SANDIA REPORT
SAND2007-0433
Unlimited Release
Printed January 2007

Design Tools for Complex Dynamic Security Systems

Kenneth N. Groom, Raymond H. Byrne, Glenn A. Laguna, Brandon R. Rohrer, David G. Wilson, Rush D. Robinett, III, John J. Harrington, Robert Bickerstaff, and Brian J. Rigdon

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.

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.osti.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

SAND2007-0433
Unlimited Release
Printed January 2007

Design Tools for Complex Dynamic Security Systems

Kenneth N. Groom, Raymond H. Byrne, Glenn A. Laguna and Brandon R. Rohrer
Intelligent Systems and Controls Department

David G. Wilson
Energy Systems Analysis Department

Rush D. Robinett, III
Energy and Infrastructure Futures Group

John J. Harrington and Robert Bickerstaff
Mobile Robots Department

Brian J. Rigdon
Intelligent Systems Principles Department

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

Abstract

The development of tools for complex dynamic security systems is not a straight forward engineering task but, rather, a scientific task where discovery of new scientific principles and math is necessary. For years, scientists have observed complex behavior but have had difficulty understanding it. Prominent examples include: insect colony organization, the stock market, molecular interactions, fractals, and emergent behavior. Engineering such systems will be an even greater challenge. This report explores four tools for engineered complex dynamic security systems: Partially Observable Markov Decision Process, Percolation Theory, Graph Theory, and Exergy/Entropy Theory. Additionally, enabling hardware technology for next generation security systems are described: a 100 node wireless sensor network, unmanned ground vehicle and unmanned aerial vehicle.

Acknowledgment

This work was sponsored by Laboratory Directed Research and Development (LDRD) funding. This SAND report serves as the final report for LDRD project #79831 entitled "Design Tools for Complex Dynamic Security Systems."

First the authors would like to give special thanks to our Sandia associates: Dan Pritchard, 6428, who provided invaluable insight into the issues of base perimeter surveillance. Hung D. Nguyen, 5432, for sharing his knowledge and vision on unattended ground sensors and wireless sensor networks.. And Cal Smith 6420 for his guidance in perimeter security.

Next we would like to thank our Colorado State University partner E. K. P. Chong and his graduate students L. W. Krakow and Y. Li for their contributions to the problem of networked wireless sensor scheduling and Partially Observable Markov Decision Process. Also, we would like to thank David Novick, 6474 for his outstanding work on an environmentally secure enclosure for sensor nodes.

Next we would like to thank members of our autonomous vehicle team: James Holmes who worked with Robert Bickerstaff of 6472 for their ground breaking work with unmanned aerial vehicles. Chris Lewis, at Kansas State University, who expanded our capability with unmanned ground vehicles.

Finally we would like to thank those that provided management for this project. John T. Feddema, our line manager, provided constant guidance and support for our work. Phillip Heermann, our level two, provided us with the overall vision and goals.

Contents

Appendix

List of Figures

List of Tables

Introduction

On July 19, 2004 Philip Heermann convened a meeting to announce the winners of the LDRD proposal request: "Design Tools for Complex Dynamic Security Systems." Approximately fifteen staff members, who had submitted proposals, were invited to attend and most where present. While a combined proposal by Kenneth Groom and Raymond Byrne was selected as the winner, Mr. Heermann communicated that the technical review committee liked elements of other proposals and wished them to be combined in some way.

The proposals for the call were in fact quite diverse. The almost certain reason for this is the vague definition of the word "complexity." One author on the subject says "I can't define what it is but I know it when I see it." To assist the principle investigators (PIs), a precise definition of "complexity" was provided: "A complex system is a large scale distributed system with many networked components, each with many states, and many coupled states between components. They have non-linear, often unknown dynamics/states. And they require attention to multiple disciplines." Further, the desired attributes of an engineered complex system was given as: "stable, self organizing, adaptive and/or learning."

The technical goal of the LDRD was given as: "Develop the core competencies needed to design and deploy complex physical systems that are robust, limited in behavior and not brittle." To ground the LDRD in reality, the application area of Physical Security was provided. Mr. Heermann and other management felt that achieving this technical goal rested on three pillars: Theory (the main output of the LDRD), Modeling & Simulation, and Physical Experiments (See Figure 1). Finally, teaming with other organizations and activities was considered by management to be very important to the project's success.

