

SANDIA REPORT

SAND2004-4763

Unlimited Release

Printed September 2004

Analysis and Control of Distributed Cooperative Systems

John T. Feddema, David A. Schoenwald, Eric P. Parker, John S. Wagner

Prepared by

Sandia National Laboratories

Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.



Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: <http://www.doe.gov/bridge>

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: <http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online>



Analysis and Control of Distributed Cooperative Systems

John T. Feddema
Intelligent Systems Sensors and Controls Department

David A. Schoenwald
Evolutionary Computing Department

Eric P. Parker, John S. Wagner
Computational Initiatives Department

Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-1003

Abstract

As part of DARPA Information Processing Technology Office (IPTO) Software for Distributed Robotics (SDR) Program, Sandia National Laboratories has developed analysis and control software for coordinating tens to thousands of autonomous cooperative robotic agents (primarily unmanned ground vehicles) performing military operations such as reconnaissance, surveillance and target acquisition; countermine and explosive ordnance disposal; force protection and physical security; and logistics support. Due to the nature of these applications, the control techniques must be distributed, and they must not rely on high bandwidth communication between agents. At the same time, a single soldier must easily direct these large-scale systems. Finally, the control techniques must be provably convergent so as not to cause undo harm to civilians. In this project, provably convergent, moderate communication bandwidth, distributed control algorithms have been developed that can be regulated by a single soldier. We have simulated in great detail the control of low numbers of vehicles (up to 20) navigating throughout a building, and we have simulated in lesser detail the control of larger numbers of vehicles (up to 1000) trying to locate several targets in a large outdoor facility. Finally, we have experimentally validated the resulting control algorithms on smaller numbers of autonomous vehicles.

Acknowledgements

We would like to thank Dr. Mark Swinson and Dr. Doug Gage, DARPA Program Managers for the Software for Distributed Robotics Program, for the opportunity to work on this challenging problem, and for their technical guidance and support throughout the project.

Contents

Abstract	3
Acknowledgments.....	4
Contents	5
Figures	6
I. Introduction	7
II. Stability Analysis.....	9
III. Simulations of Smaller Numbers.....	17
IV. Simulations of Larger Numbers	19
IV.A. Ballistic Simulation Methods.....	19
IV.B. Plasma Simulation Methods	20
V. Hardware Test Platform	22
VI. Conclusions	23

Figures

Figure 1. One-dimensional control problem. The top line is the initial state. The second line is the desired final state. The vehicles can only use their neighbors' position to reach the final goal state.	9
Figure 2. Control block diagram of two vehicle interaction problem.	10
Figure 3. N vehicle interaction problem.	12
Figure 4. Stability region for the N=2 vehicle case.	15
Figure 5. Stability region for the N=1000 vehicle case.	15
Figure 6. Detailed simulation of multiple vehicles navigating a building. The protruding green and blue cones represent the 4 IR proximity sensors.	18
Figure 7. Ballistic simulation of a swarm in a complex urban environment.	20
Figure 8. PIC simulation of a swarm in a complex urban environment.	21
Figure 9. Robotic vehicle platform used in the tests.	22
Figure 10. Surveillance task: starting in a single location, all 20 vehicles spread throughout the building while maintaining ultrasound and communication range to nearest neighbors.	23

I. Introduction

The subject of cooperative multiple autonomous vehicles has generated a great deal of interest in recent years due to the vision of these vehicles being able to perform tasks faster and more efficiently than an individual vehicle. Types of cooperative tasks range from moving large objects [1] to troop hunting behaviors [2]. Conceptually, large groups of mobile vehicles outfitted with sensors should be able to automatically perform military tasks like formation following, localization of chemical sources, de-mining, target assignments, autonomous driving, perimeter control, surveillance, and search and rescue missions [3-6]. Simulation and experiments have shown that by sharing concurrent sensory information, the group can better estimate the shape of a chemical plume and therefore localize its source [7]. Similarly, for a search and rescue operation, a moving target is more easily found using an organized team [8-9].

In the field of distributed mobile robot systems, much research has been performed and summaries are given in [10][11]. The strategies of cooperation encompass theories from such diverse disciplines as artificial intelligence, game theory/economics, theoretical biology, distributed computing/control, animal ethology, and artificial life.

Much of the early research concentrated on animal-like cooperative behavior. Arkin [12] studied an approach to "cooperation without communication" for multiple mobile robots that are to forage and retrieve objects in a hostile environment. This behavior-based approach was extended in [13] to perform formation control of multiple robot teams. Motor schemas such as avoid-static-obstacle, avoid-robot, move-to-goal, and maintain-formation were combined by an arbiter to maintain the formation while driving the vehicles to their destination. Each motor schema contained parameters such as an attractive or repulsive gain value, a sphere of influence, and a minimum range that were selected by the designer. "When inter-robot communication is required, the robots transmit their current position in world coordinates with updates as rapidly as required for the given formation speed and environmental conditions." [13]

Another behavior-based approach includes Kube and Zhang [14]. Much of their study examined comparisons of behaviors of different social insects such as ants and bees. They considered a box-pushing task and utilized a Subsumption approach [15-16] as well as ALN (Adaptive Logic Networks). Similar studies using analogs to animal behavior can be found in Fukuda et al. [17]. Noreils [18] dealt with robots that were not necessarily homogeneous. His architecture consisted of three levels: functional level, control level, and planner level. The planner level was the high-level decision maker. Most of behavior-based approaches do not include a formal development of the system controls from a stability point of view. Many of the schemes such as the Subsumption approach rely on stable controls at a lower level while providing coordination at a higher level.

More recently, researchers have begun to take a system controls perspective and analyze the stability of multiple vehicles when driving in formations. Chen and Luh [19] examined decentralized control laws that drove a set of holonomic mobile robots into a circular formation. A conservative stability requirement for the sample period is given in terms of the damping ratio and the undamped natural frequency of the system. Similarly, Yamaguchi studied line-formations [20] and general formations [21] of nonholonomic vehicles, as did Yoshida *et al.* [22]. Decentralized control laws using a potential field approach to guide vehicles away from obstacles can be found in [23-24]. In these studies, only continuous time analyses have been performed, assuming that the relative position between vehicles and obstacles can be measured at all time.

Another way of analyzing stability is to investigate the convergence of a distributed algorithm.

Beni and Liang [25] proved the convergence of a linear swarm of asynchronous distributed autonomous agents into a synchronously achievable configuration. The linear swarm is modeled as a set of linear equations that are solved iteratively. Their formulation is best applied to resource allocation problems that can be described by linear equations. Liu *et al.* [26] provide conditions for convergence of an asynchronous swarm in which swarm "cohesiveness" is the stability property under study. Their paper assumes position information is passed between nearest neighbors only and proximity sensors prevent collisions.

Also of importance is the recent research combining graph theory with decentralized controls. Most cooperative mobile robot vehicles have wireless communication, and simulations have shown that a wireless network of mobile robots can be modeled as an undirected graph [27]. These same graphs can be used to control a formation. Desai *et al.* [28-29] used directed graph theory to control a team of robots navigating terrain with obstacles while maintaining a desired formation and changing formations when needed. When changing formations, the transition matrix between the current adjacency matrix and all possible control graphs are evaluated. In the next section, the reader will notice that graph theory is also used in this paper to evaluate the controllability and observability of the system.

Other methods for controlling a group of vehicles range from distributed autonomy [30] to intelligent squad control and general purpose cooperative mission planning [31]. In addition, satisfaction propagation is proposed in [32] to contribute to adaptive cooperation of mobile distributed vehicles. The decentralized localization problem is examined by Roumeliotis and Bekey [33] and Bozorg *et al.* [34] via the use of distributed Kalman filters. Uchibe *et al.* [35] use Canonical Variate Analysis (CVA) for this same problem.

In this project, we addressed the stable control of multiple vehicles using large-scale decentralized control techniques [36]. The approach taken differs from previous efforts in that the analysis techniques are scalable to very large dimensions and they ensure stability even under structural perturbations such as communication failures and parameter variations. While this depth of analysis may not be necessary when controlling smaller numbers of vehicles, the formalism introduced here is necessary when tens to hundreds, possibly thousands of vehicles, are involved. With hundreds of vehicles, it is not feasible to experimentally determine the interaction gains and the communication rates between vehicles necessary to stabilize the system.

