2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

On the use of Capability Functions for Cooperative Objective Coverage in Robot Swarms

Bradley E. Bishop, *Member, IEEE*

Abstract— **In this paper, we present a new abstraction method by which functional capabilities of a swarm of robots can be automatically distributed across a number of objectives. To demonstrate the framework, we provide a simulation study for a swarm of autonomous aerial vehicles performing a reconnaissance mission. The framework developed provides a novel methodology for real-time resource allocation for a wide variety of missions.**

I. Introduction

Development of the controllers for swarms of autonomous vehicles requires significant care if the capabilities of the swarm as a whole are to be utilized appropriately. A great deal of research has been directed at developing centralized and decentralized control strategies for a wide variety of applications [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]. A concern that is not addressed by these controller formulations is the ability to quantify and allocate swarm resources appropriately for various missions and objectives. In this paper, we develop a control methodology, based on functional capability abstraction, which provides real-time resource allocation and role assignment for swarms of cooperating vehicles.

Small autonomous vehicles have many advantages over manned systems, in terms of cost, size and capability. As such devices are integrated into existing fielded systems, they will act as significant force multipliers and will add a great deal of capability. The cost of this capability will be, of necessity, a supervisory control structure, under which one operator will be responsible for a large number of units. That these units may be of varying types with varying capabilities and limitations places significant cognitive pressure on the operator. As has been demonstrated in urban search and rescue activities [13,14], simply operating a device in a challenging environment precludes significant secondary cognitive effort (such as scanning rubble for survivors). The difficulties will be compounded when the operator must guide multiple reconnaissance units in a rapidly evolving operational environment. In this paper, we present a robot swarm control concept, based on [1] and [15] that provides:

1) *Automatic resource allocation and re-allocation*
2) *Autonomous, on-line path planning and reactive formation generation*
3) *Flexibility in specification of multiple objectives and required total resource levels*

The remainder of the paper is organized as follows. In Section II, we discuss the problem statement and the assumptions on swarm composition. In Section III, we define the control method that is used to achieve capability-based swarm control. Section IV includes several simulations demonstrating the efficacy of the system. Finally, we offer conclusions and some ideas for future work in Section V.

II. Problem Statement

A. Vehicle Kinematics

In this paper, we will consider control of a swarm of cooperating unmanned aerial vehicles (UAVs). Let there be *n* unmanned aerial vehicles (UAVs) with simple kinematics approximating those of an autonomous helicopter (see Figure 1). Effectively, we assume that the UAVs are *holonomic* in nature, capable of hovering and moving in any direction (with limited total speed) while airborne. We simplify the complete kinematic description of the i[th] UAV system to include only position (p_i) in the world coordinate frame, as seen in Figure 1. More complete kinematic descriptions of the vehicle would include an Euler-angle representation of the pose along with pose-based velocity limits. Such a model is more complex than required for this work, which will focus on the simple holonomic model in order to demonstrate the efficacy of the control methodology.

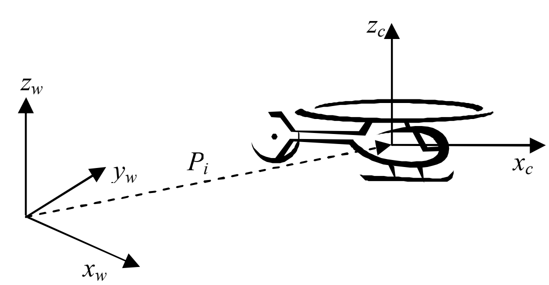

Figure 1: i[th] UAV model

We assume that there are limits on the overall velocity of each of the UAVs, although we allow that individual units

Manuscript received September 15, 2006. This work was supported in part by the Office of Naval Research under the National Naval Responsibility in Naval Engineering (NNR-NE) Program (Grant N0014-03-1-0160).

B. E. Bishop is with the United States Naval Academy, Annapolis MD 21402 (phone 410-293-6117; fax: 410-293-2215; email: bishop@usna.edu).

may have differing maximum velocities (as we may be using various types and sizes of UAV).

