COLLECTIVE BEHAVIOR AND SENSOR NETWORK
A MULTI-AGENT DYNAMIC SYSTEM APPROACH

A Thesis
Submitted to
the Temple University Graduate Board

In Partial Fulfillment
Of the Requirements for the Degree
MASTER OF SCIENCE IN ENGINEERING

---

by

Zhao Cheng
August, 2010

Dr. Li Bai

Thesis Advisor
College of Engineering
Committee Member

Dr. Saroj Biswas

College of Engineering
Committee Member

Dr. Jie Wu

College of Science and Technology
Committee Member

ABSTRACT

This research presents both theoretical foundation and numerical simulation work for design and analysis of a multi-agent dynamic system on the collective formation behavior patterns of grouped agents. A mass model with tunable control parameters is proposed. This model can realistically represent the aggregation pattern and the formation shape of multiple agents. Stability analysis is also provided to prove the stability of the second-order dynamic system. Several simulations will also be given according to the proposed model to show the aggregation patterns.

The research on self-organizing characteristics of collective agent behaviors has a wide range of applications in nature and engineering. The formation such as a flock of birds, a school of fish, or a swarm of locusts, is the emergence of ordered state in which the moving agents can organize as formation. Design and control of the self-organizing dynamic system has implications on wireless general design of mobile sensor networks, sensor network data fusion, attitude alignment of satellite clusters and congestion control of communication networks.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In this chapter, research background of collective behavior, aggregation behavior and sensor networks are presented. The organization of this thesis is given at the end of this chapter.

1.1 Collective Behavior

Collective behavior of multi-agent dynamical networks has attracted much attention in recent years. The self-organizing character of collective motion in nature, such as a flock of birds, a school of fish, or a swarm of locusts, is the emergence of *ordered state* in which the agents (particles) move in the same direction [1, 2], despite the fact that the interactions are (presumably) of short range. This issue can be further generalized into a consensus problem [3–5], i.e., groups of self-propelled particles agreeing upon certain quantities of interest such as position, temperature, and voltage, which has implications on general design of mobile sensor networks, sensor network data fusion, attitude alignment of satellite clusters, congestion control of communication networks and multi-agent formation control [6–9].

A flocking model based on computer simulation was proposed by Reynolds [10]. Three heuristic rules are prescribed,

(i) *separation*: steer to avoid crowding and collision;

(ii) *alignment*: steer towards the average heading;

(iii) *cohesion*: steer to move towards the average position.

These rules are effective and often used in the design of bio-group dynamic models (see [11–13]).

Vicsek *et al.* [1] proposed a dynamic model to describe the collective motion of self-propelled particles, which has drawn more and more attention recently from both physics and engineering communities [2, 11, 14–16]. In the *Vicsek model*, each particle tends to move in the average direction of its neighbors while being simultaneously subject to noise. As the amplitude of the noise increases, the system undergoes a transition from an *ordered state*, in which the particles move in the same direction, to a *disordered state*, where the directions are uniformly distributed. Jadbabaie *et al.* [17] provided mathematical convergence conditions for the Vicsek model, i.e., all the individuals should be linked at least during some time intervals. Grégoire and Chaté [2] modified the Vicsek model by changing the way the noise is introduced, which switched the phase transition from second- to first-order. Zhang *et al.* [4] introduced predictive mechanisms into the Vicsek model. Leonard, Sepulchre and *et al.* [18–20] proposed the design and control of mobile sensor networks for optimal data collection of ocean sampling. The model used by Leonard and Vicsek in their research is a simplified vector model, in which, each agent in the model moves with a constant unit velocity; the direction of motion is changed according to the control law. Leonard's model is based on continuous dynamics whereas the model used by Viseck et al is a discrete model.

The Gazi and Passino A/R model [11], embedded with a similar mechanism to the inter-molecular force, is derived from biological flocking/swarming behaviors. Thus far, the general understanding is that swarming behaviors resulted from an interplay between a long-range attraction and a short-range repulsion [21, 22]. In the physics community, a large volume of literature on systems with interactive particles has also adopted functions of attractive and repulsive forces to investigate the system dynamics [22–26]. With the A/R model, Gazi and

Passino [11] proved that the individuals would form a bounded cohesive swarm in finite time. It was later generalized into a social foraging swarm model [27] by adding one more factor. Under certain suitable circumstances, agents in this modified model are apt to move to the more favorable regions. The A/R model has been adopted by physicists and biologists to model self-driven particles and biological flocks [15, 28, 29]. The Gazi-Passino model is a kinematic model which could be viewed as an approximation of point mass swarm model. The control law is applied to the velocity instead of acceleration.

1.2 Aggregation Behavior

In 2004, Olfati-Saber [12] developed a theoretical framework for flocking. The model algorithms proposed a mass model based on Newton's law of motion. The moving agents based on this model will eventually gather and form an α -lattice. The case of flocking in free-space with multiple obstacles are considered, the simulation showed that the formed α-lattice has good performance in passing complex obstacles. The cost (or collective potentials) functions of the system are created and then the Lyapunov stability are proved.

Vicsek [30] discussed the universal patterns rooted in the systems consisting both living organisms and non-living objects. One of the most common and spectacular manifestation of coordinated actions, which describe the essential aspects of collective motion and provide a wide selection of systems ranging from colonies of tissue cells, flocks of birds to collectively moving robots. It has the potential of improving the interpretation of collective behavioral patterns in both living and nonliving systems to understand the interrelation of the systems by learning similar phenomena in the two domains of nature.

Juanico [31] proposed a modified kinematic model with attractive-repulsive pairwise interaction and showed an interesting simulation that agents will form

several stellate patterns due to the distribution change of preferred pairwise length. He also gave the definition and analysis of order parameter U as a measurement of pattern meta-stability. Although the variation among moving agents is a fact rooted in natural and artificial swarm systems, the role of diversity in the self-organized pattern formation has not been previously explored. However, the stellate patterns are based on kinematic model.

Chen and the author [32] introduced the attractive/repulsive (A/R) functional link between each pair of agents. By changing the slope of the A/R function, they found a dramatic transition between two different formation patterns. In the liquid-like pattern, the outer agents are sparsely distributed while the inner ones densely. And agents will remain constant distance in between in the crystal-like pattern.

1.3 Sensor Networks

A sensor network is a networked system that composes a large number of sensor nodes [6]. Mobile sensor networks are sensor networks in which sensor nodes have motion capability and can move under their own control. In 2004, Ogren [7] presented a stable control strategy for sensor nodes move and reconfigure cooperatively in response to a sensed, distributed environment. The gradient climbing strategies are applied to artificial potentials to drive the sensor nodes.

In 2005, Nguyen [33] introduced the concept of artificial potential field to guide the movement of a robot. In his research, the robot was built with artificial potential fields and was able to navigate itself to a particular location through a path with obstacles. Later, the artificial-potential filed approach was applied into management of an mobile sensor network to improve the sensor performance. Heo [34] proposed and analyzed distributed energy-efficient deployment algorithms for mobile sensor networks. In 2008, Ma [9] introduced

non-Newton's model and discussed the sensor coverage problems in sensor management of mobile sensor networks. This research is based on simulations and analysis in three cases: spatial coverage, spatial migration and spatial retreats.

1.4 Thesis Goals and Objectives

This thesis is mainly inspired by the research [32] published in 2009. In this paper, two different formation patterns are generated by changing one parameter of the model. In the crystal-like pattern, distance between agents remain constant while in the liquid-like pattern, the distance varies based on the location of agents. However, we still do not know how and why there are two different formation patterns. It is better to do more simulations, observing the collective behavior and implement mathematical analysis in order to better understand the mechanism hidden behind this phenomenon.

The objective of this research is to explore principles of collective formation with an appropriate model. Then analysis in three different aspects - stability, formation and performance - will be discussed. Besides, simulations and case study is also required to show the results and support the research. For convenience, a simulation platform could be built with graphic user interface. With this software, readers interested with this topic can design their own simulations via configuring the settings and parameters of simulation platform. In the end, a summation will be provided to summarize the accomplishment.

1.5 Organization of the Thesis

The rest of this thesis is organized as follows. In Chapter 2, several models of collective behavior are introduced. Simulations and observations are given in Chapter 3. Chapter 4,5 and 6 discusses stability, formation pattern and appli-

cations of the system respectively. Chapter 7 gives the conclusion and future work. All papers referenced in this thesis can be found in References. Simulation platform and its source code are showed in Appendix.

CHAPTER 2

MODELS OF COLLECTIVE MOTION

In this chapter, three types of models are presented: the Unit-vector Model, the Kinematic Model and the Mass Model. The Unit-vector Model is a steering control model, in which the direction of each agent is determined by a control algorithm. However, in the second model, the Mass Model, agents obey the rule of Newton's law of motion: both direction and speed of each agent was updated at each step. Similar to the second one, the the third type of model, named as Kinematic Model, is also a system whose agents have both their directions and scalar velocities updated according to the control algorithm. But, they are still different, since the control input is applied directly on the the velocity of each agent in Kinematic Model instead of acceleration (derivative of velocity) in the Mass Model. Thus, the Kinematic Model can also be viewed as an approximation of Mass Model. Note that the Kinematic Model is a first-order system while the Unit-vector Model and the Mass Model are second-order systems. Generally, the second-order systems have better stability performance than first-order systems. The Mass Model is also the closest one to movement mechanism of a real object over all three models. Because of these reasons, the Mass Model is adopted and discussed in the following chapters.

2.1 Preliminary Notations

In order to introduce the existing models, some preliminary notations are presented in this section. The model used in the thesis is a group of n moving agents (or particles) in a square cell with periodic boundary condition with equations of motion. The communication problems among agents are studied by means of graph theory. A graph G is defined as a pair $(\mathcal{V}, \mathcal{E})$ that consists of a set of vertices $\mathcal{V} = \{1, 2 \dots N\}$ and a set of edges $\mathcal{E} \subseteq \{(i, j) : i, j \in \mathcal{V}, i \neq j\}$. The moving agents are the vertices of the graph G and the connection between a pair of agents is an edge defined in G. The graph G is said to be an undirected graph if $(i, j) \in \mathcal{E} \Longleftrightarrow (j, i) \in \mathcal{E}$, otherwise, the graph G is named as a directed graph. Let $x_i, v_i, u_i \in \mathbb{R}^m$ (m=1,2,3) denote the position, velocity and control input of node i for all $i \in \mathcal{V}$, respectively.

Let r denotes the interaction range between two agents, then the set of spatial neighbors of agent i is defined by

$$\mathcal{N}_i = \{j \in \mathcal{V} : \|x_j - x_i\| < r\} \tag{2.1}$$

where $\|\cdot\|$ is Euclidean norm in $\mathbb{R}^m$. Define $|\mathcal{N}_i|$ as the total number of elements in the set $\mathcal{N}_i$. Fig. 2.1 shows the pairwise interaction of the agent 1. The agent 2 and 3 are neighboring agents of agent 1 since their distance to agent 1 is less than r.

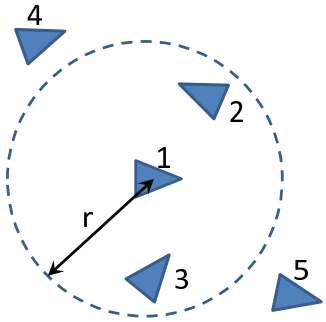


Figure 2.1: Pairwise interaction of an agent.

Collective potential function $V(x)$ of a group of n agents is nonnegative, $V \geq 0$. There are several models describing the intermolecular potential, such as the Hard-Sphere Model, the Square-Well Model and the Exponential-6 Potential Model, etc, among which, the Lennard-Jones potential is the best-known function for non-polar monatomic systems [35,36]. The Lennard-Jones potential function may be written as the summation of an attractive field and a repulsive field [37],

$$F(x) = \lambda_m x^{-m} - \lambda_n x^{-n}$$

where m and n are integers, λ_m and λ_n are the coefficients of x^{-m} and x^{-n} respectively.

The potential function for two agents is presented as Ψ_a, which is varies according to the interval x_{ij} of two agents i and j. It reaches the global minimum 0 at $x = d$ (see Fig. 2.2). And d denotes the equivalent point, where the absolute value of ψ reaches the minima, and r is the scope of an agent. The ratio of equivalent point to the scope of an agent is $k \triangleq \frac{r}{d}$.

