

# Tracking Environmental Isoclines using Polygonal Formations of Submersible Autonomous Vehicles

Shahab Kalantar, Uwe Zimmer

Australian National University  
Department of Information Engineering<sup>1</sup>

Canberra, ACT 0200, Australia  
[shahab.kalantar@rsize.anu.edu.au](mailto:shahab.kalantar@rsize.anu.edu.au), [uwe.zimmer@ieee.org](mailto:uwe.zimmer@ieee.org)

Knowledge of iso-contours of the underwater terrain can be used to reconstruct it using interpolation. Identifying a set of isoclines can be more efficient and less time-intensive than sweeping a large area. In this paper, we propose a system where a small number of agile underwater vehicles cooperatively maintain a polygonal formation on a plane above the terrain and use field values measured by the individual robots to locally reconstruct the field using interpolation schemes. The formation then tracks a desired iso-contour of the field by tracking the corresponding curve on the reconstructed field.

## 1. Introduction

Traditionally, to create a bathymetric map of the underwater terrain employing AUV's, a single bulky vehicle (moving at a certain depth) is towed to follow a path composed of a set of parallel transects (covering a large area) and record altitude values. The recorded information is then used to approximate the field in that region (see, e.g., [3]). An alternative approach is that the vehicle autonomously guide itself along lines of constant altitude. A set of these recorded iso-contours can then be used to reconstruct the field using Hermite interpolation [14]. An obvious benefit of such an approach is that only regions with high *variability* need to be covered. Given the local gradient and Hessian, this is an easy task to do. One only needs to climb the gradient path and follow the direction orthogonal to field gradient once at the desired isocline. In reality, though, these information are not readily available (have to be estimated) and are sometimes even meaningless. In [5], a single vehicle uses a history of past field values. In [6], UUV-gas theory is used where we only need to determine whether we are inside or outside the region surrounded by the iso-contour. A similar approach is described in [7]. In [4],

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

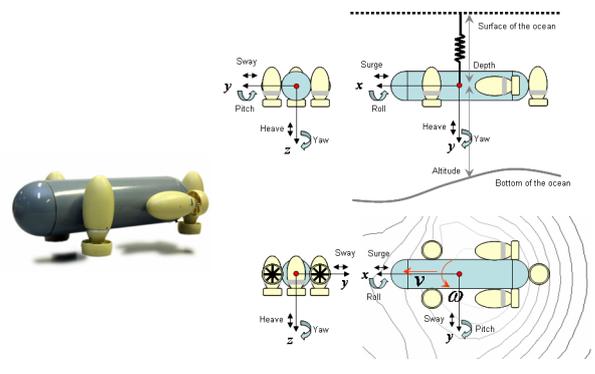


figure 1: Serafina underwater vehicle.

the trajectory of a vehicle is modeled as a damped sinusoid whose parameters are adjusted to stabilize the vehicle on the contour. Rather than using a single vehicle, we can employ a formation of communicating vehicles. This way, local characteristics of the field can be computed more easily [9]. In [8], a four-vehicle system is described which estimates the gradient and the curvature to track a contour and adjust its size to minimize estimation error.

In this paper, we propose a polygonal formation and interpolation for local reconstruction based on only static instantaneous measurements of the field at the positions of the robots. The interesting aspect of this method is that the representation of the desired isocline in the reconstructed field can be used to convert the problem of tracking tangent to the true isocline to that of tracking a well-defined path. This is because, at each particular time instant, a reconstructed *picture* of the field is made available which contains a smooth curve representing the isocline. Accordingly, robust methods for tracking such curves can be employed.

As models for our physical platforms, we use Serafina [2], a small fast five thruster vehicle equipped with a suite of sensors (figure 1). Ongoing research is aiming at low-bandwidth short-range long-wave radio communication [12], optical communication [10], scheduling strategies

[11], and acoustic range and bearing measurement [13]. In this paper, we will not be concerned with these issues. Furthermore, we will consider a very simple motion model for the vehicles, i.e., that of a non-holonomic unicycle moving on the plane:

$$\dot{\mathbf{y}}(t) = \begin{bmatrix} \cos\theta(t) & 0 & 0 \\ 0 & \sin\theta(t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ v(t) \\ \omega(t) \end{bmatrix} \quad (1)$$

where  $\mathbf{y}(t) = [x(t), y(t), \theta(t)]^T$  is the state of the robot, and  $v(t)$  and  $\omega(t)$  are, respectively, desired linear and angular velocity inputs of the yaw-surge controller. Additionally, a heave controller keeps the vehicle at the desired depth and the roll and pitch controllers maintain these angles at zero. We also assume that the ambient flow is negligible or is counteracted by the controller.

## 2. Initializing and maintaining polygonal formations

In this section, we explain how a polygonal formation of co-planar vehicles can be created and maintained. Given a desired radius  $r_p$ , a deformation of the corresponding regular polygonal formation  $\mathbf{P}$ , composed of  $N$  vehicles  $\mathbf{R}_i$ ,  $i = 0, \dots, N-1$ , with planar positions, denoted by  $\mathbf{q}_i(t) = (x_i(t), y_i(t))^T \in \mathfrak{R}^2$ , and collective state  $\mathbf{q}_p(t) = [\mathbf{q}_0(t), \dots, \mathbf{q}_{N-1}(t)]^T$ , can be defined by the formation function [15]

$$\mathbf{G}_p(\mathbf{q}_p(t)) = (\tilde{\mathbf{x}}(t) - r_p \hat{\mathbf{1}}_N)^T (\tilde{\mathbf{x}}(t) - r_p \hat{\mathbf{1}}_N) + \left( \mathfrak{g}(t) - \frac{2\pi}{N} \hat{\mathbf{1}}_N \right)^T \left( \mathfrak{g}(t) - \frac{2\pi}{N} \hat{\mathbf{1}}_N \right) \quad (2)$$

where

$$\mathbf{x}_i(t) = \mathbf{q}_i(t) - \bar{\mathbf{q}}(t), \quad \tilde{\mathbf{x}}_i(t) = \|\mathbf{x}_i(t)\|, \quad (3)$$

$$\tilde{\mathbf{x}}(t) = [\tilde{\mathbf{x}}_0(t), \dots, \tilde{\mathbf{x}}_{N-1}(t)]^T, \quad (4)$$

$$\mathfrak{g}_i(t) = \cos^{-1} \left( \frac{\mathbf{x}_i(t) \cdot \mathbf{x}_{i+1}(t)}{\|\mathbf{x}_i(t)\| \|\mathbf{x}_{i+1}(t)\|} \right), \quad (5)$$

$$\mathfrak{g}(t) = [\mathfrak{g}_0(t), \dots, \mathfrak{g}_{N-1}(t)]^T. \quad (6)$$

