NAWCWD TP 8630

An Economical Framework for Verification of Swarm-Based Algorithms Using Small, Autonomous Robots

by
James Bobinchak,
Eric Ford,
Rodney Heil,
and
Duane Schwartzwald
Weapons and Energetics Department

SEPTEMBER 2006

NAVAL AIR WARFARE CENTER WEAPONS DIVISION
CHINA LAKE, CA 93555-6100

Naval Air Warfare Center Weapons Division

FOREWORD

The research described in this report was performed at the Naval Air Warfare Center Weapons Division during fiscal year 2006 as an attempt to apply swarm-based algorithms that were previously tested in simulation to hardware. The effort was supported by 6.1 funds from the Office of Naval Research, Code 03R.

The report was reviewed for technical accuracy by Sam Ghaleb.

Approved by
M. LAMBERT, Acting Head
Weapons Engineering and Integration Division
Weapons and Energetics Department
________________ 2006

Under authority of
W. M. SKINNER
RDML, U.S. Navy
Commander

Released for publication by
S. O'NEIL
Director for Research and Engineering

NAWCWD Technical Publication 8630

Published by .... Visual Information Division (TID)
Collation.. Cover, 10 leaves
First printing .... 20 copies

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, D.C. 20503.

1. AGENCY USE ONLY *(Leave Blank)*	2. REPORT DATE September 2006	3. REPORT TYPE AND DATES COVERED Research; 2006

4. TITLE AND SUBTITLE An Economical Framework for Verification of Swarm-Based Algorithms Using Small, Autonomous Robots (U)	5. FUNDING NUMBERS
6. AUTHOR(S) James Bobinchak, Eric Ford, Rodney Heil, and Duane Schwartzwald	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Air Warfare Center Weapons Division China Lake, CA 93555-6100	8. PERFORMING ORGANIZATION REPORT NUMBER NAWCWD TP 8630
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words)*

(U) We present an economical (~$6,000) framework for verifying, in hardware, swarm-based algorithms that were previously developed in computer simulation of large numbers of weapons engaging a plurality of highly maneuverable targets. This framework consists of a maximum of 10 small, autonomous, ground robots and an overhead vision tracking system that mimics both global positioning system (GPS) localization and peer-to-peer robot communications. Robots maintain a cohesive network formation by balancing a virtual system of interconnecting spring forces. The use of an optimal target-weapon pairing algorithm and interception methods enable weapons to intercept targets while minimizing global transit distance. Experimental results indicate that network formation occurs, on the average, in less than 25 seconds for a six-node robotic swarm. Thus our framework provides an economical, simple, quick, and reliable way of investigating the interaction among the mobile nodes of a robotic swarm using embedded algorithms.

14. SUBJECT TERMS		15. NUMBER OF PAGES
Distributed robotics	Swarm-based weapons	18
Overhead vision system	Target-weapon pairing	16. PRICE CODE
Swarm robotics	Virtual coupling	

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF ABSTRACT	19. SECURITY CLASSIFICATION OF THIS PAGE	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

CONTENTS

ACKNOWLEDGMENT

We thank the following people for funding related to this effort: Pam Overfelt, Alan Marshall, Alan Van Nevel, Brent Hedman, Russ Rudin, and Sam Ghaleb.

This report contains the results of the Navy portion of a collaborative effort between Raytheon Missile Systems and the Naval Air Warfare Center Weapons Division, under Cooperative Research and Development Agreement number NCRADA-NAWCWDCL-04-124, on which Mr. Sam Ghaleb was the Navy Principal Investigator.

INTRODUCTION

A controllable swarm of autonomous vehicles is a highly desirable tactical force. Vehicles in a swarm are extremely versatile, and can be customized to perform a variety of functions efficiently. They can make decisions autonomously based solely upon local sensing and peer-to-peer communications. Their autonomy allows for reduced reliance on communications between the swarm and the operator controlling the swarm, which in turn allows the operator to easily manage a large swarm of vehicles without the need to micromanage individual units. Swarms of autonomous vehicles are, in general, highly redundant and consequently able to survive their working environment. There is no hierarchical command and control structure, and therefore no common mode failure point or vulnerability. A swarm is also scalable; concepts that apply to a small swarm apply to a very large swarm because the maximum number of neighbors a unit can have is physically constrained.Due to the simplicity and small size of the components of a swarm, overall costs can be lower than a single large unit designed for the same task.

PURPOSE

In this paper we present an economical, reliable, and low-complexity framework for implementing swarm-based algorithms in small, autonomous, ground robots. We also present experimental results on two types of swarm-based algorithms: network formation and target-weapon pairing. These algorithms were previously used successfully in computer simulation of large numbers of weapons engaging many highly maneuverable targets (Reference 1), so there was interest in determining the feasibility of using them in actual hardware.

The Swarm Robotics Laboratory was established at the China Lake Naval Air Weapons Station for the purpose of testing and demonstrating swarm algorithms using hardware instead of computer simulation. Assumptions that hold true for computer simulations are often proven false when applied to hardware. In the interest of developing a product (a weapon system), the Swarm Robotics Lab is attempting to provide an intermediary step between simulating an algorithm in software and implementing it in a weapon system. With the application of simulation methods to an intermediate hardware test system, the methods can be adapted to work properly with hardware inconsistencies and real world variables so they then may be incorporated into a prototype weapon system.

