Control of Contour Formations of Autonomous Vehicles by General Curve Evolution Theory

Shahab Kalantar & Uwe R. Zimmer

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

In this paper, we introduce the concept of a contour-shaped formation of autonomous underwater vehicles for the purpose of adaptation to isoclines of environmental fields created as a result of diffusion of chemicals. General curvature-driven curve evolution theory is used as a continuous model for such formations. A collection of robots can act as markers on an imaginary curve moving according to the local curvature and the external environmental force projected unto the normal to the curve. Basic control rules are given, and various strategies for implementation of the scheme are explored. Algorithms are discussed for formation initialization, and simulation runs are presented to demonstrate the behaviour of these types of formations.

1. Introduction

Using a group of mobile robots to localize, track, converge to, or enclose the source of a plume released into an environment has gained much attention in robotics communities. Often, the robots have to form special shapes dictated by the application. In this paper, we look at curve-like groups which are required to seek and remain on a particular contour around the plume source. A formation $\Phi$ of a set of n mobile robots $R_i$, with states $q_i(t) \in \mathbb{R}^n$, can be defined as a collection of geometric relationships constraining the aggregate state $q \in \mathbb{R}^m$. A formation function $\bar{V}_\phi(q)$ can be designed to capture these constraints. In a rigid formation, we only have strict equalities and $\bar{V}_\phi: \mathbb{R}^m \rightarrow \mathbb{R}$ (a scalar function) can be defined in such a way that a perfect formation would satisfy $\bar{V}_\phi(q) = 0$. The only degrees of freedom are those of translation and rotation. With deformable (non-rigid) formations, we have much more degrees of freedom. In this case, the vector-valued $\bar{V}_\phi: \mathbb{R}^m \rightarrow \mathbb{R}^r$ ($r$ is the number of constraints) would constitute a number of inequalities as well and a vector inequality such as $\bar{V}_\phi(q) \leq 0$ may describe an infinite number of acceptable formations (even when translation and rotation have been abstracted away). Formation control should keep the aggregate state on the formation manifold. Formations are also required to fulfill some sort of mission. We can represent a mission by a suitable mission function $\mathcal{M}_\phi: \mathbb{R}^n \otimes \Psi \rightarrow \mathbb{R}$, where $\Psi$ is a set of parameters. Mission is accomplished if $\mathcal{M}_\phi = 0$.

Rigid formations have been the topic of great many research papers (see, e.g., [3, 7, 8]), while deformable ones have not been addressed sufficiently (see [5, 9] for an example of research on deformable formations). In our particular application, a contour-shaped array of robots should adapt itself to a contour $\tau_{f_i}$, corresponding to a desired isocline (or level sets) $f_i$ of a distribution. The contour formation should satisfy constraints related to bending and stretching of the band, so that $\bar{V}_\phi$ is composed of a set of inequalities such as $d_1 \leq \|q_i - q_{i\pm\ell}\| \leq d_2$, $|\gamma_i - q_i| \leq d_3$, $|\kappa_i| \leq \kappa_0$, where it is assumed that the indexes define a total ordering between robots. $\kappa$ denotes curvature. Furthermore, $\mathcal{M}_\phi$ is defined as

$$\sum_{\ell=1}^\infty \|Q(\gamma_{f_i}, \kappa_i) - q_i\|,$$

where $Q(\gamma_{f_i}, \kappa)$ denotes the closest point on the level set to point $\gamma_{f_i}$. As $\ell \rightarrow \infty$, it is desired that $\mathcal{M}_\phi \rightarrow 0$, while $\bar{V}_\phi(q_i) \leq 0$ at all times. The approach we take in this paper is inspired by similar applications of deformable models in machine vision.

