2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009

Experimental Validation of Cooperative Environmental Boundary Tracking with On-board Sensors

Abhijeet Joshi[1], Trevor Ashley [2],Yuan R. Huang[1], and Andrea L. Bertozzi[3]

Abstract— **This paper describes a cooperative testbed implementation of a new algorithm (Jin and Bertozzi, CDC 2007) for environmental boundary tracking and estimation using only localized noisy sensors. The tracking algorithm is based on Page's cumulative sum algorithm (CUSUM) a method for change-point detection. A geometric, biologically inspired, motion control algorithm allows individual vehicles to track and follow the environmental boundary without external positioning information. Relative positioning between vehicles allows several to maintain a convoy while tracking the boundary. The algorithm performs well in the presence of moderate sensor noise.**

I. INTRODUCTION

Tracking environmental boundaries is an important and useful task for cooperative autonomous vehicles. There are numerous possible application areas ranging from monitoring hazardous conditions in the ocean (e.g. oil spills and 'dead-zones' caused by algae blooms), wild fires, temperature and salinity in the ocean, and hazardous weather conditions such as hurricanes and tropical storms. Moreover, remote sensing can be ineffective in some applications, making this problem ideally suited for robotic vehicles.

This is an active area of research. Most of the approaches to track boundaries fall into two main groups: one which is the gradient-free approach and the other one which measures the gradient of a quantity in the environment. There are a number of papers develop algorithms based on gradients of the environmental field. Bertozzi et al. design a centralized collective motion algorithm based on the *snake* algorithm in image processing to detect and track algae blooms, where each agent needs to measure the concentration gradient [1]. The ACE algorithm developed in [2] uses gradient information in the field to map contours. Zhang and Leonard [3] use four robots to compose a formation so that the gradient at the formation center can be measured in a density field. Susca et al. in [4] require the sensor nodes to collect location information about the tangent and curvature of the boundary, rather than a general gradient in concentration.

[1]Dept. of Electrical Engineering, University of California Los Angeles, Los Angeles, CA 90095 {abhijeet@ucla.edu, yuanh@seas.ucla.edu}
[3]Department of Engineering, Harvey Mudd College, Claremont, CA. {tashley@hmc.edu}
3Department of Mathematics, University of California Los Angeles, Los Angeles, CA 90095{bertozzi@math.ucla.edu}

The gradient-free approach to tracking boundaries uses less information than the gradient dependent formulae since the direction of movement at the next time-step is not completely obvious and one of the main concerns of the gradient-free approach is sensor noise clouding important information regarding the surrounding. Without gradient knowledge from the sensor data, Barat and Rendas [5] use single autonomous underwater vehicle (AUV) with a profiler sonar to detect the boundaries between distinct benthic regions. Casbeer et al [6] consider the problem of forest fire surveillance in which on board cameras obtain information about an area of the fire and local segmentation of the boundary is done on board using imaging algorithms. Andersson too takes the gradient-free approach in [7] where a feedback control mechanism is taken to rapidly capture images in an atomic force microscope (AFM). Our testbed implementation is based on the work of Kemp et al. [8] that consider a simple algorithm for multiple AUV surveillance requiring only local scalar concentration measurements to locate the contour. The path planning part of the algorithm was tested without sensor noise (and with only simulated sensors) on Caltech's multi-vehicle testbed [9]. This paper implements a refined algorithm described in [10] that includes both more efficient path planning and a CUSUM filter for change point detection to determine boundary crossing from noisy sensor data. Real on-board sensors are used and the filtering is performed locally on each robot.