In this project, we have also focused our effort on a surveillance task where large numbers of vehicles from 20 to 1000 are dispersed around a facility. The goal is for these vehicles to autonomously create a distributed communication/navigation network that links a remote base station to multiple surveillance points. We have simulated in great detail the control of low numbers of vehicles (up to 20) navigating throughout a building. These simulations include detailed models of the radio frequency communication, infrared and ultrasound ranging with the environment and amongst vehicles, and the vehicles' kinematics.

We have also simulated in lesser detail the control of larger numbers of vehicles (up to 1000) trying to locate a number of targets in a large outdoor facility. These simulations use software initially developed to simulate molecular interactions in plasma physics and celestial body interactions in galactic physics. Grid-based techniques limit the range of interaction thus reducing the computational load of computing the interactive forces.

In both the small-scale and large-scale simulations, the guidance of the vehicles is based on attractive and repulsive gradient forces. These gradient forces are derived from optimal performance indices that trade off minimizing the distance to specified goals, and optimizing the distance between vehicles to maintain a required communication distance.

To verify and validate the control algorithms, we have implemented and tested the algorithms in hardware. We have developed 20 test vehicles with the ability to navigate an indoor environment. Each vehicle contains a 4MHz 8-bit microcontroller, a 900 MHz radio, 4 infrared proximity sensors, an electromagnetic compass, and a wheel encoder. For more computationally expensive algorithms, an optional 25MHz 386EX embedded processor interfaces to the microcontroller through dual port SRAM. Each vehicle is approximately 250 mm long, 225 mm wide, and 200 mm tall. Powered by rechargeable nickel-metal hydride batteries, the vehicles are typically able to operate 30-45 minutes before needing to be recharged.

The following sections briefly describe the stability analysis, small-scale and large-scale simulations, and the hardware test platform. More details on the stability analysis and the experimental results can be found in [39-44].

II. Stability Analysis

In complex large-scale systems, it is often desirable to break up a system into smaller strongly coupled systems that are controllable. If we can prove that the smaller systems are input/output reachable and controllable, then we can prove that the large-scale system is connectively controllable [36]. Even the smaller scale systems may contain thousands of states, in which case, there exist techniques based on graph theory that can quickly tell a person whether the system is input/output reachable and structurally controllable. The example below shows some of the progress made in understanding how these techniques can be used in the design of large-scale distributed cooperative robotic vehicular systems.

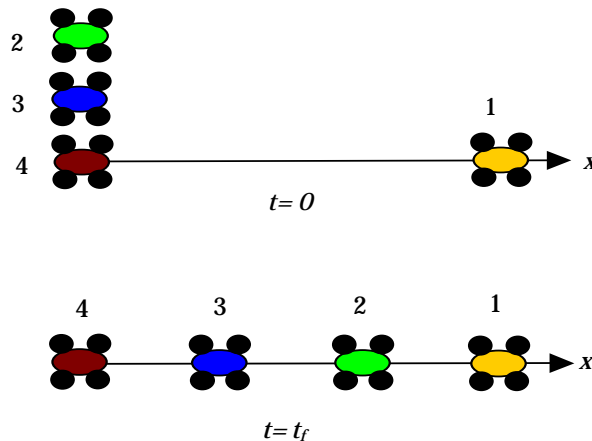


Figure 1. One-dimensional control problem. The top line is the initial state. The second line is the desired final state. The vehicles can only use their neighbors' position to reach the final goal state.

Let us start by analyzing a simple one-dimensional problem where a linear chain of interdependent vehicles is to spread out along a line as shown in Figure 1. The objective is to spread out evenly along the line using only information from the nearest neighbor. In [37], the Sandia National Laboratories had previously developed a robotic perimeter detection system that spread the vehicles uniformly around a perimeter. The vehicles shared their current positions with neighboring vehicles via radio messages. The goal position, used by the embedded control on each vehicle, was one-half the distance between the vehicles to their right and left. This goal

point moved as the neighboring vehicles moved and updated their position with their neighbors. In the experiments, the system worked, but we wondered if one-half was a magic number and if we could prove that it provides a stable solution regardless of how fast the vehicles move and how often they share their current position. The following analysis resulted.

Assume the vehicle's plant model is a simple integrator, and the commanded input is the desired velocity of the vehicle along the line. A feedback loop and a proportional gain K_p are used to control the vehicle's position. The desired position of each vehicle is one-half the sum of the position of the neighbors on each side. Figure 2 shows a block diagram of the control system. The formulation is in the discrete-time frequency domain, i.e. the z domain. Since we are interested in steady state analysis, we will make heavy use of the final value theorem, which states

$$\lim_{k \rightarrow \infty} f(kT) = \lim_{z \rightarrow 1} (1 - z^{-1}) F(z) \quad (1)$$

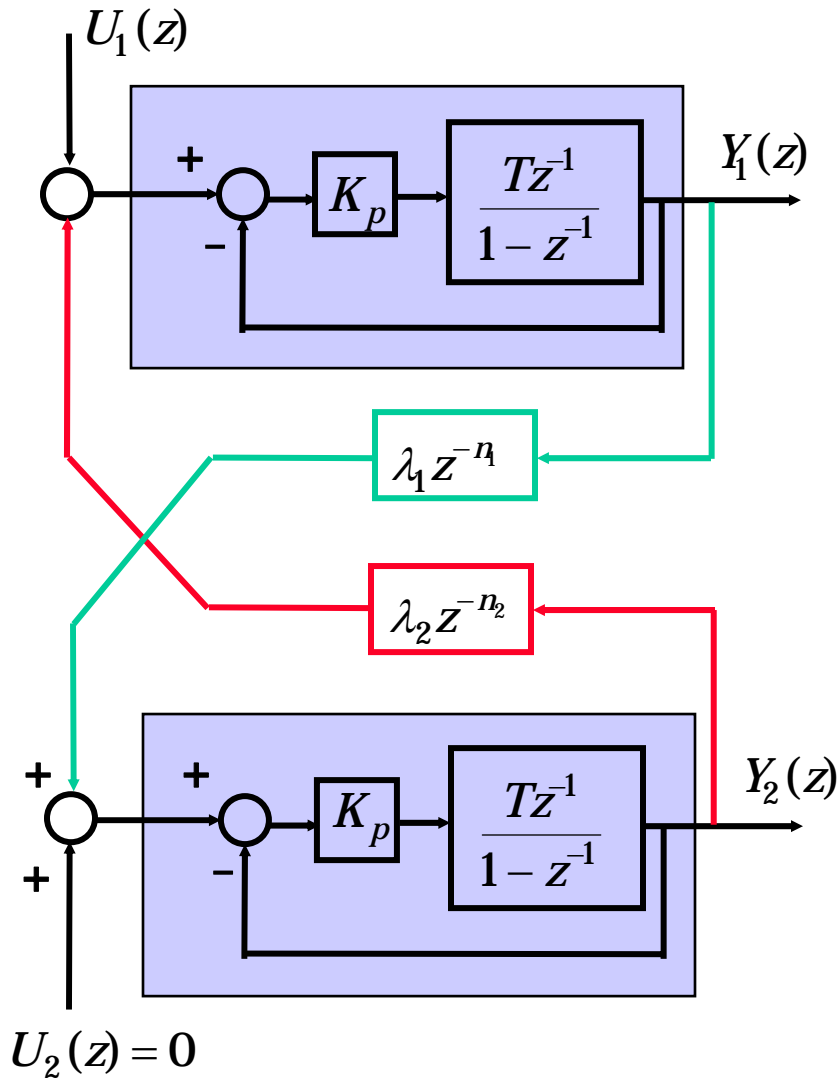


Figure 2. Control block diagram of two vehicle interaction problem.

