

# Distributed shape control of homogeneous swarms of autonomous underwater vehicles

Shahab Kalantar · Uwe R. Zimmer

Published online: 31 August 2006  
© Springer Science + Business Media, LLC 2006

**Abstract** In this paper, we study large formations of underwater autonomous vehicles for the purposes of exploration and sampling the ocean surface. The formations or aggregates we consider are composed of up to hundreds of robots with the capability of forming various complex shapes dictated by the shape of the region to be explored, as well as special shapes suitable for migration. The shapes are determined through bathymetric maps and described with reduced-dimensional representation techniques. The approach we propose is that of breaking up the control and coordination strategy into two decoupled problems, i.e., partitioning the aggregate into two non-overlapping sets: its boundary and its interior. The boundary uses general theory of curve evolution to form shapes while the interior passively complies, using attraction-repulsion forces, to form a uniform distribution inside the boundary. This makes the problem much more tractable than previous methods. Decision making by individual robots is entirely based on local information, autonomous underwater vehicles, formation control, swarm control.

---

Part of this research was done under a grant from Cooperative Research Centre for Intelligent Manufacturing Systems and Technology (CRC-IMST), Australia.

---

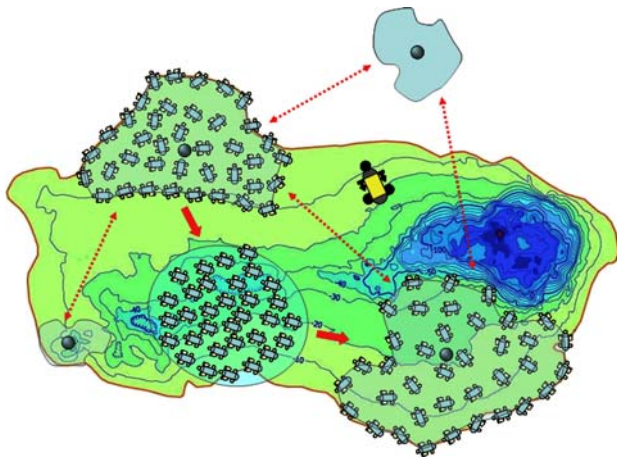
S. Kalantar (✉) · U. R. Zimmer  
Australian National University, Research School for Information Science and Engineering and the Faculty for Engineering and Information Technology,  
Canberra, ACT 0200, Australia  
e-mail: shahab.kalantar@rsise.anu.edu.au

U. R. Zimmer  
e-mail: uwe.zimmer@ieee.org

## 1. Introduction

Coordinated motion of groups of autonomous agents has been the subject of much research in the past few years. Most of the literature, either consider small formations (especially suited to aerospace applications, (Egerstedt and Hu, 2001; Zhang et al., 2003)) or swarms with no constraints on the boundary (mostly related to the study of biological organisms, (Gazi and Passino, 2004; Reif and Wang, 1999)). Only recently, researchers have started looking at large formations of robots resembling swarms but satisfying some geometrical characteristics (Belta and Kumar, 2002; Chaimowicz and Kumar, 2004; Cortes et al., 2004; Spletzer and Fierro, 2005; Chaimowicz et al., 2005). The primary application we have in mind is that of exploration of underwater environments requiring the realization of Autonomous Ocean Sampling Networks (AOSN) (Curtin et al., 1989). We are specially interested in mobile networks as opposed to static ones. In this case, autonomous submarines play the role of sensor platforms. They are equipped with special application-specific sensors as well as limited bandwidth communication capabilities. The robot aggregate is assumed to be homogeneous, consisting of identical robots. Also, interactions are limited to a small neighborhood around each robot. These local interactions should lead to the accomplishment of a global task.

In a typical underwater exploration task, a group of mobile autonomous vehicles are required to form particular shapes corresponding to underwater features and carry on some sampling process of the region. Features of interest underwater include, but are not limited to, regions with certain depth characteristics, usually represented by *bathymetric maps*, and *plumes* created as a result of emission of heat or matter from one or several sources and undergoing *diffusion*



**Fig. 1** Underwater exploration and migration

(Okubo, 1980). This latter example can include oil leakage (or other kinds of contamination) as well as explosions. Regions are usually connected together in a *topological map* of the environment. The robot aggregate needs to migrate through pathways from one node in the map to another. To do this, it may have to form particular shapes suitable for navigation and distribute again upon reaching the destination node. Figure 1 shows part of a typical scenario. In this paper, we address the issue of commanding the group to form desired shapes. We assume that signatures (shapes) of environmental features have been computed by suitable shape description methods. These signatures can be computed using groups of robots which have been adapted to the feature or they could be conveniently calculated using *bathymetric maps*, if available, or other means, such as using aerial vehicles. For more details, refer to Kalantar and Zimmer (2004).

To have a collection of robots form a certain shape we have to come up with means to describe geometric shapes. One inefficient way of doing this is to represent the position of each and every robot within the formation. This is certainly appropriate for a small number of robots such as in formation flying. Indeed, many methods have been proposed for representing small formations such as the concept of *virtual structures* (Ren and Beard, 2004). In Belta and Kumar (2002), a formation state is described as an element of an abstract manifold which is the Cartesian product of a Lie group  $G$  (an element of which represents the pose of the formation as a whole) and a shape manifold  $S$  which is a low-dimensional space representing the shape of the formation. They apply their theory to ellipsoid-shaped formations using the mean and variance as shape parameters. In this kind of approach, the positions of individual robots are not important as long as the whole group has a certain shape. This approach may prove to be very complex when it comes to describing more complex shapes. On the other hand, a multitude of canonical invariant representations have



**Fig. 2** Two of the Serafina underwater vehicles

been used by machine vision researchers over the past few decades, such as statistical moments, Fourier descriptors, differential invariants, integral invariants and wavelets, to name the most important ones. In this paper, we propose an alternative approach where the problem is decoupled into two much simpler problems: motion of robots at the boundary of the aggregate and appropriate distribution of interior robots within the confines of the boundary. The particular boundary representation we use here is Fourier descriptors which have been widely used in pattern recognition but others can also be used. Also, we consider uniform distribution for interior robots but non-uniform ones, such as the one described in Cortes et al. (2004), are also applicable.

The physical robots we consider here are small autonomous vehicles called *Serafina* (Fig. 2), which have been developed in our laboratory (for information about the technical aspects of the submersibles please have a look at the web-site (<http://users.rsise.anu.edu.au/~serafina/>)). They have limited communication, sensing as well as computational power. For a particular sampling operation, they will be equipped with special sensors. Each robot can only communicate with vehicles in its vicinity defined by a radius. In this paper, we use a very simple control system  $\dot{q}(t) = F(q(t), t)$ , basically treating Serafinas as holonomic vehicles, where  $q(t) = [x(t), y(t)]$  is the coordinates of the robot in the horizontal plane and  $F$  denotes the total force acting on the robot. The robots are stabilised at a certain fixed distance from the bottom of the ocean using  $\dot{z}(t) = K_z(z(t) - z_d)$ , so that we only consider motion in the  $(x, y)$  plane.

The paper is organized as follows. First, a very general global overview of our approach is presented. We then present a review of reduced-dimensional shape representations. Then, methods for effecting uniform distribution are discussed in Section 4. Separation of the group into interior and boundary robots is discussed in Section 5. After that, target shape reconstruction and the associated potential (Section 6), movement of the boundary as well as interior robots (Sections 7 and 8), are dealt with. Next, some special cases are considered. Finally, simulation results are presented.

## 2. Overview

Suppose that we are given a collection  $S$  of  $N$  robots  $R_i, i = 1, \dots, N$ , represented as points in  $\mathbb{R}^2$  with their positions denoted as  $q_i(t) = (x_i(t), y_i(t))$  ( $q_i(t)$  can be interpreted as the state of  $R_i$ ). The positions are measured with respect to a fixed global coordinate system or a local one moving with the robots. The state of the system  $S$  can be described by  $q(t) = (q_1(t), \dots, q_N(t)) \in \mathbb{R}^{2N}$ . We call  $S$  an *aggregate*. Each robot is uniquely identified by its label. The *centre of gravity* of an aggregate is defined as

$$\bar{q} = \frac{1}{N} \sum_{i=1}^N q_i. \tag{1}$$

A local coordinate system can be conveniently placed at  $\bar{q}$ . From now on, we assume (although not entirely necessary) that all the distances are expressed in these local coordinates.

An aggregate can be partitioned into two non-overlapping sets:  $S = I_S \cup B_S$ .  $B_S$  denotes the set of robots forming the boundary of the shape described by  $S$ , while  $I_S$  is the set of robots inside the boundary. For this definition to be meaningful, the region defined by  $S$  should be connected, in the topological sense of the word.

The robots we consider here are homogeneous, meaning that they can equally well belong to  $B_S$  or  $I_S$ , and that they can be anywhere on the boundary or inside. Any two of the robots can thus be interchanged.

Due to communication range limitations, all the processes we consider are local. Depending on which set a robot belongs to and which particular process it is involved in at a certain time instant, various *neighborhoods* may be defined. A neighborhood  $\mathcal{N}_i$  of each robot  $R_i$  defines a circular region, with radius  $r(\mathcal{N}_i)$  around it and is defined as the set of robots  $R_j$  such that  $\|q_i(t) - q_j(t)\| \leq r(\mathcal{N}_i)$ . Two robots are called *neighbors (connected)* if both of them belong to the other robot’s neighborhood.

Boundaries can be conveniently represented by continuous closed curves  $\gamma : [0, 1] \rightarrow \mathbb{R}^2$ , parametrized by arc-length  $s$ .  $\gamma(s)$  denotes the position of the point on the boundary  $s$  distance away from the starting position  $\gamma(0)$  and  $\gamma(1) = \gamma(0)$ . In our case, the boundary is actually a polygon each vertex of which denotes the position of a boundary robot. If an order  $l_B : B_S \rightarrow \{0, 1, \dots, |B_S| - 1\}$  is defined on  $S$ , then  $\gamma(l_B(R)/(|B_S| - 1))$  defines the position of  $R \in B_S$ .

In our approach, the robots are initially randomly distributed in a region. They start by moving according to a rule which tries to make the region occupied by  $S$  as close as possible to a convex shape, with robots distributed uniformly inside it:

$$\dot{q}(t) = \frac{dq(t)}{dt} = \mathcal{F}_S(q(t)) \tag{2}$$

As will be shown later,  $\mathcal{F}_S$  is a function of the attraction-repulsion forces exerted on individual robots and is computed as the gradient of a potential  $\Phi(q(t))$ . After stabilization, boundary selection process can commence. Equilibrium can be determined using

$$\left\| \Phi(q(t)) - \frac{1}{T} \int_{t-T}^t \Phi(q(t)) dt \right\| \leq \varepsilon_\Phi \tag{3}$$

In some cases, the process may also commence when the energy level falls below some limit:

$$\Phi(q(t)) \leq \varepsilon_D \tag{4}$$

In its most general case, boundary assignment is a statistical process. Denote by  $P_B(R_i, t)$  the probability of  $R_i \in B_S$  at time  $t$ . Then at a certain time  $T$  marking the end of the process, we should have  $P_B(R_i, T) = 1$  for  $R_i \in B_S$  and  $P_B(R_i, T) = 0$  for  $R_i \in I_S$ . Moreover, left and right neighbors of each boundary robot should be determined by this process. Accordingly, we will have a set of coupled differential equations:

$$\begin{aligned} \partial P_B(R_i, t) / \partial t &= \hat{P}_B(S, t) \\ \partial P_l(R_i, R_j, t) / \partial t &= \hat{P}_l(S, t) \\ \partial P_r(R_i, R_j, t) / \partial t &= \hat{P}_r(S, t) \end{aligned} \tag{5}$$

$P_l(R_i, R_j, t)$  represents the probability that  $R_j$  is  $R_i$ ’s left neighbor on the boundary, and  $P_r(R_i, R_j, t)$  is similarly defined. The functions  $P_B, P_l$ , and  $P_r$  work together to maximize some function of the curve:

$$\Gamma \left( \int_0^1 \Upsilon(\gamma(s)) ds \right) \tag{6}$$

Examples include *length* ( $\Gamma(u) = u - L_d$ , and  $\Upsilon(v) = v$ ), where  $L_d$  is a minimum required length, and *curvature* ( $\Gamma(u) = u, \Upsilon(v) = \kappa_d - \kappa(v)$ ).

