IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 6, 2018, accepted February 28, 2018, date of publication March 22, 2018, date of current version April 23, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2818622

Adaptive Navigation Control Primitives for Multirobot Clusters: Extrema Finding, Contour Following, Ridge/Trench Following, and Saddle Point Station Keeping

CHRISTOPHER A. KITTS, (Senior Member IEEE), ROBERT T. MCDONALD, (Student Member IEEE), AND MICHAEL A. NEUMANN, (Member IEEE)
Santa Clara University, Santa Clara, CA 95053, USA
Corresponding author: Christopher A. Kitts (ckitts@scu.edu)

ABSTRACT Adaptive navigation is the process of modifying a vehicle's direction or motion path based on measurements taken while moving. When exploring a scalar field, such as the temperature or the concentration level of a pollutant across a region of interest, adaptive navigation may allow the identification of locations of interest—like the maximum temperature or the source of the pollutant—without exhaustively mapping the entire region. Adaptive navigation has been hailed as a powerful capability, and significant work has been performed to explore how such techniques can be used to find the local extreme points and follow contour levels in a field. Our own prior work in this field has matured to experimentally verifying and validating such capabilities through field demonstrations. Beyond extrema-finding and contour following, however, little to no prior work has been performed on moving to/along other critical features in a scalar field, such as down ridges, up trenches, and to saddle points; performing such maneuvers can be valuable for a number of applications. In this paper, we provide and verify via simulation new multirobot adaptive navigation controllers for moving with respect to these new features. We also present a multilayered control architecture that unifies the execution of all of our multirobot adaptive navigation control primitives: extrema finding, contour following, ridge/trench following, and saddle point positioning. In addition, we review several considerations related to the performance of these controllers within unknown scalar fields. Finally, we review ongoing and future work to experimentally verify our new controllers, improve and extend their performance, and apply them to real field applications.

INDEX TERMS Adaptive navigation, adaptive sampling, gradient-climbing, differential control, multirobot formations, formation control, cluster space control.

I. INTRODUCTION

In a conventional navigation scenario, a vehicle is provided with a series of waypoints; the trajectory between these paths may also be prescribed. In contrast, Adaptive Navigation (AN) is the process of determining or modifying the vehicle's direction or path through a region based on measurements taken while moving through that region.

In the simplest form of AN, the destination is explicitly specified with alterations to the path permitted. A common example of this is rerouting an automobile trip in order to avoid traffic [1]. For robotic vehicles, a common example of this is to avoid obstacles or adjust a route due to challenging terrain. For wheeled robots, this has been achieved by modifying artificial potential field methods with fuzzy logic approaches in order to move safely through dynamic and unstructured environments [2]. Fuzzy logic has also been applied to adaptive navigation for legged robots which handle obstructions differently than wheeled robots given their ability to climb over some obstacles [3]. For next generation Mars missions, multiple capabilities such as identifying physical obstacles, estimating wheel slippage probabilities throughout the region, and creating dynamic predictions of path-following performance are integrated to incrementally plan a route to a desired location [4].

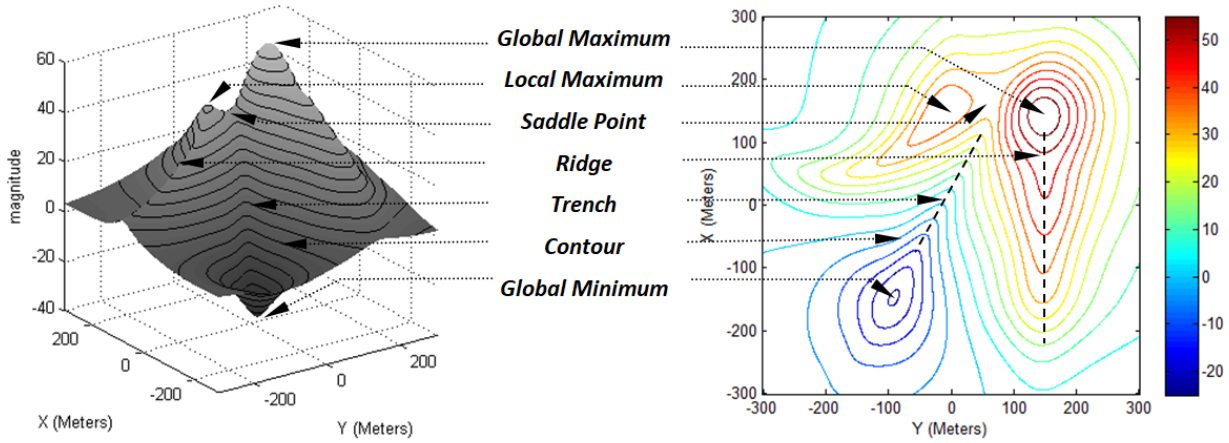


FIGURE 1. Critical Points within a Scalar Field: (left) 3D view of scalar surface, (right) overhead view of equivalent contour map.

In a second form of AN, the destination may not be specified at all. Examples of this include techniques to evade enemies in a pursuit scenario [5] and to manage evacuation routes during disasters [6].

A third form of AN consists of moving to or along defined conditions of interest within a region without knowing their explicit location *a priori*, such as navigating to the location of maximum temperature or pollutant concentration level. In such scenarios, AN can dramatically reduce time and/or energy compared to a conventional approach of identifying interesting features once a region has been exhaustively mapped; furthermore, AN can be used when the region's characteristic of interest is time-varying, a condition that can render conventional mapping approaches useless. In general, the required AN motion decisions to move to/along features of interest are based on some knowledge of the structure of or a critical characteristic of the local scalar field; such knowledge is generated through direct measurement by the vehicle(s) as it/they move through the field.

A. SCALAR FIELDS

The most prevalent version of the third class of AN, and the focus of this paper, involves navigation through scalar fields, which are fields that associate a single scalar value with each location within the field. For a planar region, the scalar value is often depicted as the altitude of a surface at the sampled point in a plane, as depicted in Figure 1. Scalar fields are often used to represent environmental parameters such as temperature or light levels, barometric pressure, radiation or pollutant concentration measurements, and so on.

Interesting features in a scalar field include the minimum and maximum values, contours of specific value, ridges and trenches within the field, and saddle points. In such fields, extreme points might represent the location of important features such as a heat or pollution source or perhaps the center of an anoxic region. Contours define the location of specific values or concentrations and are important in defining the extent of a feature, specifying safety thresholds, etc. Ridges (trenches) define maximum (minimum) signal paths from (to) critical points in a field; they also serve as divides (accumulators) for other processes that flow with respect to gradients (e.g., water flows away from ridges and gathers in streams, etc.). Finally, saddle points serve as minimum energy gateways for movement between adjoining maxima (minima) (e.g., travelers in mountainous regions move through saddle point ''passes'' in mountainous regions when journeying between adjoining peaks or valleys). Altogether, these features are the critical elements of a scalar field; all are of interest for a wide range of applications, and being able to navigate to/along all of these provides a robust set of primitive capabilities for an adaptive navigation system.

B. SINGLE VEHICLE SCALAR FIELD AN

A variety of techniques have been proposed to navigate single vehicle systems with respect to a scalar field.

Bio-inspired techniques are often intuitive but lack formality, typically requiring significant verification via simulation or experimentation. Such techniques have been explored primarily for plume tracing, in which a well-defined scalar field ridge exists and the objective is to find and move up the ridge to the source. In cases where directional or flow measurements are possible, a sensor array can be used to sense the local upstream direction and compute a gradient in gas concentration, mimicking the manner in which in which moths sense the direction of pheromone plumes [7]. Moth behavior also inspired a successful field demonstration of plume tracing using an autonomous underwater vehicle with a behavior-based control scheme; once the plume is sensed, this controller drives the vehicle perpendicular to the flow and repeatedly in and out of the plume, turning upstream each time it passes through the plume [8].

When only point measurements are possible (or when field changes at the scale of the robot are too small and/or noisy to support differential sensing), single vehicles may take several measurements in order to compute a field gradient. For

example, a sliding mode controller has been demonstrated in simulation in order to drive a single robot to extreme points without the use of gradient estimation [9].

Experimentally, the underwater robotic vehicle the Autonomous Benthic Explorer (ABE) used a bio-inspired approach consisting of periods of random spatial motion, similar to the chemotaxis behavior of E. coli, to generate a local distribution of altitude measurements; this allowed the vehicle controller to compute a bathymetric gradient and to navigate to a specific location in a lake with respect to the lakebed's topology [10], [11]. Other experimental work included adding a sinusoidal perturbation or some other spatial dither to a vehicle's path in order to gather enough local spatial information to compute the field gradient and then move accordingly [12], [13].

With respect to following contours (or similarly, level sets, boundaries, etc.), a sliding mode control approach for moving a non-holonomic vehicle along a time-varying contour without gradient information has been verified both in simulation [14] and in a hybrid experiment using a real vehicle navigating with respect to a simulated (computed, not sensed) field in an indoor $\sim 7\text{m}^2$ region [15].

C. MULTIROBOT SCALAR FIELD AN

Single vehicle approaches can suffer from the costly maneuvers associated with sensing the local gradient [16]. Multi-vehicle systems, on the other hand, are able to gather distributed spatial information instantaneously, eliminating the need for spurious motions that cause delays and which are detrimental in time-varying fields. Furthermore, multi-vehicle systems can better compensate for vehicle failures, can change their distributed aperture size and shape in response to varying spatial frequencies within the field, and can employ spatially-distributed sensing approaches other than gradient-sensing, such as differential sensing across the distributed aperture or the execution of decentralized approaches. There have been a number of studies of Multi-robot Adaptive Navigation (MAN), although few have been demonstrated experimentally and almost none have been implemented in the field.

A bio-inspired multirobot plume tracing capability was demonstrated via simulation in [17]. This approach consisted of a lead robot attempting to follow the plume using a single-robot movement strategy but with two other robots flanking the leader; as the leader moved out of the plume, one of the flankers would presumably still be in the plume and become the new leader.

With respect to more general extrema-finding, a swarm approach has been demonstrated in simulation in which individual robots repel at short distances but are attracted to those with higher field readings over longer distances, thereby moving the swarm to a local maximum [18]. An alternate decentralized approach aligns robots laterally and bases the forward speed of each robot on the local field reading in order to advance a group to an extreme point; this has been done in simulation for up to 8 robots and has been performed experimentally with two robots using the variation of light level over a distance of a few feet [19].

In [20], a group of robots successfully climbed a gradient, but this is done by a single robot using a spatial dither to compute the gradient and move towards the field maxima while also serving as the leader in a leader-follower formation. In [21], a robot group shared distributed measurements to compute a very low resolution estimate of the gradient, but then only a single vehicle would act, or the single vehicle would ascend the gradient while commanding other vehicles to move in the same direction. Demonstrated experimentally using small-scale lab equipment, the implementation used only binary ''in plume'' or ''out of plume'' data; nevertheless, it showed 25-40% time reductions in plume tracking compared to using only a single vehicle, with most of the performance gain occurring with the use of only three robots.

In [22], a probabilistic gradient estimation technique is demonstrated in simulation and shown to be robust to noise; by using an ample number of mobile sensors (more than 3 times the minimal number for an idealized planar gradient estimate), this approach is able to accommodate significant deformations in the loosely controlled group's geometry. In [23], a minimal multirobot gradient estimation technique (3 robots for planar gradient estimation) is proposed along with a coordinated navigation policy for the robot group to climb or descend the gradient. This work is matured in [16] with the introduction of a provably stable adaptive navigation strategy that decouples the formation keeping task from the gradient-ascent/descent task. This strategy was later evolved and evaluated via detailed simulations using an archive of real underwater robot temperature data, leading to the development of constraints for the specific vehicles used in that project, the assessment of specific formations, and the conclusion that gradient-based navigation would be feasible if implemented experimentally [24], [25], although such experimental implementation has yet to be accomplished. Experimental demonstration performed by [26] used three robots in a table top light-sensing scenario; this work demonstrated a centralized gradient estimation approach as well as a distributed technique appropriate for instances of limited communication among the mobile platforms. Other experimental demonstrations of gradient-based extrema finding includes our own work with wheeled robots responding to the signal strength of a radio frequency field, performed within a $\sim 1500\ \text{m}^2$ outdoor testbed [27].