B. Capability Function

We must develop a functional representation of the capabilities of the system to achieve desired objectives (such as sensing). In this work, we assume the following capability function for sensing coverage from a single unit, measured at a discrete point in space:

$$f_{ij}(p_i, o_j) = \begin{cases} G_i \sin^M\left(\frac{\pi}{2}\frac{d_{ij}}{k_i}\right) & \text{when } d_{ij} < k_i \\ G_i \sin\left(\frac{\pi}{2} e^{A_i(k_i - d_{ij})}\right) & \text{when } d_{ij} \geq k_i \end{cases} \quad (1)$$

where

$$d_{ij} = \sqrt{\left(o_j^x - p_i^x\right)^2 + \left(o_j^y - p_i^y\right)^2 + \left(o_j^y - p_i^y\right)^2} \quad (2)$$

is the distance between vehicle i (at position p_i) and objective point j (at position o_j). G_i, A_i, M and k_i are dependent on the characteristics of the sensor system. This functional is smoothly differentiable with maxima at distance k_i.

C. Swarm-level objective function

The functional description of unit capability as measured at an objective point allows us to quantify swarm capability coverage of a specific point or area in the operational environment. We define functional coverage of an objective point o_j by summing the functional values of all UAVs:

$$V_j = \sum_{i=1:n} f_{ij}\left(p_i, o_j\right) \quad (3)$$

The complete set of m objectives and their capability function values will be given by:

$$V = \begin{bmatrix} V_1 \\ \vdots \\ V_m \end{bmatrix} \quad (4)$$

The goal of the control system will be to achieve desired values of the functional coverage function V_j for each objective point. The principle is that appropriate resources must be allocated to each objective point. Several, less capable units can accomplish the same functional coverage as a single, more powerful system. Further, some units may need to move closer to a desired target than others must to achieve appropriate functional levels. In the following section, we will discuss the control methodology that will accomplish this autonomously.

III. CONTROL SYSTEM

We base our control scheme on the control methodology from [1], based on redundant manipulator analogs (later referred to as *kinematic control* in [2]). This control method regulates swarm-level functions, such as mean and variance, while still allowing the individual units some degree of autonomy. This controller, based on redundant manipulator methods such as those discussed in [16], has proven to be an extremely effective technique for swarm control.

Given a platoon of n holonomic vehicles, the 3n-dimensional state is given by $q = [p_1 \ldots p_n]^T$. Given an m-dimensional platoon-level function of the state, $V(q)$, the unit velocities are related to function velocities (in the task space) by $\dot{V}(q) = J(q)\dot{q}$, where $J(q)$ is the Jacobian of V.

For a large platoon of robots, the number of state variables (x, y and z for each unit) is typically greater than the number of task variables (objective function values to cover). This redundancy creates an infinite number of possible configurations for the platoon while still achieving the desired task profile.

A. Primary Task Controller

The primary task controller is designed to achieve desired functional coverage at a set of objective points. The j[th] column of the Jacobian for V as defined by (3) and (4) is given by:

$$\frac{\partial V_j}{\partial p_i^x} = \begin{cases} -G_i M \sin^{M-1}\left(\frac{\pi}{2}\frac{d_{ij}}{k_i}\right)\cos\left(\frac{\pi}{2}\frac{d_{ij}}{k_i}\right)\frac{\left(o_j^x - p_i^x\right)\pi}{2d_{ij}} & \text{when } d_{ij} < k_i \\ A_i G_i \cos\left(\frac{\pi}{2} e^{A_i(k_i - d_{ij})}\right)\frac{\left(o_j^x - p_i^x\right)\pi}{2d_{ij}} e^{A_i(k_i - d_{ij})} & \text{when } d_{ij} \geq k_i \end{cases} \quad (5)$$

The other columns are equivalent and are excluded for brevity. Note that the derivative is smooth and continuous.

Although each unit is constrained by the platoon-level function, there is still a great deal of flexibility and autonomy. This flexibility can be characterized by the dimension of the null space of the Jacobian. Given a platoon with a total of $3n$ degrees of freedom and an m-dimensional task function $V(q)$, there are effectively $3n - m$ degrees of freedom left over after the main task is achieved. This number is the dimension of the null space of the system, which is a manifold of positions on which the main task $V(q)$ is achieved.