If we let $H_1(z)$ be the transfer function $Y_1(z)/U_1(z)$ and $H_2(z)$ be the transfer function $Y_2(z)/U_1(z)$ then we have

$$\begin{aligned} y_1^{ss} &= \lim_{k \rightarrow \infty} y_1(kT) = \lim_{z \rightarrow 1} H_1(z) \\ y_2^{ss} &= \lim_{k \rightarrow \infty} y_2(kT) = \lim_{z \rightarrow 1} H_2(z) \end{aligned} \quad (2)$$

where we have used the fact that $U_1(z) = 1/(1-z^{-1})$, i.e., $u_1(kT)=1$ for all k . That is, the desired linear positioning behavior has been normalized to be between 0 and 1. The superscript "ss" refers to the steady state value. Carrying out the block diagram manipulation and algebra, one arrives at the formulas for the steady state position values of the two vehicles in terms of the given parameters

$$\begin{aligned} y_1^{ss} &= \frac{1}{1 - \lambda_1 \lambda_2} \\ y_2^{ss} &= \frac{\lambda_1}{1 - \lambda_1 \lambda_2} \end{aligned} \quad (3)$$

where λ_1 and λ_2 are the interaction gains. Given the steady state positions, the formulas for the interaction gains are

$$\begin{aligned} \lambda_1 &= \frac{y_2^{ss}}{y_1^{ss}} \\ \lambda_2 &= \frac{y_1^{ss} - 1}{y_2^{ss}} \end{aligned} \quad (4)$$

These formulas assume a stable configuration. We will explore stability analysis shortly. It should be noted that both steady state positions are independent of the delays n_1 and n_2 , the proportional gain K_p , and the sampling time delay T . Now consider the three-vehicle case as depicted in Figure 3. Using the same analysis as above, we can arrive at the steady state positions (assuming stability) for the three vehicles as

$$\begin{aligned} y_1^{ss} &= \frac{1 - \lambda_{32} \lambda_{23}}{1 - \lambda_{21} \lambda_{12} - \lambda_{32} \lambda_{23}} \\ y_2^{ss} &= \frac{\lambda_{12}}{1 - \lambda_{21} \lambda_{12} - \lambda_{32} \lambda_{23}} \\ y_3^{ss} &= \frac{\lambda_{12} \lambda_{23}}{1 - \lambda_{21} \lambda_{12} - \lambda_{32} \lambda_{23}} \end{aligned} \quad (5)$$

where again only the vehicle interaction gains affect the steady state position values. To solve the inverse problem, i.e., the vehicle interaction gains given the steady state positions, we must solve an undetermined system of nonlinear equations with 3 equations and 4 unknowns. This can be done using a nonlinear least squares root finding algorithm such as employed in the MATLAB routine, `fsolve`.

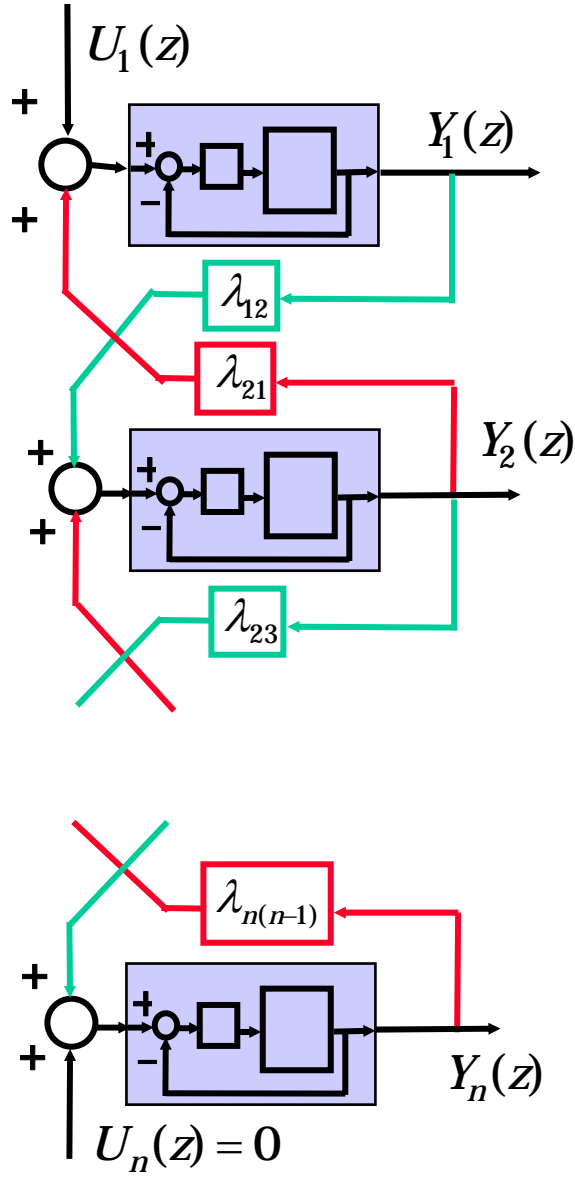


Figure 3. N vehicle interaction problem.

To generalize for the N vehicle interaction problem, we formulate a set of linear equations based on the algebra of the transfer function manipulation. Note that

$$\begin{aligned}
 Y_1^{ss} &= 1 + \lambda_{21} Y_2^{ss} \\
 Y_i^{ss} &= \lambda_{i-1,i} Y_{i-1}^{ss} + \lambda_{i+1,i} Y_{i+1}^{ss} \\
 Y_N^{ss} &= \lambda_{N-1,N} Y_{N-1}^{ss}
 \end{aligned} \tag{6}$$

where i is an integer such that $i \in [2, N-1]$.

This results in the tridiagonal [38] system of linear equations

$$\begin{bmatrix} Y_1^{ss} \\ \vdots \\ Y_i^{ss} \\ \vdots \\ Y_N^{ss} \end{bmatrix} = \begin{bmatrix} 0 & \lambda_{21} & 0 & \dots & 0 \\ \lambda_{12} & 0 & \lambda_{32} & \dots & 0 \\ 0 & \lambda_{i-1,i} & 0 & \lambda_{i+1,i} & 0 \\ \vdots & \ddots & \ddots & \ddots & \lambda_{N,N-1} \\ 0 & \dots & 0 & \lambda_{N-1,N} & 0 \end{bmatrix} \begin{bmatrix} Y_1^{ss} \\ \vdots \\ Y_i^{ss} \\ \vdots \\ Y_N^{ss} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \quad (7)$$

which can be reformulated in the familiar Ax=b form

$$\begin{bmatrix} 1 & -\lambda_{21} & 0 & \dots & 0 \\ -\lambda_{12} & 1 & -\lambda_{32} & \dots & 0 \\ 0 & -\lambda_{i-1,i} & 1 & -\lambda_{i+1,i} & 0 \\ \vdots & \ddots & \ddots & \ddots & -\lambda_{N,N-1} \\ 0 & \dots & 0 & -\lambda_{N-1,N} & 1 \end{bmatrix} \begin{bmatrix} Y_1^{ss} \\ \vdots \\ Y_i^{ss} \\ \vdots \\ Y_N^{ss} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \quad (8)$$

Thus, given the steady state position values, the required interaction gains can be solved for as a system of linear equations using least squares. The solution will not be unique; hence many different sets of interaction gains can result in the same steady state position values of the vehicles. Likewise, the inverse problem can be solved to determine the interaction gains given a set of desired steady state vehicle position values. We obtain

$$\begin{bmatrix} 0 & Y_2^{ss} & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ Y_1^{ss} & 0 & 0 & Y_3^{ss} & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & Y_2^{ss} & 0 & 0 & Y_4^{ss} & 0 & \dots & 0 \\ 0 & \dots & \dots & \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & \dots & Y_{i-1}^{ss} & 0 & 0 & Y_{i+1}^{ss} & 0 & \dots & 0 \\ 0 & \dots & \dots & \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & Y_{N-1}^{ss} & 0 \end{bmatrix} \begin{bmatrix} \lambda_{12} \\ \lambda_{21} \\ \lambda_{23} \\ \vdots \\ \lambda_{i-1,i} \\ \lambda_{i,i-1} \\ \vdots \\ \lambda_{N-1,N} \\ \lambda_{N,N-1} \end{bmatrix} = \begin{bmatrix} Y_1^{ss} - 1 \\ Y_2^{ss} \\ Y_3^{ss} \\ \vdots \\ Y_i^{ss} \\ \vdots \\ Y_N^{ss} \end{bmatrix} \quad (9)$$

In this case, Ax=b is an underdetermined system (more unknowns than equations, $2(N-1) > N$ for $N > 2$) which can be solved using QR factorization such as with the MATLAB backslash (\) operator. Alternatively, this can be solved as a constrained linear least squares problem:

$$\min_x \frac{1}{2} \|Ax - b\| \quad s.t. \quad x \geq 0, \quad Ax = b, \quad \varepsilon \leq x \leq 1 - \varepsilon \quad (10)$$

where the first constraint rejects negative interaction gains, the second constraint forces Equation (9) to be solved exactly, and the third constraint rejects zero and unity interaction gains, that is, ε is a small parameter greater than zero. The advantage of formulating Equation (9) as a constrained least squares problem is that we can eliminate nonzero interaction gains from the set of possible solutions. Since this corresponds to a vehicle not utilizing information of its nearest neighbors, it is best to look at nonzero interaction gains.