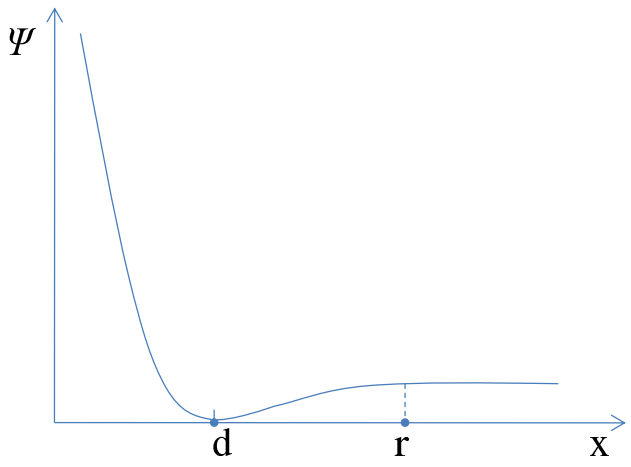


Figure 2.2: Pairwise potential $\Psi(x)$.

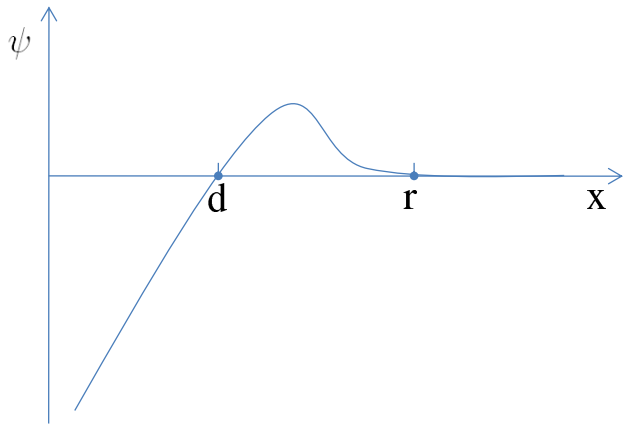


Figure 2.3: Pairwise force $\psi(x)$.

2.2 Unit-vector Model

In Unit-vector Model, each agent has its position at x_i and the heading at θ_i ($x_i \in \mathbb{R}^2$). All agents move with a constant speed v. Without loss of generosity, assign $v = 1$ as all agents move at an unit and constant speed [19, 38]

$$\begin{cases} \dot{x_i} = ve^{j\theta_i} \\ \dot{\theta}_i = u_i \end{cases} \tag{2.2}$$

where $i \in \mathcal{V}$, $j = \sqrt{-1}$. The control input u_i control the change of direction for agent i, thus the scalar u_i is also called as the steering control. The Unit-vector Model is second-order dynamics of agents with force in steering control.

The Vicsek Model [1] is also widely used. It can be viewed as a discrete variation of Unit-vector Model. Agents in the Unit-vector Model move at a universal constant speed and has only their directions updated which is the same as agents in Vicsek Model do. However, a system created with the Viscek Model is a discrete Unit-vector Model with specialized control algorithm. This will be introduced in the following paragraph.

In the Vicsek Model, n agents move with a constant speed. The velocities $\{v_i e^{j\theta_i}\}$ of the agents are determined simultaneously at each simulation step, the

next position of the i-th agent updated at time step t according to

$$x_i(t+1) = x_i(t) + v_i(t)e^{j\theta_i(t)}\Delta_i(t)$$

The absolute value of velocity v_i is assigned to be 1. The next angle of the velocity at time step t is from

$$\theta_i(t+1) = \arctan\left\{\frac{\sum_{j\in\mathcal{N}_i} sin(\theta_j(t))}{\sum_{j\in\mathcal{N}_i} cos(\theta_j(t))}\right\} + \Delta\theta$$

where $\Delta\theta$ denotes noise, a random number chosen with an uniform probability from the interval $[-\eta/2, \eta/2]$.

2.3 Kinematic Model

In the Kinematic Model, n individual agents are defined with the equation of motion

$$\dot{x_i} = u_i \tag{2.3}$$

where $i \in \mathcal{V}$. The model defined in Eq. 2.3 can be viewed as an approximation of a model with point agents whose mass are negligible, e.g. bacteria. Consider the individual agent move based on the Newton's law $m_i a_i = F_i$. This gives the equation of motion of the system

$$\begin{cases} \dot{x_i} = v_i \\ m_i\dot{v_i} = u_i \end{cases}$$

where u_i is net force on an agent i. Suppose there is a velocity damping term with the form $-k_v v_i$ in $u_i (k_v > 0)$, we have $u_i = -k_v v_i + \hat{u}_i$. Since for certain agents such as bacteria, the mass m_i is very small and the viscosity of the environment is high, we have $m_i \approx 0$ and the system equation of motion becomes

$$\dot{x_i} = \frac{1}{k_v}\hat{u}_i$$

now consider $k_v = 1$, this model is simplified as a first-order system in Eq. 2.3 from Newton's Law of motion [27].

2.4 Mass Model

Consider a group of n moving agents in $\mathbb{R}^m$, $m = 1, 2, 3$ with equation of motion [12],

$$\begin{cases} \dot{x_i} = v_i \\ \dot{v_i} = u_i \end{cases} \tag{2.4}$$

where $i \in \mathcal{V}$ and the control input of agent i consists of three terms

$$u_i = f_i^g + f_i^c + f_i^n \tag{2.5}$$

where $f_i^g = -\Delta_{x_i} V(x)$ is a gradient term, $f_i^c = \sum_{j \in \mathcal{N}_i} (v_j - v_i)$ is a velocity consensus term and $f_i^n \triangleq f_i^n(x_i, v_i, x_r, v_r)$ is a navigational term. The pair (x_r, v_r) is the navigation objective. To fully present the control input, u_i is written as below,

$$\begin{aligned} u_i &= \underbrace{c_1 \sum_{j \in \mathcal{V}} \psi(\|x_j - x_i\|) n_{ij}}_{gradient\ term} + \underbrace{c_2 \sum_{j \in \mathcal{N}_i} (v_j - v_i)}_{consensus\ term} \\ &+ \underbrace{c_3 \left(-k_x(x_i - x_r) - k_v(v_i - v_r)\right)}_{navigation\ term} \end{aligned} \tag{2.6}$$

where n_{ij} is a vector pointing from x_i to x_j, $n_{ij} = \frac{x_j - x_i}{\|x_j - x_i\|}$, $\mathcal{V}$ is the universal set of agents, and $i \in \mathcal{V}$, c_1, c_2 and c_3 denotes the weight parameter for gradient term, consensus term an navigation term respectively. The strength of this acceleration is related to the distance, x, between two particles, which is also the derivative of the potential between two particles.

2.5 Control Algorithm of Mass model

There are two options available to apply a potential function with these models: first, using the current function with a infinite cutoff; second, modify one of the models and make it with a finite cutoff $r < \infty$. The advantage of choosing a finite cutoff potential function is in computation; it only need to consider the agents within a certain cutoff range (scope of an agent) which results in less computation. However, there is actually no such cutoff existing in the real world, such that, the second option is used. With a finite cutoff, the scope of an agent can be designed as a distance, beyond which, the information is too weak to influence an agent. As a result, it could be better to utilize the property of exponential function, to define a finite cutoff distance approximately and mathematically from an infinite potential function.

The exponential function is attractive between two agents when $x \in [d, \infty)$ and repulsive when $x \in [0, d)$. Combining a exponential function and a second-order polynomial function, we have

$$\psi(x) = \begin{cases} Ax^2 + Bx + a, & x \in [0, d), \\ \frac{b}{c}(x - d)\exp\left(-\frac{(x-d)^2}{c}\right), & x \in [d, \infty). \end{cases} \tag{2.7}$$

The second-order polynomial $Ax^2 + Bx + a$ must satisfy the following conditions to ensure continuity and differentiability of $\psi(x)$ at $x = d$:

$$\begin{aligned} \psi(x) \mid_{x \to d} &= f(x)_{x=d}, \\ \psi'(x) \mid_{x \to d} &= f'(x)_{x=d}. \end{aligned}$$

The pairwise potential $\Psi(x)$ is integration of pairwise force $\psi(x)$,

$$\Psi(x) = \int_d^x \psi(s)\mathrm{d}s.$$

The parameter of A and B can be determined by ensuring continuity and differentiability of $f(x)$ at $x = d$. Parameter a and b are related to the size of

simulation and determine the maximum value of attraction and repulsion respectively. To make it suitable for a square area with edge value of 200, a is set to 5 and b is 0.2, the free parameters in this system are c, d. The parameter d is an equilibrium of two agents and it determines the interval between two agents when system is stable. When the coefficient c changes, the scale of attraction, r (cutoff distance), varies accordingly. The function $g(x) = x\exp(-x^2/c)$ will increase when x is relarively small and then decreases exponentially. When x is beyond $r = \sqrt{10c}$, the value of function $g(x)$ will be too small and negligible $g(x) \to 0$. When c is around 10, we can approximate $r = c$, beyond which the attraction is negligible. The value of c is related to the value of two other parameters a and b. When assgin $a = 5$, $b = 0.2$, the parameter c varies in the domain $[5, 17]$, above which, the attractive force is weakened so that particles do not gather and form clusters, while beyond which, the attractive force could be too strong to cause explosion in the center of the particle cluster, especially when the size of cluster is large (greater than 400).

CHAPTER 3

SIMULATION AND OBSERVATION OF COLLECTIVE MOTION

In this chapter, several simulation results are presented from the modified Mass Model introduced in Chapter 2. A list of figures are showed in this chapter to demonstrate the behaviors of agents. Under the Mass Model, agents may perform flocking and aggregation both in 2D and 3D environment. Two patterns are introduced in this chapter, the liquid-like pattern and the crystal-like pattern.

3.1 Flocking

In this section, following simulations is used: In a two-dimensional square with periodic boundary conditions, N particles are generated, whose initial locations and velocities are selected randomly. Then all particles in this region evolve according to Eq. (2.6) and the states of all particles are updated synchronously. The parameter c is set as 5 so that the ratio k stay close to 1.

Figure 3.1 shows the agents gathering and forming α-lattice structure. In the figure, each point represents a particle and the edge connected two particles is the connection between neighboring particle pair. (a) Type I. The particles are initialized and randomly distributed in the area. (b) The agents start to aggregate and more connections are created. (c) The majority of the particles

are connected within a group. (d) The α-lattice structure is showed, in which, all the agents are connected and synchronized.

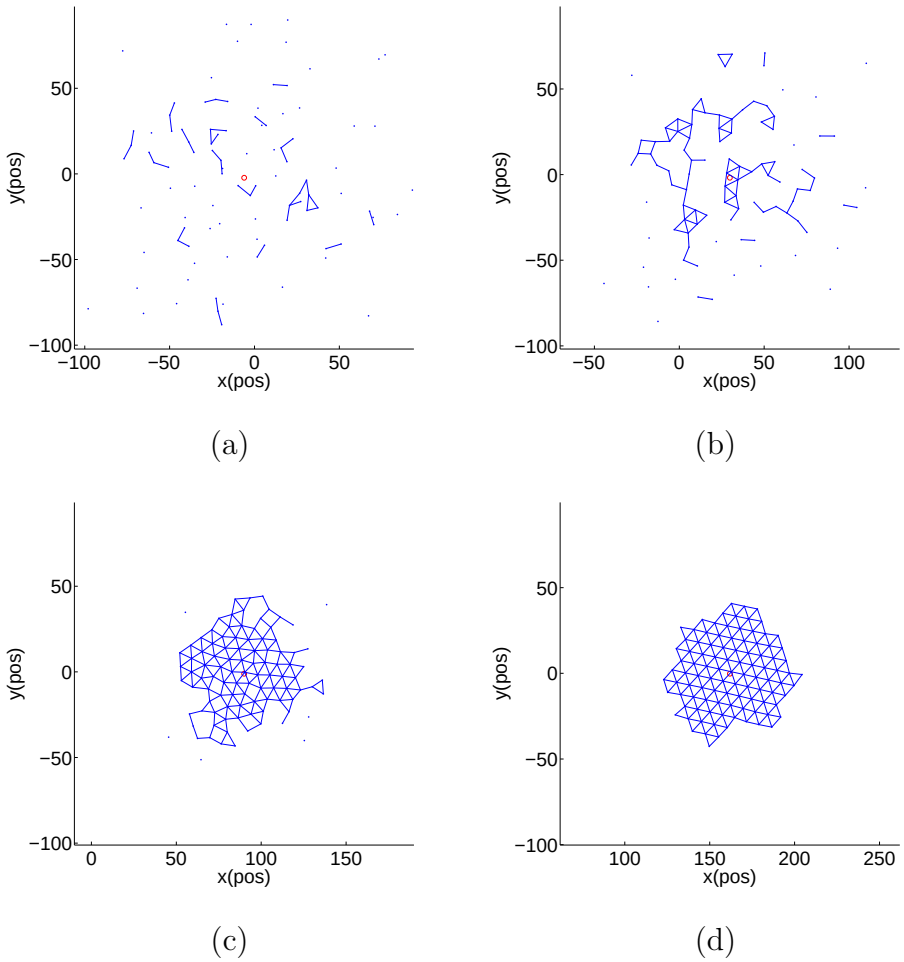


Figure 3.1: Agents gather and form an α-lattice structure.

