ON COLLABORATIVE OPTIMIZATION AND COMMUNICATION FOR A TEAM OF AUTONOMOUS UNDERWATER VEHICLES

Alberto Speranzon *** **Jorge Silva** *
João Borges de Sousa ** **Karl Henrik Johansson** ***

* *Instituto Superior de Engenharia do Porto, Rua Dr. António Bernardino de Almeida N. 431, 4200-072 Porto,Portugal*
** *Faculdade de Engenharia da Universidade do Porto, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal*
*** *Dept. of Signals, Sensors & Systems, Royal Institute of Technology, SE-100 44 Stockholm, Sweden*

Abstract: A multi-vehicle search strategy for finding an optimum of a scalar based on the simplex search algorithm is proposed. The strategy is described as hierarchical scheme with two layers. A team controller (upper layer) is described by a discrete-event system. The output of this layer is a set of waypoints for the vehicles and it is used by the vehicle controller (lower layer) to drive each vehicle to the next waypoint. The vehicles can communicate sensor data. Since underwater communication is costly in terms of energy, a protocol that reduces the average communication load is considered. Simulations are carried out in order to evaluate the performance of the search strategy for varying fields and levels of measurement noise.

1. INTRODUCTION

The problem of coordination and control of multiple heterogeneous vehicles has recently attracted the attention of researchers in control engineering and computer science. There are several aspects to this problem such as sensing capabilities, layered control strategies, stability of geometric formations and control under communication constraints, just to name a few. In this paper we are concerned with a specific multi-vehicle control problem: given a scalar field, coordinate the motions of a set of vehicles with sampling capabilities to find its minimum (maximum) in a given region. Here we report our investigations concerning the implementation of the fixed-size simplex search algorithm [12]. The simplex algorithm is a direct search method used in many practical optimization problems. Its simplicity and robustness properties [8, 10] makes it an interesting algorithm for minimum search applications with multiple vehicles. The simplex method is usually applied in situations where the cost of gradient estimation is high. The method behaves as a gradient descent method even if no explicit gradient calculation is needed. In spite of its wide application, there are few cases for which it possible to prove convergence of the simplex algorithm, for a scalar field with dimension two or higher. It was, actually, shown that the original Nelder-Mead algorithm [10] (in which the size of the simplex can change over time) can converge to a non-stationary point even for quite smooth and strictly convex functions [9]. The simplex algorithm is useful to improve an initial estimate of the solution in few iterations without explicit estimation of the gradient and with few function evaluations. An informal description of a multi-vehicle search strategy based on the simplex algorithm is proposed in [6]. Here we formalize the search strategy and discuss some of its properties. Our objective is to use the simplex algorithm to progress towards a minimum and to get as close as possible to it. In order to improve the estimate of the solution obtained with the simplex algorithm we can also resort to other kinds of search strategies combined with dynamic estimation. It is not the objective of this paper to analyze the final

Fig. 1. Autonomous Underwater Vehicle used in the PISCIS project at Porto University.

phase of the search procedure, but we plan to use the proposed algorithm as part of a global search strategy. Optimization algorithms have been used as the inspiration for other multi-vehicle search strategies. Bachmayer et al. [1] use a pure gradient-based method for scenarios where a vehicle platoon searches the minimum of a convex and smooth scalar fields. Burian et al. [2] report results with mixed strategies and present illustrative examples using real data, such as depth profiles of a lake. However, these results are drawn for single vehicle operation.

The main contributions of this paper are the definition of a new motion coordination strategy for the vehicles performing the search operation and a communications protocol for the implementation of the simplex algorithm.

Our work is mainly motivated by the PISCIS project [4] at the Underwater Systems and Technology Laboratory (USTL), Porto University, but the results can be applied also to other vehicles and scenarios. The PISCIS project offers an experimental testbed consisting of two small size autonomous underwater vehicles (Figure 1) with environmental sensors and acoustic modems for underwater communications.