In order to analyze stability of the N vehicle interaction problem, a reformulation of the vehicle dynamics into discrete-time state space is helpful. The purpose of this analysis is to determine conditions for asymptotic stability of vehicle positions with respect to the interaction gains λ and vehicle speed time constant $K_p T$. The following time-domain equations are derived from Fig. 3:

$$\begin{aligned}
 y_1(k+1) &= (1 - K_p T)y_1(k) + K_p T\lambda_{21}y_2(k-1) + K_p T u_1(k) \\
 y_i(k+1) &= (1 - K_p T)y_i(k) + K_p T\lambda_{i-1,i}y_{i-1}(k-1) + K_p T\lambda_{i+1,i}y_{i+1}(k-1), \quad 2 \leq i \leq N-1 \\
 y_N(k+1) &= (1 - K_p T)y_N(k) + K_p T\lambda_{N-1,N}y_{N-1}(k-1)
 \end{aligned} \tag{11}$$

where it is assumed that $u_N(k)=0$ and the delay between position interaction information is one sampling delay. To solve these equations, initialize by setting $y_i(0)=y_i(1)=0$ for i between 1 and N (note that initial vehicle positions do not have to start at 0, but this is the normal case). Then start the difference equation solver at $k=1$ (i.e. compute $y_i(2)$). For the stability analysis, we note that we can put (11) into a state space description. We break the analysis into two cases.

Case I: All interaction delays =0, i.e. $n_{ij}=0$. This results in the following state space description.

$$\begin{aligned}
 & \begin{bmatrix} y_1(k+1) \\ y_2(k+1) \\ \vdots \\ y_i(k+1) \\ \vdots \\ y_{N-1}(k+1) \\ y_N(k+1) \end{bmatrix} = \begin{bmatrix} 1 - K_p T & K_p T\lambda_{21} & & & & & & 0 \\ K_p T\lambda_{12} & 1 - K_p T & K_p T\lambda_{32} & & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & K_p T\lambda_{i-1,i} & 1 - K_p T & K_p T\lambda_{i+1,i} & & \\ & & & & & \ddots & \ddots & \\ & & & & & & K_p T\lambda_{N-2,N-1} & 1 - K_p T & K_p T\lambda_{N,N-1} \\ 0 & & & & & & & K_p T\lambda_{N-1,N} & 1 - K_p T \end{bmatrix} \cdot \\
 & \begin{bmatrix} y_1(k) \\ y_2(k) \\ \vdots \\ y_i(k) \\ \vdots \\ y_{N-1}(k) \\ y_N(k) \end{bmatrix} + \begin{bmatrix} K_p T \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} u_1(k)
 \end{aligned} \tag{12}$$

which is in the form: $y(k+1) = Ay(k) + Bu_1(k)$. The eigenvalues can easily be solved for in any of a number of software packages including MATLAB. For stability, we look at the maximum absolute value of all the eigenvalues of A which is a real NxN matrix. If this is inside the unit circle (less than unity magnitude) then we have asymptotic stability of the vehicle positions. Otherwise, we do not have a stable vehicle configuration. Note that the A matrix above is in tridiagonal form.

For the special case of all the interaction gains $\lambda_{ij} = \lambda$, the elements of each diagonal are equal. There is a special formula (p. 59 of [38]) for the eigenvalues of A in this case, which is

$$\text{eig}(A) = 1 - K_p T + 2K_p T \lambda \cos\left(\frac{i\pi}{N+1}\right), \quad i = 1, \dots, N \quad (13)$$

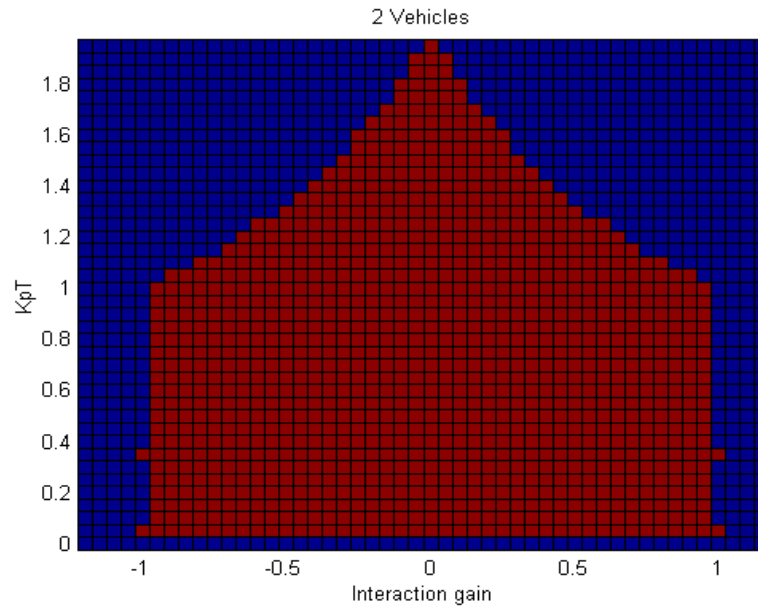


Figure 4. Stability region for the $N=2$ vehicle case.

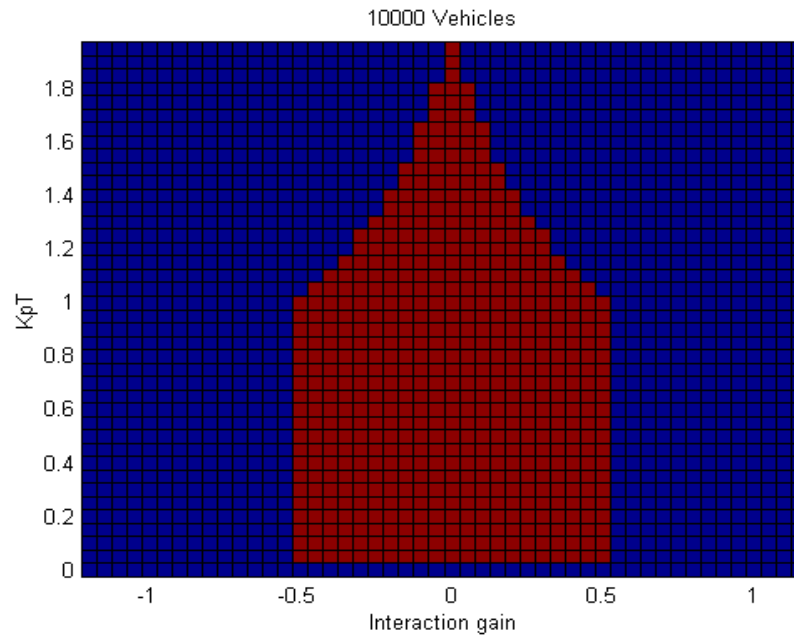


Figure 5. Stability region for the $N=1000$ vehicle case.

All the eigenvalues in (13) will be real. Figures 4 and 5 illustrate the stability region for this case. The red zone represents stable combinations of λ and $K_p T$. The blue zone represents unstable

combinations of λ and $K_p T$. We refer to this as a stability ‘‘house’’ due to the shape of the stable zone. The size of this house varies only with N . The plot shown is for $N=2$. As N is increased, the house gets smaller in width but maintains the same height and shape. Figure 5 shows the stability region for $N=1000$. From the formula in (13), we can see that as $N \rightarrow \infty$ the cosine term becomes unity. This implies that λ must stay between -0.5 and 0.5 for $K_p T$ less than one in order to maintain stability. For $K_p T$ greater than one, the admissible λ values taper off parabolically (the sloped ‘‘roof’’) until $K_p T = 2$. Computer simulations of (11) agreed with these stability results.

