A Formation Control Approach to Adaptation of Contour-Shaped Robotic Formations

Shahab Kalantar & Uwe R. Zimmer
Australian National University
Research School for Information Science and Engineering
and the Faculty for Engineering and Information Technology
Canberra, ACT 0200, Australia
shahab.kalantar@rsise.anu.edu.au | uwe.zimmer@ieee.org

1 Much research has been done in the area of robot formations. Most of them consider rigid formations where the robot aggregate forms a rigid virtual body. Relatively little has been done on deformable formations composed of rigid links as well as flexible ones. In this paper, we will examine and design controllers for a special type of robotic formations, i.e., those resembling contours. These type of formations have numerous applications in the underwater world, including adaptation to plume boundaries and isoclines of concentration fields, flock shepherding, and shape formation. We adopt general curve evolution theory as a suitable abstraction to describe the motion of such formations. We will first design controllers using simple geometrical reasoning, based on basic requirements on connectivity and mission accomplishment, and will later show that they lead to the same controller structure.

1. Introduction

The emergence of overwhelming interest in formation control within mobile robotics community is the direct result of the need to deploy multiple small relatively low-profile robotic vehicles instead of bigger more complex units. The use of multiple vehicles increases robustness and fault tolerance, while making possible the coverage of much larger areas, thus increasing sensing capabilities by orders of magnitude. In the mean time, the ability to cover large areas makes the exploration of natural phenomena, with considerable spatial extent, much simpler and time (and cost) effective. This, in turn, brings about the challenge of designing distributed control strategies to achieve coordination (including synchronization). Much of the research in this area deal with relatively small number of robots with pre-defined topologies. Parallel to this, the swarm robotics community study aggregates composed of large numbers of homogeneous autonomous units. The dichotomy between these two domains has started to fade away, though.

Formations can be achieved in various ways. Leader-following [6] (the relationship being defined by a formation graph), virtual structures (treating the formation as a rigid body and using formation functions [8], artificial potentials [4], structural potentials [9]), reactive (behaviour-based) methods [7] (in which the formation emerges), abstraction-based approach [12] (using a small set of shape descriptors), shape variables [13] and coordination variables [5] (an attempt at unifying different schemes), are among the strategies proposed to model formations, as well as numerous papers dealing with low level control schemes to realize any one of the above methods.

A rigid formation is one in which the relative states of individual robots, with respect to every other robot, are kept fixed (within certain bounds) at equilibrium. The formation may go through a number of pre-defined configurations according to the characteristics of the current environment, undergoing deformation in during transitions. While the shape of a rigid formation is fixed (at equilibrium), the shape of a deformable formation is always directly dictated by the shape of the environment. In [11], control strategies are proposed for covering a region with mobile sensors. In [10], a chain of robots enclose a target using simple behaviours. In [15], a particular type of deformable model from machine vision is used to adapt a chain of robots to the boundary of an environmental level-set. In [1] and [3], a more general method based on curve evolution is used. In this paper, we deal with this problem from a formation point of view.

The physical robots we will ultimately use to implement the proposed system are small agile subsensibles (called Serafina) developed at our laboratory. In
In this paper, we constrain the motion of the formation to a plane \( P \) parallel to the bottom of the ocean (figure 1). With suitable inputs for thrusters \( T_{x} \), \( T_{y} \), and \( T_{z} \), we can control the depth (heave motion) as well as maintaining zero roll angle. \( T_{x} \) and \( T_{y} \) can be used to move the robot on the plane. Thus, considering that swaying is not directly controllable, we can model the motion of a Serafina, confined to \( P \), as a simple non-holonomic unicycle. In this paper, for the sake of simplicity, we will treat robots as particles attached to each robot. In the simplest case, when robots are treated as particles, the state of each robot can be given by its \( x \) and \( y \) coordinates \( p_{i}(t) = (x_{i}(t), y_{i}(t)) \), measured with respect to \( \Upsilon \). Thus, the state is given by the mapping \( \sigma_{1}: R \rightarrow \mathbb{R}^{2} \), \( \sigma_{1}: R \rightarrow q_{i}(t) = p_{i}(t) \). If the orientation of the robots are important, the state is defined by \( \sigma_{2}: R \rightarrow \mathbb{R}^{2} \otimes [0, 2\pi] \), \( \sigma_{2}: R \rightarrow q_{i}(t) = (p_{i}(t), \theta_{i}(t)) \), where \( \theta_{i} \) is measured with respect to \( \Upsilon \), and defines the rotation matrix of \( \Upsilon_{i} \). To couple the robots with the surrounding environment \( E \), and hence make possible their interaction with environmental features, we define an idealized virtual sensor \( s: E \otimes T \rightarrow \mathbb{R}^{2} \otimes [0, 2\pi] \), \( s: E \otimes T \rightarrow \mathbb{R}^{2} \otimes [0, 2\pi] \), \( \sigma_{2}: R \rightarrow q_{i}(t) = (p_{i}(t), \theta_{i}(t), s_{i}(t)) \). We can now define the augmented state of each robot by \( \sigma_{1}: R \rightarrow \mathbb{R}^{2} \otimes [0, 2\pi] \otimes \mathbb{R}^{3} \), \( \sigma_{2}: R \rightarrow q_{i}(t) = (p_{i}(t), \theta_{i}(t), s_{i}(t)) \). The aggregate state of the whole collection is denoted by

