40th Southeastern Symposium on System Theory
University of New Orleans
New Orleans, LA, USA, March 16-18, 2008

Utilizing Navigational Capability to Control Cooperating Robotic Swarms in Reconnaissance-Based Operations

MIDN 1/C Evan A. Barnes
Bradley E. Bishop*
Weapons and Systems Engineering
United States Naval Academy
105 Maryland Ave (Stop 14A)
Annapolis, MD 21402
bishop@usna.edu

Abstract: This paper focuses on the development of a methodology by which multiple, heterogeneous units can deliver discontinuous capability (such as that from a computer vision system) to a specified set of targets. The centralized coordination controller commands the units such that a specified level of capability is delivered to a set of target locations while accommodating obstacle avoidance and a spectrum of additional secondary objectives. The efficacy of the controller is demonstrated through simulation.

Index Terms – Mobile robots, distributed systems, swarm robotics.

I. INTRODUCTION

While 21st century surveillance techniques have gradually relieved many of the dangers and pressures placed on a human being through the utilization of unmanned robotic platforms, the efficiency and effectiveness of such systems remains far short of the potential. The unmanned robotic vehicle's ability to carry out surveillance missions is unparalleled by any human, based on technological advances in electronic sensors. However, a single autonomous robot faces many limitations which decrease its cost efficiency and prevent it from being the best choice for various missions. On the other hand, a group of robots working together can remove many of the limitations faced by a single unit approach. However, controlling a group of robots is inherently more complex than controlling just one.

While the potential of a swarm of robots is virtually unlimited, there are still many questions concerning the proper control methodologies and swarm styles. Swarm robotics poses a difficult challenge with respect to control theory and data retrieval, including such aspects as centralization, communication, resource allocation, etc.

This work focuses on the use of a centralized "intelligent" swarm [1]. Robotic units in an "intelligent" swarm employ various sensors that allow them to cooperate, coordinate motion and communicate with other members of the swarm. The members of an "intelligent" swarm work together to accomplish a specific objective rather than acting as individual entities in a large group. They are typically able to sense each other and their surroundings and position themselves accordingly.

For centralized "intelligent" swarm controllers, there are two common methodologies used today: systems-theoretic and behavior-based. Systems-theoretic techniques guarantee strong system performance, and their control methodologies are more easily analyzed using standard mathematical tools. They involve the direct coding or command of all units in a swarm based on a desired system response for modeled stimuli. Therefore, every unit in the swarm is directly controlled to guarantee a specific outcome. However, in order for this to be a viable option for a certain mission, all of the information pertaining to the environment must be known, or at a minimum be highly predictable. Alternatively, behavior-based control approaches permit a swarm to automatically adapt to an unknown environment, and the control method is simpler in nature. The outcome of a behavior-based controller is inherently unpredictable and often gives rise to interesting emergent outcomes as a result of the swarm's interaction with its surroundings [2]. The tradeoff between these two approaches involves guaranteed performance versus system flexibility.

We base our approach to swarm control on the work of [2,3], wherein the swarm is treated as a single entity, and the task is defined as a differentiable function of the swarm state. Under this approach, techniques from redundant manipulator control were applied to achieve objectives such as control of mean position and swarm variance. This controller was successful in directing a swarm to a certain location; however, the swarm units are not in any way guaranteed to be able to use their active payload to achieve any specified task. Given a sensor with limited field of view, there is a strong possibility under this control that the swarm may be in the correct location but some of the units may be rendered useless due to their position and orientation to the target point(s).

This paper focuses on new methodologies for optimizing the performance of a heterogeneous swarm in reconnaissance based operations. By utilizing the concept of Navigational Capability, individual units are not directed based upon their location in a physical environment, but rather their ability to deliver their specific *capability* to a target location [4]. By quantifying a unit's ability to carry out its mission, our primary objective shifts from robot location to the robot's ability to influence and produce results: the mission, not the

formation. A basic example of capability delivery is the concept of a robot equipped with a light source. The capability of the robot revolves around its ability to deliver light to a certain location. This ability is quantified by measuring the intensity of the light on the target. Establishing a maximal intensity, based on the nature of the illumination source, the robot's ability to deliver light can be described as a percentage of the desired intensity at the target point(s). Converging on one location, the individual capabilities of an entire swarm can be summed up at the specified points to determine if the units are achieving the desired levels of illumination.

