OCEANS 2004, Kobe, Japan
(authors' manuscript – do not distribute)

THE AUSTRALIAN NATIONAL UNIVERSITY

Contour Shaped Formation Control for Autonomous Underwater Vehicles Using Canonical Shape Descriptors and Deformable Models

Shahab Kalantar & Uwe R. Zimmer

Australian National University
Research School for Information Science and Engineering
and the Faculty for Engineering and Information Technology[1]
Canberra, ACT 0200, Australia
shahab.kalantar@rsise.anu.edu.au | uwe.zimmer@ieee.org

Cooperative tasks such as swarms following gradients, detecting environmental boundaries (defining natural features) and exploration by aggregates of mobile autonomous robots have gained much attention in the past few years. Aggregates come in various forms including rigid formations, involving a few robots, with clearly-defined simple shapes or swarms, with up to hundreds of robots, but with no particular shape characteristics. In the aforementioned tasks, a large aggregate is required to form complex shapes. When the number of robots increases, it is very difficult or even senseless to manually determine the position of each and every agent within the formation. In this paper, we concentrate on a special kind of geometric formation (open and closed contours). We use Fourier descriptors, as one of the most natural ways of representing curves, and active contour models, to make the aggregate exhibit desirable physical behaviours. We apply these kinds of formations to the case of autonomous underwater vehicles collectively exploring, mapping, and adapting to environmental features. Many of the features found in underwater landscapes are shaped as open or closed contours so that the proposed combination makes sense. In our approach, a group of mobile agents can synthesize a curve given a small set of canonical invariant descriptors.

Keywords: **autonomous underwater vehicles, formation control, swarm control**

1. Introduction

Exploration of underwater environments is currently one of the most important tasks confronted by researchers around the world. Realization of truly Autonomous Ocean Sampling Networks (AOSN) [1] [5] is considered the holy grail of this line of research. Although an underwater landscape could in theory be navigated and explored by just one underwater vehicle, it is commonly believed that this task should in practice be approached from a multi-robot cooperative perspective. This paradigm makes ocean observation possible through numerous mobile networked sensor platforms. Autonomous submarines play the role of these sensor platforms. They are equipped with special application-specific sensors as well as limited bandwidth communication capabilities. A basic capability of these networks is the ability to form certain shapes dictated by the task.

In most cases, these robots are homogeneous (i.e., have the same structure and functionality), share local information with each other via local interactions, are run by the same control rule which together realize a global complex behaviour (i.e., the control strategy is distributed among the vehicles), and are governed by a global task. Research on robot aggregates has been divided into two schools. One is *formation control* where a relatively small number of robots are required to form simple geometric shapes. The approaches proposed for this class of problems usually don't scale up. The other one is *swarm research* where the emergence of behaviours and patterns among a large number of identical agents is of interest. In swarming research, exact shape of the aggregate is not of interest.

We are interested in large aggregates of robots and automatic methods of making them form complex geometric shapes. Obviously enough, 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. In these kinds of application, the robots form a virtual structure [2] which then has to go through rigid-body transforma-

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

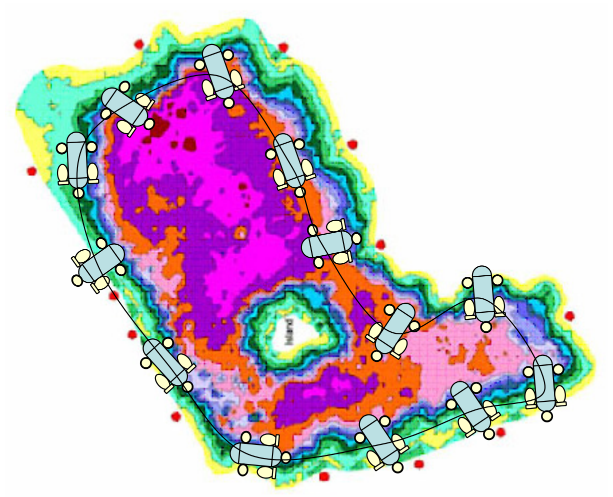

figure 1: Adapting to underwater landscape (closed contours)

tions. Formation graphs [3] have been used to capture the interrelationships between individual robots. Formation functions [4] are similarly used to describe geometric constraints between robots' positions and orientations. Using these methods, the locations of robots with respect to other robots (or a virtual leader) have to be determined beforehand. The motivation behind all these research efforts is to somehow decouple the problems of formation keeping and manoeuvring [6]. When the number of robots increases, resembling more a swarm than a simple formation, it is very hard to draw a line between formation control and swarming behaviour. Instead, the robots are generally required to form a non-rigid approximate shape rather than a rigid virtual structure. In [7], an alternative method of representation is proposed whereby 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. On the other hand, a multitude of canonical invariant representations have 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.

We only consider *contour-shaped formations*. Among the shape representation methods proposed in machine vision, Fourier descriptors are the most appropriate for curve-shaped entities. It is invariant to several transformations and it is very compact. Even such a simple setting has numerous applications. Here, we'll enumerate some of them.

a. Many of the natural features underwater (usually represented by bathometric maps) form open or closed curves. Figure 1 shows an example. Such features can be interpreted as level-sets of underwater landscapes.

b. In the case of an oil leakage, the affected area can be described by shape descriptors calculated by some aerial vehicles which are communicated to the underwater vehicles for enclosure of the covered area. From that point on, suitable potential functions can be used by the formation to track the level-set or, alternatively, updated information can come from aerial vehicles.

c. Another interesting application is that of enclosing and tracking underwater plumes. Given a rough model for the evolution of the plume (for predicting the boundary of the plume), complemented by sensor measurements obtained from individual robots, the aggregate can be made to uniformly occupy the predicted boundary. Invariant descriptors can be used to describe the boundary and elastic forces, along with some potential calculated using the physical model and estimates obtained from sensors, can be used to adapt the contour formation to the plume boundary.

d. One way of identifying features is through computing signatures, i.e., 1-D functions for representing 2-D boundaries (or areas). Many kinds of signatures have been introduced (most of them invariant with respect to certain transformations) such as $r-s$ curves, or $\phi-s$ curves. Here, we use Fourier descriptors which are descriptive and very compact at the same time, while satisfying all the desirable invariance properties.

e. If the features can be uniquely identified, then they can be compared. Recalculating a certain level-set of a landscape and comparing it with the one computed some time ago can indicate the amount of change it has gone through.

f. Similar to *e* – pre-mapped features can be systematically searched for using a rough spatial model.

g. *Objects* and *fields* are two fundamental conceptualizations of the entities comprising the natural environments [8]. Objects refer to things in the world while a field refers to a single-valued function of location in 2-D space. A collection (or aggregate) of these entities can be connected together through a topological map. Each node of such a map is an invariant description of an object or a field together with geometric (or other) attributes such as position (absolute or relative), orientation or scaling.

h. In many applications, a swarm is required to cover a certain area with a certain shape. This problem can be conveniently decomposed into two sub-problems: controlling the boundary of the shape using designated robots and having the robots inside the boundary make a uniform distribution.

We also use active contour models (elastic bands) to enforce some degree of *cohesion* and *smoothness*. It can also be used for adaptation, as the following sections will describe. This model realizes what we intuitively think a curved formation should behave like, from a physical point of view.

2. Serafina: Small Autonomous Submersibles

Serafina is the name given to small light-weight submersibles currently being developed in our laboratory. Their primary purpose is to investigate the possibility of creating schools of autonomous underwater vehicles capable of exploring the underwater landscape. They are equipped with five thrusters attached to a main hull containing the controller, the sensors and the batteries. The main circuit board consists of a serial line connector for debugging and down loading programs and data, the CPU module (Motorola MPC 555 PowerPC µcontroller unit), sonar sensor module (Airmar 25-162-01 200 kHz), the compass module (2 axis strapped down magnetometer from Tri-M Systems), linear accelerometers (2 ADXL 311 dual axis from Analog Devices), pressure sensor (26 PCB Honeywell), optical short range sensors (short-range communication), long-range receiver and transmitter, and motor drivers. Figure 2 shows a picture of this vehicle while figure 3 shows the hardware block diagram.

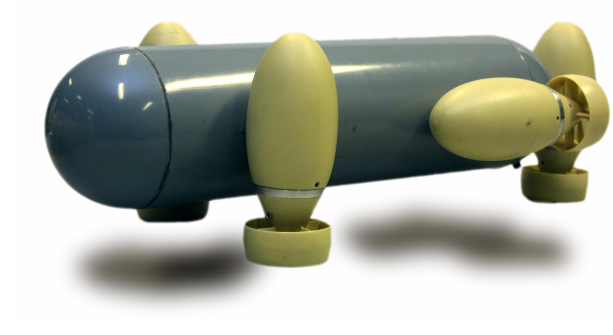

figure 2: Serafina underwater vehicle

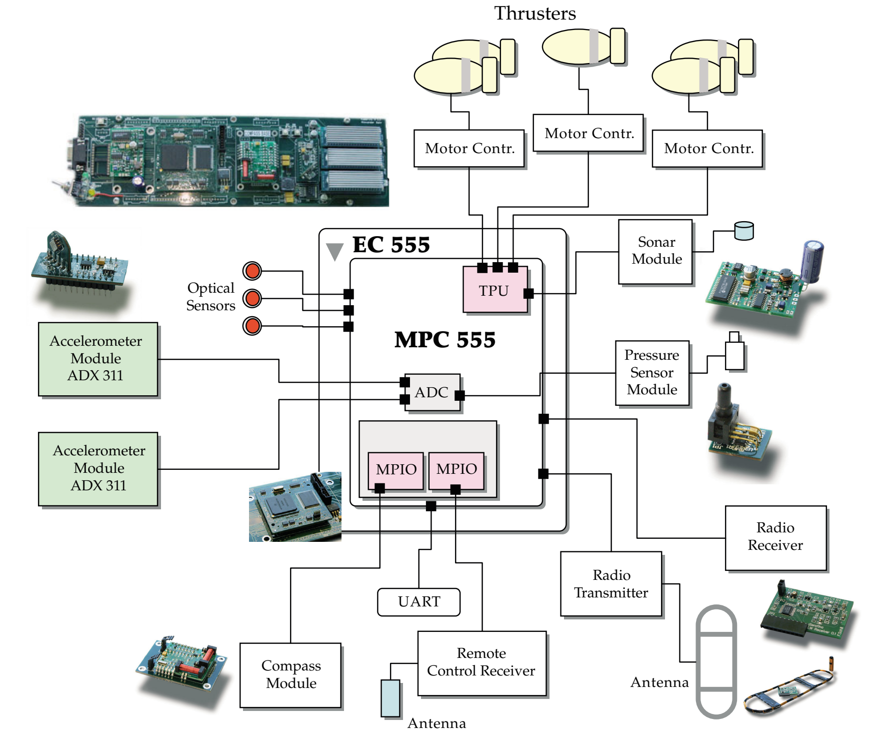

figure 3: Serafina hardware block diagram

Serafinas are neutrally buoyant and pressure sensors can be used to keep them at a certain depth. Also, the roll and pitch angles can be stabilized around zero. This way, their dynamics can be modelled as simple unicycles. The motion of the point at the middle of the axis connecting the two rear thrusters is equal to the sum of the forces acting on the robot and governed by the differential equation

