Gradient-Based Cluster Space Navigation for Autonomous Surface Vessels

Thomas Adamek, *Member, IEEE*, Christopher A. Kitts, *Senior Member, IEEE*, and Ignacio Mas, *Member, IEEE*

***Abstract*—This paper presents an experimentally demonstrated gradient-based multirobot technique for adaptively navigating within a parameter field. To implement this technique, simultaneous measurements of the parameter are made at different locations within the field by a spatially controlled cluster of mobile robots. These measurements are shared in order to compute a local gradient of the field. Depending on the task to be achieved, the multirobot cluster is directed with respect to this direction. Moving in or opposite to the gradient direction allows efficient navigation to local maxima/minima in the field, a capability of interest for applications such as detecting pollution sources or the location of resource-starved areas. Moving perpendicular to the gradient direction allows parameter contours to be navigated, a behavior useful for applications such as defining the extent of a field or establishing a safety perimeter at a defined field level. This paper describes the multirobot control technique which combines a full degree-of-freedom "cluster space" multirobot controller with a gradient-based adaptive navigation capability. Verification of the technique through field experiments using a fleet of three robotic kayaks is also presented. Finally, a discussion of results, a review of challenges, and a review of ongoing and future work are presented.**

***Index Terms*—Adaptive navigation, autonomous surface vessel, cluster space control, gradient-based navigation.**

I. Introduction

MULTIROBOT systems have the potential to dramatically impact robotic applications through improved performance and the enabling of completely new capabilities. Alone, robots offer strength, speed, precision, repeatability, and the ability to withstand extreme environments. Combined in a multirobot system, additional advantages are possible, such as redundancy, increased throughput, expanded coverage/availability, and spatially distributed sensing and actuation [1]. Multirobot systems can support applications ranging from remote [2] and *in situ* sensing [3] to the physical manipulation of objects [4], and the domains for such applications include land, sea, air, and space [5].

One characterization of multirobot systems is the degree to which their functions and spatial characteristics are coordinated. Some applications loosely coordinate such characteristics, such as the open-loop synchronization of prepositioned manipulators on an assembly line [6]. Applications requiring capabilities such as synoptic mapping, in which robot-based sensors are spatially distributed throughout a field of interest and collect data in a synchronized manner in order to represent the state of the field [7], may use closed-loop spatial control to ensure high-quality results. Other applications may require even more integrated operation and relative position control, such as proposed sparse array space telescopes [8].

Our work focused on the highly integrated end of the control spectrum, with target applications that include active escorting/guarding [9], [10], object tracking, object manipulation [11], and sparse antenna arrays [12]. Given that these applications require active control of the relative spatial characteristics of the robot formation, we have developed a flexible and powerful formation-level control architecture, known as the cluster space formation control technique [13], which provides a suitable level of abstraction at the application-formation control interface.

The work presented in this paper focuses on the use of multiple, spatially distributed robots to sense a spatially varying parameter of interest, estimate the local spatial gradient of this field, and then navigate with respect to this gradient. This approach offers the ability to navigate with respect to possible features of interest, such as local maximum/minimum locations or perhaps along a specific parameter contour. This is in contrast to traditional parameter mapping approaches in which a single platform systematically navigates through a region, often in a "mow-the-lawn" fashion, in order to map an entire area [14]. If the ultimate goal is to locate/perform gradient-related features/tasks, the multirobot gradient-based mapping capability offers benefits such as faster identification of areas of interest and the ability to dynamically track these features in time-varying fields.

There have been a number of examples of gradient-based navigation through a parameter field by a single robot through the use of bioinspired control strategies. In one system, a robot was built to follow the direction of an odor source [15]. This was done using four anemometric and gas sensors to estimate the direction of airflow carrying the molecules of interest. Given this bearing estimate, the robot moved toward the odor source using a control strategy inspired by how moths track pheromones [16], [17]. Another implemented example was executed by the autonomous benthic explorer (ABE), an underwater robot used to

Manuscript received April 29, 2013; revised July 31, 2013; accepted November 16, 2013. Recommended by Technical Editor H. Gao. This work (elements of this work, to include control system development, creation of the experimental testbed, field experimentation, etc.) was supported in part by the SCU Robotic Systems Laboratory, School of Engineering, Sustainability Program, and Technology Steering Committee; in part by NOAA under Grant NA03OAR4300104 (via subaward UAF 05-0148 from the University of Alaska Fairbanks); and in part by the National Science Foundation under Grant CNS-0619940.

T. Adamek and C. A. Kitts are with the Department Mechanical Engineering, Santa Clara University, Santa Clara, CA 95053 USA (e-mail: tadamek@scu.edu; ckitts@scu.edu).

I. Mas was with the Department Mechanical Engineering, Santa Clara University, Santa Clara, CA 95053 USA. He is now with the Instituto Tecnologico de Buenos Aires, Buenos Aires, Argentina (e-mail: ignacio.a.mas@gmail.com).

Digital Object Identifier 10.1109/TMECH.2013.2297152

find a specific location in a lake using a single beam sonar altimeter [18]. ABE's motion control strategy was modeled after the chemotaxis behavior of E. coli [19], [20], using periods of random spatial motion during which the gradient was computed followed by gradient-referenced motion. In general, while gradient-based navigation was possible, these systems were limited by their small sensing regions/baselines and the resulting slow convergence that results from having to move over time in order to compute the local gradient [21]. More recently, another technique uses a sinusoidal perturbation of the sensed parameter to compute a local gradient, thereby supporting a source seeking behavior [22], [23].

With respect to multirobot approaches, Hayes *et al.* implemented a bioinspired, multistage approach to localizing odor sources in which the robots first identified the existence of a plume, then moved toward the plume's source, and finally located the source position [24]. In this work, a single robot combined an outward spiraling motion in attempts to sense the binary presence of an odor (e.g., the sensed level was above or below a given threshold) with periodic "surges" of motion in the upstream direction, determined by a flow sensor, in order to move toward a source. Multirobot execution of this algorithm consisted of either 1) using all robots to initially find a plume but then following the plume with only the first robot to locate it; or 2) having upward-surging robots command downwind robots and any robots with no plume information to surge in the direction of the commanding robot. Although limited to binary plume information, rudimentary-robot collaboration, and verification via simulation and small-scale lab-based experiments, this study demonstrated performance improvements of 25–40% time reductions in tracking a plume to its source, with the majority of performance improvement occurring with the addition of only one or two robots.

Work by Biyik and Arcak demonstrated another approach in which a multirobot formation was steered to the maxima/minima of a parameter field. In this work [25], the leader alone performed extremum seeking by generating approximate gradients of the local field through the dithering of sensor position. Using this approach to determine the desired bearing, additional vehicles simply followed this leader using passivity-based coordination rules. This approach, which was verified via simulation but not hardware-in-the-loop experimentation, showed that under certain conditions, the following vehicles responded only to the gradient motion, effectively filtering out the dither.

Significant work by Leonard and various collaborators has explored, developed, and simulated several gradient-based adaptive sampling strategies for small autonomous underwater vehicle (AUV) fleets with the objective of detecting interesting features and tracking fronts in the marine environment [26]–[28]. In [29], the same team conducted real AUV experiments in Monterey Bay, CA, USA, to verify the performance of their core virtual bodies and artificial potentials formation-keeping architecture and to assess the feasibility of performing gradient-based maneuvers. Experimental results showed the ability to maintain adequate formation control to support gradient-based maneuvers, and through postmission analysis and simulation, the team concluded that there was good potential to successfully implement gradient-based adaptive navigation.

It is worth noting that gradient-based methods have shortcomings. First, moving up or down the gradient is limited by local minima/maxima; if identifying global extremes is critical, augmented techniques are required to identify other peaks and valleys in the parameter field. Second, discontinuities in the field can yield the technique difficult if not impractical to implement. Third, small gradients, particularly in the presence of noise, may be difficult to sense. A common strategy to address this is to perform a local search (known as "casting" in the biological literature) in order to reacquire the gradient; this behavior has been observed, for example, in moths [30].

The work presented in this paper describes an implemented technique for using a multirobot formation to perform gradient estimation while moving through a scalar parameter field and to use the gradient estimate to adaptively navigate in real time in order to locate and track features of interest. Using three autonomous surface vessels operating in a marine environment, this system uses our previously developed cluster space multirobot formation control architecture in order to spatially distribute the robots such that a reasonable gradient estimate is possible. A new adaptive navigation control system acts as an outer control loop in order to steer the robot fleet in order to ascend/descend gradients or to track field contours with specific concentration levels.

Compared to "mow-the-lawn" mapping approaches, our system enables identification and tracking of gradient-based features, even with dynamic fields, with significantly fewer operational resources (time, power, etc.). Compared to the previously discussed single-robot gradient estimation strategies, our system provides real time, continuous estimation of the gradient, ensuring faster response. Perhaps most interesting, most of the previously reviewed strategies have only been explored via analysis and simulation; our system has been verified through hardware-in-the-loop experimental demonstration in a field setting. Achieving these capabilities is now allowing us to focus on field applications with an established network of science collaborators in order to perform tasks such as identifying bathymetric features of interest to geologists, locating anoxic regions relevant to marine biological health, and isolating locations of pollutants.