With respect to tracking/following contours (level sets, boundaries), a number of research teams have used multi-robot strategies in which the contour seeking/following functionality is executed largely on an individual basis but with loose interaction among robots to achieve certain capabilities. In one such study, simulation was used to show individual robots already stationed around a time-varying elliptical contour move independently to maintain their position on the contour; they then share their location to collectively generate an estimate of the elliptical [28]. In [29], robots individually moved toward and track a contour but shared information

to establish efficient spatial spreading among the robots in order to save energy in route to the contour given the ultimate objective of collectively circumnavigating the contour; this was demonstrated in simulation, and a single robot version was demonstrated experimentally by tracking illumination level within a $\sim$1.25 m^2 field. In [30], multiple vehicles track a colored tape over a $\sim$3 m^2 field in order to track a boundary; similar experiments are performed by [31] using vehicles driving over colored mats within a $\sim$50 m^2 workspace. These systems used simple, single robot wall/tape following threshold controllers to follow the contour, with periodic interaction to avoid collision and/or promote convoy-like behavior between robots.

A number of studies have explored tighter interaction among multirobot groups in order to collectively navigate to/along contours. In one such study, a swarm of robots within a boundary use gradient information to move outward to find the desired contour but interact with each other to evenly space themselves around the field. This work exploited the ''snake algorithm'' approach for image segmentation functions in image processing. Verified in simulation, this work evolved in several steps to the point at which it can accommodate noisy sensor data and asynchronous communication [32], [33]. Another team further extended the use of the snake algorithm approach to show simulations of multirobot groups following moving two- and three-dimensional level sets/surfaces [34]. Alternatively, in [35], simulations were used to demonstrate the use of a cooperative filter using Hessian information to move a robot group along noisy level curves; active formation shaping was used to minimize errors in gradient estimation. This work was extended in [36] to enable a group of six robots to move along three-dimensional level set surfaces in order to track the principal lines of the surface's curvature.

To our knowledge, the only field-grade experimental use of MAN is our own work, conducted with non-holonomic autonomous surface vessels in Lake Tahoe, CA and Stevens Creek Reservoir, CA in order to ascend/descend bathymetric gradients and to follow bathymetric contours; these operations were performed in regions of interest on the order of $\sim$75,000 m^2 [37].

D. OTHER RELATED TECHNIQUES

MAN has similarities to potential field based navigation, which has been explored by many research teams [38]–[41]. A typical potential field-based navigation scenario involves navigation between two established points with the vehicle ''descending'' a field that is artificially created by placing a sink at the destination, peaks at the location of sensed obstacles, optional trenches along a desired trajectory, and so on.

The MAN scenario is significantly different. To begin, the destination, the desired path and the scalar field itself are all unknown a priori. In addition, the scalar field represents a real environmental phenomenon rather than artificially imposed navigation construct. Furthermore, characteristics of the local scalar field are estimated based on sensed environmental data rather than having the desired velocity or applied force be computed based on an explicitly defined potential field function. Finally, while potential fields are typically used to direct the desired velocity or force on a vehicle ''down'' the surface of the composite field, once the local field is sensed in MAN, multiple field characteristics (e.g., scalar value, gradient, differential offsets, etc.) may be applicable and they may be used in a range of ways (move along or opposite to, move perpendicular to, specify a net or differential stimulus, etc.).

If anything, the techniques are complimentary; in fact, we note that MAN field implementations may include potential field-based collision avoidance, which we have demonstrated at the level of both an individual vehicle as well as at the multirobot formation level [42], [43]. In addition, some researchers use potential field constructs to enable formation-keeping [44] as an inner loop for or independent of the MAN layer of the overall multirobot control system [16].

MAN is also different than work on scalar field estimation, which has the objective of estimating the entire field; adaptive motion control for that application usually has the objective of determining where to sample the field in order to establish the best field estimate [45]–[47]. MAN, on the other hand, seeks to adaptively move to/along specific features of interest in order to support the needs of specific applications, unconcerned with estimating the field in the vicinity of features that are not of interest.

E. CURRENT FOCUS

As a summary of the described prior work, it can be seen that MAN is a powerful technique given its ability to efficiently locate and/or move along scalar field features of interest. It is a field, however, that is still in its infancy. While outstanding analytic and simulation work has been accomplished for a wide range of MAN approaches, this work has focused completely on extrema-finding and contour following; little to no work has been done on techniques to navigate to/along other scalar field features, such as down a ridge, up a trench, or to a saddle point in a field. In addition, very little work has been done to experimentally verify any of the developed techniques, with most experiments being performed with indoor testbeds having severe constraints on the size of the field, the capability of the robotic vehicles, and the nature of the field itself. To our knowledge, the only field-grade experimental use of MAN is our previously mentioned work with autonomous surface vessels [37].

The primary contribution of this article is in the presentation and simulation-based verification of new control primitives for descending ridges, climbing trenches, and locating saddle points within a scalar field. These primitives use a minimal number of robots and simple, reactive, differential control laws that use currently sensed data; as such, they provide an initial proof-of-concept of these capabilities. An additional contribution is the use of a multilayered control framework to unify the execution of these new control primitives with our previously developed primitives for extrema finding and

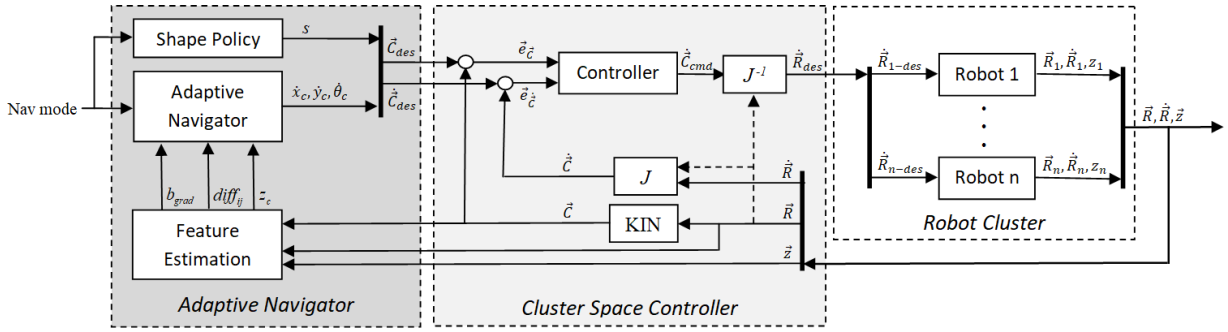


FIGURE 2. The Adaptive Navigation Layered Control Architecture: In the lowest layer (right), individual robots execute closed loop velocity commands. A formation control layer (middle) issues robot-level velocity setpoints in order to maintain a specified cluster geometry and to drive the cluster as commanded. Finally, an adaptive navigation layer (left) provides cluster geometric setpoints and drive commands based on scalar field measurements and the selected navigation mode.

contour following. A summary of these previously developed gradient-based controllers is included in order to 1) show how control architecture unification is achieved, 2) compare the behavior of all of these control primitives in the context of a single scalar field example, and 3) highlight the comprehensive nature of our collection of MAN control primitives which collectively support navigation to/along every critical feature in a scalar field. Furthermore, we discuss several considerations and performance constraints relating to the use of these control primitives. Finally, we describe ongoing work in experimentally demonstrating the new ridge/trench/saddle point primitives, improving their performance, applying them to specific field applications, and extending this work to more complex fields.

Section II of this paper reviews the layered control architecture used for our MAN work. Because performance of the primitives is a function of the relative position of the robots, the adaptive navigation control layer issues directives to a lower level formation controller that enforces the spatial geometry of the robot group; this formation controller, in turn, issues drive commands to each robot. Throughout the paper, in order to focus on the adaptive navigation control primitives, we assume ideal formation-keeping and holonomic motion behaviors at the formation and robot level. In Section III of the paper, we briefly review our three-robot, gradient-based controllers for extrema-seeking and contour following, and we illustrate their behavior via simulation through the use of a notional scalar field; we also show a simple case of contour following for a time-varying scalar field. Section IV presents our minimal five-robot differential approach for ridge descent, trench ascent, and saddle point positioning. These primitives are demonstrated via simulation using the same notional scalar field used in Section III. Section V discusses performance considerations for our suite of controllers, and Section VI highlights ongoing and future extensions of this work. Finally, Section VII summarizes the work performed and draws conclusions.

II. THE LAYERED CONTROL ARCHITECTURE

Central to our adaptive navigation technique is the ability to sense, compute, and move with respect to spatial characteristics of a scalar field. We do this with a minimal number of robots, each with the ability to sample the field and share information. Motion of the robots is specified by a formation controller that maintains the relative position of the robots to ensure the collection of scalar field data in all spatial dimensions and at an appropriate spatial resolution. At a higher level, the adaptive navigator ingests realtime scalar field samples from the robots, computes relevant field characteristics (such as the gradient or differential offsets), and determines appropriate motion commands for the multirobot formation using a reactive control policy. This layered control approach is pictured in Figure 2.

A. CLUSTER SPACE CONTROL LAYER

For formation control, we adopt a cluster space control strategy. The cluster space technique is an operational space control approach that treats the multirobot formation as a full degree-of-freedom, articulating virtual mechanism. This virtual mechanism can move through its environment while changing its shape and size as demanded by any task. The controller accepts specifications and computes compensations in the cluster space, converting to/from robot-specific state variables through the use of kinematic transforms [48].

In this technique, the pose of the robot cluster is represented by the location and orientation of the cluster frame (the location of which is defined by the user as some function of robot positions), the relative orientations of each robot with respect to the cluster frame (which are not typically freely specified due to robot nonholonomic constraints), and a set of variables representing the spatial geometry of the robots. These pose variables, defined by $\vec{\boldsymbol{C}}$, and their derivatives define the cluster state space. For a cluster of $\boldsymbol{n}$ robots each with $\boldsymbol{m}$ degrees of freedom, nonlinear kinematic transforms of the form shown in Eq. (1) are used to relate these cluster pose variables to conventional robot pose variables, defined by $\vec{\boldsymbol{R}}$. The velocity relationship between spaces is a linear (although pose-dependent) transform represented by a Jacobian matrix, $\boldsymbol{J}$, of the form shown in Eq. (2), where {G} represents a fixed Global frame of reference. The inverse relationships may also

be found.

$$\dot{\vec{C}} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_{mn} \end{pmatrix} = KIN\left({}^{G}\vec{R}\right) = \begin{pmatrix} g_1(r_1, r_2, \cdots, r_{mn}) \\ g_2(r_1, r_2, \cdots, r_{mn}) \\ \vdots \\ g_{mn1}(r_1, r_2, \cdots, r_{mn}) \end{pmatrix} \tag{1}$$

$$\dot{\vec{C}} = {}^{G}J\left({}^{G}\vec{R}\right){}^{G}\dot{\vec{R}} \tag{2}$$

Within the cluster control layer, robot space state information is converted to cluster space state values through the forward kinematic functions, KIN and J, as shown in Figure 2. Cluster space errors are computed by subtracting the computed cluster space state from the cluster space setpoints provided by the adaptive navigation control layer. An error-driven resolved rate control law computes cluster velocity compensation commands, and these commands are converted to robot-specific velocity setpoints via the inverse Jacobian transform.[1] Although not used for the simulations performed in this article, this control architecture can accommodate potential field collision avoidance capability either at the level of the individual robot or for the aggregate cluster. In addition, Lyapunov stability has been characterized for arbitrary cluster space command trajectories (such as those that will be generated by the adaptive navigation layer), with or without the collision avoidance options [43].

Use of the cluster space formation control layer allows the desired cluster geometry and motion goals to be specified and monitored in the cluster space, a capability that significantly eases supervision by a human operator and provides a convenient abstraction layer for higher level controllers such as the adaptive navigation control layer. In addition, computing control compensations in the cluster space typically leads to well-behaved geometric motion even if individual robot motion is highly complex.

