*Proceedings of the IEEE/RSJ International Conference on Intelligent ROBots and Systems (IROS), September, 22-26, 2008, Nice, France*

# *Scale-Adaptive Polygonal Formations of Submersible Vehicles and Tracking Isocontours*

*Shahab Kalantar, Uwe R. Zimmer*

Australian National University
Research School of Information Sciences and Engineering
Canberra, ACT 0200, Australia
shahab.kalantar@rsise.anu.edu.au,uwe.zimmer@ieee.org

***Abstract*** – *Building on the assumption that identification of a sufficient number of isoclines of an environmental field (such as the ocean bottom terrain) allows efficient reconstruction of the field, in this paper, a sequel to [1], we describe a system where a group of robots in a spacial arrangement (a regular polygon centred around a lead robot) locally construct the field (measured at the locations of the robots) inside the polygonal area using interpolation by barycentric coordinates. If the error of interpolation is small enough, the corresponding isocline of the interpolated field will match the real isocline accurately enough. Tracking this isocline in a certain direction will then allow robust traversal of field isoclines. We use the measurement at the centre of the formation to adjust the size of the polygon to obtain desired accuracy.*

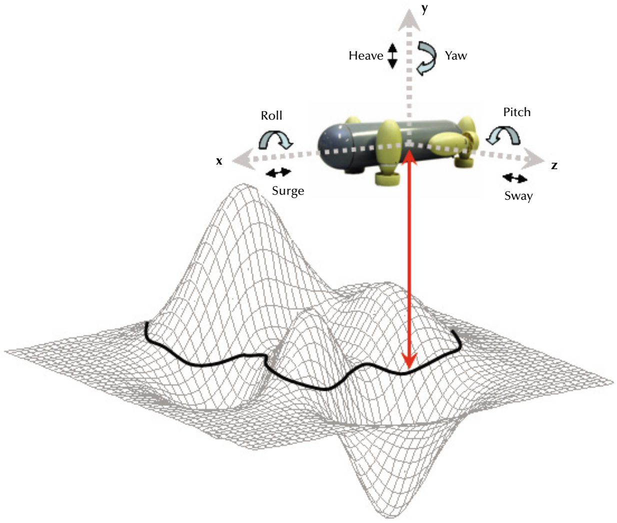

*Figure 1:* Underwater vehicle and tracking isoclines.

## 1. Introduction

Natural environments, in general, and the underwater world, in particular, abound with static or dynamic phenomena which can be modelled by mappings $f\colon D \to \mathbb{R}$, where $D \subseteq \mathbb{R}^n$ ($n \in \{2,3\}$) is a domain. Examples include ocean bottom terrain, chemical concentrations and plumes. Identification, characterization and reconstruction of these *fields* are among the most important tasks defined for autonomous vehicles. In this paper, we focus on computing *bathymetric* maps of the underwater terrain. One traditional way to do this is to have a submarine sweep an area along parallel lines [3]. More recently, researchers have suggested traversal of a sufficient number of *significant* isoclines of the field (e.g., those with higher slopes [4, 5]). Interpolation can then be used to reconstruct the field [9]. For smooth fields, gradient and Hessian information is all that is required [8, 7, 6]. For real environments, though, this information comes in the form of often unreliable estimates, even when multiple vehicles are deployed. In this paper, we employ a strategy based on polygonal interpolation and describe a scenario where a group of Serafina vehicles [2] collectively form a polygon around a specified leader robot. The leader robot, then, tracks an isoline of the interpolated field which corresponds to the actual one. This strategy is robust because the tracked path is smooth even on very rugged terrain and noisy measurements. It is also *scalable* as increase in the number of boundary robots does not affect the control strategy. On the other hand, in principle even only a single vehicle can be deployed (visiting vertices of an imaginary polygon), albeit at the expense of increased time and uncertainty. In the previous paper [1], the formation is only composed of robots on the polygon and they collectively move so that the centre of mass of the formation tracks the isoline. We also use a slightly different formation control strategy. In [1], we use an interpolation scheme which is only applicable to strictly regular polygons whereas here, we employ a method which works for *arbitrary* (*irregular*