Section II of this paper reviews the cluster space multirobot control approach. Section III presents how the cluster space controller is augmented with an adaptive navigation outer control loop that estimates the local gradient and computes motion commands for the multirobot cluster controller. Section IV reviews the design of the multikayak system that was used to demonstrate the technique during field experiments. Section V presents field data demonstrating the functionality of the technique in the context of navigating with respect to bathymetric gradients. Section VI reflects on the results of the experiments, and highlights ongoing and future work designed to extend the presented technique. Finally, Section VII summarizes the work presented in this paper.

II. Cluster Space Multirobot Framework

The cluster space control technique represents a group of robots as a virtual articulating kinematic mechanism that can be resized and reshaped as it moves through space [13]. The group of robots, termed a "cluster," is assigned a cluster reference frame, and the pose of the cluster is represented by the location and orientation of this frame, the geometric shape of the cluster, and the relative orientations of each robot with respect to the cluster frame; these pose variables and their derivatives define the cluster state space for the system. Motions are specified in terms of these cluster parameters, making specification and monitoring simple for human operators and providing a natural level of abstraction for higher level controllers. Furthermore, control compensation is computed with respect to the cluster variables, which provides well-behaved geometric motion even when individual robots may need to move in a highly nonlinear manner.

With respect to other common formation approaches, the cluster technique has several distinct characteristics. Compared to many bioinspired and "swarm" approaches [31], [32], cluster space control is a full-degree-of-freedom controller with a strong control-theoretic mathematical framework. With respect to the virtual rigid body approach [33], [34], cluster space control explicitly provides for the time-varying geometry of the formation, and control computations are directly computed as opposed to being established through an algorithmic reasoning process [35]. Compared to artificial potential field approaches, the proposed method does not rely on the generation of adequate potential functions to reach desired poses; rather, it allows for simple formation reconfiguration based on intuitive parameters [36]. Finally, with respect to leader–follower techniques, cluster control does not require the designation of a specific leader and is more flexible in accommodating relative spatial requirements among robots [37], [38].

The benefits of cluster space control, however, do not come without drawbacks. Depending on how it is implemented, the technique can be computationally expensive compared to many other methods, and the level of flexibility that it provides is unnecessary for many applications. In addition, convenient shape descriptions can suffer from geometric singularities [39]; this can be automatically managed through state space switching; however, this adds to the computational load.

Our previous and ongoing work in cluster space control includes its implementation with both human pilots and automated trajectory controllers, use with both holonomic and nonholonomic vehicles, use with linear and nonlinear controllers, implementation with both resolved rate and dynamic controllers, avoidance of obstacles, and experimental demonstration on land/sea/air systems with up to six robots [10], [40], [41]. Supporting analytic work has included proof of Lyapunov stability, dual-rate computational implementations, varying the level of (de-) centralization, and the formulation of hierarchical clusters of clusters [42], [43].

This following section defines the robot space and cluster space representations of a multirobot system and introduces the kinematic transforms that relate the positions and velocities in these spaces. The subsequent section reviews the inverse Jacobian control architecture, which is a typical way in which cluster space control is implemented. A three-robot planar cluster is used as the example throughout this section given that the experiments presented later in this paper use such a real-world cluster of robots.

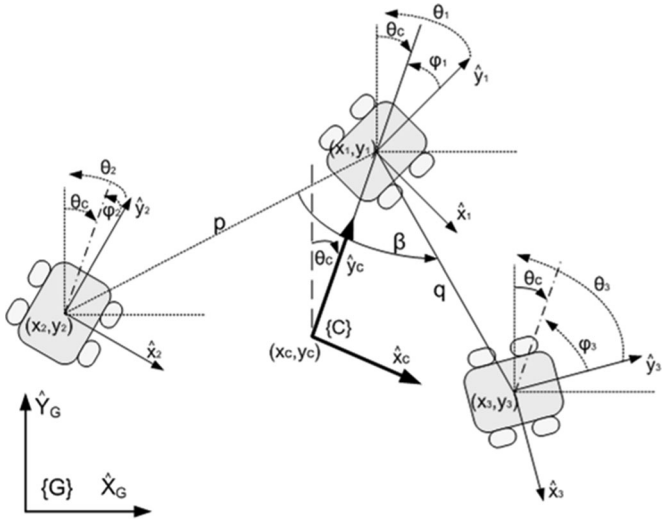

Fig. 1. Three-robot cluster, showing a cluster space representation of pose: cluster location (x_c, y_c, θ_c), cluster shape (p, q, β), and relative robot orientations with respect to the cluster $(\emptyset_1, \emptyset_2, \emptyset_3)$.

A. Kinematic Formulation

The general kinematic formulation for a cluster of n robots, each with m degrees of freedom, is provided in [13]. Here, we provide the specific formulation for a three-robot planar system, as shown in Fig. 1, which we have used to demonstrate the gradient-based navigation technique.

A conventional robot-oriented representation of this system consists of describing the system's pose in terms of the position and orientation of each robot

$$^{G}\vec{R} = (x_1, y_1, \theta_1, x_2, y_2, \theta_2, x_3, y_3, \theta_3)^T \tag{1}$$

where (x_i, y_i, θ_i) is the position and orientation of robot i for $i = 1, 2, 3$ as defined within the global frame, $\{G\}$.

To consider the system as a cluster, a cluster reference frame $\{C\}$ is defined; in this example, it is located at the centroid of the formation and oriented in the direction of robot 1. The shape of the cluster is naturally defined as a triangle, expressed in this case through a side-angle-side description of geometry. Given this, the system's cluster-oriented pose is

$$\vec{C} = (x_c, y_c, \theta_c, \emptyset_1, \emptyset_2, \emptyset_3, p, q, \beta)^T \tag{2}$$

where the values (x_c, y_c) is the position, θ_c is the orientation of the cluster frame with respect to $\{\mathrm{G}\}$, (p, q, β) quantify the side-angle-side description of the cluster's shape, and $(\emptyset_1, \emptyset_2, \emptyset_3)$ denote the relative angle of each robot with respect to the cluster

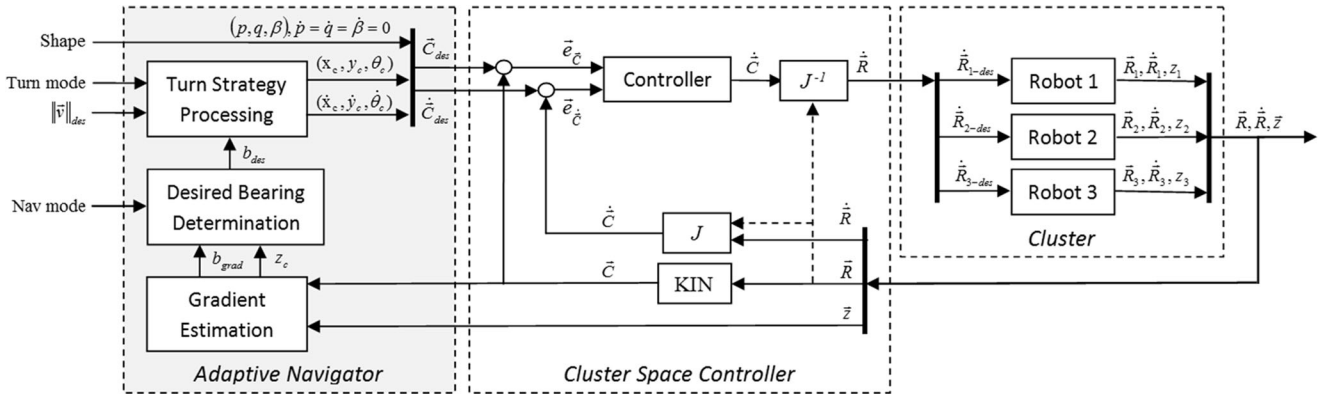

Fig. 2. Implemented gradient-based cluster space control architecture. The robot cluster is shown on the right, with each robot capable of responding to a robot-specific velocity command. The cluster space control layer is shown in the middle. This controller computes an error-drive cluster velocity command, which is converted to robot-specific velocity commands via the inverse Jacobian transform. The research presented in this paper focuses on the inclusion of the adaptive navigation layer, shown in the gray box on the left. This controller estimates the gradient direction, determines the desired bearing for the cluster, and specifies the appropriate cluster state space set-points to achieve the desired navigation task.

frame. We note that, in general, the cluster space technique provides flexibility in how the cluster frame is assigned and how the cluster shape is defined; the wide range of options drives implementation issues such as the level of (de)centralization, computational complexity, and the nature of geometric singularities. For a three-robot planar system, nine position variables represent the system's degrees of freedom; accordingly, both $\vec{R}$ and $\vec{C}$ are nine-element vectors.