For our prior implementations of MAN extrema finding and contour following, a three robot cluster was used [27], [37]. For our new controllers for ridge descent, trench ascent and saddle point positioning, we introduce a new five-robot cluster, which is described in Section IV of this paper.

B. ADAPTIVE NAVIGATION CONTROL LAYER

The primary function of the adaptive navigation control layer is to direct the motion of the robot formation in order to execute the selected navigation objective. This is a feedback process given that the control law is a function of scalar field measurements taken by each of the spatially-distributed robots.

In particular, the Feature Estimation block within the adaptive navigation layer shown in Figure 2 estimates parameters relating to the nature of the scalar field based on the realtime sensor measurements from each robot in the cluster. In our prior work focused on extrema finding and contour following, the role of this block was restricted to gradient estimation. Given the new control primitives proposed in this paper for ridge descent, trench ascent, and saddle point positioning, we have expanded the role of this function to be the location in the control architecture where estimates of scalar field parameters are generally computed. This includes gradient estimates (b_{grad}) for extrema finding and contour following as well as differential field measurements (d_{ij}) between robots in the cluster and the scalar field value at the origin of the cluster frame (z_c).

Within the Adaptive Navigator block, specification of the navigation mode (e.g., which control primitive to invoke) leads to the use of the appropriate scalar field parameter estimates generated by the Feature Estimation block, and the selection of the relevant MAN control laws. As is described in Section III, for extrema-finding and contour following, the robot cluster is controlled to move in a particular direction with respect to the field gradient. As described in Section IV, ridge/trench following is achieved by using the differential field measurements to move the cluster such that it straddles the ridge/trench while also moving in the appropriate direction along the feature; saddle point positioning generally occurs at the end of a ridge/trench maneuver if/when the cluster motion settles.

Ultimately, an additional role of the adaptive navigation layer is to specify the shape and size of the multirobot cluster to ensure that viable estimates of the field gradients and/or differential measurements are computed given the spatial frequencies within the field, the magnitude of the gradient, noise, and mission parameters. For this study, a fixed geometry appropriate for the given mission and scalar field is assumed in order to focus on the aggregate motion control strategies; modulating the geometry of the cluster, however, is an important area for future work. The adaptive navigation layer can also be used to specify a holonomic or nonholonomic style of aggregate cluster motion; in this paper, holonomic cluster motion is assumed, but details on how to accommodate both options can be found in [37].

III. ADAPTIVE NAVIGATION FOR EXTREMA FINDING AND CONTOUR FOLLOWING

In prior work, we experimentally implemented both extrema finding and contour following through navigation policies that steer the robot cluster based on the direction of the local scalar field gradient. In this Section, we include a summary of this work in order to provide a single article summarizing our full suite of MAN controllers, to illustrate the comparative behavior of all of these controllers within a prototypical scalar field, and to provide the proper context for showing the unification of all of the MAN controllers in the multilayered control architecture shown in Figure 2.

An estimate of the gradient can be established through the use of measurements of the scalar field taken by each of the three distributed robots, as shown in [37]; this estimate uses

[1] Full dynamic control is possible through the use of a controller that computes force/torque compensations, in which case a Jacobian transpose transform is used to convert these compensations to robot-specific force and torque inputs. For details, see [49].

an assumption that the three measurements establish a local, planar approximation of the scalar field. Given this, the role of the multirobot adaptive navigation policy is to direct the motion of the aggregate robot cluster with respect to the gradient given the objective of either moving toward a local field maximum, a local field minimum, or along a specific field contour.

A. EXTREMA-SEEKING CONTROL PRIMITIVE

Movement to a local maximum is achieved by driving the cluster in the direction of the gradient, b_{grad}. Conversely, movement to a field's local minimum is achieved by driving the cluster in the direction opposite of the gradient, $b_{grad} + \pi$ radians. Although other speed-setting policies are possible, in practice we have used a policy in which a constant cluster speed, s_c, is adopted while the direction of travel is modified in order to perform the adaptive navigation task of interest; we discuss this choice in Section V. Given this policy, extrema-seeking is performed by using the following set-points for the cluster-level motion controller:

$$(\dot{x_c})_{des} = s_c \left[\cos\left(b_{grad} + d\pi\right)\right] \quad (3)$$
$$(\dot{y_c})_{des} = s_c \left[\sin\left(b_{grad} + d\pi\right)\right] \quad (4)$$

where $d = 0$ for gradient ascent and $d = 1$ for gradient descent. In the simplest possible implementation, the cluster is driven as per Eqs. (3)-(4) while maintaining a constant shape, size and orientation appropriate for the field being explored; the relative robot headings are generally not specified independently for non-holonomic vehicles and are allowed to vary in order for the individual robots to maneuver as required to achieve the specified cluster-level motion and geometry. We note that, given a holonomic cluster-level motion strategy, the cluster heading setpoint $\left(\dot{\theta_c}\right)_{des}$ is arbitrary; for this work, however, it was set to the bearing of the gradient, (b_{grad}).

B. CONTOUR FOLLOWING CONTROL PRIMITIVE

Contour-oriented navigation may be performed by driving the cluster in a direction perpendicular to the gradient, $b_{grad} \pm \pi/2$ radians, for counterclockwise (CCW) vs. clockwise (CW) circumnavigation of the local maximum, respectively. However, because in application we often wish to follow a contour with a designated scalar value, we have adopted a path-following approach for the contour following behavior. Accordingly, we specify a bearing setpoint that is equal to the desired contour bearing (the steady state solution) plus a corrective term proportional to the scalar error, $(z_{des} - z_c)$, thereby steering the cluster toward the desired contour; the corrective term is limited to $\pm\pi/2$ radians, orienting the direction of the cluster's travel up or down the gradient in the direction of the desired contour for large deviations. Mathematically:

$$b_{cf} = b_{grad} + d\{sgn\left(z_{des} - z_c\right) \times min\left[K_{ct} \times \left\|z_{des} - z_c\right\|, \pi/2\right] - (\pi/2)\} \quad (5)$$

where $d = 1$ for CW navigation and $d = -1$ for CCW navigation, and K_{ct} is the scalar error correction gain. This path-following approach is strategically similar to that used for an operational single boat system that follows paths in order to perform bathymetric mapping applications [50]. The value z_c is the estimate of the scalar field value at the cluster's location, which is defined as the location of the origin of the cluster frame. If no robot is physically at that location, the value is estimated based on an interpolation of the measured values given the local planar estimate of the field.

Using the same constant cluster speed policy as previously discussed, contour-following is performed by using the following set-points for the cluster-level motion controller:

$$(\dot{x_c})_{des} = s_c \left[\cos\left(b_{cf}\right)\right] \quad (6)$$
$$(\dot{y_c})_{des} = s_c \left[\sin\left(b_{cf}\right)\right] \quad (7)$$

As before, for this initial implementation, the cluster is controlled to maintain constant shape, size and orientation appropriate for the field being explored, and the relative robot headings are not independently specified.

C. ILLUSTRATION OF EXTREMA FINDING AND CONTOUR FOLLOWING CONTROL PRIMITIVES

To demonstrate these policies via simulation, consider the scalar field shown in Figure 1, which covers a 600 m by 600 m square area and which has several localized minima and maxima; the scalar value ranges from approximately 65 units at the tallest peak to -20 units at the deepest abyss, and it approaches a value of 0 units at edges of the spatial field. The mathematical function used to generate this field is given in Appendix A. For both extrema finding and contour following, the field is explored by a 3-robot cluster moving at 1 m/s in an equilateral triangle with 5 m sides; the cluster has a steering time constant of 1 sec. These physical parameters are similar to the characteristics of the physical multirobot systems currently being implemented and operated by the authors; this includes Pioneer-class or ATV-style wheeled robots for land-based operation and custom-built automated kayaks for marine surface operations. As previously stated, in order to focus on the behavior of the adaptive navigation control primitives, we assume ideal formation-keeping and a holonomic motion behavior at the formation level. We note that cluster level motion commands can easily be verified to ensure against saturation by checking the robot space velocity commands computed by the inverse Jacobian function in the cluster space controller; although we have not implemented that here, we have demonstrated its capability in other work.

For the described scenario, Figure 3 shows the paths taken by the 3-robot cluster for the extrema finding behavior. Several different initial positions in the field are used, and initial cluster bearing angles are set perpendicular to the initial gradient in order to cause a transient in the cluster steering behavior. The simulation Paths A and B show gradient ascent, and path C shows gradient descent. As can be seen in all cases, the cluster is able to appropriately ascend/descend the scalar

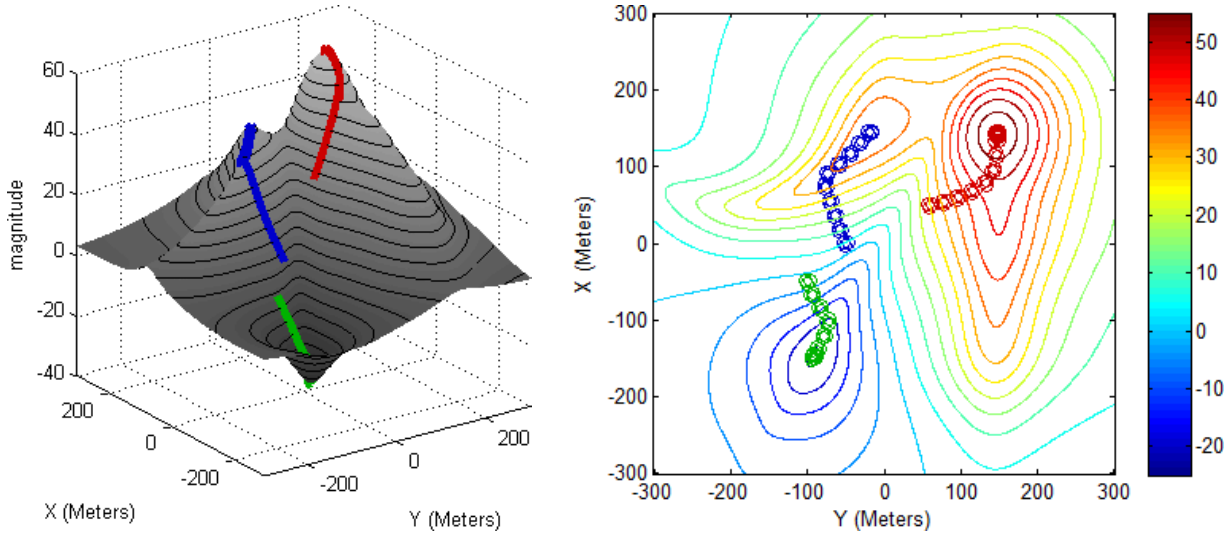


FIGURE 3. Gradient Ascent/Descent Examples: Paths A and B ascend the local gradient, with Path A ending at the global maximum and Path B ending on a local maximum. Path C descends the gradient, ending at the global minimum: (left) 3D view of scalar surface, (right) overhead view of equivalent contour map.

field with a cluster-level steering policy that drives the cluster in the direction of (Paths A and B) or opposite to the direction of (Path C) the local gradient. Steering control performance for Path B is shown in the time response plot in Figure 4; this plot shows the error in the cluster's direction of travel given an instantaneous setpoint established by the computed field gradient.

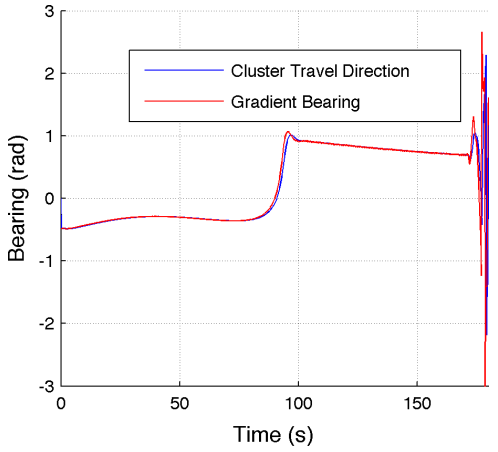


FIGURE 4. Time History of Gradient-Following Capability for Path B from Figure 3: The red line indicates the instantaneous bearing of the local gradient while the blue line indicates the actual direction of travel of the robot cluster. As expected, the cluster tracks the gradient. At approximately 170 sec, the cluster reaches the local peak. With no motion termination condition, the cluster moves back and forth in the vicinity of the peak.