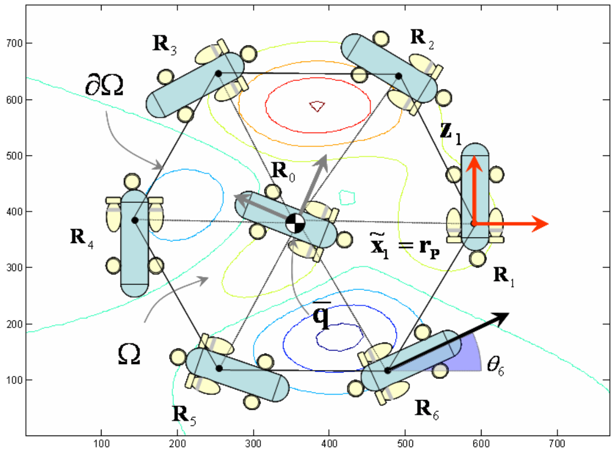

*Figure 2:* Polygonal formation with a central lead robot

but *simple*) polygons. This offers more robustness in face of uncertain inter-robot position measurements and formation keeping actuation commands. Finally, we use measurements from the lead robot to adaptively scale the size of the formation to decrease interpolation error.

We adopt a very simple model for the vehicles, i. e., a unicycle moving on a plane (figure 1). We can realize this by employing four decoupled controllers: a *heave controller* keeps the vehicle at a desired depth, *roll and pitch controllers* keeps these angles at zero, and a *surge-sway controller* moves the vehicle on the horizontal surface. We also assume the ambient flow to be negligible. For a clearer presentation, we only consider the purely kinematic case where control inputs for the robots are the linear and angular velocities ($v$ and $\omega$, respectively).

## 2. Polygonal Formations

Figure 2 shows the formation shape we use in this paper. Such a formation is composed of the lead robot $R_0$, $N$ boundary robots $R_1$, ..., $R_N$, and a desired radius $\mathrm{r_P}$. We denote the planar positions by $\mathrm{q}_i$ (Measured with respect to an inertial frame). We use $\mathrm{q_P}$ to denote the vector of all the positions of boundary robots. Such an arrangement can be described by a suitable *formation function* ([10]) $\mathrm{G_P}(\mathrm{q}_0,\mathrm{q_P})$ such that its *zero level* set gives a polygonal formation which is unique up to translation and rotation. (Sepulchre, *et al.*, 2007) describes control rules for forming a regular polygon around a beacon (the lead robot), starting from random positions. To keep this formations, each vehicle needs the distance to the lead robot as well as its two neighbours. Here, we parametrize the mission by $s$, arc-length of the path traversed by the lead robot, and we design the set of differential equations

$$\frac{\partial}{\partial t}\mathrm{q}_0 = \frac{\partial}{\partial s}\mathrm{q}_0(s)\,\dot{s},\ \frac{\partial}{\partial t}\mathrm{r_P} = \frac{\partial}{\partial s}\mathrm{r_P}(s)\,\dot{s}, \qquad (1)$$

which are the trajectory of the lead robot and the rate of change of the size of the formation, respectively. See [8] for a discussion of how to choose $s$.

## 3. Field Interpolation

We can use interpolation by *barycentric coordinates* [12] to construct a picture of the field covered by the polygonal area $\Omega_\mathrm{P}(s) \subset \mathbb{R}^2$ defined by $\mathrm{q_P}(s)$. Denote the boundary by $\partial\Omega$ and let $p \in \Omega_\mathrm{P}$. Consider a set of $N$ functions $\alpha_i\colon \mathbb{R} \otimes \mathbb{R}^{2N} \to \mathbb{R}^+$: and the vector

$$\alpha(p,\mathrm{q_P}) = [\alpha_0,\alpha_1,\ldots,\alpha_{N-1}]^T, \qquad (2)$$

where $\alpha_i \equiv \alpha_i(p,\mathrm{q_P})$.