\[
q(t) = [q_{0}(t), ..., q_{N-1}(t)]^T.
\]

A formation \( F_{R} \) on \( R \) is a collection of inequalities

\[
\{ \Lambda_{a}(\sigma_{a}(S_{\beta}), P_{\beta}) \leq 0 \}
\]

where \( \Lambda_{a}: \sigma_{a}(R) \otimes \mathbb{R}^{k} \rightarrow \mathbb{R} \) is a non-linear function, \( S_{\beta} \) is a subset of \( R \), and \( P_{\beta} \) is a vector of parameters.
The formation can, in general, be a function of time. A formation energy function can be defined as

$$E_f(R) = \sum_{\beta \in \Lambda} \Lambda_\beta(\sigma_a(S_\beta), P_\beta)$$  \hspace{1cm} (3)$$

which is a measure of deviation from a valid formation. Accordingly, a formation characteristic function determines if an aggregate of robots is a $F_k$-formation and is defined as $C_f(R) = 1$ if $E_f(R) = 0$ and 0, otherwise. A partially rigid formation is one in which at least one of the inequalities are replaced with a strict equality, provided the set of constrains are consistent. In a rigid formation, all of the constrains are equalities and in such a way that the whole aggregate forms a rigid body. In such a formation the distance between each pair of robots is kept constant. The formation can thus be defined as the zero level set of $E_f(R)$ \[8\]. On the other hand, in a deformable formation, we have, at least one inequality. Figure 3 shows examples of rigid and deformable (non-rigid) formations.

In a rigid formation, it makes sense to define a formation coordinate system $Y_g$, attached to one of the robots (the designated leader), or an imaginary point $r_0$ (usually, the centre of mass), moving with formation. The only degrees of freedom of such a formation is, thus, defined by translation of this point on the plane and rotation of the whole formation around it. This is, of course, not a convenient way of defining deformable formations.

Similar to formation functions, we can define a mission function $M_R$ to formalize the goal of a formation. This can be done by defining the collection

$$\{s_t(\Gamma_\gamma(\sigma_a(S_t), P_t)) = s_\gamma\}$$  \hspace{1cm} (4)$$

where, as before, $\Gamma_\gamma: \sigma_a(R) \otimes \mathbb{R}^n \rightarrow \mathbb{R}$ is a non-linear function, $S_t$ is a subset of $\mathbb{R}$, $P_t$ is a vector of parameters, and $s_t$ are constants defining an environmental feature. Now, the mission function can be defined as

$$M_R = \sum_{\gamma} (s_t(\Gamma_\gamma(\sigma_a(S_t), P_t)) - s_\gamma)$$  \hspace{1cm} (5)$$

which determines how far the formation is from its goal. This decoupling of formation keeping and mission accomplishment simplifies things a lot and has been exploited by some (see [4]). A very important application of formations is to localize and converge to a source of release of some environmental chemical. In the case of rigid formations, the mission can be considered accomplished if $s_\gamma(R_T) = s_M$, where $s_M$ is the maximum concentration of the plume. An alternative is for the formation to adapt itself to a certain desired iso-cline of the plume. If $s_M$ denotes the desired concentration, the mission is fulfilled if $s_\gamma(q_t) = s_M$. If $\gamma: [0, 1] \rightarrow \mathbb{R}^2$ denotes the curve induced by the concentration $s_M$, then mission accomplishment can alternatively be defined as the convergence of the robots in the formation to this curve. In this case, a deformable formation shaped as a curve is more suitable. Such a formation, which we call a contour formation, is a succession of basic structures shown in figure 3b. In the next section, we attempt to design distributed controllers for individual robots such that, at all times, we have $E_f(R) = 0$, and the motion of the formation is towards the minimization of $M_R$.