$\bar{\mathbf{q}}(t)$  is the centre of mass and  $\hat{\mathbf{1}}_N$  is a vector of size  $N$  with all the entries equal to 1. The operations on the indexes are modulus  $N$ , so that  $\mathbf{q}_N = \mathbf{q}_0$ . Any root of  $\mathbf{G}_p(\mathbf{q}_p(t)) = 0$  gives a polygonal formation which is unique up to translation and rotation. More technically,

$$\mathbf{G}_p^{-1}(0) / (\text{SO}(2) \otimes \mathfrak{R}^2) \quad (7)$$

is a singleton. The position of the centre of mass  $\bar{\mathbf{q}}(t)$  breaks the translational ( $\mathfrak{R}^2$ ) symmetry and the angle

$$\theta_0(t) = \text{atan} \left( \frac{y_0(t) - \bar{y}(t)}{x_0(t) - \bar{x}(t)} \right) \quad (8)$$

(which can be defined as the orientation of the formation) breaks the rotational ( $\text{SO}(2)$ ) symmetry. In the following,

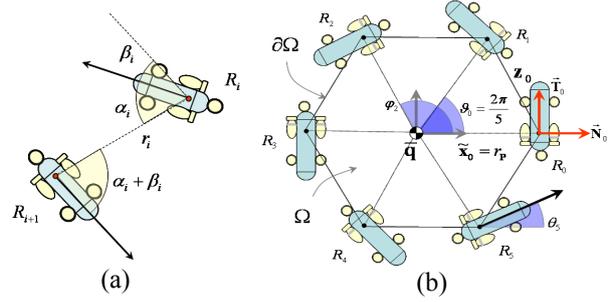


figure 2: (a) Cyclic pursuit for creating regular polygons. (b) Maintaining the polygon.

we use the notation  $\mathbf{n}(\hat{v}) = \hat{v} / \|\hat{v}\|$ . Letting  $B_\varepsilon(u)$  denote an  $\varepsilon$ -ball around  $u \in \mathfrak{R}^N$ , an  $\varepsilon$ -neighborhood of a formation is defined by  $B_\varepsilon(\mathbf{G}_p^{-1}(0))$ . A number of control strategies have been proposed in the literature for creating a polygonal formation from an arbitrary aggregate of robots. In [17], cyclic pursuit strategy is proposed for obtaining a regular polygon at equilibrium. In this scheme, each robot  $\mathbf{R}_i$  pursues  $\mathbf{R}_{i+1}$  with appropriate control inputs  $v_i$  and  $\omega_i$ . Figure 2(a) shows the variables used in this context. The evolution of these variables is given by

$$\dot{r}_i = -v_i \cos \alpha_i - v_{i+1} \cos(\alpha_i + \beta_i), \quad (9)$$

$$\dot{\alpha}_i = \frac{1}{r_i} (v_i \sin \alpha_i + v_{i+1} \sin(\alpha_i + \beta_i)) - \omega_i, \quad (10)$$

$$\dot{\beta}_i = \omega_i - \omega_{i+1}. \quad (11)$$

It is proved in [17] that if the control variables are given by  $v_i = k_v r_i$  and  $\omega_i = k_\omega \alpha_i$ , where  $k_\omega = 1$  and  $k_v = \pi \csc(\pi/N) / 2N$ , then the equilibrium state of the system would be a regular polygon with  $N$  vertices. The equilibrium values for  $\alpha_i$  and  $\beta_i$  are  $\bar{\alpha} = \pi/N$  and  $\bar{\beta} = \pi - 2\pi/N$ . Also  $\bar{r}$  is a constant and the internal angle at each vertex is given by  $\pi(1 - 2/N)$ .

Once the vehicles have converged to a sufficiently small neighborhood of a polygonal formation, they enter a formation keeping phase. The control rules for this phase have to make sure that the formation does not deviate from an ideal polygon too much, or, in other words, it stays within  $B_\varepsilon(\mathbf{G}_p^{-1}(0))$ . We will use the concept of *virtual leaders* proposed in [15]. We design the desired position for the non-holonomic vehicle  $\mathbf{R}_i$  as

$$\mathbf{q}_{i_d}(t) = p_i(s_i), \quad \dot{\mathbf{q}}_{i_d}(t) = \frac{\partial}{\partial s_i} p_i(s_i) \dot{s}_i, \quad (12)$$

where

$$\frac{\partial}{\partial s_i} p_i(s_i) = -\nabla_{\mathbf{q}_i(t)} \mathbf{G}_p(\mathbf{q}_p(t)) + \beta(\bar{\mathbf{q}}_d(t) - \bar{\mathbf{q}}(t)). \quad (13)$$

$p_i(s_i)$  denotes the trajectory of  $\mathbf{q}_{i_d}(t)$ , parametrized by  $s_i$ .  $\partial p_i(s_i) / \partial s_i$  is the tangent to the trajectory.  $\dot{s}_i$  is the speed of traversal of the trajectory.  $\mathbf{q}_{i_d}(t)$  can be regarded as the virtual leader of  $\mathbf{R}_i$ .  $\bar{\mathbf{q}}_d(t)$  is the desired position of the centre of mass and should be designed according to application. It plays the role of the virtual leader for the whole formation. Note that this control rule effectively decouples

formation keeping and maneuvering which is very desirable. The original control rule proposed in [15] is

$$\frac{\partial}{\partial s_i} p_i(s_i) = -\nabla_{\mathbf{q}_i(t)} \tilde{\mathbf{G}}_{\mathbf{P}}(\mathbf{q}_{\mathbf{P}}(t)), \quad (14)$$

where  $\tilde{\mathbf{G}}_{\mathbf{P}}$  is the same as  $\mathbf{G}_{\mathbf{P}}$  but  $\bar{\mathbf{q}}(t)$  is replaced with  $\bar{\mathbf{q}}_d(t)$ . This non-decoupled strategy may not be appropriate for every kind of formation function. Unless  $\|\bar{\mathbf{q}}_d(t) - \bar{\mathbf{q}}(t)\|$  is very small or the speed is very low, the path taken by individual robots may not be collision-free.

As an expression for  $-\nabla_{\mathbf{q}_i(t)} \mathbf{G}_{\mathbf{P}}(\mathbf{q}_{\mathbf{P}}(t))$ , we can use the resultant of a normal and a tangential motion. Thus, the dynamics of the whole formation can be concisely represented as (see figure 2(b))

$$\begin{aligned} \frac{\partial}{\partial s} p(s) &= \beta_1 \tilde{\eta}(r_{\mathbf{P}} \mathbf{1}_{N-1} - \tilde{\mathbf{x}}(t)) \bullet \tilde{\mathbf{N}} + \\ &\beta_2 \tilde{\eta}(\mathbf{z}(t) - s_1(\mathbf{z})) \bullet \tilde{\mathbf{T}} + \beta_3 \tilde{\eta}(\|\bar{\mathbf{q}}_d(t) - \bar{\mathbf{q}}(t)\|) \mathbf{1}_N \end{aligned}, \quad (15)$$

where the black circle denotes Hadamard product and

$$\frac{\partial}{\partial s} p(s) = \left[ \frac{\partial}{\partial s_0} p_0(s_0), \dots, \frac{\partial}{\partial s_{N-1}} p_{N-1}(s_{N-1}) \right]^T, \quad (16)$$

