ANNUAL REVIEWS

Annual Review of Environment and Resources

Multiple UAVs for Mapping: A Review of Basic Modeling, Simulation, and Applications

Tarek I. Zohdi

Department of Mechanical Engineering, University of California, Berkeley, California 94720-1740, USA; email: zohdi@berkeley.edu

Annu. Rev. Environ. Resour. 2018. 43:523–43

The *Annual Review of Environment and Resources* is online at environ.annualreviews.org

https://doi.org/10.1146/annurev-environ-102017-025912

Keywords

UAVs, mapping

Abstract

The goal of this article is to provide an introduction to basic modeling and simulation techniques for multiple interacting unmanned aerial vehicles (UAVs), called swarms, for applications in mapping. The target audience is senior students and young scientists. This review will serve to inform, orient, and direct someone already educated in environmental science but unaware of multiple-UAV interaction models.

Contents

1. INTRODUCTION

Research on unmanned aerial vehicles (UAVs) started in the early 1900s. Initially, most research was geared toward military applications. This research accelerated moderately during World War II with the aim of training antiaircraft gunners and flying attack missions. However, with the exception of the V-2 (Vergeltungswaffe 2) rocket system program in Germany, they were primarily "toy" airplanes. It was not until the 1960s, during the Cold War, when the United States was involved in a variety of military conflicts and the US Air Force was concerned about losing pilots over hostile territory, that UAV research started to grow rapidly. In 2012, the US Air Force was operating approximately 7,500 UAVs. As of 2017, nearly every industrialized country has a growing manufacturing base for UAVs.

Due to the dramatic increase in inexpensive UAV and camera technology, there are a wide variety of nonmilitary applications, such as worldwide antipoaching and antiwhaling efforts. UAVs have also been used for geophysical mapping (for example in oil and gas exploration), in particular for geomagnetic surveys, where measurements of the Earth's varying magnetic field strength are used to calculate the nature of the underlying magnetic rock structure in order to locate mineral deposits. Because of the huge spatial expanse associated with oil and gas pipelines, monitoring activity can be enhanced and accelerated by the deployment of UAVs. In the field of archaeology, UAVs are used to accelerate surveying to protect sites from looters. Another obvious application is cargo transport, which has been promoted by Amazon, DHL, Google, and other companies. The utility of UAVs in agriculture (e.g., for crop dusting and crop health monitoring) is also evident.

Overarching all of these applications is the real-time mapping of large areas, such as those struck by a multilocation disaster, for example an earthquake, fire, or tsunami, by multiple UAVs (which we refer to as swarms). Because of the complex, multifaceted infrastructure (roads, bridges, pipelines, power grids, and water) that needs to be mapped after a disaster, there exists a need for different mapping strategies (**Figure 1**). Such sectors need to be mapped with different technologies (e.g., infrared, radio, optical, and microwave frequencies). Small UAVs are usually battery powered (**Figure 2**), and thus they have limited range and their paths must be planned carefully to conserve power. Specifically, the objective of this work is to provide an introduction to basic modeling and simulation techniques for multiple interacting UAVs for a target audience of young scientists. Simultaneous advances in inexpensive UAVs, computational modeling techniques, and

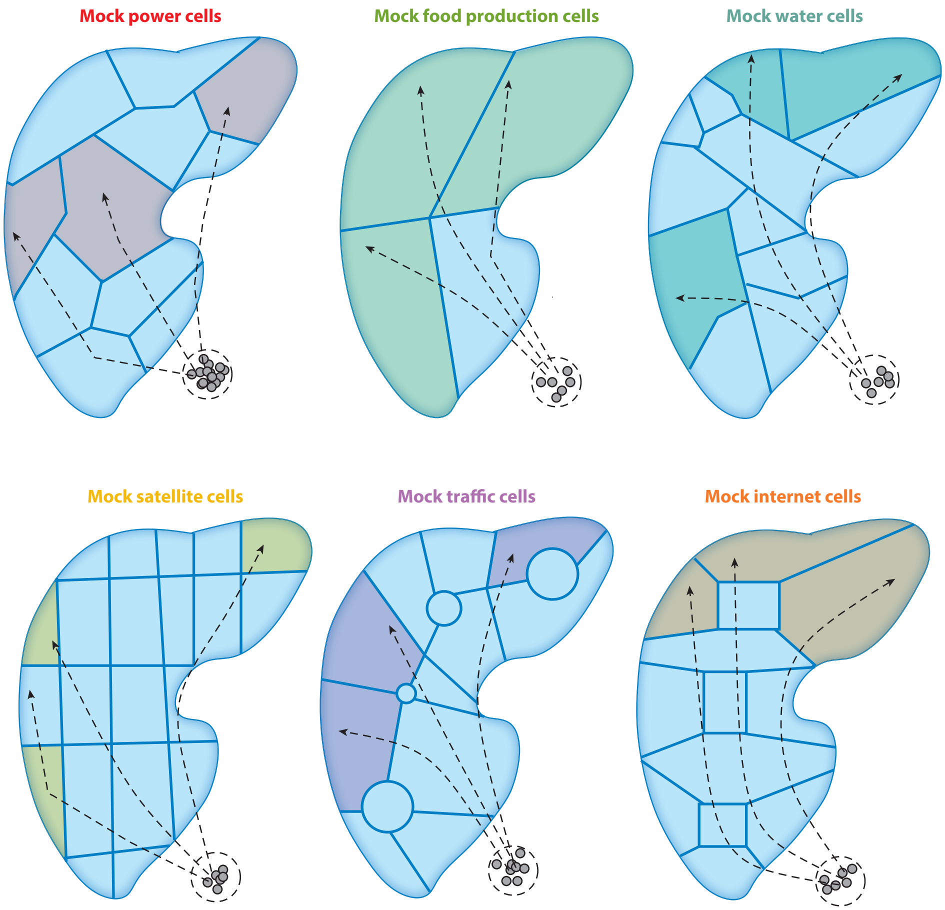

Figure 1
Different coexisting infrastructures that require different mapping strategies and path planning for UAVs.

camera and sensor technologies have made rapid postdisaster mapping a possibility. Agent-based paradigms for simulation of coupled complex systems have become powerful predictive tools. Because different infrastructures have different grids and different quantities to be mapped, the optimal path for a set of released swarms will vary over the same terrain. It is relatively easy to develop agent-based models for a team of swarm members (UAVs) with the intent of mapping large areas with various optimality conditions: minimum time, minimum energy usage, optical sensing, infrared sensing, acoustic sensing, water spillage sensing, etc.

It is important to note that new Federal Aviation Administration (FAA) regulations require eligible owners to register their UAVs prior to flight. For owners less than 13 years of age, a parent or other responsible person must file an FAA registration form, and the UAVs must have FAA-issued registration numbers. In June 2016, the FAA announced regulations for commercial operation of small UAVs, those between 0.55 and 55 pounds (about 250 g to 25 kg), including payload, that require the on-site presence of a licensed remote pilot in command (above 16 years of age). Unfortunately, the proliferation of these small flying vehicles and the resulting concerns regarding invasion of privacy have led to many cases of individuals attempting to shoot down UAVs that enter into the airspace above their property. This is technically illegal according to

Figure 2

Various models of quadcopters that appear in the popular press.

FAA regulations, primarily because the scattered debris can harm individuals and property below. It is estimated that UAVs were shot down in the United States at a rate of one UAV per month in 2016. The legality of shooting down a UAV is complex and is determined in part by the height at which the UAV is flying at the time of shooting (for news coverage, see 1). To the knowledge of the author, no works have explored the dynamical response and possible breakup of a UAV being shot at, for example, by a shotgun. Zohdi (2) used a discrete element method (DEM) to formulate the dynamical response of a quadcopter to a series of random external impulses, such as from shotgun pellets. The use of DEM allows for the fragmentation of the quadcopter and also allows one to compute the trajectories and distribution of the debris field. Various initial conditions can be tested, which can potentially aid in settling disputes after a UAV has already been shot down, for example by determining the height and location of the UAV prior to it being shot down on the basis of the debris field. This is outside the scope of the current presentation.

2. GENERAL MODELS FOR MULTIPLE UAVs

In this article, we concentrate on decentralized swarm control paradigms. It has long been recognized that interactive cooperative behavior within biological groups or swarms is advantageous in avoiding predators or, vice versa, in capturing prey. For example, one of the primary advantages of a swarm-like decentralized decision-making structure is that there is no leader, and thus the vulnerability of the swarm is substantially reduced. Furthermore, the decision making is relatively simple and rapid for each individual; however, the aggregate behavior of the swarm can be quite sophisticated.