Gradient projection controllers allow systems with more degrees of freedom than task variables to coordinate motions in a way that maintains the swarm-level function regulation while using the additional degrees of freedom to carry out secondary objectives. The basic controller is given by:

$$\dot{q}^d = J^+(K(V^d(q) - V(q)) + \dot{V}^d(q)) + K_N(I - J^+J)v \quad (6)$$

where J^+ is the pseudoinverse of J given by $J^T(JJ^T)^{-1}$, K is a controller gain matrix, K_N is the null-space gain value, V^d is the desired task function trajectory, $(I - J^+J)$ is the projection operator that guarantees coordination, and v is an encoded secondary task. The secondary task is carried out on the null space of the primary task.

B. Secondary tasks

There are several secondary tasks of interest when using a swarm to carry out missions such as reconnaissance, etc. These include obstacle avoidance, kinematic constraint satisfaction, and avoidance of capability shadows (places where obstacles and environmental effects reduce the capabilities of the system).

1) Obstacle Avoidance

For our system, we will be concerned with avoiding collisions of three types: between robots in the swarm, between robots and the environment, and between the robots and the objective points. Each of these objectives can be handled by an appropriate artificial potential field (APF) [17] as follows. The inter-robot repulsion field v_r for the system is given by

$$\begin{bmatrix} v_r(i) \\ v_r(i+n) \\ v_r(i+2n) \end{bmatrix} = \sum_{\substack{j=1:n \\ j\neq i}} \begin{cases} \left(\frac{1}{\|(p_i-p_j)\|} - \frac{1}{\rho}\right)^2 \frac{(p_i-p_j)}{\|(p_i-p_j)\|} & when \quad \|(p_i-p_j)\| < \rho \\ 0 & when \quad \|(p_i-p_j)\| \geq \rho \end{cases} \tag{7}$$

where ρ is the maximum distance for effect from the vehicle repulsion. For obstacle avoidance (v_c) and avoidance of the objective point (v_o) (assuming that the robots should not occupy the same space as the target), we use a similar composition. The null-space component of the control is then the sum of all of the obstacle avoidance vectors. While it is possible to make obstacle avoidance primary, as in [2], we choose here to project the avoidance vector onto the high-dimensional null-space of the system. When obstacle avoidance is not compatible with the primary objective of achieving functional coverage, we can promote the avoidance aspect to primary. This is easy to detect by looking for vanishing null-space projections of the avoidance vector. Thus, under normal operation, the primary objectives drive the evolution of the swarm configuration.

Because we do rely on APF techniques, local minima of the secondary functions can become a problem [17], especially in very densely cluttered environments with many units. Many methods exist for recognizing and dealing with local minima in standard path-planning approaches, but this discussion is beyond the scope of this work.

2) Capability Shadows

Another significant issue in the use of capability functions to dictate robot motion and placement for objective achievement is that most of the capabilities in which we will be interested, including reconnaissance and perhaps even ordnance delivery, require line of sight (LOS). As such, we must enforce LOS constraints on the pose of the robots in the swarm relative to the objective points. For this work, we assume that the robots are performing reconnaissance missions at low altitudes near obstacles. As such, we use the following APF-like method to force the robot out of the back-projected capability shadow of the obstacles to force LOS on the objective point (the configuration is shown in Figure 2). An interesting approach to maintaining LOS between units in a swarm can be found in [18]

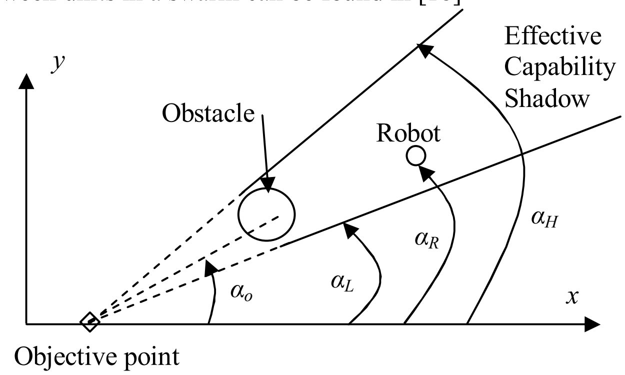

Figure 2: The robot is shown in the capability shadow for the objective point.

