

# Experimental Validation of an Algorithm for Cooperative Boundary Tracking

Chung H. Hsieh<sup>1</sup>, Zhipu Jin<sup>2</sup>, Daniel Marthaler<sup>3</sup>, Bao Q. Nguyen<sup>1</sup>, David J. Tung<sup>4</sup>  
Andrea L. Bertozzi<sup>3</sup>, Richard M. Murray<sup>5</sup>

**Abstract**—This paper demonstrates the testbed implementation of a simple algorithm for cooperatively locating and following environmental boundary in two-dimensional space. The algorithm alternates between clockwise and counter-clockwise motion along circular paths. By switching the orientation of rotation upon crossing the boundary, a vehicle is able to continue along its search. Coordination is achieved by changing vehicle speeds depending on the location of other vehicles.

## I. INTRODUCTION

This paper demonstrates the testbed implementation of a simple algorithm for cooperatively locating and following environmental boundary of any substance capable of being measured by an onboard sensor. Such general problems are of current interest for unmanned vehicle operations with specific applications ranging from coastal algae blooms [2,11], chemical plumes [5,8], and adaptive ocean sampling [7], to future applications including oil spills or hazardous chemicals.

Automated algorithms for boundary tracking are the subject of ongoing research in many fields. Many of these methods are based on simple cooperative searching through data sets. In image processing for example, image snakes [9] search to locate areas where there are sudden sharp variations in pixel intensity. The snake tries to find such areas along smooth edges in an image. This method was recently shown in simulation [13] to be adaptable to cooperative UAV boundary tracking using a platform of vehicles that possess adequate sensors for measuring spatial gradients. However, typical sensors for UAVs are often binary detectors [5,8] or at best scalar sensors. Kemp *et al* [10] proposed the “UUV-gas” algorithm as a method to track the boundary with such constraints on modern UUVs. This algorithm is designed to run on a platform in which each individual vehicle has a controller that automates either clockwise or counterclockwise motion around a nearby circle with a prescribed radius and center. There is nothing inherent about aquatic applications in this algorithm and

<sup>1</sup>Dept. of Electrical Engineering, University of California Los Angeles, Los Angeles, CA 90095 chung\_hsieh@ieee.org, bknguyen@ucla.edu

<sup>2</sup>Dept. of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125 jzp@cds.caltech.edu

<sup>3</sup>Dept. of Mathematics, University of California Los Angeles, Los Angeles, CA 90095 {daniel, bertozzi}@math.ucla.edu

<sup>4</sup>Dept. of Computer Science, University of California Los Angeles, Los Angeles, CA 90095 davetung@ucla.edu

<sup>5</sup>Control and Dynamical Systems, California Institute of Technology, Pasadena, CA 91125 murray@cds.caltech.edu

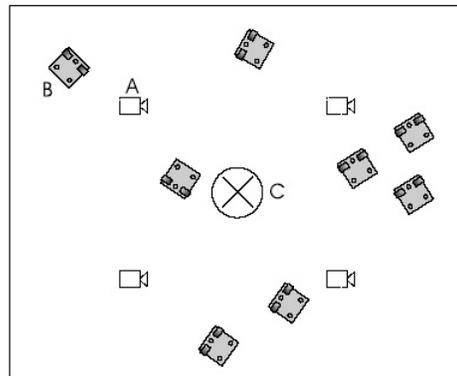


Fig. 1. An overhead view of the Caltech Multi-Vehicle Wireless Testbed (MVWT) [3]. (A) The overhead camera of the lab positioning system. (B) The vehicles. (C) A post at the center of the testbed.

indeed we test it on a land platform with second order vehicles.

The full boundary tracking algorithm uses simple circular motion as a building block for a composite path that eventually travels the entire boundary of the region. Communication between vehicles allows them to maintain adequate spacing in order to spread out around the perimeter of the region of interest. This paper demonstrates the UUV-gas boundary tracking algorithm on the Multi-Vehicle Wireless Testbed (MVWT) [3] at the California Institute of Technology (Caltech). The testbed’s second order control vehicles employ a circle tracking controller discussed below.

