



HAL
open science

Tracking Environmental Isoclines using Polygonal Formations of Submersible Autonomous Vehicles

Shahab Kalantar, Uwe Zimmer

► **To cite this version:**

Shahab Kalantar, Uwe Zimmer. Tracking Environmental Isoclines using Polygonal Formations of Submersible Autonomous Vehicles. 6th International Conference on Field and Service Robotics - FSR 2007, Jul 2007, Chamonix, France. inria-00258741

HAL Id: inria-00258741

<https://inria.hal.science/inria-00258741>

Submitted on 25 Feb 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Tracking Environmental Isoclines using Polygonal Formations of Submersible Autonomous Vehicles

Shahab Kalantar and Uwe Zimmer

Australian National University,
Department of Information Engineering, Canberra, ACT 0200, Australia
shahab.kalantar@rsise.anu.edu.au

Summary: 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

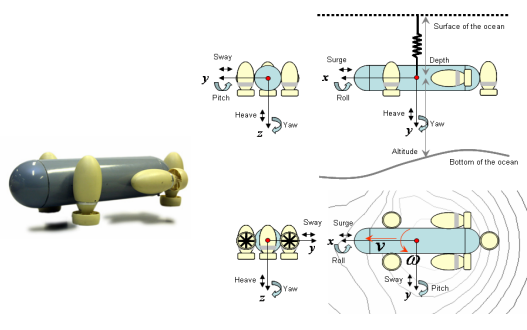


Fig. 1. Serafina underwater vehicle.

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., 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:

$$\dot{x}(t) = v(t)\cos\theta(t), \dot{y}(t) = v(t)\sin\theta(t), \dot{\theta}(t) = \omega(t), \quad (1)$$

where $v(t)$ 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.

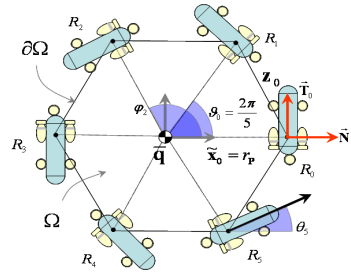


Fig. 2. Polygonal formation.

2 Initializing and maintaining polygonal formations

In this section, we explain how a polygonal formation (figure 2) 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.

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

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

$$\mathfrak{g}(t) = \cos^{-1}(\mathbf{n}(\mathbf{x}_i(t)) \cdot \mathbf{n}(\mathbf{x}_{i+1}(t))), \quad \mathbf{n}(\hat{v}) = \hat{v} / \|\hat{v}\|, \quad (3)$$

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

$\bar{\mathbf{q}}(t)$ 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 $\mathbf{G}_p(\mathbf{q}(t)) = 0$ gives a polygonal formation which is unique up to translation and rotation. More technically, $\mathbf{G}_p^{-1}(0) / (\text{SO}(2) \otimes \mathfrak{R}^2)$ is a singleton. The position of the centre of mass $\bar{\mathbf{q}}(t)$ breaks the translational (\mathfrak{R}^2) symmetry and the angle $\theta_0(t)$ (which can be defined as the orientation of the formation) breaks the rotational ($\text{SO}(2)$) symmetry. Letting $B_\varepsilon(u)$ denote an ε -ball around $u \in \mathfrak{R}^N$, an ε -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 ω_i . 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 $\mathbf{q}_{i_d}(t) = p_i(s_i)$ where $p_i(s_i)$ denotes the trajectory of $\mathbf{q}_{i_d}(t)$, parametrized by s_i , and

$$\dot{\mathbf{q}}_{i_d}(t) = \frac{\partial}{\partial s_i} p_i(s_i) \dot{s}_i, \quad (5)$$

$$\frac{\partial}{\partial s_i} p_i(s_i) = -\nabla_{\mathbf{q}_{i_d}(t)} \mathbf{G}_p(\mathbf{q}_p(t)) + \beta(\bar{\mathbf{q}}_d(t) - \bar{\mathbf{q}}(t)). \quad (6)$$