RELATED WORK

Two academic institutions, CalTech and UCLA, have established robot laboratories on which we loosely have based our own. CalTech has a Multi-Vehicle Wireless Testbed (Reference 2) for investigating motion planning methods based on cooperative swarming models and virtual potential functions. However, their testbed is very costly, using four cameras and several image processing boards to implement the overhead vision tracking system and robots consisting of a Pentium III laptop computer mounted on a chassis with two ducted fans to propel the vehicle.

The Applied Mathematics Laboratory at UCLA uses a Micro-Car Testbed (Reference 3) for implementing a UAV-routing algorithm. Their testbed is based on the one constructed at CalTech, but uses only two cameras to implement the overhead vision system, which keeps the cost of the system low. The micro-cars are controlled via RF link by a dedicated computer that uses information from the overhead vision system to determine how to move the cars around the arena. Moving the on-board processing to a dedicated off-board computer to control the cars allows the construction of the cars to be very simple and inexpensive. However, the micro-cars are controlled entirely by the central computer, eliminating their autonomy.

Our laboratory attempts to implement the best aspects of both CalTech and UCLA Labs. The idea was to keep the cost and complexity of our framework low while allowing for future expansion when the funds became available.

OVERHEAD VISION TRACKING SYSTEM

The overhead vision tracking system consists of a monochrome camera equipped with a 64 degree field-of-view (FOV) lens mounted at a height of 10 feet above the floor, which yields a visible region of approximately 12.5 feet x 9.4 feet, the entire arena. Each robot in the arena is uniquely identified using a 6-inch x 8-inch barcode attached to its top. Arena video is collected at 30 frames per second and streamed via FireWire to a 2.2 GHz Dell desktop PC running Windows 2000. The computer processes the video to identify the position and orientation of each barcode. Barcode position and orientation are encoded in a global positioning system (GPS) data packet that is sent over an RS232 serial interface to an RF transmitter that broadcasts the data packet over the arena. Due to limitations in transmitter bandwidth and robot microprocessor speed, the GPS data packets are sent at a rate of 2 packets per second per robot, so that in an arena occupied by a maximum of 10 robots the packet transmission rate is 20 packets per second. A receiver on each robot intercepts the RF transmission and the data is sent to the robots' microprocessor, which decodes the packets to obtain the positions and orientations of all the robots in the arena. Since each robot now possesses information about the position and orientation of all the other robots in the arena, we have effectively achieved distributed information sharing (which is essential for robot cooperation) without actually having to implement robot-to-robot communications. Thus, the vision system provides both a GPS localization signal and a means for easily emulating peer-to-peer network communications. See Figure 1.

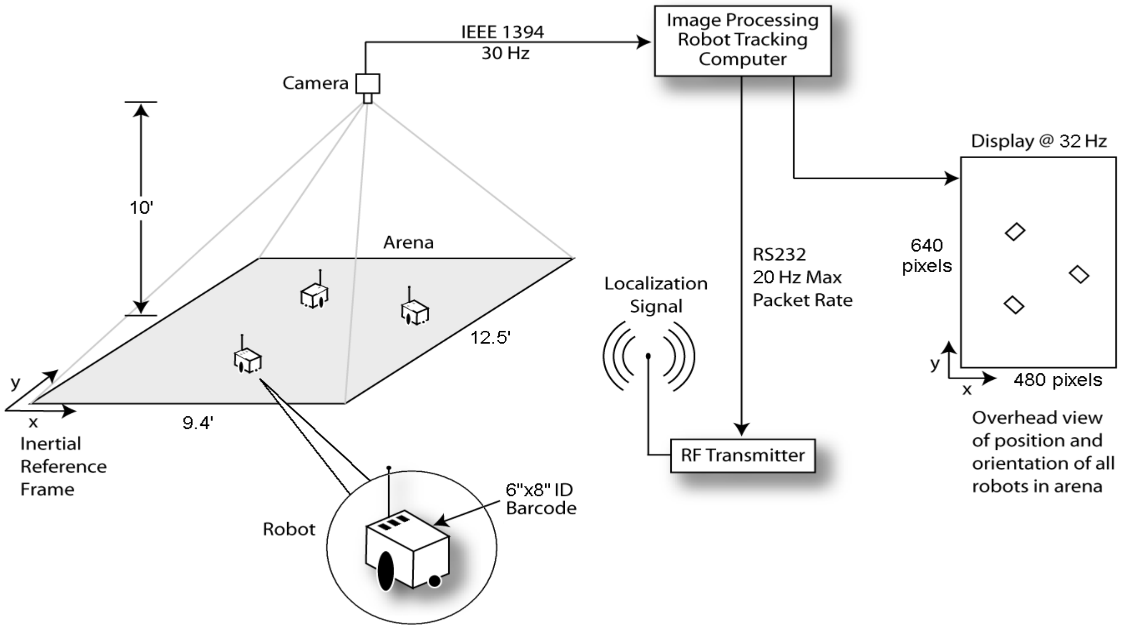


FIGURE 1. Overhead Vision Tracking System.

ARENA

The arena is approximately 12.5 feet long along the y-axis and 9.4 feet wide along the x-axis. Black rubber matting covers the floor of the arena to reduce the glare from overhead lighting and to improve the traction of the robots. A 1/4-inch CCD, progressive scan monochrome camera (DMK 21F04) suspended from the ceiling provides visual coverage of the entire arena that is sent at 30 frames per second to the computer that serves as control console. The camera focal-plane-array (FPA) is 640 x 480 pixels. When equipped with a 1/3-inch format lens (L28CSWI) the camera provides a nominal FOV of 68 degrees along the y-axis of the arena, though the actual value is about 64 degrees.