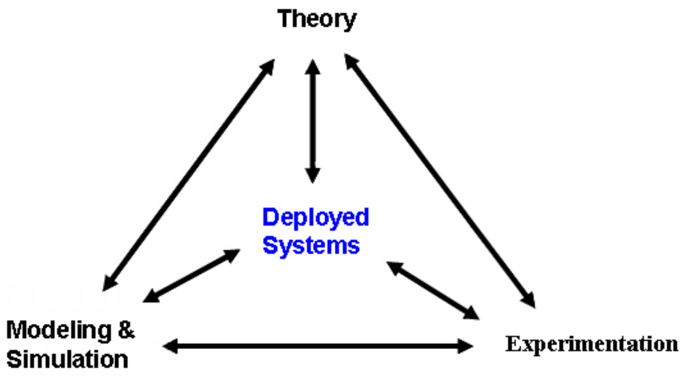


Figure 1. THEORY

The development of tools for complex dynamic security systems is not a straight forward engineering task but, rather, a scientific task where discovery of new scientific principles and math is necessary. For years, scientists have observed complex behavior but have had difficulty understanding it. Prominent examples include: insect colony organization, the stock market, molecular interactions, fractals, and emergent behavior. Engineering such systems will be an even greater challenge.

As described previously, our target application for complexity technology was physical security. While today the Robotics Center is part of the Security Systems and Technology Center (6400), at the time it was not and relationships had to be formed. Here we describe our efforts to team with 6400, and other organizations, and how they influenced our vision of security systems of the future. This vision had a direct effect on the theoretical approaches taken to engineered complex systems.

One of our first contacts was with Cal Smith, 4241 Safeguards and Security. This group provides the “guards and gates” for Sandia. We toured the Coyote Canyon test area where Cal had set up equipment to secure the area. Cal’s primary concern was that an intruder could observe the test from the rim of the canyon. In addition to security cameras placed at strategic positions, Cal planned to rent a helicopter and continuously over-fly the area during the test. This was obviously very expensive. Cal was very interested in inexpensive and widely dispersed wireless sensor networks so that his team would have a warning of an approaching intruder.

Our next contact was with Dan Pritchard of 6428, Security Systems and Technology. Having just completed the TALON ”grand challenge”, they were beginning a one year grand challenge entitled ”Virtual Perimeter Security.” We attended the kickoff meeting and many subsequent meetings. One of their primary concepts was that of a wireless sensor network arranged around the perimeter of a given base or facility. They were not concerned with the sensors being covert. In fact they wanted the sensors to be seen so as to provide a deterrent effect. They were also interested in ”annunciator” technology to provide a security officer with an informative computer interface.

Contact was also made with Hung Nguyen of 5432, Integrated Military Systems Development. Hung had also worked on the TALON project and considers unattended ground sensors, or UGS, to be his group’s specialty. Since his missions tend to be clandestine, his interest is in covert sensors placed in remote locations. He envisioned a ”coffee can” like sensor capable of fusing data from low level and high level sensors.

Our next stop was the ”Cooperative Monitoring Center”, 6900. They worked a great deal with border security, both in the US and in the former Soviet Union, where they were interested in preventing the flow of nuclear materials out of the country. They had deployed a number of wireless sensor networks and were very helpful on the practical aspects of such a deployment. 6900 currently runs a sensor network deployment on the south side of the RVR.

Many others in the security area were interviewed including: Dan Rondeau, Steve Ortiz, Rebecca Horton, Regan Stinnett, and Mike Toscano.

From these discussions a picture of the challenges faced by those in the security area began to surface. Figure 2 displays a taxonomy used by the military for sensor fusion. Overlaying the taxonomy are colored bars indicating the contributions made by various projects at Sandia to the respective areas. What was clear was that little research was being done to automatically identify the situation, assess the threat, and perhaps most importantly, provide feedback to the system in order to improve the capability to protect assets.

