Chapter 9

Robot Swarms: Dynamics and Control

Veysel Gazi\textsuperscript{1}, Barış Fidan\textsuperscript{2}, Lino Marques\textsuperscript{3}, and Raul Ordonez\textsuperscript{4}

\textsuperscript{1}Dept. of Electrical and Electronics Engineering, Istanbul Kemerburgaz University, Mahmutbey Dilmenler Cad. No: 26, 34217 Bağcılar, Istanbul, Turkey.

\textsuperscript{2}University of Waterloo, Dept. of Mechanical and Mechatronics Engineering, ON N2L 3G1, Canada.

\textsuperscript{3}Dept. of Electrical and Computer Engineering, Institute of Systems and Robotics, University of Coimbra, 3030-290 Coimbra, Portugal.

\textsuperscript{4}Dept. of Electrical and Computer Engineering, University of Dayton, Dayton, Ohio, USA.

9.1 Introduction

The field of swarms and swarm robotics, and the more general field of multi-agent dynamic systems is an active research field which has been popular for more than two decades now. Since the pioneering work by Reynolds \cite{1} on simulation of a flock of birds in flight, extensive work has been done on modeling and analysis of the swarm behavior as well as development of swarm control and coordination algorithms. Collection of some of the related results can be found in the recent special journal issues \cite{2, 3}, edited books \cite{4–8} and monographs \cite{9–14} on swarm robotics and multi-agent dynamic systems.

A mobile \textit{robot swarm}, in general, can be defined as a network of mobile robots moving on a (two-dimensional) plane or a three-dimensional space to perform certain cooperative tasks. Within this context each individual robot in the swarm is considered as a dynamic \textit{agent}. There is no direct mechanical link between pairs of robot agents within the swarm, but rather some wireless sensing and communication links between certain assigned pairs. The main purpose of using mobile robot swarms is to collectively reach goals that are difficult to achieve by an individual robot or a monolithic robot system. A sample experimental robot swarm moving on plane is shown in Figure 9.1
The swarm robotics field is interdisciplinary by its nature and there are many aspects with various modeling and analysis approaches [16]. Many studies on swarms get their inspiration from nature and concepts like stigmergy [14], bio-mimicry [11, 14], and physico-mimicry [8] are getting increasingly popular. In this chapter, we focus on system dynamics and control aspects, and review studies on a set of coordination and control problems involving robot swarms.

The increased interest in multi-agent dynamic systems and in particular in swarms of robots is due to the rich portfolio of possible applications in various fields including agriculture, health, defense and others. Of particular current interest are applications involving teams of unmanned aerial, ground, space or underwater vehicles, robots, mobile sensors, automatic life-stock control units, etc. [17–26]. From mathematical and system theoretic perspective, these applications map to the problems of aggregation and foraging [27–29], formation control and coordinated tracking [30–32], distributed agreement and output synchronization [33–46], and, source seeking [47–55]. We provide further details of these problems in Section 9.3. Note that this is not an exhaustive list and in addition to the aforementioned problems, there exist also various more specific ones considered in the robot swarm literature, including coordinated search, deployment, map building, and olfactory navigation.

For the robot swarm coordination and control problems mentioned above, various multi-agent control approaches have been developed in the literature, including behavior based methods [56], feedback linearization [57], virtual structures [58], leader-follower structure [15, 59–63], artificial potential functions [64], graph theory based [15,61,65–68] methods, extremum seeking control [55]. The details of these approaches are given in Section 9.4.

The chapter is organized as follows: Section 9.2 presents the agent (individual robot) models that are most widely used in the literature. Various types of robot swarm coordination and control problems definitions are covered in Section 9.3. Section 9.4 presents different approaches in the literature to these problems. A summary and concluding remarks of the chapter are provided in Section 9.6.
9.2 Agent Dynamics

For a robot swarm coordination and control problem one needs to consider high level dynamic behavior of the whole swarm, including interactions between individual robot agents in the swarm, as well as the low level dynamic characteristics of the agents. There are various mathematical models which can be used to describe the dynamics of the individual agents. Here we consider a classification based on actuation constraints, and present holonomic (fully actuated) and non-holonomic (involving velocity constraints) agent models in Subsections 9.2.1 and 9.2.2, respectively. Later, in Subsection 9.2.3, we discuss introduction of simplified models for convenience of control design.

9.2.1 Fully Actuated Agent Model

A general form for a fully-actuated (holonomic) agent dynamics model can be represented as

\[
M(p_i)\ddot{p}_i + f_i(p_i, \dot{p}_i) = u_i, \tag{9.1}
\]

where \(p_i, \dot{p}_i, \ddot{p}_i, u_i \in \mathbb{R}^n\), and \(M(p_i) \in \mathbb{R}^{n \times n}\) denote the position, velocity, acceleration, control input, and mass (inertia) matrix of agent/robot \(i\), respectively. The term \(f_i(p_i, \dot{p}_i)\) represents the other effects (such as centripetal, Coriolis, gravitational effects, and additive disturbances). In case it is completely unknown, \(f_i\) can be thought of as the cumulative disturbance acting on the agent dynamics.

The fully-actuated model in (9.1) can be used to represent some omni-directional robots as well as some manipulators or spacecraft [69–72]. If \(f_i\) and \(M_i\) are known, with a control input of the form

\[
u_i = f_i(p_i, \dot{p}_i) + M(p_i)\bar{u}_i
\]

the model in (9.1) can be easily reduced to the point mass model

\[
\ddot{p}_i = \bar{u}_i \tag{9.2}
\]

Therefore, usually the researchers consider either (9.2) or assume that the agent dynamics in (9.1) contain uncertainties and disturbances. One usual assumption is that

\[
f_i(p_i, \dot{p}_i) = f^k_i(p_i, \dot{p}_i) + f^u_i(p_i, \dot{p}_i), 1 \leq i \leq N,
\]

where \(f^k_i(\cdot, \cdot)\) represents the known part and \(f^u_i(\cdot, \cdot)\) represents the unknown part. The unknown part is assumed to be bounded with a known bound, i.e.,

\[
\|f^u_i(p_i, \dot{p}_i)\| \leq \tilde{f}_i(p_i, \dot{p}_i), 1 \leq i \leq N,
\]

where \(\tilde{f}_i(p_i, \dot{p}_i)\) are known for all \(i\). This in a sense incorporates model uncertainties and additive disturbances in the model. Another usual assumption is that the mass/inertia matrices \(M_i(p_i)\) are unknown for all agents/robots \(i\). However, it is also assumed that they are lower and upper bounded by known bounds, e.g., they satisfy

\[
M_i\|y\|^2 \leq y^\top M_i(p_i)y \leq \bar{M}_i\|y\|^2, 1 \leq i \leq N.
\]
for any arbitrary $y \in \mathbb{R}^n$ and some known scalars $M_i$ and $\bar{M}_i$ satisfying $0 < M_i < \bar{M}_i < \infty$. Consideration of uncertainties and disturbances affecting the agent dynamics makes the agent model more realistic. To suppress such adverse effects one can use either robust or adaptive strategies. Sample robot swarm studies considering the fully actuated dynamics in (9.1) can be seen in [31, 32, 73, 74].