UNITS OF MEASURE

With the relative sizes of the arena and the camera field of vision, there are approximately 4.3 pixels per inch. Due to camera lens barrel distortion, this number varies slightly from the edges of the image to the center; however, the difference is negligible because the resolution varies by no more than 10 percent across the entire image. Given that the y-axis of the arena covers 640 pixels, and that a robot’s y-axis position must fit into an 8-bit data type, the use of pixels as coordinates to represent position proved ineffective. In order to represent each coordinate in a single byte and thereby minimize transmission time, the pixels were scaled so the total arena length could be represented as 256 units. The resulting unit is referred to as a “dot.” Therefore the arena size is 256 x 192 dots, where one dot is approximately 0.6 inches or 3 pixels.

ROBOTS

Our robots are modified Parallax BOE-Bots®. The unmodified BOE-Bot is available from Parallax for approximately 170 dollars. Two continuous-rotation wheel servos allow for variable speed movement in forward and reverse directions, scalable turns, and stationary pivots. At full speed, the robots are capable of moving approximately 6 inches, or 10 dots, per second. In place of the default Basic Stamp™, a Javelin™ microcontroller with more onboard memory was installed. A separate circuit board added to the front of the robot supports a speaker which relays audible error messages and system feedback, a 555 timer circuit which provides the robot with a unique identification, an array of three ultrasonic rangefinder modules which can detect obstacles to the nearest inch, and an Abacom AM-RTD-0315 RF Receiver which wirelessly receives position and command data from the control console. With the demands of the additional hardware, a battery pack was designed to support eight AA rechargeable batteries. A robot can run continuously on a single battery pack for several hours before a decrease in voltage causes a negative impact on performance. The total cost for robot and additional hardware is approximately $350. See Figure 2.

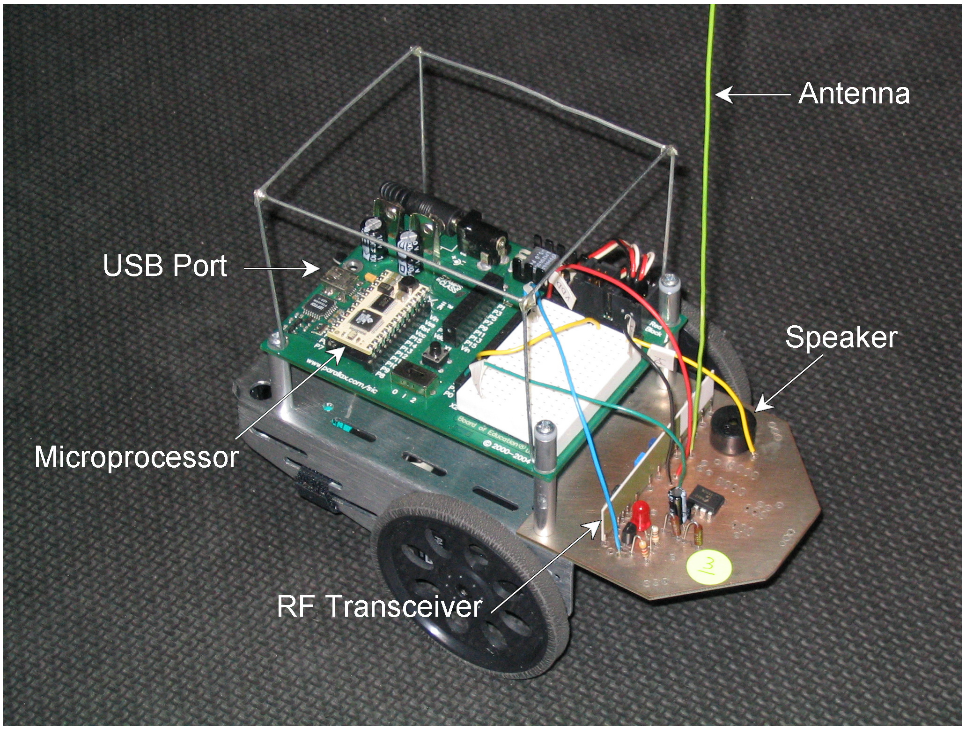


FIGURE 2. Robot Hardware.

BARCODES

Each robot is equipped with a barcode for identification of an individual robot, its spatial position, and its heading. The barcode consists of a white card with a series of black squares against one end of the card. Each of these cards has a unique number of squares, from one to ten. A robot's identification number is represented by the quantity of squares on the card. Additionally, a robot's current angle of orientation is determined by the edge of the card closest to the arrangement of squares. For this reason, there can be no robot with a zero for identifier; such a robot would have no information regarding its current heading. A robot's position within the arena is determined by the location of the center of the card's area. See Figure 3.

