A Tactical Command Approach to Human Control of Vehicle Swarms

Jacob Beal
Raytheon BBN Technologies
10 Moulton Street
Cambridge, MA, 02138

Abstract

Human control of vehicle swarms faces a dilemma: an operator must be able to exercise precise control over how a mission is executed, but controlling individual vehicles is not scalable. The Proto spatial computing language offers an intermediate representation, where the motion of a swarm is specified as a vector field, which is then approximated by the movement of individual members (Bachrach, Beal, and McLurkin 2010). I propose that this can be exploited to build a "tactical command" model of swarm control, whereby human "officers" dynamically decompose a swarm into units and task those units to carry out geometric and topological maneuvers under the constraints imposed by the platform. This abstraction may also allow situation awareness interfaces for individual agents to be extended to apply to swarm units.

Introduction

The use of individual unmanned vehicles has now become routine. On land, air, and sea they are used for a wide range of applications including conducting scientific surveys (e.g., (Leonard et al. 2007)), disaster response (e.g., (Murphy et al. 2011)), and military reconnaissance and combat (e.g., (Yamauchi 2004)). As the availability, affordability, and reliability of unmanned vehicles continues to improve, it is increasingly reasonable to contemplate deployment not just of a few vehicles at a time, but of swarms of dozens or hundreds or more. Consider, for example, large-scale oceanic or atmospheric environmental monitoring, wilderness fire-fighting, or scouting at the leading edge of a military operation.

To date, however, nearly all deployed applications have used at most a handful of vehicles. One of the important obstacles to larger deployments has been the difficulty of coordinating multi-vehicle operations. Remote manual piloting of vehicles is quite challenging, typically requiring at least one human operator per vehicle, and often more—e.g., a driver and a mission specialist (Peschel and Murphy 2011). This makes it extremely costly and difficult to control a swarm of vehicles. At a slightly higher level, a large number of control algorithms have been developed for formation control, each able to provide a particular class of maneuvers for a swarm under certain assumptions (e.g., (Hsieh et al. 2008; Ogren, Fiorelli, and Leonard 2002; Ji and Egerstedt 2007)). At the other end of the spectrum, a vast number of algorithms have been developed for autonomous performance of particular tasks. These, however, tend to be difficult to customize for the practicalities of the task or platform at hand, and give little opportunity for input from human operators. What is missing is an intermediate representation where human operators can exercise tight control over how the vehicles carry our their mission, but where their commands are given to aggregates of vehicles at a higher level of abstraction. Some prior work has been done in the area of representation, such as Mataric's work on basis behaviors (Mataric and Marjanovic 1993) or the work by Klavins (Klavins 2004) and Kloetzer and Belta (Kloetzer and Belta 2006) on description languages for swarm flocking, but the results so far have been generally unintuitive and difficult to translate into effective control for complex situations.

I propose that, for vehicles in loosely constrained spaces, this gap may be addressed with a "tactical command" architecture loosely inspired by human military command. The proposed architecture exploits a continuous space abstraction of aggregates (Beal 2004; Beal and Bachrach 2006) that has already been applied to the control of robot swarms (Bachrach, Beal, and McLurkin 2010). This paper builds on top of that prior work to develop a model whereby human "officers" can decompose a swarm on the fly into a hierarchy of units, then task those units to carry out maneuvers constrained by autonomic considerations of self-preservation, much like those that might be carried out by human troops under an officer's command. Finally, I propose how the same abstraction may be used to extend situation awareness interfaces for individual agents to apply as well to swarm units of arbitrary size.

Command and Control of Swarms

Let us begin our investigation of how to control swarms of unmanned vehicles by considering another domain where there is already a well developed model for controlling swarms: military command. Here the swarm is not of vehicles, but of humans. Such a military force is divided into units hierarchically (e.g., infantry being decomposed into

