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[1]
Autonomous Underwater Robotics Research Group
Canberra, ACT 0200, Australia
shahab.kalantar@rsise.anu.edu.au | uwe.zimmer@ieee.org

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 submersibles (called *Serafina*) developed at our laboratory. In

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

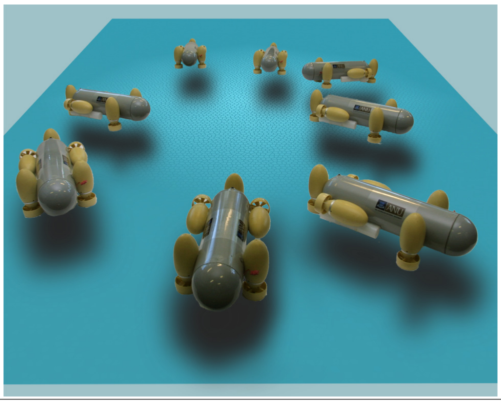

figure 1: Serafina and plane of motion

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^v_b, T^v_{fl}, and T^v_{fr}, we can control the depth (heave motion) as well as maintaining zero roll angle. T^v_{br} and T^v_{bl} 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 moving according to $m_q\ddot{q} + d_q\dot{q} = f_q$, where f_q is the outcome of our design effort. We further ignore inertial effects so that the equation of motion simplifies to $\dot{q} = (1/d_q)f_q$, which designates a path in $\Re^2$. Refer to [16] for more information on Serafinas.

The deformable formations we will discuss in this paper have many potential applications. Contour formations can be used to place a group of robots at a certain level set (iso-cline) of an environmental field, such as, temperature and salinity (figure 2). Such fields most commonly initiate from one or more sources and spread out according to a *diffusion process*, in which particles move proportional to the gradient of the concentration of heat or matter [2]. Once successfully adapted, the formation can traverse the iso-cline, producing a contour map. A variant of the above application is to have the robots enclose (or adapt) to the boundary of a diffusion process, more concretely, containment of oil spills or other contaminants. Adapting to underwater hills, *shepherding* (where a group of robotic shepherds guide a group of live entities (e.g., flocks) to desired states), and exploration of the surface of a ship or large submarine and localization of sources of anomalies (such as explosions or cracks) are among other promising application areas.

2. Modelling formations

In this section, we try to provide a general mathematical framework for describing robotic formations. Suppose that we are given a collection R of N robots