The next simulation shows agents passing through obstacles from the left-hand side to the right-hand side. In Figure 3.2, (a) agents are randomly generated in the left of the three obstacles, some connections are created; (b - e) Agents form a group and then pass through the space among the three obstacles; (f) Once the agents pass the obstacles, they start regrouping.

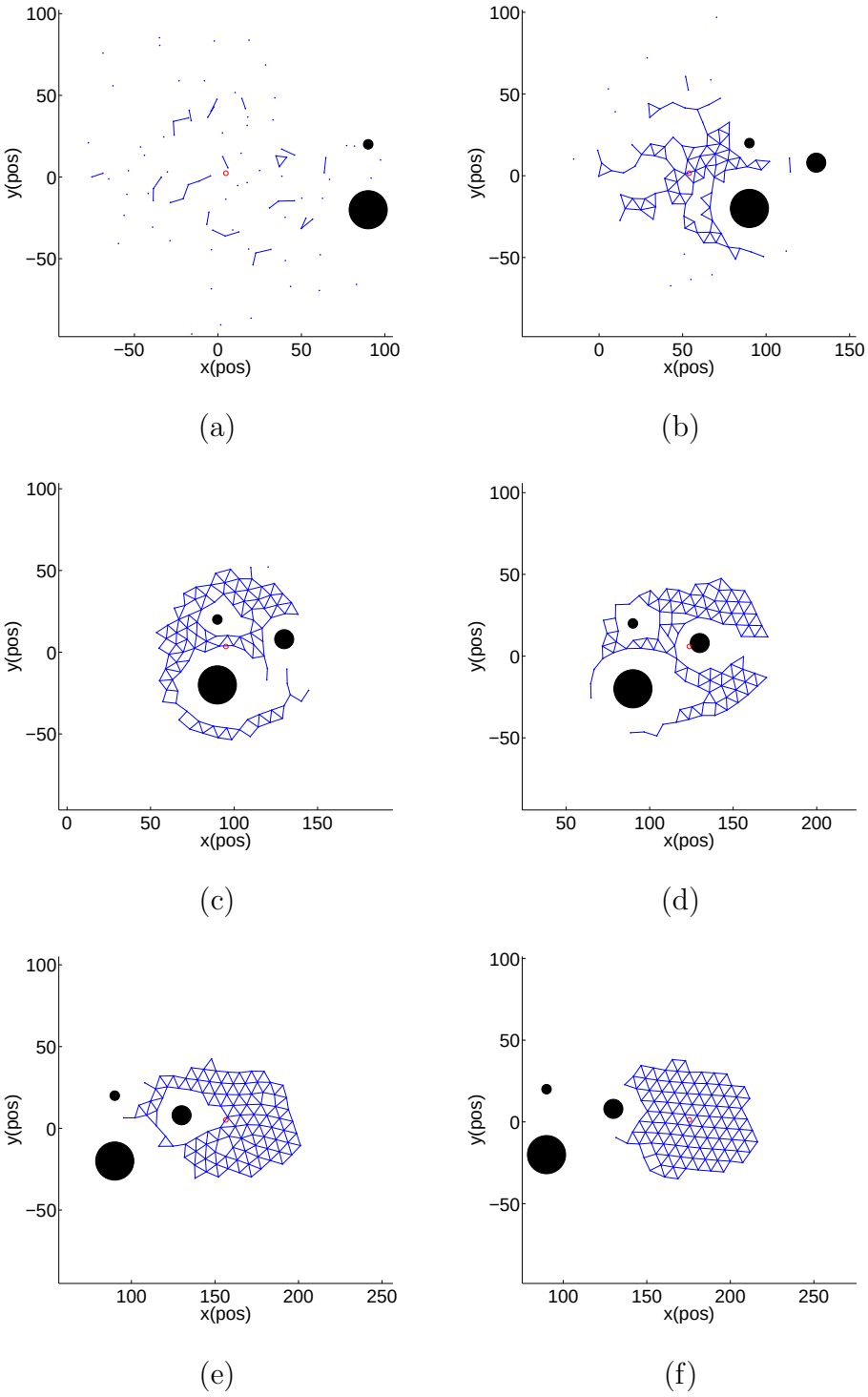


Figure 3.2: Agents pass through the obstacles.

3.2 Aggregation Formation

The following simulations are designed in both two-dimensional square and three-dimensional cubic with periodic boundary conditions, N particles are generated, whose initial locations and velocities are selected randomly. Then all particles in this region evolve according to Eq. (2.4) and the states of all particles are updated synchronously. The edge length of the square boundary is 200. From the simulations, some interesting phenomenon emerged: when the parameter c in the Eq. (2.7) is changed, a transition between different aggregation patterns, namely *liquid-like* and *crystal-like* patterns, surfaces.

The agents' movement in liquid-like pattern is showed in Figure 3.3, agent number $n = 100$ and $c = 0.25$, $d = 1$. (a) Initial positions. (b) Positions after 800 steps. (c) Positions after 1600 steps. (d) Aggregation pattern of one liquid-like cluster. The green circle surrounding the particles is a symbol to mark the cluster.

Figure 3.4 shows the agents in crystal-like pattern evolve in a bounded square area. The number of agents $n = 100$, $c = 0.05$ and $d = 1$. (a) Initial positions. (b) Positions after 800 steps. (c) Positions after 1600 steps. (d) The Aggregation pattern of one crystal-like cluster.

Figure 3.5 Aggregation structure of different patterns. The number of particles $N = 50$ with $d = 1$. (a) The "liquid-like" pattern with $c = 0.25$. (b) The "crystal-like" pattern with $c = 0.05$. (c) The transitional pattern with $c = 0.15$.

For the liquid-like case, after a certain period, some small clusters of particles are formed. Due to the 'surface tension', these particles will gradually form a circle with increasing density from exterior to interior, as shown in Fig. 3.3. Such patterns are quite similar to the formation of molecules in liquid phase and hence called liquid-like patterns. When two groups encounter each other, they will merge into a larger circle-shaped cluster. The final shape of the new

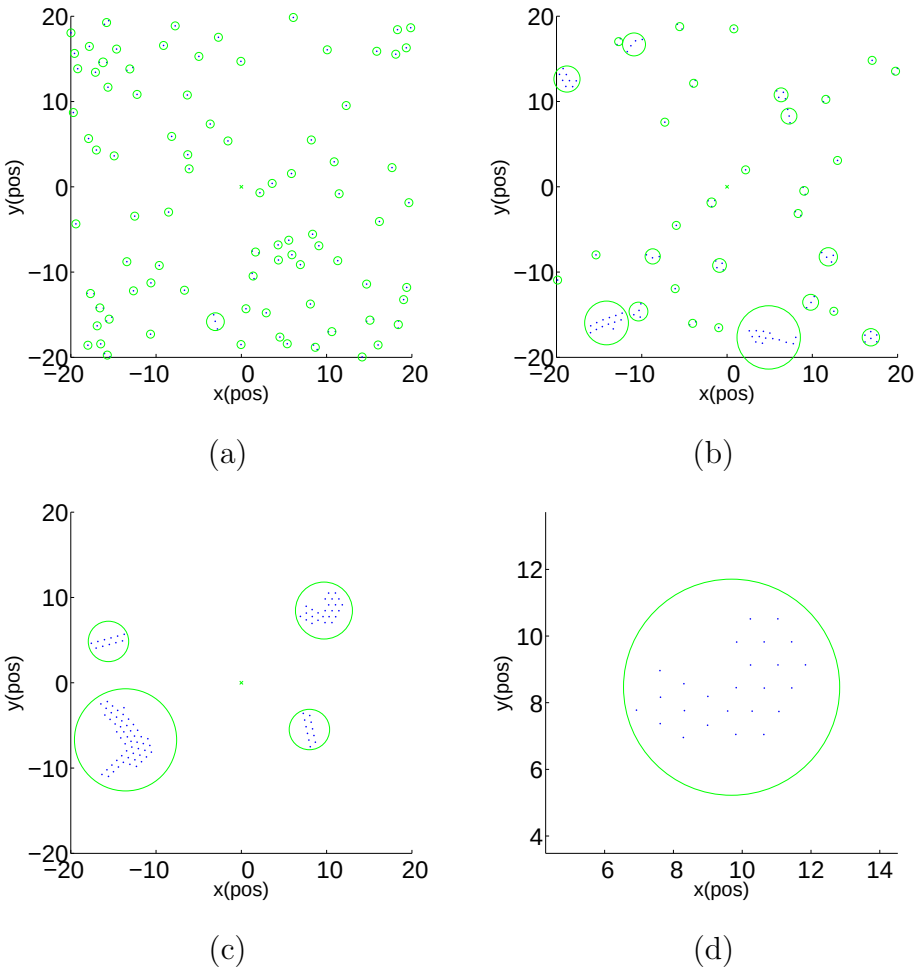


Figure 3.3: Agents aggregate under liquid-like pattern.

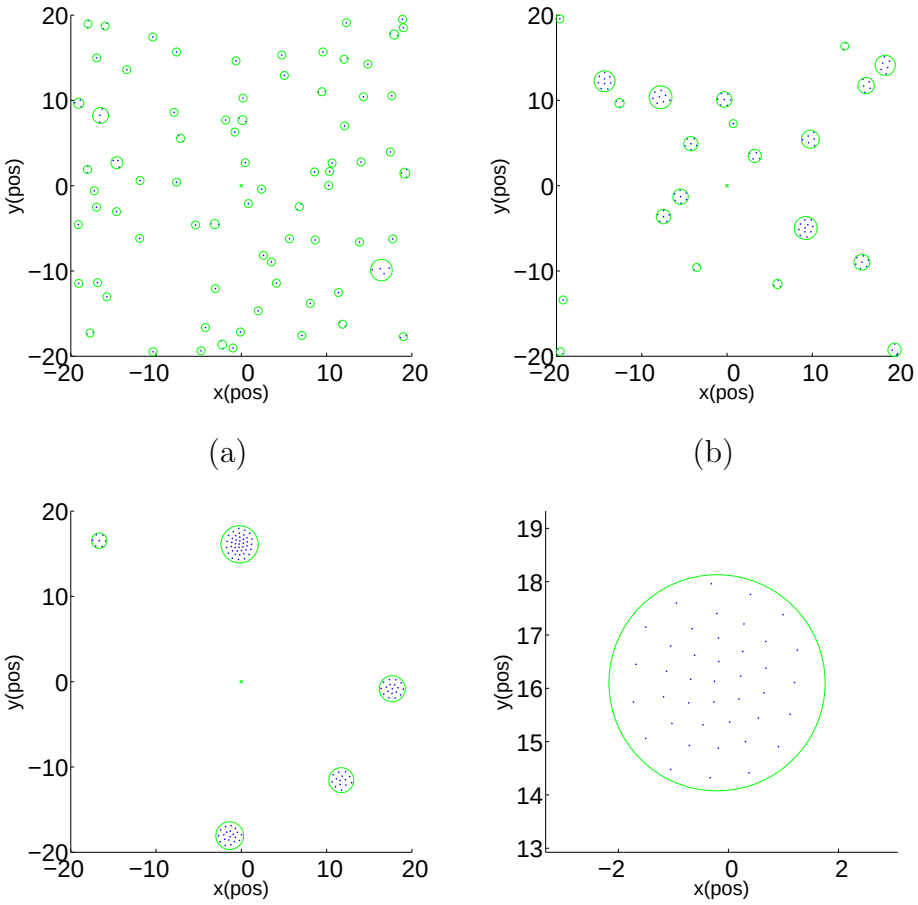
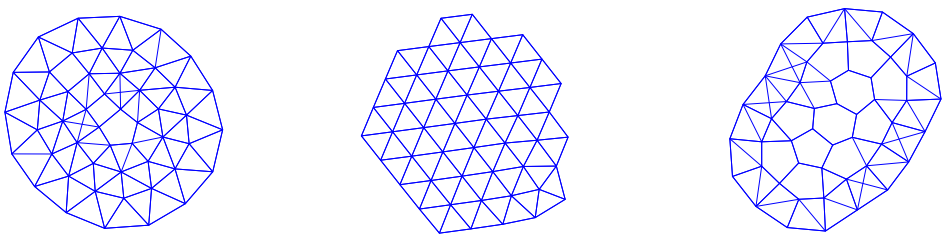


Figure 3.4: Agents aggregate under crystal-like pattern.