Once a boundary has been determined,  $S$  is partitioned into two sets of robots. Interior robots move according to

$$\frac{dq_i(t)}{dt} = \mathcal{F}_I(q(t)) = \tilde{\mathcal{F}}_{I_S}(q_I(t)) + \tilde{\mathcal{F}}_{B_S}(q_B(t)) \tag{7}$$

where  $\tilde{\mathcal{F}}_{I_S}$  and  $\tilde{\mathcal{F}}_{B_S}$  are variants of  $\mathcal{F}_S$  and represent, respectively, interaction with interior and boundary robots.

On the other hand, boundary robots will have the capability of morphing the boundary curve into a target curve  $\gamma_D$ .  $\gamma_D$  is, in our method, constructed from a small set of shape descriptors  $\Xi$ :

$$\begin{aligned} \frac{dq_i(t)}{dt} &= \mathcal{F}_I(q(t)) \\ &= \mathcal{F}_N(\gamma(t), \gamma_D, \Theta(\gamma_D)) \vec{N}(s) + \mathcal{F}_T(\gamma(s)) \vec{T}(s) \end{aligned} \tag{8}$$

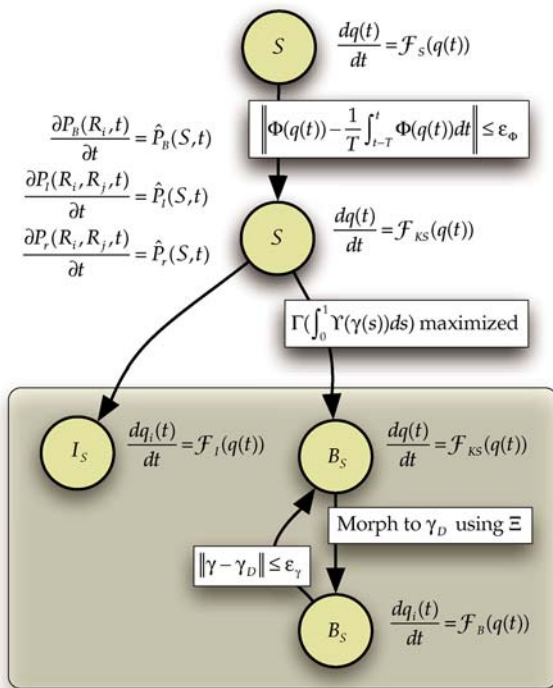


Fig. 3 Global view of the system

where  $\vec{N}(s)$  and  $\vec{T}(s)$  represent normal and tangent vectors at  $\gamma(s)$ , and  $\Theta(\gamma_D)$  is the attracting potential corresponding to  $\gamma_D$ . After convergence, we'll have  $\|\gamma - \gamma_D\| \leq \epsilon_\gamma$ .

Figure 3 shows the global state diagram of the system.

### 3. Shape representation

In this paper, we are interested in large groups of autonomous mobile vehicles which collectively form certain desired shapes. The shapes we deal with here normally correspond to environmental features such as those found in underwater landscapes, or almost stationary plumes developed through some slow diffusion process. The assembly can then act as a huge networked sensor sampling the desired feature, or simply complying with it's shape, while performing exploration tasks. In traditional formation control literature, a formation shape is usually represented by formation graphs or formation functions which encode the precise geometrical relationships between each and every robot. When the number of robots increases, the aggregate more resembles a swarm than a simple formation. On the other hand, swarm literature are interested in the inter-vehicle forces giving rise to the advection-diffusion process of swarming rather than the actual shape of the aggregate. In this paper, we combine aspects of both approaches. First, we review some of the previous attempts at finding appropriate representations.

The most reasonable way of describing the approximate shape of very large formations is to use a reduced-order

representation. This is a fairly common approach used in machine vision and pattern recognition. We review the basic concept here. Let  $S \subseteq \mathfrak{R}^M$  be a low-dimensional space of canonical invariant shape parameters with elements  $\vec{s} = \{s_0, \dots, s_{M-1}\}$  and  $G \subseteq \mathfrak{R}^3$  the Lie group of Euclidian transformations in the plane with elements  $\vec{g} = (\mu, \theta, l)$ . A certain formation  $F(t)$  can be considered as an element  $\vec{f} = (\vec{s}, \vec{g})$  of  $G \otimes S$ . Let  $Q = (q_0, \dots, q_{N-1}) \in \mathfrak{R}^N$  denote the state of the aggregate  $G$ , then  $\vec{f} = \Psi(Q)$  and  $Q = \Psi^{-1}(\vec{f})$ .  $\Psi^{-1}$  is generally not known. In such a case, we can design control laws for individual robots using a minimum norm solution of  $\Psi \dot{q} = \vec{f}$  in the least squares sense, where  $\Psi$  denotes the differential of  $\Psi$ . Thus,  $\dot{q} = \Psi^{-1} \vec{f}$ . At each time step, all the robots send their position information to a super-agent which calculates the error  $e = \|\vec{f}_d - \vec{f}\|$  and sends  $e$  back to the robots. The robots then move according to  $\dot{q} = \Psi^{-1} e$ . In Belta and Kumar (2002), this method was originally proposed and used by defining

$$\mu = \frac{1}{N} \sum_{i=0}^{N-1} q_i,$$

$$s_0 = \frac{1}{N-1} \sum_{i=0}^{N-1} x_i^2, \quad s_1 = \frac{1}{N-1} \sum_{i=0}^{N-1} y_i^2 \tag{9}$$

which are the first and second moments, and defining orientation using

$$\sum_{i=0}^{N-1} x_i y_i = 0 \tag{10}$$

to form simple ellipsoidal shapes.

For more complex shapes, higher-order moments or better shape parameters should be used. One such method is called *integral invariants*. In this method, shape parameters  $s_{mn}$  are defined by

$$s_{mn} = \frac{1}{2\pi N_x N_y} \int_X \int_Y \left( \sum_{q \in G} e^{-\sigma \|q - (x,y)\|^2} \right)^m \tag{11}$$

$$\left[ \int_0^{2\pi} \left( \sum_{q \in G} e^{-\sigma \|q - ((x,y) + R(\phi)re_x)\|^2} \right)^n d\phi \right] dx dy \tag{12}$$

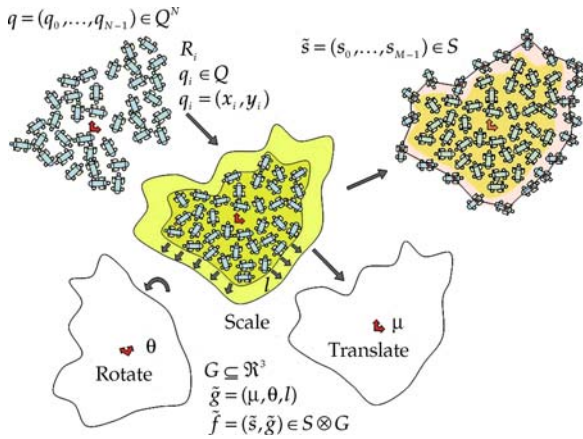
where

$$X = \left[ \min_{q_i \in G} (x_i) - d, \max_{q_i \in G} (x_i) + d \right],$$

$$Y = \left[ \min_{q_i \in G} (y_i) - d, \max_{q_i \in G} (y_i) + d \right], \tag{13}$$

$$N_x = \max_{q_i \in G} (x_i) - \min_{q_i \in G} (x_i) - 2d,$$

$$N_y = \max_{q_i \in G} (y_i) - \min_{q_i \in G} (y_i) - 2d.$$



**Fig. 4** Formation shapes and transformations

The parameters are calculated as  $s_{00}, s_{01}, s_{11}$ , etc. where  $d$  denotes a desired margin around the area covered by robots. This representation, although very powerful, is very costly and  $\Psi^{-1}$  or  $\tilde{\Psi}^{-1}$  can not be easily computed, if at all.

The complexities associated with using the above representation techniques for more complex shapes prompted us to consider a slightly different approach where the aggregate is partitioned into two non-overlapping sets: the interior and the boundary. In most applications, the interior is just required to maintain a uniform distribution inside the boundary. What was said about reduced dimensional representations need only be applied to the boundary set. Among the various proposed boundary representations, complex Fourier descriptors are the most popular. A handful of Fourier descriptors  $\mathcal{F} = \{\mathcal{F}_{-M}, \dots, \mathcal{F}_M\} \in \mathbb{R}^{2M}$ , with  $M$  usually very small, can represent fairly complex shapes and, furthermore, given  $\mathcal{F}$ , the original curve  $\gamma_{\mathcal{F}}(s) = \Psi^{-1}\{q_0, \dots, q_{N-1}\}$ , with desired granularity specified by  $N$ , can easily be recovered. The version we use here is invariant under homogeneous transformations, details of which will be presented in Section 6. Figure 4 is a visual demonstration of shapes and transformations.

#### 4. Uniform distribution

We use the recipe for swarming for two purposes. One is to make the interior robots form a uniform distribution inside the boundary and the other is to make the whole assembly form a more appropriate shape prior to the commencement of the boundary assignment process. The rationale behind this assumption is that, as will be seen shortly, the type of swarming behaviour we consider here is actually a discrete implementation of a diffusion process which relates the time derivative of the concentration to the second spatial derivative of concentration (local curvature) that will effectively act like a low pass filter on the concentration profile.

Suppose that the individual robots move according to the control law

$$\dot{q}_i(t) = \frac{dq_i(t)}{dt} = \lambda \mathcal{F}_i(q(t)), \tag{14}$$

where  $\mathcal{F}_i(q(t))$  is a force driving each robot and  $\lambda$  is a damping coefficient. Here, we have ignored inertial effects. The system  $S$  is thus governed by the coupled differential equations

$$\dot{q}(t) = \lambda \mathcal{F}(q(t)) \tag{15}$$

*Definition 1.* Let  $D_S > 0$  be a given number (comfortable distance between neighboring robots). Define a neighborhood  $\mathcal{N}_i$  around each robot with  $r(\mathcal{N}_i) = D_S$ . Let the robots be initially positioned at locations  $q_1(t_0), \dots, q_N(t_0)$  at time  $t_0$ . Let the robots evolve according to (2). Suppose that after a finite time, say  $T_c$ , all the robots reach their equilibrium (stop moving) and for every pair of robots  $R_i$  and  $R_j$ , such that  $R_i \in \mathcal{N}_j$  and  $R_j \in \mathcal{N}_i$ , we have  $\|q_i(t_c) - q_j(t_c)\| = D_S$ , where  $t_c = t_0 + T_c$ . Then, we say that the robots have formed a swarm. In absence of any external perturbations, the swarm will stay in equilibrium for all times  $t > t_c$ . We call  $\mathcal{F}$  a swarm generator. The set of states  $q(t_c)$  satisfying the above definition can be considered as the invariant set  $\Omega_{\mathcal{F}}$  for the system (3). A swarm generator  $\mathcal{F}$  is called invariant if it is invariant under affine transformations. In other words,  $q(t)$  is an equilibrium state of  $\mathcal{F}$ , then so are  $q(t) + \tilde{q}$  and  $R(\theta)q(t)$ , where  $\tilde{q}$  is a translation and  $R_z(\theta)$  represents the rotation matrix around the  $z$  axis applying a  $\theta$  degree rotation to all  $q_i$ .