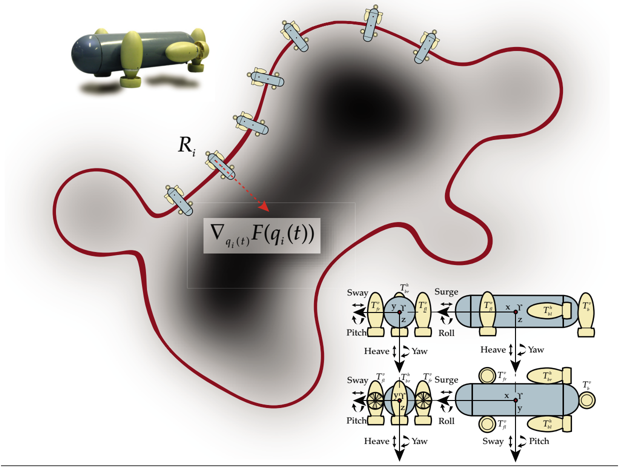

figure 2: Adaptation to fields

R_i, $i = 0, \ldots, N-1$, constrained to move on the plane. Each robot has a unique id $I_i = I(R_i)$ where $I: R \to N$. We can define $I_i = i$. Let Υ denote an inertial coordinate system attached to the plane of movement. Also, let Υ_i be a local coordinate system 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 Υ. Thus, the state is given by the mapping $\sigma_1 : R \to \Re^2$, $\sigma_1 : R_i \mapsto q_i(t) = p_i(t)$. If the orientation of the robots are important, the state is defined by $\sigma_2 : R \to \Re^2 \otimes [0, 2\pi)$, $\sigma_2 : R_i \mapsto q_i(t) = (p_i(t), \vartheta_i(t))$, where ϑ_i is measured with respect to Υ, and defines the rotation matrix of Υ_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 $s_{v_i} : E \otimes T \to \Re^s$, $t \mapsto e_i(t)$, with which every robot is equipped and measures some environmental aspect of interest. To make the presentation simpler, and more realistic, assume that all the sensors are identical and the vector $e_i(t)$ does not contain p_i and ϑ_i. In the following, we will be interested in the special case of $s = 1$. We will also drop E, as it is implicit, and define the *sensor map* as a function of the state of each robot. Thus, the sensor map can be defined by $s_v : \sigma_\alpha(R) \otimes T \to \Re$, where α is 1 or 2. Also, define $s_{v_i}(t) = s_v(q_i(t), t)$. We can now define the augmented state of each robot by $\sigma_3 : R \to \Re^2 \otimes [0, 2\pi) \otimes \Re$, $\sigma_3 : R_i \mapsto q_i(t) = (p_i(t), \vartheta_i(t), s_{v_i}(t))$. The *aggregate* state of the whole collection is denoted by

$$q(t) = [q_0(t), \ldots, q_{N-1}(t)]^T. \tag{1}$$

A *formation* F_R on R is a collection of inequalities

$$\{\Lambda_\beta(\sigma_\alpha(S_\beta), P_\beta) \leq 0\} \tag{2}$$

where $\Lambda_\beta : \sigma_\alpha(R) \otimes \Re^{n_\beta} \to \Re$ is a non-linear function, S_β is a subset of R, and P_β is a vector of parameters.

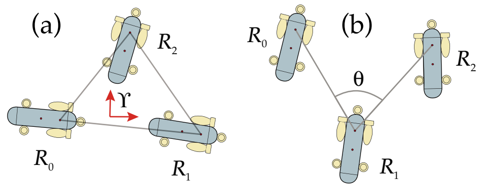

figure 3: Rigid and deformable formations. (a) In a rigid formation, the states of pairs of robots are constrained in such a way that the only degrees of freedom of the formation are those of translation and rotation of Υ_R. In this example, the defining equations are $\|q_0 - q_1\| = d_{01}$, $\|q_1 - q_2\| = d_{12}$ and $\|q_0 - q_2\| = d_{02}$. The energy function is defined as $\|q_0 - q_1\| + \|q_1 - q_2\| + \|q_0 - q_2\|$. (b) This deformable formation is defined by the rigid constraints $\|q_0 - q_1\| = d$ and $\|q_1 - q_2\| = d$, and the non-rigid constraint $|\varphi| \le \varphi_d$.

P_β can, in general, be a function of time. A *formation energy function* can be defined as

$$E_F(R) = \sum_{\beta \ni \Lambda_\beta(\sigma_\alpha(S_\beta), P_\beta) > 0} \Lambda_\beta(\sigma_\alpha(S_\beta), P_\beta) \tag{3}$$

which is a measure of deviation from a *valid* formation. Accordingly, a *formation characteristic function* determines if an aggregate of robots is a F_R-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 Υ_R, attached to one of the robots (the designated leader), or an imaginary point r_R (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

$$\{s_v(\Gamma_\gamma(\sigma_\alpha(S_\gamma), P_\gamma)) = s_\gamma\} \tag{4}$$

where, as before, $\Gamma_\gamma : \sigma_\alpha(R) \otimes \Re^{n_\gamma} \to \Re$ is a non-linear function, S_γ is a subset of $\Re$, P_γ is a vector of parameters, and s_γ are constants defining an environmental feature. Now, the mission function can be defined as

$$M_R = \sum_\gamma (s_v(\Gamma_\gamma(\sigma_\alpha(S_\gamma), P_\gamma)) - s_\gamma) \tag{5}$$