II. CAPABILITY FUNCTIONS

In this paper, we consider a simple 35mm computer vision system as the main enabling device for capability. In this section we outline the mathematical model for the capability and describe the method by which a smoothly differentiable artificial capability function can be generated for navigation purposes.

A. Pin-Hole Lens System

This pin-hole lens model is a standard in computer vision. The formula relating points in the camera frame to the image frame is given by:

$$ {}^{i}_{x}p = \frac{f * {}^{c}_{x}p}{{}^{c}_{z}p + f} \tag{1} $$

$$ D = {}^{c}_{z}p + f \tag{2} $$

where f is the focal length of the lens, ${}^{i}_{x}p$ is the distance along the x-axis in the image reference frame, ${}^{c}_{x}p$ is a distance along the x-axis in the camera reference frame, and D is the distance from the image plane to the obstacle or point in space.

B. Lens and System Parameters

Assuming a standard 35 mm lens system, the first step in the full modeling process was to develop an appropriate metric for capability delivered to a target point. Since the goal of the system is to return information on a certain target location, it was determined that the appropriate unit of measurement would be the pixel resolution (pixels/meter) at a certain distance from the lens system. The desired resolution, N_{opt}, was set to 400 pixels per meter, or .25cm per pixel. This gave us a balance between clarity in image and flexibility in movement. The robots would then be able to maintain a reasonable distance while still being able to return usable information on the objective location. In the image plane a resolution of 300 ppi (pixels per inch) was desired. This is a common standard for quality LCD image resolution and the pixel width, w, is equal to .0846 mm/pixel.

Once the basic controller parameters had been decided, (1) and (2) were used to create a basic proportional comparison where D is equal to the distance from the lens system, and f is the focal length of the lens system.

$$ N = \frac{f}{D * w} \tag{3} $$

One can now manipulate this equation for solve for the desired distance to achieve the desired pixel resolution. This distance is referred to as k and it is calculated by the following equation.

$$ k = \frac{f}{N_{opt} + w} \tag{4} $$

C. Initial Capability Functions

Once the desired distance has been established, the overall capability equations that represent the respective vision systems are developed. Using the previously determined k value, the following capability equation was generated. It was necessary to have two separate equations based on object distance because once an object is less than the desired distance away, pixel resolution becomes less of an issue and the field of view becomes the presiding limitation. The combination of these limited equations results in one differentially smooth curve.

$$ C_{ij} = f(D_{ij}) = \begin{cases} \sin\left(\dfrac{\pi * D_{ij}}{2 * k}\right)^{M_i}, & D_{ij} < k_i \\ \sin\left(\dfrac{\pi * N}{2 * N_{opt}}\right), & D_{ij} \geq k_i \end{cases} \tag{5} $$

In this equation, C_{ij} represents the overall capability at target j for unit i at the specified distance D_{ij}, while M_i and k_i depend on camera and lens parameters as well as the task. The total capability at target j is denoted C_j, and is given by the sum of the component capabilities at that point. The 2-Dimensional plot of capability for a single 35mm lens can be seen in Figure 1.

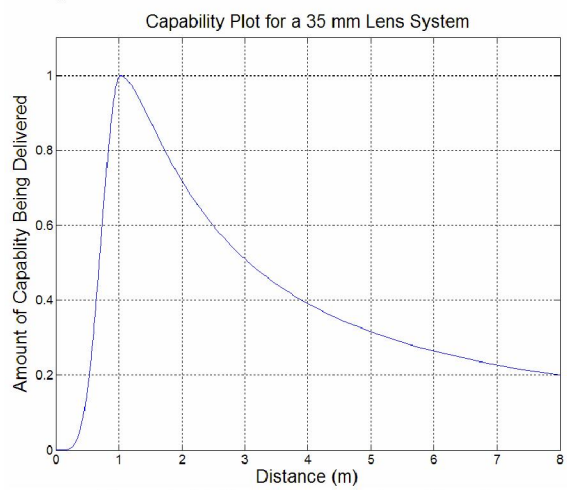

Figure 1: 2D Capability curve for a 35mm lens.