Environmental isoclines can be defined as contours of equal concentration created as a result of diffusion of some chemical released from a source. In its simplest form, diffusion follows Fick's law, which states

---

1. Part of this research was done under a grant from Cooperative Research Centre for Intelligent Manufacturing Systems and Technology (CRC-IMST), Australia.
that the flux of matter is a function of the gradient of the concentration of matter, i.e.,

$$J_x = -D_x \frac{\partial C(q)}{\partial x} - D_y \frac{\partial C(q)}{\partial y} \quad (2)$$

and the movement of particles of matter follows

$$\frac{\partial C(q)}{\partial t} = \frac{J_q}{\partial q} \quad (3)$$

where $C(q)$ denotes the concentration at $q \in \mathbb{R}^2$, $D_x$ and $D_y$ are coefficients of diffusion. A simpler and more realistic method for modelling plumes is to use the similarity hypothesis ([2]), which states that, as the patch is spread, the shape of the particle distribution is preserved. Thus, the mean concentration distribution would be

$$\overline{C}(q,t) = \frac{Q(t)}{\beta \sigma_x(t) \sigma_y(t)} C \left\{ \frac{x}{\sigma_x(t)}, \frac{y}{\sigma_y(t)} \right\} \quad (4)$$

where $Q(t)$ is the rate of release of particles, and $\sigma_x(t)$ and $\sigma_y(t)$ are, respectively, the standard deviations in the respective directions, and implicitly describe environmental conditions. The function $G$ determines the shape of the distribution, and $\beta$ is a constant depending on $G$. Usually, a Gaussian distribution for $G$ is assumed. In this paper, we will deal with slow diffusion processes, i.e., it is assumed that adaptation occurs much faster than the spread of the plume. Alternatively, it may be assumed that the rate of release is bigger than the rate of change of the standard deviations. If this is not the case, then, after reaching steady state, the plume will form a uniform patch. For the particular case of salinity or temperature, which we are primarily interested in, this condition is satisfied. Furthermore, we assume that the flow is almost laminar, so that we can ignore the effect of turbulence. Finally, external forces (wind, ocean flow) always affect the spatial distribution. They may counteract or augment diffusion. As will be shown later, the only information the robots need to ascend or descend the distribution towards the desired level set, is the local gradients of concentration. The minimal requirement is that this gradient be reasonably well-defined and smooth. Thus, the geometric form of the plume is irrelevant, as long as its scale is commensurate with the size of the formation. See [2] for more detailed information on plumes and environmental diffusion.

The control scheme to be presented in this paper is intended to be implemented on small submarines called Serafina, developed at our laboratory. Figure 1 shows this vehicle from different views. We confine ourselves to motion on the plane. As shown in figure 2, roll and pitch controllers try to neutralize the effect of disturbances on roll and pitch angles. The heave controller keeps the robot at a certain desired depth $D_h$, relative to the ocean bottom. Motion of the submarine on the $x$-$y$ plane is thus similar to a non-holonomic vehicle because swaying is not directly controllable. The yaw and surge controller guides the robot on the plane to desired locations. See [10] for details on dynamic modelling and decoupled control of marine vehicles.
2. Deformable contour formations

Before discussing actual formations, let us study the case of an idealized smooth curve which can be regarded as a continuum model for contour formations. For more details and arguments, see [4] and references there in. Let \( \gamma : [0, 1] \otimes \mathbb{R} \rightarrow \mathbb{R}^2 \) define a moving curve on the plane, parametrized by arc-length \( s \). Thus \( \gamma(s, t) = (x(s, t), y(s, t))^T \) gives the position in \( \mathbb{R}^2 \) of points constituting the curve. The general curve evolution theory is based on the fact that the flow

\[
\frac{\partial \gamma(s, t)}{\partial t} = \kappa(s, t) \bar{N}(s, t)
\]

moves the curve in the direction of the gradient of \( L(\gamma) = \int_0^1 ds \), i.e., reduces the length as fast as possible and is called the Euclidian curve shortening flow, where

\[
\kappa(s, t) = \frac{y_{ss}(t)x_{st}(t) - x_{ss}(t)y_{st}(t)}{\sqrt{(x_{st}(t))^2 + (y_{st}(t))^2}^3}
\]

is the Euclidian curvature and

\[
\bar{N}(s, t) = \frac{1}{\sqrt{x_{st}(t)^2 + y_{st}(t)^2}} \begin{pmatrix} y_{st}(t) \\ -x_{st}(t) \end{pmatrix}
\]

is the unit inward normal to the curve at \( \gamma(s, t) \). Adaptation of \( \gamma \) to a desired level set \( \mathcal{F} \) of the diffused plume \( \mathcal{F}(q) \) can be formulated as the minimization of the functional

\[
\mathcal{E}(\gamma) = \int_0^1 g(\gamma(s))ds
\]