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 $s_v(r_\Upsilon) = s_M$, 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_d denotes the desired concentration, the mission is fulfilled if $s_v(q_i) = s_d$. If $\gamma_d : [0, 1] \to \Re^2$ denotes the curve induced by the concentration s_d, 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 3.b. 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 $\Re^3$. 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 $\nabla_q F(q)$ 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: *$\nabla_q F(q)$ has non-zero projection $\nabla_q F(q)|_P$ on the plane of motion P of the formation. In the following, by $\nabla_q F(q)$, we actually mean $\nabla_q F(q)|_P$.*

Assumption 5: *R_i can measure $F(q_i)$ and $\nabla_{q_i} F(q_i)$. 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 $\gamma_d : [0, 1] \to \Re^2$, representing the target iso-cline, corresponding to the desired concentration F_d 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 $F(q_i) = s_v(q_i(t_f), t_f) = F_d$.*

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 $\|q_i - q_{i-1}\|$ 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 $|\mathrm{acos}(\Phi_i(t)) - \pi| \le \varphi_d$ where φ_d is the maximum deviation from a straight line, and*

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

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

$$\begin{aligned} E_C(q(t)) = {} & \\ \sum_{i=0}^{N-1} \Bigg\{ & (F(q_i(t)) - F_d)^2 + (\mathrm{acos}(\Phi_i(t)) - \pi)^2 \\ & + \left(\frac{\|q_i(t) - q_{i-1}(t)\|}{\|q_i(t) - q_{i-1}(t)\| + \|q_{i+1}(t) - q_i(t)\|} - \frac{1}{2} \right)^2 \Bigg\} \end{aligned} \tag{7}$$

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,

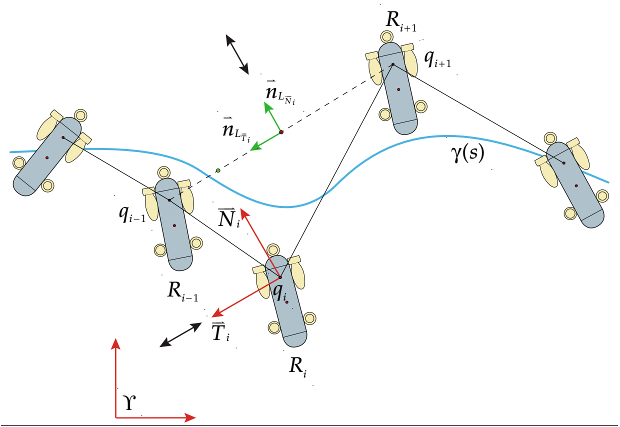

figure 4: Change of coordinates

$$\dot q_i(t) = \nabla_{q_i(t)} E(q(t)) \tag{8}$$

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 Υ. 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 Υ_i. Even so, the motions in the X_{Υ_i} and Y_{Υ_i} 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 cold be designed more intuitively. Such a coordinate system can be arrived at by modelling the contour formation as an ideal imaginary continuous curve γ_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

$$\vec{T}_i(t) = \frac{q_{i+1}(t) - q_i(t)}{\|q_{i+1}(t) - q_i(t)\|} \tag{9}$$

which is an approximation of

$$\vec{T}(s,t) = \frac{\partial\gamma_R(s)/\partial s}{\|\partial\gamma_R(s)/\partial s\|} \tag{10}$$

while the normal direction is defined by $\vec{N}_i(t) = \vec{T}_i(t)^{\perp}$. The more robots there are and the more close they are together, the more accurate these approximations would be. We will attach the coordinate system $\Upsilon_{NT_i} = \{\vec{N}_i, \vec{T}_i\}$ to R_i. A point $p_{XY} \in \Re^2$ has the representation $p_{NT} = R(\vartheta_i)(p_{XY} - q_i)$ in Υ_{NT_i}, where

$$\vartheta_i = \operatorname{atan}\left(\frac{y_{i+1} - y_{i-1}}{x_{i+1} - x_{i-1}}\right) - \frac{\pi}{2} \tag{11}$$