The paper is organized as follows. A general hierarchical model for the multi-vehicle control system is presented in Section 2. It consists of two parts: a team controller and a vehicle controller. In Section 3 we describe how to map the simplex algorithm to such hierarchical control structure. We first construct a team controller that implements the simplex search algorithm. Starting from the construction of such team controller we then extend it to the case of two independent team controllers which synchronize thought a communication channel. Since communication underwater requires a large amount of power we discuss some communication issues in Section 4. In Section 5 we describe the dynamic model of the underwater vehicles. Simulation results are reported in Section 6 and some conclusions are drawn in Section 7.

2. HIERARCHICAL MULTI-VEHICLE MODEL

Consider a compact convex set $\Omega \subset \mathbb{R}^2$ containing the origin. Define a field through a scalar-valued map $V : \Omega \to \mathbb{R}$ with a global minimum at the origin. Let $n \geq 2$ vehicles be positioned in $p_i \in \Omega$, $i = 1, \ldots, n$. Each vehicle can take measurements and communicate them to the other vehicles. Based on the measurements the vehicles are supposed to find the minimum

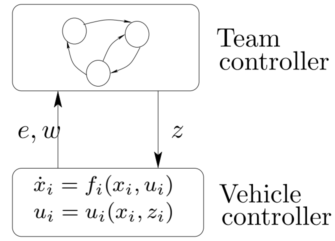


Fig. 2. Hierarchical control structure for n vehicles.

of V. In practice there are limitations on how often measurements can be taken and how accurate the communication is. We propose a hierarchical control strategy [5, 13, 14] with two layers: an upper layer, called team controller, modeled by a discrete-event system and a lower layer modeled by a continuous-time control system called vehicle controller, as shown in Figure 2. The discrete layer generates waypoints for the autonomous underwater vehicles (AUV's) according to an optimization algorithm. The continuous layer uses waypoints as target points to be reached by the AUV's. The continuous-layer therefore generates feasible trajectories for the AUV's which connects waypoints. We assume that the vehicles are able to exchange information through a communication channel.

In the next two subsections we review some definitions and properties of discrete-event systems and discuss how their interaction with continuous-time systems create a hierarchical control structure.

2.1 Discrete layer

The discrete layer is modeled by a discrete-event system [3].

Definition 2.1. A discrete-event system is a quintuple

$$D = (Z, E, W, \xi, z_0) \tag{1}$$

where E is the alphabet of events $E = \{e_0, e_1, e_2, \ldots\}$, Z is the discrete state space, W the set of inputs, $\xi : E \times Z \times W \to Z$ the transition function, and $z_0 \in Z$ the initial state.

The transition function ξ defines the evolution of the discrete-event system, i.e., it maps the current state to next state once an event happens. Note that in our definition of a discrete-event system, the transition function depends also on the input set W. In the following we recall two important concepts for discrete-event systems that we will use later in the paper.

Definition 2.2. The *language generated* by $D = (Z, E, W, \xi, z_0)$ is defined as

$$\mathcal{L}(D) = \{(e, w) \in (E \times W)^* : \xi(e, z_0, w) \text{ is feasible}\}$$

where $w = (w_1, w_2, w_2) \in W$ is a vector of three real values.

Definition 2.3. Given two discrete-event systems D_1 and D_2, they are *equivalent* if $\mathcal{L}(D_1) = \mathcal{L}(D_2)$.

2.2 Continuous layer

The continuous layer represents the dynamics of the vehicles and the continuous-time control algorithms. Each vehicle $i = 1, \dots, n$ is described by a control system

$$\dot{x}_i = f_i(x_i, u_i), \qquad u_i \in U_i$$

where $f_i : X_i \times U_i \to \Omega$ defines the dynamics of the individual vehicles with continuous state x_i and admissible continuous controls in U_i. The control u_i is a state feedback that depends on both the continuous state and the state of the discrete-event system in the discrete layer z_i i.e.,

$$u_i = u_i(x_i, z_i).$$

The interactions between the two layers are described in the following.

3. TEAM CONTROLLER

The search strategy described in this paper is based on the simplex algorithm which we describe in some detail in the next subsection. We then show how it is possible to design a team controller, modeled as a discrete-event system, that executes the simplex algorithm. An equivalent implementation with communicating team controllers is then designed.