We can define a set of position kinematic transforms expressing cluster-oriented pose variables in terms of robot-oriented pose variables and vice versa

$$\vec{C} = \text{KIN}\,({}^{G}\vec{R}) = \begin{pmatrix} g_1\,(r_1, r_2, \ldots, r_{mn}) \\ g_1\,(r_1, r_2, \ldots, r_{mn}) \\ \vdots \\ g_{mn}\,(r_1, r_2, \ldots, r_{mn}) \end{pmatrix} \tag{3}$$

$${}^{G}\vec{R} = \text{INVKIN}(\vec{C}) = \begin{pmatrix} h_1\,(c_1, c_2, \ldots, c_{mn}) \\ h_1\,(c_1, c_2, \ldots, c_{mn}) \\ \vdots \\ h_{mn}\,(c_1, c_2, \ldots, c_{mn}) \end{pmatrix}. \tag{4}$$

Taking the derivative of (3) and (4), system velocities can be related to one another through the use of a linear time-varying Jacobian matrix, J, as shown in

$$\dot{\vec{C}} = \begin{pmatrix} \dot{c}_1 \\ \dot{c}_2 \\ \vdots \\ \dot{c}_{mn} \end{pmatrix} = {}^{G}J(\vec{R})\,{}^{G}\dot{\vec{R}} \tag{5}$$

$${}^{G}\dot{\vec{R}} = \begin{pmatrix} \dot{r}_1 \\ \dot{r}_2 \\ \vdots \\ \dot{r}_{mn} \end{pmatrix} = {}^{G}J^{-1}({}^{G}\vec{R})\,\dot{\vec{C}}\,. \tag{6}$$

For our three-robot system, the forward position transforms, $\vec{C} = \text{KIN}({}^{G}\vec{R})$, are provided in the Appendix. Limited space prevents expression of the remaining sets of equations.

B. Cluster Space Formation Control Architecture

With the formal kinematics defined, the controller is composed in such a manner that the desired motions are specified and control compensations are computed in the cluster space. The resulting feedback control architecture is independent of the style of control function used (e.g., linear, nonlinear, model based, etc.) or of how the desired state values are set (real-time pilot inputs, trajectory generator, inputs from a higher level controller, etc.).

For the gradient-based navigation capability achieved in this research project, a cluster space linear PID resolved rate controller was used in order to generate an instantaneous cluster velocity command, $\dot{\vec{C}}_{\text{cmd}}$. As shown in Fig. 2, this was converted to equivalent robot-specific velocity commands through the use of the inverse Jacobian transform, J^{-1}, as defined in (6). Sensed robot positions were converted to cluster position estimates through the use of the forward kinematic functions, defined in (3) and (5).

Although it was not used for this particular research project, we note that we have developed and implemented dynamic versions of this cluster space control architecture in which the controller specifies cluster space forces and torques; these are transformed by J^T to determine the instantaneous robot-specific control forces and torques to be applied [42].

Nonholonomic drive constraints for the individual robots, such as the ones use for experimentation in this project, limit the freedom of arbitrarily specifying the $\emptyset_i$ states, which are the relative angles of each robot with respect to the cluster frame. A simple way to accommodate these constraints is to use a robot-level heading controller that orients the robot in the direction of its commanded velocity vector as specified by the cluster space controller; this is the approach that was used for the experiments reported in Section V. A more sophisticated approach achieves

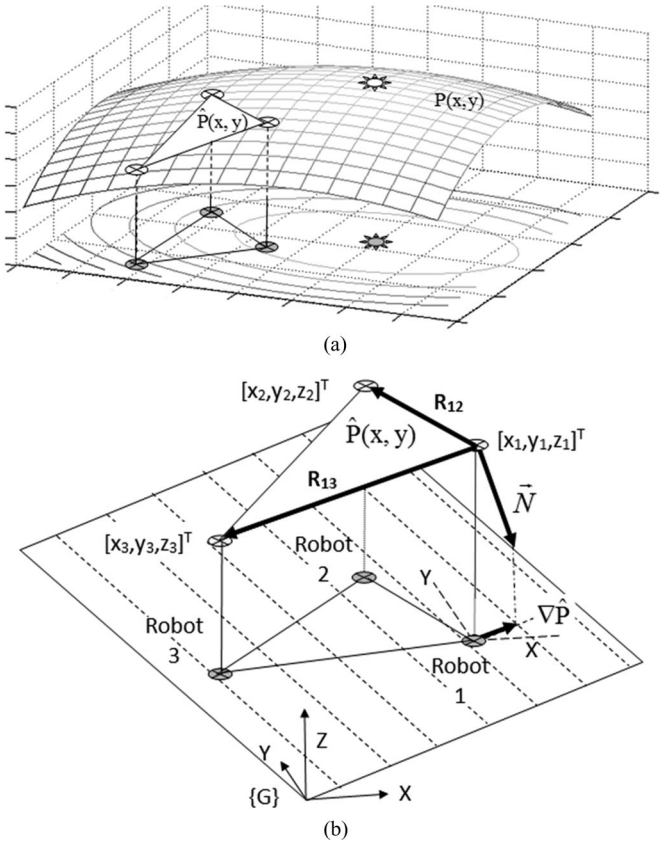

Fig. 3. Three robots within the cluster compute the field's gradient based on their locations and samples of the scalar parameter field. (a) Three robots sample the scalar parameter field $P(x, y)$, thereby creating a local approximation in the form of the plane $\hat{P}(x, y)$. (b) Three robots define vectors within the planar field approximation, allowing the direction of the field gradient to be computed.

the same result but through the explicit handing of the nonholonomic constraints within the cluster space controller [42].

III. Gradient-Based Multirobot Navigation

For this research project, we seek to adaptively navigate with respect to the gradient of a scalar parameter field, $P(x, y)$, in order to perform functions such as efficiently moving to local minima/maxima or moving along specific concentration contours of the field. Examples of parameters of interest include quantities such as temperature and the concentration of pollutants or hazardous materials. In general, the precise nature of $P(x, y)$ is unknown, although some assumptions may be reasonable regarding the magnitude of the gradient and the spatial frequencies within the field.

To adaptively navigate, we first estimate the direction of the local gradient using real-time measurements made by sensors on each of the distributed robots. Given this estimate, we then steer the cluster with respect to the gradient in order to navigate in a manner appropriate to the given task.

A. Computing the Gradient

To understand how the direction of the local gradient is estimated, consider the diagrams in Fig. 3. In Fig. 3(a), a planar region of robot motion is shown. The scalar parameter field is represented by contours in the planar region and also as a surface with a height at any point above the plane equal to the value of the scalar field at that point, $z = P(x, y)$. The maximum point in the scalar field is indicated by a star in both the plane and on the 3-D parameter surface.

At a given point in time, the three robotic vehicles are at specific locations in the planar region, as denoted by the circled locations. Each vehicle samples the parameter field at its own location such that it can be envisioned to be located on the virtual parameter surface. Given the use of three vehicles to create an instantaneous estimate of the field, the samples effectively establish a planar approximation of the parameter surface, $\hat{z} = \hat{P}(x, y)$, in the vicinity of the cluster.

In Fig. 3(b), the robots are shown again, both in the *XY* plane of motion and in the approximated planar parameter surface, $\hat{z} = \hat{P}(x, y)$, at the locations (x_i, y_i, z_i) for $i = 1, 2, 3$, where (x_i, y_i) is the location of robot i and z_i is the measurement of the field at this point. Because the approximated field is planar, the contour lines are now approximated as lines in the local region, as now shown in the *XY* plane of motion.

Given the locations of the robots on the virtual surface, we construct the vectors $\bar{R}_{12}$ and $\bar{R}_{13}$, as shown in Fig. 3(b), running from the projected robot 1 location to the projected locations of robots 2 and 3, respectively. To compute the direction of the field's gradient, shown in the *XY* plane as $\nabla\hat{P}$, the cross product $\vec{N} = -\vec{R}_{12} \times \vec{R}_{13}$ is computed and projected into the *XY* plane. The resulting $\nabla\hat{P}$ vector points in the direction of greatest parameter increase, and it is perpendicular to the local scalar field contour lines.

To summarize this estimation approach mathematically

$$\bar{R}_{12} = \begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{pmatrix} \tag{7}$$

$$\bar{R}_{13} = \begin{pmatrix} x_3 - x_1 \\ y_3 - y_1 \\ z_3 - z_1 \end{pmatrix} \tag{8}$$

$$\overrightarrow{N} = -\vec{R}_{12} \times \vec{R}_{13} \tag{9}$$

$$\nabla\hat{P} = [N_x, N_y]^T \tag{10}$$

$$b_{\text{grad}} = pi/2 - \text{ATAN2}(N_y, N_x) \tag{11}$$

where N_x and N_y are the x- and y-components of N, the surface normal vector and b_{grad} is the bearing of the field gradient (e.g., the direction of maximum parameter increase), expressed as a heading angle in $\{G\}$.