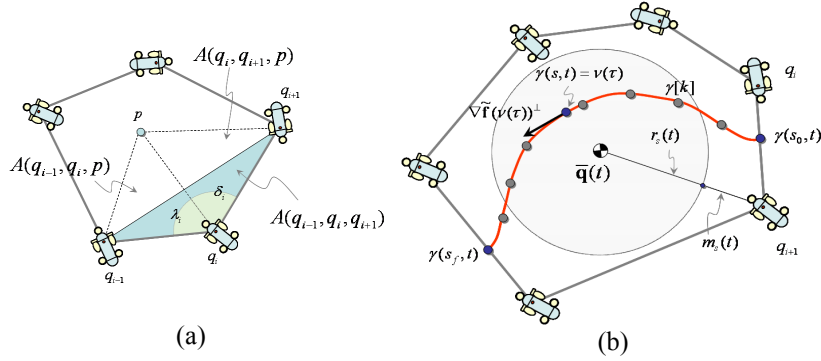


Fig. 3. (a) Interpolation on a convex polygon. (b) Construction of the interpolated isocline.

$\partial p_i(s_i)/\partial s_i$ is the tangent to the trajectory. \dot{s}_i is the speed of traversal of the trajectory. $\mathbf{q}_i(t)$ can be regarded as the virtual leader of \mathbf{R}_i . $\mathbf{q}_d(t)$ 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. Control rules for non-holonomic vehicles (15.) can be used for tracking. Initially, $\mathbf{q}_d(t_0) = \mathbf{q}(t_0)$. The trajectory for $\mathbf{q}_d(t)$ can be represented as the parametrized curve $p_0(s_0)$ such that

$$\dot{\mathbf{q}}_d(t) = \frac{\partial}{\partial s_0} p_0(s_0) \dot{s}_0. \quad (7)$$

$\partial p_0(s_0)/\partial s_0$ 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., expressions for \dot{s}_i and \dot{s}_0 are given.

3 Field interpolation by polygonal formations

Interpolation on polygons can be done by barycentric coordinates (18,19). Let the domain $\Omega_{\mathbf{p}}(t) \subset \mathcal{R}^2$ 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 $\alpha(p, \mathbf{q}_{\mathbf{p}}) = [\alpha_0, \alpha_1, \dots, \alpha_{N-1}]^T$ 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 $\alpha_i(\mathbf{q}_j, \mathbf{q}_p) = \delta_{ij}$, where δ_{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 $\tilde{\mathbf{f}}: \Omega \rightarrow \mathfrak{R}$ defined by $\tilde{\mathbf{f}}(p) = \alpha_p(p, \mathbf{q}_p)^T \mathbf{f}_p(t)$, 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 $\mathbf{f}(p)$. α 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 α should of course satisfy the above conditions. Condition 5 above makes sure that the interpolated value at a vertex is equal to node data: $\tilde{\mathbf{f}}(\mathbf{q}_i(t)) = \mathbf{f}_i(t)$. Conditions 2 and 3 ensure that the interpolated values are bounded between the minimum and maximum of the nodal values: $\min_i \{\mathbf{f}_i\} \leq \tilde{\mathbf{f}}(p) \leq \max_i \{\mathbf{f}_i\}$. Along the edges of the polygon, the interpolant must be piece-wise linear (i.e., C^0). This can be stated as

$$\tilde{\mathbf{f}}(\tau) = \tau \mathbf{f}_i + (1 - \tau) \mathbf{f}_{i+1}, \quad q = \tau \mathbf{q}_i + (1 - \tau) \mathbf{q}_{i+1}, \quad (8)$$

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 $w(p, \mathbf{q}_p) \cdot (\mathbf{q}_p - p \mathbf{1}_N) = 0$, then partition of unity coordinates can be found by the formula

$$\alpha_i(p, \mathbf{q}_p) = w_i(p, \mathbf{q}_p) / (\sum_k w_k(p, \mathbf{q}_p)). \quad (9)$$