Using the same field, Figure 5 shows paths taken by the 3-robot cluster for the contour-following control primitive. Several different initial positions in the field are used. Simulation Paths A and B show CCW contour following with initial positions below and above the desired contour, respectively; Path C shows CW contour following with an initial location below the desired contour. As can be seen in all cases, the cluster is able to travel to and follow the appropriate contour in the designated direction. As an example of steering control performance, Figure 6 shows the Path A time response of the scalar field error as well as the error in the cluster's direction of travel given the instantaneous bearing setpoint established by Eq. (8).

D. TIME-VARYING SCALAR FIELD

For single robot gradient-based navigation using spatial dithering, the robot must cycle through a spatial pattern each time a gradient is computed. For a time-varying field, this results in computational delay that distorts the gradient estimate, potentially to the point that the field cannot be navigated. The multirobot strategy does not suffer from this problem due to its ability to instantaneously sample the field and compute a realtime gradient.

As a simple demonstration of the ability of a multirobot cluster's ability to track a moving contour, Figure 7 shows results from a case using the same prototypical field and robot formation. In particular, a field of constant shape translates at a constant velocity that is 50% of the speed of the cluster, and the cluster is commanded to track a contour value of 45 units. Figure 7 shows the cluster's ability to track the specified contour in a frame fixed to the moving scalar field. Figure 7 also shows the cluster's motion in the global frame with a contour overlay showing the initial position of the field; the field moves to the right in the positive $\hat{y}_G$ direction. Figure 8 provides a time history of the cluster's scalar value, showing a small steady state scalar offset that leads or lags the desired contour as the cluster rotates around it; we note that a more sophisticated controller using integral or feed-forward control could be used to lower this error.

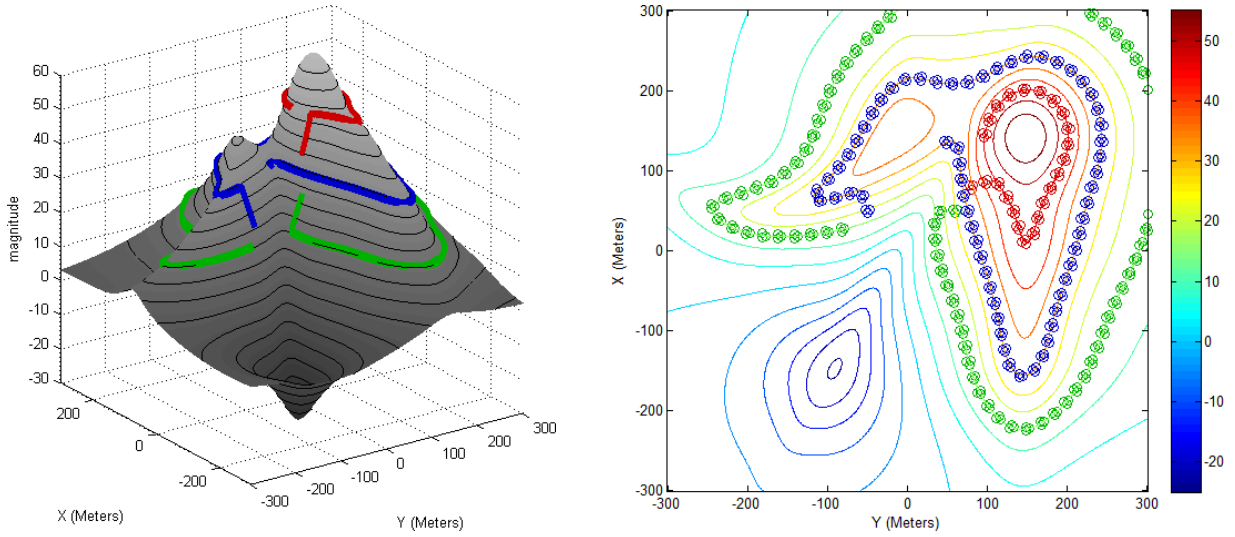


FIGURE 5. Contour-Following Examples: Paths A and B move CCW about the local peak(s), with Path A ascending the scalar field to its contour level setpoint and Path B descending to its contour level setpoint. Path C moves CW about the local peak and ascends to its contour level setpoint: (left) 3D view of scalar surface, (right) overhead view of equivalent contour map.

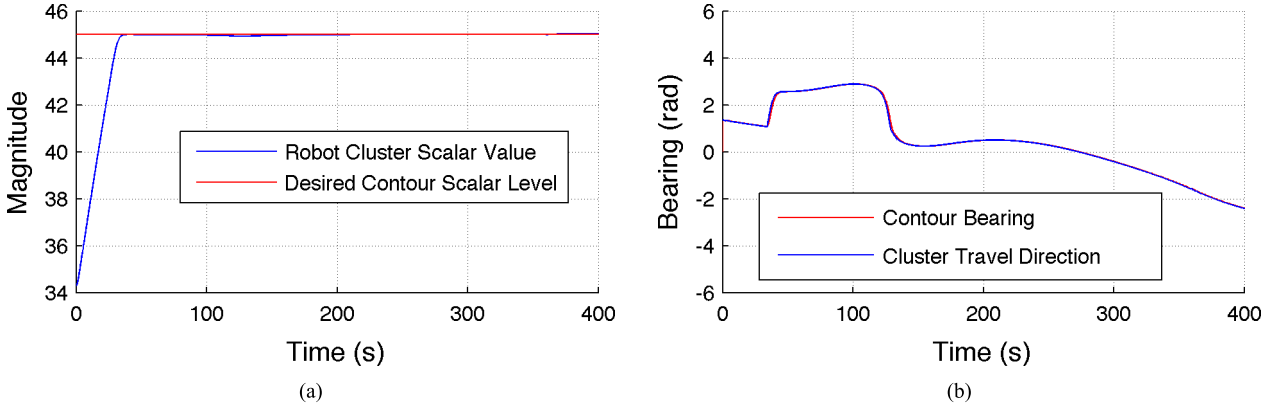


FIGURE 6. Time History of Contour-Following Capability for Path A from Figure 5. (a) Time history of the robot cluster's scalar value as it moves to the desired contour level of 45 units. (b) Time history of bearing control, showing how the robot cluster's heading tracks the desired contour bearing given the contour-following setpoint generation policy.

IV. ADAPTIVE NAVIGATION FOR RIDGE/TRENCH FOLLOWING AND SADDLE POINT STATION KEEPING

In addition to the control primitives described in Section III, we desire primitives to navigate down ridges / up trenches as well as to hold position on or identify the location of a saddle point. In the context of this work, a descending ridge / ascending trench is a continuous path on the virtualized scalar surface with a monotonically decreasing / increasing elevation and composed of points which are a surface maximum / minimum in a crossing axis. Alternatively, a saddle point is a location on the virtualized scalar surface which is a stationary point that is a relative minimum in one direction but a relative maximum along a crossing axis.

To our knowledge, little to no work has been done on adaptive navigation techniques specific to these capabilities. Ridge descent is important for applications such as following plumes downstream to locate their impact zones, identifying departure paths that provide maximum parameter "service levels" as a function of distance from the source (e.g., for extended wireless communications, etc.), locating "divides" in an environment for processes that are driven by parameter gradients (such as gravity driven flows with respect to terrain), and so on. Trenches serve as accumulators for gradient driven processes and also represent paths of minimal exposure for approaching a set of dangerous sources. Saddle points often serve as passageways for minimum energy or exposure paths between adjacent peaks or valleys.

We note that while a gradient ascent/descent approach works for ridge ascent / trench descent, the opposite – descending ridges and ascending trenches – cannot be achieved through such strategies. This is because the gradient in the vicinity of these features often diverges from the alignment of the feature of interest. Accordingly, we need a new strategy to descend a ridge and ascend a trench.

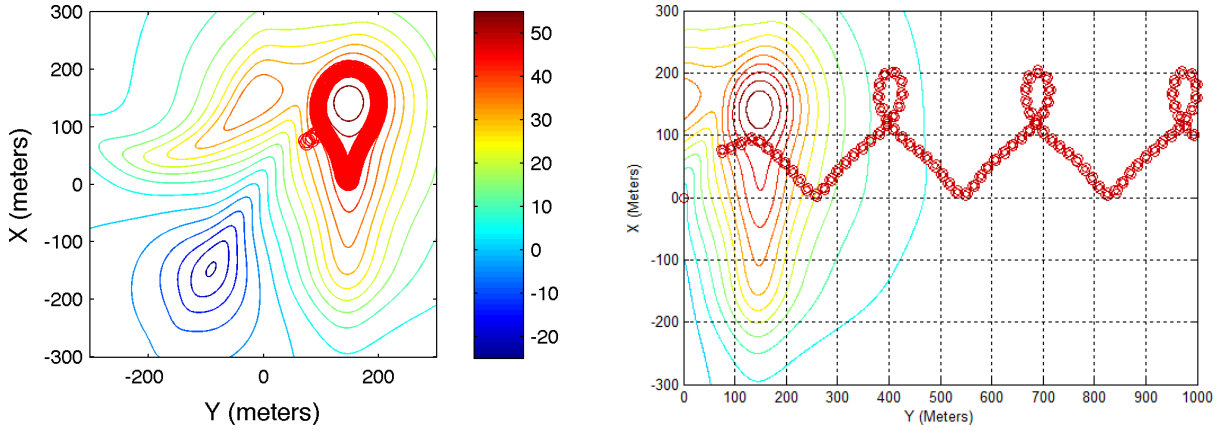


FIGURE 7. Contour Following on a Moving Scalar Field: A cluster executes contour following Scenario A from Fig. 7, moving up to and then following the desired contour, but in this case, the scalar field is moving in the positive Y direction: (left) overhead view of cluster's path in the moving field frame; (right) overhead view of the cluster's path in the non-moving global frame, with the scalar field contour shown for a time of t = 0 sec. A supplemental file associated with this article shows an animation of this behavior.

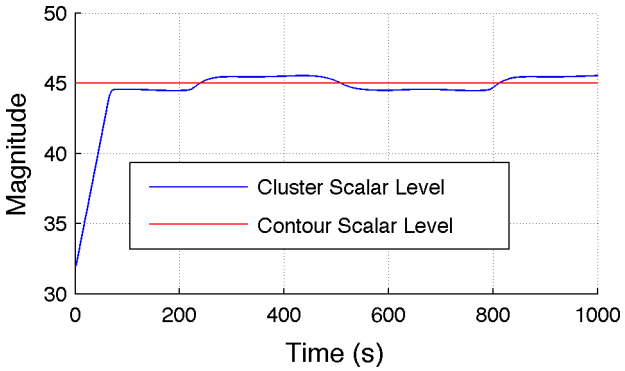


FIGURE 8. Time History of the Cluster Scalar Value while Contour Following on a Moving Scalar Field: Once transient motion has settled, the cluster exhibits a small steady-state error which lags the desired contour level in the direction of its travel; this lag, however, results in the cluster's scalar value alternating between having a lower then higher value than the desired scalar value.

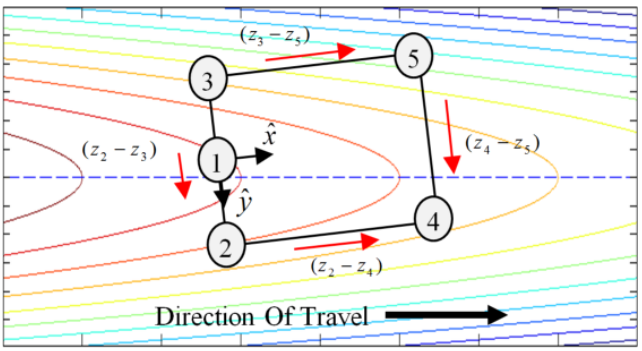


FIGURE 9. Differential Drive Compensation Signals for Ridge Following: The cluster is straddling the ridge but has both a lateral and rotational offset. Longitudinal scalar differences indicate the desired direction of travel along the ridge. Additionally, lateral scalar differentials can be used in a differential drive strategy to align the cluster laterally and rotationally.