$$m\frac{\partial^2 q_i}{\partial t^2} + \lambda\frac{\partial q_i}{\partial t} = f_i\,, \qquad (1)$$

where m is a mass-dependent coefficient and λ is the damping factor. When the motion is slow, the effect of the acceleration term is only transient. In this case, $m(\partial^2 q_i/\partial t^2) \sim 0$ and the equation reduces to $\lambda(\partial q_i/\partial t) = f_i$. Interpreting the force f_i as a vector from q_i to the desired target q_{d_i}, we can employ feedback linearization or a simple reactive controller which directly provides control values for the rear thrusters.

3. Invariant Representations and Fourier Descriptors

When an invariant representation is used, the problem of forming a shape can be effectively decoupled from that of translation, rotation and scaling (i.e., affine transformations). Suppose that S is the M-dimensional space of shape descriptors. An element $\tilde{s} = (s_0, \dots, s_{M-1}) \in S$ can be used to *uniquely* describe a certain class of shapes up to translation, rotation, and scaling. It should be noted that uniqueness is a relative concept in this context and is defined up to a tolerance level. For example, in the case of Fourier descriptors, an infinite number of variables are required to uniquely define a closed contour. The same is also true about any other canonical representation, such as higher-order moments. Two shapes $\hat{S}_1$ and $\hat{S}_2$ are then considered as the same if $\|s_{1i} - s_{2i}\| \leq \varepsilon$ for a given tolerance level ε and for each $s_{1i} \in \hat{S}_1$ and $s_{2i} \in \hat{S}_2$ where $i = 0, \ldots, M-1$. Care has to be taken to select M as large as is necessary. Now let $G \subset \Re^3$ denote the configuration space of the formation as a virtual body. Each element in the Lie group G is denoted as $\hat{g} = (\mu, \theta, l)$ where μ is the centre of mass (or gravity), θ is the orientation of the body, and l is the scaling which can be used to shrink or expand the virtual body. $\hat{f} = (\hat{s}, \hat{g}) \in S \otimes G$ uniquely defines the formation state. Let $q_i \in Q$ denote the state (position and orientation) of robot R_i where $i = 0, \ldots, N-1$. In this paper, $q_i = (x_i, y_i)$ is the position of the robot with respect to the body-fixed coordinate frame centred at μ (or μ^w). The collective state of the system is then $q = (q_0, \ldots, q_{N-1}) \in Q^N$. Elements of $\hat{s}$ and $\hat{g}$ are a function of q_i's:

$$(\hat{s}, \hat{g}) = \Psi(q_0, \ldots, q_{N-1}) = (\Psi_s(q), \Psi_g(q)) \tag{2}$$

If $\mathcal{D}\Psi$ is the differential of the map Ψ, then we will have $\mathcal{D}\Psi\dot{q} = (\dot{s}, \dot{g})$. The minimum-norm solution of the above equation in the sense of least-squares is given by

$$\dot{q} = \mathcal{D}\Psi^{-1}\dot{f} = \mathcal{D}\Psi^T(\mathcal{D}\Psi\mathcal{D}\Psi^T)^{-1} \tag{3}$$

where

$$f = (\mu^w, \theta, l, s_0, \ldots, s_{M-1})^T \equiv (f_1, \ldots, f_{M-3})^T. \tag{4}$$

Now, for every f_j we will have $dq = \mathcal{D}\Psi^{-1}df_j$, where

$$df_j = [df_{jx}, df_{jy}]^T, \tag{5}$$

$$\text{and } \mathcal{D}\Psi_{f_j} = \left[\frac{\partial f_{jx}}{\partial q}, \frac{\partial f_{jy}}{\partial q}\right]^T. \tag{6}$$

For a desired f_{jd}, we can set

$$df_j = \lambda[f_{jx} - f_{jdx}, f_{jy} - f_{jdy}]^T, \tag{7}$$

implementing a simple linear feedback loop. This is basically a first order approximation so that for descriptors more complicated than first and second order moments, it will get stuck in undesirable local minima. Fortunately, at least for the case of Fourier descriptors, we don't need to resort to such calculations because the desired contour can be easily synthesized using the descriptors and the group actions can be realized as affine transformations on the robots' positions.

Complex Fourier descriptors are among the most popular methods in image processing for describing shapes represented as closed contours [9]. Any closed curve, parametrized by s (the arc length) and with perimeter T can be described by its Fourier expansion

$$q(s) = \sum_{n=-\frac{N}{2}}^{\frac{N}{2}} c_n e^{jn\omega s}, \ \omega = \frac{n\pi}{T} \tag{8}$$

where the Fourier coefficients are given by

$$c_n = \frac{1}{T}\int_0^T q(s)e^{-jn\omega s}ds \tag{9}$$

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, $i = 0, \ldots, N-1$, the Fourier descriptors c_n, $n = -N/2, \ldots, N/2$ are the coefficients of the Fourier transform of q_i:

$$q_i = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}} c_k e^{2\pi j\frac{ki}{N}}, \ c_k = \frac{1}{N}\sum_{i=0}^{N} q_i e^{-2\pi j\frac{ik}{N}} \tag{10}$$

According to Shannon's theorem, the highest frequency is obtained for $k = N/2$. c_k's with $k > N/2$ give no information when the shape is discretized by N points. c_k's actually represent the frequency contents of the curve. As k increases, more detailed information about higher frequencies will be gained. c_0 is the areal centre of gravity of the shape. c_1 describes the size of the shape. Using only c_1 and c_{-1}, the synthesized shape will be an ellipse. Higher frequency components distort the ellipse defined by c_1. The set of Fourier descriptors invariant with respect to translation, rotation, scaling and choice of starting point are given by

$$(\mathcal{F}_{n_x}, \mathcal{F}_{n_y}) = \mathcal{F}_n = \frac{|c_n|}{|c_0|}e^{j\alpha_n} \tag{11}$$

where

$$c_n = \frac{-T}{(2\pi n)^2}\sum_{k=0}^{N-1}\left\{\frac{q_{k+1} - q_k}{\|q_{k+1} - q_k\|} - \frac{q_k - q_{k-1}}{\|q_k - q_{k-1}\|}e^{j\beta_k n}\right\} \tag{12}$$

$$c_o = \frac{1}{2T}\sum_{k=0}^{N-1}(q_{k+1} - q_k)\|q_{k+1} - q_k\| \tag{13}$$

$$T = \sum_{k=0}^{N-1}\|q_{k+1} - q_k\| \tag{14}$$

$$\beta_{kn} = -\frac{2\pi n}{T}\sum_{k-1}^{i=0}\|q_{i+1} - q_i\| \tag{15}$$

$$\alpha_n = -(\varphi_n + (1-n)\varphi_2 - (2-n)\varphi_1) \tag{16}$$

$$\varphi_n = \arg(c_n) = \tan^{-1}(c_{n_y}/c_{n_x}) \tag{17}$$

In the above formulas, the curve is described by the vertices $q_o, \ldots, q_{N-1}$ and the periodic boundary condition is defined by $q_{N-1+i} = q_{i-1}$ and $q_{-i} = q_{N-i}$, where $i = 1, \ldots, N$. For open contours, the boundary conditions are $q_{-i} = q_i$ and $q_{N-1+i} = q_{N-1-i}$. A few of the descriptors are sufficient to describe very complex shapes.

We use c_0 as the centre of gravity of the contour, viewed as a virtual body. c_0 gives the *areal centre of gravity* of the polygon defining the curve.

The parameter $\varphi_1 = \arg(c_1)$ gives the angle of the main principal axis and the descriptors $\mathcal{F}_{-1}$ and $\mathcal{F}_1$ can be used to reconstruct the loci of the best least squares fit of the object with an ellipse, this coordinate system being uniquely defined. The ellipse is given by the following formula

$$q(s) = \mathcal{F}_{-1}e^{-j\frac{2\pi}{T}s} + \mathcal{F}_1 e^{j\frac{2\pi}{T}s} \qquad (18)$$

In practice, φ_1 is dependent on the starting point and can't be used directly. In fact, the reconstructed curve is rotated with respect to the original object. This rotation actually gives the orientation of the normalized object. The difference between the orientation of the normalized object and the original one is given by $\Delta\Phi = (2\varphi_1 - \varphi_2)_{mod 2\pi} + \pi$. Using $\Delta\Phi$, we can compensate for the difference in rotation and recover the orientation of the original object.

4. Forming Shapes with Fourier Descriptors

As was seen in previous sections, c_n's give rise to two sets of parameters. One is the set of invariant features $\mathcal{F}_{-M}, \ldots, \mathcal{F}_M$ which comprises the canonical representation and the other is the set $\{\mu_d^w, \theta_d, s_d, n_d, \varphi_1, \varphi_2\}$ describing the geometrical properties of the shape. $\mu_d^w, \theta_d, s_d, n_d$ represent desired areal centre of gravity, orientation (with respect to the original shape), scaling (expansion/shrinking), and starting point, respectively. Also $\varphi_1 = \arg(c_1)$ and $\varphi_2 = \arg(c_2)$.