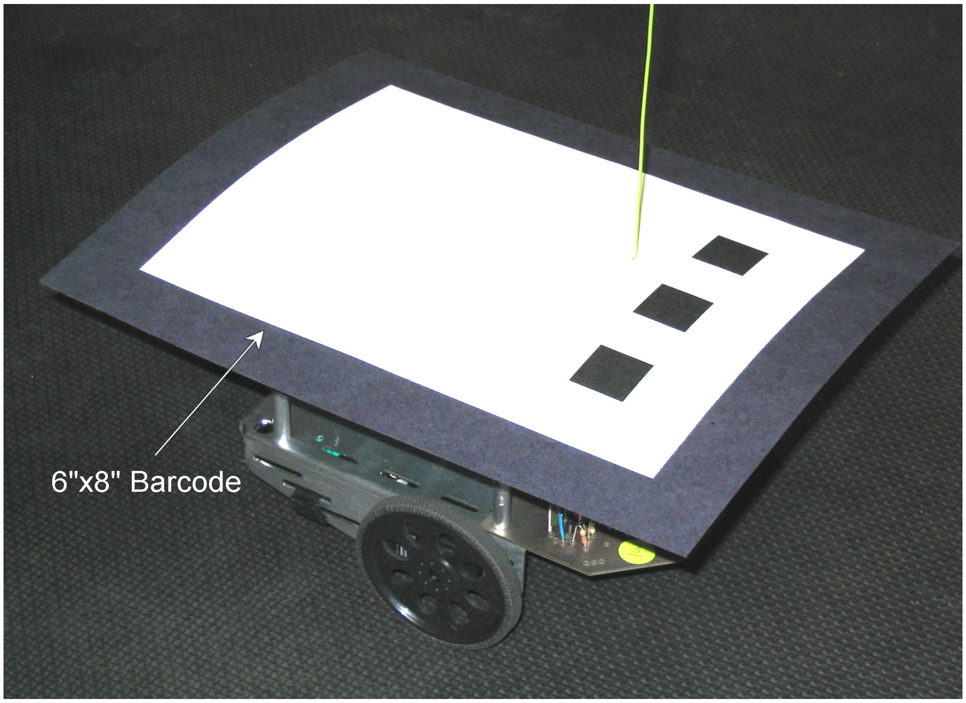


FIGURE 3. Robot With Barcode.

IMAGE PROCESSING MODULE

The video stream is refreshed at a frequency of 30 Hz and is analyzed by image processing software (OpenCV), which first locates all white rectangles in the arena in terms of the x- and y-coordinates of their centers. Then the software determines the number of black squares within each rectangle, thereby establishing each robot's identity and position. Based on the center of gravity of the squares in relation to the center of the rectangle, the software can determine the heading of each robot. Finally, the image-processing module passes the robot ID, the x- and y-coordinate position, and the heading angle for each robot to the serial port driver.

SERIAL PORT DRIVER

The serial port driver sends the data packet to an RF transmitter. The data is formatted in 6-byte packets. The first byte contains a four-bit representation of the robot ID. The second and third bytes contain the x- and y-coordinates. The fifth byte carries heading angle information. The checksum for this packet is held in the fourth byte. The sixth byte is always zero to indicate the end of the packet.

COMMUNICATIONS

Current communication protocol in the arena can support up to ten robots. Communication serves a threefold purpose. First, it simulates GPS, permitting each robot to know its own location. Secondly, it simulates peer-to-peer communications, permitting each robot to know the other robots' locations. Finally, it allows the human operator to send commands to the swarm, or to individual robots.

TRANSMITTER

An Abacom AM-RTD-0315 RF Transceiver connected to the serial port transmits GPS and command packets to the swarm. Operating at a frequency of 315 MHz, the transceiver is capable of data transmission rates up to 10 Kbps, and an output power of 1 mW.

RECEIVER

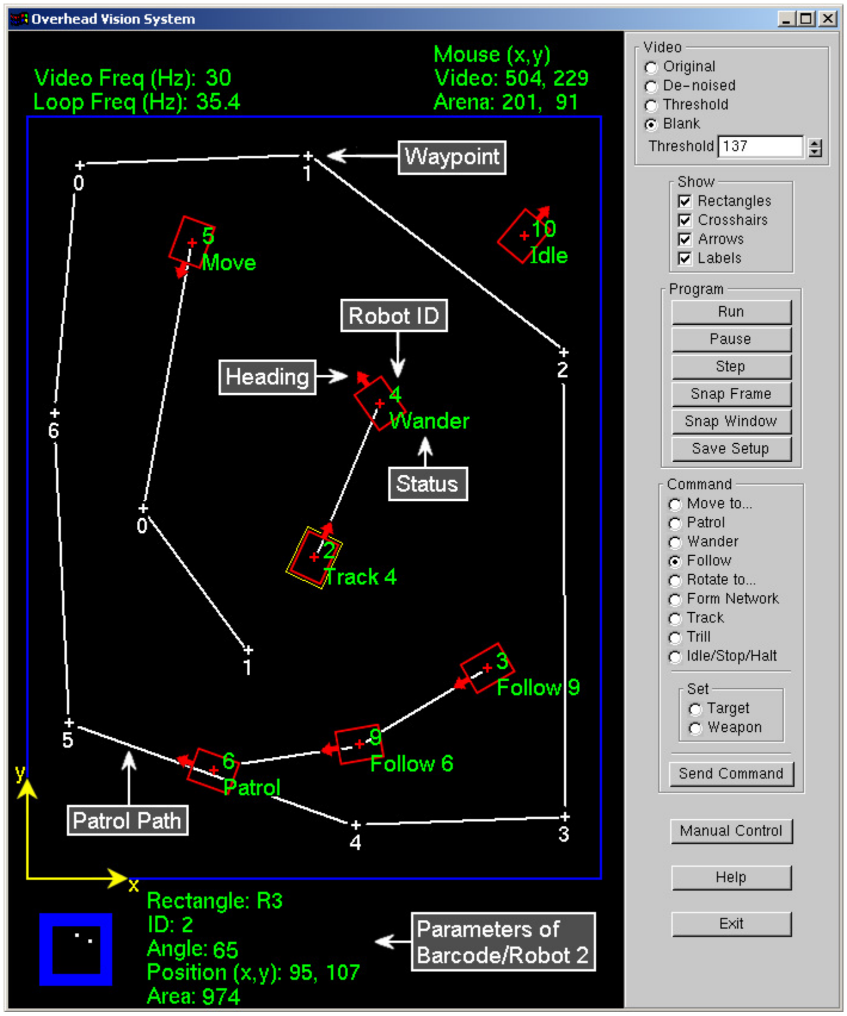
Robots are equipped with an Abacom AM-RTD-0315 RF Transceiver for receiving GPS and objective information from the control console. The Javelin microcontroller has a 256 byte UART buffer for receiving the data from the RF receiver. Although the UART is capable of more than 9600 baud, continuous transmission of data at this rate causes the buffer to overflow. Native software on the Javelin microcontroller could not process the incoming data fast enough to keep the buffer from exceeding its capacity. As a result, the protocol was modified to transmit data in short 500 mS increments while maintaining 9600 baud within each increment. The result is a data transmission rate of 2 packets per second per robot. To prevent buffer overflow the maximum data rate was set at 20 packets per second, which allows a maximum of 10 robots to occupy the arena simultaneously. However, due to susceptibility of the AM transceivers to interference from the servos and other external noise sources, the arena ideally supports six robots.