While many of the perimeter finding algorithms can be performed with a single sensor, cooperativity is an important aspect of this problem. Clark and Fierro [11],[12] try to detect and surround a dynamic perimeter using a group of nonholonomic robots equipped with collision avoidance controllers. In [13] the authors develop decentralized controllers for a team of disk-shaped robots to converge to and circulate along the boundary of a desired two-dimensional geometric pattern specified by a smooth function with collision avoidance. An algorithm to track dynamic boundaries with multiple networked sensor agents is discussed in [14]. Zhang and Leonard in [15] show methods that control cooperative Newtonian particles to generate patterns on smooth curves. Susca et al [4] propose a distributed coordination algorithm in which each vehicle tracks the boundary individually and communicates with its nearby neighbors. Vertices are generated uniformly along the boundary based on certain metric

978-1-4244-4524-0/09/$25.00 ©2009 AACC

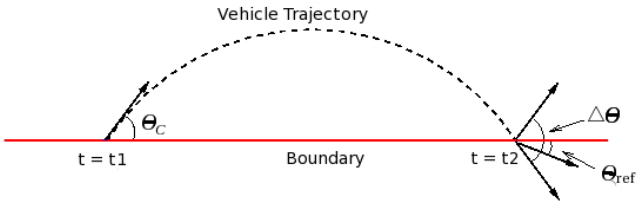

Fig. 1. Geometry between two consecutive crossing points in equation (2).

and a polygon is generated to approximate the boundary.

Although computer simulations validate a consistent control model, the real-life testbed remains an invaluable method for taking into account real-life uncertainties and errors. A economical micro car testbed was developed in [16], using 1/64 size vehicles in an integrated system with overhead camera positioning and off-board motion planning. This paper discusses the adaptation of the testbed for testing boundary tracking algorithms. The platform has previously demonstrated the viability of a multi-function cooperative testbed arena in a cost-effective design [16]. These second generation vehicles come replete with on-board radio communications setup and possess on-board processing. An overhead camera and positioning software allows for overhead tracking; this is relayed to the cars via a radio base station at 30 Hz. The path planning software could be implemented both on and off-board depending on application. In this paper we demonstrate an application where all the motion planning is done on-board and the off-board computer and communications setup is only used for communication of overhead camera information to the vehicles. The new vehicles have comparable functionality to those in [17,18], while maintaining the original low material cost at the order of $150 per car, on a sub-palm-sized chassis.

II. TRACKING ALGORITHM

Our experiment implements an effective way to track the boundary between two disparate regions using only local information in the presence of noise. We also demonstrate an accurate way to measure the change of state from one region to another in the presence of low *SNR* along with an efficient path-planning algorithm as proposed in [10]. The basic algorithm is summarized below.

A. Path Planning Algorithm

Consider a *bang-bang* type steering controller ([8], [9]),

$$\mathbf{u}(k) = \begin{cases} -\theta_{ref} & \text{when } z(k) > B \\ \theta_{ref} & \text{when } z(k) < B \end{cases} \quad (1)$$

where u is the steering control, θ_{ref} a convenient turning angle, $z(k)$ is an input to the steering control at some time-step k and B is the threshold for $z(k)$ indicating the boundary. The vehicle responds by turning with a fixed angle without regard to how the boundary changes.

In addition to the basic bang-bang controller we also implement an improved path planning algorithm proposed in [10]. The improved steering controller keeps track of the time between boundary crossings and attempts to minimize the time between them, thus effectively flattening the vehicle trajectory on a straight path. As noted in [10] this control algorithm requires mostly accurate measurement of boundary crossings, but can easily be adapted to accommodate occasional failures in identifying the boundary. The control law is explicated below.

Let $\Delta t = t_2 - t_1$, be the time passed between two boundary crossings. If θ_c is the crossing angle corresponding to t_1 then $\Delta\theta$ is the resulting difference when compared to the crossing angle at t_2. Also, let ω be the angular velocity of the vehicle as it moves along the arc (figure(1)). The steering control at t_2, $u(t_2)$, is then given by,

$$\mathbf{u}(t_2) = \begin{cases} (\Delta t \cdot \omega - 2\theta_{\text{ref}})/2 & \text{when } z(k) > B \\ -(\Delta t \cdot \omega - 2\theta_{\text{ref}})/2 & \text{when } z(k) < B \end{cases} \quad (2)$$

Setting $\Delta t = 0$ gives us the *bang-bang* type controller.