$$\tilde{\mathbf{N}} = [\tilde{\mathbf{N}}_{\mathbf{P}}(\tau_0(t)), \dots, \tilde{\mathbf{N}}_{\mathbf{P}}(\tau_{N-1}(t))]^T, \quad (17)$$

$$\tilde{\mathbf{T}} = [\tilde{\mathbf{T}}_{\mathbf{P}}(\tau_0(t)), \dots, \tilde{\mathbf{T}}_{\mathbf{P}}(\tau_{N-1}(t))]^T, \quad (18)$$

$$\tilde{\eta}([u_0, \dots, u_{N-1}]) = [\eta(u_0), \dots, \eta(u_{N-1})]^T, \quad (19)$$

$$\mathbf{z}_i(t) = \|\mathbf{q}_i(t) - \mathbf{q}_{i+1}(t)\|, \quad (20)$$

$$\mathbf{z}(t) = [\mathbf{z}_0(t), \dots, \mathbf{z}_{N-1}(t)]^T. \quad (21)$$

$s_1(\mathbf{z})$  is the shift operator, whose output is defined by  $[s_1(\mathbf{z})]_0 = \mathbf{z}_{N-1}(t)$ ,  $[s_1(\mathbf{z})]_1 = \mathbf{z}_0(t)$ , etc. Also,

$$\tilde{\mathbf{N}}_{\mathbf{P}}(\tau_i(t)) = \tilde{\mathbf{N}}_i = \mathbf{n}(\mathbf{q}_i(t) - \bar{\mathbf{q}}(t)), \quad (22)$$

$$\tilde{\mathbf{T}}_{\mathbf{P}}(\tau_i(t)) = \tilde{\mathbf{T}}_i = \mathbf{n}(\mathbf{q}_{i+1}(t) - \mathbf{q}_{i-1}(t)), \quad (23)$$

$$\tau_i(t) = \frac{\varphi_i(t)}{2\pi}. \quad (24)$$

Note that  $0 \leq \tau_i(t) \leq 1$  is a parametrization of the circle and  $\varphi_i(t)$  denotes the angle between the  $x$  axis and the vector connecting the centre of mass to the robot. Here,  $\eta(u)$  is any monotonic function of its argument. It can be used to regulate the speed. Given  $\mathbf{q}_{i_d}(t)$ , the non-holonomic control rule implemented on each robot causes  $\mathbf{q}_i(t)$  to track the desired point.

The linear and angular velocities are given by

$$v_i(t) = k_v \left( (\mathbf{q}_{i_d}(t) - \mathbf{q}_i(t)) \cdot \begin{pmatrix} \cos(\theta_i(t)) \\ \sin(\theta_i(t)) \end{pmatrix} \right), \quad (25)$$

$$\omega_i(t) = k_\omega (\theta_{i_d}(t) - \theta_i(t)) + \dot{\theta}_{i_d}(t), \quad (26)$$

$$\theta_{i_d}(t) = \text{atan} \left( \frac{y_{i_d}(t) - y_i(t)}{x_{i_d}(t) - x_i(t)} \right). \quad (27)$$

This is a well-known control rule for nonholonomic unicycles. Initially,  $\bar{\mathbf{q}}_d(t_0) = \bar{\mathbf{q}}(t_0)$ . The trajectory for  $\bar{\mathbf{q}}_d(t)$

can be represented as the parametrized curve  $p_0(s_0)$  such that

$$\dot{\bar{\mathbf{q}}}_d(t) = \frac{\partial}{\partial s_0} p_0(s_0) \dot{s}_0. \quad (28)$$

$\partial p_0(s_0)/\partial s_0$  will be designed in later sections. The integrity of the formation is maintained by regulating the speed of traversal of the paths  $p_i(s_i)$ , i.e.  $\dot{s}_i$ . In [15], the authors propose to define

$$\dot{s}_i = c \exp(-\alpha_i \|\mathbf{q}_{i_d}(t) - \mathbf{q}_i(t)\|), \quad (29)$$

$$\dot{s}_0 = \frac{v_0}{\left\| \frac{\partial}{\partial s_0} p_0(s_0) \right\|} \exp \left( -\alpha_0 \sum_{i=0}^{N-1} \|\mathbf{q}_{i_d}(t) - \mathbf{q}_i(t)\| \right), \quad (30)$$

for  $c, v_0, \alpha_i, \alpha_0 > 0$ .  $v_0$  is a nominal speed. Besides the closeness of the formation to the desired one, other factors can also affect  $\dot{s}_0$ . In the following, by  $\dot{s}_0$ , we only mean that component of the speed which is related to formation keeping. Other factors affecting the speed of the virtual leader are included as part of the design for  $\partial p_0(s_0)/\partial s_0$ .

### 3. Field interpolation by polygonal formations

Interpolation on polygons can be done by barycentric coordinates ([18],[19]). Let the domain  $\Omega_{\mathbf{P}}(t) \subset \mathfrak{R}^2$  denote the inside of the (assumed convex) polygonal formation, with boundary  $\partial\Omega$ . Let  $p \in \Omega_{\mathbf{P}}$  be a given point. Any vector of real numbers

$$\alpha(p, \mathbf{q}_{\mathbf{P}}) = [\alpha_0, \alpha_1, \dots, \alpha_{N-1}]^T \quad (31)$$

is called the *generalized barycentric coordinates* of  $p$  if (1) the coordinates have *linear precision*, i.e., they can reproduce a linear function exactly,  $p = \alpha(p, \mathbf{q}_{\mathbf{P}}) \cdot \mathbf{q}_{\mathbf{P}}$ , (2) the coordinates are non-negative and bounded to guarantee no under- or over-shooting in the coordinates,  $0 \leq \alpha_i \leq 1$ , (3) the coordinates form a *partition of unity* to assure constant precision and to make the formulation invariant to both translation and rotation,  $\sum_i \alpha_i = 1$ , (4) the coordinates are infinitely differentiable with respect to their arguments to ensure smoothness when a node  $\mathbf{q}_i$  is moved,  $\alpha_i \in C^\infty$ , (5) we have  $\alpha_i = \delta_{ij}$ , where  $\delta_{ij}$  is the Kronecker delta function. Conditions 1 and 2 are called *affine combination*. Positivity of the coordinates (condition 2) is called *convex combination*. The functions  $\alpha_i: \mathfrak{R}^2 \otimes \mathfrak{R}^{2N} \rightarrow \mathfrak{R}^+$ ,  $i = 0, \dots, N-1$ , are also called  *$C^0$  shape functions* associated with  $p$ .