Suppose that the following properties hold:

(1) $p = \alpha(p,\mathrm{q_P}) \cdot \mathrm{q_P}$ (*linear precision*),

(2) $0 \leq \alpha_i(p,\mathrm{q_P})$ (*convex combination*),

(3) $\alpha_i(p,\mathrm{q_P}) \leq 1$ (*bounded-ness*),

(4) $\sum_i \alpha_i(p,\mathrm{q_P}) = 1$ (*partition of unity*),

(5) $\alpha_i(p,\mathrm{q_P}) \in C^\infty$ (*infinite differentiability*),

(6) $\alpha_i(\mathrm{q}_j,\mathrm{q_P}) = \delta_{ij}$, $\delta_{ij}$ the delta function.

Then, $\alpha_i$'s are called $C^0$ shape functions associated with $p$, and $\alpha(p,\mathrm{q_P})$ its generalized barycentric coordinates.

Let $f\colon \mathbb{R}^2 \to \mathbb{R}$ 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 $f(\mathrm{q}_i(t)) = f_i(t)$. An *interpolation* scheme for $f$, based on polygon $\mathbf{P}$, is a function $\tilde{f}\colon \Omega_\mathrm{P} \to \mathbb{R}$ defined by

$$\tilde{f}(p) = \alpha_p(p,\mathrm{q_P})^T f_\mathrm{P}(t), \qquad (3)$$

where $f_\mathrm{P}(t) = [f_0(t),\ldots,f_{N-1}(t)]^T$. $\tilde{f}(p)$ is the interpolated value of $f(p)$. $\tilde{f}$ generates a vector of barycentric coordinates and takes the inner product of this vector and the vector composed of field values at the vertices. According to condition (5) above, $\tilde{f}(\mathrm{q}_i(t)) = f_i(t)$. Conditions (2) and (3) ensure that the interpolated values are bounded between the minimum and maximum of the nodal values: $\min_i\{f_i\} \leq \tilde{f}(p) \leq \max_i\{f_i\}$. Along the edges of the polygon, the interpolant must be piece-wise linear (i.e. $C^0$). This can be stated as

$$\begin{aligned}\tilde{f}(\tau) &= \tau f_i + (1-\tau)f_{i+1}\\ q &= \tau\mathrm{q}_i + (1-\tau)\mathrm{q}_{i+1}',\end{aligned} \qquad (4)$$

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

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

$$w(p,\mathrm{q_P}) \cdot (\mathrm{q_P} - p1_N) = 0, \qquad (5)$$

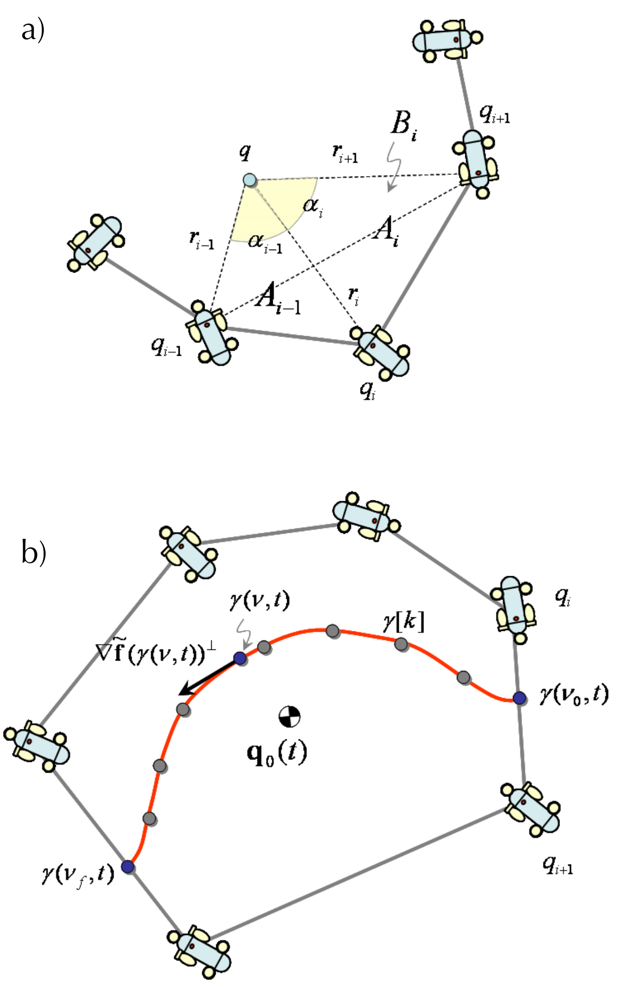

*Figure 3:* (a) Interpolation by barycentric coordinates. (b) Construction of the isocline.