D. Deliverable Capability

The available *deliverable capability* for a unit depends on its exact payload. An example for a computer vision system is shown in Figure 2. In this figure, one can see the deliverable capability for a 35mm pin-hole lens system in polar

coordinates, where the capability is determined for distance from the camera and angle from the optical axis. The origin is the center of the lens and coordinates are given in the robot's sensor-fixed frame, which may be mobile with respect to the body frame of the robot in the case of an active sensor. The capability is defined as the pixels/meter of sensing delivered at the target point. The scale for C is normalized to a maximum value of '1', representing the unit's maximal capability as a percentage of desired capability at the target (here assumed to be 100% of this unit's available capability). The nonzero area shows the field of view of the camera and the shape of the level curve shows the drop off in capability as the target point moves away from the lens, as would be expected. When calculating the capability delivered on target, the 2D Capability curve was applied to an actual 3D space and the lens' field of view was taken into account.

Because the deliverable capability curve is not differentially smooth, another function must be used to control motion, with the actual deliverable capability used only to determine mission success.

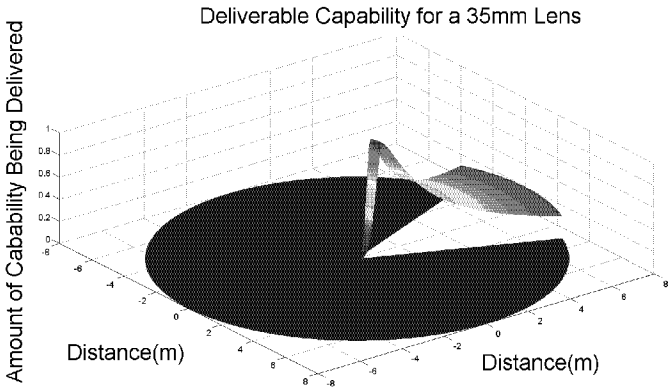

Figure 2: Deliverable Capability for a 35mm lens.

E. Navigational Capability

The difficulty in using the work of [2] and [4] with this sort of realistic capability is that the capability function in [4] must be continuously differentiable for the Jacobian-based methods to apply. As such, a new concept, *navigational capability* was developed which extends the actual deliverable capability of a system to provide a differentiable function with good characteristics.

By adding a limiting factor to the initial equation, the *navigational capability* was scaled from the *deliverable capability* based on the difference between the optical axis of the camera and the actual bearing to target. In this formulation, θ is the heading of the optical axis and α is the bearing to the target point from the robot's current position. The resulting equation is shown in (6):

$$C_{ij} = f(D_{ij}) = \begin{cases} \sin\left(\dfrac{\pi * D_{ij}}{2*k}\right)^{M_i} * 1/10^{\sin\left(\frac{\theta_i - \alpha_i}{2}\right)^4}, & D_{ij} < k_i \\ \sin\left(\dfrac{\pi * N}{2*N_{opt}}\right) * 1/10^{\sin\left(\frac{\theta_i - \alpha_i}{2}\right)^4}, & D_{ij} \geq k_i \end{cases} \quad (6)$$

This formulation provided us with a differentiable function covering a full 360° field of view and a distance of 0 to ∞, as seen in Figure 3. This capability is called the *Navigational Capability* for the individual robot. The reason for this is that the differentially smooth curve was used to calculate gradient responses to direct the desired motion of the robot to accomplish the primary task.