3.1 Simplex search

The simplex algorithm is a direct search method used in many practical optimization problems. It is usually applied in situations where the cost of function evaluation is high and gradient calculation is difficult, as happens in scalar field corrupted by noise or time-varying. The algorithm is useful to improve an initial estimate of the solution with few function evaluations. Its simplicity and robustness properties [8, 10], make it an interesting algorithm for minimum search applications with multiple vehicles. Notice that the widely used gradient based methods cannot cope with the existence of noise in the field. Moreover, the main objective in this kind of application is not an algorithm which converges in few iterations but one which enhances the synergy between the vehicles given the problem constraints.

Let us define a triangular grid $\mathcal{G} \in \Omega$ as depicted in Figure 3, with aperture $d > 0$. Introduce an arbitrary point $p_0 \in \Omega^\circ$ and a base of vectors given by b_1, b_2 such that $b_1^T b_1 = b_2^T b_2 = d^2$ and $b_1^T b_2 = d^2 \cos \pi/3$. The grid is then equal to

$$\mathcal{G} = \{p \in \Omega |\ p = p_0 + k b_1 + \ell b_2,\ k, \ell \in \mathbb{Z}\}.$$

A simplex $z = (z_1, z_2, z_3) \in \mathcal{G}^3$ is defined by three neighboring vertices of $\mathcal{G}$, which belong to a triangle.

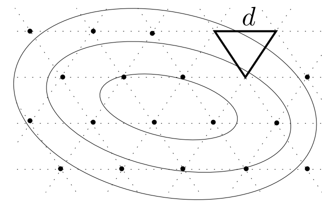


Fig. 3. A triangular grid with aperture d over a two-dimensional scalar field depicted by its level curves. The solid line triangle illustrates the state z of the discrete-event system evolving on the grid.

We suppose, without loss of generality, that $V(z_3) \geq V(z_i)$, $i = 1, 2$. Given a simplex $z = (z_1, z_2, z_3)$ the next simplex, z', is generated from z by reflecting z_3 with respect to the other vertices, namely

$$z \mapsto z' = \mathcal{S}(z_1, z_2, z_3') \qquad z_3' = z_1 + z_2 - z_3 \qquad (2)$$

where $\mathcal{S}(.)$ defines the following simplex algorithm:

$z(0) \leftarrow (z_1(0), z_2(0), z_3(0))$
$k \leftarrow 0$
while true **do**
  $i \leftarrow \arg\max_i V(z_i(k))$
  $z_i' \leftarrow z_j + z_h - z_i$ with $j, h \in \{1, 2, 3\}$ and $j \neq h, j \neq i, h \neq i$
  $z_j' \leftarrow z_j$
  $z_h' \leftarrow z_h$
  $z(k+1) \leftarrow (z_1', z_2', z_3')$
  $k \leftarrow k + 1$
  if $k \geq 2 \wedge z(k) = z(k-2)$ **then**
    stop
  end if
end while

Notice that the algorithm terminates at time instance $k = N$ with $N \geq 2$, if $z(N) = z(N-2)$. Since the algorithm is deterministic, it follows that a continuation after step N would lead to an oscillation between the two discrete states $z(N)$ and $z(N-1)$.

Definition 3.1. A fixed point χ for the simplex algorithm is a pair of two simplexes that makes the algorithm to terminate. Thus

$$\chi = \{(z(k-1), z(k)) \in \mathcal{G}^3 \times \mathcal{G}^3 : z(k-2) = z(k), k \geq 2\}.$$

3.2 Simplex search as team controller

Let $D = (Z, E, w, \xi, z_0)$ be a discrete-event system modeling the team controller. We will design D so that it implements the simplex search algorithm.

Let the state space of the discrete-event system be $Z \subset \mathcal{G}^3$ such that $z = (p_1, p_2, p_3) \in Z$ is a simplex. The event alphabet E consists of a single enabling event, $\{e\}$, triggered by the underlying continuous-time layer as shown in Figure 2. Such event is generated when the submarines reach the waypoints (z_1, z_2), vertices of the simplex z. The activation of such event is discussed in detail in section 5.