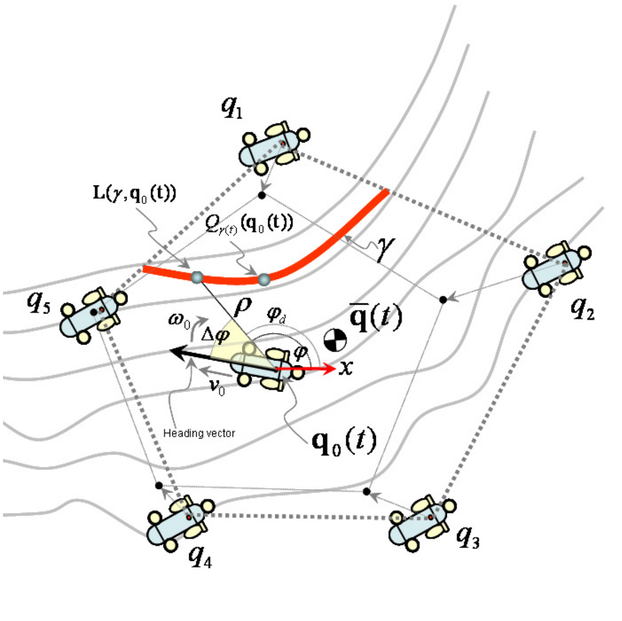

*Figure 4:* Control of the lead robot.

then partition of unity coordinates can be found by the formula

$$\alpha_i(p,\mathbf{q}_{\mathrm{P}}) = \frac{w_i(p,\mathbf{q}_{\mathrm{P}})}{\sum_k w_k(p,\mathbf{q}_{\mathrm{P}})}. \tag{6}$$

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

In this paper, we use weight functions which work for arbitrary but otherwise simple polygons [12]. Referring to figure 3.a and denoting $\mathbf{r}_i(p) = \mathbf{q}_i - p$, it can be seen that

$$\mathbf{A}_i(p) = \frac{1}{2}\|\mathbf{r}_i(p)\|\|\mathbf{r}_{i+1}(p)\|\sin\alpha_i, \text{ and} \tag{7}$$

$$\mathbf{B}_i(p) = \frac{1}{2}\|\mathbf{r}_{i-1}(p)\|\|\mathbf{r}_{i+1}(p)\|\sin(\alpha_{i-1}+\alpha_i). \tag{8}$$

It is shown in [12] that the weights

$$\begin{aligned} w_i(p) = {} & \mathbf{b}_{i-1}(p)\mathbf{A}_{i-2}(p) - \mathbf{b}_i(p)\mathbf{B}_i(p) \\ & + \mathbf{b}_{i+1}(p)\mathbf{A}_{i-1}(p) \end{aligned} \tag{9}$$

$$\mathbf{b}_j(p) = \frac{\|\mathbf{r}_i(p)\|}{\mathbf{A}_{i-1}(p)\mathbf{A}(p)}, \tag{10}$$

are homogeneous normalized barycentric coordinates of $\nu$ with respect to the polygon. This interpolation scheme has all the desired properties. It can be shown that

$$w_i(p) = \frac{2}{\|\mathbf{r}_i(p)\|}(\tan\frac{\alpha_{i-1}}{2} + \tan\frac{\alpha_i}{2}). \tag{11}$$

Note that

$$\tan\frac{\alpha_i}{2} = \frac{\|\mathbf{r}_i(p)\|\|\mathbf{r}_{i+1}(p)\| - \langle\mathbf{r}_i(p),\mathbf{r}_{i+1}(p)\rangle}{2\mathbf{A}_i(p)}. \tag{12}$$