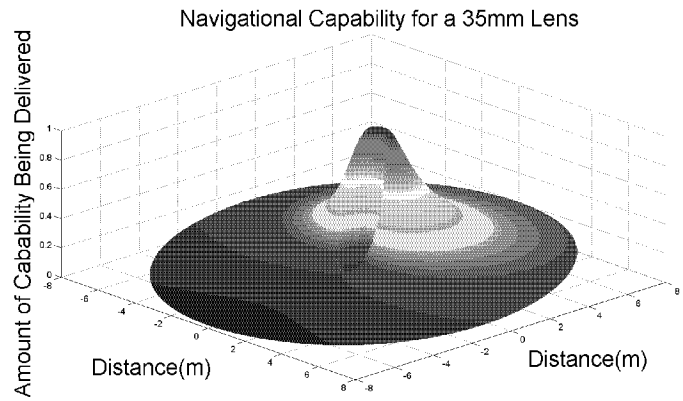

Figure 3: Navigational Capability for a 35mm lens

III. SWARM CONTROLLER

By using the *navigational capability* to direct the movement of the swarm unit, the controller is able to position the swarm in the correct position and monitor the amount of *deliverable capability* at a set of target points to ensure the primary objective is completed. However, there are numerous secondary objectives that must also be met, including: avoiding collisions (with obstacles and robots); allocating resources appropriately (including line-of-sight); low response time; and effective energy use. By using a hybrid systems-behavior control theory (based on [4]) the swarm is successfully able to complete this wide variety of secondary objectives. For the sake of brevity, the various methods used to complete the secondary objectives will not be discussed. Note, however, that while the primary objective pertains to the entire swarm as a single entity, some of the secondary objectives concern each individual unit's interaction with its surroundings and other members of the swarm. Without meeting the primary objectives, the swarm would not serve a purpose. However, it is the secondary objectives which must be met in order to make the swarm an efficient choice for any given mission.

A. Control Algorithm

For the primary objective, a systems-theoretic controller was used to provide a certain level of assurance for mission success, if possible (it is possible to ask for more capability on more targets than can be provided). The first step was to develop a Jacobian for a capability in an environment of m objectives and n robots. This is the primary task controller for the swarm and provides directional calculations to the individual members of the swarm based upon the capability

gradient. The Jacobian matrix is calculated by computing the partial derivative of each navigational capability function with respect to its individual variables. It represents how a change in any one of the robots poses will affect the capability at a target point. The robot state is (X_i, Y_i, θ_i), where θ_i represents the world-coordinate orientation of the optical axis for unit i.

$$J = \begin{bmatrix} \frac{\partial C_1}{\partial X_1} & \frac{\partial C_1}{\partial Y_1} & \frac{\partial C_1}{\partial \theta_1} & \cdots & \frac{\partial C_1}{\partial X_n} & \frac{\partial C_1}{\partial Y_n} & \frac{\partial C_1}{\partial \theta_n} \\ & \vdots & & \ddots & & \vdots & \\ \frac{\partial C_m}{\partial X_1} & \frac{\partial C_m}{\partial Y_1} & \frac{\partial C_m}{\partial \theta_1} & \cdots & \frac{\partial C_m}{\partial X_n} & \frac{\partial C_m}{\partial Y_n} & \frac{\partial C_m}{\partial \theta_n} \end{bmatrix} \quad (7)$$

The partial derivatives for the Navigation Capability functions were as follows, with target location j given by coordinates (X_{Tj}, Y_{Tj}):

$$\frac{\partial C_j}{\partial X_i} = \begin{cases} \frac{M\pi(X_i - X_{Tj})}{2D_{ij}k} * \sin\left(\frac{\pi D_{ij}}{2k}\right)^{M-1} * \cos\left(\frac{\pi D_{ij}}{2k}\right) * 1/10^{\sin\left(\frac{\theta_i - \alpha_i}{2}\right)^4}, & D_{ij} < k \\ \frac{-\pi k(X_i - X_{Tj})}{2D_{ij}^3} * \sin\left(\frac{\pi N}{2N_{opt}}\right) * \cos\left(\frac{\pi N}{2N_{opt}}\right) * 1/10^{\sin\left(\frac{\theta_i - \alpha_i}{2}\right)^4}, & D_{ij} \geq k \end{cases} \quad (8)$$