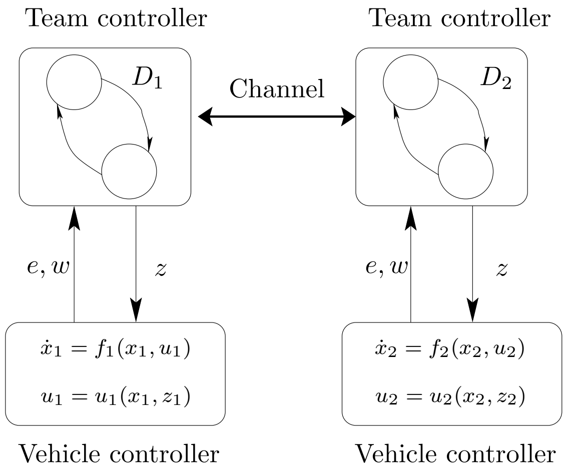


Fig. 4. Decentralized hierarchical control structure for $n = 2$ vehicles with a communication channel.

When an event is triggered the input w is also generated. It consists of the value of the scalar field V at the AUV's current position, which we indicate with p_1, p_2 and p_3. Thus $w = (V(p_1), V(p_2), V(p_3)) \in W$ is the ordered triple of measurements.

Remark 3.2. We assume that the value of the third vertex of the simplex is know to both vehicles. We will discuss later in this section how this value is available to the AUV's.

The transition function $\xi(e, z, w)$ of the discrete-event system is defined as follows

$$\xi(e,z,w) = \begin{cases} (p'_3, p_2, p_1), & \text{if } w_3 \geq \max\{w_1, w_2\} \\ (p_1, p'_2, p_3), & \text{if } w_2 \geq \max\{w_1, w_3\} \\ & \text{and } w_2 \neq w_3 \\ (p'_1, p_2, p_3), & \text{if } w_1 > \max\{w_2, w_3\} \end{cases} \tag{3}$$

where p'_k with $k \in \{1, 2, 3\}$ is the reflected vertex as defined in (2). The following proposition follows form construction.

Proposition 3.3. Let $D = (Z, E, W, \xi, z_0)$ with $z(0) = z_0$ be the team controller previously defined. Let $\mathcal{Z} = \{z(0), \ldots, z(N)\}$ be a sequence of simplexes generated by the simplex algorithm, such that $(z(N-1), z(N))$ is a fixed point. Then the language generated by D is $\mathcal{L}(D) = \mathcal{Z} \cup \{z(N-1), z(N)\}^*$.

Thus the team controller implements the simplex and in particular z is the set of waypoints where the AUV's need to move in order to find the minimizer of the field V. Such waypoints are used by the continuous-time layer to compute feasible trajectories for the AUV's to reach the vertices of the new simplex.

3.3 Communicating team controllers

Consider now the case with two communicating AUV's, see Figure 4. Let $\ell_i \in L_i \subset \mathbb{R}^2$ be the data locally available to the ith AUV, in particular $\ell_i = (V(p_i), V(p_3), i = 1, 2$. At each step the ith AUV needs to have some information about $V(p_j)$, $j \in 1, 2$, $j \neq i$ in order to compute the next simplex. We assume that such information is available through a communication channel. Let $c_i \in C_i \subset \mathbb{R}$ be the measurement received by the ith AUV, namely $c_i = V(p_j)$, with $j \neq i$. Notice that when the two AUV's have exchanged measurements they have available the data (l_i, c_i).

We, then,define the following two team controllers that generate waypoints for the AUV's

$$D_i = \big(Z, E_i, (L_i, C_i), \xi_i, z_0\big), \qquad i = 1, 2 \tag{4}$$

The event alphabet contains a common enabling event, triggered by the continuous-time layer, thus $E_i = \{e\}$. In other words, the two AUV's observe the same enabling event e which is triggered when both vehicles have reached their previously computed waypoint. The transition function ξ_i is defined as follows

$$\xi_1(e, z_1, (\ell_1, c_1)) = \begin{cases} (p'_3, p_2, p_1), & \text{if } V(p_3) \geq \max\{V(p_1), c_1\} \\ (p_1, p'_2, p_3), & \text{if } c_1 \geq \max\{V(p_1), V(p_3)\} \\ & \text{and } c_1 \neq V(p_3) \\ (p'_1, p_2, p_3), & \text{if } V(p_1) > \max\{V(p_3), c_1\} \end{cases} \tag{5}$$

and,

$$\xi_2(e, z_2, (\ell_2, c_2)) = \begin{cases} (p'_3, p_2, p_1), & \text{if } V(p_3) \geq \max\{V(p_2), c_2\} \\ (p_1, p'_2, p_3), & \text{if } V(p_2) \geq \max\{V(p_3), c_2\} \\ & \text{and } V(p_2) \neq V(p_3) \\ (p'_1, p_2, p_3), & \text{if } c_2 > \max\{V(p_2), V(p_3)\} \end{cases} \tag{6}$$