SWARM CONTROL SYSTEM

In addition to transmitting GPS packets, the vision system computer doubles as a control console, which transmits high-level objectives to the robots in the form of a 6-byte command packet. These objectives are entered using the graphical user interface and are received and interpreted by the robots which, in turn, execute the objective autonomously using an on-board software algorithm.

GRAPHICAL USER INTERFACE (GUI)

The graphical user interface, as illustrated in Figure 4, allows an operator to monitor and coordinate the robots and to establish manual control if necessary. Either the raw camera image or a filtered image can be displayed in the GUI window, showing the locations of the robots within the arena. The robot identification number and the most recent command sent to that robot are shown for each robot. The interface allows a variety of objective commands to be queued for transmission to a robot: multiple waypoint movement, pursuit and interception of other robots, stationary tracking of another robot, and target and weapon designations for target-weapon pairing studies. Manual control of forward and reverse motion, as well as rotation, is available for individual robots, however this is merely a convenient way of controlling a robot that has inadvertently moved outside the arena, not for remotely controlling the members of the swarm. In the same manner as the GPS data, the commands are sent via the RF transmitter in 6-byte packets. The first byte represents the robot ID number. The second byte specifies which objective the robot should execute while the third and fifth bytes serve as parameters for a particular objective. The interface also allows creation of a log file of all robot locations during the duration of a test, so that trajectories can be plotted in post-test analysis.



Overhead Vision System

Video Freq (Hz): 30
Loop Freq (Hz): 35.4

Mouse (x,y)
Video: 504, 229
Arena: 201, 91

Rectangle: R3
ID: 2
Angle: 65
Position (x,y): 95, 107
Area: 974

Video
○ Original
○ De-noised
○ Threshold
● Blank
Threshold 137

Show
☑ Rectangles
☑ Crosshairs
☑ Arrows
☑ Labels

Command
○ Move to...
○ Patrol
○ Wander
● Follow
○ Rotate to...
○ Form Network
○ Track
○ Trill
○ Idle/Stop/Halt

Set
○ Target
○ Weapon

FIGURE 4. Control Console GUI.

ROBOT AUTONOMOUS CAPABILITIES

While each robot receives its objective from the GUI, it interprets and executes that objective autonomously using serial multi-tasking. One task checks the UART buffer for a new packet (either GPS or Command) and extracts the individual bytes, while the other executes the extracted command or updates the robot with the extracted GPS data. For a movement command, the robot calculates the relative angle measure between its current heading and the destination and then adjusts its course appropriately by scaling the speed of one of its wheel servos. A wheel servo moves full speed forward when the angle difference is close to zero and decrements to a full stop as the angle difference approaches 45 degrees, resulting in a turn. As the angle difference approaches 90 degrees, the servo increases speed in the reverse direction, with angle differences of greater than 90 degrees causing a full stationary pivot. This method also allows for minor course corrections to counter inconsistencies in the wheel servo calibration. In a pursuit situation, the robot continuously calculates its trajectory from the GPS coordinates of the robot it is following and adjusts its movement accordingly.

COLLISION AVOIDANCE

Collision avoidance is a necessary component of swarm behavior. Robots must be able to move simultaneously in a group formation without interfering with each other. With the angle and distance to an obstacle, a robot is able to adjust its course to navigate smoothly around the obstacle with minimal divergence from its course, even if the obstacle is also in motion.

NETWORK FORMATION

Network formation allows a robotic swarm to arrange itself into a stable and uniform configuration. The formation of a network occurs through the use of separation and cohesion. This effect is achieved with “virtual coupling”, a virtual system of spring forces that connect a robot with all other robots within its “local neighborhood” or immediate surroundings. A rest distance is established within the equation representing the spring in a non-compressed state that serves as the desired spacing between any given robots. When two robots are closer to each other than the rest distance, the spring is compressed and the resulting force pushes the robots apart. Conversely, when two robots within a local neighborhood are farther apart than the rest distance, the elasticity of the spring pulls the robots closer together. When properly balanced, the system of connections between the individuals in a group of robots forces the robots to arrange themselves in a cohesive, equidistant arrangement, typically a series of equilateral triangles, as shown in Figure 5. Network formation occurs in two cases: convergent and divergent.

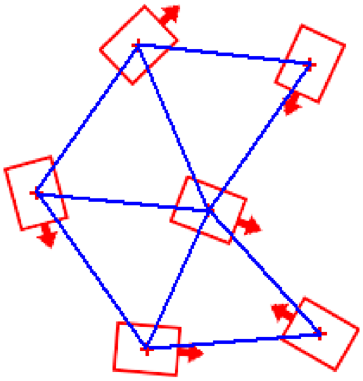


FIGURE 5. Stable Network Formation.