$$\alpha_E = \begin{cases} \alpha_R - \alpha_L & when \quad \alpha_R \leq \alpha_o \\ \alpha_H - \alpha_R & when \quad \alpha_R > \alpha_o \end{cases}$$

$$\alpha_d = \begin{cases} \alpha_o - \frac{\pi}{2} & when \quad \alpha_R \leq \alpha_o \\ \alpha_o + \frac{\pi}{2} & when \quad \alpha_R > \alpha_o \end{cases} \tag{8}$$

$$\begin{bmatrix} v_{LOS}(i) \\ v_{LOS}(i+n) \\ v_{LOS}(i+2n) \end{bmatrix} = \begin{bmatrix} \cos(\alpha_d)(\alpha_E)^3 \\ \sin(\alpha_d)(\alpha_E)^3 \\ 0 \end{bmatrix}$$

This parameterization of the shadow avoidance vector is applied only when the robot is in the shadow, and is designed to push the robot out of the region where its capability cannot be brought to bear on the objective point. The avoidance vector pushes the robot in a direction perpendicular to the objective-obstacle vector with angle α_o, which is defined slightly to one side of exact center to help alleviate problems caused by local minima in the avoidance field. The magnitude of the desired velocity is the cube of angle differential between the robot-objective vector and the nearest edge of the capability shadow, and the angle of that velocity is perpendicular to the objective-obstacle vector, again defined by α_o.

We do not consider the vertical aspect of shadow avoidance motion in this paper, working with the more restrictive assumption that the vehicles cannot fly over the obstacles. In practice, the effective capability shadow will be slightly larger than the actual obstacle would project, as shown in Figure 2, as the robots will need some free space to view around the obstacle. As such, we increase the width of the sensor shadow slightly, as shown. v_{LOS} is added to the avoidance APFs to generate a complete APF secondary objective. We note that obstacle avoidance will always dominate the APF as the vehicles approach a collision, since the v_{LOS} is a bounded function based on angle (and point again to methods by which obstacle avoidance can be made primary as necessary, through monitoring of the null-space projection). The sum of all of the APFs will be projected onto the null space of the primary Jacobian:

$$\dot{q}^d = J^+(K(V^d(q)-V(q))+\dot{V}^d(q))+K_N(I-J^+J)(v_r+v_c+v_o+v_{LOS}) \quad (9)$$

3) Kinematic constraints

While we consider holonomic models in this paper, we will enforce limitations on achievable velocity, as a method for demonstrating the capability of the control method to accommodate heterogeneous swarms as well as to account for issues that arise when individual units cannot achieve the initially-specified motions.

In [15], we developed a methodology for controlling nonholonomic swarms. The fundamental technique applied in that work accommodates any deviation between a desired and achievable unit motion. We apply the null-space projection method in that paper to enforce hard limits on individual robot speed in this work. This iterative method is outlined below for completeness:

- Compute $\dot{q}^d$ from (9)
- Set $v_{corr}(i) = 0$, $i \in [1, \ldots, 3n]$
- For j = 1:max iterations (or all velocities are achievable)
  - For i = 1:n

    Compute v_{corr}:

    $$\begin{bmatrix} v_{corr}(i) \\ v_{corr}(i+n) \\ v_{corr}(i+2n) \end{bmatrix} = \begin{bmatrix} \dot{q}^{best}(i) \\ \dot{q}^{best}(i+n) \\ \dot{q}^{best}(i+2n) \end{bmatrix} - \begin{bmatrix} \dot{q}^{d}(i) \\ \dot{q}^{d}(i+n) \\ \dot{q}^{d}(i+2n) \end{bmatrix}$$

    where $\dot{q}^{best}$ is the achievable velocity for the system closest to $\dot{q}^d$
  - Compute $\dot{q}^d := \dot{q}^d + (I - J^+J)(v_{corr})$

IV. Simulations

To test the efficacy of the proposed scheme, we perform simulated objective satisfaction tests using a swarm of four robots and two objective points. We use the following parameters for all of the simulations:

Objective points: $[1, 0, 0]^T$ and $[4, -2, 4]^T$
Number of swarm units: 4
Deployment of swarm: $q = [1\ 2\ 3\ 4\ 4\ 5\ 6\ 7\ 1\ 2\ 3\ 4]^T$
K (proportional gain): 0.2
K_N = 1000 (due to very high values for primary task)
$\rho = 0.5$

A. Example 1: Basic simulation: Homogeneous swarm

For the base simulation, we set the following parameters:
$A_i = 0.3$ for $i = 1 \ldots 4$
$G_i = 1$ for $i = 1 \ldots 4$
$k_i = 1$ for $i = 1 \ldots 4$
$V^d = [1, 1]^T$
Speed limit of 1.0 units/sec for every robot

We see the results of the simulation in Figure 3 – 4.

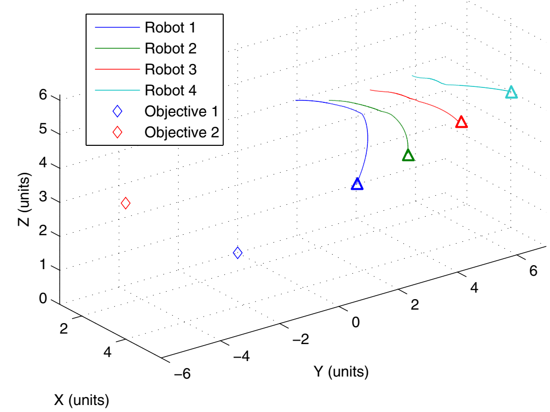

Figure 3: 3d motion, Example 1, initial positions are indicated by triangles. Note that the capability shadows are shown at ground level (z=0m) only.

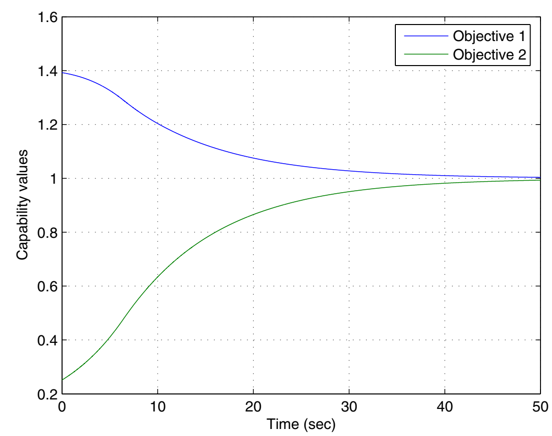

Figure 4: Capability measures for the two objective points

We see that the system achieves the desired capability measures at the two objective points. Further, the units clearly avoid collisions and do not exceed the maximum unit velocity. Note that the units are not really in a straight line at the end of this simulation, but rather that the 3D projection makes them appear that way.

B. Example 2: Heterogeneous swarm

In this simulation, we use all of the same parameters as Example 1, but we allow robot #1 to achieve higher speeds, setting the speed limit for only that robot to 2.0. We see the 3d motion in Figure 5. The capability coverage values V match the profile shown in Figure 4 exactly, as expected. We note that the abilities of the high-speed unit are utilized, resulting in less motion from the lower-speed units. Reducing the available speed for a unit also results in changes in 3d motion, but may also degrade the capability coverage profile slightly due to inability of the kinematic constraints to be satisfied for a desired set of unit velocities using the techniques from Section III.B.3 (as will be seen in Example 3b). We exclude simulations involving varying values of A_i, G_i and k_i for brevity, noting that the control method presented automatically adapts to such changes in sensing capability across units.