Proposition 3.4. The team controller D and the team controller obtained by composing D_1 and D_2 are language equivalent: $\mathcal{L}(D) = \mathcal{L}(D_1 \| D_2)$.

The result follows by construction. The parallel composition of D_1 and D_2 is the discrete-event system

$$D_p = D_1 \| D_2 = \big(Z \times Z, \{e\}, (L_1 \cup C_1) \cup (L_2 \cup C_2), \xi_p, (z_0, z_0)\big).$$

where the initial state z_0 is the same simplex for both AUV's. Note that $(L_1 \cup C_1) \cup (L_2 \cup C_2) = W$, defined as for the centralized scheme. Since the enabling event e is observed by both discrete-event system then the transition function ξ_p is defined as follows

$$\xi_p(e, (z_1, z_2), w) = \big(\xi_1(e, z_1, w), \xi_2(e, z_2, w)\big)$$

with $w \in W$ and where $(z_1, z_2) \in Z \times Z$. Since the AUV's know the position of each other and $\xi_1 = \xi_2$ it follows that if the initial condition z_0 is the same for D_1 and D_2 then their future states are the same, i.e. $z_1(k) = z_2(k)$ for all $k \geq 0$. This implies that $\xi(e, z_1, w) = \xi(e, z_2, w)$, thus $\xi_p(e, (z_1, z_2), w)$ represents a single state of Z. Then we conclude that $\mathcal{L}(D_p) = \mathcal{L}(D)$.

Remark 3.5. The team controllers specify how the discrete-event systems compute a new simplex. However, they do not define which AUV moves to the reflected point. Such decision is implicitly defined by the transition functions ξ_i (cf. equations (5) and (6)) if we interpret the next state $\xi_i(e, z_i, (\ell_i, c_i))$ as an ordered tuple, namely the next waypoint for the ith AUV is the ith element of $\xi_i(e, z_i, (\ell_i, c_i))$, with $i = 1, 2$.

4. COMMUNICATION ISSUES

Underwater communication is very costly in terms of energy since the SNR is generally very low [11]. From the transition functions (5) and (6) we notice that both AUV's need to know the value of the field V at all the three vertices of the current simplex in order to generate the next simplex. This is achieved transmitting measurement using underwater acoustic modems. We propose a communication protocols in which measurements and very short (in term of bits/symbols) synchronization codes are used. In doing this the average data rate is decreased compare to a protocol where the raw measurements are communicated.

If the communicated data are not available then the discrete event systems D_1 and D_2 are non-deterministic. This fact follows from that there are three possible states $z_i' = \xi_i(e, z_i, (\ell_i, \emptyset))$, that can be reached.

We consider the following communication strategy. At each step the first AUV sends its measurement, $V(p_1)$ to the second. This AUV then replies using a message γ_k depending on the comparison of the field's value at the vertices of the current simplex. Summarizing we have

message	condition
γ_1	$V(p_3)$ is the largest
γ_2	$V(p_2)$ is the largest
γ_3	$V(p_1)$ is the largest

The messages γ_k need tp be designed very small, in term of bits/symbols, compared with a measurement.

Observe that the second AUV, after receiving $V(p_1)$, knows the field's value at all vertices of the current simplex, thus it can determine uniquely the message γ_k, $k = 1, \ldots, 3$.