B. Decision Algorithm

The key to success of the boundary tracking method is an way to determine in real time, from the limited noisy sensor data available, that the vehicle has indeed crossed the boundary. The cumulative sum (CUSUM) filter (implemented for this task in simulation [10]) has also been successfully used on the testbed vehicles to accurately detect changes in data from infrared range sensor for obstacle avoidance [16]. Let $z(k)$ be the input to the filter and the parameters c_u and c_l control the rate of accumulation; $z(k)$ is assumed to be noisy. Here we consider two independent CUSUM filters to detect crossings from one side to the other and vice-versa as established in equations (3) and (4) below. For instance a high value for $U(k)$ indicates a the teal tape and a low value for $L(k)$ indicates that the sensor is on the black tape (top plots in figures(4) and (5)). The rising edge of the high-side filter and the falling edge of the low-side filter are then good indications of boundary crossings [10]. The iterative formulae for the CUSUM filters are as follows:

$$U(k) = \begin{cases} 0 & k = 0 \\ \max(0, z(k) - B - c_u + U(k-1)) & k > 0, \end{cases} \quad (3)$$

$$L(k) = \begin{cases} 0 & k = 0 \\ \min(0, z(k) - B + c_l + L(k-1)) & k > 0. \end{cases} \quad (4)$$

III. HARDWARE MODIFICATION

A. Sensor

We sense the boundary using a downward looking forward infrared range sensor. The Fairchild Semiconductor QRB1134 IR sensor was chosen for its low current consumption and ease of being implemented on the vehicle. It

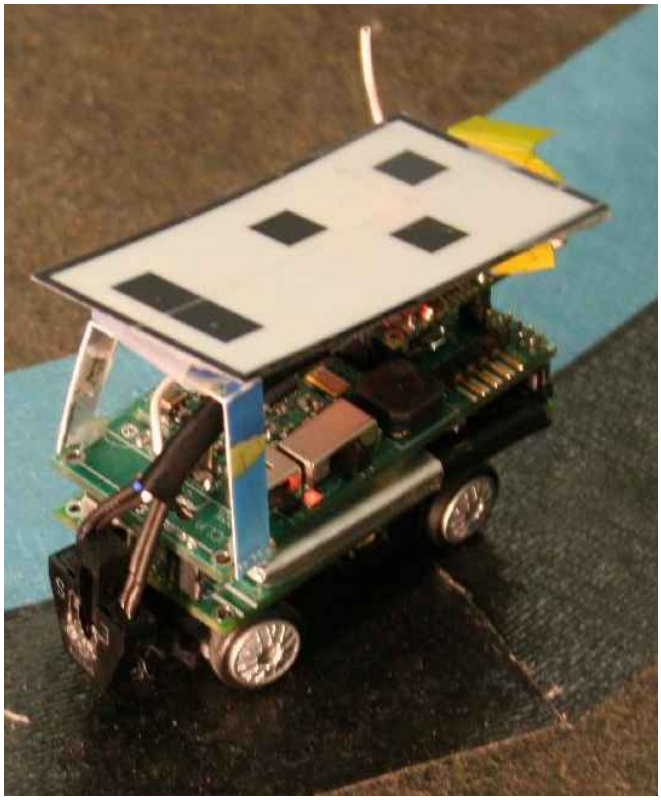

Fig. 2. Autonomous vehicle with downward looking IR device, on colored tape indicating boundary.

is an NPN-type silicon phototransistor mounted next to an IR emitting diode on a converging optical axis. A photocurrent is generated when the IR beam reflects off a suitable surface. The area of optimum response is a circle of diameter 0.200in / 0.5cm [19]. The optimal installation height was found to be 0.150in / 0.38cm. The phototransistor's output was attached to an ADC channel on the Atmega8 processor on-board each vehicle. The figure (2) shows the final installation of the sensor on the vehicle.

B. Testbed

The modification to the testbed in [16] included constructing a physical boundary for the vehicles to follow. The boundary was created of a pair of contrasting tapes - teal blue and black - laid such that a cross-sectional swath of the tape would result in the following transition: testbed-black-teal-testbed. The testbed surface is asphalt grey. The transitions of importance are black-to-teal and the teal-to-black. According to our scheme black-to-teal transition implies a movement to the right. The colors were chosen to provide the widest possible difference in sensor output to detect the transition.