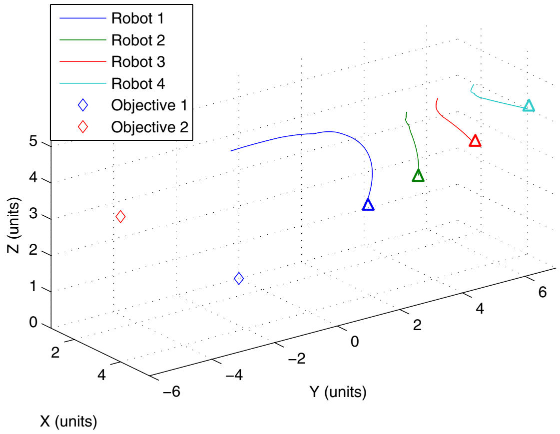

Figure 5: 3d motion, Example 2 [heterogeneous swarm]

C. *Example 3a & 3b: With obstacles and LOS control*

For Example 3a, we add to the setup in Example 2 a cylindrical obstacle with radius 0.5 at position [3, 2]. We see from the top-down view of the simulation in Figure 6 that the units fly out of the capability shadows to achieve coverage (full 3D view is shown in Figure 7). While it appears that robot 1 is following the shadow edge proximal to objective 1, it is actually moving in toward the objective while the other units are moving outward to achieve sensing on both targets. The capability coverage values V match the profile seen in Example 1 and 2 exactly.

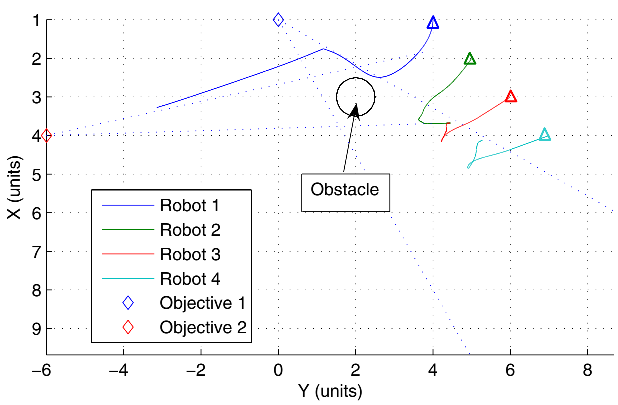

Figure 6: Top-down view of Example 3a. Shadow lines are dotted, and only influence motion distal from the projecting objective point.

Example 3b is identical except that robot #1 and robot #2 now have a limit on speed of 0.1 units/sec, while the other units retain their 1.0 units/sec limit. The results are shown in Figure 8. The capability coverage profile is shown in Figure 9, clearly degraded from the previous examples due to the changes in overall swarm capabilities. It is a testament to the effectiveness of the velocity re-distribution method that the deviation is not much more significant.