We will design and express the controllers with respect to Υ_{NT_i}. The form of motion equations would be

$$\dot{q}_{NT_i}(t) = \begin{pmatrix} \eta_{\vec{N}_i}(t) \\ \eta_{\vec{T}_i}(t) \end{pmatrix} \tag{12}$$

which has the representation

$$\begin{aligned} \dot{q}_i(t) &= \eta_{\vec{N}_i}(t)\vec{N}_i(t) + \eta_{\vec{T}_i}(t)\vec{T}_i(t) \\ &= \eta_{\vec{N}_i}(t)R_z(\vartheta_i)\vec{\imath} + \eta_{\vec{T}_i}(t)R_z(\vartheta_i)\vec{\jmath} \end{aligned} \tag{13}$$

in Υ_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

$$G(x,\sigma) = e^{-\|x\|_n^{\,2}/\sigma^2} \tag{14}$$

where $x \in \Re^n$ and $\|x\|_n$ denotes the standard Cartesian metric in $\Re^n$. $G(x)$ is bounded between zero and 1, 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 σ. 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 $H^s(x,\sigma) = sign(x)H(x,\sigma)$, where $x \in \Re$, will be used to simplify formulas.

Adaptation: To achieve $F(q_i) = F_d$, for a given concentration F_d, we can make use of gradient information in a control law such as

$$\begin{aligned} &\dot{q}_i(t) = C_{F_i}(t) = \\ &-\beta_1 s(F(q_i(t)))d(F(q_i(t)))\nabla_{q_i(t)}F(q_i(t)) \end{aligned} \tag{15}$$

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 $\nabla_{q_i(t)}F(q_i(t))$. More specifically,

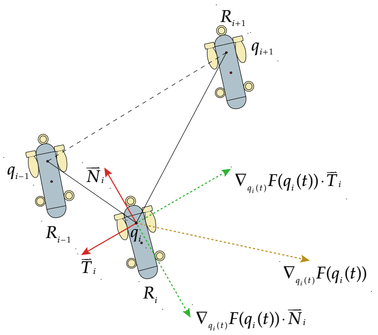
R_{i+1}
q_{i+1}
q_{i-1}
$\vec{N}_i$
$\nabla_{q_i(t)}F(q_i(t)) \cdot \vec{T}_i$
R_{i-1}
q_i
$\vec{T}_i$
$\nabla_{q_i(t)}F(q_i(t))$
R_i
$\nabla_{q_i(t)}F(q_i(t)) \cdot \vec{N}_i$

figure 5: Gradient following

$$\begin{aligned} \dot{q}_i(t) = C_{F_i}(t) = -\beta_1 H(F(q_i) - F_d, \sigma_1) \\ G\left(\nabla_{q_i(t)}F(q_i(t)), \frac{1}{\sqrt{\alpha_1}}\right)\nabla_{q_i(t)}F(q_i(t)) \end{aligned} \tag{16}$$

The $G(\bullet,\bullet)$ term slows down the motion in areas with large gradients, preventing the undesired oscillations. In some applications, a bound $M_{\nabla F}$ on the gradient of the field may be know. In that case, ∇F can be normalized by this bound. This can help us to have more control over the speed. Expressed in Υ_{NT_i},

$$\dot{q}_i(t) = C^N_{F_i}(t)\vec{N}_i(t) + C^T_{F_i}(t)\vec{T}_i(t) \tag{17}$$

where

$$\begin{aligned} C^N_{F_i}(t) &= \langle C_{F_i}(t), \vec{N}_i(t)\rangle, \\ C^T_{F_i}(t) &= \langle C_{F_i}(t), \vec{T}_i(t)\rangle \end{aligned} \tag{18}$$

Smoothing: To achieve $|\varphi_i(t)| \to \pi$, we refer to figure 6. As can be seen from the figure, we just need to set

$$\dot{q}_i(t) = C_{T_i}(t) = \beta_2 H(z_i(t), \sigma_2)\frac{z_i(t)}{\|z_i(t)\|} \tag{19}$$

where $z_i(t) = (1/2)(q_{i+1}(t) - q_{i-1}(t)) - q_i(t)$. $C^N_{T_i}(t) = C_{T_i}(t) \cdot \vec{N}_i(t)$ acts towards minimization of the objective function

$$\Delta_i(t) = \frac{1}{2}\|Q(q_i(t), L(q_{i-1}(t), q_{i+1}(t))) - q_i(t)\|^2 \tag{20}$$

where $Q(\bullet,\bullet)$ denotes the distance of $q_i(t)$ from the line passing through $q_{i-1}(t)$ and $q_{i+1}(t)$ (denoted by $L(q_{i-1}(t), q_{i+1}(t)) = L_{T_i}$) and is given by