The modeling of swarm-like behavior has biological research origins, dating back at least to the work of Breder (3) in 1952. It is commonly accepted that a central characteristic of swarm-like behavior is the trade-off between long-range interaction and short-range repulsion between individuals. Models describing clouds or swarms of particles, where their interaction is constructed from attractive and repulsive forces, dependent on the relative distance between individuals, are commonplace (for reviews, see References 4–6). The field is large and encompasses a wide variety of applications, for example, the behavior of flocks of birds, schools of fish, flows of traffic, and crowds of human beings. Loosely speaking, swarm analyses are concerned with the complex aggregate behavior of groups of simple members, which frequently are treated as particles (for example, see Reference 7).

A central objective of this article is to provide basic mechanistic models and numerical solution strategies for the direct simulation of the motion of swarms that can be achieved using relatively standard computing equipment. The usual approach to modeling such systems is to use a combination of short-range and long-range interaction forces (4–9). Early approaches that rely on decentralized organization can be found in the work of Beni (10), Brooks (11), Dudek et al. (12), Cao et al. (13), and Liu & Passino (14). However, there are alternative, rule-driven swarms where the interaction is governed not by forces but by proximal instructions, such as: (*a*) "If a fellow swarm agent gets close to me, attempt to retreat as far as possible," (*b*) "follow the leader," or (*c*) "stay in clusters." For example, ant colonies (15) exhibit a foraging-type behavior, aided by a trail-laying and trail-following mechanism, for finding food sources. They deposit a chemical substance, called a pheromone, which decays over time. The fellow swarm agents detect paths with a high pheromone concentration (where the food source is highly concentrated) and follow them (6, 15–18). For certain swarms, the visual field of the individual agents may play a significant role, while in others, this is a nonissue, for example if the agents are robots or UAVs for which the communication is electronic. In some systems, agents interact with a specific set of swarm agents regardless of whether they are far away (19). For example, Ballerini et al. (20) concluded, on the basis of careful observations, that a starling (*Sturnus vulgaris*) communicates with a certain number of starlings surrounding it, regardless of the distance, attributing this to a perceptual limit in the number of objects that starlings can track. In the references, an extensive list of works have been included. While these rule-driven paradigms are usually easy to construct, they are difficult to analyze mathematically. It is primarily for this reason that a mechanical approach is adopted in this article.

3. BASIC MODELING OF POINT-MASS SWARMS

Throughout our analysis, the objects are assumed to be small enough to be considered (idealized) as point masses, and the effect of their rotation with respect to their center of mass is considered unimportant to their overall motion.

3.1. Notation

Here, boldface symbols represent vectors or tensors. A fixed Cartesian coordinate system is used throughout. The unit vectors for such a system are the mutually orthogonal triad $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$. We denote the position of a point (swarm) in space by the vector $\mathbf{r}$. In fixed Cartesian coordinates, we have

$$\mathbf{r} = r_1\mathbf{e}_1 + r_2\mathbf{e}_2 + r_3\mathbf{e}_3; \qquad 1.$$

for velocity,

$$\mathbf{v} = \dot{\mathbf{r}} = \dot{r}_1\mathbf{e}_1 + \dot{r}_2\mathbf{e}_2 + \dot{r}_3\mathbf{e}_3; \qquad 2.$$

for acceleration,

$$\mathbf{a} = \ddot{\mathbf{r}} = \ddot{r}_1\mathbf{e}_1 + \ddot{r}_2\mathbf{e}_2 + \ddot{r}_3\mathbf{e}_3. \qquad 3.$$

3.2. Construction of a Swarm

In our analysis, we treat the swarm members as point masses; i.e., we ignore their dimensions. For each swarm member (N_s in total), the equations of motion are

$$m_i\dot{\mathbf{v}}_i = m_i\ddot{\mathbf{r}}_i = \mathbf{\Psi}_i^{\mathrm{mt}}(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_{N_\mathrm{s}}), \qquad 4.$$

where $\mathbf{\Psi}^{\mathrm{mt}}$ represents the forces of interaction between swarm member i and targets to be mapped and other swarm members.

3.3. A Model Problem: Mapping of a Region

An algorithm is as follows.

3.3.1. Algorithm.

1. Initialize the locations of the targets to be mapped: $\mathbf{T}_i = (T_x, T_y, T_z)_i$ for $i = 1, 2, \ldots, N_\mathrm{T}$, where N_T is the number of targets.
2. Initialize the locations of the swarm members (UAVs): $\mathbf{r}_i = (r_x, r_y, r_z)_i$ for $i = 1, 2, \ldots, N_\mathrm{s}$.
3. For each swarm member i, determine the distance to each target j,

$$\|\mathbf{r}_i - \mathbf{T}_j\| = \left[(r_{ix} - T_{jx})^2 + (r_{iy} - T_{jy})^2 + (r_{iz} - T_{jz})^2\right]^{1/2}, \qquad 5.$$

   and the direction to each target,

$$\mathbf{n}_{ij} = \frac{\mathbf{T}_j - \mathbf{r}_i}{\|\mathbf{r}_i - \mathbf{T}_j\|}. \qquad 6.$$

4. For each swarm member i, determine the force of interaction as a function of the distances:

$$\mathbf{\Psi}_{ij}^{mt} = \mathcal{F}(\|\mathbf{r}_i - \mathbf{T}_j\|, \mathbf{n}_{ij}). \qquad 7.$$

   For example, we compute a weighted direction to each target,

$$\mathbf{N}_i = \sum_{j=1}^{N_T} w_j \mathbf{n}_{ij} e^{-a\|\mathbf{r}_i - \mathbf{T}_j\|}, \qquad 8.$$

   where w_j is a weight reflecting the importance of that target and a is a decay parameter, which is normalized to give an overall direction in which to move:

$$\mathbf{n}_i^* = \frac{\mathbf{N}_i}{\|\mathbf{N}_i\|}. \qquad 9.$$

The forces are then constructed by multiplying the available thrust, F_i, by the overall normalized direction:

$$\mathbf{\Psi}_i^{mt} = F_i \mathbf{n}_i^*. \tag{10.}$$

5. Integrate the equations of motion

$$m_i \dot{\mathbf{v}}_i = \mathbf{\Psi}_i^{mt}, \tag{11.}$$

yielding

$$\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i(t) + \frac{\Delta t}{m_i} \mathbf{\Psi}_i^{mt}(t) \tag{12.}$$

and

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + \Delta t \mathbf{v}_i(t). \tag{13.}$$

Note that if

$$\|\mathbf{v}_i(t + \Delta t)\| > v_{\max}, \tag{14.}$$

then we define $\mathbf{v}_i^{\text{old}}(t + \Delta t) = \mathbf{v}_i(t + \Delta t)$, and the velocity is rescaled:

$$\mathbf{v}_i^{\text{new}}(t + \Delta t) = \mathbf{v}_i(t + \Delta t) \frac{v_{\max}}{\|\mathbf{v}_i^{\text{old}}(t + \Delta t)\|}, \tag{15.}$$

with $\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i^{\text{new}}(t + \Delta t)$.

6. Determine whether any targets have been mapped by checking the distance between swarm members and targets:

$$||\mathbf{r}_i - \mathbf{T}_j|| \leq \text{tolerance}. \tag{16.}$$

For any $\mathbf{T}_j$, if any swarm member has satisfied this criterion, take $\mathbf{T}_j$ out of the system for the next time step so that no swarm member wastes resources by attempting to map $\mathbf{T}_j$.

7. The process is then repeated for the next time step.

3.3.2. Numerical example. We consider the following parameters:

- weights $w_i = 1$,
- decay parameter $a = 0.01$,
- ratio of thrust to mass $F_i/m_i = 10^5$ N/kg,
- 100 swarm members,
- 100 targets,
- $T = 50$ s,
- $\Delta t = 0.001$ s,
- initial swarm velocity $\mathbf{v}_i(t = 0) = \mathbf{0}$ m/s,
- initial swarm domain (10 m, 10 m, 10 m),
- domain to be mapped (500 m, 500 m, 10 m), and
- maximum velocity of a swarm member $v_{\max} = 100$ m/s.

The results are shown in **Figures 4** and **5**. As can be seen from the progression of unmapped to mapped targets, the algorithm is quite adept in picking up missed targets by successive sweeps.

The model presented here is relatively simple and can be implemented in any computing environment. However, to really capture swarm-like behavior, more sophisticated models are constructed, in particular addressing the interaction between swarm members, obstacles, and targets. This is done next.