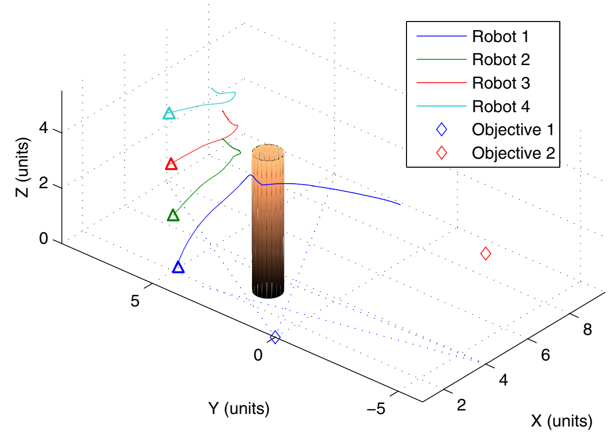

Figure 7: 3d motion, Example 3a

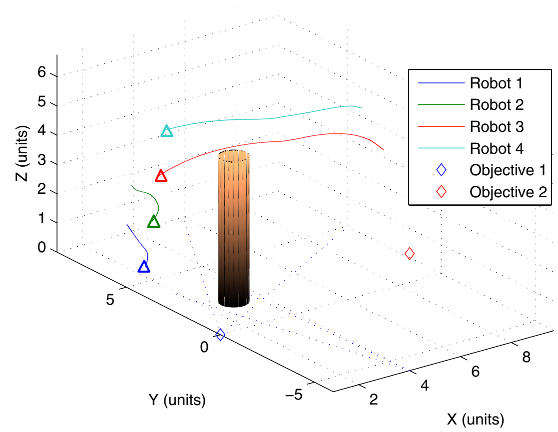

Figure 8: 3d motion, example 3b

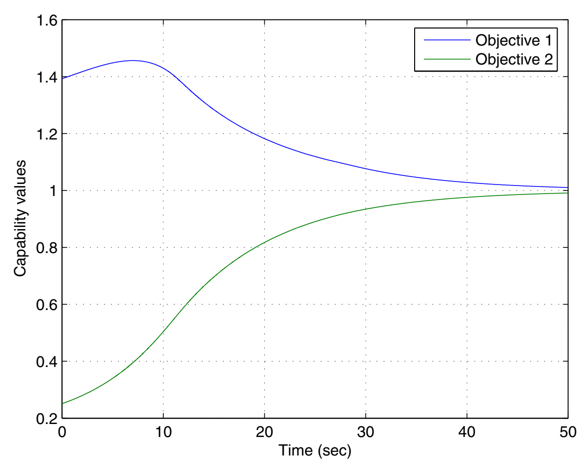

Figure 9: Capability coverage, Example 3b

D. *Example 4: High V^d values*

In this example, starting from Example 3a, we allow desired capability coverage values to exceed the maximum G_i, to demonstrate the coordinated nature of the controller. We set $V^d = [0.5, 2]$. We also modify the starting positions of the units from the line configuration to a more scattered initial deployment. The result, seen in Figure 10, demonstrates a much richer set of motions than the previous examples. However, all of the vehicles avoid obstacles, emerge from the capability shadow and achieve the desired capability coverage at the two objective points (see Figure 11).

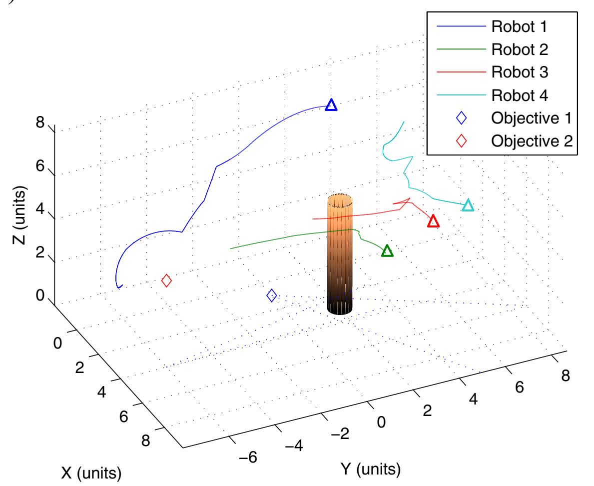

Figure 10: 3d motion, Example 4

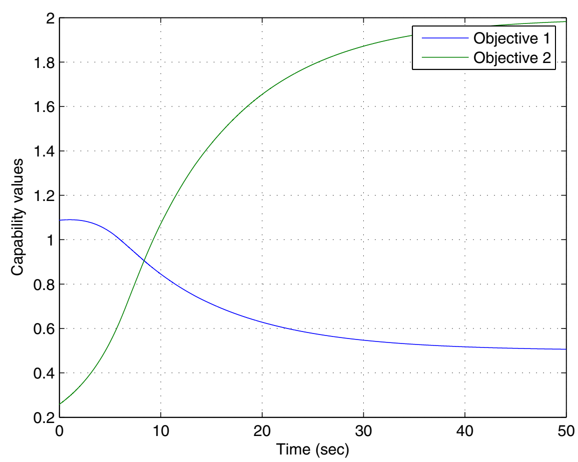

Figure 11: Capability coverage at the objectives, Example 4

V. Conclusion

In this work, we have presented a new formulation of swarm objectives and capabilities that allows for the use of standard swarm control techniques to achieve more abstracted objectives than those based entirely on position and swarm formation. We have developed a complete control system that achieves capability coverage for multiple objectives in the presence of obstacles. The system developed allows for heterogeneous swarms with varying sensing capabilities and kinematic constraints.

Future work involves application of these concepts to multi-modal swarms and varying capability modalities, as well as adapting this approach to nonholonomic systems such as those from [15], and utilizing the dynamics-based techniques from [19].

References

[1] B.E. Bishop, "On the Use of Redundant Manipulator Techniques for Control of Platoons of Cooperating Robotic Vehicles," *IEEE Transactions on Systems, Man and Cybernetics,* Vol. 33, No. 5, Sept. 2003, pp. 608 – 615.