## II. MULTI-VEHICLE WIRELESS TESTBED

### A. Experimental Platform

The Caltech MVWT [3] consists of a lab positioning system composed of four CCD cameras (Fig. 1A) mounted on the ceiling and an image processor that determines the vehicle’s location and orientation. This information is transmitted wirelessly via IEEE 802.11b to individual vehicles (Fig. 1B). There is a center post (Fig. 1C) that remains from a previous unrelated experiment. The testbed represents the first quadrant of the Cartesian coordinate system with dimension of 6.7 meters by 7.3 meters.

The vehicles used for this experiment are named Kelly (Fig. 2,3). A Kelly is powered by two ducted fans separated by length  $2r_f$  producing forces  $F_R$  and  $F_L$  for the right and left fan, respectively. The orientation is represented by the angle  $\theta$ . Kelly has onboard a 700Mhz laptop with 802.11b



Fig. 2. Frontal view of the MVWT vehicle "Kelly"

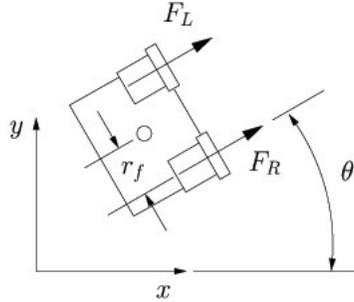


Fig. 3. An overhead representation of the MVWT vehicle "Kelly" [3].

capability. The Kelly is a non-linear second order dynamic vehicle whose  $(x, y)$  coordinate motion and orientation  $\theta$  are described by the following equations:

$$m\ddot{x} = -\mu\dot{x} + (F_R + F_L) \cos(\theta), \quad (1)$$

$$m\ddot{y} = -\mu\dot{y} + (F_R + F_L) \sin(\theta), \quad (2)$$

$$J\ddot{\theta} = -\psi\dot{\theta} + (F_R - F_L)r_f, \quad (3)$$

where  $\mu$  is the linear coefficient of friction and  $\psi$  is the rotational friction. The mass and inertial moment of the vehicle are  $m$  and  $J$ .

### B. Vehicle controller for circular motion

The UUV-gas algorithm discussed in the next section requires a controller for individual vehicle motion in a circle with prescribed speed, radius, and orientation. Simply prescribing fan speeds on the vehicle without the assistance of an closed loop controller causes the vehicle to become unstable from any disturbances such as imperfections on the floor, drop position packets from the wireless system, and even turbulence from another vehicle's push fans. Therefore, a local closed loop controller for following a circular path is used to maintain stability. The controller follows a radius  $r_{ref}$ , with a speed  $s_{ref}$  about a center  $\vec{C}$  at  $(x_c, y_c)$ .

The equilibria of (1-3) are constant position and orientation with zero velocity. However, the linearized dynamics are not controllable around such equilibria. We consider the error dynamics around a constant velocity and use this to approximate circular motion through piecewise linear patches. We take  $\dot{x}_e, y_e, \dot{y}_e, \theta_e, \dot{\theta}_e$  as the state variables and

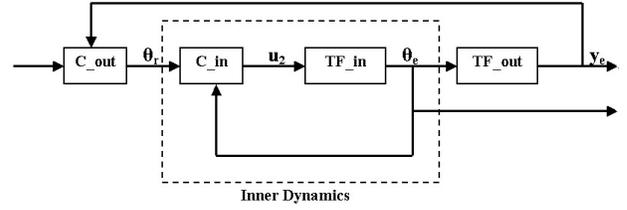


Fig. 4. Classical controller for the MVWT vehicle.