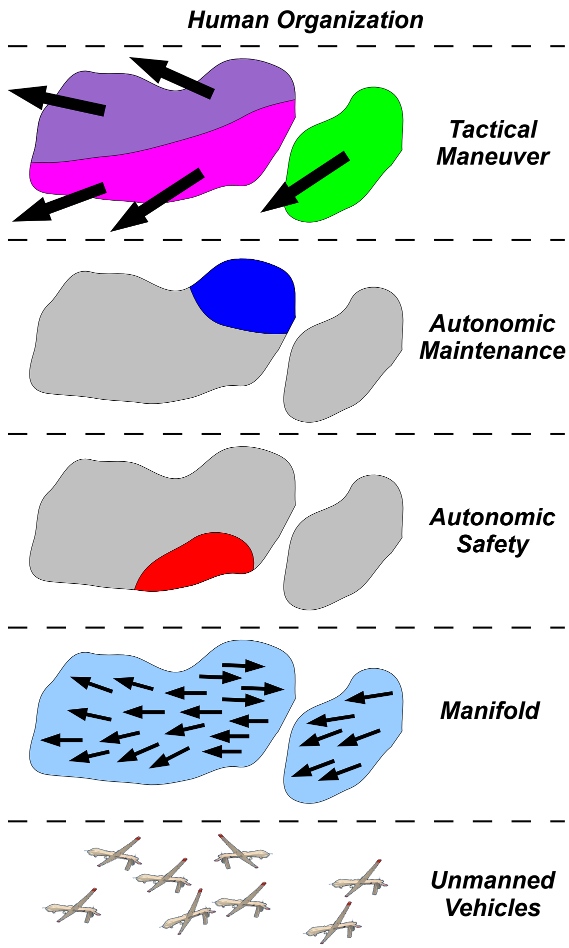


Figure 1: Proposed tactical command architecture for human control of unmanned vehicle swarms: low-level control of the actual unmanned vehicles supports a vector-field abstraction, on which autonomic safety and maintenance procedures are built. Human "officers" can then control the swarm subject to these constraints, using tactical maneuver commands that act on swarm structure and geometry.

companies, platoons, and squads), and may be commanded at any level of the hierarchy.[1] A commander can then formulate orders in terms of aggregates of humans rather than individuals.

Interestingly, military thinking and orders often make a close connection with spatial concepts. For example, responsibilities of units are often based on geographical regions, e.g., "Company A will take Hill 103 while Company B takes Hill 107," or "3rd Battalion is responsible for everything South of the river." Another example is how strength of forces is often expressed in terms of density, e.g., concentrating firepower or being spread too thin.

Humans, of course, are much more intelligent than unmanned vehicles. It is unreasonable to expect vehicles to be able to handle commands like "take that hill," or "stop that sniper" any time soon. Simpler tactical commands, however, are quite possible to implement with currently available formation control algorithms. Many tactical commands affect only the arrangement of swarm members. For example, implementing "spread out and advance 200 meters" requires only adjusting the separation between vehicles and translating them through space. Other tactical maneuvers may restructure the swarm and assign tasks to subsets: for example, "engage targets by squads" could be implemented by splitting the vehicles into groups, running a target selection algorithm on each group, and using a coloring algorithm to ensure that groups pick different targets.

Taking inspiration from this, we can propose a *tactical command* architecture for human control of vehicle swarms. The idea here is to allow a human to act as an "officer" issuing orders to units that may comprise many vehicles. The vehicles, meanwhile, first see to their own safety and supply needs, and execute the orders only so far as these do not conflict. Thus, vehicles that are neglected or given bad commands may not act effectively, but will at least not be endangered. Figure 1 illustrates this architecture, which comprises six levels of abstraction:

1. **Unmanned Vehicles:** The lowest level of abstraction is the operation of sensors, actuators, and communication systems on particular unmanned vehicles. For example, Figure 1 shows a collection of eight unmanned aircraft, organized into a group of six on the left and a group of two on the right.

2. **Manifold:** At the next level, the collection of vehicles is viewed as an approximation of a continuous manifold with local information sharing—an *amorphous medium* (Beal 2004). As established in (Bachrach, Beal, and McLurkin 2010), the motion of the swarm can be specified here as a vector field over the manifold. The values of this vector field will be determined by the levels above.

3. **Autonomic Safety:** A human officer should not be expected to ensure that individual vehicles avoid obstacles and other vehicles. Instead, the autonomic safety level takes on the task of avoiding collisions, using sensor information and the position and velocity of neighboring vehicles to modulate the commanded velocity. For example, in Figure 1 two of the vehicles have become too close together, and their velocities need be perturbed away from the commanded velocity in order to separate them.

4. **Autonomic Maintenance:** Similarly, a human officer should not be expected to track the fuel, supplies, and faults of every vehicle in the swarm. Instead, the autonomic maintenance level should handle this by assuming control over vehicles that need service, effectively removing them from the "mission" portion of the swarm until servicing is complete. For example, in Figure 1 one of the vehicles is heading in the opposite direction of the rest, as it returns for service. The responsibility of the human officer is to ensure that their maneuver specifications can tolerate the absence of vehicles needing service and to ensure that servicing facilities are available.