This formula is used when $\mathbf{A}_i(p) \neq 0$. Otherwise, $p$ lies on the link connecting $\mathbf{q}_i$ and $\mathbf{q}_{i+1}$, in which case we can use the linearity of the interpolated field. Thus, for $p$ on or very close to the boundary, we set

$$\tilde{f}(p) = \tau(p)f(\mathbf{q}_i) + (1-\tau(p))f(\mathbf{q}_{i+1}), \tag{13}$$

$$\tau(p) = \frac{\|(p-\mathbf{q}_i(t))\cdot(\mathbf{q}_{i+1}(t)-\mathbf{q}_i(t))\|}{\|\mathbf{q}_{i+1}(t)-\mathbf{q}_i(t)\|}. \tag{14}$$

To construct the isoline, we need to compute the gradient

$$\nabla_p\tilde{f}(p) = \frac{\partial}{\partial p}\tilde{f}(p) = \sum_{i=0}^{N-1}\frac{\partial}{\partial p}\alpha_i(p,\mathbf{q}_{\mathrm{P}})f_i(t). \tag{15}$$

The computation is lengthy but tractable.

For a desired isocline value $\mathbf{c}_d$, the *interpolated isocline* $\gamma(t,\nu)$ ($\nu$ denoting the arc-length parametrization) is the solution of the equation $\tilde{f}(p) = \mathbf{c}_d$, $p \in \Omega_{\mathrm{P}}$, which can be easily computed. Referring to figure 3.b, The basic idea is to first find one of the end-points on $\partial\Omega$ using linearity along the edges and then trace the curve using $\nabla_p\tilde{f}(p)^{\perp}$ (we've used discrete samples of the curve; see [1] for more detail). If this solution is not a unique curve, the size of the polygon is deemed to be too big locally (see section 5).

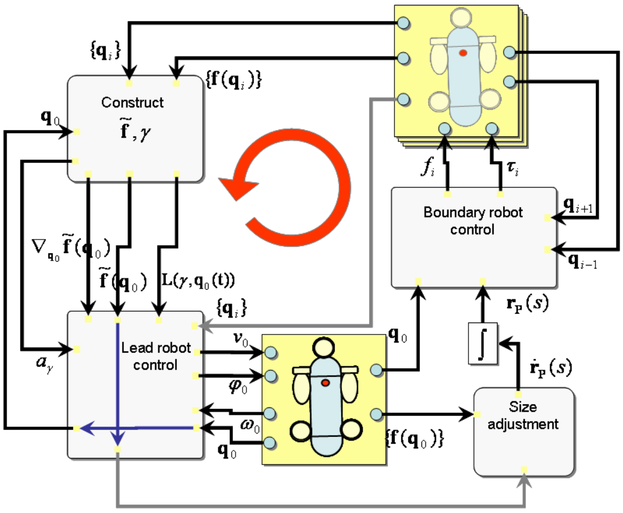

*Figure 5:* Closed loop control system

## 4. Tracking Control

To track the constructed isocline, we design control rules for the lead robot. Let $Q_{\gamma(t)}(q_0(t))$ denote the closest point on $\gamma(t)$ to $q_0(t)$, corresponding to the parameter value $\nu_Q$. One strategy is to track this point directly. A more robust method is to track a *look-ahead point*. Using a look-ahead makes tracking more robust to fluctuations of the path (whose observation changes over time). Referring to figure 4, we compute the *fixed* look-ahead point as

$$L(\gamma, q_0(t)) = \gamma(t, \nu_Q + D\nu_L), \tag{16}$$

where $\nu_L$ is the look-ahead length and $D$ determines direction of traversal. The control laws for linear and rotational movements of the lead robot are then [11]

$$\begin{aligned} v_0(s) &= a_\gamma \frac{v_n}{\rho_T} \| q_0(t) - L(\gamma, q_0(t)) \| \cos(\Delta\varphi) \\ &\quad + (1 - a_\gamma)\beta(c_d - \tilde{f}(q_0(t)))\, \nabla_{q_0(t)}\tilde{f}(q_0(t)) \end{aligned} \tag{17}$$

$$\begin{aligned} \varpi_0(s) &= a_\gamma(k\Delta\varphi + \dot{\varphi}_d) + (1 - a_\gamma) \cdot \\ &\quad k\left(\varphi_0 - \cos^{-1}\left(\frac{\nabla_{q_0(t)}\tilde{f}(q_0(t))}{\| \nabla_{q_0(t)}\tilde{f}(q_0(t)) \|}\right)\right) \end{aligned} \tag{18}$$