(a) (b) (c)

Figure 3.5: Aggregation structure of different patterns.

larger cluster is independent on the original orientations and velocities of the two clusters before merging with each other.

For the crystal-like case, small clusters of particles emerge after a period of time with a more even distribution of particles compared with the liquid-like pattern, as shown in Fig. 3.4. Such patterns are quite similar to the formation of molecules in crystal phase and hence called crystal-like patterns. When two clusters encounter, a larger cluster is formed. This new large cluster will not be round in shape. The final shape of the new cluster is dependent on the original orientations and velocities of the two clusters prior to merging with each other.

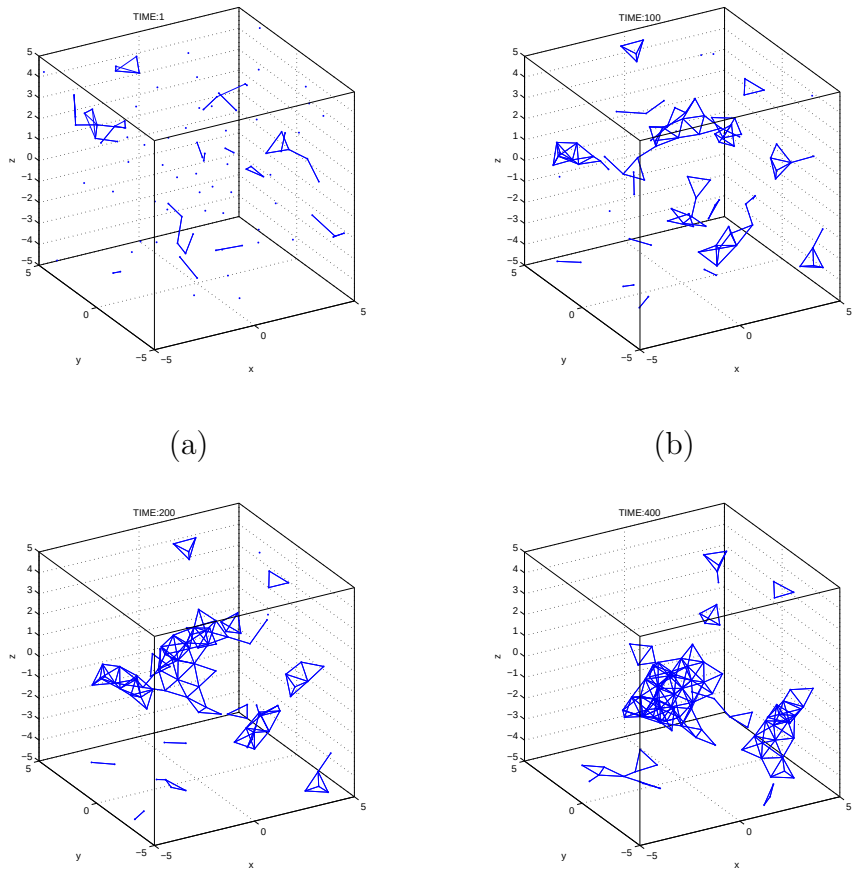
Between the two patterns, there exists a certain unstable pattern, like a transitional phase, as shown in Fig. 3.5(c). In this pattern, there is the crystal-like irregular shape together with the liquid-like round features. In general, the internal distribution of particles is uniform, but with exterior irregular shapes.

The determining factor that influences the two above-mentioned aggregation patterns is believed to be the distance of attraction. The reason can be explained as follows. When the effective distance of attraction is short, a particle will only affect the adjacent particles. As a result, the distribution of particles will be uniform and the distance between two particles will not be affected by the size of the group (see Fig. 3.5(b)). In comparison, when the effective distance of attraction is long enough, particles will be attracted by not only the adjacent particles but also some particles further away. As a result, the inner particles will be pressed closer to their neighbors and the outer particles enjoy larger separations as the pressure among them is smaller (see Fig. 3.5(a)). The outer particles will also form a circle/sphere to surround the group, which is similar to the surface tension in liquid.

3.3 Aggregation formation in 3-D

In the three-dimensional environment, the agents' movement in both liquid-like pattern and crystal pattern is quite similar with that in two-dimensional environment. Figure 3.7 shows the movement in liquid pattern, with agent number $n = 100$ and $c = 0.25$, $d = 1$. (a) Initial positions. (b) Positions after 1000 steps. (c) Positions after 2000 steps. (d) (c) Positions after 4000 steps. In this case, some small clusters of particles are formed shortly after the simulation starts. Due to the 'surface tension', these particles will gradually contract to form a circle with increasing density from exterior to interior, as shown in Fig. 3.7. Such patterns are quite similar to the formation of molecules in liquid phase The final shape of the cluster is independent on the original orientations and velocities of the two clusters before merging with each other.

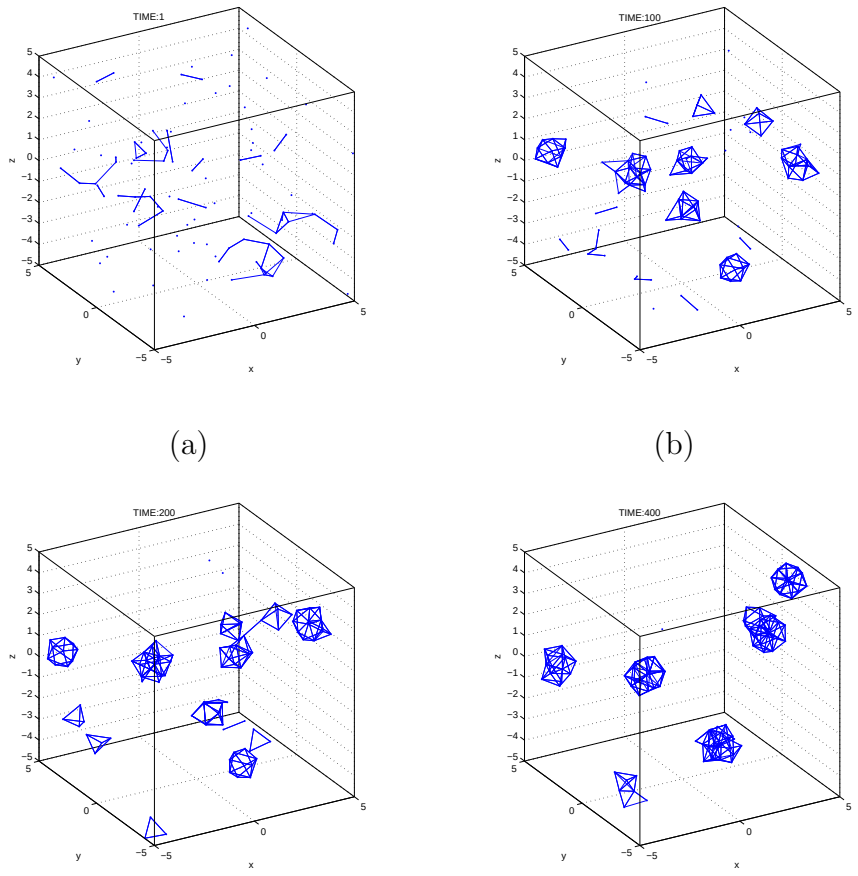
In the crystal-like case, small clusters of particles emerge after a period of time with a more even distribution of particles compared with the liquid-like pattern, as shown in Fig. 3.6. When two clusters encounter, a larger cluster is formed. This new larger cluster will not be sphere in shape. The final shape of the new cluster is dependent on the original orientations and velocities of the two clusters prior to merging with each other. Which is similar to the two-dimensional simulation for the crystal-like pattern.



(a) (b)

(c) (d)

Figure 3.6: Agents aggregate under crystal-like pattern in 3D environment.



(a) (b)

(c) (d)

Figure 3.7: Agents aggregate under liquid-like pattern in 3D environment.

CHAPTER 4

STABILITY ANALYSIS

The system of multiple agents can be considered as a dynamical system of movement of agents. A widely used method named Lyapunov stability is to analyze the stability of this system. In this chapter, Lyapunov stability theorem is introduced. Energy function of a system plays an important role in order to apply Lyapunov stability theorem. In this case, the Hamiltonian is chosen as the energy function and is used to analyze the state of this system.

4.1 Lyapunov Stability

Consider an autonomous nonlinear dynamical system

$$\dot{x} = f(x(t)), \quad x(0) = x_0, \tag{4.1}$$

where $x(t) \in \mathcal{D} \subseteq \mathbb{R}^n$ denotes the system state vector, $\mathcal{D}$ is an open set that contains the origin, and $f : \mathcal{D} \to \mathbb{R}^n$ is a locally Lipschitz map. Suppose f has an equilibrium; without loss of generality, we may assume that it is at origin $f(0) = 0$.

The equilibrium point $x = 0$ of E.q. (4.1) is defined as [39]:

1. stable, if, for each $\epsilon > 0$, there exists a $\delta = \delta(\epsilon) > 0$ such that

$$\|x(0)\| < \delta \Rightarrow \|x(t)\| < \epsilon, \quad \forall t \geq 0$$

2. unstable, if not stable.

3. asymptotically stable, if it is stable and there exist $\delta > 0$ such that

$$\|x(0)\| < \delta \Rightarrow \lim_{t\to\infty} x(t) = 0$$

Conceptually, the meanings of the above terms are the following:

1. Lyapunov stability of an equilibrium means that solutions starting "close enough" to the equilibrium (within a small distance δ from it) remain "close enough" forever (within a small distance ϵ from it).

2. Asymptotic stability means that solutions that start close enough not only remain close enough but also eventually converge to the equilibrium.

3. Exponential stability means that solutions not only converge, but in fact converge exponentially to zero.

4.2 Hamiltonian and Stability

The value of the Hamiltonian is defined as the total energy of the system. For a closed system, it is the sum of the kinetic energy and potential energy. Hamiltonians can be used to describe simple systems as bouncing ball, pendulum or oscillating spring in which energy changes from kinetic to potential and back again over time. The model of the energy of complex dynamic system such as planetary orbits in celestial mechanics and in quantum mechanics are widely used.

The Hamiltonian H of the system is often created to help for the Lyapunov stability analysis. For the system defined in 2.4, the Hamiltonian H consists of two terms, collective potential $V(x)$ and collective kinetic energy $K(v)$

$$H(x, v) = V(x) + K(v) \tag{4.2}$$

The collective potential is the summation of potentials defined by connections in

the graph G

$$V(x) = \frac{1}{2}\sum_i \sum_{j \neq i} \Psi(x_j - x_i)$$

where $i, j \in \mathcal{V}$. The collective kinetic energy $K(v)$ is the summation of kinetic energy of all agents defined in G

$$K(v) = \frac{1}{2}\sum_i \|v_i\|^2$$

where $i \in \mathcal{V}$. We define a m-dimensional graph Laplacian as

$$\hat{L} = L \otimes I_m$$

where L is the scalar graph Laplacian, a $n \times n$ matrix of graph G, the symbol $\otimes$ denotes the Kronecker product. This multi-dimensional Laplacian satisfies the sum of squares property

$$z^T \hat{L} z = \frac{1}{2}\sum_{(i,j)\in\mathcal{E}} a_{ij}\|z_j - z_i\|^2, \ z \in \mathbb{R}^{mn}$$

where $z = col(z_1, z_2, \ldots, z_n)$.

Take the derivative of $H(x, v)$, we have

$$\begin{aligned} \dot{H}(x,v) &= \langle \nabla V(x), dx/dt \rangle - \langle dv/dt, v \rangle \\ &= \langle \nabla V(x), v \rangle + \langle -\nabla V(x) + \hat{L} v^T, v \rangle \\ &= -v^T \hat{L}(x) v \\ &\leq 0 \end{aligned} \tag{4.3}$$

where $\dot{H}(x, v) = 0$, only if $v = 0$, that means there is no interval movement between any two agents, $dx = 0$.

To allow the usage of LaSalle's invariance principle, the key stability analysis of collective dynamics is chosen as the appropriate coordinate system. The approach of using $H(x, v)$ in the (x, v)-coordinates does not work since one cannot establish the bound of solutions. According to Olfati-Saber's method [12],

a moving frame of the average of the whole group is treated as the coordinates. The group center of position $x_c = \frac{1}{N}\sum_{i=1}^{N} x_i$ and the average velocity is $v_c = \frac{1}{N}\sum_{i=1}^{N} v_i$. Then the position and velocity of agent i in the moving frame is given by

$$\begin{cases} \hat{x}_i = x_i - x_c \\ \hat{v}_i = v_i - v_c \end{cases} \tag{4.4}$$

In the $(\hat{x}, \hat{v})$-coordinate, we can also prove Eq. (4.5).

$$\begin{aligned} \dot{H}(\hat{x}, \hat{v}) &= \langle \nabla V(\hat{x}), d\hat{x}/dt \rangle - \langle d\hat{v}/dt, \hat{v} \rangle \\ &= \langle \nabla V(\hat{x}), \hat{v} \rangle + \langle -\nabla V(\hat{x}) + \hat{L}\hat{v}^T, \hat{v} \rangle \\ &= -\hat{v}^T \hat{L}(\hat{x})\hat{v} \\ &\leq 0 \end{aligned} \tag{4.5}$$

Note that both Eq. (4.3) and Eq. (4.5) are describing the same objectives, they are mathematically equivalent. The LaSalle's invariance principle can be applied to Eq. (4.5), which finally proved the Lyapunov Stability.