CONVERGENT NETWORK FORMATION

In the convergent case, the robots begin separated from each other and converge toward the center of the arena. As the robots approach the center of the arena, they begin to enter each other's local neighborhoods and alter their respective courses. An ad-hoc network forms near the center of the arena as the robots' virtual springs reach their rest distances. The total network formation time is comprised of not only the time for networking, but also the transit time as the robots proceed toward the center of the arena. Refer to Figure 6 box plots (Reference 4) for observed convergent network formation times for two to six robots. Each box plot represents ten trials.

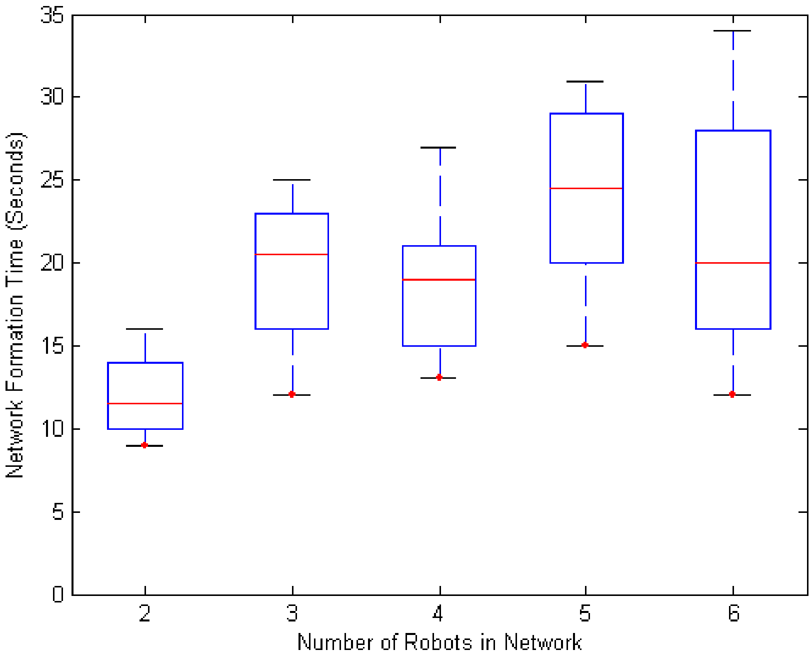


FIGURE 6. Observed Convergent Network Formation Times vs. Network Size.

DIVERGENT NETWORK FORMATION

In the divergent case, the robots begin in a closely spaced cluster and diverge to form an ad-hoc network, due to the repulsion caused by the compressed virtual spring network. The total network formation time is representative of time to form a network because the robots begin within each other's local neighborhood and, as a result, there is no independent transit time. Refer to Figure 7 for divergent network formation times for two to six robots. Each box plot represents ten trials.

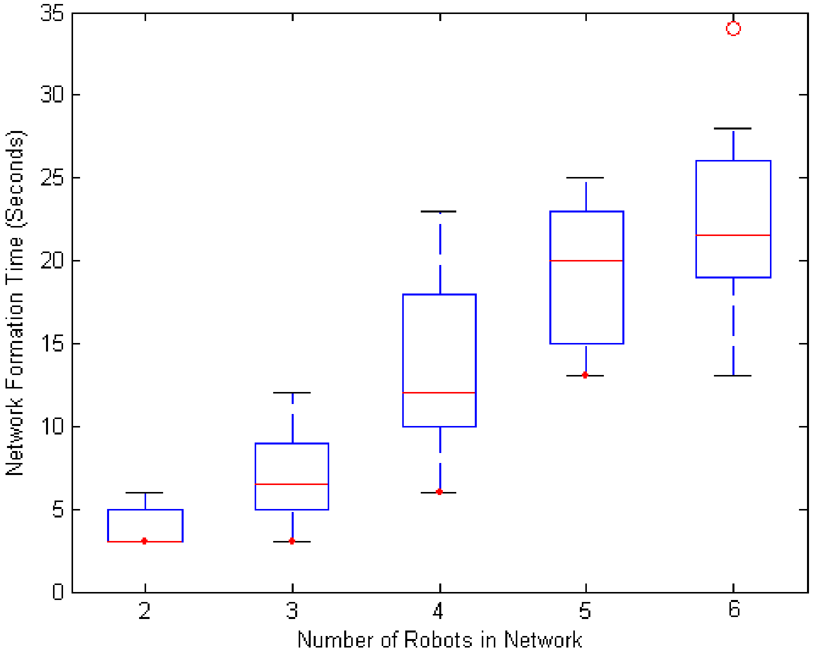


FIGURE 7. Observed Divergent Network Formation Times vs. Network Size.

TARGET-WEAPON PAIRING

Consider the asymmetric multi-assignment problem, where we want to assign n weapons to m targets. Each weapon is capable of intercepting no more than one target; however each target may be attacked by more than one weapon. The probability that a weapon can intercept a target is used as the cost benefit for pairing weapons with targets. A table of probabilities is generated for every possible target-weapon combination and the goal is to determine the optimum target-weapon assignment.