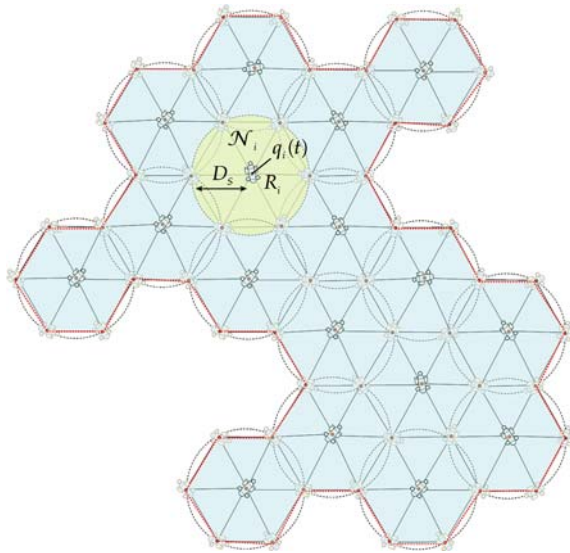
It is easy to see that the only local arrangement, satisfying (4), is a hexagon centred around each robot, as Fig. 5 depicts. In other words, if there exists a swarm generator  $\mathcal{F}$ , then after the swarm has been formed, the structure would locally look like the one in Fig. 5. Thus, every neighborhood consists of just seven robots, with one robot at the centre and the other six distributed around the perimeter of a circle with radius  $D_S$ .

For the class of swarm generators we are interested in,  $\Omega_{\mathcal{F}}$  is infinite, i.e., there are an infinite number of equilibrium states. This is always true for invariant generators.

In real environments, there are always some sort of external disturbance that tries to destabilize the swarm. In such a case, we require that a bound such as  $D_S - \epsilon \leq \|q_i(t_c) - q_j(t_c)\| \leq D_S + \epsilon$  be actually satisfied.

One way of designing a swarm generator  $\mathcal{F}$  is to first design a potential function  $\Psi(q, t)$  whose minimums correspond to the invariant set  $\Omega_{\mathcal{F}}$ . Then, we can define

$$\mathcal{F}(q(t)) = -\nabla_q \Omega(q, t) \tag{16}$$



**Fig. 5** Uniform distribution by swarming

The control laws for individual robots would thus be defined as

$$\mathcal{F}_i(q(t)) = -\nabla_{q_i} \Omega(q, t) \tag{17}$$

Although the forces  $\mathcal{F}(q(t))$  can be designed without having to define a potential first, nevertheless, to every such force there always corresponds a potential:

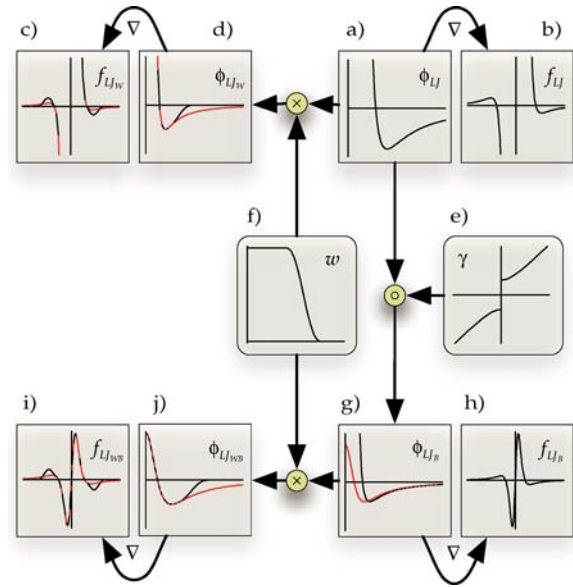
$$\Omega(q, t) = \int_{t_0}^t \mathcal{F}(q(t)) dt \tag{18}$$

It is worth noting that  $\Psi(q, t)$  can be used as a Lyapunov function for the aggregate.

There are quite a number of ways that a swarm generator can be designed. Here, we consider a special type of swarm generator constructed from a potential. Let

$$\phi_{LJ}(\tau) = \frac{-\mathcal{E}_D}{m-n} \left( m \left( \frac{\mathcal{D}_S}{\tau} \right)^n - n \left( \frac{\mathcal{D}_S}{\tau} \right)^m \right). \tag{19}$$

(depicted in Fig. 6(a)). Then, the potential energy between two robots  $R_i$  and  $R_j$  is defined by  $\phi_{LJ}^{ij} \equiv \phi_{LJ}(\|q_i(t) - q_j(t)\|)$  and is called the *Lennard-Jones bi-reciprocal potential*.  $\mathcal{E}_D$  is called the *dissociation energy* and  $\mathcal{D}_S$  is called the *equilibrium separation distance*. When the two robots  $R_i$  and  $R_j$  are in equilibrium,  $\phi_{LJ}(\mathcal{D}_{ij}) = -\mathcal{E}_D$ , its minimum value, and the distance between them would be equal to  $\mathcal{D}_S$ . Lennard-Jones potentials were initially used to model interaction forces between molecules. It has since been used for geometric modelling (Tonnesen, 1998) and other similar applications. Here, we use it to create uniform distributions of robots in a spatial extent.



**Fig. 6** Swarming potentials and associated forces

We have

$$\nabla_{q_i} \phi_{LJ}^{ij} = \frac{d}{d\tau} \phi_{LJ}(\|q_i(t) - q_j(t)\|) \frac{q_i(t) - q_j(t)}{\|q_i(t) - q_j(t)\|} \tag{20}$$

where

$$\frac{d}{d\tau} \phi_{LJ}(\tau) = \frac{\mathcal{E}_D m n}{m-n} \left( \frac{\mathcal{D}_S^n}{\tau^{n+1}} - \frac{\mathcal{D}_S^m}{\tau^{m+1}} \right) \tag{21}$$

Now, define

$$f_{\phi_{LJ}}^{ij} = f_{\phi_{LJ}}(q_i(t) - q_j(t)) = -\nabla_{q_i} \phi_{LJ}^{ij} \tag{22}$$

as the force between two robots  $R_i$  and  $R_j$  and define

$$f_{\phi_{LJ}}^i(q(t)) = \sum_{j=1}^N f_{\phi_{LJ}}(q_i(t) - q_j(t)) \tag{23}$$

as the total force experienced by  $R_i$ . If we set

$$\mathcal{F}_i(q(t)) = f_{\phi_{LJ}}^i(q(t)) \tag{24}$$

then the system defined by (2) will strive to minimize the *total potential of the aggregate* (also, the Lyapunov function) which is defined as

$$\phi_{LJ}^S(q(t)) \equiv \frac{1}{2} \sum_{i=1}^N \sum_{j=1, j \neq i}^N \phi_{LJ}(\|q_i(t) - q_j(t)\|) \tag{25}$$

The Lennard-Jones potential  $\phi_{LJ}$  goes to zero at infinity and at a point called the *collision distance*  $\mathcal{C}_D$ . If  $n = 2m$ , then  $\mathcal{C}_D = \mathcal{D}_S(2^{-1/m})$ .

Refer to Tonnesen (1998) for a detailed description of the effect of varying  $\mathcal{E}_D$ ,  $n$  and  $m$  on *compressibility* and *brittleness* of the swarm.

As is clear from previous section,  $\phi_{LJ}(\|q_i(t) - q_j(t)\|)$  and  $f_{LJ}(q_i - q_j)$  go to infinity as  $\|q_i - q_j\| \rightarrow 0$ . This causes problems when implementing on a real computer with limited computing capabilities. Moreover, when the robots get very close together, the Lennard-Jones potential applies very large forces on robots. In simulations, this effect is seen as an explosion of the swarm. Although the robots have physical constraints limiting their actual velocities, nevertheless, it is highly desirable if the actual computed forces somehow captured this limitation. Bounding the potential at a certain maximum  $\phi_M$  is tantamount to asserting that below a certain distance, the force (pressure) exerted on each robot does not change. We can realize this using a transformation on  $\tau$ , i.e., the distance between two robots. Suppose that  $\tau_M = \phi^{-1}(\phi_M)$ , such that  $\tau_M < \mathcal{C}_D$ . Now consider the mapping

$$\gamma(\tau) = \tau_M \sqrt{\beta \tau^2 + 1} \tag{26}$$

The graph of this mapping is shown in Fig. 6(e) with  $\beta = (1/\tau_M)^2$  (which corresponds to a slope of 1, because the slope equals  $\alpha\sqrt{\beta}$ ). As can be seen in the figure, the mapping is almost linear and one-to-one when  $\tau$  is greater than some value, and decays to  $\tau_M$  as  $\tau \rightarrow 0$ .

The bounded potential is now defined by

$$\phi_{LJ_B}(\tau) = \phi_{LJ} \circ \gamma(\tau) \tag{27}$$

with corresponding force calculated as

$$f_{LJ_B}(\zeta) = -\frac{d}{d\gamma} \phi_{LJ}(\gamma(\|\zeta\|)) \frac{d}{d\tau} \gamma(\|\zeta\|) \frac{\zeta}{\|\zeta\|} \tag{28}$$

Here,  $\zeta$  is the vector representing the difference of two positions on the plane, e.g.,  $q_i(t) - q_j(t)$ , and let  $\tau = \|\zeta\|$ . Note that  $\partial\tau/\partial\zeta = \zeta/\|\zeta\|$ . The critical points of  $\phi_{LJ_B}$  occur at points  $\tau$  where  $\phi'_{LJ_B}(\tau) = \phi'_{LJ}(\gamma(\tau))\gamma'(\tau) = 0$ . The one corresponding to  $\gamma'(\tau) = 0$  is zero (the point at which the potential attains its maximum) and the one corresponding to  $\phi'_{LJ}(\gamma(\tau)) = 0$  is equal to

$$\mathcal{D}_{S_B} = \sqrt{\left(\left(\frac{\mathcal{D}_S}{\tau_M}\right)^2 - 1\right) / \beta} \tag{29}$$

which is also the point at which the force is zero. This means that with bounded potentials, the comfortable distance would be  $\mathcal{D}_{S_B}$ . If we plug this value into the expression for  $\phi_{LJ_B}$ , we get  $-\mathcal{E}_D$ , so that the minimum value of the original potential is not changed under the action of  $\gamma(\mathcal{E}_{D_B} = \mathcal{E}_D)$  as long as

$\tau_M < \mathcal{C}_D$ . For a desired comfortable distance  $\tilde{\mathcal{D}}_S$ , one should put  $\mathcal{D}_S = \tau_M \sqrt{\beta \tilde{\mathcal{D}}_S^2 + 1}$ .

To prevent the discontinuity at neighborhood boundary, weighting has been proposed in Tonnesen (1998) to make the potential smoothly die off at the boundaries. Again, refer to Tonnesen (1998) for details. Now, the potential is defined as

$$\phi_{LJ_w}(\tau) = w(\tau)\phi_{LJ}(\tau) \tag{30}$$

(Fig. 6(c)). The way  $w(\tau)$  (Fig. 6(c)) is defined ensures that for every robot  $R_i$ ,

$$\sum_{j=1, j \neq i}^N w^{ij} \phi_{LJ}^{ij} = \sum_{j \in \mathcal{N}_i} w^{ij} \phi_{LJ}^{ij} \tag{31}$$

where  $w^{ij} = w(\|q_i(t) - q_j(t)\|)$ . The interaction force applied to a robot  $R_i$  from another robot  $R_j$  would thus be

$$\begin{aligned} f_{LJ_w}(q_i(t) - q_j(t)) &= -\nabla_{q_i}(w(\tau)\phi_{LJ}(\tau_{ij})) \\ &= \left( w(\tau_{ij}) \frac{d}{d\tau} \phi_{LJ}(\tau_{ij}) + \phi_{LJ}(\tau_{ij}) \frac{d}{d\tau} w(\tau_{ij}) \right) \frac{q_i(t) - q_j(t)}{\tau_{ij}} \end{aligned} \tag{32}$$

where  $\tau_{ij} = \|q_i(t) - q_j(t)\|$  and is depicted in Fig. 6(d).