linearize the dynamics at the equilibrium point  $[C, 0, 0, 0, 0]$ , and obtain the linearized error dynamics as:

$$\begin{bmatrix} \ddot{x}_e \\ \dot{y}_e \\ \ddot{y}_e \\ \dot{\theta}_e \\ \ddot{\theta}_e \end{bmatrix} = \begin{bmatrix} -\frac{\mu}{m} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{\mu}{m} & \frac{C\cdot\mu}{m} & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -\frac{\psi}{J} \end{bmatrix} \begin{bmatrix} \dot{x}_e \\ y_e \\ \dot{y}_e \\ \theta_e \\ \dot{\theta}_e \end{bmatrix} + \begin{bmatrix} \frac{1}{m} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{r_f}{J} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (4)$$

where  $u_1$  and  $u_2$  are propulsion and torque in the error dynamics.

At each particular time, we measure the current states, calculate the reference state in polar coordinates, and determine the error vector  $e$ . Here we assume that the current and reference states have the same angle with respect to the circle's center. We use the local controller to generate the control laws, i.e. the fan forces. Below we describe two types of controllers designed for the Kelly vehicle, a classical controller and a LQR controller. For the trajectory, the equilibrium point is  $[r_{ref} \cdot \omega, 0, 0, t \cdot \omega + \theta_0, \omega]$ . Here note that  $\dot{x}$  as a constant is meant to locally approximate the boundary of a circle while  $\dot{\theta}$  as a constant is maintained throughout the circular path.

1) *Classical Controller*: According to the linearized error dynamics, the dynamics between  $\dot{x}_e$  and  $u_1$  is totally decoupled from other state variables and inputs. So we can view this system as two parts:

- Speed dynamics. The speed  $\dot{x}_e$  is totally controlled by  $u_1$ . This is a first-order subsystem.
- Departure dynamics. The departure  $y_e$  are determined by  $\theta_e$ , and  $\theta_e$  is driven by  $u_2$ . This is a two-layer second-order subsystem.

The speed dynamics are described by a first-order transfer function that decouples from other variables, so we use a PI controller. For the departure dynamics, we need an inner controller for  $\theta_e$  and an outer controller for  $y_e$ . Fig. 4 shows how this strategy works. If the inner layer responds quicker than the outer layer, then we can design the controllers separately.

2) *LQR Controller*: We also have a LQR controller based on the error dynamics. We select the diagonal matrices  $Q$  and  $R$  for any particular circle trajectory and compute the gain matrix  $G$  by using the MATLAB function  $LQR(A, B, Q, R)$ . The controller law is

$$\begin{aligned} F_e &= -G \cdot e, \\ F_{nom} &= \begin{bmatrix} F_{L,nom} \\ F_{R,nom} \end{bmatrix}, \\ &= \begin{bmatrix} 0.5(\sqrt{(mr_{ref}\omega^2)^2 + (\mu r_{ref}\omega)^2} - \frac{\psi\omega}{r_f}) \\ F_{L,nom} + \frac{\psi\omega}{r_f} \end{bmatrix}, \\ F &= \bar{F}_e + F_{nom}. \end{aligned}$$

Note that the nominal fan forces  $F_{nom}$  are different since the vehicle keeps turning when it follows a circle. In the experimental results presented later in this paper, the classical controller is used.

### III. UUV-GAS ALGORITHM

The UUV-gas algorithm is designed for vehicle platforms with sensors capable of either binary state detection or scalar readings. Even with scalar sensor it is still very difficult to have a reliable gradient estimator from the sensors. The UUV-gas is gradient-free method for tracking the boundary.

#### A. UUV-gas theory

The UUV-gas algorithm has two parts. The first is a single vehicle boundary tracking method comprised of circular motion and the second is a cooperative motion algorithm to space apart multiple vehicles.