As will be described, differential sensing can be used to generate control signals that adjust the lateral position of a cluster that straddles a ridge/trench and orient a cluster properly as it travels down/up a ridge/trench.

A. DIFFERENTIAL SENSING AND CONTROL STRATEGY

Consider the rectangular five-robot cluster shown in Figure 9. The cluster space definition assumed for formation control purposes is detailed in Appendix B. The desired motion for this cluster is to move down the ridge, represented by the dotted line in the contour field shown in Figure 9, while straddling it laterally and maintaining rotational alignment with it. When straddling the ridge, differential scalar field measurements can be generated both laterally and longitudinally across the cluster. These differential signals can be used to generate cluster velocity commands that will produce the control motion to maintain a straddling pose with respect to the ridge.

More specifically, consider the differential scalar field signals shown in Figure 9:

- Longitudinal differentials $(z_2 - z_4)$ and $(z_3 - z_5)$ point in the desired direction of travel with respect to the cluster frame's $\hat{x}$ dimension in order to drive down the ridge;
- The average of the lateral differentials $(z_2 - z_3)$ and $(z_4 - z_5)$ points in the desired direction of travel with respect to the cluster frame's $\hat{y}$ dimension in order to correct for lateral displacement;
- The difference of the lateral differentials $(z_2 - z_3)$ and $(z_4 - z_5)$ can be used to establish the sense of cluster rotation required to correct for angular displacement.

For trench following, we note that the differential arrows in Figure 9 would be drawn in the opposite directions of those shown; accordingly, the sign of corrective motions will be the opposite of those used for ridge following.

Given the assumption that the cluster is straddling the ridge, the differential signals generated by Robots 2-5 contain the critical information for guiding the cluster properly along

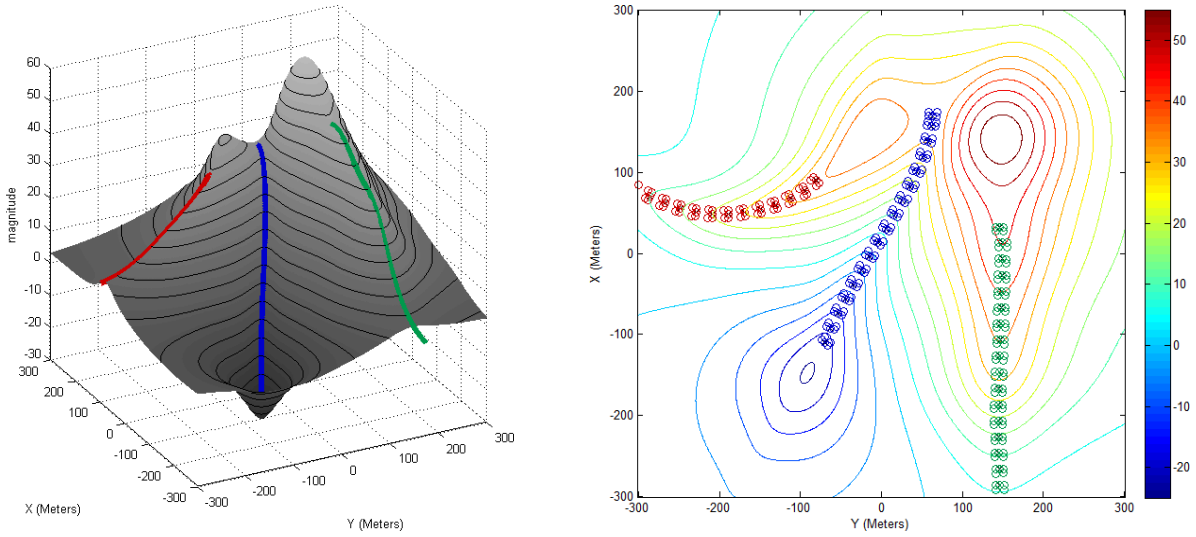


FIGURE 10. Ridge/Trench-Following Examples: Paths A and C move down ridges, while Path B moves up a trench: (left) 3D view of scalar surface, (right) overhead view of equivalent contour map.

the ridge. Robot 1 is not used in the creation of these differential signals. Instead, Robot 1 is used to verify that assumption that the cluster is, in fact, straddling the ridge. Given a well-conditioned ridge/trench, if $[(z_1 > z_2)\ and\ (z_1 > z_3)]$ / $[(z_1 < z_2)\ and\ (z_1 < z_3)]$, then Robots 2 and 3 are straddling the ridge/trench and the assumption holds. If this is not the case, the cluster may or may not be straddling the ridge/trench and a different control strategy may be necessary.

B. RIDGE/TRENCH FOLLOWING CONTROL PRIMITIVE

Given the use of longitudinal and lateral scalar field differentials generated by combinations of robot sensor data, we can now specify the motion control primitives for ridge/trench following.

$$(\dot{x}_c)_{des} = d \times v_x \{sgn[(z_2 - z_4) + (z_3 - z_5)]\} \quad (8)$$

$$(\dot{y}_c)_{des} = d \times v_y \{sgn[(z_2 - z_3) + (z_4 - z_5)]\} \quad (9)$$

$$(\dot{\theta}_c)_{des} = d \times \omega_z \{sgn[(z_4 - z_5) - (z_2 - z_3)]\} \quad (10)$$

where v_x, v_y, and ω_z are the constant velocity setpoints for each velocity component, and $d = 1$ for ridge following and $d = -1$ for trench following; the longitudinal and lateral scalar field differentials are used only to establish the sign of these discrete velocity commands. The reason for using discrete values for velocity components is discussed in Section V.

To demonstrate this policy via simulation, we continue to use the scalar field shown in Figure 1. This field is explored by a five-robot rectangular cluster with a length of 5 m (in the direction of motion) and a width of 10 m. The component velocity setpoints used in Eqs. (11)-(13) are $v_x = 1\ m/s$, $v_y = 1\ m/s$, and $w_z = 0.4\ rad/s$, values that are consistent with the capabilities of the physical robots available to the authors given the size of the cluster. For this scenario, Figure 10 shows the paths taken by the five-robot cluster for three cases. Simulation Paths A and C show motion down two different ridges, and Path B shows movement up a trench. As can be seen in all cases, the cluster is able to travel successfully along the appropriate feature in the designated direction.

As an example of lateral control performance, Figure 11 shows the time response of both lateral scalar differentials: the "front" lateral differential, $z_4 - z_5$, as well as the "rear" lateral differential, $z_2 - z_3$. After the initial transient dies out, the steady state offset for both differential values settles to within ± 0.005 scalar units; for this particular scenario, this is equivalent to a settling value for the rotational alignment error of approximately 0.2 degrees.

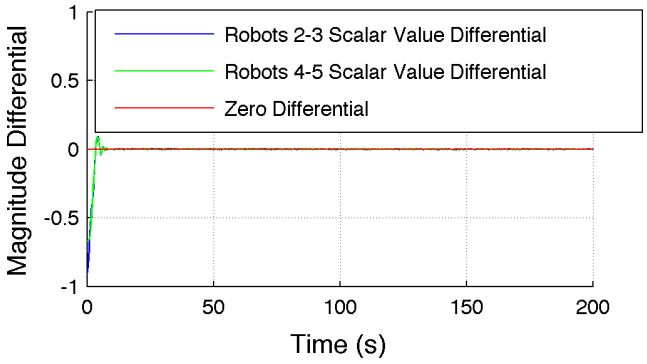


FIGURE 11. Lateral Differentials for Ridge Following Example: As the cluster moves down the ridge along Path A, the lateral differential scalar values settle to within +/−0.005 scalar units.

C. SADDLE POINT KEEPING CONTROL PRIMITIVE

The control primitive for moving to and holding position at a saddle point is identical to the primitive used for ridge/trench following, as provided by Eqs. (8)-(10). This is convenient since saddle points are generally arrived at by descending ridges or ascending trenches.

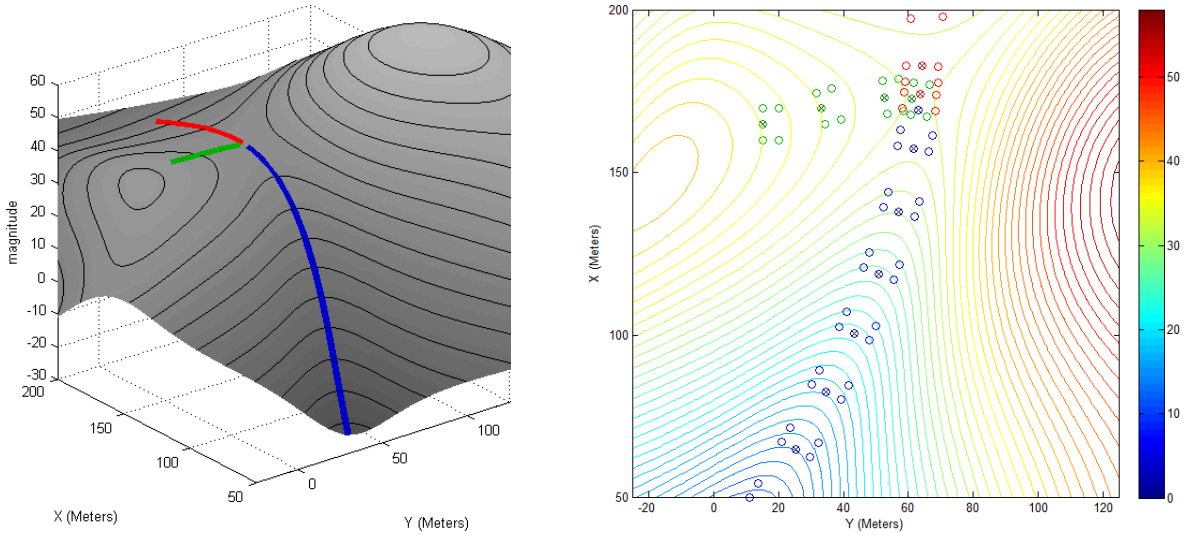


FIGURE 12. Saddle Point Positioning: Paths A and C move up a trench and settle at the saddle point, while Path B moves down a ridge and settles at the saddle point: (left) 3D view of scalar surface, (right) overhead view of equivalent contour map.

To demonstrate this via simulation, we continue to use the scalar field shown in Figure 1, the control velocities prescribed by Eqs. (8)-(10), and the velocity setpoints used in the ridge/trench following examples. Figure 12 shows the paths taken by the five-robot cluster for three cases. Simulation Paths A and C show the five-robot cluster moving up two different trenches and coming to rest at the field's saddle point. Similarly, Path B shows the cluster moving down a ridge and coming to rest at the same saddle point.

For the trench scenario, Figure 13 shows the time response of the scalar differentials, the two lateral differentials as well as the two longitudinal differentials. As expected, the two lateral differentials rapidly converge, settling to within ± 0.01 scalar units as the cluster aligns itself laterally during the climb up the trench. The longitudinal differentials are negative for the majority of the motion as the cluster moves up the trench (longitudinal differentials cause motion along the feature, and for trenches this differential is negative); however, near the end of the run as the cluster approaches the saddle point, these differentials converge causing the cluster to terminate its motion at the saddle point, with the differential values settling to within ± 0.001 scalar units.