IV. SINGLE VEHICLE IMPLEMENTATION

This section describes several single vehicle tracking tests using either the original bang-bang steering control law

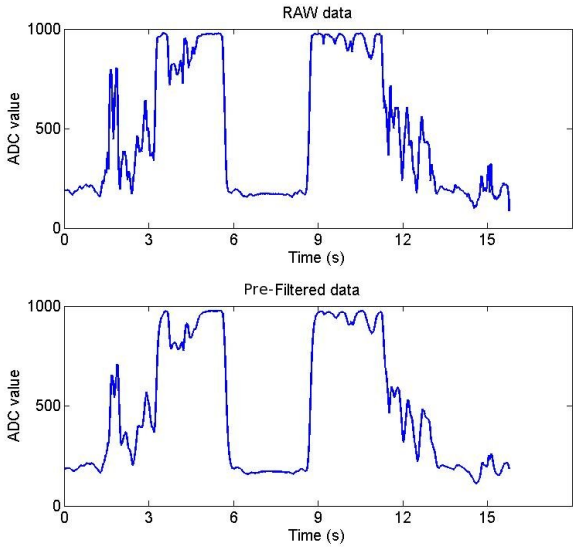

Fig. 3. Comparison of pre-filtered (bottom) and raw sensor data (top). The sensor was over the teal tape between the time instances of 3 and 6 seconds and between 9 and 12 seconds.

(1) or the improved steering control law (2) and various combinations of CUSUM filter and pre-filter.

The pre-filter is a modified Kalman filter with a fixed gain:

$$Inn = z(k) - \hat{z(k)} \tag{5}$$

$$\hat{z(k+1)} = \hat{z(k)} + Gain \cdot Inn \tag{6}$$

The $Gain$ is obtained from previous sensor data by Kalman filtering the signal. $z(k)$ is the same as defined above and $\hat{z(k)}$ is the pre-filtered signal. The computational processing power on the vehicle is sufficient to implement this modified Kalman filter but not good enough to implement a full Kalman filter on board.

To illustrate the need for filtering, figure (3) shows data from the on-board ADC as the vehicle makes two crossing of the boundary. The sensor was over the teal tape between times 3 and 6 seconds and then again between times 9 and 12 seconds. The top half of figure (3) shows the raw data while the bottom half shows pre-filtered data using equations (5) and (6). Figure (4) shows the CUSUM filter applied to the raw data, and resulting decision value for the steering control. A pre-filter was found to be an effective way to weakly damp the raw signal before it was supplied to the CUSUM filter for further analysis. Results using both the pre-filter and CUSUM filter are shown in figure 5. Sensor noise, especially in the black region (see figure (3) circa 2 sec), still has the potential to generate false positives in the decision algorithm causing the vehicle to veer off course. This can be corrected with a simple modification of the bang-bang or modified path planning algorithm whereby a

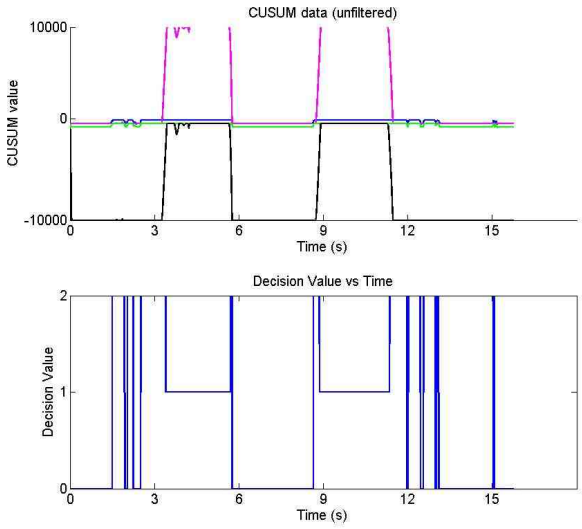

Fig. 4. (top) CUSUM filter applied to the raw data from figure 3. (bottom) Steering control decisions as based on data from top.