### 9.2.2 Non-Holonomic Agent Dynamics

Another agent/robot model that is commonly used in the literature is given by

\[
\begin{align*}
\dot{x}_i &= v_i \cos(\theta_i), \\
\dot{y}_i &= v_i \sin(\theta_i), \\
\dot{\theta}_i &= \omega_i, \\
\dot{v}_i &= \frac{1}{m_i} \left[ F_i + f_{v_i} \right], \\
\dot{\omega}_i &= \frac{1}{I_i} \left[ \tau_i + f_{\omega_i} \right],
\end{align*}
\]

(9.3)

where $p_i(t) = [x_i(t), y_i(t)]^\top \in \mathbb{R}^2$ denotes the position of agent $i$ at time instant $t$ in cartesian coordinates, $\theta_i$ is its steering angle, $v_i$ is its linear speed, and $\omega_i$ is its angular speed. These terms are graphically shown in Figure 9.2. The quantities $m_i$ and $I_i$ are positive constants and represent the mass and the moment of inertia of agent $i$, respectively. The control input for agent $i$ is $u_i = [F_i, \tau_i]^\top$ where $F_i$ is the force input and $\tau_i$ is the torque input.

The terms $f_{v_i}$ and $f_{\omega_i}$ in (9.3) represent modeling uncertainties and additive disturbances for agent $i$. These terms can be assumed to be known or unknown. In case they are known they can easily be compensated for with appropriate controller design. Therefore, in order to obtain more general and more practical results, usually they are assumed to be unknown but with known bounds. In other words, usually it is assumed that $|f_{v_i}| < f_v^+$ and $|f_{\omega_i}| < f_{\omega}^+$, for known bounds $f_v^+$ and $f_{\omega}^+$. Another common assumption is that the exact values of the mass $m_i$ and the inertia $I_i$ for agent $i$ are unknown. Still, as in the case for the inertia matrix $M_i(p_i)$ for the fully actuated model in (9.1), it is assumed that they are lower and upper bounded in the form $0 < M_i < m_i < \bar{M}_i$ and $0 < I_i < I_i < \bar{I}$, where the bounds $M, \bar{M}, I, \bar{I}$ are known.

The model (9.3) can be used to represent the dynamics of differentially driven mobile robots available in many research laboratories throughout the world. It can also be used to represent the dynamics of a unicycle and therefore sometimes is called the unicycle model. If the robots are moving
with low speed, the acceleration dynamics can be removed and the model can be simplified to a *kinematic* agent model to include only the first three equations

\[
\begin{align*}
\dot{x}_i &= v_i \cos(\theta_i), \\
\dot{y}_i &= v_i \sin(\theta_i), \\
\dot{\theta}_i &= \omega_i,
\end{align*}
\]  

(9.4)

with the linear speed \(v_i\) and the angular speed \(\omega_i\) as the control inputs, i.e., \(u_i = [v_i, \omega_i]^T\). The model (9.4) is widely used in the literature as well.

The agent dynamics in (9.3) (and those in (9.4)) have a non-holonomic velocity constraint. In particular, the agents cannot move along the axis connecting the two actuated wheels in Figure 9.2. This constraint can be expressed in the form

\[
\dot{x}_i \sin(\theta_i) - \dot{y}_i \cos(\theta_i) = 0,
\]

and needs to be taken into account in controller design. In fact, compared to controller development for the model (9.1), controller development for the model (9.3) is in general more difficult: It is known that \([75, 76]\) systems such as (9.3) “cannot be stabilized via continuously differentiable, time invariant, state feedback control laws.” This brings an important restriction leading to the fact that controllers stabilizing (9.3) have to be either discontinuous or time varying.

Depending on the available agent/robot dynamics one can usually use either the fully actuated holonomic model (9.1) or the non-holonomic model in (9.3). However, there are exceptional problems where both models can be employed simultaneously. For example, as noted in [77], the formation control problem can be stated as the control of systems having both non-holonomic (vehicle dynamics) and holonomic (formation and tracking) constraints.

Another issue to mention about the non-holonomic model is that since the angles are \(2\pi\) periodic it is possible to view \(\theta_i\) as a continuous variable taking values from \((-\infty, +\infty)\) or as a discontinuous variable taking values in an interval of length \(2\pi\) such as \([-\pi, \pi)\). The second approach requires also special attention to be taken for operations with angles. In particular, one can perform all addition operations on the angles (mod \(2\pi\)) with \(-\pi\) radians shift. For example, \(\theta_1 + \theta_2\) and \(\theta_1 - \theta_2\) can be calculated as \([(\theta_1 + \theta_2 + \pi)(\text{mod } 2\pi) - \pi]\) and \([(\theta_1 - \theta_2 + \pi)(\text{mod } 2\pi) - \pi]\), respectively. Similarly, \(\dot{\theta}_i(t)\) can be defined as

\[
\dot{\theta}_i(t) = \lim_{\Delta t \to 0} \frac{(\theta_i(t) - \theta_i(t - \Delta t) + \pi)(\text{mod } 2\pi) - \pi}{\Delta t}.
\]

Then, the corresponding control laws can be modified accordingly. It is possible to use various strategies in order to steer the robotic agents in a desired direction [78].

### 9.2.3 Simplified or High-Level Agent Models

The agent models discussed in the preceding subsections are realistic models representing the physical dynamics of agents/robots. They can be viewed as low-level agent dynamics models. In order to focus on the overall behavior of the swarm and simplify the analysis, some studies further simplify
or ignore the individual agents dynamics. In other words, there are studies using high-level models for describing the motion of the agents, and they do not consider the low-level agent dynamics. Such models can be thought of as tools for planning agent paths and generating way-points for the agents to visit in order to achieve the desired behavior. Usually it is easier to analyze the qualitative overall behavior of the swarm using such simplified models. However, for implementations on real robotic swarms one still needs to consider the low-level agent dynamics based on robots used for experimentation.