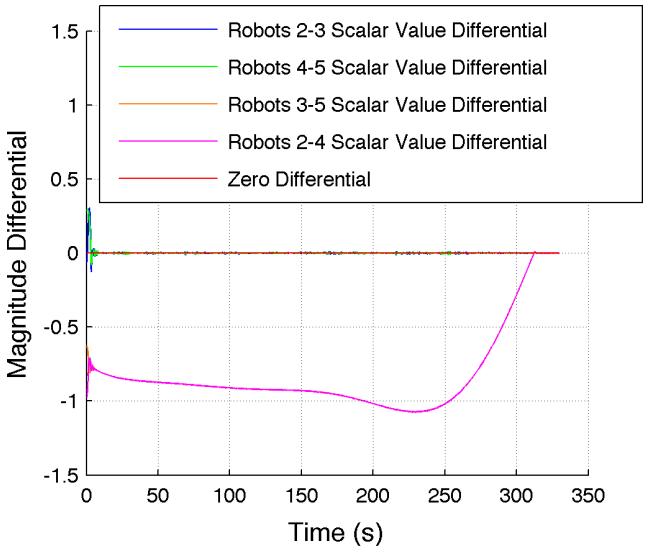


FIGURE 13. Lateral and Longitudinal Differentials for Trench Following with Settling at a Saddle Point: As the cluster moves up the trench along Path A, the lateral differential scalar values settle to within ±0.01 scalar units. The longitudinal differential scalar values are negative as the cluster ascends the trench, but then they converge to within is ±0.001 scalar units as the cluster nears and then stops at the saddle point.

V. DISCUSSION

We have described a family of simple, minimal, reactive control primitives for navigating to/along critical features within a planar scalar field. The performance of each control primitive is a function of robot operating parameters, cluster configuration, and the characteristics of the scalar field. In this section, we review several implications of the interplay between these values.

A. CLUSTER SIZE IMPACT ON GRADIENT FOLLOWING

The gradient-based controllers in this paper compute the local gradient with the assumption that the virtual plane created by the three robots is tangential to the local scalar field. As the size of the cluster increases relative to the size of the field's spatial features, however, this assumption becomes less valid. To visualize this, consider a three robot cluster positioned in field defined by a one-dimensional spatial sine wave as shown in Figure 14. Clearly, as the size of the cluster increases relative to the spatial wavelength, the slope of the cluster-defined plane diverges from the slope of the field.

Figure 15 characterizes this relationship in the form of an amplitude response, plotting the ratio of the estimated slope to the real slope as a function of the ratio of the cluster's size

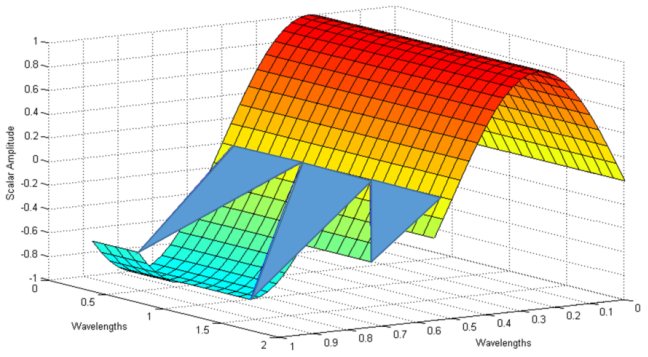


FIGURE 14. Gradient Estimation Accuracy as a Function of Cluster Size: Three triangular 3-robot clusters of varying length are shown on a scalar field that has the shape of a sine wave. As the size of the cluster increases, the slope of its planar estimate of the field deviates more and more from the local slope of the scalar field.

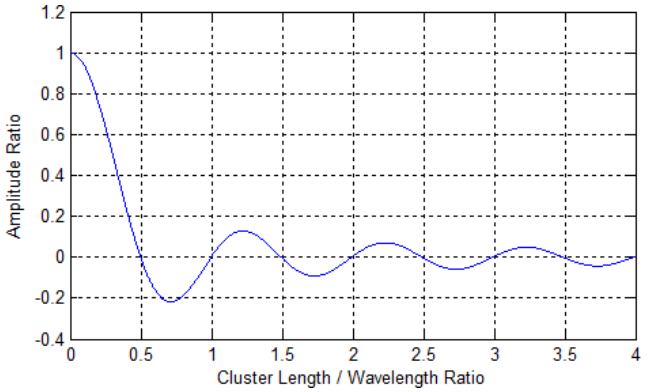


FIGURE 15. Gradient Estimate Amplitude Response: As the ratio of the cluster's length to wavelength increases, the fidelity of the cluster's ability to estimate the gradient of the field decays.[2]

to the field's wavelength. This figure implies a low pass filter behavior such that the gradient estimation ''signal'' is best passed for features with a relatively large spatial wavelength; alternatively, the gradient estimate is severely attenuated for features with a relatively small wavelength.

This relationship has an impact on appropriate cluster sizing. Clusters that are too big will filter out spatial frequencies of interest, thereby missing important application-specific features. Clusters that are too small, on the other hand, may pass noise that will corrupt the gradient estimate. Noise sources include traditional sensor and data acquisition noise as well as low amplitude, high frequency ''spatial noise'' that may exist within the scalar field.

Ideally, clusters should be small with respect to the feature sizes of interest but large compared to wavelengths characteristic of noise contributions. For example, based on the Figure 15 scenario, a mission designer might choose to limit cluster size to no more than ∼20% the size of the smallest spatial wavelength of interest (to achieve ∼80% fidelity in gradient estimation for that wavelength) while also ensuring that cluster size was at least 40% the size of the largest spatial wavelength attributable to noise (to limit the impact of noise to < 10% variation in the gradient estimate). With respect to noise, a prior simulation demonstrated improved turning performance of a gradient-following cluster as a function of increasing cluster size as well as the gradient value itself (a steeper field lowers the impact of noise on a slope computation) [37]. Furthermore, as a real world example, a previous MAN mission in Lake Tahoe, CA involving the exploration of bathymetric formations focused on features that were larger than 50 meters in size and ''noise'' on the order of one meter (which was the resolution of the sonar sensors as well as the maximum size of small rocks and boulders we wished to ignore). A triangular cluster with sides on the order of 15-25 meters was used to successfully follow contours and ascend/descend gradients.

B. CLUSTER SIZE IMPACT ON RIDGE/TRENCH TRACKING BOUNDS

The ridge/trench following primitives are designed to drive a multirobot cluster along these features within a scalar field. As previously described, lateral differential scalar signals for a ridge/trench following five-robot cluster are computed between the ''rear'' pair of robots (Robots 2 and 3) and the ''front'' robots (Robots 4 and 5) in order to compute the corrective velocities given in Eqs. 12 and 13. It was also noted that the role of Robot 1 was not to contribute to any of the differentials but to provide an instantaneous verification that the cluster was straddling the ridge/trench.

Given this, consider Figure 16, showing a cross-section of an idealized symmetric ridge and the placement of Robots 1-3 (the three robots aligned laterally in the rear of the cluster). In the left scenario, the cluster is not properly following the ridge by straddling it. In the middle scenario, the rear of the cluster is, in fact, properly straddling the ridge; however, the scalar measurements cannot be used to guarantee this given that Robot 1's measurement is less than the measurement from one of the other robots. Only the right scenario, when Robot 1's measurement is the maximum value, ensures that the cluster is straddling the ridge (given our previous

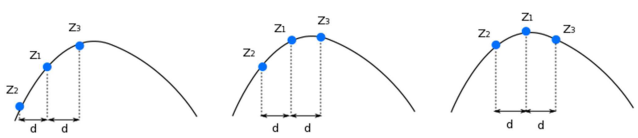


FIGURE 16. Ridge Following Verification: The left scenario shows a lateral cluster position that is not straddling the ridge. The middle scenario shows a lateral position that straddles the ridge, but measurements are unable to confirm this. Only in the right scenario does the cluster straddle the ridge in a manner that can be confirmed by sensor data. We require this in order to execute the ridge following control primitive. A similar requirement is used for trench following such that the middle robot's sensor reading is less than that for robot's 2 and 3.

[2]Although not detailed here, we note that the gradient response function depends the location of the cluster in the wavelength as well as where the cluster frame is defined within the cluster geometry. The amplitude response shown in Fig. 15 is specific to the triangular base being located at a location of $\pi \pm 2\pi$ and the cluster's frame located on the triangle's base.

assumption of a ''small'' cluster given the size of scalar field features).

Therefore, the acceptable tracking performance is bounded by a maximum allowable offset from being perfectly positioned on the ridge. For a symmetric ridge, this offset is $p = d/2$, or $\frac{1}{4}$ the lateral width of the cluster, as derived in Appendix C; furthermore, perfect straddling will result in a zero steady state offset of the center robot with respect to the ridgeline. These values are for the idealized and symmetric ridge assumptions. Although not detailed in this paper, relationships that accommodate noise and/or asymmetric ridge models can be developed as well.

C. SPEED CONSIDERATIONS

Because field characteristics can vary so dramatically, we have used velocity commands in our control primitives, with the gradient and/or differential computations only used to specify the direction of these velocities. Without this, the nature of the field would play the role of a highly variable gain, demanding faster responses when the cluster is positioned on steeper features. As a result, this strategy allows the control primitive to be appropriate over a broader range of spatial frequencies. More aggressive velocity policies are of interest in order to reduce navigation times; this will require consideration of robot dynamics as well as formal Lyapunov stability constraints.

Another speed-related issue arises when considering the ability to follow a ridge/trench that curves. Given the speed policies in Eq. 11 and 13, the cluster is limited in the radius of curvature it can follow to $r = v_x/\omega_z$, where v_x is the forward velocity of the cluster and ω_z is its allowable turnrate. As a result, it is possible for a cluster to lose track of a ridge/trench if its forward velocity is too fast for the features being followed; given the desire to follow these features with specific radii of curvature, the forward and turn velocities can be adjusted accordingly.

VI. FUTURE WORK

Multirobot adaptive navigation is a compelling capability, and the full suite of scalar field control primitives presented in this paper provides a foundation for a number of ongoing and future research initiatives. Acknowledging that the control primitives presented here constitute a simple, minimal, reactive approach to scalar field navigation, there is great potential for work that will refine these primitives. To begin, we are particularly interested in improved and extended performance through the incorporation of a) temporal filtering at several levels (robot-level sensor data, cluster-level field characteristic estimates, phased array filtering of sensors across the cluster, etc.), b) adaptive modification of cluster shape and size (for noise filtering, tuning the cluster to spatial frequencies of interest, etc.), c) more sophisticated controllers beyond the simple proportional and discrete strategies used here, d) additional robots and formation constructs, and so on.

From a verification perspective, we have started work to refine the primitives through more detailed simulation that includes cluster and robot level dynamics and through experimental demonstration using relevant land rover and marine surface vessel testbeds [27], [51]. We have also started to extend the primitives to navigation within three-dimensional scalar fields; this work is currently being verified in simulation and will be explored experimentally with our aerial and underwater multirobot testbeds [52], [53]. We are also exploring the use of an application-oriented state machine that sequences control primitives in order to methodically achieve certain tasks within a scalar field; initial scenarios that have been demonstrated in simulation include searching for a field's global maximum by sequentially trekking between adjoining local peaks, and moving between waypoints while ensuring that a minimum level of service is available (or maximum level of exposure is assured).

There is also work to be performed to compare navigation performance from a time and energy perspective to more conventional approaches, to develop stability guarantees, to address aliasing, etc. Other interesting extensions include the accommodation of turbulent scalar fields and the application of MAN approaches to vector fields. Finally, we are actively working with industry partners to implement our techniques in field grade multirobot systems in order to perform compelling, professional applications.

VII. SUMMARY AND CONCLUSIONS

We have presented and demonstrated via simulation new differential-based MAN control laws for scalar field navigation down ridges, up trenches, and to saddle points. These complement existing gradient-based control primitives for finding extreme points and following contours, which together establish a comprehensive suite of controllers that address all critical features in a scalar field. We have also described a multilayered control framework that unifies the execution of these control laws. This architecture uses a formation-keeping layer in order to maintain a spatial distribution of the robots appropriate for estimating the gradient or differential measurements used by the control laws. We also discuss several implementation considerations such as the impact of cluster size and velocity on MAN performance.

Our proposed controllers are admittedly simple, reactive, and minimal in size; even in their current form, however, they are capable of demonstrating a comprehensive suite of critical MAN behaviors. Our ongoing and future research activities are targeted to mature these controllers, to verify and validate them experimentally and in field applications, and to extend these MAN techniques to three dimensional fields and vector fields.