For contour following, the location of the cluster in the parameter field must be approximated. Given that the origin of $\{\text{C}\}$ represents the cluster's location and given the planar assumption of the field in the local area, the parameter field value at the cluster's location is $\hat{z}_c = \hat{P}(x_c, y_c)$.

B. Gradient-Based Navigation

With an estimate of the bearing of the field gradient now available, this knowledge can be incorporated into the cluster's

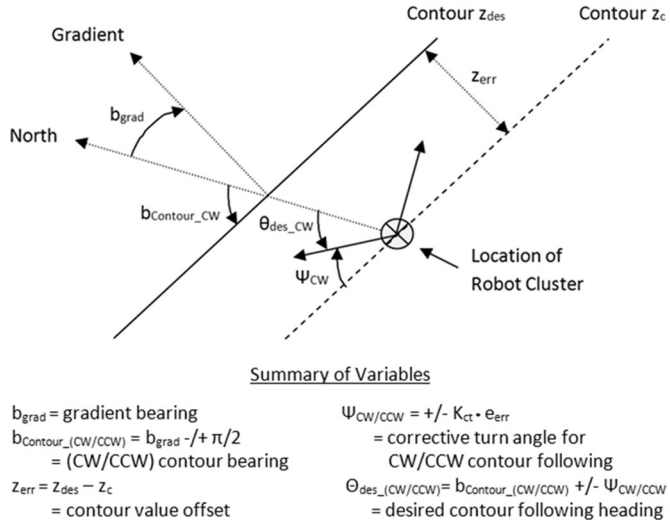

Fig. 4. Cluster contour following strategy (for clarity, only CW-related quantities are shown).

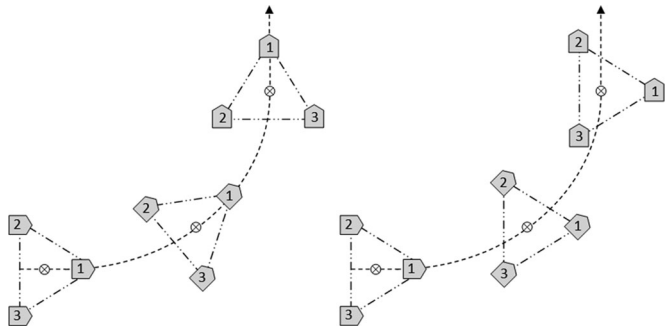

Fig. 5. Examples of Nonholonomic cluster rotation (left) and holonomic cluster rotation (right).

real-time navigation strategy in order to adaptively drive the cluster as a function of the sensed environment. Although a variety of navigation strategies can be considered, here we focus on two specific strategies which we believe hold specific promise for applications we are pursuing: 1) navigating to local minima/maxima in the field; and 2) navigating along specific contour levels within the field.

To navigate to the local minimum or maximum, $b_{\rm grad}$ provides the heading of the greatest rate of parameter increase. The opposite direction is the heading of the greatest rate of parameter decrease. Accordingly, for gradient climbing/descent mode, the desired bearing of travel is

$$b_{\rm des} = b_{\rm grad} + (d * \pi) \tag{12}$$

where $d = 0$ for gradient ascent and $d = 1$ for gradient descent. We note that this navigation strategy simply directs the cluster along the local direction of maximum/minimum parameter change; there is no attempt to remain on any specific gradient line.

Navigating along a field contour requires more sophistication given that this strategy implies not just the desire to move in the direction of the contours but also the desire to move to and follow a specific contour line with a given parameter value. First, the direction of the contour lines must be determined. Given that contour lines are perpendicular to the gradient, the bearing of what we term the clockwise (CW) contour direction (which implies a CW rotation around the parameter field if the field was a simple single peak) has a value of $[b_{\rm grad} - (\pi/2)]$. Similarly, the bearing of the contour for counterclockwise (CCW) travel is $[b_{\rm grad} + (\pi/2)]$.

To follow a specific contour of value $z_{\rm des}$, a simple cross-track controller is used, as depicted in Fig. 4. This strategy specifies a heading set point equal to the desired contour bearing plus a corrective bearing term proportional to the cross track error, $(z_{\rm des} - z_c)$, which biases travel toward the desired contour line. The corrective term is limited to 90° such that the cluster heads directly toward the contour line in a perpendicular fashion for large deviations. Mathematically,

$$\theta_{\rm des} = b_{\rm grad} + d * \{\mathrm{sgn}(z_{\rm des} - z_c) * \min[K_{\rm ct} * ||z_{\rm des} - z_c||, \pi/2] - \pi/2\} \tag{13}$$

where d is 1 for CW navigation and –1 for CCW navigation, and $K_{\rm ct}$ is the cross-track correction gain. This path-following approach is similar to that used for an operational single boat system that follows paths in order to perform bathymetric mapping applications [14].

C. Cluster Turning Options

In our work to date, we have specified a constant shape, size, and forward speed for the cluster and have simply steered the cluster in order to travel along the desired bearing. Given a desired bearing based on the navigation strategy as formulated in Section III-B, there are two distinct options for how the cluster can be commanded to move, as depicted in Fig. 5.

The first strategy consists of constraining the cluster's translational velocity to act only in a single direction with respect to the cluster frame and to accommodate lateral travel by turning the aggregate cluster. For example, the cluster's translational velocity can be constrained to act only in the cluster frame's y-direction (e.g., $\dot{x}_c = 0$; $\dot{y}_c = v_{\rm des}$) and the cluster is turned by aligning the cluster heading with the desired bearing $((\theta_c)_{\rm des} = b_{\rm des})$ as determined in Section III-B. This establishes a nonholonomic-like aggregate drive characteristic for the cluster in which one "side" of the cluster is generally aligned with the direction of travel and there is no "side-slip" component of cluster velocity.

Alternatively, the second strategy allows the cluster to adopt a translational velocity with components in both the x- and y-cluster frame directions. This is achieved by varying the ratio $\dot{y}_c/\dot{x}_c$, given a constant translational cluster speed, $v_{\rm des}$, in order to align the cluster's velocity vector with the desired bearing. This is done independently of the cluster's heading, which may be kept constant (as shown in Fig. 5) or oriented based on some independent criteria. The effect of this motion strategy is to provide a holonomic-like side-slip capability for the overall cluster, which in practice has been found to be more agile when turning.

D. Integrated Controller

The previous sections described how the field gradient is estimated (Section III-A), how the navigation strategy (gradient climbing/descending or CW/CCW contour-following) exploits this knowledge to specify a desired bearing for cluster travel (Section III-B), and how the cluster turn strategy (rotate or side-slip) incorporates the desired bearing into cluster state space set-points for the cluster space controller. This information is used to specify set-points for the cluster variables x_c, y_c, θ_c, and their derivatives. Set-points for the cluster shape variables are independently specified given the current policy of constant size and shape. Finally, the relative rotation variables, $\emptyset_i$, are not independently specified given the nonholonomic motion constraints for the individual robots.

The shaded portion in Fig. 2 shows how these cluster space set-points are provided to the cluster space control architecture by the adaptive navigation system.

Simulation work was performed to verify the controller and to demonstrate several gradient-based maneuvers, as reported in [44]. Furthermore, simulation was used to evaluate the effects of noise, gradient strength, and formation-keeping capability on performance, as summarized in Section VI.

IV. Multikayak Experimental Testbed

To experimentally demonstrate the gradient-based adaptive control system, an existing set of three robotic kayaks was used as seen in Fig. 6. These kayaks were used to sense and navigate with respect to the underwater topography, with bathymetric depth serving as the parameter of interest. To do this, the kayaks were outfitted with simple, single beam sonar sensors in order to perform local depth measurements. These measurements and the previously described adaptive navigation control elements were integrated with the fleet's existing cluster space control system.

The baseline multikayak system is a low-cost, student-developed testbed that has existed for several years and that has been used to demonstrate a variety of multirobot navigation capabilities with up to six boats [10]. Each boat is identical, with a commercially available "sit-on-top" kayak serving as the hull. The kayak is propelled by two Minn Kota Endura 30 trolling motors arranged in a differential drive configuration and controlled by a Roboteq AX3500 motor controller. These motors are powered by a marine deep-cycle battery, allowing 3 h of standard operations and speeds up to 5 kn. Simple aluminum and PVC chassis elements attach electronic components to the hull and provide for rapid assembly of the system in the field.

Each kayak uses a Garmin 18 differential GPS unit and a digital Devantech CMPS30 compass for position sensing, providing sensing accuracy on the order of ± 3 m and 3°, respectively. For depth readings, a Garmin Intelleducer sonar provides 1 Hz data up to a maximum depth of 275 m with an accuracy of ± 1 m. Two on-board BasicX microcontrollers provide basic data acquisition and formatting. They also handle the parsing functions and serve as an interface between the on-board sensors/actuators and a wireless communication system that integrates the system with the off-board cluster control system. An isolated 12 V battery system provides power to the sensor, computing, and communications components.