Remark 1: The concepts of rigid and deformable formations are intimately related. If the set of configurations a rigid formation goes through is indexed by a discrete set, whereas that for a deformable formation is a continuous set.

Remark 2: We do not consider tolerance bounds as deformations.

Remark 3: Our definition implies the existence of a fixed connectivity graph. As such, swarms are not included.

3. The approach

Adaptation to an iso-cline of a field $F$ is made possible through climbing the concentration gradient while maintaining formation integrity. We will stick to the following assumptions:

Assumption 1: The concentration field $F(q)$ is assumed to be defined in a domain $D$ in $\mathbb{R}^3$. We also assume that it is continuous everywhere in this domain. This also implies that we only consider the case of turbulent-free flow.

Assumption 2: The concentration gradient $\nabla F(q)$ is assumed to be defined everywhere and smooth in $D$. Again, this is too much of an assumption in practice. We assume that a suitable filtering procedure provides a sufficiently smooth gradient field.

Assumption 3: The formation is initialized inside the domain where the field is defined. If this is not the case, then the formation will have to march, almost randomly, to get in contact with the domain.
Assumption 4: \( \nabla F(q) \) has non-zero projection \( \nabla F(q) \parallel \) on the plane of motion \( P \) of the formation. In the following, by \( \nabla F(q) \), we actually mean \( \nabla F(q) \parallel \).

Assumption 5: \( R_i \) can measure \( F(q_i) \) and \( \nabla F(q_i) \). For some special kinds of fields and using special sensors, this may be possible. This problem will be dealt with in a sequel paper.

Assumption 6: \( R_i \) is able to measure the relative positions \( q_{i-1} \) and \( q_{i+1} \) (through communication or active sensing, or a combination of them).

Assumption 7: In the simulations, we will only consider the case of an open contour with fixed (manually controlled) end points which has the same behaviour as a closed chain with \( R_N = R_0 \) and \( R_{-1} = R_{N-1} \). This will be relaxed in future papers.

Assumption 8: the curve \( \gamma_d : [0, 1] \rightarrow \mathbb{R}^2 \), representing the target iso-cline, corresponding to the desired concentration \( F_0 \) of the field \( F \), is a simple smooth closed curve.

We can now state the requirements imposed on the controller:

Requirement 1: (Stabilization) The formation has to stop eventually. This means that the group should asymptotically converge to an equilibrium point of the energy function defined below (which may not be the iso-cline). In other words, we should have \( \dot{q}_i(t_f) = 0 \), where \( t_f \) is some finite time instant.

Requirement 2: (Adaptation) The formation should converge to and stop at the desired iso-cline. This means that, after stabilization, we should have \( F(q_i) = s_d(q_i(t_f), t_f) = F_d \).

Requirement 3: (Separation) At all times, the constraint \[ \| q_i(t) - q_{i-1}(t) \| = \| q_{i+1}(t) - q_i(t) \| \] should be satisfied, which basically means that the robots should be uniformly distributed on the imaginary curve representing the formation. In the case of an open contour, the motion of end robots has to make sure that \( \| q_i - q_{i-\alpha} \| \) lies within some specified bounds.

Requirement 4: (Smoothness) At all times, the contour formation has to satisfy some form of smoothness (which will be explained in more detail when we discuss relationship with active contours). The pertinent constraint can be stated as \[ | \text{acos}(\Phi_i(t)) - \pi | \leq \varphi_d \] where \( \varphi_d \) is the maximum deviation from a straight line, and

\[
\Phi_i(t) = \frac{(q_{i+1}(t) - q_{i-1}(t)) \cdot (q_i(t) - q_{i-1}(t))}{\| q_{i+1}(t) - q_{i-1}(t) \| \| q_i(t) - q_{i-1}(t) \|} \tag{6}
\]