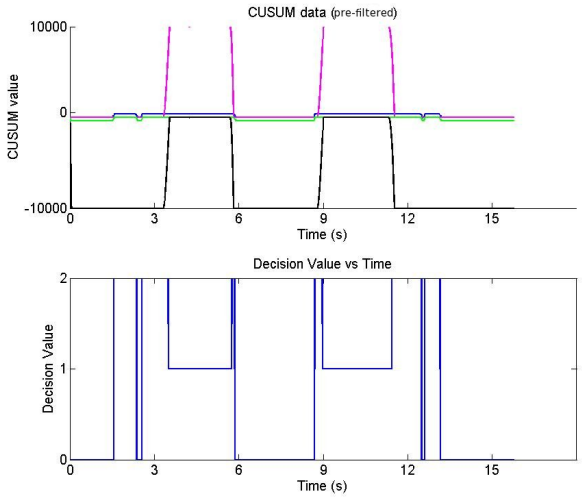

Fig. 5. The same calculations as in figure (4) applied to pre-filtered data from figure (3-bottom).

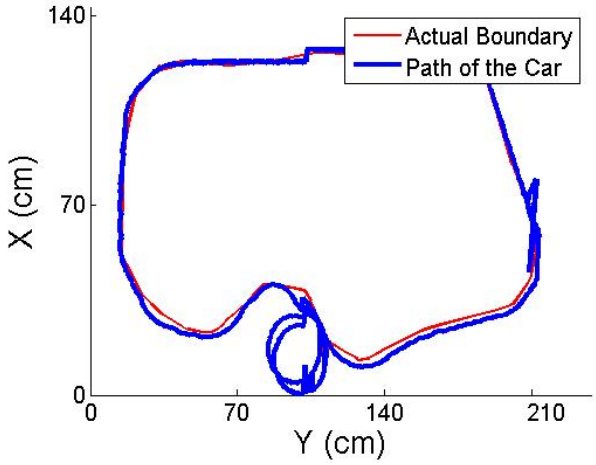

Fig. 6. Car path, without the pre-filter, using modified controller (2), at speed 0.3 m/s.

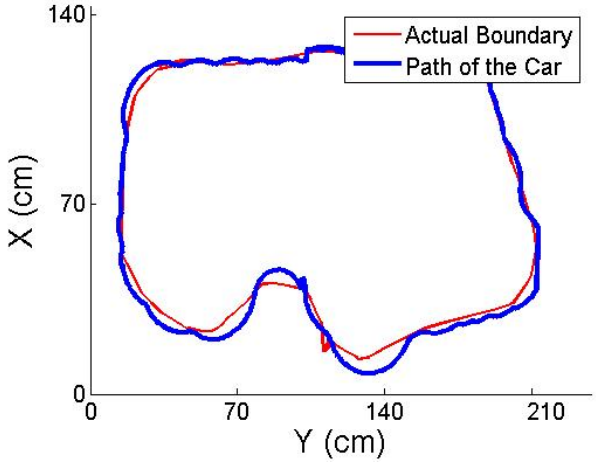

Fig. 7. Car Path, with the pre-filter, using modified controller (2), at speed 0.3 m/s.

vehicle retries the path using a slightly different circle after missing boundary detection. In figure (4) the raw data from figure (3)(top) has been analyzed by the CUSUM filter, with significant improvement over the raw data, yet still leading to occasional decision changes where they ought not to be, eg: t : 2 sec. and t : 12 to 13 sec. After the signal was damped using the pre-filter the result is very clean (see figure (3) bottom and figure(5)).

Figures (6-9) show different runs at various speeds and using the two control algorithms (1) and (2). All runs use the CUSUM filter and some use the pre-filter while others used the CUSUM directly on the raw data. Figure (6) shows a single vehicle path using the bang-bang controller (1) and the CUSUM filter (but no pre-filter). Note that the basic algorithm does a reasonable job of following the boundary yet veers off occasionally due to false positives. This could be easily resolved by automating the return of the vehicle to the last known position of the boundary or to the average position of the rest of the team. Figures (7-9) all show successful following of the boundary. The faster speeds in figures (8-9) show larger departures from the boundary due to the change in path from the speed. The CUSUM filter is very important for successful implementation of this algorithm.