CHAPTER 5

FORMATION ANALYSIS

This chapter is focused on Formation analysis of aggregation patterns: the crystal-like pattern and the liquid-like pattern. Theorems and proofs for lattice formation and sphere formation are given. Performance analysis is also provided for better understanding the system.

5.1 Case 1: Lattice Formation

In this case, particles form crystal-like pattern in the stable state, with $1 < k < 2$.

Theorem 1(Lattice): Consider a group of N agents defined in Eq. 4.1, $x_i, v_i \in \mathbb{R}^m (m = 1, 2, 3)$ with the same equivalent point d and scope r, if they are in Lyapunov stability and $1 < k < 2 (k = \frac{r}{d})$, the group forms a lattice, with neighboring positions $\{x_{i,j}\}$ remain constant.

Proof: In the case of $1 < k < 2$, an agent interacts with only one neighboring agent in any direction. The hamiltonian H is given by the summation of total potential and kinetics, see Eq. 4.2.

Consider the potential function

$$V = \sum_{i \in v} \sum_{i,j \in v} V_{i,j} = \sum_{i \in v} \sum_{i,j \in v} \psi_{i,j}$$

In any direction, agent i and its neighboring agent j has an interaction force $\varphi_{i,j} = 0$. Assume that if the interaction force $\varphi_{i,j} \neq 0$, then the whole group can be divided into two parts by an line orthogonal to and across the center of the edge of agent i and agent j. The two parts will move due to the inter-force

in between, which contradict with the stable condition $v = 0$; so the interaction force $\varphi_{i,j} = 0$.

The parameter $\varphi_{i,j} = 0$ indicates the equivalent point of potential ψ, which indicates that the interval $x_{i,j}$ between agent i and agent j is a constant value, $x_{i,j} = d$.

5.2 Case 2: Sphere Formation

In this case, particles are forming liquid-like pattern in the stable state, with $k > 2$.

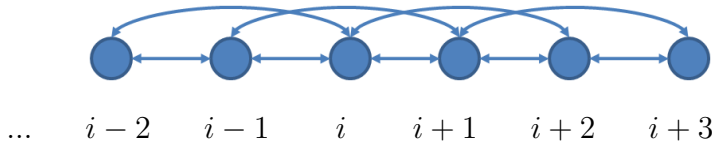


Figure 5.1: Interactions among particles (a).

Consider a group of N agents defined in Eq. refeq: system, $x_i, v_i \in \mathbb{R}^m$ with the same equivalent point d and scope r. As m=1, all N agents are distributed in a line with only two applicable moving direction, the positive and the negative. Assume that most of the agents has at least two neighbors in one direction except those close to the boundary, where the interval between agent i and j is defined as

$$d_i \triangleq x_{i,j} = \|x_i - x_j\|$$

Lemma 1(Symmetry), In any direction, agent i and its neighboring agent j has an interaction force,

$$\sum_j \varphi_{i,j} = 0, i, j \in \mathcal{V}, j = i-1, i+2, i+3...$$

Assume that $\sum_j \varphi_{i,j} = C(C \neq 0)$, then in the opposite direction, for neigh-

boring agents

$$j', j' \in \mathcal{V}, j = i+1, i-2, i-3, ...$$

As in stable state,

$$\dot{v_i} = 0 \Rightarrow v_{i,j} = 0 \Rightarrow \dot{x_{i,j}} = 0$$

so that

$$\sum_{j'} \varphi_{i,j'} = \sum_{j} \varphi_{i,j} = C$$

which means,

$$\dot{v}_{i,j} \neq 0$$

This is contradict with the Lyapunov Stability condition. Therefore,

$$\sum_{j} \varphi_{i,j} = 0, i, j \in \mathcal{V}, j = i-1, i+2, i+3, ...$$

Lemma 2(Boundary), $d_i < d, i \in \mathcal{V}$.

From Lemma 1, it is proved that agent i and its neighboring agent j has an interaction force canceled each other in any direction,

$$\sum_{j} \varphi_{i,j} = 0, j = i-1, i+2, i+3, ...$$

For agent i and its non-adjacent neighboring agents $\hat{j}$, $x_{i,\hat{j}} > r$ then

$$\sum_{j} \varphi_{i,\hat{j}} > 0, \hat{j} = i+2, i+3, ...$$

such that

$$\forall i, \varphi_{i,i+1} < 0$$

then

$$\forall i, x_{i,i+1} < d$$

So

$$d_i = x_{i,i+1} < d$$

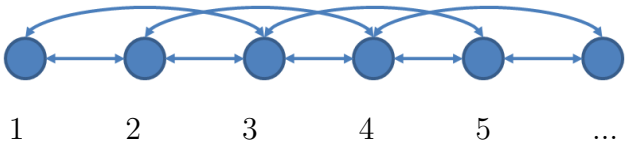


Figure 5.2: Interactions among particles (b).

Lemma 2 is proved.

Theorem 2(Sphere): Consider a group of N agents defined in Eq. refeq: system, $x_i, v_i \in \mathbb{R}^m (m = 1)$ with the same equivalent point d and scope r, if the following conditions is fulfilled,

1. The system is Lyapunov stable;

2. $\forall \varepsilon \in \mathcal{R}, \psi(x + \varepsilon) > \psi(x)$ and $\forall \varepsilon < \min(d, r - d)$, $\psi(d - \varepsilon) \gg \psi(\varepsilon - d)$;

3. $k \triangleq \frac{r}{d} > 2$;

4. $\forall i \in \mathcal{V}, x_{i,i+2} > d$.

the group forms a sphere (or circle for m=2), or $x_{i,j} > x_{ij-1},\ j \in \mathcal{V}$.

Proof: From Lemma 1, the forces on agent i are balanced,

$$\psi_{i,i-1} + \psi_{i,+} = \psi_{i,i+1} + \psi_{i,-} = 0 \tag{5.1}$$

where $\psi_{i,+}$ and $\psi_{i,-}$ is defined as summation of force from agent i to its non-adjacent neighboring agents.

$$\psi_{i,+} \triangleq \psi_{i,j}, j \in \mathcal{V}, j = i + 2, i + 3, \ldots$$

$$\psi_{i,-} \triangleq \psi_{i,j}, j \in \mathcal{V}, j = i - 2, i - 3, \ldots$$

when $x_{1i} < r$, in the negative direction, the number of neighboring agents are less than the positive direction. From Lemma 2, $d_i < d$ and $\forall \varepsilon \in \mathcal{R}, \psi(x + \varepsilon) > \psi(x)$. We get

$$\psi_{i,+} < \psi_{i,-}$$

Combined with Eq .5.1, we get

$$\psi_{i,i-1} > \psi_{i,i+1}$$

The corresponding interval distance is

$$x_{i,i-1} > x_{i,i+1}$$

when $x_{1i} \geq r$, the number of neighboring agents are the same than the positive direction. However, $x_{i,i-1} > x_{i,i+1}$ is maintained while $x_{1i} < r$. Still have

$$\psi_{i,+} < \psi_{i,-}$$

The corresponding interval distance is

$$x_{i,i-1} > x_{i,i+1}$$

One particle has only one neighboring particle in repulsive range in any possible direction

5.3 Performance Analysis

In order to quantitatively analyze the dynamics of the different patterns, we adopt two indexes, namely d_a and v_a, to measure the average neighboring distance and average velocity, respectively, as below,

$$d_a = \frac{\sum_{i\in\mathcal{V}} \sum_{j\in\mathcal{N}_i} \|x_i - x_j\|}{\sum_{i\in\mathcal{V}} \sum_{j\in\mathcal{N}_i} \|N_i\|}, \tag{5.2}$$

$$v_a = \frac{\| \sum_{i\in\mathcal{N}_i} v_i\|}{\sum_{i\in\mathcal{N}_i} \|v_i\|} \tag{5.3}$$

with $s_{i,j}$ give in Eq. (2.1). Clearly, the value $v_a \to 1$ and $\dot{d}_a \to 0$ as the velocities of the particles achieve synchronization, so both v_a and $\dot{d}_a$ can be regarded as an *order parameter*. Note that $\dot{d}_a$ demonstrates the evolution of the average

inter-particle distance, thus it contains more information than v_a and we display both v_a and $\dot{d}_a$ in Figs. 5.3 and 5.5, respectively. Indeed, due to the periodical boundary condition, the particles can communicated with the other ones for sufficient times, and hence the particles in all these three patterns will eventually reach a synchronized velocity [12], which is also verified by Fig. 5.3. Moreover, it is also exhibited that the synchronization procedure of the liquid-like pattern is quicker than that of the crystal-like one. The underlying reason roots in that the former has larger individual vision scope and tighter clustering formation, which implies more connections in the proximity net who accelerate the consensus procedure [12].

One can understand more deeply about the dynamics of the system from the evolution of the average neighboring distance d_a and its derivative $\dot{d}_a$ in Figs. 5.4 and 5.5, respectively. For the liquid-like pattern, since the average distance d_a is much smaller (see Fig. 3.3(d)) than that of the crystal-like one (see Fig. 3.4(d)), its average distance derivative $\dot{d}_a$ will experience a negative value during a long period until reaching a sufficient small d_a in Fig. 5.4, which well explains the negative overshooting of the liquid-like pattern in Fig. 5.5. Afterwards, its $\dot{d}_a$ value settles down to zero quicker than that of crystal-like pattern due to its larger number of neighboring connections induced by larger individual vision scope and tighter clustering formation, which reveals the distinct forming procedures of the different aggregation patterns.

The following figures show the dynamics of order parameters, with $N = 200$ and $L = 14.1$. All results are the average over 100 independent runs. All the other settings are the same as Fig. 3.3.

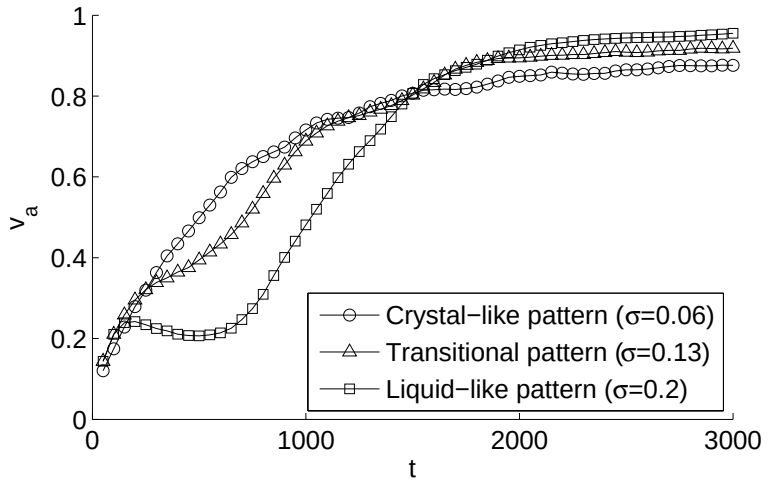


Figure 5.3: The average velocity v_a of particles.

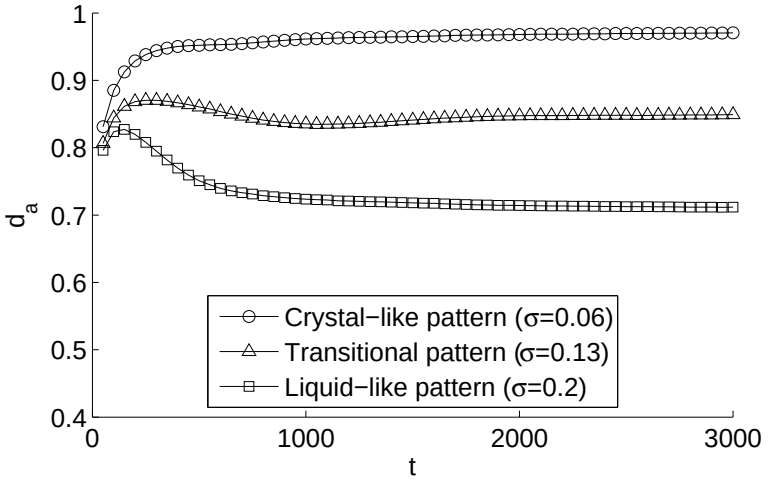


Figure 5.4: The evolution of average neighboring distance d_a.