For a closed contour formation, the energy function, representing all of the requirements can be stated as

\[
E_c(q(t)) = \sum_{i=0}^{N-1} \left( (F(q_i(t)) - F_d)^2 + (\text{acos}(\Phi_i(t)) - \pi)^2 + \frac{\| q_i(t) - q_{i-1}(t) \|}{\| q_i(t) - q_{i+1}(t) \|} + \frac{1}{2} \right) \tag{7}
\]

With fixed end-points, \( i \) runs from 1 to \( N - 2 \). Note that, when in an equilibrium, the separation constraints are automatically attained through the last term. It is straightforward to have the robots move according to the gradient of the energy function, finally ending up in a local equilibrium state. More formally,

\[
\dot{q}_i(t) = \nabla_{q_i(0)} E(q(t)) \tag{8}
\]

will give the steepest descent direction. As will be seen later, the formation may converge to undesirable local minima. The mechanism used to escape these local minima has to be incorporated in a meaningful and intuitive way into the above control system. Moreover, similar to what was said about decoupling formation control and mission accomplishment, it would be desirable to decouple the motion of the formation into simpler behaviours. Finally, by decoupling, we will arrive at a framework very similar to that used in curve evolution, as will be seen later.

As far as contour formations are concerned, the controller given in equation (8) has a number of drawbacks. First, the control law is expressed in the global inertial coordinate system \( \Sigma \). This implicitly implies that such a system should exist and be known to all the robots. This problem can be remedied by expressing the control law in the local coordinates \( \Sigma_i \). Even so, the motions in the \( \Sigma_i \) and \( \Sigma_\gamma \) directions are not decoupled. It would have been much better if we could exploit the nature of a contour formation to define more natural coordinates in which controllers could be designed more intuitively. Such a coordinate system can be arrived at by modelling the contour formation as an ideal imaginary continuous curve \( \gamma_R \) passing through all the robots. For such a curve, motion of every point \( \gamma_R(s) \) can be decomposed into a normal and a tangential component. Translating back to the formation, the tangent direction for \( R_i \) can be defined by

\[
\dot{T}_i(t) = \frac{q_{i+1}(t) - q_i(t)}{\| q_{i+1}(t) - q_i(t) \|} \tag{9}
\]
\[ T(s, t) = \frac{\partial \gamma(s)}{\partial s} \left( \frac{\partial \gamma(s)}{\partial s} \right) \]  

while the normal direction is defined by \( \bar{N}(t) = T(t) \). The more robots there are and the more close they are together, the more accurate these approximations would be. We will then attach the coordinate system \( Y_{NT} = \{ \bar{N}, \bar{T}, \} \) to \( R_i \). A point \( p_{XY} \in \mathbb{R}^2 \) has the representation \( p_{NT} = R(\theta_j)(p_{XY} - q_i) \) in \( Y_{NT} \), where

\[
\theta_j = \arctan \left( \frac{y_{i+1} - y_{i-1}}{x_{i+1} - x_{i-1}} \right) - \frac{\pi}{2} 
\]

We will then design and express the controllers with respect to \( Y_{NT} \). The form of motion equations would be

\[
\dot{q}_{NT}(t) = \begin{pmatrix} \eta_2(t) \\ \eta_1(t) \end{pmatrix} 
\]

which has the representation

\[
\dot{q}_{(t)} = \eta_{\gamma_1}(t) \bar{N}(t) + \eta_{\gamma_2}(t) \bar{T}(t) = \eta_{\gamma_1}(t) R_i(\bar{\theta}_i) \dot{\bar{\theta}}_i + \eta_{\gamma_2}(t) R_i(\bar{\theta}_i) \]  

in \( Y_i \). Note that this change of coordinates has effectively transformed the problem into two one-dimensional problems. This way, stability results for one-dimensional problems can be applied.

### 4. Controller design

In this section, we will design decoupled controllers for fulfilling each of the above requirements. We will often make use of the function

\[ G(x, \sigma) = e^{-\|x\|^2/\sigma^2} \]

where \( x \in \mathbb{R}^n \) and \( \| \cdot \| \) denotes the standard Cartesian metric in \( \mathbb{R}^n \). \( G(x) \) is bounded between zero and one, and is used to exponentially slow down a process \( p(x) \) (via using \( G(x, \sigma)p(x) \)) when \( x \) is grown beyond a certain limit, determined through the parameter \( \sigma \). We will also use the function \( H(x, \sigma) = 1 - G(x, \sigma) \) to slow down the process \( p(x) \) when in a certain neighborhood of the origin. Finally, the notation \( H'(x, \sigma) = sign(x)H(x, \sigma) \), where \( x \in \mathbb{R} \), will be used to simplify formulas.