The team controllers require the AUV's to know the value of the field V at the common point p_3 at each step. The communication strategy suggested, implicitly assigns the role of master to one of the two AUV's. Thus this constraint can be relaxed and it is necessary that only the master always knows the field's value at that point. We consider as example a possible case that could occur. Suppose $V(p_3)$ is the largest value. This case is shown in Figure 6(a). The field assumes largest value at p_3. Thus the next simplex is $\{p_1, p_2, p_3'\}$. With the communication protocol proposed, the vehicle in p_2 receives the measurement taken by the vehicle in p_1, namely $V(p_1)$, shown with

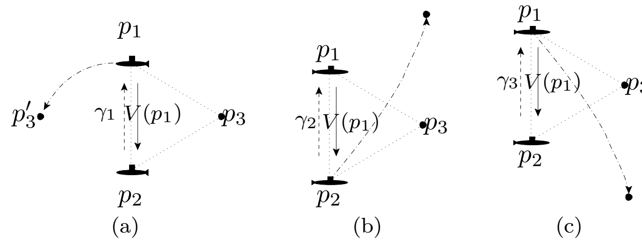


Fig. 6. Motion of the AUV's and communicated date for two different scenarios. In dotted line is shown the current simplex. With the arrowed solid line is represented the communication of raw measurement, while with a dashed line is represented the transmission of a message. In deash-dotted line is shown the trajectory of the AUV's toward the destination vertexes.

a solid arrows line. At this point the vehicle in p_2 has all the measurements available and computes which vertex of the current simplex should be reflected. The message γ_3 is then communicate the the vehicle in p_1. The transition functions ξ_i give

$$\xi_2(e, p_2, \{V(p_1), V(p_2), V(p_3)\}) = (p_3', p_2, p_1)$$

and,

$$\xi_1(e, p_1, \{V(p_1), \gamma_3, V(p_3)\}) = (p_3', p_2, p_1).$$

Thus the next simplex is the ordered triple (p_3', p_2, p_1). This means that vehicle in p_1 will need to move to p_3' and vehicle p_2 should remain standing still.

At the next step the value of $V(p_1)$ is known only to the vehicle in p_2, but after the communication of $V(p_3')$ it can compute which of the vertexes should be reflected and transmit back the corresponding message.

The other two cases, shown in Figure 6(b) and Figure 6(c), follow similarly.

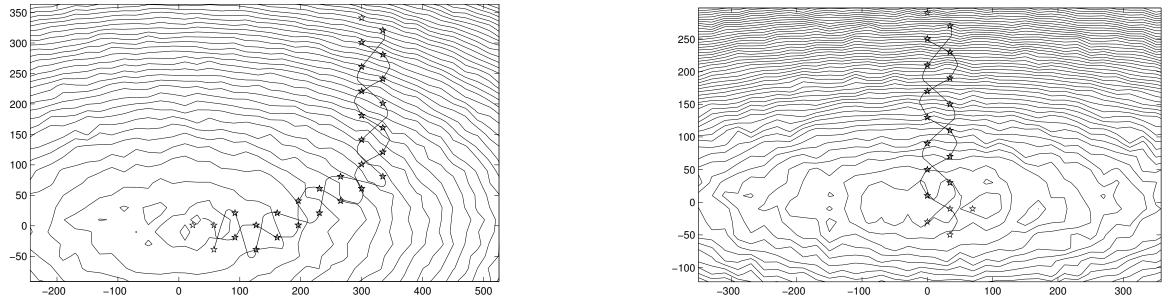
5. VEHICLE CONTROLLER

A complete nonlinear model of a Autonomous Underwater Vehicles can be found in [7]. In this work we consider the problem of controlling the AUV's on a plane. The nonlinear model of the system is

$$\begin{aligned}\dot{x}_i &= V\cos\psi_i \\ \dot{y}_i &= V\sin\psi_i \\ \dot{\psi}_i &= r_i \qquad\qquad i = 1, 2\end{aligned}$$

where $(x_i, y_i)^T$ is the position of the ith AUV with respect to a global coordinate frame, ψ_i is the yaw angle and r_i is the yaw rate. The velocity V is assumed constant. The guidance in the horizontal plane is achieved using a "line of sight" control law: at each time step the vehicle is commanded to head towards the reference waypoint z_i. The continuous-time controller for the yaw is a PID

$$r_i(x_i, z_i) = k_p\varepsilon_i + k_i\int_0^t \varepsilon_i(\tau)d\tau - k_d\frac{d\psi_i}{dt}$$