The performance of the formation can be assessed using some suitable measure. For the case of Fourier descriptors, a commonly used measure is the normalized difference between the desired and actual descriptors. Combined with other shape parameters, the relation

$$\sum_{n=-M, n\neq 0}^{M} (\|\mathcal{F}_n - \mathcal{F}_{n_d}\| + \|\mu_d^w - \mu^w\| + \|\theta_d - \theta\| + \|s_d - s\|) \qquad (19)$$

provides a reasonable similarity measure. The desired final location for a robot R_i is computed as

$$q_{i_d} = s_d R(\theta_d - (2\varphi_1 - \varphi_2)_{mod 2\pi} - \pi) q_i^{\mathcal{F}_d} + \mu_d^w \qquad (20)$$

where $R(\vartheta)$ denotes the rotation matrix around the z axis by ϑ degrees, $\phi = (2\varphi_1 - \varphi_2)_{mod 2\pi} - \pi$, and

$$q_i^{\mathcal{F}_d} = \sum_{n=-M, n\neq 0}^{M} \mathcal{F}_n e^{\frac{2jn\pi i}{N}} \qquad (21)$$

The error between the current position and the desired one is therefore $q_{i_e} = q_{i_d} - q_i$ and we can use a control rule such as the following to drive the robot to the desired location:

$$\frac{\partial q_i}{\partial t} = \gamma \frac{q_i - q_{i_d}}{\|q_i - q_{i_d}\|} = \gamma \frac{q_{i_e}}{\|q_{i_e}\|} \qquad (22)$$

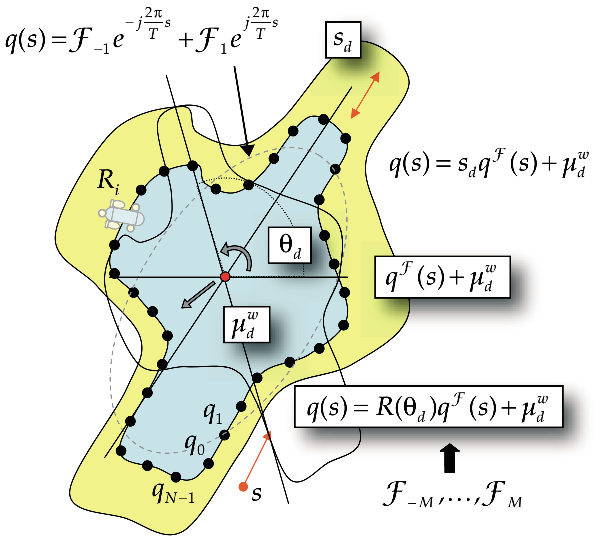
$q(s) = \mathcal{F}_{-1}e^{-j\frac{2\pi}{T}s} + \mathcal{F}_1 e^{j\frac{2\pi}{T}s}$
s_d
R_i
$q(s) = s_d q^{\mathcal{F}}(s) + \mu_d^w$
θ_d
$q^{\mathcal{F}}(s) + \mu_d^w$
μ_d^w
$q(s) = R(\theta_d) q^{\mathcal{F}}(s) + \mu_d^w$
q_1
q_0
q_{N-1}
s
$\mathcal{F}_{-M}, \ldots, \mathcal{F}_M$

figure 4: Shape formation using descriptors

where γ is the strength of attraction (actually, a spring constant). This force corresponds to a quadratic potential $P_{\mathcal{F}} = \|q - q_d\|^2/2$. Thus

$$\frac{\partial q_i}{\partial t} = -\nabla P_{\mathcal{F}}(q) \qquad (23)$$

In practice, the distance between two neighbouring robots should satisfy a bound such as $\delta_{min} \leq \|q_{i+1} - q_i\| \leq \delta_{max}$. This implies that we should always have $\delta_{min} \leq \|q_d(s_{i+1}) - q_d(s_i)\| \leq \delta_{max}$. Using Fourier descriptors, these bounds may sometimes be at odds. If one of them, e.g. the upper bound, is assumed to be satisfied, we can scale the target curve to satisfy the lower bound. To do this, we simply multiply $q_i^{\mathcal{F}_d}$ by $\eta(\delta_{min}/\arg min_{s_i}\|q_d(s_{i+1}) - q_d(s_i)\|)$ where $\eta(\bullet)$ is approximately 1 when the argument is below 1 and linearly ascends, with appropriate slope, for values greater than 1.

5. Calculating Environmental Signatures

As demonstrated in [10], a robot formation shaped as a closed contour and under the pressure of snake and external potential forces can be made to adapt itself to environmental level sets. After the formation has been stabilized, a descriptor-computing mechanism can be switched on to register the current shape of the contour. This will give the canonical representation of the level set. This can be done in a distributed sequential manner by the robots. Neighbouring should synchronize on transmission and reception of signals. The process starts with R_0. R_0 transmits its estimated location q_0 to R_1. R_1 computes the values $c_{0,1} = (q_0 + q_1) \cdot \|q_1 - q_0\|$ and $T_1 = \|q_1 - q_0\|$. It then

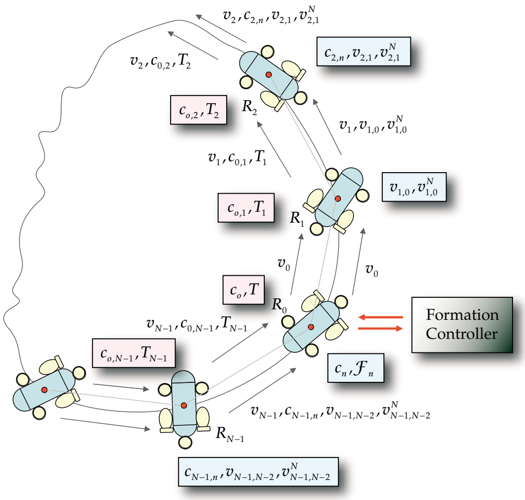
$v_2, c_{2,n}, v_{2,1}, v^N_{2,1}$
$c_{2,n}, v_{2,1}, v^N_{2,1}$
$v_2, c_{0,2}, T_2$
$c_{o,2}, T_2$
R_2
$v_1, v_{1,0}, v^N_{1,0}$
$v_1, c_{0,1}, T_1$
$v_{1,0}, v^N_{1,0}$
$c_{o,1}, T_1$
R_1
v_0
v_0
c_o, T
R_0
Formation Controller
$v_{N-1}, c_{0,N-1}, T_{N-1}$
$c_{o,N-1}, T_{N-1}$
$c_n, \mathcal{F}_n$
$v_{N-1}, c_{N-1,n}, v_{N-1,N-2}, v^N_{N-1,N-2}$
R_{N-1}
$c_{N-1,n}, v_{N-1,N-2}, v^N_{N-1,N-2}$

figure 5: Calculation of descriptors

transmits $c_{0,1}$ and T_1, together with q_1 to R_2. R_2 will then calculate

$$c_{0,2} = (q_1 + q_2) \cdot \|q_2 - q_1\| \text{ and} \tag{24}$$

$$T_2 = \|q_2 - q_1\| \tag{25}$$

This process goes on until R_0 is reached by which time R_0 can compute

$$T = \|q_0 - q_{N-1}\| \text{ and} \tag{26}$$

$$c_0 = ((q_0 + q_{N-1}) \cdot \|q_0 + q_{N-1}\| + c_{0,N-1})/(2T) \tag{27}$$

Now the second round can begin. R_0 transmits its position q_0 and T to R_1. R_1 calculates

$$q_{1,0} = \|q_1 - q_0\| \text{ and} \tag{28}$$

$$q^N_{1,0} = (q_1 - q_0)/\|q_1 - q_0\| \tag{29}$$

and transmits these along with q_1 and T to R_2. R_2 can now compute partial $c_{-M}, \ldots, c_M$ by the formulas

$$c_{2n} = \left(\frac{q_2 - q_1}{\|q_2 - q_1\|} - q^N_{1,0}\right) \cdot e^{-jn\frac{2\pi}{T}\left(\|q_2 - q_0\| + q^N_{1,0}\right)} \tag{30}$$

As previously, by the time R_0's turn is reached, all the c_n's will have their correct values by the formulas

$$c_n = \frac{T}{(2\pi n)^2}\left(c_{N-1,n} + \frac{q_0 - q_{N-1}}{\|q_0 - q_{N-1}\|} - q^N_{N-1,N-2}\right) \cdot e^{-jn\frac{2\pi}{T}\left(\|q_0 - q_{N-1}\| + q^N_{N-1,N-2}\right)} \tag{31}$$

Now the values of $\mathcal{F}_{-M}, \ldots, \mathcal{F}_M$ can be easily computed and transmitted to the formation controller. The partial values $c_{2,n}, \ldots, c_{N-1,n}$ are among the items that have to be passed along to consecutive robots. Figure 5 shows the flow of information. The flow in the interior of the contour corresponds to the first round and the one on the outside corresponds to the final round.

6. *Active Contours*

Elastically deformable contour models (or snakes) were originally proposed by machine vision researchers [11] and used principally for boundary finding (segmentation) and motion tracking. The basic idea is to use the analogy of a rubber band fitting itself to arbitrary shapes. The elastic band evolves according to laws of linear elasticity and as a response to two kinds of forces. The elastic (internal) forces oppose stretching (expansion) and bending, and the environment (external) forces are normally gradients of a certain potential field guiding the snake to boundaries of features. In this paper, we use the model for two purposes. One is to maintain the smoothness of the closed curve at all times and the other one is to keep the robots together (so that we don't need separate attractive forces between the robots). In [10], an application of snakes to collective adaptation to environmental boundaries (or level sets) is described but we are using snakes primarily for formation control rather than tracking boundaries, although the latter can also be easily incorporated into our framework as well. In this section, we give a brief overview of these models, together with some of their properties.