One possible simplification is the use of a single integrator model instead of the double integrator model in (9.2). In other words, the agents can be assumed to move based on

\[ \dot{p}_i = \ddot{u}_i \]  

(9.5)

and perform the analysis accordingly. Then the results obtained for the swarms composed of agents with dynamics obeying (9.5) can easily be adapted to swarms composed of agents with dynamics obeying (9.2) with inclusion of appropriate damping to the developed control inputs and without change in the overall qualitative behavior.

Other high-level or simplified models commonly used in the literature are discrete time models. Probabilistic Markovian models, evolutionary models, optimization based deterministic or nature inspired models are examples of such high-level discrete time representations [16]. Then, given the used model the overall swarm dynamics depend on the properties of the underlying motion and coordination algorithm and the relevant tools appropriate for the employed model/algorithm can be used to analyze the overall qualitative behavior.

### 9.3 Problem Definitions

The control theoretic studies on swarms are concerned with developing appropriate individual control laws \( u_i \) for the agents/robots which will lead to achieving a desired swarm behavior for the given agent/robot and swarm settings. Various behaviors have been studied in the literature. We discuss some of these behaviors in more detail below. More detailed investigations can be found in [10, 16]. Note that the problems presented here provide just a glimpse of the enormous potential research issues and agent behaviors, and the potential applications of swarms.

### 9.3.1 Aggregation and Social Foraging

Cooperative collective behavior usually occurs in aggregated swarms in nature. Therefore, aggregation is an important fundamental behavior for swarms in nature and therefore for swarm robotic systems. In swarms in nature, aggregation usually also occurs during social foraging, which is a behavior with many advantages such as increasing probability of success for the individuals [79, 80]. The aggregation and social foraging behaviors have been studied also in the engineering literature using various approaches.

Given a swarm of \( N \) agents with positions \( p_i(t) \) the objective in the swarm aggregation problem is
to design the control inputs $u_i(t)$ for every agent $i$ such that

$$\lim_{t \to \infty} \| p_i(t) - p_j(t) \| \leq \varepsilon, \ \forall i, j \in \{1, \ldots, N\}, i \neq j,$$

(9.6)

for some $\varepsilon > 0$ [10] (see Figure 9.3(a) for illustration). A commonly used approach to solve this problem is the potential function modeling [27, 29]. The studies in [27, 29] rigorously investigate the dynamics of an artificial potential function based model of swarm aggregation, and derive bounds on the swarm size for swarms composed of agents with single integrator dynamics. Double integrator agents are considered in [81, 82] with similar results to those in [27, 29] about the qualitative properties of the swarm dynamics. The article [31] considers swarms composed of agents with dynamics obeying the fully actuated model (9.1) with uncertainties and develops a sliding mode controller to achieve swarm aggregations and suppress the uncertainty effects.

Figure 9.3: Illustration of typical robot swarm coordination and control problems: (a) Aggregation. (b) Formation acquisition and maintenance.

Social foraging is similar to aggregation in the sense that usually aggregation also occurs during social foraging. However, it is also different since in social foraging the environment affects the motion dynamics of the individuals. The objective in social foraging is to increase the energy intake by the individuals and the swarm. Therefore, in swarms in nature the swarm moves towards regions with higher food/nutrient concentration (which can be referred to as favorable regions) and away from regions containing toxic or hazardous substances (which can be referred to as unfavorable regions). These concepts can be easily extended to swarm robotic systems such that favorable regions represent targets or goals, whereas unfavorable regions represent threats or obstacles.

In the light of the above observations given a swarm of $N$ agents with positions $p_i(t)$ and a function $\sigma : \mathbb{R}^n \to \mathbb{R}$ representing the environment (usually called the resource profile) the objective in the
social foraging problem is to design the control inputs $u_i(t)$ for every agent $i$ such that

$$\lim_{t \to \infty} \| p_i(t) - c_{\sigma_j} \| \leq \varepsilon_j, \forall i \in \{1, \ldots, N\},$$

(9.7)

for some $\varepsilon_j > 0$ and $c_{\sigma_j} \in \mathbb{R}^n$ such that $\sigma(c_{\sigma_j}) \leq \sigma(x)$ for all $x$ in some neighborhood of $c_{\sigma_j}$. In other words, defining the resource profile function $\sigma$ such that the regions with lower $\sigma$ values are more favorable than those with larger $\sigma$ values, the swarm is required to converge into the vicinity of a local minimizer of $\sigma$. Note that satisfaction of (9.7) by all the agents in the swarm results in satisfaction of (9.6) as well.

Social foraging for robot swarms using potential functions has been studied in [28, 83] using single integrator and double integrator dynamics with noise. Similarly, the works [84, 85] study aggregation as well as social foraging using double integrator dynamics. The article [85] considers an energy based Lagrangian approach for investigating swarm dynamics. It brings an alternative strategy for analyzing the dynamics of swarms and unifies the analysis strategies used for biological swarms and engineering multi-agent dynamic systems. The article [86] considers swarms composed of non-holonomic agent dynamics described by (9.3) and achieves similar qualitative results.

Inspired from social foraging of different species, researchers have developed various optimization algorithms. Examples of such algorithms include the ant colony optimization method [14], the particle swarm optimization method [10, 87, 88], and the bacterial foraging optimization method [10, 11, 89]. As mentioned in the preceding section, such algorithms can effectively be used also for high-level path planning for robot swarms for various applications such as cooperative search, contaminating source localization, chemical concentration mapping [90, 91] and others, in addition to being powerful, general-purpose optimization methods.

Probabilistic [92] and evolutionary [93] models and studies for swarm aggregations can also be found in the literature. In particular, probabilistic strategies can be combined with rule based approaches, with each rule being applied with some probability to achieve aggregation. Evolutionary strategies, on the other hand, can effectively be used to tune controllers with unknown parameters (such as neural network controllers) for achieving desired swarm behavior such as aggregation.