where $v_n$ is a nominal speed, $\rho_T$ is the desired maximum distance between the look-ahead point and $q_0$, $\varphi_d$ is the desired heading towards the look-ahead, $\Delta\varphi = \varphi_0 - \varphi_d$, $\beta$ and $k$ are positive gains. $a_\gamma = 1$ if $\gamma$ exists and is zero otherwise. These commands implicitly define $\partial q_0(s) / \partial s$ and correspond to the purely kinematic version of the system where dynamic effects are assumed negligible. A more elaborate approach [11] would be to define the look-ahead as an independently moving *virtual path tracker* $L(\nu)$ and modulate its speed with an expression such as

$$\dot{\nu} = \frac{1}{\| \dot{\gamma} \|} e^{\alpha(\rho_T - \rho(t))} v_n, \; \rho(t) = \| q_0 - L(\nu) \|. \tag{19}$$

However, due to the fact that the constructed curve changes with each observation, it is not clear how to define initial positions for the look-ahead and the way it is to jump between two consecutive curves, whose parametrizations differ. See [14] for a possible solution. The described system forms a closed loop. Boundary robots collectively move in such a way that their aggregate centre of mass track the lead robot while forming a regular polygon, the lead robot tries to track the virtual path tracker while regulating its speed based on feedback from the rest of the formation, the virtual path tracker tracks the path while controlling its speed through feedback from the lead robot, and finally, the path itself is updated as a result of motion of the boundary robots (figure 5).

## 5. Scale Adaptation

The accuracy of the proposed method relies on the error of interpolation (assuming this error is much larger than those of sensory measurements). It turns out that this error is bounded by the size of the polygon and the second variation of the underlying field. For a polygonal formation P with size $r_P$ and $N$ vehicles, we have that

$$\| f - \tilde{f} \|_{L^2(\Omega)} \le 4Cr_P^2 \| f \|_{2,L^2(\Omega)} \tag{20}$$

where the $L^2$-norms are defined by

$$\| u \|_{L^2(\Omega)} = \sqrt{\int_\Omega u^2(x, y)\, dxdy}, \text{ and} \tag{21}$$

$$\| u \|_{2,L^2(\Omega)} = \sqrt{\int_\Omega (u_{xx}^2 + u_{xy}^2 + u_{yy}^2)\, dxdy}. \tag{22}$$

This is a straightforward extension of a similar theorem for error of interpolation by triangular finite elements [15]. If, in a sufficiently small neighbourhood around $q_0$, $\| f \|_{2,L^2(\Omega)}$ is bounded by some threshold, then interpolation error can be decreased by shrinking the size. Note, however, that too small a formation may raise safety issues, as well as slower operation. We use the field measurement by the leader robot to assess the accuracy of interpolation by boundary robots. This is a good practical approximation. We simply keep shrinking the formation until the error falls below some desired margin. Note that, we this method, decreasing the size of the formation may increase the error if variation in the second derivative is not small enough.

Let $\hat{r}_P$ and $\check{r}_P$ denote some upper and lower bounds, and $f^*$ the estimated maximum value of the field in normal operating conditions. It is required that the formation expand to maximum if the error of interpolation falls below some tolerance bound $\mathbf{e}_T$ which is expressed as the percentage $\mathbf{e}_T = \mu f^*$. The absolute error is defined to be

$$\mathbf{e}_I(t) = \left|\tilde{f}(\mathbf{q}_0(t)) - f(\mathbf{q}_0(t))\right|. \tag{23}$$

We use a moving average $\bar{\mathbf{e}}_I(t)$ of this error. Define the normalized error by

$$\tilde{\mathbf{e}}(t) = \frac{\bar{\mathbf{e}}_I(t) - \mathbf{e}_T}{(1-\mu)f^*}. \tag{24}$$