Let  $\mathbf{f}: \mathfrak{R}^2 \rightarrow \mathfrak{R}$  represent a field defined on the plane (altitude measurements). We assume that the vehicles can measure the field at their respective positions, giving the values  $\mathbf{f}(q_i(t)) = \mathbf{f}_i(t)$ . An *interpolant* (or *interpolation scheme*) of  $\mathbf{f}$ , based on polygon  $\mathbf{P}$ , is a function  $\mathbf{f}: \Omega \rightarrow \mathfrak{R}$  defined by

$$\tilde{\mathbf{f}}(p) = \alpha_p(p, \mathbf{q}_{\mathbf{P}})^T \mathbf{f}_{\mathbf{P}}(t), \quad (32)$$

where  $\mathbf{f}_p(t) = [\mathbf{f}_0(t), \dots, \mathbf{f}_{N-1}(t)]^T$ .  $\tilde{\mathbf{f}}(p)$  is the interpolated value of  $\mathbf{f}(p)$ .  $\mathbf{f}$  generates a vector of barycentric coordinates and takes the inner product of this vector and the vector composed of field values at the vertices. The coordinates generated by  $\mathbf{f}$  should of course satisfy the above conditions. Condition 5 above makes sure that the interpolated value at a vertex is equal to node data:  $\mathbf{f}(\mathbf{q}_i(t)) = \mathbf{f}_i(t)$ . Conditions 2 and 3 ensure that the interpolated values are bounded between the minimum and maximum of the nodal values:  $\min_i\{\mathbf{f}_i\} \leq \mathbf{f}(p) \leq \max_i\{\mathbf{f}_i\}$ . Along the edges of the polygon, the interpolant must be piece-wise linear (i.e.,  $C^0$ ). This can be stated as

$$\tilde{\mathbf{f}}(\tau) = \tau \mathbf{f}_i + (1 - \tau) \mathbf{f}_{i+1}, \quad (33)$$

$$q = \tau \mathbf{q}_i + (1 - \tau) \mathbf{q}_{i+1}, \quad (34)$$

where  $q \in \partial\Omega$  and  $\tau \in [0, 1]$ .

If  $w(p, \mathbf{q}_P) = [w_0(p, \mathbf{q}_P), \dots, w_{N-1}(p, \mathbf{q}_P)]^T$  is a vector of real numbers such that

$$w(p, \mathbf{q}_P) \cdot (\mathbf{q}_P - p \mathbf{1}_N) = 0, \quad (35)$$

then partition of unity coordinates can be found by the formula

$$\alpha_i(p, \mathbf{q}_P) = w_i(p, \mathbf{q}_P) / (\sum_k w_k(p, \mathbf{q}_P)). \quad (36)$$

$w_i(p, \mathbf{q}_P)$  's are called (non-normalized) *weight functions*.

There are a plethora of methods for defining shape functions. Some do not satisfy all the desired properties. Some are only defined for regular polygons. In this section, we will present a method based on *Wachspress* construction which can be used for irregular but convex polygons (thus providing for some robustness to deviations from a perfect polygon). Wachspress formula for non-normalized weight functions is

$$w_i = A(\mathbf{q}_{i-1}, \mathbf{q}_i, \mathbf{q}_{i+1}) \prod_{k \in \{i, i+1\}} A(\mathbf{q}_{k-1}, \mathbf{q}_k, p). \quad (37)$$

It can be proved that coordinates based on this formula satisfy all the properties. For the case of a triangle, it reduces to the original triangulat barycentric coordinates. In [20], a simpler local formula is given for when  $p$  is strictly inside the polygon. If this is the case, then

$$w_i(p, \mathbf{q}_P) = \frac{A(\mathbf{q}_{i-1}, \mathbf{q}_i, \mathbf{q}_{i+1})}{A(\mathbf{q}_{i-1}, \mathbf{q}_i, p) A(\mathbf{q}_i, \mathbf{q}_{i+1}, p)}. \quad (38)$$

It can be shown that this expression simplifies to

$$w_i(p, \mathbf{q}_P) = \frac{\cot(\delta_i(t)) + \cot(\lambda_i(t))}{\|p - \mathbf{q}_i(t)\|^2}. \quad (39)$$

Figure 3(a) shows the various terms used in these formulas. For numerical stability reasons, instead of  $\cot(\delta_i(t))$  and  $\cot(\lambda_i(t))$ , we use the expressions  $\tilde{u}_{\cot}(p, \mathbf{q}_i, \mathbf{q}_{i-1})$  and  $\tilde{u}_{\cot}(p, \mathbf{q}_i, \mathbf{q}_{i+1})$ , respectively, where

$$\tilde{u}_{\cot}(z_1, z_2, z_3) = \frac{(z_3 - z_2) \cdot (z_1 - z_2)}{\|(z_3 - z_2) \times (z_1 - z_2)\|}. \quad (40)$$

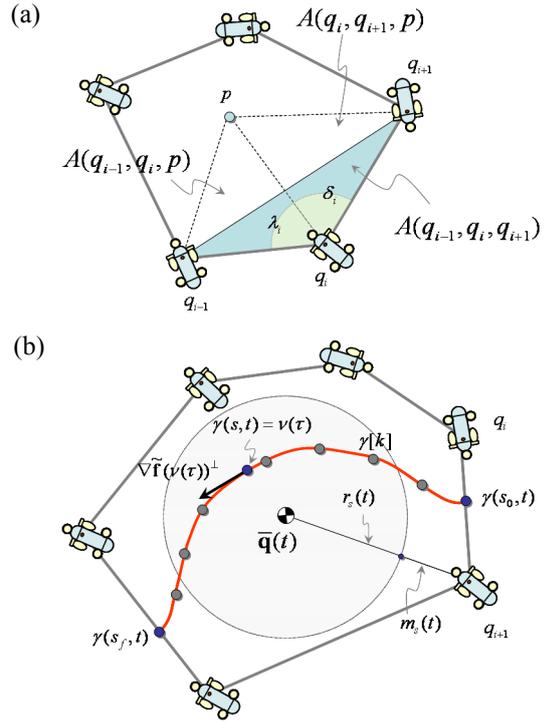


figure 3: (a) Interpolation on a convex polygon. (b) Construction of the interpolated isocline.

If  $p$  is very close to the boundary, a simple linear interpolation or the original Wachspress formula should be used. Closeness to boundary is determined by the condition

$$\|(\mathbf{q}_{i+1} - \mathbf{q}_i) \times (p - \mathbf{q}_i(t))\| \leq \varepsilon \|\mathbf{q}_{i+1} - \mathbf{q}_i\|, \quad (41)$$

in which case we can define

$$\tilde{\mathbf{f}}(p) = \tau(p) \mathbf{f}(\mathbf{q}_i) + (1 - \tau(p)) \mathbf{f}(\mathbf{q}_{i+1}), \quad (42)$$

where

$$\tau(p) = \frac{\|(p - \mathbf{q}_i(t)) \cdot (\mathbf{q}_{i+1}(t) - \mathbf{q}_i(t))\|}{\|\mathbf{q}_{i+1}(t) - \mathbf{q}_i(t)\|}. \quad (43)$$