### 9.3.2 Formation Control and Swarm Tracking

Formation of geometric patterns by robot swarms is an important engineering problem. Such behavior can be seen in swarms in nature during cooperative behaviors such as migration, object transportation, and others. The formation control problem can be divided into various stages such as formation acquisition, formation maintenance, and formation reconfiguration (see Figure 9.3(b) for illustration). These can be viewed also as different behaviors exhibited by the swarm robotic systems. Formation acquisition is concerned with achieving a predefined geometric shape from any initial positions and orientations of the swarm members. Similarly, formation maintenance is concerned with keeping the acquired formation during motion of the swarm. Formation reconfiguration, on the other hand, is concerned with various behaviors such as splitting or joining formations or changing the geometric shape of the formation. Such behaviors or tasks might need to be performed by a swarm robotic system while performing higher level missions or goals.
To express the formation control problem more formally consider a swarm of $N$ agents with positions $p_i(t)$, the specific agent motion dynamics, and control inputs $u_i(t)$. Then the problem of formation control is concerned with designing the control inputs $u_i(t)$ for every agent $i$ such that

$$\lim_{t \to \infty} \| p_i(t) - p_j(t) \| = d_{ij}, \forall i \neq j \in \{1, \ldots, N\},$$

(9.8)

for a given set of desired inter-agent distances $\{d_{ij}|i, j \in \{1, \ldots, N\}, i \neq j\}$, where $d_{ij}$ denotes the desired distance between agents $i$ and $j$. Sometimes achieving/satisfying condition (9.8) exactly might be very difficult and small tolerance might be allowable. In such cases, the condition can be stated as

$$\lim_{t \to \infty} \| p_i(t) - p_j(t) \| - d_{ij} \leq \epsilon, \forall i \neq j \in \{1, \ldots, N\},$$

(9.9)

for some $\epsilon > 0$.

The above problem definitions involve only the agents and there are no external influences affecting the motion of the swarm. External environmental effects similar to those in the foraging problem can also be included in the definition of the model. Sometimes an external effect of particular interest is the case of external agents (agents not belonging to the swarm), which can represent friendly or hostile entities. Pursuit-evasion by the swarm and unwanted intruders under various scenarios are interesting research directions. Here we would like to emphasize the problem that we refer to as the swarm tracking problem, in which a swarm of agents is required to capture/enclose a moving target, achieve a geometric formation around the target, and continue tracking it (i.e., in a sense escort it) keeping the formation. It is related to the formation control problem in the sense that the condition (9.8) (or condition (9.9)) need to be satisfied. However, in addition to (9.8) or (9.9) the agent controllers are required to ensure that

$$\lim_{t \to \infty} p_T(t) \in \text{conv}\{p_1(t), \ldots, p_N(t)\},$$

(9.10)

is satisfied. Here $p_T(t)$ denotes the position of the mobile target and $\text{conv}\{p_1, \ldots, p_N\}$ denotes the convex hull of $p_1, \ldots, p_N$. Note that the swarm tracking behavior can be thought of as composed of two parallel subtasks - that of formation acquisition/maintenance and cooperative enclosing/tracking the maneuvering target.

The formation control and/or swarm tracking problems have been investigated in many works in the literature including [32, 74, 85, 86, 94, 95]. The work in [85] considers double integrator dynamics and uses an energy approach, the articles [32, 74] use the fully actuated model, whereas the works in [86, 94, 95] use the nonholonomic agent model for the individual agent dynamics. Other articles on target tracking and/or formation control of swarms of agents include [17, 57, 64, 96–99].

### 9.3.3 Source Seeking

Source seeking, or source localization, i.e., the design of control algorithms for autonomous agents to seek a source with unknown spatial distribution is of great interest, where for example techniques such as extremum seeking control [55] have been used in the design. Source seeking is an application having great theoretical interest, and it also has a significant impact on engineering applications: for instance, in the problem of developing vehicles with more autonomy, such as the situation where no
GPS information is available, or to reduce cost due to position sensors. Some of the direct applications of source seeking can be found in contaminant plume control, autonomous odor sensing or toxic gas leakage localization. Source seeking for fish models is discussed in [100, 101] and an overview of source seeking can be found in [102]. Sliding mode extremum seeking control for non-holonomic vehicle source seeking is investigated in [103, 104].

In the application of source seeking, the task of the vehicle is to find a source that emits a signal that decays as a function of distance away from the source, where the signal field is unknown and only the measurement of the signal at the current agent location is available.

A basic diagram of source seeking using extremum seeking control is provided in Figure 9.4, where the control goal is for the agent to seek an extremum of an unknown signal field based on the measurement of the signal only. Source seeking is first discussed for systems with moderately unstable single poles in [47], where the autonomous vehicle is modeled as a single or a double integrator, such as in (9.2). A formal study on source seeking for the unicycle model (9.4) first appeared in [48], where the design keeps the angular velocity constant and tunes the forward speed by extremum seeking, which generates triangle or star pattern vehicle motions that drift towards the source. A more challenging analysis is performed on a different strategy in [49], where the approach is to keep the forward speed constant and tune the angular velocity by extremum seeking. The resulting motion sinusoidally converges to the source and settles in a ring around the source as the forward speed is constant. Based on these two strategies, adding a simple derivative-like feedback to the forward speed in the angular velocity tuned by extremum seeking loop [52] allows the vehicle to slow down as it gets closer to the source and converges closer to the source without giving up convergence speed. Source seeking for slow or drifting sensors is explored in [53]. Rather than sinusoidal perturbations, a stochastic source seeking control law is used to tune the forward velocity in [105, 106] and the angular velocity in [107].

When source seeking is performed using a group of robots, the problem may become mathematically equivalent to swarm tracking, as described in Section 9.3.2. The main difference is conceptual, in that when performing swarm tracking together with, say, artificial potentials, the entire potential is typically assumed known. In contrast, in group source seeking (or swarm seeking), at least part of the potential function is assumed unknown, and is maximized or minimized collectively by the swarm. In this case, the maximization or minimization results in the localization of the unknown source, while also for instance attaining and maintaining a desired formation. Various strategies based on extremum seeking control have been explored in [55].
9.4 Control Design Approaches

There are various approaches for high-level control or path planning and low-level agent control in multi-agent dynamic systems. It is possible to design high-level path planning or high-level control strategies without considering the individual agent dynamics, although taking them into account can be also beneficial. In contrast, low-level control of individual agents by its nature deals with the agent dynamics. Since the interest in multi-agent dynamic systems is in the collective behavior of the agents the inter-agent distances play an important role and are of particular interest. In the preceding section, the swarm control problems are defined in terms of constraints on the inter-agent or agent-extremum point distances. Therefore, the developed control strategies need to take into account the distance requirements on the agents.