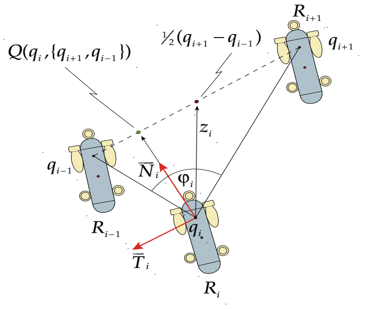

figure 6: Achieving smoothness

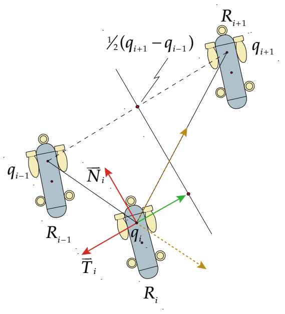

figure 7: Uniform spacing

$$Q(q_i(t), L_{T_i}) = q_{i-1}(t) + \vec{u}(q_{i+1}(t)) - q_{i-1}(t)$$

$$\vec{u} = \frac{(q_i(t) - q_{i-1}(t)) \cdot (q_{i+1}(t) - q_{i-1}(t))}{\|q_{i+1}(t) - q_{i-1}(t)\|} \tag{21}$$

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

$$\dot{q}_i(t) = \beta_2 H(Q(q_i, L_{T_i}) - q_i(t), \sigma_2) \frac{Q(q_i, L_{T_i}) - q_i(t)}{\|Q(q_i, L_{T_i}) - q_i(t)\|} \tag{22}$$

Note that $(Q(q_i, L_{T_i}) - q_i(t))/\|Q(q_i, L_{T_i}) - q_i(t)\|$ 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

$$-2\sqrt{\Delta_i(t)} \cdot \left(\frac{(q_{i+1}(t) - q_{i-1}(t))(q_{i+1}(t) - q_{i-1}(t))^T}{\|q_{i+1}(t) - q_{i-1}(t)\|^2} - I_2\right) \tag{23}$$

Separation: To satisfy the separation constraint, we can use

$$\dot{q}_i(t) = C_{S_i}(t) = \beta_3 H(z^{-}{}_i, \sigma_3)\frac{z_i^-}{\|z_i^-\|} + \beta_3 H(z^{+}{}_i, \sigma_3)\frac{z_i^+}{\|z_i^+\|} \tag{24}$$

where

$$z_i^- = \left(q_{i-1}(t) + \frac{q_i(t) - q_{i-1}(t)}{\|q_i(t) - q_{i-1}(t)\|}\left(\frac{1}{2}\rho_i\right)\right) - q_i(t)$$

$$z_i^+ = \left(q_{i+1}(t) + \frac{q_i(t) - q_{i+1}(t)}{\|q_i(t) - q_{i+1}(t)\|}\left(\frac{1}{2}\rho_i\right)\right) - q_i(t)$$

$$\rho_i(t) = \|q_i(t) - q_{i-1}(t)\| + \|q_{i+1}(t) - q_i(t)\| \tag{25}$$

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

$$w_1 = (1/2)(q_{i+1} + q_{i-1}) \text{ and} \tag{26}$$

$$w_2 = (1/2)(q_{i+1} + q_{i-1}) + \|Q(q_i, L_{T_i})\|\vec{N}_i(t) \tag{27}$$

$C^T_{S_i}(t) = C_{S_i}(t) \cdot \vec{T}_i(t)$ acts towards minimization of the objective function

$$\Xi_i(t) = \frac{1}{2}\|Q(q_i(t), L_{N_i}) - q_i(t)\|^2 \tag{28}$$

which can be achieved by the control law

$$\dot{q}_i(t) = \beta_2 H(Q(q_i, L_{N_i}) - q_i(t), \sigma_2) \frac{Q(q_i, L_{N_i}) - q_i(t)}{\|Q(q_i, L_{N_i}) - q_i(t)\|} \tag{29}$$

where $(Q(q_i, L_{T_i}) - q_i(t))/\|Q(q_i, L_{T_i}) - q_i(t)\|$ 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

$$-2\sqrt{\Xi_i(t)}\left(\frac{(w_1(t) - w_2(t))(w_1(t) - w_2(t))^T}{\|w_1(t) - w_2(t)\|^2} - I_2\right)$$

