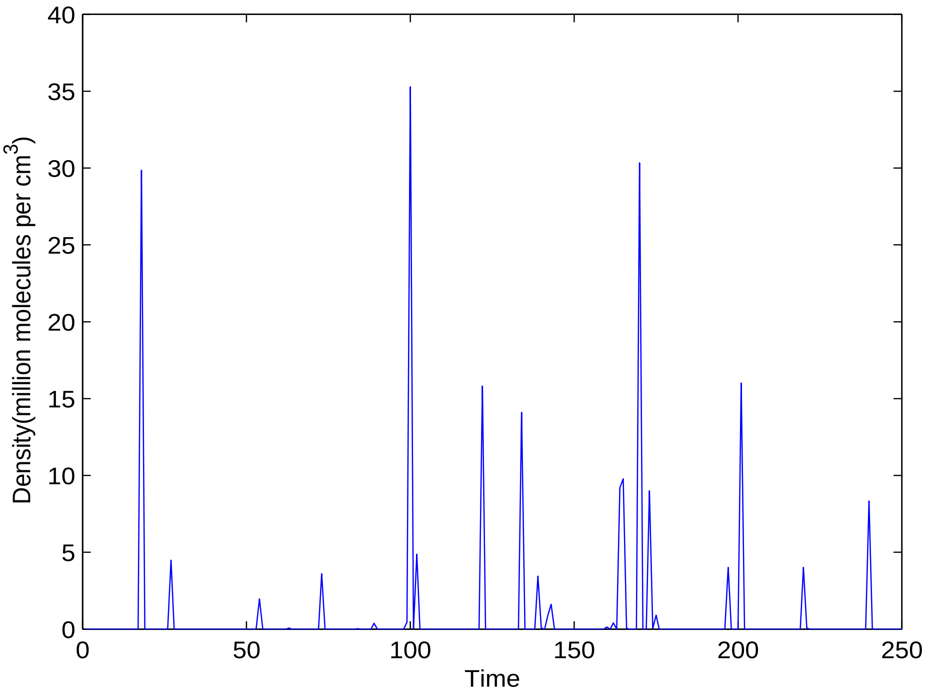


Figure 1.2: Concentration fluctuates at a concrete position (80m,0m) from $t = 0$ to $t = 250$.

where N is the number of filaments; $C_i(z,t)$ is the concentration of the ith filament at the location z and is obtained by

$$\begin{aligned} C_i(z,t) &= \frac{Q}{\sqrt{8\pi^3}R_i^3(t)}\exp\left(\frac{-r_i^2(t)}{R_i^2(t)}\right) \\ r_i(t) &= \|z - p_i(t)\|_2 \end{aligned}$$

where Q is the amount of the molecules per filament; $R_i(t)$ is the ith filament's size at time t; and $\exp(\cdot)$ refers to the exponential function e.

Based on (1.1), (1.2), (1.3), and (1.4), the position and size of the ith filament can be calculated at any time t, and the chemical concentration at any position can also be obtained in (1.5). By using this model, Figure 1.1 shows the shape of an instantaneous plume, which consists of filaments released by an odour source. Figure 1.2 shows that the density at a concrete position (80m,0m) fluctuates over time from $t = 0$ to $t = 250$. Hence, from two figures, one can see that the odour

Table 1.1: The parameters of the Farrell's odour dispersion model.

Variables	Parameters
Area(m×m)	100 × 100
Odour source position(m)	(10,0), (10,15),(10,-15)
Q	5123.7618
K_x, K_y	10,10
Growth rate	0.001
Initial wind velocity(m/s)	S(0.8,0), M(1,0),L(1.5,0)

model proposed by Farrell *et al.* (2002) [28] simulates the dynamical movement process of the filaments. Moreover, one can also see that the chemical concentration at a concrete position is time-varying, multimodal, and discontinuous. Therefore, in this dissertation, we will use the Farrell's odour dispersion model to build the simulation environment whose parameters are shown in Table 1.1 where "S", "M", and "L" denote the small initial wind speed, the medium initial wind speed, and the large initial wind speed, respectively.

1.1.3 Dynamics of Differentially Driven Mobile Robots

In order to search for and locate the odour source, we will use the differentially driven mobile robots, such as P3-DX (www.mobilerobots.com) shown in Figure 1.3. This type of robot will be employed to validate the proposed decision algorithms and controllers. The dynamics can be described by

$$\begin{pmatrix} \dot{r}_{xi} \\ \dot{r}_{yi} \\ \dot{\theta}_i \\ \dot{\nu}_i \\ \dot{\omega}_i \end{pmatrix} = \begin{pmatrix} \nu_i \mathrm{cos}\theta_i \\ \nu_i \mathrm{sin}\theta_i \\ \omega_i \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{m_i} & 0 \\ 0 & \frac{1}{J_i} \end{pmatrix} \begin{pmatrix} F_i \\ \tau_i \end{pmatrix} \tag{1.6}$$

where $r_i = (r_{xi}, r_{yi})^T$ is the position of the ith ($i \in \{1, 2, \ldots, N\}$) robot; θ_i denotes the orientation; ν_i is the linear velocity; ω_i is the angular velocity; τ_i is the torque; F_i is the force; m_i is the mass; and J_i is the moment of inertia. Let $y_i = (r_i^T, \theta_i, \nu_i, \omega_i)^T$ be the state of the ith robot and $p_i = (F_i, \tau_i)^T$ be the control input.

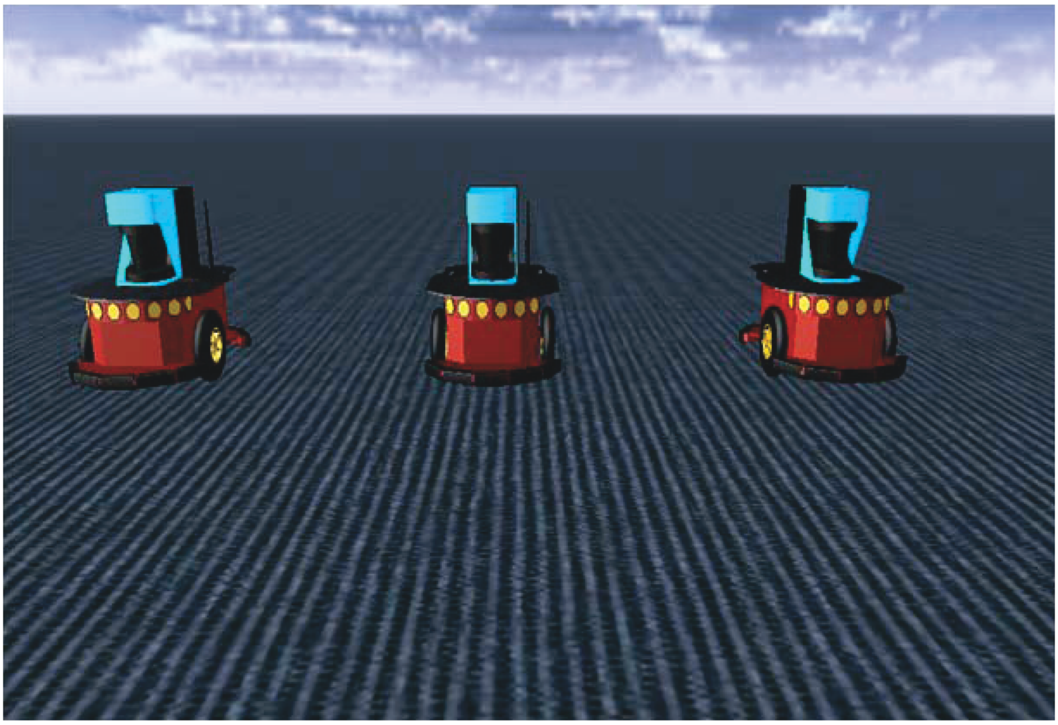


Figure 1.3: P3-DX mobile robots.

Because the nonholonomic systems cannot be stabilized with continuous static state feedback, we make use of the "hand position" instead of the "center position" of the robot (see details in [68, 118]). Let (1.7) be the dynamics of the "hand position" of the robot, and L_i be a distance between the "hand position" and the "center position" along the line that is perpendicular to the wheel axis [68, 118].

$$\left\{ \begin{array}{l} \dot{x}_i = v_i \\ \dot{v}_i = u_i \quad i \in \{1, 2, \ldots, N\} \end{array} \right. \tag{1.7}$$

where x_i and v_i, respectively, denote the position and velocity for the robot i at the "hand position"; and u_i is a control signal for the robot i at the "hand position". The variable relationship between the "hand position" and the "center position" is shown in

$$x_i = r_i + L_i \begin{pmatrix} \cos\theta_i \\ \sin\theta_i \end{pmatrix} \tag{1.8}$$

$$v_i = \begin{pmatrix} \cos\theta_i & -L_i\sin\theta \\ \sin\theta_i & L_i\cos\theta_i \end{pmatrix} \begin{pmatrix} \nu_i \\ \omega_i \end{pmatrix} \tag{1.9}$$

In the light of (1.8) and (1.9), we can obtain the position and velocity of the

Table 1.2: The parameters of P3-DX mobile robots in the simulations.

m_i (kg)	L_i (m)	J_i (kg m^2)	b (m)	l (m)	J_{wheel} (kg m^2)	$\tau_{l,r}^{max}$	$\tau_{l,r}^{min}$
9	0.12	0.2	0.079	0.381	0.07	0.8	-0.8

"hand position" of the robot, and then calculate the control law u_i for the double-integrator system (1.7). Finally, we obtain the control input (1.10) for the system (1.6) [68] as

$$p_i = \begin{pmatrix} \frac{1}{m_i}\cos\theta_i & -\frac{L_i}{J_i}\sin\theta_i \\ \frac{1}{m_i}\sin\theta_i & \frac{L_i}{J_i}\cos\theta_i \end{pmatrix}^{-1} \left[u_i - \begin{pmatrix} -\nu_i\omega_i\sin\theta_i - L_i\omega_i^2\cos\theta_i \\ \nu_i\omega_i\cos\theta_i - L_i\omega_i^2 sin\theta_i \end{pmatrix} \right] \tag{1.10}$$

Usually, the applied torques for left wheel and right wheel are calculated by

$$\tau_l = \frac{J_{wheel}}{b}\left(\frac{F_i}{m_i} - \frac{\tau_i l}{2J_i}\right) \tag{1.11}$$

$$\tau_r = \frac{J_{wheel}}{b}\left(\frac{F_i}{m_i} + \frac{\tau_i l}{2J_i}\right) \tag{1.12}$$

where b is the radius of the wheel; l denotes the axis length between two wheels; J_{wheel} is a moment of inertia of the wheel; τ_l and τ_r refer to the applied torques for the left wheel and right wheel, respectively. The parameters that are used for P3-DX are shown in Table 1.2 where the maximum output torque $\tau_{l,r}^{max}$ of the motors of the left and right wheels is limited as 0.8 in the anticlockwise direction. Correspondingly, in the clockwise direction, the maximum output torque of the motors of the left and right wheels is also 0.8. But, from the the anticlockwise direction, the minimum output torque $\tau_{l,r}^{min}$ of the motors of the left and right wheels is -0.8. It is worth mentioning that the maximum and minimum torques have been defined in Microsoft Robotics Studio in order to control the P3-DX robots.

1.1.4 Odour Source Localisation Using a Single Robot

For this case, i.e. odour source localisation using a single robot, the major approaches that have been reported are chemotaxis [50, 43], anemotaxis [80, 44], and the odour source mapping method [29, 111].

An initial approach is to calculate a concentration gradient of odour in order to orient the robot's movement to search for the odour source, which is called chemotaxis [66, 125, 48]. In [43], for example, several chemotaxis-based algorithms were employed to control a robot lobster to track a statistically characterized turbulent plume by using two concentration sensors. The research results showed that these algorithms were effective only in a narrow search region. In [120], the search results of three chemotaxis algorithms observed in the bacterium *E. coli* [46], the silkworm moth *Bombyx mori* [57], and the dung beetle *Geotrupes stercorarius* [50] were reported and showed that the *E. coli* algorithm may only be useful in a very small scale where the plume shape is dominated by the diffusion of molecules. The *Bombyx mori* algorithm can be applied in the environment where the chemical plume rapidly fluctuates, and the *Geotrupes stercorarius* algorithm can work well for the strong plumes. In [88, 89], three biologically inspired search strategies including chemotaxis, biased random walk, and a combination of chemotaxis and biased random walk, were proposed. The experimental results showed that chemotaxis is a more efficient but a less robust strategy than the biased random walk method. The combined approach is the most efficient one among the three strategies. From the aforementioned research results, one can conclude that chemotaxis is not feasible in an environment where the evolution of the chemical distribution is dominated by turbulence. As a result, the current research based on a single robot mainly focuses on anemotaxis and the odour source mapping method.

In order to enable the robot to locate the odour source in a turbulent environment, anemotaxis, where wind information is used to guide the robot, was proposed [80, 44]. In [75, 76, 78, 27], a strategy was designed by simulating the manoeuvres of moths flying upwind along a pheromone plume. Briefly speaking, instead of moving directly upwind, the robot uses a counter-turn behavior across the wind for tracing the plume. In [123, 124, 121, 122], a strategy, which is similar to the counter-turn behavior observed in moths and is called chemo-kino-kinesis, was also designed. In this approach, the velocity of robot is decomposed into two components: a component normal to the wind direction and another component tangential to the wind direction. By controlling the turn rate of the component normal to the wind direction, a zigzag behavior can be obtained. Different from the zigzag behavior, spiral-surge is another moth-inspired anemotaxis approach. In [37, 35, 34, 36, 44], the robot moved along spirals, which can be reset based on the concentration detected. However, it is worth mentioning that the speed and heading of robots are generated by using instantaneous sensor readings. Once the search task fails, the robot cannot provide any information about the position of the odour source. In order to deal with this issue, the odour source mapping approach, which not only makes use of the instantaneous sensor information but also utilizes historical sensor information, was proposed [29, 111].

For the odour source mapping approach, a source-likelihood map is estimated and a path between the current position and the predicted position of the odour source, which is most likely to detect chemical concentrations, is planned based on this map. The robot moves along the path to the predicted location of the odour source. During moving to the predicted position, the sensor readings are used to update the source-likelihood map, and then a path is planned again. Finally, the location of the odour source is declared according to the source-likelihood map. For example, Farrell *et al.* (2003) [29] developed a hidden Markov based search algorithm where the location of the odour source is predicted and a most likely path taken by odour to a given location is designed. In [111], the results in [29] were improved. Pang *et al* (2006) [111] proposed a new source-likelihood mapping approach, which is derived based on a stochastic process theory and a Bayesian inference method. According to the measured wind information and the detection or undetection events, the location of the odour source can be predicted. However, the odour source mapping approach requires that the robot is equipped with devices that have better sensing capabilities and stronger computational capabilities.

1.1.5 Odour Source Localisation Using a Multi-Robot System

Because the multi-robot system can detect a wider region to capture the time-varying plume and obtain multiple detection information [19, 56, 68, 30, 65, 20], how to locate the odour source using a *multi-robot system* has received increasing interest from researchers [70, 148, 69, 148, 149, 134]. For example, Hayes *et al.* (2002) [44] extended the spiral surge algorithm by using several simple types of communication such that the extended spiral surge algorithm can be used to coordinate a multi-robot system to search for the plume and locate the source of odour. The overall search process is divided into two parts, namely, *plume finding* and *plume tracing*. For *plume finding*, the outward spiral search pattern allows for the coverage of the local space and provides an effective search for odour clues. For *plume tracing*, the robot samples the wind direction and moves upwind for a set distance to locate the odour source. Marques *et al.* (2002) [91, 92] used a genetic algorithm to coordinate a group of mobile robots with random initial positions to locate the source of odour. Hayes *et al.* (2003) [45] proposed a swarm intelligence based distributed search algorithm where a team of robots can locate the odour source more efficiently than a single robot. It is worth noting that the swarm intelligence techniques used in [45] involve the study of the collective behaviors of biological groups found abundantly in nature, including ant colonies, bird flocking, and fish schooling [98, 60].

The particle swarm optimization (PSO) algorithm, as one of the swarm intelligence techniques, has been studied in the domain of cooperative search [113, 114, 115, 112, 91] and been used to coordinate a multi-robot system to deal with the odour source localisation problem [93, 53, 52]. To search for an odour source, for instance, Marques *et al.* (2006) [93] used two search strategies: a global search strategy and a local search strategy. In the global search strategy, robots cross wind in a search space to search for odour cues. In the local search strategy, the PSO algorithm is used to coordinate a multi-robot system to locate the source of odour. To avoid trapping in local maximal concentrations, Jatmiko *et al.* (2007) [53] improved the commonly used PSO algorithm through analogy with an electrical charge theory. In the improved algorithm (CPSO), two types of robots (neutral and charged robots) are used. Among neutral robots, there is no repulsive force while among charged robots a mutual repulsive force is generated in order to maintain positional diversity of robots. To use the PSO algorithm for odour source localisation, Lu and Han (2010) [81] proposed a distributed coordination control architecture where the PSO algorithm is divided into three parts (prediction, plan, and control). Accordingly, the cooperative control system consists of three levels: a group level, a trajectory level, and a robot level. In the group level, swarm information and individual information are used to predict the probable position of the odour source. In the trajectory level, a movement trajectory of the robot is planned from the current position to the probable position of the odour source. In the robot level, a control law is designed to enable the robot to move along the planned trajectory. This control architecture makes the control system robust and evolvable [42]. In terms of this control architecture, the search performance of the robot group coordinated by the CPSO algorithm [53] is improved. To quickly locate the odour source, Lu and Han (2011) [85] proposed a probability particle swarm optimization with an information-sharing mechanism. Because of the introduction of a distribution estimation algorithm, each robot can be provided an opportunity to choose an appropriate position in the search space such that the search performance of the robot group can be improved. However, the PSO algorithm is a kind of the concentration-based approach [93, 53, 81, 85] . One can see that the local convergence of the multi-robot system coordinated by the PSO algorithm often appears in the environment characterized by high Reynolds numbers [86, 81]. Moreover, the search results of the multi-robot system are always significantly influenced by the detection precision of concentration sensors equipped on mobile robots [29, 111].

It should be pointed out that the decision on the position of the odour source is the first and important step for odour source localisation. But, the position of the odour source is not directly obtained. The PSO algorithm only presents a kind of current understanding on the position of the odour source such that the error between the decided position and the real position of the odour source is bigger, especially in the initial search stage. Even if several researchers utilize wind information to estimate the position of the odour source, the computational cost of the proposed approaches is also higher [29, 111]. Therefore, how to design a distributed decision algorithm with "computational simplicity" is the first motivation of this dissertation.

Moreover, the controller design should satisfy the several requirements in the dynamical search environment. From the perspective of control engineering, the developed controller is required to be better disturbance rejection and robustness against uncertainties. Moreover, the developed controller is also required to be better tracking performance. From the perspective of characteristics of odour source localisation, in order to adapt the dynamical search environment, the developed controller is required to enable the system to be convergence within a finite-time interval. Furthermore, there exists a mutual influence between decision and control, which originates from the unknown search environment. In other words, the knowledge about the search environment is only from the current and previous information of the robot group. Consequently, we always require that the robot can rapidly move to the target such that the new information can be obtained to update the decision [138]. It is obvious that the PSO algorithm cannot satisfy aforementioned requirements. Therefore, how to design the finite-time cooperative motion control algorithms is the second motivation of this dissertation.

Before we design the distributed decision algorithm and the finite-time cooperative motion control algorithms, we need to present an architecture that can guide the design of the decision algorithm and the cooperative control algorithm and illustrate the relationship of information exchange between the decision algorithm and the cooperative control algorithm. Therefore, how to design a distributed coordination control architecture is the third motivation of this dissertation.

1.2 Significance of This Dissertation

The significance of this dissertation can be stated in two aspects.

From the perspective of the research problem, this dissertation focuses on the problem of odour source localisation, which is of practical significance for human security such as searching for the source of toxic gas leakage and locating the origin of a fire at its initial stage. Moreover, this problem possesses several important characteristics, which will advance technologies. Hence, a meaningful research topic is discussed in this dissertation.

From the perspective of the solutions, the characteristics of odour source localisation can impose several constraints for the distributed decision and the cooperative control of the multi-robot system. For example, how do the intermittent detection events influence the convergence of a distributed decision algorithm? How should the cooperative control laws be designed to enable the robot group to rapidly find the odour source in a dynamical environment? In this dissertation, several effective solutions to deal with the aforementioned issues are proposed. Hence, the research presented in this dissertation provides important contributions to the knowledge in the fields of artificial intelligence and cooperative control.

1.3 Contributions and Outline of This Dissertation

1.3.1 Contributions

In this dissertation, in order to deal with the problem of odour source localisation by using a multi-robot system, a distributed coordination control architecture which consists of two levels: a decision level and a control level, is proposed. The main contribution for the proposed distributed coordination control architecture is that the decision about the position of the odour source in the decision level and the cooperative control of the robot group in the control level can be separately designed.

For the decision on the position of the odour source, a distributed decision algorithm with computational simplicity is formulated and its main contribution is to use both concentration information and wind information to more accurately and rapidly predict the position of the odour source. In particular, based on the observation model of the position of the odour source, wind information is used to predict the position of the odour source. The predicted position is then combined with the position estimated by the concentration information as a final decision on the position of the odour source.

For the cooperative control of the robot group, three kinds of cooperative control algorithms, i.e., the PSO-based finite-time motion control algorithm, the consensus-based finite-time motion control algorithm, and the potential-based finite-time motion control algorithm, are presented. The main contributions for three cooperative control algorithm are to introduce the idea of finite-time into the design of control laws such that the multi-robot system can finish the group objective in finite-time. Firstly, a PSO-based finite-time motion control algorithm, which can control a single robot to trace the plume, is designed. The analogy of the proposed PSO-based finite-time motion control algorithm is the foraging behaviors of birds. Secondly, a consensus-based finite-time motion control algorithm, which consists of a finite-time parallel motion control algorithm that can control the robot group to trace the plume and a finite-time circular motion control algorithm that can control the robot group to circle the predicted position of the odour source in order to search for the odour clues, is developed. The idea of the proposed consensus-based finite-time motion control algorithm is that we can flexibly arrange the formation shape of the robot group and utilize the characteristic of the wider detection region of the multi-robot system. Finally, a potential-based finite-time motion control algorithm, which also consists of a finite-time parallel motion control algorithm and a finite-time circular motion control algorithm, is derived. The idea of the proposed potential-based finite-time motion control algorithm is to integrate the obstacle avoidance function into the cooperative control law.

The effectiveness of the proposed coordination control architecture, the distributed decision algorithm, and the cooperative motion controllers is validated by benchmark testing and comparison of the other solutions in the same simulation environment with the same odour dispersion model and the dynamics of the differentially driven mobile robots.

1.3.2 Outline

The outline of the dissertation is briefly summarized in Figure 1.4.

Chapter 2 provides a distributed coordination control architecture, which consists of two levels: a decision level and a control level.

In the decision level, a distributed decision algorithm is proposed in Chapter 3 to make a decision on the position of the odour source. Specifically, an observation model for the position of the odour source based on wind information is derived. A Kalman filter is then used to estimate the probable position of the odour source. Moreover, a dynamic finite-time consensus fusion algorithm is designed to fuse the estimated observation results from other robots. The position of the odour source obtained by using wind information is combined with that of the odour source

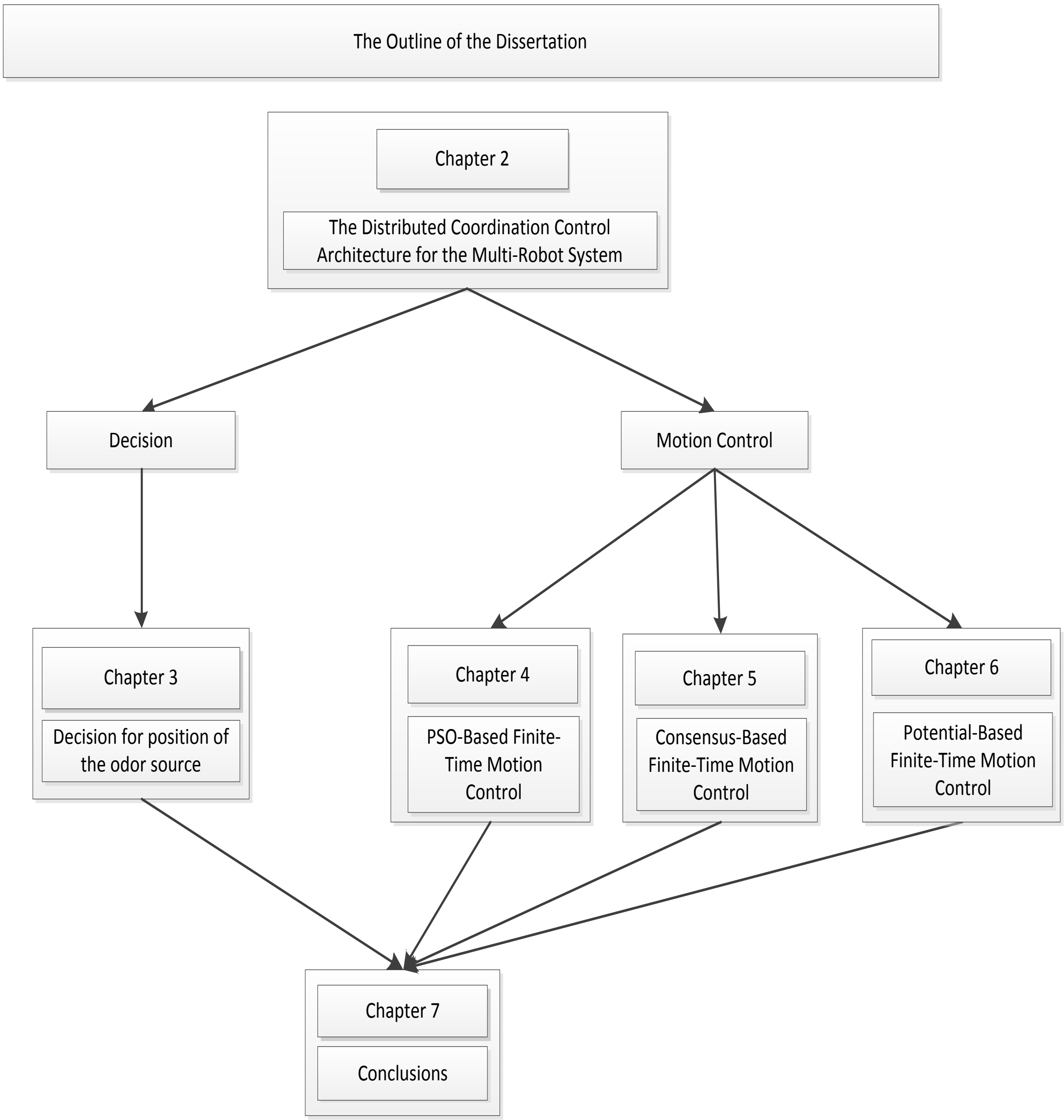

The Outline of the Dissertation
Chapter 2
The Distributed Coordination Control Architecture for the Multi-Robot System
Decision
Motion Control
Chapter 3
Decision for position of the odor source
Chapter 4
PSO-Based Finite-Time Motion Control
Chapter 5
Consensus-Based Finite-Time Motion Control
Chapter 6
Potential-Based Finite-Time Motion Control
Chapter 7
Conclusions

Figure 1.4: The outline of this dissertation.

obtained by using concentration information in order to make a final decision on the position of the odour source. In addition, the movement direction of the robot group is planned according to the predicted position of the odour source in order to trace the plume.

In the control level, Chapter 4 develops the PSO-based finite-time motion control algorithm, i.e. continuous-time finite-time particle swarm optimization (FPSO) by analyzing the continuous-time model of the PSO algorithm. As a discrete version of the FPSO algorithm, the discrete-time FPSO algorithm is also proposed by employing the same discretization scheme as the generalized particle swarm optimization (GPSO) and a linear matrix inequality (LMI) is used to obtain the convergence condition of the discrete-time FPSO algorithm. On the basis of the distributed decision algorithm, simulations are presented to illustrate the effectiveness of the PSO-based finite-time motion control algorithm for the problem of odour source localisation.

In order to efficiently use the characteristic of the wider detection of the multi-robot system, a consensus-based finite-time motion control algorithm is designed in Chapter 5 in the control level based on a new nonlinear finite-time consensus algorithm. This motion control algorithm consists of the finite-time parallel motion control algorithm that can control a group of robots to parallel move to trace the plume and a finite-time circular motion control algorithm that can enable the robot group to circle the predicted position of the odour source. The effectiveness of the consensus-based finite-time motion control algorithm is also illustrated for the problem of odour source localisation.

By introducing the potential function into controllers, Chapter 6 achieves a potential-based finite-time motion control algorithm in the control level based on a new finite-time coordination control algorithm and a new finite-time tracking control algorithm. This motion control algorithm also consists of a finite-time parallel motion control algorithm and a finite-time circular motion control algorithm. Similarly, the effectiveness of the potential-based finite-time motion control algorithm is illustrated for the problem of odour source localisation.

Chapter 7 provides final remarks on the contributions of this dissertation and suggests future works in the problem of odour source localisation.

Chapter 2

A Distributed Coordination Control Architecture

2.1 Introduction

In its traditional application, the PSO algorithm enables the velocity of the robot to be determined by using odour concentration information of the group and individual. We therefore begin this chapter by discussing the PSO algorithm in Section 2.2. A distributed coordination control architecture is then proposed in Section 2.3. Next, the effectiveness of the distributed coordination control architecture for the odour source localisation problem is illustrated in Section 2.4. Finally, we give the conclusion in Section 2.5.

2.2 The Analysis of the PSO Algorithm

In the last decade, the particle swarm optimization (PSO) [64] algorithm has been widely studied [98, 140, 113, 114, 60, 58, 59, 45]. Empirical evidences have been accumulated to show that PSO is a useful tool for optimization problems [140]. Due to lacking precision in a local search solution, an inertia factor ω in the velocity updating equation is introduced into the original version of PSO [55, 26], which gives rise to a commonly used form of PSO described by

$$\begin{cases} v_i(k+1) = g(v_i(k), u_i(k)) \\ x_i(k+1) = x_i(k) + v_i(k+1) \end{cases} \tag{2.1}$$

where

$$g(v_i(k), u_i(k)) \quad = \quad \omega v_i(k) + u_i(k) \tag{2.2}$$

$$u_i(k) \quad = \quad \alpha_1(x_l(k) - x_i(k)) + \alpha_2(x_g(k) - x_i(k)) \tag{2.3}$$

with

$$\begin{aligned} x_l(k) &= \arg\max\{f(x_i(k)), f(x_l(k-1))\} \\ x_g(k) &= \arg\max\{f(x_1(k)), \ldots, f(x_N(k)), f(x_g(k-1))\} \end{aligned}$$

where $v_i(k) = \begin{pmatrix} v_{i1}(k) & v_{i2}(k) & \cdots & v_{in}(k) \end{pmatrix}^T$ $(i = 1, 2, \cdots, N)$ is a velocity vector; $u_i(k)$ is a control vector; $x_l(k)$ denotes the previously best position of the ith particle; $x_g(k)$ refers to the globally best position of the swarm; ω is the inertia factor; N is the number of particles; $g : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ is a map; $f : \mathbb{R}^n \to \mathbb{R}$ is a fitness function; and α_j $(j = 1, 2)$ are random parameters, called acceleration coefficient.

In order to analyze how PSO algorithms are used to deal with the problem of odour source localisation, we introduce the oscillation center $p_i(k)$ given in (2.4) into (2.3).

$$p_i(k) \quad = \quad \frac{\alpha_1 x_l(k) + \alpha_2 x_g(k)}{\alpha_1 + \alpha_2} \tag{2.4}$$

Then, we have

$$u_i(k) \quad = \quad (\alpha_1 + \alpha_2)(p_i(k) - x_i(k)) \tag{2.5}$$

From (2.5), one can see that $u_i(k)$ can be regarded as a “P” controller and can keep the system (2.1) stable at the equilibrium $p_i(k)$ calculated by using swarm information $x_g(k)$ and individual information $x_l(k)$ under several parameter conditions [33]. Therefore, each particle uses both swarm information and individual information to predict a probable target position $p_i(k)$, and then employs a “P” controller $u_i(k)$ to adjust the movement direction of the particle toward the target position. In other words, the PSO algorithm provides a kind of "*decision-control mechanism*", i.e. a decision $p_i(k)$ and a control signal $u_i(k)$ [71].

From the perspective of "*decision-control mechanism*", the study of PSO algorithms consists of two categories: optimization performance improvement [81, 31, 146, 1] and stability analysis [15, 55, 32]. For optimization performance improvement, how to design a new $p_i(k)$ based on the characteristics of optimization problems is one main research content, such as the problem of odour source localisation [82, 83], the problem of disassembly sequencing [146], and the problem of vertical electrical sounding [31]. For stability analysis, how to analyze the convergence of a particle swarm [15, 55, 32] under a given control law $u_i(k)$ is another main research content. Accordingly, the commonly used analysis tools include a Lyapunov approach and a passivity approach. Current results of stability analysis [15, 55, 32] indicate that the particle swarm can converge under several conditions when $k \to \infty$. It is worth mentioning that the convergence analysis of PSO algorithms is of practical significance because one can see that the better results can be obtained only in the convergence region of PSO algorithms [31].

To sum up, one can see that PSO algorithms are suitable to coordinate a multi-robot system to deal with the problem of odour source localisation. Each robot can receive the current position information $x_j(k), j = 1, 2, \ldots, N$ and the concentration information $f(x_j(k)), j = 1, 2, \ldots, N$ via the local sensors and wireless communication networks such that the oscillation center $p_i(k)$ in (2.4) can be calculated. Then, the control signal $u_i(k)$ is generated in terms of (2.5). If one only considers the kinematics of mobile robots, the real control signal as an input to the mobile robot is $v_i(k+1)$ (The readers can refer to [93, 53, 93]), which is calculated in (2.2). Hence, we propose a formal control architecture, which is described in Figure 2.1 where $p_i(k)$ and $v_i(k+1)$ are produced in the decision module and the control module, respectively. As a result, PSO algorithms can be employed to coordinate a robot

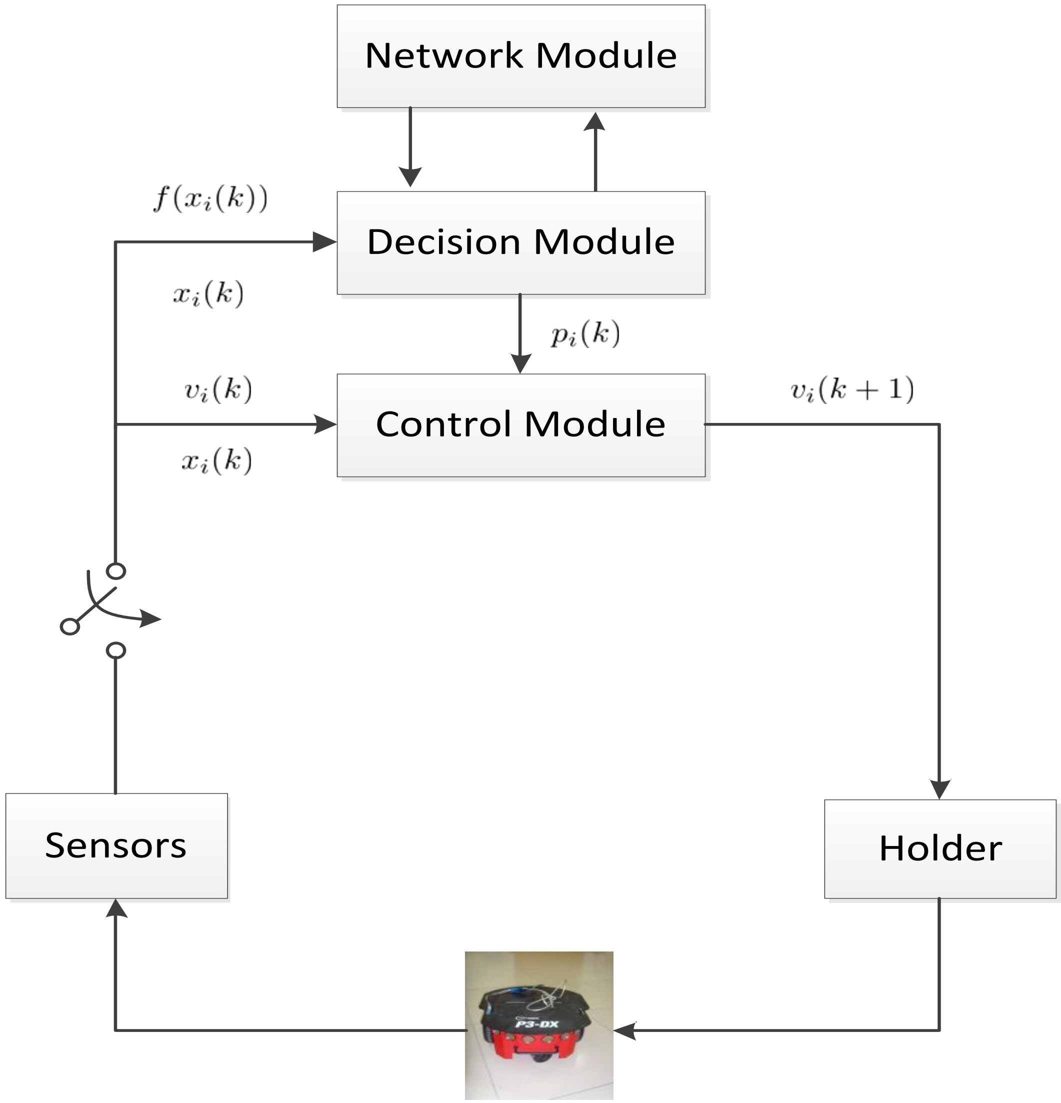


Figure 2.1: The distributed coordination control architecture for the discrete-time control signal.

group to deal with the problem of odour source localisation [93, 53, 93].

However, one can notice that several improved PSO algorithms cannot be directly used to coordinate the robot group for the problem of odour source localisation because they do not possess the form of control law. Due to the same reason, other kinds of evolutionary algorithms such as genetic algorithms cannot also be directly used for the problem of odour source localisation. Moreover, the researchers, who design a new algorithm to coordinate the multi-robot system to deal with the problem of odour source localisation, not only consider how to design the algorithm, but also notice that the developed algorithm should hold the expression of control laws. If the improved PSO algorithms and evolutionary algorithms can be used in the decision module to predict the probable position of the odour source, control algorithms can be separately designed in the control module in terms of problem characteristics. Therefore, we propose the following the distributed coordination control architecture.

2.3 A Distributed Coordination Control Architecture

In this section, we will propose a distributed coordination control architecture and consider a continuous-time kinematics model of N identical robots, which is described by

$$\dot{x}_i = u_i \quad i = 1, 2, \dots N \tag{2.6}$$

where x_i denotes the position for the robot i.

It is obvious that a continuous-time control signal is utilized to control the mobile robot in (2.6). Moreover, multiple robots need to communicate with each other in order to coordinate their motion behaviors. Consequently, the distributed coordination control architecture shown in Figure 2.1 is modified and illustrated in Figure 2.2. From this figure, one can see that the probable position of the odour source $p_i(k)$ is generated in the decision module and can be calculated in (2.4). In the control module, the continuous-time control signal $u_i(t)$ is produced based on the current individual information, group information, and the probable position of the odour source $p_i(k)$. As an example, we can use a consensus algorithm proposed by Ren [118] (2008) as a cooperative control law. This algorithm is given by

$$u_i(t) \;=\; -\alpha_i(x_i(t) - p_i(k)) - \sum_{j=1}^{N} a_{ij} \Big[(x_i(t) - p_i(k)) - (x_j(t) - p_j(k)) \Big] \tag{2.7}$$

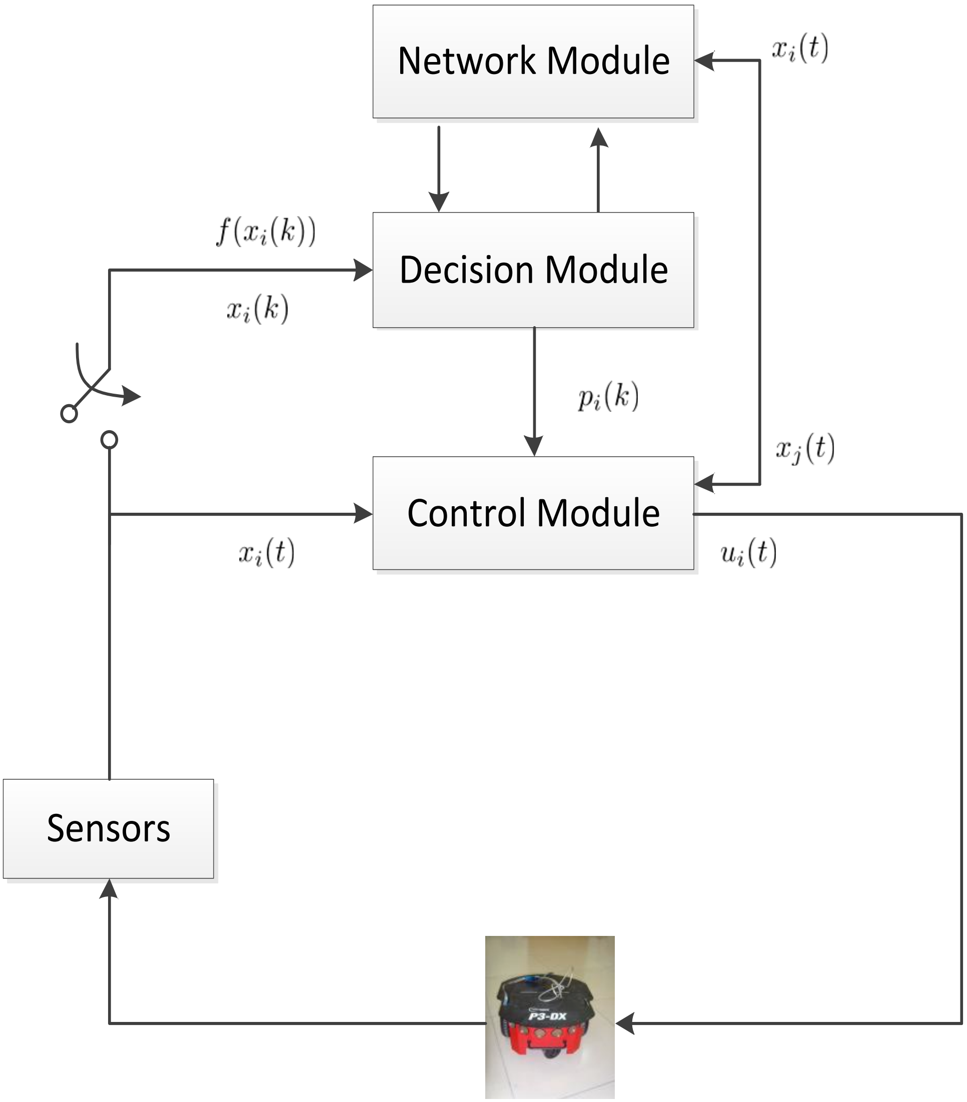


Figure 2.2: The distributed coordination control architecture for the continuous-time control signal.

where $p_i(k)$ and $p_j(k)$ denote the probable positions of the odour source for the ith robot and the jth robot, respectively; $x_i(t)$ and $x_j(t)$ are the real positions of the ith robot and the jth robot, respectively; a_{ij} is the (i, j) entry of an adjacent matrix A related with a communication topology; α_i is a positive scalar; and $u_i(t)$ is a control signal. Proposition 1 will guarantee that the system (2.6) is stable under a given cooperative control law (2.7).

Proposition 1. *The consensus algorithm (2.7) can guarantee that* $x_i(t) \to p_i(k)$*,*

$x_j(t) \to p_j(k)\ ,\ i = 1, 2, \ldots N\ \ as\ t \to \infty.$

Proof. It is omitted because the proof is similar to Theorem 3.6 in [118, 119]. □

Remark 1. *It should be pointed out that the decision algorithm in the decision module and the control law in the control module can be separately designed by using the distributed coordination control architecture shown in Figure 2.2. Researchers can focus on one of two algorithms according to the characteristics of the problem of odour source localisation. Therefore, the evolutionary theory in the computational intelligence field can be employed to design the decision algorithm and the cooperative control theory in the control field can be used to develop the control algorithm.*

From the characteristics of the odour source localisation problem, one can notice that tracing the plume is a better approach to locate the odour source. Therefore, when a position of the odour source is predicted in the decision module, the robot should not directly move toward the predicted position. Instead, the robot should trace the plume based on the predicted position. As a result, we introduce a planning module into the distributed coordination control architecture, which is shown in Figure 2.3.

In the planning module, based on the predicted position of the odour source, we will plan the movement direction of the robot in order to trace the plume. For instance, we introduce a spiral trajectory from the current position to the predicted position. This trajectory is shown in Figure 2.4 where "o" denotes the current position of the robot $x_i(k)$ and "*" denotes the predicted position of the odour source $h^{ic}(k)$.

The model of the spiral trajectory of the ith robot, which originates from logarithmic spiral, is described by

$$\begin{aligned} \begin{pmatrix} p_{ix}(k') \\ p_{iy}(k') \end{pmatrix} &= \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 \\ \sin\theta_1 & \cos\theta_1 \end{pmatrix} \begin{pmatrix} ae^{b\theta(k')}\cos\theta(k') - a \\ ae^{b\theta(k')}\sin\theta(k') \end{pmatrix} \\ &\quad + \begin{pmatrix} h_x^{ic}(k) \\ h_y^{ic}(k) \end{pmatrix} \end{aligned} \tag{2.8}$$

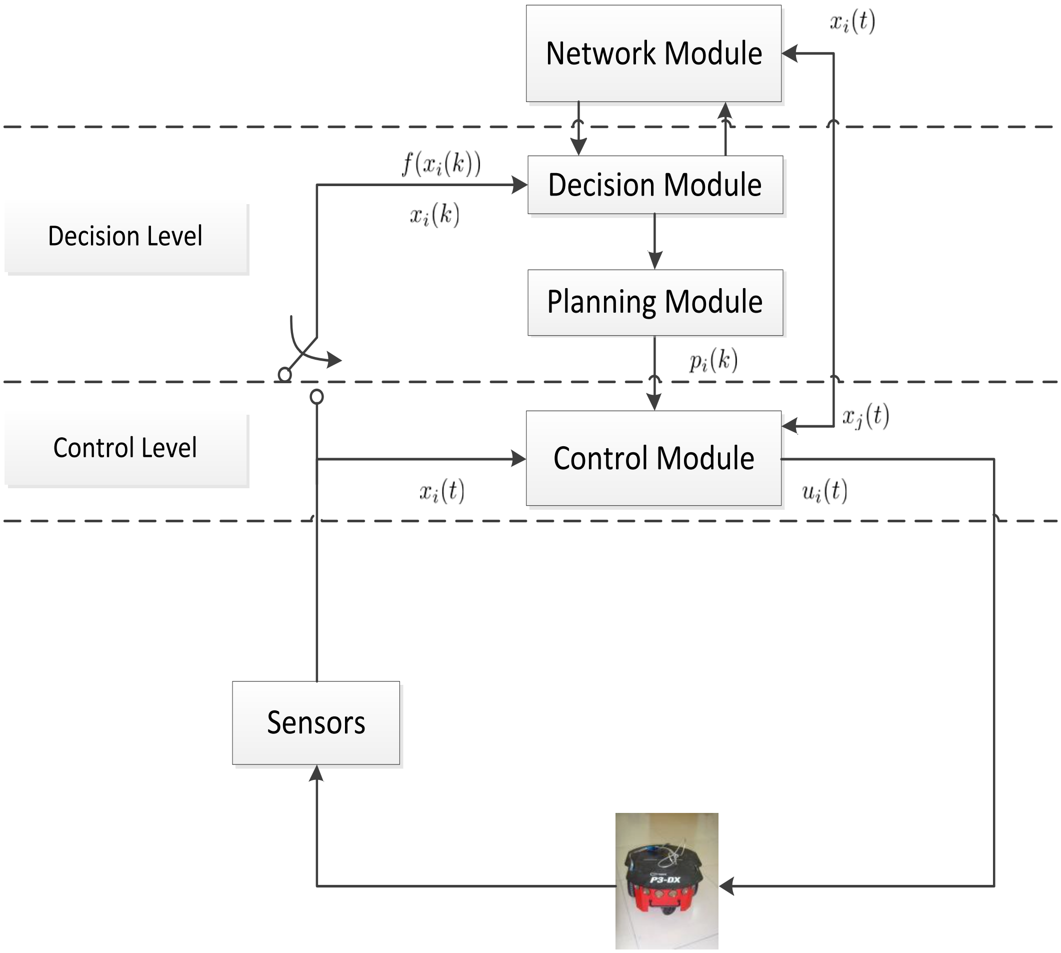


Figure 2.3: The proposed distributed coordination control architecture.

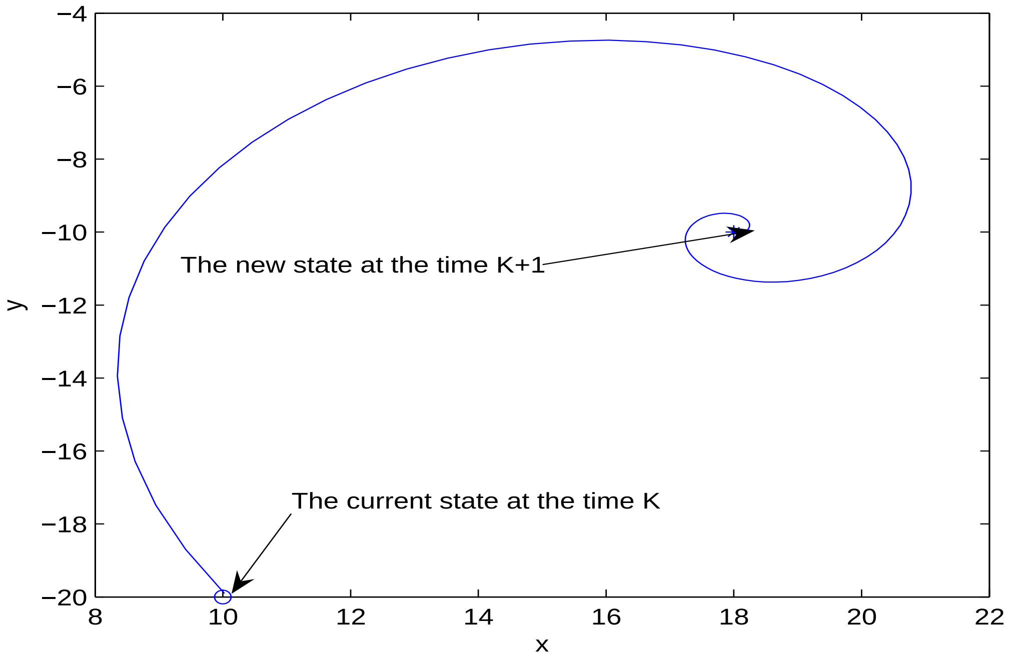


Figure 2.4: A spiral trajectory.

where $p_i(k')(p_{ix}(k'), p_{iy}(k'))$ is the position of the ith robot at time $k'(k-1 < k' \leq k)$; a and b ($a = 0.1, b = 0.4$ in the following simulations) are two parameters; $h^{ic}(k)(h_x^{ic}(k), h_y^{ic}(k))$, which is a predicted position of the odour source for the ith robot; $\theta(k')$ is a spiral angle, which decreases from $\theta_{max}(k)$ to 0 based on $\theta(k') = \theta(k'-1) - 0.1$ where $\theta_{max}(k)$ can be computed in (2.9).

$$\theta_{max}(k) \quad = \quad (1/b)\log(r/a) \tag{2.9}$$

where r is a distance between the position $p_i(k)$ and the position $h^{ic}(k)$. In the equation (2.8), θ_1 can be calculated by

$$\theta_1 = \theta_{11} - \theta_{12}$$

where

$$\theta_{11} \quad = \quad \begin{cases} \theta'_{11}, & if\ h_1\ is\ satisfied; \\ 2\pi + \theta'_{11}, & if\ h_2\ is\ satisfied; \\ \pi + \theta'_{11}, & otherwise. \end{cases}$$

$$\theta_{12} \quad = \quad \begin{cases} \theta'_{12}, & if\ h_3\ is\ satisfied; \\ 2\pi + \theta'_{12}, & if\ h_4\ is\ satisfied; \\ \pi + \theta'_{12}, & otherwise. \end{cases}$$

with

$$
\begin{aligned}
\theta'_{11} &= \operatorname{atan}((p_{iy}(k) - h_y^{ic}(k))/(p_{ix}(k) - h_x^{ic}(k))) \\
\theta'_{12} &= \operatorname{atan}((ae^{b\theta_{\max}} \sin\theta_{\max})/(ae^{b\theta_{\max}} \cos\theta_{\max})) \\
h_1 &= p_{ix}(k) > h_x^{ic}(k) \; and \; p_{iy}(k) > h_y^{ic}(k) \\
h_2 &= p_{ix}(k) > h_x^{ic}(k) \; and \; p_{iy}(k) < h_y^{ic}(k) \\
h_3 &= ae^{b\theta_{\max}} \cos\theta_{max} > 0 \; and \; ae^{b\theta_{max}} \sin\theta_{max} > 0 \\
h_4 &= ae^{b\theta_{\max}} \cos\theta_{max} > 0 \; and \; ae^{b\theta_{max}} \sin\theta_{max} < 0
\end{aligned}
$$

Remark 2. *The idea of spiral trajectories is also used in [37, 80]; however, there exist two different points between spiral trajectories introduced in this section and the ones described in [37, 80]. Firstly, the mathematical model of spiral trajectories is clearly illustrated while in [37, 80] the mathematical model of spiral trajectories is not given. In fact, the robots move along diamond shapes with a certain length and angle, which have been determined before starting experiments. Secondly, the robots move inward to trace the plume whereas in [37, 80] the robots move outward along diamond shapes in order to search for chemical clues.*

2.4 Odour Source Localisation

In what follows, we will give two examples to show the influence of the distributed coordination control architecture on the search efficiency of the multi-robot system for the problem of odour source localisation.

2.4.1 The Distributed Coordination Control Architecture for the Continuous-Time Control Signal

In this subsection, we will test the distributed coordination control architecture shown in Figure 2.2. PSO [93] and CPSO [53] algorithms as examples are implemented to predict the probable position of the odour source in the decision module,

Table 2.1: The parameters of the cooperative control algorithm in (2.7).

α_i	v_{max}(m/s)	ω_{max}(rad/s)
0.35	0.8	1.57

respectively. A consensus algorithm (2.7), which parameters are shown in Table 2.1, is then applied at the control module. PSO-A and CPSO-A are used to name the two algorithms with the architecture, respectively. In the light of the different sampling time Δt ($\Delta t = 10s, 20s, 30s$), PSO algorithms without the architecture are named by PSO-10, PSO-20, and PSO-30, respectively. CPSO algorithms have similar names.

In addition, the simulations are carried out under three kinds of environments, which are characterized by small wind, medium wind, and large wind, respectively. CPSO and PSO algorithms' parameters chosen in the simulations can be found in [93] and [53], respectively. It is worth mentioning that if any robot detects a maximum concentration predefined, the search behaviors of robots are terminated, and then the search time is recorded (The max search time and concentration are limited to 2500s and 19515ppt, respectively.). Tables 2.2-2.4 describe the success rates for different algorithms under the small wind environment, the medium wind environment, and the large wind environment. Correspondingly, Figures 2.5-2.7 show the average search time for different algorithms under the small wind environment, the medium wind environment, and the large wind environment.

From simulation results, one can see that the distributed coordination control architecture exerts different influence on PSO and CPSO algorithms, respectively. CPSO-A can attain the better results for various number of robots under the three kinds of wind speed. However, PSO-A only can obtain the better results than PSO-30 for 5 robots under the three kinds of wind speed. For 10 robots, PSO-A can generate very competitive results. But, for 15 and 20 robots, PSO-30 can obtain much better results than PSO-A. To sum up, the distributed coordination

Table 2.2: The success rates under the small wind environment based on 50 runs for the coordination control architecture shown in Figure 2.2.

Robots	PSO-10	PSO-20	PSO-30	PSO-A
5	0	2	34	30
10	0	34	82	70
15	2	56	70	74
20	4	68	74	92
Robots	CPSO-10	CPSO-20	CPSO-30	CPSO-A
5	2	24	30	100
10	46	88	76	100
15	68	72	78	100
20	84	76	76	100

Table 2.3: The success rates under the medium wind environment based on 50 runs for the coordination control architecture shown in Figure 2.2.

Robots	PSO-10	PSO-20	PSO-30	PSO-A
5	0	2	20	50
10	0	38	86	82
15	2	80	92	84
20	16	90	96	92
Robots	CPSO-10	CPSO-20	CPSO-30	CPSO-A
5	14	22	34	100
10	74	80	90	100
15	88	92	88	100
20	90	98	86	100

control architecture can improve the search performance for the CPSO algorithm and provide competitive results for the PSO algorithm.

2.4.2 The Proposed Distributed Coordination Control Architecture

In this subsection, we will test the proposed distributed coordination control architecture shown in Figure 2.3. PSO [93] and CPSO [53] are still used as examples to

Table 2.4: The success rates under the large wind environment based on 50 runs for the coordination control architecture shown in Figure 2.2.

Robots	PSO-10	PSO-20	PSO-30	PSO-A
5	0	4	26	52
10	6	68	92	92
15	10	90	100	88
20	26	94	98	100
Robots	CPSO-10	CPSO-20	CPSO-30	CPSO-A
5	16	38	44	100
10	82	88	96	100
15	98	94	98	100
20	100	100	96	100

evaluate the proposed distributed coordination control architecture. PSO-Consensus (PSO-C), and CPSO-Consensus (CPSO-C) are used to name two algorithms using the distributed coordination control architecture shown in Figure 2.2 while PSO-S-Consensus (PSO-S-C) and CPSO-S-Consensus (CPSO-S-C) are used to name two algorithms using the proposed distributed coordination control architecture shown in Figure 2.3.

Figures 2.8-2.10 show the average search time for different algorithms under the small wind environment, the medium wind environment, and the large wind environment. Correspondingly, Tables 2.5-2.7 describe the success rates for different algorithms under the small wind environment, the medium wind environment, and the large wind environment. From simulation results, one can see that the proposed distributed coordination control architecture can provide a flexible mechanism for the CPSO algorithm and the PSO algorithm such that the researchers can focus on the design of the decision algorithm or the cooperative control algorithm.

Table 2.5: The success rates under the small wind environment based on 50 runs for the coordination control architecture shown in Figure 2.3.

Robots	PSO	PSO-C	PSO-S-C	CPSO	CPSO-C	CPSO-S-C
5	34	30	80	30	100	100
10	82	70	100	76	100	100
15	70	74	100	78	100	100
20	74	92	100	76	100	100

Table 2.6: The success rates under the medium wind environment based on 50 runs for the coordination control architecture shown in Figure 2.3.

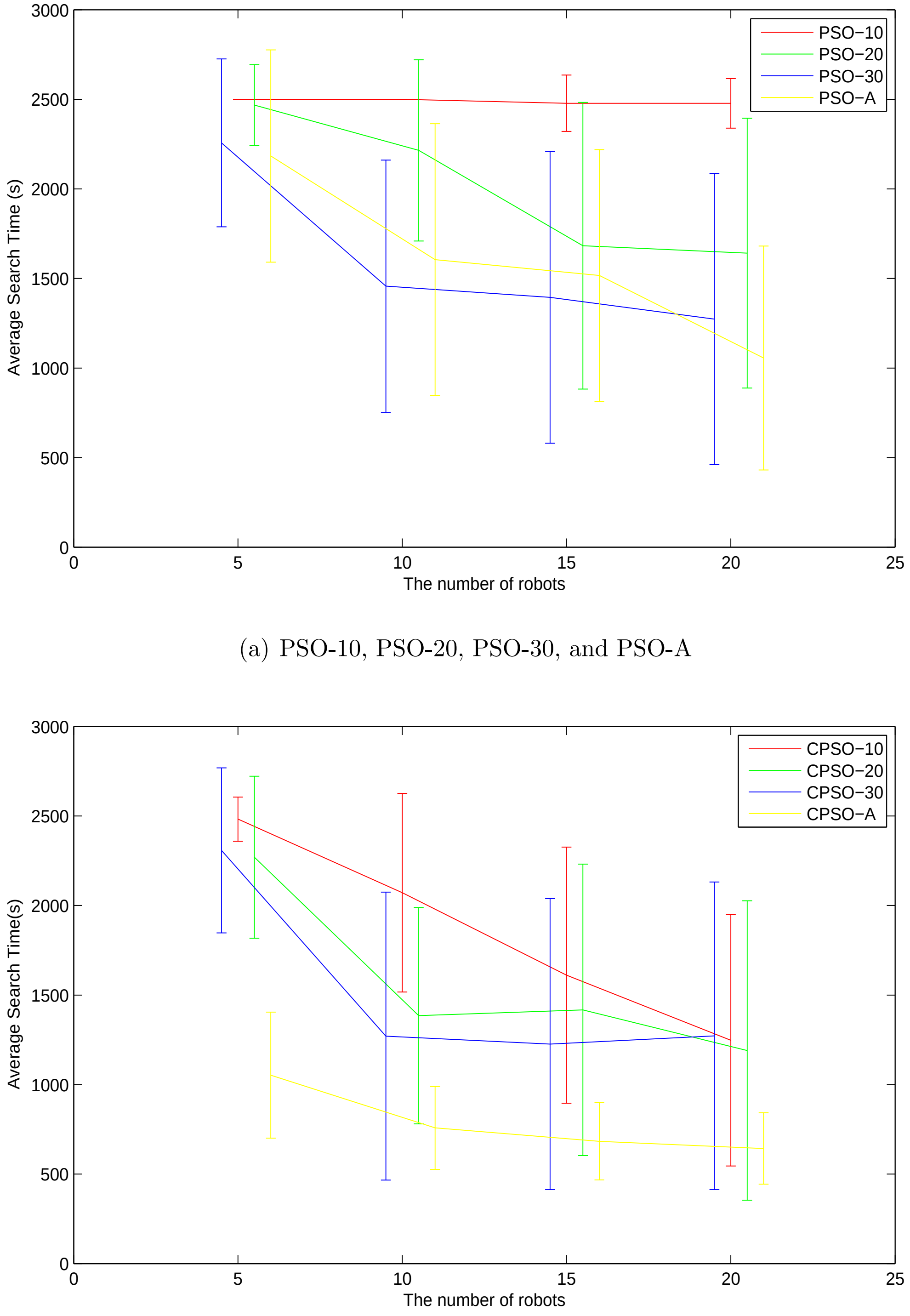
Robots	PSO	PSO-C	PSO-S-C	CPSO	CPSO-C	CPSO-S-C
5	20	50	88	34	100	100
10	86	82	100	90	100	100
15	92	84	100	88	100	100
20	96	92	100	86	100	100

Table 2.7: The success rates under the large wind environment based on 50 runs for the coordination control architecture shown in Figure 2.3.

Robots	PSO	PSO-C	PSO-S-C	CPSO	CPSO-C	CPSO-S-C
5	26	52	100	44	100	100
10	92	92	100	96	100	100
15	100	88	100	98	100	100
20	98	100	100	96	100	100

2.5 Conclusion

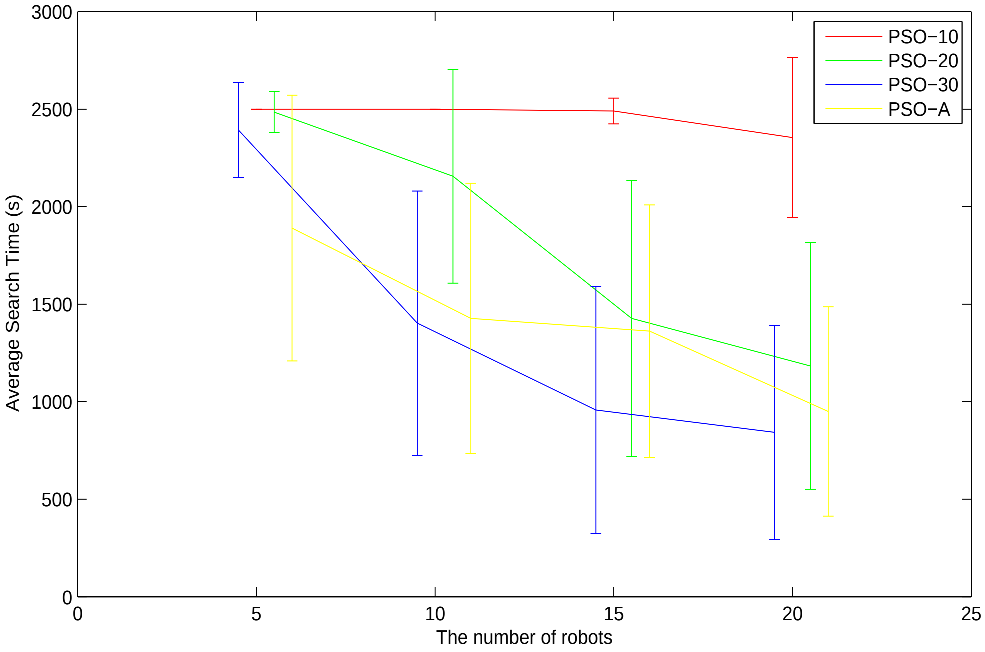
We have proposed a distributed coordination control architecture for a multi-robot system. The proposed distributed coordination control architecture consists of two levels: a decision level and a control level. In the decision level, we have predicted the probable position of the odour source by using evolutionary algorithms. Moreover, we have given the movement direction of the robot based on the predicted position of the odour source in order to trace the plume. In the control level, we have employed a consensus algorithm, which can enable robots to move along the given movement direction. Finally, we have illustrated the effectiveness of the proposed distributed coordination control architecture.



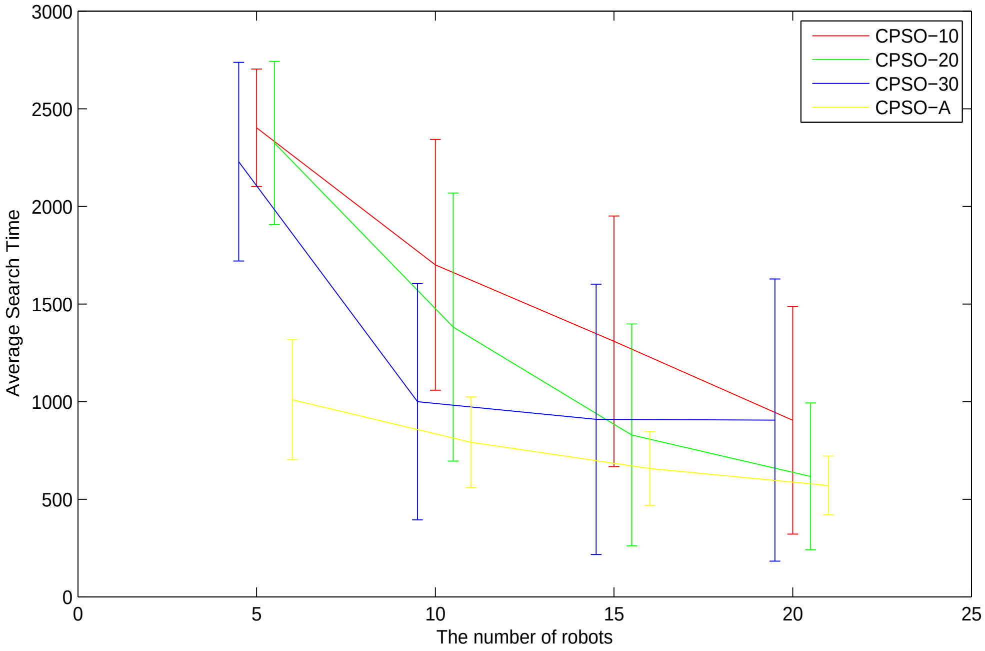
(a) PSO-10, PSO-20, PSO-30, and PSO-A

(b) CPSO-10, CPSO-20, CPSO-30, and CPSO-A

Figure 2.5: The average search time under the small wind environment for the coordination control architecture shown in Figure 2.2.

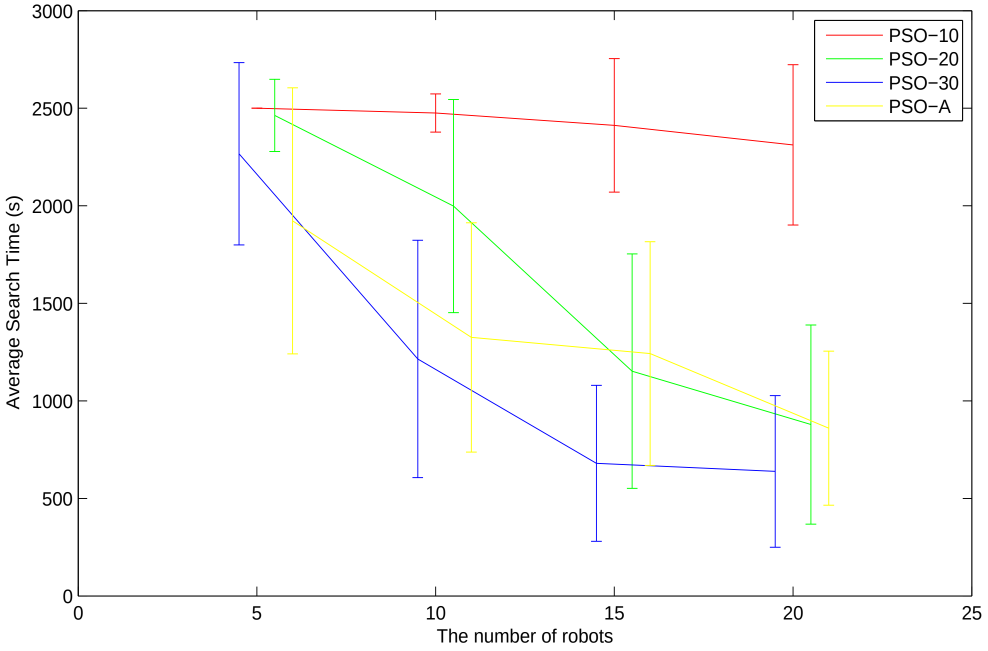


(a) PSO-10, PSO-20, PSO-30, and PSO-A

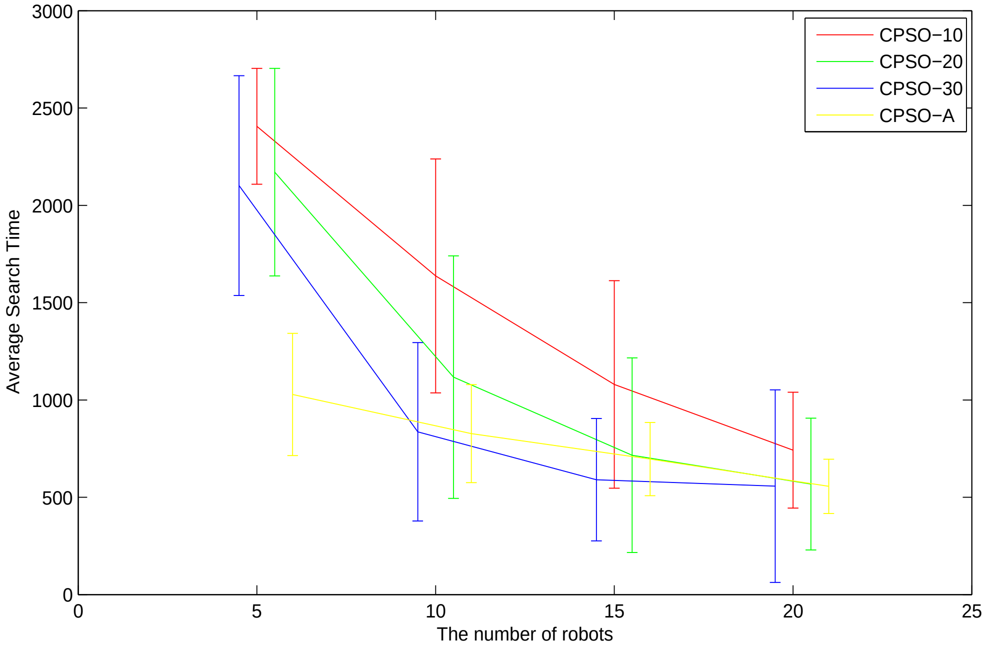


(b) CPSO-10, CPSO-20, CPSO-30, and CPSO-A

Figure 2.6: The average search time under the medium wind environment for the coordination control architecture shown in Figure 2.2.