where \( g : [0, max_q \mathcal{F}] \rightarrow \mathbb{R}^+ \) is a strictly decreasing function, i.e. \( g(q) \rightarrow 0 \) as \( \mathcal{F}(q) \rightarrow \mathcal{F}_0 \). \( g \) is related to \( \mathcal{F} \) through the definition \( g(q) = h_{\mathcal{F}, q} \mathcal{F}(q) \). \( h_{\mathcal{F}, q} \) serves to stop the curve from evolving past the desired iso-cline and a good choice for it is

\[
h_{\mathcal{F}, q}(\zeta) = 1 - \frac{1}{\sigma^2} e^{-\frac{(\zeta - \mathcal{F}(q))^2}{\sigma^2}}
\]

The gradient descent of \( \mathcal{E}(\gamma) \) gives the flow

\[
\frac{\partial \gamma(s, t)}{\partial t} = \nabla g(\gamma(s, t)) \kappa(s, t) \bar{N}(s, t)
\]

where

\[
\nabla g(q) = \nabla \mathcal{F}(q) h_{\mathcal{F}, q}(\mathcal{F}(q)) \nabla \mathcal{F}(q)
\]

In the discrete implementation of the contour evolution, where the curve is composed of a finite number of nodes (as is the case in a robotic contour formation), moving towards the normal to the curve will cause the nodes to come very close to each other. On the other hand, in an open contour, it would be desirable to keep the length of the curve fixed, which is not enforced by normal motion alone. To remedy these problems, some sort of attractive-repulsive forces are required. It can be proved that if these forces act towards the tangent to the curve, the geometry of the curve will not be affected. It will only change the parametrization in the continuum model. This will lead to a more general form of curve evolution

\[
\frac{\partial \gamma(s, t)}{\partial t} = \eta_\alpha(s, t) \bar{N}(s, t) + \eta_\tau(s, t) \hat{T}(s, t)
\]

where

\[
\hat{T}(s, t) = \frac{\partial \gamma(s, t)/ds}{\|\partial \gamma(s, t)/ds\|} = \gamma(s, t)^\perp
\]

is the tangent to the curve at \( \gamma(s, t) \). The non-zero tangential component \( \eta_\tau \) can be designed according to desired behaviour. For a closed curve, we use the boundary condition \( \gamma(1, t) = \gamma(0, t) \). For open curves, the end points are kept fixed in time \( \gamma(0, t) = \gamma(1, t) = \theta \).

Based on the above continuum model for curve motion under the influence of an external force, we now proceed to define control laws for the motion of robots \( R_s \). Let us set

\[
\frac{\partial \mathcal{F}(q)}{\partial t} = \eta_{\mathcal{F}}(q, t) \bar{N}_s(q, t) + \eta_{\tau}(q, t) \hat{T}_s(q, t)
\]

where

\[
\eta_{\mathcal{F}}(q, t) = \beta_1 \kappa(q, t) g(q, t)
\]

\[
- \beta_2 (h_{\mathcal{F}}(\mathcal{F}(q, t))) \nabla_{\mathcal{F}(q, t)} \mathcal{F}(q, t)) \bar{N}_{\mathcal{F}(q, t)}\mathcal{F}(q, t))
\]

The normal, and tangent vectors, and the curvature are defined, respectively, by

\[
\bar{N}_s(q, t) = \begin{pmatrix} (y_{s+1}(q, t) - y_{s-1}(q, t)) \\ -y_{s+1}(q, t) + x_{s+1}(q, t) \end{pmatrix} / \|q_{s+1}(q, t) - q_{s-1}(q, t)\|
\]

\[
\hat{T}_s(q, t) = q_{s+1}(q, t) - q_{s-1}(q, t) / \|q_{s+1}(q, t) - q_{s-1}(q, t)\|
\]

\[
\kappa(q, t) = 4 \|q_{s+1}(q, t) - q_{s-1}(q, t)\|^{3} / \|q_{s+1}(q, t) - q_{s-1}(q, t)\|
\]
\[\ddot{\kappa}(t) = \left( y_{i-1}(t) - x_{i-1}(t) \right) (x_{i+1}(t) - x_{i-1}(t)) \]
\[\quad - \left( x_{i-1}(t) - 2x_{i}(t) + x_{i+1}(t) \right) (y_{i+1}(t) - y_{i-1}(t)) \]