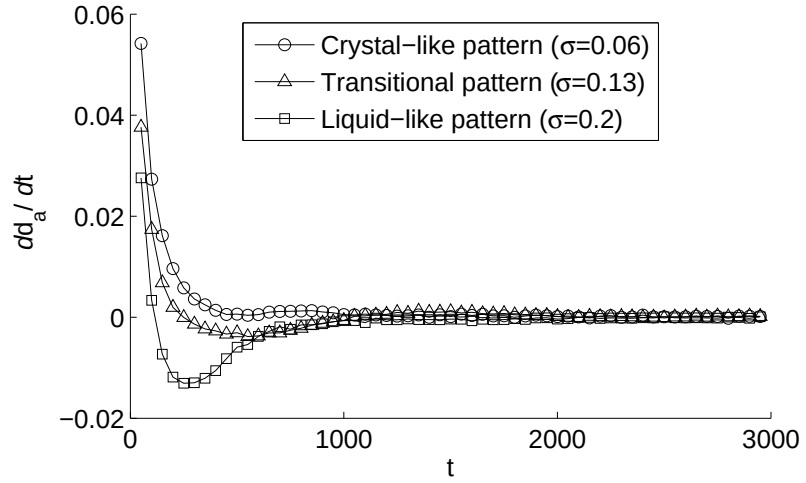


Figure 5.5: The evolution of the derivative of average neighboring distance $\dot{d}_a$.

CHAPTER 6

APPLICATION OF COLLECTIVE BEHAVIOR

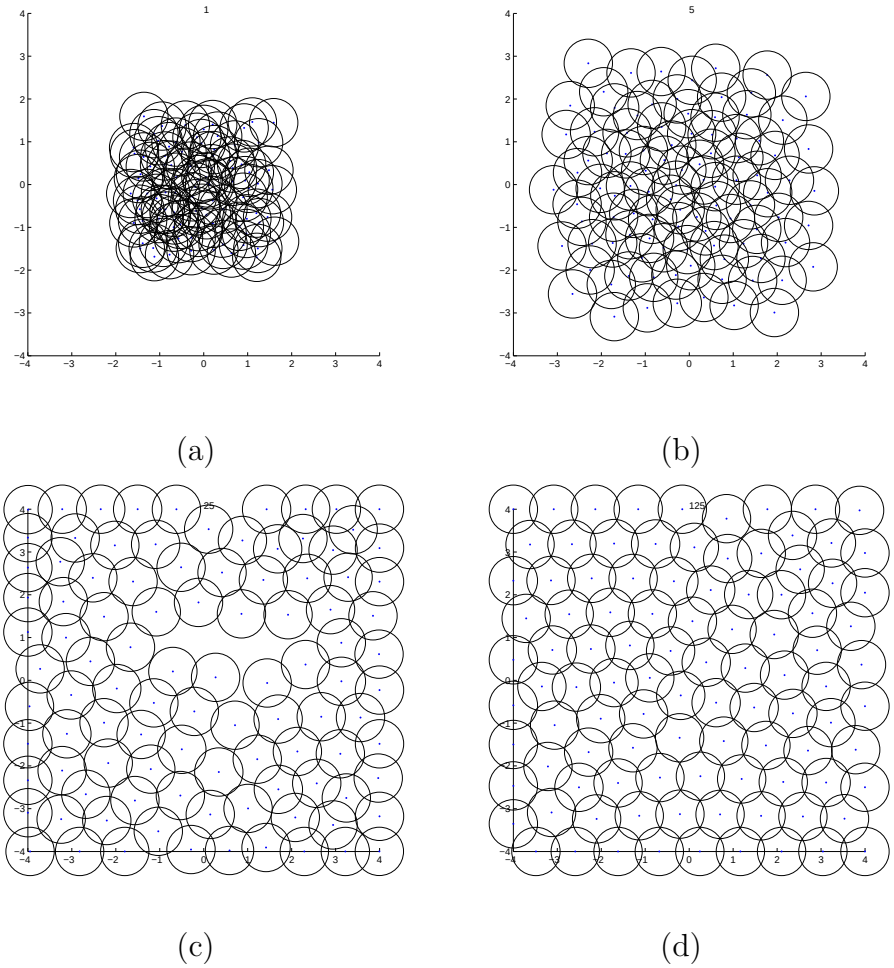
Collective behavior has applications in large areas, one of which is in the mobility of sensor networks. This chapter gives introduction and simulations on mobile sensor network and demonstrate the capability in spatial coverage and obstacle avoidance. Figs. 3.1 and 3.2 show the capability of spatial movement and avoiding multiple stand-alone obstacles.

6.1 Mobile Sensor Networks

We design the following simulations: in a two-dimensional square with boundaries, N particles are generated, whose initial locations and velocities are selected randomly within this square. Then all particles in this region evolve according to Eq. (2.6) and the states of all particles are updated synchronously. The parameter c is set as 5 so that the ratio k is close to 1.

Figure 6.2 shows the agents expanding and filling the whole area. In the figure, each point represents a particle and the edge connected two particles is the connection between neighboring particle pair. (a) The agents are initialized and randomly distributed in the center of the square area. (b) The position of agents start to expand and more area are covered after 50 steps. (c) After 250 steps, the agents reached to the boundary of this square and cover most of the area. (d) After 1,250 steps, agents cover al the area and stop moving. Positions

of all the agents are connected and synchronized.



(a) (b)

(c) (d)

Figure 6.1: Agents span and fill a square area.

Although Fig. 3.2, already showed the capability of avoiding multiple standing obstacles. It is still not clear whether the agents could expand the cover the area as complex as inside a building. In this case, a square area are divided into two parts with a wall, the wall is defined as elastic boundary, when agents encountered it, they will be assigned a backward speed with elastic force. The divided two parts are only partially split by the wall with a narrow entrance connected with these two, see Fig. 6.2. Agents are initialized in the right side of this area, they are expected to find the entrance and cover as large area as they can in

the simulation. (a) The agents are initialized and randomly distributed in the up right of the square area. (b) The position of agents start to expand and more area are covered after 50 steps. (c) After 250 steps, the agents reached the boundary and cover all area of the right part of the square area. (d) After 1,000 steps, agents reached the boundary of left side of the square area. (e) more area are covered after 2,000 steps. (f) After 4,000 steps, agents cover most of the square area and stop moving.

6.2 Biofilms

A biofilm is an aggregate of microorganisms. Cells in this aggregate adhere to each other or to a surface. Recently, the formation of biofilms are drawing more attention. A ship with biofims formed on the surface will have larger resistance when sailing in the water. Its steer surface may also be damaged by biofims. In microbiology, the growth of bacterial biofims may reduce the effect of antibacterial medicine. However, it will be an effective solution by introducing the research of collective behavior into the control of formation of biofilms.

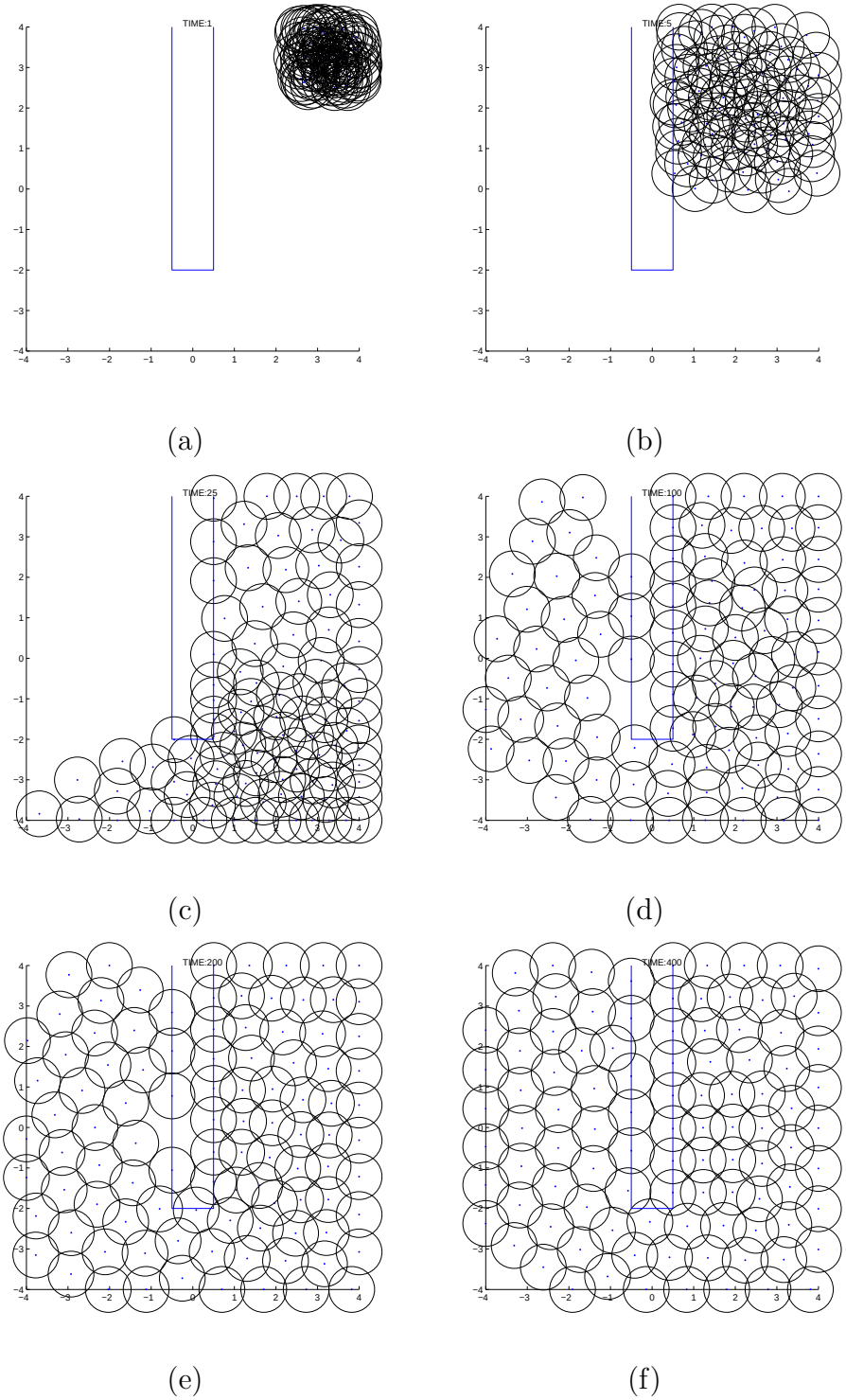


Figure 6.2: Agents span and fill a partially blocked area.

CHAPTER 7

CONCLUSION AND FUTURE WORK

All the research progress in this article is showed in the previous chapters. In this chapter, a summery is presented to review the major accomplishment of this thesis. Those goals that are not fully achieved and some valuable future work are also suggested.

7.1 Conclusion

This research has done both theoretical foundation and numerical simulation work for design and analysis of a multi-agent dynamic system on collective formation. A best Mass Model is chosen from three popular existing models. The control algorithm of the model is carefully designed with a pairwise potential function. The system created with the given model is proved to be Lyapunov stable. About the formation, two theorems are proposed and proved so as to explain how and why liquid-like pattern and crystal-like pattern exist. In the performance analysis, we know a system with the liquid-like pattern ends up in a more stable state but that with the crystal-like pattern is faster in synchronization. simulations and case study is also presented to show the capability of this research. Systems based on this model can form a flock/swarm, passing obstacles, spanning in a bounded area and entering a room.

7.2 Future Work

There are several interesting topics this thesis dose not include. One is the proof of theorems in Chapter 5. The proof presented in this chapter are with a strict condition - system in one dimension environment. A expansion to higher dimension will be valuable implementing. The next one, further comparison between liquid-like pattern and crystal-like pattern may be done in the future. And also the benefits and defect of these two patterns can be studied. At last, Design and control of the self-organizing dynamic system can be applied on sensor network data fusion, attitude alignment of satellite clusters and congestion control of communication networks to see the performance and features.

References

[1] T. Vicsek, A. Czirok, E. Ben-Jacob, I. I. Cohen, and O. Shochet, "Novel type of phase transition in a system of self-driven particles," *Phys Rev Lett*, vol. 75, no. 6, pp. 1226–1229, 1995.

[2] G. Gregoire and H. Chate, "Onset of collective and cohesive motion," *Phys Rev Lett*, vol. 92, no. 2, p. 025702, 2004.

[3] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.