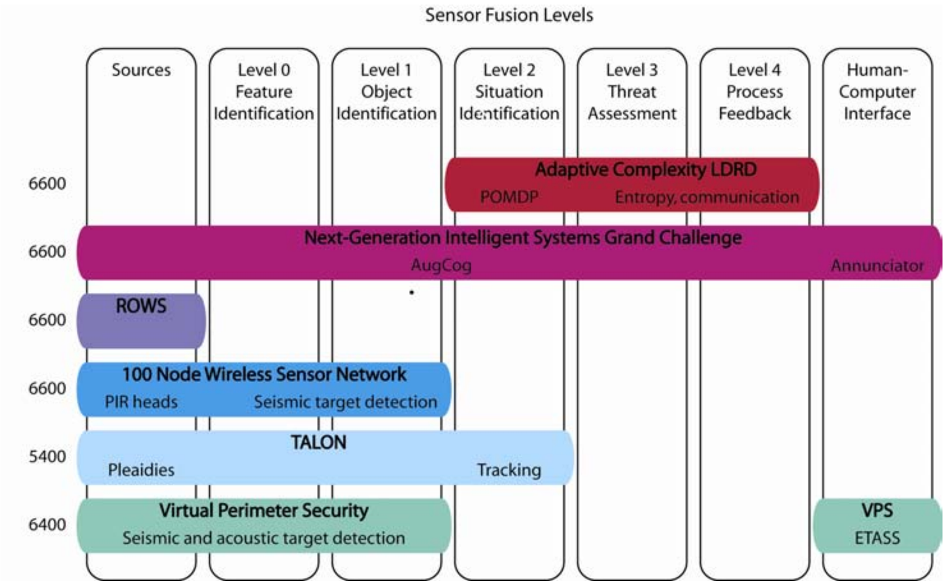

Sensor Fusion Levels
Sources
Level 0 Feature Identification
Level 1 Object Identification
Level 2 Situation Identification
Level 3 Threat Assessment
Level 4 Process Feedback
Human-Computer Interface
6600
Adaptive Complexity LDRD
POMDP
Entropy, communication
6600
Next-Generation Intelligent Systems Grand Challenge
AugCog
Annunciator
6600
ROWS
6600
100 Node Wireless Sensor Network
PIR heads
Seismic target detection
5400
TALON
Pleaidies
Tracking
6400
Virtual Perimeter Security
Seismic and acoustic target detection
VPS
ETASS

Figure 2. Sandia security project space

To make these concepts more concrete, a notional security system was developed. The system, shown in Figure 3, would consist of a very large number of inexpensive sensor nodes widely distributed through the area to secure. These low power nodes would provide the initial detection and tracking of an intruder. Augmenting these sensors would be dynamic assets such as UGVs, UAVs, and remotely/robotically operated weapons systems (ROWS).

All the aforementioned components are being considered by the military and others for security. However, as the number of these assets grows, management becomes overwhelming. What technologies would make such a complex system possible?

Our efforts to team in the complexity area were less productive than those for security, primarily due to the loose definition of the term. We attended brown bags hosted by Bob Floran, met with interested members of the Advanced Concepts Group, and even attended a summer program run by the New England Institute. We found that some people, followers of Bar-Yam, viewed complexity as a human interactions problem. Some considered it