[1] Though in practice the hierarchy is typically not strict.

5. **Tactical Maneuver:** The human officer actually commands the swarm with combinations of three classes of operations: 1) subdividing the swarm into units, 2) delegating units of the swarm to command by other humans, and 3) specifying maneuvers in terms of geometric and topological changes (which will be modulated by the lower levels). For example, Figure 1 shows the left group being split into two subgroups, and each group given a movement command. We will discuss the form of these commands in detail in the next section.
6. **Human Organization:** Finally, for completeness we include the human or humans who are commanding the swarm. Note that when different humans command different portions of the swarm, the humans necessarily will have some organization of their own for determining who commands which portion of the swarm and for coordinating their commands.

The key change of representation in this architecture is the abstraction of the collection of unmanned vehicles as a continuous manifold. This is what allows commands to be given for aggregate portions of the swarm, without detailed knowledge about the individual vehicles involved. The manifold model also means that commands can scale and adapt to vehicles coming and going for service without the involvement of the human officer, as presented in (Beal and Schantz 2010).

This representation, and thus the architecture as a whole, depends on three assumptions about vehicles, as established in (Bachrach, Beal, and McLurkin 2010):

- Vehicles must have relatively inexpensive and fairly reliable communication with all other nearby vehicles (though this may be inhibited by obstacles or other features of the environment).
- Vehicles must be able to determine approximate distance and range to their neighbors and must share relative coordinates (global coordinates are useful to have, but not necessary).
- Vehicles must be able to approximately move according to an arbitrary limited velocity vector (e.g., a winged flying vehicle is fine as long as its turning radius is smaller than the precision of control needed). Note that this means dexterous actions, such as opening doors or capping wells, are not supported by this abstraction, and neither are vehicles maneuvering in highly constrained spaces.

If these assumptions are satisfied, then the continuous space abstraction can be valid, and the movement of the swarm can be adequately specified by means of a vector field, meaning the proposed tactical command architecture should be viable.

Tactical Maneuver Commands

The interface between a human officer and the vehicle swarm is tactical maneuver commands. Commands need to be both simple for a human to understand and also fairly simple to implement with a robust distributed algorithm on a swarm of vehicles. This also means the commands must implicitly tolerate perturbations in the velocity due to autonomic safety and changes in the population of vehicles due to autonomic maintenance. Commands must also implicitly adapt to complex operating environments, such as for marine vehicles in a harbor.

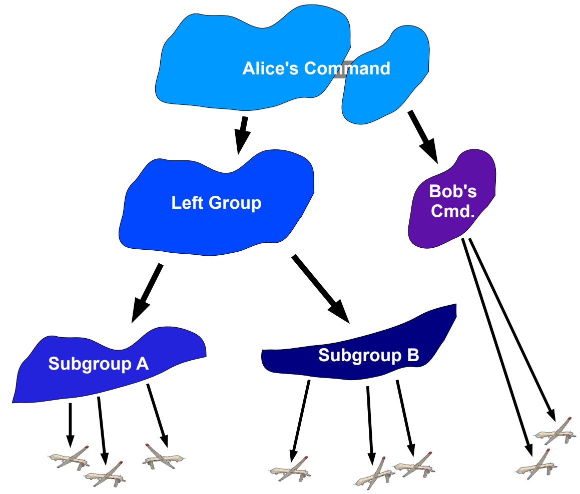


Figure 2: The unit organization of a swarm may be expressed as a multitree, subdividing the swarm into units to be commanded and delegating command of units to various human officers. For example, the units from Figure 1 might be organized into a nested set of five units under the command of two different officers.

In this section, we present an example family of such tactical maneuver commands, based on hierarchical unit organization and the moments of a vehicle distribution. While this set is no doubt imperfect, meaning that some applications likely cannot be expressed well without additional primitives, these examples are sufficient to demonstrate how succinct command and control of swarms is possible using the manifold abstraction.

Primitive Tactical Maneuvers

The basis for our family of maneuver commands are primitive operators affecting the logical and physical organization of the swarm. In this initial paper, we do not propose particular algorithms for carrying out these primitive maneuvers; for now, it is enough to note that all of these primitives are easily within the scope of existing control techniques.