In this case, we have

$$\phi_{LJ_{BW}}(\tau) = w(\tau)\phi_{LJ_B}(\tau), \tag{33}$$

$$f_{LJ_{BW}}(q_i(t) - q_j(t)) = -\nabla_{q_i}(w(\tau)\phi_{LJ_B}(\tau_{ij})). \tag{34}$$

The latter is shown in Fig. 6(i), while the former can be seen in Fig. 6(j). Figure 6 shows the various potentials and the associated forces. Finally, the bounded forces, although within the limitations of the processor, are still too big to be realized by real robots. To dampen out the spikes that occur when robots get close together, we use the *shunting model*. In this model the actual force applied to a robot,  $\tilde{f}_{LJ_{BW}}^i(t)$ , is the solution to the differential equation

$$\begin{aligned} \frac{\partial \tilde{f}_{LJ_{BW}}^i(t)}{\partial t} &= -D \tilde{f}_{LJ_{BW}}^i(t) + (B_U - \tilde{f}_{LJ_{BW}}^i(t)) \max(0, \tilde{f}_{LJ_{BW}}^i(t)) \\ &\quad - (B_L + \tilde{f}_{LJ_{BW}}^i(t)) \max(0, -\tilde{f}_{LJ_{BW}}^i(t)) \end{aligned} \tag{35}$$

where  $D$  is the decay rate and  $B_U$  and  $B_L$  are the upper and lower bounds. We set  $D = \mathcal{E}_D/10$ . In the discrete

implementation,  $\Delta t = 1/D$  should be used. Shunting model was introduced by Grossberg and was originally used in neural networks but can generally be used to limit an arbitrary signal to lower and upper bounds. It also exhibits some memory effects in that big spikes are not just cut out but retain their action over an extended period of time. This will produce smooth transitions.

### 5. Boundary assignment

Assigning boundary robots is indeed the most crucial phase in the proposed system. The whole aggregate should self-organize itself into two non-overlapping sets. It is a difficult problem even in the presence of global information. In our case, all the decisions have to be made with respect to local neighborhoods, which makes the problem even harder. In its most general form, the solution can be formulated as the solution of a set of coupled selection equations with dynamic variables  $\xi_i \in [0, 1]$  ( $\xi_i = 1$  means  $R_i \in B_S$ ),  $\zeta_{ij}^i \in [0, 1]$  ( $\zeta_{ij}^i = 1$  means  $R_i \in B_S, R_j \in B_S$ , and  $R_j$  is  $R_i$ 's left neighbor on the boundary),  $\zeta_{ij}^r \in [0, 1]$  (defined similarly to  $\zeta_{ij}^l$ ), subject to constraints imposed on the desired quality of the boundary (length, curvature, ...). The resulting  $B_S$  should of course be a boundary for  $S : R_i \in B_S$  should form a closed polygon without intersections and all  $R \in I_S (= S \setminus B_S)$  should be inside this polygon. The system can be written

$$\varrho = \varsigma \left( \xi, \zeta^l, \zeta^r, \left\{ \Gamma_\alpha \left( \int_0^l \Upsilon_\alpha(\gamma(s)) ds \right) \right\} \right) \quad (36)$$

where  $\varrho = [\xi, \zeta^l, \zeta^r]^T$ . The assignment should emerge in the limit as an asymptotically stable point of the above dynamics (refer to Starke (1987) for a similar problem).

In this section, rather than attempting to solve the general problem, we present a procedure which can be considered as a rough discrete approximation of the continuous dynamics. It is always successful if certain conditions are met.

As before,  $\mathcal{N}_r(R_i)$  denotes the neighborhood of  $R_i$  with radius  $r$ . Define

$$\begin{aligned} \rho &: \mathcal{N}_r(R_i) \rightarrow [0, 2\pi], \\ R_\alpha &\rightarrow s_{2\pi}(\text{atan}((y_{R_\alpha} - y_{R_1}) / (x_{R_\alpha} - x_{R_1}))), \\ s_{2\pi}(\theta) &= \theta - 2\pi \lfloor \theta / (2\pi) \rfloor \end{aligned} \quad (37)$$

Define a partial order  $\leq$  on  $\mathcal{N}_r(R_i)$  such that  $R_\alpha \leq R_\beta \Leftrightarrow \rho(R_\alpha) \leq \rho(R_\beta)$ . Now, re-label the robots in  $\mathcal{N}_r(R_i)$  as  $\{R_0, \dots, R_{|\mathcal{N}_r(R_i)|-1}\}$  according to the order. Also define the mappings

$$\iota_+ : \mathcal{N}_r(R_i) \rightarrow \mathcal{N}_r(R_i), R_\alpha \rightarrow R_{(\alpha+1) \bmod |\mathcal{N}_r(R_i)|} \quad (38)$$

$$\iota_- : \mathcal{N}_r(R_i) \rightarrow \mathcal{N}_r(R_i), R_\alpha \rightarrow R_{(\alpha-1) \bmod |\mathcal{N}_r(R_i)|} \quad (39)$$

Finally, consider the maps  $l, r : B_S \rightarrow B_S$ , which define the left and right neighbors of  $R_i \in B_S$ . The boundary assignment scheme should determine  $B_S, l(B_S)$  and  $r(B_S)$ .

We now describe 5 heuristic rules which will identify  $B_S$ . They should be active during the process. First, fix some  $\theta_G$ .  $\theta_G$  can be used to control the length and curvature of the resulting curve. Furthermore, let  $B_S = \emptyset$  initially.

**Rule 1.** Suppose there exists a  $R_\alpha \in \mathcal{N}_r(R_i)$  such that for every  $R_\beta \in \mathcal{N}_r(R_i), R_\beta \neq R_\alpha$ , we have  $\mathcal{G}(R_\alpha) \geq \mathcal{G}(R_\beta)$  and  $\mathcal{G}(R_\alpha) \geq \theta_G$ , where

$$\mathcal{G}(R_\alpha) = \text{acos} \frac{(R_\alpha - R_i)(\iota_+(R_\alpha) - R_i)}{\|R_\alpha - R_i\| \|(\iota_+(R_\alpha) - R_i)\|} \quad (40)$$

Then, let  $B_S \leftarrow B_S \cup \{R_i\}$  and define  $\iota(R_i) = \iota_+(R_\alpha), r(R_i) = R_\alpha$ .

**Rule 2.** Suppose  $R_\alpha \in \mathcal{N}_r(R_i)$  such that  $R_\alpha \geq l(R_i), r(R_\alpha) = R_i$ , and  $\|q_\alpha(t) - q_i(t)\| \leq \|q_{l(R_i)}(t) - q_i(t)\|$ . Then define  $l(R_i) = R_\alpha$ .

**Rule 3.** Suppose  $R_\beta \in \mathcal{N}_r(R_i)$  such that  $R_\alpha \leq r(R_i), l(R_\alpha) = R_i$ , and  $\|q_\beta(t) - q_i(t)\| \leq \|q_{r(R_i)}(t) - q_i(t)\|$ . Then define  $r(R_i) = R_\beta$ .

**Rule 4.** Suppose  $R_j \in \mathcal{N}_r(R_i)$  such that

$$\chi(q_{r(R_i)}(t), q_i(t), q_{l(R_j)}(t), q_j(t)) > 0, \quad (41)$$

where

$$\begin{aligned} \chi(q_q, \tilde{q}_1, q_2, \tilde{q}_2) &= (l_1 - \|\mathcal{U}_1\|)(l_2 - \|\mathcal{U}_2\|), \\ l_1 &= \frac{1}{2} \|q_1 - \tilde{q}_1\|, \mathcal{U}_1 = Q - m_1, m_1 = \frac{1}{2}(q_1 + \tilde{q}_1) \\ l_2 &= \frac{1}{2} \|q_2 - \tilde{q}_2\|, \mathcal{U}_2 = Q - m_2, m_2 = \frac{1}{2}(q_2 + \tilde{q}_2) \\ x_Q &= (1/D) \left( (x_{q_2} - x_{\tilde{q}_2})(x_{q_1}, y_{q_1} - y_{q_1} x_{q_1}) - (x_{q_1 - x_{\tilde{q}_1}}) \right. \\ &\quad \left. (x_{q_2} y_{q_2} - y_{q_2} x_{q_2}) \right) \\ y_Q &= (1/D) \left( (y_{q_2} - y_{\tilde{q}_2})(x_{q_1} y_{q_1} - y_{q_1} x_{q_1}) - (y_{q_1} - y_{\tilde{q}_1}) \right. \\ &\quad \left. (x_{q_1} y_{q_2} - y_{q_2} x_{q_2}) \right) \\ \mathcal{D} &= (x_{q_1} - x_{\tilde{q}_1})(y_{q_2} - y_{\tilde{q}_2}) - (y_{q_1} - y_{\tilde{q}_1})(x_{q_2} - x_{\tilde{q}_2}) \end{aligned} \quad (42)$$

Then, let  $r(R_i) = R_j$ .

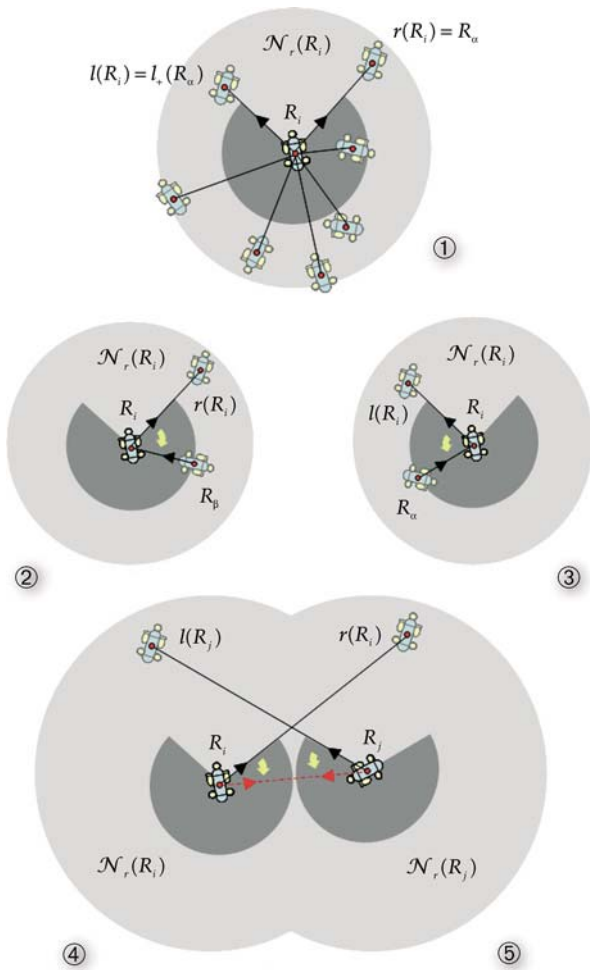
**Rule 5.** Suppose  $R_i \in \mathcal{N}_r(R_j)$  such that

$$\chi(q_{l(R_j)}(t), q_j(t), q_{r(R_i)}(t), q_i(t)) > 0. \quad (43)$$

Then, set  $l(R_j) = R_i$ .

Figure 7 shows the rules. To make the formulas in the following sections more readable, instead of using  $l(R_i)$  and  $r(R_i)$  for referring to a robot's left and right neighbors, we imagine that the boundary robots are re-labelled in such a way that





**Fig. 7** Boundary-assignment rules

their labels parametrize the boundary. The  $R_i \in B_S$  with the least  $i$  (in fact, any robot can be picked for this purpose) can be chosen as the starting point of the curve ( $s = 0$ ). The robot immediately to its left is labelled  $s = 1$  and so on. The robot to the right of the starting robot would thus be labelled  $s = |B_S| - 1$ .