4. GENERALIZATIONS: CONSTRUCTION OF AN INTERACTING SWARM

In the following analysis, we treat the swarm members as point masses; i.e., we ignore their dimensions. (The swarm member centers, which are initially nonintersecting, also cannot intersect later due to repulsion terms.) For each swarm member (N_s in total), the equations of motion are

$$m_i \ddot{\mathbf{r}}_i = \boldsymbol{\Psi}_i(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_{N_s}), \tag{17}$$

where $\boldsymbol{\Psi}$ represents the forces of interaction between swarm member i and the target, obstacles, and other swarm members. We consider the following decomposition of interaction forces,

$$\boldsymbol{\Psi}_i = \boldsymbol{\Psi}_i^{\mathrm{mm}} + \boldsymbol{\Psi}_i^{\mathrm{mt}} + \boldsymbol{\Psi}_i^{\mathrm{mo}}, \tag{18}$$

where between swarm members (member–member) we have

$$\boldsymbol{\Psi}_i^{\mathrm{mm}} = \sum_{j \neq i} N_s \left[\left(\underbrace{\alpha_1^{\mathrm{mm}} \|\mathbf{r}_i - \mathbf{r}_j\|^{\beta_1^{\mathrm{mm}}}}_{\text{attraction}} - \underbrace{\alpha_2^{\mathrm{mm}} \|\mathbf{r}_i - \mathbf{r}_j\|^{-\beta_2^{\mathrm{mm}}}}_{\text{repulsion}} \right) \underbrace{\frac{\mathbf{r}_j - \mathbf{r}_i}{\|\mathbf{r}_i - \mathbf{r}_j\|}}_{\mathbf{n}_{ij} \stackrel{\mathrm{def}}{=} \text{unit vector}} \right]; \tag{19}$$

here $\|\cdot\|$ represents the Euclidean norm in R^3, and the normalized direction is determined by the difference in the position vectors of the particles' centers:

$$\mathbf{n}_{ij} \stackrel{\mathrm{def}}{=} \frac{\mathbf{r}_j - \mathbf{r}_i}{\|\mathbf{r}_i - \mathbf{r}_j\|}. \tag{20}$$

Between the swarm members and the target we have (member–target)

$$\boldsymbol{\Psi}_i^{\mathrm{mt}} = \left(\alpha^{\mathrm{mt}} \|\mathbf{r}_i - \mathbf{T}\|^{\beta^{\mathrm{mt}}} \right) \frac{\mathbf{T} - \mathbf{r}_i}{\|\mathbf{r}_i - \mathbf{T}\|}, \tag{21}$$

and for the repulsion between swarm members and obstacles (member–obstacle) we have

$$\boldsymbol{\Psi}^{\mathrm{mo}} = -\sum_{j=1}^{q} \left[\left(\alpha^{\mathrm{mo}} \|\mathbf{r}_i - \mathbf{O}_j\|^{-\beta^{\mathrm{mo}}} \right) \frac{\mathbf{O}_j - \mathbf{r}_i}{\|\mathbf{r}_i - \mathbf{O}_j\|} \right], \tag{22}$$

where q is the number of obstacles and where all of the design parameters, represented in each equation by α and β, are nonnegative.

Note that the specific configuration of interparticle forces chosen here is only one of many possibilities to model the interaction. A variety of alternative forms are available from the field of molecular dynamics, which is typically concerned with the calculation of thermochemical and thermomechanical properties of gases, liquids, and solids by using models of systems of atoms or molecules. In such models, each atom or molecule is represented by a material point and is treated as a point mass whose motion is described by Newton's second law, with the forces computed from a prescribed potential energy function, $V(\mathbf{r})$, $m\ddot{\mathbf{r}} = -\nabla V(\mathbf{r})$ (for example, see Reference 21).

Note also that the surrounding environment (for example, a fluid such as air) is a source of damping for the system. The simplest model is of the form (for swarm member i)

$$\boldsymbol{\Psi}_i^{\mathrm{env}} = -c^{\mathrm{env}}(\mathbf{v}_i - \mathbf{v}^{\mathrm{env}}), \tag{23}$$

where $\mathbf{v}_i$ is the velocity of the i-th member and $\mathbf{v}^{\mathrm{env}}$ is the local velocity of the ambient medium. In summary, we have the following forces acting on each member of the swarm:

$$\boldsymbol{\Psi} = \boldsymbol{\Psi}^{\mathrm{mm}} + \boldsymbol{\Psi}^{\mathrm{mt}} + \boldsymbol{\Psi}^{\mathrm{mo}} + \boldsymbol{\Psi}^{\mathrm{env}}. \tag{24}$$

The problem of fully coupled (two-way) particle–fluid interaction is beyond the scope of this presentation. Generally, this requires the use of staggering-type schemes (22–25).

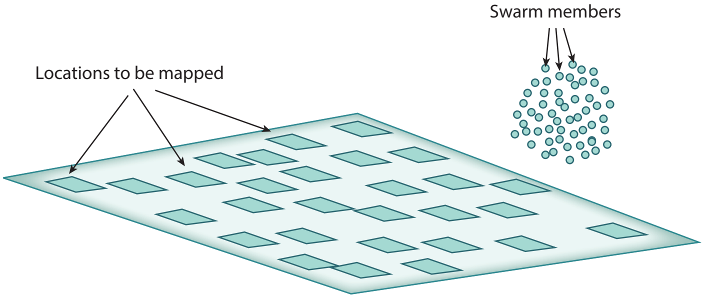

Figure 3

A typical setup for mapping of a region by a swarm.

5. EXAMPLES

To illustrate such models, we review the results of Zohdi (8).

5.1. Example: Chasing a Moving Target

As a representative of a class of model problems, we now consider a normalized performance function (normalized by the total simulation time and the initial separation distance) representing both the time it takes for the swarm members to get to the target and the distance of the swarm members from the target:

$$\Pi = \frac{\int_0^{\mathcal{T}} \sum_{i=1}^{N_s} \|\mathbf{r}_i - \mathbf{T}\| \, \mathrm{d}t}{\mathcal{T} \sum_{i=1}^{N_s} \|\mathbf{r}_i(t=0) - \mathbf{T}\|}, \qquad 25.$$

where the total simulation time is $\mathcal{T} = 30$ s and $\mathbf{T}$ is the position of the target (**Figure 3**). The components of the initial position vectors of the nonintersecting swarm members, each assigned a mass of 10 kg, were given random values of $-1 \leq r_{ix}, r_{iy}, r_{iz} \leq 1$. The location of the moving target was given by the following function:

$$\begin{aligned} T_x &= x_0 + a_1 \cos(a_2 t) + a_3 t, \\ T_y &= y_0 + b_1 \sin(b_2 t) + b_3 t, \\ T_z &= z_0 + c_1 \cos(c_2 t) + c_3 t, \end{aligned} \qquad 26.$$

where the parameters are listed in **Table 1**.

The location of the center of the (rectangular) obstacle array was (1.5, 0, 0). A 100-obstacle fence was set up in a 10 m×10 m array with a spacing of 0.2 m between obstacle centers. For illustrative purposes, 200 swarm members were used. The parameters selected were: $\alpha_1^{\mathrm{mm}} = 1$, $\alpha_2^{\mathrm{mm}} = 1$, $\alpha^{\mathrm{mt}} = 200$, $\alpha^{\mathrm{mo}} = 100$, $\beta_1^{\mathrm{mm}} = 2$, $\beta_2^{\mathrm{mm}} = 2$, $\beta^{\mathrm{mt}} = 2$, and $\beta^{\mathrm{mo}} = 2$. The environmental damping was set to $c^{\mathrm{env}} = 1$. Simulations were run (**Figure 6**), with the performance being $\Pi = 0.2712$.

Table 1 Table of parameters for moving target (see Equation 27)

(x_0, y_0, z_0)	a_1	a_2	a_3	b_1	b_2	b_3	c_1	c_2	c_3
(4, 0, 0)	1	1	0.5	1	1	0.5	1	1	0.5