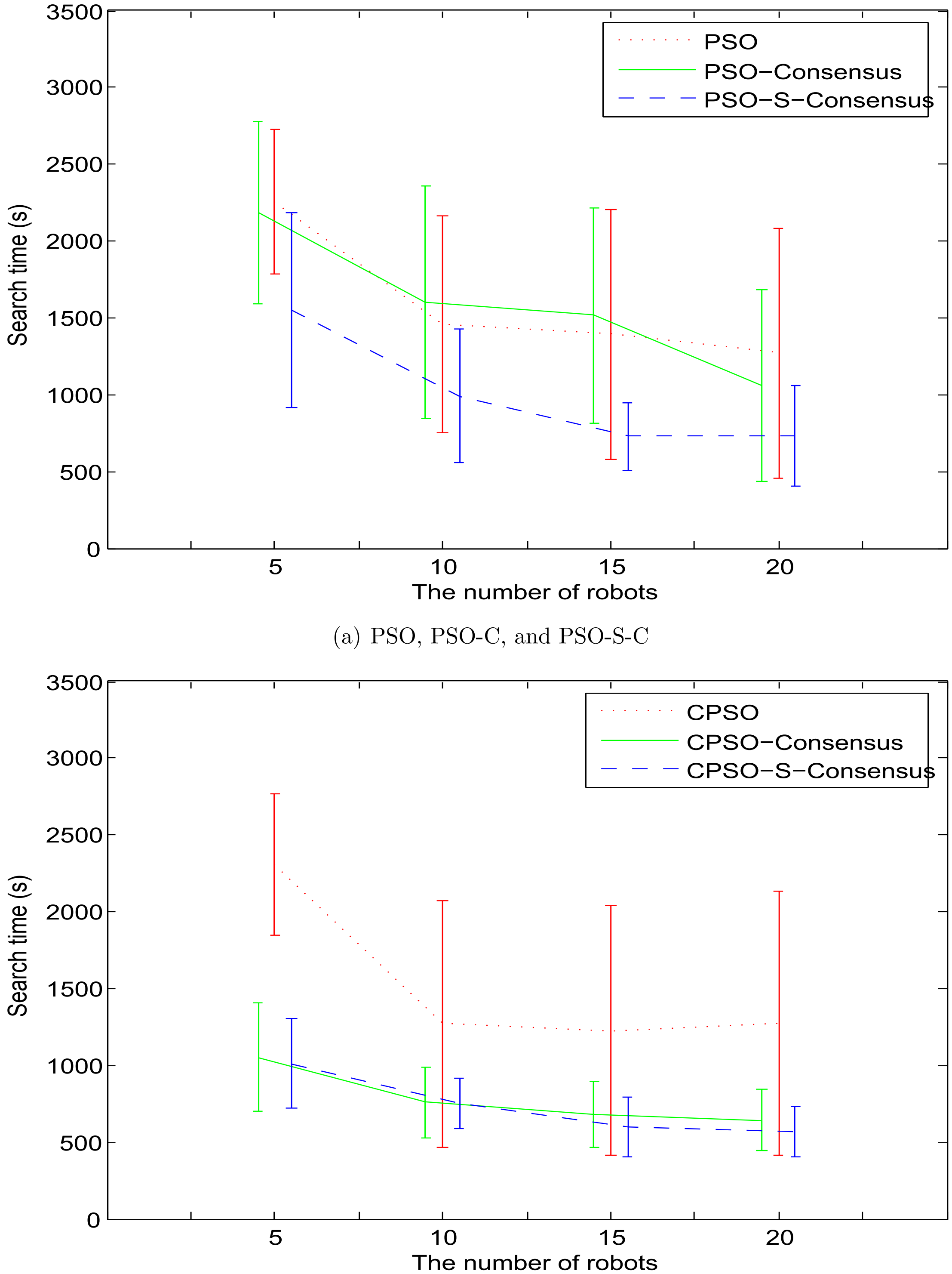


(a) PSO-10, PSO-20, PSO-30, and PSO-A



(b) CPSO-10, CPSO-20, CPSO-30, and CPSO-A

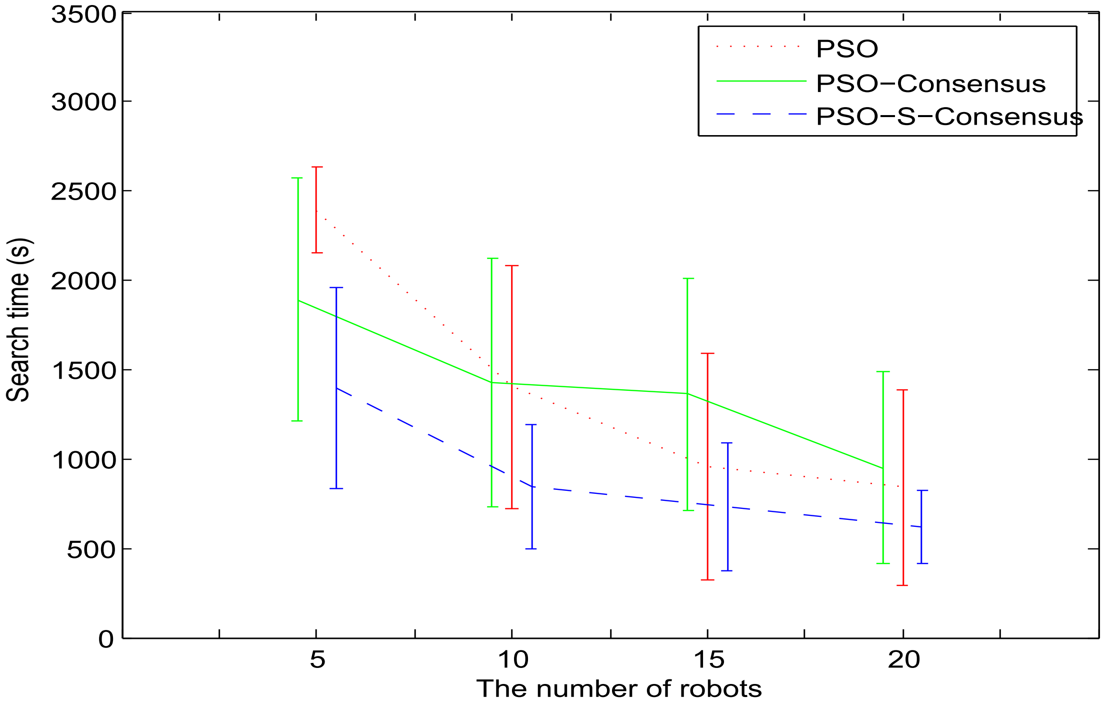
Figure 2.7: The average search time under the large wind environment for the coordination control architecture shown in Figure 2.2.



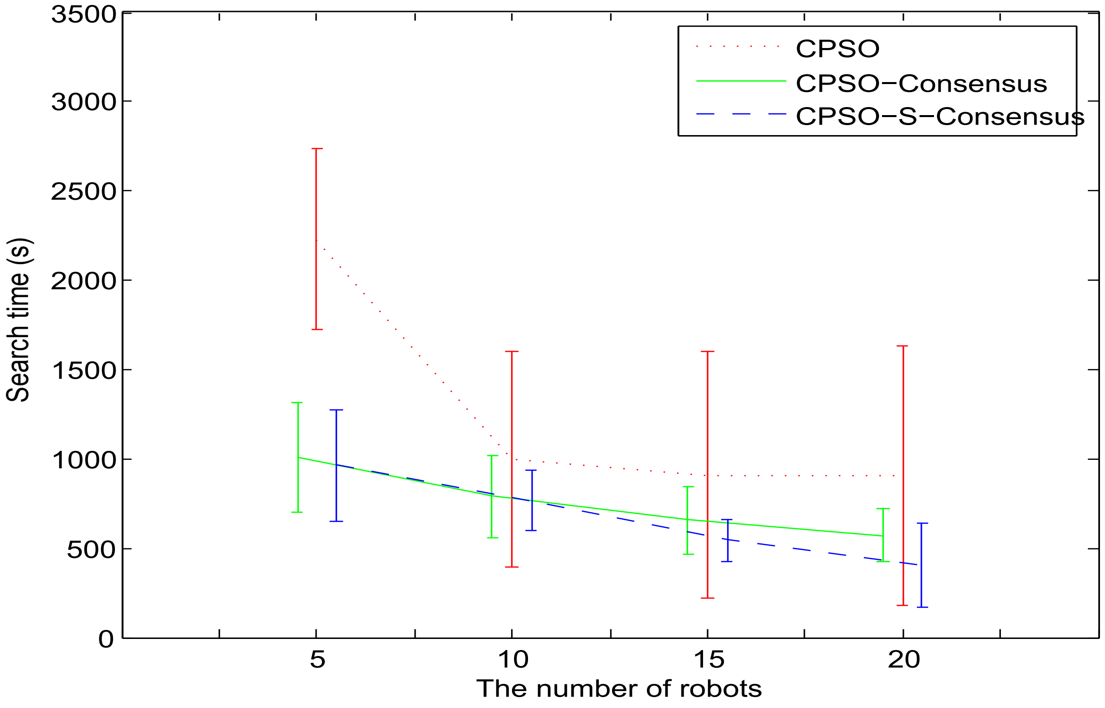
(a) PSO, PSO-C, and PSO-S-C

(b) CPSO, CPSO-C, and CPSO-S-C

Figure 2.8: The average search time under the small wind environment for the coordination control architecture shown in Figure 2.3.

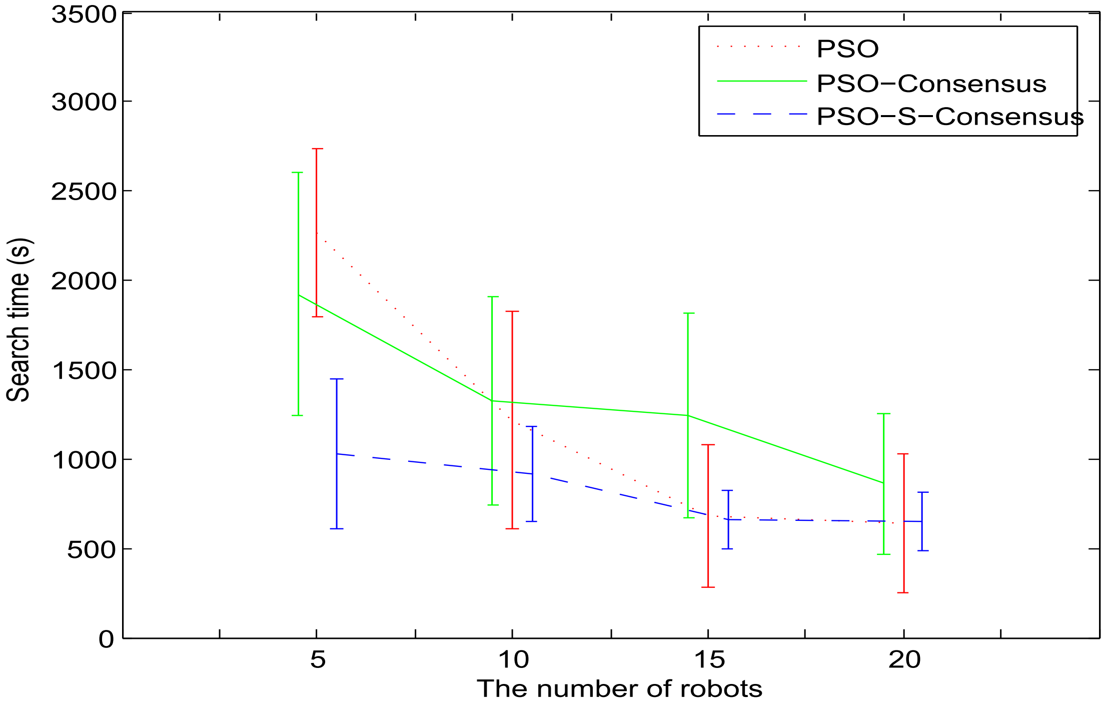


(a) PSO, PSO-C, and PSO-S-C

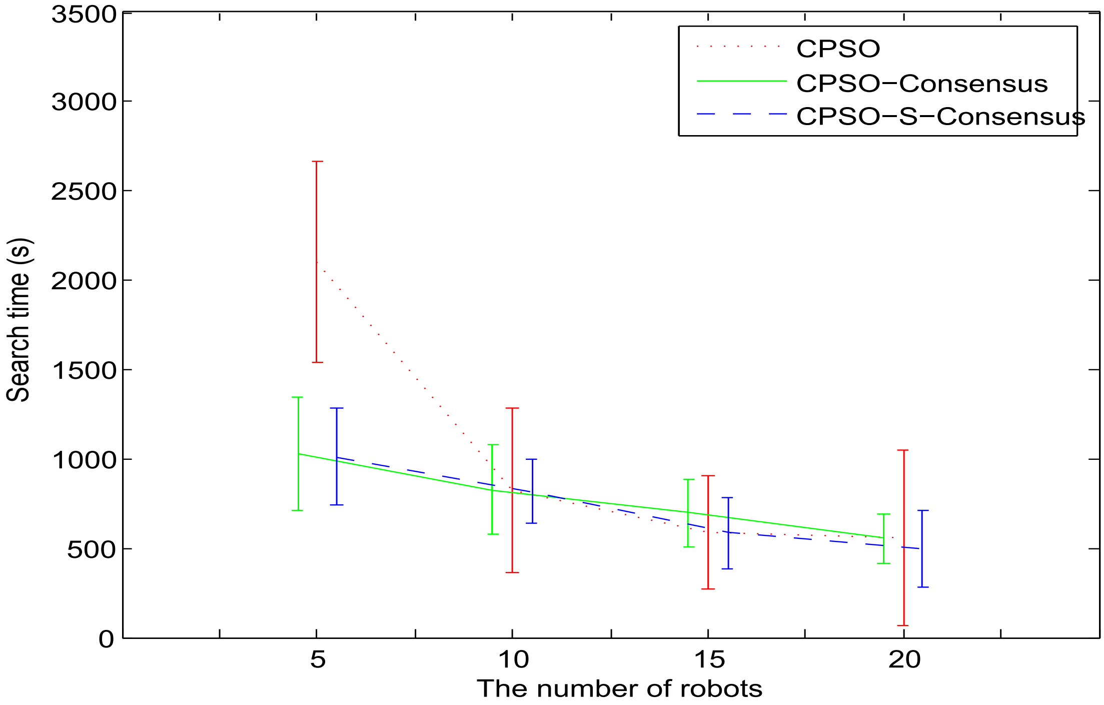


(b) CPSO, CPSO-C, and CPSO-S-C

Figure 2.9: The average search time under the medium wind environment for the coordination control architecture shown in Figure 2.3.



(a) PSO, PSO-C, and PSO-S-C



(b) CPSO, CPSO-C, and CPSO-S-C

Figure 2.10: The average search time under the large wind environment for the coordination control architecture shown in Figure 2.3.

Chapter 3

A Distributed Decision Algorithm

3.1 Introduction

It has been established in Chapter 2 that decision algorithms and controllers can be separately designed in the distributed coordination control architecture. This chapter presents several preliminaries and derives the distributed decision algorithm with computational simplicity in Section 3.2 and Section 3.3, respectively. The effectiveness of the proposed distributed decision algorithm for the problem of odour source localisation is then illustrated in Section 3.4. In Section 3.5, the conclusion is given.

3.2 Preliminaries

In this section, a continuous-time dynamic model of N identical robots is considered and given in (1.7). Moreover, l_N denotes the index set $\{1, 2, \ldots, N\}$. Let $\text{sig}(r)^a = \text{sign}(r)|r|^a$, where $0 < a < 1$, $r \in R$, and $\text{sign}(\cdot)$ is a sign function. Next, we give a definition of finite-time convergence [7, 6].

Definition 1. *Consider the system $\dot{x} = f(x(t))$. The origin is said to be a finite-time-stable equilibrium if there exists an open neighborhood $\mathcal{N} \subseteq \mathcal{D}$ of the origin and a function $T : \mathcal{N} \setminus \{0\} \to (0, \infty)$, such that for every $x_0 \in \mathcal{N} \setminus \{0\}$, $x(t)$ is defined for $t \in [0, T(x_0)]$, $x(t) \in \mathcal{N} \setminus \{0\}$, for $t \in [0, T(x_0)]$, and $\lim_{t \to T(x_0)} x(t) = 0$.*

If $\mathcal{D} = \mathcal{N} = \mathrm{R}^n$, the origin is said to be a globally finite-time-stable equilibrium.

Correspondingly, we give the following lemmas [7, 6], which will be used in the convergence analysis.

Lemma 1. *Suppose there exists a continuously differentiable function $V : \mathcal{D} \to \mathrm{R}$, real number $k > 0$ and $a \in (0, 1)$, and a neighborhood $\mathcal{U} \subset \mathcal{D}$ of the origin such that V is positive definite on $\mathcal{U}$ and $\dot{V} + kV^a$ is negative semidefinite on $\mathcal{U}$. Then the origin is a finite-time-stable equilibrium of the system $\dot{x} = f(x(t))$. Moreover, if T is the settling time, then $T(x_0) \leq \frac{1}{k(1-a)} V(0)^{1-a}$ for all x_0 in the open neighborhood of the origin.*

Moreover, let $G = (\nu, \mathcal{E}, A)$ be a weighted directed graph of order N with the set of nodes $\nu = \{\nu_1, \nu_2, \cdots, \nu_N\}$, the set of edges $\mathcal{E} \subseteq \nu \times \nu$, and a weighted adjacency matrix $A = [a_{ij}]$ with nonnegative adjacency elements a_{ij}. Let $L(A) = [l_{ij}] \in \mathbb{R}^{N \times N}$ denotes the graph Laplacian of $G = (\nu, \mathcal{E}, A)$, which is defined by

$$l_{ij} = \begin{cases} \sum_{j=1, j \neq i}^{N} a_{ij}, & i = j \\ -a_{ij}, & i \neq j \end{cases} \tag{3.1}$$

The graph $G = (\nu, \mathcal{E}, A)$ is used to model the communication topology among robots and we have the following property [144].

Lemma 2. *If $G = (\nu, \mathcal{E}, A)$ has a directed spanning tree, then eigenvalue 0 is algebraically simple and all other eigenvalues are with positive real parts for the Laplacian matrix $L(A)$.*

3.3 A Distributed Decision Algorithm

In this section, we will first give a measurement model for the position of the odour source based on wind information, and then use a Kalman filter to estimate the position of the odour source. Next, we will describe the details of a dynamic finite-time consensus fusion algorithm. Finally, we will give the distributed decision algorithm.

3.3.1 A Measurement Model

For short time-scales, various studies show that the filament movement can be modeled as a random walk superimposed on the downflow advection [29, 111]. Based on these studies, the movement process of a single filament can be modeled as

$$\dot{x}(t) \quad = \quad u(t) + n(t) \tag{3.2}$$

where $x(t)$ is the position of the filament at time t; $u(t)$ is the mean airflow velocity at the position $x(t)$; and $n(t)$ denotes a random process, which will be assumed to be Gaussian with zero mean and σ^2 variance.

If the odour source released this filament at time t_l, the position $x(t_l, t_k)$ of the filament at time $t_k > t_l$ can be calculated by integrating (3.2) as

$$x(t_l, t_k) \quad = \quad \int_{t_l}^{t_k} u(\tau)d\tau + \int_{t_l}^{t_k} n(\tau)d\tau + x_s(t_l) \tag{3.3}$$

where $x_s(t_l)$ refers to the position of the odour source at time t_l. Because we only have sensor measurements at the discrete times, $\{t_i\}_{i=l}^{k} = \{t_l, t_{l+1}, \ldots, t_{k-1}, t_k\}$ can be seen as the discrete-time indexes. Hence, $\int_{t_l}^{t_k} u(\tau)d\tau$ is approximately computed as $\int_{t_l}^{t_k} u(\tau)d\tau \approx \sum_{i=t_l}^{t_{k-1}} u(i)\Delta t$. Moreover, since we only consider a stationary odour source, which means that the position of the odour source is not changed with the increase of time, we have

$$x_s(t_k) \quad = \quad x_s(t_l) \tag{3.4}$$

Define $v(t_l, t_k) = \sum_{i=t_l}^{t_{k-1}} u(i)\Delta t$ and $w(t_l, t_k) = \int_{t_l}^{t_k} n(\tau)d\tau$ that is a Gaussian process with zero mean and $(t_k - t_l)\sigma^2$ variance; therefore, the position of the filament can be rewritten in discrete time as

$$x(t_l, t_k) \quad = \quad x_s(t_k) + v(t_l, t_k) + w(t_l, t_k) \tag{3.5}$$

However, time t_l at which the source started to release the filament is unknown and earlier than the start time of this mission. But, we only record wind information after starting this mission. Therefore, the start time of this mission is seen as the releasing time of filaments. Moreover, we must account for all possible releasing time t_l as

$$\bar{v}(t_k) \;=\; \frac{1}{k}\sum_{t_l=0}^{t_{k-1}} v(t_l, t_k) \tag{3.6}$$

Similarly, we have

$$\bar{w}(t_k) \;=\; \frac{1}{k}\sum_{t_l=0}^{t_{k-1}} w(t_l, t_k) \tag{3.7}$$

where $\bar{w}(t_k)$ is also a Gaussian process with zero mean and $\frac{1}{k}\sum_{t_l=0}^{t_{k-1}}(t_k - t_l)\sigma^2$ variance. Moreover, if the ith robot detects the filament, the real movement distance $v(t_l, t_k)$ of the filament advected by wind flow should satisfy $\| v(t_l, t_k) - x_i(t_k) \|_2 < \alpha$ where α is a control parameter and $x_i(t_k)$ is the current position of the ith robot at time t_k.

Finally, in the following, we use the discrete-time index k instead of the t_k to describe the current time. Hence, the movement process of this filament is modeled as

$$x(k) \;=\; x_s(k) + \bar{v}(k) + \bar{w}(k) \tag{3.8}$$

In order to derive the measurement model, we define $z_i(k) = x(k) - \bar{v}(k)$ in (3.8), and then we get

$$z_i(k) \;=\; x_s(k) + \bar{w}(k) \tag{3.9}$$

where $z_i(k)$ is regarded as a measurement for $x_s(k)$ at time k; and $\bar{w}(k)$ is regarded as a measurement noise, which is a Gaussian process with zero mean and $\frac{1}{k}\sum_{l=0}^{k-1}(k - l)\sigma^2$ variance.

Remark 3. *It is worth mentioning that the control parameter α has a significant impact on the measurement value $z_i(k) = x(k) - \bar{v}(k)$. If the inequality $\| v(t_l, t_k) -$*

$x_i(t_k) \|_2 < \alpha$ is not satisfied, the movement distance of the filament advected by airflows will be neglected. Therefore, the control parameter α will influence the quality of data obtained.

3.3.2 A Kalman Filter with Intermittent Observations and Uncertain Measurement Noise

In order to use Kalman filters, we need to reformulate the odour source localisation problem into a position estimation problem. Firstly, we have a measurement model described by (3.9). Secondly, based on (3.4), the movement process of the odour source can be modeled as

$$x_s(k) = x_s(k-1) \tag{3.10}$$

As a consequence, the odour source localisation problem can be dealt with as a position estimation problem; therefore, we use a discrete Kalman filter method, which consists of two parts: time update equations and measurement update equations. For the first part, time update equations are described by

$$\hat{x}_s^{-i}(k) = \hat{x}_s^i(k-1) \tag{3.11}$$

$$P^{-i}(k) = P^i(k-1) \tag{3.12}$$

where $\hat{x}_s^{-i}(k)$ is *a priori* position of the odour source at time k for the ith robot given knowledge of the process prior to time $k-1$; $\hat{x}_s^i(k-1)$ is *a posteriori* position estimate at time $k-1$ for the ith robot; and $P^{-i}(k)$ is *a priori* estimate error covariance at time k for the ith robot while $P^i(k-1)$ is *a posteriori* estimate error covariance at time $k-1$ for the ith robot.

For the second part, measurement update equations are given by

$$K^i(k) = P^{-i}(k)(P^{-i}(k) + R_0(k))^{-1} \tag{3.13}$$

$$\hat{x}_s^i(k) = \hat{x}_s^{-i}(k) + K^i(k)(z_i(k) - \hat{x}_s^{-i}(k)) \tag{3.14}$$

$$P^i(k) = (I - K^i(k))P^{-i}(k) \tag{3.15}$$

where $K^i(k)$ is chosen to be the gain that minimizes *a posteriori* error covariance; $R_0(k)$ is a measurement noise covariance matrix; $\hat{x}_s^i(k)$ is *a posteriori* position estimate at time k for the ith robot; and $z_i(k)$ is obtained by the observation model (3.9).

However, there exist two issues that must be considered when we use the Kalman filters to estimate the position of the odour source. One issue is about the measurement noise covariance matrix $R_0(k)$ that is known and approximately calculated (see the measurement model (3.9)). In fact, the exact measurement noise covariance is not known, i.e. $R_i(k) = R_0(k) + \gamma_i \Psi_i \Psi_i^T$ where $R_i(k)$ is the actual measurement noise covariance and $\gamma_i \Psi_i \Psi_i^T (\gamma_i \geq 0)$ is unknown. Hence, we are interested in how the uncertainties in the measurement noise covariance affect the state estimate. Another issue is about the intermittent observations. Due to the characteristics of the problem of odour source localisation, the robot cannot always detect the odour clues. Therefore, we are interested in whether the Kalman filter converges in this case.

We define the arrival of the observation at time k as a binary random variable λ_k with probability distribution $P(\lambda_k = 1)$. In the light of the reference [133], we rewrite (3.14) and (3.15) as

$$\hat{x}_s^i(k) = \hat{x}_s^{-i}(k) + \lambda_k K^i(k)(z_i(k) - \hat{x}_s^{-i}(k)) \tag{3.16}$$

$$P^i(k) = (I - \lambda_k K^i(k))P^{-i}(k) \tag{3.17}$$

From the measurement model (3.9), we have

$$z_i(k) = x_s^i(k) + \bar{\omega}_i(k)$$

where $z_i(k)$ is the measurement for the robot i at time k; $x_s^i(k)$ is the position of the odour source for the robot i at time k; and $\bar{\omega}_i(k)$ is the measurement noise for the robot i at time k. Moreover, based on (3.11), we obtain

$$\begin{aligned}\hat{x}_s^i(k) &= \hat{x}_s^i(k-1)+\lambda_k K^i(k)(x_s^i(k)-\hat{x}_s^i(k-1)+\bar{\omega}_i(k)) \\ &= \hat{x}_s^i(k-1)+\lambda_k K^i(k)(x_s^i(k-1)-\hat{x}_s^i(k-1)+\bar{\omega}_i(k)) \\ &= \hat{x}_s^i(k-1)+\lambda_k K^i(k)(e^i(k-1)+\bar{\omega}_i(k))\end{aligned}$$

where $e^i(k-1)=x_s^i(k-1)-\hat{x}_s^i(k-1)$. Furthermore, we can derive

$$e^i(k)=e^i(k-1)-\lambda_k K^i(k)(e^i(k-1)+\bar{\omega}_i(k))$$

Let $\bar{P}^i(k)$ be a real error covariance matrix. We have

$$\begin{aligned}\bar{P}^i(k) &= \mathbb{E}[e^i(k)e^i(k)^T] \\ &= (\lambda_k K^i(k)-I)\bar{P}^i(k-1)(\lambda_k K^i(k)-I)^T \\ &\quad +\lambda_k^2 K^i(k)R_0(k)K^i(k)^T+\gamma_i\lambda_k^2 K^i(k)\Psi_i\Psi_i^T K^i(k)^T\end{aligned}$$

where $\mathbb{E}$ is an expected symbol. When $\lambda_k=0$, i.e. no observations, we have $\bar{P}^i(k)=\bar{P}^i(k-1)$. Otherwise, when $\lambda_k=1$, we have

$$\begin{aligned}\bar{P}^i(k) &= (K^i(k)-I)\bar{P}^i(k-1)(K^i(k)-I)^T \\ &\quad +K^i(k)R_0(k)K^i(k)^T+\gamma_i K^i(k)\Psi_i\Psi_i^T K^i(k)^T \qquad (3.18)\end{aligned}$$

In the following, we will describe that there exits a unique $\bar{P}^i(k)$ as $k\to\infty$. Hence, we first give two lemmas [5, 130].

Lemma 3. *Consider the discrete-time Lyapunov equation*

$$HXH^T-X+M=0$$

where M is Hermitian. Then, the following are true.

1. *If H is stable, then X is unique and Hermitian and*

$$X=\sum_{k=0}^{\infty}H^kMH^k$$

2. *If H is stable and $M > 0$, then X is unique, Hermitian and $X > 0$.*

We introduce (3.12) and (3.13) into (3.17) to obtain the discrete-time Riccati equation, which is described by

$$P^i(k) \;=\; P^i(k-1) - \lambda_k P^i(k-1)[P^i(k-1) + R_0]^{-1} P^i(k-1) \tag{3.19}$$

Lemma 4. *Consider the discrete-time Riccati equation (3.19) where $\lambda_k = 1$ and $P^i(0) \geq 0$. Then*

1. *There exists a $P \geq 0$ such that for all $P^i(0) \geq 0$,*

$$\lim_{k\to\infty} P^i(k) \;=\; P$$

Furthermore, P is the unique solution of the algebraic matrix equation

$$P[P + R_0]^{-1} P = 0$$

within the class of positive semidefinite symmetric matrices.

2. *The eigenvalues of the matrix*

$$D = I - [P + R_0]^{-1} P$$

are strictly within the unit circle.

Therefore, we have the following proposition.

Proposition 2. *Consider the actual error covariance $\bar{P}^i(k)$ as $k \to \infty$. $\bar{P}^i(\infty)$ is the unique solution that satisfies*

$$\begin{aligned} \bar{P}^i(\infty) \;=\; & (K^i(\infty) - I)\bar{P}^i(\infty)(K^i(\infty) - I)^T \\ & + K^i(\infty) R_0 K^i(\infty)^T + \gamma_i K^i(\infty)\Psi_i \Psi_i^T K^i(\infty)^T \end{aligned} \tag{3.20}$$

Furthermore,

$$\bar{P}^i(\infty) \;=\; \bar{P}^i(\infty)|_{\gamma_i=0} + \gamma_i V_1$$

where

$$\begin{aligned} M &= K^i(\infty)\Psi_i\Psi_i^T K^i(\infty)^T \\ V_1 &= \sum_{k=0}^{\infty}(K^i(\infty) - I)^k M((K^i(\infty) - I)^T)^k \end{aligned}$$

Proof. In terms of the second part of Lemma 4, $I - [P + R_0]^{-1}P$ is stable, which means that$-(I - [P^i(\infty) + R_0]^{-1}P^i(\infty))^T$ is stable. Hence, we can derive that $K^i(\infty) - I$ is also stable. In the light of Lemma 3, $P^i(\infty)$ is the unique solution to (3.20) and we can get

$$\bar{P}^i(\infty) \;=\; \bar{P}^i(\infty)|_{\gamma_i=0} + \gamma_i V_1$$

Hence, this completes the proof of Proposition 2. □

Remark 4. *From Proposition 2, one can see that the actual error covariance $\bar{P}^i(\infty)$ is bounded even if observation is intermittent and there exist uncertainties in the measurement noise. Therefore, we can use the Kalman filter to estimate the position of the odour source.*

Example 1: Consider the following process

$$\begin{aligned} x(k) &= x(k-1) \\ z(k) &= x(k) + \bar{w}(k) \end{aligned}$$

where $\bar{w}(k)$ satisfies a normal distribution with zero mean and 2.1 variance. The uncertain measurement noise variance is assumed to satisfy a normal distribution with zero mean and 10.3 variance. The probabilities of measurement arrival is $P(\lambda_k = 1) = 0.9$, $P(\lambda_k = 1) = 0.5$, and $P(\lambda_k = 1) = 0.1$, respectively. We use the Kalman filter to estimate the state with initial state $\hat{x}_s^i(0) = 0$ and covariance $P^i(0) = 10^3$. Figure 3.1 shows the filtering results for the different probabilities of measurement arrival. From this figure, one can see that process states estimated by Kalman filters converge even though there exist uncertainties in the measurement

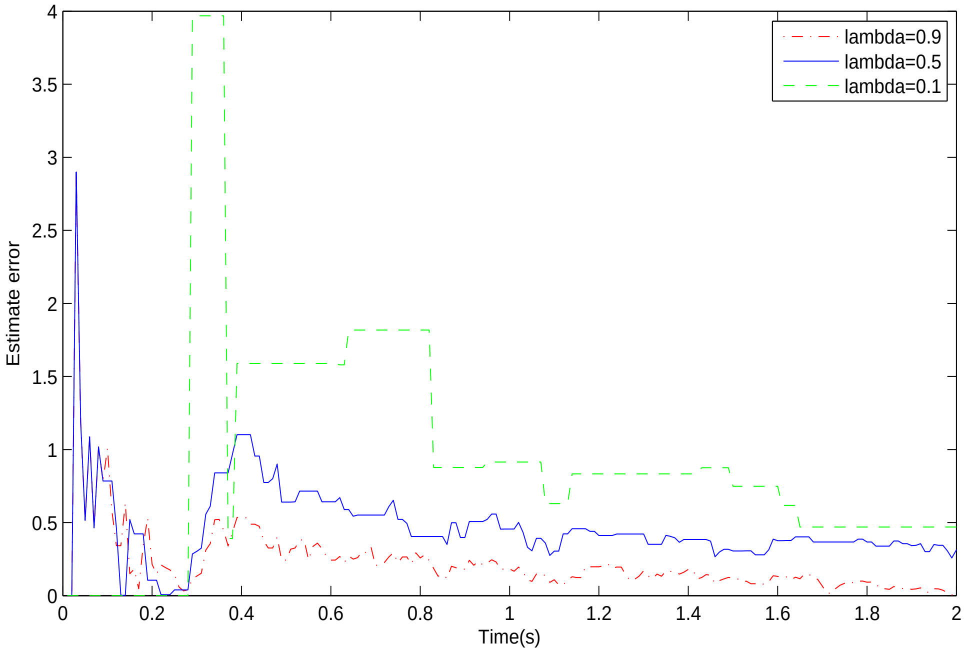


Figure 3.1: Estimate errors for the different probabilities of measurement arrival.

noise and intermittent observations. Moreover, the results are better when we increase the probability of measurement arrival, which means that we need to design an efficiently cooperative control algorithm that can control the robot group to trace the plume in order to obtain more observations.

3.3.3 Information Fusion for Multiple Observations

From the left side of the measurement model (3.9), one can see that the measurement value $z_i(k) = x(k) - \bar{v}(k)$ can be calculated only when the robot detects the odour clues where $x(k) = x_i(k)$ and $x_i(k)$ is the position of robot i at time k. Therefore, there exist two issues that need to be considered. One issue is that there exist noises on the position sensor and wind velocity sensor, which results in uncertainties in the measurement value $z_i(k)$. Another issue is that there exist multiple observations from neighborhood robots, which requires the robot to fuse multiple observations. In the following, we provide a dynamic finite-time consensus fusion algorithm to deal with two issues.

$$\begin{aligned} \dot{x}_i(t) &= \dot{u}_i(t) + \sum_{j=1}^{N} a_{ij}(x_j(t) - x_i(t)) + \lambda(u_i(t) - x_i(t)) \\ &\quad + \beta \text{sig}\left(\sum_{j=1}^{N} a_{ij}(x_j(t) - x_i(t)) + \lambda(u_i(t) - x_i(t)) \right)^a \qquad (3.21\text{a}) \\ y_i(t) &= x_i(t) \qquad (3.21\text{b}) \end{aligned}$$

where $\beta > 0$, $\lambda > 0$, and $0 < a < 1$; $x_i(t)$ is regarded as a decision state for the ith robot; $u_i(t)$ is the input of measurement and $y_i(t)$ is the corresponding fused output. In order to guarantee the convergence of the dynamic finite-time consensus fusion algorithm (3.21), we have the following proposition.

Proposition 3. *Consider the dynamic finite-time consensus fusion algorithm, (3.21) with the input $u_i(t) = r(t)$ and $\lambda > 0$. If the interaction topology $G(\nu, \mathcal{E}, A)$ is undirected and connected, the dynamic finite-time consensus fusion algorithm converges within a finite-time interval $\left[0, \frac{(1+a)V(0)^{\frac{1-a}{1+a}}}{k(1-a)}\right]$, i.e. $x_i(t) \to x_j(t)$ and $x_i(t) \to r(t)$ as $t \to \frac{(1+a)V(0)^{\frac{1-a}{1+a}}}{k(1-a)}$.*

Proof. Let $s_i(t) = x_i(t) - r(t)$ and introduce $s_i(t)$ into (3.21). Then, we have

$$\begin{aligned} \dot{s}_i(t) &= \sum_{j=1}^{N} a_{ij}(s_j(t) - s_i(t)) - \lambda s_i(t) \\ &\quad + \beta \text{sig}\left(\sum_{j=1}^{N} a_{ij}(s_j(t) - s_i(t)) - \lambda s_i(t) \right)^a \qquad (3.22) \end{aligned}$$

Further, set $\phi_i(t) = \sum_{j=1}^{N} a_{ij}(s_j(t) - s_i(t)) - \lambda s_i(t)$. Hence, we can obtain

$$\dot{s}_i(t) = \phi_i(t) + \beta \text{sig}(\phi_i(t))^a \qquad (3.23)$$

Let $s(t) = [s_1(t), \cdots, s_N(t)]^T$ and $\phi(t) = [\phi_1(t), \cdots, \phi_N(t)]^T$. We rewrite (3.23) as

$$\dot{s}(t) = \phi(t) + \beta \text{sig}(\phi(t))^a \qquad (3.24)$$

We choose a Lyapunov candidate as

$$V(t) = \sum_{i=1}^{N} \left(\frac{\beta}{1+a} |\phi_i(t)|^{a+1} + \frac{1}{2} \phi_i(t)^2 \right) \qquad (3.25)$$

Obviously $V(t) \geq 0$ and along the closed-loop trajectories,

$$\begin{aligned}\frac{dV(t)}{dt} &= \sum_{i=1}^{N}(\beta \text{sig}(\phi_i(t))^a \dot{\phi}_i(t) + \phi_i \dot{\phi}_i(t)) \\ &= \sum_{i=1}^{N}(\beta \text{sig}(\phi_i(t))^a \ + \phi_i(t))\dot{\phi}_i(t) \\ &= (\beta \text{sig}(\phi(t))^a \ + \phi(t))^T \dot{\phi}(t)\end{aligned}$$

Moreover, we have

$$\dot{\phi}(t) = -(L(A) + \lambda I)(\beta \text{sig}(\phi(t))^a \ + \phi(t))$$

Hence, we obtain

$$\frac{dV(t)}{dt} = -\xi^T(L(A) + \lambda I)\xi$$

where $\xi = \beta \text{sig}(\phi(t))^a \ + \phi(t)$.

In the light of Lemma 2, we have $\frac{dV(t)}{dt} < 0$. The finite-time consensus fusion algorithm (3.21) is asymptotically stable, i.e. $x_i(t) \rightarrow r(t)$ as $t \rightarrow \infty$. Similar to the proof process in [144], the fusion algorithm (3.21) is a finite-time convergence one and the settling time $T \leq \frac{(1+a)V(0)^{\frac{1-a}{1+a}}}{k(1-a)}$. □

However, it is worth mentioning that the signal $\dot{r}(t)$ is hard to be obtained. Hence, we will use an approximately dynamic finite-time consensus fusion algorithm, which is given by

$$\begin{aligned}\dot{x}_i(t) &= \sum_{j=1}^{N} a_{ij}(x_j(t) - x_i(t)) + \lambda(u_i(t) - x_i(t)) \\ &\quad + \beta \text{sig}\left(\sum_{j=1}^{N} a_{ij}(x_j(t) - x_i(t)) + \lambda(u_i(t) - x_i(t))\right)^a \qquad (3.26a) \\ y_i(t) &= x_i(t) \qquad (3.26b)\end{aligned}$$

Remark 5. *It is obvious to see that if $\beta = 0$ and $\lambda = 1$, then the dynamic finite-time consensus fusion algorithm described by (3.26) becomes a linear consensus algorithm, which was studied by R. Olfati-Saber* et al. *(2005) [109, 105], R. A. Freeman* et al.

(2006) [39, 38], and P. Yang [145] (2008). Hence, the linear consensus algorithm described in [109, 105, 39, 38, 145] can be regarded as special cases of the nonlinear consensus algorithm (3.26).

For the proposed consensus fusion algorithm (3.26), we have the following convergence proposition.

Proposition 4. *Let $r(t)$ be a signal with a uniformly bounded rate $|\dot{r}(t)| \leq \nu$. Consider the dynamic finite-time consensus fusion algorithm (3.26) with the input $u_i(t) = r(t)$, $\lambda > 0$, and $\epsilon = \max\left\{ \frac{\nu\sqrt{N}}{\lambda_{min}(L(A)+\lambda I)(\beta+1)}, \frac{\nu\sqrt{N}}{\lambda_{min}(L(A)+\lambda I)} - \frac{\beta\sqrt{N}}{||L(A)+\lambda I||_2} \right\}$. If the interaction topology $G(\nu, \mathcal{E}, A)$ is undirected and connected, the dynamic finite-time consensus fusion algorithm is a ϵ-convergence within a finite-time interval.*

Proof. Defining the error variable $s_i(t) = x_i(t) - r(t)$ gives

$$\begin{aligned} \dot{s}_i(t) &= -\dot{r}(t) + \sum_{j=1}^{N} a_{ij}(s_j(t) - s_i(t)) - \lambda s_i(t) \\ &\quad +\beta \mathrm{sig}\left(\sum_{j=1}^{N} a_{ij}(s_j(t) - s_i(t)) - \lambda s_i(t) \right)^a \end{aligned} \tag{3.27}$$

Let

$$\phi_i(t) = \sum_{j=1}^{N} a_{ij}(s_j(t) - s_i(t)) - \lambda s_i(t) \tag{3.28}$$

and $s(t) = [s_1(t), \cdots, s_N(t)]^T$ and $\phi(t) = [\phi_1(t), \cdots, \phi_N(t)]^T$. We can rewrite (3.27) as

$$\dot{s}(t) = -\mathbf{1}_N \dot{r}(t) + \phi(t) + \beta \mathrm{sig}(\phi(t))^a \tag{3.29}$$

where $\mathbf{1}_N = [\underbrace{1, 1, \ldots, 1}_{N}]^T$.

We choose a Lyapunov candidate as

$$V(t) = \sum_{i=1}^{N} \left(\frac{\beta}{1+a} |\phi_i(t)|^{a+1} + \frac{1}{2}\phi_i(t)^2 \right) \tag{3.30}$$

Obviously $V(t) \geq 0$ and along the closed-loop trajectories,

$$\begin{aligned}\frac{dV(t)}{dt} &= \sum_{i=1}^{N}(\beta \text{sig}(\phi_i(t))^a \dot{\phi}_i(t) + \phi_i \dot{\phi}_i(t)) \\ &= \sum_{i=1}^{N}(\beta \text{sig}(\phi_i(t))^a \ + \phi_i(t))\dot{\phi}_i(t) \\ &= (\beta \text{sig}(\phi(t))^a \ + \phi(t))^T \dot{\phi}(t)\end{aligned}$$

Moreover, we have

$$\dot{\phi}(t) = -(L(A) + \lambda I)\dot{s}(t)$$

Hence, we obtain

$$\begin{aligned}\frac{dV(t)}{dt} &= -\xi^T(L(A) + \lambda I)\xi + \xi^T(L(A) + \lambda I)\mathbf{1}_N \dot{r}(t) \\ &\leq -\lambda_{min}(L(A) + \lambda I)||\xi||_2^2 \\ &+\nu\sqrt{N}||L(A) + \lambda I||_2||\xi||_2\end{aligned}$$

where $\xi = \beta \text{sig}(\phi(t))^a \ + \phi(t)$. Furthermore, when

$$||\xi||_2 > \frac{\nu\sqrt{N}||L(A) + \lambda I||_2}{\lambda_{min}(L(A) + \lambda I)}$$

we get $\frac{dV(t)}{dt} < 0$. In addition, we have

$$\begin{aligned}||\xi||_2 &= ||\beta \text{sig}(\phi(t))^a + \phi(t)||_2 \\ &\leq ||\beta \text{sig}(\phi(t))^a||_2 + ||\phi(t)||_2\end{aligned}$$

If $|\phi_i(t)| \geq 1$, we can obtain

$$\begin{aligned}||\xi||_2 &\leq \beta||\phi(t)||_2 + ||\phi(t)||_2 \\ &\leq (\beta + 1)||L(A) + \lambda I||_2||s(t)||_2\end{aligned}$$

Hence, we derive

$$||s(t)||_2 \geq \frac{\nu\sqrt{N}}{\lambda_{min}(L(A) + \lambda I)(\beta + 1)}$$

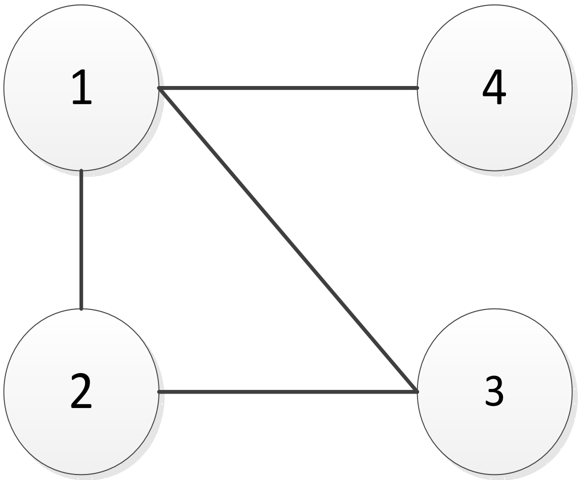


Figure 3.2: The communication topology among four robots.

If $0 \leq |\phi_i(t)| < 1$, we can obtain

$$||\xi||_2 \leq \beta\sqrt{N} + ||L(A) + \lambda I||_2||s(t)||_2$$

Therefore, we get

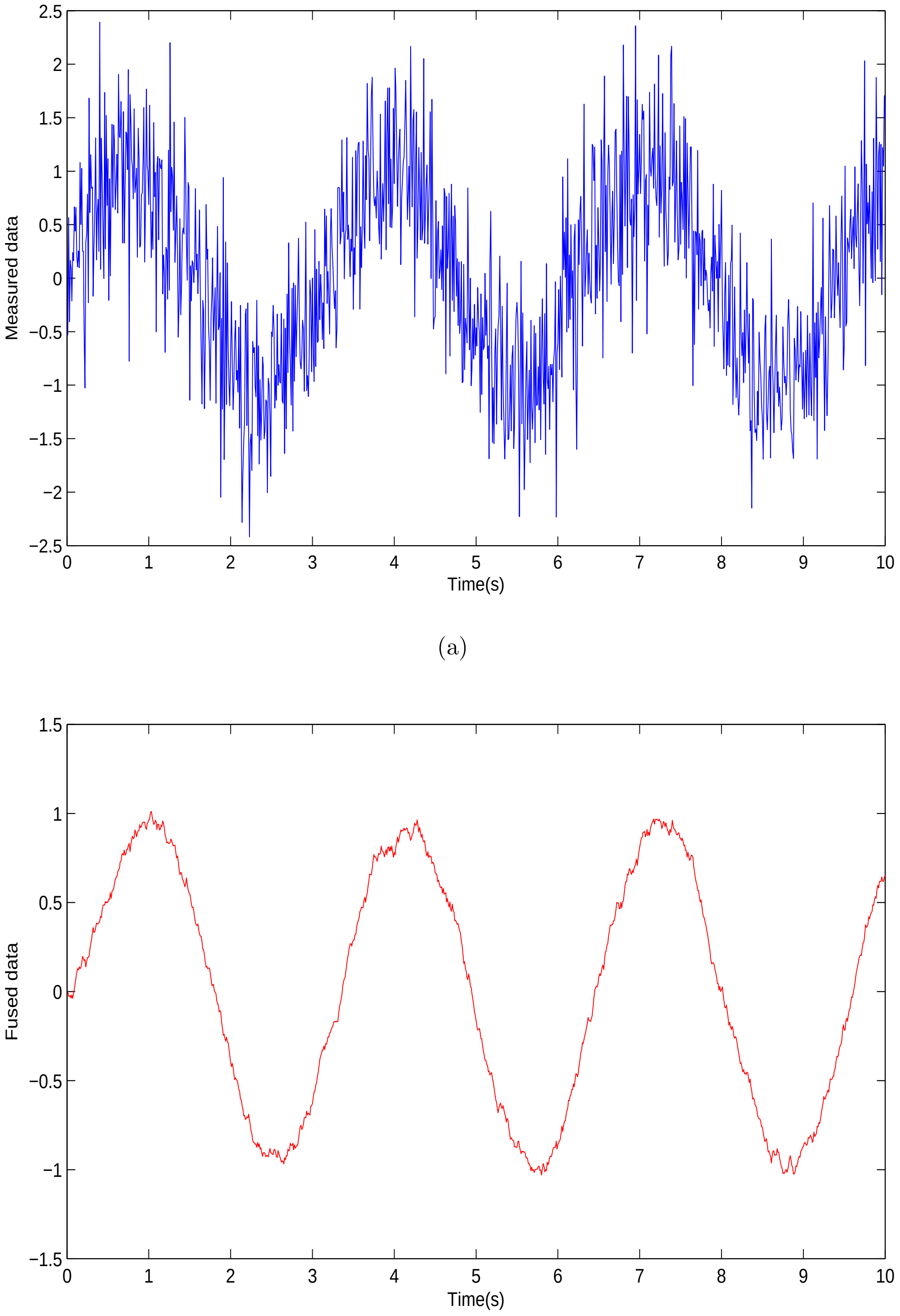
$$||s(t)||_2 \geq \frac{\nu\sqrt{N}}{\lambda_{min}(L(A) + \lambda I)} - \frac{\beta\sqrt{N}}{||L(A) + \lambda I||_2}$$

Let B_ρ be a closed ball centered at $s(t) = 0$ with radius

$$\rho = \max\left\{\frac{\nu\sqrt{N}}{\lambda_{min}(L(A) + \lambda I)(\beta + 1)}, \frac{\nu\sqrt{N}}{\lambda_{min}(L(A) + \lambda I)} - \frac{\beta\sqrt{N}}{||L(A) + \lambda I||_2}\right\}$$

As a consequence, any solution (3.27) starting in R^N/B_ρ satisfies $\frac{dV(t)}{dt} < 0$. Thus, it enters B_ρ in finite time and remains in B_ρ thereafter. This guarantees finite-time ϵ-stability of $s(t) = 0$ with a radius $\epsilon = \rho$. □

Remark 6. *From the proposed dynamic finite-time consensus fusion algorithm (3.26), one can see that each robot can access to the observations $u_j(t)$ detected by the robot j via the state $x_j(t)$ transferred by communication networks. Because the state $x_i(t)$ of the fusion algorithm (3.26) can track the average measurement value, the dynamic finite-time consensus fusion algorithm (3.26) acts as a low-pass filter.*



(a)

(b)

Figure 3.3: (a) sensor measurements $u_i(t) = r(t) + \omega_i(t)$ and (b) fused data $y_i(t)$ via the proposed finite-time consensus fusion algorithm (3.26).

Remark 7. *From Proposition 4, one can see that ϵ will be decreased if we increase the parameter β, which means that the state $x_i(t)$ rapidly reaches $r(t)$ in finite time with the lesser stable error. Therefore, we can adjust the parameter β to obtain a better fusion result.*

Example 2: Let the Figure 3.2 show the communication topology among four robots. It is obvious that the communication topology is undirected and connected. Set parameters $\beta = 4$, $\lambda = 0.8$, $a = 0.5$ and the initial state $x_i(t) = 0, i = 1, 2, 3, 4$ for the finite-time consensus fusion algorithm (3.26). The sensing model is $u_i(t) = r(t) + \omega_i(t)$ where $r(t) = \sin(2t)$ and the covariance of $\omega_i(t)$ is 0.3. Figure 3.3(a) shows the sensor measurements for the first robot and Figure 3.3(b) shows the fused sensor data via the proposed dynamic finite-time consensus fusion algorithm (3.26).

3.3.4 A Distributed Decision Algorithm

In terms of the probable position of the odour source $y_i(k)$ in (3.26), we use concentration information $\hat{p}_i(k)$ to obtain the final estimated position of the odour source, which is described by

$$h^{ic}(k) \quad = \quad c_1 y_i(k) + c_2 \hat{p}_i(k) \tag{3.31}$$

where $y_i(k)$ is the position of the odour source estimated by the ith robot in (3.26); $\hat{p}_i(k)$ denotes the position of the odour source estimated by the ith robot in (2.4); and $c_j (j = 1, 2)$ are weighted parameters and satisfy $c_1 + c_2 = 1$. To sum up, the distributed decision algorithm is described in Algorithm 1.

3.4 Odour Source Localisation

In order to validate the effectiveness of the distributed decision algorithm, we will plan the movement direction of the robot group based on the estimated position of the odour source. Then, we will design a simple cooperative control law. Finally, we will show the simulation results of the distributed decision algorithm.

Algorithm 1 A distributed decision algorithm

```
/*Initialization*/
Initialize parameters $\alpha$, $\beta$, $a$, $c_1$, $c_2$ and $\lambda$;
Initialize *a posteriori* estimate error covariance $P^i(0)$ and a measurement noise covariance matrix $R(0)$;
Initialize fusion algorithm's state in (3.26) and the source estimate position $\hat{x}_s(0)$ using the current position $x_i(t)$ when the first concentration detection event occurs;
Set $robot.l = 0$ and $robot.windcount = 0$; /*Releasing time and an accumulator that can record the number of wind velocity*/
/*Main Body*/
**repeat**
   /*Store wind velocity within 100s*/
   $robot.wind[robot.windcount] = wind$;
   Perform (2.4) to get $\hat{p}_i(k)$.
   /*Concentration detection events occur*/
   **if** $robot.snsd > 0$ **then**
      /*Calculate the movement distance of filaments*/
      Set $sumtemp = 0$ and $kk = 0$;
      **for** $i = 0$; $i < robot.windcount$; $i++$ **do**
         Set $sum = 0$;
         **for** $j = i$; $j < robot.windcount$; $j++$ **do**
            $sum = sum + robot.wind[j]$;
         **end for**
         **if** $sum < robotposition + \alpha \wedge sum > robotposition - \alpha$ **then**
            $sumtemp = sumtemp + sum$;$kk = kk + 1$;
         **end if**
      **end for**
      Calculate the measurement $z_i(k) = robotposition - sumtemp/kk$;
   **end if**
   Perform (3.11) and (3.12) to obtain *a priori* position of the odour source $\hat{x}_s^{-i}(k)$ and *a priori* estimate error covariance $P^{-i}(k)$;
   /*Calculate measurement noise variance*/
   Set $kk = 0$ and $sum = 0$;
   **for** $i = robot.l$; $i < \text{int}(CurrentTime)$; $i++$ **do**
      $sum = sum + (\text{int}(CurrentTime) - i) * R(k-1)$; $kk = kk + 1$;
   **end for**
   $R(k) = sum/kk$;
   Perform (3.13) to calculate the Kalman gain;
   Perform (3.15) to calculate *a posteriori* estimate error covariance $P^i(k)$;
   Perform (3.14) to generate *a posteriori* position estimate of the odour source $\hat{x}_s^i(k)$;
   Perform (3.26) to obtain the fused measurement $y_i(k)$;
   Perform (3.31) to generate the final position estimate of the odour source $h^{ic}(k)$;
**until** Termination conditions are satisfied.
```

3.4.1 The Planning of Movement Direction of the Robot Group

In this subsection, we use a different movement planning method from Chapter 2 to trace the plume. The target position $p_i(k)$ of the ith robot is shown in Figure 3.4. From this figure, one can see that the robot probably detects the plume at the direction from the current position to the target position under an assumption that filaments that are released at the estimated position $h^{ic}(k)$ may move along the current wind direction. In order to let the robot detect the odour and move toward the estimated position, the target position of the robot can be calculated according to (3.32) where a center point between the estimated position and the current position at the x direction is chosen due to simple computation. Moreover, the short blue arrow in Figure 3.4 denotes the wind direction and magnitude. Further, $h^{ic}_x(k)$ and $h^{ic}_y(k)$ denote the coordinates of the estimated position at the x axis and y axis, respectively. x_i is the current position of the robot. $x_{xi}(k)$ and $x_{yi}(k)$ are coordinates of x_i at the x axis and y axis, respectively. Similarly, w_x and w_y are the velocities of wind at the x and y axis, respectively. $\frac{|x_{xi}(k)-h^{ic}_x(k)|}{2}$ is the coordinate of the goal position at the x axis if the estimated position is regarded as the origin. Based on the rule of similar triangles, $\mathrm{sign}(w_y) \times \frac{|x_{xi}(k)-h^{ic}_x(k)|}{2} \times \frac{|w_y|}{w_x}$ is the coordinate of the goal position at the y axis if the estimate position is regarded as the origin. Therefore, in terms of the coordinate system shown in Figure 3.4, we have the goal position described by

$$
\begin{aligned}
p_{xi}(k) &= h^{ic}_x(k) + \frac{|x_{xi}(k) - h^{ic}_x(k)|}{2} && (3.32a)\\
p_{yi}(k) &= h^{ic}_y(k) + \mathrm{sign}(w_y) \times \frac{|x_{xi}(k) - h^{ic}_x(k)|}{2} \times \frac{|w_y|}{w_x} && (3.32b)
\end{aligned}
$$

where $p_{xi}(k)$ and $p_{yi}(k)$ are the coordinates of the target position $p_i(k)$ of the ith robot at the x and y axis, respectively.

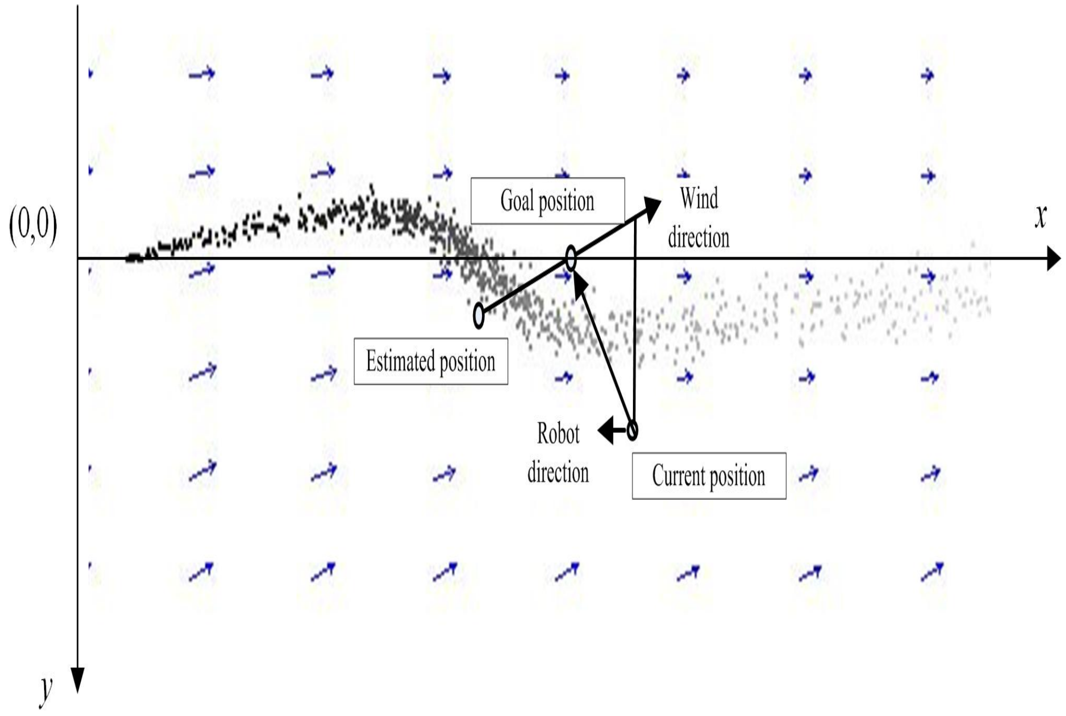


Figure 3.4: A target position of the ith robot.

3.4.2 A Cooperative Control Algorithm

The robot will move along the direction from the current position to the target position. Moreover, if time of the robot group failing to detect the plume is over 20s, the robot group will move toward the estimated position $h^{ic}(k)$. Based on these movement strategies, we give the following finite-time control law as

$$\begin{aligned} u_i(t) &= \dot{v}_c + \sum_{j=1}^{N} a_{ij}(v_j - v_i) + \delta(v_c - v_i) \\ &\quad + \mathrm{sig}\left(\sum_{j=1}^{N} a_{ij}(v_j - v_i) + \delta(v_c - v_i)\right)^a \end{aligned} \tag{3.33}$$

where $0 < a < 1$, $\delta > 0$ and v_c is a reference velocity of the robot group and is decided by the above movement strategies. We have the following proposition to guarantee the stability of the system (1.7) under the control law (3.33).

Proposition 5. *Consider the robot dynamics (1.7) with the control input (3.33). If the interaction topology $G(\nu, \mathcal{E}, A)$ is undirected and connected, the control law* *guarantees $v_i \to v_j$ and $v_i \to v_c$ within a finite-time interval* $\left[0, \frac{(1+a)V(0)^{\frac{1-a}{1+a}}}{k(1-a)}\right]$.

Proof. Let $q_i = v_i - v_c$ and $p_i = x_i - \int v_c dt$. Following the similar step of Proposition 3, we can prove that the proposed control law (3.33) guarantees $v_i \to v_j$ and $v_i \to v_c$ within a finite-time interval $\left[0, \frac{(1+a)V(0)^{\frac{1-a}{1+a}}}{k(1-a)}\right]$. □

However, each robot has its individual reference velocity based on the distributed decision algorithm and movement strategies. Hence, the control law (3.33) can be modified by

$$\begin{aligned} u_i(t) &= \dot{v}_c^i + \sum_{j=1}^{N} a_{ij}(v_j - v_i) + \delta(v_c^i - v_i) \\ &\quad + \mathrm{sig}\left(\sum_{j=1}^{N} a_{ij}(v_j - v_i) + \delta(v_c^i - v_i)\right)^a \end{aligned} \tag{3.34}$$

where v_c^i is the reference velocity of the ith robot.

Moreover, in order to provide robots with obstacle avoidance capabilities, we give the following control law which was proposed by Choi *et al.* (2009) [14].

$$u_i(t) = -\nabla U(x_i) \tag{3.35}$$

where $\nabla U(x_i)$ is the gradient of the potential U with respect to x_i for robot i.

$$\begin{aligned} U &= \sum_i \sum_{j\in N(i)} U_{ij}(||x_i - x_j||^2) \\ &= \sum_i \sum_{j\in N(i)} U_{ij}(r_{ij}) \end{aligned} \tag{3.36}$$

where $N(i)$ denotes a set among which each robot can communicate with the robot i, and

$$U_{ij}(r_{ij}) = \frac{1}{2}\left(\log(\kappa + r_{ij}) + \frac{\kappa + d^2}{\kappa + r_{ij}}\right)$$

where $r_{ij} = ||x_i - x_j||^2$; κ and d are parameters. If the distance between any two robots is smaller than d_0, we use the control law (3.35) and (3.34) otherwise.

Table 3.1: The parameters used in the distributed decision algorithm.

β	λ	a	c_1	c_2	α(m)	Sampling time(s)
8	0.8	0.5	0.8	0.2	40	1

Table 3.2: The parameters used in (3.34) and (3.35) for the cooperative control algorithm.

a	δ	κ	d_0	d	v_{max}(m/s)	ω_{max}(rad/s)
0.5	0.2	0.5	6.6	4	0.8	1.57

Example 3: Consider three robots. Set parameters $\delta = 0.1$, $a = 0.5$, $d_0 = 6.6$, $\kappa = 0.8$, and $d = 4$. Let $v_c = [9 - 9]$. Figure 3.5 shows the parallel movement of three robots under the control laws (3.34) and (3.35).

3.4.3 Simulation Results

In this subsection, we will test the proposed distributed decision algorithm (DDA) which parameters are shown in Table 3.1. It should be pointed out that the parameters β and λ in Table 3.1 are to guarantee the convergence of the proposed dynamic finite-time consensus fusion algorithm. α can influence the quality of data received. Moreover, we use the given cooperative control algorithm (3.34) and (3.35) which parameters are shown in Table 3.2 where the parameters v_{max} and ω_{max} are to limit the maximum linear velocity and angular velocity of robots, respectively. Further, we utilize a circle where the real position of the odour source is viewed as a center with a predefined radius $1m$ as one of termination conditions, which means that the search task is finished if any robot enters the circle. (The other termination condition is the maximal search time $1500s$). In addition, it is worth mentioning that the robot group will search for the odour clues along the direction of y axis from the initial positions (right-up corner) to the (80m,50m). Once the odour clues are detected by any robot, the proposed DDA will start to run.

The motion process of the robot group with five robots under all to all communication topology is illustrated in Figures 3.6-3.9. In Figure 3.6, the initial positions of the robot group are set at the right-up corner in the search region. In Figure 3.7 and Figure 3.8, the parallel motion is used to track the plume and to move along the plume in terms of the probable positions of the odour source given by the distributed decision algorithm. In Figure 3.9, the robot group finds the real odour source. The corresponding prediction distribution of the position of the odour source is shown in Figures 3.10-3.13. The prediction errors of five robots about the position of the odour source are shown in Figure 3.14 where the robots keep the predefined position (80m,0m) of the odour source from 0s to about 40s. After about 40s, the robot group detects the odour clues, and then the proposed DDA starts to run. It is worth mentioning that the prediction accuracy of the position of the odour source is higher if more odour information can be obtained, which implies that we need to enable the robot group to rapidly arrive the target position.

3.5 Conclusion

We have proposed a distributed decision algorithm, which is used to make a decision on the position of the odour source. We have first derived an observation model for the position of the odour source based on wind information. In terms of the observation, we have used a Kalman filter to estimate the position of the odour source. Moreover, we have developed a dynamic finite-time consensus fusion algorithm to fuse observations from other robots. As a final decision on the position of the odour source, we have proposed a distributed decision algorithm, which combines the source position predicted by concentration information with one estimated by wind information. Finally, we have illustrated the effectiveness of the proposed distributed decision algorithm for the problem of odour source localisation.

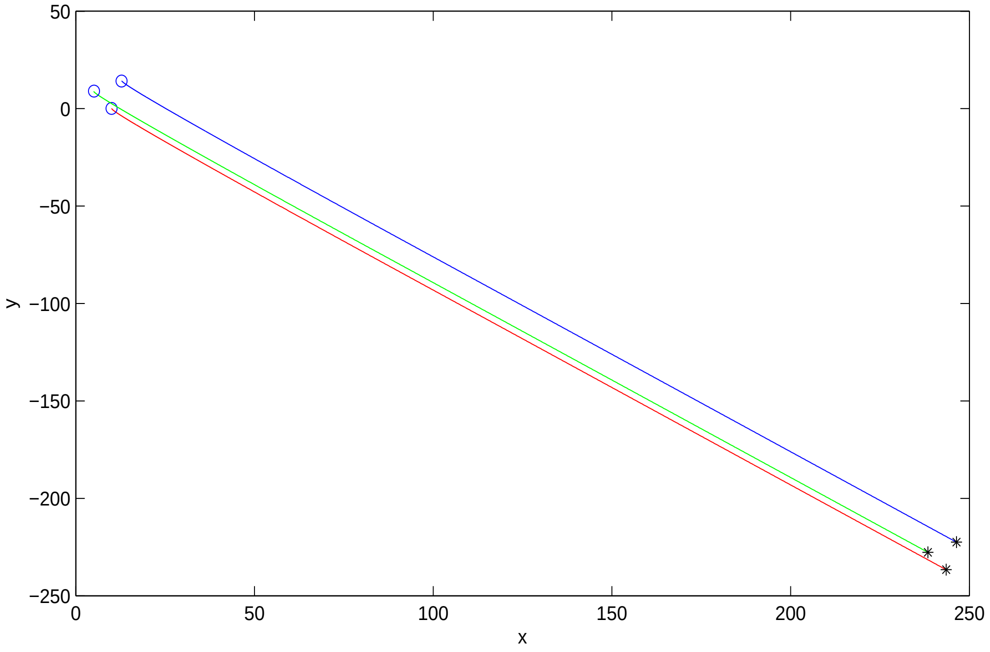


Figure 3.5: The parallel movement for three robots. "o" and "*" denote the initial position and the end position, respectively.

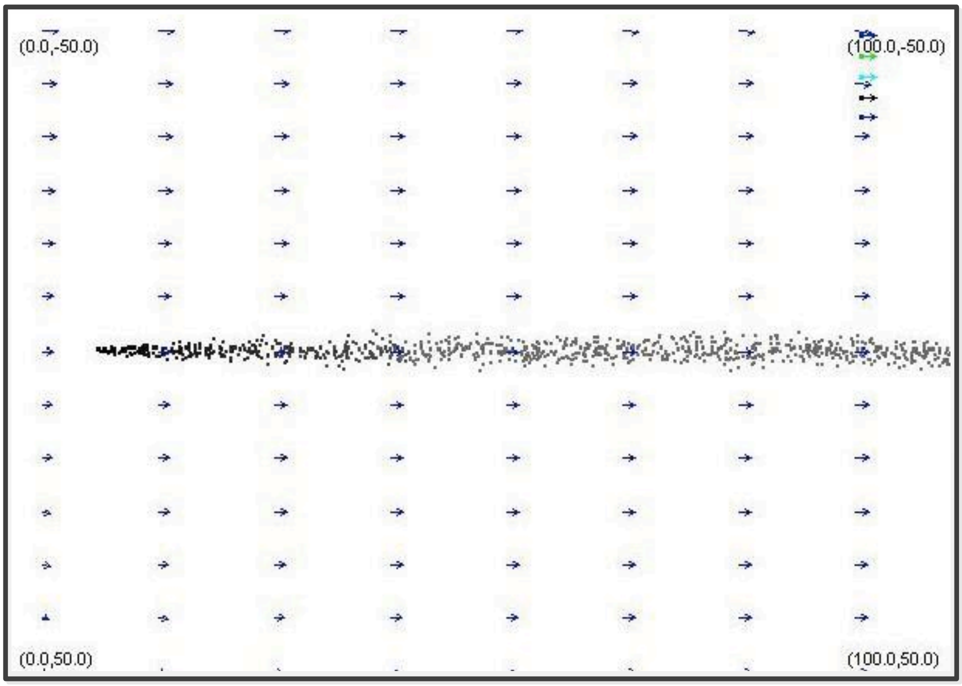


Figure 3.6: The search process of five robots at $T = 0s$.

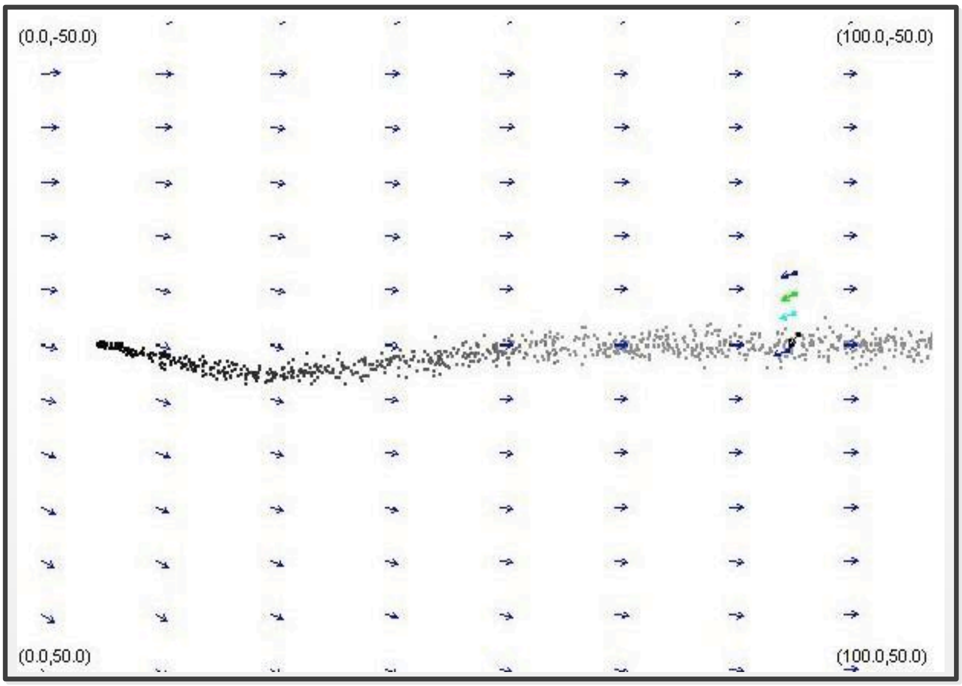


Figure 3.7: The search process of five robots at $T = 47s$.

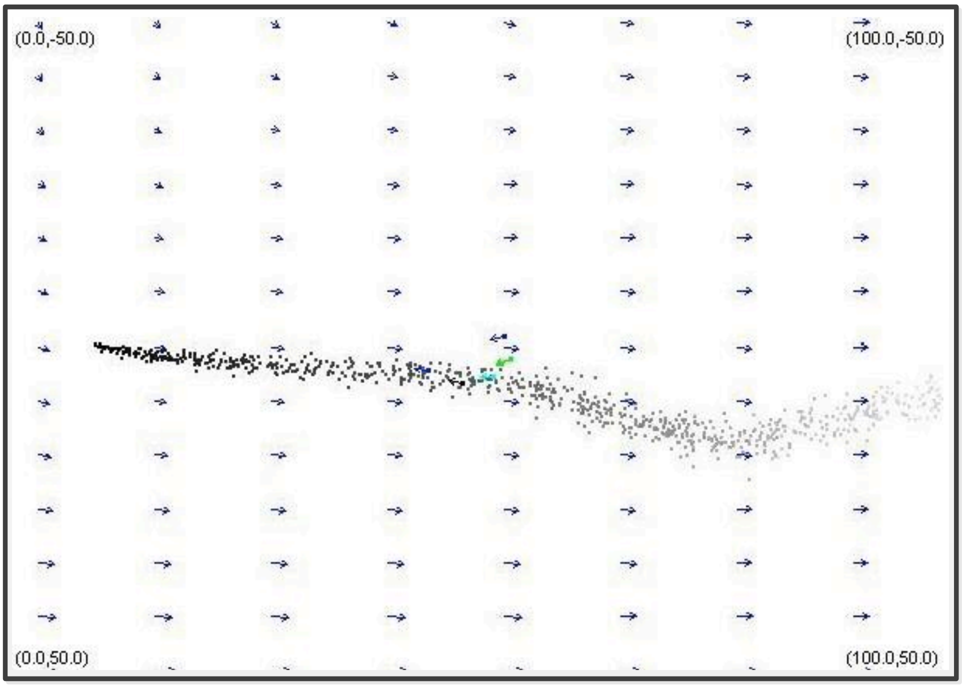


Figure 3.8: The search process of five robots at $T = 98s$.