**6. Target shape and potential**

Curve evolution, as discussed in Section 8, requires that a potential around the desired shape be defined. In our approach, a shape is simply represented by its boundary. Among the many approximate methods devised to describe shapes, Fourier descriptors can very compactly encode shapes and provide us with a simple reconstruction recipe. Any closed curve, parametrized by  $s$  (the arc length) and with perimeter  $T$  can be described by its Fourier expansion

$$q(s) = \sum_{n=-\frac{N}{2}}^{\frac{N}{2}} c_n e^{jn\omega s}, \omega = \frac{n\pi}{T} \tag{44}$$

where the Fourier coefficients are given by

$$c_n = \frac{1}{T} \int_0^T q(s) e^{-jns} ds. \tag{45}$$

Here, we assume that the perimeter is normalized by  $T = 2\pi$ , i.e.  $\omega = 1$ . For a shape described by a set of vertices  $q_i, i = 0, \dots, N - 1$ , the Fourier descriptors  $c_n, n = -N/2, \dots, N/2$  are the coefficients of the Fourier transform of  $q_i$ :

$$q_i = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}} c_k e^{2\pi j \frac{ki}{N}}, c_k = \frac{1}{N} \sum_{i=0}^N q_i e^{-2\pi j \frac{ik}{N}} \tag{46}$$

Refer to Kalantar and Zimmer (2004) for the calculation of the set of Fourier descriptors  $\{\mathcal{F}_n = (\mathcal{F}_{n_x}, \mathcal{F}_{n_y})\}$ , invariant with respect to translation, rotation, scaling and choice of starting point, given the set  $\{q_0, \dots, q_{N-1}\}$  of vertices. The periodic boundary condition is defined by  $q_{N-1+i} = q_{i-1}$  and  $q_{-i} = q_{N-i}$ , where  $i = 1, \dots, N$ .

Given a set of invariant features  $\mathcal{F}_{-M}, \dots, \mathcal{F}_M$  (comprising the canonical representation of  $\gamma_D$ ) and the set  $\{\mu_d^w, \theta_d, s_d\}$  of geometrical properties of the shape, where  $\mu_d^w, \theta_d, s_d$  represent desired areal centre of gravity, orientation (with respect to the original shape), and scaling (expansion/shrinking), the formula

$$v_i = s_d R(\theta_d) v_i^{\mathcal{F}_d} + \mu_d^w \tag{47}$$

gives the positions of the vertices  $\{v_0, \dots, v_{N-1}\}$  of the polygon approximating the target shape, where  $R(\vartheta)$  denotes the rotation matrix around the  $z$  axis by  $\vartheta$  degrees, and

$$v_i^{\mathcal{F}_d} = \sum_{n=-M, n \neq 0}^M \mathcal{F}_n e^{\frac{2jn\pi i}{N}} \tag{48}$$

A potential around  $\gamma_D$  can be constructed using a distance function  $d(\gamma_D, q(t))$  giving the closest point on  $\gamma_D$  to a point  $q$  inside or outside the boundary. In Wang et al. (2002) a simple method is proposed.  $\gamma_D(s)$  is first approximated with a cubic spline function  $\tilde{\gamma}_D(s)$  with equally spaced breakpoints  $\{s_0, \dots, s_{N-1}\}$ , with  $s_0 = 0$  and  $s_{N-1} = 1$ . The distance between  $q(t)$  and position  $\tilde{\gamma}_D(s)$  is  $D(s) = \|\tilde{\gamma}_D(s) - q(t)\|$ . The value  $s^*$  minimizing  $D(s)$  is the closest point to  $q(t)$ ,  $\overline{D(s^*)q(t)}$  is perpendicular to the curve and its distance is the shortest distance between  $q(t)$  and the curve. For each segment  $[s_i, s_{i+1}]$ , let  $P(s)$  be the quadratic polynomial interpolating  $\tilde{\gamma}_D(s)$  at  $s_i, (s_i + s_{i+1})/2$ , and  $s_{i+1}$ . At each iteration, the value among  $\{s_i, (s_i + s_{i+1})/2, s_{i+1}, s^*\}$  which gives the largest  $P$  is eliminated and the process continued

until either  $P(s^*) \leq \varepsilon_P$  or  $s^*$  goes out of the segment. In case of convergence, Newton’s formula

$$s^* = s^* - D'(s^*)/D''(s^*) \tag{49}$$

is used to get a better solution. Otherwise, another segment is tried.

Here, we use a much simpler strategy based on the polygonal approximation of  $\gamma_D$ . For each segment  $S_i = \{v_i, v_{i+1}\}$  and point  $q(t)$ , let

$$\tilde{Q}(q(t), S_i) = v_i + \tilde{u}(v_{i+1} - v_i) \tag{50}$$

$$\tilde{u} = \frac{\langle (q(t) - v_i), (v_{i+1} - v_i) \rangle}{\|v_{i+1} - v_i\|^2} \tag{51}$$

$$Q(q(t), S_i) = \begin{cases} \tilde{Q}(q(t), S_i); & \|M_{S_i} - \tilde{Q}q, S_i\| \leq \frac{v_{i+1} - v_i}{2} \\ \iota(\tilde{Q}(q(t), S_i), v_{i+1}, v_i); & \text{otherwise} \end{cases} \tag{52}$$

$$\iota(\tilde{Q}(q(t), S_i), v_{i+1}, v_i) = \begin{cases} v_{i-1}; & \|\tilde{Q}q, S_i - v_i\| > \|\tilde{Q}q, S_i - v_{i+1}\| \\ v_i; & \text{otherwise} \end{cases} \tag{53}$$

where  $M_{S_i} = (v_{i+1} + v_i)/2$ . Moreover, let

$$\phi_{v_{i+1}-v_i} = \text{atan}(y_{v_{i+1}-v_i}/x_{v_{i+1}-v_i}) \tag{54}$$

$$(q - v_i)^* = R_z(-\phi_{v_{i+1}-v_i})(q - v_i) \tag{55}$$

$$\phi_{(q-v_i)^*} = \text{atan}(y_{(q-v_i)^*}/x_{(q-v_i)^*}) \tag{56}$$

Now, the distance function is defined as

$$d(\gamma_D, q(t)) = \min_{S_i} \|q(t) - Q(q(t), S_i)\|, \tag{57}$$

where the minimum is taken over all  $S_i$  subject to the constraints

$$\phi_{(q-v_i)^*} \neq \pi \quad \text{and} \quad \phi_{(q-v_i)^*} \neq 0. \tag{58}$$

The in-out function is 1 if  $q(t)$  is outside the shape and is  $-1$  if it is inside and is defined by:

$$f_{IO}(q(t), \gamma_D) = \begin{cases} 1; & 0 \leq \phi_{(q-v_i)^*} < \pi \\ -1; & \pi \leq \phi_{(q-v_i)^*} < 2\pi \end{cases} \tag{59}$$

$Q(q(t), S_i)$  is the closest point on  $\gamma_D$  to  $q(t)$  and we denote it by  $Q_{\gamma_D}(q(t))$ .

The distance function provides a scalar field over  $\mathfrak{R}^2$ . To create a vector field, let’s define

$$g(q(t)) = \xi(1 - G_\sigma(d(\gamma_D, q(t)))) \tag{60}$$

where  $G_\sigma(\cdot) \sim \mathcal{N}(0, \sigma)$ . The gradient is computed as

$$\begin{aligned} \nabla_{q(t)}g(q(t)) &= -\xi \dot{G}_\sigma(d(\gamma_D, q(t))) \frac{N_{\gamma_D}(q(t))}{\|N_{\gamma_D}(q(t))\|} \\ &= -\xi \frac{1}{\sigma^2} G_\sigma(d(\gamma_D, q(t))) N_{\gamma_D}(q(t)) \end{aligned} \tag{61}$$

where

$$\begin{aligned} N_{\gamma_D}(q(t)) &= Q_{\gamma_D}(q(t)) - q(t), \\ \|N_{\gamma_D}(q(t))\| &= d(\gamma_D, q(t)). \end{aligned} \tag{62}$$

$\nabla g(\cdot)$  defines a force field as required by curve evolution schemes.

### 7. Boundary morphing

The shape of the aggregate is controlled by the motion of the boundary robots; the interior robots passively comply. This is equivalent to continuous deformation of the initial closed simple curve  $\gamma_0 = \gamma(0)$  into another desired curve  $\gamma_\infty$ , ensuring collision-free paths for the robots and no self-intersections. There exists several possible solutions. In Kalantar and Zimmer (2004), an ad hoc strategy is proposed in which the initial curve, represented by a set of Fourier descriptors, successively evolves into intermediate curves constructed with decreasing subsets of the descriptor set, to form an ellipsoidal shape in the limit, and then evolving into curves given by increasing subsets of the descriptor set of the target curve. This method (shape interpolation) involves unnecessary intermediate curves, several re-parametrizations, and substantial synchronization.

A more formal method relies on general curve evolution theory (Kichenassamy et al., 1995; Chenyang et al., 2000; Belyaev et al., 1999). Let  $\mathcal{U} : \gamma \rightarrow \mathcal{U}(\gamma) \in \mathfrak{R}$  denote a cost functional along  $\gamma$ . A curve evolution equation gives a one-parameter family of curves  $\{\gamma(t)\}$ ,  $t \in \mathfrak{R}^+$ , such that  $\gamma_t|_{t=0} = \gamma_0$  and  $\gamma_\infty = \lim_{t \rightarrow \infty} \gamma_t$  is a local minimum of  $\mathcal{U}$ . The evolution equation takes the form of a partial differential equation  $\partial\gamma(t)/\partial t = \Omega(\gamma(t), \nabla\mathcal{U}(\gamma(t)))$  and follows the gradient descent of  $\mathcal{U}$ . These curves are usually called *active contours*, especially in machine vision literature.

There are two different approaches to representation and implementation of active contours. *Parametric* ac-

tive contours are explicitly represented as curves  $\gamma(s, t) = [x(s, t), y(s, t)]$  parametrized by the arclength  $s \in [0, 1]$ . The curve evolves according to the *Lagrangian* dynamic reaction-diffusion equation  $\partial\gamma(s, t)/\partial t = \Omega(\gamma(s, t), \nabla\mathcal{U}(\gamma(s, t)))$ . The force  $\Omega$  can be decomposed into its normal and tangential components. Thus, the equation would be

$$\frac{\partial\gamma(s, t)}{\partial t} = \Omega_N(s, t) \vec{N}_\gamma(s, t) + \Omega_T(s, t) \vec{T}_\gamma(s, t) \quad (63)$$

where  $\vec{N}_\gamma(s, t)$  and  $\vec{T}_\gamma(s, t)$  denote, respectively, the inward normal and the tangent to the curve at  $s$ . The tangential component only changes the parametrization, while the normal component modifies the shape of the curve and is usually called the *speed function*. A classic example is the deformable *active contour (snake)*:

$$\frac{\partial\gamma(s, t)}{\partial t} = (I(s, t) - \nabla P(\gamma(s, t)) \cdot \vec{N}_\gamma(s, t)) \vec{N}_\gamma(s, t) \quad (64)$$

$$I(s, t) = \frac{\partial}{\partial s} \left( \alpha(s) \frac{\partial\gamma(s, t)}{\partial s} \right) - \frac{\partial^2}{\partial s^2} \left( \beta(s) \frac{\partial^2\gamma(s, t)}{\partial s^2} \right) \quad (65)$$

where  $\alpha(s)$  and  $\beta(s)$  control, respectively, the *elasticity* and *rigidity* of the contour.  $\nabla P$  is a force field directing  $\gamma$  to the final desired curve  $\gamma_D$ . Also,

$$\mathcal{U}(\gamma) = \frac{1}{2} \int_0^1 \left( \alpha(s) \left\| \frac{\partial\gamma(s)}{\partial s} \right\|^2 + \beta(s) \left\| \frac{\partial^2\gamma(s)}{\partial s^2} \right\|^2 \right) ds - \int_0^1 P(\gamma(s)) ds \quad (66)$$