In the next section, we will try to reconstruct a particular isocline of the interpolated field. To do this, we need to compute the gradient of  $\mathbf{f}(p)$ . Simple calculations show that

$$\nabla_p \tilde{\mathbf{f}}(p) = \frac{\partial \tilde{\mathbf{f}}(p)}{\partial p} = \sum_{i=0}^{N-1} \frac{\partial}{\partial p} \alpha_i(p, \mathbf{q}_P) \mathbf{f}_i(t), \quad (44)$$

where

$$\frac{\partial}{\partial p} \alpha_{p_i}(p, \mathbf{q}_P) = \frac{\partial}{\partial p} (w_i(p, \mathbf{q}_P) / (\sum_k w_k(p, \mathbf{q}_P))) \quad (45)$$

$$= \left( \sum_k w_k(p, \mathbf{q}_P) \right)^{-1} \frac{\partial}{\partial p} w_i(p, \mathbf{q}_P) \quad (46)$$

$$- w_i(p, \mathbf{q}_P) \left( \sum_k w_k(p, \mathbf{q}_P) \right)^{-2} \sum_k \frac{\partial}{\partial p} w_k(p, \mathbf{q}_P)$$

where

$$\frac{\partial}{\partial p} w_i(p, \mathbf{q}_p) = \frac{-2(p - \mathbf{q}_i)}{\|p - \mathbf{q}_i\|} (\cot(\delta_i) + \cot(\lambda_i)) + \mu \quad (47)$$

$$\mu = \frac{1}{\|p - \mathbf{q}_i\|^2} \left( - \frac{\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} (\mathbf{q}_{i-1} - \mathbf{q}_i)}{\|(\mathbf{q}_{i-1} - \mathbf{q}_i) \times (p - \mathbf{q}_i)\|^2} - \frac{\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} (\mathbf{q}_{i+1} - \mathbf{q}_i)}{\|(\mathbf{q}_{i+1} - \mathbf{q}_i) \times (p - \mathbf{q}_i)\|^2} \right) \quad (48)$$

If  $p$  is close to the boundary, as determined by non-equality (41), we use a simple linear interpolation:

$$\frac{\partial}{\partial p} \tilde{\mathbf{f}}(p) = \frac{\mathbf{f}_{i+1}(t) - \mathbf{f}_i(t)}{\|\mathbf{q}_{i+1}(t) - \mathbf{q}_i(t)\|} (\mathbf{q}_{i+1}(t) - \mathbf{q}_i(t)) . \quad (49)$$

$\partial \alpha_i(p, \mathbf{q}_p) / \partial p$ , like  $\alpha_{p_i}(p, \mathbf{q}_p)$  are invariant to translation and rotation of the formation.

To track a desired isocline value  $\mathbf{c}_d$  of the underlying field, we construct the corresponding isocline of the interpolated field and track it instead. This *interpolated isocline* is our observation of the real iso-contour. Tracking this curve and making sure that it is accurate enough (i.e., close enough to the real one) are decoupled in our method. The desired curve is the solution to the equation  $\mathbf{f}(p) = \mathbf{c}_d$ , where  $p \in \Omega$ . We are assuming that this solution is a unique curve. If that is not the case, then the size of the formation is too large with respect to variations in the underlying field. To reconstruct the curve, we should first find one of its end-points. This end-point lies on one of the edges of the polygon. Let  $\gamma(s, t)$  denote the curve,  $\gamma(s_0, t)$  is the position of one of the end-points and  $\gamma(s_f, t)$  denotes the other end-point.

To locate  $\gamma(s_0, t)$ , we iterate over all the edges and find the edge  $\{\mathbf{q}_i, \mathbf{q}_{i+1}\}$ , with the smallest  $i$ , such that  $\mathbf{c}_d \in [\mathbf{f}_i(t) - \mathbf{d}_i(t), \mathbf{f}_i(t) + \mathbf{d}_i(t)]$  where

$$\tilde{\mathbf{f}}_i(t) = \frac{1}{2} (\mathbf{f}_i(t) + \mathbf{f}_{i+1}(t)) , \quad (50)$$

and

$$\mathbf{d}_i(t) = \frac{1}{2} |\mathbf{f}_{i+1}(t) - \mathbf{f}_i(t)| . \quad (51)$$

Now, we have

$$\gamma(s_0, t) = \mathbf{q}_-(t) + w(\mathbf{q}_+(t) - \mathbf{q}_-(t)) , \quad (52)$$

$$w = \frac{\mathbf{c}_d - \mathbf{f}(\mathbf{q}_-(t))}{\mathbf{f}(\mathbf{q}_+(t)) - \mathbf{f}(\mathbf{q}_-(t))} , \quad (53)$$

where

$$\mathbf{q}_-(t) = \operatorname{argmin}_{\mathbf{q} \in \{\mathbf{q}_i, \mathbf{q}_{i+1}\}} \{\mathbf{f}_i(t), \mathbf{f}_{i+1}(t)\} \quad (54)$$

and similarly for  $\mathbf{q}_+(t)$  (replace *min* with *max*).

Next, we have to determine the direction of traversal of  $\gamma$ . This direction should be towards the inside of the polygon and is given by

$$\delta_\gamma(t) = -\operatorname{sign}\left( (\mathbf{q}_{i+1}(t) - \mathbf{q}_i(t)) \times \frac{\partial}{\partial p} \tilde{\mathbf{f}}(\gamma(s_0, t)) \right) . \quad (55)$$

Letting  $\mathbf{v}(\tau_0) = \gamma(s_0, t)$ , initially, the trace of  $\mathbf{v}$  under the evolution equation

$$\begin{aligned} \frac{\partial}{\partial \tau} \mathbf{v}(\tau) &= \alpha_1 \delta_\gamma(t) \frac{\partial}{\partial (x, y)} (\tilde{\mathbf{f}}(\mathbf{v}(\tau)))^\perp + \\ &\alpha_2 \operatorname{sign}(\mathbf{c}_d - \tilde{\mathbf{f}}(\mathbf{v}(\tau))) \frac{\partial}{\partial (x, y)} \tilde{\mathbf{f}}(\mathbf{v}(\tau)) \end{aligned} \quad (56)$$

would be the desired curve. Depending on the gains  $\alpha_1$  and  $\alpha_2$ , tracking the contour can be made as accurate as desired. Evolution stops when, for some  $\tau$ ,  $\mathbf{v}(\tau) \notin \Omega$ . Also,  $\gamma(s_f, t) = \mathbf{v}(\tau_f)$  such that  $\mathbf{v}(\tau_f) \in \partial\Omega$ , where  $\tau_f \neq \tau_0$ . In practice, we use a discrete version of  $\gamma$  which is sampled along the way. We denote the sample points by  $\gamma[k]$ ,  $k = 0, \dots, M-1$ , where  $\gamma[0]$  and  $\gamma[M-1]$  denote the end-points.