**Adaptation:** To achieve \( F(q_i) = F_{Ad} \) for a given concentration \( F_{Ad} \), we can make use of gradient information in a control law such as

\[
\dot{q}_i(t) = C_{\gamma_j}(t) = -\beta_1 s(F(q_i(t)))(d(F(q_i(t)))) \nabla_{\gamma_0}(F(q_i(t))) 
\]

where \( s(F(q_i(t))) \) slows down the robot in the vicinity of the iso-cline and stops it when the desired level set has been reached, and \( d(F(q_i(t))) \) determines the direction of motion along the field gradient \( \nabla_{\gamma_0}(F(q_i(t))) \). More specifically,
Section: Controller design

\[
Q(q_i(t), L_{T}) = q_{i-1}(t) + n(q_{i+1}(t)) - q_{i-1}(t)
\]

\[
n = \frac{(q_i(t) - q_{i-1}(t)) \cdot (q_{i+1}(t) - q_{i-1}(t))}{\|q_{i+1}(t) - q_{i-1}(t)\|^2}
\]

Thus, the motion in the direction of the normal can be achieved by

\[
\dot{q}_i(t) = \beta_2 H(Q(q_i, L_T) - q_i(t), \sigma_2) \frac{Q(q_i, L_T) - q_i(t)}{\|Q(q_i, L_T) - q_i(t)\|}
\]

Note that \((Q(q_i, L_T) - q_i(t)) / \|Q(q_i, L_T) - q_i(t)\|\) is co-linear with \(\vec{N}_i(t)\). Alternatively, the same thing can be done by the control law \(\dot{q}_i(t) = -\nabla q_i(t) \Delta_i(t)\) which is equal to

\[
-2\sqrt{\Delta_i(t)} \cdot \left( \frac{(q_{i+1}(t) - q_{i-1}(t)) \cdot (q_{i+1}(t) - q_{i-1}(t))^T}{\|q_{i+1}(t) - q_{i-1}(t)\|^2} - I_2 \right)
\]

**Separation:** To satisfy the separation constraint, we can use

\[
\dot{q}_i(t) = C_{\gamma_i}(t) = \beta_3 H(z_i, \sigma_3) \begin{pmatrix} z_i \end{pmatrix} + \beta_3 H(z_i^*, \sigma_3) \begin{pmatrix} z_i^* \end{pmatrix}
\]

where

\[
\begin{align*}
\dot{z}_i &= \left( q_{i-1}(t) + \frac{q_i(t) - q_{i-1}(t)}{\|q_i(t) - q_{i-1}(t)\|} \cdot 2 \rho_i(t) \right) - q_i(t) \\
\dot{z}_i^* &= \left( q_{i+1}(t) + \frac{q_i(t) - q_{i-1}(t)}{\|q_i(t) - q_{i-1}(t)\|} \cdot 2 \rho_i(t) \right) - q_i(t) \\
\rho_i(t) &= \|q_i(t) - q_{i-1}(t)\| + \|q_{i+1}(t) - q_i(t)\|
\end{align*}
\]

Referring to figure 7, the line \(L_{N_i} = L(w_1(t), w_2(t))\) is the loci of points that satisfy the separation constraint, where

\[
w_1 = (1/2)(q_{i+1} + q_{i-1}) \quad \text{and} \quad w_2 = (1/2)(q_{i+1} + q_{i-1}) + \|Q(q_i, L_T)\| \vec{N}_i(t)
\]

\[
C_{\gamma_i}(t) = C_{\gamma_i}(t) \cdot \vec{T}_i(t)
\]

acts towards minimization of the objective function

\[
\Xi_i(t) = \frac{1}{2} \left\| Q(q_i(t), L_{N_i}) - q_i(t) \right\|^2
\]

which can be achieved by the control law

\[
\dot{q}_i(t) = \beta_2 H(Q(q_i, L_{N_i}) - q_i(t), \sigma_2) \frac{Q(q_i, L_{N_i}) - q_i(t)}{\|Q(q_i, L_{N_i}) - q_i(t)\|}
\]