1) *Single vehicle UUV-gas*: Assuming that only a binary state sensor is available, the vehicle travels around an arc in a clockwise direction when inside the region of interest and counterclockwise when outside the region of interest. That is:

$$\frac{d\theta}{dt} = \begin{cases} +\omega & \text{inside,} \\ -\omega & \text{outside,} \end{cases} \quad (5)$$

where  $\theta$  is the heading of the vehicle and  $\omega$  is the angular rate of change. The speed of the vehicle  $s_{ref}$  is constant,  $s_{ref} = r_{ref}\omega$ . Following this simple process, the vehicle is able to follow the boundary using knowledge from binary sensors of whether it is inside or outside the region.

The stability, convergence and coverage of the UUV-gas algorithm is discussed in detail in [10]. We briefly discuss the important features. First, UUV-gas assumes a priori knowledge of a point  $X_p$  inside the region. If the vehicle starts outside the region it will head toward this point. Second, as the vehicle turns in and out of the boundary, the radius of curvature  $r$  must be less than the minimum radius of curvature  $R_c$  of the boundary ( $r < R_c$ ) to guarantee convergence. Third, even if stable and convergent, the UUV-gas may not give total coverage of the boundary as shown in Fig. 6. To ensure total coverage, there is a minimum distance  $D_m$  between any two points on the boundary, where  $D_m > 2\pi R_c$ .

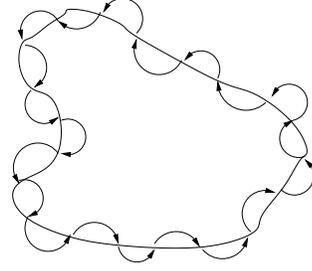


Fig. 5. Shows the path of the vehicle as it goes in and out of the boundary.

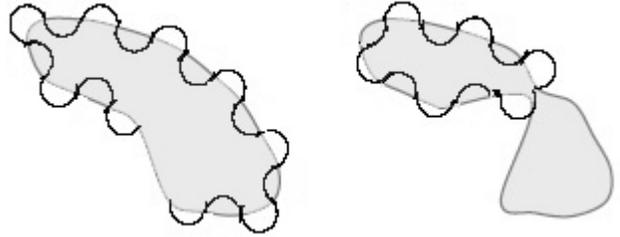


Fig. 6. Incomplete coverage of the boundary due to a narrow section in the region [10].

2) *Multiple vehicles*: With  $N$  vehicles each running the gradient-free boundary tracking algorithm, they will converge on and track the boundary. However, their spacing may become quite nonuniform, possibly causing collisions. The UUV-gas algorithm spreads the vehicles by changing the vehicle's speed rather than its velocity [10]:

$$\left| \frac{dZ_n}{dt} \right| = U_o \left( 1 + g t_n \cdot \sum_{m \neq n} f(Z_m - Z_n) \right), \quad (6)$$

$$f(Z) = \frac{Z}{|Z| \exp\left(\frac{-|Z|}{\lambda}\right)}. \quad (7)$$

The speed without other vehicles is  $U_o$ , and  $t_n$  is a vector tangent to the boundary,  $f$  is the repulsion vector,  $g$  is the speed gain, typically between 0.1-0.5, with higher values producing faster spreading [10]. The speed update should occur less frequently than boundary crossings.

#### B. UUV-gas implementation on MVWT

The Caltech Multi-Vehicle Wireless Testbed (MVWT) is an excellent platform to test the merits of the UUV-gas algorithm, since many path following controllers are available on the platform (e.g. section II-B).

Instead of simply varying  $\frac{d\theta}{dt}$ , the UUV-gas is modified to move its center of rotation (CoR) each time the vehicle crosses the boundary. Fig. 7 shows the path of the vehicle, and the projection of a new center each time it makes a crossing. The vehicle is aware of its CoR and crossing point  $\vec{B}$  at coordinate  $(x_b, y_b)$  (Fig. 7A). A unit vector  $\vec{Z}$  from