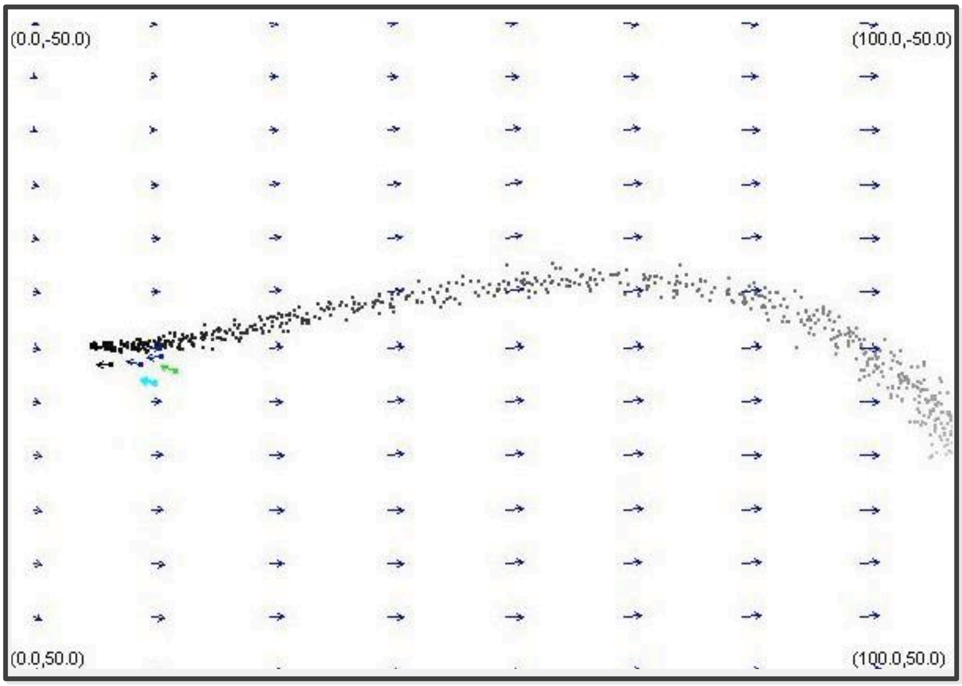


Figure 3.9: The search process of five robots at $T = 185s$.

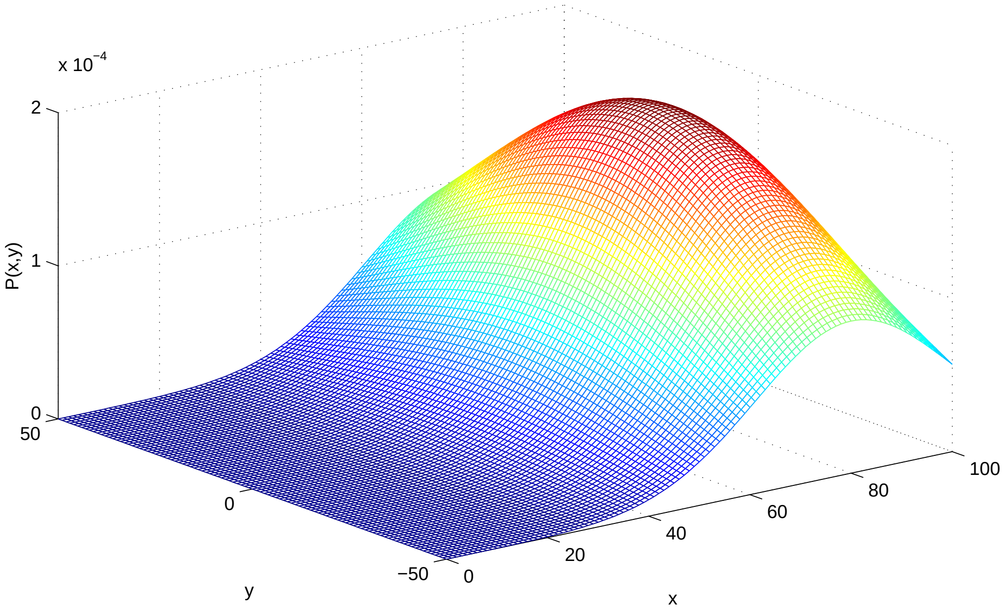


Figure 3.10: The probability distribution of the position of the odour source estimated by five robots at $T = 0s$.

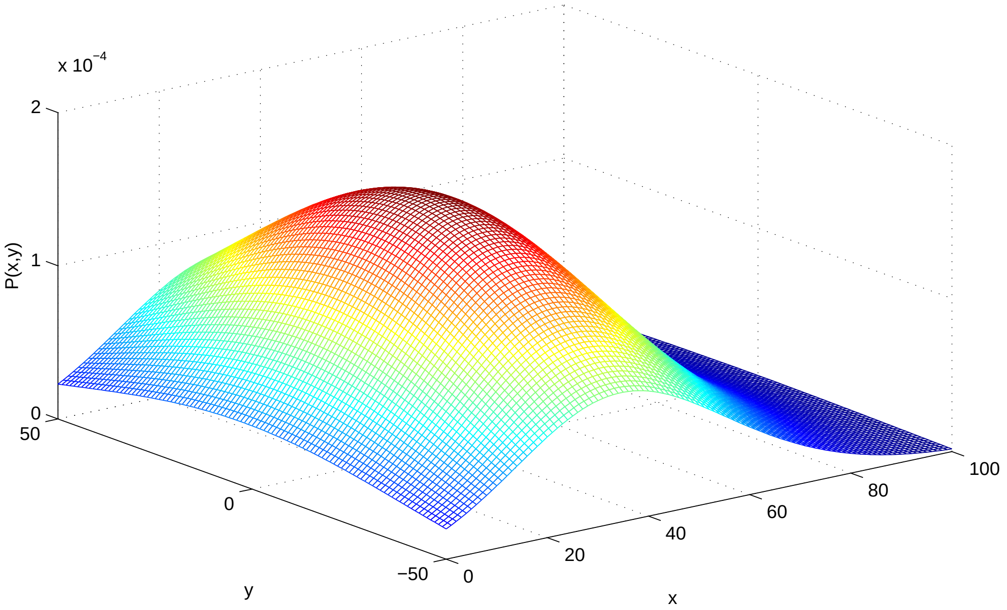


Figure 3.11: The probability distribution of the position of the odour source estimated by five robots at $T = 47s$.

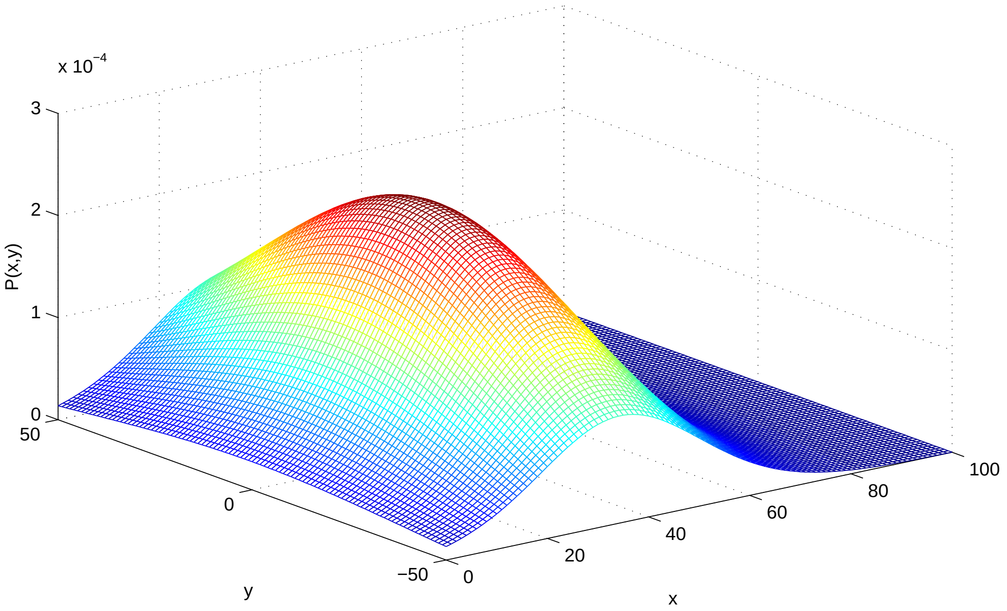


Figure 3.12: The probability distribution of the position of the odour source estimated by five robots at $T = 98s$.

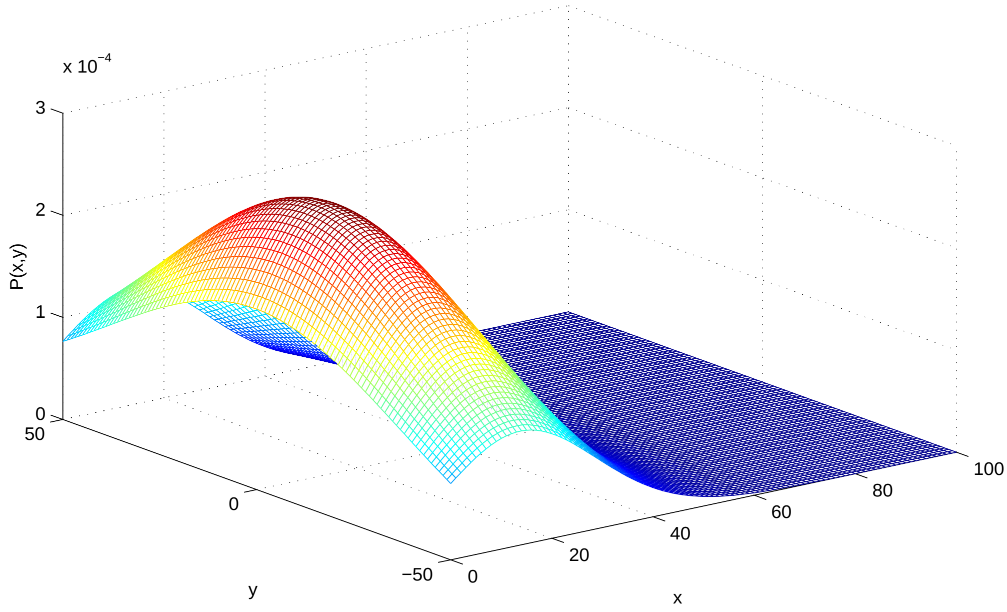


Figure 3.13: The probability distribution of the position of the odour source estimated by five robots at $T = 185s$.

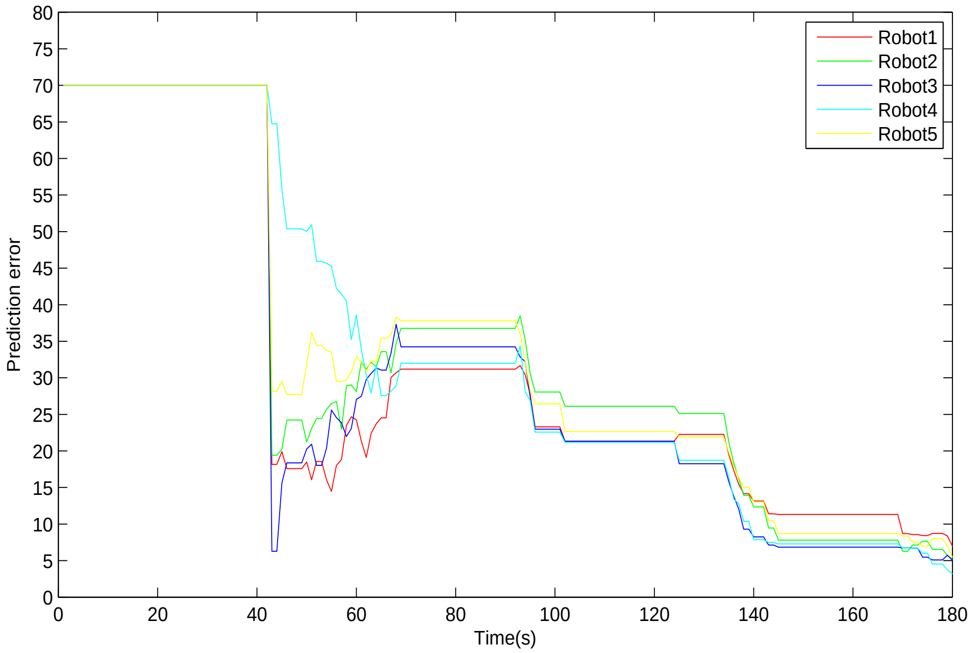


Figure 3.14: Prediction error $\|h^{ic}(k) - x_s\|_2$ over time. x_s is a real position of the odour source.

Chapter 4

Particle Swarm Optimization Based Finite-Time Motion Control

4.1 Introduction

Accordingly, in the control level of the distributed coordination control architecture presented in Chapter 2, a PSO-based finite-time motion control algorithm will be designed in Section 4.2. In Section 4.3, we will propose a discrete-time finite-time PSO (FPSO) algorithm and prove its convergence. Then, the benchmark functions will be used to validate the performance capabilities of the discrete-time FPSO algorithm in Section 4.4. The effectiveness of the PSO-based finite-time motion control algorithm for odour source localisation will be illustrated in Section 4.5. Finally, we will give the conclusion in Section 4.6.

4.2 A PSO-Based Finite-Time Motion Control Algorithm

In this section, we will briefly describe the continuous model of the PSO algorithm. Then, we will derive a continuous-time FPSO algorithm that is used as the PSO-based finite-time motion control algorithm. Finally, we will analyze its finite-time convergence property by using a Lyapunov approach.

4.2.1 A Continuous-Time Model of the PSO Algorithm

The stochastic differential model of the PSO algorithm, i.e. the continuous-time model of the PSO algorithm, can be given by

$$\ddot{x}_i(t) + (1-\omega)\dot{x}_i(t) + \alpha x_i(t) = \alpha_1 x_l(t) + \alpha_2 x_g(t) \tag{4.1}$$

with

$$\begin{aligned} \alpha &= \alpha_1 + \alpha_2 \\ x_i(0) &= x_0 \\ \dot{x}_i(0) &= v_0 \end{aligned}$$

where $x_l(t)$ and $x_g(t)$ are the trajectories of the local and global best positions associated with the ith ($i \in l_N$) particle, respectively; ω is an inertial factor; α_j ($j = 1, 2$) are random variables; $x_i(0)$ and $\dot{x}_i(0)$ are the initial states at time $t = 0$.

This continuous-time PSO model is derived by Fernández Martínez *et al.* [33] (2008) in terms of physical analogy with a damped mass-spring oscillator. We use the following discretization scheme in (4.2), and then introduce it into the stochastic differential model of the PSO algorithm in (4.1).

$$\dot{x}_i(t) \simeq \frac{x_i(t) - x_i(t-\Delta t)}{\Delta t} \tag{4.2a}$$

$$\ddot{x}_i(t) \simeq \frac{x_i(t+\Delta t) - 2x_i(t) + x_i(t-\Delta t)}{\Delta t^2} \tag{4.2b}$$

Consider one case, i.e. $\Delta t = 1$ and $t = k$, we have

$$x_i(k+1) = -(\alpha - \omega - 1)x_i(k) - \omega x_i(k-1) + \alpha_1 x_l(k) + \alpha_2 x_g(k) \tag{4.3}$$

with

$$\begin{aligned} x_i(0) &= x_0 \\ x_i(1) &= (1-\alpha)x_0 + \omega v_0 + \alpha_1 x_l(0) + \alpha_2 x_g(0) \end{aligned}$$

Moreover, the difference form of the PSO algorithm in (4.3) can be rewritten in terms of the position and velocity. Consequently, we can derive the most commonly used form, which is presented by (2.1).

Consider another case, i.e. $\Delta t = \Delta k > 0$ and $t = k$, we can obtain the generalized PSO (GPSO) algorithm [31] as

$$x_i(k + \Delta k) = \gamma_1 x_i(k) + \gamma_2 x_i(k - \Delta k) + \Delta k^2(\alpha_1 x_l(k) + \alpha_2 x_g(k)) \tag{4.4}$$

with

$$\begin{aligned} \gamma_1 &= 2 - (1 - \omega)\Delta k - \alpha \Delta k^2 \\ \gamma_2 &= (1 - \omega)\Delta k - 1 \end{aligned}$$

We rewrite the equation (4.4) based on the position and velocity $(x_i(k), v_i(k))$ as

$$\begin{aligned} v_i(k + \Delta k) &= (1 - (1 - \omega)\Delta k)v_i(k) \\ &\quad + \alpha \Delta k \left(\frac{\alpha_1 x_l(k) + \alpha_2 x_g(k)}{\alpha_1 + \alpha_2} - x_i(k) \right) && \text{(4.5a)} \\ x_i(k + \Delta k) &= x_i(k) + v_i(k + \Delta k)\Delta k && \text{(4.5b)} \end{aligned}$$

As pointed out by Fernández Martínez *et al.* [33] (2008), the particle swarm movement controlled by the GPSO algorithm becomes more elastic and less damped when $\Delta k \to 0$.

Remark 8. *Both the PSO algorithm (2.1) and the GPSO algorithm (4.5) can be derived according to the same the continuous-time model (4.1) under different conditions. The motivation of which Fernández Martínez* et al. *[33] (2008) proposed the GPSO algorithm is that the search performance of the GPSO algorithm approaches that of the corresponding continuous-time model when $\Delta k \to 0$.*

4.2.2 A Continuous-Time Model of the FPSO Algorithm

In this subsection, in the light of the continuous model of the PSO algorithm (4.1), we will first give a continuous-time FPSO algorithm, and then present remarks about parameters introduced by the continuous-time FPSO algorithm. Let $\xi_i(t) = x_i(t) - p_i(t)$ where $p_i(t)$ is described in (2.4) and introduce $\xi_i(t)$ into the continuous model of the PSO algorithm in (4.1). We have

$$\ddot{\xi}_i(t) + (1-\omega)\dot{\xi}_i(t) + \alpha\xi_i(t) = -\ddot{p}_i(t) - (1-\omega)\dot{p}_i(t) \tag{4.6}$$

Moreover, set $y_1(t) = \xi_i(t)$ and $y_2(t) = \dot{\xi}_i(t)$. Then, the equation (4.6) is rewritten as

$$\dot{y}_1(t) = y_2(t) \tag{4.7a}$$

$$\dot{y}_2(t) = -(1-\omega)y_2(t) - \alpha y_1(t) - (1-\omega)\dot{p}_i(t) - \ddot{p}_i(t) \tag{4.7b}$$

In the stagnation and deterministic case, i.e., $p_i(t)$ is stable and α is a constant, we have

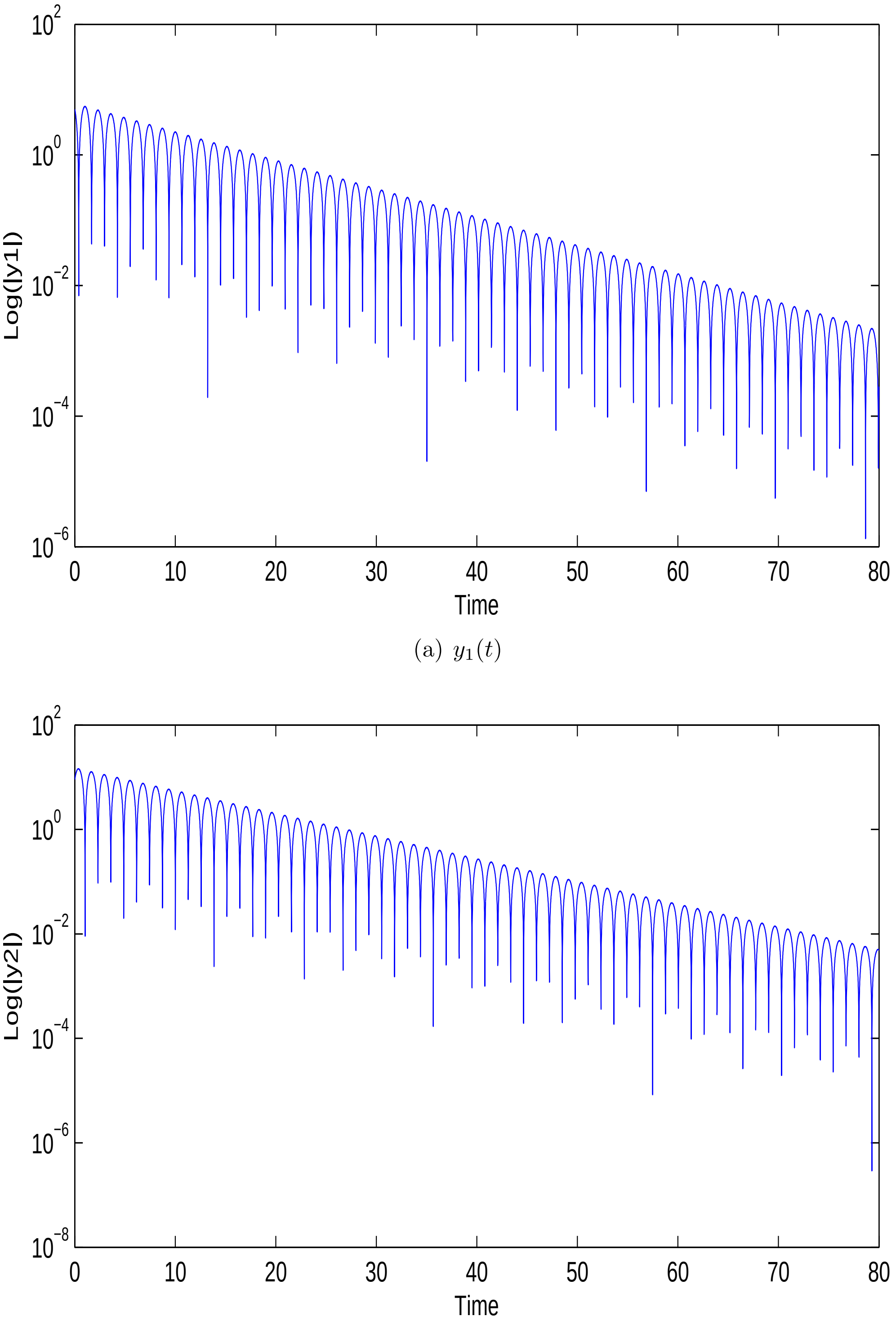
$$\dot{y}_1(t) = y_2(t) \tag{4.8a}$$

$$\dot{y}_2(t) = -(1-\omega)y_2(t) - \alpha y_1(t) \tag{4.8b}$$

The system described by (4.8) is asymptotically stable at the origin when $t \to \infty$, if the parameters ω and α satisfy $\omega < 1$ and $\alpha > 0$, respectively. Figure 4.1 illustrates the asymptotical stability of the system given by (4.8).

Remark 9. *It is obvious that the system described by (4.7) is not stable at the origin if the $p_i(t)$ is time-varying, which enables us to consider the stable system (4.8). Moreover, one can see that the continuous-time PSO algorithm (4.7) does not possess a good tracking performance, which may result in the higher oscillation of convergence results for a class of optimization problems. Therefore, this issue motivates us to introduce the control theory to modify the continuous-time PSO algorithm (4.7) under a given decision on $p_i(t)$.*

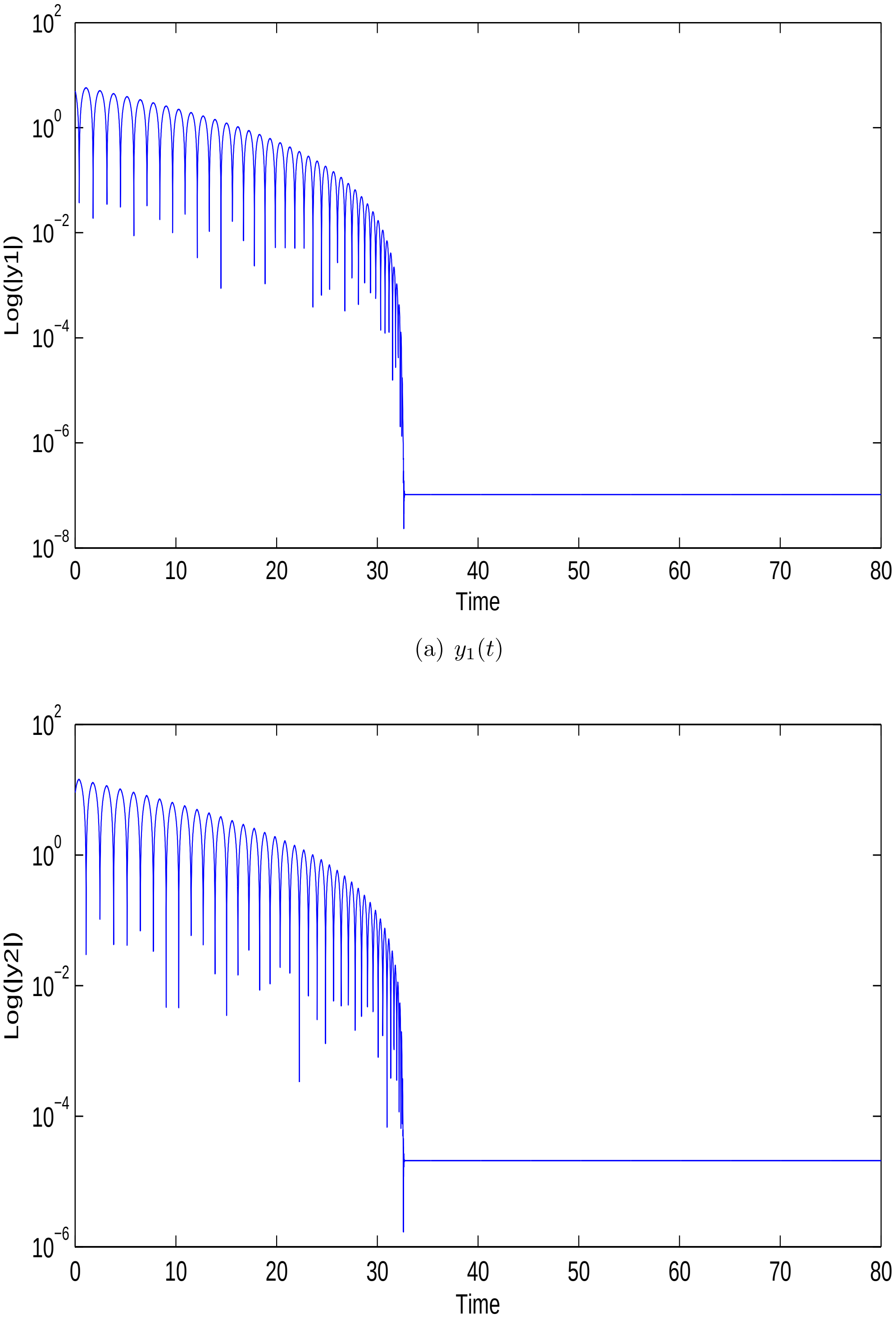
In what follows, we provide a nonlinear finite-time PSO algorithm. In order to enable the states of the system (4.8) to rapidly converge to the origin and enlarge



(a) $y_1(t)$

(b) $y_2(t)$

Figure 4.1: The convergence curves of the system states in (4.8) ($\omega = 0.8$, $\alpha = 6$, $y_1(0) = 5$, and $y_2(0) = -9$).



(a) $y_1(t)$

(b) $y_2(t)$

Figure 4.2: The convergence curves of the system states in (4.9) ($\omega = 0.8$, $\alpha = 6$, $a = 0.5$, $\beta = 1.9$, $\gamma = 0.5$, $y_1(0) = 5$, and $y_2(0) = -9$).

the magnitude of state oscillation, a nonlinear damping item and a parameter γ are added into the system (4.8) as

$$\begin{aligned} \dot{y}_1(t) &= y_2(t) && \text{(4.9a)} \\ \dot{y}_2(t) &= -\gamma\Bigg((1-\omega)y_2(t)+\alpha y_1(t)\Bigg) - \beta\mathrm{sig}\Bigg((1-\omega)y_2(t)+\alpha y_1(t)\Bigg)^a && \text{(4.9b)} \end{aligned}$$

where $0 < a < 1$, $\beta > 0$, and $0 < \gamma \leq 1$.

It is obvious that if $\beta = 0$ and $\gamma = 1$, then the system described by (4.9) becomes a linear system (4.7) in the stagnation case, which was studied by Fernández Martínez *et al.* [33] (2008). Hence, the system (4.7) in the stagnation case can be regarded as a special case of the system (4.9). Figure 4.2 illustrates the convergence of the system states in (4.9) under the same parameter values ω, α and initial states $(y_1(0),\, y_2(0))$ in the deterministic case.

By letting $y_1(t) = \xi_i(t)$ and $y_2(t) = \dot{\xi}_i(t)$, the system (4.9) can be written as

$$\ddot{\xi}_i(t) + \gamma(1-\omega)\dot{\xi}_i(t) = -\gamma\alpha\xi_i(t) - \beta\mathrm{sig}\Bigg((1-\omega)\dot{\xi}_i(t)+\alpha\xi_i(t)\Bigg)^a \qquad (4.10)$$

Moreover, by setting $\xi_i(t) = x_i(t) - p_i(t)$, the continuous-time FPSO algorithm is presented by

$$\begin{aligned} \ddot{x}_i(t) &= -\gamma(1-\omega)(\dot{x}_i(t)-\dot{p}_i(t)) - \gamma\alpha(x_i(t)-p_i(t)) \\ &\quad -\beta\mathrm{sig}\Bigg((1-\omega)(\dot{x}_i(t)-\dot{p}_i(t))+\alpha(x_i(t)-p_i(t))\Bigg)^a + \ddot{p}_i(t) \qquad (4.11) \end{aligned}$$

Consider the dynamics model (1.7) of the robot. We derive the control algorithm (4.12) as the PSO-based finite-time motion control algorithm.

$$\begin{aligned} u_i(t) &= -\gamma(1-\omega)(v_i(t)-\dot{p}_i(t)) - \gamma\alpha(x_i(t)-p_i(t)) \\ &\quad -\beta\mathrm{sig}\Bigg((1-\omega)(v_i(t)-\dot{p}_i(t))+\alpha(x_i(t)-p_i(t))\Bigg)^a + \ddot{p}_i(t) \qquad (4.12) \end{aligned}$$

If the five parameters $(\omega, \alpha, a, \gamma, \beta)$ fall into the following set (4.13), the continuous-time FPSO algorithm is finite-time stable.

$$\Omega_c = \{(\omega, \alpha, a, \gamma, \beta)|\ \omega < 1, \alpha > 0, 0 < a < 1, 0 < \gamma \leq 1, \beta > 0\} \qquad (4.13)$$

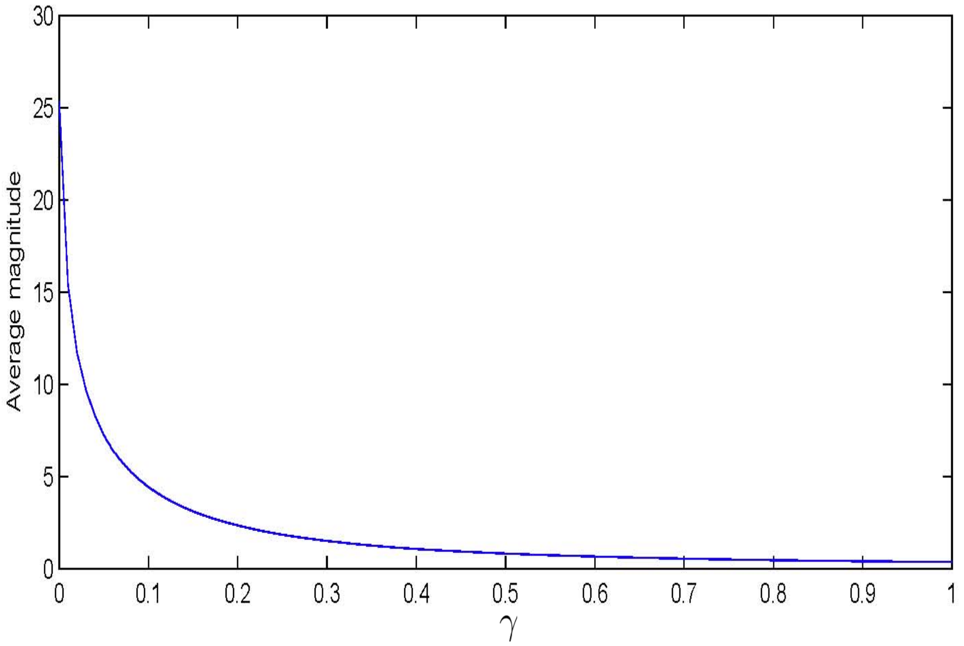


Figure 4.3: The curve of the average oscillation magnitude for the parameter γ for the system state $y_2(t)$ in (4.9) ($\omega = 0.8$, $\alpha = 6$, $a = 0.5$, $\beta = 0.1$, $y_1(0) = 5$, and $y_2(0) = -9$).

It is worth mentioning that the parameters γ and β are used to control the oscillation magnitude and the convergence speed of the state trajectory of the particle, respectively. From Figure 4.3 and Figure 4.4, one can see that increasing the parameter γ and β means the decrease of the average oscillation magnitude and the convergence time, respectively. Hence, the proposed continuous-time FPSO algorithm provides *a flexible mechanism to control "frequency" and "magnitude"* such that we can always adjust two parameters to deal with the problem of odour source localisation in terms of problem's characteristics.

Remark 10. *$\dot{p}_i(t)$ describes the velocity of the variable $p_i(t)$ decided by $\frac{\alpha_1 x_l(t)+\alpha_2 x_g(t)}{\alpha_1+\alpha_2}$. Moreover, the decision process is a discrete-time process, which results in that $p_i(t)$ is piece-wise continuous. Consequently, the direction of $\dot{p}_i(t)$ can be approximately calculated from $p_i(k-1)$ to $p_i(k)$ $(k-1 < t \leq k)$ and the magnitude of that is given based on the maximum linear velocity of the robot. $\ddot{p}_i(t)$ can be approximately*

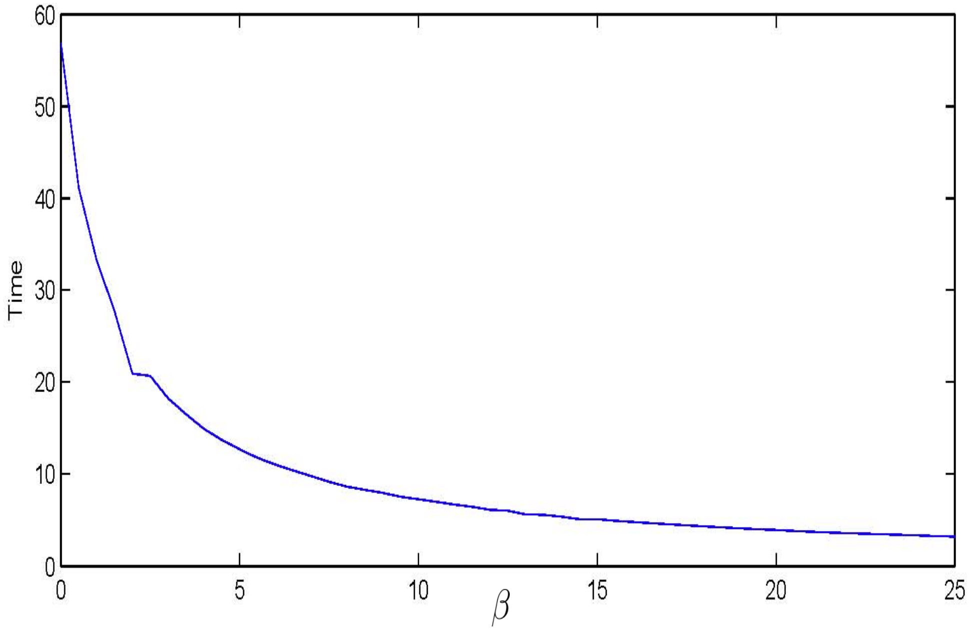


Figure 4.4: The curve of the convergence time for the parameter β for the system state $y_2(t)$ in (4.9) ($\omega = 0.8$, $\alpha = 6$, $a = 0.5$, $\gamma = 1$, $y_1(0) = 5$, and $y_2(0) = -9$).

calculated based on $\dot{p}_i(t)$ and previous velocity information.

4.2.3 Convergence Analysis

In this subsection, we will prove the finite-time convergence of the continuous-time model of the FPSO algorithm. The following Lyapunov analysis provides a global convergence result.

Proposition 6. *Consider the continuous-time model of the FPSO algorithm (4.11) with $(\omega, \alpha, a, \gamma, \beta) \in \Omega_c$ in (4.13) and α is a constant in the deterministic case. The continuous-time FPSO algorithm converges within $\left[0, \frac{(1+a)V(0)^{\frac{1-a}{1+a}}}{k(1-a)}\right]$ where $V(0)$ and k can be calculated based on (4.15) and (4.20), respectively, i.e. $x_i(t) \to p_i(t)$ and $v_i(t) \to \dot{p}_i(t)$ when $t \to \frac{(1+a)V(0)^{\frac{1-a}{1+a}}}{k(1-a)}$.*

Proof. Introduce $\xi_i(t) = x_i(t) - p_i(t)$ into (4.11) and set $y_1(t) = \xi_i(t)$, $y_2(t) = \dot{\xi}_i(t)$. As a result, the system (4.9) can be obtained. Consider the deterministic case, we can write the system (4.11) as

$$\begin{aligned} \dot{y}_1(t) &= y_2(t) \\ \dot{y}_2(t) &= -\gamma\phi_m - \beta \mathrm{sig}(\phi_m)^a \end{aligned} \tag{4.14}$$

for every $a \in (0, 1)$, where $\phi_m = (1 - \omega)y_2(t) + \alpha y_1(t)$.

If the origin (0, 0) of the system (4.14) is a finite-time-stable equilibrium, $x_i(t)$ and $v_i(t)$ will reach $p_i(t)$ and $\dot{p}_i(t)$ in finite-time. Choose a Lyapunov candidate as

$$V(t) = \frac{\beta(1-\omega)}{a+1}|\phi_m|^{a+1} + \frac{\gamma(1-\omega)}{2}\phi_m^2 + \frac{\alpha(1-\omega)}{2}y_2^2 \tag{4.15}$$

Obviously, $V(t) \geq 0$ and along the closed-loop trajectories, we obtain

$$\begin{aligned} \frac{dV(t)}{dt} &= \beta(1-\omega)\mathrm{sig}(\phi_m)^a\dot{\phi}_m + \gamma(1-\omega)\phi_m\dot{\phi}_m + \alpha(1-\omega)y_2\dot{y}_2 \\ &= -(1-\omega)^2(\gamma\phi_m + \beta\mathrm{sig}(\phi_m)^a)^2 \end{aligned} \tag{4.16}$$

Given initial state $y_1(0)$ and $y_2(0)$, if there exists a constant $k > 0$ such that

$$\frac{dV(t)}{dt} \leq -kV(t)^{\frac{2a}{a+1}} \tag{4.17}$$

$V(t)$ will reach zero in finite-time $t^* = \frac{(1+a)V(0)^{\frac{1-a}{1+a}}}{k(1-a)}$ in terms of Lemma 1, which implies that $y_1(t)$ and $y_2(t)$ will be zero.

Suppose $V(t) \neq 0$ (This proof is trivial when $V(t) = 0$) and let $\Upsilon_1 = -\frac{\frac{dV(t)}{dt}}{V(t)^{\frac{2a}{1+a}}}$. Therefore,

$$\begin{aligned} \Upsilon_1 &= \frac{(1-\omega)^2(\gamma\phi_m + \beta\mathrm{sig}(\phi_m)^a)^2}{\left[\rho_1|\phi_m|^{a+1} + \rho_2\phi_m^2 + \rho_3 y_2^2\right]^{\frac{2a}{1+a}}} \\ &\geq \frac{(1-\omega)^2\beta^2|\phi_m|^{2a}}{\rho_1^{\frac{2a}{1+a}}|\phi_m|^{2a} + \rho_2^{\frac{2a}{1+a}}|\phi_m|^{\frac{4a}{1+a}} + \rho_3^{\frac{2a}{1+a}}|y_2|^{\frac{4a}{1+a}}} \\ &\triangleq \Upsilon_2(y_1, y_2) \end{aligned}$$

where $\rho_1 = \frac{\beta(1-\omega)}{a+1}$, $\rho_2 = \frac{\gamma(1-\omega)}{2}$, and $\rho_3 = \frac{\alpha(1-\omega)}{2}$; The inequality follows from the fact that $\phi_m = \mathrm{sig}(\phi_m)$.

It should be pointed out that it is obvious that $\phi_m = 0$ if $y_1(t) = 0$ and $y_2(t) = 0$. Moreover, we have the following claim.

Claim 1: If $\phi_m = 0$, then $y_1(t) = 0$ and $y_2(t) = 0$.

This can be proved as follows. If $\phi_m = 0$, which implies $(1-\omega)y_2(t)+\alpha y_1(t) = 0$, $\begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} = h \begin{pmatrix} -\frac{1-\omega}{\alpha} \\ 1 \end{pmatrix}$ where $h \in \mathbb{R}$. If $h = 0$, then $y_1(t) = 0$ and $y_2(t) = 0$. If $h \neq 0$, both $y_1(t)$ and $y_2(t)$ are constants and are not zero. In terms of (4.14), we can obtain $\phi_m = 0$ and $y_2(t) = 0$, which means $y_1(t) = 0$ that is contradict. Hence, we have $h = 0$, i.e. $y_1(t) = 0$ and $y_2(t) = 0$. The proof of this claim is finished. Because we suppose $V(t) \neq 0$, $\phi_m \neq 0$ by Claim 1.

Due to $\phi_m = (1-\omega)y_2(t) + \alpha y_1(t) = \begin{pmatrix} \alpha & 1-\omega \end{pmatrix} \zeta(t)$ where $\zeta(t) = \begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix}$ presents the system state, we have

$$\phi_m^2 \;=\; \zeta(t)^T \begin{pmatrix} \alpha^2 & \alpha(1-\omega) \\ \alpha(1-\omega) & (1-\omega)^2 \end{pmatrix} \zeta(t) \qquad (4.18)$$

Let $M = \begin{pmatrix} \alpha^2 & \alpha(1-\omega) \\ \alpha(1-\omega) & (1-\omega)^2 \end{pmatrix}$ and the eigenvalues of the matrix M are $\lambda_1 = 0$ and $\lambda_2 = \alpha^2 + (1-\omega)^2$, respectively. Hence, the matrix M is a semi-positive definite one, i.e. $M \geq 0$. For any vector $\mathbf{x} \neq 0$, $\mathbf{x}^T M \mathbf{x} = 0$ implies $\mathbf{x} = h \begin{pmatrix} -\frac{1-\omega}{\alpha} \\ 1 \end{pmatrix}$ where $h \in \mathbb{R}$ and $h \neq 0$. Consider the set $\mathcal{U} = \{\xi \in \mathbb{R}^2 : \xi^T \xi = 1\}$. Then $\mathcal{U}$ is a bounded closed set. It is obvious that $\mathbf{x} \notin \mathcal{U}$. Since the function $\xi^T M \xi$ is continuous with respect to ξ and for any $\xi \in \mathcal{U}$, $\xi^T M \xi \neq 0$, we have that $\mathbf{min}_{\xi \in \mathcal{U}} \xi^T M \xi$, denoted by k_1, exists and is larger than zero.

Consider (4.18) that can be rewritten as

$$\phi_m^2 \;=\; \left(\frac{\zeta}{||\zeta||_2}\right)^T \begin{pmatrix} \alpha^2 & \alpha(1-\omega) \\ \alpha(1-\omega) & (1-\omega)^2 \end{pmatrix} \frac{\zeta}{||\zeta||_2} ||\zeta||_2^2 \qquad (4.19)$$

where $||\cdot||_2$ presents the 2-norm. Since $\frac{\zeta}{||\zeta||_2} \in \mathcal{U}$, $\phi_m^2 \geq k_1 ||\zeta||_2^2 \geq k_1 y_2^2$, which implies $|\phi_m|^{\frac{4a}{1+a}} \geq k_1^{\frac{2a}{1+a}}|y_2|^{\frac{4a}{1+a}}$. Therefore, we have

$$\begin{aligned}
\Upsilon_2(y_1,y_2) &= \frac{(1-\omega)^2\beta^2|\phi_m|^{2a}}{\rho_1^{\frac{2a}{1+a}}|\phi_m|^{2a}+\rho_2^{\frac{2a}{1+a}}|\phi_m|^{\frac{4a}{1+a}}+\rho_3^{\frac{2a}{1+a}}|y_2|^{\frac{4a}{1+a}}} \\
&\geq \frac{(1-\omega)^2\beta^2|\phi_m|^{2a}}{\rho_1^{\frac{2a}{1+a}}|\phi_m|^{2a}+\rho_2^{\frac{2a}{1+a}}|\phi_m|^{\frac{4a}{1+a}}+\rho_3^{\frac{2a}{1+a}}(\frac{1}{k_1})^{\frac{2a}{1+a}}|\phi_m|^{\frac{4a}{1+a}}} \\
&\geq \frac{(1-\omega)^2\beta^2}{\rho_0}\frac{|\phi_m|^{2a}}{|\phi_m|^{2a}+|\phi_m|^{\frac{4a}{1+a}}} \\
&\triangleq \frac{(1-\omega)^2\beta^2}{\rho_0}\Delta(|\phi_m|)
\end{aligned}$$

where $\rho_0 = \max\left\{\rho_1^{\frac{2a}{1+a}}, 2\times\max\left\{\rho_2^{\frac{2a}{1+a}}, \rho_3^{\frac{2a}{1+a}}(\frac{1}{k_1})^{\frac{2a}{1+a}}\right\}\right\}$. If the minimum of $\Delta(|\phi_m|)$ can be obtained and is larger than zero, (4.17) will be satisfied.

It follows from $|\phi_m| = \left|\begin{pmatrix}\alpha & 1-\omega\end{pmatrix}\zeta(t)\right|$ that $|\phi_m| \leq \left\|\begin{pmatrix}\alpha & 1-\omega\end{pmatrix}\right\|_2 ||\zeta(t)||_2$ based on the Hölder inequality, which implies $|\phi_m| \leq \sqrt{\alpha^2+(1-\omega)^2}||\zeta(t)||_2$. In the light of (4.16), one can see that the state of the system asymptotically converges to the origin, which means $||\zeta(t)||_2 \leq ||\zeta(0)||_2$ where $\zeta(0)$ is the initial state and can be given in advance. Furthermore, because we suppose $V(t) \neq 0$, $\phi_m \neq 0$ by Claim 1. Thus, we have $0 < |\phi_m| \leq \sqrt{\alpha^2+(1-\omega)^2}||\zeta(0)||_2$.

In order to obtain the minimum of $\Delta(|\phi_m|)$, which is a nonlinear function for $|\phi_m|$, two cases are considered: one is the set $\chi_1 = \{|\phi_m| : 1 \leq |\phi_m| \leq \sqrt{\alpha^2+(1-\omega)^2}||\zeta(0)||_2\}$ while the other is the set $\chi_2 = \{|\phi_m| : 0 < |\phi_m| < 1\}$. It is worth mentioning that $\chi_1 = \emptyset$ if $\sqrt{\alpha^2+(1-\omega)^2}||\zeta(0)||_2 < 1$. On the one hand, if $\chi_1 \neq \emptyset$, it is a compact set, for any $|\phi_m| \in \chi_1$, $\Delta(|\phi_m|) \neq 0$. Hence, $k_2 = \min_{|\phi_m|\in\chi_1}\Delta(|\phi_m|)$ exists, and is larger than zero. On the other hand, if $\chi_1 = \emptyset$ or $|\phi_m| \notin \chi_1$, we have $|\phi_m| \in \chi_2$, which implies $|\phi_m|^{\frac{2}{1+a}} < 1$. Hence, $|\phi_m|^{\frac{4a}{1+a}} < |\phi_m|^{2a}$. In this case, $\Delta(|\phi_m|) > \frac{1}{2}$.

Let

$$k = \frac{(1-\omega)^2\beta^2}{\rho_0}\times\mathbf{min}\left\{k_2, \frac{1}{2}\right\} \tag{4.20}$$

Since $V(t)$ is radially unbounded and $\dot{V}(t)$ is negative definite, global stability holds. □

Remark 11. *From Proposition 6, one can see that the proposed continuous-time FPSO algorithm possesses a good tracking performance. Especially, when $p_i(t)$ is time-varying, the proposed continuous-time FPSO algorithm can always track $p_i(t)$ within a finite-time interval, which enables this algorithm to efficiently deal with a class of ill-posed and dynamical optimization problems such as odour source localisation.*

Remark 12. *It should also be pointed out that Proposition 6 gives a convergence condition for the deterministic system, i.e., α is a constant. If α is a random number, Proposition 6 describes an expected convergence condition in the mean square.*

4.3 A Discrete-Time FPSO Algorithm

4.3.1 A Discrete-Time Model of the FPSO Algorithm

As a part of the FPSO algorithm, we discretize the system (4.9) to derive the discrete-time model of the FPSO algorithm by employing (4.2). The discrete-time model is described by

$$
\begin{aligned}
y_1(k+\Delta k) &= y_1(k) + y_2(k+\Delta k)\Delta k && \text{(4.21a)}\\
y_2(k+\Delta k) &= y_2(k) - \gamma\Delta k\phi_a(k) - \beta\Delta k \mathrm{sig}(\phi_a(k))^a && \text{(4.21b)}\\
\phi_a(k) &= (1-\omega)y_2(k) + \alpha y_1(k) && \text{(4.21c)}
\end{aligned}
$$

By introducing (4.22) into (4.21), the discrete-time model of the FPSO algorithm can be obtained.

$$
\begin{aligned}
y_1(k+\Delta k) &= x_i(k+\Delta k) - p_i(k+\Delta k) && \text{(4.22a)}\\
y_2(k+\Delta k) &= v_i(k+\Delta k) - \frac{p_i(k+\Delta k) - p_i(k)}{\Delta k} && \text{(4.22b)}
\end{aligned}
$$

Based on the forms of the position and velocity, the discrete-time model of the FPSO algorithm is given by

$$\begin{aligned} v_i(k+\Delta k) &= v_i(k) + \frac{p_i(k+\Delta k) - 2p_i(k) + p_i(k-\Delta k)}{\Delta k} \\ &\quad -\gamma \Delta k \phi_a(k) - \beta \Delta k \mathrm{sig}(\phi_a(k))^a \qquad (4.23a) \\ x_i(k+\Delta k) &= x_i(k) + v_i(k+\Delta k)\Delta k \qquad (4.23b) \end{aligned}$$

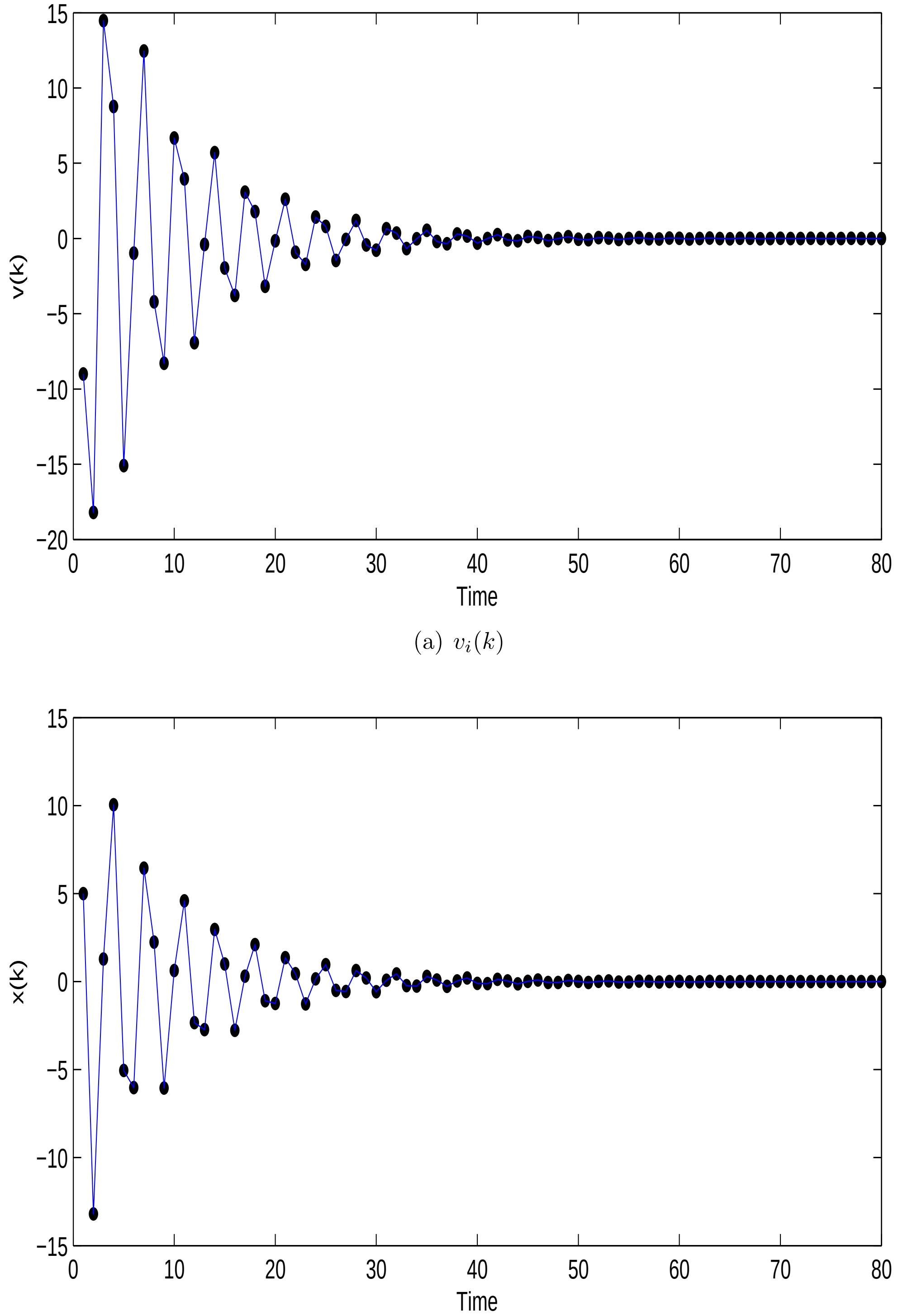
with

$$\phi_a(k) = (1-\omega)v_i(k) + \alpha x_i(k) - \frac{1-\omega+\alpha\Delta k}{\Delta k} p_i(k) + \frac{1-\omega}{\Delta k} p_i(k-\Delta k)$$

Remark 13. *From (4.23), one can see that* $\ddot{p}_i(t) \simeq \frac{p_i(k+\Delta k)-2p_i(k)+p_i(k-\Delta k)}{\Delta k^2}$. *However, it is not easy to calculate* $p_i(k+\Delta k)$. *Hence, we will design a kind of method to calculate* $p_i(k+\Delta k)$ *in the following simulations.*

To illustrate the characteristic of the discrete-time FPSO algorithm, we set $p_i(k) = 0$ and use the PSO algorithm and the GPSO algorithm as comparison examples whose results were also presented in [33]. Figures 4.5-4.7 show the corresponding results. The magnitude of position oscillation denotes the exploration capability of particles while the number of sampling points refers to the exploiting capability of particles. From these figures, one can see that the discrete-time FPSO algorithm provides a flexible mechanism to tradeoff the exploration capability and the exploiting capability of the particle swarm.

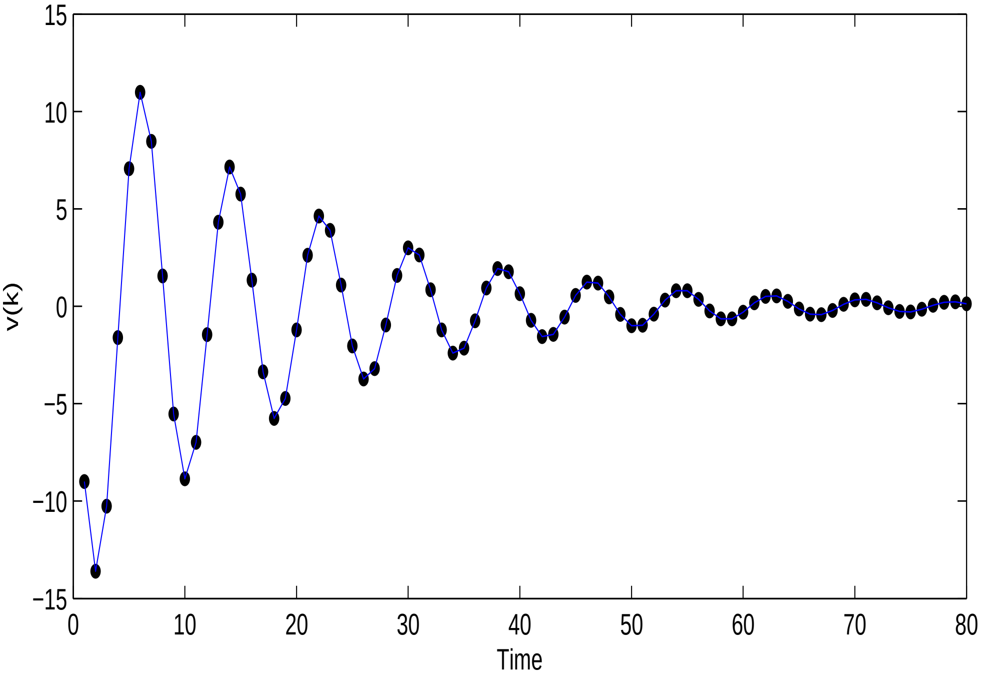
Remark 14. *In Figures 4.5-4.7, the trajectories of all particles are the same due to the fact that* $p_i(k) = 0$ *and other parameters are constant. One reason is that the movement of the particle from the same initial position to the same equilibrium position can be easily shown in the same environment. The other reason is that the characteristics of the discrete-time FPSO algorithm can also be easily illustrated.*



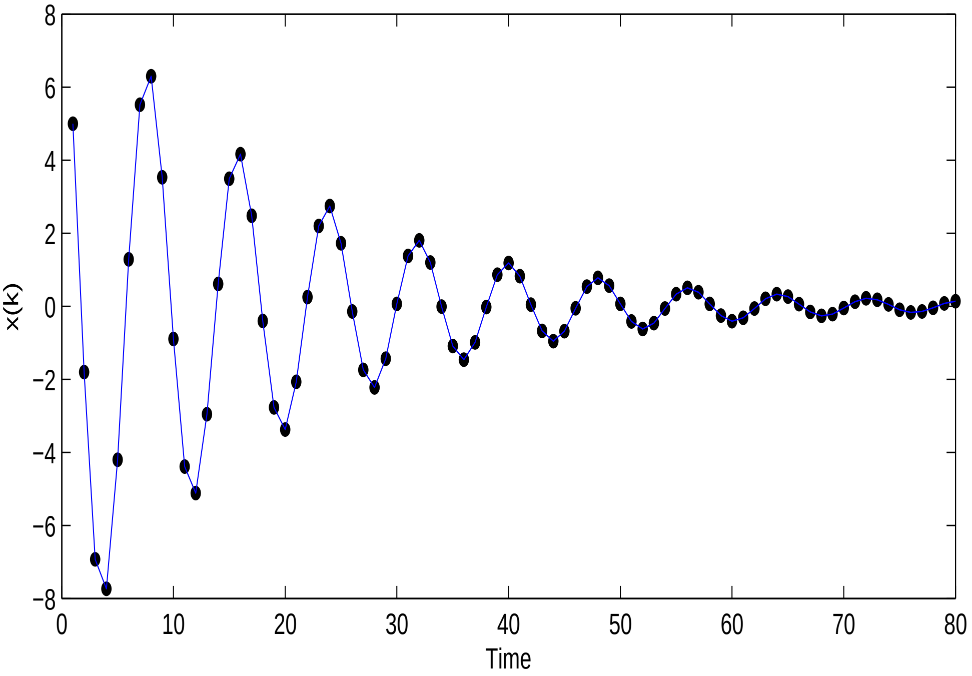
(a) $v_i(k)$

(b) $x_i(k)$

Figure 4.5: The convergence curves of the states of the basic model of the discrete PSO ($\omega = 0.8$, $\alpha = 2.2$, $\beta = 0$, $\gamma = 1$, $\Delta k = 1$, $x_i(0) = 5$, and $v_i(0) = -9$).

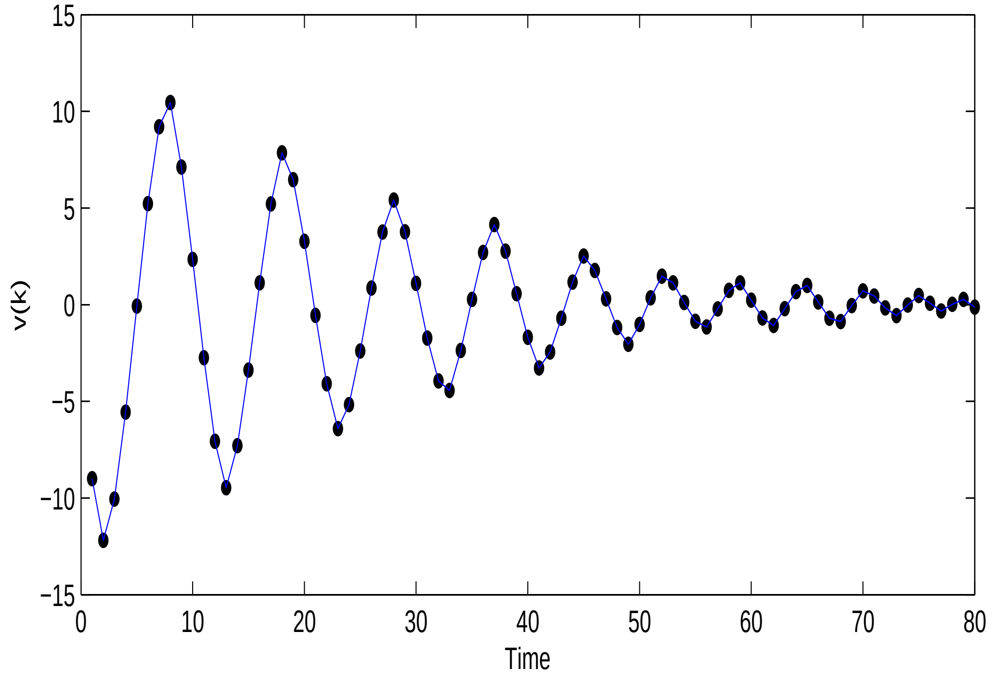


(a) $v_i(k)$

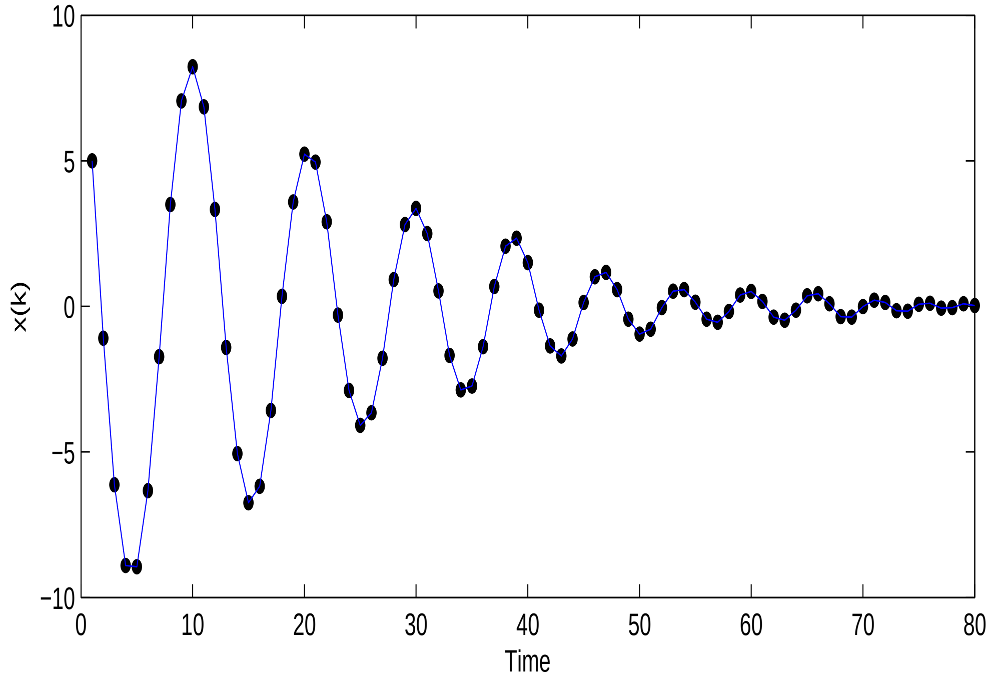


(b) $x_i(k)$

Figure 4.6: The convergence curves of the states of the generalized model of the discrete PSO ($\omega = 0.8$, $\alpha = 2.2$, $\beta = 0$, $\gamma = 1$, $\Delta k = 0.5$, $x_i(0) = 5$, and $v_i(0) = -9$).



(a) $v_i(k)$



(b) $x_i(k)$

Figure 4.7: The convergence curves of the states of the finite-time model of the discrete PSO ($\omega = 0.8$, $\alpha = 2.2$, $\beta = 0.5$, $\gamma = 0.5$, $\Delta k = 0.5$, $x_i(0) = 5$, and $v_i(0) = -9$).

4.3.2 Convergence Analysis

We first give a Schur complement lemma, which will be used in the convergence proof of the discrete-time FPSO algorithm.

Lemma 5. *(Schur complement) Given constant matrices P_1, P_2 and P_3 with appropriate dimensions, where $P_1^T = P_1$ and $P_2^T = P_2 > 0$, then $P_1 + P_3^T P_2^{-1} P_3 < 0$ if and only if*

$$\begin{pmatrix} P_1 & P_3^T \\ P_3 & -P_2 \end{pmatrix} < 0, \text{ or } \begin{pmatrix} -P_2 & P_3 \\ P_3^T & P_1 \end{pmatrix} < 0$$

For the discrete-time model of the FPSO algorithm, a convergence condition is given in Proposition 7.

Proposition 7. *Consider the discrete-time model of the FPSO algorithm (4.23) with $(\omega, \alpha, \gamma, a) \in \Omega_d$ in (4.24) and α is a constant in the deterministic case. If there exist a positive-definite matrix $Q = Q^T > 0$ and a parameter β such that the LMI (4.25) holds, the discrete-time FPSO algorithm converges within a finite-time interval.*

$$\begin{aligned} \Omega_d \;=\; & \left\{ (\omega, \alpha, \gamma) : 1 - \frac{2}{\gamma \Delta k} < \omega < 1, \right. \\ & 0 < \alpha < \frac{4 - 2\gamma\Delta k(1-\omega)}{\gamma \Delta k^2}, \\ & \left. 0 < \gamma \leq 1, 0 < a < 1 \right\} \end{aligned} \tag{4.24}$$

$$\begin{pmatrix} I - Q & AQ & 0 \\ QA^T & -Q & QH^T \\ 0 & HQ & -\frac{1}{\varpi} I \end{pmatrix} \;<\; 0 \tag{4.25}$$

where ϖ is a positive constant, I is a unit matrix, $H = \begin{pmatrix} \alpha & 1-\omega \\ 0 & 0 \end{pmatrix}$, and $A =$

$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$ *with*

$$\begin{aligned} A_{11} &= 1-\gamma\Delta k^2\alpha \\ A_{12} &= \Delta k-\gamma\Delta k^2(1-\omega) \\ A_{21} &= -\gamma\Delta k\alpha \\ A_{22} &= 1-\gamma\Delta k(1-\omega) \end{aligned}$$

Proof. Because the discrete-time model of the FPSO algorithm (4.23) can be derived in terms of (4.21), we write the system (4.21) in the Matrix-Vector form:

$$\begin{aligned} \begin{pmatrix} y_1(k+\Delta k) \\ y_2(k+\Delta k) \end{pmatrix} &= \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}\begin{pmatrix} y_1(k) \\ y_2(k) \end{pmatrix} \\ &\quad + \begin{pmatrix} -\beta\Delta k^2\mathrm{sig}(\phi_a(k))^a \\ -\beta\Delta k\mathrm{sig}(\phi_a(k))^a \end{pmatrix} \end{aligned} \tag{4.26}$$

with

$$\begin{aligned} A_{11} &= 1-\gamma\Delta k^2\alpha \\ A_{12} &= \Delta k-\gamma\Delta k^2(1-\omega) \\ A_{21} &= -\gamma\Delta k\alpha \\ A_{22} &= 1-\gamma\Delta k(1-\omega) \end{aligned}$$

Set $A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$, $\eta(k) = \begin{pmatrix} y_1(k) \\ y_2(k) \end{pmatrix}$, and $f(\eta(k)) = \begin{pmatrix} -\beta\Delta k^2\mathrm{sig}(\phi_a(k))^a \\ -\beta\Delta k\mathrm{sig}(\phi_a(k))^a \end{pmatrix}$. The system (4.26) can be rewritten as

$$\eta(k+\Delta k) = A\eta(k) + f(\eta(k)) \tag{4.27}$$

If the nonlinear item $f(\eta(k)) = 0$, the stability region of the system (4.27) is the part of the space (ω, α), where the roots of the characteristic equation

$$\lambda^2 + (\alpha\gamma\Delta k^2 + \gamma\Delta k(1-\omega) - 2)\lambda + (1-\gamma\Delta k(1-\omega)) = 0$$

are in the unit circle. Based on Routh-Hurwitz criteria [102], this region turns to be

$$\begin{aligned} \Omega_d &= \Bigg\{(\omega,\alpha,\gamma) : 1-\frac{2}{\gamma\Delta k} < \omega < 1, 0 < \alpha < \frac{4-2\gamma\Delta k(1-\omega)}{\gamma\Delta k^2}, \\ &\qquad 0 < \gamma \leq 1,\ 0 < a < 1\Bigg\} \end{aligned} \tag{4.28}$$

If the nonlinear item $f(\eta(k)) \neq 0$, we have

$$f^T(\eta(k))f(\eta(k)) \;=\; (\beta^2\Delta k^4 + \beta^2\Delta k^2)|\phi_a|^{2a}$$

Consider the case where $|\phi_a| > 1$. We obtain

$$\begin{aligned} f^T(\eta(k))f(\eta(k)) \;&\leq\; (\beta^2\Delta k^4 + \beta^2\Delta k^2)|\phi_a|^2 \\ &= \delta\eta(k)^T H^T H\eta(k) \end{aligned} \tag{4.29}$$

where

$$\begin{aligned} \delta &= \beta^2\Delta k^4 + \beta^2\Delta k^2 \\ H &= \begin{pmatrix} \alpha & 1-\omega \\ 0 & 0 \end{pmatrix} \end{aligned}$$

Consider another case where $0 < |\phi_a| \leq 1$. There exits a positive constant $\varepsilon > 0$ such that

$$f^T(\eta(k))f(\eta(k)) \;\leq\; \delta\varepsilon|\phi_a|^2 \tag{4.30}$$

Therefore, $\varpi = \max\{\delta, \delta\varepsilon\}$. Choose a Lyapunov candidate as

$$V(\eta(k)) \;=\; \eta(k)^T P\eta(k) \tag{4.31}$$

where $P = P^T > 0$ and $P \in R^{2\times 2}$.

$$\begin{aligned} \triangle V(\eta(k)) \;=\; & V(\eta(k+\Delta k)) - V(\eta(k)) \\ & = (A\eta(k) + f(\eta(k)))^T P(A\eta(k) + f(\eta(k))) - \eta(k)^T P\eta(k) \\ & = -(A\eta(k) + f(\eta(k)))^T P(A\eta(k) + f(\eta(k))) - \eta(k)^T P\eta(k) \\ & +2(A\eta(k) + f(\eta(k)))^T P(A\eta(k) + f(\eta(k))) \\ & = -(A\eta(k) + f(\eta(k)))^T P(A\eta(k) + f(\eta(k))) - \eta(k)^T P\eta(k) \\ & +(A\eta(k) + f(\eta(k)))^T PA\eta(k) + (A\eta(k) + f(\eta(k)))^T Pf(\eta(k)) \\ & +\eta(k)^T A^T P(A\eta(k) + f(\eta(k))) \\ & +f(\eta(k))^T P(A\eta(k) + f(\eta(k))) \end{aligned} \tag{4.32}$$

Based on the inequality $x^Ty + y^Tx \leq x^Tx + y^Ty$ for any $x, y \in R^n$, we can derive

$$\begin{aligned}\triangle V(\eta(k)) \;\leq\; & -(A\eta(k) + f(\eta(k)))^T P(A\eta(k) + f(\eta(k))) - \eta(k)^T P\eta(k) \\ & +(A\eta(k) + f(\eta(k)))^T PA\eta(k) \\ & +\eta(k)^T A^T P(A\eta(k) + f(\eta(k))) + f(\eta(k))^T f(\eta(k)) \\ & +(A\eta(k) + f(\eta(k)))^T P^2(A\eta(k) + f(\eta(k)))\end{aligned}$$

In terms of (4.30), we have

$$\begin{aligned}\triangle V(\eta(k)) \;\leq\; & (A\eta(k) + f(\eta(k)))^T(P^2 - P)(A\eta(k) + f(\eta(k))) \\ & +\eta(k)^T(\varpi H^T H - P)\eta(k) + (A\eta(k) + f(\eta(k)))^T PA\eta(k) \\ & +\eta(k)^T A^T P(A\eta(k) + f(\eta(k)))\end{aligned}$$

Therefore, we obtain

$$\begin{aligned}\triangle V(\eta(k)) \;\leq\; & \begin{pmatrix} P(A\eta(k) + f(\eta(k))) \\ P\eta(k) \end{pmatrix}^T \times \begin{pmatrix} I - P^{-1} & AP^{-1} \\ P^{-1}A^T & \varpi P^{-1}H^T HP^{-1} - P^{-1} \end{pmatrix} \\ & \times \begin{pmatrix} P(A\eta(k) + f(\eta(k))) \\ P\eta(k) \end{pmatrix}\end{aligned}$$

If

$$\begin{pmatrix} I - P^{-1} & AP^{-1} \\ P^{-1}A^T & \varpi P^{-1}H^T HP^{-1} - P^{-1} \end{pmatrix} \;<\; 0 \tag{4.33}$$

we have $\triangle V(\eta(k)) < 0$.