A curve can be defined by the parametric equation $q(s, t) = (x(s, t), y(s, t))$ where $0 \le s \le 1$ is the arc-length parameter, $t \ge 0$ denotes time, x and y are the coordinates of each point on the curve. One way of deriving snake models is to use the energy formulation. The contour is influenced by a potential energy $P_{\mathcal{F}}(q(s, t))$ and a deformable energy $\mathcal{E}_{\mathcal{D}}(q(s, t))$. The potential energy pulls the points of the curve towards desired locations. In vision applications, this field is defined as a function of image coordinates, whereas in our case, $P_{\mathcal{F}}$ attracts the curve towards a desired curve described by a set of Fourier descriptors. Thus, we can define

$$P_{\mathcal{F}}(q(s, t)) = \int_0^1 \left\| q(s) - \sum_{n=-M}^{M} \mathcal{F}_n e^{jn\omega s} \right\| ds \tag{32}$$

The deformation energy $\mathcal{E}_{\mathcal{D}}(q(s, t))$ is defined as

$$\frac{1}{2}\int_0^1 \left(\alpha(s) \left| \frac{\partial q(s, t)}{\partial s} \right|^2 + \beta(s) \left| \frac{\partial^2 q(s, t)}{\partial s^2} \right|^2 \right) ds \tag{33}$$

where $\alpha(s)$ and $\beta(s)$ are tension (resistance to stretching) and rigidity (resistance to bending) parameters, respectively, associated with the elastic band. The kinetic energy is defined as

$$\mathcal{E}_K(q(s)) = \frac{1}{2}\int_0^1 \mu(s) \left| \frac{\partial^2 q(s, t)}{\partial t^2} \right|^2 ds \tag{34}$$

and damping energy by

$$\mathcal{D}(q(s)) = \frac{1}{2}\int_0^1 \lambda(s) \left| \frac{\partial q(s, t)}{\partial t} \right|^2 ds \tag{35}$$

where $\mu(s)$ and $\lambda(s)$ are the mass and damping coefficient, respectively. The whole energy is equal to

$$\begin{aligned}\mathcal{E}(q(s,t)) &= \mathcal{E}_{\mathcal{D}}(q(s,t)) + \mathcal{E}_K(q(s)) \\ &\quad + \mathcal{D}(q(s)) - P_{\mathcal{F}}(q(s,t))\end{aligned} \tag{36}$$

and the snake deforms in such a way as to minimize it. The complete Euler-Lagrange differential equation of the system is then

$$\begin{aligned}&\mu(s)\frac{\partial^2 q}{\partial t^2} + \lambda(s)\frac{\partial q}{\partial t} - \frac{\partial}{\partial s}\left(\alpha(s)\frac{\partial q}{\partial s}\right) + \frac{\partial^2}{\partial s^2}\left(\beta(s)\frac{\partial^2 q}{\partial s^2}\right) \\ &= -\nabla P_{\mathcal{F}}(q)\end{aligned} \tag{37}$$

where $\nabla P_{\mathcal{F}}(q) = q(s) - q_d(s)$.

The curve is actually represented by the set of discrete points $q_0 = v(s_0), \ldots, q_{N-1} = q(s_{N-1})$, where $s_i = i/N$, N being the number of samples around the contour. It is temporally evolved using finite differences to approximate $\partial^2 q/\partial s^2$ and $\partial^4 q/\partial s^4$, and forward differences to approximate $\partial q/\partial t$.

7. Curve Evolution

Suppose that $C^I(\mathcal{F}^I_{-M}, \ldots, \mathcal{F}^I_M)$ is the curve defined by the initial positions of robots and $C^F(\mathcal{F}^F_{-M}, \ldots, \mathcal{F}^F_M)$ denotes a final desired curve for C^I to evolve into. In section 4, every robot R_i with position $q_{C^I}(s_i)$ followed a straight line to $q_{C^F}(s_i)$. Suppose, for the moment, that the centres of gravity of C^I and C^F are co-located. Depending on the geometry of the curves, their relative orientation, and the positions of the starting points $s_{0_{C^I}}$ and $s_{0_{C^F}}$, this simple scheme can potentially lead to collisions between robots. This means that C^I may cross itself at one or more points while evolving. It can be likened to twisting a generalized cylinder (with cross section equal to the current formation shape on one side and the target shape on the other) along its main axis. To let the formation proceed to the target formation from the current state will clearly make the contour fold unto itself and the sum of snake and repulsion forces will certainly prevent the formation from convergence. It is observed that this *torsion* effect can be alleviated (or altogether eliminated, depending on case) if the starting position of one of the contours is shifted to either direction for a sufficient amount. Fortunately, torsion can be quantified and a method for decreasing it can be devised. Suppose, at the moment, that the robot collection is homogeneous (the relative positions of robots in the formation do not matter).

To quantify, we consider the intersection of the paths of two robots R_{s_i} (going from $q_{C^I}(s_i) = q(s_i)$ to $q_{C^F}(s_i) = q_d(s_i)$) and R_{s_j} (going from $q_{C^I}(s_j) = q(s_j)$ to $q_{C^F}(s_j) = q_d(s_j)$). The point of intersection of the lines defined by the vectors $v_i = q_d(s_i) - q(s_i)$ and $v_j = q_d(s_j) - q(s_j)$ (denoted by Q_{s_i,s_j}) is given by

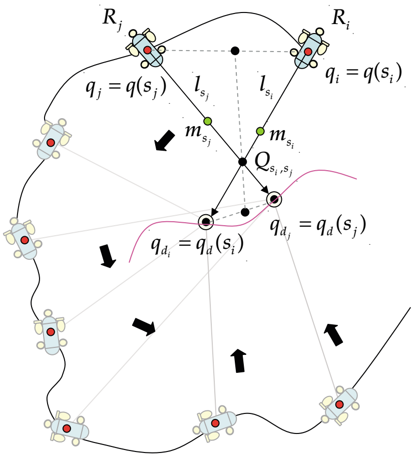

figure 6: Calculating torsion and its elimination

$$\begin{aligned}x_{Q_{s_i,s_j}} &= (1/D_{s_i,s_j}) \\ &\cdot((x_{q(s_j)} - x_{q_d(s_j)})(x_{q(s_i)}y_{q_d(s_i)} - y_{d(s_i)}x_{q_d(s_i)}) \\ &-(x_{q(s_i)} - x_{q_d(s_i)})(x_{q(s_j)}y_{q_d(s_j)} - y_{d(s_j)}x_{q_d(s_j)}))\end{aligned} \tag{38}$$

$$\begin{aligned}y_{Q_{s_i,s_j}} &= (1/D_{s_i,s_j}) \\ &\cdot((y_{q(s_j)} - y_{q_d(s_j)})(x_{q(s_i)}y_{q_d(s_i)} - y_{d(s_i)}x_{q_d(s_i)}) \\ &-(y_{q(s_i)} - y_{q_d(s_i)})(x_{q(s_j)}y_{q_d(s_j)} - y_{d(s_j)}x_{q_d(s_j)}))\end{aligned} \tag{39}$$

where

$$\begin{aligned}D_{s_i,s_j} &= Det(q(s_i), q_d(s_i), q(s_j), q_d(s_j)) \\ &= (x_{q(s_i)} - x_{q_d(s_i)})(y_{q(s_j)} - y_{q_d(s_j)}) \\ &\quad -(y_{q(s_i)} - y_{q_d(s_i)})(x_{q(s_j)} - x_{q_d(s_j)})\end{aligned} \tag{40}$$

Now, define

$$m_{s_i} = \frac{(q_d(s_i) - q(s_i))}{2} \;;\; m_{s_j} = \frac{(q_d(s_j) - q(s_j))}{2} \tag{41}$$

$$l_{s_i} = \frac{\|q_d(s_i) - q(s_i)\|}{2} \;;\; l_{s_j} = \frac{\|q_d(s_j) - q(s_j)\|}{2} \tag{42}$$

$$U_{s_i} = Q_{s_i,s_j} - m_{s_i} \quad ;\; U_{s_j} = Q_{s_i,s_j} - m_{s_j} \tag{43}$$

Necessary conditions for no intersection are $\|U_{s_i}\| > l_{s_i}$ and $\|U_{s_j}\| > l_{s_j}$ or

$$\frac{\|Q_{s_i,s_j} - (q_d(s_i) - q(s_i))/2\|}{\|q_d(s_i) - q(s_i)\|} > \frac{1}{2} \text{ and} \tag{44}$$

$$\frac{\|Q_{s_i,s_j} - (q_d(s_j) - q(s_j))/2\|}{\|q_d(s_j) - q(s_j)\|} > \frac{1}{2} \tag{45}$$

Therefore, the quantity

$$\begin{aligned}\Phi_{C^I,C^F} &= \int_0^1\!\!\int_0^1\left(\frac{\|Q_{s_i,s_j} - (q_d(s_i) - q(s_i))/2\|}{\|q_d(s_i) - q(s_i)\|}\right. \\ &\quad \left.+ \frac{\|Q_{s_i,s_j} - (q_d(s_j) - q(s_j))/2\|}{\|q_d(s_j) - q(s_j)\|}\right)ds_i ds_j\end{aligned} \tag{46}$$

is a measure of torsion. The greater the Φ_{C^I,C^F}, the less the torsion would be.