Unit Organization and Command The unit organization of a swarm can be expressed as a multitree constructed using two operators:

- designating a sub-unit within a larger unit, and
- assignment of unit command to a subordinate human officer

In each case, the subset being acted upon can be identified by an indicator function over the space, marking each vehicle as part of the affected unit or not. Figure 2 shows a simple example, in which the units from the top level of Figure 1 are

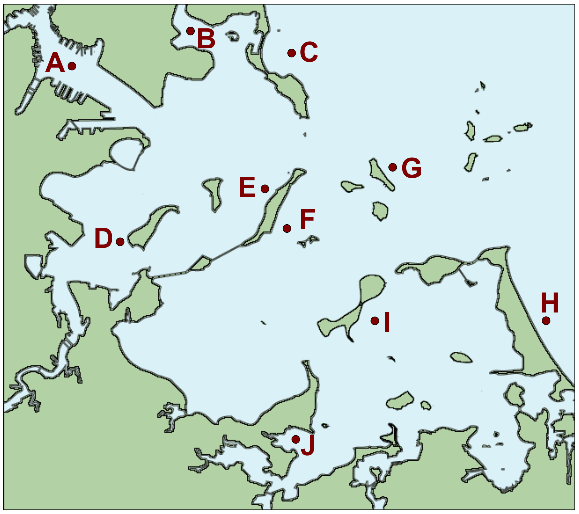


Figure 3: Geometric operators for maneuvering a swarm or controlling its motion must be able to conform to complex manifolds. For example, a swarm of water-based vehicles maneuvering in Boston Harbor cannot construct a triangular formation or path that connects any three of the points labelled on the map above. Constructs like shortest path, bisector, or boundary, however, are compatible with manifolds and remain well defined.

organized into five units under the command of two different officers.

Unit Geometry and Topology Once parsed into units, the arrangement of vehicles in a unit can be controlled in terms of their collective geometry and topology. The continuous abstraction represents each unit as a manifold that covers the space through which the unit's vehicles are scattered, plus a density function indicating how closely packed the vehicles are at each point in space (Bachrach, Beal, and McLurkin 2010). A maneuver is thus a mass-flow function over the manifold that shifts it over time into the desired manifold and density function. This may then be implemented approximately by the motion of individual vehicles, which represent quanta of mass.

While there are a vast number of possible geometric commands, not all are compatible with this manifold abstraction. For example, "form a triangle" does not work on a seascape with islands and peninsulas, where the notion of "triangle" may be ill-defined, as illustrated in Figure 3.

One set of geometric properties that certainly are safe to measure and control, however, are the moments of the swarm unit. In the continuous model, this means the moments of inertia of the manifold considered as a rigid body. These continuous moments may then be mapped directly to the statistical moments of the distribution of vehicles, forming a valid approximation.

The first several moments are:

- **Zeroth moment:** total mass of the swarm
- **First moment:** center of mass of the swarm
- **Second moment:** volume, aspect-ratio, and pose of the swarm
- **Third moment:** asymmetries (skewness) in the distribution of the swarm.

Each moment is a tensor, with the kth moment containing D^k elements, where D is the number of dimensions (typically two or three for a swarm). The set of moments is countably infinite, but only the lowest few have clear physical intuitions associated with them.

Given a distributed estimator for computing the moments of a swarm unit, it is possible to implement operators for controlling the geometry of the swarm by acting on these moments or the relations between them. For example:

- Controlling density sets the mass/volume relationship
- Controlling position can implement way-point following
- Controlling aspect ratio and pose can create formations

Similarly, topological operators can be used to regulate the relations between swarm units or between the elements of a unit. Just as with geometric operators, some topological operators can be problematic to implement. For example, having a unit connect to the exterior of another unit requires categorizing surfaces into exterior and interior, which may be difficult in a complex environment. Operators that are certainly safe to implement, however, are those that control connectedness. For example:

- Ensuring elements of a unit remain nearby one another
- Ensuring two units remain separated from one another