Set $Q = Q^T = P^{-1}$ and (4.33) can be written as

$$\begin{pmatrix} I - Q & AQ \\ QA^T & \varpi QH^T HQ - Q \end{pmatrix} \;<\; 0 \tag{4.34}$$

Further, we have

$$\begin{pmatrix} I - Q & AQ \\ QA^T & -Q \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & \varpi QH^T HQ \end{pmatrix} \;<\; 0 \tag{4.35}$$

In order to use Lemma 5, we rewrite (4.35) as

$$\begin{pmatrix} I - Q & AQ \\ QA^T & -Q \end{pmatrix} + \begin{pmatrix} 0 \\ QH^T \end{pmatrix} \varpi I \begin{pmatrix} 0 & HQ \end{pmatrix} \;<\; 0 \tag{4.36}$$

Set $P_1 = \begin{pmatrix} I-Q & AQ \\ QA^T & -Q \end{pmatrix}$, $P_2 = \frac{1}{\varpi}I$, and $P_3 = \begin{pmatrix} 0 & HQ \end{pmatrix}$. In terms of Lemma 5, the matrix inequality can be described in the form of the linear matrix inequality given in Proposition 7. Hence, the system (4.26) is asymptotically stable, and is also finite-time stable since it is a discrete-time version of the FPSO algorithm, which means that the discrete-time FPSO algorithm (4.23) converges within a finite-time interval. □

Remark 15. *In fact, if the parameters α, β, γ, a, and ω can satisfy the linear matrix inequality (4.25), the discrete-time FPSO algorithm (4.23) converges within a finite-time interval. However, how to choose the parameters is a hard problem. In order to deal with this problem, we first select the parameters γ and a where γ is used to control exploration ability and a does not influence the search ability of the discrete-time FPSO algorithm. Then, we choose α and ω from Ω_d in (4.24), which can influence the convergence of the discrete-time FPSO algorithm. Finally, by using Matlab LMI toolbox to solve the linear matrix inequality (4.25), we can get the parameter β which is used to control convergence time.*

4.4 Benchmark Functions

In this section, we will illustrate the optimization characteristics and performance of the discrete-time FPSO algorithm through three ill-posed functions and twenty-five benchmark functions.

4.4.1 Three Ill-Posed Functions

In this subsection, we will illustrate the characteristics of the proposed discrete-time FPSO (DFPSO) algorithm based on three ill-posed functions: Griewank, Rastrigin, and Ackley with two dimensions. A similar analysis has been given for the PSO algorithm and the GPSO algorithm in [31, 33]. Moreover, the max function evaluation is limited to 4000 and the population size is 40. The parameters ω and α

are given such that the corresponding algorithms converge. For the PSO algorithm, $0 < \alpha < 4$ and $-1 < \omega < 1$. For the GPSO algorithm, $0 < \alpha < 16$, $-3 < \omega < 1$, and $\Delta t = 0.5$. For the DFPSO algorithm, $0 < \alpha < 32$, $-7 < \omega < 1$, $\Delta t = 0.5$, $\gamma = 0.5$, and $\beta = 0.01$. The success rates and average iterations over 100 simulations (within a tolerance of 10^{-4}) are shown in Figures 4.8-4.16 for Griewank, Rastrigin, and Ackley, respectively. In these figures, the triangles denote the stability region for three algorithms, that is, three algorithms converge when the parameters fall into the triangles.

Remark 16. *It should be pointed out that the decision on $p_i(k)$ are the same for three algorithms. But, three algorithms possess different controllers $u_i(k)$.*

From Figures 4.8-4.16, one can see that the proposed DFPSO algorithm can obtain the higher success rates (A wider stability region with dark red) and lower average iterations (A wider stability region with dark blue) within a wider range of parameters. Hence, from more simulation results, we can conclude that the better optimization results can be obtained only in the stability regions of three algorithms. Moreover, the DFPSO algorithm provides **a flexible mechanism of parameter choice** to obtain the better results for a class of ill-posed optimization problems, which implies that **the controller $u_i(k)$ has an important impact on optimization results**. In addition, one can also see that the higher success rates and the lower average iterations are got for the DFPSO algorithm when the parameters fall into $-2.5 < \omega < 0.2$ and $0 < \alpha < 10$.

4.4.2 Twenty-Five Benchmark Functions

In this subsection, we will test the performance of the DFPSO algorithm based on twenty-five benchmark functions listed in Table 4.1. It is worth mentioning that f_1-f_{10} are the unimodal functions where f_6-f_{10} are from the reference [135] (The readers can refer to [135] and visit the website http://www.ntu.edu.sg/home/EPNSugan/

Table 4.1: The test functions.

Test functions	n	Optimum	Domain	Name
$f_1(\mathbf{x}) = \sum_{i=1}^{n} x_i^2$	30	0	[-100,100]	Sphere
$f_2(\mathbf{x}) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	0	[-10,10]	Schwefel's P2.22
$f_3(\mathbf{x}) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	30	0	[-100,100]	Schwefel's P1.2
$f_4(\mathbf{x}) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	0	[-2,2]	Rosenbrock
$f_5(\mathbf{x}) = \sum_{i=1}^{n} (\lfloor x_i + 0.5 \rfloor)^2$	30	0	[-100,100]	Step
$f_6(\mathbf{x}) = \sum_{i=1}^{n} z_i^2 + f_{bias_1}, \mathbf{z} = \mathbf{x} - \mathbf{o}$	30	-450	[-100,100]	Shifted Sphere
$f_7(\mathbf{x}) = \sum_{i=1}^{n} (\sum_{j=1}^{i} z_j)^2 + f_{bias_2}, \mathbf{z} = \mathbf{x} - \mathbf{o}$	30	-450	[-100,100]	Shifted Schwefel's P1.2
$f_8(\mathbf{x}) = \sum_{i=1}^{n} (10^6)^{\frac{i-1}{n-1}} z_i^2 + f_{bias_3}, \mathbf{z} = \mathbf{x} - \mathbf{o}$	30	-450	[-100,100]	Shifted Rotated High Conditioned Elliptic
$f_9(\mathbf{x}) = (\sum_{i=1}^{n} (\sum_{j=1}^{i} z_j)^2)(1 + 0.4\lvert N(0,1) \rvert) + f_{bias_4}, \mathbf{z} = \mathbf{x} - \mathbf{o}$	30	-450	[-100,100]	Shifted Schwefel's P1.2 with Noise in Fitness
$f_{10}(\mathbf{x}) = \max\{\lvert \mathbf{A}_i\mathbf{x} - \mathbf{B}_i \rvert\} + f_{bias_5}$	30	-310	[-100,100]	Schwefel's P2.6 with Global Optimum on Bounds
$f_{11}(\mathbf{x}) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert})$	30	-12596.5	[-500,500]	Schwefel
$f_{12}(\mathbf{x}) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	0	[-5.12,5.12]	Rastrigin
$f_{13}(\mathbf{x}) = \sum_{i=1}^{n} [y_i^2 - 10\cos(2\pi y_i) + 10]$	30	0	[-5.12,5.12]	Noncontinuous Rastrigin
$f_{14}(\mathbf{x}) = -20\exp(-0.2\sqrt{1/n \sum_{i=1}^{n} x_i^2}) - \exp(1/n \sum_{i=1}^{n} \cos 2\pi x_i) + 20 + e$	30	0	[-32,32]	Ackley
$f_{15}(\mathbf{x}) = 1/4000 \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(x_i/\sqrt{i}) + 1$	30	0	[-600,600]	Griewank
$f_{16}(\mathbf{x}) = \pi/n\{10\sin^2(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4), y_i = 1 + \frac{1}{4}(x_i + 1)$[1]	30	0	[-50,50]	Generalized Penalized
$f_{17}(\mathbf{x}) = \frac{1}{10}\{\sin^2(3\pi x_1) + \sum_{i=1}^{n} (x_i - 1)^2[1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$[1]	30	0	[-50,50]	Generalized Penalized
$f_{18}(\mathbf{x}) = \sum_{i=1}^{n} (\sum_{k=0}^{20} [a^k \cos(2\pi b^k (x_i + 0.5))]) - n \sum_{k=0}^{20} [a^k \cos(\pi b^k)], a = 0.5, b = 3$	30	0	[-0.5,0.5]	Weierstrass
$f_{19}(\mathbf{x}) = \sum_{i=1}^{n-1} (100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2) + f_{bias_6}, \mathbf{z} = \mathbf{x} - \mathbf{o}$	30	390	[-100,100]	Shifted Rosenbrock's Function
$f_{20}(\mathbf{x}) = \sum_{i=1}^{n} \frac{z_i^2}{4000} - \prod_{i=1}^{n} \cos(\frac{z_i}{\sqrt{i}} + 1 + f_{bias_7}), \mathbf{z} = (\mathbf{x} - \mathbf{o})\mathbf{M}$	30	-180	[0,600]	Shifted Rotated Griewank's Function without Bounds
$f_{21}(\mathbf{x}) = -20\exp(-0.2\sqrt{1/n \sum_{i=1}^{n} z_i^2}) - \exp(1/n \sum_{i=1}^{n} \cos 2\pi z_i) + 20 + e + f_{bias_8}, \mathbf{z} = (\mathbf{x} - \mathbf{o})\mathbf{M}$	30	-140	[-32,32]	Shifted Rotated Ackley's Function with Global Optimum on Bounds
$f_{22}(\mathbf{x}) = \sum_{i=1}^{n} [z_i^2 - 10\cos(2\pi z_i) + 10] + f_{bias_9}, \mathbf{z} = \mathbf{x} - \mathbf{o}$	30	-330	[-5,5]	Shifted Rastrigin's Function
$f_{23}(\mathbf{x}) = \sum_{i=1}^{n} [z_i^2 - 10\cos(2\pi z_i) + 10] + f_{bias_{10}}, \mathbf{z} = (\mathbf{x} - \mathbf{o})\mathbf{M}$	30	-330	[-5,5]	Shifted Rotated Rastrigin's Function
$f_{24}(\mathbf{x}) = \sum_{i=1}^{n} (\sum_{k=0}^{20} [a^k \cos(2\pi b^k (x_i + 0.5))]) - n \sum_{k=0}^{20} [a^k \cos(\pi b^k)] + f_{bias_{11}}, a = 0.5, b = 3, \mathbf{z} = (\mathbf{x} - \mathbf{o})\mathbf{M}$	30	90	[-0.5,0.5]	Shifted Rotated Weierstrass Function
$f_{25}(\mathbf{x}) = \sum_{i=1}^{n} (\mathbf{A}_i - \mathbf{B}_i(\mathbf{x}))^2 + f_{bias_{12}}, A_i = \sum_{i=1}^{n} (a_{ij} \sin\alpha_j + b_{ij} \cos\alpha_j)$	30	-460	$[-\pi, \pi]$	Schwefel's P2.13

[1] $u_i(x_j, a, k, m) = \begin{cases} k(x_j - a)^m, & x_j > a \\ 0, & -a \le x_j \le a \\ k(-x_j - 1)^m, & x_j < -a \end{cases}$

to obtain the benchmark functions and the corresponding Matlab codes.). For the multimodal functions, $f_{11} - f_{18}$ are basic multimodal functions while $f_{19} - f_{25}$ are complex multimodal functions. In the simulations, the number of dimensions of all the functions is set as 30 and a maximum function evaluation number is 1×10^5. Each test is repeated 50 times independently. All simulations are carried out on a laptop with two Cores(TM) Duo CPU T6670 running at 2.20GHz with 3GB of RAM. The operation system is Windows Vista and the compiler is Matlab.

In order to illustrate the effectiveness of the proposed DFPSO algorithm, seven state-of-the-art PSO variants, which include the global version PSO-1 algorithm with a fixed inertia weight $\omega = 0.4$ [131], the global version PSO-2 algorithm where ω linearly changes from 0.9 to 0.4 [132], the ALC-PSO algorithm [13], the local version SPSO2007 [110], the generalized PSO algorithm [32], the hierarchical version CLPSO algorithm [77], and the SPSO2011 algorithm [110], are used as comparison algorithms. The reasons of selecting the aforementioned PSO variants are stated in the following. On the one hand, the six algorithm consisting of the PSO-1 algorithm, the PSO-2 algorithm, the ALC-PSO algorithm, the SPSO2007 algorithm, the SPSO2011 algorithm, and CLPSO algorithm are representative and well-performed PSO algorithms that cover the two categories of the improved PSO algorithms. For instance, the PSO-1 algorithm and the PSO-2 algorithm are a class of the well-known PSO variants where the algorithm's parameters are adjusted to obtain the new $p_i(k)$. The ALC-PSO algorithm, the SPSO2007 algorithm, the SPSO2011 algorithm, and the CLPSO algorithm algorithm are another class of well-known PSO variants where the communication topologies are defined to get the new $p_i(k)$. On the other hand, the GPSO algorithm and the DFPSO algorithm provide the new $u_i(k)$ and a flexible mechanism, which $p_i(k)$ proposed by well-known PSO variants can be used to obtain the better search performance under the given $u_i(k)$. Moreover, the parameters of the comparison algorithms can be found in the corresponding references. For the DFPSO algorithm, we set $\omega = -0.7$ if the test functions are the unimodal functions or $\omega = -2.1$ if the test functions are the multimodal functions. Other parameters are $\Delta t = 0.5$, $\gamma = 0.5$, $\alpha = 8.5$, and $\beta = 0.01$. The population size is set to 20 for all the algorithms.

Remark 17. *It is worth noting that the proposed DFPSO algorithm only provides a kind of the motion control mechanism of the particle swarm, that is, the DFPSO algorithm gives a cooperative controller $u_i(k)$ with a finite-time convergence property. In this dissertation, we mainly illustrate the significance of the motion control mechanism. In other words, we do not design the decision algorithm $p_i(k)$. Instead, we plan the motion trajectory of the particle swarm. Moreover, the proposed FPSO algorithm is motivated by the problem of odour source localisation and can effectively locate the odour source in a dynamical environment. Hence, the corresponding decision algorithm, i.e. $p_i(k)$, is also developed according to the characteristics of the odour source localisation problem. But, for the discrete-time optimization problems, we do not give the corresponding decision algorithm. Hence, in numerical simulations, we use (2.4) to calculate $p_i(k)$ and the method proposed by the SPSO2011 algorithm [110] to compute $p_i(k+1)$.*

Results on Unimodal Functions

In Table 4.2, the mean function errors, the minimal function errors, and standard errors given by the different PSO variants after $1E+05$ are reported. One can see from this table that the results yielded by the DFPSO algorithm are better than those obtained by the other algorithms for a majority of test functions. Moreover, the statistical results are also reported by using a two-sample Wilcoxon rank sum test. The Wilcoxon rank sum test as a nonparametric statistical test can be used to determine the statistical significance of the difference between two independent samples. For the most test functions, the difference between the results yielded by the DFPSO algorithm and the results obtained by the other algorithms are statistically significant. Finally, the number of functions where the DFPSO algorithm yields significantly better or worse results is also shown in Table 4.2.

Results on Basic Multimodal Functions and Complex Multimodal Functions

For the multimodal functions, the global optimum is difficult to be located, especially for the complex multimodal functions. In Table 4.3, the results for the basic multimodal functions are reported. Form this table, the proposed DFPSO algorithm can produce the very competitive results for the most test functions. In order to further test the effectiveness of the proposed algorithm, the results for the complex multimodal functions are also given in Table 4.4. Similarly, the statistical test results by using the Wilcoxon rank sum test for the basic multimodal functions and the complex multimodal functions are also listed in Table 4.3 and Table 4.4, respectively. In addition, it is not real that we require that the DFPSO algorithm can obtain the better optimization results for all the test functions. From Table 4.4, one can see that the number of functions where the ALC-PSO algorithm can obtain the better results is more than the one of functions where the DFPSO algorithm can get the better results. One reason is that the DFPSO algorithm is designed for the problem of odour source localisation, which means that we develop the proposed algorithm according to the characteristics of the odour source localisation problem. If the characteristics of test functions are different from the odour source localisation problem, then the DFPSO algorithm maybe obtain the worse results.

4.5 Odour Source Localisation

In this section, we will deal with the problem of odour source localisation by using the PSO-based finite-time motion control algorithm.

Table 4.2: The function error values on the unimodal functions based on 50 runs.

Func.	Index	DFPSO	PSO-1	PSO-2	ALC-PSO	SPSO2007	GPSO	SPSO2011	CLPSO
f_1	mean[1]	4.14E-300	8.83E-17	1.44E-21	7.36E-70	7.15E-77	8.45E-191	9.09E-99	3.80E-12
	best[2]	2.54E-314	1.45E-78	8.29E-26	1.07E-89	3.56E-83	6.53E-205	2.64E-104	4.76E-13
	std[3]	0	4.36E-16	8.83E-21	5.15E-69	2.56e-76	0	3.50E-98	2.61E-12
	p_{value}[4]	–	9.43E-14* [5]	9.43E-14*	9.43E-14*	9.43E-14*	9.43E-14*	9.43E-14*	9.43E-14*
f_2	mean	7.92E-148	6.46E-32	9.42E-08	1.50E-31	4.33E-37	3.98E-67	3.01	3.20E-08
	best	6.92E-153	1.64E-37	4.97E-16	4.49E-39	3.80E-16	3.28E-87	5.69E-01	1.14E-08
	std	3.05E-147	2.02E-31	4.38E-07	8.65E-31	1.69E-36	1.78E-66	1.67	1.89E-08
	p_{value}	–	9.43E-14*	9.32E-14*	9.43E-14*	9.43E-14*	9.43E-14*	9.43E-14*	9.43E-14*
f_3	mean	5.72E-136	5.10E-03	1.81E+04	2.90E-03	6.50E-05	7.92E-132	3.71E-10	9.58E+03
	best	5.27E-152	5.82E-05	180.64	1.14E-04	7.67e-07	1.87E-178	1.78E-11	4.77E+03
	std	2.74E-135	1.16E-02	8.29E+03	4.50E-03	9.03E-05	3.65E-131	5.20E-10	2.01E+03
	p_{value}	–	9.43E-14*	9.43E-14*	9.43E-14*	9.43E-14*	0.21	9.43E-14*	9.43E-14*
f_4	mean	25.39	25.37	33.82	24.64	17.45	28.90	23.02	25.01
	best	25.18	8.58	8.09E-01	8.95	14.30	28.80	20.62	22.05
	std	1.21E-01	14.67	26.51	12.91	2.26	3.76E-02	1.81	9.32E-01
	p_{value}	–	7.92E-06*	1.73E-02*	3.57E-08*	9.43E-14*	9.43E-14*	8.96E-09*	0.06
f_5	mean	0	18.38	1.50	0	15.56	0	5.56	0
	best	0	1	0	0	0	0	0	0
	std	0	35.94	1.77	0	58.98	0	3.32	0
	p_{value}	–	1.88E-14*	3.14E-09*	0.27	3.05E-11*	0.27	4.87E-14*	0.27
f_6	mean	6.82E-14	2.50E-03	337.54	3.30E-03	7.95E-14	1.17E-12	7.38E-14	7.85E-13
	best	5.68E-14	1.13E-13	1.13E-13	1.13E-13	5.68E-14	2.27E-13	0	2.84E-13
	std	2.31E-14	1.65E-02	867.04	2.35E-02	5.74E-14	1.96E-12	2.87E-14	3.46E-13
	p_{value}	–	1.21E-13*	9.54E-14*	5.20E-14*	0.59	4.11E-14*	0.32	4.07E-14*
f_7	mean	1.36E-05	6.90E-03	1.41E+04	2.20E-03	1.08E-04	37.39	3.03E-09	4.87E+03
	best	5.46E-06	1.42E-04	339.06	2.00E-04	3.25E-06	1.57E-01	6.01E-11	3.12E+03
	std	3.80E-06	1.78E-02	1.14E+04	2.3E-03	1.27E-04	92.48	4.62E-09	1.24E+03
	p_{value}	–	9.43E-14*	9.43E-14*	3.01E-11*	1.72E-06*	9.43E-14*	9.43E-14*	3.01E-11*
f_8	mean	7.29E+05	4.51E+06	1.78E+07	4.15E+06	1.01E+06	9.77E+05	6.14E+05	4.16E+07
	best	3.54E+05	1.07E+06	3.38E+06	1.25E+06	4.42E+05	2.97E+05	1.53E+05	2.44E+07
	std	3.11E+05	2.55E+06	1.10E+07	2.52E+06	3.34E+05	4.23E+05	2.57E+05	1.06E+07
	p_{value}	–	2.08E-13*	9.17E-14*	7.01E-11*	7.58E-04*	3.24E-02*	0.06	2.82E-11*
f_9	mean	1.98E+03	6.19E+03	3.59E+04	4.76E+03	6.33E+02	3.17E+04	3.26E+03	1.22E+04
	best	6.62E+02	6.27E+02	1.18E+04	3.60E+02	8.39E+01	2.20E+04	8.55E+02	5.46E+03
	std	1.36E+03	5.47E+03	1.37E+04	3.67E+03	6.90E+02	6.16E+03	1.51E+03	3.80E+03
	p_{value}	–	4.95E-06*	9.43E-14*	1.17E-04*	1.49E-06*	3.01E-11*	6.88E-05*	3.11E-11*
f_{10}	mean	5.33E+03	6.50E+03	7.99E+03	6.16E+03	4.11E+03	7.83E+03	6.64E+03	3.69E+03
	best	4.07E+03	3.97E+03	3.42E+03	3.28E+03	2.06E+03	4.56E+03	4.38E+03	2.56E+03
	std	8.91E+02	1.58E+03	2.40E+03	1.50E+03	9.75E+02	1.63E+03	1.33E+03	4.43E+02
	p_{value}	–	1.80E-02*	1.58E-08*	6.14E-04*	1.07E-05*	3.86E-08*	2.44E-05*	1.05E-10*
sum.	better[6]	–	9	10	8	6	8	6	7
	worse[7]	–	1	0	1	3	0	2	1

[1] mean: The value denotes the mean function error based on 50 runs.

[2] best: The value is the minimal function error among 50 runs.

[3] std: The value denotes the standard error based on 50 runs.

[4] p_{value}: The value is a p value of a two-sample Wilcoxon rank sum test for the function error between the DFPSO algorithm and other comparison algorithms in the corresponding column. The p value being smaller means that the results obtained by the DFPSO algorithm are significant compared with ones obtained by other algorithms.

[5] $*$: The difference between two samples is significant at level $\alpha = 0.05$.

[6] better : The number of functions where the DFPSO algorithm obtains significantly better results.

[7] worse : The number of functions where the DFPSO algorithm obtains significantly worse results.

Table 4.3: The function error values on the basic multimodal functions based on 50 runs.

Func.	Index	DFPSO	PSO-1	PSO-2	ALC-PSO	SPSO2007	GPSO	SPSO2011	CLPSO
f_{11}	mean[1]	5.66E+03	3.44E+03	1.51E+03	1.32	4.37E+03	5.06E+03	5.02E+03	3.78E+01
	best[2]	4.01E+03	2.05E+03	1.07E+03	2.33E-02	2.19E+03	2.95E+03	2.68E+03	3.11E+01
	std[3]	8.54E+02	6.59E+02	2.70E+02	1.70	7.95E+02	8.70E+02	1.00E+03	3.61
	p_{value}[4]	–	1.32E-10*[5]	2.66E-11*	3.01E-11*	1.60E-06*	1.69E-02*	1.76E-02*	3.01E-11*
f_{12}	mean	9.63E-11	7.28E+01	8.05E+01	1.86E+01	4.99E+01	3.54E+01	3.25E+01	1.85E+01
	best	1.70E-13	3.58E+01	2.68E+01	8.95	2.78E+01	1.20E+01	1.69E+01	1.01E+01
	std	1.24E-10	2.11E+01	3.23E+01	5.66	1.29E+01	1.91E+01	9.12	6.70
	p_{value}	–	2.91E-11*	2.91E-11*	2.88E-11*	2.90E-11*	2.91E-11*	2.89E-11*	2.91E-11*
f_{13}	mean	2.89E-10	6.04E+01	1.16E+02	4.2E-01	4.78E+01	3.94E+01	5.85E+01	1.96E+01
	best	6.82E-13	7	4.20E+01	6.20E-10	2.9E+01	1.00E+01	2.4E+01	1.04E+01
	std	4.11E-10	3.00E+01	4.41E+01	2.6E-01	1.40E+01	2.71E+01	2.50E+01	5.48
	p_{value}	–	2.87E-11*	2.88E-11*	3.42E-11*	2.85E-11*	2.89E-11*	2.87E-11*	2.89E-11*
f_{14}	mean	6.83E-07	2.48	1.99E+01	1.95	1.73	5.32E-01	2.26	6.11E-02
	best	1.11E-09	5.77E-14	1.97E+01	9.31E-01	7.99E-15	1.06E-01	9.31E-01	7.57E-05
	std	7.47E-07	1.60	3.78E-02	6.81E-01	6.64E-01	3.32E-01	5.75E-01	1.87E-01
	p_{value}	–	8.45E-09*	1.93E-11*	3.01E-11*	5.43E-10*	3.01E-11*	2.99E-11*	3.01E-11*
f_{15}	mean	6.28E-14	4.91E-02	9.8E-03	4.19E-02	4.48E-02	1.50E-02	9.8E-03	3.96E-04
	best	0	0	0	2.22E-16	0	1.12E-31	1.11E-16	2.51E-08
	std	1.07E-13	7.93E-02	1.21E-02	6.42E-02	1.48E-01	2.67E-02	9.3E-03	6.15E-04
	p_{value}	–	1.58E-04*	0.27	3.08E-04*	2.5E-03*	3.79E-10*	4.7E-03*	2.99E-11*
f_{16}	mean	3.17E-25	9.50E-01	1.76E-01	6.23E-02	2.56E-01	2.21E-01	1.38	1.24E-09
	best	7.53E-26	2.08E-32	3.84E-21	1.82E-32	1.57E-32	2.14E-06	1.57E-32	2.45E-11
	std	2.16E-25	1.59	2.85E-01	1.14E-01	3.72E-01	2.60E-01	2.16	1.50E-09
	p_{value}	–	7.9E-03*	2.84E-11*	0.18	0.37	2.91E-11*	9.23E-06*	2.91E-11*
f_{17}	mean	1.16E-01	1.68E-01	1.05E-02	1.03E-02	1.37E-01	1.7E-03	9.4E-03	3.09E-10
	best	1.34E-11	2.58E-32	1.41E-22	2.95E-32	1.34E-32	1.08E-08	1.34E-32	3.74E-11
	std	4.81E-01	5.10E-01	1.36E-02	1.51E-02	4.35E-01	7.8E-03	2.25E-02	4.99E-10
	p_{value}	–	1.5E-03*	1.01E-04*	8.30E-05*	1.45E-04*	4.49E-07*	1.19E-06*	9.91E-08*
f_{18}	mean	7.1E-03	6.01	6.35E-01	2.15	2.36	8.69	9.95	3.00E-04
	best	1.0E-03	1.85	3.03E-04	1.39E-01	4.95E-01	4.09	5.56	1.47E-04
	std	5.3E-03	2.04	6.6E-01	1.26	1.39	2.49	1.65	1.11E-04
	p_{value}	–	3.01E-11*	7.68E-08*	3.01E-11*	3.01E-11*	3.01E-11*	3.01E-11*	3.01E-11*
sum.	better[6]	–	7	5	5	6	6	6	5
	worse[7]	–	1	2	2	1	2	2	3

[1] mean: The value denotes the mean function error based on 50 runs.

[2] best: The value is the minimal function error among 50 runs.

[3] std: The value denotes the standard error based on 50 runs.

[4] p_{value}: The value is a p value of a two-sample Wilcoxon rank sum test for the function error between the DFPSO algorithm and other comparison algorithms in the corresponding column. The p value being smaller means that the results obtained by the DFPSO algorithm are significant compared with ones obtained by other algorithms.

[5] $*$: The difference between two samples is significant at level $\alpha = 0.05$.

[6] better : The number of functions where the DFPSO algorithm obtains significantly better results.

[7] worse : The number of functions where the DFPSO algorithm obtains significantly worse results.

Table 4.4: The function error values on the complex multimodal functions based on 50 runs.

Func.	Index	DFPSO	PSO-1	PSO-2	ALC-PSO	SPSO2007	GPSO	SPSO2011	CLPSO
f_{19}	mean[1]	1.95E+02	5.29E+01	5.26E+07	7.35E+01	4.90E+01	1.71E+02	1.04E+03	1.23E+02
	best[2]	2.61E+01	1.38E-01	1.17E-01	3.46E-02	2.30E-02	2.54E-01	1.85E+01	4.23E+01
	std[3]	2.47E+02	7.02E+01	1.31E+08	9.64E+01	6.95E+01	2.09E+02	1.96E+03	6.23E+01
	p_{value}[4]	–	3.66E-06*[5]	0.55	2.20E-04*	1.82E-07*	0.65	4.95E-02*	0.51
f_{20}	mean	4.69E+03	4.71E+03	4.69E+03	2.89E+03	4.69E+03	4.69E+03	4.89E+03	4.69E+03
	best	4.69E+03	4.69E+03	4.69E+03	2.17E+03	4.69E+03	4.69E+03	4.77E+03	4.69E+03
	std	3.67E-12	3.39E+01	3.67E-12	3.47E+02	3.67E-12	3.67E-12	9.33E+01	3.67E-12
	p_{value}	–	4.67E-19*	Nan	3.37E-20*	Nan	Nan	3.31E-20*	Nan
f_{21}	mean	20.9264	20.9449	20.9383	20.9496	20.9587	20.9489	20.9259	20.9889
	best	20.7679	20.7679	20.7465	20.7207	20.7672	20.7682	20.7311	20.8461
	std	6.01E-02	6.79E-02	6.38E-02	6.7E-02	6.74E-02	6.67E-02	7.25E-02	5.33E-02
	p_{value}	–	1	0.3431	3.7E-02*	5.9E-02*	7.81E-02*	0.8442	8.06E-07*
f_{22}	mean	32.40	81.96	117.46	20.30	51.19	103.79	73.27	72.89
	best	11.45	43.75	63.28	5.96	22.88	42.78	42.78	61.28
	std	2.13E+01	2.05E+01	2.06E+01	7.33	1.64E+01	2.60E+01	1.80E+01	6.27
	p_{value}	–	1.6E-10*	2.31E-13*	4.8E-03*	8.49E-06*	2.71E-12*	1.26E-09*	5.09E-10*
f_{23}	mean	130.47	146.28	204.53	152.05	78.84	170.57	83.61	190.55
	best	100.98	77.60	106.15	77.60	31.83	103.47	41.80	157.32
	std	1.92E+01	4.77E+01	4.09E+01	4.43E+01	3.74E+01	3.56E+01	2.43E+01	1.52E+01
	p_{value}	–	0.24	5.59E-11*	2.7E-02*	1.02E-07*	2.49E-07*	3.94E-10*	1.33E-12*
f_{24}	mean	26.02	29.22	32.11	30.71	35.26	26.48	30.24	33.49
	best	20.28	22.46	25.96	20.97	28.88	18.62	18.80	29.19
	std	2.97	2.84	2.65	3.44	2.33	2.61	5.15	1.54
	p_{value}	–	7.80E-05*	1.44E-11*	3.16E-07*	1.47E-13*	0.86	2.92E-05*	1.27E-13*
f_{25}	mean	9.39E+05	4.64E+04	1.92E+05	4.63E+04	8.05E+05	8.22E+05	1.09E+06	6.14E+05
	best	6.78E+05	3.74E+03	7.95E+04	3.24E+03	3.09E+04	1.58E+05	6.81E+05	4.55E+05
	std	1.23E+05	3.68E+04	8.77E+04	3.23E+04	1.82E+05	2.42E+05	1.94E+05	7.36E+04
	p_{value}	–	9.43E-14*	9.43E-14*	9.43E-14*	4.95E-04*	0.06	1.08E-04*	2.88E-13*
sum.	better[6]	–	3	3	3	3	3	5	4
	worse[7]	–	2	1	4	3	0	1	1

[1] mean: The value denotes the mean function error based on 50 runs.

[2] best: The value is the minimal function error among 50 runs.

[3] std: The value denotes the standard error based on 50 runs.

[4] p_{value}: The value is a p value of a two-sample Wilcoxon rank sum test for the function error between the DFPSO algorithm and other comparison algorithms in the corresponding column. The p value being smaller means that the results obtained by the DFPSO algorithm are significant compared with ones obtained by other algorithms.

[5] $*$: The difference between two samples is significant at level $\alpha = 0.05$.

[6] better : The number of functions where the DFPSO algorithm obtains significantly better results.

[7] worse : The number of functions where the DFPSO algorithm obtains significantly worse results.

Table 4.5: The parameters used in (4.12) for the motion control.

β	γ	α	a	ω	v_{max}(m/s)	ω_{max}(rad/s)
0.01	0.5	8	0.5	-0.5	0.8	1.57

4.5.1 Motion Process of a Group of Robots

In the following, we will test the search efficiency of a multi-robot system coordinated by the proposed PSO-based finite-time motion control algorithm (Continuous-time FPSO algorithm, CFPSO). The parameters β, γ, α, a, ω, which are given in Table 4.5 in terms of the numerical results for the DFPSO algorithm, are used to guarantee the convergence of the motion control while the parameters v_{max} and ω_{max} are used to limit the maximum linear velocity and angular velocity of robots, respectively. It is worth mentioning that the robot group will search for the odour clues along the direction of y axis from the initial positions (right-up corner) to the target positions (right-down corner) in the initial stage. Once the odour clues are detected by any robot, the proposed CFPSO algorithm will start to run. Moreover, we use a circle where the real position of the odour source is regarded as a center with a predefined radius $1m$ as one of termination conditions, which means that the search task is terminated if any robot enters the circle. The maximal search time $1500s$ is used as another termination condition.

As an example, the motion process of the robot group controlled by the proposed CFPSO algorithm is illustrated in Figures 4.17-4.18. In Figure 4.17(a), the initial positions of the robot group are set at the right-up corner in the search region. In Figure 4.17(b) and Figure 4.18(a), the robot group controlled by $u_i(t)$ traces the plume and moves along the plume in the light of the probable positions of the odour source $h_i(k)$. In Figure 4.18(b), the robot group finds the real odour source. From 0s to about 40s, the robots keep the predefined position (80m,0m) of the odour source. After about 40s, the robot group detects the odour clues, and then the proposed

CFPSO algorithm starts to run. Correspondingly, the prediction errors and average prediction errors based on 50 runs of five robots for the position of the odour source are shown in Figure 4.19 where the real position of the odour source is located at (10m,0m). From Figure 4.19, one can see that the average prediction error of $h^{ic}(k)$, $i = 1, 2, 3, 4, 5$ in (3.31) for the position of the odour source approaches to 2m. Among 5 robots, the best prediction prediction error is lesser than 1m with a higher proportion for 50 runs. The aforementioned simulation results mainly illustrate that the prediction error for $h^{ic}(k)$ calculated in (3.31) under a given finite-time cooperative controller (4.12) can provide stable prediction results for the probable position of the odour source, which reflects the importance of finite-time convergence for motion control.

4.5.2 Comparison Results

In this subsection, we will compare the search efficiency of the multi-robot system coordinated by the proposed CFPSO algorithm with several selected algorithms, which include the PSO algorithm [93], the PPSO-IM algorithm [85], the LPSO algorithm [87], the CPSO algorithm [53], the WUI-45 algorithm [53], and the WUII algorithm [53]. It is worth mentioning that the PPSO-IM, PSO, and CPSO algorithms only utilize concentration magnitude information while the WUI-45, WUII, and LPSO algorithms make use of not only concentration magnitude information but also wind information. The parameters of the seven algorithms can be found in [85], [53], [93], and [87], respectively. Furthermore, we use three evaluation indexes to estimate the search performance of multi-robot systems coordinated by these algorithms. These indexes are success rate, search time, and consumed energy. Finally, we will test all the algorithms in the nine scenarios listed in Table 4.6, but use the new termination conditions, that is, the maximum search time is set as 1000s for shortening the experimental time and the radius of the circle is still set as 1m.

Table 4.6: The nine scenarios.

cases	The position of odour source (x, y)	The initial wind speed (v_x, v_y)
case 1	(10, 0)	(1,0)
case 2	(30, 10)	(1,0)
case 3	(30, -20)	(1,0)
case 4	(10, 0)	(1.5,0)
case 5	(30, 10)	(1.5,0)
case 6	(30, -20)	(1.5,0)
case 7	(10, 0)	(0.8,0)
case 8	(30, 10)	(0.8,0)
case 9	(30, -20)	(0.8,0)

Remark 18. *The seven algorithms are chosen because they have been used to deal with the problem of odour source localisation.* ***Hence, it is unfair to choose other kinds of PSO algorithms that are not designed for the problem of odour source localisation.*** *It should be indicated that the seven algorithms, which include the PSO algorithm[93], the CPSO algorithm [53], the WUI-45 algorithm [53], the WUII algorithm [53], the PPSO-IM algorithm [85], and the LPSO algorithm [87], use the same initial search process as the proposed CFPSO algorithm and run at the same simulation environment. Moreover, the simulation results not only reflect the importance on the position prediction of the odour source, but also illustrate the effectiveness of the finite-time motion control in the dynamical environment.*

Remark 19. *It should also be pointed out that the initial wind speed and the position of the odour source are two key parameters in Table 1.1 for the dynamical environment because a combination of the wind speed and the position of the odour source reflects a class of the environmental condition in the real world. Therefore, we design nine scenarios described in Table 4.6 in order to validate the effectiveness of the proposed CFPSO algorithm.*

Remark 20. *In the simulations, we choose 3 robots and 5 robots to locate the odour* *source, respectively. The reasons are stated in the following. From the viewpoint of engineering, if we can use a few robots to successfully deal with this problem, we will do not use more robots due to costs. In addition, In terms of the third characteristics of the odour source localisation problem described in Chapter 1, a few robots can also sample sufficient odour concentrations through the appropriate design of the the robot behavior.*

In Table 4.7 and Table 4.8, compared with these methods like the PSO algorithm, the CPSO algorithm, the WUI-45 algorithm, the WUII algorithm, the PPSO-IM algorithm, and the LPSO algorithm, the CFPSO algorithm generates the better success rates and search time for the different numbers of robots. Correspondingly, the statistical results by using the Wilcoxon rank sum test are also given in Table 4.7 and Table 4.8. Moreover, since path length denotes the energy consumed by robots, the shorter path length means the better search performance. From Figures 4.20-4.29, one can also see that robots coordinated by the CFPSO algorithm consume the lesser energy than those coordinated by the PSO algorithm, the CPSO algorithm, the WUI-45 algorithm, the WUII algorithm, the PPSO-IM algorithm, and the LPSO algorithm.

4.6 Conclusion

A PSO-based finite-time motion control algorithm has been proposed to deal with the problem of odour source localisation. Specifically, we have derived a continuous-time FPSO algorithm by introducing a nonlinear damping item and a parameter into the continuous model of the PSO algorithm such that the continuous-time FPSO algorithm can converge within a finite-time interval and its exploration capability can be improved. Then, we have used a Lyapunov approach to analyze the finite-time convergence of the continuous-time FPSO algorithm. Next, we have given a discrete-time FPSO algorithm by employing the same dicretization scheme as the

Table 4.7: The statistical results of search time for 3 robots based on 50 runs.

Robots	Index	CFPSO	PSO	CPSO	WUI-45	WUII	PSO-IM	LPSO
case 1	suc%[1]	100	4	6	2	34	80	100
	mean$_{time}$[2]	162.51	991.53	990.29	995.90	837.40	602.15	289.79
	std$_{time}$[3]	18.76	43.99	42.04	29.01	265.65	264.16	71.55
	p_{time}[4]	–	6.63E-20* [6]	9.12E-20*	4.73E-20*	1.74E-18*	6.80E-18*	1.95E-17*
	rank [5]	1	6	5	7	4	3	2
case 2	suc%	98	54	32	32	20	86	94
	mean$_{time}$	201.12	860.28	894.94	917.18	894.69	398.28	348.87
	std$_{time}$	158.12	176.30	187.32	149.54	243.62	324.99	216.94
	p$_{time}$	–	1.08E-16*	2.53E-17*	2.11E-17*	3.55E-17*	1.92E-7*	1.05E-9*
	rank	1	4	5	7	6	3	2
case 3	suc%	100	28	14	14	22	84	96
	mean$_{time}$	123.85	933.28	967.13	960.97	871.66	389.52	237.72
	std$_{time}$	46.39	137.94	94.08	109.70	260.19	333.28	196.32
	p$_{time}$	–	1.12E-18*	2.75E-19*	2.75E-19*	1.10E-18*	6.35E-12*	1.34E-10*
	rank	1	5	6	7	4	3	2
case 4	suc%	98	4	4	2	34	74	100
	mean$_{time}$	216.19	987.95	991.50	997.87	826.31	680.63	292.40
	std$_{time}$	120.37	59.64	54.23	15.30	282.60	262.37	76.79
	p$_{time}$	–	2.79E-19*	2.72E-19*	1.87E-19*	9.58E-16*	7.44E-16*	1.22E-9*
	rank	2	5	6	7	4	3	1
case 5	suc%	100	48	50	22	22	90	94
	mean$_{time}$	187.45	887.01	825.84	936.26	830.12	338.01	440.65
	std$_{time}$	106.52	161.25	211.22	143.65	294.16	290.64	277.63
	p$_{time}$	–	6.25E-18*	1.87E-17*	1.03E-18*	8.24E-17*	9.74E-6*	6.19E-11*
	rank	1	6	4	7	5	3	2
case 6	suc%	100	34	40	16	18	86	98
	mean$_{time}$	137.50	898.92	878.68	959.89	884.71	350.81	247.74
	std$_{time}$	72.09	171.79	189.88	123.71	264.31	299.42	171.16
	p$_{time}$	–	2.10E-18*	2.82E-18*	3.73E-19*	2.46E-18*	1.57E-9*	5.36E-10*
	rank	1	5	4	7	6	3	2
case 7	suc%	100	4	0	0	42	70	100
	mean$_{time}$	191.26	991.01	1000	1000	823.40	705.99	305.69
	std$_{time}$	37.71	68.16	0	0	258.32	278.97	69.34
	p$_{time}$	–	6.63E-20*	3.31E-20*	3.31E-20*	1.61E-17*	2.61E-17*	1.12E-14*
	rank	1	5	6	6	4	3	2
case 8	suc%	100	50	34	24	40	92	96
	mean$_{time}$	158.87	856.64	894.72	950.38	826.42	328.47	402.56
	std$_{time}$	45.22	185.59	185.61	104.28	257.13	270.58	268.09
	p$_{time}$	–	3.89E-18*	1.85E-18*	7.98E-19*	4.33E-18*	2.42E-12*	2.14E-14*
	rank	1	5	6	7	4	3	2
case 9	suc%	100	6	6	4	24	84	98
	mean$_{time}$	129.29	987.23	979.37	997.16	835.19	370.34	242.93
	std$_{time}$	58.33	51.78	91.22	16.91	284.96	312.06	166.77
	p$_{time}$	–	9.12E-20 *	9.11E-20*	6.63E-20*	3.06E-18*	3.90E-10*	1.23E-10*
	rank	1	6	5	7	4	3	2
avg-rank		1.11	5.22	5.22	6.88	4.55	3	1.88

[1] suc%: The value denotes the success rate among 50 runs.

[2] mean$_{time}$: The value is the mean of search time among 50 runs.

[3] std$_{time}$: The value is the standard error of search time among 50 runs.

[4] p_{time}: The value is a p value of a two-sample Wilcoxon rank sum test for search time between the CFPSO algorithm and other comparison algorithms in the corresponding column. The p value being smaller means that the results obtained by the CFPSO algorithm are significant compared with results obtained by other algorithms.

[5] Composite ranking is applied to all the algorithms, which is evaluated by the descending order of suc% and the ascending order of search time.

[6] $*$: The difference between two samples is significant at level $\alpha = 0.05$.

Table 4.8: The statistical results of search time for 5 robots based on 50 runs.

Robots	Index	CFPSO	PSO	CPSO	WUI-45	WUII	PSO-IM	LPSO
case 1	suc%[1]	100	10	4	8	72	90	100
	mean$_{time}$[2]	165.06	983.61	993.15	975.72	718.74	616.85	267.54
	std$_{time}$[3]	24.55	53.38	37.68	85.26	256.87	206.78	53.14
	p_{time}[4]	–	1.63E-19*[6]	6.63E-20*	1.23E-19*	1.16E-17*	1.72E-17*	3.01E-16*
	rank[5]	1	6	7	5	4	3	2
case 2	suc%	100	82	74	68	52	100	100
	mean$_{time}$	141.25	698.38	745.51	793.25	687.27	246.89	403.68
	std$_{time}$	32.96	197.59	203.52	217.49	334.58	165.65	258.71
	p$_{time}$	–	6.87E-18*	6.50E-18*	6.83E-18*	1.10E-17*	1	7.07E-16*
	rank	1	4	5	7	6	2	3
case 3	suc%	100	8	14	6	36	98	92
	mean$_{time}$	103.05	984.29	967.58	983.06	798.46	207.22	349.77
	std$_{time}$	20.09	67.87	100.20	80.12	311.03	174.64	279.45
	p$_{time}$	–	1.23E-19*	2.75E-19*	9.11E-20*	2.53E-18*	1.12E-12*	2.41E-4*
	rank	1	7	5	6	4	2	3
case 4	suc%	100	20	18	26	68	98	100
	mean$_{time}$	168.50	962.01	969.25	945.41	655.06	526.41	256.18
	std$_{time}$	22.57	110.19	93.63	104.02	308.83	216.85	50.23
	p$_{time}$	–	5.42E-19*	4.38E-19*	9.52E-19*	8.18E-18*	8.46E-18*	8.37E-16*
	rank	1	6	7	5	4	3	2
case 5	suc%	100	72	68	62	28	98	94
	mean$_{time}$	151.45	742.11	764.81	817.17	834.95	244.33	366.92
	std$_{time}$	62.07	229.53	210.05	200.53	283.17	184.68	262.45
	p$_{time}$	–	1.47E-17*	8.69E-18*	6.94E-18*	3.47E-18*	1.54E-8*	4.19E-13*
	rank	1	4	5	6	7	2	3
case 6	suc%	100	64	38	48	38	98	90
	mean$_{time}$	101.45	799.81	871.54	887.91	789.09	222.06	380.05
	std$_{time}$	11.84	199.61	189.20	158.13	303.71	166.67	319.44
	p$_{time}$	–	5.67E-18*	2.23E-18*	3.60E-18*	2.23E-18*	1.73E-15*	3.30E-17*
	rank	1	4	7	6	5	2	3
case 7	suc%	100	6	2	6	64	94	100
	mean$_{time}$	175.58	984.26	997.51	986.49	725.37	583.19	306.42
	std$_{time}$	34.56	85.47	17.65	60.61	287.51	202.52	76.40
	p$_{time}$	–	9.12E-20*	4.73E-20*	9.12E-20*	6.14E-17*	1.36E-17*	1.31E-15*
	rank	1	5	7	6	4	3	2
case 8	suc%	100	68	74	58	46	98	94
	mean$_{time}$	132.69	737.01	756.74	822.34	762.34	240.75	403.15
	std$_{time}$	20.36	217.92	203.45	198.85	304.75	185.45	273.43
	p$_{time}$	–	6.05E-18*	6.50E-18*	4.97E-18*	3.99E-18*	9.66E-13*	6.71E-17*
	rank	1	5	4	7	6	2	3
case 9	suc%	100	2	0	2	52	98	100
	mean$_{time}$	108.38	999.29	1000	999.48	748.79	206.56	237.65
	std$_{time}$	27.48	5.08	0	3.74	314.80	189.82	234.70
	p$_{time}$	–	4.73E-20*	3.30E-20*	4.73E-20*	7.23E-18*	7.64E-10*	8.81E-10*
	rank	1	5	7	6	4	3	2
average rank		1	5.11	6	6	4.88	2.44	2.55

[1] suc%: The value denotes the success rate among 50 runs.

[2] mean$_{time}$: The value is the mean of search time among 50 runs.

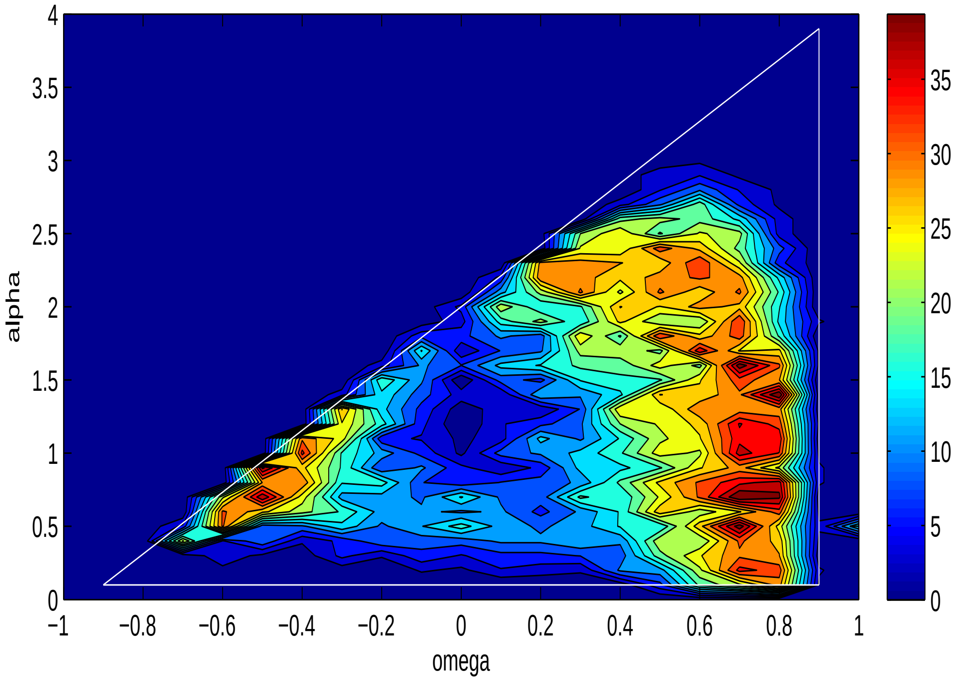
[3] std$_{time}$: The value is the standard error of search time among 50 runs.

[4] p_{time}: The value is a p value of a two-sample Wilcoxon rank sum test for search time between the CFPSO algorithm and other comparison algorithms in the corresponding column. The p value being smaller means that the results obtained by the CFPSO algorithm are significant compared with results obtained by other algorithms.

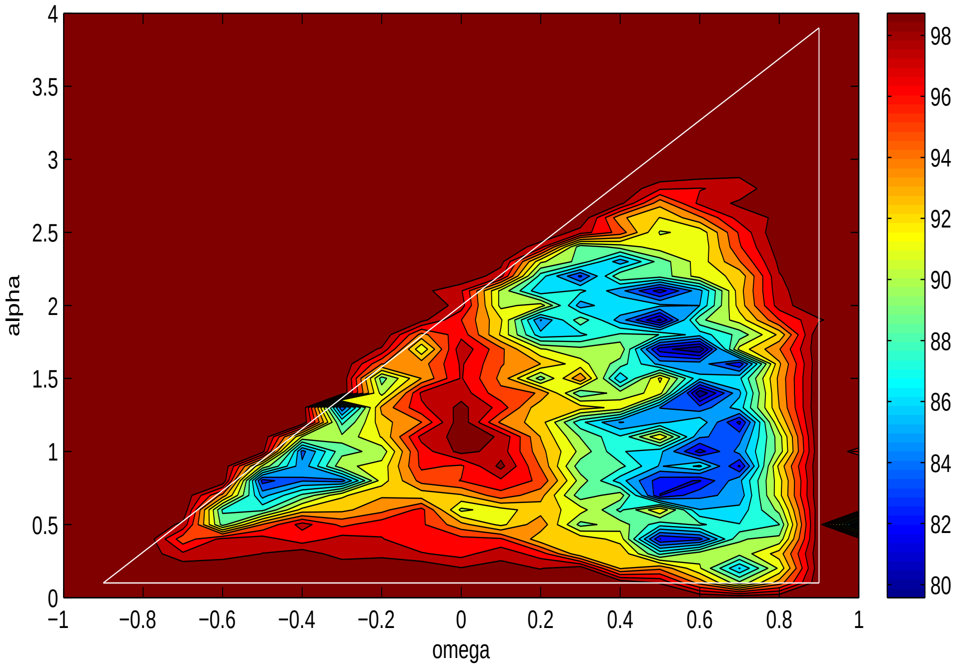
[5] Composite ranking is applied to all the algorithms, which is evaluated by the descending order of suc% and the ascending order of search time.

[6] $*$: The difference between two samples is significant at level $\alpha = 0.05$.

GPSO algorithm and analyzed the convergence of the discrete-time FPSO algorithm. Finally, this study has shown the characteristics of the discrete-time FPSO algorithm through numerical simulations on the benchmark functions and the performance capabilities of the continuous-time FPSO algorithm that is used as the PSO-based finite-time motion control algorithm for the problem of odour source localisation.

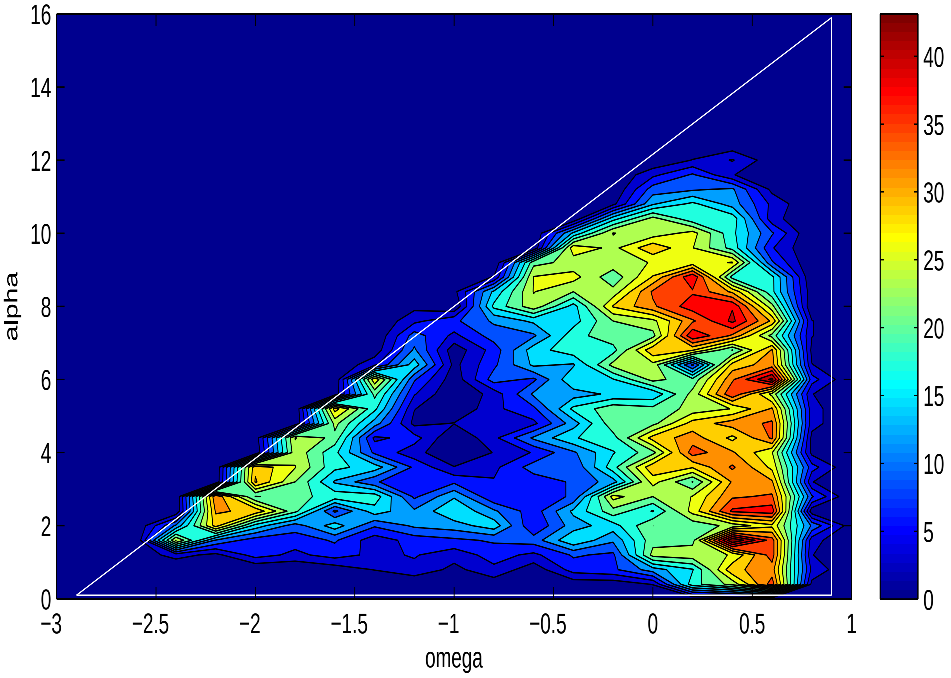


(a) Success rates for PSO

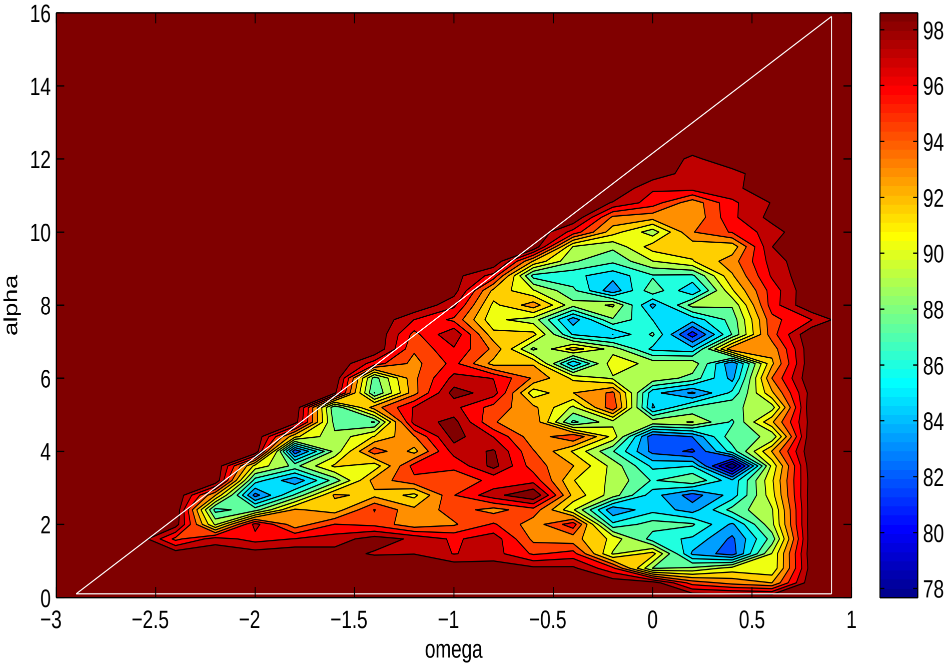


(b) Average iterations for PSO

Figure 4.8: Success rates and average iterations for the Griewank function for PSO.

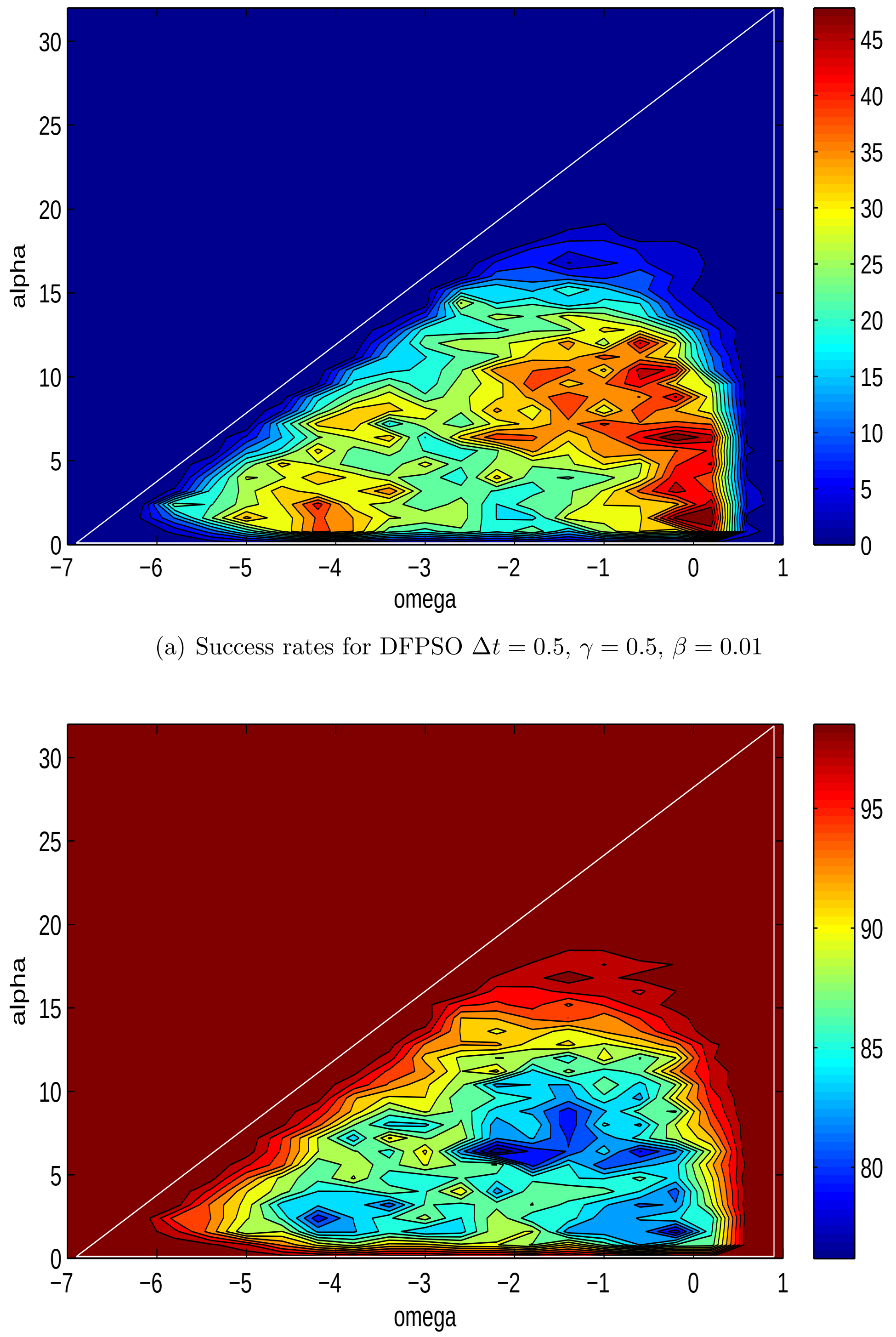


(a) Success rates for GPSO $\Delta t = 0.5$



(b) Average iterations for GPSO $\Delta t = 0.5$

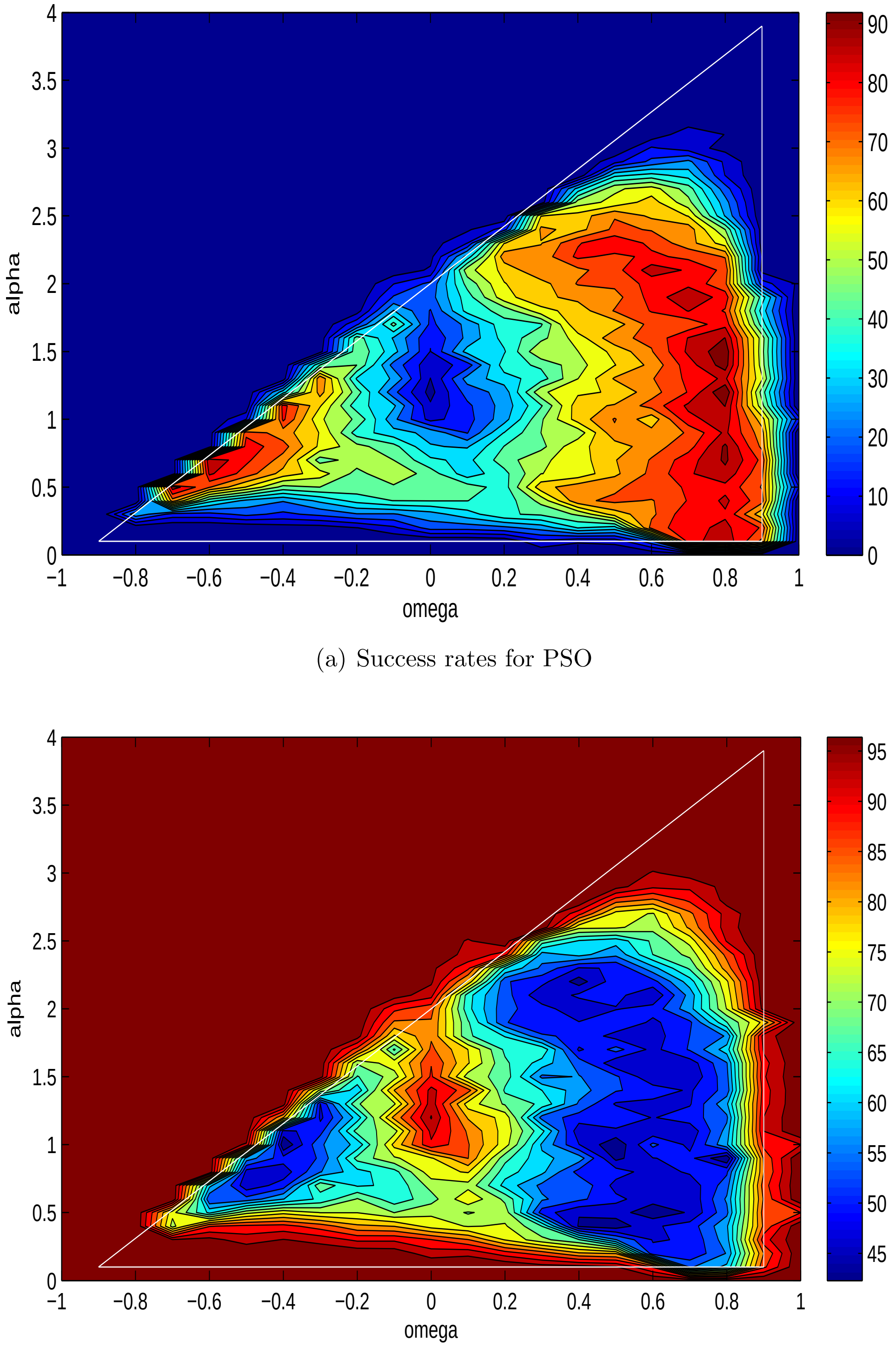
Figure 4.9: Success rates and average iterations for the Griewank function for GPSO.



(a) Success rates for DFPSO $\Delta t = 0.5$, $\gamma = 0.5$, $\beta = 0.01$

(b) Average iterations for DFPSO $\Delta t = 0.5$, $\gamma = 0.5$, $\beta = 0.01$

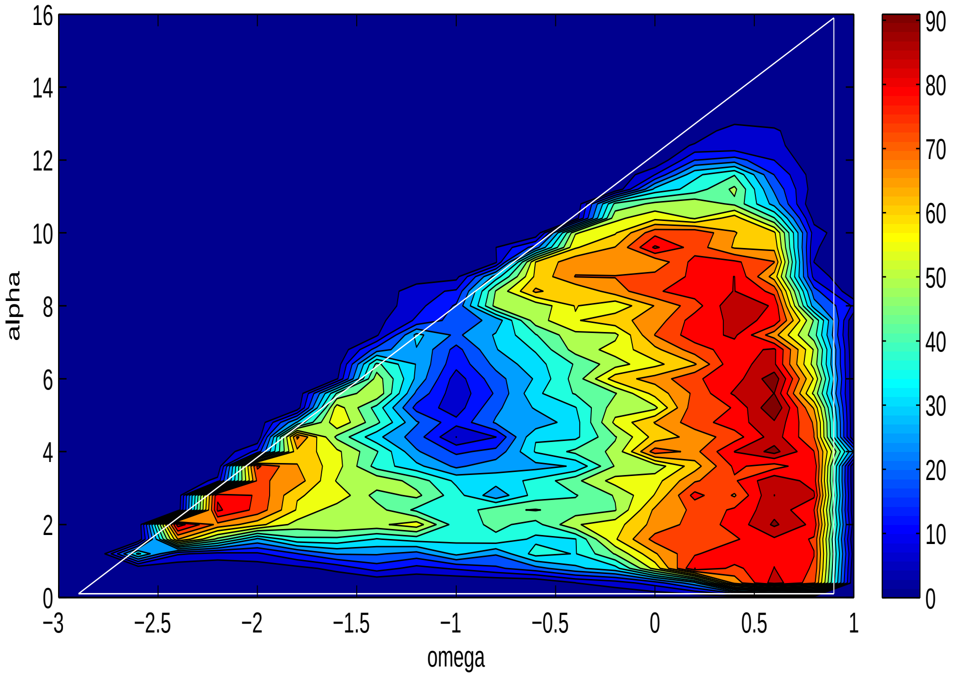
Figure 4.10: Success rates and average iterations for the Griewank function for DFPSO.



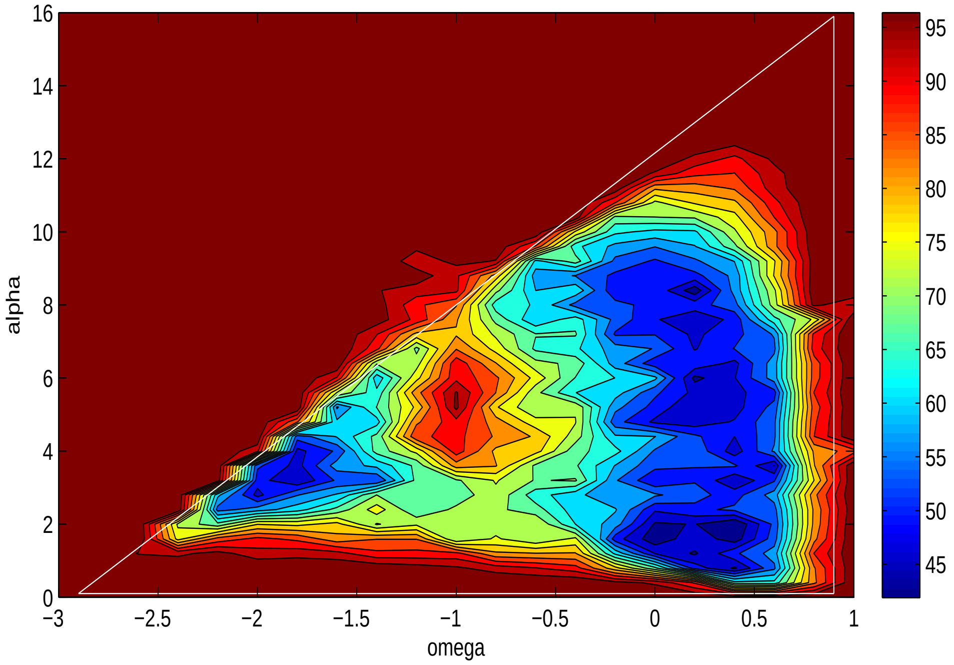
(a) Success rates for PSO

(b) Average iterations for PSO

Figure 4.11: Success rates and average iterations for the Rastrigin function for PSO.

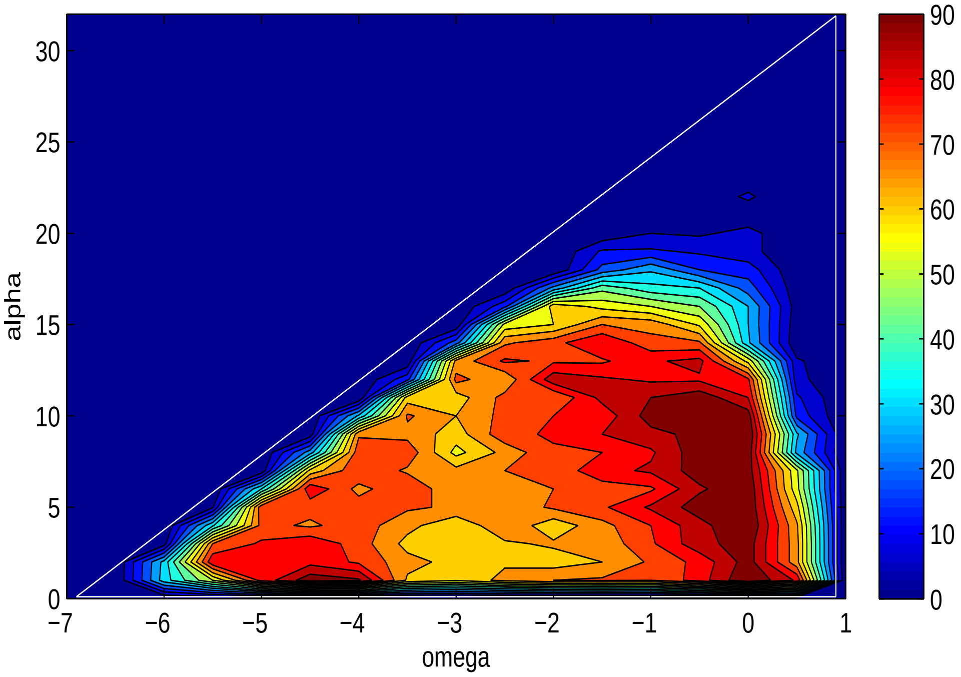


(a) Success rates for GPSO $\Delta t = 0.5$

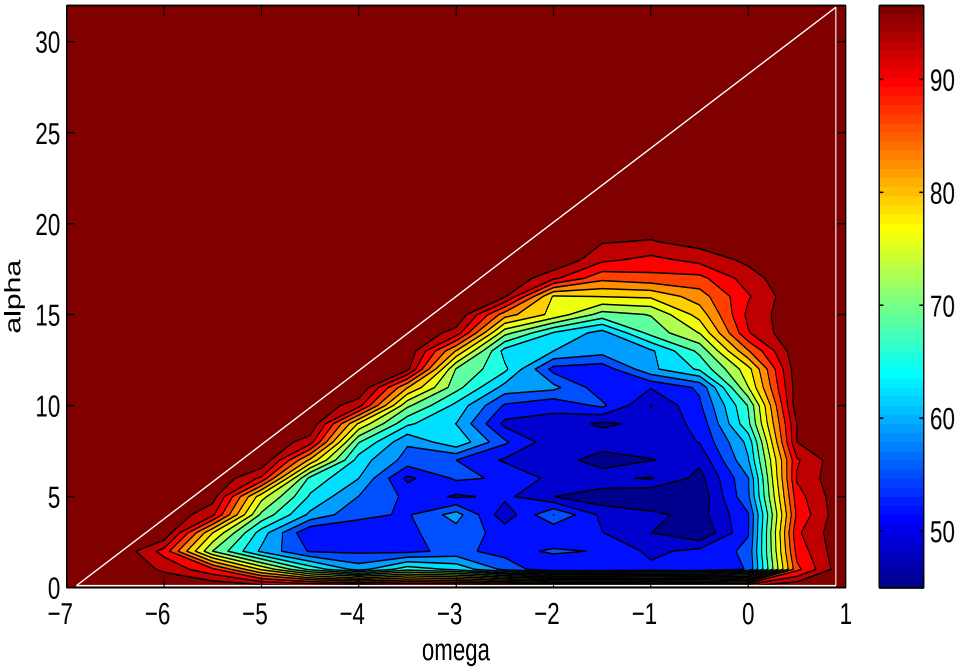


(b) Average iterations for GPSO $\Delta t = 0.5$

Figure 4.12: Success rates and average iterations for the Rastrigin function for GPSO.