9.4.1 Artificial Potential Functions

Artificial potential functions are commonly used in multi-agent dynamic systems in order to specify inter-agent interactions and interactions of the agents with their environment. They can be used to construct attractive and repulsive forces between agents and between the agents and the environment [17, 31, 32, 86, 94, 99, 108, 109]. Artificial potentials can be thought of as the functions representing the forces on the agents due to the potential energy arising due to the positions of the agents relative to each other or relative to targets or threats in the environment [85]. In particular, these forces can be thought of as acting so as to decrease the overall potential energy, and thereby move the swarm towards a configuration with lower potential energy. Depending on the function that they perform the potential functions can be categorized in different ways including swarm potential, environmental potential, target or threat potential, and others [85]. Such potentials can be used to represent interactions in both biological swarms and swarm robotic systems. While the resultant behavior in biological systems needs to be biologically realistic no such requirement exists in engineering multi-agent dynamic systems. Instead, engineering swarms should safely satisfy other requirements/constraints such as achieving the system objectives or mission goals, including the problems defined in the preceding section.

The potential functions constitute, in a sense, a high-level control or path planning approach. Therefore, once the interaction requirements are specified by the potential functions and given the agent dynamics, low-level control algorithms need to be designed in order to force the agents to move based on the potential function requirements.

Aggregation and Social Foraging

The aggregation potential is usually composed of two terms - one for attraction and the other for repulsion between individuals. The attraction term dominates on long distances in order to keep the individuals together, whereas the repulsion term dominates on short distances in order to avoid collision when agents are in close proximity. The potential energy due to the aggregation potential can be represented in the form

\[
J_{agg}(p) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \left[ J_a(\|p_i - p_j\|) - J_r(\|p_i - p_j\|) \right]
\]  \hspace{1cm} (9.11)

where \( J_a : \mathbb{R}^+ \rightarrow \mathbb{R} \) represents the attraction component, whereas \( J_r : \mathbb{R}^+ \rightarrow \mathbb{R} \) represents the repulsion component. Various potentials can serve as aggregation potentials. One particular potential is the so-called Morse potential

\[
J_{agg}(p) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \left[ -a \exp \left( -\frac{\|p_i - p_j\|}{c_a} \right) + b \exp \left( -\frac{\|p_i - p_j\|}{c_r} \right) \right]
\] (9.12)

where \( a, b, c_a, c_r > 0, b > a, c_a > c_r \), and \( \left( \frac{b}{a} \right) \left( \frac{c_a}{c_r} \right)^2 < 1 \) are satisfied. There are many other possible potentials that can lead to aggregation behavior [27–29, 85, 86, 109–114].

In the case of social foraging, in addition to the interactions between individuals, interaction of the individuals with the environment needs to be represented as well. This can be done by introducing a resource profile to model the environment or simply an environmental potential \( J_{env} : \mathbb{R}^n \rightarrow \mathbb{R} \), which represents the potential energy due to the position of the agents in the resource profile. The resources in the resource profile include goals and targets that are attractive to the agents and obstacles and threats that are repulsive to the agents. An example of a resource profile is the multimodal Gaussian resource profile

\[
J_{env}(y) = -\sum_{i=1}^{M} \frac{A_{ei}}{2} \exp \left( -\frac{\|y - c_{ei}\|^2}{l_{ei}} \right) + b_e
\]

where \( c_{ei} \in \mathbb{R}^n, l_{ei} \in \mathbb{R}^+, A_{ei} \in \mathbb{R}, \forall i = 1, \ldots, M, \) and \( b_e \in \mathbb{R} \). See Figure 9.5 for example plot of such a profile. Depending on the properties of the environment, one may use other resource profile functions as well [28, 83, 85, 86]. Note that the potential function \( J_{env}(y) \) discussed here corresponds to the resource profile function \( \sigma \) mentioned in the preceding section. There could be also alternative non-potential function based methods for representing the environment.

Figure 9.5: Example resource profile (Multimodal Gaussian profile [28]).

Formation Control and Swarm Tracking

Potential functions can also be used to specify the inter-agent distances in the formation control problem and to achieve geometric formations. This can be achieved easily by a simple modification of the aggregation potential to have pair dependent attraction repulsion parameters depending on the desired distances between the pairs of agents in the geometric formation. For example, the Morse potential
(9.12) can be modified as
\[ J_{frm}(p) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \left[ -a_{ij} \exp \left( -\frac{\|p_i - p_j\|}{c_{aij}} \right) + b_{ij} \exp \left( -\frac{\|p_i - p_j\|}{c_{rij}} \right) \right] \]
where \( a_{ij}, b_{ij}, c_{aij}, c_{rij} > 0, b_{ij} > a_{ij}, c_{aij} > c_{rij} \), and \( \left( \frac{b_{ij}}{a_{ij}} \right) \left( \frac{c_{rij}}{c_{aij}} \right)^2 < 1 \) are satisfied. These parameters can have different values for different \((i, j)\) pairs, depending on their relative location in the desired geometric formation.

In the problem of swarm tracking the agents are required to track a moving target in addition to acquiring and maintaining a geometric formation. Therefore, in the case of swarm tracking the overall potential consists of two terms - one for formation control and one for target tracking. The target tracking potential can be a simple quadratic potential or any other attractive potential. One example of such potential is
\[ J_{tar}(p, p_t) = \sum_{i=1}^{N} \frac{1}{4} \|p_i - p_t\|^4, \] (9.13)
where \( p_t \) denotes the position of the target. Note that in this formulation there is an attraction to every agent from the target. Using a target tracking potential of the form (9.13), it is possible to show that the swarm will enclose the target as desired. In the mean time the formation potential will guarantee that the agents acquire and keep the desired geometric shape. In case there is a need to evade the target then the attractive potential can be replaced with a repulsive potential.

One issue to mention about potential functions is that they can suffer from the problem of existence of local minima. In other words, the dynamics can get stuck at a local minimum and convergence to a global minimum cannot be guaranteed. This can be critical in particular in the case of formation control and swarm tracking tasks, since achieving the desired geometric shape cannot be guaranteed globally. In that case, in order to achieve convergence to the desired formation the agents must start their motion with an initial configuration that is in the attraction region of the desired configuration. Nevertheless, there are also strategies such as simulated annealing, which can be used to overcome local minima. There are various studies in the literature considering the formation control or the swarm tracking problems using potential functions [29, 32, 85, 86].