Fig. 6. Three kayak cluster operating in Stevens Creek Reservoir.

Each kayak is wirelessly connected to a remote control station, which executes the adaptive navigation controller and serves as an operator interface to the system. The wireless system uses two Metrocom Ricochet transceivers capable of 128 Kb/s speeds and robust communications up to 1.5 mi. The station consists of a standard Windows-based laptop computer running the controller, which executes within a MATLAB/Simulink environment. The DataTurbine streaming software connects the Simulink controller with a simple serial port application that manages the interface with the wireless communication equipment. We note that this software architecture is used extensively by the research team for several other low-cost multirobot testbeds; although it has performance limitations, its capability is more than sufficient for the control requirements of these systems, it is easily maintained and configured by a student research team, and it provides simple integration with a variety of other networked tools, interfaces, and simulators available to the team.

V. Experimental Results

To experimentally verify the gradient-based adaptive navigation technique, distributed depth measurements were made in order to perform tasks such as navigating up/down underwater slopes and following bathymetric contours. Depth was used as the parameter of interest for initial field testing because such fields are static, they are easily measured using the existing multirobot system, and we can create high resolution truth data using an alternate system that performs science-grade bathymetric mapping [14].

Experiments were performed at two sites. The first was Stevens Creek Reservoir in Cupertino, CA, USA, which is a routine test location for various marine robotic systems developed at Santa Clara University. As a man-made entity, the bathymetric profile is a simple concave shape with contours that follow the coastline and depths that are completely in range of the sonar units on each boat. In addition, the main part of this reservoir had been extensively mapped by the team prior to navigation experiments as part of a separate research effort. The second site was in Lake Tahoe approximately a half mile off the coast of Camp Richardson on the Southwestern shore of the Lake, a location known to have a descending ravine but for which detailed maps were unavailable. After the experiments were executed, the team used the Lab's bathymetric mapping system to map a portion of the region of operation in order to verify results.

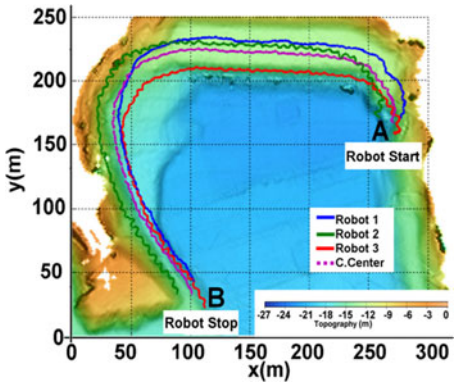

Fig. 7. Track of the kayak cluster during a holonomic contour following operation in August 2012 at Stevens Creek Reservoir. A depth contour of 11.5 m was specified, with the cluster commanded to maintain a triangular formation of $(p, q, \beta) = [18\text{ m}, 18\text{ m}, 90^\circ]$.

Experiments were performed to verify both gradient climbing/descent as well as contour following. In addition, both cluster motion modes, nonholonomic-like and holonomic-like, were demonstrated; however, in this paper, we only present holonomic-like maneuvers given our preference for that option and limitations on space.

A. Steven's Creek Contour Following Demonstration

The Steven's Creek tests were run during algorithm development in order to iteratively test and improve the control system. The lack of long paths of depth change prohibited compelling demonstrations of gradient ascent/descent. However, the man-made topography provided an outstanding venue for demonstrating the contour-following capability. Fig. 7 shows the result of such a contour-following experiment, with the cluster moving CCW from locations A to B, around the northern edge of the reservoir, following a depth value of $z = 11.5$ m while maintaining a desired shape of $[p, q, \beta] = [18\text{ m}, 18\text{ m}, 90^\circ]$.

Fig. 8 shows the precise behavior of the cluster during this experiment. In Fig. 8(a), the sensor data from each robot and the computed centroid depth are shown. As can be seen, the cluster centroid depth successfully tracks the desired value of 11.5 m with an rms error of 1.2 m. Fig. 8(b) and (c) indicates the ability of the cluster to maintain its specified shape of $[p, q, \beta] = [18\text{ m}, 18\text{ m}, 90^\circ]$ during this contour-following operation. Shape, β, is maintained with an rms error of 13.1°, and the size parameters p and q are controlled to within an rms error of 3.4 and 4.6 m, respectively. Formation-keeping performance and its impact on adaptive navigation performance are discussed in the next section.

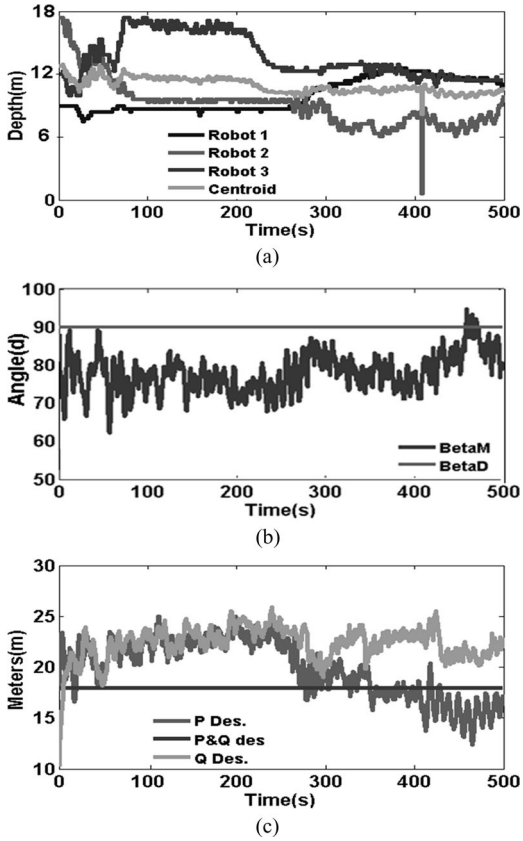

Fig. 8. Time histories of a contour following operation in August 2012 at Stevens Creek Reservoir. (a) Time response of robot depths, with the centroid depth maintained at 11.5 m. (b) Time response of cluster shape, β, with a desired value of 90°. (c) Time response of cluster size parameters, p and q, with a desired value of 18.5 m for each.

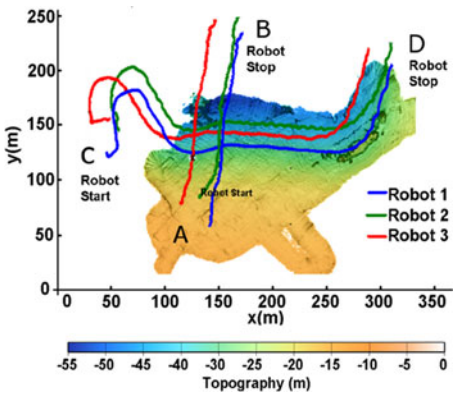

Fig. 9. Track of the kayak cluster during two separate operations in August 2012 in Lake Tahoe, CA, USA. Tracks A–B show a holonomic gradient descent operation and tracks C–D show a holonomic contour following operation.

B. Lake Tahoe

The Lake Tahoe tests were executed in the general area in which a shallow, descending ravine was known to exist, but for which there was no available high-resolution bathymetric data. Once the experiments were executed, the team used a different automated mapping system to create the bathymetric data used for verification. Fig. 9 shows the paths of the cluster robots superimposed on this bathymetric data for two different experimental runs. The tracks running from points A to B in Fig. 9 show how the cluster moved in response to a command to descend the local gradient while maintaining a shape of $[p, q, \beta] = [30\text{ m}, 30\text{ m}, 90^\circ]$. The partial bathymetric data upon which the paths are overlaid show that the cluster successfully performs this task.

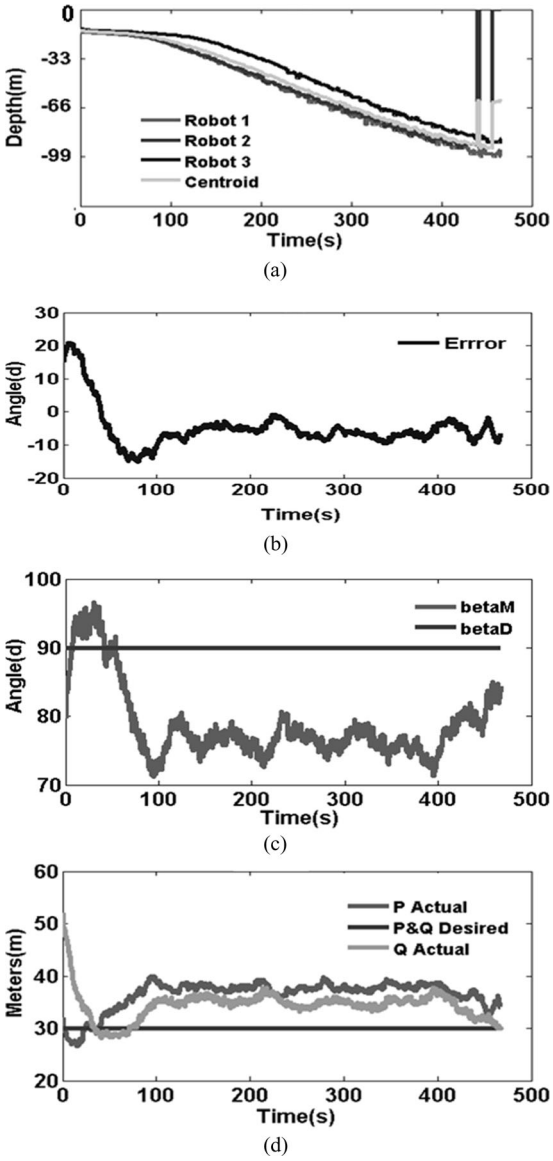

Fig. 10. Time histories for a gradient descent operation in August 2012 in Lake Tahoe. (a) Time histories of robot depth and centroid depth, showing a steady gradient descent. (b) Error between the estimated gradient and cluster's direction of travel. (c) Time history of shape, β, with a desired value of 90°. (d) Time response of cluster size parameters, p and q, with a desired value of 30 m for each.