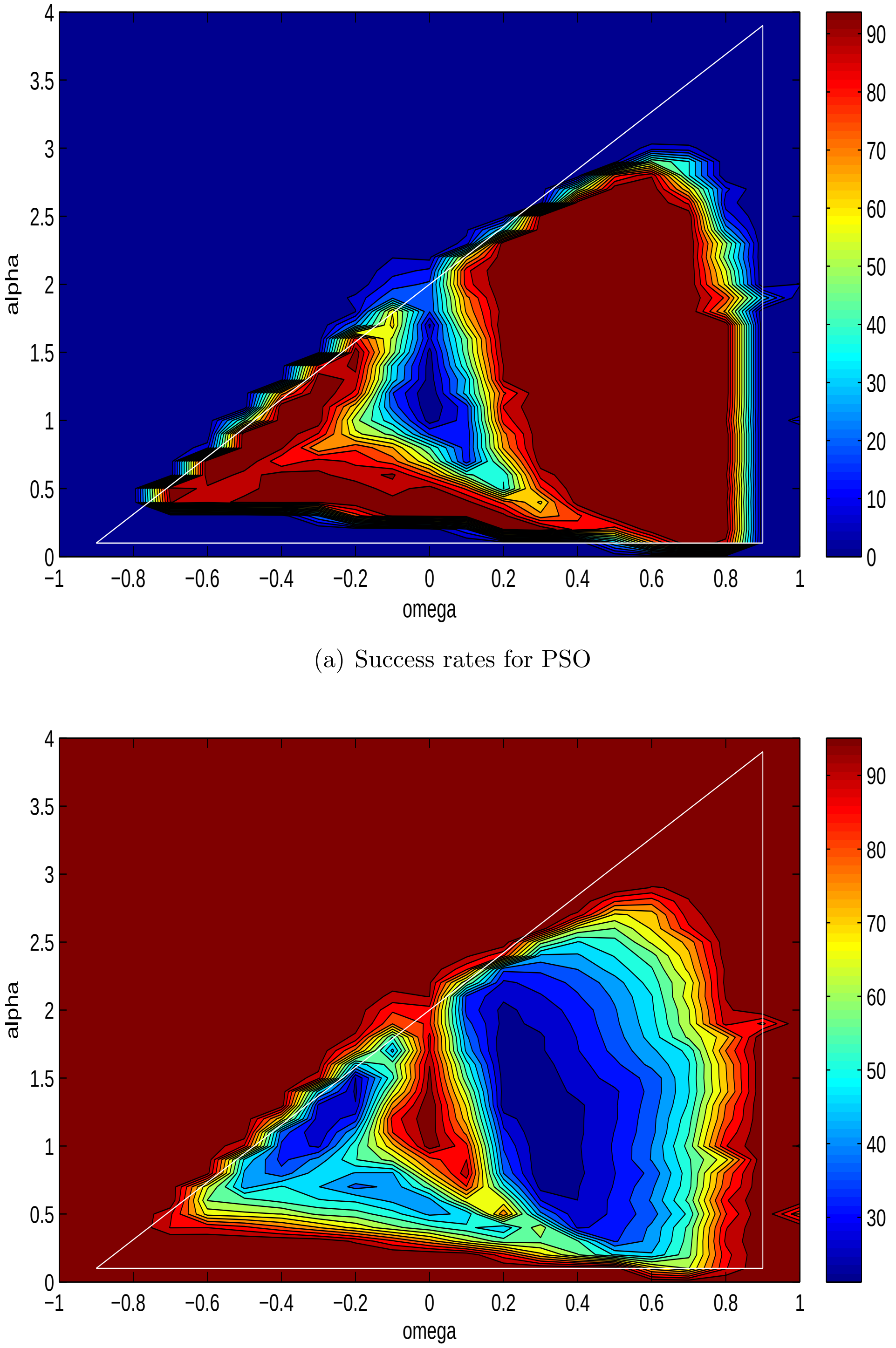


(a) Success rates for DFPSO $\Delta t = 0.5$, $\gamma = 0.5$, $\beta = 0.01$



(b) Average iterations for DFPSO $\Delta t = 0.5$, $\gamma = 0.5$, $\beta = 0.01$

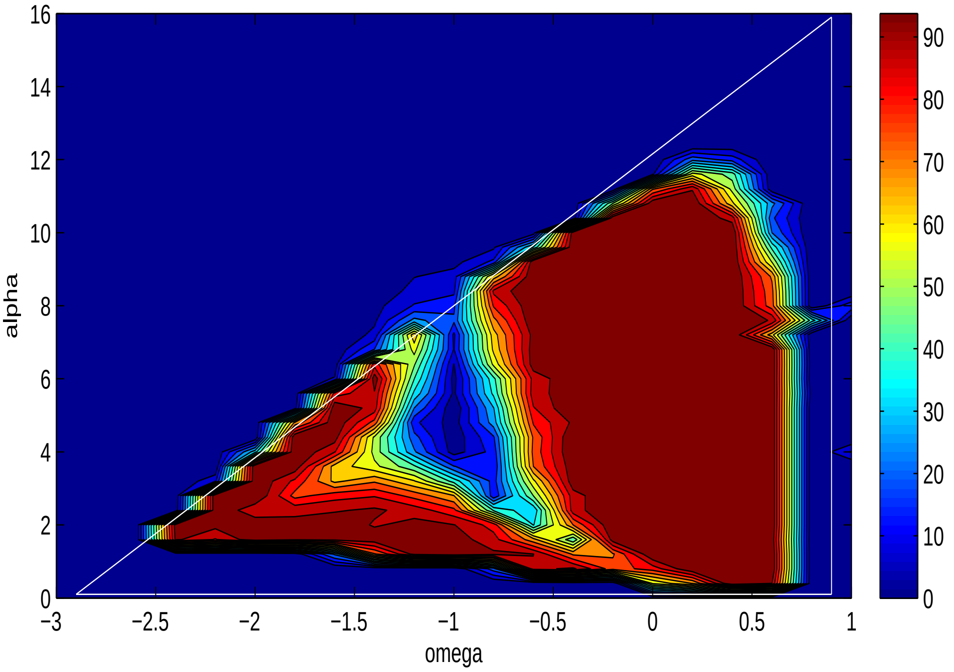
Figure 4.13: Success rates and average iterations for the Rastrigin function for DFPSO.



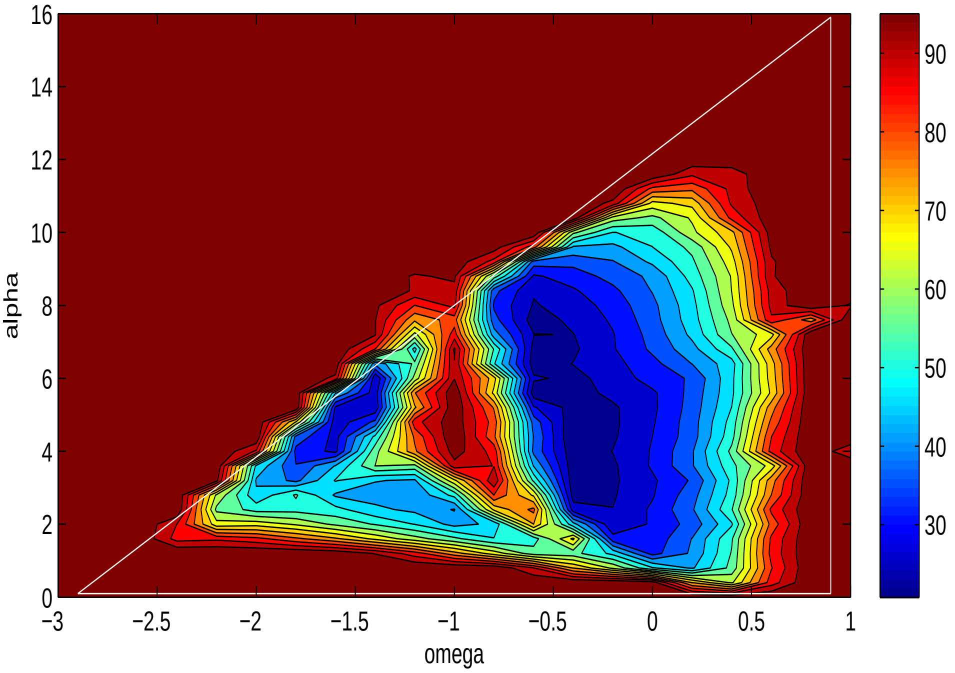
(a) Success rates for PSO

(b) Average iterations for PSO

Figure 4.14: Success rates and average iterations for the Ackley function for PSO.

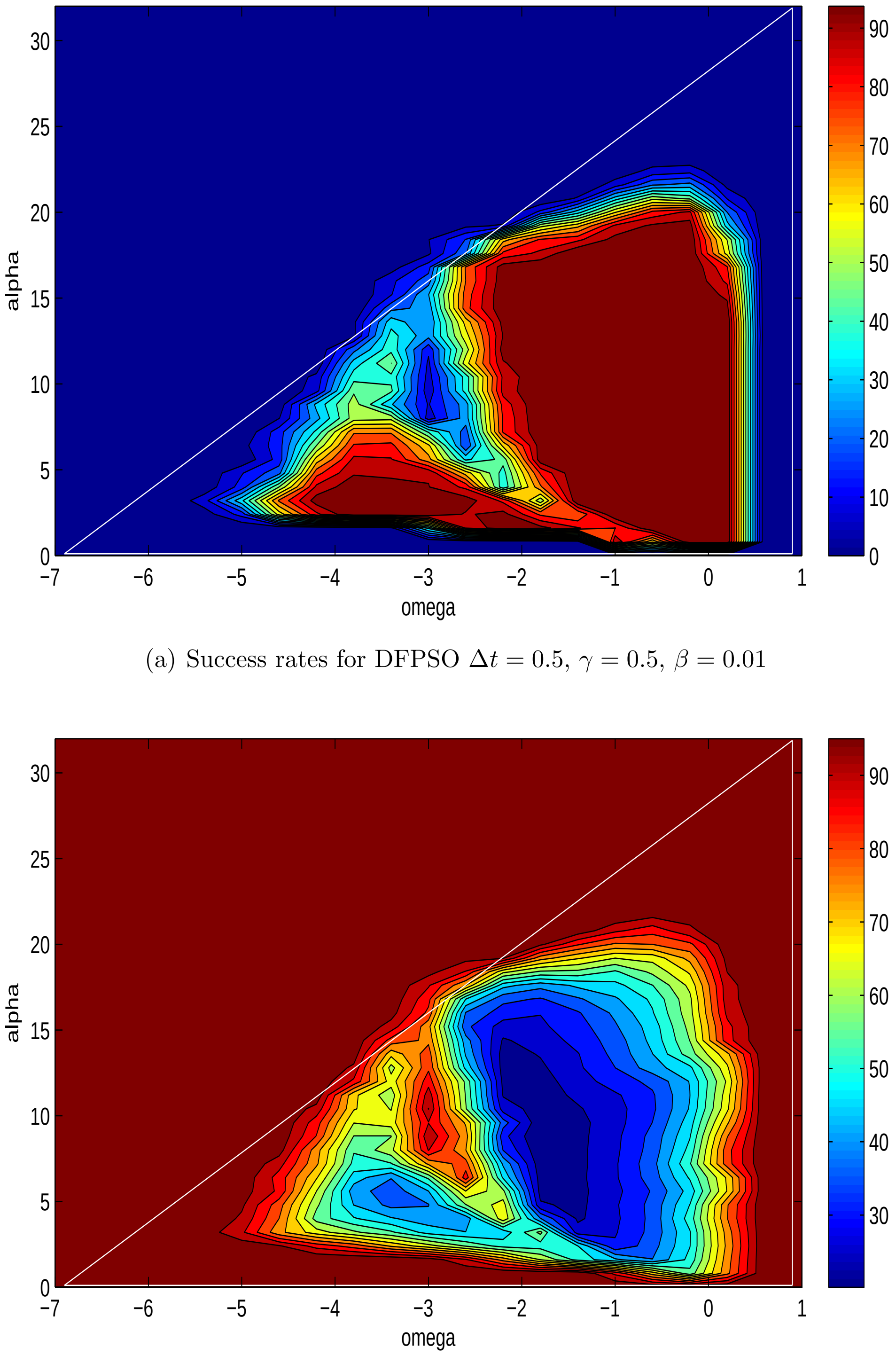


(a) Success rates for GPSO $\Delta t = 0.5$



(b) Average iterations for GPSO $\Delta t = 0.5$

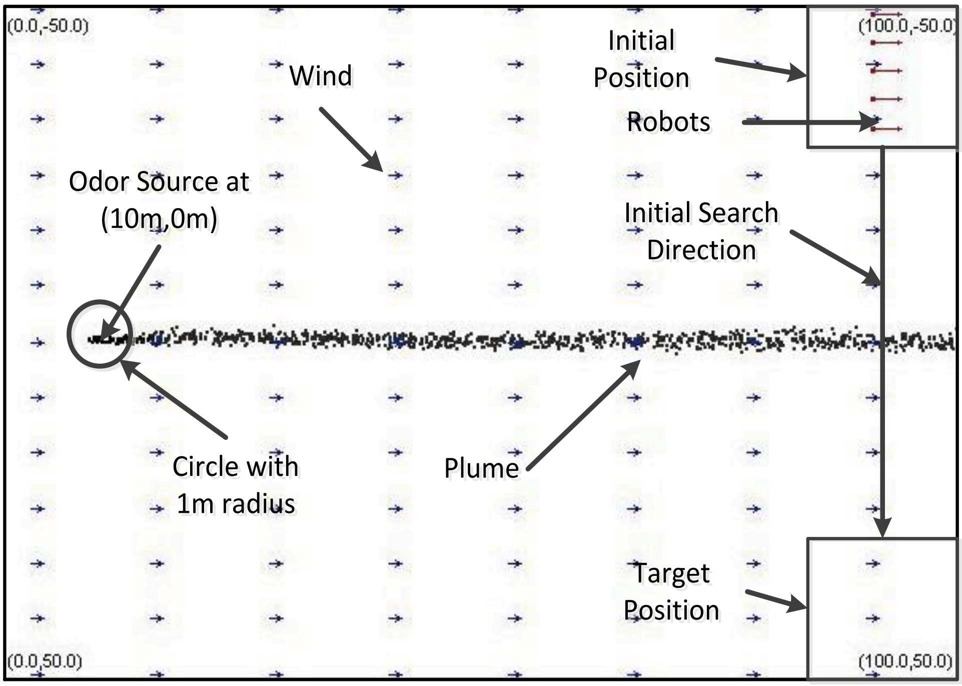
Figure 4.15: Success rates and average iterations for the Ackley function for GPSO.



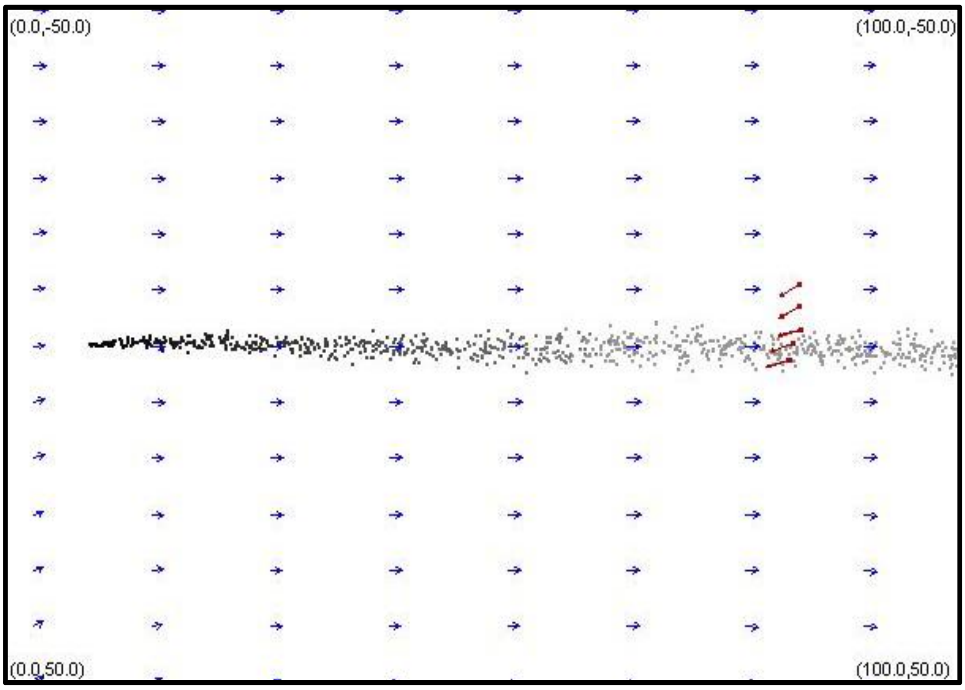
(a) Success rates for DFPSO $\Delta t = 0.5$, $\gamma = 0.5$, $\beta = 0.01$

(b) Average iterations for DFPSO $\Delta t = 0.5$, $\gamma = 0.5$, $\beta = 0.01$

Figure 4.16: Success rates and average iterations for the Ackley function for DFPSO.

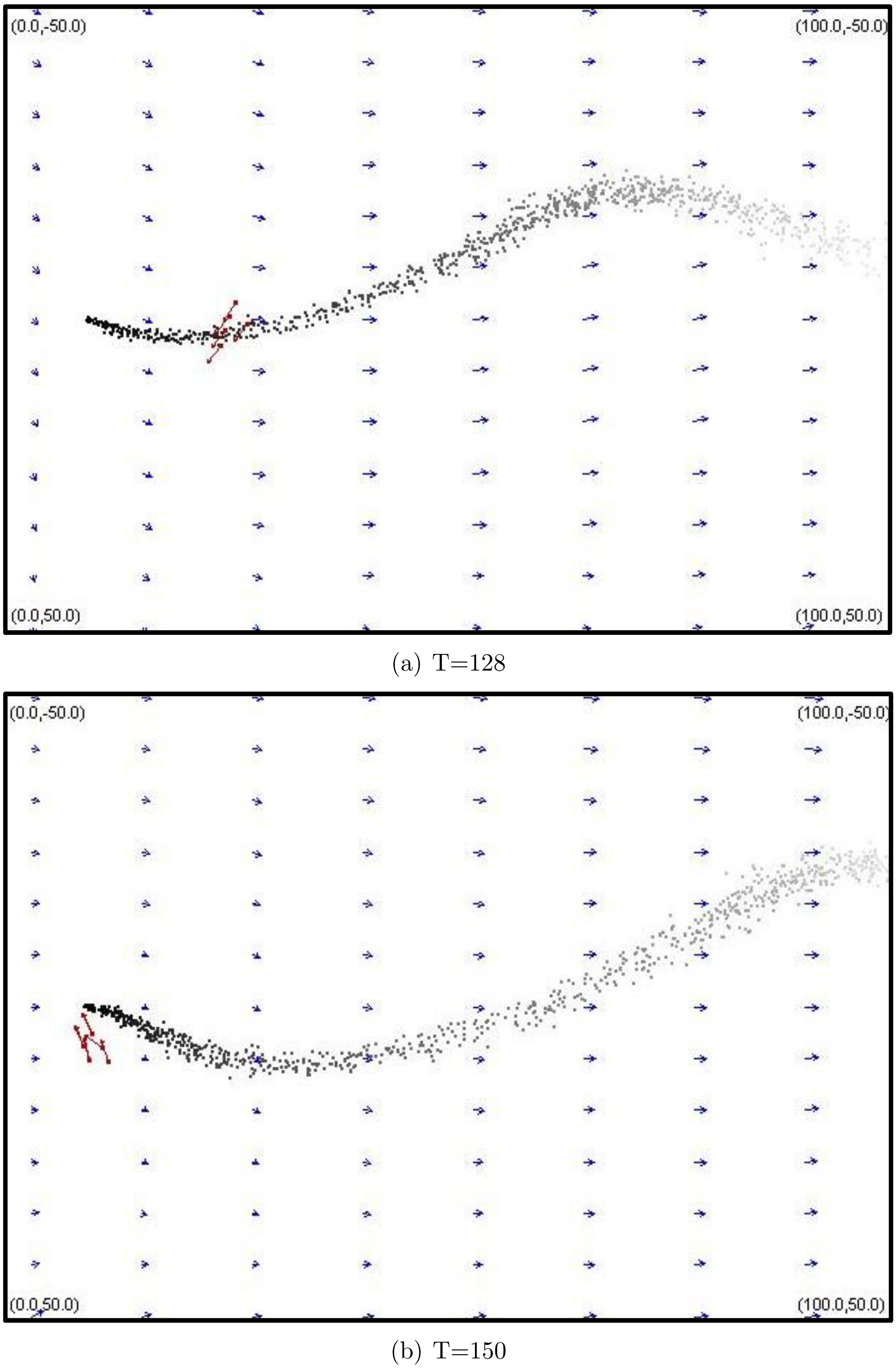


(a) T=0s



(b) T=49s

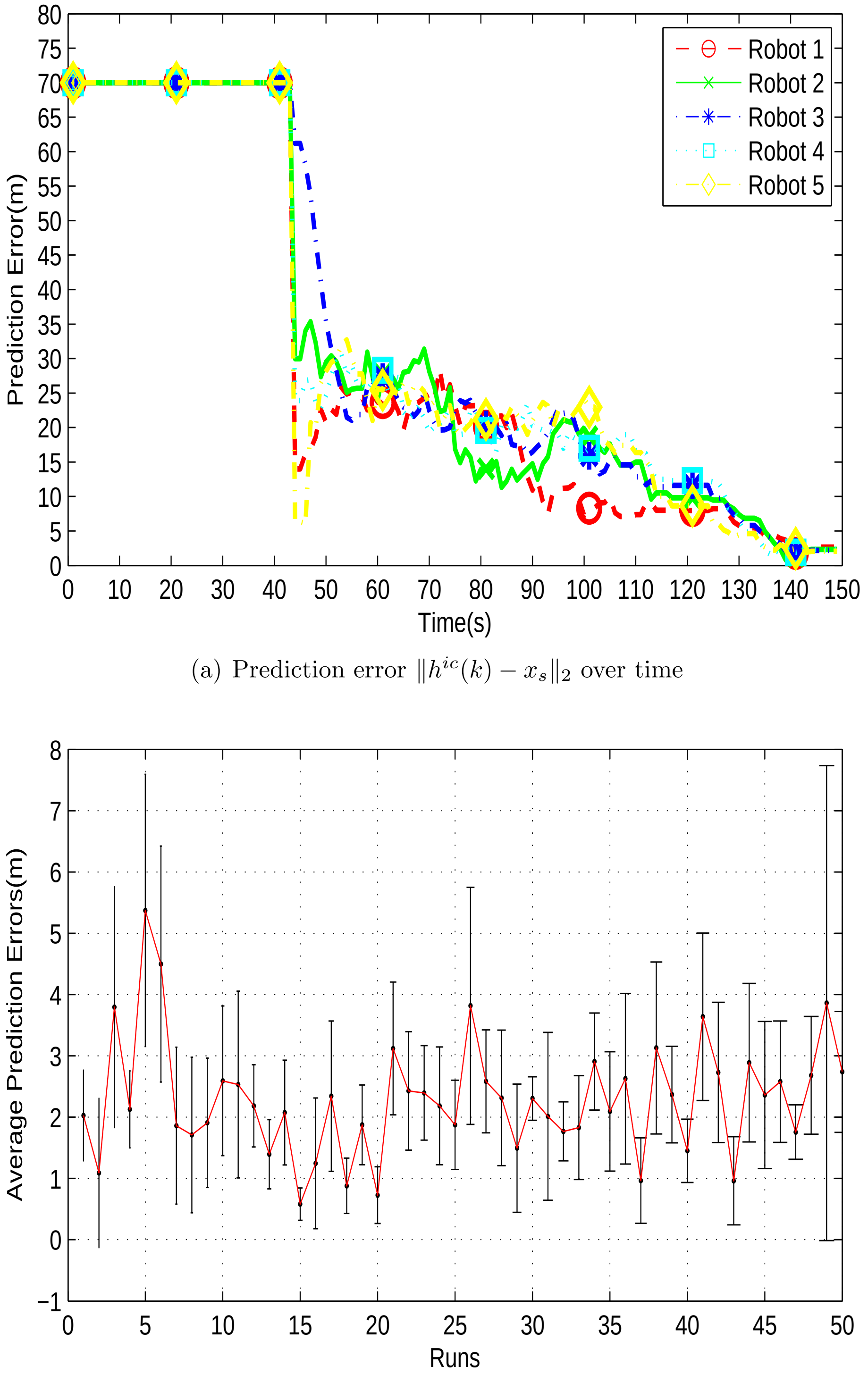
Figure 4.17: The search process of five robots for T=0s and T=49s.



(a) T=128

(b) T=150

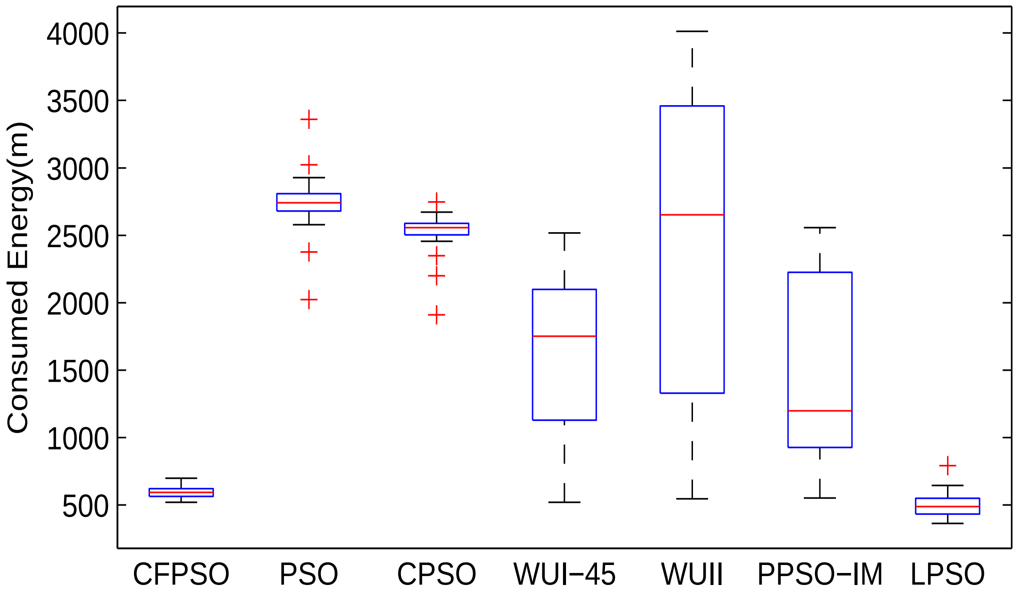
Figure 4.18: The search process of five robots for T=128s and T=150s.



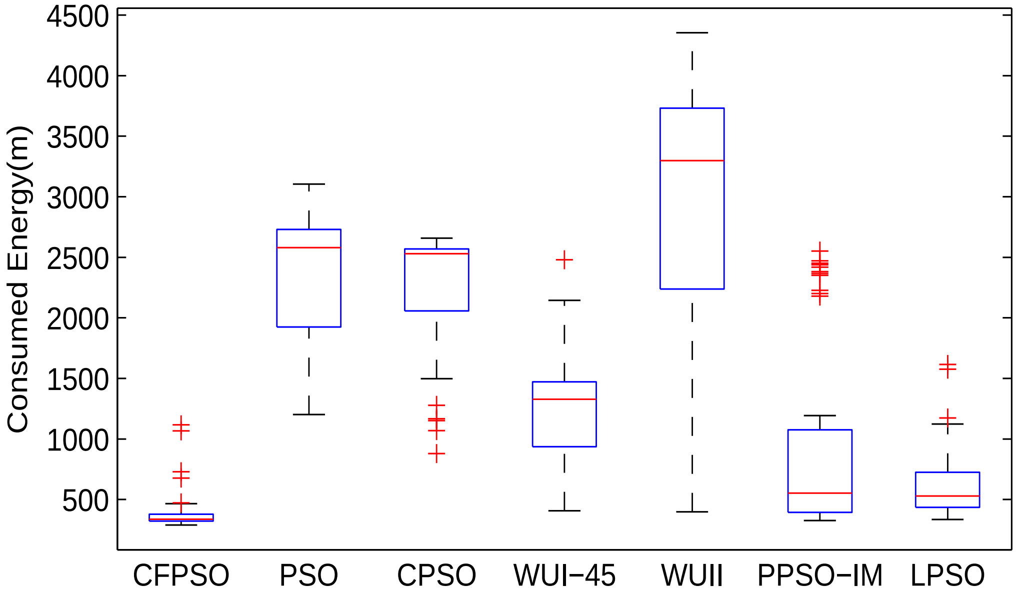
(a) Prediction error $\|h^{ic}(k) - x_s\|_2$ over time

(b) The average prediction errors $\frac{\sum_{i=1}^{5} ||h^{ic}(k)-x_s||_2}{5}$ with 50 runs

Figure 4.19: The prediction errors. x_s is the real position of the odour source.

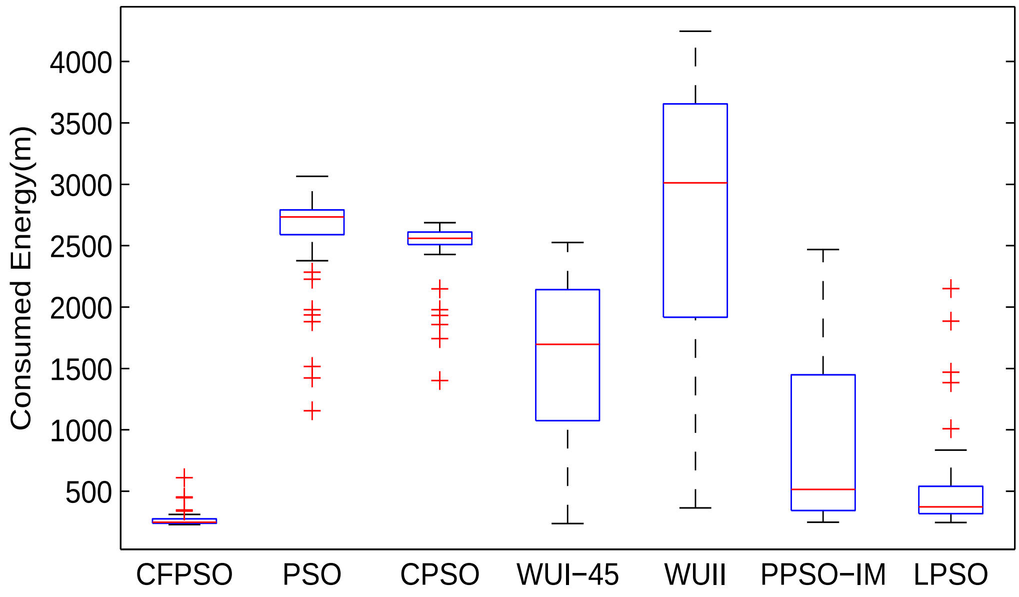


(a) Case 1

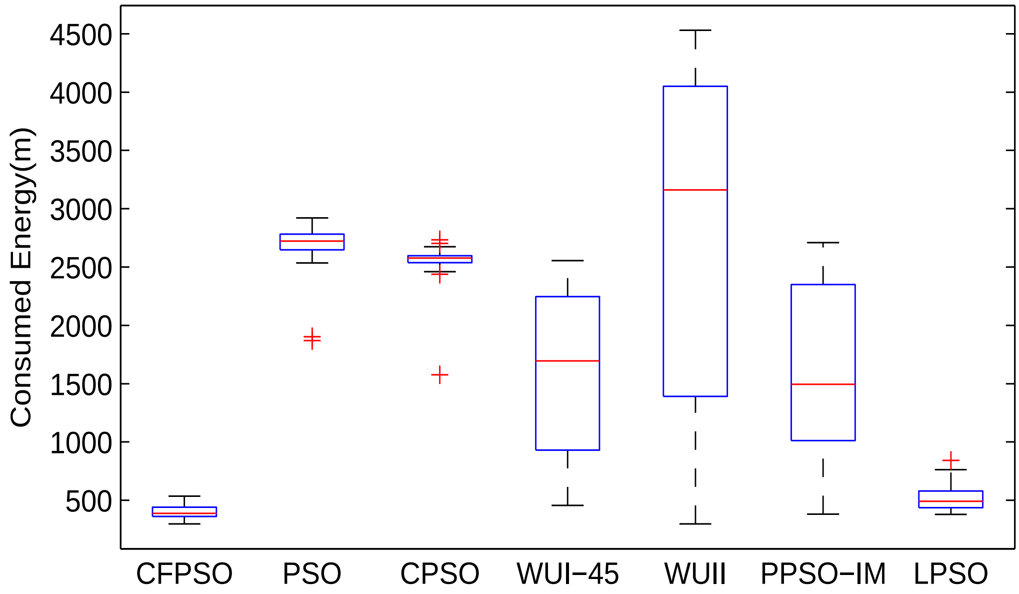


(b) Case 2

Figure 4.20: The consumed energy for case 1 and case 2 based on 3 robots with 50 runs.

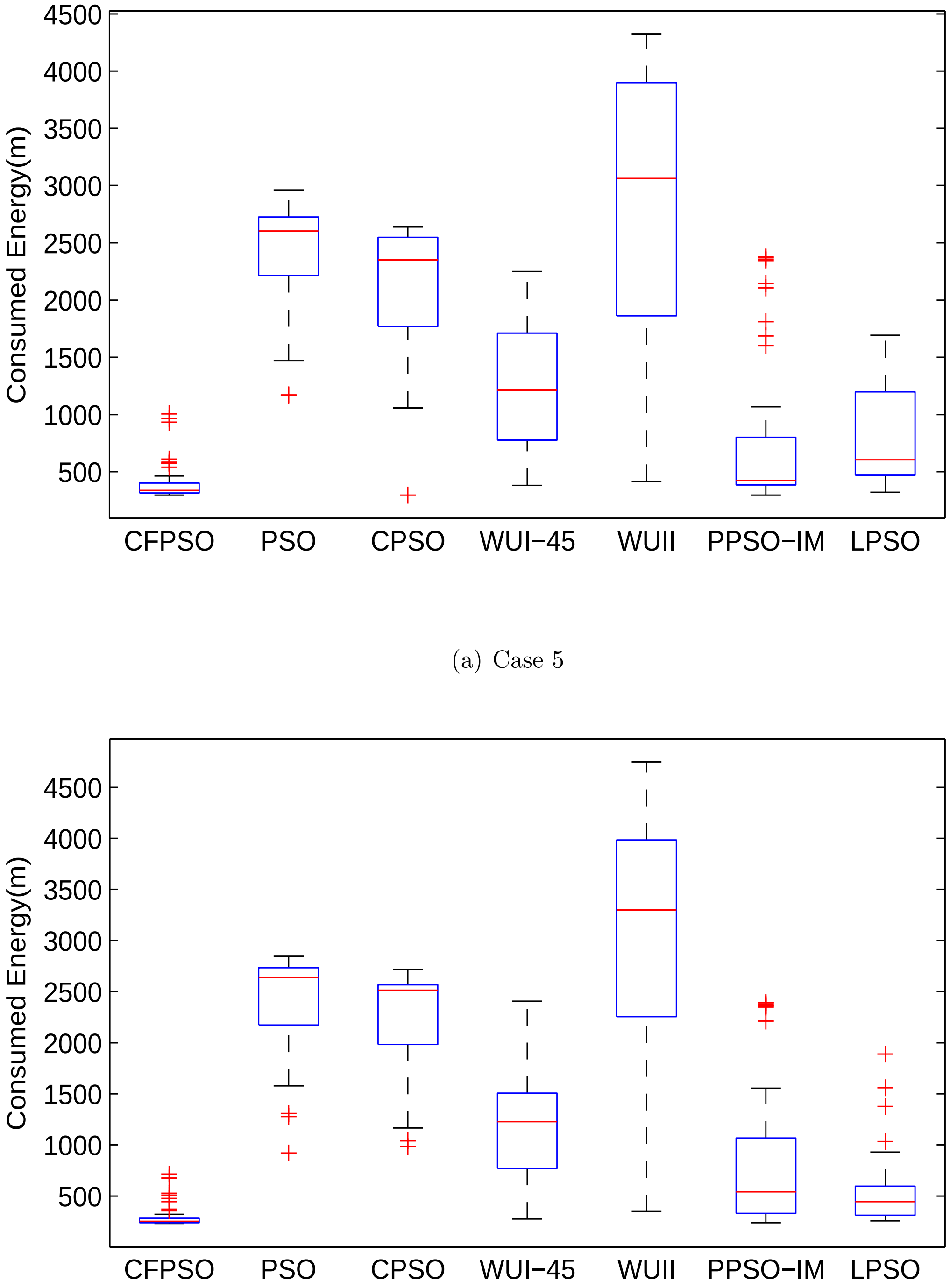


(a) Case 3



(b) Case 4

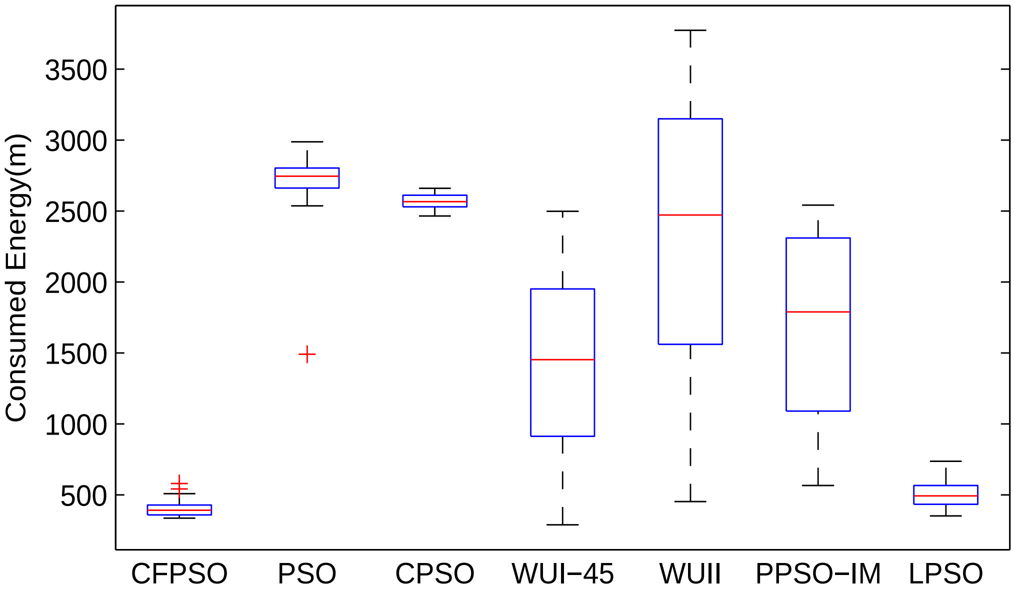
Figure 4.21: The consumed energy for case 3 and case 4 based on 3 robots with 50 runs.



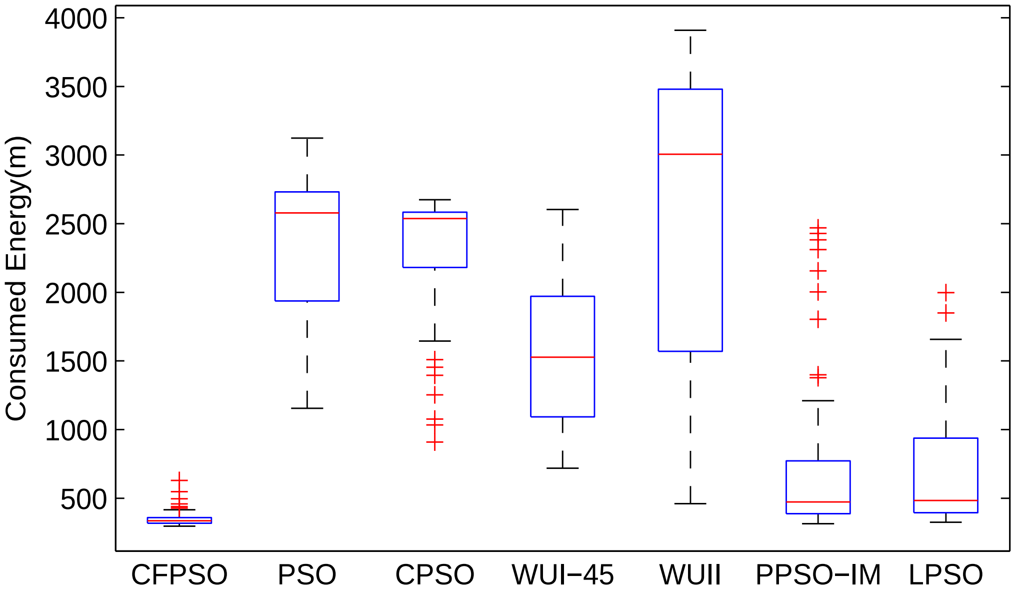
(a) Case 5

(b) Case 6

Figure 4.22: The consumed energy for case 5 and case 6 based on 3 robots with 50 runs.



(a) Case 7



(b) Case 8

Figure 4.23: The consumed energy for case 7 and case 8 based on 3 robots with 50 runs.

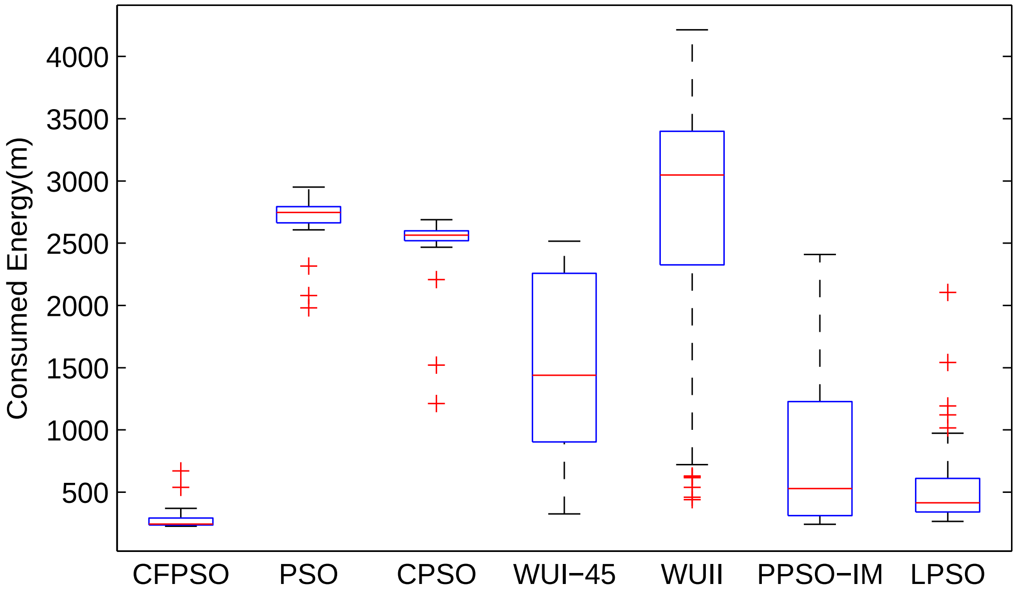
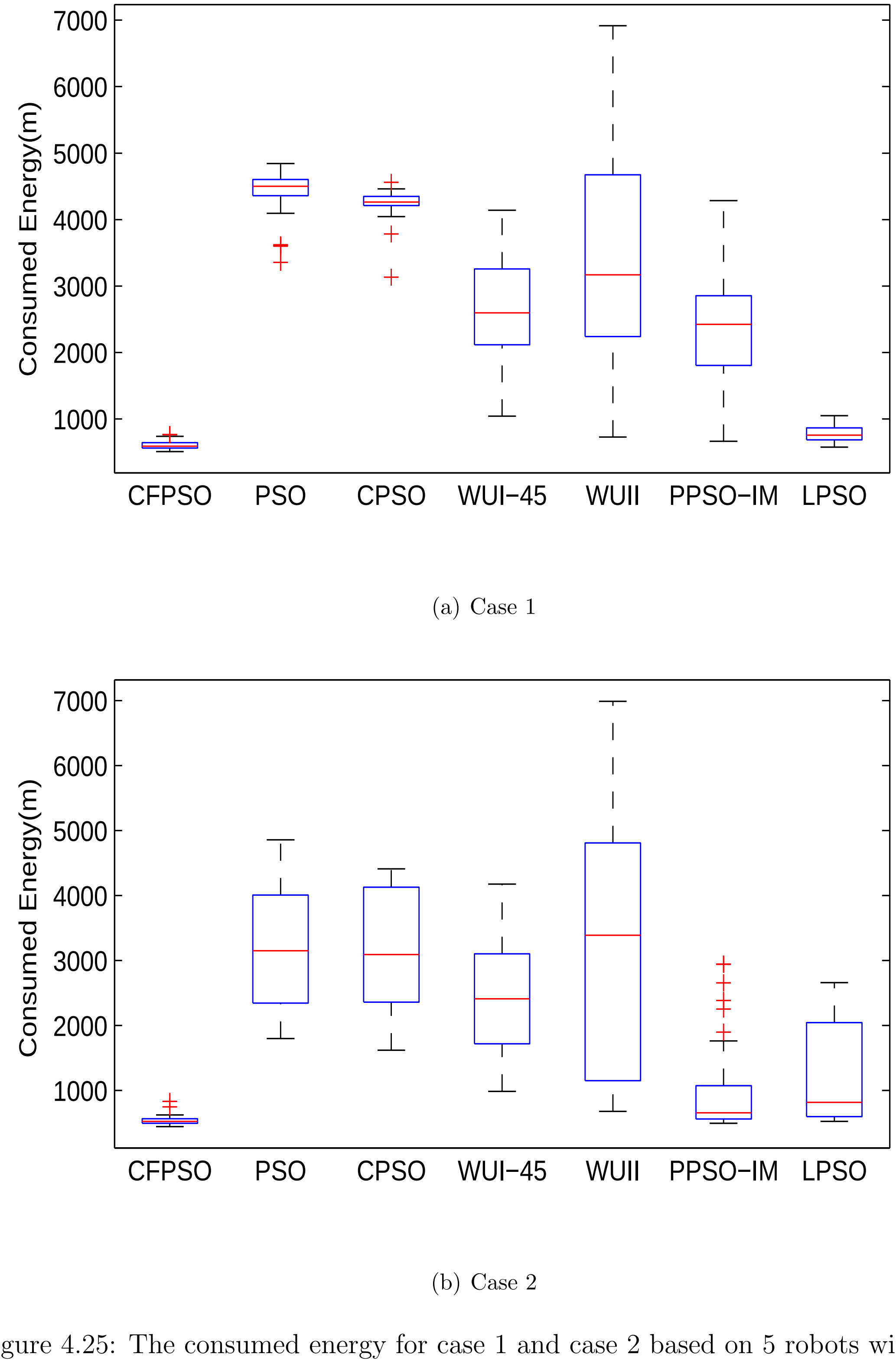


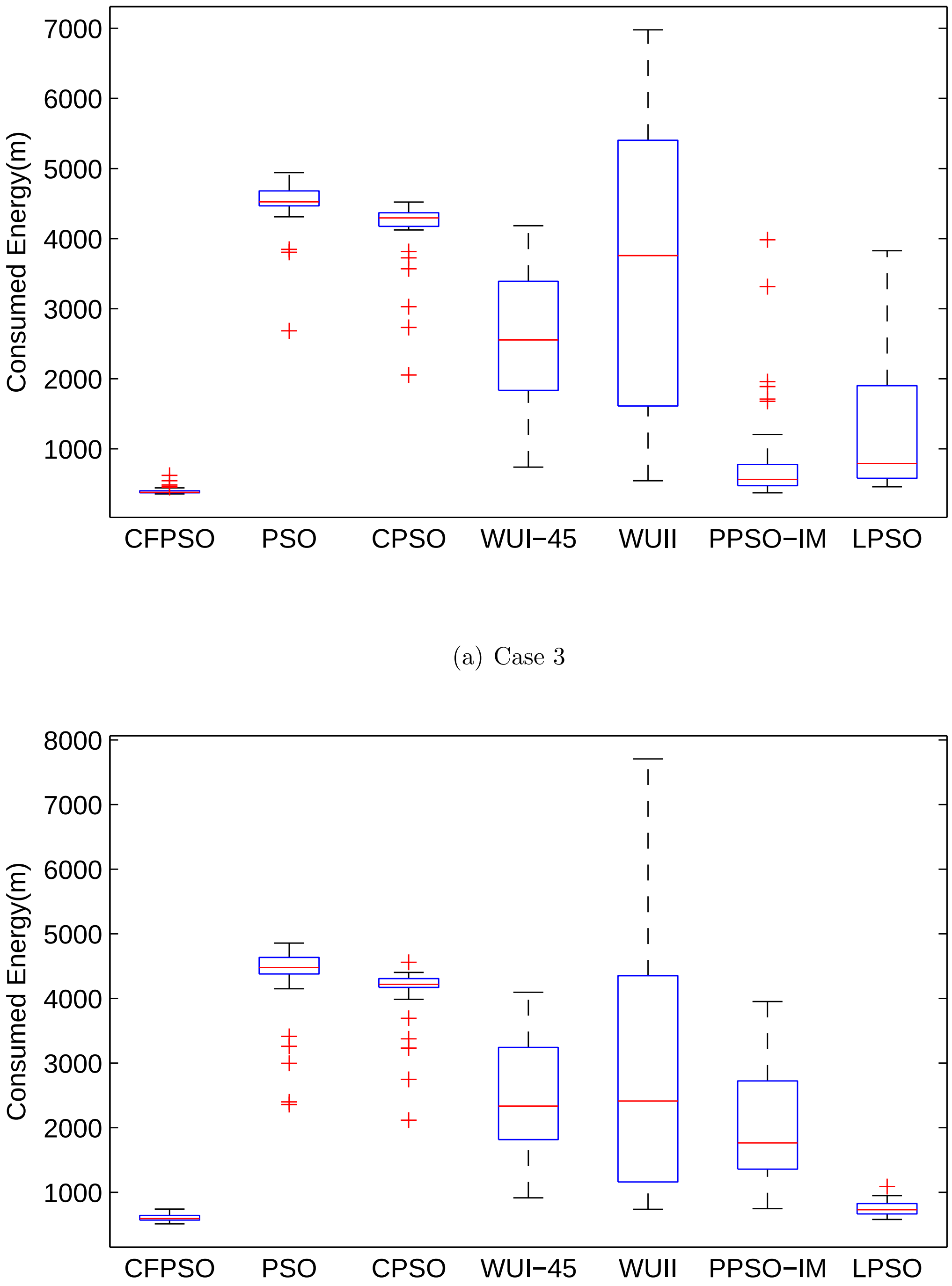
Figure 4.24: The consumed energy for case 9 based on 3 robots with 50 runs.



(a) Case 1

(b) Case 2

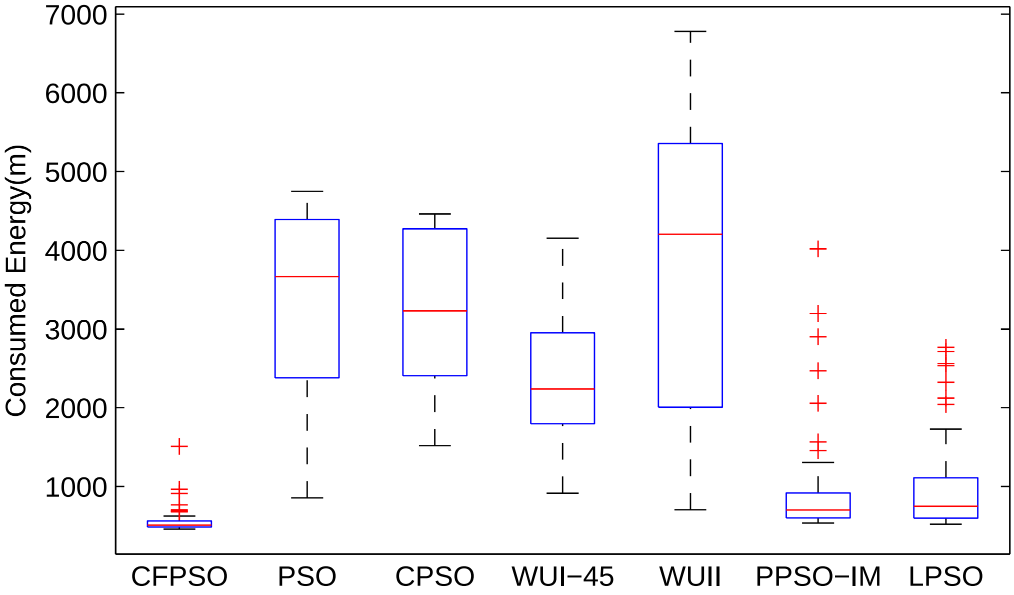
Figure 4.25: The consumed energy for case 1 and case 2 based on 5 robots with 50 runs.



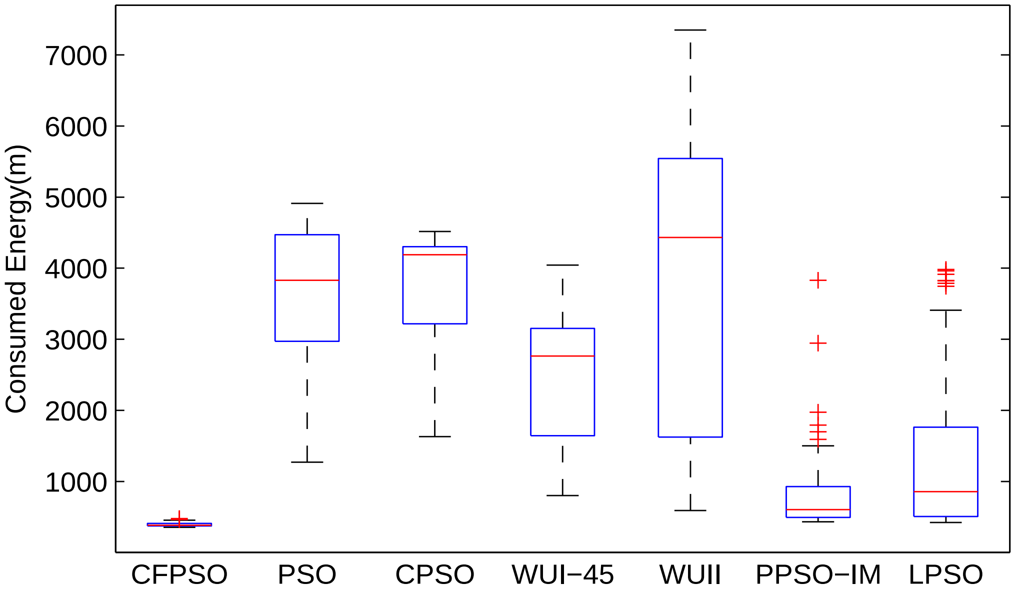
(a) Case 3

(b) Case 4

Figure 4.26: The consumed energy for case 3 and case 4 based on 5 robots with 50 runs.

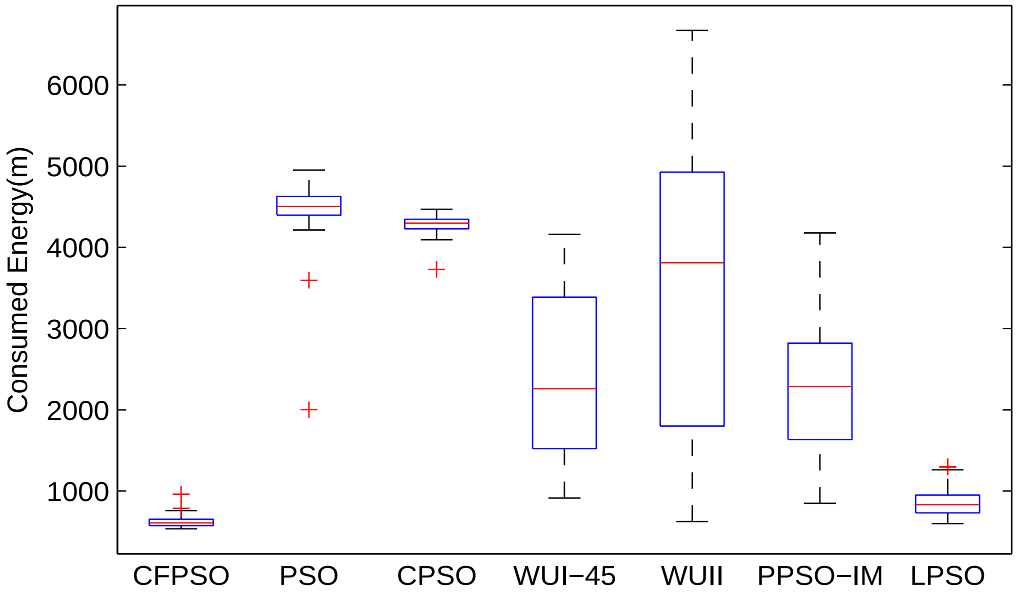


(a) Case 5

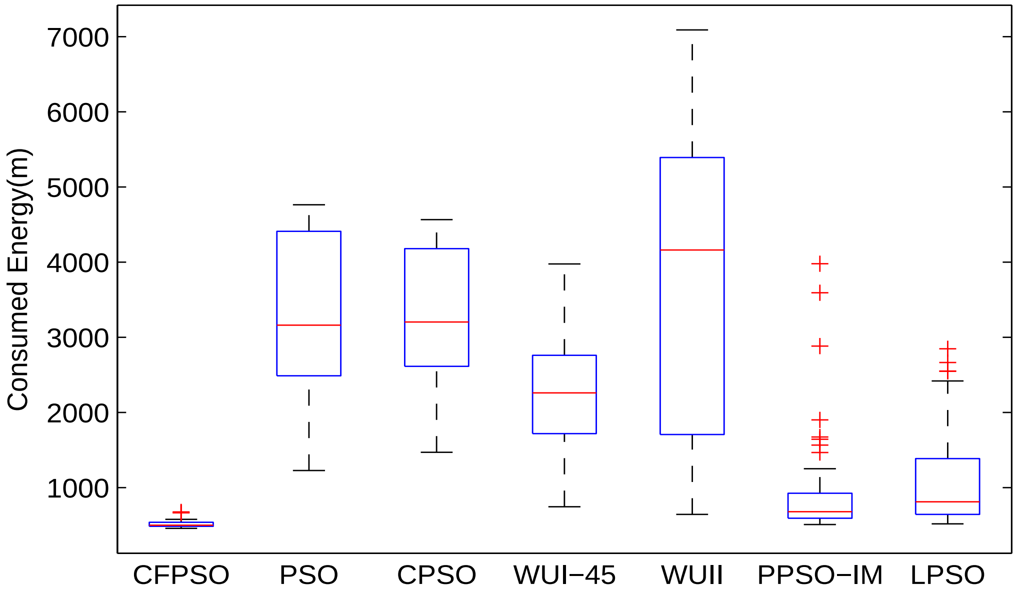


(b) Case 6

Figure 4.27: The consumed energy for case 5 and case 6 based on 5 robots with 50 runs.



(a) Case 7



(b) Case 8

Figure 4.28: The consumed energy for case 7 and case 8 based on 5 robots with 50 runs.

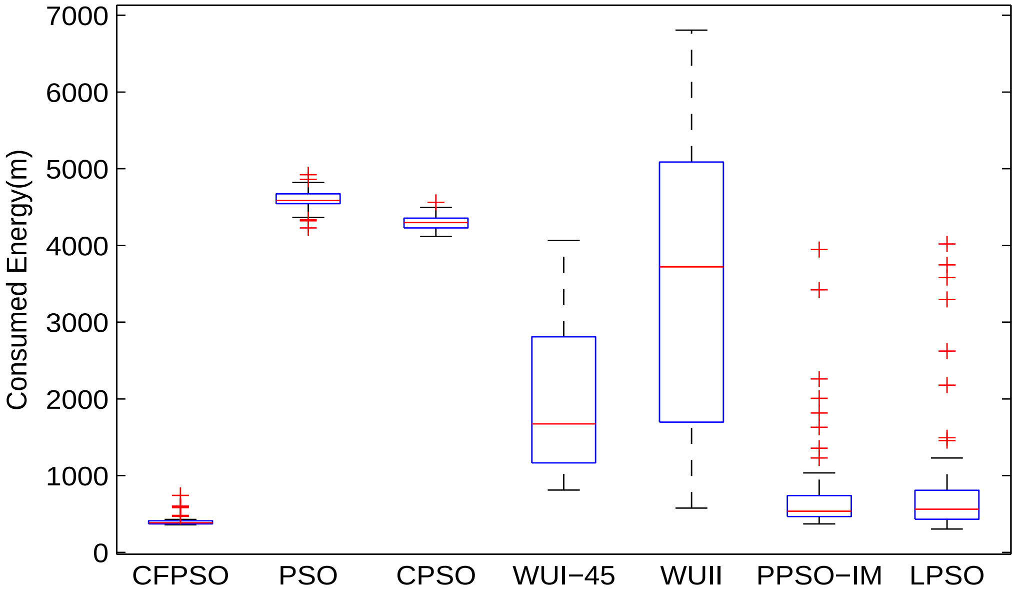


Figure 4.29: The consumed energy for case 9 based on 5 robots with 50 runs.

Chapter 5

Consensus-Based Finite-Time Motion Control

5.1 Introduction

In order to use the characteristic of the wider detection region of the multi-robot system, a consensus-based finite-time motion control algorithm, which consists of a finite-time parallel motion control algorithm and a finite-time circular motion control algorithm, is proposed in Section 5.2. Then, we test the performance capabilities of the finite-time parallel motion control algorithm and the finite-time circular motion control algorithm in Section 5.3. Next, the effectiveness of the consensus-based finite-time motion control algorithm for odour source localisation is illustrated in Section 5.4. Finally, the conclusion is presented in Section 5.5.

5.2 A Consensus-Based Finite-Time Motion Control Algorithm

In this section, we will briefly describe the background of consensus algorithm and then propose a finite-time consensus algorithm. According to the proposed algorithm, we will give a finite-time parallel motion control algorithm and a finite-time circular motion control algorithm. Finally, we will summarize the consensus-based finite-time motion control algorithm.

5.2.1 The Consensus Algorithm

Recently, there has been a surge of interest among control scientists in consensus problems [2, 143, 150, 8, 22, 74, 96, 10, 100, 151, 147, 11] due to the work of Olfati-Saber and Murray [108] (2004). On the basis of the research work [108], for consensus problems, discrete-time controllers [51, 99, 12] and continuous-time controllers [117, 118, 9] were developed for a discrete-time multi-robot system and a continuous-time multi-robot system, respectively. For the discrete-time multi-robot system, a controller classification is briefly shown in Figure 5.1 and mainly includes linear controllers [12, 99, 51, 30] and cost function based controllers (prediction control model [62, 25, 61, 54] and game theory [127, 126]) for one-order kinematics and two-order dynamics. For the continuous-time multi-robot system, a controller classification [95, 107] is briefly shown in Figure 5.2 and mainly consists of linear controllers [108, 117, 118], variable structure controllers [9], and finite-time controllers [144, 142] for one-order kinematics and two-order dynamics.

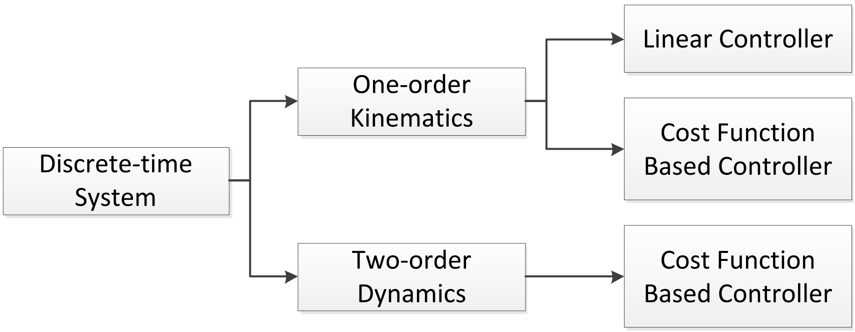


Figure 5.1: A controller classification for a discrete-time multi-robot system.

Because of the time-varying characteristic of the plume, there exist two cases for the problem of odour source localisation. One case is that the multi-robot system detects the chemical signal while the other case is that the multi-robot system loses the chemical signal. When the multi-robot system receives the chemical signal, the plume linking the odour source with the multi-robot system provides an important clue for the position of the odour source. Moving along the plume is an efficient

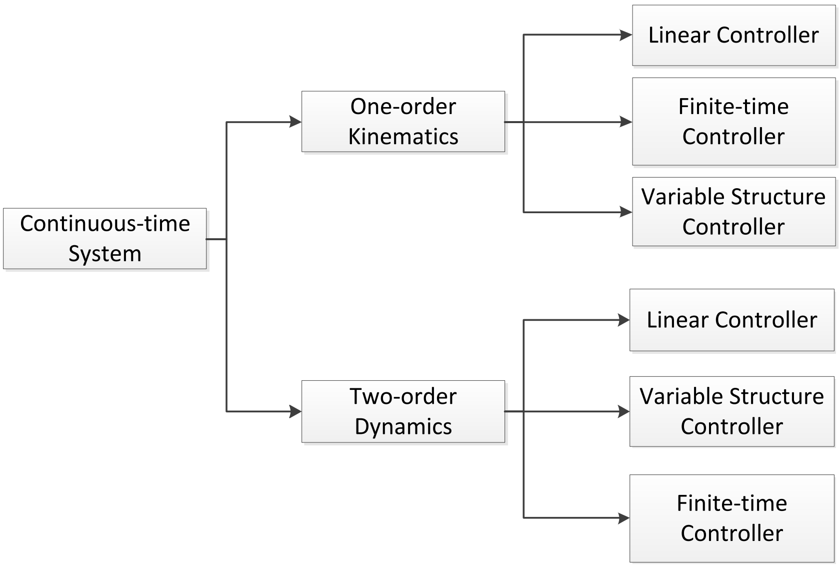


Figure 5.2: A controller classification for a continuous-time multi-robot system.

approach to locate the odour source. When the multi-robot system may not detect the chemical signal, Circling a probable position of the odour source is an efficient approach to collect odour clues. According to the aforementioned idea, the design of controllers for the single robot should enable the robot group to parallel move for tracing the plume [128, 129, 103] and to circularly move for searching for odour clues [94, 72, 104, 79, 23] such that the characteristic of the wider detection region of the multi-robot system can be efficiently used. Hence, the consensus algorithm with finite-time convergence can be used to design the parallel motion control algorithm and the circular motion control algorithm. The idea is in part from the work on the design of finite-time controllers [17, 144, 97, 136, 41] and from the study of swarming behaviors [106, 18, 32, 101].

5.2.2 The Finite-Time Consensus Algorithm

In this subsection, we provide a class of new nonlinear finite-time consensus protocol, which is the basis of the finite-time parallel motion control algorithm and the finite-time circular motion control algorithm. The proposed finite-time consensus protocol is given by

$$\begin{aligned} u_i &= \operatorname{sig}\left(\sum_{j=1}^{N} a_{ij}\left(\beta(x_j - x_i) + \gamma(v_j - v_i)\right)\right)^a \\ &\quad + \sum_{j=1}^{N} a_{ij}\left(\beta(x_j - x_i) + \gamma(v_j - v_i)\right) \end{aligned} \tag{5.1}$$

where $0 < a < 1$, $\beta > 0$, and $\gamma > 0$.

Remark 21. *It is obvious to see that if the nonlinear item is canceled, then the consensus protocol described by (5.1) becomes a linear consensus protocol, which was studied by Ren and Beard [118] (2008). If $\gamma = 0$ for a single integrator, then the consensus protocol given by (5.1) has been investigated by Xiao* et al. *[144] (2009). Hence, the linear consensus protocol described in [118] and the nonlinear consensus protocol given in [144] can be regarded as special cases of the nonlinear consensus protocol (5.1).*

The following proposition characterizes the finite-time convergence property of the nonlinear consensus protocol (5.1) as

Proposition 8. *Consider the continuous-time model (1.7). Let μ_i denote the* ith *nonzero eigenvalue of $L(A)$. If the interaction topology G is an undirected and connected graph and*

$$\gamma > \max_{i \in l_N} \sqrt{\frac{4\beta}{3\mu_i}} \tag{5.2}$$

then the consensus protocol (5.1) guarantees $x_i \to x_j$ and $v_i \to v_j$, $\forall i \in l_N$, within a finite-time interval $\left[t_0, t_0 + \frac{2V(0)^{\frac{1-a}{2}}}{k(1-a)}\right]$ where t_0 is the initial time.

Proof. Without loss of generality, we assume $t_0 = 0$ and let

$$\begin{aligned} y_i &= \sum_{j=1}^{N} a_{ij}(x_j - x_i) \\ z_i &= \sum_{j=1}^{N} a_{ij}(v_j - v_i) \end{aligned}$$

$y = [y_1, y_2, \ldots, y_N]$, and $z = [z_1, z_2, \ldots, z_N]$. Then, we have

$$\begin{aligned} \dot{y} &= z \\ \dot{z} &= -L(A)(\mathrm{sig}(\phi)^a + \phi) \\ \phi &= \beta y + \gamma z \end{aligned}$$

Choose a Lyapunov candidate as

$$V(t) = \begin{pmatrix} y^T & z^T \end{pmatrix} P \begin{pmatrix} y \\ z \end{pmatrix}$$

where

$$P = \begin{pmatrix} \frac{\beta\gamma}{2} L(A)^2 & \frac{\beta}{2} L(A) \\ \frac{\beta}{2} L(A) & \frac{\gamma}{2} L(A) \end{pmatrix}$$

Let $L(A) = \Gamma^{-1}\Theta\Gamma$ be diagonalized where $\Theta = \mathrm{diag}\{\mu_1, \mu_2, \ldots, \mu_N\}$ with μ_i being the ith eigenvalue of $L(A)$. Hence, P can be written as

$$P = \begin{pmatrix} \Gamma & 0 \\ 0 & \Gamma \end{pmatrix}^{-1} \begin{pmatrix} \frac{\beta\gamma}{2}\Theta^2 & \frac{\beta}{2}\Theta \\ \frac{\beta}{2}\Theta & \frac{\gamma}{2}\Theta \end{pmatrix} \begin{pmatrix} \Gamma & 0 \\ 0 & \Gamma \end{pmatrix}$$

If $\Omega = \begin{pmatrix} \frac{\beta\gamma}{2}\Theta^2 & \frac{\beta}{2}\Theta \\ \frac{\beta}{2}\Theta & \frac{\gamma}{2}\Theta \end{pmatrix}$ is a positive semi-definite matrix, P is also a positive semi-definite matrix. To find the eigenvalues of Ω, we can solve the equation $\det(\lambda I_{2N} - \Omega) = 0$, where $\det(\lambda I_{2N} - \Omega)$ is the characteristic polynomial of Ω. Note that

$$\begin{aligned} \det(\lambda I_{2N} - \Omega) &= \det \begin{pmatrix} \lambda I_N - \frac{\beta\gamma}{2}\Theta^2 & -\frac{\beta}{2}\Theta \\ -\frac{\beta}{2}\Theta & \lambda I_N - \frac{\gamma}{2}\Theta \end{pmatrix} \\ &= \det\Bigg[\lambda^2 I_N - \lambda(\frac{\gamma}{2}\Theta + \frac{\beta\gamma}{2}\Theta^2) \\ &\quad + \frac{\beta\gamma^2}{4}\Theta^3 - \frac{\beta^2}{4}\Theta^2\Bigg] \\ &= \prod_{i=1}^{N}\Bigg[\lambda^2 - \lambda(\frac{\gamma}{2}\mu_i + \frac{\beta\gamma}{2}\mu_i^2) \\ &\quad + \frac{\beta\gamma^2}{4}\mu_i^3 - \frac{\beta^2}{4}\mu_i^2\Bigg] \end{aligned}$$

If $\frac{\beta\gamma^2}{4}\mu_i^3 - \frac{\beta^2}{4}\mu_i^2 > 0$ and $\frac{\gamma}{2}\mu_i + \frac{\beta\gamma}{2}\mu_i^2 > 0$, we have the positive and zero eigenvalues of Ω. In terms of Lemma 2 and (5.2), we have $\Omega \geq 0$. It is straightforward to see that

$\{(y,z)|z = \mathbf{0}_N,\ y = \mathbf{1}_N d\}$ where $d \in \mathbb{R}$ is a equilibrium set. In the following, we will see that the equilibrium $(\mathbf{0}_N, \mathbf{0}_N)$ is stable while others are not stable. Obviously $V(t) \geq 0$ and along the closed-loop trajectories,

$$\begin{aligned}\frac{dV(t)}{dt} &= \beta\gamma y^T L(A)^2 z + \beta z^T L(A) z + \phi^T L(A)\dot{z} \\ &= -\zeta^T \begin{pmatrix} \beta^2 L(A)^2 & \frac{\beta\gamma}{2} L(A)^2 \\ \frac{\beta\gamma}{2} L(A)^2 & \gamma^2 L(A)^2 - \beta L(A) \end{pmatrix} \zeta \\ &\quad -\phi^T L(A)^2 \mathrm{sig}(\phi)^\alpha\end{aligned}$$

where $\zeta = \begin{pmatrix} y \\ z \end{pmatrix}$. Since $\phi^T L(A)^2 \mathrm{sig}(\phi)^a = |\phi|^T L(A)^2 |\phi|^a$, we have $\phi^T L(A)^2 \mathrm{sig}(\phi)^a \geq 0$.