To maximize Φ_{C^I, C^F}, we have to reorder the robots' id's. This is made possible through re-parametrization of C^I (or, alternatively, C^F), or homogeneous robots. This amounts to shifting the id's, in either direction, until Φ_{C^I, C^F} is maximum. If $\tau \in [0, 1]$ denotes the amount of shift, then $q_{C^I}(s)$ will go to $q_{C^F}(s - \tau)$. The optimal $\tau_{optimal}$ is, therefore, the solution to the optimization problem

$$max_{\tau \in [0, 1]} \Phi_{C^I, C^F}(\tau) \tag{47}$$

where $\Phi_{C^I, C^F}(\tau) =$

$$\int_0^1 \int_0^1 \left(\frac{\| Q_{s_i, s_j}(\tau) - (q_d(s_i - \tau) - q(s_i))/2 \|}{\| q_d(s_i - \tau) - q(s_i) \|} + \frac{\| Q_{s_i, s_j}(\tau) - (q_d(s_j - \tau) - q(s_j))/2 \|}{\| q_d(s_j - \tau) - q(s_j) \|} \right) ds_i ds_j \tag{48}$$

$$\begin{aligned} x_{Q_{s_i, s_j}}(\tau) &= (1/D_{s_i, s_j}(\tau)) \\ &\cdot ((x_{q(s_j)} - x_{q_d(s_j - \tau)})(x_{q(s_i)} y_{q_d(s_i - \tau)} - y_{d(s_i)} x_{q_d(s_i - \tau)}) \\ &- (x_{q(s_i)} - x_{q_d(s_i - \tau)})(x_{q(s_j)} y_{q_d(s_j - \tau)} - y_{d(s_j)} x_{q_d(s_j - \tau)})) \end{aligned} \tag{49}$$

$$\begin{aligned} y_{Q_{s_i, s_j}}(\tau) &= (1/D_{s_i, s_j}(\tau)) \\ &\cdot ((y_{q(s_j)} - y_{q_d(s_j - \tau)})(x_{q(s_i)} y_{q_d(s_i - \tau)} - y_{d(s_i)} x_{q_d(s_i - \tau)}) \\ &- (y_{q(s_i)} - y_{q_d(s_i - \tau)})(x_{q(s_j)} y_{q_d(s_j - \tau)} - y_{d(s_j)} x_{q_d(s_j - \tau)})) \end{aligned} \tag{50}$$

$$\begin{aligned} &D_{s_i, s_j}(\tau) \\ &= Det(q(s_i), q_d(s_i - \tau), q(s_j), q_d(s_j - \tau)) \\ &= (x_{q(s_i)} - x_{q_d(s_i - \tau)})(y_{q(s_j)} - y_{q_d(s_j - \tau)}) \\ &\quad - (y_{q(s_i)} - y_{q_d(s_i - \tau)})(x_{q(s_j)} - x_{q_d(s_j - \tau)}) \end{aligned} \tag{51}$$

It turns out that we don't have to examine every point on the curve with every other point. The contribution of every pair of two neighbouring robots R_i and R_{i+1} to the cost function is sufficient. This, combined with the fact that a discrete version of $\Phi_{C^I, C^F}(\tau)$ will be used in practice, leads to the following maximization problem:

$max_{\tau \in \{0, \ldots, N-1\}} \Phi_{C^I, C^F}(\tau)^d =$

$$\sum_{i=0}^{N-1} \left(\frac{\| Q_i(\tau) - (q_{d_{i-\tau}} - q_i)/2 \|}{\| q_{d_{i-\tau}} - q_i \|} + \frac{\| Q_i(\tau) - (q_{d_{i+1-\tau}} - q_{i+1})/2 \|}{\| q_{d_{i+1-\tau}} - q_{i+1} \|} \right) \tag{52}$$

$$\begin{aligned} x_{Q_i(\tau)} &= (1/D_i^d(\tau)) \\ &\cdot ((x_{q_{i+1}} - x_{q_{d_{i+1-\tau}}})(x_{q_i} y_{q_{d_{i-\tau}}} - y_{q_i} x_{q_{d_{i-\tau}}}) \\ &- (x_{q_i} - x_{q_{d_{i-\tau}}})(x_{q_{i+1}} y_{q_{d_{i+1-\tau}}} - y_{q_{i+1}} x_{q_{d_{i+1-\tau}}})) \end{aligned} \tag{53}$$

$$\begin{aligned} y_{Q_i(\tau)} &= (1/D_i^d(\tau)) \\ &\cdot ((y_{q_{i+1}} - y_{q_{d_{i+1-\tau}}})(x_{q_i} y_{q_{d_{i-\tau}}} - y_{q_i} x_{q_{d_{i-\tau}}}) \\ &- (y_{q_i} - y_{q_{d_{i-\tau}}})(x_{q_{i+1}} y_{q_{d_{i+1-\tau}}} - y_{q_{i+1}} x_{q_{d_{i+1-\tau}}})) \end{aligned} \tag{54}$$

$$\begin{aligned} &D_i^d(\tau) \\ &= Det^d(q_i, q_{d_{i-\tau}}, q_{i-1}, q_{d_{i+1-\tau}}) \\ &= (x_{q_i} - x_{q_{d_{i-\tau}}})(y_{q_{i+1}} - y_{q_{d_{i+1-\tau}}}) \\ &\quad - (y_{q_i} - y_{q_{d_{i-\tau}}})(x_{q_{i+1}} - x_{q_{d_{i+1-\tau}}}) \end{aligned} \tag{55}$$

If the lines are parallel, the D_{s_i, s_j} would be zero and Q_{s_i, s_j} is not defined any more. To get around this problem and make the computations more efficient, at the same time, rather than the above maximization problem, we consider the minimization problem $min_{\tau \in \{0, \ldots, N-1\}} \tilde{\Phi}_{C^I, C^F}(\tau)$, where

$$\tilde{\Phi}_{C^I, C^F}(\tau) = \sum_{i=0}^{N-1} D_i^d(\tau) \tag{56}$$

Denote the optimal τ by τ_{C^I, C^F}. At the beginning of motion, all the robots simultaneously calculate the *same* optimal τ and then proceed. For inhomogeneous robots (where the positions of individual robots in the formation matters), the re-parametrization process is implemented by actually moving the robots around the perimeter of the curve, i.e., they have to follow the tangent vectors to the points on the contour. The applied force is

$$f_T = \frac{\nabla C^I(q)}{\| \nabla C^I(q) \|} \tag{57}$$

which can be approximated by

$$f_T = \frac{q_{i+1} - q_i}{\| q_{i+1} - q_i \|} \tag{58}$$

Using this method, the formation will eventually drift away, especially when there are high curvature points on the curve. A better and simpler way is to shift the desired locations q_{d_i} around the contour, i.e., to lead R_i to $q_{d_{i+1}}$ or $q_{d_{i-1}}$, depending on direction.

It should be noted that the forgoing scheme works only when the direction of increase of the parameter s is the same for both C^I and C^F. If this is not the case, then no amount of shifting can prevent the evolving curve from intersecting itself. Throughout this section, we implicitly assumed that the robots know the Fourier descriptors for the initial configuration, i.e., $\mathcal{F}^I_{-M}, \ldots, \mathcal{F}^I_M$ are known to each robot R_i. Furthermore, we also made the assumption that robot R_i is located at $q_{C^I}(s_i)$. Based on these assumptions, R_i needs only to calculate the optimal τ and proceed to

$$q_{C^F}(s_i - \tau_{C^I, C^F}) = q_{C^F}(s_{i - \tau_{C^I, C^F}}). \tag{59}$$

In practice, when a motion command is issued, information about the initial positions of robots is not available. This means that the robots have to go through a *configuration identification* phase first to compute $\mathcal{F}^I_{-M}, \ldots, \mathcal{F}^I_M$. We assume that the ordering and positions of robots are such that C^I is a *simple curve*.

Even when C^I has been computed, a re-parametrization is required to place R_i at $q_{C^I}(s_i)$. The formula $(N/(2\pi))(\varphi_2 - \varphi_1)_{mod 2\pi}$ gives an approximation to the amount of shift but this is not a reliable measure. The correct amount of shift τ is that which places R_0 at $q(s_0)$.

The id shifting process is initiated by R_0. When R_0 reaches the closest point to $q(s_0) = q_0$, the process stops and the robots start moving. After reaching their goals, the position of R_i, i.e. q_i, would be $q(s_i)$. For this process to succeed, we should make an assumption. Define the neighbourhood of a point $q(\tilde{s})$ as

$$\mathcal{N}_{\tilde{s}} = \{q(s) | \|q(\tilde{s}) - q(s)\|_C < \delta_{\tilde{s}}\} \tag{60}$$

where $\|q(s_1) - q(s_2)\|_C$ means the distance between $q(s_1)$ and $q(s_2)$ travelled over the curve C, which is equal to $|s_2 - s_1|$. We assume that for a given neighbourhood size δ_s, for every $s_1, s_2 \in [0, 1]$, $\|q(s_1) - q(s_2)\|_C < \delta_{s_1}$ implies that $q(s_2) \in \mathcal{N}_{s_1}$ and $q(s_1) \in \mathcal{N}_{s_2}$. This means that no two points of the curve come closer than δ_s to each other. Here, we set $\delta_s = \delta_{min}$. In the discrete case, δ_{min} is equal to the maximum allowable distance between two robots. If $\|q(\tilde{s}_0) - q(s_0)\|_C < \delta_{min}$, where $q(\tilde{s}_0)$ is the actual position of R_i on the curve, then R_0 will directly move to $q(s_0)$ which is not desirable in some anomalous situations. We should also assume that the uncertainty in the position of a robot does not exceed δ_{min}: $\|q_i - q(\tilde{s}_i)\|_C \le \delta_{min}$.

8. Step-Wise Deformation

Even though the procedure outlined above determines the best possible configuration for evolving C^I into C^F, it does not guarantee that there would be no intersections at all. The reason lies in the fact that the energy difference between C^I into C^F may be very substantial. Very much like the discussion in section 6, we can define the *internal energy* of a curve C by

$$\mathcal{E}_C = \frac{1}{2}\int_0^1 \left(\left|\frac{\partial q_C(s)}{\partial s}\right|^2 + \left|\frac{\partial^2 q_C(s)}{\partial s^2}\right|^2 \right) ds \tag{61}$$

and the *energy difference* between C^I and C^F by

$$d\mathcal{E}_{C^I, C^F} = \frac{1}{2}\int_0^1 \left(\left|\frac{\partial q_{C^F}(s)}{\partial s} - \frac{\partial q_{C^I}(s)}{\partial s}\right|^2 + \left|\frac{\partial^2 q_{C^F}(s)}{\partial s^2} - \frac{\partial^2 q_{C^I}(s)}{\partial s^2}\right|^2 \right) ds \tag{62}$$

We *argue* that the greater the value of $d\mathcal{E}_{C^I, C^F}$, the less the value of $\Phi_{C^I, C^F}(\hat{\tau}_{C^I, C^F})$ (i.e., greater the amount of torsion) would be. Thus, a safer method for deformation would be to successively evolve C^I into curves with lesser energy (to go downhill) until the energy reaches a minimum and then climb up the energy hill corresponding to C^F.