### 9.4.2 Neighborhood Topologies
The neighborhood structure or topology is very important for the information flow in multi-agent dynamic systems [10, 13, 115]. It determines how quickly and how broadly the information is spread within the swarm. Therefore, the topology has an effect also on the stability and performance of the control algorithms. One possible neighborhood strategy is to use all-to-all interactions. However, as the number of agents increases or the operational area of the swarm gets larger, all-to-all interaction becomes impractical, or simply infeasible. Therefore, other structures such as nearest neighbors rules and leader-follower type structures have also been considered in the literature.

Two distinct neighborhoods can be defined for a given multi-agent system - one for sensing/communication between the agents and one for satisfying control/task constraints. In other words, the set of agents from which a given agent obtains information (either via sensing or communication) and the set of
agents with which it has to perform a common task (such as keep a distance) might be different. The interactions in both neighborhoods can be defined as unidirectional or bidirectional. Moreover, it is possible to have static and dynamic (i.e., time-varying) topologies.

A common approach to represent or model the neighborhood structure/topology in multi-agent dynamic systems is to use tools from algebraic graph theory. This approach can be used together with potential functions or with other methods alike. Introducing graphs to model neighborhoods allows usage of techniques from graph theory for analyzing the connectivity properties of the swarm and derivation of conditions for the necessary or sufficient information flow for achieving the desired task.

9.4.3 Gradient Based, Lyapunov, and Sliding Mode Methods

Many approaches using potential functions and in particular the extremum seeking control strategies utilize the gradient of the potential function (or the related other functions) in order to minimize (or maximize) the potential (the objective function). Since the potential functions constitute a high-level method for modeling agent interactions and path planning, it is easier to develop the control/motion strategy assuming that the motion dynamics of the agents are governed by the point-mass single integrator model in (9.5) where $\bar{u}_i \in \mathbb{R}^n$ is the control input for the $i^{th}$ agent. This allows the designer to focus on the higher-level task of coordinated control design. Then the obtained results for the single integrator model in (9.5) can be extended to more realistic models such as the double integrator point-mass dynamics in (9.2) and the non-holonomic agent dynamics in (9.3) using techniques such as phase lead compensator design [47] and sliding mode control [94].

The potential function, which can be viewed as a signal to be tracked by the agents, can have many unknown minima. Its value/strength is assumed to be measured by the agents. Based on the application this in a sense means that the agents can measure their distance to their neighbors, their distance to a possibly moving target, the value of the environment/resource profile (i.e., their distance to the obstacles/threats and/or targets/goals in the environment), etc. Then the purpose of the design is to minimize the potential and to achieve the desired goal, which can be aggregation, social foraging, source seeking, formation control, collision avoidance, obstacle avoidance, etc.

As mentioned in the preceding sections, depending on the application, the potential function can be composed of several components. For example in the swarm tracking problem, it can be composed of an inter-agent interaction component and a tracking component. The inter-agent interaction component puts a constraint on the agents, based on their neighbors’ positions, in order to maintain a group structure and includes functions of the relative distance between each pair of neighbors. Its specific form is defined according to the desired geometric formation. The tracking component, on the other hand, contains constraints with respect to the target(s) or the source(s) of the particular tracking problem. It is typically defined in terms of a scalar signal to be tracked in order to direct the group’s behavior for target tracking or source seeking. This signal can be an artificial potential function given the knowledge of target position, or an actual signal depending on the particular application, such as concentration of a chemical source, an electromagnetic signal emitted by the source of interest, an acoustic signal or a thermal diffusion field.

The choice of the potential function is important in the control design, because different potentials
might result in different performances even with the same control algorithm. In particular, existence of multiple local minima in the potential function results in only being able to guarantee local convergence to the desired formation. Nevertheless, it is possible to show that by appropriate choice of the potential function one can always guarantee that eventually the target will be surrounded or “enclosed” by the tracking agents.

It is possible to combine the potential functions using different methods. The simplest method is to take a linear combination of the potential functions for the subtasks. For example, for the swarm tracking problem the overall potential function can be combined as

$$J(p, p_t) = K_{tar}J_{tar}(p, p_t) + K_{frm}J_{frm}(x)$$

(9.14)

where $K_{tar} > 0$ and $K_{frm} > 0$ are the weights of the potential components.

A block diagram of the overall control structure for the multi-agent dynamic system can be found in Figure 9.6 [55]. In the scheme in Figure 9.6 the extremum seeking controller is assumed to be a gradient based controller although it can be any controller. Then with appropriate choice of the potential functions if the extremum seeking controller for each agent can minimize the performance function (9.14), then we can achieve our objective of source seeking, formation control and collision avoidance (this objective is achieved by adding a repulsive potential between the agents and obstacle into (9.14)). This control design will be decentralized if each agent has its own performance function.

If the velocity $\dot{p}_t$ of the target is known then for every agent $i$ the control input in the form

$$\bar{u}_i = \dot{p}_t - k_i \nabla p_i J(p, p_t),$$

(9.15)

where $k_i > 0$ guarantees that the Lyapunov function $V = J(p, p_t)$ satisfies

$$\dot{V} = -\sum_{i=1}^{N} k_i \| \nabla p_i J(p, p_t) \|^2 \leq 0.$$

(9.16)
ically converges to values for which \( \| \nabla_p J(p, p_t) \| = 0 \) and \( \| \nabla_p J(p, p_t) \| = 0 \). Moreover, it is also possible to show that as \( t \to \infty \) we will have

\[
p_t \to \text{conv}\{p_1, p_2, \ldots, p_N\},
\]

where \( \text{conv}\{p_1, p_2, \ldots, p_N\} \) is the convex hull of the positions of the agents. In other words, the agents will “surround” or “enclose” the target.

In case \( \dot{p}_t \) is not known then under the assumption that \( \| \dot{p}_t \| \leq \gamma_t \) for some known \( \gamma_t > 0 \) the controller

\[
\ddot{u}_i = -k_i \nabla_p J(p, p_t) - \beta_i \text{sgn}(\nabla_p J(p, p_t)) \tag{9.17}
\]

where \( k_i > 0, \beta_i > \gamma_t \) and \( \text{sgn}(\cdot) \) is the signum function guarantees again that (9.16) is satisfied.

If only the sign of the gradient \( \nabla_p J(p, p_t) \) is known the controller can further be relaxed to

\[
\ddot{u}_i = -(k_i + \beta_i) \text{sgn}(\nabla_p J(p, p_t)), \tag{9.18}
\]

guaranteeing the same result.