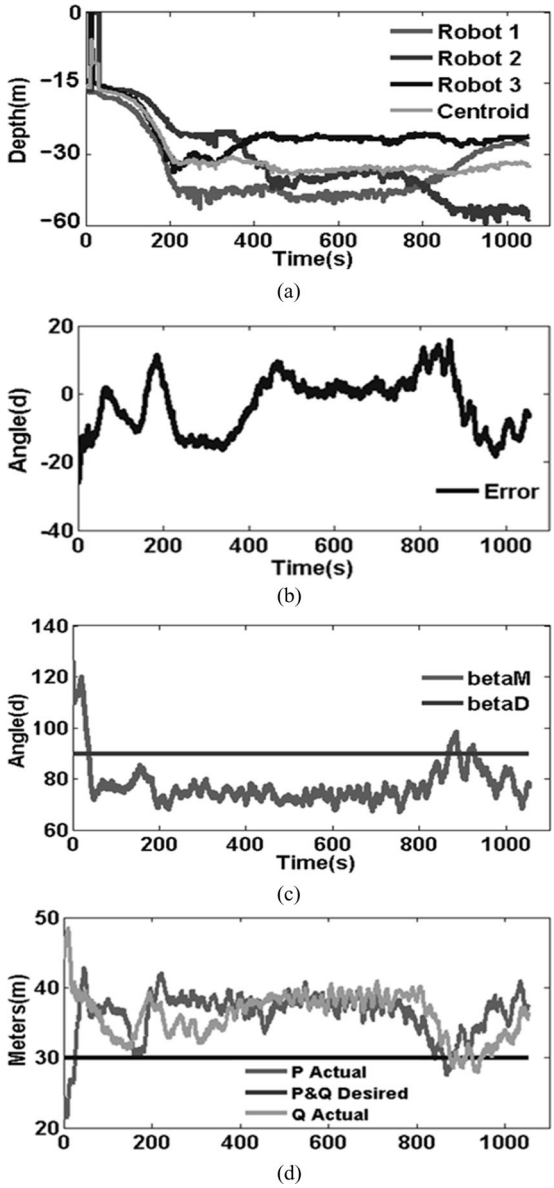

Fig. 11. Time histories for a gradient tracking operation in August 2012 in Lake Tahoe. (a) Time histories of robot depth and centroid depth, showing a steady gradient tracking path. (b) Error between the estimated gradient and cluster's direction of travel. (c) Time history of shape, β, with a desired value of 90°. (d) Time history of size, p and q, with a desired value of 30 m for both.

More precisely, Fig. 10 shows time history data for various sensed parameters during this maneuver. In Fig. 10(a), the depth data from each robot are shown; as expected, the data show steadily increasing depth as the cluster descends the gradient. During the maneuver, these data are used to compute the bearing of maximum descent and the cluster is steered to that bearing. Fig. 10(b) shows the bearing error over time as the cluster attempts to track the direction of maximum descent. Once the initial transient dies out, this steering maneuver is accomplished with an rms error of 6.1°. Finally, Fig. 10(c) and (d) shows the degree to which cluster shape is maintained during the maneuver. Shape, β, is maintained to its desired value of 90° with an rms error of 13.6°. The cluster size parameters, p and q, are controlled to within an rms error of 7.0 and 6.1 m, respectively, of their desired values of 30 m. The impact of formation-keeping performance on adaptive navigation functionality is discussed in the next section.

In Fig. 9, the tracks running from points C to D show robot motion when the cluster was commanded to follow a CCW contour of $z = 33$ m while maintaining a shape of $[p, q, \beta] =$ [30 m, 30 m, 90°]. As can be seen, the cluster moves along a topographic contour as expected. Fig. 11 shows the precise behavior of the cluster during this experiment. In Fig. 11(a), the sensor data from each robot are shown. After an initial transient, the robots descend from a depth of about 17 m to the desired parameter value of 33 m. At about $t = 200$ s, the centroid of

the cluster reaches the desired parameter value and follows that contour with an rms error of 0.9 m. The ability to track the gradient is also characterized in Fig. 11(b), which shows the computed instantaneous bearing of the contour and the travel bearing of the cluster; the cluster follows the contour bearing with an rms error of 8.9°.

Fig. 11(c) and (d) shows the degree to which cluster shape is maintained during the contour following maneuver. Shape, β, maintains its desired value with an rms error of 15.3°. The cluster size parameters p and q are controlled to within an rms error of 7.3 and 6.7 m, respectively.

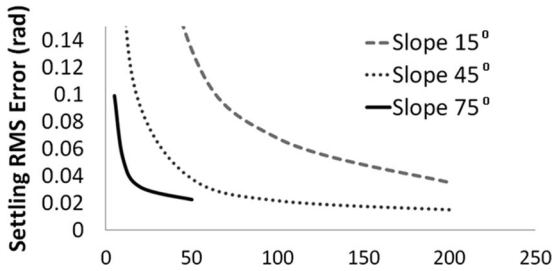

Fig. 12. Characterization of cluster turning performance as a function of cluster size and the magnitude of the scalar field gradient.

VI. Discussion of Results

The previous section provides positive experimental verification of the system's ability to perform gradient-based navigation for the purposes of descending gradients and following contours. That said, there are a number of interesting issues relating to the implementation of this capability both with the inner loop cluster controller as well as with the outer-loop adaptive navigation process.

A. Formation-Keeping Performance

With respect to performance of the inner loop cluster controller, shown in Figs. 8, 10, and 11, it is apparent that high precision control has not been achieved. One contributing factor is the quality of navigation sensors being used. The depth sensors have a ± 1 m resolution, and the GPS sensors are limited to ± 3 m of accuracy; performance could be improved through the use of a filtering or estimation process. Furthermore, a very simple proportional, resolved rate controller was used; performance could be enhanced through the addition of integral control, the use of a nonlinear controller, and/or the use of a true dynamic controller, all of which we have implemented in previous investigations [42]. Nevertheless, this linear controller was more than capable enough to provide the level of performance necessary for the adaptive navigation loop to function well. As discussed next, this is largely due to the insensitivity of gradient estimation over the error ranges encountered for the system (e.g., cluster shape was kept in its benign range, and a slightly larger than desired cluster may have even improved gradient estimation).

B. Adaptive Navigation Performance

Beyond sensing or control limitations, adaptive navigation has been implemented to date with a constant cluster geometry. From supporting simulation work and ad hoc field tests, it is clear that the selection of the cluster's shape and size can significantly affect performance given the magnitude of the gradient, the presence of noise, and the structure (shape, frequency content, etc.) of the parameter field.

To demonstrate this, we considered the performance of a simple turn maneuver within a simple inclined, planar scalar field, assuming noisy sensor readings that approximated the ± 1 m magnitude errors that we experience with our depth sensors. A series of simulations were run in which the cluster size, (p, q), was varied while holding the shape constant at $\beta = 90°$. Fig. 12 summarizes the results of this exercise for three different slopes of the planar scalar field.

Two predominant effects can be seen. First, performance improves dramatically as the slope of the scalar field increases. Second, as size increases, performance also improves. Both of these results are due to the same phenomena. Because the gradient estimate relies on the scalar differences across the cluster, in the presence of noise, larger scalar field differences across the cluster lead to a smaller noise impact. Both a more prominent scalar field gradient and larger clusters lead to larger scalar field differences. We note that the rapidly decreasing performance (e.g., increasing rms errors) at very small cluster sizes is due predominantly to nearing a singular point in the cluster geometry (which occurs when two or more robots are at the same location), which leads to computational amplification of errors [35].

Similar simulations show the effect of cluster shape, β, on the ability of the cluster to navigate with respect to gradients. This result showed satisfactory results over most of the range of β, except in the vicinity of $\beta = 0°$ and $\beta = 180°$. In these configurations, the triangular cluster becomes a straight line leading to the loss of gradient information in one of the two sensing degrees of freedom.