Let

$$Q = \begin{pmatrix} \beta^2 L(A)^2 & \frac{\beta\gamma}{2} L(A)^2 \\ \frac{\beta\gamma}{2} L(A)^2 & \gamma^2 L(A)^2 - \beta L(A) \end{pmatrix}$$

Similarly, we have

$$Q = \begin{pmatrix} \Gamma & 0 \\ 0 & \Gamma \end{pmatrix}^{-1} \begin{pmatrix} \beta^2\Theta^2 & \frac{\beta\gamma}{2}\Theta^2 \\ \frac{\beta\gamma}{2}\Theta^2 & \gamma^2\Theta^2 - \beta\Theta \end{pmatrix} \begin{pmatrix} \Gamma & 0 \\ 0 & \Gamma \end{pmatrix}$$

The eigenvalues of Q can be calculated as

$$\begin{aligned}\det(\lambda I_{2N} - Q) &= \prod_{i=1}^{N} \Bigg[\lambda^2 + \lambda(-\gamma^2\mu_i^2 + \beta\mu_i - \beta^2\mu_i^2) \\ &\quad -\beta^2\mu_i^2(-\gamma^2\mu_i^2 + \beta\mu_i) - \frac{\beta^2\gamma^2}{4}\mu_i^4 \Bigg]\end{aligned}$$

If $-\gamma^2\mu_i^2 + \beta\mu_i - \beta^2\mu_i^2 < 0$ and $-\beta^2\mu_i^2(-\gamma^2\mu_i^2 + \beta\mu_i) - \frac{\beta^2\gamma^2}{4}\mu_i^4 > 0$, we have the positive and zero eigenvalues of Q. Since the graph G is undirected and connected, it is obvious (by Lemma 2) and (5.2) to see that Q is a positive semi-definite matrix. Therefore, we have $\frac{dV(t)}{dt} \leq 0$.

Set $\mathcal{E} = \{(y,z) \in R^{2N} | \dot{V} = 0\}$. It is straightforward to see that $(\mathbf{0}_N, \mathbf{0}_N) \in \mathcal{E}$ and $(\mathbf{1}_N d, \mathbf{0}_N) \in \mathcal{E}$ where $d \neq 0, \mathbf{1}_N = \underbrace{(1,1,\ldots,1)}_{N}^T\}$. Consider the equilibrium $(\mathbf{1}_N d, \mathbf{0}_N) \in \mathcal{E}$. Because $z = -L(A)v$ and $\mathrm{rank}(L(A)) = N-1$, $z = \mathbf{0}_N$ implies that

$v_1 = v_2 = \cdots = v_N$. Hence, we have

$$u_i = \mathrm{sig}\left(\sum_{j=1}^{N} a_{ij}\left(\beta(x_j - x_i)\right)\right)^a + \sum_{j=1}^{N} a_{ij}\left(\beta(x_j - x_i)\right)$$

Since $v_1 = v_2 = \cdots = v_N$ implies $u_i = 0$, then we have

$$\mathbf{0}_N \;\; = \;\; \mathrm{sig}(\beta y)^a + \beta y$$

that is, $y = \mathbf{0}_N$, which is a contradiction. Therefore, $M = \{(\mathbf{0}_N, \mathbf{0}_N)\}$ is the largest invariant set, and by the LaSalle Invariance principle, solutions converge to the largest invariant set M, which implies $x_1 = x_2 = \cdots = x_N$ and $v_1 = v_2 = \cdots = v_N$.

To show $(\mathbf{0}_N, \mathbf{0}_N)$ is a finite-time-stable equilibrium, we suppose $V(t) \neq 0$, and then let $\Upsilon_1 = -\frac{\frac{dV(t)}{dt}}{V(t)^{\frac{1+\alpha}{2}}}$ and $\mathcal{U} = \{\xi \in \mathbb{R}^N |\ \xi^T \xi^\hbar = 1, 0 < \hbar \leq 1$ and the nonzero terms of $\xi_1, \ldots, \xi_N$ are not the same$\}$. $\mathcal{U}$ denotes a compact set. Hence, $\xi^T L(A)^2 \xi^\hbar > 0$ for any $\xi \in \mathcal{U}$. Let $\min_{\xi\in\mathcal{U}} \xi^T L(A)^2 \xi^\hbar = k_1 > 0$. Since $\phi = (\beta I_N, \gamma I_N)\zeta$, we have $\phi^T\phi = \zeta^T \begin{pmatrix} \beta^2 I_N & \beta\gamma I_N \\ \beta\gamma I_N & \gamma^2 I_N \end{pmatrix} \zeta$. Moreover, $B = \begin{pmatrix} \beta^2 I_N & \beta\gamma I_N \\ \beta\gamma I_N & \gamma^2 I_N \end{pmatrix} \geq 0$ and $\zeta^T B \zeta$ is a continuous function with respect to ζ, which implies $||\zeta||_2^2 \leq k_2 ||\phi||_2^2 = k_2 \sum_{i=1}^N \phi_i^2$. where k_2 is a positive constant. Hence,

$$\begin{aligned}
\Upsilon_1 \;\; = \;\; & \frac{\zeta^T Q \zeta + |\phi|^T L(A)^2 |\phi|^\alpha}{V(t)^{\frac{1+\alpha}{2}}} \\
\geq \; & \frac{|\phi|^T L(A)^2 |\phi|^\alpha}{(\lambda_{max}(P)^{\frac{1+\alpha}{2}})(||\zeta||^2)^{\frac{1+\alpha}{2}}} \\
\geq \; & \frac{\frac{|\phi|^T}{\sqrt{|\phi|^T|\phi|^\alpha}} L(A)^2 \frac{|\phi|^\alpha}{\sqrt{|\phi|^T|\phi|^\alpha}} |\phi|^T |\phi|^\alpha}{(\lambda_{max}(P)^{\frac{1+\alpha}{2}})(k_2 \sum_{i=1}^N \phi_i^2)^{\frac{1+\alpha}{2}}} \\
\geq \; & \frac{k_1 \sum_{i=1}^N |\phi_i|^{\alpha+1}}{\lambda_{max}(P)^{\frac{1+\alpha}{2}} k_2^{\frac{1+\alpha}{2}} (\sum_{i=1}^N \phi_i^2)^{\frac{1+\alpha}{2}}} \\
\geq \; & \frac{k_1 (\sum_{i=1}^N \phi_i^2)^{\frac{\alpha+1}{2}}}{\lambda_{max}(P)^{\frac{1+\alpha}{2}} k_2^{\frac{1+\alpha}{2}} (\sum_{i=1}^N \phi_i^2)^{\frac{1+\alpha}{2}}} \\
= \; & \frac{k_1}{\lambda_{max}(P)^{\frac{1+\alpha}{2}} k_2^{\frac{1+\alpha}{2}}}
\end{aligned}$$

Let $k = \frac{k_1}{\lambda_{max}(P)^{\frac{1+\alpha}{2}} k_2^{\frac{1+\alpha}{2}}}$. And we have $\frac{dV(t)}{dt} \leq -kV(t)^{\frac{1+\alpha}{2}}$. By Lemma 1, the equilibrium $(\mathbf{0}_N, \mathbf{0}_N)$ is finite-time stable. □

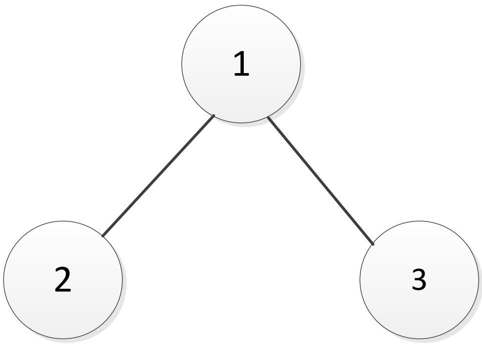


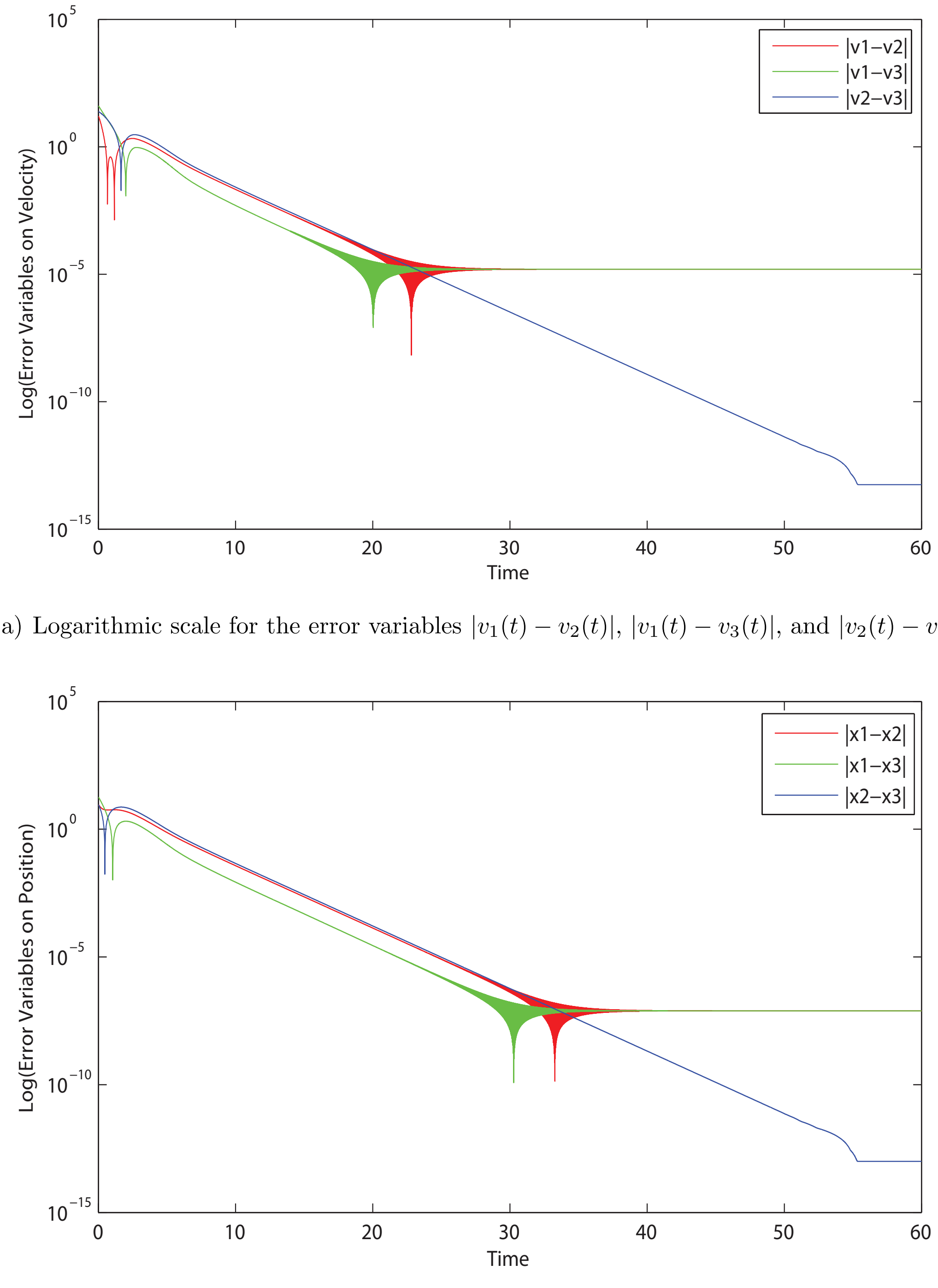
Figure 5.3: The communication topology among three robots.

Example: Let the Figure 5.3 show the communication topology among three robots. It is obvious that the communication topology is undirected and connected. In order to illustrate the finite-time convergence, we use the logarithmic scale for error variables $|x_1 - x_2|$, $|x_1 - x_3|$, $|x_2 - x_3|$, $|v_1 - v_2|$, $|v_1 - v_3|$, $|v_2 - v_3|$ and set parameters $\beta = 0.5, \alpha = 0.6$, $\gamma = 0.9$, initial positions $x_1(0) = -9$, $x_2(0) = 0$, $x_3(0) = 9$, and initial velocities $v_1(0) = 20$, $v_2(0) = 4$, $v_3(0) = -20$. Figure 5.4(a) and Figure 5.4(b) show finite-time convergence results on error variables for $v_i(t)$ and $x_i(t)$ $(i \in l_3)$.

Remark 22. *In the simulations, one can see that the system (1.7) is stable with the given control law (5.1) at* $T > 55.32$. *Therefore, the proposed finite-time consensus protocol can guarantee that the system (1.7) converges within a finite-time interval and the max settling time can be calculated based on* $\frac{2V(0)^{\frac{1-\alpha}{2}}}{k(1-\alpha)}$.

5.2.3 A Finite-Time Parallel Motion Control Algorithm

In this subsection, we will propose a parallel motion control algorithm, which can coordinate robots to form a parallel motion through the interaction with its neighbors and environment. We first give a finite-time formation algorithm that can keep a certain distance among robots. This finite-time formation algorithm is described



(a) Logarithmic scale for the error variables $|v_1(t) - v_2(t)|$, $|v_1(t) - v_3(t)|$, and $|v_2(t) - v_3(t)|$

(b) Logarithmic scale for the error variables $|x_1(t) - x_2(t)|$, $|x_1(t) - x_3(t)|$, and $|x_2(t) - x_3(t)|$

Figure 5.4: The convergence curves for the system (1.7) under a given control law (5.1).

by

$$\begin{aligned}u_i{=}\mathrm{sig}&\left(\sum_{j=1}^{N} a_{ij}\left(\beta((x_j-h_j)-(x_i-h_i))+\gamma(v_j-v_i)\right)\right)^{\alpha}\\&+\sum_{j=1}^{N} a_{ij}\left(\beta((x_j-h_j)-(x_i-h_i))+\gamma(v_j-v_i)\right)\end{aligned} \tag{5.3}$$

where $h_i (i \in l_N)$ is a constant.

Corollary 1. *Consider the continuous-time model (1.7). If the interaction topology* G *is an undirected and connected graph and* $\gamma > \max_{i \in l_N} \sqrt{\frac{4\beta}{3\mu_i}}$*, then the formation algorithm (5.3) guarantees* $x_i - h_i \to x_j - h_j$ *and* $v_i \to v_j$*,* $\forall i \in l_N$*, within a finite-time interval.*

Proof. By observing (5.3) and Proposition 8, $x_i - h_i$ and v_i $(i \in l_N)$, respectively, will reach a consensus within a finite-time interval, i.e., if $t > T$ (T is the settling time), $x_i - h_i = x_j - h_j$ and $v_i = v_j$. □

Therefore, we have the following parallel motion control algorithm.

$$\begin{aligned}u_i \;=\; & \dot{v}_c + \mathrm{sig}\left(\sum_{j=1}^{N} a_{ij}\left(\beta((x_j-h_j)-(x_i-h_i))\right.\right.\\&\left.\left.+\gamma(v_j-v_i)\right)+\lambda(v_c-v_i)\right)^{\alpha}\\&+\sum_{j=1}^{N} a_{ij}\left(\beta((x_j-h_j)-(x_i-h_i))\right.\\&\left.+\gamma(v_j-v_i)\right)+\lambda(v_c-v_i)\end{aligned} \tag{5.4}$$

where v_c is a reference velocity of the robot group and $\lambda > 0$.

Corollary 2. *Consider the continuous-time model (1.7). If the interaction topology* G *is an undirected and connected graph and* $\gamma > \max_{i \in l_N} \sqrt{\frac{4\beta}{3\mu_i}}$*, then the parallel motion control algorithm (5.4) guarantees* $x_i - h_i \to x_j - h_j$*,* $v_i \to v_j$*, and* $v_i \to v_c$*,* $\forall i \in l_N$*, within a finite-time interval.*

Proof. Let $\bar{x}_i = x_i - h_i$, $\tilde{x} = \bar{x}_i - \bar{x}_c$ where $\bar{x}_c = \int_0^t v_c(\tau)d\tau$, and $\tilde{v}_c = v_i - v_c$. By Corollary 1 and the proof process of Proposition 8, $x_i - h_i$ and v_i $(i \in l_N)$, respectively, will reach a consensus within a finite-time interval, i.e., after the settling time T, $x_i - h_i = x_j - h_j$, $v_i = v_j = v_c$. □

Example: Let the Figure 5.3 show the communication topology among three robots and set parameters $\beta = 0.5$, $\alpha = 0.6$, $\gamma = 0.9$, $\lambda = 0.1$, $h_i = 10 \times \begin{pmatrix} \cos(\frac{2\pi i}{3} + \frac{\pi}{6}) \\ \sin(\frac{2\pi i}{3} + \frac{\pi}{6}) \end{pmatrix}$ $(i = 1, 2, 3)$, and $v_c = \begin{pmatrix} 9 & -9 \end{pmatrix}^T$. The initial positions for three robots are (41.7194, -21.4161), (25.72, 25.3729), and (-11.9554,6.7822), respectively. And the initial velocities of three robots are (-8.4829, -8.9210), (0.6160,5.5833), and (8.6802,-7.4019), respectively. Figure 5.5 shows the movement of three robots controlled by the finite-time parallel motion control algorithm and Figure 5.6 illustrates the convergence of velocity. Moreover, one can see that the real settling time is $T = 46.05$ in this case.

However, it should be pointed out that each robot has its own decision about the movement direction due to limited communication links, which means that the reference velocity v_c is different for the different robot. Hence, for each robot, the parallel motion control algorithm is described by

$$\begin{aligned} u_i &= \dot{v}_{ic} + \mathrm{sig}\Bigg(\sum_{j=1}^{N} a_{ij}\Bigg(\beta((x_j - h_j) - (x_i - h_i)) \\ &\quad +\gamma(v_j - v_i)\Bigg) + \lambda(v_{ic} - v_i)\Bigg)^{\alpha} \\ &\quad +\sum_{j=1}^{N} a_{ij}\Bigg(\beta((x_j - h_j) - (x_i - h_i)) \\ &\quad +\gamma(v_j - v_i)\Bigg) + \lambda(v_{ic} - v_i) \end{aligned} \tag{5.5}$$

where v_{ic} is a velocity decided by the current position of the ith robot and the probable position of the odour source $h^{ic}(k)$, which is shown in Chapter 3.

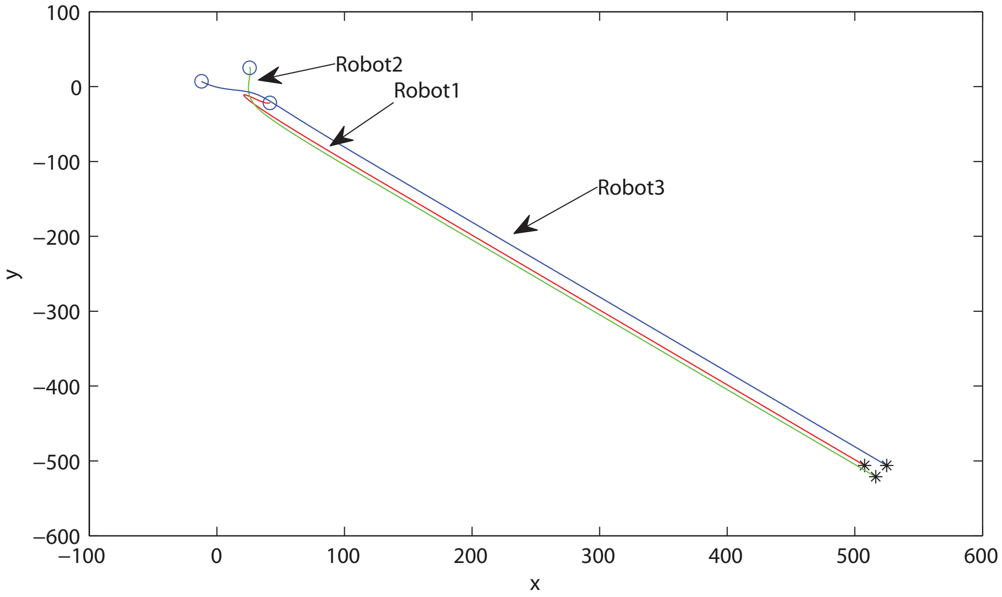


Figure 5.5: The parallel movement for three robots. "o" and "*" denote the initial position and the end position, respectively.

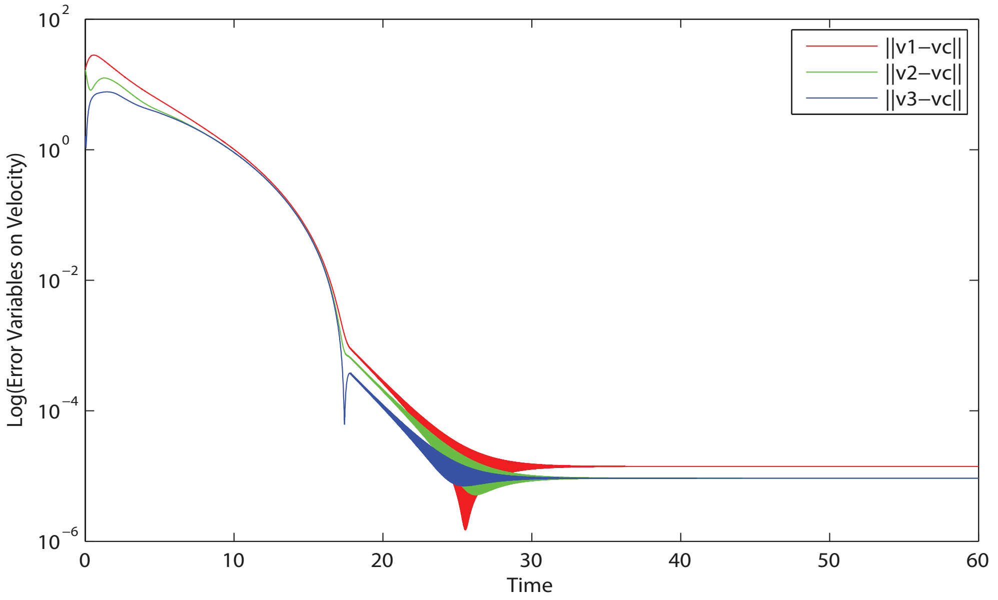


Figure 5.6: The finite-time convergence results on error variables $||v_1 - v_c||_2$, $||v_2 - v_c||_2$, and $||v_3 - v_c||_2$.

5.2.4 A Finite-Time Circular Motion Control Algorithm

In this subsection, we provide a finite-time circular motion control algorithm, which is given by

$$\begin{aligned}
u_i &= \ddot{h}_i + \mathrm{sig}\Bigg(\sum_{j=1}^{N} a_{ij}\Bigg(\beta((x_j - h_j) - (x_i - h_i)) \\
&\quad +\gamma((v_j - \dot{h}_j) - (v_i - \dot{h}_i))\Bigg) + \lambda_1(\dot{h}_i - v_i) \\
&\quad +\lambda_2(h^c - (x_i - h_i))\Bigg)^{\alpha} \\
&\quad +\sum_{j=1}^{N} a_{ij}\Bigg(\beta((x_j - h_j) - (x_i - h_i)) \\
&\quad +\gamma((v_j - \dot{h}_j) - (v_i - \dot{h}_i))\Bigg) + \lambda_1(\dot{h}_i - v_i)) \\
&\quad +\lambda_2(h^c - (x_i - h_i)) \qquad (5.6)
\end{aligned}$$

where $h_i(t) = \begin{pmatrix} r\cos(\theta_i) & r\sin(\theta_i) \end{pmatrix}^T$; $\dot{\theta}_i = \omega_0$; r denotes the radius; h^c refers to the center; ω_0 denotes the angular velocity; and $\lambda_1 > 0$, $\lambda_2 > 0$.

The following corollary can guarantee the finite-time convergence of the proposed circular motion control algorithm.

Corollary 3. *Consider the continuous-time model (1.7). If the interaction topology G is an undirected and connected graph and $\gamma > \max_{i\in l_N}\sqrt{\frac{4\beta}{3\mu_i}}$, then the circular motion control algorithm (5.6) guarantees $x_i - h_i \to x_j - h_j \to h^c$, $v_i - \dot{h}_i \to v_j - \dot{h}_j$, and $v_i \to \dot{h}_i$, where $h_i(t) = \begin{pmatrix} r\cos(\theta_i) & r\sin(\theta_i) \end{pmatrix}^T$ and $\dot{\theta}_i = \omega_0$, $\forall i \in l_N$, within a finite-time interval.*

Proof. Let $\bar{x}_i = x_i - h_i$, $\dot{\bar{x}}_i = v_i - \dot{h}_i$. Further let $\xi_i = \bar{x}_i - h^c$ and $\zeta_i = \dot{\bar{x}}_i$. By the proof process of Proposition 8, we have $\xi_i \to 0$ and $\zeta_i \to 0$, which implies that $x_i - h_i \to h^c$ and $v_i \to \dot{h}_i$, $\forall i \in l_N$ within a finite-time interval, i.e., after the settling time T, $x_i - h_i = x_j - h_j = h^c$ and $v_i = \dot{h}_i$. □

Similarly, the finite-time circular motion control algorithm for each robot is given by

$$
\begin{aligned}
u_i \;=\; & \ddot{h}_i + \mathrm{sig}\Bigg(\sum_{j=1}^{N} a_{ij} \Bigg(\beta((x_j - h_j) - (x_i - h_i)) \\
& + \gamma((v_j - \dot{h}_j) - (v_i - \dot{h}_i)) \Bigg) + \lambda_1(\dot{h}_i - v_i) \\
& + \lambda_2(h^{ic} - (x_i - h_i)) \Bigg)^{\alpha} \\
& + \sum_{j=1}^{N} a_{ij} \Bigg(\beta((x_j - h_j) - (x_i - h_i)) \\
& + \gamma((v_j - \dot{h}_j) - (v_i - \dot{h}_i)) \Bigg) + \lambda_1(\dot{h}_i - v_i)) \\
& + \lambda_2(h^{ic} - (x_i - h_i))
\end{aligned} \tag{5.7}
$$

Example: Let the Figure 5.3 show the communication topology among three robots and set parameters $\beta = 0.5$, $\alpha = 0.6$, $\gamma = 0.9$, $\lambda_1 = 0.1$, $\lambda_2 = 0.1$, $r = 10$, $\omega_0 = 2.5$, $h^c = \begin{pmatrix} 20 & 20 \end{pmatrix}^T$. Figure 5.7 shows the movement of three robots controlled by the finite-time circular motion control algorithm.

Remark 23. *For each robot, the reference velocity or the reference center are different. For the parallel motion, the proposed control algorithm (5.5) also provides a group decision process, i.e., the result of information exchange is to adjust the velocity of each robot such that $v_i \simeq v_j$ $(i, j \in l_N)$. Similarly, for the circular motion (5.7), the result of information exchange is to adjust the circular center such that $x_i - h_i \simeq x_j - h_j$ $(i, j \in l_N)$. If $\|v_{ic} - v_{jc}\|_2$ and $\|h^{ic} - h^{jc}\|_2$ are bounded, i.e. $\max_{i,j \in l_N} \|v_{ic} - v_{jc}\|_2 < \epsilon_1$ (ϵ_1 is a given parameter.) for the parallel motion and $\max_{i,j \in l_N} \|h^{ic} - h^{jc}\|_2 < \epsilon_2$ (ϵ_2 is a given parameter.) for the circular motion, the proposed motion control algorithms (5.5) and (5.7) converge.*

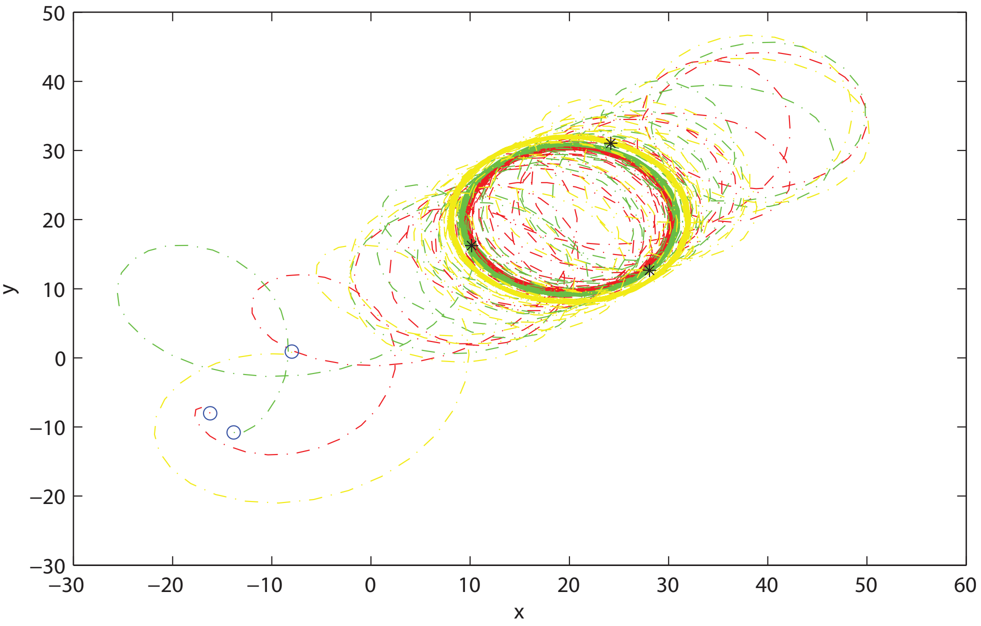


Figure 5.7: The circular movement for three robots. "o" and "*" denote the initial position and the end position, respectively.

5.2.5 Consensus-Based Finite-Time Motion Control

The consensus-based finite-time motion control algorithm includes a finite-time parallel motion control algorithm and a finite-time circular motion control algorithm. In terms of the detection event of the multi-robot system, the different motion control algorithm is used. To sum up, the consensus-based finite-time motion control algorithm is described in Algorithm 2.

5.3 Performance Capabilities of the Consensus-Based Finite-Time Motion Control

We will first make use of the communication topology shown in Figure 5.3 to test the consensus-based finite-time motion control algorithm. Let the sampling time be 0.01s and $h_i = 5 \times \begin{pmatrix} \cos(\frac{2\pi i}{3} + \frac{\pi}{6}) \\ \sin(\frac{2\pi i}{3} + \frac{\pi}{6}) \end{pmatrix}$ $(i = 1, 2, 3)$. Table 5.1 and Table 5.2 show the parameters of the consensus-based finite-time motion control algorithm.

Algorithm 2 The consensus-based finite-time motion control algorithm

```
/*Initialization*/
Initialize parameters $\beta$, $a$, $\gamma$, and $\lambda$ for the parallel motion control algorithm (5.5);
Initialize parameters $\lambda_1$, $\lambda_2$, $\gamma$, $\beta$, $\alpha$, $r$, and $\omega_0$ for the circular motion algorithm (5.7);
Run the distributed decision algorithm to obtain the reference motion direction of the robot and the position of the odour source;
/*Main Body* for the ith robot/
repeat
    IF the robot group does not detect the chemical signal and the time $T > 20s$
    Perform (1.8) and (1.9) to get the real position and velocity of the "hand position";
    Perform (5.5) to calculate the control law $u_i(t)$;
    Perform (1.10) to calculate the control input $p_i$;
    Perform (1.11) and (1.12) to obtain the applied torques for the left and right wheels;
    ELSE
    Perform (1.8) and (1.9) to get the real position and velocity of the "hand position";
    Perform (5.7) to calculate the control law $u_i(t)$;
    Perform (1.10) to calculate the control input $p_i$;
    Perform (1.11) and (1.12) to obtain the applied torques for the left and right wheels;
until Termination conditions are satisfied.
```

Table 5.1: The parameters of controllers (5.5).

γ	λ	β	α	v_c(m/s)
1.2	2	0.5	0.4	(0.4,-0.2)

Table 5.2: The parameters of controllers (5.7).

γ	λ_1	λ_2	β	α	r_1(m)	r_2(m)	r_3(m)	ω_0(radian/s)	h^c(m,m)
1.2	3.2	3.2	0.5	0.4	3	8	12	0.1	(0,0)

Moreover, we record the data at every 0.5s. Figures 5.8-5.12 show the results of parallel movement whereas Figures 5.13-5.17 show the results of circular movement. In order to further test the proposed consensus-based finite-time motion control algorithms, we use the communication topology shown in Figure 5.18. The results for parallel movement and circular movement are shown in Figures 5.19-5.26.

5.4 Odour Source Localisation

In this section, we will use the finite-time parallel motion control law (5.5) and the finite-time circular motion control law (5.7) to control a group of robots to trace the plume and search for the odour clues, respectively. The simulation consists of

Table 5.3: The parameters used in (5.5) for the parallel motion control for odour source localisation.

β	γ	α	λ	δ	h_i
0.5	2	0.6	0.1	0.3	$(5\cos(2\pi i/5+\pi/6), 5\sin(2\pi i/5+\pi/6))$

Table 5.4: The parameters used in (5.7) for the circular motion control for odour source localisation.

β	γ	α	λ_1	λ_2	δ	ω_0	r	θ_i	v_{max}(m/s)	ω_{max}(rad/s)
0.5	2	0.6	0.1	0.1	0.3	0.1	10	$2\pi i/5$	0.8	1.57

two parts. In the first part, we will show the motion process of the multi-robot system. In the second part, we will compare the search performance of the multi-robot system with that of the multi-robot system coordinated by several selected algorithms. The parameters used in this section can be found in Table 5.3 and Table 5.4. In addition, the maximum search time is 1500s and the radius is 1m.

The prediction errors of five robots about the position of the odour source are shown in Figure 5.27 where the robots keep the predefined position (80m,0m) of the odour source from 0s to about 50s. After about 50s, the robot group detects the odour clues, and then the proposed motion control algorithms start to run. Correspondingly, the motion process of the robot group is illustrated in Figures 5.28-5.29. In Figure 5.28(a), the initial positions of the robot group are set at the right-up corner in the search region. In Figures 5.28(b) and 5.29(a), the parallel motion is used to track the plume and to move along the plume in the light of the probable positions of the odour source. In Figure 5.29(b), the real odour source is found.

Then, we will compare the search efficiency of the multi-robot system coordinated by the decision-control systems (DCS) which include a distributed decision algorithm and the consensus-based finite-time motion control algorithm with several selected algorithms which include PSO [93], PPSO-IM [85], LPSO [87], CPSO [53], WUI-

Table 5.5: The success rates (%) based on 50 runs.

Algorithms	3 robots	5 robots	7 robots	9 robots
PSO [93]	36	54	70	68
PPSO-IM [85]	92	100	100	100
LPSO [87]	100	100	100	100
CPSO [53]	42	46	70	76
WUI-45 [53]	40	70	90	80
WUII [53]	58	86	98	96
CCS(L=2) [84]	96	100	100	100
DCS	100	100	100	100

45 [53], WUII [53], and CCS(L=2) [84]. For all algorithms, the robot group will search for the odour clues along the direction of y axis from the initial positions (right-up corner) to the (80m,0m). Once the odour clues are detected by any robot, these algorithms will start to run. In Table 5.5, the success rates obtained by eight algorithms are shown. Figure 5.30 shows less search time obtained from DCS. From Figure 5.31, one can see that robots coordinated by DCS consume the lesser energy than those coordinated by other compared algorithms.

5.5 Conclusion

We have developed the consensus-based finite-time motion control algorithm to control the robot group to trace and search for the plume. Specifically, we have first proposed a nonlinear finite-time consensus algorithm and used a Lyapunov approach to analyze the finite-time convergence of the proposed consensus algorithm. Then, in terms of the proposed finite-time consensus algorithm, we have derived a finite-time parallel motion algorithm and a finite-time circular motion algorithm. Next, we have summarized the consensus-based finite-time motion control algorithm. Finally, the effectiveness of the consensus-based finite-time motion control algorithm is illustrated for the odour source localisation problem.

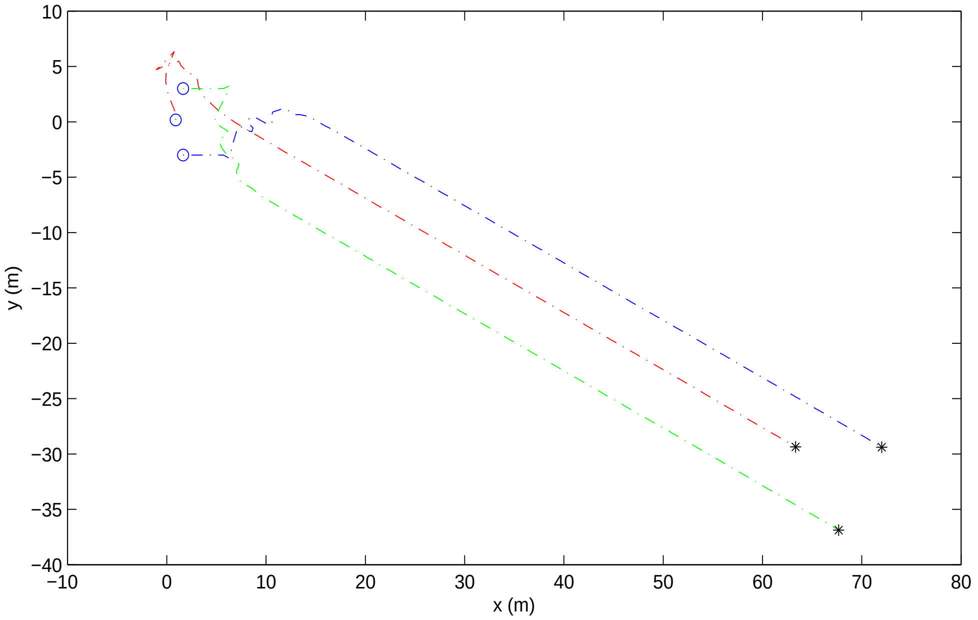


Figure 5.8: The parallel movement with communication topology shown in Figure 5.3.

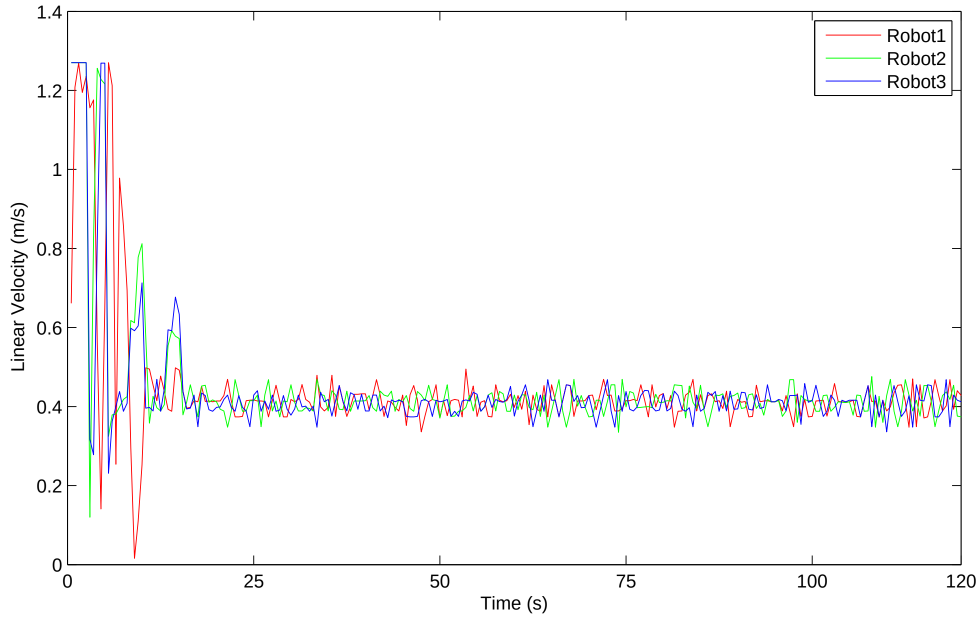


Figure 5.9: The change curve of linear velocity for parallel movement with communication topology shown in Figure 5.3.

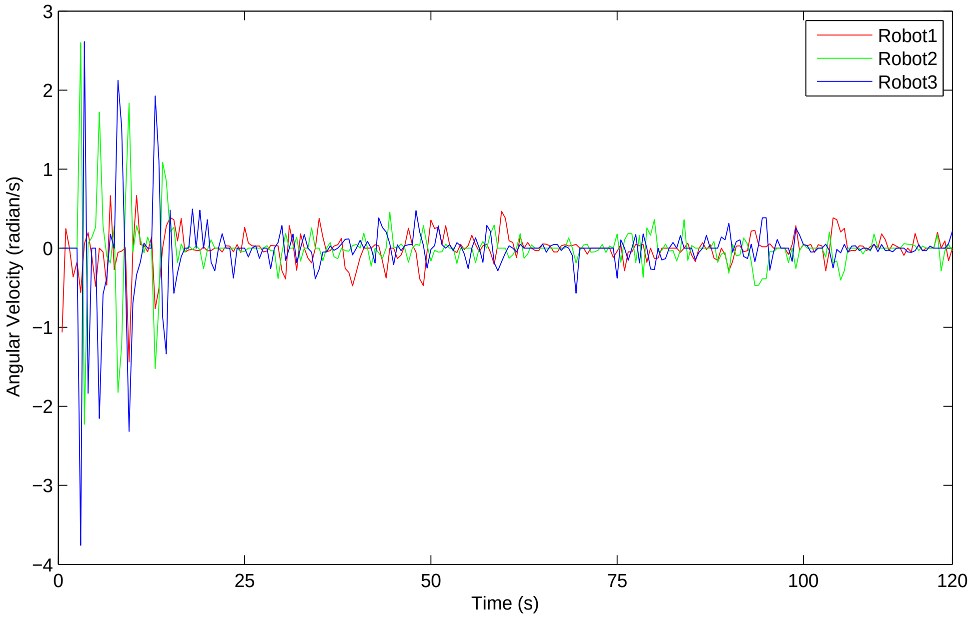


Figure 5.10: The change curve of angular velocity for parallel movement with communication topology shown in Figure 5.3.

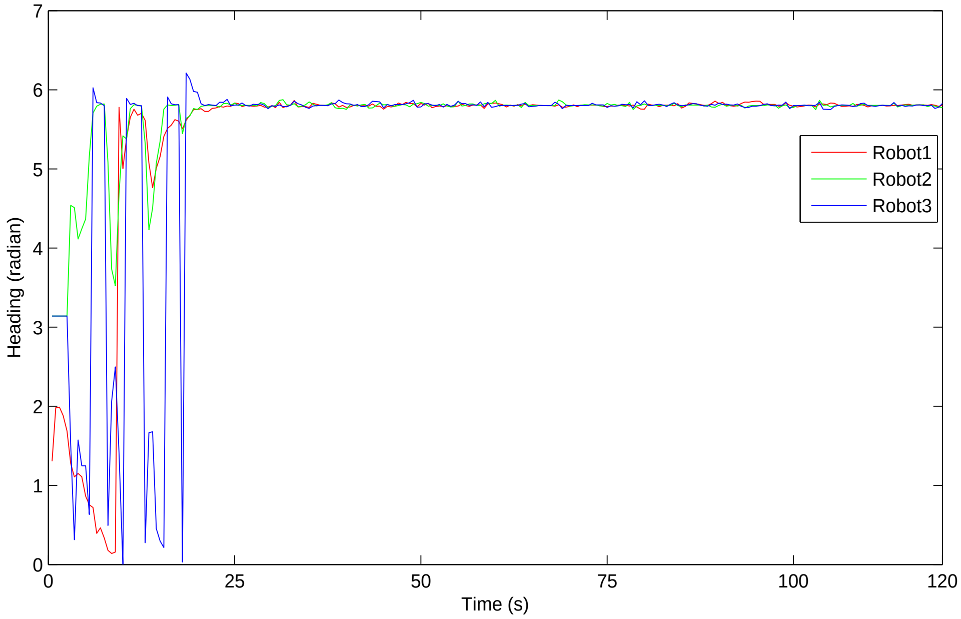
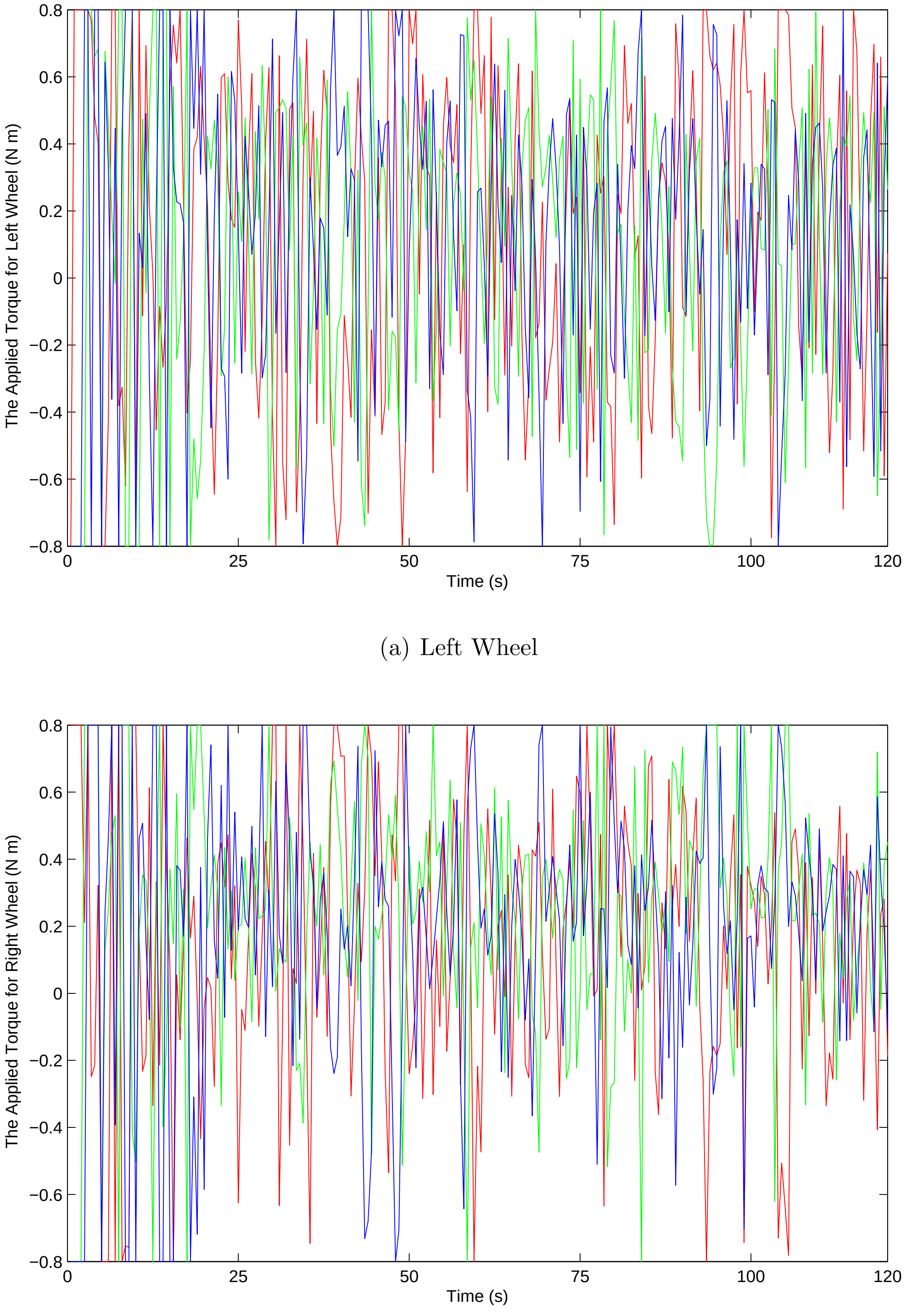


Figure 5.11: The change curve of the heading for parallel movement with communication topology shown in Figure 5.3.



(a) Left Wheel

(b) Right Wheel

Figure 5.12: The change curves of the applied torques for left wheel and right wheel for the parallel movement with communication topology shown in Figure 5.3.

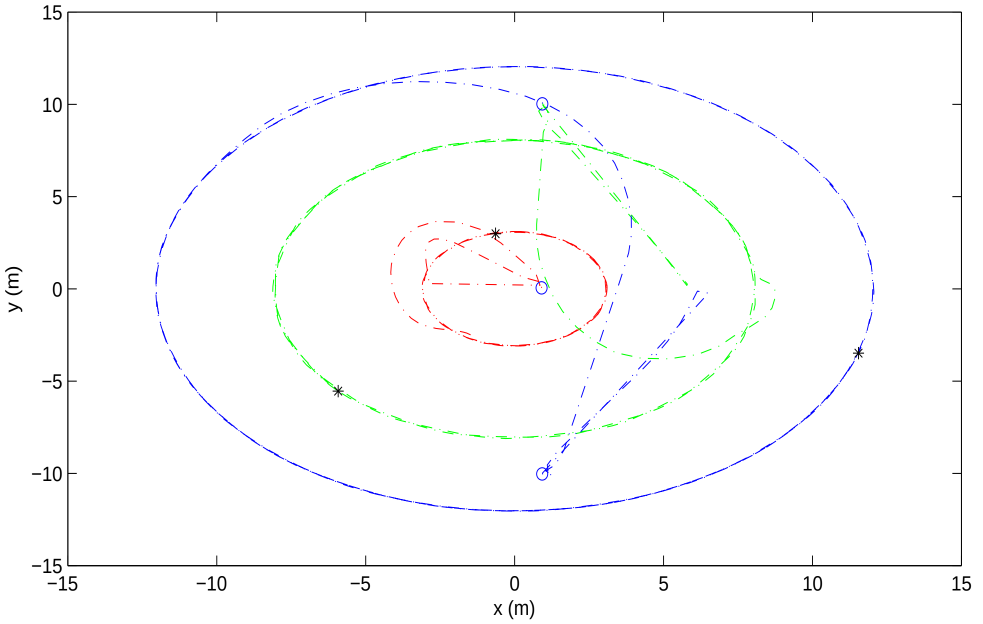


Figure 5.13: The circular movement with communication topology shown in Figure 5.3.

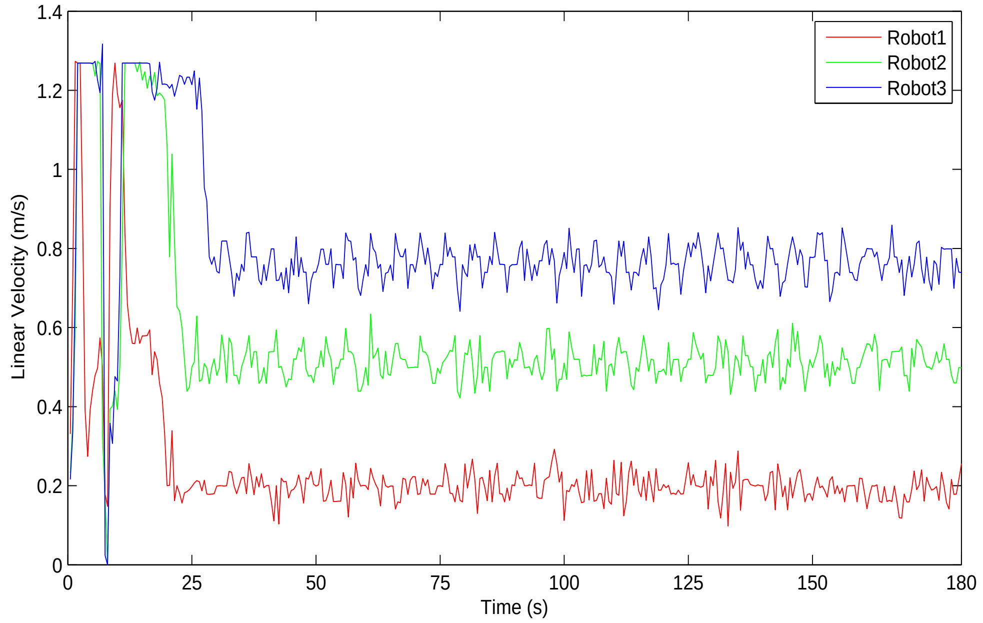


Figure 5.14: The change curve of linear velocity of circular movement with communication topology shown in Figure 5.3.

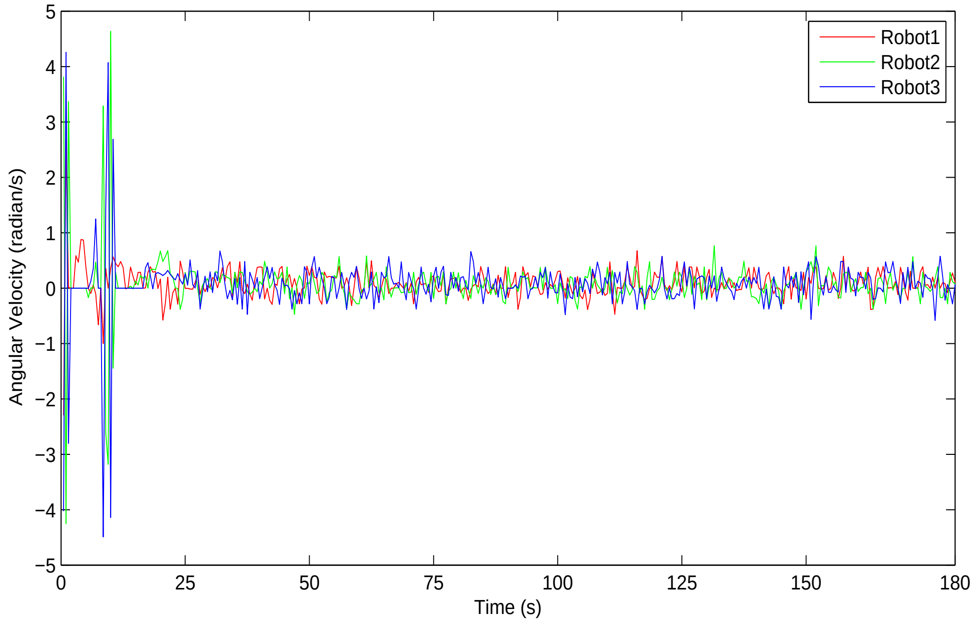


Figure 5.15: The change curve of angular velocity for circular movement with communication topology shown in Figure 5.3.

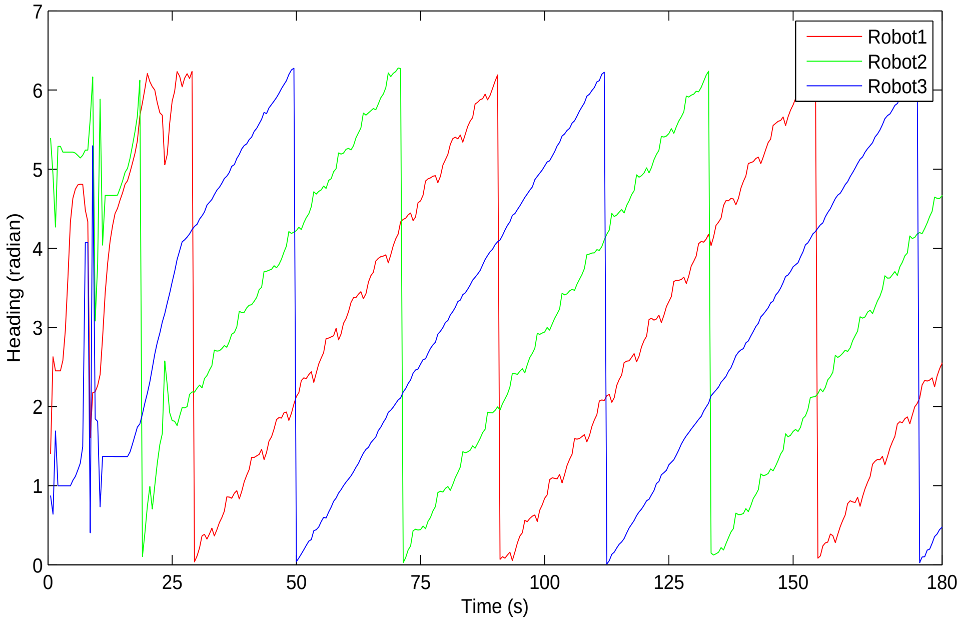
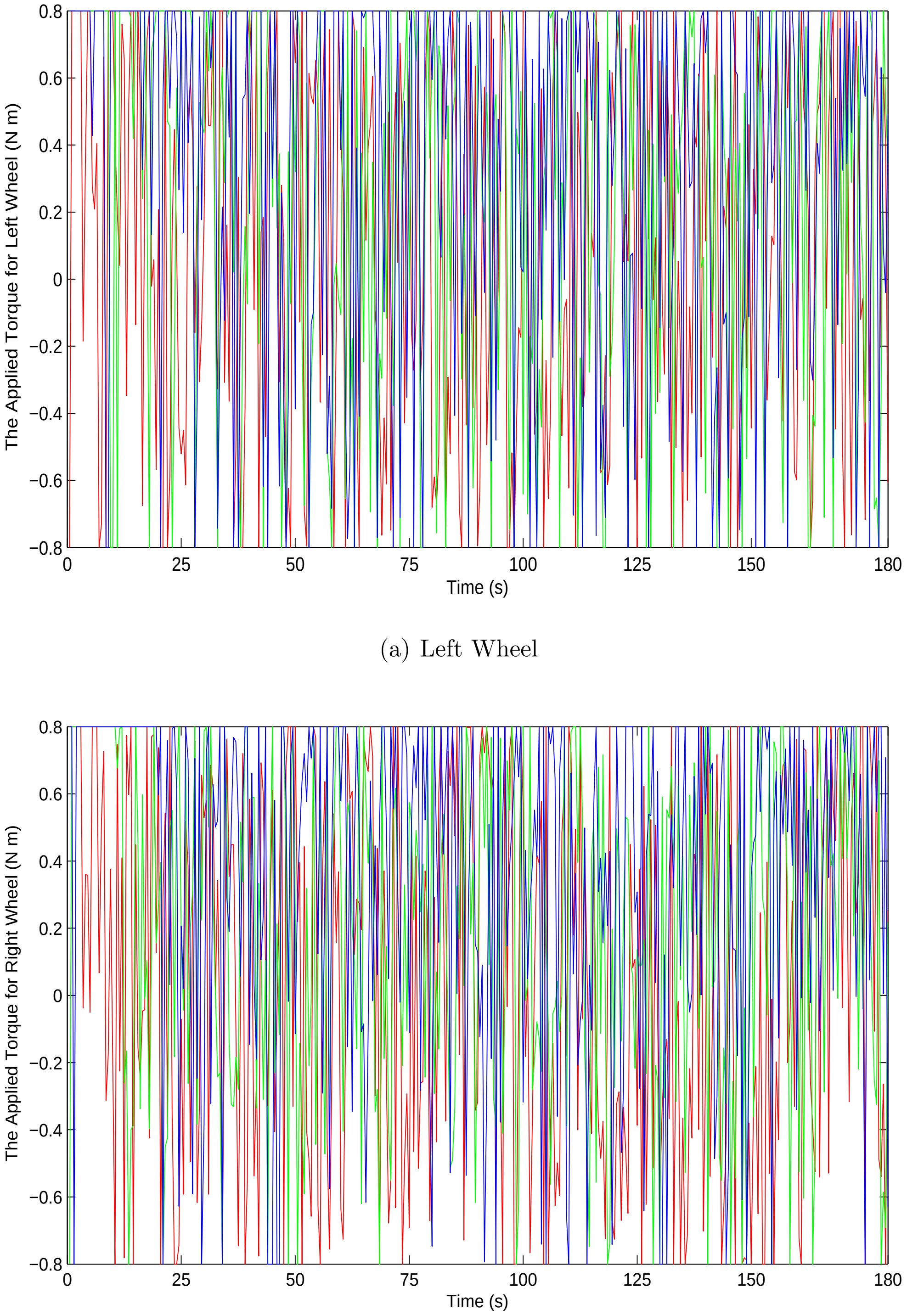


Figure 5.16: The change curve of the heading for circular movement with communication topology shown in Figure 5.3.



(a) Left Wheel

(b) Right Wheel

Figure 5.17: The change curves of the applied torques for left wheel and right wheel for circular movement with communication topology shown in Figure 5.3.

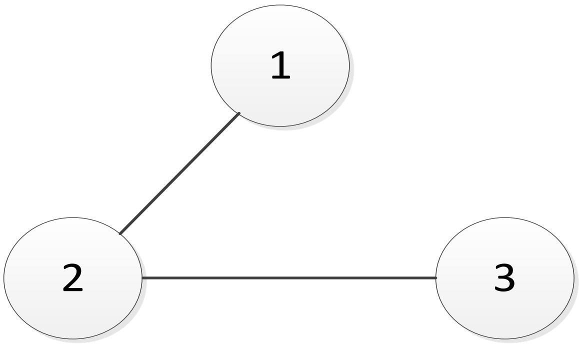


Figure 5.18: The communication topology among three robots.

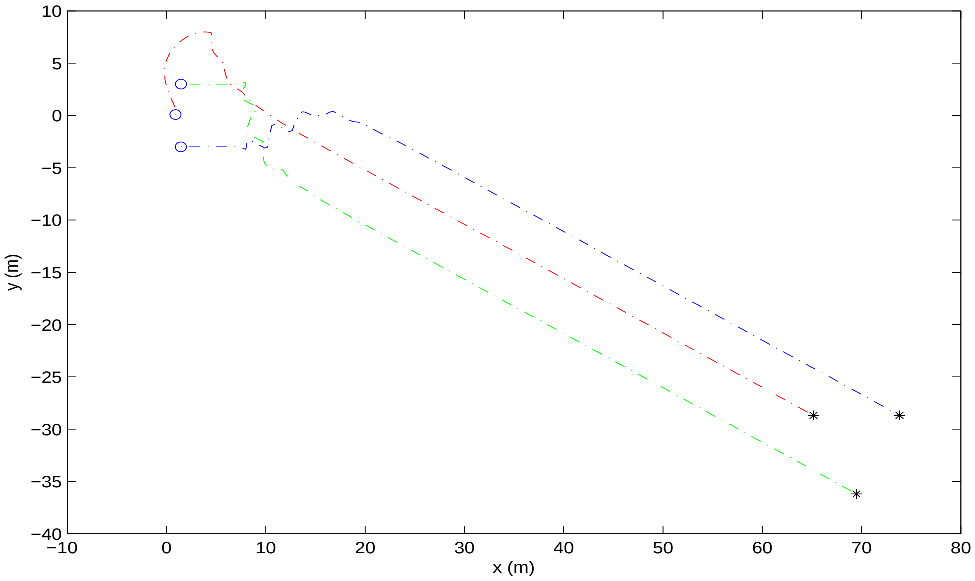


Figure 5.19: The parallel movement with communication topology shown in Figure 5.18.

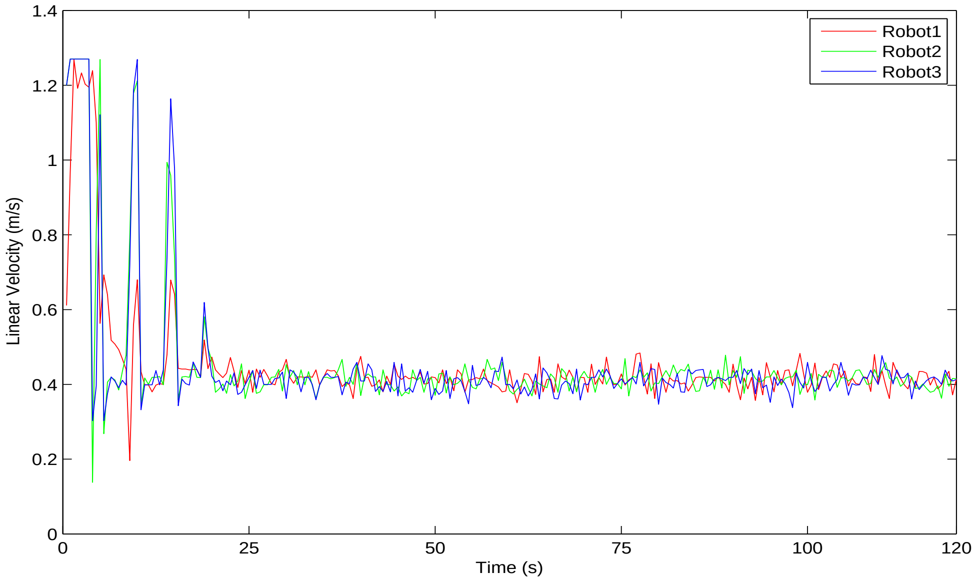


Figure 5.20: The change curve of linear velocity for parallel movement with communication topology shown in Figure 5.18.

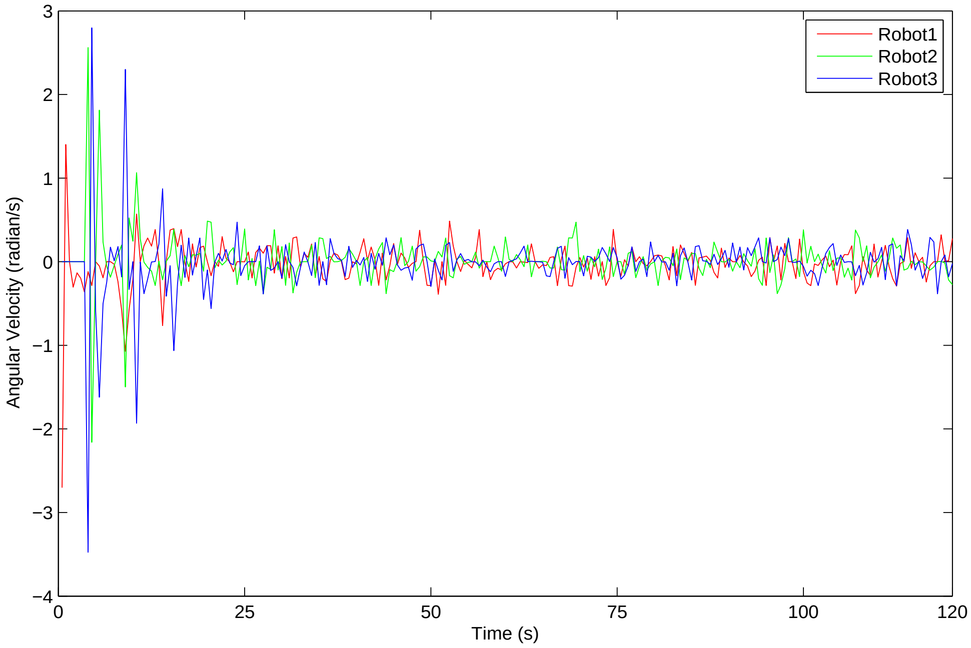


Figure 5.21: The change curve of angular velocity for parallel movement with communication topology shown in Figure 5.18.

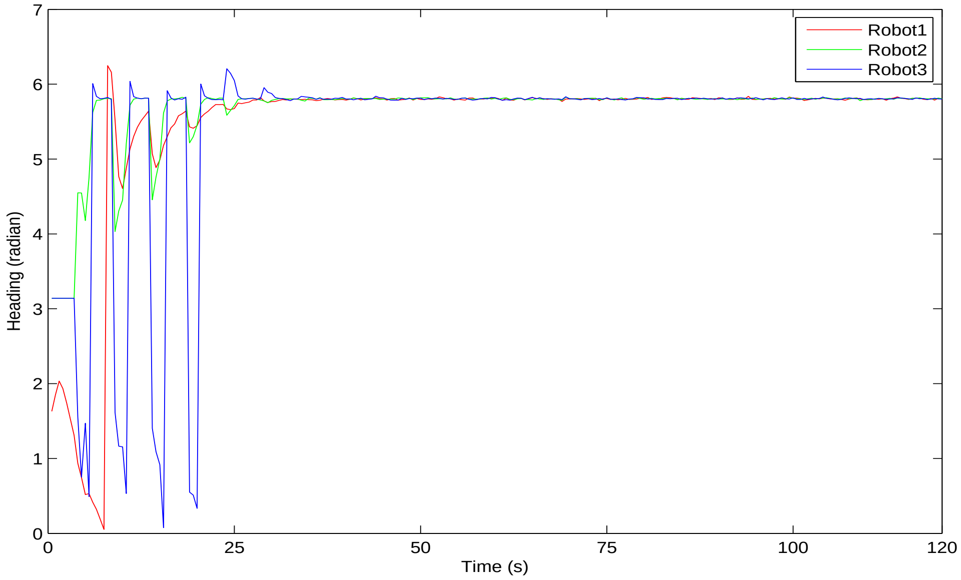


Figure 5.22: The change curve of the heading for parallel movement with communication topology shown in Figure 5.18.

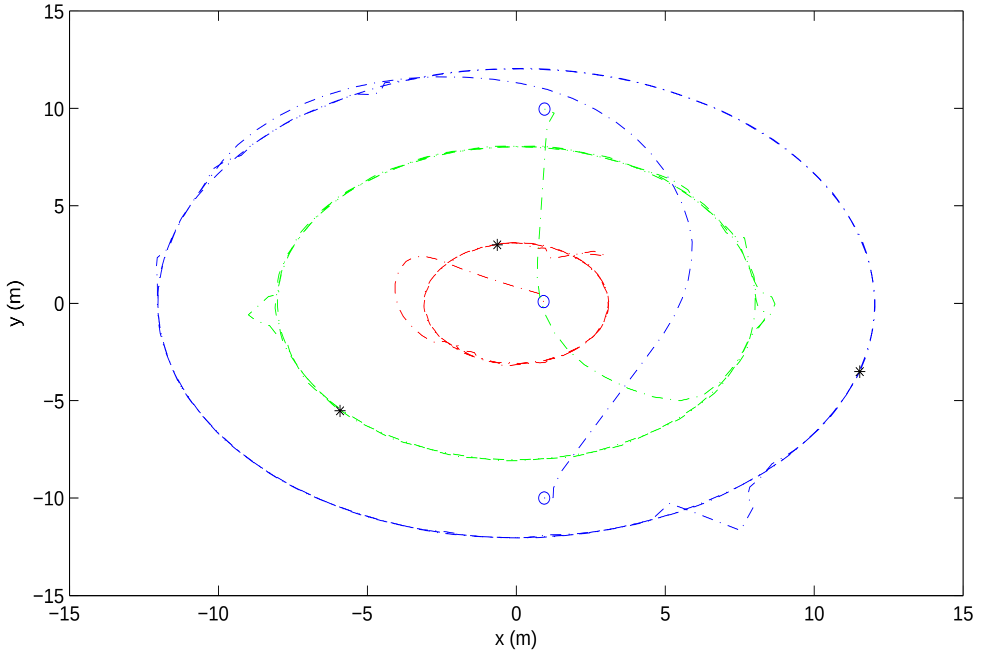


Figure 5.23: The circular movement with communication topology shown in Figure 5.18.