SOLUTION TO THE ASYMMETRIC MULTI-ASSIGNMENT PROBLEM

Once a table of possible target-weapon intercept probabilities is generated, an assignment algorithm is used to maximize the global probability of intercepting all targets. The actual linear programming problem to be solved is

$$\text{maximize} \sum_{(i,j)\in A} a_{ij}x_{ij}$$

subject to

$$\begin{aligned} 1 \le \sum_{j \in \mathrm{A}(i)} x_{ij} \le \alpha_i \quad & \forall i = 1,\dots,m \\ \sum_{i \in \mathrm{B}(j)} x_{ij} = 1 \quad & \forall j = 1,\dots,n \\ 0 \le x_{ij} \quad & \forall (i,j) \in A \\ \sum_{i=1,\dots,m} \alpha_i \ge n & \end{aligned}$$

where

x_{ij} = decision variable (0 or 1)

$\mathrm{A}(i)$ = set of weapons to which target i can be assigned

$\mathrm{B}(j)$ = set of targets to which weapon j can be assigned

A = set of all possible pairs (i, j)

a_{ij} = probability of weapon j intercepting target i

α_i = upper bound on the number of weapon to which target i can be assigned

m = total number of targets

n = total number of weapons

This problem states that the global probability of intercept must be maximized, while ensuring that every target i is assigned to at least one weapon, but no more than α_i weapons, and every weapon j is assigned to exactly one target. Because α_i is an upper limit on the assignment, this is a constrained multi-assignment problem. To generate an unconstrained multi-assignment problem, let $\alpha_i \to \infty$.

Using duality theory, the unconstrained multi-assignment problem becomes

$$\text{minimize} \quad \sum_{i=1}^{m} \pi_i + \sum_{j=1}^{n} p_j + (n-m)\lambda$$

subject to

$$\begin{aligned} \pi_i + p_j \ge a_{ij} \quad & \forall (i,j) \in A \\ \lambda \ge \pi_i \quad & \forall i = 1,\dots,m \end{aligned}$$

where

π_i = profit of target i

p_j = price of weapon j

λ = maximum profit

One method of solving the unconstrained multi-assignment problem is the forward/reverse auction algorithm proposed by Bertsekas (Reference 5). Because the robot's microprocessor does not have the required memory capacity or computational capability to run the auction algorithm, it is instead implemented on the computer and the results of the assignment are then transmitted to the robot. See example of asymmetric unconstrained multi-assignment using four weapons and two targets in Figure 8.

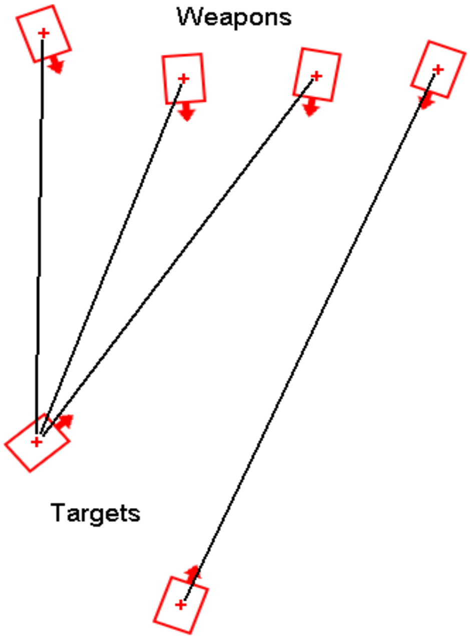


FIGURE 8. Four vs. Two Target-Weapon Pairing.

PURSUIT AND INTERCEPT

Once targets have been assigned to available weapons, two methods are possible for weapon guidance. A weapon can pursue the target, continuously directing itself toward the target's current location. A weapon can also attempt to intercept the target, to direct itself toward the position where the target will hypothetically be located at the time the weapon will reach that position.

PURSUIT

In pursuit mode, the weapon will always move toward the target's current location. By definition, a weapon in pursuit mode will typically achieve a condition where it is approaching the target from behind. If the target is moving on an orthogonal vector to the weapon, the weapon will ultimately execute a broad arc at the expense of time and energy. See Figure 9.

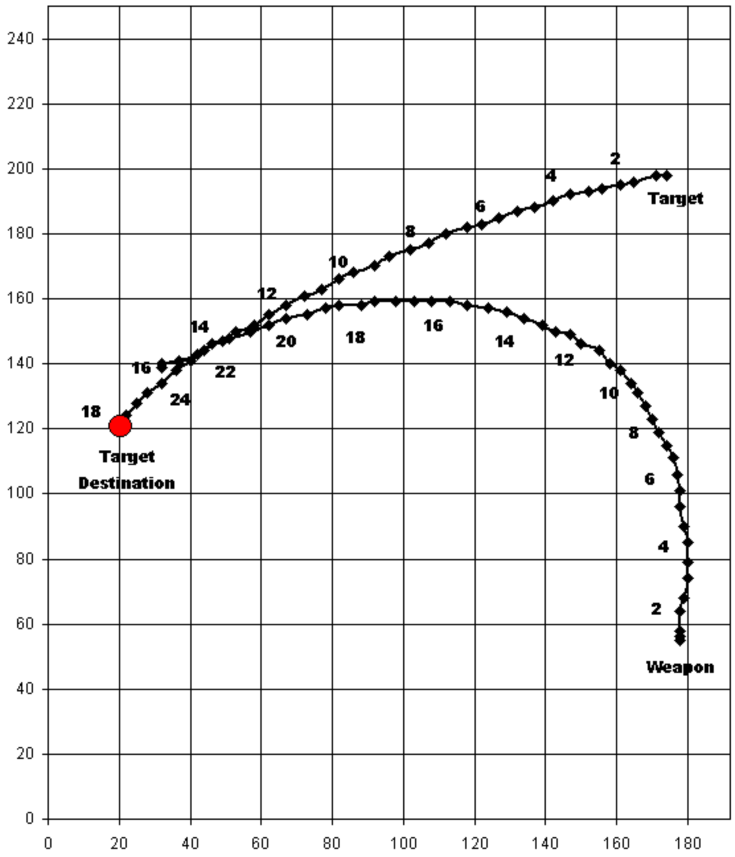


FIGURE 9. One vs. One Pursuit Mode.