$w_i(p, \mathbf{q}_p)$'s are called (non-normalized) *weight functions*. There are a plethora of methods for defining shape functions. We will use 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). In 20., a simple local formula is given for when p is strictly inside the polygon:

$$w_i(p, \mathbf{q}_p) = \frac{A(\mathbf{q}_{i-1}, \mathbf{q}_i, \mathbf{q}_{i+1})}{A(\mathbf{q}_{i-1}, \mathbf{q}_i, p)A(\mathbf{q}_i, \mathbf{q}_{i+1}, p)} = \frac{\cot(\delta_i(t)) + \cot(\lambda_i(t))}{\|p - \mathbf{q}_i(t)\|^2}. \quad (10)$$

Figure 3(a) shows the various terms used in these formulas. If p is very close to the boundary, a simple linear interpolation should be used. To reconstruct a particular isoline of the interpolated field, we need to compute the gradient of $\tilde{\mathbf{f}}(p)$:

$$\nabla_p \tilde{\mathbf{f}}(p) = \frac{\partial}{\partial p} \tilde{\mathbf{f}}(p) = \sum_{i=0}^{N-1} \frac{\partial}{\partial p} \alpha_i(p, \mathbf{q}_p) \mathbf{f}_i(t), \quad (11)$$

To track a desired isoline value \mathbf{c}_d of the underlying field, we construct the corresponding isoline of the interpolated field and track it instead. This *interpolat-*

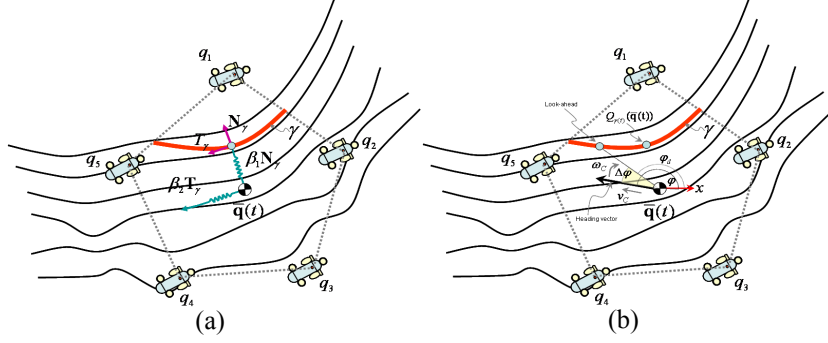


Fig. 4. (a) Tracking the reconstructed isocline. (b) Lateral and longitudinal control of the formation centre.

ed isoline 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. To reconstruct the curve $\gamma(s, t)$, we first find its end-points ($\gamma(s_0, t)$ and $\gamma(s_f, t)$), which lie on edges. Next, we traverse the curve sampling it along the way. We can also define a *safety circle* sufficiently away from the edges. See figure 3(b).

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))

$$\begin{aligned} \frac{\partial}{\partial s_0} p_0(s_0) &= a_\gamma (\beta_1 \bar{\mathbf{N}}_\gamma(\tau_C, t) + \beta_2 \bar{\mathbf{T}}_\gamma(\tau_C, t)) + \\ & (1 - a_\gamma) \beta_3 (\mathbf{c}_d - \tilde{\mathbf{f}}(\bar{\mathbf{q}}(t))) \nabla_{\bar{\mathbf{q}}(t)} \tilde{\mathbf{f}}(\bar{\mathbf{q}}(t)) \end{aligned} \quad (12)$$

where $a_\gamma = 1$ if γ exists and is zero otherwise, β_3 is a gain, τ_C denotes the value of the parametrization of γ for 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 $\bar{\mathbf{N}}_\gamma(\tau_C, t) = \mathbf{n}(Q_{\gamma(t)}(\bar{\mathbf{q}}(t)))$ is normal to the curve at τ_C , and $\bar{\mathbf{T}}_\gamma(\tau_C, t)$ is the local tangent at τ_C . The gain β_1 determines how fast the formation should react to *normal* variations in the tracked position $\gamma(t, \tau_C)$ and β_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 β_1 and β_2 functions of the normal