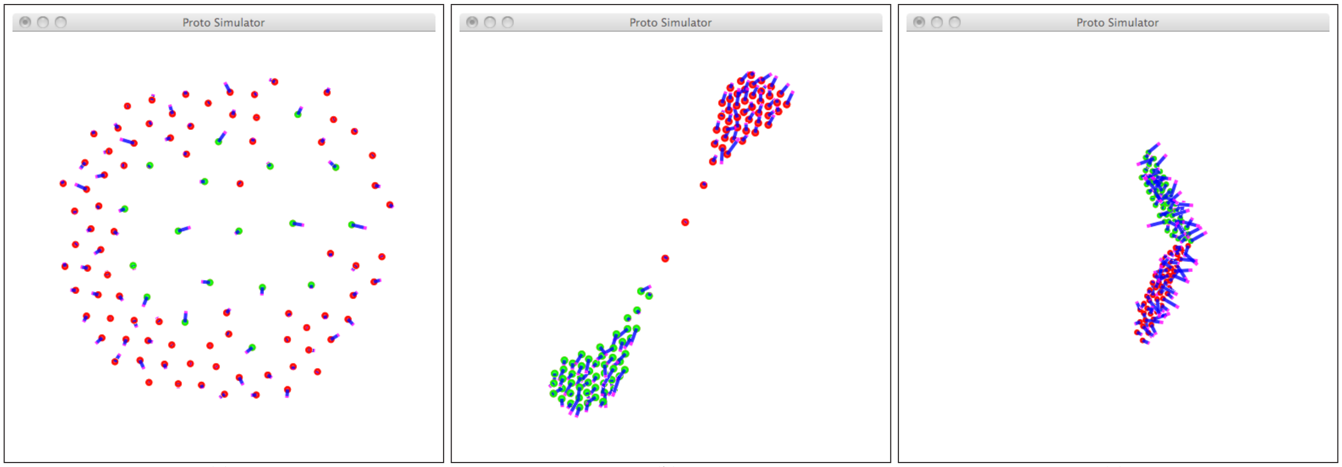
Composite Maneuvers

Note that the geometric and topological operations that we have discussed so far are largely independent of one another. For example, a swarm can change its density, spreading out or bunching up, without affecting its position, aspect ratio, or connectedness. A maneuver for a unit may thus be constructed from any non-conflicting combination of operators these properties, plus a specification of the speed with which the maneuver is to be carried out.

Figure 4 shows several examples of such maneuvers, as implemented on swarms using Proto (Beal and Bachrach 2006; MIT Proto Retrieved June 22nd 2012). Each example uses a set of independent controls on geometric properties of the units of the swarm. Each control computes a vector field over the swarm. These vector fields are then added together and modulated by the topological controls to produce a vector field specifying the tactical maneuver. The maneuver field is then modulated by the autonomic controls to produce the vector field that ultimately controls vehicle movement.

More complex maneuvers can be constructed by splicing together vector fields piecewise. Most immediately, this means that the vector fields computed for particular units can be spliced together to form a vector field specifying a heterogeneous composite maneuver for the entire swarm.

In general, any given vehicle may have multiple maneuvers computed that apply to it. Unlike the relatively independent aspects of a unit's geometry, however, blending such



(a) (b) (c)

Figure 4: Examples of maneuvers built from geometric and topological control primitives, demonstrated on a simulated 100-vehicle swarm: (a) red and green units are maintaining connectivity, while red attempts to be four times as dense as green; (b) red and green units are moving away from one another while maintaining connectedness at a priority over tight density; (c) red and green units are maintaining a thin aspect ratio as they move slowly apart and to the right, creating a chevron formation.

maneuvers together is likely to result in failure, since they are not independent. The organizational heterarchy of the swarm, however, can be used to determine which command takes priority, with the most specific unit dominating. Finally, if no maneuver applies to a particular unit, it may take the default action of doing nothing, while the autonomic layers of the architecture ensure that neglect remains benign.

Maneuvers may also be sequenced in time, making transitions dictated by the progress of time, by the completion of prior maneuvers, or by external events. With a human operator overseeing the swarm and the limited degree of true mission autonomy available in the near term, this type of composition is likely to be much less heavily used.

Command and Control Interface

Swarm operators should not be required to write complex textual programs in order to command their vehicles. The sort of geometric operators that we have described, however, are potentially much more intuitive for a human to work with. If we neglect sequencing and only allow the current maneuver for each unit to be specified, then a simple graphical interface may be sufficient for programming the behavior of a swarm.

Assuming that the vehicles are able to report position information, the swarm can be displayed embedded into a map. The operator would then be able to select vehicles by region, by unit, or individually, and issue them commands that reorganize units or delegate units to the control of other operators. Maneuvers could be commanded by selecting a unit and then selecting which to control out of the primitive geometric and topological properties discussed above. In this way, a single human operator might control the movement and configuration of a large number of vehicles, though with only limited mission autonomy.