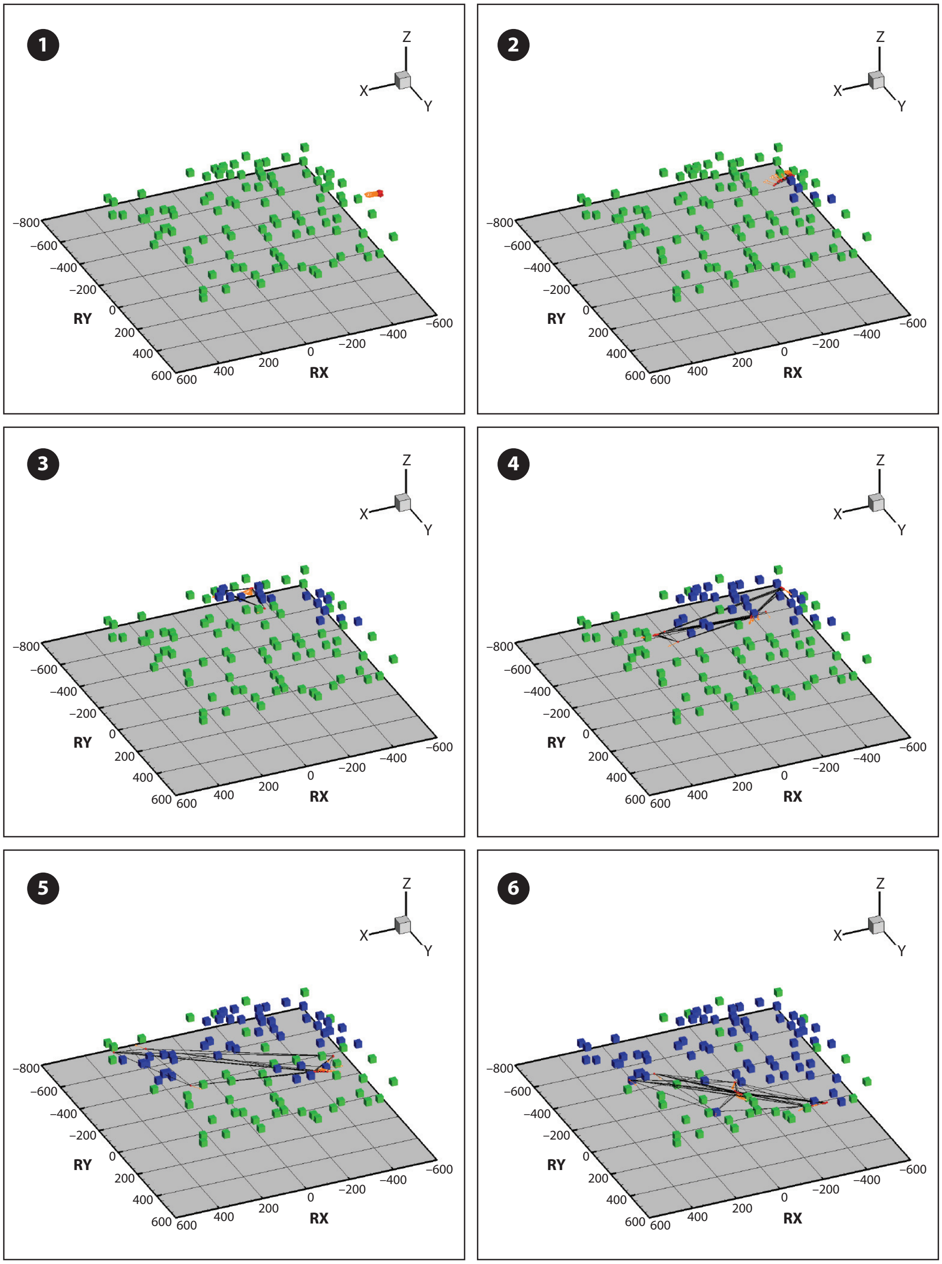

Figure 4

From top to bottom and from left to right: steps in the model mapping problem described in Sections 3.3.1 and 3.3.2. Red dots represent swarm members, and cubes represent unmapped (*green*) and mapped (*blue*) targets. Vector arrows on swarm members represent velocities, and lines are drawn as an aid to visualizing the connectivity between the swarm members. Steps continue in **Figure 5**.

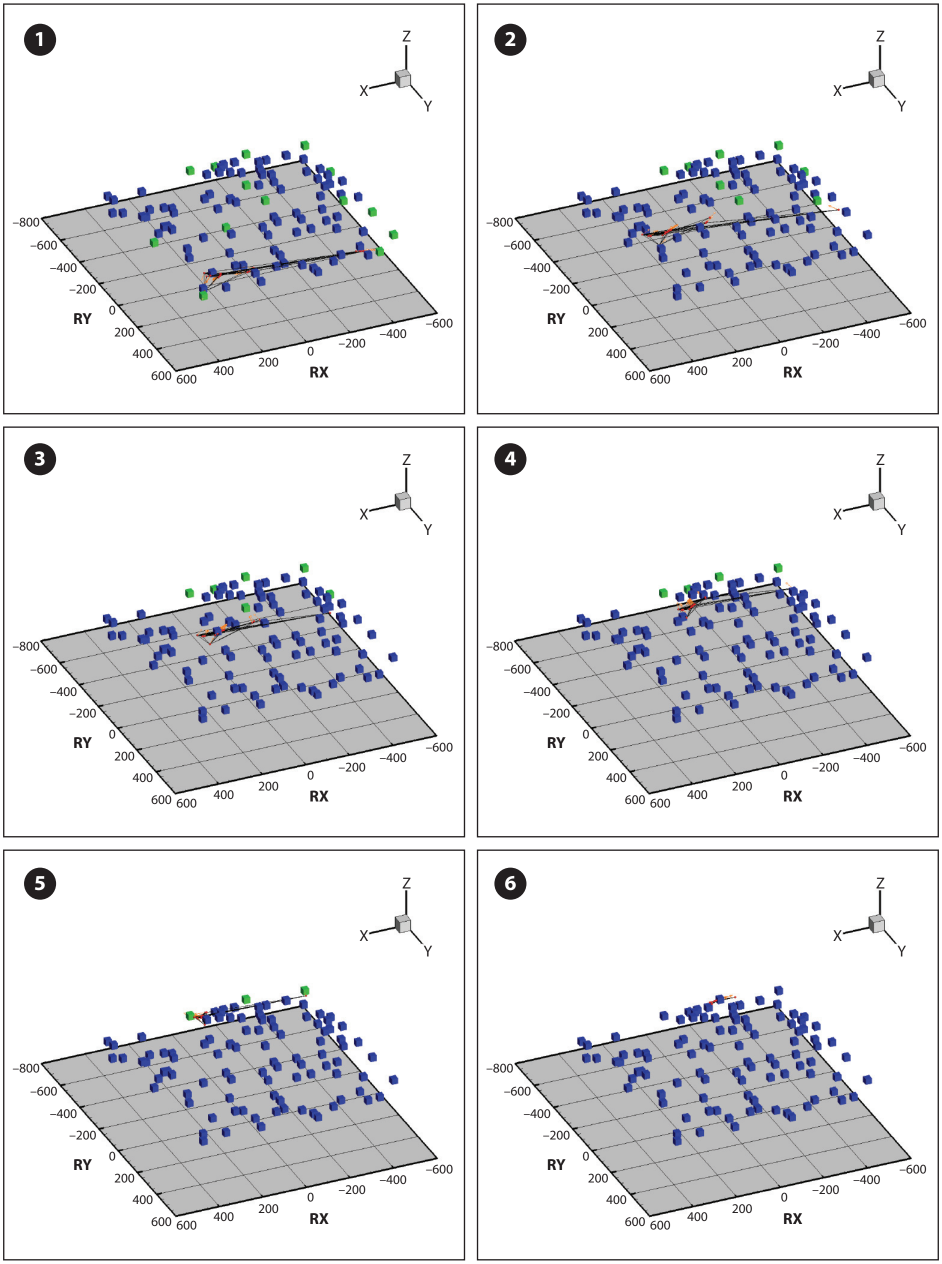

Figure 5

From top to bottom and from left to right: steps in the model mapping problem described in Sections 3.3.1 and 3.3.2. Red dots represent swarm members, and cubes represent unmapped (*green*) and mapped (*blue*) targets. Vector arrows on swarm members represent velocities, and lines are drawn as an aid to visualizing the connectivity between the swarm members. Steps are continued from **Figure 4**.