distance. Let us define $\beta_1 = v_N \sigma_{\varepsilon, \beta}(\mathbf{Q}(t))$, $\beta_2 = v_T(1 - \sigma_{\varepsilon, \beta}(\mathbf{Q}(t)))$, $\mathbf{Q}(t) = \|\mathcal{Q}_{\gamma(t)}(\bar{\mathbf{q}}(t))\|$, where $\sigma_{p,q}(u)$ is the sigmoidal function defined in 1. to implement a smooth switching mechanism between the two tangential and normal behaviours. 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 $\beta = |\ln(\delta/(1-\delta))|/4\varepsilon$, 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). figure 4(b) illustrates this. Denoting by \mathbf{L} this lookahead point, the virtual centre of mass is moved according to the following control laws (16.) for longitudinal and lateral movements:

$$\begin{aligned} \mathbf{v}_c(t) &= a_\gamma \frac{v_n}{\rho_T} \|\bar{\mathbf{q}}(t) - \mathbf{L}(\gamma, \bar{\mathbf{q}}(t))\| \cos(\Delta\phi) \dot{s}_0 + \\ &(1 - a_\gamma) \beta_3 (\mathbf{c}_d - \tilde{\mathbf{f}}(\bar{\mathbf{q}}(t))) \nabla_{\bar{\mathbf{q}}(t)} \tilde{\mathbf{f}}(\bar{\mathbf{q}}(t)) \end{aligned} \quad (13)$$

and $\omega_c(t) = a_\gamma(k\Delta\phi + \dot{\phi}_d)$, where v_n is a nominal speed, ρ_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 *responsive-ness* play the same role as tangential and normal responsive-ness 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 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)

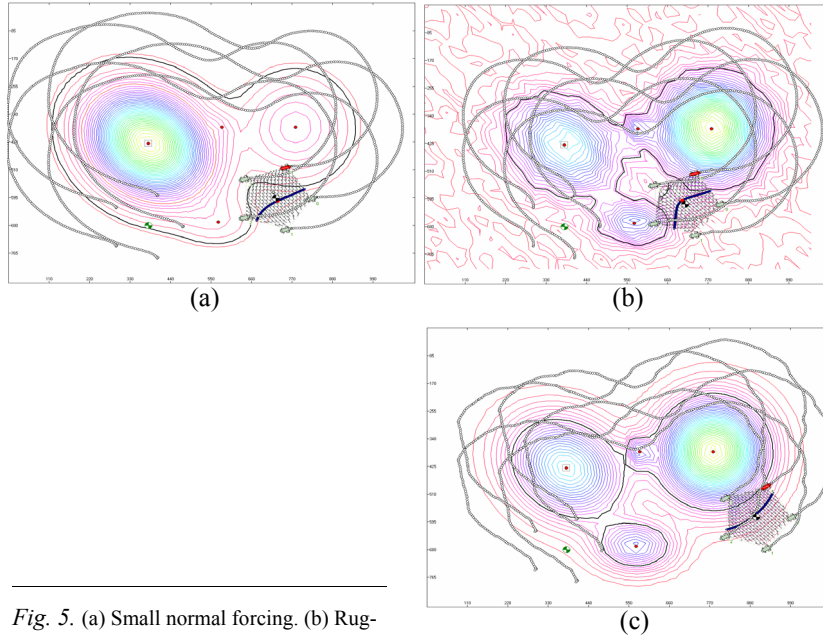


Fig. 5. (a) Small normal forcing. (b) Rugged terrain. (c) High normal gain.

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 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,

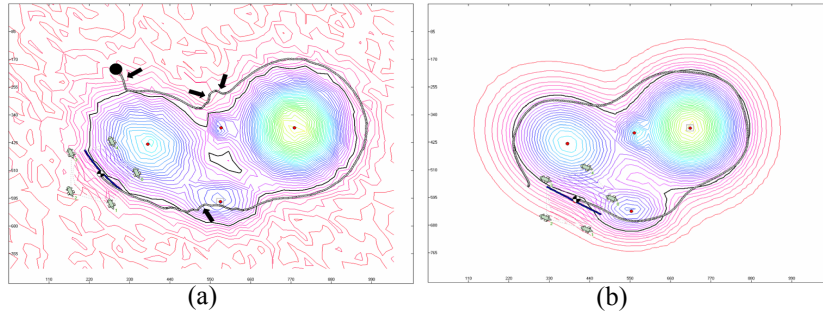


Fig. 6. (a) Smooth switching between normal and tangential motions. (b) Non-holonomically modelled formation centre, tracking a lookahead on the reconstructed curve.