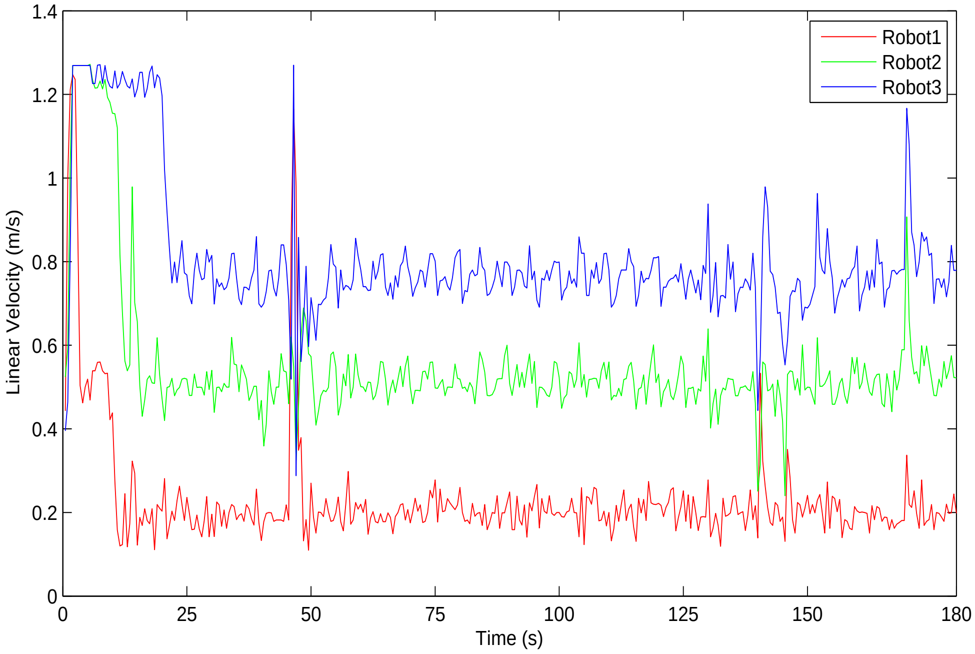


Figure 5.24: The circular movement with communication topology shown in Figure 5.18.

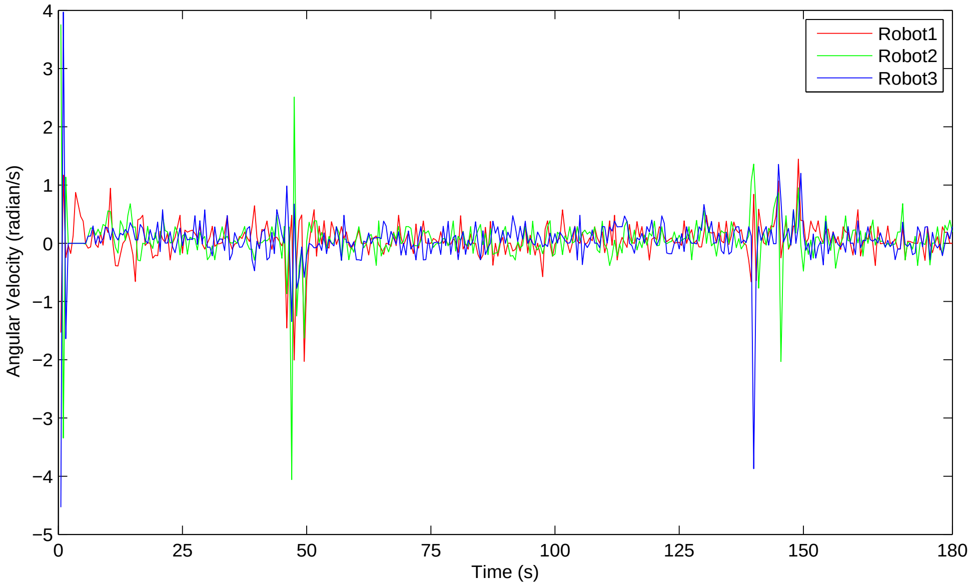


Figure 5.25: The change curve of angular velocity for circular movement with communication topology shown in Figure 5.18.

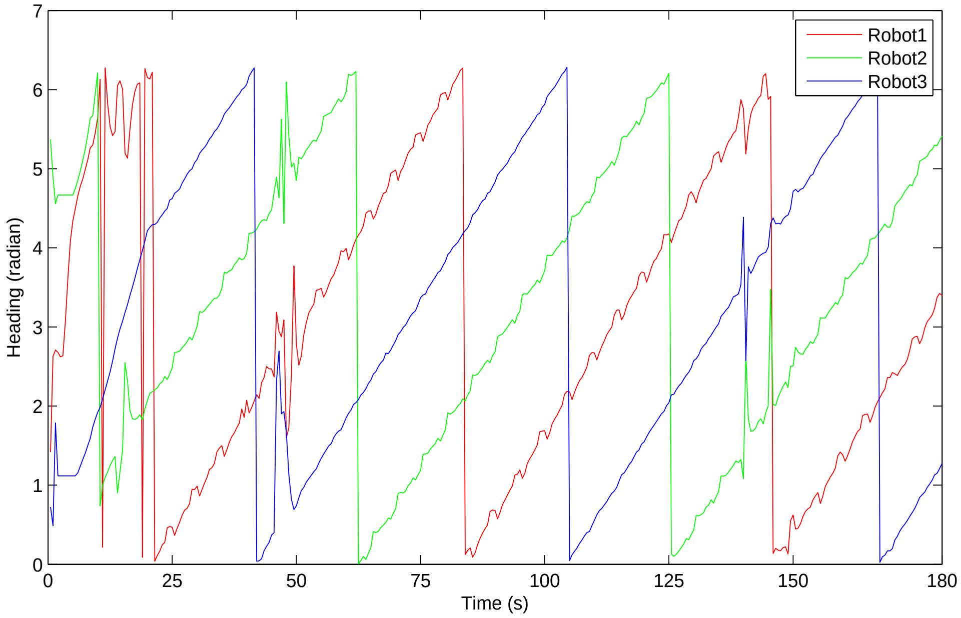


Figure 5.26: The change curve of the heading for circular movement with communication topology shown in Figure 5.18.

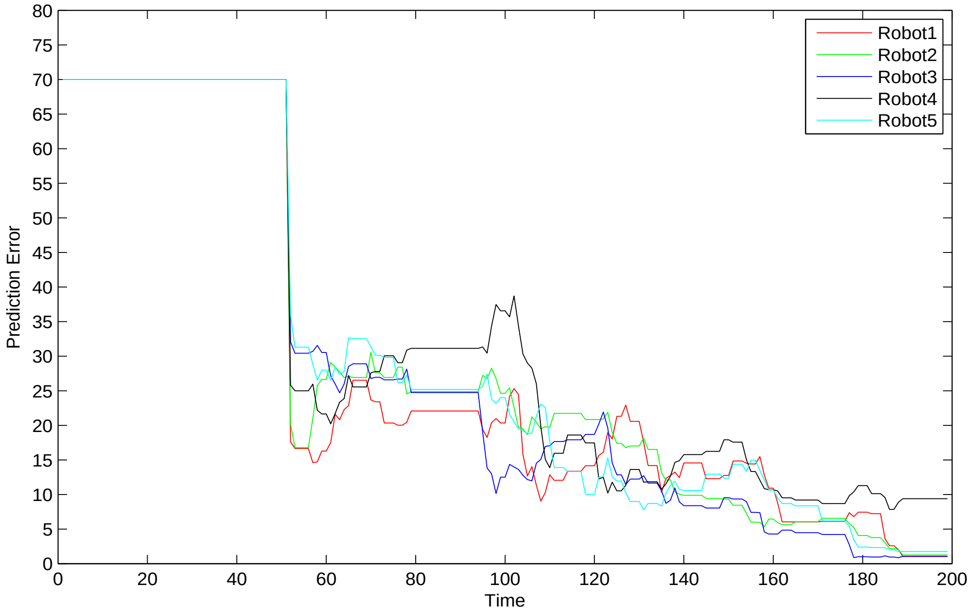
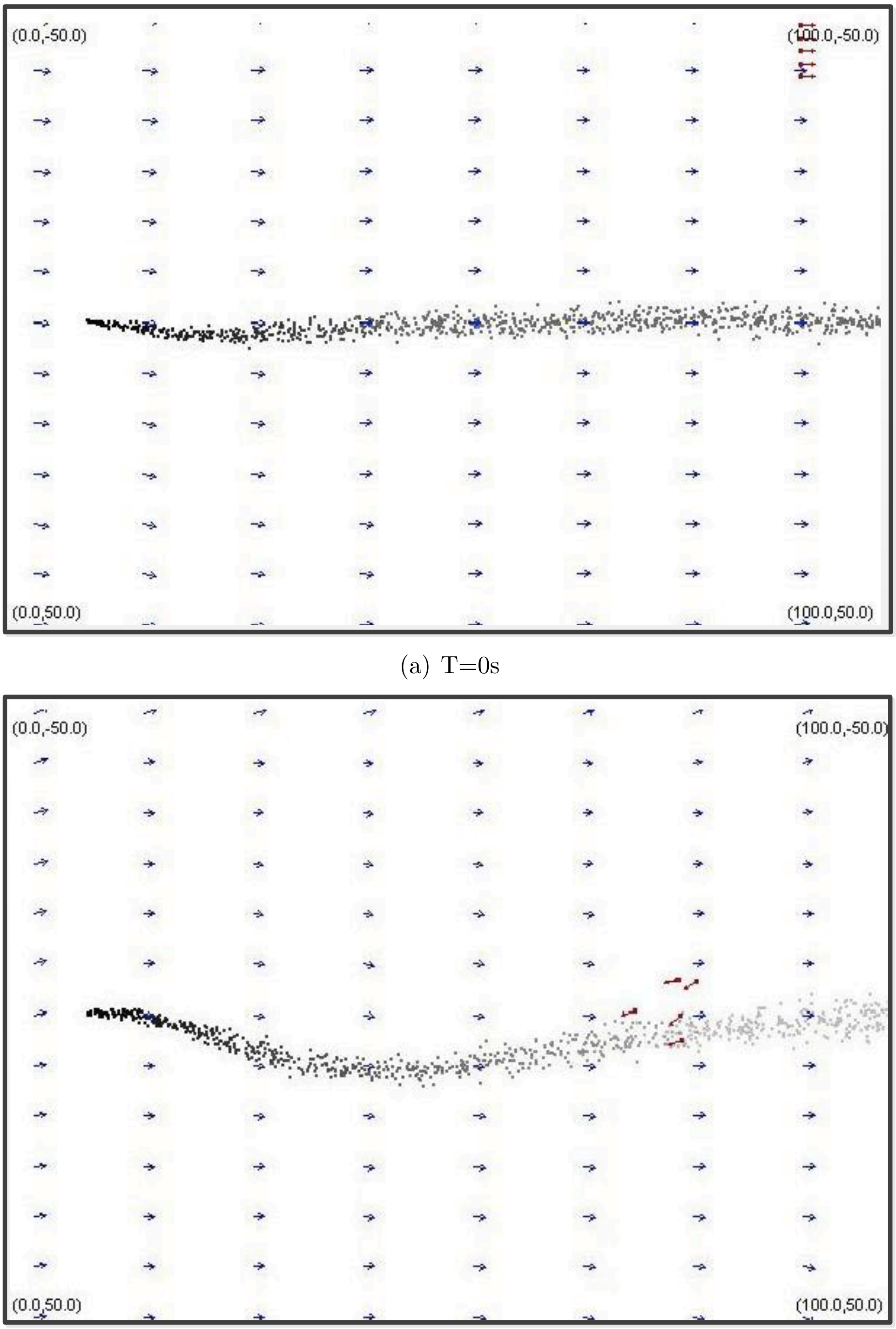


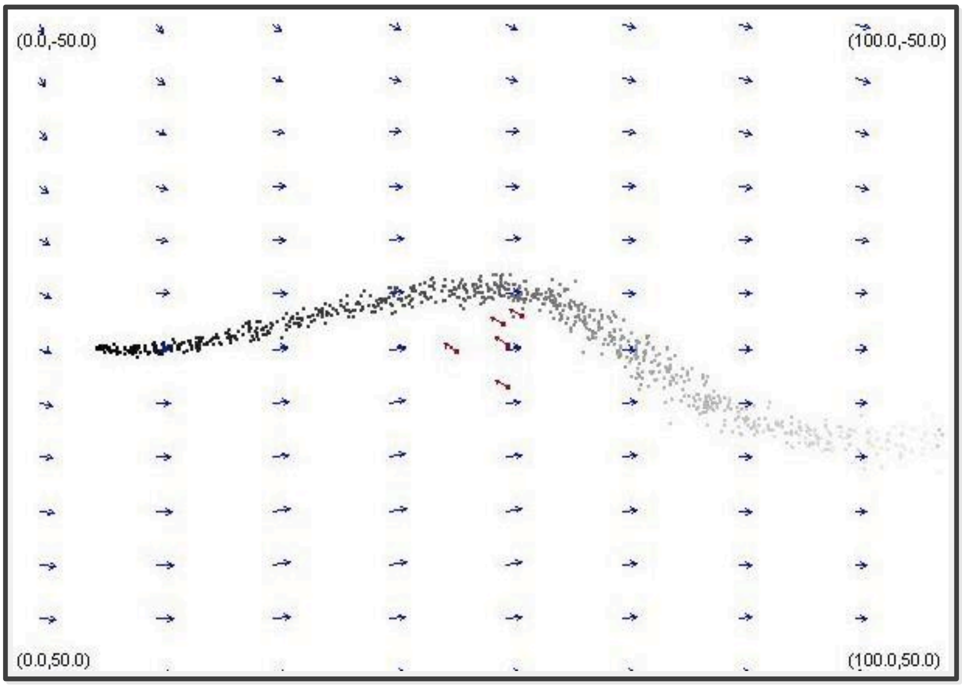
Figure 5.27: Prediction error $\|h^{ic}(k) - x_s\|_2$ over time. x_s is a real position of the odour source.



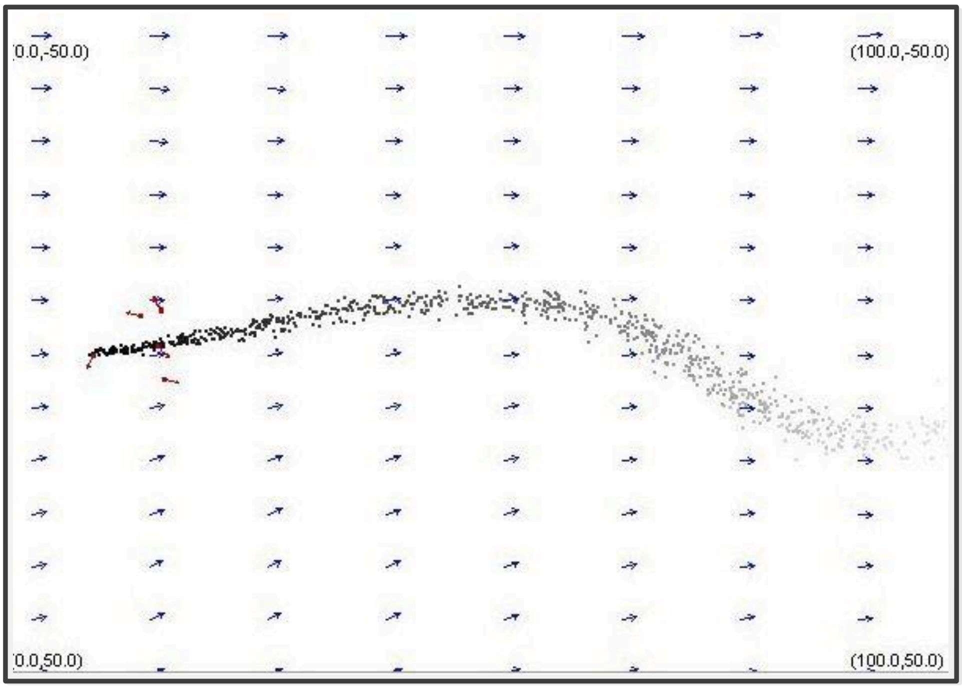
(a) T=0s

(b) T=66s

Figure 5.28: The search process of five robots at T=0s and T=53s.



(a) T=115s



(b) T=203s

Figure 5.29: The search process of five robots at T=103s and T=146s.

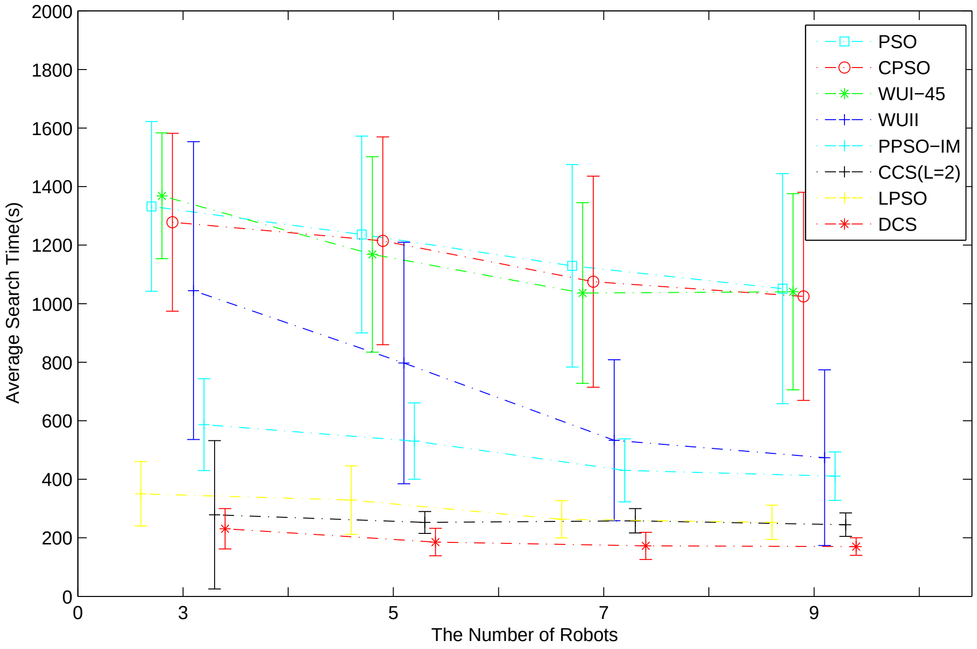


Figure 5.30: The average search time based on 50 runs.

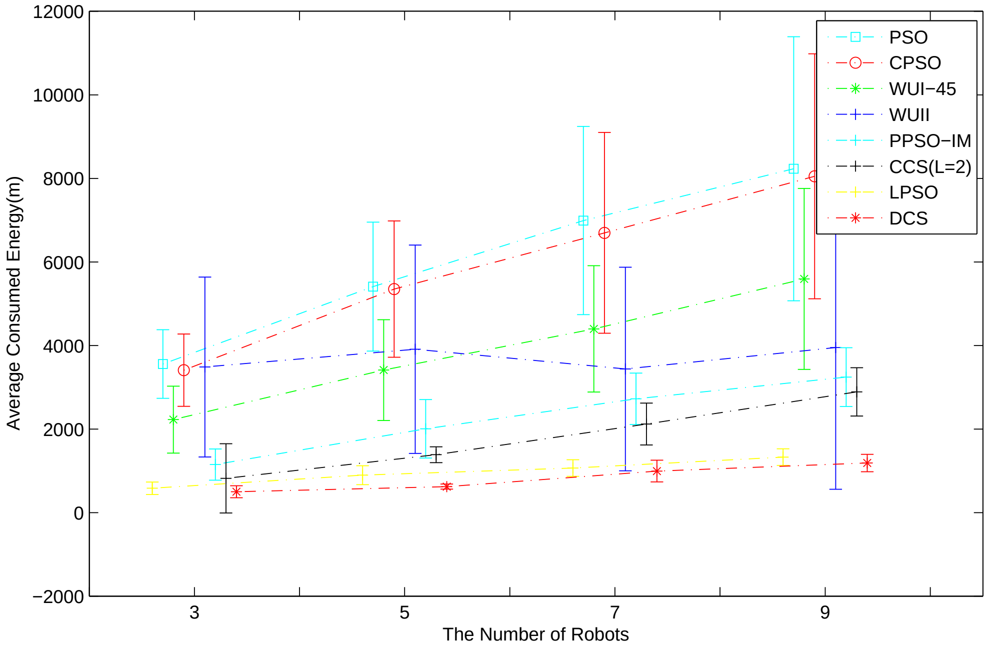


Figure 5.31: The average consumed energy based on 50 runs.

Chapter 6

Potential-Based Finite-Time Motion Control

6.1 Introduction

In order to introduce the obstacle avoidance function into cooperative control laws, a potential-based finite-time motion control algorithm, which consists of a finite-time parallel motion control algorithm and a finite-time circular motion control algorithm, is designed in Section 6.2. In Section 6.3, the performance capabilities of the finite-time parallel motion control algorithm and the finite-time circular motion control algorithm are tested. Moreover, the effectiveness of the potential-based finite-time motion control algorithm for odour source localisation is illustrated in Section 6.4. Finally, the conclusion is given in Section 6.5.

6.2 Potential-Based Finite-Time Motion Control

In this section, we will introduce the potential function that can be used for obstacle avoidance, the coordination control algorithm that can enable the virtual formation and the velocity consensus to be reached, and the tracking control algorithm that can enable the real formation and velocity to track the virtual formation and velocity. In terms of the coordination control algorithm and the tracking control algorithm, we will give the finite-time parallel motion control algorithm and the finite-time circular motion control algorithm. Finally, we will summarize the potential-based finite-time motion control algorithm and give stability analysis about the coordination control algorithm and the tracking control algorithm.

6.2.1 The Potential Function

The potential functions have been widely used for the robot navigation and control [3]. Various functions have been proposed for different aims [116, 67, 40, 21]. In this chapter, we consider the potential function in which it does not include any environmental terms. In particular, consider the following potential function form as

$$J = f(\|x_i - x_j\|_2)$$

where $f : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ denotes a map; $\|x_i - x_j\|_2$ refers to the distance between the ith robot and the jth robot; and $\|\cdot\|_2$ denotes the Euclidian norm. Moreover, the potential function has the following properties.

- Set $x_i = \xi_i - x_0$ where x_0 is a constant vector. The potential function J satisfies $f(\|x_i - x_j\|_2) = f(\|\xi_i - \xi_j\|_2)$ and
$$\frac{\partial J}{\partial x_i} = \frac{\partial J}{\partial \xi_i}$$
- The potential function J has the zero minimum at $\|x_i - x_j\|_2 = d$ where d is a predefined parameter.
- There exists a constant c such that the potential function satisfies
$$J \le c \sum_{i=1}^{N} \left\| \frac{\partial J}{\partial x_i} \right\|_2^2$$

It is obvious that we can always build a potential function which satisfies the aforementioned properties. For example, the potential function proposed by Olfati-Saber [106] (2006) satisfies the above properties.

6.2.2 Coordinating Control

In this subsection, we provide the following finite-time coordination control algorithm (6.1), which can enable the virtual velocity consensus and the accurate formation shaped by the potential function to be reached within a finite-time interval.

$$\dot{\hat{x}}_i(t) = \hat{v}_i(t) - \beta \left(\frac{\partial J}{\partial \hat{x}_i} \right)^{\frac{2a-1}{2a+1}} \tag{6.1a}$$

$$\dot{\hat{v}}_i(t) = \gamma \left(\sum_{j=1}^{N} a_{ij}(\hat{v}_j - \hat{v}_i) \right)^{\frac{2a-1}{2a+1}} \tag{6.1b}$$

where $a > 0.5$, $\beta > 0$, $\gamma > 0$; N is the number of robots; $\hat{x}_i(t)$ and $\hat{v}_i(t)$ denote the virtual position and velocity of the ith robot, respectively; J is a potential function which satisfies the aforementioned properties; and a_{ij} refers to the communication link between the ith robot and the jth robot. If there exists a communication link between the ith robot and the jth robot, $a_{ij} = 1$ and $a_{ij} = 0$ otherwise.

It is obvious to see that in (6.1), the velocity and position are decoupled, which means that the virtual velocity consensus of the robot group can be first reached, and then the virtual position reaches the one decided by the potential function. Hence, in the stable state, virtual dynamics in (6.1) equals to the real dynamics in (1.7). Moreover, the proposed finite-time coordination control algorithm is continuous, smooth, and easily to be programmed compared with other finite-time controllers [63, 142, 73, 9, 47, 141] because of no“sign” function.

The objective of the finite-time coordination control algorithm (6.1) is to guarantee that the potential J is minimized and $\hat{v}_i \to \hat{v}_j$, $\forall j$ within a finite-time interval. The following Example 1 will illustrate the objective of the finite-time coordination control algorithm.

Example 1: Consider six robots and use the potential function given in [106]. For the potential function, the parameters are $d = 7$, $r = 1.2d$, $a = b = 5$, $\epsilon = 0.1$, and $h = 0.2$. For the finite-time coordination control algorithm (6.1), the parameters

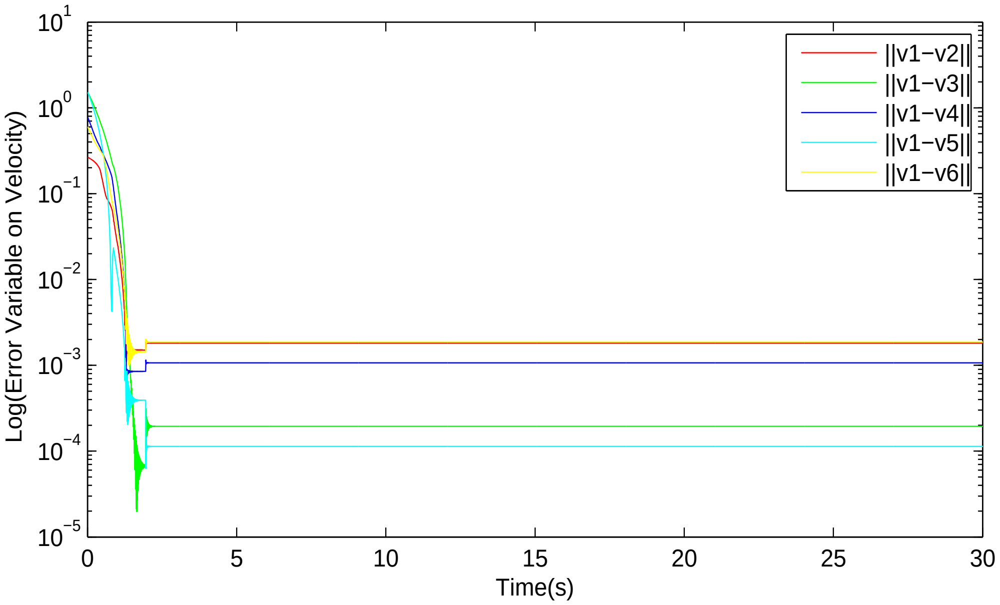


Figure 6.1: The finite-time convergence results on error variables $||v_1 - v_2||_2$, $||v_1 - v_3||_2$, $||v_1 - v_4||_2$, $||v_1 - v_5||_2$, and $||v_1 - v_6||_2$ where v_i denotes the virtual velocity of the ith robot.

are $a = 1$, $\beta = 0.6$, and $\gamma = 0.9$. Figure 6.1 and Figure 6.2 show the finite-time convergence results of the virtual velocity and position in the logarithmic scale for the robot 1. From two figures, one can see that the virtual velocity reaches the consensus after about 3s while the virtual position is the position decided by the potential function after about 6s.

6.2.3 Tracking Control

In order to enable the real velocity and position for the second-order dynamics (1.7) to track the virtual velocity and position, we give the following finite-time tracking control algorithm, which is described by

$$u_i(t) \quad = \quad \dot{\hat{v}}_i - k((v_i - \hat{v}_i)^{1/q} + {k_1}^{1/q}(x_i - \hat{x}_i))^{2q-1} \tag{6.2}$$

where x_i is the real position of the ith robot; $\hat{x}_i$ is the virtual position of the ith robot; v_i is the real velocity of the ith robot; $\hat{v}_i$ is the virtual velocity of the ith robot; $k_1 > \frac{1}{3}$; $k > 2^{\frac{2+q}{2}} + 2(2-q)k_1^{1/q}|v_i - \hat{v}_i|_{\max} + k_1 - \frac{1}{3}$; $|v_i - \hat{v}_i|_{\max}$ is the max

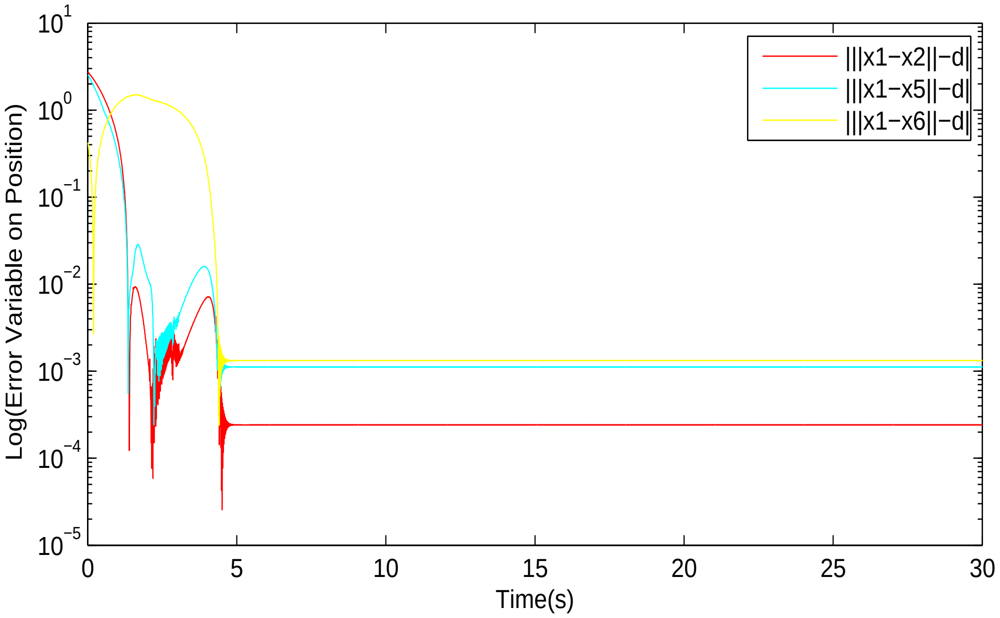


Figure 6.2: The finite-time convergence results on error variables $|||x_1 - x_2||_2 - d|$, $|||x_1 - x_4||_2 - d|$, $|||x_1 - x_5||_2 - d|$, and $|||x_1 - x_6||_2 - d|$ where x_i denotes the virtual position of the ith robot and d denotes the predefined distance between two robots.

bound for $v_i - \hat{v}_i$; and $q = \frac{2a-1}{2a+1} > \frac{1}{2}$ because $2q - 1$ should be larger than zero and equal $\frac{2a-3}{2a+1}$ where both $2a - 3$ and $2a + 1$ are odd integers.

The objective of the finite-time tracking control algorithm (6.2) is to guarantee that $x_i \to \hat{x}_i$, and $v_i \to \hat{v}_i$ within a finite-time interval. The following Example 2 will illustrate the objective of the finite-time tracking control algorithm.

Example 2: Consider that the finite-time coordination control algorithm has converged, i.e. $\hat{v}_i = 0.8$ and $\dot{\hat{x}}_i = 0.8$. For the finite-time tracking control algorithm (6.2), the parameters are $q = 3/5$, $k_1 = 0.5$, and $k = 7$. Figure 6.3 and Figure 6.4 show the finite-time convergence results of the real position and velocity in the logarithmic scale for the robot i. From two figures, one can see that the real velocity tracks the virtual velocity after about 4s while real position equals to the virtual one after about 5s.

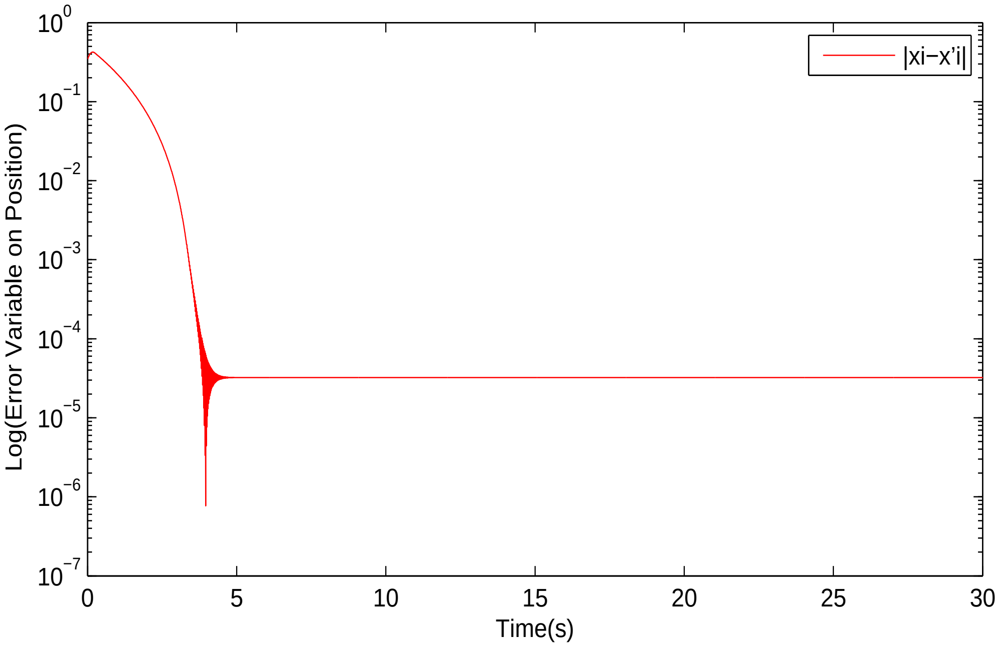


Figure 6.3: The finite-time convergence results on error variables $|x_i - x'_i|$ where x_i and x'_i denote the real position and virtual position of the ith robot, respectively.

6.2.4 Finite-Time Parallel Motion Control

Based on the finite-time coordination control algorithm and the finite-time tracking control algorithm, we will propose a finite-time parallel motion control algorithm, which can coordinate the robot group to parallel move in order to trace the plume. Therefore, we use the following control algorithm to take place of the finite-time coordination control algorithm (6.1).

$$\dot{\hat{x}}_i(t) = \hat{v}_i(t) - \beta\left(\frac{\partial J}{\partial \hat{x}_i}\right)^{\frac{2a-1}{2a+1}} \tag{6.3a}$$

$$\dot{\hat{v}}_i(t) = \dot{v}_c + \gamma\left(\sum_{j=1}^{N} a_{ij}(\hat{v}_j - \hat{v}_i) + \lambda(v_c - \hat{v}_i)\right)^{\frac{2a-1}{2a+1}} \tag{6.3b}$$

where $a > 0.5$, $\beta > 0$, $\gamma > 0$, $\lambda > 0$; $\hat{x}_i(t)$ and $\hat{v}_i(t)$ denote the virtual position and velocity of the ith robot, respectively; v_c is the reference velocity. J is a potential function that can be constructed based on the real requirements.

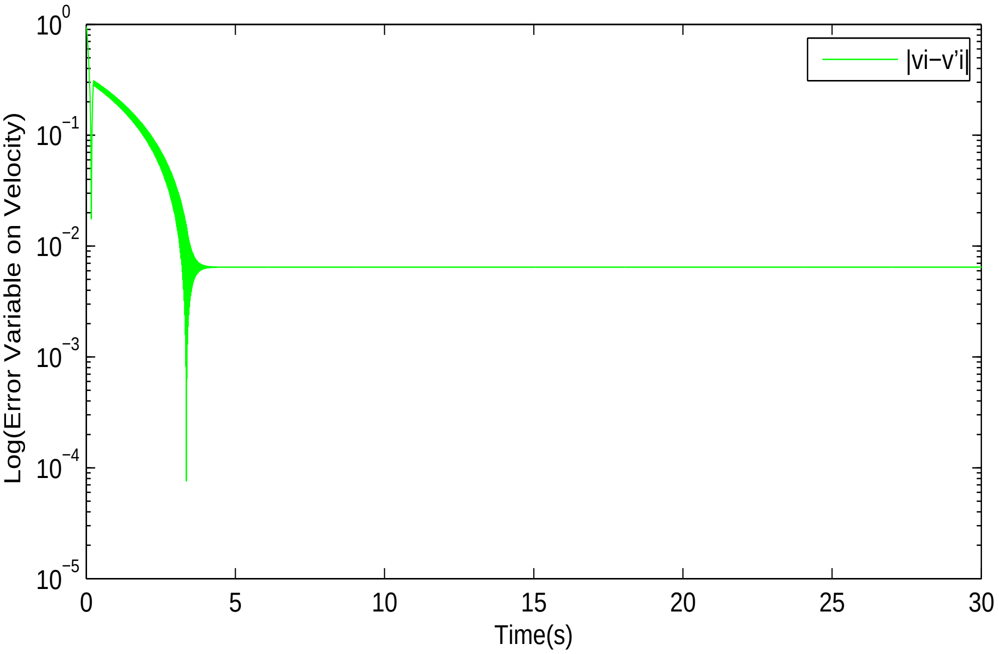


Figure 6.4: The finite-time convergence results on error variables $|v_i - v'_i|$ where v_i and v'_i denote the real velocity and virtual velocity of the ith robot, respectively.

It is straightforward to see that the reference velocity v_c from the decision level is introduced into the finite-time coordination control algorithm (6.1). The objective of the control algorithm (6.3) is to guarantee that the potential J is minimized and $v_i \to v_c$, $v_j \to v_c$. Therefore, the finite-time parallel motion control algorithm is described in Algorithm 3.

Example 3: Consider the double-integrator dynamics (1.7). The conditions and parameters are the same as Example 1 and Example 2, and set $\lambda = 0.5$ and $v_c = [-1.9, 1.9]$. Figure 6.5 shows that the parallel movement of six robots. Figure 6.6 and Figure 6.7 show the finite-time convergence results of the real velocity and position in the logarithmic scale under the reference velocity v_c.

However, each robot has its individual reference velocity. Hence, the motion

Algorithm 3 The finite-time parallel motion control algorithm

```
/*Initialization*/
Initialize parameters $\beta$, $a$, $\gamma$, and $\lambda$ for the coordination control algorithm (6.3);
Initialize parameters $k$, $k_1$, and $q$ for the tracking control algorithm (6.2);
Initialize parameters for the potential function $J$;
Perform (1.8) and (1.9) to get the real position and velocity of the "hand position";
Run the the distributed decision algorithm to obtain the reference motion direction of the
    robot and initialize the virtual position $\hat{x}_i(0)$ and velocity $\hat{v}_i(0)$;
/*Main Body* for the ith robot/
repeat
    Perform (6.3) to obtain the virtual position $\hat{x}_i(t)$ and velocity $\hat{v}_i(t)$;
    Perform (1.8) and (1.9) to get the real position and velocity of the "hand position";
    Perform (6.2) to calculate the control law $u_i(t)$;
    Perform (1.10) to calculate the control input $p_i$;
    Perform (1.11) and (1.12) to obtain the applied torques for the left and right wheels;
until Termination conditions are satisfied.
```

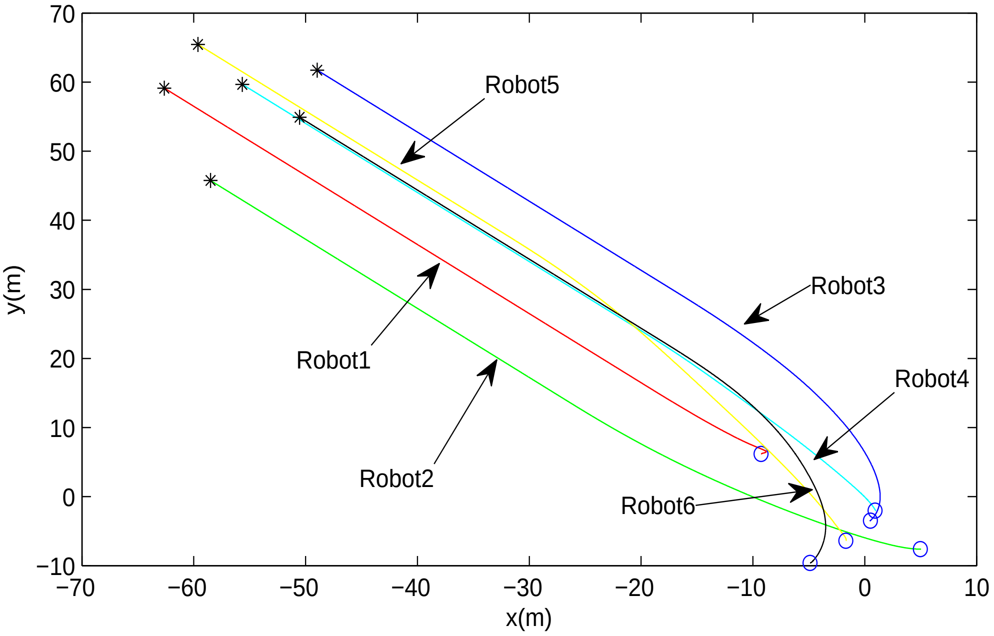


Figure 6.5: The parallel movement for six robots where 'o' denotes the initial position and '*' denotes the end position.

control algorithm (6.3) can be modified by

$$\dot{\hat{x}}_i(t) = \hat{v}_i(t) - \beta\left(\frac{\partial J}{\partial \hat{x}_i}\right)^{\frac{2a-1}{2a+1}} \tag{6.4a}$$

$$\dot{\hat{v}}_i(t) = \dot{v}_c^i + \gamma\left(\sum_{j=1}^{N} a_{ij}(\hat{v}_j - \hat{v}_i) + \lambda(v_c^i - \hat{v}_i)\right)^{\frac{2a-1}{2a+1}} \tag{6.4b}$$

where v_c^i is the reference velocity of the ith robot.

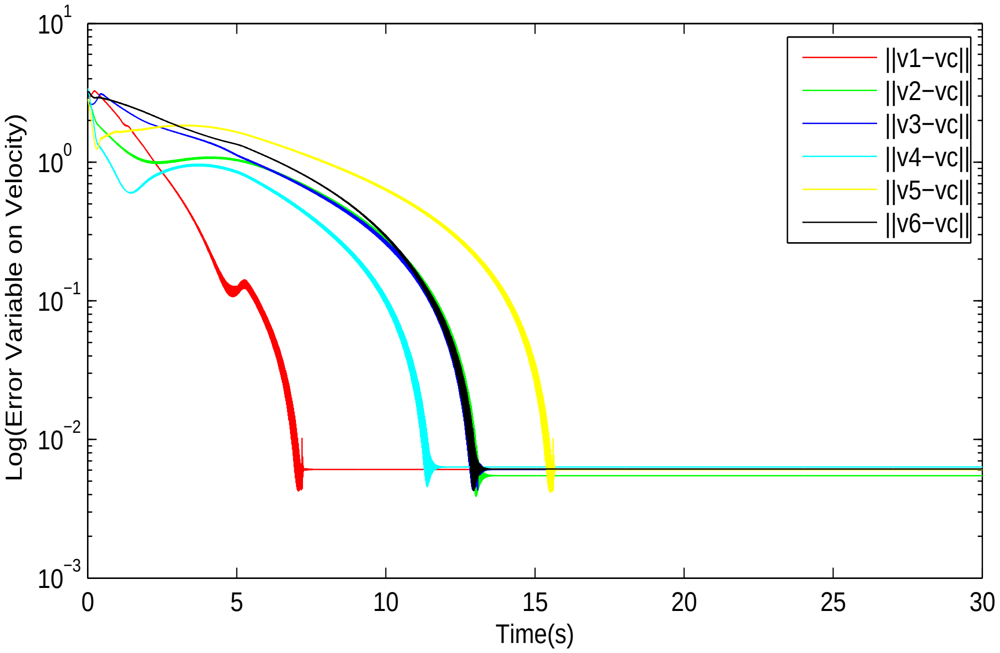


Figure 6.6: The finite-time convergence results on error variables $||v_1 - v_c||_2$, $||v_2 - v_c||_2$, $||v_3 - v_c||_2$, $||v_4 - v_c||_2$, $||v_5 - v_c||_2$, and $||v_6 - v_c||_2$ where v_i denotes the real velocity of the ith robot.

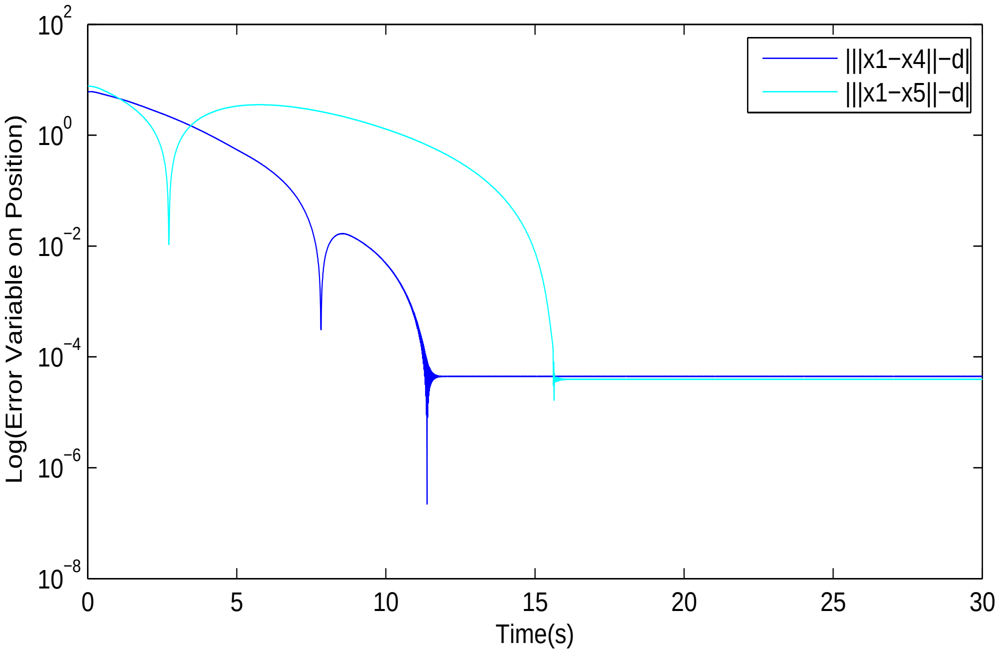


Figure 6.7: The finite-time convergence results on error variables $|||x_1 - x_4||_2 - d|$, and $|||x_1 - x_5||_2 - d|$ where x_i denotes the real position of the ith robot and d denotes the predefined distance between two robots.

6.2.5 Finite-Time Circular Motion Control

Based on the finite-time coordination control algorithm and the finite-time tracking control algorithm, we will propose a finite-time circular motion control algorithm, which can coordinate the robot group to circle a predicted odour source position in order to search for odour clues. We first introduce a virtual robot that can communicate with all real robots, which is shown in Figure 6.8 where ξ^r denotes the virtual robot. Accordingly, the position and velocity of the virtual robot are h_r and $\dot{h}_r$, respectively. $h_r(t) = \begin{pmatrix} h_x^c + R\cos(\theta_r) & h_y^c + R\sin(\theta_r) \end{pmatrix}^T$; $\dot{\theta}_r = \omega_0$; R denotes the radius; ω_0 denotes the angular velocity; and $h^c(h_x^c, h_y^c)$ denotes the probable position of the odour source.

Then, we use the following control algorithm to take place of the finite-time coordination control algorithm (6.1).

$$\begin{aligned} \dot{\hat{x}}_i(t) &= \hat{v}_i(t) - \beta \left(\frac{\partial \breve{J}}{\partial \hat{x}_i} \right)^{\frac{2a-1}{2a+1}} && \text{(6.5a)} \\ \dot{\hat{v}}_i(t) &= \ddot{h}_r + \gamma \left(\sum_{j=1}^{N} a_{ij}(\hat{v}_j - \hat{v}_i) + \lambda(\dot{h}_r - \hat{v}_i) \right)^{\frac{2a-1}{2a+1}} && \text{(6.5b)} \end{aligned}$$

where $a > 0.5$, $\beta > 0$, $\gamma > 0$, $\lambda > 0$; $\hat{x}_i(t)$ and $\hat{v}_i(t)$ denote the virtual position and virtual velocity of the ith robot, respectively; $\dot{h}_r$ is the velocity of the virtual robot. $\breve{J}$ is an extended potential function for all robots including the virtual robot.

The objective of the control algorithm (6.5) is to guarantee that the extended potential $\breve{J}$ is minimized and $v_i \to \dot{h}_r$, $v_j \to \dot{h}_r$. Therefore, the finite-time circular motion control algorithm is described in Algorithm 4. Similarly, each robot may obtain the different reference center h^c, i.e. h^{ic}.

Example 4: We still use five robots marked by black star and a virtual robot marked by red star and the same initial conditions as Example 3 except for $d = 2.5$. Moreover, let parameters $R = 15$, $\omega_0 = 0.1$, and $h^c = [9, 9]$. Figure 6.9 shows the circular movement process of six robots.

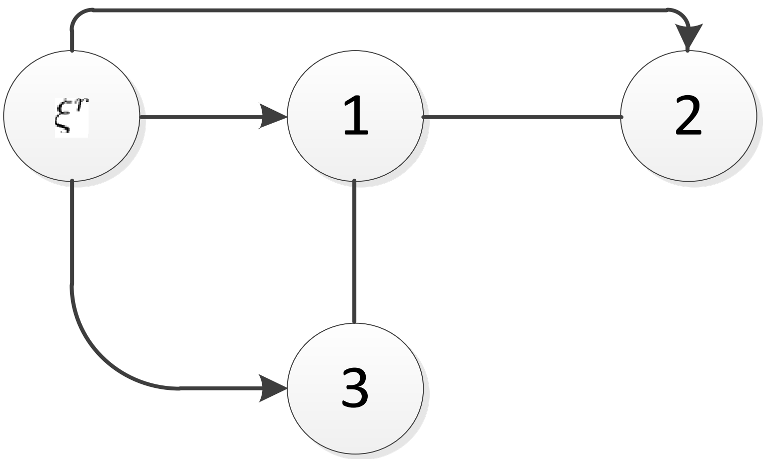


Figure 6.8: The communication topology for three robots with a virtual robot.

Algorithm 4 The finite-time circular motion control algorithm

```
/*Initialization*/
Initialize parameters $\beta$, $\gamma$, $a$, and $\lambda$ for the coordination control algorithm (6.5);
Initialize parameters of the virtual robot $R$, $\theta_r(0)$, and $\omega_0$;
Initialize parameters $k$, $k_1$, and $q$ for the tracking control algorithm (6.2);
Initialize parameters for the potential function $J$;
Initialize the virtual position $\hat{x}_i(0)$ and velocity $\hat{v}_i(0)$;
Run the distributed decision algorithm to obtain the position of the odour source;
/*Main Body* for the ith robot/
repeat
    Perform (6.5) to obtain the virtual position $\hat{x}_i(t)$ and velocity $\hat{v}_i(t)$;
    Perform (1.8) and (1.9) to get the real position and velocity of the “hand position”;
    Perform (6.2) to calculate the control law $u_i(t)$;
    Perform (1.10) to calculate the control input $p_i$;
    Perform (1.11) and (1.12) to obtain the applied torques for the left and right wheels;
until Termination conditions are satisfied.
```

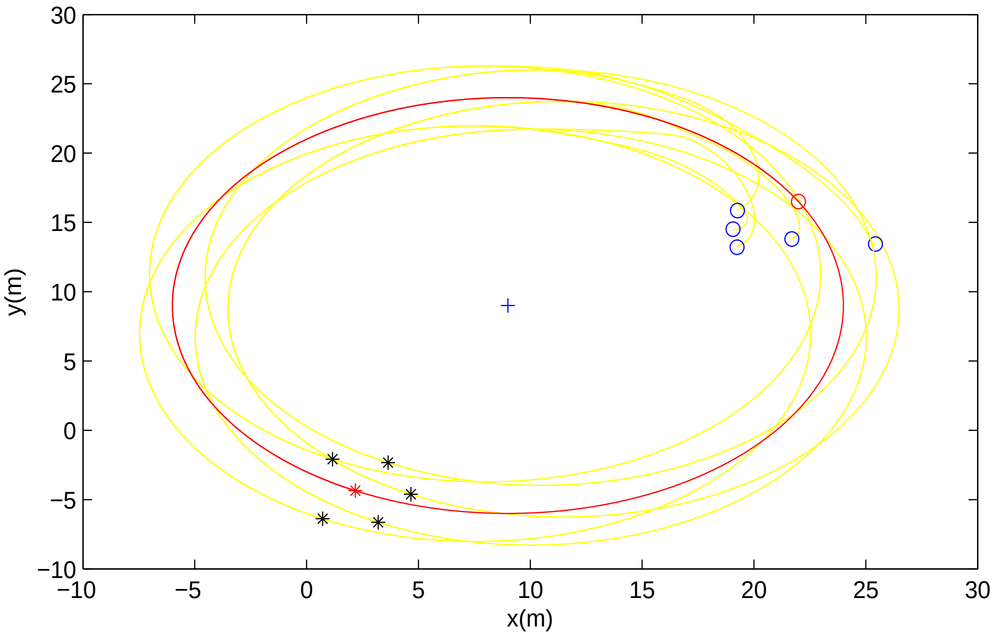


Figure 6.9: The circular movement for six robots where 'o' denotes the initial position, '*' denotes the end position, '+' refers to the position of the odour source, the red line denotes the trajectory of the virtual robot, and the yellow line denotes the trajectories of the real robot.

6.2.6 Potential-Based Finite-time Motion Control Algorithm

The potential-based finite-time motion control algorithm includes a finite-time parallel motion control algorithm and a finite-time circular motion control algorithm. In terms of the detection event of the multi-robot system, the different motion control algorithms are used. To sum up, the potential-based finite-time motion control algorithm is described in Algorithm 5.

6.2.7 Stability Analysis

In this subsection, we will prove the convergence of the finite-time coordination control algorithm (6.1) and the finite-time tracking control algorithm (6.2). Moreover, the proof processes of the motion control algorithms (6.3) and (6.5) are omitted because they are similar to that of the finite-time coordination control algorithm

Algorithm 5 The potential-based finite-time motion control algorithm

```
/*Initialization*/
Initialize parameters $\beta$, $\gamma$, $a$, and $\lambda$ for the coordination control algorithm (6.5);
Initialize parameters of the virtual robot $R$, $\theta_r(0)$, and $\omega_0$;
Initialize parameters $k$, $k_1$, and $q$ for the tracking control algorithm (6.2);
Initialize parameters for the potential function $J$;
Initialize parameters $\beta$, $a$, $\gamma$, and $\lambda$ for the coordination control algorithm (6.3);
Initialize parameters for the potential function $J$;
Perform (1.8) and (1.9) to get the real position and velocity of the "hand position";
Run the the distributed decision algorithm to obtain the reference motion direction of the
robot and initialize the virtual position $\hat{x}_i(0)$ and velocity $\hat{v}_i(0)$;
/*Main Body* for the ith robot/
repeat
    IF the robot group does not detect the chemical signal and the time $T > 20s$
    Perform (6.5) to obtain the virtual position $\hat{x}_i(t)$ and velocity $\hat{v}_i(t)$;
    Perform (1.8) and (1.9) to get the real position and velocity of the "hand position";
    Perform (6.2) to calculate the control law $u_i(t)$;
    Perform (1.10) to calculate the control input $p_i$;
    Perform (1.11) and (1.12) to obtain the applied torques for the left and right wheels;
    ELSE
    Perform (6.3) to obtain the virtual position $\hat{x}_i(t)$ and velocity $\hat{v}_i(t)$;
    Perform (1.8) and (1.9) to get the real position and velocity of the "hand position";
    Perform (6.2) to calculate the control law $u_i(t)$;
    Perform (1.10) to calculate the control input $p_i$;
    Perform (1.11) and (1.12) to obtain the applied torques for the left and right wheels;
until Termination conditions are satisfied.
```

(6.1). We give the following lemmas [7, 6, 24], which will be used in convergence analysis.

Lemma 6. *For any real numbers x_i, $i = 1, \ldots, n$ and $0 < b \leq 1$, the following inequality holds:*

$$(|x_1| + \cdots + |x_n|)^b \leq |x_1|^b + \cdots + |x_n|^b$$

When $b = p/q \leq 1$, where $p > 0$ and $q > 0$ are odd integers,

$$|x^b - y^b| \leq 2^{1-b}|x - y|^b$$

Lemma 7. *Let c, d be positive real numbers and $\gamma(x, y) > 0$ a real-valued function. Then,*

$$|x|^c|y|^d \leq \frac{c\gamma(x,y)|x|^{c+d}}{c+d} + \frac{d\gamma^{-c/d}(x,y)|y|^{c+d}}{c+d}$$

The following proposition will show that the coordination control algorithm (6.1) converges in finite-time.

Proposition 9. *Consider the coordination control algorithm (6.1). If the interaction topology $G(\nu, \mathcal{E}, A)$ among robots is undirected and connected, the finite-time coordination control algorithm guarantees that the potential J is minimized and $\hat{v}_i \to \hat{v}_j$, $\forall j$ within a finite-time interval.*

Proof. For the coordination control algorithm (6.1), position and velocity are decoupled. Hence, we first show the velocity consensus in finite-time.

Let $\phi_i = \sum_{i=1}^{N} a_{ij}(\hat{v}_i - \hat{v}_j)$ and choose a Lyapunov candidate as

$$V = \frac{1}{2}\sum_{i=1}^{N} \phi_i^2$$

Along the trajectory (6.1), we have

$$\begin{aligned}\dot{V} &= \sum_{i=1}^{N} \phi_i \dot{\phi}_i \\ &= -\gamma \phi^T L(A) \phi^{\frac{2a-1}{2a+1}} \\ &\leq -k_1 \gamma \sum_{i=1}^{N} \phi_i^{\frac{4a}{2a+1}}\end{aligned}$$

Where $L(A)$ is Laplacian matrix. Because $\dot{V} < 0$ at $\hat{v}_i \neq \hat{v}_j$, $\hat{v}_i \to \hat{v}_j$ as $t \to \infty$. Moreover, we can get

$$\begin{aligned}V^{\frac{2a}{2a+1}} &= \left(\frac{1}{2}\right)^{\frac{2a}{2a+1}} \left(\sum_{i=1}^{N} \phi_i^2\right)^{\frac{2a}{2a+1}} \\ &\leq k_2 \left(\frac{1}{2}\right)^{\frac{2a}{2a+1}} \sum_{i=1}^{N} \phi_i^{\frac{4a}{2a+1}}\end{aligned}$$

Let $k = \frac{k_1 \gamma}{k_2 (1/2)^{\frac{2a}{2a+1}}}$. Hence, we have $\dot{V} + kV^{\frac{2a}{2a+1}} \leq 0$. By Lemma 1, $\hat{v}_i \to \hat{v}_j$ as $t \to \frac{2a+1}{k} V(0)^{\frac{1}{2a+1}}$.

Assume that $\hat{v}_i = \hat{v}_j = v_0 = \dot{x}_0$ when $t > \frac{2a+1}{k} V(0)^{\frac{1}{2a+1}}$. We can rewrite (6.1a) as

$$\dot{\hat{x}}_i(t) = \dot{x}_0(t) - \beta \left(\frac{\partial J}{\partial \hat{x}_i}\right)^{\frac{2a-1}{2a+1}} \tag{6.6}$$

Let $\xi_i(t) = \hat{x}_i(t) - x_0(t)$ and then get

$$\dot{\xi}_i(t) \;=\; -\beta\left(\frac{\partial J}{\partial \xi_i}\right)^{\frac{2a-1}{2a+1}} \tag{6.7}$$

Choose a nonnegative function as

$$V = J$$

Along the system trajectories (6.7), we can derive

$$\begin{aligned}\frac{dV}{dt} &= \sum_{i=1}^{N}\left(\frac{\partial J}{\partial \xi_i}\right)^{T}\dot{\xi}_i(t)\\ &= -\beta\sum_{i=1}^{N}\left(\left\|\frac{\partial J}{\partial \xi_i}\right\|_2^2\right)^{\frac{2a}{2a+1}}\\ &\leq -\beta\left(\sum_{i=1}^{N}\left\|\frac{\partial J}{\partial \xi_i}\right\|_2^2\right)^{\frac{2a}{2a+1}}\end{aligned}$$

which means the function V is monotonically decreasing for all $t > \frac{2a+1}{k}V(0)^{\frac{1}{2a+1}}$. From LaSalle's invariance principle, all solutions starting in $\Omega_c = \{\xi_i | \|\xi_i - \xi_j\|_2 = d\}$ converge to the largest invariance set in $E = \{\xi_i \in \Omega_c | \frac{dV}{dt} = 0\}$. Hence, $\xi_i(t)$ will asymptotically converge to the local minimum of the potential function J, which means $\|\xi_i - \xi_j\|_2 = d$, i.e. $\|x_i - x_j\|_2 = d$.

To show that $\|x_i - x_j\|_2 = d$ is a finite-time-stable equilibrium, we suppose $V(t) \neq 0$, and then let $\Upsilon_1 = -\frac{\frac{dV(t)}{dt}}{V(t)^{\frac{2a}{2a+1}}}$. Hence,

$$\begin{aligned}\Upsilon_1 &\geq \frac{\beta(\sum_{i=1}^{N}\|\frac{\partial J}{\partial \xi_i}\|_2^2)^{\frac{2a}{2a+1}}}{J^{\frac{2a}{2a+1}}}\\ &\geq \frac{\beta}{c^{\frac{2a}{2a+1}}}\end{aligned}$$

where the second inequality comes from the property of the potential function. Let $k' = \frac{\beta}{c^{\frac{2a}{2a+1}}}$ and we have $\dot{V} + kV^{\frac{a+1}{2}} \leq 0$. Hence, when $t > \frac{2a+1}{k}V(0)^{\frac{1}{2a+1}} + \frac{2a+1}{k'}J(0)^{\frac{1}{2a+1}}$, $\xi_i(t)$ will converge to the local minimum of the potential function J in finite-time. □

The following proposition guarantees the finite-time convergence of the tracking control algorithm (6.2).

Proposition 10. *Consider the robot dynamics (1.7) with the control input (6.2). The finite-time tracking control algorithm guarantees that $v_i \to \hat{v}_i$ and $x_i \to \hat{x}_i$, $i \in l_N$ within a finite-time interval.*

Proof. When $t > \frac{2a+1}{k}V(0)^{\frac{1}{2a+1}} + \frac{2a+1}{k'}J(0)^{\frac{1}{2a+1}}$, let $x_1 = x_i - \hat{x}_i$ and $x_2 = v_i - \hat{v}_i$, and consider the following Lyapunov candidate as

$$V = \frac{1}{2}x_1^2 + \int_{x_2^*}^{x_2} (s^{1/q} - x_2^{*1/q})^{2-q}\mathrm{d}s \tag{6.8}$$

where $q = \frac{2a-1}{2a+1}$; $x_2^* = -k_1 x_1^q$, $k_1 > 0$.

Along the trajectory of (6.2), we have

$$\begin{aligned} \dot{V} &= x_1x_2 + (x_2^{1/q} - x_2^{*1/q})^{2-q}u_i \\ &\quad -(2-q)x_2\frac{\partial x_2^{*1/q}}{\partial x_1}\int_{x_2^*}^{x_2}(s^{1/q} - x_2^{*1/q})^{1-q}\mathrm{d}s \end{aligned} \tag{6.9}$$

Let $\xi_1 = x_1$ and $\xi_2 = x_2^{1/q} - x_2^{*1/q}$.

$$\begin{aligned} \dot{V} &\leq \xi_1(x_2 - x_2^*) + x_1x_2^* + \xi_2^{2-q}u_i \\ &\quad +(2-q)k_1^{1/q}|x_2||x_2 - x_2^*||x_2^{1/q} - x_2^{*1/q}| \end{aligned} \tag{6.10}$$

By Lemma 6 and Lemma 7,

$$\begin{aligned} \dot{V} &\leq 2|\xi_1||x_2^{1/q} - x_2^{*1/q}|^q - k_1x_1^{q+1} + \xi_2^{2-q}u_i \\ &\quad +2(2-q)k_1^{1/q}|x_2||x_2^{1/q} - x_2^{*1/q}|^q|\xi_2| \\ &\quad \leq 2|\xi_1||\xi_2|^q - k_1x_1^{q+1} + \xi_2^{2-q}u_i \\ &\quad +2(2-q)k_1^{1/q}|x_2||\xi_2|^{1+q} \\ &\quad \leq \frac{1}{3}|\xi_1|^{1+q} + 2^{\frac{2+q}{q}}|\xi_2|^{1+q} - k_1\xi_1^{q+1} + \xi_2^{2-q}u_i \\ &\quad +2(2-q)k_1^{1/q}|x_2||\xi_2|^{1+q} \end{aligned}$$

Because $|x_2| < \sigma$, let $k_2 = 2^{\frac{2+q}{q}} + 2(2-q)k_1^{1/q}\sigma$ and $d = q + 1$.

$$\dot{V} \;\leq\; -(k_1 - \frac{1}{3})|\xi_1|^d + \xi_2^{2-q}u_i + k_2|\xi_2|^d$$

Let $k_1 > \frac{1}{3}$, $k = k_2 + k_1 - \frac{1}{3}$ and $\xi_2^{2-q}u_i + k|\xi_2|^d = 0$. Then, we have

$$\dot{V} \;\leq\; -(k_1 - \frac{1}{3})(|\xi_1|^d + |\xi_2|^d)$$

Moreover,

$$V \;\leq\; 2|\xi_1|^2 + 2|\xi_2|^{d+1}$$

Because the system (1.7) is asymptotical convergence with the control input (6.2), consider the set $\mathcal{U} = \{\zeta||\zeta| < 1\}$. When $\xi_2 \in \mathcal{U}$, we derive

$$V \;\leq\; 2|\xi_1|^2 + 2|\xi_2|^2$$

Then

$$V^{d/2} \;\leq\; 2^{d/2}(|\xi_1|^d + |\xi_2|^d)$$

Let $k' = \frac{k_1-2}{2^{d/2}}$ and we obtain

$$\dot{V} + k'V^{d/2} \;\leq\; 0$$

By Lemma 1, the system (1.7) is finite-time stable with the control input (6.2). □

6.3 Performance Capabilities of the Potential-Based Finite-time Motion Control

In this section, we will implement the potential-based finite-time motion control algorithm (FTMCS) in a physics simulation platform provided by Microsoft Robotics Studio. It should be pointed that the state data such as ν_i, r_i, θ_i, and ω_i are mainly

Table 6.1: The parameters of the parallel motion control algorithm.

λ	γ	β	q	k	k_1	v_c(m/s)	r(m)	d(m)	ϵ	h
0.4	1	0.5	0.6	7	0.5	(0.4,-0.2)	5	2.8	0.1	0.2

Table 6.2: The parameters of the circular motion control algorithm.

λ	γ	β	q	k	k_1	R(m)	ω_0(radian/s)	h^c(m,m)	r(m)	d(m)	ϵ	h
0.6	1	0.3	0.6	7	0.5	8	0.02	(0,0)	9	2.8	0.1	0.2

provided by the Physics Engine of Microsoft Robotics Studio. The parameters for FTMCS are listed in Table 6.1 and Table 6.2.

Figure 6.10 shows the parallel movement of three robots. The corresponding linear velocity, angular velocity, heading, and torques are shown in Figure 6.11, Figure 6.12, Figure 6.13, and Figure 6.14, respectively. Figure 6.15 shows the circular movement of three robots. The corresponding linear velocity, angular velocity, heading, and torques are shown in Figure 6.16, Figure 6.17, Figure 6.18, and Figure 6.19, respectively. From Figure 6.10 and Figure 6.15, one can see that the parallel motion and the circular motion has been realized in the physics environment by employing FTMCS. Moreover, from other figures, one can also see that the error accumulation is inevitable and puts an impact on the linear velocity, angular velocity, and heading. An important characteristic of the finite-time controller is to enable the system states to rapidly arrive in the vicinity of equilibriums, which is efficient especially for the dynamical search environment because rapid formation can better capture the time-varying plume.

6.4 Odour Source Localisation

In this section, we will test the effectiveness of FTMCS for odour source localisation. For the parallel motion, the parameters can be found in Table 6.1 except for $d = 7$ and $r = 1.2d$. For the circular motion, the parameters can also be found in Table 6.2 except for $d = 2.5$, $r = 1.2d$, $R = 15$, and $\omega_0 = 0.1$. Moreover, we use a circle where the real position of the odour source is viewed as a center with a predefined radius $1m$ as one of termination conditions, which means that the search task is finished if any robot enters the circle. (The other termination condition is the maximal search time $1500s$). In addition, it should be pointed out that the robot group will search for the odour clues along the direction of y axis from the initial positions (right-up corner) to the (80m,0m). Once the odour clues are detected by any robot, FTMCS will start to run.

The prediction errors of five robots about the position of the odour source are shown in Figure 6.20 where the robots keep the predefined position (80m,0m) of the odour source from 0s to about 50s. After about 50s, the robot group detects the odour clues, and then FTMCS starts to run. Correspondingly, the motion process of the robot group controlled by FTMCS is illustrated in Figure 6.21 and Figure 6.22. In Figure 6.21(a), the initial positions of the robot group are set at the right-up corner in the search region. In Figures 6.21(b) and 6.22(a), the parallel motion is used to track the plume and move along the plume in terms of the probable positions of the odour source. In Figure 6.22(b), the robot group finds the real odour source.

Then, we will compare the search efficiency of the multi-robot system coordinated by FTMCS with several selected algorithms which include PSO [93], PPSO-IM [85], LPSO [87], CPSO [53], WUI-45 [53], WUII [53], and CCS(L=2) [84]. The parameters of the seven algorithms can be found in [85], [53], [93], [87], and [84], respectively. For all algorithms, the robot group will search for the odour clues along the direction of y axis from the initial positions (right-up corner) to the (80m,0m). Once the odour clues are detected by any robot, these algorithms will start to run. In Table 6.3, the success rates obtained by eight algorithms are shown. Figure 6.23 shows less search time obtained from FTMCS. Moreover, since path length denotes the energy consumed by robots, the shorter path length means the better search performance. From Figure 6.24, one can see that robots coordinated by FTMCS

Table 6.3: The success rates (%) based on 50 runs.

Algorithms	3 robots	5 robots	7 robots	9 robots
PSO [93]	36	54	70	68
PPSO-IM [85]	92	100	100	100
LPSO [87]	100	100	100	100
CPSO [53]	42	46	70	76
WUI-45 [53]	40	70	90	80
WUII [53]	58	86	98	96
CCS(L=2) [84]	96	100	100	100
FTMCS	100	100	100	100

consume the lesser energy than those coordinated by other compared algorithms.

6.5 Conclusion

We have developed a potential-based finite-time motion control algorithm to deal with the problem of odour source localisation. Firstly, we have developed a finite-time coordination control algorithm such that the virtual shape decided by the potential function and virtual velocity consensus can be reached in finite-time. Secondly, we have derived a finite-time tracking control algorithm such that the real velocity and position can track the virtual velocity and position. On the basis of the finite-time coordination control algorithm and the finite-time tracking control algorithm, we have developed a potential-based finite-time motion control algorithm, which includes a finite-time parallel motion control algorithm and a finite-time circular motion control algorithm. Finally, we have illustrated the effectiveness of the potential-based finite-time motion control algorithm for the problem of odour source localisation.

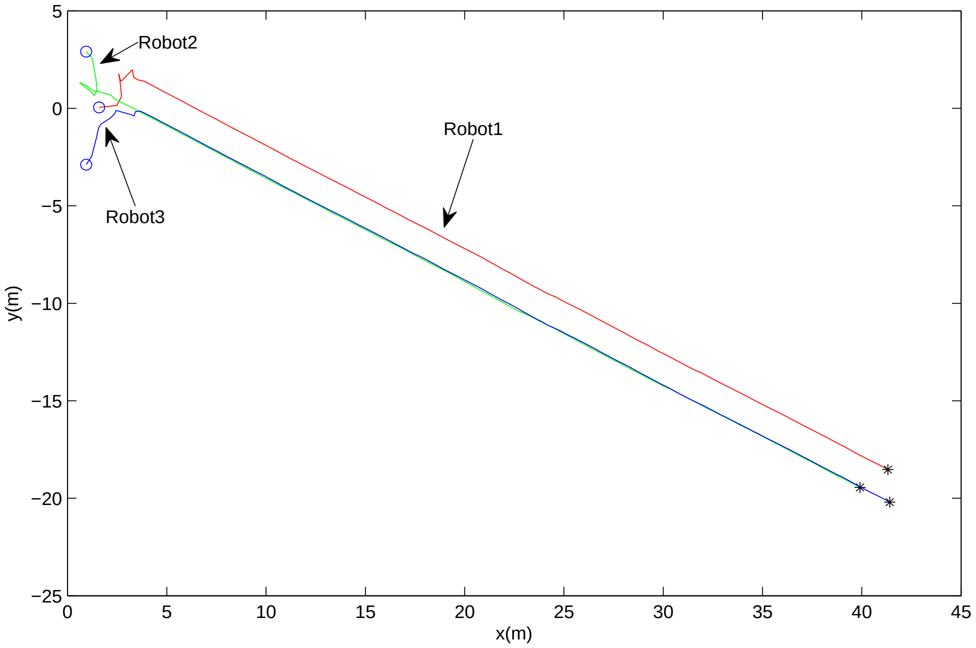


Figure 6.10: The parallel movement for three robots where ‘o’ denotes the initial position and ‘*’ denotes the end position.