To maintain a rigid inter-robot distance \( d = L/N \), where \( L \) is a desired contour length, we define
\[\eta_1(t) = \frac{\beta_i e^{|\beta_i - \eta_i| (1 - d/N)^2} - e^{|\beta_i + \eta_i| (1 - d/N)^2}}{p_i(t) = \|q_i(t) - q_{i-1}(t)\| + \|q_i(t) - q_{i+1}(t)\|} \]

The end robots, in an open contour formation, have to be treated differently because the curvature is not defined at these two positions. Moreover, the normal direction is ill-defined. In [2], we let the two end robots escape stagnation when \( V_{q_i}(0) \bar{F}(q_i(t)) \) becomes almost co-linear with \( T_0 \) (and similar for \( q_i \)).

### 3. Implementation and hybrid modelling

When implementing the control scheme described in the previous section, several possibilities can be considered. We have assumed that each robot can measure the gradient at its position so that the only external information needed by each robot are the positions of its two neighboring robots with respect to its body-fixed local coordinate system. This information can be gained through active sensing, explicit communication, or, more realistically, a combination of both. There is an inevitable delay associated with this process. Figure 4.a shows a simple model for communication channels between adjacent robots. In a fully synchronized scheme, the robot formation implements a discrete version of equation (14): at each discrete instant \( t \), all the robots start synchronizing on shared information, while keeping station, before moving to the next location given by
\[\dot{q}_{i+1}^d = q(t) + \Delta t(\eta_1(t) N(t) + \eta_1(t) T(t)) \]

using the linear control rule
\[\dot{q}(t) = \nu \frac{q_{i+1}^d - q(t)}{\|q_{i+1}^d - q(t)\|} \]

where \( \nu \) ensures that \( q_{i+1}^d \) is reached in \( \Delta t \). In practice, synchronization takes some time, so that there will be intervals of inaction in between successive movements. \( \Delta t \) is selected based on the characteristics of the differential equation ([9]) and should, in general, be small enough. Figure 4.b shows the timed automaton for the synchronization phase. In a fully asynchronous scheme, we have
\[\dot{q_i}(t) = \Xi(q_i(t), q_{i-1}(t - \tau_i(t)), q_{i+1}(t - \tau_{i+1}(t))) \]

where \( \Xi \) denotes the curve evolution equation, and \( \tau_i(t) \) and \( \tau_{i+1}(t) \) denote, respectively, the times (measured from \( t \)) when the positions \( \dot{q}_{i-1} \) and \( \dot{q}_{i+1} \) were updated. If \([t \tau_i(t) - \tau_{i+1}(t)] \), \( t - \tau_i(t), \) and \( t - \tau_{i+1}(t) \) are within reasonable bounds, equation (26) would be nearly as good as equation (24). The bounds should be determined through analysis and, in general, depend on the speed of individual robots. If the bounds can become relatively large, we can extrapolate the motions of \( R_{i-1} \) and \( R_{i+1} \) using the laws

\[\hat{N}_{i+1}(t) = \frac{1}{2} \left( \hat{N}_{i+1}(t) + \frac{\hat{N}_{i+1}(t) - \hat{N}_{i+1}(t)}{\|\hat{N}_{i+1}(t) - \hat{N}_{i+1}(t)\|} \right) \]