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

$$\begin{aligned} \dot{q}_i(t) = {} & (C_{F_i}(t) \cdot \vec{N}_i(t) + C_{T_i}(t) \cdot \vec{N}_i(t) + C_{S_i}(t) \cdot \vec{N}_i(t))\vec{N}_i(t) \\ & + (C_{F_i}(t) \cdot \vec{T}_i(t) + C_{T_i}(t) \cdot \vec{T}_i(t) + C_{S_i}(t) \cdot \vec{T}_i(t))\vec{T}_i \end{aligned} \tag{30}$$

Now, consider the following facts:

1. $C_{F_i}(t) \cdot \vec{T}_i(t)$ (the tangential component of the field gradient) can in many situations counteract $C_{S_i}(t) \cdot \vec{T}_i(t)$ (or $C_{T_i}(t) \cdot \vec{T}_i(t)$) 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. $C_{S_i}(t) \cdot \vec{T}_i(t)$ and $C_{T_i}(t) \cdot \vec{T}_i(t)$ 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 $C_{S_i}(t) \cdot \vec{T}_i(t)$.

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

4. $C_{F_i}(t) \cdot \vec{N}_i(t)$ and $C_{T_i}(t) \cdot \vec{N}_i(t)$ can be at odds with each other. This presents a trade-off between smoothness and adaptation and should be balance by proper values for β_1 and β_2. With these considerations, the final controller would be

$$\begin{aligned} \dot{q}_i(t) & \\ = {} & (\beta_1 C^N{}_{F_i}(t) + \beta_2 C^N{}_{T_i}(t))\vec{N}_i(t) + \beta_3 C^T{}_{S_i}(t)\vec{T}_i(t) \\ = {} & (-\beta_1 H^s(F(q_i) - F_d, \sigma_1)\langle\nabla_{q_i(t)}F(q_i(t)), \vec{N}_i(t)\rangle \\ & + \beta_2 sign((q_i(t) - q_{i-1}(t)) \times \vec{n}_{L_{T_i}}) \\ & \quad H(Q(q_i, L_{T_i}) - q_i(t), \sigma_2))\vec{N}_i(t) \\ & + \beta_3 sign((q_i(t) - (1/2)(q_{i-1}(t) - q_{i+1}(t))) \times \vec{n}_{L_{N_i}}) \\ & \quad H(Q(q_i, L_{N_i}) - q_i(t), \sigma_3))\vec{T}_i(t) \end{aligned} \tag{31}$$

where $\vec{n}_L$ 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 R^2, under the influence of external forces. If $\gamma : [0,1] \rightarrow \Re^2$ is a closed curve, the objective is to minimize the functional $\int_0^1 g(\gamma(s, t))ds$

where

$$g_{F_d}(q_i(t)) = g(F_d, q_i(t)) = h_{F_d} \circ F(q_i)$$

$$h_{F_d}(u) = 1 - \frac{1}{1 + \sigma_1 |u - F_d|^2} \tag{32}$$

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

$$\frac{\partial}{\partial t}\gamma(s, t) = g(\gamma(s, t))\kappa(s, t)\vec{N}(s, t) - (\nabla_{\gamma(s, t)} g(\gamma(s, t)) \cdot \vec{N}(s, t))\vec{N}(s, t) \tag{33}$$

where $\kappa(s, t)$ denotes the curvature of the curve at s. A discrete version would look like