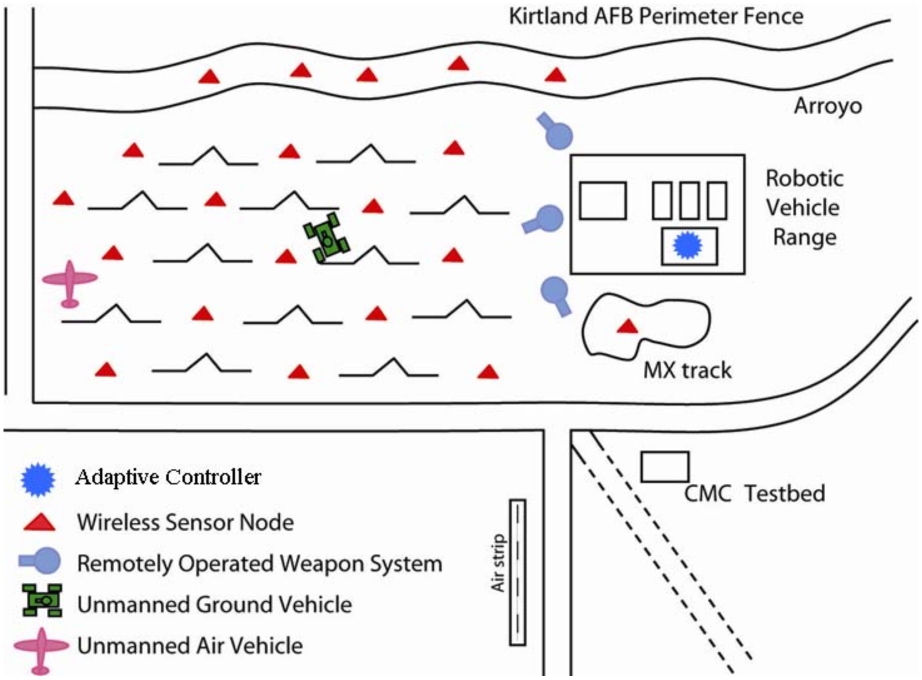
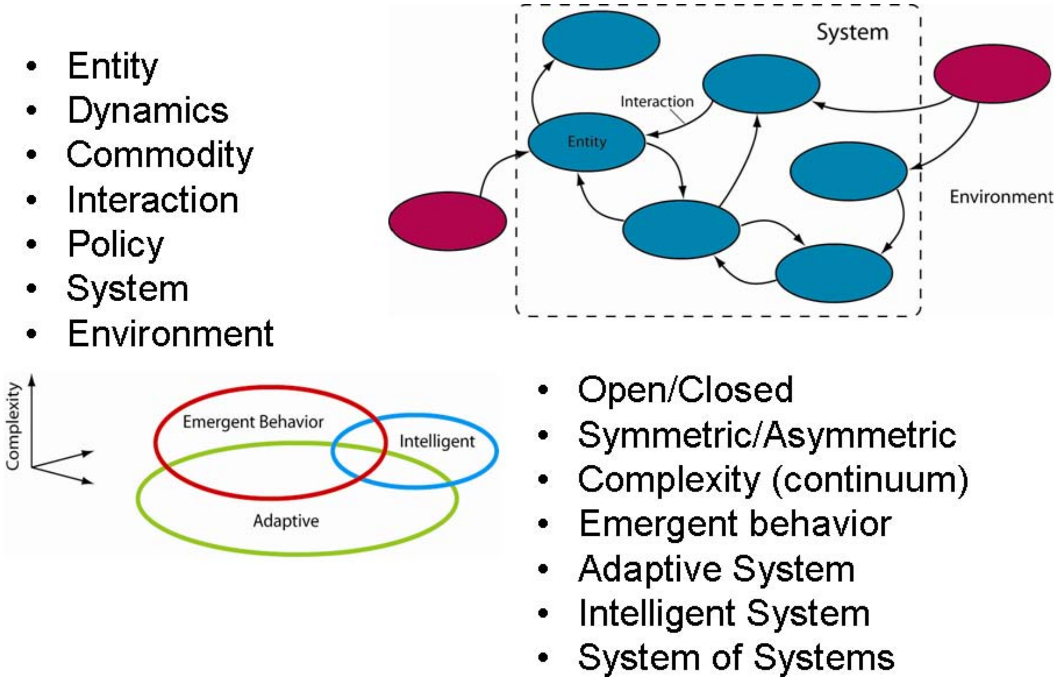


Figure 3. System consisting of large number of inexpensive sensor nodes

essential to the next generation Computer Aided Design (CAD) of sophisticated devices. Still others focus on creating a simulation environments were complexity can be observed. Reviewing some of the popular books on the subject led to the definitions shown in figure 4. One of the common elements in most definitions of complexity is the interaction of many entities in a system. To narrow the area of interest, we stuck with the definition provided by Phillip Heermann at the kick-off meeting: “A complex system is a large scale distributed system with many networked components, each with many states, and many coupled states between components. They have non-linear, often unknown dynamics/states.” The notional security system fits this definition well.

The first pillar of the LDRD, and the initial section of this report, involves the theoretical concepts we planned to employ to engineer complex dynamic security systems. Three techniques were chosen. Chapter 1, Partially Observable Markov Decision Process, provides a method to make decisions in a security system that are optimal and non-myopic. Chapter 2, Percolation and Graph theory, provides tools to analyze the connectivity and performance of networks, including robustness to attacks. Chapter 3, Exergy/entropy theory, provides tools to investigate stability of nonlinear and complex systems and to help investigate the stability of a collective group of dynamic security system components.

The second pillar of the LDRD, and the middle section of this report, involves the simulation tools used to help develop our complexity theories. Because an implementation of POMDP requires a close link to the environment of interest, the decision was made early to



- Entity
- Dynamics
- Commodity
- Interaction
- Policy
- System
- Environment

- Open/Closed
- Symmetric/Asymmetric
- Complexity (continuum)
- Emergent behavior
- Adaptive System
- Intelligent System
- System of Systems

Figure 4. Complexity definitions

port the POMDP code to the Umbra simulation environment. Exergy and Percolation theories, however, were more amenable to developed in the abstract and were thus simulated in MATLAB. Chapter 4 describes details of these simulations.