Note that this may hinder the formation from adapting but it will ensure its integrity. See figure 3 for a visual demonstration of vectors. This particular choice may be problematic with very large curvatures. On the other hand, it may help the end robots escape stagnation when \( V_{q_i}(0) \bar{F}(q_i(t)) \) becomes almost co-linear with \( T_0 \) (and similar for \( q_i \)).
\[ \hat{q}_{i-1}(t) = q_{i-1}(t - \tau) + (t - \tau)\dot{q}_{i-1}(t - \tau) \]
\[ \hat{q}_{i+1}(t) = q_{i+1}(t - \tau) + (t - \tau)\dot{\hat{q}}_{i+1}(t - \tau) \]
where \( \ddot{q}_{i-1} \) and \( \ddot{\hat{q}}_{i+1} \) denote the estimated velocities at the time when \( q_{i+1} \) and \( q_{i-1} \) were last updated. It is desirable to bound
\[
\|q_i(t) - \hat{q}_i(t)\| = \int_0^t [\dot{q}_i(t) - \dot{\hat{q}}_i(t)] dt
\]
\[
\leq \int_0^t [\dot{q}_i(t) - \dot{\hat{q}}_i(t)] dt
\]
\[
\leq \int_0^t \left( \|\eta_j N_i(t) - \tilde{\eta}_j N_i(t)\| + \|\eta_j \ddot{N}_i(t) - \tilde{\eta}_j \ddot{N}_i(t)\| \right) dt
\]
where \( t_0 = \min\{\tau, \tau_i\} \), \( \hat{q}_i(t) \) is the position of \( R_i \) at time \( t \) using the rule
\[
\hat{q}_i(t) = \Xi(\hat{q}_i(t), q_{i-1}(t), \hat{q}_{i+1}(t))
\]
Moreover,
\[
\ddot{\hat{N}}_i(t) = -\begin{bmatrix} 1 & 0 \end{bmatrix} \dddot{\hat{N}}_i(t) \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 \end{bmatrix} \dddot{\hat{N}}_i(t) \begin{bmatrix} 1 \\ 0 \end{bmatrix}
\]
Qualitatively speaking, \( \eta_j \) and \( \eta_j \) try to keep \( q_i \), \( q_{i-1} \) and \( q_{i+1} \) within certain bounds. Intuitively enough, a reasonable way of bounding \( \|q_i(t) - \hat{q}_i(t)\| \) is to bound \( \|\hat{q}_i(t) - \hat{q}_{i-1}(t)\| \), \( \|\hat{q}_i(t) - \hat{q}_{i+1}(t)\| \) and \( \|\hat{q}_{i+1}(t) - \hat{q}_{i-1}(t)\| \) thus preventing a suitable defined energy function to fall below a certain threshold. This denotes an event-triggered synchronisation, as an alternative to traditional time-triggered synchronisation. As long as certain bounds are satisfied, the linear interpolation scheme is a good approximation. In practice, even if the above bounds are satisfied, deviation of \( v_{i-1}(t) \) and \( v_{i-1}(t) \) from their estimated values at times \( t - \tau \) and \( t - \tau \) necessitate synchronization after some time. To decide when this should happen, we can test for the violation of the bounds
\[
(\ddot{v}_{i-1} + \epsilon)(t + \Delta t - \tau) \leq \delta \]
\[
(\ddot{v}_{i-1} + \epsilon)(t + \Delta t - \tau) \leq \delta
\]
where \( \epsilon \) is a small positive constant. Figure 4.b shows the hybrid automaton modelling this semi-asynchronous scheme.

4. Formation initialization

The robots are initially scattered throughout a region with no definite structure (randomly distributed). Assume that a circular neighborhood \( N_i(q_i(t)) \), with radius \( r \), is defined around each robot \( R_i \). This neighborhood can designate the limited communication range of the robots. \( R_i \in N_i(q_i(t)) \) if and only if \( \|q_i(t) - q_i(t)\| \leq r \). A connection graph \( G \) can be constructed in which nodes correspond to robots and an edge \( e_i \) is present between two nodes \( R_i \) and \( R_j \) if \( R_i \in N_i(q_i(t)) \) and \( R_j \in N_j(q_j(t)) \). We will assume that the initial graph is connected. Before adaptation is attempted, the robots have to self-organize into a formation suitable the problem at hand, i.e., a roughly straight line in the case of open contours and a circle in the case of closed ones. In [6], algorithms are proposed for these two cases. Here, we will adapt their method to the case of real robots with dimensions (instead of points). We also slightly modify and reformulate the original versions. First, note that it is desirable to bound the differentiation of \( q_i(t) \) by \( \|\dot{q}_i(t)\| \leq \Delta \).
Given a vector $\mathbf{r}$ located at the origin. Moreover, assume that each robot has the means to measure the distances and bearings to the robots in its neighborhood. No two robots are allowed to come closer than a distance $D_M$ to each other. Also, let $L$ denote the desired inter-robot distance between a pair of robots on the final line formation. Implementing the proposed algorithm, the final line would spread along the $X$ axis of a global coordinate system $\mathcal{Y}$ whose $Y$ axis is aligned with the north. The position of the $X$ axis along $Y$ direction would be the mean of the $Y$ coordinates of all the robots, if measured with respect to this global system.