Geometric active contours, on the other hand, are implicitly defined as the zero level set of a scalar function  $\psi(x, t), x \in \mathbb{R}^2$ . The curves evolve according to the *Eulerian* (heat) equation

$$\frac{\partial\psi(x, t)}{\partial t} = g(x)(\kappa + v_0)\|\nabla\psi\| + \nabla g(x) \cdot \nabla\psi, \quad (67)$$

where  $g(x)$  is the same function as in Eq. (60) in Section 6,  $\kappa$  is the curvature and  $v_0$  is a constant. For our application, the Lagrangian formulation is more suitable because it lends itself to distributed implementation. The evolution equation we use here is the parametrized equivalent of the general Eq. (67),

$$\begin{aligned} \frac{\partial\gamma(s, t)}{\partial t} &= (\alpha + \beta\kappa(s, t)) \cdot g(\gamma(s, t)) \vec{N}_\gamma(s, t) \\ &\quad - \langle \nabla g(s, t), \vec{N}_\gamma(s, t) \rangle \vec{N}_\gamma(s, t) \\ &\quad + \omega(\gamma(s, t)) \vec{T}_\gamma(s, t) \end{aligned} \quad (68)$$

where

$$\alpha = f_{IO}(\gamma(s, t), \gamma_D), \quad (69)$$

$$g(\gamma(s, t)) = G_\sigma(d(\gamma_D(t), \gamma(s, t))), \quad (70)$$

$$\vec{N}_\gamma(s, t) = \frac{y_s(s, t) - x_s(s, t)}{\sqrt{x_s(s, t)^2 + y_s(s, t)^2}}, \quad (71)$$

$$\vec{T}_\gamma(s, t) = \frac{\partial\gamma(s, t)/\partial s}{\|\partial\gamma(s, t)/\partial s\|} = \vec{N}_\gamma(s, t)^\perp, \quad (72)$$

$$\kappa(s, t) = \frac{y_{ss}(s, t) \cdot x_s(s, t) - x_{ss}(s, t) \cdot y_s(s, t)}{\sqrt{x_s(s, t)^2 + y_s(s, t)^2}} \quad (73)$$

$g$  is zero at the desired boundary and stops the propagating front, the term involving  $\kappa$  smooths the curve such that those areas of the boundary with higher curvatures move faster.  $\omega(\gamma(s, t))$  does not have any effect other than re-parametrizing but is crucial for numerical implementation, as will be discussed next. Figure 8 shows the evolution of the boundary under the influence of the force field.

Boundary robots use a discretized version of Equ. (68) to move. We use the following rule:

$$\begin{aligned} q_i[n] &= q_i[n - 1] + \Delta t(f_{IO}(\gamma_i[n]) + \beta\kappa_i[n]) \cdot g(\gamma_i[n]) \cdot \vec{N}_{\gamma_i}[n] \\ &\quad - (\nabla_{q_i[n]}g(q_i[n]) \cdot \vec{N}_{\gamma_i}[n]) \cdot \vec{N}_{\gamma_i}[n] + \omega_i[n] \cdot \vec{T}_{\gamma_i}[n] \end{aligned} \quad (74)$$

We use the following central difference approximations:

$$\begin{aligned} dx_i[n]/ds &\approx \Delta x/2\Delta s, d^2x_i[n]/ds^2 \approx \Delta x^2/\Delta s^2, \\ dy_i[n]/ds &\approx \Delta y/2\Delta s, d^2y_i[n]/ds^2 \approx \Delta y^2/\Delta s^2, \end{aligned} \quad (75)$$

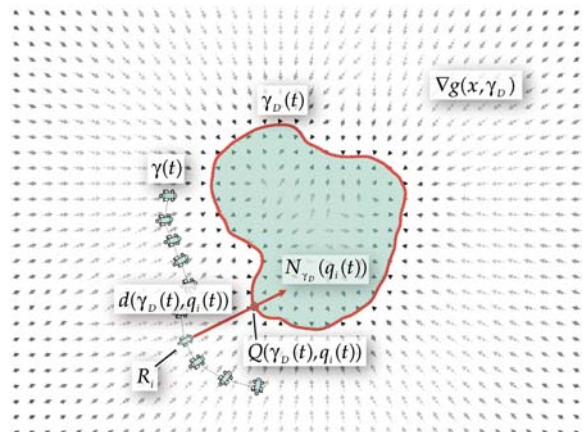


Fig. 8 Motion of curves under shape potentials

where

$$\begin{aligned} \Delta x &= x_{i+1}[n] - x_{i-1}[n], \\ \Delta y &= y_{i+1}[n] - y_{i-1}[n], \\ \Delta_{xx} &= x_{i-1}[n] - 2x_i[n] + x_{i+1}[n], \\ \Delta_{yy} &= y_{i-1}[n] - 2y_i[n] + y_{i+1}[n], \end{aligned} \tag{76}$$

$$\text{and } \Delta s = \int_0^1 \frac{\gamma(s)ds}{\|B_S\|}$$

Thus

$$\vec{N}_\gamma[n] = \frac{\Delta y - \Delta x}{\sqrt{\Delta x^2 + \Delta y^2}}, \tag{77}$$

$$\kappa_i[n] = 4 \frac{\Delta_{yy} \cdot \Delta x - \Delta_{xx} \cdot \Delta y}{(\Delta x^2 + \Delta y^2)^{3/2}} \tag{78}$$

In the above formulas  $\Delta s$  does not appear so that as robots get closer to each other, the calculations will get more unstable due to accumulation of errors. That’s the reason why a non-zero tangential force is used. This force should uniformly distribute the robots along the boundary. Rather than calculating the perimeter of the boundary and dividing it by the number of robots to determine the spacing between robots, we use a simple local force acting on each robot that achieves the same thing upon convergence:

$$\omega_i(t) = \zeta \left( e^{\frac{4(q_i(t)-q_{i-1}(t))^2}{\zeta(t)^2}} - e^{\frac{4(q_{i+1}(t)-q_i(t))^2}{\zeta(t)^2}} \right) \tag{79}$$

where  $\zeta(t) = \|q_{i+1} - q_i\| + \|q_i - q_{i-1}\|$ .

It should be noted that instead of gradients constructed out of descriptor sets for desired geometric shapes, we can use concentration gradients of a diffusion process and stop at an isocline. This way, the geometric shape is not predefined and is the shape of the isocline (Bertozzi et al., 2005).

### 8. Interior motion

The interior robots should at all times distribute themselves uniformly inside the boundary while staying far enough away from the boundary to prevent possible leakage out of it. Each interior robot has to handle two different interactions: one with other interior robots and the other with boundary robots. Interaction with fellow interior robots is the same as motion under the influence of Lennard–Jones forces described in Section 4. If there are no boundary robots in the neighborhood of an interior robot, the net force on it is simply calculated using Eq. (23). If the boundary robots are treated the same way as interior robots, interior robots can in many situation easily escape the boundary by squeezing through boundary robots. This can happen when two boundary robots

are further than some certain distance apart and the aggregated force from the interior on a particular interior robot in the vicinity of the boundary exceeds the one imposed on it by the boundary robots. This means that the forces from the boundary robots and from the interior robots should be balanced. One might be tempted to prevent the interior robots from moving in the direction of the boundary by using the formula

$$\left\langle \mathcal{F}_I, \left( \frac{\mathcal{F}_B}{\|\mathcal{F}_B\|} \right)^\perp \right\rangle + \mathcal{F}_B \tag{80}$$

when the angle between  $\mathcal{F}_B$  and  $\mathcal{F}_I$  is greater than  $\pi/2$ . This will cause the interior robots in the neighborhood of the boundary to keep moving along the boundary. Another solution is to balance the forces using a weighted summation  $w_B \mathcal{F}_B + w_I \mathcal{F}_I$  but then the optimal weights have to be determined.

The strategy we will use here is to treat the boundary as being composed of segments rather than individual robots and calculate the force from the boundary as a function of the distance to these segments. For each interior robot  $R_i$ , consider the set

$$\begin{aligned} \Omega_i &= \{\{q_\alpha, q_\beta\} | R_\alpha, R_\beta \in B_S, \\ &(R_\alpha = l(R_\beta), R_\beta = r(R_\alpha)), \\ &(R_\alpha \in N_i \cap B_S) \vee (R_\beta \in N_i \cap B_S)\} \end{aligned} \tag{81}$$

Now, define the motion of  $R_i$  as

$$\begin{aligned} \dot{q}_i(t) &= \mathcal{F}_I^i(q(t)) \\ &= \sum_{R_\eta \in I_S} f_{LJ_{BW}}(q_i(t) - q_\eta(t)) \\ &\quad + \sum_{\{q_\alpha, q_\beta\} \in \Omega_i} \frac{\partial \phi_{LJ_{BW}}}{\partial t}(d(q_i(t), \{q_\alpha, q_\beta\})) \\ &\quad \times \frac{Q(q_i(t), \{q_\alpha, q_\beta\}) - q_i(t)}{\|Q(q_i(t), \{q_\alpha, q_\beta\}) - q_i(t)\|} \end{aligned} \tag{82}$$

where

$$\tilde{Q}(q_i(t), \{q_\alpha, q_\beta\}) = q_\alpha + \tilde{u}(q_\beta - q_\alpha), \tag{83}$$

$$\tilde{u} = \frac{\langle (q_i(t) - a_\alpha), (q_\beta - q_\alpha) \rangle}{\|q_\beta - q_\alpha\|^2}, \tag{84}$$

$$\begin{aligned} &Q(q(t), \{q_\alpha, q_\beta\}) \\ &= \begin{cases} \tilde{Q}(q(t), \{q_\alpha, q_\beta\}); & \varrho \leq 0 \\ \iota(\tilde{Q}(q(t), \{q_\alpha, q_\beta\}), q_{\beta, q_\alpha}); & \text{otherwise} \end{cases} \end{aligned} \tag{85}$$

$$\varrho = \|M_{\{q_\alpha, q_\beta\}} - \tilde{Q}(q, \{q_\alpha, q_\beta\})\| - \|q_\beta - q_\alpha\|/2 \quad (86)$$

$$\begin{aligned} & \iota(\tilde{Q}(q(t), \{q_\alpha, q_\beta\}), q_{\beta, q_\alpha}) \\ &= \begin{cases} q_\beta; & \|(\tilde{Q}(q, \{q_\alpha, q_\beta\}) - q_\alpha)\| > \|\tilde{Q}(q, \{q_\alpha, q_\beta\}) - q_\beta\| \\ q_\alpha; & \text{otherwise} \end{cases} \end{aligned} \quad (87)$$

where  $M_{\{q_\alpha, q_\beta\}} = (q_\beta + q_\alpha)/2$ . This scheme is like applying usual particle-based forces for a very dense boundary (Fig. 9).