The third pillar of the LDRD, and the final section of this report, involves experimentation designed to demonstrate the theoretical and simulation portions of the LDRD. Since the theoretical approaches taken to develop complexity tools were so diverse, the experiment was also designed to pull the various ideas together. The experiment, or demo, was based on the security system vision presented previously. It consisted of a large scale wireless sensor network, an unmanned ground vehicle, an unmanned aerial vehicle, and a remotely operated weapon system (using a laser tag weapon, a paintball gun or simulated by a camera and pan/tilt platform), see Figures 3 and 5.

For the given system, POMDP provides a method to selectively power on and off sensor nodes so as to minimize power utilization and tracking error while tracking one or more intruders through the sensor network.

To investigate network centric warfare concepts, exergy/entropy collectives control design and analysis would be an ideal tool for understanding throughput of the combined system including multiple targets and weapons. One of the fundamental questions would be when is the combined system network saturated? Or what are the stability boundaries of the collective system? Fundamentally exergy/entropy collective techniques could also identify what level of optimality would work best for weapon to target assignments subject to minimizing information flow through the network. The demonstration of some of these concepts on the experimental testbed would have given insight on the potential solution to the problem.

Percolation and Graph theory provides a highly efficient method to move messages through the sensor network as well as to the UGV and UAV.

While funding shortfalls prevented a fully integrated demonstration of the system, development of the individual components did conclude successfully. This third part describes the development of the wireless sensor network (Chapter 5), the UGV (Chapter 6) and the UAV (Chapter 7), respectively.

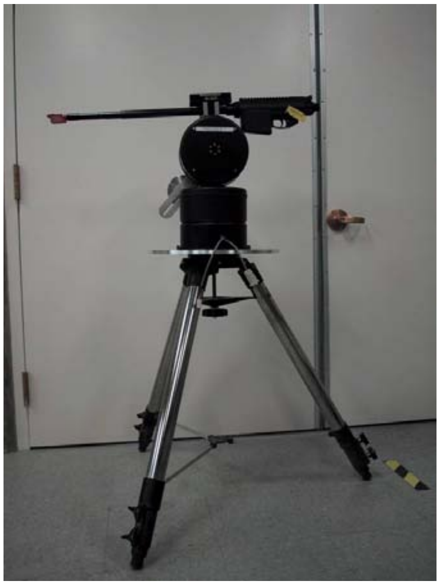
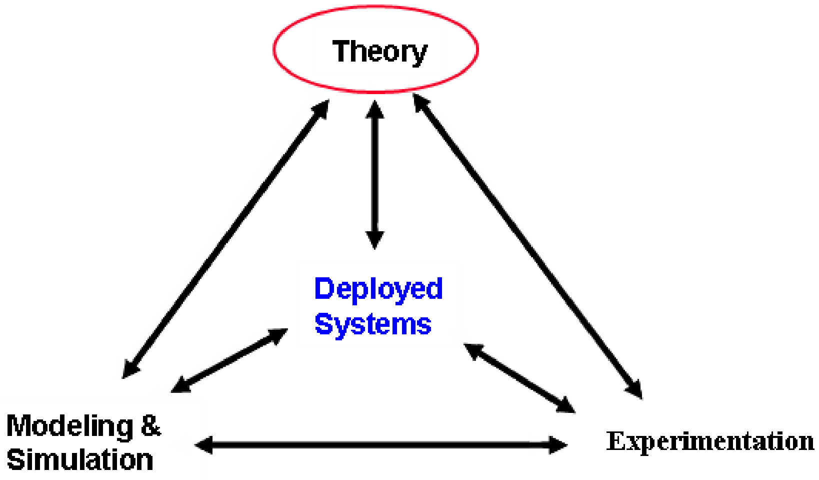


Figure 5. Paintball gun on tripod

Part I

Theory


Theory
Deployed Systems
Modeling & Simulation
Experimentation

Chapter 1

Partially Observable Markov Decision Process

1.1 Introduction

The purpose of sensor scheduling is to select the number and combination of sensors to activate over time. A typical goal is to trade off tracking performance and sensor usage. Sensor-scheduling problems are addressed in [1, 2, 3], where the problem is formulated as an optimization problem to minimize the instantaneous estimation error or to maximize the information gain. Because these schemes only consider the instantaneous performance, they are said to be "myopic." Non-myopic sensor scheduling has gained interest, focusing on the use of stochastic dynamic programming, for example in [4, 5, 6].