Fortunately, Fourier descriptors directly provide us with an automatic method for doing this. Suppose C is described by $\mathcal{F}^C_{-M}, \ldots, \mathcal{F}^C_M$ and let C_L denote the curve reconstructed using the first L descriptors, i.e.

$$q_{C_L}(s) = \sum_{n=-L}^{L} \mathcal{F}^C_n e^{jn\omega s} \tag{63}$$

It is our *conjecture* that $\mathcal{E}_{C_L} \ge \mathcal{E}_{C_{L-1}}$. We propose that, instead of directly deforming C^I into C^F, we follow the path

$$C^I_M \to \ldots \to C^I_1 \Rightarrow C^F_1 \to \ldots \to C^F_M \tag{64}$$

Now, rename this path as

$$\tilde{C}_{(1)} \to \ldots \to \tilde{C}_{(M)} \Rightarrow \tilde{C}_{(M+1)} \to \ldots \to \tilde{C}_{(2M)} \tag{65}$$

If the above conjecture is true, then for every $L_1, L_2, P_1, P_2 \in \{1, \ldots, 2M\}$, with $P_1 \le L_1 < L_2 \le P_2$, we will have that

$$d\mathcal{E}_{\tilde{C}_{(L_1)}, \tilde{C}_{(L_2)}} \le d\mathcal{E}_{\tilde{C}_{(P_1)}, \tilde{C}_{(P_2)}}, \text{ and that} \tag{66}$$

$$\Phi_{\tilde{C}_{(L_1)}, \tilde{C}_{(L_2)}}(\hat{\tau}_{\tilde{C}_{(L_1)}, \tilde{C}_{(L_2)}}) \ge \Phi_{\tilde{C}_{(P_1)}, \tilde{C}_{(P_2)}}(\hat{\tau}_{\tilde{C}_{(P_1)}, \tilde{C}_{(P_2)}}) \tag{67}$$

or, alternatively,

$$\tilde{\Phi}_{\tilde{C}_{(L_1)}, \tilde{C}_{(L_2)}}(\hat{\tau}_{\tilde{C}_{(L_1)}, \tilde{C}_{(L_2)}}) \le \tilde{\Phi}_{\tilde{C}_{(P_1)}, \tilde{C}_{(P_2)}}(\hat{\tau}_{\tilde{C}_{(P_1)}, \tilde{C}_{(P_2)}}) \tag{68}$$

This is intuitively appealing and simulations support it. It should be noted that it is not at all required to go through all the nodes in the path; many of the intermediate deformations are actually redundant. Suppose that C^I ($\tilde{C}_{(1)}$) has evolved into $\tilde{C}_{(R)}$. If

$$\begin{aligned} &\Phi_{\tilde{C}_{(R)}, \tilde{C}_{(R+2)}}(\hat{\tau}_{\tilde{C}_{(R)}, \tilde{C}_{(R+2)}}) \\ &\ge \Phi_{\tilde{C}_{(R)}, \tilde{C}_{(R+1)}}(\hat{\tau}_{\tilde{C}_{(R)}, \tilde{C}_{(R+1)}}) \end{aligned} \tag{69}$$

(the amount of torsion does not increase), then there is no point in evolving $\tilde{C}_{(R)}$ into $\tilde{C}_{(R+1)}$; in such a case, $\tilde{C}_{(R)}$ can directly proceed to $\tilde{C}_{(R+2)}$. In general, the next curve to deform into is $\tilde{C}_{(R+\kappa)}$ such that

$$\begin{aligned} &\Phi_{\tilde{C}_{(R)}, \tilde{C}_{(R+\kappa)}}(\hat{\tau}_{\tilde{C}_{(R)}, \tilde{C}_{(R+\kappa)}}) \\ &\le \Phi_{\tilde{C}_{(R)}, \tilde{C}_{(R+1)}}(\hat{\tau}_{\tilde{C}_{(R)}, \tilde{C}_{(R+1)}}) \end{aligned} \tag{70}$$

It should be noted that due to discretization, it may even be the case that

$$\begin{aligned} &\Phi_{\tilde{C}_{(R)}, \tilde{C}_{(R+\kappa)}}(\hat{\tau}_{\tilde{C}_{(R)}, \tilde{C}_{(R+\kappa)}}) \\ &< \Phi_{\tilde{C}_{(R)}, \tilde{C}_{(R+1)}}(\hat{\tau}_{\tilde{C}_{(R)}, \tilde{C}_{(R+1)}}) \end{aligned} \tag{71}$$

C^F incorporates any desired rotation or scaling but no translation. Translation is decoupled from curve evolution and can be done at any time before, during, or after the curve has been evolved.

9. Behaviour-based Motion Algorithm

In this section, we propose a simple behaviour-based algorithm for guiding robots to their destinations. $C^I(\mathcal{F}^I_{-M}, \ldots, \mathcal{F}^I_M)$ and $C^F(\mathcal{F}^F_{-M}, \ldots, \mathcal{F}^F_M)$ are assumed to be described by $q(s)$ and $q_d(s)$, respectively. Although the robots can follow a straight line towards

q_d, a better way is to adopt ideas from curve evolution literature where evolution is conducted in a different coordinate system. When a curve is evolving, the motion of points attached to the boundary of the curve can be characterized by motion along the tangent and normal vectors to the curve at that point:

$$\frac{dq(s,t)}{dt} = \dot{q}_{\vec{N}}(t)\vec{N} + \dot{q}_{\vec{T}}(t)\vec{T} \tag{72}$$

where $\dot{q}_{\vec{N}}$ and $\dot{q}_{\vec{T}}$ denote speeds along $\vec{N}$ (normal) and $\vec{T}$ (tangent), respectively. Motion along $\vec{T}$ does not change the geometry of the curve; it only affects a re-parametrization of the curve. By defining time-changing rules for $\vec{N}$ and $\vec{T}$, the curve can be evolved. In most of the curve evolution literature, motion along $\vec{T}$ is ignored. A proper $\dot{q}_{\vec{N}}$ is sought to suit the particular problem. Intuitively, the best case is when the desired goal $q_d(s)$ is situated on the normal $\vec{N}$ to the curve at $q(s)$. In this case, R_s can directly proceed to $q_d(s)$. If this is not the case, a proper strategy would be to let R_s drive along $\vec{T}(s)$ at $q(s)$ until $q_d(s)$ falls on $\vec{N}(s)$. In this mode (mode 1), the equations of motion are

$$\dot{q}(t) = \dot{q}^{(1)}_{\vec{T}}\vec{T};\ \dot{q}^{(1)}_{\vec{T}} = \dot{\beta}_1\zeta_{\vec{T}}(\vec{N});\ \dot{\beta}_1 = \rho \tag{73}$$

where the tangent is defined by

$$\vec{T} = \frac{q(s)}{\|q(s)\|} = \frac{(x_s, y_s)}{\sqrt{x_s^2 + y_s^2}} = \frac{q(s_{i+1}) - q(s_{i-1})}{\|q(s_{i+1}) - q(s_{i-1})\|} \tag{74}$$

and

$$\begin{aligned}\zeta_{\vec{T}}(\vec{N}) &= \operatorname{sgn}((R_z(\pi/2)(q(s_{i+1}) - q(s_{i-1}))\\ &\qquad \times R_z(\pi/2)(q(s_{i-1}) - q_d(s_i)))\\ &\quad /\|(R_z(\pi/2)(q(s_{i+1}) - q(s_{i-1})))\|)\\ &= \begin{cases} 1 & \text{if } q_d(s) \text{ is on the left of } \vec{N}\\ -1 & \text{if } q_d(s) \text{ is on the right of } \vec{N}\end{cases}\end{aligned} \tag{75}$$

The robots remain in this mode until $\vartheta_i \cong 0$ where ϑ_i is the angle between $\vec{N}$ and $q_d(s_i) - q(s_i)$ and is computed as

$$\begin{aligned}\vartheta_i = min(&\operatorname{acos}(((q_i - q_{d_i}) \cdot \vec{N})/\|q_i - q_{d_i}\|),\\ &\operatorname{acos}((-(q_i - q_{d_i}) \cdot \vec{N})/\|q_i - q_{d_i}\|))\end{aligned} \tag{76}$$