We define three virtual sensors $s_R$, $s_V$, $s_{\ell}$, and $s_{\ell+}$ as follows. $s_R$ indicates if the path of the robot is obstructed in the easterly direction. The other sensors serve the same purpose for west, north, and south directions, respectively. More formally:

**Definition 1:** $s_R = 1$ for $R$, if there exists a $R_i \in \mathcal{N}(q_i(t))$, $R_i \neq R$, such that

$$\|q_i(t) - q_j(t)\| \leq D_M,$$

$$\|q_i(t) - q_j(t)\| = \inf_{R_i \in \mathcal{N}(q_i(t))} \|q_i(t) - q_j(t)\|$$

and

$$|\theta_j| = \frac{1}{\chi - \chi} |(\frac{y_j}{x_j} - \frac{y_j}{x_j})| \leq \theta_S$$

where $\theta_S = \theta_S$. $S_R = 0$ otherwise. $\theta_S$ is an angle determining the degree of openness in a direction. To avoid collisions, $\theta_S$ should be bigger that $\pi/4$. The other sensors are similarly defined. For $s_{\ell}$ we define $\theta_{\ell} = \pi - \theta_S$. Also, $\theta_{\ell+} = \pi/2 - \theta_S$, and $\theta_{\ell+} = 3\pi/2 - \theta_S$. We denote $R_i$ by $R_i^\chi$, where $\chi \in \{L, R, V, V+, V-\}$.

**Figure 5:** Notations for formation initialization

**Figure 6:** Automaton for formation initialization. $M$ is composed of two automata, run in parallel, for motions in $x$ and $y$ directions. States are named according to the states of sensors. Also, $\xi = \gamma_i, (\|\theta_{\mathcal{N}(q_i)} - y_i\|)\gamma_i$ and $\xi = \gamma_v, (\|\theta_{\mathcal{N}(q_v)} - y_v\|)\gamma_v$

Now, let each robot move according to the following control rule:

$$\dot{\theta}_j(t) = [\dot{x}_j(t), \dot{y}_j(t)]^T$$

$$\dot{x}_j(t) = (s_L\theta_R + s_P\theta_L)(\|d_i - d_j\|)\phi(d_i, d_j)$$

$$\dot{y}_j(t) = s_V\phi(\theta_{\mathcal{N}(q_j)}(t), y_j(t))$$

$$\dot{x}_j(t) = 1 - 2\gamma_0(t)\xi_j$$

$$\dot{y}_j(t) = (\|\theta_{\mathcal{N}(q_j)} - y_j\|)\gamma_0(\|\theta_{\mathcal{N}(q_j)} - y_j\|)\gamma_v$$
where
\[ \bar{y}_{\mathcal{N},(q_i)}(t) = \frac{1}{n} \mathbb{E}_r \sum_{i=1}^{n} y_i(t) \] (39)
is the Y coordinate of the centre of gravity of robots in \( \mathcal{N},(q_i) \),
\[ s_Y = z_v, \bar{\delta}(\bar{y}_{\mathcal{N},(q_i)}(t), y_i(t)) + s_v, \Phi(\bar{y}_{\mathcal{N},(q_i)}(t), y_i(t)) \]
\( n \) denotes the boolean negation of the boolean variable \( u \), \( \omega(t) \) is a binary random variable, \( q_i(t) = k_{q_i}(t) - q_i^v(t) \), \( d_i = \|k_{q_i}(t) - q_i^v(t)\| \), \( q_i^v(t) \) and \( q_i^v(t) \) being the positions of the robots on the left and right of \( R_i \). \( \Phi(u_1, u_2) = 1 \) if \( u_1 < u_2 \) and 0 otherwise. \( v_v \) and \( \nu_v \) are nominal constant speeds in the respective (east-west, south-north) directions. \( \eta(u) \) monotonically goes to zero as \( u \to 0 \). \( \gamma_e(u) \) is a monotonic function which increases as \( u \to 0 \). Likewise, \( \gamma_v(u) \) is a monotonically decreasing function as \( u \to 0 \). The intuition behind the use of \( \gamma_e \) and \( \gamma_v \) is that as the robot approaches the line, its horizontal velocity should decrease while its vertical speed should increase. This cooperative behaviour will help the faster convergence of the robots to the line.