[4] H.-T. Zhang, M. Z. Chen, G.-B. Stan, T. Zhou, and J. M. Maciejowski, "Collective behavior coordination with predictive mechanisms," *IEEE CIRCUITS AND SYSTEMS MAGAZINE*, vol. 3, pp. 65–87, 2008.

[5] H.-T. Zhang, M. Z. Q. Chen, and T. Zhou, "Improve consensus via decentralized predictive mechanisms," *EPL*, vol. 86, no. 4, pp. –, 2009.

[6] I. F. Akyildiz and H. Rudin, "Computer networks, the journal, and computer networks, the technology," *Computer Networks-the International Journal of Computer and Telecommunications Networking*, vol. 40, no. 4, pp. 475–475, 2002.

[7] P. Ogren, 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, 2004.

[8] D. P. Bertsekas and J. N. Tsitsiklis, "Comments on "coordination of groups of mobile autonomous agents using nearest neigbbor rules"," *IEEE Transactions on Automatic Control*, vol. 52, no. 5, pp. 968–969, 2007.

[9] K. Ma, Y. Y. Zhang, and W. Trappe, "Managing the mobility of a mobile sensor network using network dynamics," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 1, pp. 106–120, 2008.

[10] C. W. Reynolds, "Flocks, herds, and schools: A distributed behavioral model," *Computer Graphics*, vol. 21, no. 4, pp. 25–34, 1987.

[11] V. Gazi and K. M. Passino, "Stability analysis of swarms," *IEEE Transactions on Automatic Control*, vol. 48, no. 4, pp. 692–697, 2003.

[12] 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, 2004.

[13] W. Ren and R. W. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Transactions on Automatic Control*, vol. 50, no. 5, pp. 655–661, 2005.

[14] 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, no. 7025, pp. 513–6, 2005.

[15] 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. –, 2006.

[16] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Transactions on Automatic Control*, vol. 50, no. 2, pp. 169–182, 2005.

[17] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[18] 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, 2007.

[19] R. Sepulchre, D. A. Paley, and N. E. Leonard, "Stabilization of planar collective motion: All-to-all communication," *Automatic Control, IEEE Transactions on*, vol. 52, no. 5, pp. 811–824, 2007.

[20] ——, "Stabilization of planar collective motion with limited communication," *Automatic Control, IEEE Transactions on*, vol. 53, no. 3, pp. 706–719, 2008.

[21] C. M. Breder, "Equations descriptive of fish schools and other animal aggregations," *Ecology*, vol. 35, no. 3, pp. 361–370, 1954.

[22] K. Warburton and J. Lazarus, "Tendency distance models of social cohesion in animal groups," *Journal of Theoretical Biology*, vol. 150, no. 4, pp. 473–488, 1991.

[23] P. Schlottmann, "Integrable 2-band model with attractive and repulsive interactions," *Physical Review Letters*, vol. 68, no. 12, pp. 1916–1919, 1992.

[24] B. I. Lev, "Exactly solvable three-dimensional lattice model with attractive and repulsive interactions," *Physical Review E*, vol. 58, no. 3, pp. R2681–R2684, 1998.

[25] A. Melzer, V. A. Schweigert, and A. Piel, "Transition from attractive to repulsive forces between dust molecules in a plasma sheath," *Physical Review Letters*, vol. 83, no. 16, pp. 3194–3197, 1999.

[26] A. J. Archer and N. B. Wilding, "Phase behavior of a fluid with competing attractive and repulsive interactions," *Physical Review E*, vol. 76, no. 3, pp. –, 2007.

[27] V. Gazi and K. M. Passino, "Stability analysis of social foraging swarms," *IEEE Trans Syst Man Cybern B Cybern*, vol. 34, no. 1, pp. 539–57, 2004.

[28] Y. L. Chuang, M. R. D'Orsogna, D. Marthaler, A. L. Bertozzi, and L. S. Chayes, "State transitions and the continuum limit for a 2d interacting, self-propelled particle system," *Physica D-Nonlinear Phenomena*, vol. 232, no. 1, pp. 33–47, 2007.

[29] L. Q. Peng, Y. Zhao, B. M. Tian, J. Zhang, B. H. Wang, H. T. Zhang, and T. Zhou, "Consensus of self-driven agents with avoidance of collisions," *Physical Review E*, vol. 79, no. 2, p. 6, 2009.

[30] T. Vicsek, "Universal patterns of collective motion from minimal models of flocking," in *Self-Adaptive and Self-Organizing Systems, 2008. SASO '08. Second IEEE International Conference on*, 2008, pp. 3–11.

[31] D. E. O. Juanico, "Self-organized pattern formation in a diverse attractive-repulsive swarm," *EPL*, vol. 86, no. 4, pp. –, 2009.

[32] M. Z. Q. Chen, Z. Cheng, H.-T. Zhang, T. Zhou, and I. Postlethwaite, "Collective aggregation pattern dynamics control via attractive/repulsive function," in *Complex Sciences*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, J. Zhou, Ed. Springer-Verlag, 2009, vol. 5, pp. 2064–2077.

[33] B. Nguyen, Y.-L. Chuang, D. Tung, C. Hsieh, Z. Jin, L. Shi, D. Marthaler, A. Bertozzi, and R. Murray, "Virtual attractive-repulsive potentials for co-

operative control of second order dynamic vehicles on the caltech mvwt," *Proc. 2005 Am. Control Conf*, pp. 1084–1089, 2005.

[34] N. Heo and P. K. Varshney, "Energy-efficient deployment of intelligent mobile sensor networks," *Ieee Transactions on Systems Man and Cybernetics Part a-Systems and Humans*, vol. 35, no. 1, pp. 78–92, 2005, 880RP Times Cited:65 Cited References Count:30.

[35] W. W. M. Rigby, E.B. Smith and G. Maitland, *The Forces Between molecules*, New York, 1986.

[36] N. March and J. Mucci, *Chemical Physics of Free Molecules*, New York, 1993.

[37] J. E. Jones, "On the determination of molecular fields - ii from the equation of state of a gas," *Proceedings of the Royal Society of London Series a-Containing Papers of a Mathematical and Physical Character*, vol. 106, no. 738, pp. 463–477, 1924.

[38] E. W. Justh and P. S. Krishnaprasad, "Equilibria and steering laws for planar formations," *Systems and Control Letters*, vol. 52, no. 1, pp. 25–38, 2004.

[39] H. K. Khalil, *Nonlinear Systems*, New York, 1992.

APPENDIX A

SIMULATION PLATFORM

In order to design simulations of collective motion in different initial conditions, a platform is designed to provide a graphic use interface. There are four sections in this platform: the menu section, the configuration section, the execution section and the display section. The menu section is on the top of this platform, it has four items. Those items are used to achieve file, edit, display and help functionality respectively. And the configuration section is responsible to set each of the listed parameters and configure the model. After everything is set, the platform is ready to run. To execute the model, simply push the"Run" in the Execution section and the result will appear in Display section. Resource code is available in Appendix B.

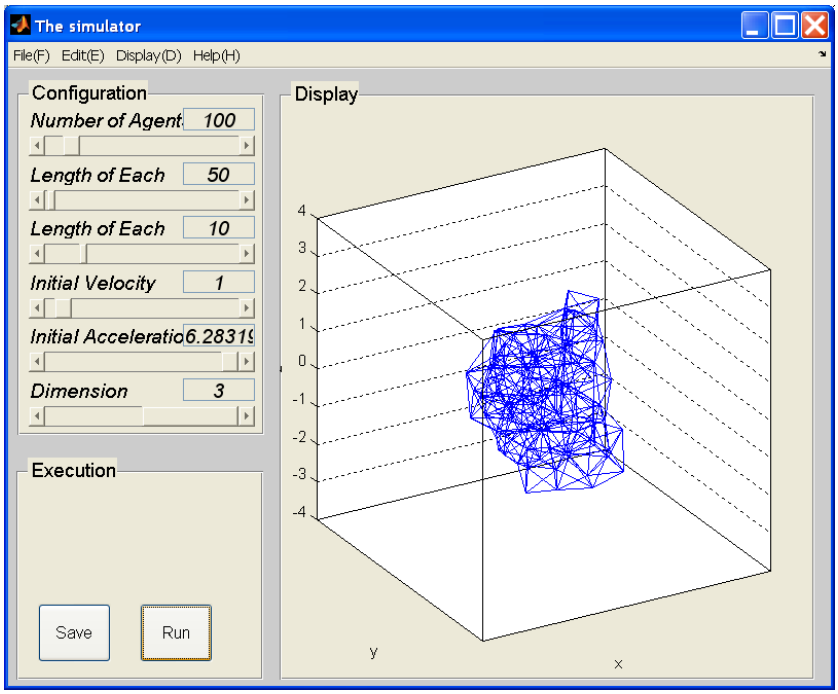


Figure A.1: The GUI of simulation platform.

APPENDIX B

SOURCE CODE

B.1 Source Code for Graphic Use Interface

```
% main.m
%
% Description: The main entrance to the GUI.
% Written by : Zhao Cheng
% Last update: June 1, 2010


close all
FONTSIZE=12;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Figure
figure('Position',[5,39,800,600],...
'Name','The simulator',...
'Menubar','none','Resize','on','pointer','arrow',...
'numbertitle','off','SelectionType','open');
% crosshair — arrow — watch — topl —
% topr — botl — botr — circle — cross —
% fleur — left — right — top — bottom —
% fullcrosshair — ibeam — custom

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Parameter Zone

PI=3.1415926;
try
```

```
load config.mat confValue
paraval(:,3) = confValue;
catch me
confValue = [ 100, 1000,10,1,2*PI,2];
paraval(:,3) = confValue;
%throw me;
end
paraval(:,2)=[ 1, 1, 1, 0, 0,2];%Min value
paraval(:,1)=[ 1000,5000,50,20,2*PI,3];%Max value
paraval(:,4)=[ 10/999,10/4999,0.1/49,0.01,0.01,0.1];%Step1
paraval(:,5)=[100/999,50/4999, 1/49, 0.1, 0.1, 1];%Step2
dataccs( 1).name='Number of Agents';
dataccs( 2).name='Length of Each Run';
dataccs( 3).name='Length of Each Step';
dataccs( 4).name='Initial Velocity';
dataccs( 5).name='Initial Acceleration';
dataccs( 6).name='Dimension';
dataccs( 7).name='5';
for i=1:length(paraval(:,1))
dataccs(i).value=zeros(1,5);
dataccs(i).value(:)=paraval(i,:);
end
AdvSet.TEXT_W=150;
AdvSet.EDIT_W=70;
AdvSet.HEIGHT=52.5;
parabars(10,245,length(paraval(:,1)),dataccs,AdvSet);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Execution Zone
hp_func = uipanel('Title','Execution',...
'Position',[0.01 0.01 0.3 0.36],...
'FontSize',FONTSIZE);
% Popups
% POPUP_WIDTH=160;
% uicontrol('Parent',hp_func,'style','popup','Tag','pop1',...
% 'Units','normalized','Position',[0.08,0.85,0.7,0.1],...
% 'Fontsize',FONTSIZE-2,'String','choose one',...
```

```
% 'Callback','call select');
%
% uicontrol('Parent',hp_func,'style','popup','Tag','pop2',...
% 'Units','normalized','Position',[0.08,0.70,0.7,0.1],...
% 'Fontsize',FONTSIZE-2,'String','choose one',...
% 'Callback','call select');

% PushButton
uicontrol('Parent',hp_func,'style','pushbutton',...
'Units','normalized','Position',[0.08,0.08,0.3,0.3],...
'String','Save','Fontsize',FONTSIZE-2,...
'Callback',...
['h=findobj(''Style'',''slider'');'...
'confValue = get(h,''Value'');'...
'confValue=fliplr( [confValue:] );'...
'save config.mat confValue;']);
uicontrol('Parent',hp_func,'style','pushbutton',...
'Units','normalized','Position',[0.5,0.08,0.3,0.3],...
'String','Run','Fontsize',FONTSIZE-2,...
'Callback','runmodel true');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Display Zone
hp_plot = uipanel('Title','Display',...
'Position',[0.33 0.01 0.65 0.97],...
'FontSize',FONTSIZE);
% Axes1=axes('Parent',hp_plot,'Tag','axes1','FontSize',FONTSIZE-3,...
% 'Position',[0.03 0.03 0.93 0.93],...
% 'Units','normalized','XTick',[],'YTick',[],'box','on');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MENU
% 'Color' ,'blue'
% 'NumberTitle' ,'off'
% 'Name' ,'My Figure'
% 'Menubar' ,'none'/'figur'
File=uimenu('Label','File(&F)');
```

```
% uimenu(File,'Label','Open(&O)','Callback','uigetfile');
% uimenu(File,'Label','Save(&S)','Callback','uiputfile');
uimenu(File,'Label','Import results','Callback','call import results');
uimenu(File,'Label','Import library','Callback','call import library');
uimenu(File,'Label','Export(&E)','Callback','call export');
uimenu(File,'Label','Config(&C)','Callback','config');
uimenu(File,'Label','Quit(&Q)','callback','close(gcf)');
Edit=uimenu('Label','Edit(&E)');
uimenu(Edit,'Label','grid on','CallBack','call com gridon');
uimenu(Edit,'Label','grid off','CallBack','call com gridoff');
uimenu(Edit,'Label','Pointer','CallBack','call com mmxy');
Disp=uimenu('Label','Display(&D)');
L2Examp1=uimenu(Disp,'Label','Sorted Plot','CallBack','call draw');
L2Examp2=uimenu(Disp,'Label','Cluster Tree','CallBack','call draw1');
Help=uimenu('Label','Help(&H)','CallBack','');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% END
```