Case II: all interaction delays = 1, i.e. $n_{ij}=1$

For this case, we get a more complex state space description

$$\begin{bmatrix} y_1(k+1) \\ \vdots \\ y_i(k+1) \\ \vdots \\ y_N(k+1) \\ y_1(k) \\ \vdots \\ y_N(k) \end{bmatrix} = \begin{bmatrix} 1-K_p T & & & 0 & 0 & K_p T \lambda_{21} & & & & & \\ & \ddots & & & & \ddots & & & & & \\ & & 1-K_p T & & & K_p T \lambda_{i-1,i} & 0 & & K_p T \lambda_{i+1,i} & & \\ & & & \ddots & & \ddots & \ddots & & \ddots & & \\ 0 & & & & 1-K_p T & & & & K_p T \lambda_{N-1,N} & & 0 \\ & & & & & & & & & & \\ & & & & & & & & & & \\ & & I_{N \times N} & & & & & & & & \\ & & & & & & & & & & \\ & & & & & & & & O_{N \times N} & & \\ & & & & & & & & & & \end{bmatrix} \cdot \begin{bmatrix} y_1(k) \\ \vdots \\ y_i(k) \\ \vdots \\ y_N(k) \\ y_1(k-1) \\ \vdots \\ y_N(k-1) \end{bmatrix} + \begin{bmatrix} K_p T \\ 0 \\ \vdots \\ 0 \end{bmatrix} u_1(k) \quad (14)$$

The above still fits the $y(k+1)=Ay(k)+bu_1(k)$ formulation. Note now that A is a $2N \times 2N$ matrix. It is also no longer a tridiagonal matrix. There is no simple formula for the eigenvalues of A in this case even if all $\lambda_{ij} = \lambda$. The eigenvalues can still be solved for using standard linear algebra software, but this becomes numerically unreliable for large N . However, if a software package has techniques for handling large sparse matrices (as does MATLAB) then it becomes more tractable. In the above description, only $4N-2$ elements of A are nonzero in general out of $4N^2$ total elements. Thus for large N , the A matrix is sparse. Though a formula is lacking, computer simulation of (4) and solving for the maximum absolute eigenvalue of A above resulted in very nearly the same stability regions with respect to λ and $K_p T$. In other words, the delay in interaction gains between vehicles did not affect the stability of the vehicle positions to any appreciable degree.

A more specific case can be studied in which all forward λ_{ij} 's are equal (i.e. $\lambda_{12} = \lambda_{23} = \dots = \lambda_F$) and all backward λ_{ij} 's are equal (i.e. $\lambda_{21} = \lambda_{32} = \dots = \lambda_B$). In this case the stability region has a three-dimensional structure, $K_p T$ vs. λ_F vs. λ_B . Numerical simulation of this case revealed that for various fixed $K_p T$ contours from 1 to 2, the stability region for λ_F and λ_B looked like a $1/\lambda$ surface that increased in size as $K_p T$ decreased from 2 down to 1. This is intuitive because we expect the range of λ 's for stability to shrink as the speed gain is increased. This is essentially a three dimensional version of the ‘‘roof’’ of the stability house in Figs. 4 and 5.

Several conclusions can be drawn from the stability analysis. First, asymptotic stability of vehicle positions depends on vehicle responsiveness K_p , communication sampling period T , and vehicle interaction gain λ . Second, if the vehicle is too fast (large K_p) or the sample period is too long (large T), then the vehicles will go unstable. There is a dependence on interaction gain for stability as well. Third, the interaction gains can be used to bunch the vehicles closer together or spread them out. Fourth, the stability region shrinks as the number of vehicles, N , increases but only to a defined limit. Finally, we can give a two-step process for placing the vehicles into an arbitrary position. First, solve Eq. (3) for the λ_{ij} 's necessary to achieve these vehicle positions. Then, use the above stability analysis (the stability "house") to determine the upper limits for $K_p T$ to maintain stability.

III. Simulations of Smaller Numbers

The multiple vehicle problem in planar space is essentially a generalization of the above analysis. But when obstacles such as walls and other vehicles as well as the need for communication between vehicles are taken into account, the ability to analytically solve the problem becomes very difficult. Thus, we implemented a study of multiple vehicles under such constraints in a simulation environment developed at Sandia National Laboratories called Umbra. Umbra allows the simulation of multiple autonomous agents with a variety of physical phenomena such as RF communications, interactions with solid objects (i.e. collisions), ultrasound communication, IR detection of objects, vehicle physics, terrain descriptions, and other phenomena the user wishes to study. All of these physical attributes can be simulated simultaneously with a graphical visualization that allows the monitoring of the vehicles' performance over the terrain.

Such a simulation was implemented for the case of multiple, small, wheeled vehicles traversing a single floor in a building with multiple corridors, rooms, and entrances. The vehicles are modeled after the vehicles that will be used in the hardware tests. Each vehicle contains 4 IR sensors for detecting objects between 6" and 2.5' on all 4 sides of itself (see Figure 6). The vehicles also contain RF communication devices to be able to converse with other vehicles within a 100' line of sight range or roughly 30' through walls. They also have ultrasound capability to measure the distance between them provided they are within 30' of each other and in line of sight range. The vehicle physics are quite simple and proved adequate on a smooth surface. The building model was generated as a CAD model and contains several connected hallways as well as a multitude of variable size rooms. The control algorithms for the vehicles must avoid contact with walls and other vehicles. Beyond that, the control goals can vary depending on the motives of the operator. For instance, the vehicles can spread out to provide maximum coverage of the building or they can stay within a prescribed area, or they can maintain a particular formation. Note that a strict mathematical model of this situation is intractable. This is due to both discrete event-based as well as dynamic physics with very complicated interactions. Thus, a stability analysis is more qualitative in nature rather than strictly mathematical.

The restriction that vehicles can't run into walls, doors, or each other essential ensures they remain inside the building. This is accomplished via rules that use the IR sensors to follow walls down a hallway. This will enable the vehicles to move throughout the building, though not necessarily in any prescribed fashion. Further restrictions on the vehicles involve the maintenance of a continuous RF communication network. This requires that vehicles stay within 100' of each other or less if line of sight (LOS) is lost (i.e. they may have to stay at a wall junction to maintain LOS). A more stringent condition is the ability for each vehicle to know its absolute (x,y) position with respect to some global coordinate system. This requires triangulation off of two or more known vehicles using ultrasound as a distance measurement. This implies that at least two vehicles must remain in fixed known locations until the other vehicles can triangulate off of them. There are a number of techniques to accomplish this that were investigated in Umbra. These include the law of cosines triangulation, steepest descent triangulation, and conjugate

gradient triangulation [42]. All had advantages and disadvantages depending on the number of vehicles and the on-board processing power and memory. Finally, there is the constraint that the vehicles spread out and “cover” the building uniformly. A gradient-based scheme was used to repel the vehicles from each other to diffuse through the building while the aforementioned constraints keep them close enough to communication and compute absolute position [41].

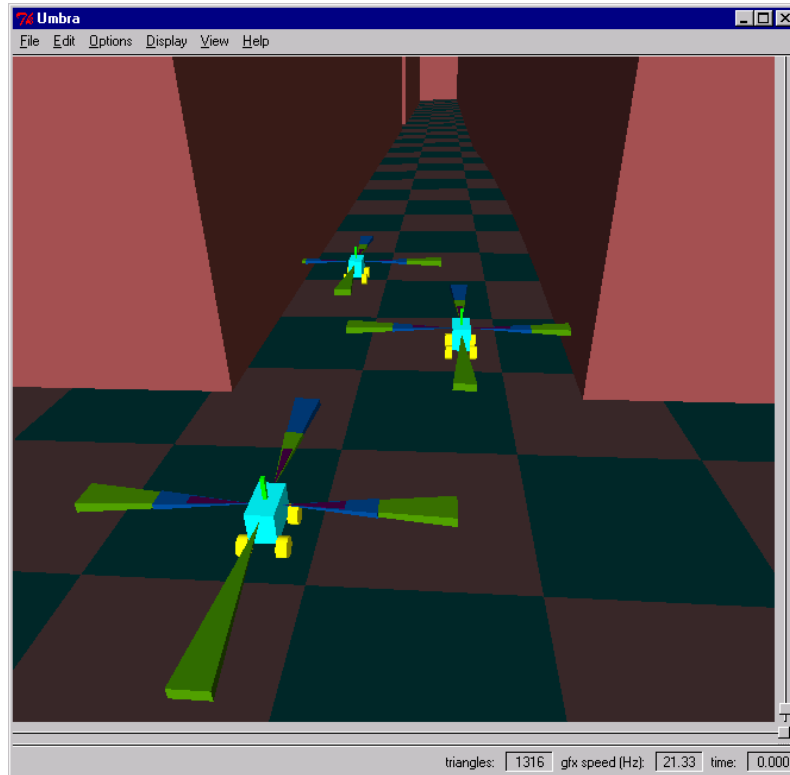


Figure 6. Detailed simulation of multiple vehicles navigating a building. The protruding green and blue cones represent the 4 IR proximity sensors.