where

\[
(Q(q_i, L_{T}) - q_i(t)) / \|Q(q_i, L_{T}) - q_i(t)\|\]

is co-linear with \(\vec{T}_i(t)\), or, alternatively, the law \(\dot{q}_i(t) = -\nabla q_i(t) \Xi_i(t)\) which is equal to

\[
-2\sqrt{\Xi_i(t)} \left( \left( w_1(t) - w_2(t) \right) \left( w_1(t) - w_2(t) \right)^T - I_2 \right)
\]

**Final controller:** Collecting all the controllers together, grouping normal and tangential components, we have

\[
\dot{q}_i(t) = (C_{\gamma_i}(t) \cdot \vec{N}_i(t) + C_{\gamma_i}(t) \cdot \vec{T}_i(t) + C_{\gamma_i}(t) \cdot \vec{T}_i(t) + C_{\gamma_i}(t) \cdot \vec{T}_i(t)) \vec{T}_i
\]

Now, consider the following facts:

1. \(C_{\gamma_i}(t) \cdot \vec{T}_i(t)\) (the tangential component of the field gradient) can in many situations counteract \(C_{\gamma_i}(t) \cdot \vec{T}_i(t)\) which is vital for maintaining the integrity of the formation. We should therefore delete this term. This may, of course, affect the adaptation. Later on, we will discuss methods of going around this deficiency.

2. \(C_{\gamma_i}(t) \cdot \vec{T}_i(t)\) and \(C_{\gamma_i}(t) \cdot \vec{T}_i(t)\) act in the same direction (pulling \(R_i\) towards \(L_{N_i}\), so that just one of them is needed. In accordance with our adherence to decoupling strategy, we reserve \(C_{\gamma_i}(t) \cdot \vec{T}_i(t)\).

3. \(C_{\gamma_i}(t) \cdot \vec{N}_i(t)\) and \(C_{\gamma_i}(t) \cdot \vec{N}_i(t)\) can act in opposite directions. Since, as was mentioned earlier, for separation purposes, convergence to \(L_{N_i}\) suffices, \(C_{\gamma_i}(t) \cdot \vec{N}_i(t)\) is not really needed and can be removed.

4. \(C_{\gamma_i}(t) \cdot \vec{N}_i(t)\) and \(C_{\gamma_i}(t) \cdot \vec{N}_i(t)\) can be at odds with each other. This presents a trade-off between smoothness and adaptation and should be balance by proper values for \(\beta_3\) and \(\beta_2\). With these considerations, the final controller would be

\[
\dot{q}_i(t) = (\beta_3 C_{\gamma_i}(t) + \beta_2 C_{\gamma_i}(t)) \vec{N}_i(t) + (\beta_3 C_{\gamma_i}(t)) \vec{T}_i(t)
\]

\[
= (-\beta_1 H(F(q_i) - F_d, \sigma_1) \langle \nabla L, F(q_i, \vec{N}_i(t)) \rangle + \beta_3 \text{sign}((q_i(t) - q_{i-1}(t)) \cdot \vec{n}_{\gamma_i})) H(Q(q_i, L_T) - q_i(t), \sigma_2) \vec{N}_i(t)
\]

\[
+ \beta_3 \text{sign}((q_i(t) - (1/2)(q_{i-1}(t) - q_{i+1}(t))) \cdot \vec{n}_{\gamma_i})) H(Q(q_i, L_{N_i}) - q_i(t), \sigma_2) \vec{T}_i(t)
\]

where \(\vec{n}_{\gamma_i}\) denotes a unit vector along the line \(L\).
5. Relation to curve evolution

The problem formulated in previous sections is indeed a discrete version of the general case of continuous curves, i.e., the evolution of curves embedded in \( \mathbb{R}^2 \), under the influence of external forces. If \( \gamma : [0,1] \to \mathbb{R}^2 \) is a closed curve, the objective is to minimize the functional \( \int_0^1 g(\gamma'(s), t) \, ds \)

where

\[
g_{\gamma}(q(t)) = g(F_{\gamma}, q(t)) = h_{\gamma} \circ F(q(t))
\]

\[
h_{\gamma}(u) = 1 - \frac{1}{1 + \sigma_1|u - F|}
\]

(32)

Using variational calculus, the steepest descent motion for the curve is found to be

\[
\frac{\partial}{\partial \gamma}(\gamma(s), t) = g(\gamma'(s), t) \kappa(s, t) \mathbf{N}(s, t)
\]

\[-(\nabla_{\gamma} g(\gamma(s), t) \cdot \mathbf{N}(s, t)) \mathbf{N}(s, t)
\]

(33)

where \( \kappa(s, t) \) denotes the curvature of the curve at \( s \).