The behaviour of the interpolation scheme is problematic near the boundary and if we use simple linear interpolation at points close to the boundary, the interpolated field and its gradient experience discontinuity at the boundary separating the region *safely* inside the polygon and the one very close to  $\partial\Omega$ . This would be seen as a small deviation of the constructed curve towards alignment of its normal with polygon sides which is the result of the strategy we use to interpolate at and near polygon boundary. To alleviate this effect, it is beneficial to define a *safety circle*  $C_s(t)$  centred at  $\bar{\mathbf{q}}(t)$  with a sufficiently small radius  $r_s(t)$ . A rule of thumb for selecting a value for this radius is to put

$$r_s(t) = \min_i \{ \|\mathbf{q}_i(t) - \bar{\mathbf{q}}(t)\| \} - m_s , \quad (57)$$

where  $m_s$  is a safety margin. Given this circle, we can extract that part of  $\gamma$  which is inside this circle. Suppose  $\tilde{s}_0$  is the first  $s$  such that  $\gamma(\tilde{s}_0, t)$  is on or inside the safety circle and  $\tilde{s}_f$  is the last  $s$  with this property. Then  $\gamma(\tilde{s}_0, t)$  and  $\gamma(\tilde{s}_f, t)$  would mark the two end-points of the contour. In the discrete version, we find the least  $k$  (denoted  $k_0$ ) and the biggest  $k$  (denoted  $k_f$ ) such that  $\gamma[k_0]$  and  $\gamma[k_f]$  are on or inside the circle and use the sequence  $\gamma[k_0]$ ,  $\gamma[k_0 + 1], \dots, \gamma[k_f]$  as the sampled curve. See figure 3(b) for illustration.

## 4. Isocline tracking control

Here, we present three simple tracking strategies.

**1.** The simplest way to track an isocline is to move the virtual leader (the desired goal for the centre of mass) towards the interpolated isocline and, at the same time, move along it. This behaviour is captured in the control law (figure 4(a))

$$\begin{aligned} \frac{\partial}{\partial s_0} p_0(s_0) &= a_\gamma (\beta_1 \vec{\mathbf{N}}_\gamma(\tau_C, t) + \beta_2 \vec{\mathbf{T}}_\gamma(\tau_C, t)) + \\ &(1 - a_\gamma) \beta_3 (\mathbf{c}_d - \tilde{\mathbf{f}}(\bar{\mathbf{q}}(t))) \nabla_{\bar{\mathbf{q}}(t)} \tilde{\mathbf{f}}(\bar{\mathbf{q}}(t)) \end{aligned} \quad (58)$$

where  $a_\gamma = 1$  if  $\gamma$  exists and is zero otherwise,  $\beta_3$  is a gain,  $\tau_C$  denotes the value of the parametrization of  $\gamma$  for

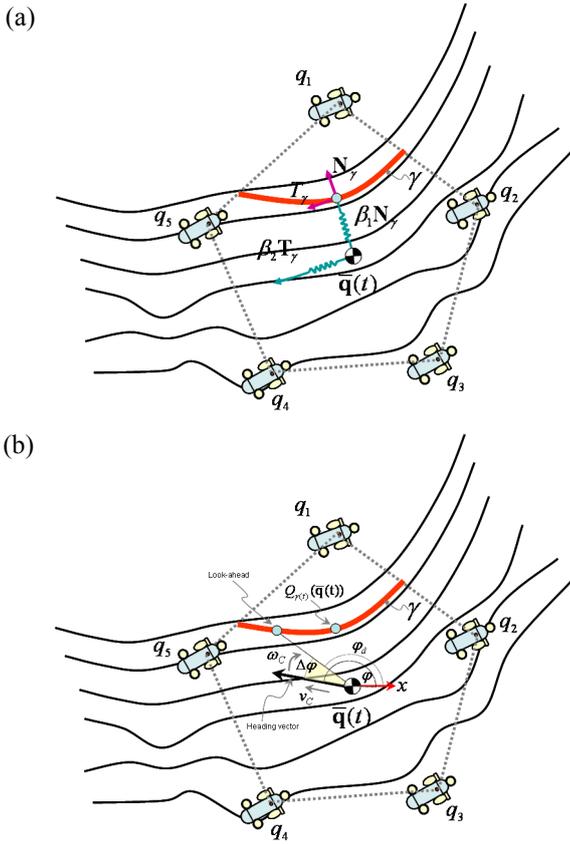


Figure 4: (a) Tracking the reconstructed isocline. (b) Lateral and longitudinal control of the formation centre.

which  $\gamma(t, \tau_c)$  is the closest point to  $\bar{\mathbf{q}}(t)$ . This closest point is also denoted  $Q_{\gamma(t)}(\bar{\mathbf{q}}(t))$ . Also

$$\bar{\mathbf{N}}_{\gamma}(\tau_c, t) = \mathbf{n}(Q_{\gamma(t)}(\bar{\mathbf{q}}(t))) \quad (59)$$

is normal to the curve at  $\tau_c$ , and

$$\bar{\mathbf{T}}_{\gamma}(\tau_c, t) = \frac{\dot{\gamma}(t, \tau_c)}{\|\dot{\gamma}(t, \tau_c)\|}, \quad \dot{\gamma}(t, \tau_c) = \frac{\partial}{\partial p} \tilde{\mathbf{f}}(\gamma(t, \tau_c))^\perp \quad (60)$$

is the local tangent at  $\tau_c$ . The gain  $\beta_1$  determines how fast the formation should react to *normal* variations in the tracked position  $\gamma(t, \tau_c)$  and  $\beta_2$  determines the speed of traversal.

2. To reduce chattering when close to the isocline and discourage moving along the tangent when far from it, we can make  $\beta_1$  and  $\beta_2$  functions of the normal distance. Let us define

$$\beta_1 = v_N \sigma_{\varepsilon, \beta}(\mathbf{Q}(t)), \quad \text{and} \quad (61)$$

$$\beta_2 = v_T (1 - \sigma_{\varepsilon, \beta}(\mathbf{Q}(t))), \quad (62)$$

where  $\mathbf{Q}(t) = \|Q_{\gamma(t)}(\bar{\mathbf{q}}(t))\|$ , and

$$\sigma_{p, q}(u) = \frac{1}{1 + \exp(-4q(u - p))}. \quad (63)$$

This implements a smooth switching mechanism between the two tangential and normal behaviours [1].  $v_N$  and  $v_T$  denote the nominal normal and tangential speeds. For particular values chosen for  $\varepsilon > 0$  and small  $\delta > 0$ , if we set

$$\beta = \frac{1}{4\varepsilon} \left| \ln \left( \frac{\delta}{1 - \delta} \right) \right|, \quad (64)$$

then we will have  $\beta_1 \geq v_N(1 - \delta)$  ( $\beta_1 \leq v_N\delta$ ) and  $\beta_2 \leq v_T\delta$  ( $\beta_2 \geq v_T(1 - \delta)$ ) when  $\mathbf{Q}(t) \geq \varepsilon + \varepsilon$  ( $\mathbf{Q}(t) \leq \varepsilon - \varepsilon$ ).