The maximum area covered by an aggregate of robots is a function of the comfortable distance  $D_S$ . Suppose  $D_S$  is fixed, and denote the area by  $\mathcal{A}_{D_S}$ . If the area of the target curve  $\gamma_D$ ,

$$\mathcal{A}_{\gamma_D} = \frac{1}{2} \oint f(\gamma_D(s)) \gamma'_D(s) ds, \quad (88)$$

$f(x, y) = (y - x)$  is smaller than  $\mathcal{A}_{D_S}$ , then the interior robots will be compacted together but will still cover this area uniformly. On the other hand, if it is larger, then part of the area surrounded by  $\gamma_D$  will be empty. This means that  $D_S$  should increase to compensate.  $D_{S_i}(t)$  can be changed adaptively as a function of the difference  $\mathcal{A}_{\gamma_D} - \mathcal{A}_{D_S}$ . Note the dependence on time as well as the individual. This method can also be implemented locally. Define the measure of pressure (or lack of it) on an interior robot  $R_i$  as

$$\begin{aligned} \Psi_i(t) = & \sum_{\alpha \in \hat{N}_{R_i}(t)} \frac{\|q_\alpha(t) - q_i(t)\| - (D_{S_i}(t) - \varepsilon)}{D_{S_U} - D_{S_L}} \\ & + \frac{(\theta_{\alpha+1} - \theta_\alpha) - \pi/3}{2\pi} \end{aligned} \quad (89)$$

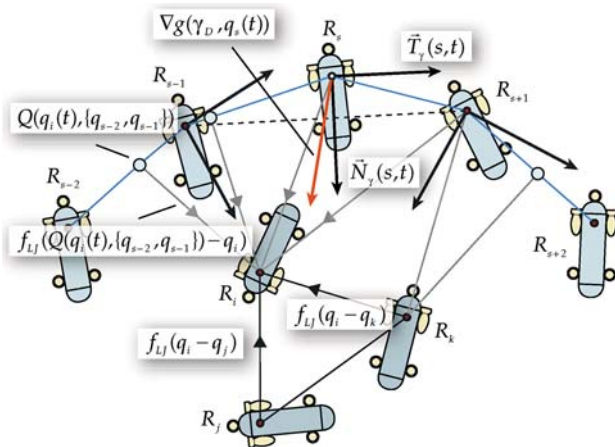


Fig. 9 Motion of boundary and interior robots

where  $\hat{N}_{R_i}$  is the same as  $N_{R_i}$  except that the elements are ordered according to

$$\theta_\alpha = \text{atan2}(y_{q_\alpha(t)-q_i(t)}, x_{q_\alpha(t)-q_i(t)}) \quad (90)$$

in the interval  $[0, 2\pi]$ ,  $D_{S_U}$  and  $D_{S_L}$  denote bounds on  $D_S(t) \cdot \Psi_i(t)$  is actually measure of how far the local status is from a perfect hexagon. The minimum of  $\Psi_i(t)$  (maximum pressure) is

$$L_{\Psi_i} = -(\|S\| - 1) \left( \frac{D_{S_i}(t) - \varepsilon}{D_{S_U} - D_{S_L}} + \frac{1}{6} \right) \quad (91)$$

(corresponding to a situation where all of the aggregate is located at  $q_i(t)$ ) and the maximum (least pressure) occurs when there is only one robot at the farthest end of the neighborhood and is calculated as

$$U_{\Psi_i} = \frac{r(N_i) - D_{S_i}(t) + \varepsilon}{D_{S_U} - D_{S_L}} + \frac{5}{6}. \quad (92)$$

When  $\hat{N}_{R_i} = \emptyset$ , we put  $\Psi_i(t) = U_{\Psi_i}$ . Define

$$\Lambda(v) = \begin{cases} v \cdot (D_{S_U} - D_{S_i})/U_{\Psi_i}; & v > 0 \\ 0; & v = 0 \\ v \cdot (D_{S_L} - D_{S_i})/L_{\Psi_i}; & v < 0 \end{cases} \quad (93)$$

The algorithm is as follows: when an equilibrium is reached,  $D_{S_i}(t)$  is updated using

$$D_{S_i}(t) = D_{S_i}(t) + \Lambda(\Psi_i(t)). \quad (94)$$

This will perturb the system from equilibrium. To achieve uniform coverage, all the  $D_{S_i}(t)$  should be the same. This can be achieved via synchronization among interior robots. A strategy for this is evolving  $D_{S_i}(t)$  according to the formula

$$D_{S_i}(t) = \frac{D_{S_i}(t) + \sum_{R_j \in N_i(t) \cap I_S} D_{S_j}}{1 + \|N_i(t) \cap I_S\|} \quad (95)$$

This scheme has been proved to be convergent (Jadbabaie et al., 2003). After convergence,  $D_{S_i}$  will converge to the average

$$\frac{1}{\|I_S\|} \sum_{R_i \in I_S} D_{S_i}. \quad (96)$$

Here, we instead use the simplest solution. We put  $D_S$  to a large value such we will always have  $\mathcal{A}_{\gamma_D} \ll \mathcal{A}_{D_S}$ . This simple solution has the drawback that the motion of the interior robots has to be sufficiently shunted to prevent large abrupt forces.

### 9. Special cases

In this section, we discuss two classes of shapes. One is the *ellipsoid*, which can be related to previous work in Belta and Kumar (2002), and the other is one which facilitates *splitting* and *joining* two assemblies of robots.

Unlike small formations, join and split operations for a large aggregate with approximate shapes is not trivial. Here, we just mention a possible avenue. The main idea is that the boundary shape descriptors can be calculated in such a way that the resulting shape is more appropriate for these operations. As an example, consider the shape in Fig. 10, composed of two overlapping closed curves (circles in this case). This shape is clearly more suitable for splitting. Let  $\gamma_r(s)$  denote a circle with radius  $r$  located at  $(0, 0)$ . Let  $0 < L < 2r$  be given. From the figure, it is obvious that  $\theta = \text{asin}(L/(2r))$  and that  $d = r \cos(\theta)$ . Now, define the shape by

$$\gamma_S(s) = \begin{cases} \gamma_r(s) + d; & 0 \leq s < \frac{1}{4} \\ \gamma_r\left(s + \frac{\pi - \theta}{4\theta}\right) - d; & \frac{1}{4} \leq s < \frac{3}{4} \\ \gamma_r(s) + d; & \frac{3}{4} \leq s < 1 \end{cases} \quad (97)$$

Denote the scaled, translated and rotated curve by  $\tilde{\gamma}_S(s) = s_S R_z(\phi_S)\gamma_S(s) + t_S$ . Suppose we discretize  $\gamma_S$  with  $N$  points and compute the Fourier descriptors. Let  $\mathcal{F} = \{\mathcal{F}_{-M}, \dots, \mathcal{F}_M\}$  be the calculated descriptors. Using  $\mathcal{F}$ , the boundary robots will form an approximate shape (shown in the figure with thick lines). Define the  $b$ -Neighborhood of  $R_i \in B_S$  by

$$N_b^B(R_i) = \{R_j \in B_S | j \leq (i + b)_{\text{mod}N}, j \geq (i - b)_{\text{mod}N}\} \quad (98)$$

Suppose, for a certain  $R_i \in B_S$ , there exists a  $R_j \in (B_S \cap N_r(R_i)) \setminus N_b^B(R_i)$ , such that  $\|q_i(t) - q_j(t)\|$  is the minimum, and similarly for  $R_j$ . If there is just one

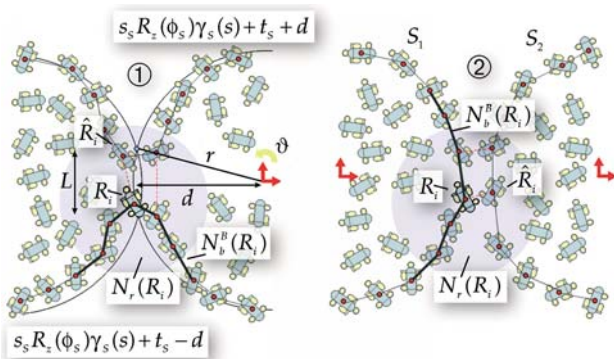


Fig. 10 Split and join operations

such pair  $\{R_i, \hat{R}_i = R_j\}$ , then, for splitting, we set

$$\begin{aligned} l(R_i) &= \hat{R}_i, r(\hat{R}_i) = R_i, \quad \text{and we put} \\ r(R_{i+1}) &= R_{j-1}, \quad \text{and} \quad l(R_{j-1}) = R_{i+1}. \end{aligned} \quad (99)$$

Alternatively, one might define

$$\begin{aligned} r(R_i) &= \hat{R}_i, l(\hat{R}_i) = R_i, \\ l(R_{i-1}) &= R_{j+1}, \quad \text{and} \quad r(R_{j+1}) = R_{i-1}. \end{aligned} \quad (100)$$

After this, a reordering should be done on the newly created curves.  $L$  should be chosen small enough so that no interior robots get trapped inside the narrow region where the two ends of the curve are close together.

For joining two aggregates  $S_1$  and  $S_2$ , they first have to form circular shapes with roughly the same scale and get close enough together. The procedure is very similar to splitting but encompasses two aggregates. Suppose, for a certain  $R_i \in B_{S_1}$ , there exists a  $R_j \in B_{S_2} \cap N_r(R_i)$ , such that  $\|q_i(t) - q_j(t)\|$  is the minimum in  $R_b^B(R_i)$ , and similarly for  $R_j$ . Then, we can set

$$\begin{aligned} l(R_i) &= R_j, r(R_j) = R_i, \\ r(R_{i+1}) &= R_{j-1}, \quad \text{and} \quad l(R_{j-1}) = R_{i+1}, \end{aligned} \quad (101)$$

or, alternatively,

$$\begin{aligned} r(R_i) &= R_j, l(R_j) = R_i, \\ l(R_{i-1}) &= R_{j+1}, \quad \text{and} \quad r(R_{j+1}) = R_{i-1}, \end{aligned} \quad (102)$$

The two alternative cases in both operations can be selected according to some bound on maximum curvature. In both operations, if there are more than one pair satisfying the minimum condition, a more involved strategy should be devised. In Belta and Kumar (2002), control rules are developed to form ellipsoidal shapes with large number of robots. In our approach, such ellipsoids can be formed using just the two descriptors  $\{\mathcal{F}_{-1}, \dots, \mathcal{F}_1\}$ . They identify the main axes of the ellipsoid (the best fit to a shape).  $\mathcal{F}_{-1}$  and  $\mathcal{F}_1$  would be the result of applying Fourier transformation to the unit circle. An ellipsoidal curve with axes  $0 < A_x \leq 1$  and  $0 < A_y \leq 1$ , can then be constructed using the formula

$$sR(\phi)(\mathcal{F}_{-1}e^{-2j\pi \frac{s}{N}} + \mathcal{F}_1e^{-2j\pi \frac{s}{N}}) + t, \quad (103)$$

where the equation

$$\begin{aligned} &\frac{(\mathcal{F}_{1x} \cos(2\pi i/N) + \mathcal{F}_{-1x} \cos(-2\pi i/N))^2}{A_x^2} \\ &+ \frac{(\mathcal{F}_{1y} \sin(2\pi i/N) + \mathcal{F}_{-1y} \sin(-2\pi i/N))^2}{A_y^2} = 1 \end{aligned} \quad (104)$$

should be satisfied. We can arbitrarily choose three of the parameters and calculate the fourth.

### 10. Simulation results

In this section, we present two simulations. The first one shows the boundary-assignment procedure in action. The second one demonstrates the morphing process through snapshots of the aggregate.

In Fig. 11(1), an initial assembly of robots is shown. The initial distribution must be such that the formation graph is connected (in the sense defined in Section 2). This is, of course, just a necessary but not a sufficient condition for the whole aggregate to form a single cohesive group. Figure 11(2) through to 11(4) show the evolution in time of the swarm. In this simulation, the boundary-determination process is active from the start. Here, we have chosen  $\theta_G = 70$ . Note that, as the energy of the whole system,  $\phi_{LJ}^S(q(t))$ , is decreased, the quality of the boundary improves. In snapshots 1, 2, and 3, only rule 1 is active. In snapshot 4, the other rules have been turned on as well.