[2] G. Antonelli and S. Chiaverini, "Fault tolerant kinematic control of platoons of autonomous vehicles," in *Proc. IEEE Int. Conf. on Robotics and Automation*, New Orleans, LA, 2004, pp. 3313–3318.

[3] C. Belta and V. Kumar, "Abstraction and control for groups of robots," *IEEE Trans. Robot.*, vol. 20, no. 5, pp. 865–875, 2004

[4] G. Yunyuan and W. Wei, "A New Multi-Robot Self-Determination Cooperation Method based on Immune Agent Network," *Proc. IEEE Int. Conf. on Robotics and Automation*, Barcelona, Spain 2005, pp. 392 – 397.

[5] E. Roszkowska, "Provably Correct Closed-Loop Control for Multiple Mobile Robot Systems," *Proc. IEEE Int. Conf. on Robotics and Automation*, Barcelona, Spain 2005, pp. 2821 - 2827.

[6] T. Balch and R. Arkin, "Communication in reactive multiagent robotic systems," *Autonomous Robots*, vol. 1, no. 1, pp. 27–52, 1994.

[7] X. Yun, "Line and circle formation of distributed physical mobile robots," *Journal of Robotic Systems*, vol. 14, no. 2, pp. 63–81, 1997.

[8] V. Gazi and K. M. Passino, "Stability analysis of swarms," *IEEE Trans. Automat. Contr.*, vol. 48, no. 4, pp. 692–697, 2003.

[9] N. E. Leonard and E. Fiorelli, "Virual leaders, artificial potentials, and coordinated control of groups," in *Proc. IEEE Conf. Decision and Control*, Orlando, FL, 2001, pp. 2968 – 2973.

[10] F. Zhang, M. Goldgeier, and P. S. Krishnaprasad, "Control of small formations using shape coordinates," in *Proc. IEEE Int. Conf. on Robotics and Automation*, Taipai, Taiwan, September 2003, pp. 2510–2515.

[11] A. D'Angelo and E. Pagello, "Making Collective Behaviours to Work through Implicit Communication", *Proc. IEEE Int. Conf. on Robotics and Automation*, Barcelona, Spain 2005, pp. 81 – 87.

[12] D.M. Stipanovic, G. Inalhan, R. Teo, and C.J. Tomlin, "Decentralized Overlapping Control of a Formation of Unmanned Aerial Vehicles," *Automatica*, Vol. 40, No. 8, pp. 1285-1296, 2004.

[13] J.L. Burke & R.R. Murphy, "Human-Robot Interaction in USAR Technical Search: Two Heads Are Better Than One, " *Proc. 2004 IEEE International Workshop on Robot and Human Interactive Communication,* Kurashiki, Okayama Japan, pp. 307- 312,

[14] J. Casper and R. Murphy, "Human-robot interaction during the robot-assisted urban search and rescue effort at the world trade center," *IEEE Trans. Syst. Man, Cybern. B,* vol. 33, pp. 367-385, June 2003.

[15] B. E. Bishop, "Control of Platoons of Nonholonomic Vehicles Using Redundant Manipulator Analogs," *ASME Journal of Dynamic Systems, Measurements and Control, Special Issue on Novel Robotics and Control*, Vol. 128, No. 1, March 2006, pp. 171 – 175.

[16] B. Siciliano, "Kinematic Control of Redundant Robot Manipulators: A Tutorial," *Journal of Intelligent Robotic Systems*, Vol. 3, pp. 201-212, 1990.

[17] J.-C. Latombe, *Robot Motion Planning,* Kluwer Academic Press, 1991.

[18] J.M Esposito and T.W. Dunbar,, "Maintaining wireless connectivity constraints for swarms in the presence of obstacles", *Proc. 2006 IEEE International Conference on Robotics and Automation,* May 15-19, 2006 Page(s):946 - 951

[19] B.E. Bishop, "Dynamics-based control of robotic swarms" *Proc. 2006 IEEE International Conference on Robotics and Automation*, May 15-19, 2006 Page(s):2763 - 2768