C. Comparison to Other Gradient-Based Navigation Methods

As previously mentioned, multirobot gradient estimation allows for faster and more responsive adaptive navigation than the reviewed techniques relying on a single robot for gradient estimation. This is because a single robot must take additional time to perform the spatial sampling necessary to generate a gradient estimate; this approach is further challenged if the field is time varying.

Compared to the more sophisticated multirobot estimation techniques explored by Leonard *et al.*, we first note that we have experimentally implemented our system. Our system adaptively navigates by continuous updating fleet velocity set-points in real time; because the Leonard group's work focuses on underwater robots that are usually out of communication contact, their approach proposes to adaptively navigate by replanning and periodically reloading waypoint-based trajectories that the AUVs would then execute in isolation [27]. Finally, relevant to adaptive navigation robustness to formation errors, we note that our surface position errors of a few meters is orders of magnitudes less than the simulated errors of 100s and at times 1000s of

meters of position errors achieved by Leonard's group; their work showed that their adaptive navigation approach is still feasible even with errors of this magnitude [29].

D. Future Work

Of course, a significant challenge remains for fields with a more complex spatial structure. In our current three-robot implementation, fields are approximated as being locally planar; when this approximation is poor, performance decays. To date, we have avoided this by operating in conditions such that cluster size was much smaller than the wavelength of any predominant parameter variation.

We have several initiatives to extend this adaptive navigation work in multiple ways. Regarding control system performance, estimation and filtering techniques will be adopted to enhance performance given sensor noise; similarly, a model-based dynamic cluster space controller could be used [42]. In terms of field systems, we will soon be re-equipping the kayak fleet with marine sensors in order to perform several environmentally oriented missions of interest to our science partners. Examples of these include identifying anoxic regions in estuaries (oxygen minima, in partnership with the Monterey Bay Aquarium Research Institute), finding and navigating "up" pollutant plumes along the coast (such as nitrogen from fertilizer run off, in partnership with scientists from NOAA), and finding specific underwater topographic features (such as the tips of underwater glacial moraines, in partnership with geologists from the U.S. Geological Survey).

We are also extending conceptual elements of this work. This includes developing methods for dynamically reshaping the cluster to adapt to the spatial frequency content within the parameter field. It also includes how to employ and geometrically control additional robots in order to effectively perform gradient-based navigation. We also plan to explore adaptive navigation within time-varying parameter fields and to address current limitations relating to being captured by local extrema. We also plan to apply the adaptive navigation technique to different domains through the use of terrestrial, aerial, and underwater robots, given that we have multirobot testbeds for each. These extensions include the estimation and navigation of gradients in 3-D volumes. Identifying, finding the sources of (gradient following), and exploring the extent of (via contour following) plumes in aerial and underwater environments are specific missions that we have been requested to support using our techniques.

Beyond adaptive navigation, we continue to develop the theoretical elements of our cluster control technique through the development of hierarchical cluster-of-clusters constructs and of propagation algorithms to dynamically generate the kinematic transforms for arbitrary cluster state space definitions. We are also using the cluster space control architecture as the foundation of other task-specific controllers in order to achieve functions for enabling reconfigurable sparse antenna arrays [12], for implementing mobile multistatic tracking networks and for manipulating objects.

VII. Conclusion

This paper presents an experimentally demonstrated multirobot control architecture that uses real-time estimates of parameter gradients to adaptively drive a cluster of robots relative to features within a sensed parameter field. Distributed measurements of the field are taken and used to compute the field's gradient and steering set-points are generated for the aggregate system in order to perform functions such as ascending/descending the gradient to find local maxima/minima and following contours in order to explore the extent of the field.

The system's architecture uses a cluster space controller as an inner control loop, with the gradient-based adaptive navigation controller serving as an outer control loop. The functionality of this controller has been verified through field experiments using three robotic kayaks and a centralized off-board control computer. These kayaks operate in a plane and sense the bathymetric depth field below them. Experiments demonstrate the ability of the three-robot cluster to follow gradients (drive the cluster to deeper water) and contours (drive the cluster along topographic ridges). Ongoing and future work is extending this technique to accommodate larger robotic clusters, more complex parameter fields, and 3-D fields appropriate for aerial and underwater applications.

Appendix

The forward position kinematics for the three-robot planar system used in this experiment are developed in detail in [13]. These relationships are given by

$$x_c = \frac{x_1 + x_2 + x_3}{3} \tag{14}$$

$$y_c = \frac{y_1 + y_2 + y_3}{3} \tag{15}$$

$$\theta_C = a\tan 2\left(2x_1 - x_2 - x_3, 2y_1 - y_2 - y_3\right) \tag{16}$$

$$\emptyset_1 = \theta_1 + \theta_C \tag{17}$$

$$\emptyset_2 = \theta_2 + \theta_C \tag{18}$$

$$\emptyset_3 = \theta_3 + \theta_C \tag{19}$$

$$p = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{20}$$

$$q = \sqrt{(x_3 - x_1)^2 + (y_1 - y_3)^2} \tag{21}$$

$$\begin{aligned}\beta = a\tan 2((x_3 - x_1)\sin(\alpha) + (y_3 - y_1)\cos(\alpha),\\ (x_3 - x_1)\cos(\alpha) - (y_3 - y_1)\sin(\alpha))\end{aligned} \tag{22}$$

$$\alpha = a\tan 2\left(y_1 - y_2, x_2 - x_1\right).$$

Acknowledgment

The authors would also like to extend their thanks to several individuals who have supported this effort: M. Neumann for feedback on this publication and his support in executing field experiments, P. Mahacek for his early work in developing

the multikayak experimental testbed, M. Rasay for his work in developing the distributed software architecture used in the experimental system, and V. Howard for his work in codeveloping early versions of the gradient-climbing capability.

REFERENCES

[1] C. A. Kitts and M. Egerstedt, "Design, control, and applications of real-world multirobot systems [from the Guest Editors]," *IEEE Robot. Autom. Mag.*, vol. 15, no. 1, p. 8, Mar. 2008. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=04476318

[2] P. Yuanteng and M. W. Mutka, "STARS: Static relays for remote sensing in multirobot real-time search and monitoring," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 10, pp. 2079–2089, Oct. 2012. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6336742

[3] S. G. Sukhatme, D. Caron, and M. J. Mataric, "Proposed approach for combining distributed sensing, robotic sampling, and offline analysis for in-situ marine monitoring," *Proc. SPIE*, vol. 4205, pp. 278–288, 2001.

[4] O. Khatib, "Mobile manipulation: The robotic assistant," *Robot. Auton. Syst.*, vol. 26, no. 2, pp. 175–183, 1999.

[5] A. Winfield, "Future directions in tele-operated robotics," *Telerobotic Appl.*, pp. 147–163, 2000.

[6] A. E. Quaid and R. L. Hollis, "Cooperative 2-DOF robots for precision assembly," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 3, Minneapolis, MN, USA, Apr. 1996, pp. 2188–2193.

[7] A. Gasparri, S. Panzieri, and A. Priolo, "A fitness-sharing based genetic algorithm for collaborative multi-robot localization," *Intell. Serv. Robot.*, vol. 3, no. 3, pp. 137–149, 2010.

[8] D. W. Miller, R. J. Sedwick, E. M. C. Kong, and S. Schweighart, "Electromagnetic formation flight for sparse aperture telescopes," in *Proc. IEEE Aerosp. Conf.*, Big Sky, MT, USA, Mar. 2002, vol. 2, pp. 729–741.

[9] I. Mas, S. Li, J. Acain, and C. A. Kitts, "Entrapment/escorting and patrolling missions in multi-robot cluster space control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, St. Louis, MO, USA, Oct. 2009, pp. 5855–5861.

[10] P. Mahacek, C. A. Kitts, and I. Mas, "Dynamic guarding of marine assets through cluster control of automated surface vessel fleets," *IEEE/ASME Trans. Mechatronics*, vol. 17, no. 1, pp. 65–75, Feb. 2012. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6097059

[11] I. Mas and C. A. Kitts, "Object manipulation using cooperative mobile multi-robot systems," in *Proc. World Congr. Eng. Comput. Sci.*, San Francisco, CA, USA, Oct. 2012, pp. 324–329.

[12] G. Okamoto, C. Chen, and C.A. Kitts, "Beamforming performance for a reconfigurable sparse array smart antenna system via multiple mobile robotic systems," *Proc. SPIE*, vol. 7706, pp. 77060–77071, Apr. 2010.

[13] C. A. Kitts and I. Mas, "Cluster space specification and control of mobile multirobot systems," *IEEE /ASME Trans. Mechatronics*, vol. 14, no. 2, pp. 207–218, Apr. 2009.

[14] C. A. Kitts, P. Mahacek, T. Adamek, K. Rasal, V. Howard, S. Li, A. Badaoui, W. Kirkwood, G. Wheat, and S. Hulme, "Field operation of a robotic small waterplane area twin hull boat for shallow-water bathymetric characterization," *J. Field Robot.*, vol. 29, no. 1, pp. 924–938, 2012.