B.2 Source Code for Parameter Configuration

```
% runmodel.m
    %
    % Description: Run the model.
    % Written by : Zhao Cheng
    % Last update: June 2, 2010

    function runmodel
    (AgentNum,RunTimeNum,RunStepLen,AxisPos,InitVel,InitAcc,Dimension,Obstacle)
    if nargin==0
    clc
    clear
    close all

    RunStepLen=10;Obstacle=1;
    SaveFlag=0;
    Dimension=3;
```

```
    AgentNum =100; RunTimeNum= 4000;
    InitVel = 1; InitAcc = 2*pi;
    AXISWIDTH = sqrt(AgentNum)*0.8;
    AxisPos = [-AXISWIDTH/2,AXISWIDTH/2,-AXISWIDTH/2,AXISWIDTH/2,-
AXISWIDTH/2,AXISWIDTH/2];
    eval('javaaddpath class/CollectiveBehavior.jar; import model;');
    figure(1),set(gcf,'position',[5,25,1.08*700,700])
    elseif nargin==1
    try
    load config.mat confValue
    catch me
    error('check configuration');
    end
    confValue
    AgentNum = confValue(1);
    RunTimeNum = confValue(2);
    RunStepLen = confValue(3);
    InitVel = confValue(4);
    InitAcc = confValue(5);
    Dimension = confValue(6);
    Obstacle=1; SaveFlag=0;
    AXISWIDTH = sqrt(AgentNum)*0.8;
    AxisPos = [-AXISWIDTH/2,AXISWIDTH/2,-AXISWIDTH/2,AXISWIDTH/2,-
AXISWIDTH/2,AXISWIDTH/2];
    eval('javaaddpath class/CollectiveBehavior.jar; import model;');
    h = findobj('Title','Display');
    FONTSIZE = get(h,'fontsize');
    delete(findobj('Parent',h))
    axes('Parent',h,'Tag','axes1','FontSize',FONTSIZE-3,...
    'Position',[0.03 0.03 0.93 0.93],...
    'Units','normalized','XTick',[],'YTick',[],'box','on');
    end

    if SaveFlag==1,SaveFilm('start',RunTimeNum,'1.avi');end

    thisModel = model();
```

```
% ParaSet[0] = (double)DM;
% ParaSet[1] = PARA_d;
% ParaSet[2] = PARA_c;
% ParaSet[3] = PARA_dt;
% ParaSet[4] = PARA_df;
% ParaSet[8-13] =X-Y-Z;

switch(Dimension)
case 2
X=0;Y=0;hold on
ParaSet=[2,1.0, 0.05, 0.02, 1, 0,0,0, AxisPos, 3, InitVel, InitAcc];
thisModel.creatAgent(AgentNum,ParaSet);
for RunTime=1:RunTimeNum
% RunTime
thisModel.runModel(RunStepLen,Obstacle);
pos = thisModel.getPos();
vel = thisModel.getVel();
% adj = thisModel.getAdj(ParaSet(2));
aveDis(RunTime) = thisModel.getAveDis(ParaSet(2));
aveVel(RunTime) = thisModel.getAveVel();
adj = thisModel.getAdjShow(ParaSet(2));
cla
plot(pos(:,1),pos(:,2),'.b'); % draw non-leader
plot(pos(:,1),pos(:,2),'or','LineWidth',0.5,...
'MarkerEdgeColor','k',...
'MarkerFaceColor','none',...
'MarkerSize',50)
axis([X+AxisPos(1),X+AxisPos(2),Y+AxisPos(3),Y+AxisPos(4)])
text(-0.015*(AxisPos(2)-AxisPos(1)),(AxisPos(4)-AxisPos(3))/2*(1.02),
    sprintf('TIME:%d',RunTime))
% % plot edge
% for i=1:AgentNum
% for j=1:AgentNum
% if(adj(i,j)==1)
% plot([pos(i,1),pos(j,1)],[pos(i,2),pos(j,2)],'b');
% end
```

```
% end
% end

% % plot trajactory
% Pos(RunTime,:,:) = pos;
%
% for j=1:AgentNum
% plot(Pos(:,j,1),Pos(:,j,2),'b');
% end

plot([0.5 0.5 -0.5 -0.5],[4 -2 -2 4])
drawnow
if SaveFlag==1,SaveFilm('save'),end
end


case 3
X=0;Y=0;Z=0;
xlabel('x');ylabel('y');zlabel('z');
axis square; box on; grid on;
view(-30,25)
ParaSet=[3,1.0, 0.05, 0.02, 1, 0,0,0, AxisPos, 3, InitVel, InitAcc];
thisModel.creatAgent(AgentNum,ParaSet);
for RunTime=1:RunTimeNum
thisModel.runModel(5);
pos = thisModel.getPos();
vel = thisModel.getVel();
adj = thisModel.getAdjShow(ParaSet(2));
%dir = vel./repmat(sqrt(sum(vel.^2,2)),1,3);% normalize
cla
hold on
%scatter3(pos(:,1),pos(:,2),pos(:,3),10,'b','filled');
% quiver3(pos(:,1),pos(:,2),pos(:,3),dir(:,1),dir(:,2),dir(:,3),...
% 0.15,'b',...'filled',...
% 'LineWidth',1.5,'MarkerFaceColor','b','MarkerSize',6)
for i=1:AgentNum
```

```
for j=1:AgentNum
if(adj(i,j)==1)
plot3([pos(i,1),pos(j,1)],[pos(i,2),pos(j,2)],[pos(i,3),pos(j,3)],'b');
end
end
end

plot3(pos(:,1),pos(:,2),pos(:,3),'.b'); % draw non-leader
axis([X+AxisPos(1),X+AxisPos(2),Y+AxisPos(3),Y+AxisPos(4),...
    Z+AxisPos(5),Z+AxisPos(6)])
text(0,(AxisPos(4)-AxisPos(3))/2*(1.1),(AxisPos(6)-AxisPos(5))/2*(1.1),...
    sprintf('TIME:%d',RunTime))
axis vis3d;
% str=sprintf('%3.1f',RunTime*Parameter(3))
% text(-5,AxisWidth/2+5,str);


%pause(0.1);
drawnow
if SaveFlag==1,SaveFilm('save'),end
end


otherwise
warning('do not have the case: dimension is 2 or 3');
end

if SaveFlag==1,SaveFilm('close'),end
```

B.3 Source Code for Multi-agent System Model

```
% MADS_acc1_obstacle.m
    %
    % Description: The main entrance to the GUI.
    % Written by : Zhao Cheng
    % Last update: June 7, 2010
```

```
function MADS_acc1_obstacle(RunTimeNum)
%
%clc
%clear
%close all

if nargin==0
RunTimeNum=1200;
end

SaveFlag=0;
sym i;
AgnetNum=100;
AxisWidth=100;%140
pos = random('Normal',0,40,1,AgnetNum)+i*random('Normal',0,40,1,AgnetNum);
vel = random('Normal',0, 0.1,1,AgnetNum)+i*random('Normal',0, 0.1,1,Ag-
netNum);
ObsNum=0;RadObs=[20+8 5+8 10+8];
PosObs = [90-20*i,90+20*i,130+8*i];
RadPar=2.5
ho=0.1;eo=0.1;
posGoal = 180 - 0*i;
%posGoal = 140 - 0*i;
velGoal = 0 + 0*i;
X=0;Y=0;

figure(1)
set(gcf,'position',[5,35,1.08*700,700])
plot(0,0,'black')

h=0.2;
d=8;%global minimum position
r=1.2*d;%1.2*d %cut off
e=0.1;
%%%%%%%%%%%%%%%%%%???
a=50;b=d2̂;c=0.05;
```

```
if SaveFlag==1,SaveFilm('start',1000,'1.avi');end
for RunTime=1:RunTimeNum
AgentCentre=0+0i;
for I1=1:AgnetNum
acc(I1)=0;
for I2=1:AgnetNum
if I1==I2,continue;end
x=pos(I2)-pos(I1);
% y=BumpFunc(h,abs(x)/r)*SigmFunc(abs(x)-d);
% cons = 2e-4*(posGoal-pos(I1)) + 1e-3*(velGoal-vel(I1));
% cons = 1e-3*(velGoal-vel(I1));
% acc(I1) = acc(I1) + y*x/sqrt(1+e*abs(x)^2) + cons ;% +...
% BumpFunc(h,abs(x)/r)*(vel(I2)-vel(I1));
% y=BumpFunc1(abs(x)-sqrt(b))*ARFunc(abs(x),a,b,c);
y=ARFunc(abs(x),a,b,c);
cons = 2e-4*(posGoal-pos(I1)) + 5e-4*(velGoal-vel(I1));
acc(I1) = acc(I1) + y*x/sqrt(1+eo*abs(x)^2) + cons;
%BumpFunc(h,abs(x)/r)*(vel(I2)-vel(I1));
end
%
for I=1:ObsNum
do=RadObs(I);ro=1.1*do;
x=PosObs(I)-pos(I1);
y=BumpFunc(ho,abs(x)/ro)*SigmFunc(abs(x)-do);
acc(I1) = acc(I1) + y*x/sqrt(1+eo*abs(x)^2);
end
end
for I1=1:AgnetNum
vel(I1)=vel(I1) + acc(I1);
pos(I1)=pos(I1) + 0.01*vel(I1);
%hold on,plot(real(pos(I1)),imag(pos(I1)),'ro')
AgentCentre = AgentCentre+pos(I1);
end
AgentCentre=AgentCentre/AgnetNum;
delete(gca)
hold on
```

```
% plot(agent(1,:),agent(2,:),'.')
plot(real(pos(:)),imag(pos(:)),'.b');
%draw obstacle
for I=1:ObsNum
plot(real(PosObs(I)),imag(PosObs(I)),'o',...
'LineWidth',1,...
'MarkerEdgeColor','k',...
'MarkerFaceColor','k',...
'MarkerSize',RadPar*(RadObs(I)-d))
end

%plot(real(AgentCentre),imag(AgentCentre),'or');
if 0 % axis moves with objects? 1:Yes;0:no.
if abs(X-real(AgentCentre))¿80
X=real(AgentCentre);
end
if abs(Y-imag(AgentCentre))¿80
Y=imag(AgentCentre);
end
else
X=real(AgentCentre);
Y=imag(AgentCentre);
end
axis([X-AxisWidth,X+AxisWidth,Y-AxisWidth,Y+AxisWidth])
edge = zeros(AgnetNum);
for I1=1:AgnetNum
for I2=I1+1:AgnetNum
if abs(abs(pos(I1)-pos(I2))-d)¡0.2*d,
edge(I1,I2)=1;
plot(real([pos(I1),pos(I2)]),imag([pos(I1),pos(I2)]));
end
end
end
set(gca,'fontsize',24)
xlabel('x(pos)'),ylabel('y(pos)')
drawnow
```

```
if(RunTime==1) pause;end
if SaveFlag==1,SaveFilm('save');RunTime,end
end

if SaveFlag==1,SaveFilm('close');end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function result=NormFunc(e,z)
result=(sqrt(1+e*z2̂)-1)/e;
function x=ARFunc(t,a,b,c)
x=a*(t.2̂./b-1)*exp(-c*t2̂);
function result=BumpFunc1(z)
if z¿=-1 && z¡=1
result=(1-cos(pi.*z))/2;
else
result=1;
end

function result=BumpFunc(h,z)
if z>0 && z<h
result=1;
elseif z>=h && z<=1
result=(1+cos(pi*(z-h)/(1-h)))/2;
else
result=0;
end

function result=SigmFunc(z)
e=0.1;
a=5;b=5;c=abs(a-b)/sqrt(4*a*b);
result=((a+b)*(z+c)/sqrt(1+e*(z+c)2̂)+a-b)/2;
```