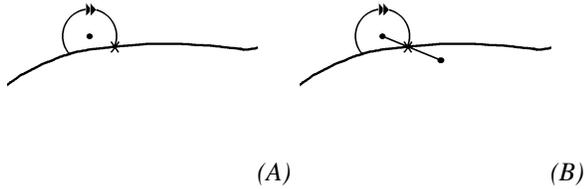


Fig. 7. Projecting a new center of rotation using the previous center and crossing point.

the center to the crossing point is found by:

$$R = \sqrt{(x_c - x_b)^2 + (y_c - y_b)^2}, \quad (8)$$

$$\vec{Z} = \frac{(x_b - x_c)\hat{x} + (y_b - y_c)\hat{y}}{R}, \quad (9)$$

after which a new CoR can be found simply by  $\vec{B} + r_{ref}\vec{Z}$ , which results in a new CoR along  $\vec{Z}$  distance  $r_{ref}$  away (Fig. 7B). In our implementation of the cooperative UUV-gas algorithm, we use the CoR instead of vehicle position to determine the speed ( $s_{ref}$ ). This method is more stable because the CoR does not vary as rapidly as the actual vehicle position.

#### IV. EXPERIMENTAL RESULTS

Our experimental tests are conducted using virtual binary state sensors. The virtual sensors use the lab positioning system to determine the vehicle position and check against a virtual region we defined on the testbed. If the vehicle coordinates are inside the region, the virtual sensor is define as having detected the region and not detected the region when outside. As the testbed has limited space and has a post at its center, we consider simple regions such as a circle and a composite region of overlapping circles.

##### A. Single Vehicle

Fig. 8 shows the path of a vehicle tracking a virtual region composed of one circle located at  $(3, 3.5)$  with radius of  $1.7m$ . The dotted line is the actual vehicle path, with the square marking the start and triangle marking the end. Only the first lap around the boundary is presented for clarity. The radius ( $r_{ref}$ ) of the prescribed path (the small circles) is  $0.5m$  with speed ( $s_{ref}$ ) of  $0.3m/s$ . The dots in the small circles are the center of rotation (CoR). This is an early experiment, but there are a few key items of interest. The vehicle response to the circle controller has transient errors when the CoR and direction of rotation changes. The jagged edges of the path are a result of the vehicle drifting while a new CoR is projected and the vehicle locking on the new circle. Even with these deviations the UUV-gas method still works. Throughout testing, only when the vehicle become completely unstable (e.g. goes into uncontrollable spin) does the UUV-gas method fail.

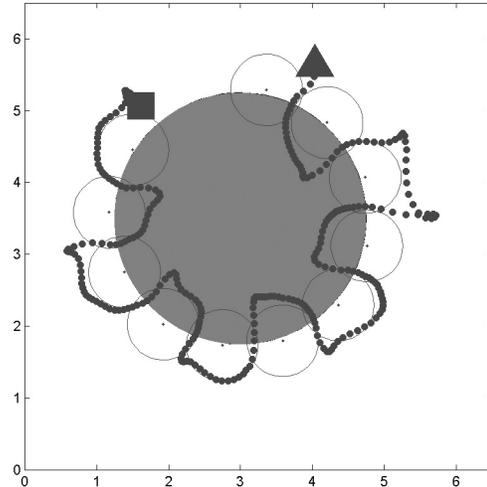


Fig. 8. Vehicle path plot for a boundary represented by a circle of radius  $1.7m$ . The circle controller uses a radius of  $0.5m$  with a vehicle speed of  $0.3m/s$ . The axes are measured in meters.

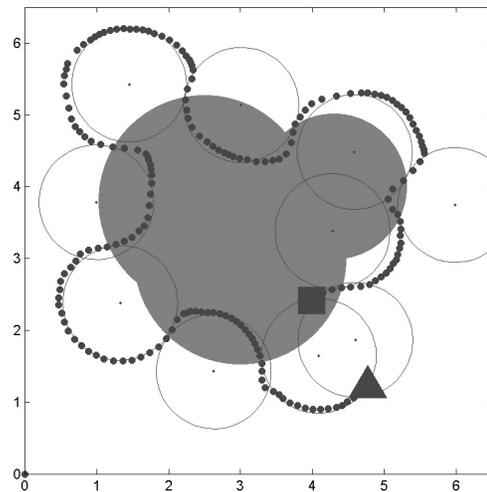


Fig. 9. Vehicle path plot of boundary represented by three intersecting circles. The local circle controller uses a radius of  $0.8m$  with a vehicle speed of  $0.15m/s$ . Square is the starting point and triangle is the ending point. The axes are measured in meters.