$$\frac{\partial C_j}{\partial Y_i} = \begin{cases} \frac{M\pi(Y_i - Y_{Tj})}{2D_{ij}k} * \sin\left(\frac{\pi D_{ij}}{2k}\right)^{M-1} * \cos\left(\frac{\pi D_{ij}}{2k}\right) * 1/10^{\sin\left(\frac{\theta_i - \alpha_i}{2}\right)^4}, & D_{ij} < k \\ \frac{-\pi k(Y_i - Y_{Tj})}{2D_{ij}^3} * \sin\left(\frac{\pi N}{2N_{opt}}\right) * \cos\left(\frac{\pi N}{2N_{opt}}\right) * 1/10^{\sin\left(\frac{\theta_i - \alpha_i}{2}\right)^4}, & D_{ij} \geq k \end{cases} \quad (9)$$

$$\frac{\partial C_j}{\partial \theta_i} = \begin{cases} -2 * \sin\left(\frac{\pi D_{ij}}{2k}\right)^M * \cos\left(\frac{\theta_i - \alpha_i}{2}\right) * \sin\left(\frac{\theta_i - \alpha_i}{2}\right)^3 * \ln(10) * \frac{1}{10^{\sin\left(\frac{\theta_i - \alpha_i}{2}\right)^4}}, & D_{ij} < k \\ -2 * \sin\left(\frac{\pi N}{2N_{opt}}\right) * \cos\left(\frac{\theta_i - \alpha_i}{2}\right) * \sin\left(\frac{\theta_i - \alpha_i}{2}\right)^3 * \ln(10) * \frac{1}{10^{\sin\left(\frac{\theta_i - \alpha_i}{2}\right)^4}}, & D_{ij} \geq k \end{cases} \quad (10)$$

$$D = \sqrt{(X_{Tj} - X_i)^2 + (Y_{Tj} - Y_i)^2} \quad (11)$$

Using standard redundant manipulator methods, (7) – (11) can be manipulated to find the desired changes in the swarm state as a function of desired change in capability. This allows the controller to achieve its primary objective by providing directional commands to each individual unit until the desired capability is reached:

$$\dot{q} = \begin{bmatrix} \dot{X}_1 & \dot{Y}_1 & \dot{\theta}_1 & \cdots & \dot{X}_n & \dot{Y}_n & \dot{\theta}_n \end{bmatrix}^T = J^+ * K_p \left(\begin{bmatrix} C_1^d \\ \vdots \\ C_m^d \end{bmatrix} - \begin{bmatrix} C_1 \\ \vdots \\ C_m \end{bmatrix} \right) \quad (12)$$

where K_p is a controller gain, C_j^d is the desired capability delivered to target point j, and $J^+ = J^T(JJ^T)^{-1}$ is the pseudoinverse of the Jacobian matrix.

In order to achieve secondary objectives, we use tasks encoded as artificial potential fields generated by the swarms interaction with its surrounding environment. These must be combined into one by projecting the potential fields onto the null space of the primary controller. The complete swarm controller is given in (13).

$$\dot{q} = J^+ * K_p \left(\begin{bmatrix} C_1^d \\ \vdots \\ C_m^d \end{bmatrix} - \begin{bmatrix} C_1 \\ \vdots \\ C_m \end{bmatrix} \right) + K_t * (I - J^+ J) * v \quad (13)$$

As one can see it is simply the primary controller with an added operation on the end. This mathematical operation is the projection of 'v' onto the null space. The variable 'v' is a vectorized sum of all of the unit level secondary objectives, which are defined purely on (X_i, Y_i). It includes unit level control for obstacle avoidance, robot avoidance, achieving line of sight and others. K_t is a system gain value that is used to control the amount of influence the secondary objectives have on the entire swarm. It is very important to establish an appropriate balance between K_p and K_t. If K_p is too high, the secondary objectives are overwhelmed and the potential fields generated will not be enough to prevent the robots from colliding into obstacles. In the end, the combination of these two controllers provides the swarm with the velocity vector, $\dot{q}$, which provides individual velocity commands to the units in order to accomplish the primary and all secondary objectives (assuming no deadlock, which can occur with artificial potential fields).