Situation Awareness for Swarms

The same aggregate programming ideas that we can consider as a means of controlling a swarm can also be applied to situation awareness for swarms. The problem to be addressed here is that if there is a small amount of information coming from each vehicle, the volume of information from the whole collection of vehicles is likely to be overwhelmingly large relative to human cognitive capabilities.

As before, a potentially good answer is to interface with the human operator in terms of units of vehicles rather than in terms of individual vehicles. For any given class of information, there are three basic strategies available for reducing the values of a unit down to a single element presented to the operator:

- **Construct a field over the unit's sub-manifold:** This retains all of the original values, but represents them a form that can be easily digested by an operator. Since the continuous space approach abstracts a group of vehicles as a continuous space, it is natural to view the values of individual units as samples of a scalar or vector field over that space. By representing values in a continuous visual map, it should be simple for an operator to gain the gist of a situation at a glance, while allowing for detailed study as necessary. For example, heat or radiation measurements can form a scalar field, while wind or current measurements can form a vector field.
- **Aggregate individual values into a unit statistic:** This strategy is essentially map/reduce, and operates by compressing the original values into a single output. It is appropriate when the operator is concerned with the amount of resources available but not the values of individual vehicles. For example, an operator controlling UAVs for wilderness firefighting might want to know the mean remaining fuel in the vehicles of a unit, or the total amount

of fire retardant in all of their tanks.

- **Take a movable perspective:** This strategy discards all but a single value, but allows the user to select that value by shifting their perspective over space. This approach is appropriate when dealing with sensors, such as video, that are highly useful to an operator but difficult to combine. At and given time, the user sees from the perspective of a particular location in the space occupied by the unit, as represented by the sensory feed from the vehicle closest to that location. The user may then shift perspective by shifting location continuously through the space occupied by the unit, which may provide more continuity in perception as the shift from vehicle to vehicle is incremental in space.

In all three cases, these techniques effectively transform a collection of information from many vehicles into a single representation for the entire unit. In this way, any situation awareness method that is designed for a single vehicle might be extended to work for a swarm as well, by treating the unit as a whole as though it were a single spatially-extended virtual vehicle.

Conceptual Example: Wilderness Fire Control

Let us illustrate these ideas with an example scenario, of how they might be applied to the domain of wilderness fire control. In the mountainous Western areas of the U.S., there are vast tracts of fire dependent forest in highly inaccessible wilderness land. Management of fires in this region often makes heavy use of aerial vehicles: for surveillance, for suppression by dropping water or spraying fire retardant, and for delivery of parachuting or rappelling firefighters.

Consider a firefighting scenario in which some of the surveillance and retardant-spraying aircraft have been replaced by UAVs. In order to integrate the human-controlled aircraft with the tactical command architecture, they just need to have a positional transponder that the UAVs can sense and be represented as being under another command. This makes them inaccessible to the operators, but still considered by the autonomic safety level.

Let us say that the UAVs are being controlled by a team of three UAV operators working together, Alice, Bob, and Carol. Alice is designated as the commander of the team, and she takes the scouting UAVs for herself, then splits the retardant-spraying UAVs into a unit for Bob and a unit for Carol. Each operator's interface is based on a map, which shows the position of the UAVs and current knowledge about the extent of the fire, plus annotations they have made about other firefighting units based on their conversations with the overall fire commander. Each operator sees the estimated mean fuel for their units of UAVs, and the retardant-spraying units also report their estimated total available retardant.

While Bob and Carol each handle one front of the fire, Alice has spread her scouting UAVs out to low density and begins investigating the next valley over. She shifts her perspective through the swarm, looking out its cameras for signs that the fire is spreading. Seeing a smoke plume, she increases the density of her UAVs and concentrates them on the area so that she can evaluate it thoroughly.

It turns out that this smoke plume is the mark of a new hot-spot, and needs immediate attention, so Alice asks Bob and Carol if either of them can spare some UAVs. Bob checks with his liaison Dave amongst the firefighters on the ground by his front and determines that things are fairly under control, so he splits his unit and sends half of it over toward Alice's UAVs, while delegating control of the other half to Carol while he focuses on the new problem. Arriving at Alice's hot-spot, he arranges his retardant UAVs into a sweep formation—a dense formation with a high aspect-ratio orthogonal to the direction of travel—and makes a retardant spraying pass over the hot-spot. Hot-spot under control, he sends his unit back toward the front he was previously working, and some of them peel off autonomously to return to base and get their retardant tanks refilled.