When ϑ_i is approximately zero, the robot can proceed along $\vec{N}$ using the following equation (making up mode 2):

$$\dot{q}(t) = \dot{q}^{(2)}_{\vec{N}}\vec{N};\ \dot{q}^{(2)}_{\vec{N}} = \dot{\alpha}_2\zeta_{\vec{N}}(\vec{T});\ \dot{\alpha}_2 = \rho \tag{77}$$

where the normal is defined by

$$\vec{N} = (-y_s, x_s)/(\sqrt{x_s^2 + y_s^2}) = R_z(\pi/2)\vec{T} \tag{78}$$

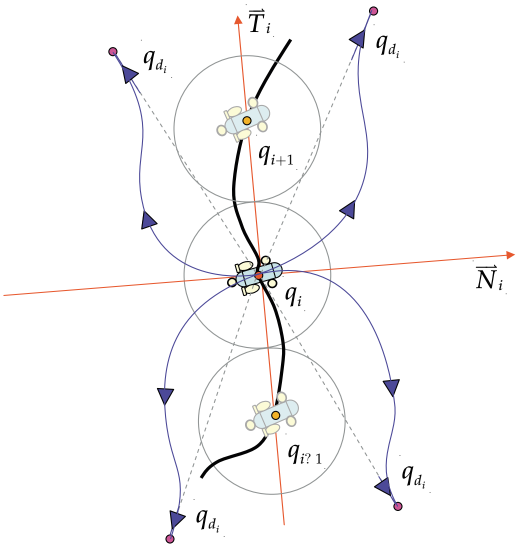

figure 7: Behaviour-based motion strategy

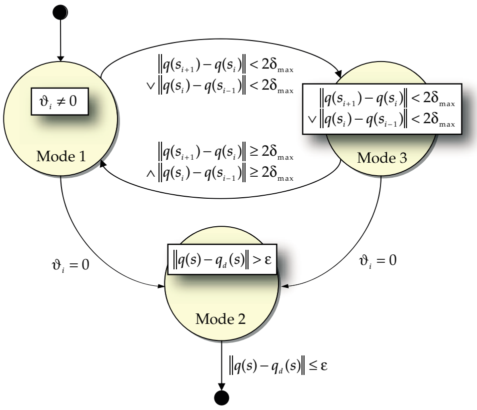
$\vartheta_i \neq 0$
Mode 1
$\|q(s_{i+1}) - q(s_i)\| < 2\delta_{max}$
$\vee \|q(s_i) - q(s_{i-1})\| < 2\delta_{max}$
$\|q(s_{i+1}) - q(s_i)\| < 2\delta_{max}$
$\vee \|q(s_i) - q(s_{i-1})\| < 2\delta_{max}$
Mode 3
$\|q(s_{i+1}) - q(s_i)\| \geq 2\delta_{max}$
$\wedge \|q(s_i) - q(s_{i-1})\| \geq 2\delta_{max}$
$\vartheta_i = 0$
$\|q(s) - q_d(s)\| > \varepsilon$
$\vartheta_i = 0$
Mode 2
$\|q(s) - q_d(s)\| \leq \varepsilon$

figure 8: Hybrid automaton for the motion algorithm

and

$$\begin{aligned}\zeta_{\vec{T}}(\vec{T}) &= -\operatorname{sgn}(((q(s_{i+1}) - q(s_{i-1}))\\ &\qquad \times (q(s_{i-1}) - q_d(s_i)))\\ &\quad /\|(q(s_{i+1}) - q(s_{i-1}))\|)\\ &= \begin{cases} 1 & \text{if } q_d(s) \text{ is on the right of } \vec{T}\\ -1 & \text{if } q_d(s) \text{ is on the left of } \vec{T}\end{cases}\end{aligned} \tag{79}$$

The robot remains in this mode until
$\|q(s) - q_d(s)\| \leq \varepsilon$, at which time the robot stops.

In the two previous cases, we assumed that the robot can continue along $\vec{T}$ without interruption. In some cases, though, the robot's path along $\vec{T}$ may be obstructed by R_{i+1} or R_{i-1}, depending on direction of motion. Referring to figure 7, when R_i closer than $2\delta_{min}$ to the adjacent robot, it should conduct a rolling motion past it. In this mode (mode 3), the equations of motion would be:

$$\dot{q}(t) = \dot{q}^{(3)}_{\vec{N}}\vec{N} + \dot{q}^{(3)}_{\vec{T}}\vec{T} \tag{80}$$

where

$$\dot{q}^{(3)}_{\vec{N}} = \dot{\alpha}_3\zeta_{\vec{N}}(\vec{T})\,;\ \dot{q}^{(3)}_{\vec{T}} = \dot{\beta}_3\zeta_{\vec{T}}(\vec{N}) \tag{81}$$

$$\dot{\alpha}_3 = 2\delta_{min}\sin(\rho/(2\delta_{min})) \tag{82}$$

$$\dot{\beta}_3 = 2\delta_{min}(1-\cos(\rho/(2\delta_{min}))) \tag{83}$$

Figure 8 shows the *hybrid automaton* governing mode changes.

We can use sigmoidal functions to combine the individual behaviours in each mode [12]. This will lead to a single differential equation, producing continuous dynamics and smooth trajectories. It also helps us ignore the problem of when a behaviour should be deactivated and another one started. At each particular instant, two adjacent behaviours can be active at the same time but to different degrees. It should be noted that this will not affect the global behaviour of the robots at all because the switching operation acts locally, in a small neighbourhood of *equilibrium points*. Thus, the equations for the motion described in this section will be:

$$\lambda\frac{\partial q_i(t)}{\partial t} = (1-\sigma_1\chi)\dot{q}_{KS} + \sigma_1\chi\dot{q}_i^{1} \tag{84}$$

$$\dot{q}_i^{1} = (1-\sigma_2)\dot{q}_{KS} + \sigma_2\dot{q}_i^{2} \tag{85}$$

$$\dot{q}_i^{2} = (1-\sigma_3)\dot{q}_i^{M_3} + \sigma_3\dot{q}_i^{M_1} \tag{86}$$

$$\dot{q}_i^{M_1} = (1-\sigma_4)\dot{q}_i^{M_2} + \sigma_4\dot{q}_i^{M'_1} \tag{87}$$

$$\dot{q}_i^{M'_1} = (1-\sigma_3)(\dot{q}^{(3)}_{\vec{T}} + \dot{q}_{snake}) + \sigma_3\dot{q}_i^{M_3} \tag{88}$$

$$\dot{q}_i^{M_3} = (1-\sigma_4)\dot{q}_i^{M_2} + \sigma_4\dot{q}_i^{M'_3} \tag{89}$$

$$\dot{q}_i^{M'_3} = \sigma_3(\dot{q}^{(3)}_{\vec{T}} + \dot{q}^{(3)}_{\vec{N}} + \dot{q}_{snake}) + (1-\sigma_3)\dot{q}_i^{M_1} \tag{90}$$

$$\dot{q}_i^{M_2} = (1-\sigma_1)\dot{q}_{KS} + \sigma_1\dot{q}_i^{M'_2} \tag{91}$$

$$\dot{q}_i^{M'_2} = \sigma_4(\dot{q}^{(2)}_{\vec{N}} + \dot{q}_{snake}) + (1-\sigma_4)\dot{q}_i^{M_1} \tag{92}$$

where

$$\sigma_1 = \sigma_{\varepsilon_G,\beta_G}(\|q_i(t) - q_{d_i}(t)\|) \tag{93}$$

$$\sigma_2 = \sigma_{\varepsilon_W,\beta_W}\left(\frac{1}{W}\sum_{\bar{t}=t-W\Delta t}^{t} q_i(\bar{t})\right) \tag{94}$$

$$\sigma_3 = \sigma_{2\delta_{min},\beta_{M_1M_3}}(\|q_{i+1}(t) - q_i(t)\|) \tag{95}$$

$$\sigma_4 = \sigma_{\varepsilon_\theta,\beta_\theta}(\|\theta_i(t)\|) \tag{96}$$

and

$$\sigma_{\alpha,\beta}(x) = 1/(1+e^{-4\beta(x-\alpha)}) \tag{97}$$

In the sigmoidal function, α is the switching point and β determines the slope of switching (how fast the switching should occur). $\dot{q}_{KS}$ stands for station keeping behaviour and the snake force is defined as:

$$\dot{q}_{snake} = \alpha\frac{\partial^2 q_i}{\partial s^2} - \alpha\beta\frac{\partial^4 q_i}{\partial s^4} \tag{98}$$

ε_G indicates how close the robot should be to the goal before it should attempt to stop. As mentioned in section 6, when snake forces are present, the resulting curve is a smooth approximation to the desired curve, so that some of the robots may never reach their exact desired locations due to elastic constraints (this actually corresponds to a local minimum). To know when this situation has occurred, each robot uses a fixed window of size W. If the robot's position has not changed considerably (quantified by ε_W) during this time frame, then for all practical purposes, the robot can assume that it is stuck in a local minimum and there is no sense in continuing. The parameters β_G, β_W, $\beta_{M_1M_3}$, β_θ and ε_θ can be selected empirically.

10. Virtual System Architecture

Now that we have considered individual aspects, we can discuss the architecture of the whole system. In [2], the concept of a *virtual structure* is proposed for robot formations. This framework can also be extended to our situation. The structure we use is basically very similar to the ones discussed by [2][7]. In fact, it is a combination of the two with the addition of local interactions between individuals which is missing in the said references. Figure 9 shows our system architecture. R_i represents the i´th Serafina equipped with the local control system C_i. u_i is the actuator signal and y_i is the sensor readings (or outputs) of R_i. M represents the formation controller which can be a larger AUV/ROV or a designated Serafina. The formation controller implements a discrete event system (an automaton) G which determines the mode of operation of the whole formation. The formation controller broadcasts the formation