(a) (b)

Fig. 5. Trajectories of two AUV's controlled by the hierarchical control algorithm proposed in the paper.

where

$$\varepsilon_i = \angle(z_i - (x_i, y_i)^T) - \psi_i$$

The event e is generated when the vehicles reach the assigned waypoints. Due to the vehicles' control limitations, in practice, it is not possible to assure that the vehicles will reach the exact waypoint. To overcome this difficulty, when a vehicle reaches a neighborhood of radius δ of the assigned vertex of the simplex, a new measurement is taken and the event e is triggered. Therefore, the values are not always sampled at the exact grid intersections. Additionally, position estimation errors may lead to sampling being done even further from the desired point. Monte Carlo simulations, reported in the following section, have been performed in order to test the robustness properties of the algorithm performing.

6. SIMULATION

In this section, we present simulation results to illustrate the implementation of the simplex based search strategy, namely the interaction between team and vehicle controllers. We consider the two autonomous underwater vehicles identical.

In what concerns the size of the simplex (or grid aperture) we are interested in setting it as small as possible because the smaller the simplex the closer we can get to the minimizer. However, the grid aperture is limited below by the dynamic behavior of the vehicle, namely the vehicles' turning radius. The considered simplex size is $d = 40$ m. Positioning errors were modeled by considering a worst case estimation error and by enlarging the acceptance neighborhood in order to encompass this error. In practice, each the vehicle performs a local filtering of the acquired samples along its trajectory. This was also considered in our simulations allowing higher levels of sensor and field noise.

Figure 5 shows a simulation run for the scalar field $V(p_1, p_2) = p_1^2 + 4p_2^2$ with gaussian noise superimposed, modeling measurement noise, and vehicles departing from the vertexes of the simplex with centroid at (320,320) (m). Simulations were stopped when the simplex reached a fixed point. The vertexes of the successive simplexes are marked with stars. The same scenario, with different noise realizations, was simulated 100 times. We also considered three different noise levels Results are collected in Table 1. As we can notice the vehicles are able to arrive very close to the minimizer. On the other hand, a fixed point is reached much earlier when the noise level is high as shown in the "Completion Time" row.

We have also simulated the behavior of the proposed control structure when the scalar field is time-varying. In Figure 7, we have simulated a scalar field that drifts at constant speed (a rough simulation of the stream's effect on a temperature field, for example). Figure 7 shows four snapshots of the evolution of the AUV's. As illustrated in the figure, the vehicles are able to move very close to the minimizer of the field.

7. CONCLUSIONS

In this paper we have considered the problem of finding the minimizer of a scalar field using a team of autonomous underwater vehicles. We proposed a hierarchical control strategy in which the high level consists of a team controller, modeled as a discrete-event system, which generates waypoints according to the simplex algorithm. The low level consists of a vehicle controller which generates continuous-time control commands that move the AUV's between waypoints.

A communication protocol have been designed in order to reduce the average bit-rate. Simulation results have been carried out to show how the hierarchical control system moves the AUV's towards the minimizer of a time-invariant and a time-varying vector field.

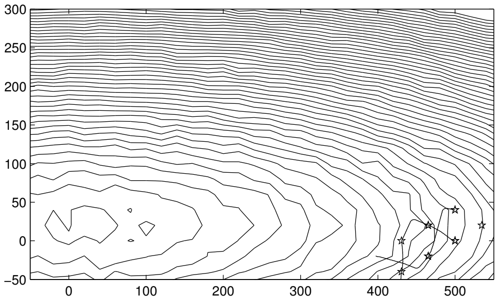
REFERENCES

[1] Ralf Bachmayer and Naomi Ehrich Leonard. Vehicle networks for gradient descent in a sampled environment. In *IEEE Conference on Decision and Control*, 2002.