### 9.4.4 Adaptive Control Approaches

Various adaptive techniques have been utilized for formation control and coordination in multi-agent dynamic systems [116–124], mainly for estimation and/or suppression of uncertainties and unknown disturbances present in the system dynamics.

When the uncertainties are parametric or the disturbance can be expressed in a parametric form, various identifier based adaptive control techniques [125] can be applied. In contrast, if it is not possible to properly parameterize the uncertainties and/or the controller, then it is possible to use universal approximators such as polynomial based ones, neural networks, and fuzzy systems to estimate and to cancel the uncertainty effects [126–131].

A particular application of the identifier based adaptive control for cooperative target localization and tracking by robot swarms is presented later in Subsection 9.5.3. Next we summarize another series of studies on adaptive formation control using universal approximation and fuzzy techniques, as an example application of this approach: In [132], a distributed adaptive fuzzy control scheme was developed for formation control and target tracking, in the existence of nonlinear and uncertain agent dynamics which are transformable to normal form. This scheme requires the agents to know not only their relative positions to the other agents and the target, but also the time derivatives of these positions up to the order of their relative degrees. Later in [133], this requirement was relaxed by employing high gain observers to estimate the derivatives of the combined agent formation and tracking errors. In [133], application of the revised scheme involving high gain observers was demonstrated for various formation maneuvers such as contraction/expansion, rotation, and reconfiguration of the formation. The overall adaptive scheme has involved sliding mode rules and bounding terms as well, in order to guarantee boundedness and asymptotic stability of the formation and tracking errors.
9.4.5 Other Nonlinear Methods

In addition to the methods mentioned in the previous subsections, there exist various other nonlinear control techniques used in coordination and control of robot swarms. Among the most popular and promising approaches, we see feedback linearization [57,134,135], neural network [136,137], model prediction [134,138,139], output regulation (linear or nonlinear servomechanism) [7,30,39–46,140–143], and passivity [9,119] based ones.

9.5 Swarm Robotic Applications

Although no commercial application using swarm robotics approaches exists so far, these approaches have been proposed in the past fifteen years or so for a wide variety of potential applications and a corresponding number of research projects have demonstrated their applicability, either with physical or with simulated robots for all domains: ground [144], aerial [145], and marine [146] (the provided references contain examples to the respective domains).

Generally speaking, we can say that swarm robotics concepts are appropriate for spatially distributed tasks benefiting from easy and robust scalability, where adding or removing elements to the swarm increases or decreases respectively its performance, but does not compromise the global ability to perform the task. Examples of such tasks are static or dynamic coverage and transportation. Static coverage is the problem of positioning robots in a stationary configuration that maximizes the probability of detecting an event of interest. Usually, in this type of coverage every point in the environment is under the robots’ detection range (i.e. covered) at every instant of time. A related problem is the deployment of communication nodes in a sensor network guaranteeing that all nodes are reachable (i.e., under communication range). Static coverage is particularly relevant for monitoring dynamic environments or intermittent events that may happen anywhere and anytime, requiring global coverage. References [147–149] provide some proposals to solve this problem.

Dynamic coverage, on the other hand, is the problem of sequentially covering all the workspace. This is particularly relevant for monitoring static or slowly changing environments with scarce resources, being physically impossible to cover the entire environment with the existing robots [150]. The dynamic coverage of unknown environments is often called exploration [151] while a dynamic coverage aiming to visit regularly different places of the environment is called patrolling [152]. Exploration with swarm robotic systems is frequently accomplished with dispersive behaviors or by means of formations that increase the global sensing width. Both types of coverage may be applied to areas (area coverage) or to area boundaries (boundary coverage).

Searching for a given target inside an environment may require to sense all points of that environment (i.e., exploring the environment) or it may be a faster task if the target provides some kind of long range clues that may be used to guide the searching agents (e.g., a chemical source generating a chemical plume) [153]. If the found targets should be transported to a given collecting point, then we will be talking about a foraging problem [154]. Some items may not be transportable by a single robot and require the cooperation of multiple ones. This raises the problem of coordinating individuals in order to perform collective transportation [155].
9.5.1 Static Coverage

The static area coverage by a robot swarm in order to optimally detect some kind of intermittent event or source of energy or chemical requires the positioning of the swarm elements in a given formation. This formation may correspond to a regular pattern for omnidirectional sensors positioned in open environments or to distorted patterns for directional sensors or environments with avoiding areas such as obstacles. Example applications for static coverage are the maintenance of the communication infrastructure, environmental and wild-life monitoring, detection of chemical leaks, hot-spots, sound [156] or radio frequency sources, and surveillance. Most of these applications have been addressed within a sensor network framework (i.e., fixed nodes), but some works employed a robotic swarm approach, allowing the adjustment of proper formations to changing environments or swarm disturbances.

The SMAVNET project provides an example of static coverage applied to real-world applications. This project aims to use swarms of Unmanned Aerial Vehicles (UAVs) in disaster scenarios to create and maintain a communication network for first responders. The motivation behind flying robots was their mobility and the benefit of providing line-of-sight communication [157]. A very different example is the detection of chemical traces. Odor sources release chemical substances that are transported by the fluid flow and dispersed by turbulent phenomena, generating a flow-biased Gaussian plume [153] whose orientation may change in time. The optimal detection of such plumes may be achieved by the positioning of a sensor array aligned with the flow, at lateral and longitudinal distances that depend from the flow speed, turbulence and sensors’ sensitivity. A swarm equipped with sensors to measure such variables may adjust itself to be always optimally positioned to detect a chemical release inside the workspace [158]. Another interesting approach is changing the swarm distribution in order to capture a monitored phenomenon with different levels of detail, minimizing the global uncertainty. Lynch et al. [159] considered the problem of modeling environmental fields such as the temperature or salinity of a region of an ocean. The authors propose a scalable and decentralized approach to fuse sensor data into a global model of the environment. Additionally, the mobile sensors are moved to maximize their sensory information relative to current uncertainties in the model. On the other hand, the most common situation is dealing with static environments and more or less radial phenomena. Cortes et al. [149] surveyed the coverage with radial sensors and proposed an asynchronous distributed implementation for coverage control with mobile sensing networks. Dirafzoon and Lobaton [160] employed a minimalist approach to infer the topology of an environment using agents without explicit localization abilities. The agents moved at constant velocity, following walls to explore the environment (like cockroaches). When pair of agents interacts near a wall, that interaction contains information allowing inferring global topological information of the physical environment (i.e., a map).