APPENDIX A
SCALAR FIELD EQUATION

The scalar field ($\mathbf{S}$) used for the simulations is formed by the summation of six feature fields: two maximums and a minimum ($\mathbf{M}_1$, $\mathbf{M}_2$, and $\mathbf{M}_3$), two ridges ($\mathbf{R}_1$ and $\mathbf{R}_2$) and

a trench ($\mathbf{T}_1$).

$$\mathbf{S} = \mathbf{M}_1 + \mathbf{M}_2 + \mathbf{M}_3 + \mathbf{R}_1 + \mathbf{R}_2 + \mathbf{T}_1 \qquad \text{(A-1)}$$

The equation for a maximum (hill) or minimum (valley) is given by:

$$\mathbf{M} = \frac{m_{\text{height}}}{m_{rolloff}\left((x - x_m)^2 + (y - y_m)^2\right) + 1} \qquad \text{(A-2)}$$

For these equations m_{height} is the height of the maximum or minimum, $m_{rolloff}$ is a coefficient that controls how quickly the feature rolls off (and thus how wide it is), and x_m and y_m provide the location of the extremum. The three maximum/minimum components for the simulated scalar field used the values shown in Table 1.

TABLE 1. Maximum/ minimum feature parameters used in simulation.

Feature	Feature Description	Variables			
		m_{height}	$m_{rolloff}$	x_m	y_m
$\mathbf{M}_1$	Global Max	50	0.0001	150	150
$\mathbf{M}_2$	Local Max	20	0.0001	0	150
$\mathbf{M}_3$	Global Min	-25	0.0001	-100	-150

The linear ridge ($\mathbf{R}_1$) descending from the global maximum is given by the equation:

$$\mathbf{R}_1 = \frac{r_{1\,\text{height}}}{\left(\left(r_{1\,end\,rolloff}\,dy\right)^4 + 1\right)\left(r_{1\,rolloff}\left(r_{1\,rolloff\,x}dx + dy\right)^2 + 1\right)} \qquad \text{(A-3)}$$

where $dy = y - y_{r1}$ and $dx = x - x_{r1}$. For these equations $r_{1\,\text{height}}$ is the height of the ridge feature, $r_{1\,end\,rolloff}$ is a coefficient that determines the roll off at the end of the ridge feature (and thus its length), $r_{1\,rolloff}$ specifies how quickly the sides of the ridge roll off, $r_{1\,rolloffx}$ modifies how the steep the ridge falls off in the x direction, x_{r1} is the x value of the center of the feature and y_{r1} is the y value of the center of the feature. The values used for the simulated scalar field are provided in Table 2.

TABLE 2. Ridge feature parameters used in simulation.

Variable	Value
$r_{1\,\text{height}}$	30
$r_{1\,end\,rolloff}$	0.007
$r_{1\,rolloff}$	0.0006
$r_{1\,rolloff\,x}$	-25
x_{r1}	150
y_{r1}	-75

The curved ridge ($\mathbf{R}_2$) descending from the local maximum is given by the equation:

$$\mathbf{R}_1 = \frac{r_{2\,\text{height}}}{\left(\left(r_{2\,end\,rolloff}\left(dx^2 + dy^2\right)^{0.5}\right)^4 + 1\right)\left(r_{2\,rolloff}\left(\left(dxc^2 + dyc^2\right)^{0.5} - r_{r_2}\right)^2 + 1\right)} \qquad \text{(A-4)}$$

where $dy = y - y_{r2}$, $dx = x - x_{r2}$, $dyc = y - y_{r2c}$, $dxc = x - x_{r2c}$, $r_{2\,\text{height}}$ is the height of the ridge feature, $r_{2\,end\,rolloff}$ is a coefficient that controls how quickly the end of the ridge rolls off (and thus how long the ridge is), y_{r2} is the y coordinate for the center of the ridge, x_{r2} is the x coordinate for the center of ridge 2, $r_{2\,rolloff}$ is a coefficient controlling how quickly the sides of ridge two roll off, y_{r2c} specifies the y coordinate of the center of the circle defining ridge 2, x_{r2c} specifies the x coordinate of the center of the circle defining ridge 2, and r_{r2} is the radius of the circle defining the curved ridge. The values used for the simulated scalar field are provided in Table 3.

TABLE 3. Curved ridge feature parameters used in simulation'.

Variable	Value
$r_{2\,\text{height}}$	25
$r_{2\,end\,rolloff}$	0.007
$r_{2\,rolloff}$	0.005
x_{r2}	-50
y_{r2}	-75
x_{r2c}	-200
y_{r2c}	250
r_{r2}	200

The trench equation is given by:

$$\mathbf{T} = \frac{t_{\text{height}}}{\left(\left(t_{end\,rolloff}\,dx\right)^4 + 1\right)\left(\left(t_{rolloff}\,d_t\right)^2 + 1\right)} \qquad \text{(A-5)}$$

where $dx = x - x_t$ and d_t is the distance from the trench line given by:

$$d_t = \frac{\left|t_{dy}x - t_{dx}y + t_{x2}t_{y1} - t_{y2}t_{x1}\right|}{\left(t_{dy}^2 + t_{dx}^2\right)^{.5}}. \qquad \text{(A-6)}$$

The parameters t_{dx} and t_{dy} are the x and y displacements between the two points (t_{x1}, t_{y1}) and (t_{x2}, t_{y2}) specifying the line along which the ridge is formed:

$$t_{dx} = t_{x2} - t_{x1} \qquad \text{(A-7)}$$

$$t_{dy} = t_{y2} - t_{y1} \qquad \text{(A-8)}$$

The values used for the aforementioned variables are shown in Table 4.

TABLE 4. Trench feature parameters used in simulation.

Variable	Value
t_{height}	-15
$t_{end\,rolloff}$	0.01
$t_{rolloff}$	0.03
x_t	0
t_{x1}	-100
t_{y1}	-150
t_{x2}	75
t_{y2}	150

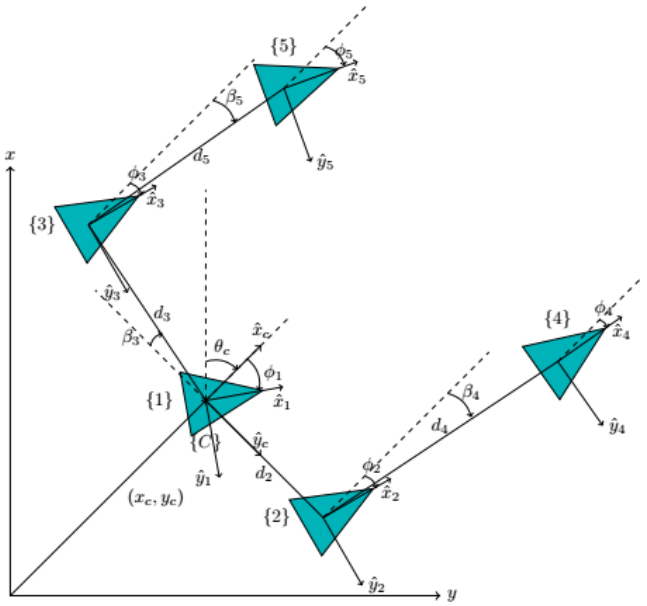


FIGURE 17. Five Robot Cluster Pose Definition: Cluster space pose variables for a five robot cluster with the cluster frame assigned to the rear-center robot and the position of the other robots defined serially within two different chains of the virtual mechanism.

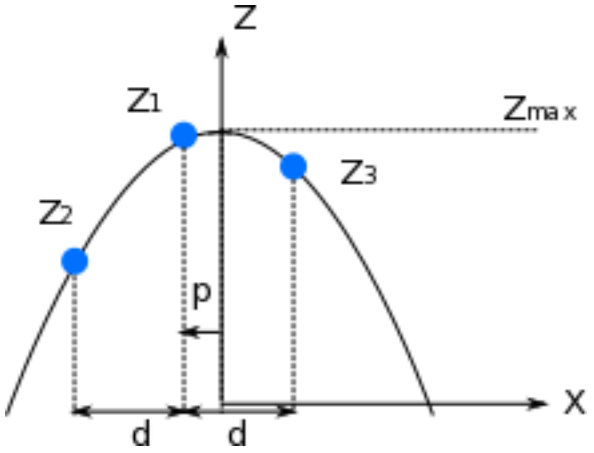


FIGURE 18. Ridge Following Lateral Offset for a Symmetric Ridge: When z_1 is the maximum scalar reading for Robots 1-3 we know that the cluster straddles the ridge. Given this, the lateral offset of Robot 1 from the top of the ridge is p, and p is bounded by the quantity d/2.

APPENDIX B
CLUSTER SPACE DEFINITION OF THE FIVE ROBOT CLUSTER

In Section IV, the five-robot cluster in a rectangular configuration is used, as shown in Figure 17.[3]

The cluster parameters used to demonstrate the ridge/trench following and saddle point positioning MAN primitives

$$\mathrm{d}_1 = \mathrm{d}_2 = \mathrm{d}_3 = \mathrm{d}_4 = 5\mathrm{m}$$
$$\beta_3 = \beta_4 = \beta_5 = 0^\circ$$

APPENDIX C
LATERAL OFFSET CONSTRAINT FOR RIDGE FOLLOWING

Section V discussed the lateral offset permitted when following a ridge, referring to the scenarios in Figure 16. Assuming a symmetric ridge described by the function g(x) and with a maximum value $z_{max} = g(0)$, the constraint for p, the maximum permitted lateral deviation of Robot 1 from the top of the ridge, may be found, given that we require $z_1 > z_3$.

$$z_1 > z_3 \tag{C-1}$$
$$g(-p) > g(d-p) \tag{C-2}$$
$$p < d - p \tag{C-3}$$
$$p < (d/2) \tag{C-4}$$

Eq. C-4 provides our constraint, which is that the allowable lateral offset in following the ridge is half the size of the distance d, as defined in Figure 18. As a result, this maximum allowable lateral offset is $\frac{1}{4}$ the lateral size of the overall five-robot cluster.

[3]Given our work with land, sea, air and space robots, we have adopted a standard aerospace frame for vehicles with $\hat{x}_i$ pointing out the front of each vehicle, $\hat{z}_i$ pointing down, and $\hat{y}_i$ oriented to complete a right hand Cartesian frame. Our global frame is oriented with $\hat{x}_G$ oriented North, $\hat{z}_G$ pointed down, and $\hat{y}_G$ pointed East such that when a vehicle is located at the origin with 0° of pitch/roll/yaw, the vehicle frame aligns with the global frame.