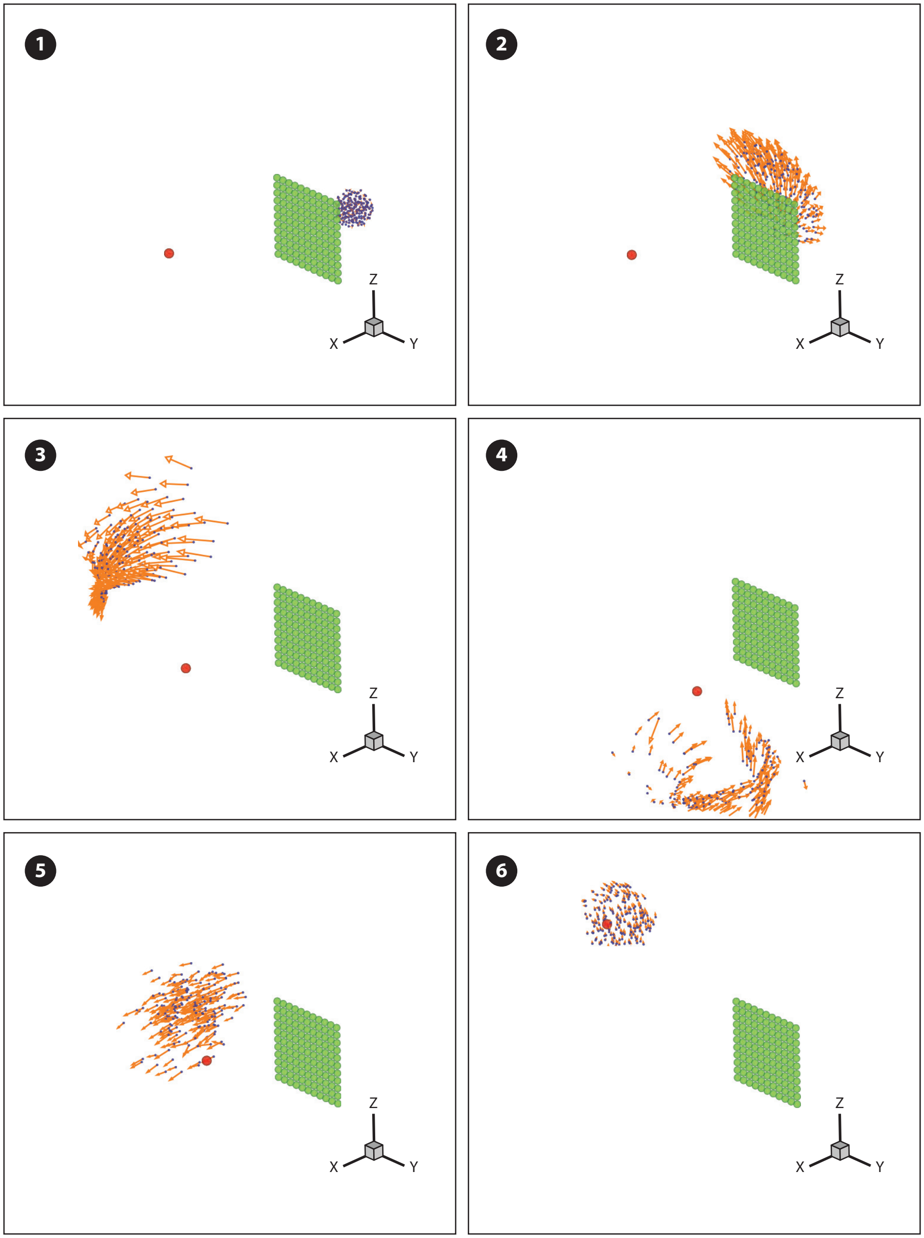

Figure 6

From left to right and from top to bottom, a swarm moves over an obstacle fence.

5.2. Example: Multisite Search

As another model problem, consider 400 swarm members and 200 randomly dispersed target sites that the swarm is tasked to visit (**Figure 7**). The algorithm is as follows: (*a*) Each swarm member is attracted to the nearest target location, and (*b*) if a site has been visited, then it is inactive (the

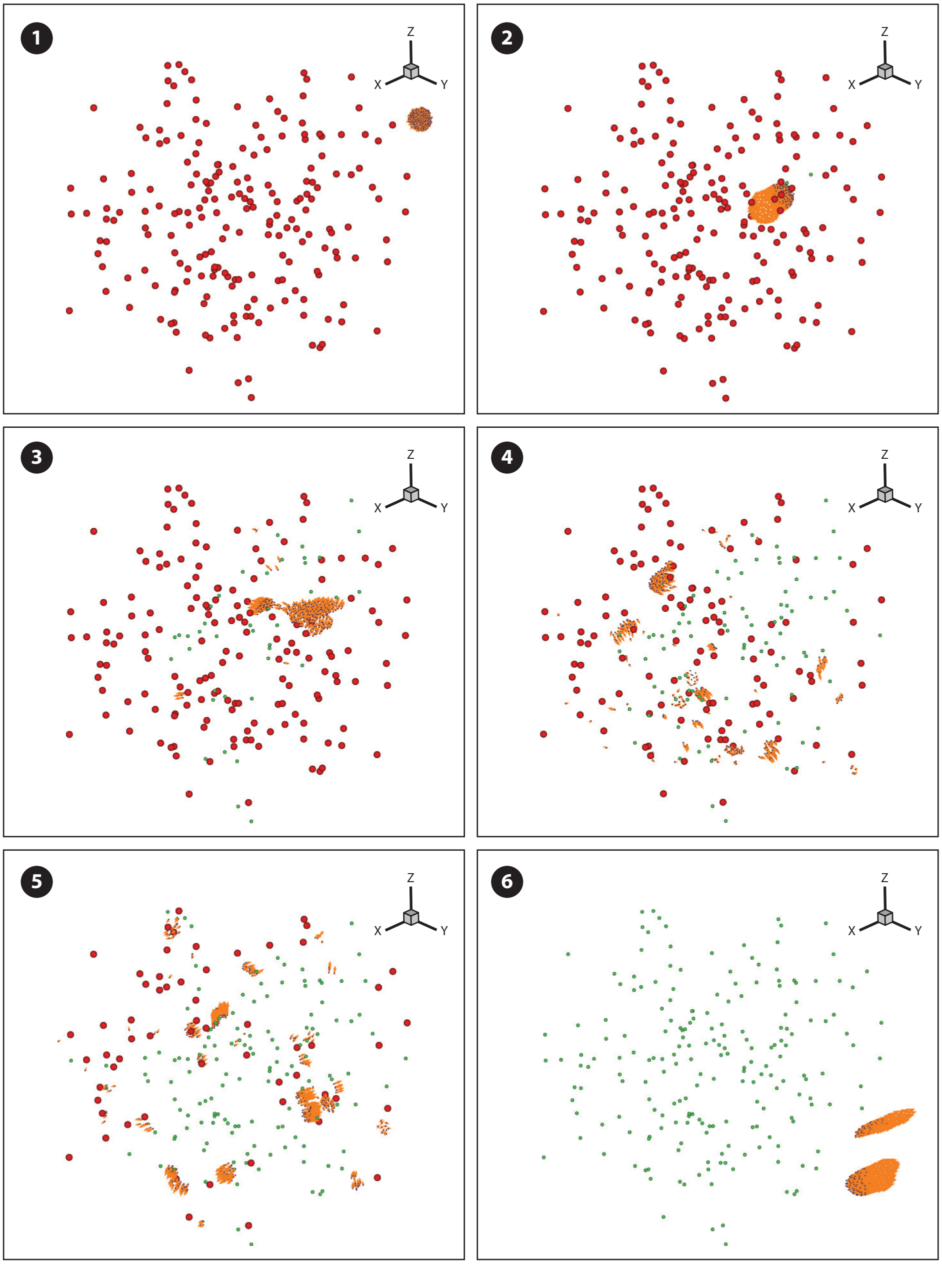

Figure 7

From left to right and from top to bottom, a swarm moves through a search space. Red sites are visited, and green sites are unvisited.

swarm is not attracted to it). As **Figure 7** indicates, the swarm has a natural tendency to divide and conquer the domain.

Typically, for systems with a finite number of particles, there will be slight variations in the performance for different starting configurations. In order to stabilize the objective function's value with respect to the randomness of the swarm starting configuration, for a given parameter selection ($\mathbf{\Lambda}$, characterized by the various α and β values), a regularization procedure is applied, whereby the performances of a series of different random starting configurations are averaged until the (ensemble) average converges, i.e., until the following condition is met:

$$\left|\frac{1}{E+1}\sum_{i=1}^{E+1}\Pi^{(i)}(\mathbf{\Lambda}^{\mathbf{I}})-\frac{\mathbf{1}}{\mathbf{E}}\sum_{\mathbf{i=1}}^{\mathbf{E}}\mathbf{\Pi}^{(\mathbf{i})}(\mathbf{\Lambda}^{\mathbf{I}})\right| \leq \mathrm{TOL}\left|\frac{1}{E+1}\sum_{i=1}^{E+1}\Pi^{(i)}(\mathbf{\Lambda}^{\mathbf{I}})\right|, \qquad 27.$$

where index i indicates a starting random configuration ($i = 1, 2, \ldots, E$) that has been generated and E indicates the total number of configurations tested. For swarms of the sizes tested, two or three sample realizations were typically needed for averaging.

Zohdi (7) tested differently sized swarms and tabulated the resulting optimal strategies (attraction and repulsion coefficients). It became clear from the results that, in some cases, if the swarm is small enough, bunching up and moving through the obstacle course is the optimal strategy. Generally, the best strategy depends strongly on the obstacle course's size and shape, the swarm's size, and the target's location. A strategy for estimating the parameters, based on genetic algorithms, is given in Section 7.

If one wishes to enforce the condition that, if a swarm member gets too close to an obstacle, it becomes immobilized, a side condition can be introduced of the following form: for all t and all r_{oj} and for $\tau < \mathcal{T}$, if

$$\|\mathbf{r}_i(t=\tau)-\mathbf{O}_j\| \leq R, \qquad 28.$$

then $\mathbf{r}_i = \mathbf{r}_i(t=\tau)$ for all $t \geq \tau$, where the unilateral condition represents the effect of being near a destructive obstacle. The swarm member is stopped in the position where it enters the radius of destruction, R. Further, the swarm performance (Π) is severely penalized if it loses members to the obstacles.