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 isoclines should be addressed.

References

1. Jaeger, H., Christaller, T. (1998) *Dual Dynamics: Designing Behaviour Systems for Autonomous Robots*. Artificial Life and Robotics, Vol. 2.
2. <http://users.rsise.anu.edu.au/~serafina/>
3. Yoerger, D., et al. (1999) *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.
4. Rendas, M.J., Folcher, J.-P., Lourtie, I.M. (2002) *Contour Tracking with Video and Altimeter*. Project SUMARE Deliverable 4.1.
5. Reet, A.V. (2005) *Contour Tracking for the REMUS Autonomous Underwater Vehicle*. Master's thesis, Dept. of Mech. Eng., US Naval Postgraduate School.
6. Kemp, M., Bertozzi, A.L., Marthaler, D. (2004) *Multi-UUV Perimeter Surveillance*. IEEE/OES Conf. on Autonomous Underwater Vehicles.
7. Bennett, A.B., Leonard, J.J., Bellingham, J.G. (1995) *Bottom Following for Survey Class Autonomous Underwater Vehicles*. Proc. Int. Symp. on Unmanned Untethered Submersible Technology, New Hampshire.
8. Zhang, F., Leonard, N.E. (2005) *Generating Contour Plots using Multiple Sensor Platforms*. Proc. IEEE Swarm Intelligence Symposium.

9. Ogren, P., Fiorelli, E., Leonard, N.E. (2002) *Formations with a Mission: Stable Coordination of Vehicle Group Maneuvres*. Proc. 15th Int. Symp. on Math. Theory of Networks and Systems.
10. Schill, F., Zimmer, U.R., Trumpf, J. (2004) *Visible Spectrum Optical Communication and Distance Sensing for Underwater Applications*. Proc. of ACRA, Canberra, Australia.
11. Schill, F., Zimmer, U.R., Trumpf, J. (2005) *Towards Optimal TDMA Scheduling for Robotic Swarm Communication*, Proc. of TAROS, London.
12. Schill, F., Zimmer, U.R., Trumpf (2006) *Effective Communication in Schools of Submersibles*, Proc. of OCEANS, Singapore.
13. Kottege, N., Zimmer, U.R. (2006) *MLS-based, Distributed Bearing, Range, and Posture Estimation for Schools of Submersibles*. Proc. of 10th Int. Symp. on Experimental Robotics, Rio de Janeiro.
14. Hormann, K., Spinello, S., Schroder, P. (2003) *C^1 -Continuous Terrain Reconstruction from Sparse Contours*, 8th Int. Fall Workshop on Vision and Visualization, Munich, Germany.
15. Egerstedt, M., Hu, X. (2001) *Formation Constrained Multi-Agent Control*. IEEE Trans. on Robotics and Automation, 17(6).
16. Egerstedt, M., Hu, X., Stosky, A. (2001) *Control of Mobile Platforms Using a Virtual Vehicle Approach*. IEEE Trans. on Automatic Control, 46(11).
17. Marshall, J.A., Lin, Z., Brouke, M.E., Francis, B.A. (2003) *A Pursuit Strategy for Wheeled-Vehicle Formations*. Proc. of 42nd IEEE Conf. on Decision and Control, Hawaii.
18. Sukumar, N., Malsch, E.A. (2006) *Recent Advances in the Construction of Polygonal Finite Element Interpolants*. Archives of Computational Methods in Engineering, 13(1).
19. Floater, M.S., Hormann, K., Kos, G. (2006) *A General Construction of Barycentric Coordinates over Convex Polygons*. Advances in Computational Mathematics, 24(1).
20. Meyer, M., Lee, H., Barr, A., Desbrun, M. (2002) *Generalized Barycentric Coordinates on Irregular Polygons*. Journal of Graphics Tools, 7(1).