Figure 12 shows the sequence of snapshots of an aggregate composed of  $N$  robots going through the morphing process. Here, the target curve coincides with the centre of mass of the initial aggregate. Translation and rotation can be carried out using the simple law

$$\dot{q}_i(t) = v_0 \frac{\mu_d - \mu(t)}{\|\mu_d - \mu(t)\|} + (e^{i(h\omega_0 t)q_i(t)} - q_i(t)) \tag{105}$$

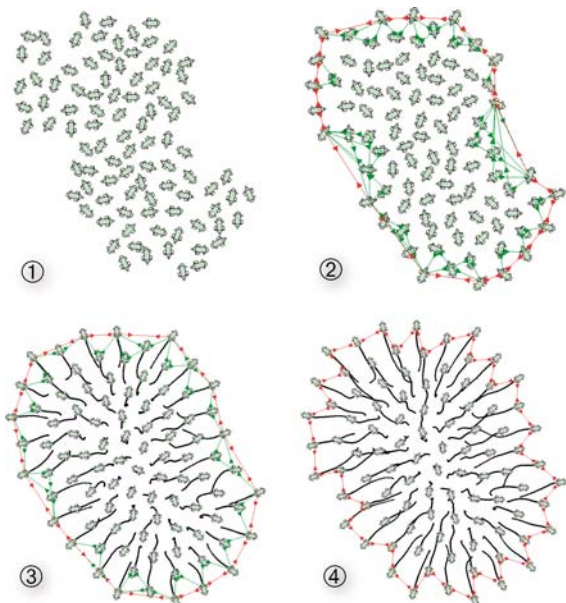


Fig. 11 Boundary self-organization

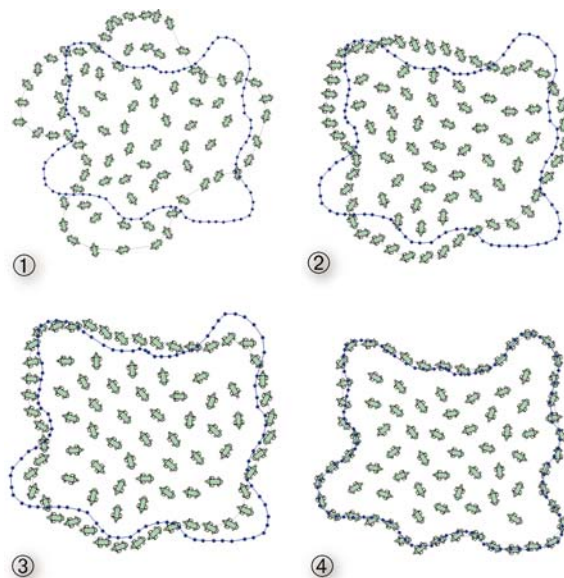


Fig. 12 Morphing

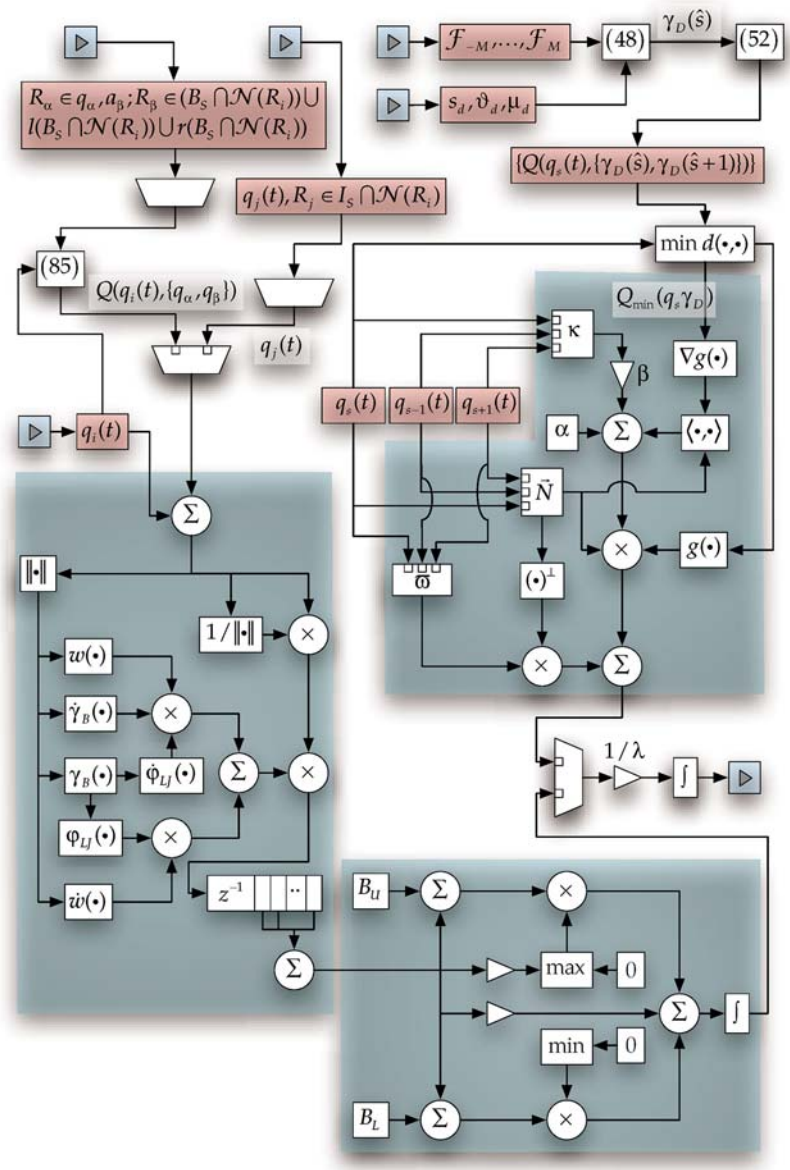
where  $v_0$  and  $\omega_0$  denote nominal linear and angular velocities, and  $h$  determines the sense of rotation. Alternatively,  $\mu_d$  and  $\theta_d$  can be entered into formula (47) defining  $\gamma_D$ .

Figure 13 shows the *function block diagram* of the control system when in morphing mode. The various equations derived in the paper are shown in the corresponding blocks.

### 11. Conclusions and further research

We showed that large aggregates of autonomous vehicles with limited communication capabilities can be commanded to form complex geometric shapes using low-dimensional shape descriptors. We also showed that the problem can be further simplified if the aggregate is treated as being composed of two components, its boundary and its interior, with the boundary performing shape formation and the interior part maintaining a uniform distribution. We solved the problem of boundary assignment through a simple self-organization strategy. We also derived precise distributed control rules, relying on neighborhood information, implementing these behaviours. The validity of the process was demonstrated by simulation results. There is definitely a lot of work to be done before this scheme can be implemented on real robots, and is currently a topic of further research. Also, we just addressed the 2-D case. The method should be extended to motion on surfaces. Also, we have only considered synchronous update; a communication strategy with delays is more close to reality. It is worthwhile to note that the formulation we used here can be extended to a continuous one. Thus, the formation can be described by  $\mathcal{R}(t) = (\gamma(t), \rho(t))$ ,

**Fig. 13** Control system function block diagram



where  $\gamma : [0, 1] \otimes \mathfrak{R}_+ \rightarrow \mathfrak{R}^2$  is a family of curves, and  $\rho : I(\gamma(t)([0, 1])) \rightarrow \mathfrak{R}$  is the concentration function, with  $I$  giving the interior of a closed curve in  $\mathfrak{R}^2$ . The motion of  $\gamma$  is given in Eq. (75), and the motion of the interior, i.e.  $\rho$ , is governed by the diffusion equation

$$\frac{\partial \rho(x, y, t)}{\partial t} = D \left( \frac{\partial^2 \rho}{\partial x^2} + \frac{\partial^2 \rho}{\partial y^2} \right) \tag{106}$$

where  $D = 1 - e^{-\alpha(\rho(t) - \rho_d)}$  is the diffusion rate corresponding to the force acting on a particle, subject to the boundary condition

$$\frac{\partial \rho(x, y, t)}{\partial \bar{N} \gamma(s, t)} = 0 \tag{107}$$

**References**

Belta, C. and Kumar, V. 2002. Towards abstraction and control for large groups of robots. In *2nd International Workshop on Control Problems in Robotics and Automation*, Las Vegas, NV, U.S.A.

Belyaev, A.G., Anoshkina, E.V., and Yoshizawa, S. 1999. Nonlinear spline generation with curve evolutions driven by curvature. *Shape Modelling International*. Aizu, Wakamatsu, Japan, pp. 146–153.

Bertozzi, A.L., Kemp, M., and Marthaler, D. 2005. Determining environmental boundaries: asynchronous communication and physical scales. In V. Kumar, N. Leonard, and A.S. Morse (eds.), *Proceedings of the Block Island Workshop on Cooperative Control 2003*, Lecture Notes in Control and Information Systems, Springer-Verlag, Vol. 309.

Chaimowicz, L. and Kumar, V. 2004. A framework for the scalable control of swarms of unmanned ground vehicles with unmanned aerial vehicles. In *Proceedings of the 10th International Conference on Robotics and Remote Systems for Hazardous Environments*. Gainesville, FL, USA.



- Chaimowicz, L., Lathan, M., and Knmar, V. 2005. Controlling swarms of robots using interpolated implicit functions. In *Proc. of the IEEE Int. Conf. on Robotics & Automation*. Barcelona, Spain.
- Chenyang, X., Yezzi, A., and Prince, J.L. 2000. On the relationship between parametric and geometric active contours. In *Proc. of 34th Asilomar Conf. on Signals, Systems, and Computers*.
- Cortes, J., Martinez, S., Karatas, T., and Bullo, F. 2004. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255.
- Curtin, T.B., Bellingham, J.G., Catipovic, J., and Webb, D. 1989. Autonomous oceanographic sampling networks. *Ocean-ography*, 6:86–94.
- Egerstedt, M. and Hu, X. 2001. Formation constrained multi-agent control. *IEEE Trans. on Robotics and Automation*, 17(6):947–951.
- Gazi, V. and Passino, K.M. 2004. Stability analysis of social foraging swarms. *IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics*, 34(1).
- Jadbabaie, A., Lin, J., and Morse, A.S. 2003. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6).
- Kalantar, S. and Zimmer, U. 2004. Contour-shaped formation control for autonomous underwater vehicles using canonical shape descriptors and deformable models. In *Proceedings of the IEEE International Conference on Marine Technology and Ocean Science (OCEANS 2004)*. Kobe, Japan.
- Kichenassamy, S., Kumar, A., Oliver, P., Tannenbaum, A., and Yezzi, A. 1995. *Gradient Flows and Geometric Active Contour Models*. Fifth International Conference on Computer Vision, MIT, Cambridge, Massachusetts.
- Okubo, A. 1980. *Diffusion and Ecological Problems*, Vol. 10 in Biomathematics Series, Springer-Verlag.
- Reif, J.H. and Wang, H. 1999. Social potential fields: A distributed control for autonomous robots. *Robotics and Autonomous Systems*, 27.
- Ren, W. and Beard, R.W. 2004. A decentralized scheme for spacecraft formation flying via the virtual structure approach. *AIAA Journal of Guidance, Control and Dynamics*, 27(1).
- Spletzer, J. and Fierro, R. 2005. *Optimal Positioning Strategies for Shape Changes in Robot Teams*. ICRA 2005, Barcelona, April 2005.
- Starke, J. 1997. *Kombinatorische Optimierung auf der Basis gekoppelter Selektionsgleichungen*, Ph.D. thesis, University Stuttgart, Verlag Shaker, Aachen, Germany.
- Tonnesen, D.L. 1998. *Dynamically coupled particle systems for Geometric modelling, reconstruction, and animation*, PhD thesis, University of Toronto.
- Wang, H., Kearney, J., and Atkinson, K. 2002. Robust and efficient computation of the closest point on a spline curve. In *Proc. of the 9th Intl. Conf. on Curves and Surfaces*, Saint-Malo, France.
- Zhang, F., Goldgeier, M., and Krishnaprasad, P.S. 2003. Control of small formations using shape coordinates. In *Proc. of 2003 IEEE International Conference on Robotics and Automation*. Taipei, Taiwan, pp. 2510–2515.



**Shahab Kalantar** got his Master's degree in Robotics and Machine Intelligence in 1997 from Tehran University and his Bachelor's in Applied Mathematics in 1991. He is now a PhD candidate at the Australian National University. His current research interest is the interaction of deformable formations of autonomous vehicles with environmental fields.



**Uwe R. Zimmer** is currently a fellow at the Australian National University at the Faculty of Engineering and Information Technology and at the Research School of Information Sciences and Engineering. He received his Dr. rer. nat. degree in 1995 at the University of Kaiserslautern, Germany. Building up an underwater research group in Germany at the former German National Research Center for Information Technology, and setting up a research laboratory for autonomous robust systems in Kyushu, Japan have been the other major activities since then. He is currently the head of the underwater robotics group at the Australian National University. His major research interests are in distributed swarm control and communication with application to real-time and real-world environments.