It is important to note that if the interaction is only between the nearest neighbors, and if there is no inertial reference point for the swarm members to refer to, instabilities (collisions) may occur (26–30). In the present analysis, such inertial reference points were provided by the swarm's knowledge of the stationary obstacles and target.

If one wishes to have more detailed descriptions beyond a point-mass model (for example, a quadcopter), one must augment the balance of linear momentum ($\dot{\mathbf{G}}_{cm} = \mathcal{M}\ddot{\mathbf{r}}_{cm}$), given by

$$\dot{\mathbf{G}}_{\mathrm{cm}} = \mathcal{M}\ddot{\mathbf{r}}_{\mathrm{cm}} = \sum_{i=1}^{N_c}\psi_i^{\mathrm{ext}} \stackrel{\mathrm{def}}{=} \mathbf{\Psi}^{\mathrm{EXT}}, \qquad 29.$$

with a balance of angular momentum, given by

$$\dot{\mathbf{H}}_{\mathrm{cm}} = \frac{\mathrm{d}(\overline{\mathcal{I}}\cdot\boldsymbol{\omega})}{\mathrm{d}t} = \sum_{i=1}^{N_c}\mathbf{r}_{\mathrm{cm}\rightarrow i}\times\psi_i^{\mathrm{ext}} \stackrel{\mathrm{def}}{=} \mathbf{M}_{\mathrm{cm}}^{\mathrm{EXT}}, \qquad 30.$$

where $\mathbf{M}_{\mathrm{cm}}^{\mathrm{EXT}}$ is the total external moment about the center of mass. There are various numerical methods that are capable handling complex interaction of multiple vehicles (for example, see Reference 2). Another issue that has not been taken into account is detailed treatment of the actuation and motor control that appear in the models as simply attraction and repulsion. For detailed modeling of the dynamics and control of UAVs, we refer the reader to the work of Mueller

& D'Andrea (31, 32), Mueller et al. (33), Hehn et al. (34), and Houska et al. (35), including the ability of a quadcopter to maintain flight despite the complete failure of a propellor.

6. SUMMARY AND EXTENSIONS

The dramatic increase in inexpensive UAV and camera technology has made the real-time mapping of areas struck by disasters, such as fires, earthquakes, and tsunamis, a reality. Proper deployment of UAVs promises to provide first responders with timely information in multiple locations. Critical infrastructure is now multifaceted, comprising water grids, power, traffic, etc. In many municipalities, UAV mapping is being proposed, and in some cases it has already been deployed. Technological advances and societal changes, such as massive numbers of cost-effective UAVs, are game-changers in terms of the ability to both (*a*) monitor and control events in a disaster and (*b*) facilitate long-term planning. MIT recently developed a gas-powered drone that was able to stay airborne for 5 days at a time (36).

It is important to realize that the models presented are also suitable for unmanned water vehicles (UWVs) and unmanned ground vehicles (UGVs) and for associated mapping applications where human divers are not feasible or safe (37). Much of the UWV market is driven by oil and gas operators, who not only require accurate measurements on which to base their decisions but also must obtain this information in the most cost-effective way in remote or environmentally sensitive areas. Importantly, this technology significantly lowers costs by reducing the personnel and equipment necessary to operate in remote locations. This also eliminates the expense and risk associated with traditional ship-based data collection solutions. Operations that were once cost-prohibitive are now economically feasible. The advantages of robotic solutions are numerous: (*a*) cost-effective operations and access to data; (*b*) elimination of risk of decompression sickness in human divers; (*c*) faster mapping of benthic surfaces; (*d*) reduced exposure and footprint; (*e*) less downtime due to poor weather conditions; (*f*) increased access to restricted, remote, or frontier areas; and (*g*) respect for and protection of environmentally sensitive areas such as coral reefs. Furthermore, given the complex, multifaceted biodiversity that may need to be mapped, there exists a need for flexible mapping strategies. Because of their size, small UWVs have attractive properties, such as low cost; ease of storage, maintenance, and deployment; and inherent safety (compared to manned ships). The ability to rapidly gather information in a remote zone can be used to improve the efficacy of policy efforts and may also allow for the identification of associated hazards, such as a rapidly spreading pollution. A viable approach is to use large numbers of small UWVs, equipped with a variety of sensors, to cooperatively survey the affected area. The use of a large number of UWVs increases the resilience of the data-gathering effort, as the loss of any individual member has only a small effect on the performance of the group. This is of particular importance in a storm scenario, as the UWVs could experience adverse conditions. The potential scale of areas to be mapped provides constraints on UWVs directly related to energy consumption and sensing, which in turn are related to travel time and range. There is thus substantial potential benefit in improving the energy efficiency of UWVs, either by adapting their mechanical design or by adapting their control and trajectories, in addition to utilizing solar energy.

A key condition for multi-UAV or multi-UWV technology to flourish is efficient mapping algorithms. In this regard, agent-based algorithms are a viable approach. Agent-based paradigms for simulation of coupled complex systems have become powerful predictive tools. Because different infrastructures have different grids and different quantities to be mapped, the optimal path for a set of released swarms will vary over the same terrain. The objective of this article was to expose the reader to the multitude of applications of UAV systems in laymen's terms and to provide a basic introduction to the mathematical construction of such a system and examples using agent-based models. Any agent-based model for a team of UAV's intending to map an area must contend with various optimality conditions, for example, minimum time, minimum energy usage, optical sensing, infrared sensing, acoustical sensing, or water spillage sensing. The deliverables of such a model constitute a multiple-UAV management tool that is easy to program and modify. These types of modeling tools are particularly timely given the multifaceted nature of today's critical infrastructure. For example, in the case of a disaster, rapid mapping is needed. The applications are growing rapidly and appear to be endless. UAV technology is now ubiquitous. Some particularly fascinating, recently proposed uses are drone-delivered automatic external defibrillators (AEDs) (38). AEDs were first approved by the US Food and Drug Administration in 1998 and have proven to be life-savers when put in public areas and applied by bystanders when a person has a heart attack. Quick drone delivery offers the possibility of increasing their efficacy.

7. APPENDIX: A GENETIC ALGORITHM

As with all mathematical models, the identification of parameters is important. Typically, for the class of problems considered in this work, the corresponding formulations depend in a nonconvex and nondifferentiable manner on the system parameters. Classical gradient-based deterministic optimization techniques are not robust, due to difficulties with objective function nonconvexity and nondifferentiability. Classical gradient-based algorithms are likely to converge toward only a local minimum of the objective functional if an accurate initial guess to the global minimum is not provided. Also, it is usually extremely difficult to construct an initial guess that lies within the global convergence radius of a gradient-based method. These difficulties can be circumvented by the use of a certain class of nonderivative search methods, usually termed genetic algorithms, before applying gradient-based schemes. Genetic algorithms are search methods based on the principles of natural selection, employing concepts of species evolution, such as reproduction, mutation, and crossover. Implementation typically involves a randomly generated population of fixed-length elemental strings, "genetic information," each of which represents a specific choice of system parameters. The population of individuals undergo "mating sequences" and other biologically inspired events to find promising regions of the search space. Such methods primarily stem from the work of Holland (39). Reviews of such methods have been provided by Goldberg (40), Davis (41), Onwubiko (42), Kennedy & Eberhart (6), and Goldberg & Deb (43).

Adopting the approaches used by Zohdi (7), a genetic algorithm has been developed to treat nonconvex inverse problems involving various aspects of multiobject mechanics. The central idea is that the system parameters form a genetic string, and a survival-of-the-fittest algorithm is applied to a population of such strings. The overall process is as follows: (*a*) A population (S agents in total) of different parameter sets is generated at random within the parameter space, each set being represented by a genetic string (N) of the system parameters; (*b*) the performance of each parameter set is tested; (*c*) the parameter sets are ranked from top to bottom according to their performance; (*d*) the best-performing parameter sets (parents) are mated pairwise producing two offspring (children); i.e., each best pair exchanges information by taking random convex combinations of the parameter set components of the parents' genetic strings; and (*e*) the worst-performing genetic strings are eliminated, new replacement parameter sets (genetic strings) being introduced into the remaining population of best-performing genetic strings. The process (*a*–*e*) is then repeated. The term fitness of a genetic string is used to indicate the value of the objective function. The most fit genetic string is the one with the smallest objective function. The retention of the most fit genetic strings from a previous optimization generation (parents) is critical, since if the objective functions are highly nonconvex (the present case), there exists a clear possibility that inferior offspring will replace superior parents. When the fittest parents are retained, the minimization of the cost function is guaranteed to be monotonic (guaranteed improvement) with increasing optimization generations. There is no guarantee of successive improvement if the top parents are not retained, even though nonretention of parents allows more new genetic strings to be evaluated in the next optimization generation. Numerical studies conducted by the author imply that, for sufficiently large populations, the benefits of parent retention outweigh this advantage and any disadvantages of inbreeding, i.e., a stagnant population (for more details on this inheritance property, see References 6 and 41). In the algorithm below, inbreeding is mitigated since, with each new optimization generation, new parameter sets, selected at random within the parameter space, are added to the population. Additionally, parent retention is computationally less expensive, since these parameter sets do not have to be reevaluated in the next optimization generation. Genetic algorithms can be used, for example, to search for parameter sets yielding maximal coverage of a desired area to be mapped. Mathematically speaking, this can be expressed by writing $\min_{\boldsymbol{\Lambda}} \boldsymbol{\Pi}(\boldsymbol{\Lambda})$. An implementation of such optimization ideas follows (7):