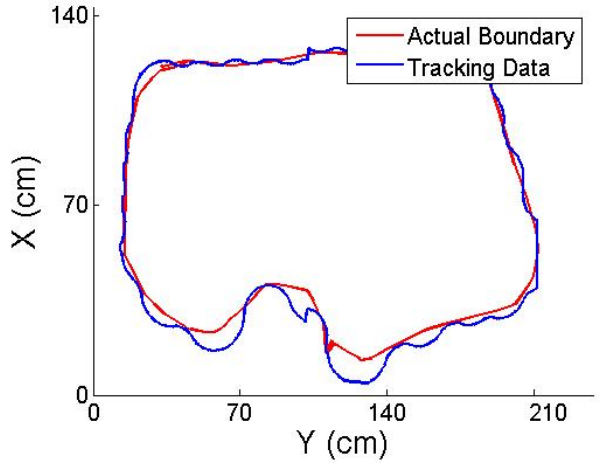

Fig. 8. Car Path, without the pre-filter, using Bang-Bang Control (1), at speed 0.48 m/s.

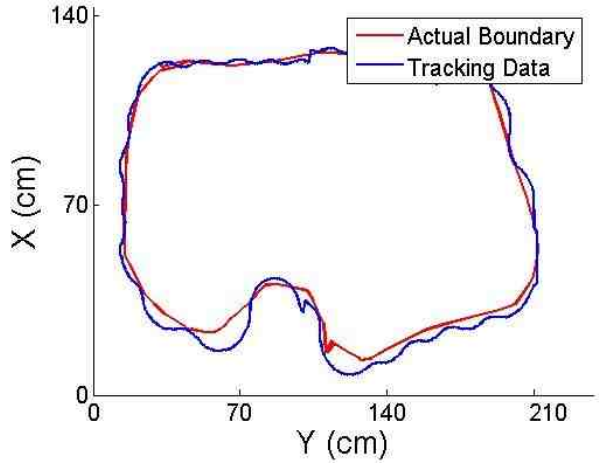

Fig. 9. Car Path, without the pre-filter, using modified controller (2), at speed 0.48 m/s.

V. COOPERATIVE BOUNDARY TRACKING

A. Description

Cooperative boundary tracking utilizes multiple vehicles to achieve efficiency, high-reliability in performing the task at hand. With this in mind we extend the above local control to multiple vehicles moving in a convoy along the boundary.

The co-operative boundary tracking scheme implemented requires the use of three cars tracking the boundary as individuals and also collecting and analyzing position data from other cars. The control law based on the position data is the global control law binding the cars into a collective.

B. Global Control Structure

The control structure maintains the convoy by prioritizing tasks. The boundary following function is given a higher

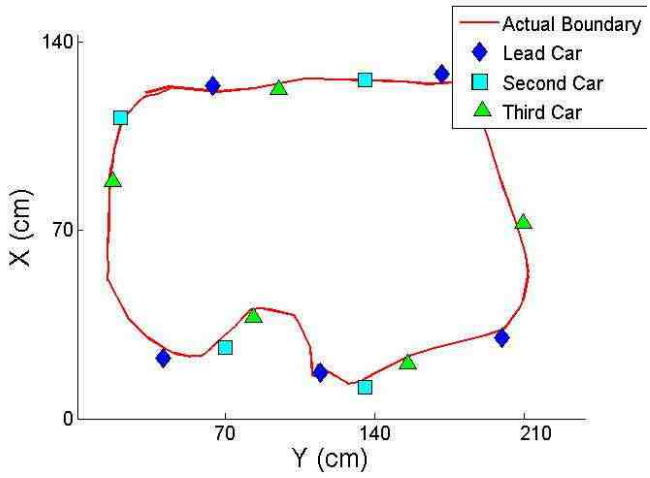

Fig. 10. Convoy of three cars, showing snapshots in time superimposed, time difference 24s.