$$\frac{\partial}{\partial t}q_i(t) = g(q_i(t))\kappa_i(t)\vec{N}_i(t) - (\nabla_{q_i(t)} g(q_i(t)) \cdot \vec{N}_i(t))\vec{N}_i(t) + \eta(q_i(t), q_{i-1}(t), q_{i+1}(t))\vec{T}_i(t) \tag{34}$$

where

$$\kappa_i(t) = \frac{4(q_{i-1}(t) - 2q_i(t) + q_{i+1}(t))^{\perp^T}(q_{i+1}(t) - q_{i-1}(t))}{\|q_{i+1}(t) - q_{i-1}(t)\|^2} \tag{35}$$

and $\nabla_{q_i(t)} g(q_i(t)) = \dot{h}_{F_d}(F(q_i(t)))\nabla_{q_i(t)} F(q_i(t))$

$\eta(\bullet)$ 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 $\nu(s, t)$ (or, equivalently, $\nu_i(t)$), 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 *iso-cline 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

$$\dot{q}_i(t) = (\beta_i u_i(t)\langle\nabla_{q_i(t)}F(q_i(t)), \vec{N}_i\rangle + \beta_2\nu_i(t))\vec{N}_i + \beta_3 w_i(t)\vec{T}_i \tag{36}$$

where

$$u_i(t) = 2(F(q_i) - 1)(1 - g_i(t))^2$$

$$F(q_i) = e^{-\frac{\|q_i(t) - Q(q_i(t), F)\|^2}{\sigma^2}}$$

$$w_i(t) = -H\left(\|q_i - q_{i-1}\|, \frac{\rho_i}{2}\right) + H\left(\|q_i - q_{i+1}\|, \frac{\rho_i}{2}\right) \tag{37}$$

$g_i(t) = g(1, q_i(t))$, $\nu_i(t) = g_i(t)\kappa_i(t)$ and ρ_i is defined by (25). Note that the external force profile is not the actual level set. The overall performance can be measured by

$$s(t) = \frac{1}{N-2}\sum_{i=1}^{N-2}\|q_i(t) - Q(q_i((t), F))\| \tag{38}$$

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

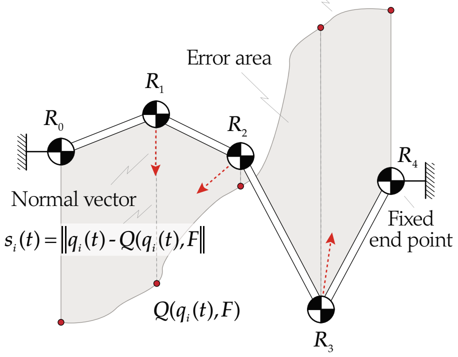

figure 8: Simulation snapshot

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

[1] Kalantar, S., Zimmer, U., *Contour Shaped Robotic Formations for Isocline Adaptation*, Proc. of the intl. conf TAROS '05, London, U.K., 2005.

[2] Okubo, A., *Diffusion and Ecological Problems: Mathematical Models*, Vol. 10 in Biomathematics Series, Springer-Verlag, 1980.

[3] Kalantar, S., Zimmer, U., *Control of Contour Formations of Autonomous Vehicles by General Curve Evolution Theory*, Proc. of IWUR'05, Genoa, Italy, 2005.

[4] Leonard, N.E., Fiorelli, E., *Virtual Leaders, Artificial Potentials and Coordinated Control of Groups*, Proc. 40th IEEE Conf. Decision and Control, 2001.

[5] Beard, R.W., Lawton, J., Hadaegh, F.Y., *A Coordination Architecture for Spacecraft Formation Control*, IEEE Trans. on Control Systems Technology, Vol. 9, No 6, Nov. 2001.

[6] Desai, J.P., Ostrowski, J., Kumar, V., *Controlling Formations of Multiple Mobile Robots*, Proc. IEEE Intl. Conf. Robotics and Automations, Belgium, May 1998.

[7] Balch, T., Arkin, R.C., *Behavior-based formation control for multiagent robot teams*, IEEE Trans. on Robotics and Automation, December, 1998.