IV. Simulations of Larger Numbers

In this project, models and simulations were developed that lead directly to the capability to simulate and understand the introduction and behavior of swarms of semi-autonomous robotic agents in urban and military environments. The physics-based modeling of swarms of autonomous robots utilizes the approaches of statistical mechanics, molecular dynamics, and plasma physics. The advantages of this approach include leveraging a large body of work on stability, fluctuation spectra, equilibrium, and efficient computation of the dynamics of potentially large ensembles of interacting objects.

Plasma simulation methods make possible the comparative theoretical study of cooperative behavior in certain limits. We have been investigating swarming and collaborative behavior from a statistical mechanics point of view where the motion or flight of vehicles is dictated by the physics of particles in potential fields. The motion is a combination of real potentials (gravity, drag, propulsion) and fictitious potentials (anti-collision and target seeking) generated by models and sensors. In the limit where the real and fictitious potentials are similar to physical, Coulomb,

or EM electrodynamics, the mature sciences of plasma and statistical mechanics theory can be applied to the system to study stability, fluctuations, dissipation, and entropy in provable limits.

IV.A. Ballistic simulation methods

In plasma physics particles are considered to have mass and charge and they move self-consistently in both self-generated and externally applied electric and magnetic potential fields. The interaction forces can be long range (electromagnetic) or short range (coulomb collisions). In molecular dynamics (and lattice gas dynamics) the particles are considered uncharged and generally move according to more simple short-range collisional forces and external boundary conditions. This model may be simplified further by considering particles moving in the ballistic limit, in which case only instantaneous impulsive forces act at the beginning of a simulation and then cease to exist.

As an example consider Figure 7, which is from a Particle-In-Cell (PIC) code simulation taken to the ballistic limit. This simulation models the injection and swarming behavior of 10^3 autonomous agents deployed at two locations within Area 1 of the main campus of Sandia National Laboratories. In this simulation the autonomous vehicles are assumed to be man made, with the ability to communicate over a specified distance. After injection into Area 1, all members of the swarm travel in a straight line at a constant velocity until they collide with an obstacle. Momentum is conserved during all collisions, whether with other members of the swarm or with walls within the simulated urban environment. As the simulation progresses, members of the swarm fill the area, searching for two targets, shown in purple. Each member of the swarm acts with complete autonomy until it either finds one of the targets or comes into communications range of another member of the swarm that has. When a target is located, the robot in question stops and broadcasts a signal stating that a target has been found. Upon entering communications range other members of the swarm also stop moving and broadcast the same signal. Over time the swarm gradually ceases to move as more and more robots relay the message that a target has been located.

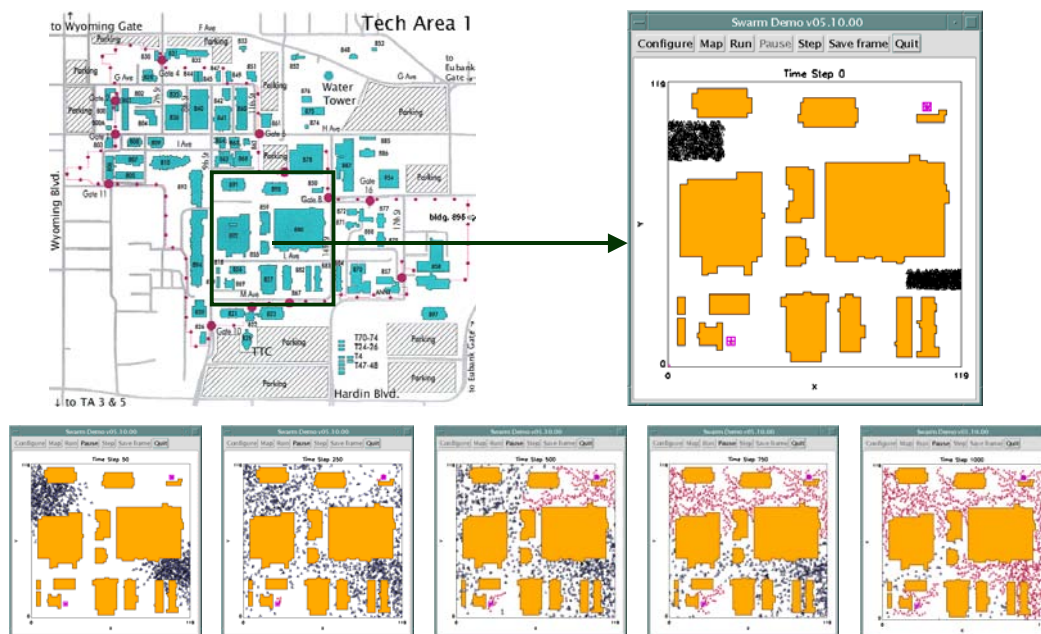


Figure 7. Ballistic simulation of a swarm in a complex urban environment.

VI.B. Plasma Simulation Methods

Modern plasma simulations accurately follow large numbers ($\sim 10^6$) of charged particles interacting with each other self-consistently. The particles move according to the forces from both applied (external) fields as well as fields the particles generate themselves. Abstractly, this is mathematically equivalent to a swarm of communicating vehicles moving according to the forces applied to them, as shown below. In this example, as with the first, a PIC code was modified to simulate the collective behavior of a swarm of 10^3 robots injected into a portion of Sandia's Area 1. In this simulation friction, drag, inertia, and a pursuer's swarming and target seeking forces were added to the model. The effect of adding these forces is striking. When the simulation begins, the swarm is divided into two tightly packed groups. Intermediate range swarming forces and target seeking forces pull the separate halves of the swarm towards each other and towards the two targets very quickly. As the simulation progresses, nearest-neighbor repulsive forces prevent the separate groups from re-forming a single, compact group. Eventually the forces begin to reach equilibrium, with groups of robots near each target, and a larger number distributed throughout the search area.

As the example below shows, plasma simulation codes can accurately compute complex trajectories and efficiently handle large numbers of particles and vehicles. Large populations are an issue because N interacting or communicating objects generally require N^2 communication events, which is an unfavorable numerical scaling for a simulation that must be time stepped to resolve group dynamics. Plasma simulation codes efficiently handle this communication

bottleneck by grouping near (strongly communicating) and far (weakly communicating) neighbors onto virtual meshes or into linked lists. The computational requirements of PIC codes scale as $O(N)$, where N is the number of particles. As stated earlier, gridless particle simulation codes scale as $O(N^2)$, and grid-optional methods scale as $O(N)$, $O(N \log(N))$, or $O(N^2)$ depending on parameters set at run-time.

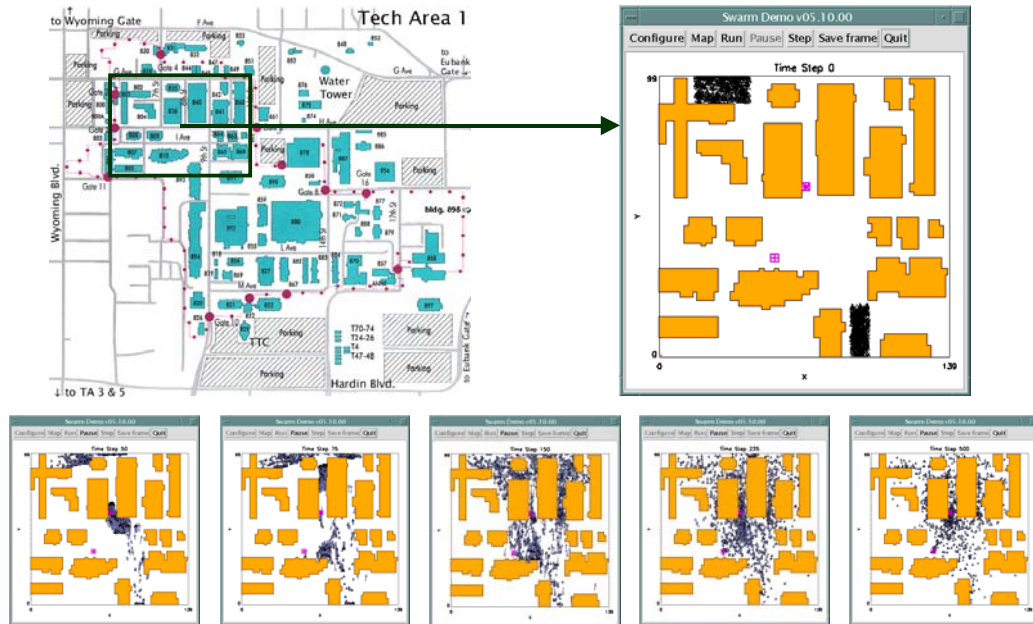


Figure 8. PIC simulation of a swarm in a complex urban environment.