Fig. 9 shows a virtual region composed of three circles. One circle centered at  $(4.3, 4)$  with radius of  $1m$ , the other two circles with radius of  $1.5m$  centered at  $(3, 3)$  and  $(2.5, 3.8)$ . The large dotted line is the actual path with the square marking the start and triangle marking the end of the run. Only the first lap around the boundary is presented to prevent overlap of vehicle path. The small circles are the local reference path of the vehicle with radius  $0.8m$ . The vehicle travels at  $0.15m/s$ . The fans are shut off during boundary crossing to control the drifting noted in Fig. 8. From Fig. 9 we can see these changes keep the vehicle approximately  $r_{ref}$  from the CoR, and the actual path in Fig. 9 is closer to that of the ideal UUV-gas method.

## B. Multiple vehicles

Fig. 10 and Fig. 11 are data plots from an experiment with three Kelly vehicles simultaneously exploring the same virtual region as Fig. 9. The three vehicles are initially placed randomly outside the virtual region. The experiment lasted 160 seconds, only the first 120 seconds are presented to prevent overlapping of vehicle path for clarity.

For the vehicles to act cooperatively, they must communicate with each other. In the UUV-gas implementation only the CoR coordinates and the last crossing coordinates are transmitted from one vehicle to another. Each vehicle chooses its reference speed  $s_{ref}$  based on other vehicles' CoR. All the known boundary coordinates are stored in each vehicle for redundancy. The two sets of coordinates total 128bits of data or four floating-point numbers. We purposely limit the data transmitted over the 802.11b for two reasons. The first is to remain faithful to the spirit of the UUV-gas, which has low-bandwidth communication as one of its constraints [10]. The second is to not introduce additional delay into the local control loop. The vehicle receives its position data via the same 802.11b with a period of  $1m.s$  and communicates at a period of  $16m.s$ , thus keeping the transmitted data at a minimum is beneficial to the local controller.

Fig. 10 shows the evolution of the CoR spacing over time. Using the CoR in UUV-gas does not cause a rapid change to  $s_{ref}$  which contributes to keeping the overall system stable. Note in Fig. 10B that the vehicle's CoR can get close to each other immediately after a crossing event. At this time the trailing vehicle slows down to avoid collision with the lead vehicle. In this manner UUV-gas serves to keep the vehicles spaced apart.

Fig. 11 shows the actual path of each vehicle. The paths are plotted individually for clarity. The square, diamond, circle and triangle represent time at 1, 40, 80 and 120 seconds respectively and is presented as a common frame of reference with Fig. 10. What is interesting about the three different vehicle paths is while vehicle 1 and vehicle 3 (Fig. 11A, 11C) are traversing the boundary smoothly, vehicle 2 (Fig. 11B) is not. The UUV-gas method is robust enough to compensate for variations in vehicle performance.

In the time it took the trailing vehicle, the slowest due to the distributional aspect of the UUV-gas, to complete one lap around the boundary, the three vehicles collected enough points to form a recognizable boundary (Fig. 11D). Compared to individual crossing points of Fig. 11A, 11B or 11C the utility of cooperative vehicles in boundary tracking is evident.