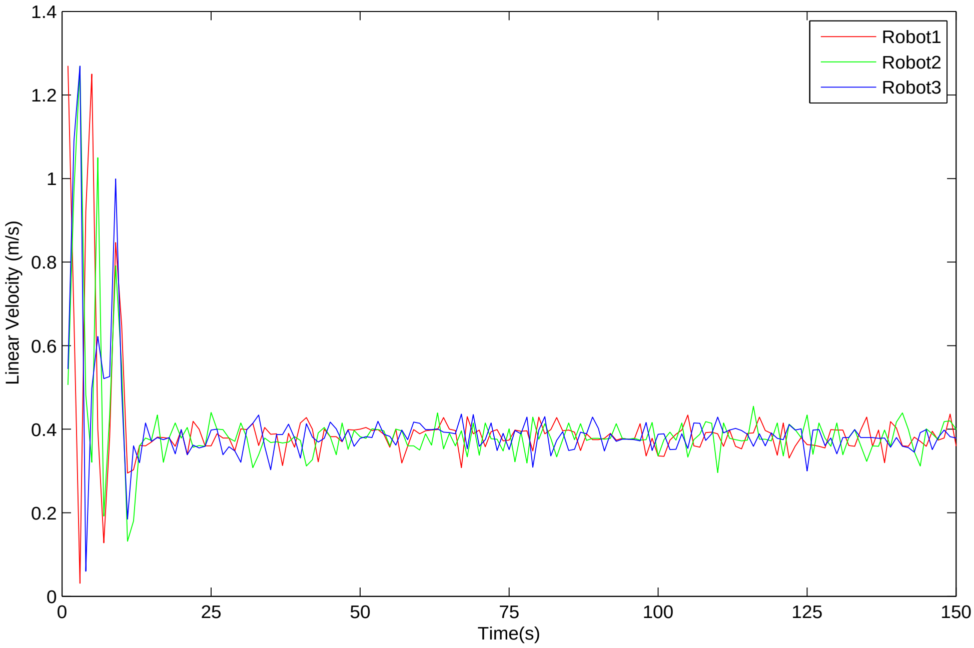


Figure 6.11: The change curve of linear velocity for parallel movement.

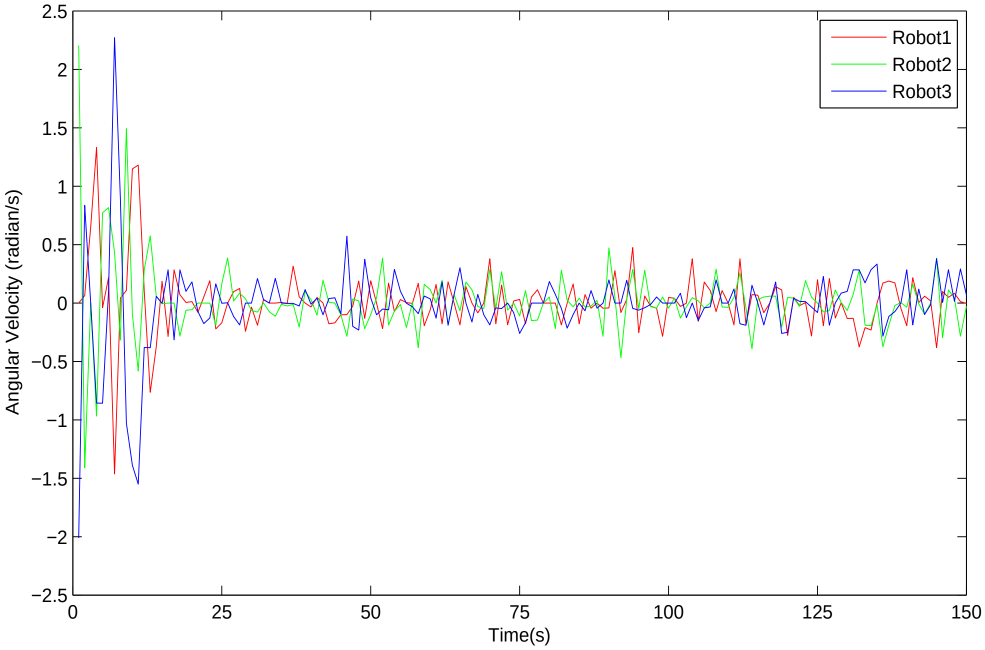


Figure 6.12: The change curve of angular velocity for parallel movement.

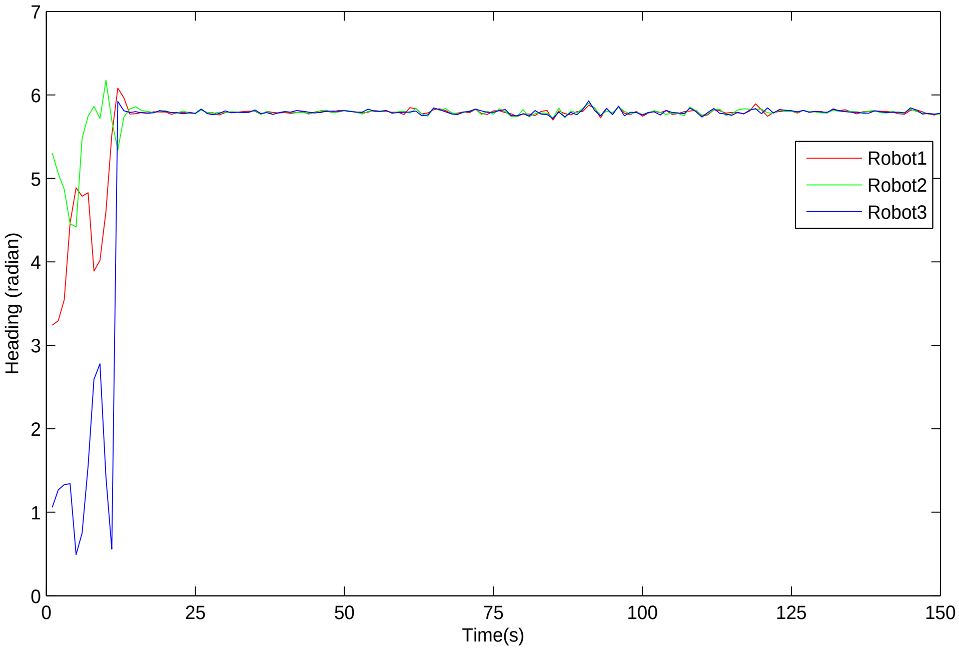
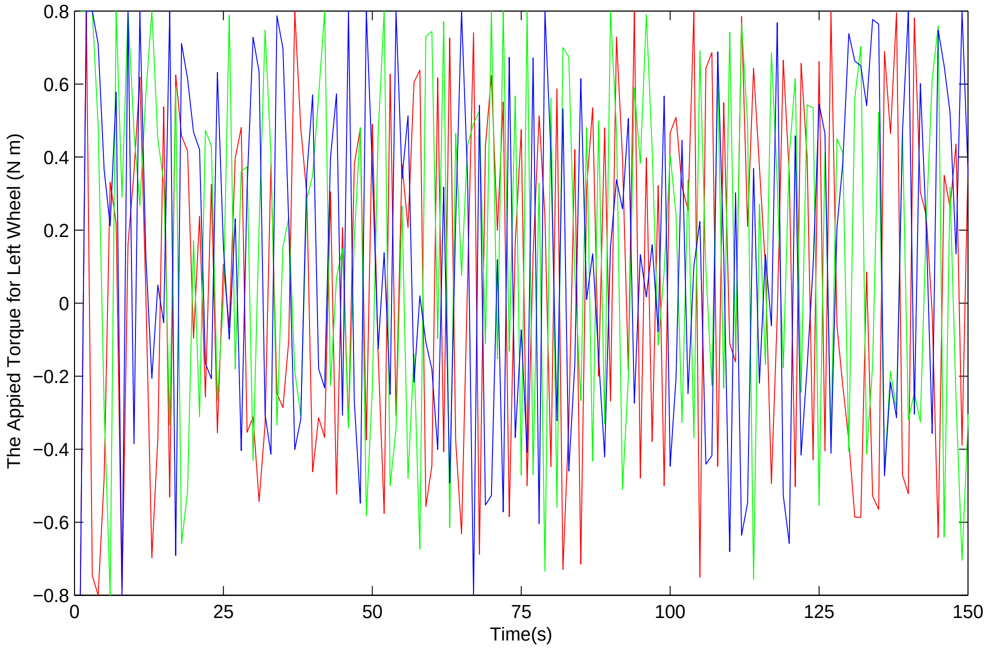
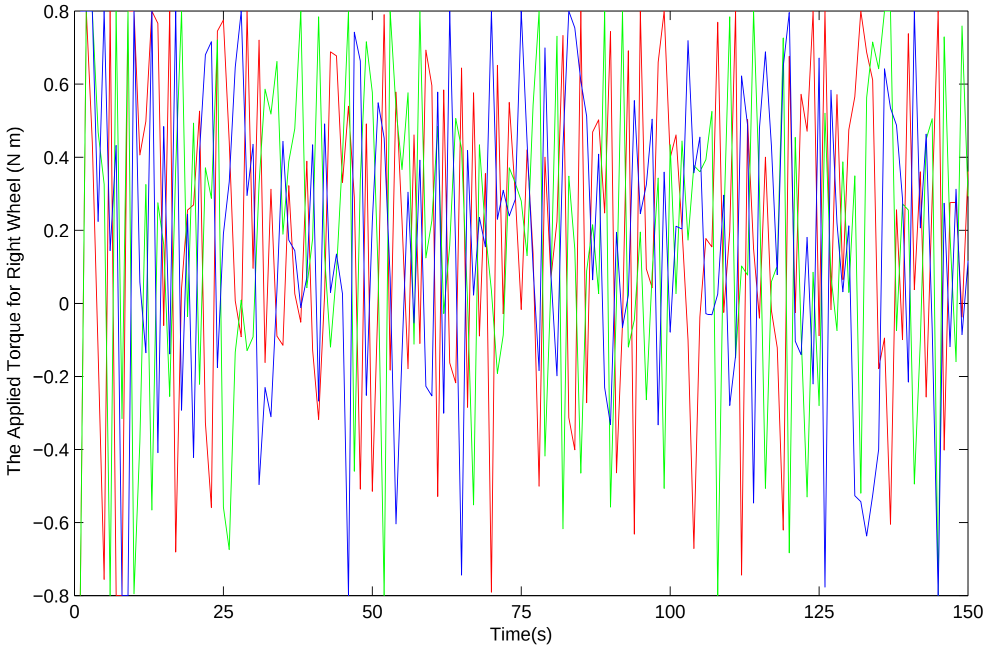


Figure 6.13: The change curve of the heading for parallel movement.



(a) Left Wheel



(b) Right Wheel

Figure 6.14: The change curves of the applied torques for left wheel and right wheel for the parallel movement.

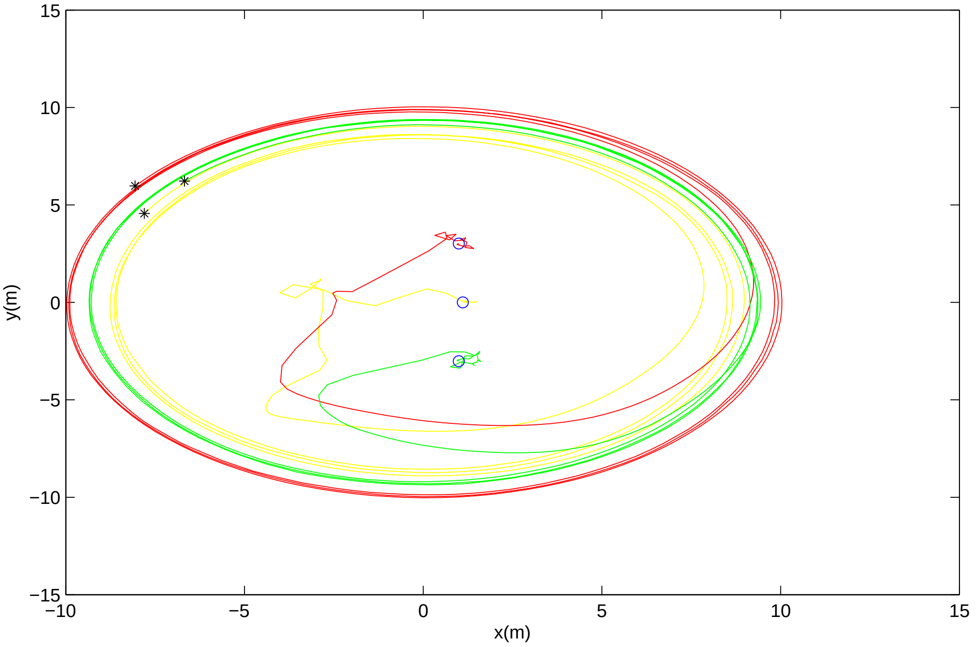


Figure 6.15: The circular movement for three robots where ‘o’ denotes the initial position and ‘*’ denotes the end position.

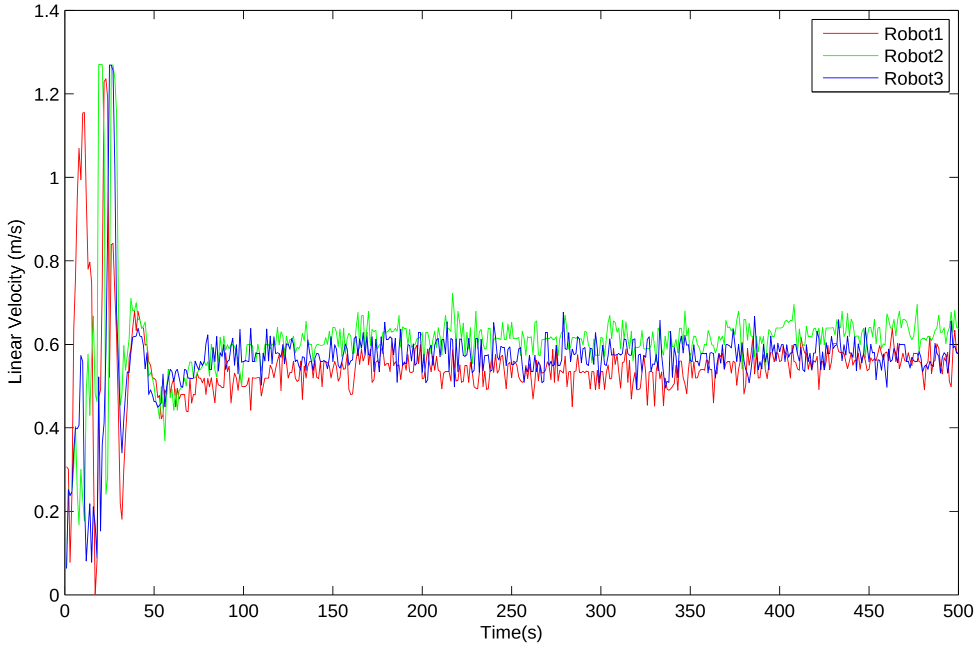


Figure 6.16: The change curve of linear velocity for circular movement.

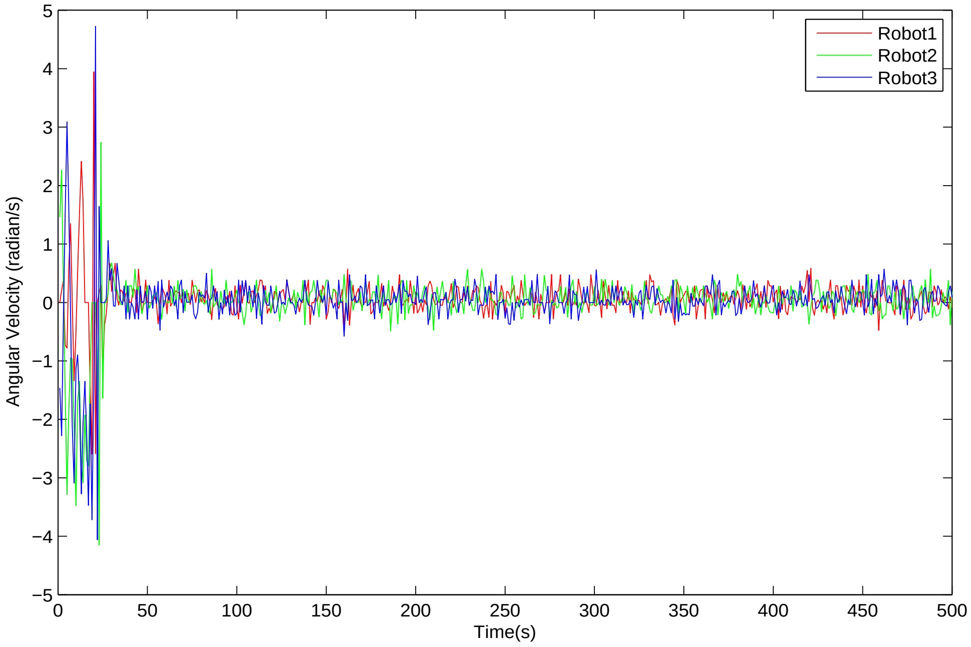


Figure 6.17: The change curve of angular velocity for circular movement.

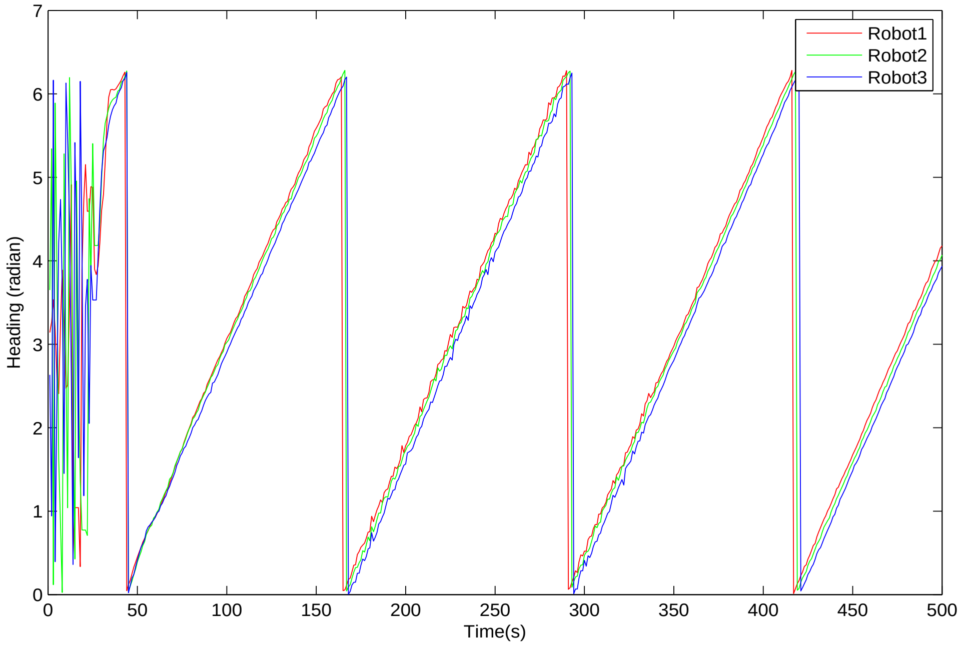
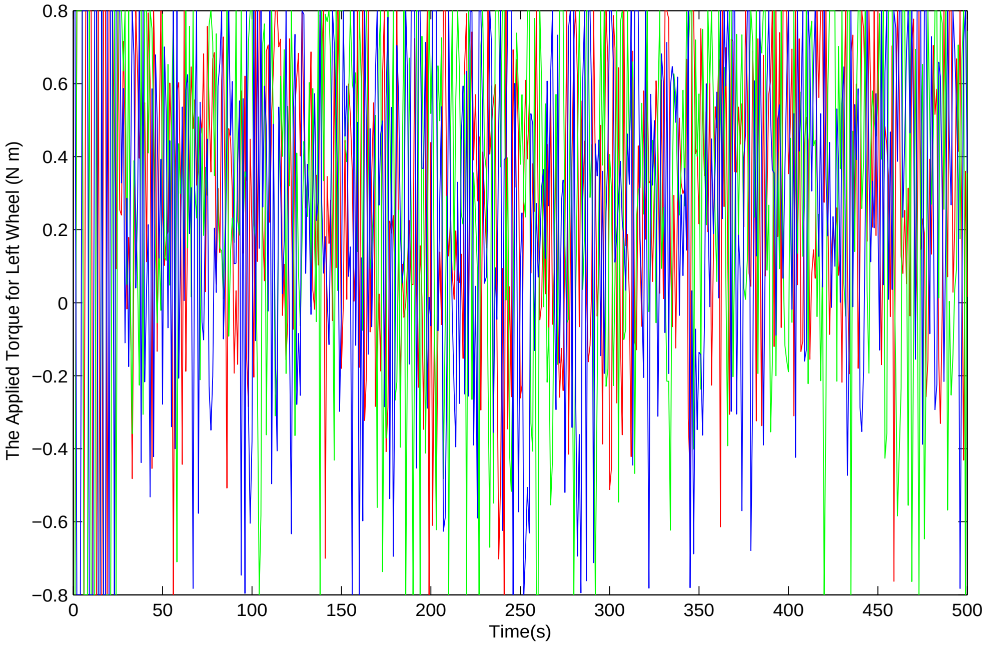
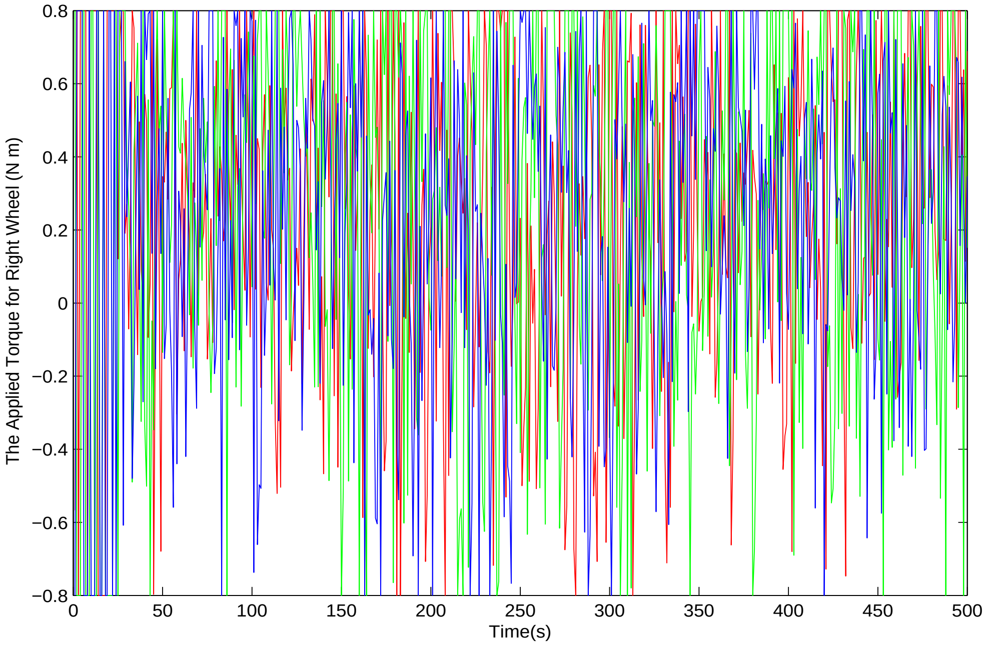


Figure 6.18: The change curve of the heading for circular movement.



(a) Left Wheel



(b) Right Wheel

Figure 6.19: The change curves of the applied torques for left wheel and right wheel for the circular movement.

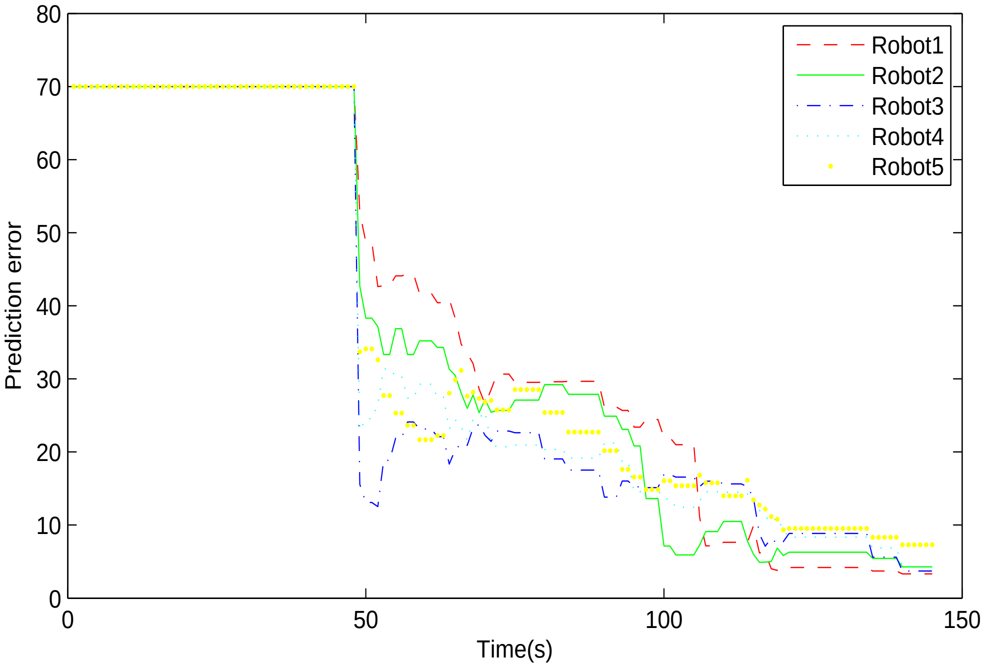
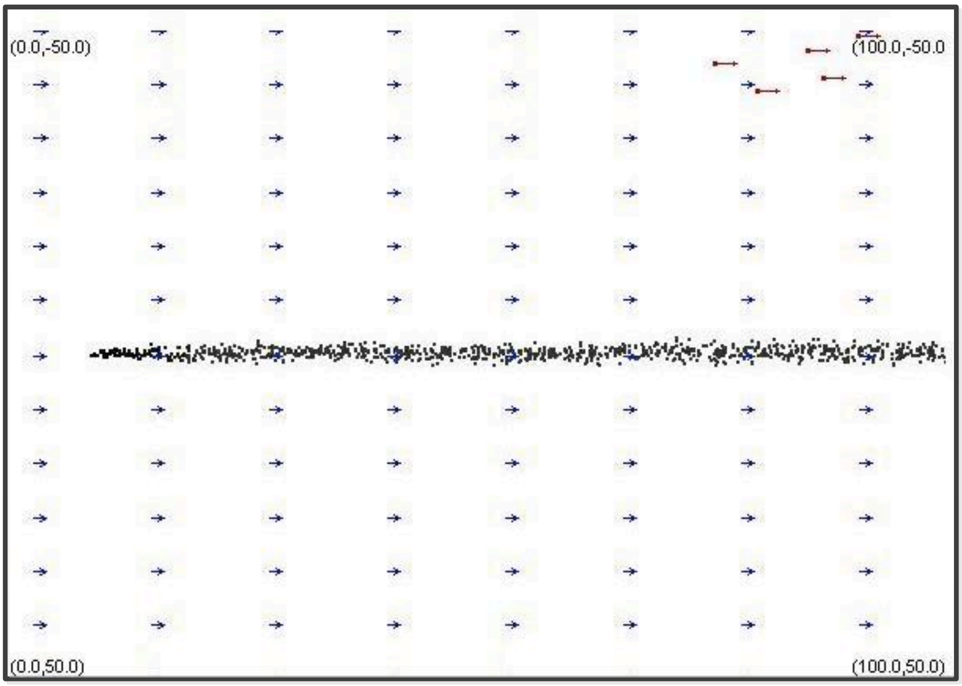
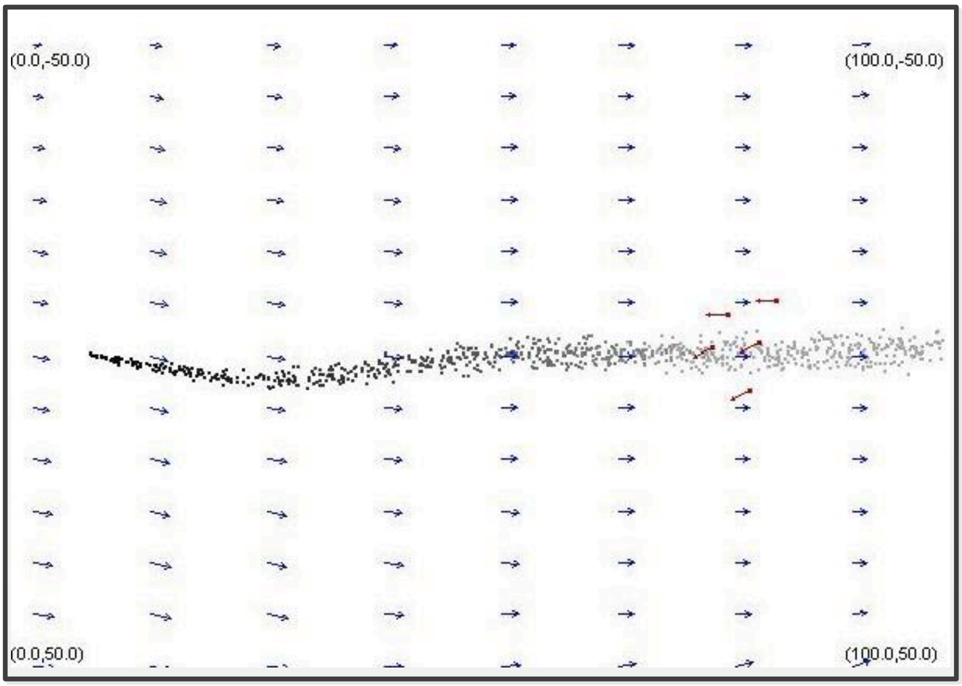


Figure 6.20: Prediction error $\|h^{ic} - x_s\|_2$ over time. x_s is a real position of the odour source and h^{ic} is estimate position of the odour source.

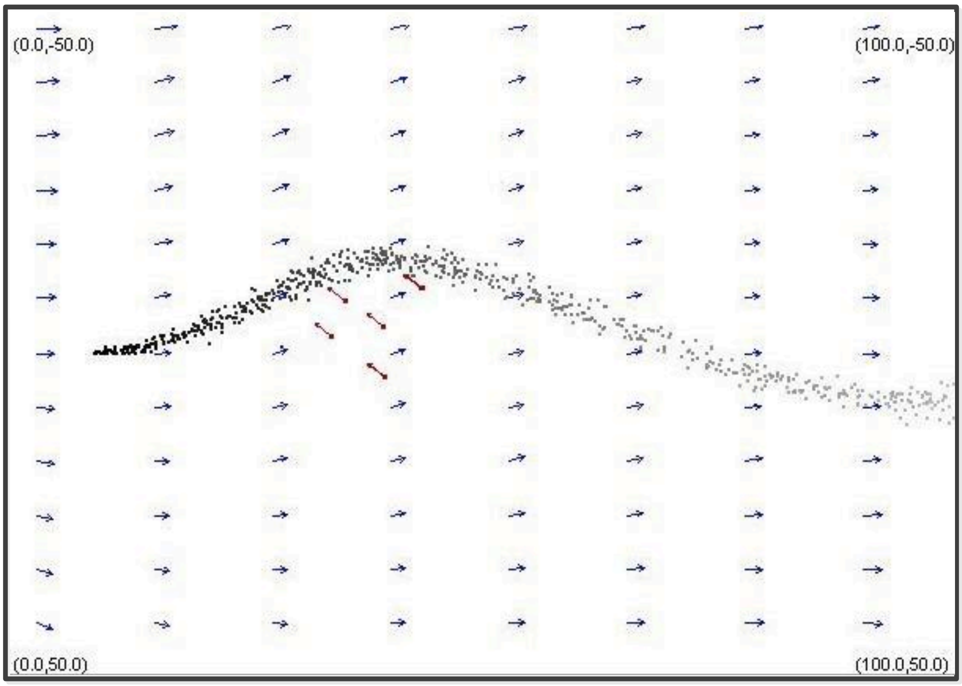


(a) T=0s

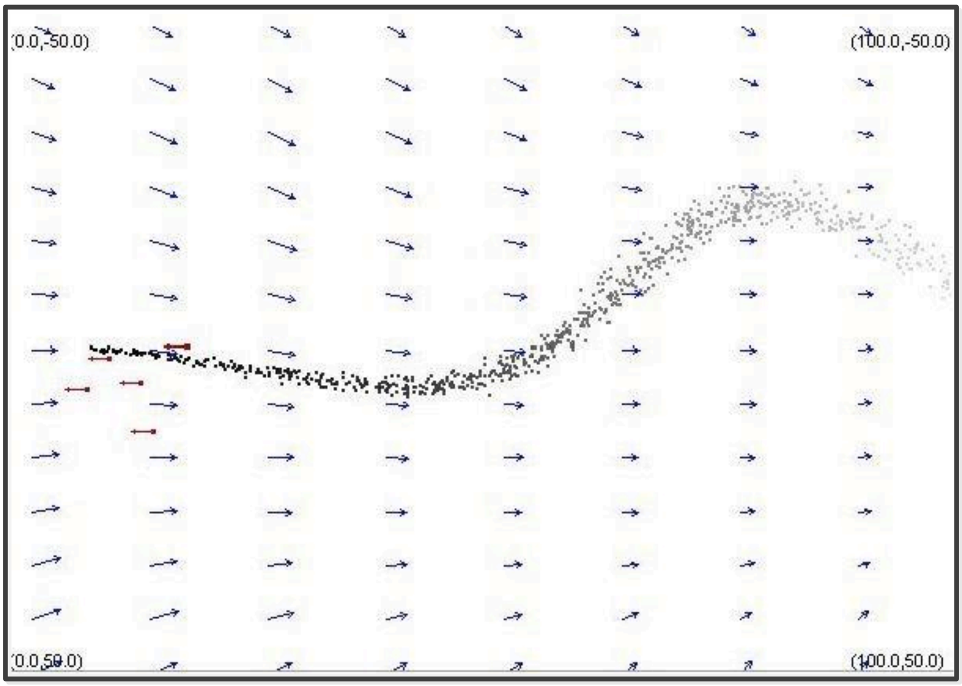


(b) T=57s

Figure 6.21: The search process of five robots at T=0s and T=57s.



(a) T=113s



(b) T=148s

Figure 6.22: The search process of five robots at T=113s and T=148s.

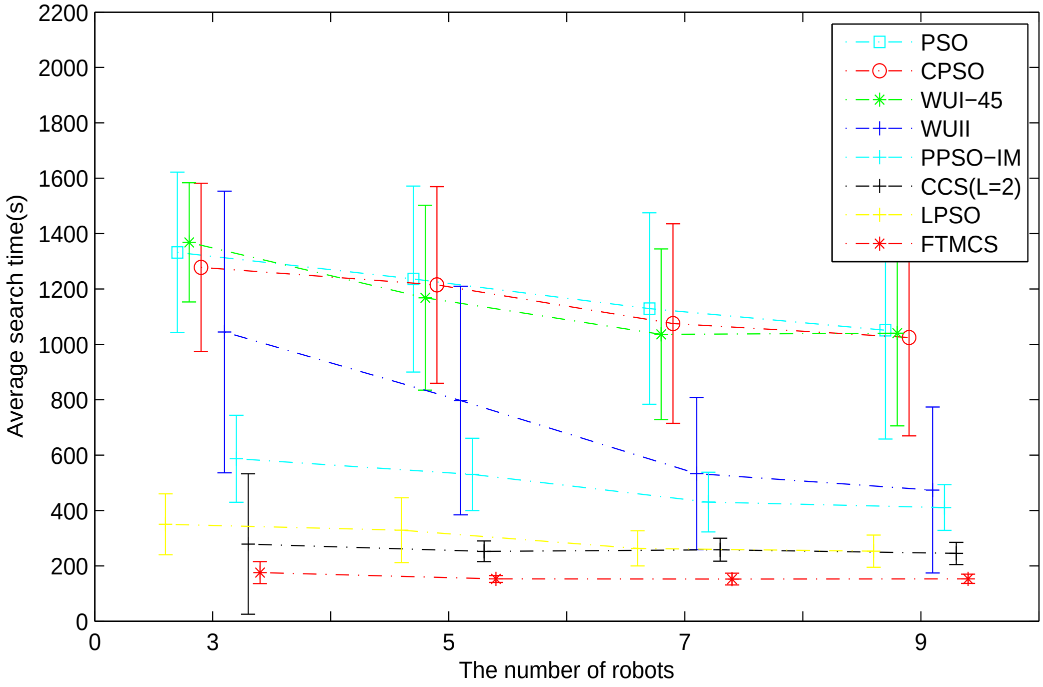


Figure 6.23: The average search time based on 50 runs.

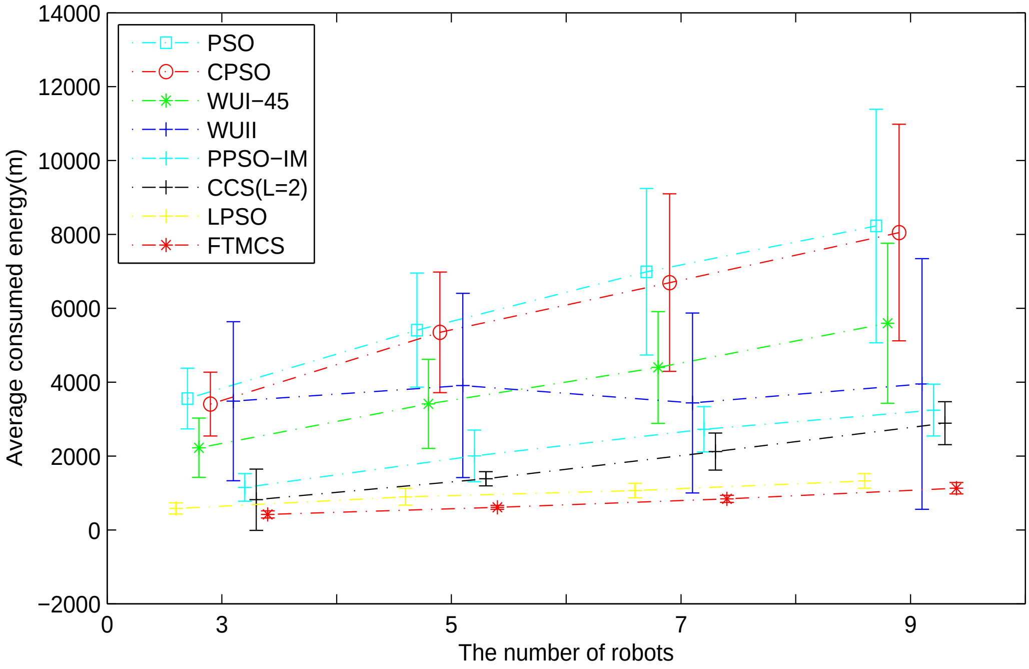


Figure 6.24: The average consumed energy based on 50 runs.

Chapter 7

Conclusions and Future Works

7.1 Conclusions

This dissertation has dealt with the problem of odour source localisation based on a multi-robot system. By analyzing PSO algorithms, we have derived a distributed coordination control architecture, which consists of two levels: a decision level and a control level. In the decision level, environmental information and individual information has been used to predict a probable position of the odour source. Moreover, the movement direction for each robot based on the predicted position of the odour source has been designed in order to trace the plume. In the control level, a cooperative control law has been developed to enable robots to move along the planned movement direction. Based on the distributed coordination control architecture, decision and cooperative control approaches can be separately designed. Consequently, we have developed a distributed decision algorithm and three cooperative control algorithms including a PSO-based finite-time motion control algorithm, a consensus-based finite-time motion control algorithm, and a potential-based finite-time motion control algorithm.

For the distributed decision algorithm, which can be used to make a decision on the position of the odour source, we have derived an observation model for the position of the odour source based on wind information and used a Kalman filter to estimate the probable position of the odour source. We have designed a dynamic finite-time consensus filter to fuse observations from other robots. In terms of fused observation, we have combined the position of the odour source obtained by using concentration information with that of the odour source obtained by using wind information in order to make a final decision on the position of the odour source. Moreover, we have planned the movement direction of the robot group in terms of the predicted position of the odour source in order to trace the plume.

For the PSO-based finite-time motion control algorithm, we have derived a continuous-time FPSO algorithm by introducing a nonlinear damping item and a parameter into the continuous model of the PSO algorithm such that the continuous-time FPSO algorithm can converge within a finite-time interval and its exploration capability can be improved. We have used a Lyapunov approach to analyze the finite-time convergence of the continuous-time FPSO algorithm, given a discrete-time FPSO algorithm by employing the same dicretization scheme as the GPSO algorithm, and analyzed the convergence of the discrete-time FPSO algorithm. We have shown the characteristics of the discrete-time FPSO algorithm through numerical simulations on the benchmark functions and the performance capabilities of the PSO-based finite-time motion control algorithm for the problem of odour source localisation.

For the consensus-based finite-time motion control algorithm, we have proposed a nonlinear finite-time consensus algorithm and used a Lyapunov approach to analyze the finite-time convergence of the proposed consensus algorithm. In terms of the proposed algorithm, we have derived a consensus-based finite-time motion control algorithm, which includes a finite-time parallel motion control algorithm and a finite-time circular motion control algorithm. On the basis of the distributed decision algorithm, we have shown the effectiveness of the consensus-based finite-time motion control algorithm for the problem of odour source localisation.

For the potential-based finite-time motion control algorithm, we have given an artificial potential function to provide obstacle avoidance capabilities for a group of robots. Based on the artificial potential function, we have proposed a finite-time coordination control algorithm and a finite-time tracking control algorithm. In terms of the two algorithms, we have derived a potential-based finite-time motion control algorithm, which includes a finite-time parallel motion control algorithm and a finite-time circular motion control algorithm. On the basis of the distributed decision algorithm, we have shown the effectiveness of the potential-based finite-time motion control algorithm for the problem of odour source localisation.

7.2 Future Works

As described in Chapter 1, the problem of odour source localisation is of practical significance for human security. Even if we have proposed several solutions for this problem in this dissertation, there still exist several works that need to be done in the near future.

- How to accurately estimate the position of the odour source by analyzing odour clues is still a challenging topic in our future works because of four main characteristics of the problem of odour source localisation.

- Several communication topologies such as the directed topology and the switching topology can occur when the wireless communication networks are used to exchange information among the multi-robot systems. Hence, how to analyze and design a stable and effective cooperative controller under these communication topologies is also a hard topic in our future works.

- A general framework for consensus has been proposed based on Lie group theory for the single integrator. How to develop a general framework for finite-time consensus such that a theory guidance for finite-time controller design can be provided is one of our future works.

- How to use our solutions to control the *real* robot group to deal with the problem of odour source localisation is an aim of our future works.

References

[1] S. Agrawal, Y. Dashora, M. K. Tiwari, and Y.-J. Son, "Interactive particle swarm: A pareto-adaptive metaheuristic to multiobjective optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, vol. 38, no. 2, pp. 258–277, March 2008.

[2] M. Arcak, "Passivity as a design tool for group coordination," *IEEE Transactions on Automatic Control*, vol. 52, no. 8, pp. 1380–1390, August 2007.

[3] N. Ayanian and V. Kumar, "Decentralized feedback controllers for multiagent teams in environments with obstacles," *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 878–887, October 2010.

[4] E. Balkovsky and B. I. Shraiman, "Olfactory search at high reynolds number," *Proceedings of the National Academy of Sciences USA*, vol. 99, no. 20, pp. 12 589–12 593, 2002.

[5] D. Bertsekas, *Dynamic Programming and Optimal Control.* Athena Scientific, 2000.

[6] S. P. Bhat and D. S. Bernstein, "Continuous finite-time stabilization of the translational and rotational double integrators," *IEEE Transactions on Automatic Control*, vol. 43, no. 5, pp. 678–682, May 1998.

[7] S. P. Bhat and D. S. Bernstein, "Finite-time stability of continuous autonomous systems," *SIAM Journal of Control and Optimization*, vol. 38, no. 3, pp. 751–766, 2000.

[8] T. Bretl, "Minimum-time optimal control of many robots that move in the same direction at different speeds," *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 351–363, April 2012.

[9] Y. Cao, W. Ren, and Z.-Y. Meng, "Decentralized finite-time sliding mode estimators and their applications in decentralized finite-time formation tracking," *Systems and Control Letters*, vol. 59, no. 4, pp. 522–529, 2010.

[10] Y. Cao, D. Stuart, W. Ren, and Z. Meng, "Distributed containment control for multiple autonomous vehicles with double-integrator dynamics: algorithms and experiments," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 4, pp. 929–938, July 2011.

[11] Y. Cao and L. Wang, "Sampled-data based consensus of continuous multiagent systems with time-varying topology," *IEEE Transactions on Automatic Control*, vol. 56, no. 5, pp. 1226–1231, May 2011.

[12] R. Carli, F. Fagnani, A. Speranzon, and S. Zampieri, "Communication constraints in the average consensus problem," *IEEE Transaction on Automatic and Control*, vol. 44, no. 3, pp. 671–684, 2008.

[13] W.-N. Chen, J. Zhang, Y. Lin, N. Chen, Z.-H. Zhan, H. S.-H. Chung, Y. Li, and Y.-H. Shi, "Particle swarm optimization with an aging leader and chanllengers," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 2, pp. 241–258, April 2013.

[14] J. Choi, S. Oh, and R. Horowitz, "Distributed learning and cooperative control for multi-agent systems," *Automatica*, vol. 45, no. 12, pp. 2802–2814, 2009.

[15] M. Clerc and J. Kennedy, “The particle swarm-explosion, stability and convergence in a multidimensional complex space,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 58–73, February 2002.

[16] T. Consi, F. Grasso, D. Mountain, and J. Atema, “Explorations of turbulent odour plumes with an autonomous underwater robot,” *Biological Bulletin*, vol. 189, no. 2, pp. 231–232, 1995.

[17] J. Cort*é*s, “Finite-time convergent gradient flows with applications to network consensus,” *Automatica*, vol. 42, no. 11, pp. 1993–2000, 2006.

[18] I. D. Couzin, J. Krause, N. R. Franks, and S. A. Levin, “Effective leadership and decision-making in animal groups on the move,” *Nature*, vol. 433, pp. 513–516, February 2005.

[19] C. David, K. Derek, B. Randal, M. Timonthy, and S. Li, “Cooperative forest fire surveillance using a team of small unmanned air vehicles,” *International Journal of Systems Science*, vol. 37, no. 6, pp. 351–360, 2006.

[20] D. V. Dimarogonas and K. J. Kyriakopulos, “On the rendezvous problem for multiple nonholonomic agents,” *IEEE Transactions on Automatic Control*, vol. 52, no. 5, pp. 916–922, May 2007.

[21] K. D. Do, “Flocking for multiple elliptical agents with limited communication ranges,” *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 931–942, September 2011.

[22] W. Dong and J. A. Farrell, “Cooperative control of multiple nonholonomic mobile agents,” *IEEE Transactions on Automatic Control*, vol. 53, no. 6, pp. 1434–1448, July 2008.

[23] M. R. D'Orsogna, Y. L. Chuang, A. L. Bertozzi, and L. S. Chayes, "Self-propelled particles with soft-core interactions: Patterns, stability and collapse," *Physical Review Letters*, vol. 96, no. 10, pp. 104 302–1–104 302–4, 2006.

[24] H. Du, S. Li, and C. Qian, "Finite-time attitude tracking control of spacecraft with application to attitude synchronization," *IEEE Transactions on Automatic Control*, vol. 56, no. 11, pp. 2711–2717, November 2011.

[25] W. B. Dunbar and R. M. Murray, "Distributed receding horizon control for multi-vehicle formation stabilization," *Automatica*, vol. 42, no. 4, pp. 549–558, April 2006.

[26] R. C. Eberhart and Y. Shi, "Parameter selection in particle swarm optimization," in *Proceedings of the 7th International Conference on Evolutionary Programming*, San Diego, CA, USA, 1998, pp. 591–600.

[27] S. Edwards, A. J. Rutkowski, R. D. Quinn, and M. A. Willis, "Moth-inspired plume tracking strategies in three-dimensions," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Barcelona, Spain, 2005, pp. 1669–1674.

[28] J. A. Farrell, J. Murlis, X. Z. Long, W. Li, and T. Ring, "Filament-based atmospheric dispersion model to achieve short time-scale structure of odour plumes," *Environment Fluid Mechanics*, vol. 2, no. 1-2, pp. 143–169, 2002.

[29] J. A. Farrell, S. Pang, and W. Li, "Plume mapping via hidden Markov methods," *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 33, no. 6, pp. 850–863, December 2003.

[30] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1465–1467, September 2004.

[31] J. L. Fernández-Martínez and E. García-Gonzalo, “The generalized PSO: A new door to PSO evolution,” *Journal of Artificial Evolution and Applications*, vol. 2008, pp. 1–15, 2008.

[32] J. L. Fernández-Martínez and E. García-Gonzalo, “Stochastic stability analysis of the linear continuous and discrete PSO models,” *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 3, pp. 405–423, June 2011.

[33] J. L. Fernández-Martínez, E. García-Gonzalo, and J. P. Fernández-Alvarez, “Theoretical analysis of particle swarm trajectories through a mechnical analogy,” *International Journal of Computational Intelligence Research*, vol. 4, no. 2, pp. 93–104, 2008.

[34] G. Ferri, E. Caselli, V. Mattoli, A. Mondini, B. Mazzolai, and P. Dario, “A biologically-inspired algorithm implemented on a new highly flexible multi-agent platform for gas source localization,” in *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, Pisa, Italy, 2006, pp. 573–578.

[35] G. Ferri, E. Caselli, V. Mattoli, A. Mondini, B. Mazzolai, and P. Dario, “Explorative particle swarm optimization method for gas/odour source localization in an indoor environment with no strong airflow,” in *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, Sanya, China, 2007, pp. 841–846.

[36] G. Ferri, E. Caselli, V. Mattoli, A. Mondini, B. Mazzolai, and P. Dario, “biologically-inspired algorithm for gas/odour source localization in an indoor environment with no strong airflow: First expertimental results,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, Rome, Italy, 2008, pp. 566–571.

[37] G. Ferri, E. Caselli, V. Mattoli, A. Mondini, B. Mazzolai, and P. Dario, “Spiral: A novel biologically-inspired algorithm for gas/odour source localization in an indoor environment with no strong airflow,” *Robotics and Autonomous Systems*, vol. 57, no. 4, pp. 393–402, 2009.

[38] R. A. Freeman, P. Yang, and K. M. Lynch, “Distributed estimation and control of swarm formation statistics,” in *Proceedings of the 2006 American Control Conference*, Minneapolis, Minnesota, USA, 2006, pp. 749–755.

[39] R. A. Freeman, P. Yang, and K. M. Lynch, “Stability and convergence properties of dynamic average consensus estimators,” in *Proceedings of the IEEE Conference on Decision and Control*, San Diego, CA, USA, 2006, pp. 398–403.

[40] V. Gazi, “Swarm aggregations using artificial potentials and sliding-mode control,” *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1208–1214, December 2005.

[41] V. Gazi and K. M. Passono, “Stability analysis of swarms,” *IEEE Transaction on Automatic Control*, vol. 48, no. 4, pp. 692–697, April 2003.

[42] S. Graham, G. Baliga, and P. R. Kumar, “Abstraction, architecture, mechanism, and a middleware for networked control,” *IEEE Transactions on Automatic Control*, vol. 54, no. 7, pp. 1490–1503, July 2009.

[43] F. W. Grasso, T. T. Consi, D. C. Mountain, and J. Atema, “Biomimetic robot lobster performs chemo-orientation in turbulence using a pair of spatially separated sensors: Progress and challenges,” *Robotics and Autonomous Systems*, vol. 30, no. 1, pp. 115–131, 2000.

[44] A. T. Hayes, A. Martinoli, and R. M. Goodman, “Distributed odour source localization,” *IEEE Sensors Journal*, vol. 2, pp. 260–271, 2002.

[45] A. T. Hayes, A. Martinoli, and R. M. Goodman, "Swarm robotic odour localization: Off-line optimization and validation with real robots," *Robotica*, vol. 21, no. 4, pp. 427–441, 2003.

[46] O. Holland and C. Melhuish, "Some adaptive movements of animats with single symmetrical sensors," in *Proceedings of the International Conference on Sumulation of Adaptive Behaviour*, Cape Cod, USA, 1996, pp. 55–64.

[47] Q. Hui, "Finite-time rendezvous algorithms for mobile autinomous agents," *IEEE Transactions on Automatic Control*, vol. 56, no. 1, pp. 207–211, January 2011.

[48] H. Ishida, Y. Kagawa, T. Nakamoto, and T. Moriizumi, "Odour-source localization in the clean room by an autonomous mobile sensing system," *Sensors and Actuators B*, vol. 33, no. 1-3, pp. 115–121, 1996.

[49] H. Ishida, T. Nakamoto, T. Moriizumi, T. Kikas, and J. Janata, "Plume-tracking robots: A new application of chemical sensors," *Biological Bulletin*, vol. 200, no. 2, pp. 222–226, 2001.

[50] H. Ishida, K. Suetsugu, T. Nakamoto, and T. Moriizumi, "Study of autonomous mobile sensing system for localization of odour source using gas sensors and anemometric sensors," *Sensors and Actuators A*, vol. 45, no. 2, pp. 153–157, 1994.

[51] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transaction on Automatic and Control*, vol. 48, no. 6, pp. 988–1001, June 2003.

[52] W. Jatmiko, P. Mursanto, B. Kusumoputro, K. Sekiyama, and T. Fukuda, "Modified PSO algorithm based on flow wind for odour source localization

problems in dynamic environments," *WSEAS Transactions on Systems*, vol. 7, no. 2, pp. 106–113, 2008.

[53] W. Jatmiko, K. Sekiyama, and T. Fukuda, "A PSO-based mobile robot for odour source localization in dynamic advection-diffusion with obstacles environment: Theory, simulation and measurement," *IEEE Computational Intelligence Magazine*, vol. 2, no. 2, pp. 37–51, May 2007.

[54] B. Johansson, A. Speranzon, M. Johansson, and K. H. Johansson, "On decentralized negotiation of optimal consensus," *Automatica*, vol. 44, no. 4, pp. 1175–1179, 2008.

[55] V. Kadirkamanathan, K. Selvarajah, and P. J. Fleming, "Stability analysis of the particle dynamics in particle swarm optimizer," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 245–255, June 2006.

[56] G. Kantor, S. Singh, R. Peterson, A. D. D. Rus, V. Kumar, G. Pereira, and J. Speletzer, "Distributed search and rescue with robot and sensor teams," in *Proceedings of the IEEE International Conference on Field and Service Robotics*, Tokyo, Japan, 2003, pp. 529–538.

[57] R. Kanzaki, N. Sugi, and T. Shibuya, "Self-generated zigzag turning of *Bombyx mori* males during pheromone-mediated upwind walking," *Zoological Science*, vol. 9, no. 3, pp. 515–527, 1992.

[58] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Network*, Perth, Australia, 1995, pp. 1928–1942.

[59] J. Kennedy and R. Eberhart, "A new optimizer using particle swarm theory," in *Proceedings of International Symposium on Micro Machine and Human Science*, Nagoya, Japan, 1995, pp. 39–43.

[60] J. Kennedy and R. Mendes, "Neighborhood topologies in fully informed and best-of-neighborhood particle swarms." *IEEE Transactions on System, Man, and Cybernetics- Part C: Application and Reviews*, vol. 36, pp. 515–519, 2004.

[61] T. Keviczky, F. Borrelli, and G. J. Balas, "Decentralized receding horizon control for large scale dynamically decoupled systems," *Automatica*, vol. 42, no. 12, pp. 2105–2115, 2006.

[62] T. Keviczky and K. H. Johansson, "A study on distributed model predictive consensus," in *Proceedings of the 17th IFAC World Congress*, Seoul, Korea, 2008, pp. 1516–1521.

[63] S. Khoo, L.-X. Xie, and Z.-H. Man, "Robust finite-time consensus tracking algorithm for multirobot systems," *IEEE/ASME Transactions on Mechatronics*, vol. 14, no. 2, pp. 219–228, April 2009.

[64] J. Krause, G. D. Ruxton, and S. Krause, "Swarm intelligence in animals and humans," *Trends in Ecology and Evolution*, vol. 25, no. 1, pp. 28–34, September 2009.

[65] M. J. B. Krieger, J. B. Billeter, and L. Keller, "Ant-like task allocation and recruitment in cooperative robots," *Nature*, vol. 406, no. 6799, pp. 992–995, 2000.

[66] Y. Kuwana and I. Shimoyama, "A pheromone-guided mobile robot that behaves like a silkworm moth with living antennae as pheromone sensors," *International Journal of Robotics Research*, vol. 17, no. 9, pp. 924–933, 1998.

[67] T. M. Lam, H. W. Boschloo, M. Mulder, and M. M. van Paassen, "Artificial force field for haptic feedback in uav teleoperation," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 39, no. 6, pp. 1316–1330, December 2009.

[68] J. R. T. Lawton, R. W. Beard, and B. J. Young, “A decentralized approach to formation maneuvers,” *IEEE Transactions on Robotics and Automation*, vol. 19, no. 6, pp. 933–941, December 2003.

[69] N. E. Leonard, D.A.Paley, F. Lekien, R. Sepulchre, D. M. Fratantoni, and R. E. Davis, “Collective motion, sensor networks, and ocean sampling,” *Proceedings of The IEEE*, vol. 95, no. 1, pp. 48–74, January 2007.

[70] N. E. Leonard and E. Fiorelli, “Virtual leaders, artificial potentials and coordinated control of groups,” in *Proceedings of the 40th IEEE Conference on Decision and Control*, Hawaii, USA, 2001, pp. 2968–2973.

[71] N. E. Leonard, T. Shen, B. Nabet, L. Scardovi, I. D. Couzin, and S. A. Levin, “Decision versus compromise for animal groups in motion,” *Proceedings of the National Academy of Sciences*, vol. 109, no. 1, pp. 227–232, January 2011.

[72] H. Levine, W.-J. Rappel, and I. Cohen, “Self-organization in systems of self-propelled particles,” *Physical Review E*, vol. 63, no. 1, pp. 1–4, 2000.

[73] S.-H. Li, H.-B. Du, and X.-Z. Lin, “Finite-time consensus for multi-agent systems with double-integrator dynamics,” *Automatica*, vol. 47, no. 6, pp. 1706–1712, 2011.

[74] T. Li, M. Fu, L. Xie, and J.-F. Zhang, “Distributed consensus with limited communication data rate,” *IEEE Transactions on Automatic Control*, vol. 56, no. 2, pp. 279–292, February 2011.

[75] W. Li, J. A. Farrell, and R. T. Cardé, “Tracking of fluid-advected odour plumes: Strategies inspired by insect orientation to pheromone,” *Adaptive Behavior*, vol. 9, no. 3-4, pp. 143–170, 2001.

[76] W. Li, J. A. Farrell, S. Pang, and R. M. Arrieta, "Moth-inspired chemical plume tracing on an autonomous underwater vehicle," *IEEE Transactions on Robotics*, vol. 22, no. 2, pp. 292–307, April 2006.

[77] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particles swarm optimization for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, June 2006.

[78] A. J. Lilienthal, D. Reiman, and A. Zell, "Gas source tracing with a moile robot using an adapted moth strategy," in *Proceedings of Autonomous Mobile System*, Stuttgart, Germany, 2003, pp. 150–160.

[79] Z.-Y. Lin and B. Francis, "Local control strategies for groups of mobile autonomous agents," *IEEE Transactions on Automatic Control*, vol. 49, no. 4, pp. 622–629, April 2004.

[80] T. Lochmatter, X. Raemy, L. Matthey, S. Indra, and A. Martinoli, "A comparision of casting and sprialing algorithms for odour source localization in laminar flow," in *Proceedings of IEEE International Conference on Robotics and Automation*, CA, USA, 2008, pp. 19–23.

[81] Q. Lu and Q.-L. Han, "A distributed coordination control scheme for odour source localization," in *Proceedings of the 36th Annual Conference of the IEEE Industrial Electronics Society*, Phoenix, AZ, USA, 2010, pp. 1413–1418.

[82] Q. Lu and Q.-L. Han, "Cooperative control of a multi-robot system for odour source localization," in *Proceedings of the 37th Annual Conference of the IEEE Industrial Electronics Society*, Melbourne, Australia, 2011.

[83] Q. Lu and Q.-L. Han, "Decision-making in a multi-robot system for odour source localization," in *Proceedings of the 37th Annual Conference of the IEEE Industrial Electronics Society*, Melbourne, Australia, 2011.

[84] Q. Lu and Q.-L. Han, "On designing a cooperative control system of multi-robot for odour source localization," in *Proceedings of the 18th World Congress of the International Federation on Automatic Control*, Milano, Italy, 2011, pp. 9446–9451.

[85] Q. Lu and Q.-L. Han, "A probability particle swarm optimizer with information-sharing mechanism for odour source localization," in *Proceedings of the 18th World Congress of the International Federation on Automatic Control*, Milano, Italy, 2011, pp. 9440–9445.

[86] Q. Lu, S.-R. Liu, and X.-R. Qiu, "A distributed architecture with two layers for odour source localization in multi-robot systems," in *Proceedings of the IEEE World Congress on Computational Intelligence*, Barcelona, Spain, 2010, pp. 153–159.

[87] Q. Lu and P. Luo, "A learning particle swarm optimization for odour source localization," *International Journal of Automation and Computing*, vol. 8, no. 3, pp. 371–380, August 2011.

[88] C. Lytridis, E. E. Kadar, and G. S. Virk, "A systematic approach to the problem of odour source localisation," *Autonomous Robot*, vol. 20, no. 3, pp. 261–276, 2006.

[89] C. Lytridis, G. S. Virk, Y. Rebour, and E. E. Kadar, "Odour-based navigational stratugies for mobile agents," *Adaptive Behaviour*, vol. 9, no. 3-4, pp. 171–187, 2001.

[90] D. Malakoff, "Following the scent of avian olfaction," *Science*, vol. 286, no. 5440, pp. 704–705, 1999.

[91] L. Marques, U. Nunes, and A. de Almeida, "Cooperative odour field exploration with genetic algorithms," in *Proceedings of Portuguese Conference on Automatic Control*, Aveiro, Portugal, 2002, pp. 138–143.

[92] L. Marques, U. Nunes, and A. de Almeida, "Odour searching with autonomous mobile robots: a evolutionary-based approach," in *Proceedings of the IEEE International Conference on Advanced Robotics*, Taipei, China, 2003, pp. 494–500.

[93] L. Marques, U. Nunes, and A. T. de Almeida, "Particle swarm-based olfactory guided search," *Autonomous Robots*, vol. 20, no. 3, pp. 277–287, 2006.

[94] J. A. Marshall, M. E. Broucke, and B. A. Francis, "Formations of vehicles in cyclic pursuit," *IEEE Transactions on Automatic Control*, vol. 49, no. 11, pp. 1963–1974, November 2004.

[95] S. Martinez, J. Cortes, and F. Bullo, "Motion coordination with distributed information," *IEEE Control Systems Magazine*, vol. 27, no. 4, pp. 75–88, August 2007.

[96] P. Massioni, T. Keviczky, E. Gill, and M. Verhaegen, "A decomposition-based approach to linear time-periodic distributed control of satellite formations," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 3, pp. 481–492, May 2011.

[97] S. Mastellone, P. Dorato, and C. T. Abdallah, "Finite-time stability of discrete-time nonlinear systems: Analysis and design," in *Proceedings of the 43rd IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, 2004, pp. 2572–2577.

[98] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: Simpler maybe better," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 204–210, June 2004.

[99] J. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Transactions on Automatic Control*, vol. 50, no. 2, pp. 169–182, February 2005.

[100] U. Munz, A. Papachristodoulou, and F. Allgower, "Robust consensus controller design for nonlinear relative degree two multi-agent systems with communication constraints," *IEEE Transactions on Automatic Control*, vol. 56, no. 1, pp. 145–151, January 2011.

[101] B. Nabet, N. E. Leonard, I. D.Couzin, and S. A. Levin, "Dynamics of decision making in animal group motion," *Nonlinear Science*, vol. 19, no. 2, pp. 399–435, 2009.

[102] K. Ogata, *Matlab for Control Engineers.* Upper Saddle River, New Jersey, USA: Prentice Hall, 2008.

[103] P. Ögren, M. Egerstedt, and X.-M. Hu, "A control lyapunov function approach to multiagent coordination," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 847–851, October 2002.

[104] P. Ögren, E. Fiorelli, and N. E. Leonard, "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *IEEE Transactions on Automatic Control*, vol. 49, no. 8, pp. 1292–1302, August 2004.

[105] R. Olfati-Saber, "Distributed kalman filter with embedded consensus filters," in *Proceedings of the IEEE Conference on Decision and Control*, Seville, Spain, 2005, pp. 8179–8184.

[106] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, March 2006.

[107] R. Olfati-Saber, J. A. Fax, and M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, January 2007.

[108] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, September 2004.

[109] R. Olfati-Saber and J. S. Shamma, "Consensus filters for sensor networks and distributed sensor fusion," in *Proceedings of the IEEE Conference on Decision and Control*, Seville, Spain, 2005, pp. 6698–6703.

[110] M. Omran and M. Clerc, "A modified particle swarm optimizer," in *SPSO2011-Particle Swarm Central*, http://www.particleswarm.info/Programs.html, 2011.

[111] S. Pang and J. A. Farrell, "Chemical plume source localization," *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 36, no. 5, pp. 1068–1080, October 2006.

[112] J. Pugh and A. Martinoli, "Multi-robot learning with particle swarm optimization," in *Proceedings of International Conference on Autonomous Agents*, Hakodate, Japan, 2006, pp. 441–448.

[113] J. Pugh and A. Martinoli, "Inspiring and modelling multi-robot search with particle swarm optimization," in *Proceedings of the IEEE Swarm Intelligence Symposium*, Hawaii, USA, 2007, pp. 332–339.

[114] J. Pugh and A. Martinoli, "Distributed adaptation in multi-robot search using particle swarm optimization," in *Proceedings of Simulation of Adaptive Behavior*, Osaka, Japan, 2008, pp. 393–402.

[115] J. Pugh and A. Martinoli, "Distributed scalable multi-robot learning using particle swarm optimization," *Swarm Intelligence*, vol. 3, no. 3, pp. 203–222, 2009.

[116] B. Ranjbar-Sahraei, F. Shabaninia, A. Nemati, and S.-D. Stan, "A novel robust decentralized adaptive fuzzy control for swarm formation of multiagent systems," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 8, pp. 3124–3134, August 2012.

[117] W. Ren, "Consensus based formation control strategies for multi-vehicle systems," in *Proceedings of the 2006 American Control Conference*, Minneapolis, Minnesota, USA, 2006, pp. 4237–4242.

[118] W. Ren and R. Beard, *Distributed Consensus in Multi-vehicle Cooperative Control - Theory and Applications.* Berlin: Springer-Verlag, 2008.

[119] W. Ren and N. Sorensen, "Distributed coordination architecture for multi-robot formation control," *Robotics and Autonomous Systems*, vol. 56, pp. 324–333, 2008.

[120] R. A. Russell, A. Bab-Hadiashar, R. L. Shepherd, and G. G. Wallace, "A comparison of reactive chemotaxis algorithms," *Robotics and Autonomous Systems*, vol. 45, no. 2, pp. 83–97, 2003.

[121] A. J. Rutkowski, S. Edwards, M. A. Willis, R. D. Quinn, and G. C. Causey, "A robotic platform for testing moth-inspired plume tracking strategies," in *Proceedings of the IEEE International Conference on Robotics and Automation*, New Orleans, USA, 2004, pp. 3319–3324.

[122] A. J. Rutkowski, R. D. Quinn, and M. A. Willis, "Biologically inspired self-motion estimation using the fusion of airspeed and optical flow," in *Proceedings of American Control Conference*, Minneapolis, USA, 2006, pp. 2712–2717.

[123] A. J. Rutkowski, R. D. Quinn, and M. A. Willis, "A sensor fusion approach to odour source localization inspired by the pheromone tracking behavior of moths," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Rome, Italy, 2007, pp. 10–14.

[124] A. J. Rutkowski, M. A. Willis, and R. D. Quinn, "Simulated odour tracking in a plane normal to the wind direction," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Irlando, USA, 2006, pp. 2047–2052.

[125] G. Sandini, G. Lucarini, and M. Varoli, "Gradient driven self-organising systems," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Yokohama, Japan, 1993, pp. 429–432.

[126] E. Semsar-Kazerooni and K. Khorasani, "Optimal consensus algorithms for cooperative team of agents subject to partial information," *Automatica*, vol. 44, no. 11, pp. 2766–2777, 2008.

[127] E. Semsar-Kazerooni and K. Khorasani, "Multi-agent team cooperation: A game theory approach," *Automatica*, vol. 45, no. 10, pp. 2205–2213, October 2009.

[128] R. Sepulchre, D. A. Paley, and N. E. Leonard, "Stabilization of plannar collective motion: All to all communication," *IEEE Transactions on Automatic Control*, vol. 52, no. 5, pp. 811–824, May 2007.

[129] R. Sepulchre, D. A. Paley, and N. E. Leonard, “Stabilization of planar collective motion with limited communication,” *IEEE Transactions on Automatic Control*, vol. 53, no. 3, pp. 706–719, April 2008.

[130] L. Shi, K. H. Johansson, and R. M. Murray, “Kalman filtering with uncertain process and measurement noise covariances with application to state estimation in sensor networks,” in *Proceedings of the IEEE International Conference on Control Applications*, Singapore, 2007, pp. 1031–1036.

[131] Y. Shi and R. C. Eberhart, “A modified particle swarm optimizer,” in *Proceedings of the IEEE Congress on Evolutionary Computation*, May 4-9, Anchorage, Alaska, USA, 1998, pp. 69–73.

[132] Y. Shi and R. C. Eberhart, “Fuzzy adaptive particle swarm optimization,” in *Proceedings of the IEEE Congress on Evolutionary Computation*, May 27-30,Seoul, Korea, 2001, pp. 101–106.

[133] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, “Kalman filtering with intermittent observations,” *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, September 2004.

[134] P. Stone and M. Veloso, “Multiagent systems: a survey from a machine learning perspective,” *Autonomous Robots*, vol. 8, no. 3, pp. 345–382, 2000.

[135] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger, and S. Tiwari, “Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization,” in *Nanyang Technological University, Singapore, KanGAL Report Number 2005005*, Singapore, 2005, pp. 1–50.

[136] S. Sundaram and C. H. Hadjicostis, “Finite-time distributed consensus in graphs with time-invariant topologies,” in *Proceedings of the 2007 American*

Control Conference, Marriott Marquis Hotel, New York City, USA, 2007, pp. 711–716.

[137] D. T. Swan, I. D. Couzin, and N. E. Leonard, "Real-time feedback-controlled robotic fish for behavioral experiments with fish schools," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 150–163, January 2012.

[138] A. Tahirovic and G. Magnani, "General framework for mobile robot navigation using passivity-based mpc," *IEEE Transactions on Automatic Control*, vol. 56, no. 1, pp. 184–190, Janauary 2011.

[139] N. J. Vickers, "Mechanisms of animal navigation in odour plumes," *Biological Bulletin*, vol. 198, no. 2, pp. 203–212, 2000.

[140] M. Wachowiak, R. Smolikova, Y. Zheng, and A. Elmaghraby, "An approach to multimodal biomedical image registration utilizing particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 289–301, June 2004.

[141] L. Wang and F. Xiao, "Finite-time consensus problems for networks of dynamic agents," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 950–955, April 2010.

[142] X.-L. Wang and Y.-G. Hong, "Finite-time consensus for multi-agent networks with second-order agent dynamics," in *Proceedings of the 17th World Congress-The International Federation of Automatic Control*, Seoul, Korea, 2008, pp. 15 185–15 190.

[143] A. Widyotriatmo and K.-S. Hong, "Navigation function-based control of multiple wheeled vehicles," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 5, pp. 1896–1906, May 2011.

[144] F. Xiao, L. Wang, J. Chen, and Y.-P. Gao, "Finite-time formation control for multi-agent systems," *Automatica*, vol. 45, no. 11, pp. 2605–2611, 2009.

[145] P. Yang, R. A. Freeman, and K. M. Lynch, "Multi-agent coordination by decentralized estimation and control," *IEEE Transactions on Automatic Control*, vol. 53, no. 11, pp. 2480–2496, December 2008.

[146] W.-C. Yeh, "Optimization of the disassembly sequencing problem on the basis of self-adaptive simplified swarm optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, vol. 42, no. 1, pp. 250–261, January 2012.

[147] W. Yu, G. Chen, and M. Cao, "Consensus in directed networks of agents with nonlinear dynamics," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1436–1441, June 2011.

[148] F. Zhang, D. M. Fratantoni, D. A. Paley, J. M. Lund, and N. E. Leonard, "Control of coordinated patterns for ocean sampling," *International Journal of Control*, vol. 80, no. 7, pp. 1186–1199, July 2007.

[149] F. Zhang, E. W. Justh, and P. S. Krishnaprasad, "Boundary following using gyroscopic control," in *Proceedings of the 43th IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, 2004, pp. 5204–5209.

[150] H. Zhang, F. L. Lewis, and Z. Qu, "Lyapunov, adaptive, and optimal design techniques for cooperative systems on directed communication graphs," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 7, pp. 3026–3041, July 2012.

[151] H.-T. Zhang, C. Zhai, and Z. Chen, "A general alignment repulsion algorithm for flocking of multi-agent systems," *IEEE Transactions on Automatic Control*, vol. 56, no. 2, pp. 430–435, February 2011.

[152] R. Zimmer and C. Butman, "Lobster orientation in turbulent odour plumes: simultaneous measurement of tracking behavior and temporal odour patterns," *Biological Bulletin*, vol. 187, no. 2, pp. 272–273, 1994.

[153] R. Zimmer and C. Butman, "Chemical signaling processes in the marine environment," *Biological Bulletin*, vol. 198, no. 2, pp. 168–187, 2000.