priority and generally occurs with a frequency of 2Hz. The convoy maintaining part of the algorithm, on the other hand, is relatively naive and takes place at a lower rate. The three vehicles were programmed with cooperative speed control as follows:

$$speed = \begin{cases} speed_l & \text{when} D \leq D_{critical} \\ & \text{or} D \geq D_{critical} \\ & \text{Curent Car is Lead Car} \\ speed_h & \text{when} D \leq D_{critical} \\ & \text{or} D \geq D_{critical} \\ & \text{Current Car is not Lead Car} \end{cases} \quad (7)$$

where $speed$ is the control speed and $speed_l$ and $speed_h$ are lower and higher speed values respectively. D is the radial distance between cars 1 and 2, and 2 and 3. The measure $D_{critical}$ was empirically ascertained.

Figures (10-11) shows sample output from the testbed with three cars maintaining a convoy while tracking the boundary. Figure (11) shows the individual paths of each car while Figure (11) shows their relative positions at different snapshots in time, superimposed on one graph.

VI. CONCLUSION AND FUTURE WORK

This paper illustrates the testbed implementation of a cooperative boundary tracking algorithm using simple local sensors in the presence of noise. It was found that implementing the pre-filter in series with the CUSUM filter along with the time-dependant control algorithm was the most reliable method to detect the boundary. False positives were occasionally detected far away from the boundary causing the vehicles to turn abruptly. This could however be remedied by automating the return of the errant vehicle closer to the boundary. The simple global control law (7) was intended to ensure that the vehicle could maintain some cohesion as a convoy of boundary tracking vehicles. The modulation in

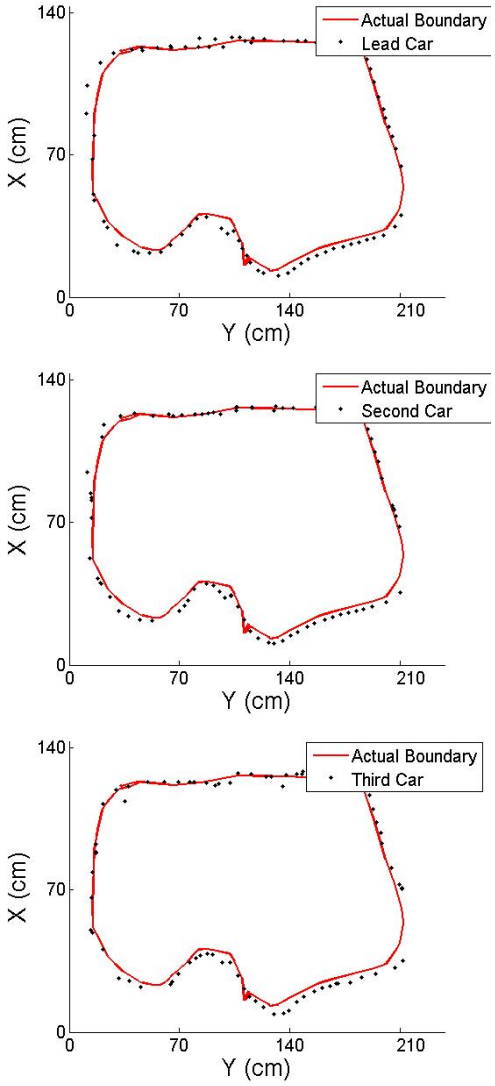

Fig. 11. Paths of Individual Cars in the convoy shown in figure 10.

speed was necessary since there is significant variance in speed on these primitive vehicles. Future work could include a more sophisticated cooperative control algorithm to use separate measurements from vehicles to better avoid false positive boundary detection due to noise. Also a performance analysis of this method could be studied using a combination of geometric estimates and performance analyses of the various filters.

ACKNOWLEDGMENTS

This research is supported by ARO MURI grant 50363-MA-MUR, ONR grant N000140610059, and ARO grant W911NF-05-1-0112. In addition, TA was supported by NSF research training group grant DMS-0601395.