- Step 1: Randomly generate a population of S starting genetic strings, $\boldsymbol{\Lambda}^{\mathbf{i}}$ (for $i = 1, \ldots, S$): $\boldsymbol{\Lambda}^{\mathbf{i}} \stackrel{\text{def}}{=} \{\boldsymbol{\Lambda}^{\mathbf{i}}_1, \boldsymbol{\Lambda}^{\mathbf{i}}_2, \boldsymbol{\Lambda}^{\mathbf{i}}_3, \boldsymbol{\Lambda}^{\mathbf{i}}_4, \ldots, \boldsymbol{\Lambda}^{\mathbf{i}}_{\mathbf{N}}\} = \{\boldsymbol{\alpha}^{\mathbf{i}}_1, \boldsymbol{\beta}^{\mathbf{i}}_1, \boldsymbol{\alpha}^{\mathbf{i}}_2, \boldsymbol{\beta}^{\mathbf{i}}_2, \ldots\}$.
- Step 2: Compute the fitness of each string, $\boldsymbol{\Pi}(\boldsymbol{\Lambda}^{\mathbf{i}})$ (for $i = 1, \ldots, S$).
- Step 3: Rank genetic strings $\boldsymbol{\Lambda}^{\mathbf{i}}$ (for $i = 1, \ldots, S$).
- Step 4: Mate nearest pairs and produce two offspring $\lambda^{i} \stackrel{\text{def}}{=} \Phi^{(\mathrm{I})}\boldsymbol{\Lambda}^{\mathbf{i}} + (1 - \Phi^{(\mathbf{I})})\boldsymbol{\Lambda}^{\mathbf{i+1}}$, $\lambda^{\mathbf{i+1}} \stackrel{\text{def}}{=} \Phi^{(\mathrm{II})}\boldsymbol{\Lambda}^{\mathbf{i}} + (1 - \Phi^{(\mathbf{II})})\boldsymbol{\Lambda}^{\mathbf{i+1}}$ (for $i = 1, \ldots, S$).
- Note: $\Phi^{(\mathrm{I})}$ and $\Phi^{(\mathrm{II})}$ are random numbers such that $0 \leq \Phi^{(\mathrm{I})}, \Phi^{(\mathrm{II})} \leq 1$, and are different for each component of each genetic string.
- Step 5: Kill off bottom $M < S$ strings and keep top $K < N$ parents and top K offspring (K offspring $+$ K parents $+$ $M = S$).
- Step 6: Repeat steps 1–5 with top gene pool (K offspring and K parents) plus M new, randomly generated strings.
- Option: Rescale and restart search around best-performing parameter set every few optimization generations.
- Option: Gradient-based methods are sometimes useful for postprocessing solutions found with a genetic algorithm if the objective function is sufficiently smooth in that region of the parameter space. In other words, if one has located a convex portion of the parameter space with a global genetic search, one can employ gradient-based procedures locally to further minimize the objective function. In such procedures, to obtain a new directional step for $\boldsymbol{\Lambda}$, one must solve the system $[\mathbf{H}]\{\Delta\boldsymbol{\Lambda}\} = -\{\mathbf{g}\}$, where $[\mathbf{H}]$ is the Hessian matrix ($N \times N$), $\{\Delta\boldsymbol{\Lambda}\}$ is the parameter increment ($N \times 1$), and $\{\mathbf{g}\}$ is the gradient ($N \times 1$). We do not employ this second (postgenetic) stage in this work. Reviews of these methods are provided by Luenberger (44) and Gill et al. (45).

To compute the fitness of a parameter set, one must go through the procedure above, requiring a full-scale simulation. It is important to scale the system variables, for example, to be positive numbers and of comparable magnitude, to avoid dealing with large variations in the parameter vector components. Typically, for populations with a finite number of agents, there will be slight variations in the performance for different random starting configurations. In order to stabilize the objective function's value with respect to the randomness of the starting configuration, for a given parameter selection ($\boldsymbol{\Lambda}$), a regularization procedure is applied within the genetic algorithm whereby the performances of a series of different random starting configurations are averaged until the (ensemble) average converges, i.e., until the following condition is met: $|\frac{1}{Z+1}\sum_{i=1}^{Z+1}\Pi^{(i)}(\boldsymbol{\Lambda}^{\mathbf{I}}) - \frac{1}{\mathbf{Z}}\sum_{\mathbf{i=1}}^{\mathbf{Z}}\Pi^{(\mathbf{i})}(\boldsymbol{\Lambda}^{\mathbf{I}})| \leq \mathrm{TOL}|\frac{1}{Z+1}\sum_{i=1}^{Z+1}\Pi^{(i)}(\boldsymbol{\Lambda}^{\mathbf{I}})|$, where index i indicates a starting random configuration ($i = 1, 2, \ldots, Z$) that has been generated and Z indicates the total number of configurations tested. In order to implement this in the genetic algorithm, in step 2, one simply replaces "compute" with "ensemble compute", which requires a further inner loop to test the performance of multiple starting configurations. Similar ideas have been applied by Zohdi (7) to other types of randomly dispersed multibody systems.

SUMMARY POINTS

1. The dramatic increase in inexpensive Unmanned Aerial Vehicle (UAV) and camera technology has made the real-time mapping of areas struck by disasters, such as fires, earthquakes, tsunamis, etc., a reality.
2. Proper deployment of UAVs promises to provide first responders with timely information in multiple locations.
3. Critical infrastructure is now multifaceted, comprising water grids, power, traffic, etc., and in many municipalities UAV mapping is being proposed, and in some cases already deployed.
4. Technological advances and societal changes such as massive numbers of cost-effective UAVs are now game-changers in terms of the ability to both (*a*) monitor and control events in a disaster and (*b*) facilitate long-term planning.
5. Key contributors to multi-UAV or multi-UWV technology flourishing are efficient mapping algorithms. In this regard, agent-based algorithms provide are a viable approach. Agent-based paradigms for simulation of coupled complex systems have become powerful predictive tools.
6. Because different infrastructures have different grids and different quantities to be mapped, the optimal path for a set of released swarms will vary over the same terrain.
7. The objective of this article was to expose the reader to the multitude of applications of UAV systems in layman's terms and then to provide a relatively basic introduction to the mathematical construction of such a system and to provide examples using agent-based models.
8. These types of modeling tools are particularly timely given critical infrastructure is now multifaceted, comprising water grids, power, traffic, etc. For example, in the case of a disaster, rapid mapping is needed. The applications are growing rapidly and appear to be endless. UAV technology is now ubiquitous.

FUTURE ISSUES

1. Any agent-based model for a team of UAV's intending to map an area must contend with various optimality conditions: minimum time, minimum energy usage, optical sensing, infrared sensing, acoustical sensing, water spillage sensing, etc. The deliverables of such a model is a multiple UAV management tool that is easy to program and modify.
2. The models presented are also suitable for Unmanned Water Vehicles (UWVs) and Unmanned Ground Vehicles (UGVs) and associate mapping applications where human divers are not feasible or safe.

3. Much of the UWV market is driven by oil, and gas operators must not only have accurate measurements on which to base their decisions, but they need to obtain this information in the most cost-effective way in remote or environmentally sensitive areas.
4. Importantly, this technology significantly lowers costs by reducing the personnel and equipment necessary to operate in remote locations.
5. This also eliminates the expense and risk associated with traditional ship-based data collection solutions.
6. Operations that were once cost prohibitive are now economically feasible. The advantages of robotic solutions are numerous: (*a*) cost-effective operations and access to data; (*b*) elimination of risk of decompression sickness of human divers; (*c*) faster mapping of the benthic surface; (*d*) reduced exposure and footprint; (*e*) less downtime due to poor weather conditions; (*f*) increased access to restricted, remote, or frontier areas; and (*g*) respect and protection of environmentally sensitive areas such as coral reefs.
7. Furthermore, given the complex multifaceted biodiversity that may need to be mapped, there exists the need for flexible mapping strategies. Because of their size, small UWVs have attractive properties such as low cost; ease of storage, maintenance and deployment; and (relative) inherent safety (compared to manned ships).