At each instant of time, the error of the formation is defined in [6] as
\[ q_e^v(t) = \sum_{i=0}^{N-1} q_e^v(t) = \left( \begin{array}{c} x_e(t) \\ y_e(t) \end{array} \right) = \bar{q}(t) + \left( \begin{array}{c} N + 2i - 1 \\ 0 \end{array} \right) - q_i(t) \] (41)
Note that the line formation strategy can also be used to form lines rotated with respect to Y. For initialization of closed contour formations, the robots should form a circle with an appropriate radius (say \( r_c \)) around the centroid of the initial assembly. In this paper, as we are focusing on open contours, we will not elaborate on realistic control laws for this case. Assuming that robots are point particles and collisions are not an issue, the following control rule will achieve the circular formation:
\[ \dot{q}_e(t) = \left( \begin{array}{c} \dot{x}_e(t) \\ \dot{y}_e(t) \end{array} \right) = \left( \begin{array}{cc} \cos(\theta_e(t)) & -\sin(\theta_e(t)) \\ \sin(\theta_e(t)) & \cos(\theta_e(t)) \end{array} \right) \right) \times \left( \begin{array}{c} r_c \\ \phi(\theta_e(t)+\theta_v(t)) \end{array} \right) \] (42)
where \( \nu_r \) and \( \nu_v \) denote, respectively, nominal radial and tangential speeds,
\[ \bar{q}(t) = \frac{1}{N} \sum_{i=0}^{N-1} q_i(t) \] (43)
\( t_0 \) being the initial time.

5. Simulation results

First, we present a simulation run for formation initialization. It should be noted that the success of this process depends on the parameters involved: the radius of the neighborhood \( r \), the desired final distance between robots in the final formation \( L \), the minimum allowable distance between two robots \( D \),
the nominal speeds $v_x$ and $v_y$, the rates of change of $\gamma_x$ and $\gamma_y$, and the random variable $\omega$. $r$ should be large enough to ensure connectivity and convergence. In our experiments, it was found that $L$ should be sufficiently larger that $D$. Setting $L = 2D$ gives good results. $\gamma_x$ continuously goes from $1/4$ to $1$, as $\|P_{N(q_t)} - y\|$ goes from $r$ to $0$. Likewise, $\gamma_y$ continuously goes from $1$ to $1/4$, as $\|P_{N(q_t)} - y\|$ goes from $r$ to $0$. To prevent the robot motion from jittering when close to the line, the rate of change of $\omega(t)$ should be small. To make this happen, it is enough to retain the last value of the random variable for some time, i.e., the robot will move according to the value of $\omega(t)$, determined at time instant $t$, for the duration of $\left[ t, t + \delta t \right]$. Figure 7 shows the sequence of motions of an initial random aggregate.

Figure 8 shows the sequence of snapshots of the adaptation of a group of robots to a level set of an imaginary field. The red circles show the sites from which the plume has been released.

Figure 9.a shows the adaptation of a group of robots which had previously initialized into a line oriented with respect to the major axis of the field. Figure 9.b shows the adaptation when the initial line overlaps with the field. This latter case is the most problematic. The final shape depends on the shape of the plume and is very difficult to determine beforehand.

6. Conclusions and future research

We showed that a formation shaped as a contour can adapt itself to areas of equal concentration produced when some chemical is diffused into an environment. The proposed scheme is scalable and there is no need for a global reference frame. The measurements and decision making are with respect to local coordinate systems fixed on each robot. These coordinates are aligned with the tangent to the curve and the normal to it. We also discussed synchronous and semi-synchronous implementation methods, as well as algorithms for the initialization of a random aggregate into suitable shapes. There are still some issues which have to be addressed. We assumed that each robot can measure the local gradient at its position. A more realistic approach would be to design contour formations of rigid formations, instead of single vehicles, which can collectively estimate the gradient. It was assumed that the environment is obstacle-free which is rarely the case. Finally, the motion of the formation is confined to a plane parallel to the ocean bottom. Studying motion constrained to surfaces as well as in three dimensions are among possible extensions to the work presented here.

References