REFERENCES

[1] Daniel Marthaler and Andrea L. Bertozzi L, *Collective motion algorithms for determining environmental boundaries*,SIAM Conference Applications of Dynamical Systems, 2003.

[2] Sumana Srinivasan, Krithi Ramamritham, Purushottam Kulkarni, *ACE in the Hole: Adaptive Contour Estimation Using Collaborating Mobile Sensors*, pp. 147-158, 2008 International Conference on Information Processing in Sensor Networks (ipsn 2008), 2008.

[3] F. Zhang, E. Fiorelli, and N. E. Leonard, *Exploring scalar fields using multiple sensor platforms: Tracking level curves*, IEEE Conference on Decision and i Control, 2007.

[4] Susca, S.; Bullo, F.; Martinez, S., *Monitoring Environmental Boundaries With a Robotic Sensor Network*, Control Systems Technology, IEEE Transactions on , vol.16, no.2, pp.288-296, March 2008.

[5] C. Barat and M. J. Rendas, *Benthic boundary tracking using a profiler sonar*, in Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 1, Oct 2003, pp. 830-835.

[6] David W. Casbeer, Derek B. Kingston, Randal W. Beard, Timothy W. McLain, Sai-Ming Li, and Raman Mehra, *Cooperative Forest Fire Surveillance Using a Team of Small Unmanned Air Vehicles*, International Journal of Systems Science, vol. 37, no. 6, May, 2006, p. 351-360.

[7] Sean B. Andersson, *Curve Tracking for Rapid Imaging in AFM*, IEEE Transactions on Nanobioscience, vol. 6,no. 4, December 2007.

[8] M. Kemp, A. L. Bertozzi, and D. Marthaler, *Multi-UUV perimeter surveillance*, Proceedings of the 2004 IEEE/OES Workshop on Autonomous Underwater Vehicles, pp. 102-107, 2004.

[9] Chung H. Hsieh , Zhipu Jin , Daniel Marthaler , Bao Q. Nguyen, David J. Tung, Andrea L. Bertozzi, Richard M. Murray, *Experimental Validation of an Algorithm for Cooperative Boundary Tracking*, Proceedings of the 2005 American Control Conference, Portland, OR, June 8-10.

[10] Zhipu Jin; Bertozzi, A.L., *Environmental boundary tracking and estimation using multiple autonomous vehicles*, Decision and Control, 2007 46th IEEE Conference on , vol., no., pp.4918-4923, 12-14 Dec. 2007.

[11] J. Clark and R. Fierro, *Mobile robotic sensors for perimeter detection and tracking*, ISA Transactions, **46**[1], February 2007, pp. 3-13.

[12] Clark, J.; Fierro, R., *Cooperative hybrid control of robotic sensors for perimeter detection and tracking*, American Control Conference, 2005, vol. 5, pp. 3500-3505, 8-10 June 2005

[13] Hsieh, M.A.; Loizou, S.; Kumar, V., *Stabilization of Multiple Robots on Stable Orbits via Local Sensing*, Robotics and Automation, 2007 IEEE International Conference on , vol., no., pp.2312-2317, 10-14 April 2007.

[14] Seyong Jang, Bongsob Song and Suk-Kyo Hong, *Dynamic Boundary Tracking in Active Sensor Networks*, International Conference on Control, Automation and Systems October 2007, pg 2368-2373.

[15] F. Zhang and N. Leonard, *Coordinated patterns on smooth curves*, Proceedings of the IEEE International Conference on Networking, Sensing, and Control, pp. 434-440, 2006.

[16] Kevin K. Leung, Chung H. Hsieh, Yuan R. Huang, Abhijeet Joshi, Vlad Voroninski, and Andrea L. Bertozzi, *A second generation micro-vehicle testbed for cooperative control and sensing strategies*, Proceedings of the 2007 American Control Conference, New York, NY, July 11-13.

[17] 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.

[18] 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.

[19] Fairchild Semiconductor QRB1134 Datasheet from Mouser Electronics (http://www.mouser.com)