The value of plasma simulation methods extends beyond just the simulation utility. Simple biomimetic behavior such as flocking can be modeled effectively using coulomb-like potentials. More complex biomimetic behavior arises when the vehicles are given more complicated controllers. For example, a program or neural net can make decisions and adapt to changing conditions in an attempt to achieve a goal. For further ground based robotic studies we can use an underlying lattice gas model instead of a plasma model. A more complete model using genetic algorithms to globally optimize terrestrial robotic behavior is currently under development.

In summary, there are several reasons to use a particle simulation code to model swarm dynamics. First, particle simulation codes are benchmarked. They have been shown to be stable and accurate models of particle dynamics with both applied and self-consistent forces. Second, particle simulation codes have well-researched theoretical underpinnings. The statistical mechanics of finite-sized particles moving under common force laws are well documented and understood. Third, particle simulation codes are efficient. Grid-optional codes are highly efficient for large numbers of particles. The N^2 -algorithm is only used for near-neighbors; multiple expansions are used for distant neighbors. Particle simulation codes have been implemented for both serial and parallel processors. Fourth, particle simulation codes are feature rich. Inertial forces, platform constraints, friction and drag effects, gravitational forces, and boundary conditions are easily modeled. A complete set of diagnostics is also available. Finally, particle

simulation codes support heterogeneous particles. Friend and foe behavior are analogous to the physical properties of different interacting molecules.

V. Hardware Test Platform

To test the algorithms and software developed in this project, we have built 20 low-cost robotic vehicle platforms that contain the necessary processing and sensing to navigate and traverse a building. The platforms are built on top of a \$60 commercial radio control car called Super Rebound from Tyco. We have removed the external housing and replaced the radio control and motor amplifier electronics with our own custom circuit boards (see Figure 10). A smaller circuit board in the back of the vehicle contains the power conditioning and two motor amplifier H-bridges. The larger circuit board on top of the vehicle contains a 4MHz 8-bit microcontroller, 900MHz radio, 4 infrared sensors, wheel encoder interface electronics, ultrasound interface electronics, and an electromagnetic compass. This board also has 16 kilobytes of dual port SRAM which allows it to interface to a commercially available embedded processing board with a 25MHz 386EX processor. This processor will be used for computationally intensive tasks such as computing the location of the vehicles from the ultrasound ranging information. On top of the vehicle is an omni-direction ultrasound transceiver. Using the RF radio to indicate the start of a ultrasound chirp, the ultrasound transceivers can be used to measure the distance (via time of flight) between vehicles. With three or more vehicles, we can triangulate to determine their x,y position in a plane. Using this technique, the system has a positioning accuracy of 25-50 mm.



Figure 9. Robotic vehicle platform used in the tests.

Many of the algorithms described above were implemented on these 20 vehicles (see [39-44] for more details). Our goal was to demonstrate that a cooperative group of robotic vehicles can form a communication/navigation network, and that this network could be applied to a military surveillance task.



Figure 10. Surveillance task: starting in a single location, all 20 vehicles spread throughout the building while maintaining ultrasound and communication range to nearest neighbors.

VI. Conclusions

In both the small-scale and large-scale simulations, the guidance of the vehicles is based on attractive and repulsive gradient forces. These gradient forces are derived from optimal performance indices that trade off minimizing the distance to specified goals, and optimizing the distance between vehicles to maintain a required communication distance.

Using state space control analysis and/or a vector Liapunov method (illustrated in [43]), we have proven that properly designed gradient-force-based control laws are asymptotically connectively stable. Also, we have shown that these gradient-force-based control laws can be used to generate many of the meta-level behaviors such as dispersion, following, clustering, and orbiting. This is a major break-through since we can now design provably stable and convergent control laws for large numbers of autonomous vehicles.

References

- [1] K. Kosuge, T. Oosumi, M. Satou, K. Chiba, K. Takeo, "Transporation of a Single Object by Two Decentralized-Controlled Nonholonomic Mobile Robots," *Proceedings of the 1998 Conference on Robotics & Automation*, Leuven, Belgium, May 1998, 2989-2994.
- [2] H. Yamaguchi, "A Cooperative Hunting Behavior by Mobile Robot Troops," *Proceedings of the 1998 Conference on Robotics & Automation*, Leuven, Belgium, May 1998, 3204-3209.
- [3] F.R. Noreils, "Multi-Robot Coordination for Battlefield Strategies," *Proceedings of the 1992 IEEE.RSJ International Conference on Intelligent Robots and Systems*, Raleigh, NC, July 1992, 1777-1784.
- [4] D. F. Hougén, M.D. Erickson, P.E. Rybski, S.A. Stoeter, M. Gini, N. Papanikolopoulos, "Autonomous Mobile Robots and Distributed Exploratory Missions," *Distributed Autonomous Robotic Systems 4*, ed. L.E. Parker, G. Bekey, J. Barhen, Springer-Verlag 2000, pp. 221-230.
- [5] B. Brumitt and M. Hebert, "Experiments in Autonomous Driving with Concurrent Goals and Multiple Vehicles," *Proceedings of the 1998 International Conference on Robotics & Automation*, Leuven, Belgium, May 1998, pp.1895-1902.
- [6] T. Kaga, J. Starke, P. Molnar, M. Schanz, and T. Fukuda, "Dynamic Robot-Target Assignment – Dependence of Recovering from Breakdowns on the Speed of the Selection Process," *Distributed Autonomous Robotic Systems 4*, ed. L.E. Parker, G. Bekey, J. Barhen, Springer-Verlag 2000, pp. 325-334.
- [7] J.E. Hurtado, R.D. Robinett, C.R. Dohrmann, S.Y. Goldsmith, "Distributed Sensing and Cooperating Control for Swarms of Robotic Vehicles," *Proc. IASTED Conference Control & Applications*, Honolulu, Hawaii, Aug. 12-14, 1998.
- [8] J.S. Jennings, G. Whelan, W.F. Evans, "Cooperative Search and Rescue with a Team of Mobile Robotis," *Proc. IEEE International Conference of Advanced Robotics*, Monterey, CA, 1997.
- [9] S. Goldsmith, J. Feddema, R. Robinett, "Analysis of Decentralized Variable Structure Control for Collective Search by Mobile Robots," SPIE98, *Proc. Sensor Fusion and Decentralized Control in Robotic Systems*, Boston, November 1-6, 1998.
- [10] Y. Uny Cao, Alex S. Fukunaga, and Andrew B. Kahng, "Cooperative Mobile Robotics: Antecedents and Directions," *Proceedings of the 1995 IEEE/RSJ IROS Conference*, 226-234.
- [11] L.E. Parker, "Current State of the Art in Distributed Autonomous Mobile Robotics," *Distributed Autonomous Robotic Systems 4*, ed. L.E. Parker, G. Bekey, J. Barhen, Springer-Verlag 2000, pp. 3-12.
- [12] R.C. Arkin, "Cooperation Without Communication: Multiagent Schema-Based Robot Navigation," *Journal of Robotic Systems* 9(3), 1992, pp. 351-364.
- [13] T. Balch, and R.C. Arkin, "Behavior-Based Formation Control for Multirobot Teams," *IEEE Transactions on Robotics and Automation*, Vol. 14, No. 6, December 1998.
- [14] R.C. Kube, and H. Zhang, "Collective Robotics: From Social Insects to Robots," *Adaptive Behavior*, Vol. 2, No.2, pp. 189-218.
- [15] R.A. Brooks, and A.M. Flynn, "Fast, Cheap and Out of Control: A Robot Invasion of the Solar System," *Journal of the British Interplanetary Society*, Vol. 42, 1989, pp. 478-485.
- [16] R.A. Brooks, "A Robust Layered Control System for a Mobile Robot," *IEEE Journal of Robotics and Automation*, Vol. RA-2, No. 1, March 1986, pp. 14-23.
- [17] T. Fukuda, H. Mizoguchi, K. Sekiyama, F. Arai, "Group Behavior Control for MARS (Micro Autonomous Robotic System)," *Proceedings of the 1999 International Conference on Robotics & Automation*, Detroit, MI, May 1999, pp.1550-1555.
- [18] F.R. Noreils, "Toward a Robot Architecture Integrating Cooperation Between Mobile Robots: Application to Indoor Environment," *The International Journal of Robotics Research*, Vol. 12, No. 1, February 1993, pp. 79-98.
- [19] Q. Chen, and J.Y.S. Luh, "Coordination and Control of a Group of Small Mobile Robots," *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 3, 1994, pp. 2315-2320.