B. Results

In order to test the controller in a variety of different environments, several test beds were developed to perform the simulation. For brevity, only two of these tests are presented:

1. The robots each possess unique lens systems as well as different locomotive constraints, also on a single target point.
2. The robots converge on multiple target locations with varying desired capability levels.

The simulation results for each test bed can be seen in Figure 4 through Figure 6. The system gain values for M_i (for all units), K_p, and K_t are 5, 1, and 5 respectively.

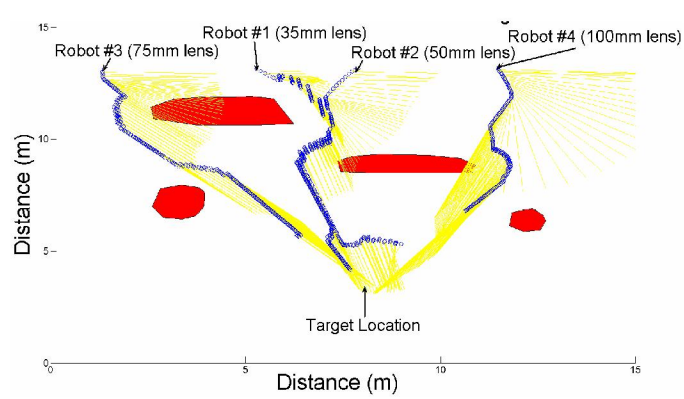

Figure 4: Test Bed #1

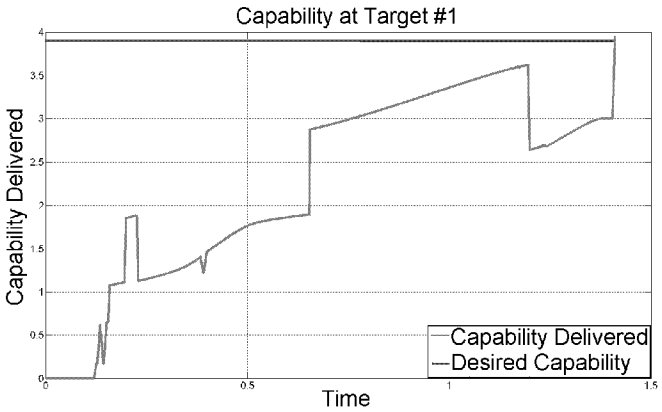

Figure 5: Delivered Capability for Test Bed #1

Test bed #1 requires a capability on target of 4, and each robot is equipped with a different lens. Therefore, the robots had to position themselves according to their specific capability. As seen in Figure 5, the delivered capability reaches the target value of 4.

Test bed #2 required a capability of 1 to be delivered to both targets, as shown in Figure 6. Once again, the test was successful. Each of the 4 robots delivered a level of capability of approximately .25 to target #1, while robots 3 and 4 also delivered a capability of over .5 each to target #2 (see Figure 7 and Figure 8). Thus, both targets were in the field of view of the systems on robots 3 and 4, and the sum of all delivered capability was at least 1 for each target.

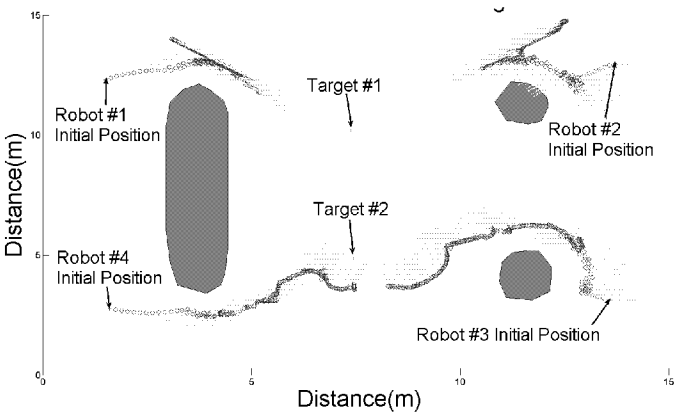

Figure 6: Test Bed #2

IV. CONCLUSIONS

In this work, we have demonstrated a method for controlling a swarm of heterogeneous robots delivering discontinuous capability (visual sensing) to a specified set of points in an obstacle-laden environment. The controller accounts for a variety of secondary objectives, including line-of-sight, velocity limits on units, and obstacle avoidance, although details were excluded for brevity.