3. An alternative way of moving towards the reconstructed isocline is to model the motion of the centre of the formation as a non-holonomic device. To do this, we can associate a *heading vector* with the centre. In this case, instead of determining normal and tangential speeds, we should design *lateral* and *longitudinal* speeds. Also, instead of moving towards the closest point and sliding along the tangent to the curve, we should now designate a *lookahead point* on the curve which would serve as the desired goal for the centre of mass of the formation. Using a lookahead makes tracking more robust to fluctuations of the path (whose observation changes over time). This is the method usually employed for tracking of a known path by non-holonomic robots. Figure 4(b) illustrates this. The lookahead point is computed as

$$\mathbf{L}(\gamma, \bar{\mathbf{q}}(t)) = \gamma(t, \tau_c + D\tau_l), \quad (65)$$

$$D = \text{sign}((\gamma[k_C + 1] - \gamma[k_C]) \cdot \bar{\mathbf{T}}_{\gamma}(\tau_c, t)) \quad (66)$$

where  $k_C$  is the index of the sample point immediately *before*  $\gamma(\tau_c)$ .  $\{\gamma[k_C], \gamma[k_C + 1]\}$  is thus the segment on which  $\mathbf{Q}(t)$  is located. Also,  $\tau_l$  is the lookahead length. If the computed lookahead point exceeds the end points of the curve, then we set  $\mathbf{L}(\gamma, \bar{\mathbf{q}}(t)) = \gamma(t, 1)$  or  $\mathbf{L}(\gamma, \bar{\mathbf{q}}(t)) = \gamma(t, 0)$ , depending on the direction. The virtual centre of mass is moved according to the following control laws ([16]) for longitudinal and lateral movements:

$$\mathbf{v}_c(t) = a_v \frac{v_n}{\rho_T} \|\bar{\mathbf{q}}(t) - \mathbf{L}(\gamma, \bar{\mathbf{q}}(t))\| \cos(\Delta\varphi) \dot{s}_0 + \quad (67)$$

$$(1 - a_v) \beta_3 (\mathbf{c}_d - \tilde{\mathbf{f}}(\bar{\mathbf{q}}(t))) \nabla_{\bar{\mathbf{q}}(t)} \tilde{\mathbf{f}}(\bar{\mathbf{q}}(t))$$

$$\omega_c(t) = a_v (k\Delta\varphi + \dot{\varphi}_d), \quad (68)$$

where  $v_n$  is a nominal speed,  $\rho_T$  is a desired distance between the centre of mass the lookahead and  $k$  is a positive gain.  $\dot{s}_0$  denotes the formation feedback. Note that with this control scheme, longitudinal and lateral *responsiveness* play the same role as tangential and normal responsiveness in the previous scheme. If they are designed carefully, it is possible to produce smooth motion even against very rugged terrain or the presence of large noises.

## 5. Simulations

Figure 5(a) shows a simulation run for a formation with  $N = 5$  vehicles and  $r_P = 100$ . We have set  $\beta_1 = 40$  and  $\beta_2 = 400$ . Also,  $k_v = 5.5$  and  $k_\omega = 50$ . The simulations are synchronous and the vehicles carry on identical motions. It is furthermore assumed that the designed linear and angular velocities are within the tolerance of the vehicles and there is no error in measurement, although there is

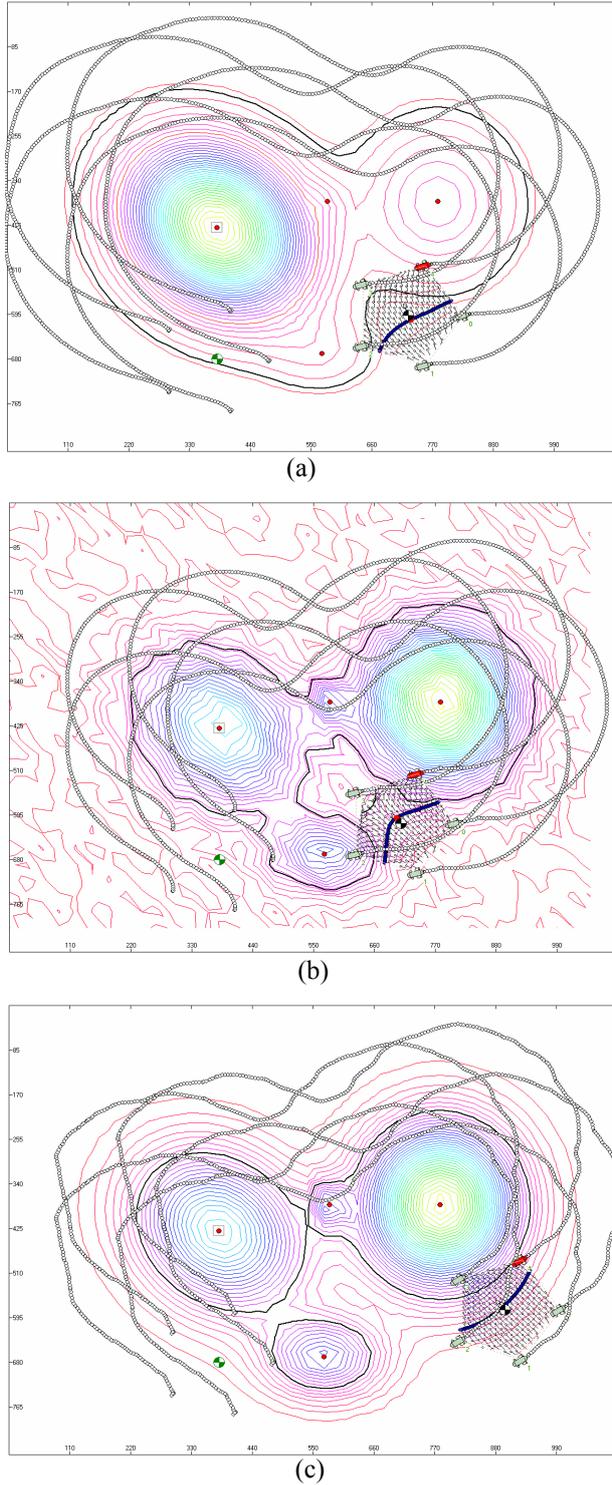


figure 5: (a) Small normal forcing. (b) Rugged terrain. (c) High normal gain.

error in interpolation because of the size of the formation. As is seen, the produced path for the robots is quite smooth. The reason is that the formation is always moving *forward* on the line on which the tangent to the curve lies and the spring for normal motion is not very stiff. Figure 5(b) shows the same formation in a very rugged terrain. The maximum field value is 6000. Underlying smooth field values have been contaminated with a uniform noise