DISCLOSURE STATEMENT

The author is not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review.

ACKNOWLEDGMENTS

This research was funded by a Pacific Earthquake Engineering Research Center grant. For more information, visit **http://www.me.berkeley.edu/people/faculty/tarek-i-zohdi**.

LITERATURE CITED

1. Totaro P, Kakaes K. 2016. Whose sky is it anyway? U.S. drone case tests rights to air space. *Reuters*, Sep. 21. **https://www.reuters.com/article/us-property-drones-rights/whose-sky-is-it-anyway-u-s-drone-case-tests-rights-to-air-space-idUSKCN11R1ND**
2. Zohdi TI. 2017. On the dynamics and breakup of quadcopters using a discrete element method framework. *Comput. Methods Appl. Mech. Eng.* 327:503–21
3. Breder CM. 1954. Equations descriptive of fish schools and other animal aggregations. *Ecology* 35(3):361–70
4. Gazi V, Passino KM. 2002. Stability analysis of swarms. In *Proceedings of the 2002 American Control Conference, May 8–10, Hilton Anchorage and Egan Convention Center, Anchorage, AK*, Vol. 3, pp. 1813–18. Piscataway, NJ: IEEE
5. Bender J, Fenton R. 1970. On the flow capacity of automated highways. *Transport Sci.* 4:52–63
6. Kennedy J, Eberhart R. 2001. *Swarm Intelligence*. San Francisco: Morgan Kaufmann
7. Zohdi TI. 2003. Computational design of swarms. *Int. J. Numer. Methods Eng.* 57:2205–19
8. Zohdi TI. 2009. Mechanistic modeling of swarms. *Comput. Methods Appl. Mech. Eng.* 198(21–26):2039–51
9. Zohdi TI. 2017. An agent-based computational framework for simulation of competing hostile planet-wide populations. *Comput. Methods Appl. Mech. Eng.* 314:513–26

10. Beni G. 1988. The concept of cellular robotic system. In *Proceedings: IEEE International Symposium on Intelligent Control 1988, Aug. 24–26, Arlington, VA*, ed. HE Stephanou, A Meystel, JYS Luh, pp. 57–62. Piscataway, NJ: IEEE
11. Brooks RA. 1991. Intelligence without reason. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI-91), Aug. 24–30*, Vol. 1, ed. J Mylopoulos, R Reiter, pp. 569–95. San Francisco: Morgan Kaufmann
12. Dudek G, Jenkin M, Milios E, Wilkes D. 1996. A taxonomy for multi-agent robotics. *Auton. Robots* 3:375–97
13. Cao YU, Fukunaga AS, Kahng A. 1997. Cooperative mobile robotics: antecedents and directions. *Auton. Robots* 4(1):7–27
14. Liu Y, Passino KM. 2000. *Swarm intelligence: literature overview*. Tech. Rep., Ohio State Univ.
15. Bonabeau E, Dorigo M, Theraulaz G. 1999. *Swarm Intelligence: From Natural to Artificial Systems*. New York: Oxford Univ. Press
16. Dorigo M, Maniezzo V, Colorni A. 1996. Ant system: optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. B* 26(1):29–41
17. Bonabeau E, Meyer C. 2001. Swarm intelligence: a whole new way to think about business. *Harvard Bus. Rev.* 79(5):106–14
18. Fiorelli E, Leonard NE, Bhatta P, Paley D, Bachmayer R, Fratantoni DM. 2004. Multi-AUV control and adaptive sampling in Monterey Bay. In *2004 IEEE/OES Autonomous Underwater Vehicles, June 17–18, Sebasco, ME*, pp. 134–47. Piscataway, NJ: IEEE
19. Feder T. 2007. Statistical physics is for the birds. *Phys. Today*. 60(10):28
20. Ballerini M, Cabibbo N, Candelier R, Cavagna A, Cisbani E, et al. 2008. Interaction ruling animal collective behavior depends on topological rather than metric distance: evidence from a field study. *PNAS* 105(4):1232–37
21. Haile JM. 1992. *Molecular Dynamics Simulations: Elementary Methods*. New York: Wiley
22. Zohdi TI. 2002. An adaptive-recursive staggering strategy for simulating multifield coupled processes in microheterogeneous solids. *Int. J. Numer. Methods Eng.* 53:1511–32
23. Zohdi TI. 2004. Modeling and simulation of a class of coupled thermo-chemo-mechanical processes in multiphase solids. *Comput. Methods Appl. Mech. Eng.* 193(6–8):679–99
24. Zohdi TI, Wriggers P. 2008. *Introduction to Computational Micromechanics*. Berlin: Springer
25. Zohdi TI. 2010. Simulation of coupled microscale multiphysical-fields in particulate-doped dielectrics with staggered adaptive FDTD. *Comput. Methods Appl. Mech. Eng.* 199:79–101
26. Hedrick JK, Swaroop D. 1993. Dynamic coupling in vehicles under automatic control. In *The Dynamics of Vehicles on Roads and on Tracks. Proceedings of the 13th IAVSD Symposium, Aug. 23–27, Chengdu, Sichuan, China*, pp. 209–20. Amsterdam: Swets & Zeitlinger
27. Hedrick JK, Tomizuka M, Varaiya P. 1994. Control issues in automated highway systems. *IEEE Control Syst. Mag.* 14(6):21–32
28. Swaroop D, Hedrick JK. 1996. String stability of interconnected systems. *IEEE Trans. Autom. Control* 41(4):349–56
29. Swaroop D, Hedrick JK. 1999. Constant spacing strategies for platooning in automated highway systems. *J. Dyn. Syst. Meas. Control.* 121:462–70
30. Shamma JS. 2001. A connection between structured uncertainty and decentralized control of spatially invariant systems. In *Proceedings of the 2001 American Control Conference, June 25–27, Crystal Gateway Marriot, Arlington, VA*, Vol. 4, pp. 3117–21. Piscataway, NJ: IEEE
31. Mueller MW, D'Andrea R. 2014. Stability and control of a quadrocopter despite the complete loss of one, two, or three propellers. In *2014 IEEE International Conference on Robotics and Automation (ICRA), May 31–June* 7, pp. 45–52. Piscataway, NJ: IEEE
32. Mueller MW, D'Andrea R. 2015. Relaxed hover solutions for multicopters: application to algorithmic redundancy and novel vehicles. *Int. J. Robot. Res.* 35(8):873–89
33. Mueller MW, Hehn M, D'Andrea R. 2015. A computationally efficient motion primitive for quadrocopter trajectory generation. *IEEE Trans. Robot.* 31(8):1294–310
34. Hehn M, Ritz R, D'Andrea R. 2012. Performance benchmarking of quadrotor systems using time-optimal control. *Auton. Robots* 33(1–2):69–88

35. Houska B, Ferreau H, Diehl M. 2011. ACADO toolkit: an open source framework for automatic control and dynamic optimization. *Optimal Control Appl. Methods* 32(3):298–312
36. Heater B. 2017. MIT's gas-powered drone is able to stay in the air for five days at a time. *TechCrunch*, June 27. **https://techcrunch.com/2017/06/27/mits-gas-powered-drone-is-able-to-stay-in-the-air-for-five-days-at-a-time/**
37. Alam M-R, Saadat M, Grenfell PW, Messner S, Jalali M-A. 2016. *Supermaneuverable autonomous swimmer*. Paper presented at the 31st Symposium on Naval Hydrodynamics, Monterey, CA, Sept. 11–16
38. Mark DB, Hansen SM, Starks ML, Cummings ML. 2017. Drone-based automatic external defibrillators for sudden death. *Circulation* 135:2466–69
39. Holland JH. 1975. *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: Univ. Mich. Press
40. Goldberg DE. 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley
41. Davis L, ed. 1991. *Handbook of Genetic Algorithms*. New York: Van Nostrand Reinhold
42. Onwubiko C. 2000. *Introduction to Engineering Design Optimization*. Upper Saddle River, NJ: Prentice-Hall
43. Goldberg DE, Deb K. 2000. Special issue on genetic algorithms. *Comput. Methods Appl. Mech. Eng.* 186(2–4):121–24
44. Luenberger D. 1974. *Introduction to Linear and Nonlinear Programming*. Menlo Park, CA: Addison-Wesley
45. Gill P, Murray W, Wright M. 1982. *Practical Optimization*. Bingley, UK: Emerald Group

Annual Review of Environment and Resources

Volume 43, 2018

Contents

I. Integrative Themes and Emerging Concerns

II. Earth's Life Support Systems

Indexes

Errata

An online log of corrections to *Annual Review of Environment and Resources* articles may be found at http://www.annualreviews.org/errata/environ