The algorithm is being implemented on a hardware platform, using Khepera II mobile robots and simulated GPS to determine the efficacy of the controller in a real-world environment.

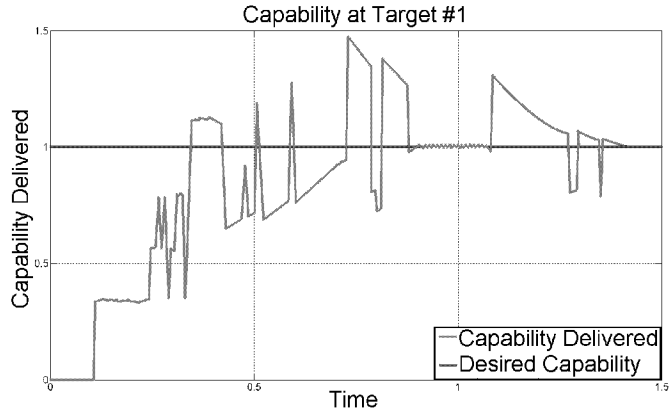

Figure 7: Delivered Capability at Target #1 for Test Bed #2

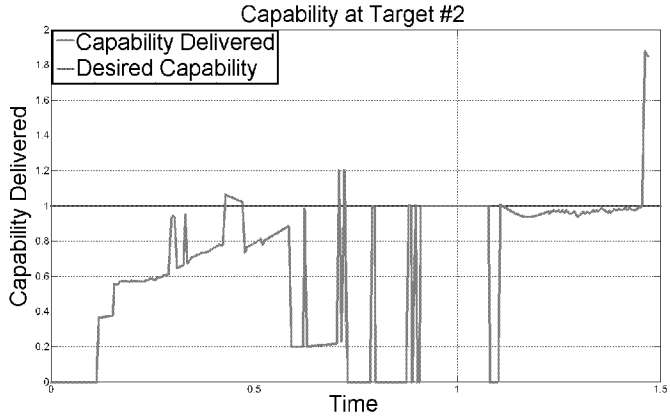

Figure 8: Delivered Capability at Target #2 for Test Bed #2

V. REFERENCES

[1] Goldbeck, J., "Evolving optimal parameters for swarm control" *NASA/DoD Conference on Evolvable Hardware,* 2002.

[2] Yong Chye Tan; Bishop, B.E., "Combining Classical and Behavior-Based Control for Swarms of Cooperating Vehicles" *Proceedings of the 2005 IEEE International Conference on Robotics and Autonomation*, 2005

[3] Antonelli, G., Chiaverini, S., "Kinematic Control of a Platoon of Autonomous Vehicles," *Proc. 2003 IEEE International Conference on Robotics and Automation*, pages 1464–1469, Taipei, Taiwan, Sept. 2003.

[4] Bishop, B.E., "On the use of Capability Functions for Cooperative Objective Achievement in Robot Swarms" Sept, 2006

[5] Magg, S. "Pattern Formation in Homogeneous and Heterogeneous Swarms: Differences Between Versatile and Specialized Agents" *2007 IEEE Symposium on Artificial Life.* April 2007.

[6] Barnes, L.; Alvis, W.; Fields, M.; Valavanis, K.; Moreno, W., "Swarm Formation Control with Potential Fields Formed by Bivariate Normal Functions" *14th Mediterranean Conference on Control and Automation*, 2006

[7] Bishop, B.E., "Control of Platoons of Nonholonomic Vehicles Using Redundant Manipulator Analogs" *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005

[8] Bishop, B.E., "On the use of redundant manipulator techniques for control of platoons of cooperating robotic vehicles" *IEEE Transactions on Systems, Man, and Cybernetics, Part A,* Sept. 2003.

[9] Y. Yang, O. Brock and R. A. Grupen, "Exploiting Redundancy to Implement Multi-Objective Behavior," *Proceedings of the 2003 IEEE International Conference on Robotics and Automation,* September 2003, pp. 3385 – 3390.