Note that $0 \leq \tilde{\mathbf{e}}(t)u(\tilde{\mathbf{e}}(t)) \leq 1$, where $u$ is the step function. The differential equation governing the evolution of the side length is

$$\dot{\mathbf{r}}_P(t) = \frac{k\mathbf{r}^e_P(t) - \beta\tilde{\mathbf{e}}(t)u(\tilde{\mathbf{e}}(t))}{D_1 u(\dot{\mathbf{r}}_P(t)) + D_2(1-u(\dot{\mathbf{r}}_P(t)))}, \tag{25}$$

where $D_1$ and $D_2$ are tunable damping coefficients and $\mathbf{r}^e_P(t) = \hat{\mathbf{r}}_P - \mathbf{r}_P(t)$. $\mathbf{r}_P(t)$ is also saturated from below. The values of the spring constant $k$ and the compressing force gain $\beta$ determine when and how the forces are balanced. As a rule of thumb, the force trying to expand the formation should be weaker than that trying to shrink it. This will decrease the chances of missing a high curvature area.

## 6. Simulations

For the simulations, we have used superimposition of Gaussians corrupted with random noise. Figure 6 shows two sample runs of the described system. Compare the field inside the polygons with the interpolated contour lines. Figure 7.a shows a snapshot of a simulation run for which we have chosen $p = 0.01$, $D_1 = D_2 = 1$, $k = 1/50$, $\beta = 300$, $\hat{\mathbf{r}}_P = 150$, and $\check{\mathbf{r}}_P = 50$. Figure 7.b and 7.c show the plots of averaged absolute errors $\tilde{\mathbf{e}}(t)$ and sizes. Figure 7.d shows the sequence of polygons generated by the formation. Note again that the curvature in environment depicted in the simulation changes too quickly with respect to the average size of the formation. As can be seen from the trace, the rate of change of curvature is much higher than the settling time of the differential equation governing size change. In some positions, the formation expands because it is experiencing low curvatures only to enter a high curvature zone shortly afterwards.

## 7. Conclusions

In this paper, we discussed a very robust adaptive method for tracking field isoclines. We only considered deterministic interpolation error. Detailed probabilistic modelling of sensing errors such as inter-robot distances and field observation (which is affected by depth and altitude measurements in the case of terrains) would be highly desirable.

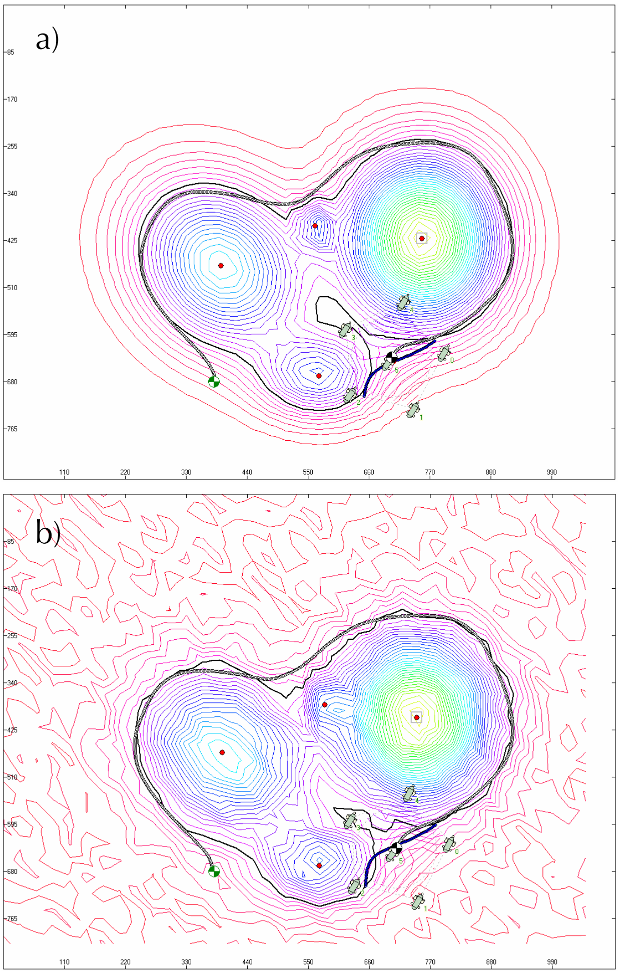

*Figure 6:* Two sample simulation runs. (a) Smooth terrain. (b) Rugged terrain.