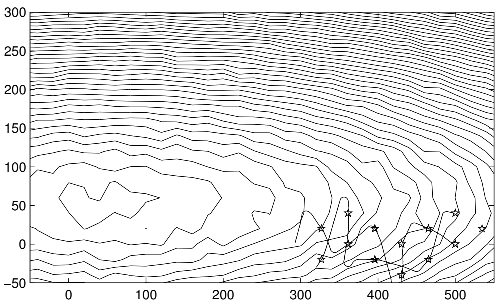
[2] E. Burian, D. Yoerger, A. Bradley, and H. Singh. Gradient search with autonomous underwater vehicles using scalar measurements. In *IEEE*

Table 1. Simulation Results (Average/Standard Deviation)

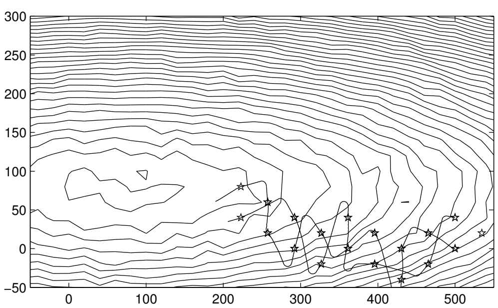
	Noise level (Std. Deviation)			
	0	1000	2500	5000
Completion time (s)	627	609/48	615/69	585/83
Total travelled distance (m)	1632	1585/124	1600/180	1525/218
Distance to minimizer (m)	22	32/16	40/22	75/41



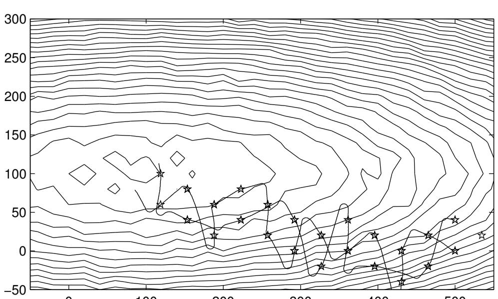
(a) Situation after 6 time steps.



(b) Situation after 12 time steps.



(c) Situation after 18 time steps.



(d) Situation after 24 time steps.

Fig. 7. The figures show the simplex, the trajectories of the AUV's and a moving scalar field at different time steps. The scalar field, shown through level curves, is a quadratic function with superimposed white noise.

Symp. Autonomous Underwater Vehicle Technology, pages 86–89, 1996.

[3] C.G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Kluwer Academic, 1999.

[4] N. Cruz, A. Matos, J.B. de Sousa, F.L. Pereira, J. Silva, E.P. Silva, J. Coimbra, and E.B. Dias. Operations with multiple autonomous underwater vehicles: the piscis project. In *Proceedings of the Second Annual Symposium on Autonomous Intelligent Networks and Systems*, 2003.

[5] N.G. Datta, J. Lygeros, and S. Sastry. Hierarchical hybrid control: a case study. In *Hybrid Systems*, pages 166–190, 1994.

[6] J. Borges de Sousa and Fernando Lobo Pereira. On coordinated control strategies for networked dynamic control systems - an application to auvs. In *Proceedings of International Symposium of Mathematical Theory of Networks and Systems (MTNS)*, 2002.

[7] T. I. Fossen. *Guidance and Control of Ocean Vehicles*. John Wiley and Sons Ltd., 1994.

[8] Jeffrey C. Lagarias, James A. Reeds, Margaret H. Wright, and Paul E. Wright. Convergence properties of the nedler-mead simplex in low dimensions. *SIAM Journal of Optimization*, 9(1):112–147, 1998.

[9] K.I.M. McKinnon. Convergence of the nelder mead simplex method to a nonstationary point. *SIAM Journal of Optimization*, 148-158, 1999.

[10] J.A. Nelder and R. Mead. A simplex method for function minimization. *Computing Journal*, 7:308–313, 1965.

[11] J. Proakis, E. Sozer, J. Rice, and M. Stojanovic. Shallow water acoustic networks. *IEEE Communications Magazine*, 39(11):114–119, 2001.

[12] W. Spendley, G.R. Hext, and F.R. Himsworth. Sequential applications of simplex designs in optimization and evolutionary operation. *Technometrics*, 4:441–461, 1962.

[13] P. Varaiya. Theory of hierarchical, multilevel systems. *IEEE Transaction on Automatic Control*, 17(2):280–281, 1972.

[14] P. Varaiya. A question about hierarchical systems. In In T. Djaferis and I. Schick, editors, *System theory: modeling, analysis and control*. Kluwer, 2000.