- [20] H. Yamaguchi, and T. Arai, "Distributed and Autonomous Control Method for Generating Shape of Multiple Mobile Robot Group," *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Vol. 2, 1994, pp.800-807.
- [21] H. Yamaguchi, J.W. Burdick, "Asymptotic Stabilization of Multiple Nonholonomic Mobile Robots Forming Group Formations," *Proceedings of the 1998 Conference on Robotics & Automation*, Leuven, Belgium, May 1998, 3573-3580.
- [22] E. Yoshida, T. Arai, J. Ota, and T. Miki, "Effect of Grouping in Local Communication System of Multiple Mobile Robots," *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Vol. 2, 1994, pp. 808-815.
- [23] P. Molnar and J. Starke, "Communication Fault Tolerance in Distributed Robotic Systems," *Distributed Autonomous Robotic Systems 4*, ed. L.E. Parker, G. Bekey, J. Barhen, Springer-Verlag 2000, pp. 99-108.
- [24] F. E. Schneider, D. Wildermuth, H.-L. Wolf, "Motion Coordination in Formations of Multiple Robots Using a Potential Field Approach," *Distributed Autonomous Robotic Systems 4*, ed. L.E. Parker, G. Bekey, J. Barhen, Springer-Verlag 2000, pp. 305-314.
- [25] G. Beni and P. Liang, "Pattern Reconfiguration in Swarms – Convergence of a Distributed Asynchronous and Bounded Iterative Algorithm," *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 3, June 1996, pp. 485-490.
- [26] Y. Liu, K. Passino, and M. Polycarpou, "Stability Analysis of One-Dimensional Asynchronous Swarms", 2001 American Control Conference, Arlington, VA, June 25-27, 2001, pp. 716-721.
- [27] A. Winfield, "Distributed Sensing and Data Collection via Broken Ad Hoc Wireless Connected Networks of Mobile Robots," *Distributed Autonomous Robotic Systems 4*, ed. L.E. Parker, G. Bekey, J. Barhen, Springer-Verlag 2000, pp. 273-282.
- [28] J.P. Desai, J. Ostrowski, V. Kumar, "Controlling Formations of Multiple Mobile Robots," *Proceedings of the 1998 Conference on Robotics & Automation*, Leuven, Belgium, May 1998, 2864-2869.
- [29] J. P. Desai, V. Kumar, J.P. Ostrowski, "Modeling and Control of Formations of Nonholonomic Mobile Robots," *IEEE Transactions on Robotics and Automation*, Vol. 17, No. 6, December 2001, pp. 905-908.
- [30] T. Fukuda, et al., "Evaluation on Flexibility of Swarm Intelligent System," *Proceedings of the 1998 Conference on Robotics & Automation*, Leuven, Belgium, May 1998, 3210-3215.
- [31] B. Brummitt and A. Stentz, "GRAMMPS: A Generalized Mission Planner for Multiple Mobile Robots In Unstructured Environments," *Proceedings of the 1998 Conference on Robotics & Automation*, Leuven, Belgium, May 1998, 1564-1571.
- [32] O. Simonin, A. Liegeois, P. Rongier, "An Architecture for Reactive Cooperation of Mobile Distributed Robots," *Distributed Autonomous Robotic Systems 4*, ed. L.E. Parker, G. Bekey, J. Barhen, Springer-Verlag 2000, pp. 35-44.
- [33] S.I. Roulmeliotis and G. A. Bekey, "Distributed Multi-Robot Localization," *Distributed Autonomous Robotic Systems 4*, ed. L.E. Parker, G. Bekey, J. Barhen, Springer-Verlag 2000, pp. 179-188.
- [34] M. Bozorg, E. M. Nebot, and H.F. Durrant-Whyte, "A Decentralised Navigation Architecture," *Proceedings of the 1998 International Conference on Robotics & Automation*, Leuven, Belgium, May 1998, pp. 3413-3418.
- [35] E. Uchibe, M. Asada, K. Hosoda, "Cooperative Behavior Acquisition in Multi Mobile Robots Environment by Reinforcement Learning Based on State Vector Estimation," *Proceedings of the 1998 International Conference on Robotics & Automation*, Leuven, Belgium, May 1998, pp.1558-1563.
- [36] D. D. Siljak, *Decentralized Control of Complex Systems*, Academic Press, 1991.
- [37] J.T. Feddema, C. Lewis, P. Klarer, R. Eisler, R. Caprihan, "Cooperative Robotic Sentry Vehicles," *Proceedings of SPIE Vol. 3839, Sensor Fusion and Decentralized Control in Robotic Systems II*, Boston, September 19-20, 1999.
- [38] G. D. Smith, *Numerical Solution of Partial Differential Equations: Finite Difference Methods*, Third Edition, Oxford University Press, 1985.
- [39] J. T. Feddema and D. A. Schoenwald, "Decentralized Control of Cooperative Robotic Vehicles," *Proceedings of SPIE*, Vol. 4364, AeroSense, Orlando, Florida, April 16, 2001.

- [40] D. A. Schoenwald, J. T. Feddema, F. J. Opper, "Decentralized Control of a Collective of Autonomous Robotic Vehicles," *Proceedings of 2001 American Control Conference*, Arlington, VA, June 25-27, 2001.
- [41] J. T. Feddema and D. A. Schoenwald, "Stability Analysis of Decentralized Cooperative Controls," *Multi-Robot Systems: From Swarms to Intelligent Automata*, ed. A.C. Shultz and L.E. Parker, Kluwer Academic Publishers, pp. 133-122, 2002.
- [42] J. T. Feddema and D. A. Schoenwald, "Distributed Communication/Navigation Robot Vehicle Network," *Proceedings of World Automation Congress*, Orlando, Florida, June 9-13, 2002.
- [43] J.T. Feddema, C. Lewis, and D.A. Schoenwald, "Decentralized Control of Cooperative Robotic Vehicles: Theory and Application," *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 5, pp. 852-864, October 2002.
- [44] D.A. Schoenwald, J.T. Feddema, "Stability Analysis of Distributed Autonomous Vehicles," *Proceedings of IEEE Conference on Decision and Control*, Las Vegas, NV, pp. 887-892, December 2002.

Specified Dissemination:

Distribution

1 Dr. Doug Gage
DARPA Information Processing Technology Office
3701 North Fairfax Drive
Alexandria, VA 22203-1714

1 Judith Hase Judd
Science & Technology Associates
4100 Fairfax Drive, Suite 910
Alexandria, VA 22203

5 MS0318 Dave Schoenwald, 9216
1 MS0318 Jennifer Nelson, 9216
1 MS1002 Philip Heermann, 15203
10 MS1003 John Feddema, 15211
1 MS1004 Ray Harrigan, 15221
1 MS1005 Kelly Jorgensen, 15222
1 MS1125 Phil Bennett, 15252
1 MS1007 Larry Shippers, 15272
1 MS1010 Steve Roehrig, 15200
5 MS1188 Eric Parker, 15311
5 MS1188 John Wagner, 15311
1 MS0741 Rush Robinett, 6200

1 MS9018 Central Technical Files, 8945-1
2 MS0899 Technical Library, 9616