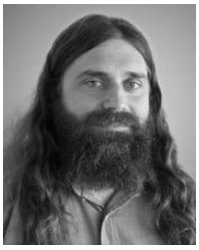
REFERENCES

[1] L. Xiao and H. K. Lo, "Adaptive vehicle navigation with en route stochastic traffic information," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 1900–1912, Oct. 2014.
[2] A. Melingui, T. Chettibi, R. Merzouki, and J. B. Mbede, "Adaptive navigation of an omni-drive autonomous mobile robot in unstructured dynamic environments," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2013, pp. 1924–1929.
[3] X. Chen, K. Watanabe, K. Kiguchi, and K. Izumi, "An ART-based fuzzy controller for the adaptive navigation of a quadruped robot," *IEEE/ASME Trans. Mechatronics*, vol. 7, no. 3, pp. 318–328, Sep. 2002.
[4] D. M. Helmick, A. Angelova, M. Livianu, and L. H. Matthies, "Terrain Adaptive Navigation for Mars Rovers," in *Proc. IEEE Aerosp. Conf.*, Big Sky, MT, USA, Mar. 2007, pp. 1–11.
[5] S. M. Amin, E. Y. Rodin, M. K. Meusey, T. W. Cusick, and A. Garcia-Ortiz, "Evasive adaptive navigation and control against multiple pursuers," in *Proc. Amer. Control Conf.*, vol. 3. Jun. 1997, pp. 1453–14573.
[6] A. Filippoupolitis, G. Gorbil, and E. Gelenbe, "Autonomous navigation systems for emergency management in buildings," in *Proc. IEEE GLOBECOM Workshops (GC Wkshps)*, Houston, TX, USA, Dec. 2011, pp. 1056–1061.
[7] H. Ishida, K. Suetsugu, T. Nakamoto, and T. Morizumi, "Study of autonomous mobile sensing system for localization of odor source using gas sensors and anemometric sensors," *Sens. Actuators A, Phys.*, vol. 45, no. 2, pp. 153–157, Nov. 1994.
[8] W. Li, J. A. Farrell, S. Pang, and R. M. Arrieta, "Moth-inspired chemical plume tracing on an autonomous underwater vehicle," *IEEE Trans. Robot.*, vol. 22, no. 2, pp. 292–307, Apr. 2006.
[9] A. S. Matveev, H. Teimoori, and A. V. Savkin, "Navigation of a unicycle-like mobile robot for environmental extremum seeking," *Automatica*, vol. 47, no. 1, pp. 85–92, Jan. 2011.
[10] E. Burian, E. Yoerger, A. Bradley, and H. Singh, "Gradient search with autonomous underwater vehicle using scalar measurements," in *Proc. Symp. Auto. Underwater Vehicle Technol.*, 1996, pp. 86–98.
[11] J. Adler, "Chemotaxis in bacteria," *Science*, vol. 153, no. 3737, pp. 708–716, Aug. 1966.
[12] C. Zhang, A. Siranosian, and M. Krstić, "Extremum seeking for moderately unstable systems and for autonomous vehicle target tracking without position measurements," *Automatica*, vol. 43, pp. 1832–1839, Oct. 2007.
[13] N. Atanasov, N. J. Le, N. Michael, and G. J. Pappas, "Stochastic source seeking in complex environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Saint Paul, MN, USA, May 2012, pp. 3013–3018.

[14] A. S. Matveev, M. C. Hoy, A. M. Anisimov, and A. V. Savkin, "Tracking of deforming environmental level sets of dynamic fields by a nonholonomic robot without gradient estimation," in *Proc. 10th IEEE Int. Conf. Control Autom. (ICCA)*, Jun. 2013, pp. 1754–1759.
[15] A. S. Matveev, M. C. Hoy, K. Ovchinnikov, A. Anisimov, and A. V. Savkin, "Robot navigation for monitoring unsteady environmental boundaries without field gradient estimation," *Automatica*, vol. 62, pp. 227–235, Dec. 2015.
[16] P. Ögren, E. Fiorelli, and N. E. 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.
[17] X. Kang and W. Li, "Moth-inspired plume tracing via multiple autonomous vehicles under formation control," *Adapt. Behav.*, vol. 20, no. 2, pp. 131–142, Apr. 2012.
[18] V. Gazi, F. Fidan, L. Marques, and R. Ordonez, "Robot swarms: Dynamics and control," in *Mobile Robots for Dynamic Environments*, E. F. Kececi and M. Ceccarelli, Eds. New York, NY, USA: ASME, 2015, ch. 4, pp. 79–107.
[19] W. Wu, I. D. Couzin, and F. Zhang, "Bio-inspired source seeking with no explicit gradient estimation," in *Proc. 3rd IFAC Workshop Distrib. Estimation Control Netw. Syst.*, Santa Barbara, CA, USA, Sep. 2012, pp. 240–245.
[20] E. Biyik and M. Arcak, "Gradient climbing in formation via extremum seeking and passivity-based coordination rules," in *Proc. 46th IEEE Conf. Decision Control*, Dec. 2007, pp. 3133–3138.
[21] 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.
[22] N. A. Atanasov, N. J. Le, and G. J. Pappas, "Distributed algorithms for stochastic source seeking with mobile robot networks," *J. Dyn. Syst., Meas., Control*, vol. 173, no. 3, pp. 1–9, 2015.
[23] R. Bachmayer and N. E. Leonard, "Vehicle networks for gradient descent in a sampled environment," in *Proc. 41st IEEE Conf. Decision Control*, vol. 1. Dec. 2002, pp. 112–117, doi: 10.1109/CDC.2002.1184477.
[24] E. Fiorelli, P. Bhatta, N. Leonard, and I. Shulman, "Adaptive sampling using feedback control of an autonomous underwater gliderfleet," in *Proc. 13th Int. Symp. Unmanned Untethered Submersible Technol.*, 2003, pp. 1–16.
[25] E. Fiorelli, N. E. Leonard, P. Bhatta, D. A. Paley, R. Bachmayer, and D. M. Fratantoni, "Multi-AUV Control and Adaptive Sampling in Monterey Bay," *IEEE J. Ocean. Eng.*, vol. 31, no. 4, pp. 935–948, Oct. 2006.
[26] 26 Formatted S. Li, R. Kong, and Y. Guo, "Cooperative distributed source seeking by multiple robots: Algorithms and experiments," *IEEE/ASME Trans. Mechatron.*, vol. 19, no. 6, pp. 1810–1820, Dec. 2014.
[27] J. Acain, C. Kitts, T. Adamek, K. Ebadi, and M. Rasay, "A multi-robot testbed for adaptive sampling experimentation via radio frequency fields," in *Proc. ASME Int. Design Eng. Tech. Conf.*, Boston, MA, USA, Aug. 2015, pp. 1–8.
[28] Z. Jin and A. L. Bertozzi, "Environmental boundary tracking and estimation using multiple autonomous vehicles," in *Proc. 46th IEEE Conf. Decision Control*, Dec. 2007, pp. 4918–4923.
[29] S. Srinivasan, K. Ramamritham, and P. Kulkarni, "ACE in the hole: Adaptive contour estimation using collaborating mobile sensors,"in *Proc. Int. Conf. Inf. Process. Sensor Netw.*, Apr. 2008, pp. 147–158.
[30] A. Joshi, T. Ashley, Y. R. Huang, and A. L. Bertozzi, "Experimental validation of cooperative environmental boundary tracking with on-board sensors," in *Proc. Amer. Control Conf.*, 2009, pp. 2630–2635.
[31] J. Clark and R. Fierro, "Cooperative hybrid control of robotic sensors for perimeter detection and tracking," in *Proc. Amer. Control Conf.*, 2005, pp. 3500–3505.
[32] D. Marthaler and A. L. Bertozzi, "Tracking environmental level sets with autonomous vehicles," in *Recent Developments in Cooperative Control and Optimization*, S. Butenko, R. Murphey, and P. M. Pardalos, Eds. Norwell, MA, USA: Kluwer, 2003.
[33] A. L. Bertozzi, M. Kemp, and D. Marthaler, "Determining environmental boundaries: Asynchronous communication and physical scales," in *Cooperative Control* (Lecture Notes in Control and Information Science), vol. 309, V. Kumar, N. Leonard, and A. S. Morse, Eds. Berlin, Germany: Springer, 2004.
[34] I. Triandaf and I. B. Schwartz, "A collective motion algorithm for tracking time-dependent boundaries," *Math. Comput. Simul.*, vol. 70, no. 4, pp. 187–202, 2005.
[35] F. Zhang and N. E. Leonard, "Cooperative filters and control for cooperative exploration," *IEEE Trans. Autom. Control*, vol. 55, no. 3, pp. 650–663, Mar. 2010.
[36] W. Wu and F. Zhang, "Cooperative exploration of level surfaces of three dimensional scalar fields," *Automatica*, vol. 47, no. 9, pp. 2044–2051, 2011.
[37] T. Adamek and C. M. I. Kitts, "Gradient-based cluster space navigation for autonomous surface vessels," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 2, pp. 506–518, Apr. 2015.
[38] O. Khatib, "The potential field approach and operational space formulation in robot control," in *Adaptive and Learning Systems*, K. S. Narendra, Ed. Boston, MA, USA: Springer, 1986.
[39] K. Konolige, "A gradient method for realtime robot control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Takamatsu, Japan, vol. 1. Jun. 2000, pp. 639–646.
[40] H. M. Choset, *Principles of Robot Motion: Theory, Algorithms, and Implementation*. Cambridge, MA, USA: MIT Press, 2005.
[41] R. C. Arkin, "Motor schema-based mobile robot navigation," *Int. J. Robot. Res.*, vol. 8, no. 4, pp. 92–112, 1989.
[42] C. Kitts, K. Stanhouse, and P. Chindaphorn, "Cluster space collision avoidance for mobile two-robot systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, St. Louis, MO, USA, Oct. 2009, pp. 1941–1948.
[43] I. Mas and C. Kitts, "Obstacle avoidance policies for cluster space control of nonholonomicmultirobot systems," *IEEE/ASME Trans. Mechatronics*, vol. 17, no. 6, pp. 1068–1079, Dec. 2012.
[44] N. E. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *Proc. 40th IEEE Conf. Decision Control*, vol. 3. Dec. 2001, pp. 2968–2973.
[45] H. M. La, W. Sheng, and J. Chen, "Cooperative and active sensing in mobile sensor networks for scalar field mapping," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 1, pp. 1–12, Jan. 2015.
[46] R. Cui, Y. Li, and W. Yan, "Mutual information-based multi-AUV path planning for scalar field sampling using multidimensional RRT," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 7, pp. 993–1004, Jul. 2016.
[47] B. Zhang and G. S. Sukhatme, "Adaptive sampling for estimating a scalar field using a robotic boat and a sensor network," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, Roma, Italy, Apr. 2007, pp. 3673–3680.
[48] C. 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.
[49] I. Mas and C. Kitts, "Dynamic control of mobile multi-robot systems: The cluster space formulation," *IEEE Access*, vol. 2, pp. 558–570, 2014.
[50] C. Kitts *et al.*, "Field operation of a robotic small waterplane area twin hull boat for shallow?water bathymetric characterization," *J. Field Robot.*, vol. 29, no. 6, pp. 924–938, 2012.
[51] P. Mahacek, I. P. O. Mas, J. Acain, and C. Kitts, "Cluster space control of autonomous surface vessels," *Marine Tech. Soc. J.*, vol. 1, pp. 13–20, Sep. 2009.
[52] J. Cashbaugh, A. Mahacek, C. Kitts, C. Zempel, and A. Sherban, "Quadrotor testbed development and preliminary results," in *Proc. IEEE Aerosp. Conf.*, Mar. 2015, pp. 1–12.
[53] C. Kitts *et al.*, "An underwater robotic testbed for multi-vehicle control," in *Proc. IEEE/OES Auto. Underwater Vehicles*, Oct. 2014, pp. 1–8.

CHRISTOPHER A. KITTS received the B.S.E. degree in mechanical and aerospace engineering from Princeton University, the M.P.A degree in international and defense policy from the University of Colorado, and the M.S. degree in aeronautics and astronautics and the Ph.D. degree in mechanical engineering from Stanford University. His industrial background includes service as a Satellite Constellation Mission Controller with the U.S. Air Force and work as a Computer Scientist while he was a Contractor with the NASA Ames Research Center's Computational Sciences Division. He is currently an Associate Professor of mechanical engineering, the Director of the Robotic Systems Laboratory, and the Associate Dean of the Research and Faculty Development for the School of Engineering, Santa Clara University. He is a Fellow of the American Society of Mechanical Engineers.

ROBERT T. MCDONALD received the B.S. degree and the M.S. degree in mechanical engineering from Santa Clara University, Santa Clara, CA, USA, in 2015 and 2017, respectively, where he is currently pursuing the Ph.D. degree in mechanical engineering. He was a Teaching Assistant with the Department of Mechanical Engineering, Santa Clara University, and a Research Assistant with the Santa Clara's Robotic Systems Laboratory.



MICHAEL A. NEUMANN received the B.S. degree in mechanical engineering, the M.S. degree in mechanical engineering and applied mathematics, and the Ph.D. degree in mechanical engineering from Santa Clara University. He was a Teaching Assistant and a Research Assistant with the Robotic Systems Laboratory. He has also served in the Peace Corps for three years.