[8] Egerstedt, M., Hu, X., *Formation Constrained Multi-Agent Control*, IEEE Conf. on Robotics and Automation, Seoul, Korea, May 2001.

[9] Olfati-Saber, R., Murray, R.M., *Distributed Cooperative Control of Multiple Vehicle Formations Using Structural Potential Functions*, 15th IFAC World Congress, Barcelona, Spain, 2002.

[10] Yamaguchi, H., *A Cooperative Hunting Behaviour by Mobile-Robot Troops*, The Intl. Journal of Robotics Research, Vol. 18, No. 8, Sept. 1999.

[11] Cortes, J., Martinez, S., Bullo, F., *Coverage control for mobile sensing networks*, EEE Trans. on Robotics and Automation, Vol. 20, No. 2, 2004.

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

[13] Krishnaprasad, P.S., Zhang, F., Goldgeier,M., *Control of Small Formations Using Shape Coordinates*, Proc. IEEE Intl. Conf. on Robotics and Automation, 2003.

[14] Caselles, V., Kimmel, R., Sapiro, G., *Geodesic Active Contours*, Intl. Journal of Computer Vision, Vol. 22, No. 1, 1997.

[15] Marthaller, D., Bertozzi, A.L., *Tracking environmental level sets with autonomous vehicles*, Proc. of the Conf. on Cooperative Control and Optimization, Gainesville, Florida, Dec. 2002.

[16] http://users.rsise.anu.edu.au/~serafina/

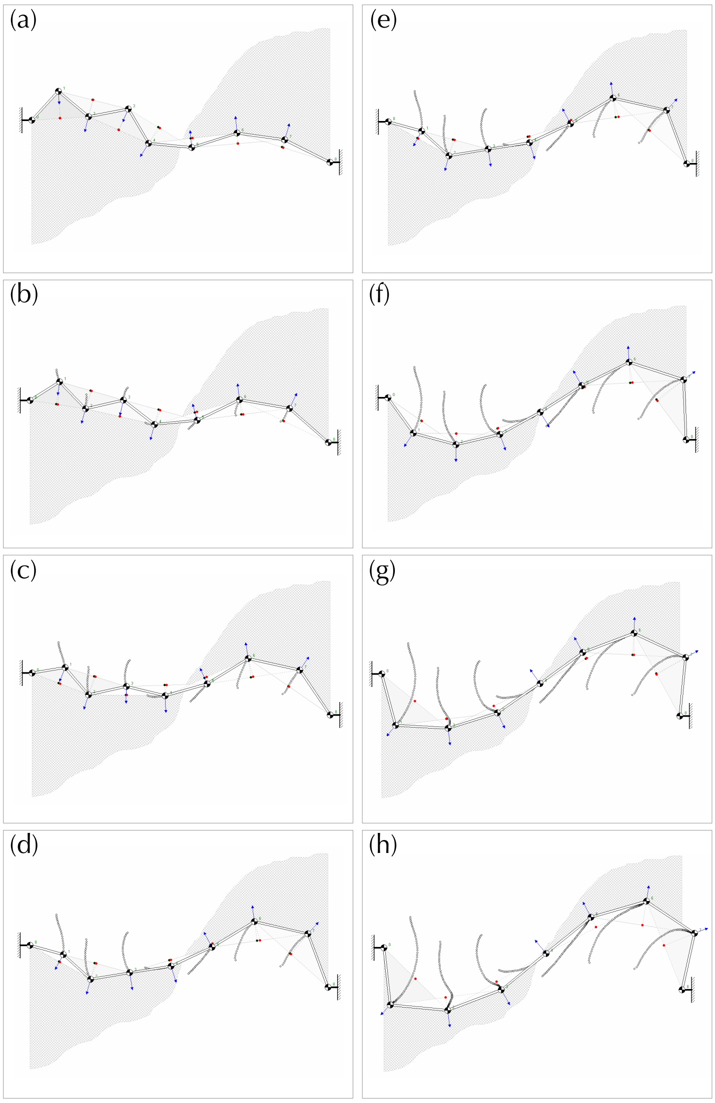

figure 9: Example simulation