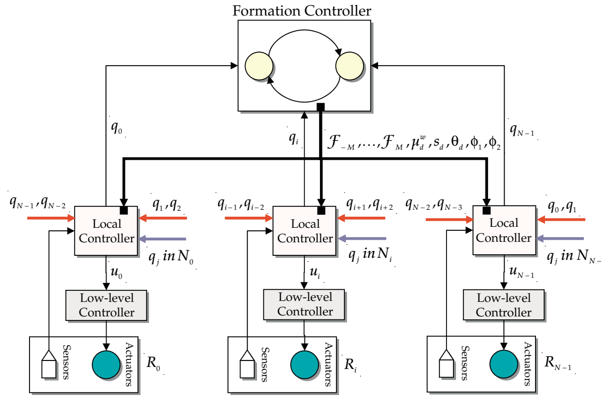

figure 9: System block diagram

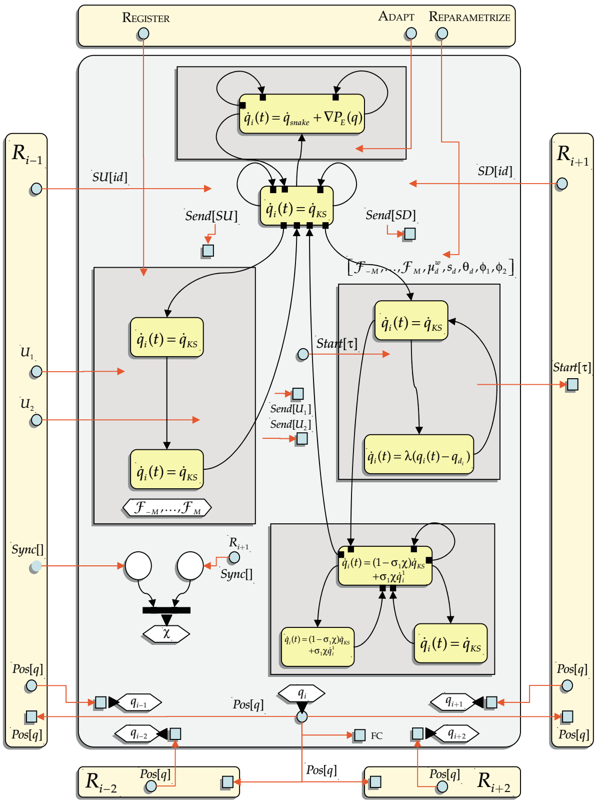
REGISTER
ADAPT
REPARAMETRIZE
R_{i-1}
R_{i+1}
R_{i-2}
R_{i+2}
$\dot{q}_i(t) = \dot{q}_{snake} + \nabla P_E(q)$
$\dot{q}_i(t) = \dot{q}_{KS}$
SU[id]
SD[id]
Send[SU]
Send[SD]
$[\mathcal{F}_{-M}, \ldots, \mathcal{F}_M, \mu^w_d, s_d, \theta_d, \phi_1, \phi_2]$
u_1
u_2
Start[τ]
Send[U₁]
Send[U₂]
$\dot{q}_i(t) = \lambda(q_i(t) - q_{d_i})$
$\mathcal{F}_{-M}, \ldots, \mathcal{F}_M$
Sync[]
R_{i+1}
Sync[]
χ
$\dot{q}_i(t) = (1-\sigma_i\chi)\dot{q}_{KS} + \sigma_i\chi\dot{q}^1_i$
Pos[q]
q_{i-1}
q_i
q_{i+1}
q_{i-2}
q_{i+2}
FC

figure 10: Detailed view of the local controller

variable ζ to designated robots along with mode of operation. The formation variable consists of shape parameters (Fourier descriptors $\mathcal{F}_{-M}, \ldots, \mathcal{F}_M$), geometric parameters (μ^w_d, s_d, θ_d, φ_1, and φ_2), and physical parameters (elasticity, damping, attraction and repulsion strengths). Three basic modes of operation are particularly important: (1) Enclose of or adaptation to an environmental level set through snake operation and some environmental potential, (2) Calculating the signature for an environmental feature (i.e., calculate the shape variables), and (3) Forming a particular shape using shape and geometrical variables. Depending on the mode of operation, all or a set of designated robots send relevant information (performance variable ς) to the controlling robot. It consists of the calculated current descriptors $\mathcal{F}_{-M}, \ldots, \mathcal{F}_M$, the centre of mass μ^w, the scale s, the orientation θ, and the current positions q_i. Using this feedback, formation performance can be measured by the master controller. Local interactions between the robots consist of those required for implementing repulsion and cohesion (snake) forces. The virtual structure's motion is characterized by the area centre of gravity (μ^w_x, μ^w_y) and the rotation θ around this point. Figure 10 shows a detailed view of the *local controller* implemented in each robot (the one for R_0 is slightly different). R_i maintains communication channels with R_{i-1}, R_{i-2}, R_{i+1}, R_{i+2}, and the formation controller. The three operation modes are triggered by *ADAPT*, *REGISTER*, and *RE-PARAMETRIZE*/ *EVOLVE* commands. The diagram uses notation inspired by state-charts, Petri nets, and communicating systems.

11. Simulation Results

To demonstrate the described strategy, we present a simple example. The arrangement in figure 11 (1), composed of $N = 102$ robots, is required to evolve into the final curve C^F shown in figure 11 (20) described by $\{\mathcal{F}^F_{-10}, \mathcal{F}^F_{10}\}$. C^F was descretized using $N = 82$ points. Here, $M = 10$, μ^w_d is equal to the centre of gravity of the initial formation μ_{C^I} (no translation), $\theta_d = 45$ (rotation with respect to C^F), $s_d = 1.1$ (scaling). The evolved curve is depicted superimposed on the non-rotated, non-scaled final curve. Figure 11 (1) depicts the initial re-parametrization phase, after which the robot's positions are those determined by the calculated descriptors $\{\mathcal{F}^I_{-10}, \mathcal{F}^I_{10}\}$. The amount of initial re-parametrization is $\tau_{init} = 6$. Figure 11 (2) through 11 (20) show the sequence of deformations of C^I. Solid circles indicate the current positions of the robots and hollow circles those of desired points (the next curve in sequence). The amount of $\hat{\tau}$ is shown in brackets. At step three, we have

$$\tilde{\Phi}_{C^I_{M-2}, C^I_{M-5}} > \tilde{\Phi}_{C^I_{M-2}, C^I_{M-3}} > \tilde{\Phi}_{C^I_{M-2}, C^I_{M-4}} \qquad (99)$$

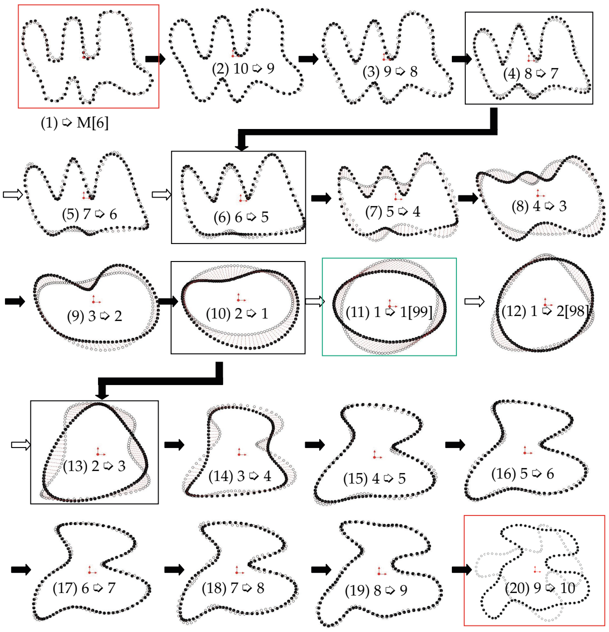

figure 11: Curve evolution in simulation

This means that C^I_{M-2} can directly deform into C^I_{M-4} (but not C^I_{M-5}). Similarly, we have

$$\tilde{\Phi}_{C_2^F, C_3^F} > \tilde{\Phi}_{C_2^I, C_1^I} > \tilde{\Phi}_{C_1^I, C_1^F} > \tilde{\Phi}_{C_1^F, C_2^F} \qquad (100)$$

so that C_2^I should evolve into C_2^F. It should be noted that $\tilde{\Phi}_{C_1, C_2}$ is not the best method of deciding when to jump; as is evident from the simulation, many other jumps can also be taken without causing any troubles. Maybe, a measure of closeness of the two curves can play a role here.

12.Conclusion and Future Research

In this paper, we showed that a formation control problem can be solved using a coupled geometrical-physical approach: canonical invariant representations for describing shapes and elastic models for enforcing natural physical behaviours. In the particular case considered in this paper (i.e., contour formations), Fourier descriptors together with snake models are natural choices. Desired points on the target curve can be synthesized using a bunch of Fourier descriptors which can describe reasonably complex shapes in a robust way. There are a number of issues which have to be addressed in future research, including extension to the case of open contours, starting from an arbitrary formation, extension to 3-D, extension to flat surfaces with defined boundaries, *split* and *join* operation, interaction of multiple formations using a spatial map, and finally, implemention on Serafinas.

References

[1] T.B.Curtin, J.G.Bellingham, J.Catipovic, and D.Webb; *Autonomous Oceanagraphic Sampling Networks*; Oceanagraphy, 6:86-94, 1989.

[2] Randal W. Beard, Jonathan Lawton, Fred Y. Hadaegh; *A Coordination Architecture for Spacecraft Formation Control*; IEEE Transactions on Control Systems Technology, Vol. 9, No. 6, Nov. 2001, pp. 777-790.

[3] R.Olfati-Saber and R.M.Murry; *Distributed Cooperative Control of Multiple Vehicle Formations Using Structural Potential Functions*; The 15th IFAC World Conf., June 2002.

[4] M. Egerstedt, X. Hu; *Formation Constrained Multi-Agent Control*; IEEE Trans. on Robotics and Automation, Vol. 17, No. 6, Dec. 2001, pp. 947-951.

[5] E. Fiorelli, P. Bhatta, N. E. Leonard; *Adaptive Sampling Using Feedback Control of an Autonomous Underwater Glider Fleet*; Proc. of 12th Int. Symp. on Unmanned Untethered Submersible Technology (UUST), Aug. 2003.

[6] P.Ogren, E. Fiorelli, and N.E.Leonard; *Formations with a Mission: Stable Coordination of Vehicle Group Maneuvers*; Proc. Symp. Math. Theory of Networks and Systems, Aug. 2002.

[7] C. Belta, V. Kumar; *Towards abstraction and control for large groups of robots*; 2nd International Workshop on Control Problems in Robotics and Automation, Las Vegas, NV, Dec. 2002.

[8] R. Haining; *Spatial Data Analysis*; Cambridge University Press, 2003.

[9] H. Burkhardt; *Transformationen zur lageinvarianten Merkmalgewinnung*; Habilitationsschrift, Karlsruhe University, 1979.

[10] A.L. Bertozzi, M. Kemp, D. Marthaler; *Determining Environmental Boundaries: Asynchronous communication and physical scales*; Proceedings of the Block Island Workshop on Cooperative Control 2003, Lecture Notes in Control and Information Systems, Springer-Verlag.

[11] M. Kass, A. Witkin, D. Terzopoulos; *Snakes: Active Contour Models*; International Journal of Computer Vision, Vol.1, 1988, pp. 321-331.

[12] H. Jaeger; *The Dual Dynamics Design Scheme for Behavior-based Robots: a Tutorial*; Arbeitspapiere der GMD 966, GMD, St. Augustin, 1996.