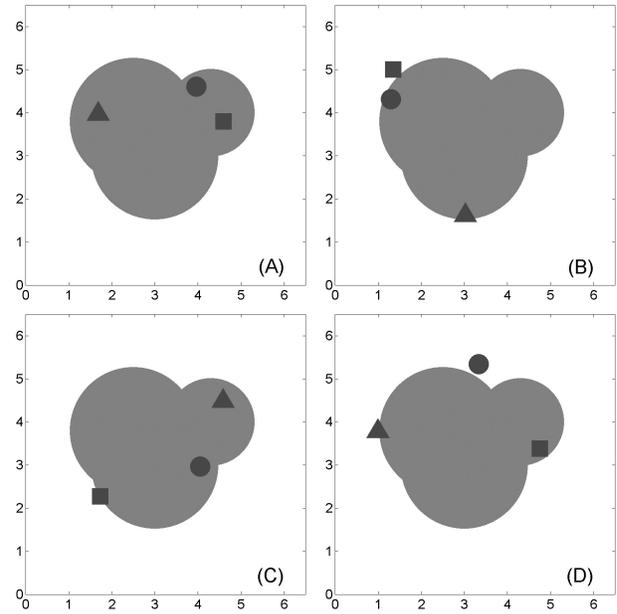


Fig. 10. Cooperative boundary tracking on MVWT testbed. Plots of three vehicles' center of rotation (CoR) at (A) 1 second, (B) 40 seconds, (C) 80 seconds, (D) 120 seconds. The square, circle and triangle represent vehicles 1, 2 and 3 respectively, each executing single vehicle boundary tracking and vehicle spacing using UUV-gas. The axes are measured in meters.

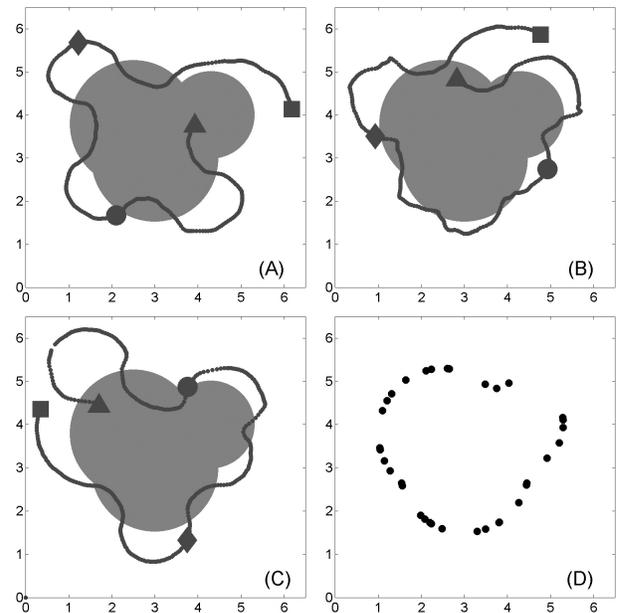


Fig. 11. Individual path of (A) Vehicle 1, (B) Vehicle 2, (C) Vehicle 3 while cooperatively searching the boundary in the same experiment as Fig. 10. The square, diamond, circle, and triangle represent 1, 40, 80, 120 seconds respectively of each vehicle's path. (D) Union of boundary crossing point from the three vehicles over 160 seconds. The axes are measured in meters.

## V. CONCLUSION

The testbed implementation of the UUV-gas algorithm illustrates several properties. First, the individual vehicle boundary tracking is stable and can be implemented on any platform possessing a controller for circular motion and a binary sensor to measure presence of the environmental field. Fig. 8 demonstrates that the single agent algorithm is robust to local deviations from the circle tracking algorithm. Using the GAS method, multiple vehicle searches work well, however in real field applications local obstacle avoidance is also important. Such avoidance will cause deviations in the motion planning; if these deviations are on the same order of magnitude as the errors shown in Fig. 8 then the algorithm will similarly be robust to obstacles. One aspect of the field not tested here is the robustness of the algorithm to sensor noise. This is discussed theoretically in [10,13] and is an interesting point for further testbed research. Another aspect is the two-dimensionality of the method which makes it appropriate for applications on the surface of the ocean or over land. Perimeter surveillance of the ground from the air is also a possible application. Another interesting point for further research is the extension to fully three-dimensional environmental fields.