A discrete version would look like

\[
\frac{\partial}{\partial t} q_i(t) = g(q_i(t), \kappa_i(t)) \mathbf{N}(t)
\]

\[-(\nabla_{q_i} g(q_i(t), \mathbf{N}(t)) \cdot \mathbf{N}(t)) \mathbf{N}(t)
\]

\[+ \eta(q_i(t), q_{i-1}(t), q_{i+1}(t)) \mathbf{T}(t)
\]

(34)

where

\[
\kappa_i(t) = \frac{4(q_{i-1}(t) - 2q_i(t) + q_{i+1}(t))^T (q_{i+1}(t) - q_{i-1}(t))}{\|q_{i+1}(t) - q_{i-1}(t)\|^2}
\]

and

\[
\nabla_{q_i} g(q_i(t)) = h_{\gamma}(F(q_i(t))) \nabla_{q_i} F(q_i(t))
\]

\( \eta(\bullet) \) is used to maintain the numerical stability of this Lagrangian scheme. The normal motion, as dictated by (33) has two distinct components, i.e., an internal force for maintaining smoothness and an external force. As can be seen, this is very similar to the controller we designed from scratch. The main difference is that moving by curvature is provably the fastest way of shrinking (smoothing) a curve. Also note that in the continuous version, there are no tangential velocities. This is because this velocity will only change the parametrization not the geometry of the curve. In the discrete version, though, it is necessary. The continuous case can serve as a continuum model for contour formations which might be useful when analyzing very large formations. In the rest of the paper, we will denote, by \( v(s, t) \) (or equivalently, \( \nu(t) \)), a general speed function, which can be a function of the curvature or be the one we designed previously. Also, \( u_i(t) \) will denote a general isocline approach function, while \( w_i(t) \) is used to denote a general tangential speed function. See [14] for more details on curve evolution.

6. Simulation results

In this section, we present a simulation run demonstrating the behaviour of chain formations under the influence of internal and external forces. Figure 8 shows the generic shape of a simulation snapshot together with various pictorial items. The robots move according to

\[
\dot{q}_i(t) = (\beta_1 u_i(t)(\nabla_{q_i(t)} F(q_i(t)), \mathbf{N})) + \beta_2 v_i(t) \mathbf{N}_i
\]

(36)

where

\[
u_i(t) = 2(F(q_i) - 1)(1 - g_i(t))
\]

\[
\frac{\|F(q_i) - Q(q_i(t), F)\|^2}{\sigma_i^2}
\]

(37)

\[
g_i(t) = g_1(q_i(t), v_i(t) = g_i(t)\kappa_i(t) \text{ and } \rho_i \text{ is defined by (25). Note that the external force profile is not the actual level set. The overall performance can be measured by}
\]

\[
s(t) = \frac{1}{N-2} \sum_{i=1}^{N-2} \|g_i(t) - Q(q_i(t), F)\|
\]

(38)

Figure 9 shows a sequence of snapshots of a formation decreasing the error area while keeping the smoothness.

7. Conclusions and future research

We showed that controllers for deformable formations can be designed using simple geometrical considerations and that the resultant has the same structure as differential equations for evolution of plane curves under the influence of internal forces (based on curvature) and external forces. In practice, it is required that certain distance, as well as, smoothness constraints be enforced. Possible avenues for future research include designing reliable controllers, based
on the basic structure, to satisfy connectivity and smoothness constraints, designing appropriate motions for the end robots in an open contour formation, rigorous results for stability and convergence, interaction with humans, gradient estimation, implementation, and methods for dealing with the effect of turbulence. Also of interest would be motion on surfaces (motion constrained to a manifold) and obstacle avoidance.

Acknowledgments
The authors want to thank Jochen Trumpf for highly valuable technical advice.

References