9.5.2 Dynamic Coverage

In a dynamic coverage framework, the swarm has no physical capacity to cover all the workspace at once, covering it sequentially according to some criteria. This is a common approach when searching stationary targets across large search spaces with small detection width sensors (e.g., for de-mining [161]) or small actuators (e.g., for cleaning [162]).

Some common examples of dynamic area coverage are floor and ocean cleaning, de-mining, and inspection of structures. Rutishauser et al. [163] studied and proposed an on-line coverage algorithm
for inspection of turbine blades with a swarm of miniature robots (less than one cubic inch). Larionova et al. [162] shown cooperative floor cleaning with mobile robots equipped with smell sensors. The robots sensed cleaning chemicals released from the ground, being able to automatically coordinate their motion in order to perform complete coverage of dirty areas without explicit or centralized coordination. A related approach, but validated only in simulation, was proposed by Jin and Ray for oil spill cleaning in a harbor environment [164].

Perimeter detection and boundary coverage approaches are also frequently employed in oil spill monitoring and other applications involving well defined frontiers, such as forest fire surveillance [165], crop monitoring, and border or facility surveillance [166]. After some severe oil spill accidents occurred in the past years, like the recent Deepwater Horizon oil spill in the Gulf of Mexico, a growing interest has emerged to develop technologies and methods to mitigate the effects and cost of such events. Some examples of these are the EU-MOP [167] and the SeaSwarm projects [168] and several research works dealing with boundary detection and tracking using simple reactive motion control [169] or a hybrid hierarchical control based on three behaviors: perimeter searching, pursuing, and tracking [170].

Extending the previous example of odor plume detection to a complete formulation of plume search, plume tracking and odor source localization in a dynamic coverage approach, we may consider a swarm in optimal formation sweeping the environment in a crosswind-biased direction while searching for odor plumes [171]. If a swarm element senses an odor clue (i.e., it gets in contact with a plume), the swarm, or a part of it, may self-organize, tracking the plume in the upwind direction. This approach has the benefit of providing spatial filtering to the measured concentrations, providing a more robust estimate of the concentration gradient [172]. When a group of elements cover the source area, a vector divergence operator may be employed to detect the existence of a source [173]. Later, its properties may be sensed by a swarm element staying in the neighborhood and propagated to the other swarm elements, so they keep searching other odor sources [174].

When searching targets with reduced probability of being detected, like an odor source, it provides similar performance searching the environment systematically or randomly. In this case, metaheuristic search algorithms, with an explorative behavior while the system detected no clue about the target location, but that turns exploitative when some clues are detected, have been demonstrated to perform better than conventional searching algorithms for similar conditions [153, 175, 176]. A limitation of these algorithms used to be dealing with multiple sources. This problem has been addressed by Menon and Ghose with a Glowworm-inspired algorithm for simultaneous source localization and boundary mapping [177].

9.5.3 Cooperative Target Localization and Tracking

The term target localization refers to estimation of the precise location of a target $T$ by a set $S = \{S_1, \ldots, S_N\}$ of sensory agents from multiple and typically noisy and nonlinear geometrical measurements related to relative positions of $S_1, \ldots, S_N$ to $T$ [50, 178]. The sensory agent set $S$ may be composed of a single sensory agent as well as multiple agents working cooperatively. The target $T$ is either a foreign entity to be identified and probably captured, or an agent $S_i$ within the set $S$ which needs self localization, or a beacon signal source to be used for self-localization of the whole sensor set $S$. 


For localization, the agents within $S$ are typically equipped with one particular type of measurement units. Most common measurement techniques include bearing measurements based on signal angle-of-arrival; range measurements based on received signal strength or signal time-of-arrival; range difference measurements based on signal time-difference of arrival; and scan based measurements [115, 179].

In [178], relative sensor-target geometries are investigated resulting in minimization of various measures of the uncertainty in location estimate of the target produced by the sensors cooperatively. A similar study with focus on target tracking is presented in [180]. Such studies justify the importance of maintaining certain geometric formations for a given sensor agent set $S$ in target localization and tracking tasks.

The aforementioned cooperative target localization and tracking tasks, will require use of distributed dual estimator/adaptive formation control algorithms. The distributed algorithms can be designed integrating various formation control schemes, some of which are mentioned in the previous sections, and suitable optimal cooperative localization algorithms such as the one studied in [50]. The first step of such a design is presented in [51].

Target localization and target tracking methods are particularly useful for emergency scenarios’ first responding teams. Typical examples of such scenarios are firefighting and search and rescue after earthquakes. The GUARDIANS EU project provides some interesting examples of using robot swarms to support human firefighters during the early phase of industrial fires, searching for hot spots, fire flames and chemical sources while assessing the risk level inside areas filled with smoke, toxic gases and inflammable materials [144]. Furthermore, the system provides human-swarming interaction, letting a firefighter guide a group of agents during the exploration phase or in a reverse behavior, being helped by a swarm of robots through escaping trajectories [181]. An additional service provided by the swarm is maintaining communication links among the operating agents and an external supervising and commanding station [182].

9.6 Concluding Remarks

In this chapter, a brief review of robot swarm systems is provided mainly from the aspects of dynamics and control. Within this context the most popular robot agent dynamic models, swarm coordination and control problems, various control design approaches to these problems were presented together with some applications. Due to space limitations, the presentation is kept very brief and the discussed literature is not exhaustive. Nevertheless, an extensive list of references has been provided for the readers to search for details of the briefly introduced models, problems, control designs, and applications. The field of robot swarm systems or multi-agent dynamic systems is still an active research field. Although there are many solved problems, there also many potentially fruitful new directions. Future research in the filed can concentrate on extending the research on robust and adaptive strategies for decentralized coordination and control under model and sensor uncertainties. Developing hardware specific more realistic sensor/interaction models may also be useful approach. Communication constrained swarm coordination and control is also an important topic of present interest. Moreover, developing model and strategies for human-swarm interaction, intelligent learning strategies and var-
ious levels of autonomy/intelligence for individual agents and the swarm as a whole are important topics of future research.
References

[38] Z. Lin, B. Francis, and M. Maggiore. Necessary and sufficient graphial conditions for forma-


[74] V. Gazi and R. Ordóñez. Target tracking using artificial potentials and sliding mode control.


J. T. Spooner and K. M. Passino. Stable adaptive control using fuzzy systems and neural