Any autonomous vehicle with gradient or gradient-free sensor capable of circular motion in both clockwise and counterclockwise directions can employ this method as basis for perimeter surveillance. The use of the Kelly vehicle on the MVWT shows that this method can successfully run on second order control vehicles which are typically found in underwater or aerial applications.

## ACKNOWLEDGMENTS

We thank Ling Shi for helpful comments. This work was supported by ONR grant N000140410054, ARO grant DAAD19-02-1-0055, NSF grants ACI-0321917 and DMS-9983726.

## REFERENCES

- [1] J. G. Bellingham, H. Schmidt, C. Chryssostomidis, 1998, *Real-time oceanography with autonomous ocean sampling networks*, ONR annual report, available on-line at <http://auvlab.mit.edu/MURI/annual-report98.htm>.
- [2] W. R. Boynton, L. Murray, J. Hagy, C. Stokes, and W. M. Kemp, 1996. *A comparative analysis of eutrophication patterns in a temperate coastal lagoon*, *Estuaries* 19(2B), 408.
- [3] T. Chung, L. Cremean, W.B. Dunbar, Z. Jin, E. Klavins, D. Moore, A. Tiwari, D. van Gogh, and S. Waydo, *A platform for cooperative and coordinated control of multiple vehicles: The Caltech Multi-Vehicle Wireless Testbed*, Proc. of the 3rd Conference on Cooperative Control and Optimization, Dec. 2002.
- [4] Z. Jin and S. Waydo and E. B. Wildanger and M. Lammers, H. Scholze and P. Foley and D. Held and R. M. Murray, *MVWT-II: The Second Generation Caltech Multi-vehicle Wireless Testbed*, Proc. of the 2004 American Control Conference, pp. 5321-5326, 2004.
- [5] J. A. Farrell, W. Li, S. Pang, and R. Arrieta, 2003, *Chemical Plume Tracing Experimental Results with a REMUS AUV*, Proceedings of Oceans.
- [6] E. Fiorelli, P. Bhatta, N. Leonard, I. Shulman, 2003, *Adaptive sampling using feedback control of an autonomous underwater glider fleet*, Proceedings of Unmanned Untethered Submersible Technology.
- [7] E. Fiorelli, N.E. Leonard, P. Bhatta, D. Paley, R. Bachmayer, D.M. Fratantoni 2004, *Multi-AUV Control and Adaptive Sampling in Monterey Bay*, Proc. IEEE Autonomous Underwater Vehicles 2004: Workshop on Multiple AUV Operations (AUV04), Sebasco, ME, June 2004, <http://www.princeton.edu/naomi/AUV04.html>.
- [8] B. Fletcher, 2001, *Chemical plume mapping with the remus AUV*, Proceedings of Unmanned Untethered Submersible Technology.
- [9] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. *Int. J. Comput. Vis.*, 1:321–331, 1987.
- [10] M. Kemp, A.L. Bertozzi, D. Marthaler, *Multi-UUV Perimeter Surveillance*, edited by C., Brancart, E. An, M. Benjamin (2004) (IEEE OES Workshop on Multiple AUV Operations) Sebasco Estates, Maine.
- [11] S. MacIntyre, K. M. Flynn, R. Jellison and J. R. Romero, 1999, *Boundary mixing and nutrient flux in Mono Lake, CA*. *Limnol. Oceanogr.* 44, 512.
- [12] S. T. Makrinos, 2004, *Port security in the war on terrorism*, *Sea Technology* 45(3), 33.
- [13] D. Marthaler and A. L. Bertozzi, 2003, *Tracking environmental level sets with autonomous vehicles*, S. Butenko, R. Murphey and P.M. Pardalos (editors), "Recent Developments in Cooperative Control and Optimization", Kluwer Academic Publishers.