INTERCEPT

With the intercept navigation system used in the arena, weapons calculate the target's current trajectory using its previous and current positions. Using the calculated speed of the target and its own speed, the weapon determines the probable intercept location and adjusts its trajectory, minimizing the time from assignment to target intercept. See Figure 10.

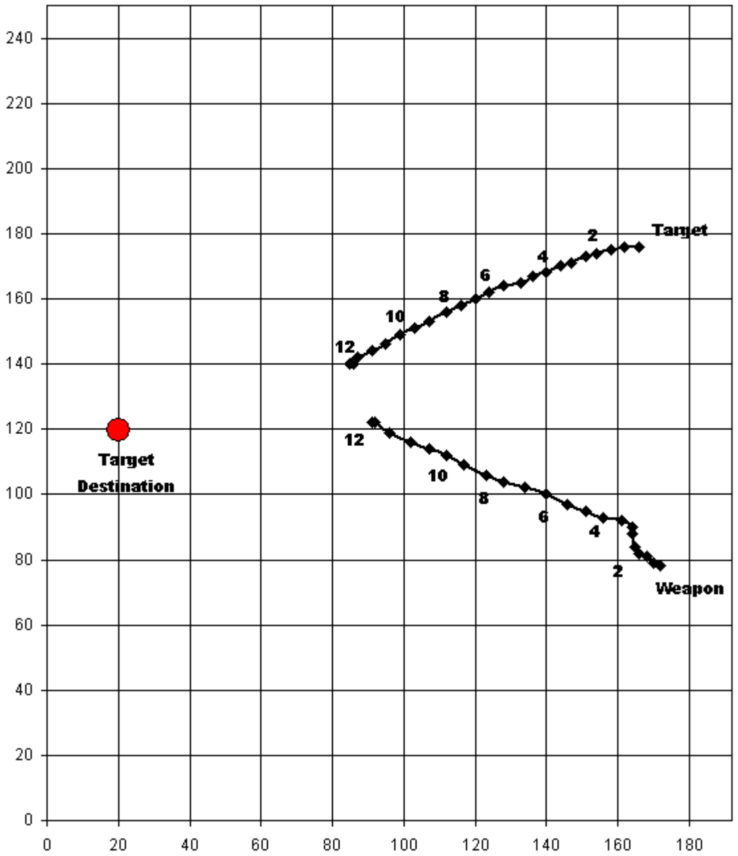


FIGURE 10. One vs. One Intercept Mode.

SUMMARY AND FUTURE WORK

With improved hardware, the capabilities and versatility of our framework could be extended. Servos or motors with accurate calibration methods and integrated encoder hardware would improve the accuracy of robot trajectories. More capable processors would allow for parallel multi-tasking, faster communication, and the potential for more complex calculations. This would allow the use of navigation and avoidance using virtual potential fields (Reference 2). Better communication modules would enable peer-to-peer communication, allowing robots to communicate their current status and location with each other and the control console. A variety of different algorithms could be implemented and tested using the upgraded framework, such as those used by Spears, et al (Reference 6). Additionally, there is an interest to apply these concepts to a three-dimensional environment with airborne test modules, such as UAVs, helicopters, or blimps.

REFERENCES

1. Naval Air Warfare Center Weapons Division. *Swarm Counter-Asymmetric-Threat (CAT) 6-DOF Dynamics Simulation*, by J. Bobinchak and G. Hewer. China Lake, California, NAWCWD, July 2005. (NAWCWD TP 8593, publication UNCLASSIFIED.)

2. B. Q. Nguyen, Y. Chuang, D. Tung, C. H. Hsieh, Z. Jin, L. Shi, D. Marthaler, A. L. Bertozzi, and R. M. Murray. "Virtual Attractive-Repulsive Potentials for Cooperative Control of Second Order Dynamic Vehicles on the Caltech MVWT." American Control Conference, June 2005. Pp. 1084-1089. (Conference and presentation UNCLASSIFIED.)
   http://math.ucla.edu/~bertozzi/papers/potential.pdf

3. C. H. Hsieh, Y. Chuang, Y. Huang, K. K. Leung, A. L. Bertozzi, and E. Frazzoli. "An Economical Micro-Car Testbed for Validation of Cooperative Control Strategies." American Control Conference, June 2006. Pp. 1446-1451. (Conference and presentation UNCLASSIFIED.)
   http://math.ucla.edu/~bertozzi/papers/ACC2006.pdf

4. W. S. Cleveland. The Elements of Graphing Data. Murray Hill, New Jersey, AT&T Bell Laboratories, 1985. (Report UNCLASSIFIED.)

5. D. P. Bertsekas, D. A. Castañon, and H. Tsaknakis. "Reverse Auction and the Solution of Inequality Constrained Assignment Problems," *SIAM J. Optimization*, Vol. 3, No. 2 (May 1993). Pp. 268-297.

6. W. M. Spears, D. F. Spears, R. Heil, W. Kerr, and S Hettiarachchi. "An Overview of Physicomimetics," *Swarm Robotics*. SAB International Workshop, July 2004. Pp. 84-97. (Workshop and presentation UNCLASSIFIED.)