This scenario illustrates how the tactical command model may be a natural fit for human control of vehicle swarms. All of the types of maneuvers being carried out are fairly simple to implement, and the interfaces described are not particularly exotic. The key to making such a system work is the manifold abstraction that will allow an operator to control a group as though it were a single spatially-extended individual, and the autonomic safety and maintenance layers that can lessen the need for vehicle micromanagement.

Contributions

In this paper I have proposed a tactical command architecture for swarm control based on a continuous space abstraction. Under this model, a swarm of vehicles is organized into units under the command of various human operator "officers," and those operators command units in terms of their aggregate geometric and topological properties. This has the potential to greatly reduce cognitive load by allowing an operator to think of large sets of vehicles as single space-spanning objects. The abstraction away from individual devices also allows the insertion of autonomic maintenance and safety processes that may temporarily assume command of particular vehicles in order to ensure that operator neglect remains benign. Taken together, this architecture has the potential to greatly simplify the direction of swarms of vehicles, as well as offering mechanisms by which situation awareness methods for individual vehicles may be extended to operate on swarms as well.

References

Bachrach, J.; Beal, J.; and McLurkin, J. 2010. Composable continuous space programs for robotic swarms. *Neural Computing and Applications* 19(6):825–847.

Beal, J., and Bachrach, J. 2006. Infrastructure for engineered emergence in sensor/actuator networks. *IEEE Intelligent Systems* 21:10–19.

Beal, J., and Schantz, R. 2010. A spatial computing approach to distributed algorithms. In *45th Asilomar Conference on Signals, Systems, and Computers*.

Beal, J. 2004. Programming an amorphous computational medium. In *Unconventional Programming Paradigms International Workshop*, volume 3566 of *Lecture Notes in Computer Science*, 121–136. Springer Berlin.

Hsieh, M.; Halász, Á.; Berman, S.; and Kumar, V. 2008. Biologically inspired redistribution of a swarm of robots among multiple sites. *Swarm Intelligence* 2(2):121–141.

Ji, M., and Egerstedt, M. 2007. Distributed coordination control of multiagent systems while preserving connectedness. *Robotics, IEEE Transactions on* 23(4):693–703.

Klavins, E. 2004. A language for modeling and programming cooperative control systems. In *Proceedings of the International Conference on Robotics and Automation*.

Kloetzer, M., and Belta, C. 2006. Hierarchical abstractions for robotic swarms. In *IEEE International Conference on Robotics and Automation*.

Leonard, N.; Paley, D.; Lekien, F.; Sepulchre, R.; Fratantoni, D.; and Davis, R. 2007. Collective motion, sensor networks, and ocean sampling. *Proceedings of the IEEE* 95(1):48–74.

Mataric, M., and Marjanovic, M. 1993. Synthesizing complex behaviors by composing simple primitives. In *Proceedings, Self Organization and Life, From Simple Rules to Global Complexity, European Conference on Artificial Life (ECAL-93)*, 698–707.

Retrieved June 22nd, 2012. MIT Proto. software available at `http://proto.bbn.com/`.

Murphy, R.; Dreger, K.; Newsome, S.; Rodocker, J.; Steimle, E.; Kimura, T.; Makabe, K.; Matsuno, F.; Tadokoro, S.; and Kon, K. 2011. Use of remotely operated marine vehicles at minamisanriku and rikuzentakata japan for disaster recovery. In *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*, 19–25.

Ogren, P.; Fiorelli, E.; and Leonard, N. 2002. Formations with a mission: Stable coordination of vehicle group maneuvers. *Proc. 15th International Symposium on Mathematical Theory of Networks and Systems* 170.

Peschel, J. M., and Murphy, R. R. 2011. Mission specialist interfaces in unmanned aerial systems. In *Proceedings of the 6th international conference on Human-robot interaction*, HRI '11, 225–226. New York, NY, USA: ACM.

Yamauchi, B. M. 2004. PackBot: a versatile platform for military robotics. In Gerhart, G. R.; Shoemaker, C. M.; and Gage, D. W., eds., *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 5422, 228–237.