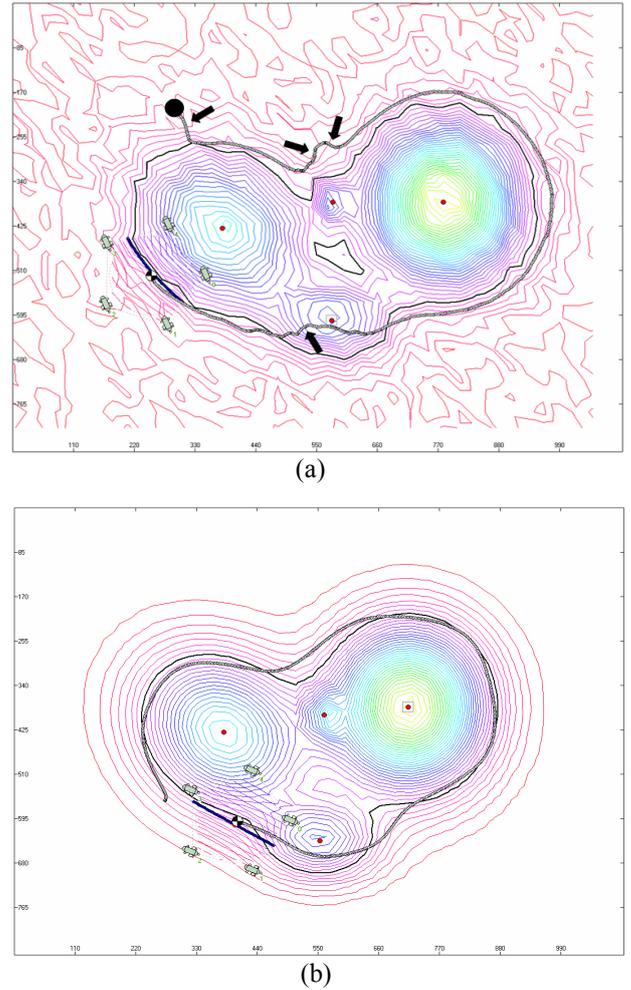


figure 6: (a) Smooth switching between normal and tangential motions. (b) Non-holonomically modelled formation centre, tracking a lookahead on the reconstructed curve.

in the range  $[-60, 60]$ . It is seen that the behaviour has not changed and the paths are still smooth. Figure 5(c) shows the paths for a formation on a smooth terrain with  $\beta_1 = 200$ . Figure 6(a) shows a simulation run of a formation employing adaptive gain strategy. The initial position is marked by a black disk and the arrows show parts of the path where motion is mainly translation, along the normal, towards the curve. In this simulation,  $\varepsilon = 5$ ,  $\varepsilon = 10$ ,  $\delta = 0.00001$ , and  $v_N = v_T = 200$ . Figure 6(b) shows a simulation run for a formation employing the third control strategy.

## 6. Conclusions and future research

In this paper, we presented a gradient-free method for tracking iso-contours using a robotic network. The basic idea was that of transforming the problem of following the gradient (and its conjugate) into that of tracking a curve which is an isocline of a locally reconstructed distribution. The particular polygonal shape of the formation allows the use of interpolation schemes. With this robust method, smooth paths can be produced even when the underlying

field is very rugged or measurements and/or actuation are noisy. There are several issues which merit further consideration. First of all, to minimize interpolation error, the size of the formation has to be adaptively changed. Relevant initial results will be presented elsewhere. Secondly, the interpolation method used allows for some deviation from the rigid shape as long as it remains convex. We will, in the future, consider methods which can handle non-convex polygons as well. Future research should also aim at explicitly dealing with noise in various sensor/actuator modalities. Finally, autonomous selective sampling of iso-lines should be addressed.

## References

- [1] Jaeger, H., Christaller, T., *Dual Dynamics: Designing Behaviour Systems for Autonomous Robots*, Artificial Life and Robotics, Vol. 2, 1998.
- [2] <http://users.rsise.anu.edu.au/~serafina/>
- [3] Yoerger, D., et al., *High Resolution Mapping of a Fast Spreading Mid Ocean Ridge with the Autonomous Benthic Explorer*, Proc. of the 11th Symp. on Unmanned Submersible Technology, Durham, NH, USA, 1999.
- [4] Rendas, M.J., Folcher, J.-P., Lourtie, I.M., *Contour Tracking with Video and Altimeter*, Project SUMARE Deliverable 4.1 (<http://www.mumm.ac.be/SUMARE/>), 2002.
- [5] Reet, A.V., *Contour Tracking for the REMUS Autonomous Underwater Vehicle*, Master's thesis, Dept. of Mech. Eng., US Naval Postgraduate School, 2005.
- [6] Kemp, M., Bertozzi, A.L., Marthaler, D., *Multi-UUV Perimeter Surveillance*, IEEE/OES Conf. on Autonomous Underwater Vehicles, 2004.
- [7] Bennett, A.B., Leonard, J.J., Bellingham, J.G., *Bottom Following for Survey Class Autonomous Underwater Vehicles*, Proc. Int. Symp. on Unmanned Untethered Submersible Technology, New Hampshire, 1995.
- [8] Zhang, F., Leonard, N.E., *Generating Contour Plots using Multiple Sensor Platforms*, Proc. IEEE Swarm Intelligence Symp., 2005.
- [9] Ogren, P., Fiorelli, E., Leonard, N.E., *Formations with a Mission: Stable Coordination of Vehicle Group Maneuvres*, Proc. 15th Int. Symp. on Math. Theory of Networks and Systems, 2002.
- [10] Schill, F., Zimmer, U.R., Trumppf, J., *Visible Spectrum Optical Communication and Distance Sensing for Underwater Applications*, Proc. of ACRA, Canberra, Australia, 2004.
- [11] Schill, F., Zimmer, U.R., Trumppf, J., *Towards Optimal TDMA Scheduling for Robotic Swarm Communication*, Proc. of TAROS, London, 2005.
- [12] Schill, F., Zimmer, U.R., Trumppf, *Effective Communication in Schools of Submersibles*, Proc. of OCEANS, Singapore, 2006.
- [13] Kottege, N., Zimmer, U.R., *MLS-based, Distributed Bearing, Range, and Posture Estimation for Schools of Submersibles*, Proc. of 10th Int. Symp. on Experimental Robotics, Rio de Janeiro, 2006.
- [14] Hormann, K., Spinello, S., Schroder, P.,  *$C^1$ -Continuous Terrain Reconstruction from Sparse Contours*, 8th Int. Fall Workshop on Vision and Visualization, Munich, Germany, 2003.
- [15] Egerstedt, M., Hu, X., *Formation Constrained Multi-Agent Control*, IEEE Trans. on Robotics and Automation, 17(6), 2001.
- [16] Egerstedt, M., Hu, X., Stosky, A., *Control of Mobile Platforms Using a Virtual Vehicle Approach*, IEEE Trans. on Automatic Control, 46(11), 2001.
- [17] Marshall, J.A., Lin, Z., Brouke, M.E., Francis, B.A., *A Pursuit Strategy for Wheeled-Vehicle Formations*, Proc. of 42nd IEEE Conf. on Decision and Control, Hawaii, 2003.
- [18] Sukumar, N., Malsch, E.A., *Recent Advances in the Construction of Polygonal Finite Element Interpolants*, Archives of Computational Methods in Engineering, 13(1), 2006.
- [19] Floater, M.S., Hormann, K., Kos, G., *A General Construction of Barycentric Coordinates over Convex Polygons*, Advances in Computational Mathematics, 24(1), 2006.
- [20] Meyer, M., Lee, H., Barr, A., Desbrun, M., *Generalized Barycentric Coordinates on Irregular Polygons*, Journal of Graphics Tools, 7(1), 2002.