[15] H. Ishidaa, K. Suetsugua, T. Nakamotoa, and T. Moriizumia, "Plume-tracking robots: A new application of chemical sensors," *Biol. Bull.*, vol. 200, pp. 222–226, Apr. 2001.

[16] M. A. Willis and E. A. Arbas, "Odor-modulated upwind flight of the sphinx moth, Manduca sexta," *J. Comparative Physiol.*, vol. A169, pp. 427–440, 1991. Available: http://link.springer.com/article/10.1007%2FBF00197655#page-2

[17] W. Li, J. A. Farrell, and S. Pang, "Moth-inspired chemical plume tracing on an autonomous underwater vehicle," *IEEE Trans. Robot.*, vol. 22, no. 2, pp. 292–307, Apr. 2006. Available: http://www.public.navy.mil/spawar/Pacific/Robotics/Documents/Publications/2006/IEEE_RO_A05_254LiFarrellPangArrieta.pdf

[18] E. Burian, D. Yoerger, A. Bradley, and H. Singh, "Gradient search with autonomous underwater vehicle using scalar measurements," in *Proc. IEEE Symp. Auton. Underwater Veh. Technol.*, Monterey, CA, USA, Jun. 1996, pp. 86–98.

[19] J. Adler, "Chemotaxis in bacteria," *Science*, vol. 153, no. 3737, pp. 708–716, 1966.

[20] R. Bachmayer and N. Ehrich Leonard, "Experimental test-bed for multi-vehicle control, navigation and communication," presented at the 12th International Symposium on Unmanned Untethered Submersible Technology, Durham, NH, USA, Aug. 2001.

[21] J. Choi, S. Oh, and R. Horowitz, "Cooperatively learning mobile agents for gradient climbing," in *Proc. 46th IEEE Conf. Decision Control*, Los Angeles, CA, USA, Dec. 2007, pp. 3139–3144.

[22] C. Zhang, A. Siranosian, and M. Krstic, "Extremum seeking for moderately unstable systems and for autonomous target tracking without position measurements," presented at the IEEE American Control Conference, Minneapolis, MN, USA, Jun. 2006.

[23] C. Zhang, D. Arnold, N. Ghods, A. Siranosian, and M. Krstic, "Source seeking with nonholonomic unicycle without position measurement—Part I: Tuning of forward velocity," in *Proc. 45th IEEE Conf. Decision Control*, San Diego, CA, USA, 2006, pp. 3040–3045.

[24] A. T. Hayes, A. Martinoli, and R. M. Goodman, "Distributed odor source localization," *IEEE Sensors J.*, vol. 2, no. 3, pp. 260–271, Jun. 2002.

[25] E. Biyik and M. Arcak, "Gradient climbing in formation via extremum seeking and passivity-based coordination rules," *Asian J. Control*, vol. 10, no. 2, pp. 201–211, 2008.

[26] R. Bachmayer and N. Leonard, "Vehicle networks for gradient descent in a sampled environment," in *Proc. 41st IEEE Conf. Decision Control*, Las Vegas, NV, USA, 2002, pp. 112–117.

[27] E. Fiorelli, P. Bhatta, N. Leonard, and I. Shulman, "Adaptive sampling using feedback control of an autonomous underwater glider fleet," in *Proc. 13th Int. Symp. Unmanned Untethered Submersible Technol.*, Durham, NH, USA, 2003, pp. 1–16.

[28] P. Ogren, E. Fiorelli, and N. Leonard, "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *IEEE Trans Autom. Control*, vol. 49, no. 8, pp. 1292–1302, Aug. 2004.

[29] E. Fiorelli, N. Leonard, N. Ehrich, P. Bhatta, D. Paley, R. Bachmayer, and D. Fratantoni, "Multi-AUV control and adaptive sampling in Monterey Bay," *IEEE J. Ocean. Eng.*, vol. 31, no. 4, pp. 935–948, Oct. 2006. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=04089061

[30] N. J. Vickers and T. C. Baker, "Reiterative responses to single strands of odor promote sustained upwind flight and odor source location by moths," *Proc. Nat. Acad. Sci.*, vol. 91, pp. 5756–5760, 1994.

[31] K. Lerman, M. Alcherio, and G. Aram, "A review of probabilistic macroscopic models for swarm robotic systems," in *Proc. Int. Conf. Swarm Robot.*, 2005, pp. 143–152.

[32] E. Şahin, "Swarm robotics: From sources of inspiration to domains of application," in *Proc. Int. Conf. Swarm Robot.*, 2005, pp. 10–20.

[33] K.-H. Tan and M. Lewis, "Virtual structures for high-precision cooperative mobile robotic control," in *Proc. IEEE /RSJ Int. Conf. Intell. Robots Syst.*, Nov. 1996, vol. 1, pp. 132–139.

[34] N. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *Proc. 40th IEEE Conf. Decision Control*, 2001, vol. 3, pp. 2968–2973.

[35] J. Fredslund and J. M. Maja, "A general algorithm for robot formations using local sensing and minimal communication," *IEEE Trans. Robot. Autom.*, vol. 18, no. 5, pp. 837–846, Oct. 2002.

[36] F. Schneider and D. Wildermuth, "A potential field based approach to multi robot formation navigation," in *Proc. IEEE Int. Conf. Robot., Intell. Syst. Signal Process.*, Oct. 2003, vol. 1, pp. 680–685.

[37] Z. Wang and D. Gu, "A local sensor based leader-follower flocking system," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 3790–3795.

[38] R. Fierro, A. Das, J. Spletzer, J. Esposito, V. Kumar, J. Ostrowski, G. Pappas, C. Taylor, Y. Hur, R. Alur, I. Lee, G. Grudic, and B. Southall, "A framework and architecture for multi-robot coordination," *Int. J. Robot. Res.*, vol. 21, no. 10/11, pp. 977–995, Oct./Nov. 2002.

[39] I. Mas, J. Acain, O. Petrovic, and C. A. Kitts, "Error characterization in the vicinity of singularities in multi-robot cluster space control," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, Bangkok, Thailand, Dec. 2008, pp. 1911–1917.

[40] I. Mas and C. A. Kitts, "Obstacle avoidance policies for cluster space control of nonholonomic multirobot systems," *IEEE /ASME Trans. Mechatronics*, vol. 17, no. 6, pp. 1068–1079, Dec. 2012.

[41] S. Agnew, P. Dal Canto, C. A. Kitts, and S. Li, "Cluster space control of aerial robots," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics*, Montréal, QC, Canada, Jul. 2010, pp. 1305–1310.

[42] I. Mas, C. A. Kitts, and R. Lee, "Model-based nonlinear cluster space control of mobile robot formations," *Multi-Robot Systems, Trends and Development*, Rijeka, Croatia: In Tech, ch. 4, 2011, pp. 53–71.

[43] I. Mas and C. A. Kitts, "Centralized and decentralized multi-robot control methods using the cluster space control framework," in *Proc. IEEE /ASME Int. Conf. Adv. Intell. Mechatronics*, Montreal, QC, Canada, 2010, pp. 115–122.

[44] C. Kitts, P. Mahacek, T. Adamek, and I. Mas, "Experiments in the control and application of automated surface vessel fleets," in *Proc. OCEANS*, Waikaloa, HI, USA, Sep. 2011, pp. 1–7.

Thomas Adamek (S'99–M'11) received the B.S. degree from the University of Windsor, Windsor, ON, Canada, and the M.S. degree in mechanical engineering from Santa Clara University, Santa Clara, CA, USA.

He was a Development Test Engineer in the automotive sector, following which he was Technical Director working with WWF in Peru. He is currently a Research Associate at Santa Clara University.

Christopher A. Kitts (S'98–A'00–M'03–SM'05) received the B.S.E. degree from Princeton University, Princeton, NJ, USA, the M.P.A. degree from the University of Colorado, Boulder, CO, USA, and the M.S. and Ph.D. degrees from Stanford University, Stanford, CA, USA.

He was a Research Engineer and an Operational Satellite Constellation Mission Controller. He was an Officer in the U.S. Air Force Space Command, a NASA Contractor with Caelum Research Corporation, Department of Defense Research Fellow, and the Graduate Student Director of the Space Systems Development Laboratory, Stanford University. He is currently an Associate Professor at Santa Clara University, Santa Clara, CA, where he is also the Director of the Robotic Systems Laboratory.

Ignacio Mas (S'06–M'12) received the Engineering degree in electrical engineering from the Universidad de Buenos Aires, Buenos Aires, Argentina, and the Ph.D. degree in mechanical engineering from Santa Clara University, Santa Clara, CA, USA.

He was a Satellite Systems Engineer and Spacecraft Systems Designer at the NASA Ames Research Center. He is currently a CONICET Assistant Researcher at the Instituto Tecnologico de Buenos Aires, Buenos Aires. His research interests include multirobot systems, coordinated navigation, and formation control.