Concerning the moving look-ahead, we have to find methods to relate the position of the tracker on the previous observed path to its postulated position on the current one. Finally, strategies for adaptively tuning the parameters to cope with high curvature environments remains a research topic.

## References

[1] Kalantar, S., Zimmer, U., *Tracking Environmental Isoclines using Polygonal Formations of Submersible Autonomous Vehicles*, Submitted to the 6th Intl. Conf. on Field and Service Robotics (FSR'07), Chamonix, France, 2007.

[2] http://serafina.anu.edu.au/

[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 Altmeter, Project SUMARE Deliverable 4.1 (http://www.mumm.ac.be/SUMARE/), 2002.

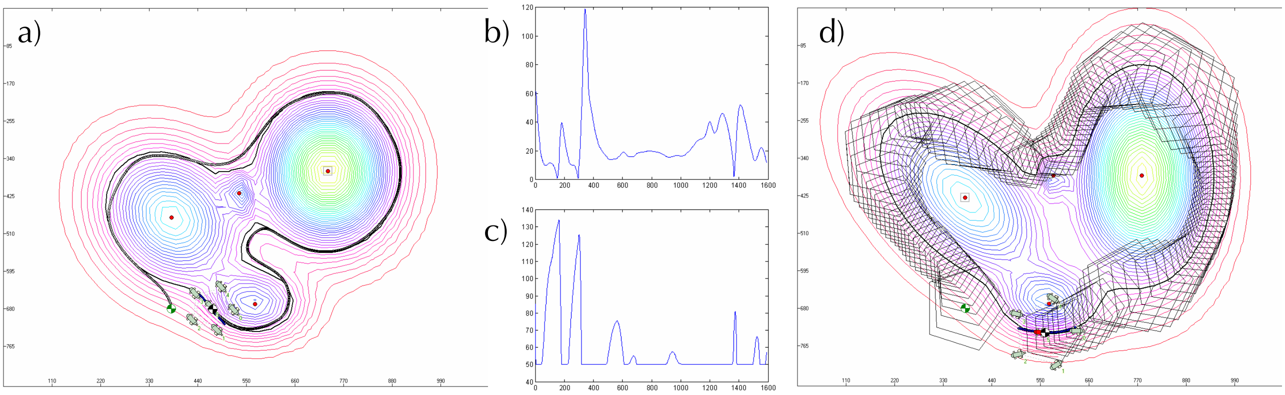

*Figure 7:* Size adjustment. (a) Run. (b) Errors. (c) Sizes. (d) Polygons.

[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] Zhang, F., Leonard, N.E., *Generating Contour Plots using Multiple Sensor Platforms*, Proc. IEEE Swarm Intelligence Symp., 2005.

[8] 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.

[9] Hormann, K., Spinello, S., Schroder, P., *Continuous Terrain Reconstruction from Sparse Contours*, 8th Int. Fall Workshop on Vision and Visualization, Munich, Germany, 2003.

[10] Egerstedt, M., Hu, X., *Formation Constrained Multi-Agent Control*, IEEE Trans. on Robotics and Automation, 17(6), 2001.

[11] Egerstedt, M., Hu, X., Stosky, A., *Control of Mobile Platforms Using a Virtual Vehicle Approach*, IEEE Trans. on Automatic Control, 46(11), 2001.

[12] Meyer, M., Lee, H., Barr, A., Desbrun, M., *Generalized Barycentric Coordinates on Irregular Polygons*, Journal of Graphics Tools, 7(1), 2002.

[13] Nguyen, B.Q., Yao-Ling Chuang, Tung, D., Chung Hsieh, Zhipu Jin, Ling Shi, Marthaler, D., Bertozzi, A., Murray, R.M., *Virtual attractive-repulsive potentials for cooperative control of second order dynamic vehicles on the Caltech MVWT*, Proc. of American Control Conf., 2005.

[14] Macek, K., Petrovic, I., Siegwart, R., *A Control Method for Stable and Smooth Path Following of Mobile Robots*, Proc. of 2nd European Conf. on Mobile Robots, 2005.

[15] Hughes, T.J.R., *The Finite Element Method*, Prentice-Hall, Stanford, 1987.