We formulate the sensor-scheduling problem as a Partially Observable Markov Decision Process (POMDP) to include long-term performance considerations [4, 5]. The underlying process in the POMDP framework [7, 8] is a controlled Markov process. The sensor-scheduling decision in a POMDP is based on recursively calculating the *belief state*, the posterior distribution of the underlying state given the history of measurements and sensor-scheduling actions. In general, analytical calculation of the belief state is impossible. In this chapter, we employ a Monte Carlo approach that combines two techniques: particle filtering for belief-state estimation, and a simulation-based Q-value approximation method for decision making via "lookahead." Particle filtering [9, 10] is a Monte Carlo method for belief-state estimation. At each time step, the output of the particle filter is a set of particles (samples) that represents the current belief state. The simulation-based Q-value approximation method uses Monte Carlo simulation to evaluate, at the current belief state, the expected cumulative cost of candidate actions as approximations of Q-values, which are then used to select the optimal action. As particle filtering provides a set of particles for the Q-value approximation method to initiate the evaluation, these two techniques dovetail naturally in our approach.

Because of the POMDP formulation, our approach can take both long-term and short-term costs into consideration. Furthermore, because it is a Monte Carlo method, it does not rely on analytical tractability, and therefore it can incorporate sophisticated target dynamics and sensor models.

In our previous work [11, 12, 13], particle filtering and policy rollout (a simulation-based Q-value approximation method) were applied to the scenario with single sensor activation for single-target tracking. This work is an extension of our previous work to the multisensor multitarget case. (A preliminary version of this paper was presented in [14].) A number of new issues are solved in this extension. In the belief-state estimation, there is the *data association* problem to decide which target is associated with each observation, and the *sensor data fusion* problem to combine information from multiple sensors. We develop an innovative tracking algorithm that integrates multisensor data fusion, multitarget data association, and particle-filter tracking techniques. For decision making, we develop a variation of policy rollout called Completely Observable rollout (CO-rollout). In this paper, the problem formulation and the sensor model are different from those of [11, 12, 13]. We use the algorithm here for studying the trade-off between tracking error and the sensor usage cost.

Our experiments involve the activation of up to two of four sensors to track two targets. The results verify that our integrated multisensor Joint Probabilistic Data Association (MS-JPDA) and particle-filter tracking algorithm works correctly, illustrate that our Q-value approximation method is effective in improving the total cost in heterogeneous sensor networks, and show the trade-off between tracking performance and sensor usage cost.

1.2 Problem Formulation

A POMDP [7, 8] is specified by its state space $\mathcal{X}$, action space $\mathcal{U}$, observation space $\mathcal{Y}$, state transition law $K(X'|X,u)$ $(X,X'{\in}\mathcal{X}$ and $u{\in}\mathcal{U})$, observation law $L(Z|X,u)$ $(Z{\in}\mathcal{Y})$, initial state distribution p^0, and one-step cost function $r(X,u)$. It is basically a Markov Decision Process (MDP) where the state is only partially observable through L.

Starting at time 0 from the initial state X^0 with known distribution p^0, a POMDP evolves as follows. At time step k, the state of the system is X^k and the observation Z^k is available. Then the action u^k is selected and a cost $r(X^k,u^k)$ is incurred. After that the system moves to the state X^{k+1} according to the transition law $K(X^{k+1}|X^k,u^k)$, and an observation Z^{k+1} is generated randomly according to the observation law $L(Z^{k+1}|X^{k+1},u^k)$.

Since the state is not directly observable, a POMDP keeps track of the belief state b^k, defined as $p(X^k|I^k)$, the posterior probability distribution of state X^k conditioned on the observation and action history $I^k := (p^0,u^0,Z^1,u^1,Z^2,\cdots,u^{k-1},Z^k)$. The goal here is to choose an action u^k, based on the belief state b^k, from a set of available actions $U(b^k)$ to minimize the expected total cost. A *policy* is defined as a sequence of mappings from belief states to actions $\pi = \left\{\pi^k\right\}$.

Let the expected total cost starting from initial belief state p^0 and using policy π over a horizon of H steps to be $J_H(p^0,\pi) = E\left(\sum_{k=0}^{H-1} r(X^k,u^k) \mid p^0,\pi\right)$, where $u^k = \pi^k(b^k)$, and the expectation is taken over all possible state and observation sequences. The objective here is to find an optimal policy $\pi^* = \left\{\pi^{*k}\right\}$ to minimize $J_H(p^0, \pi)$.

We define the cost at belief state b^k by taking action u as

$$g(b^k, u) = \int r(X, u) b^k(X) dX.$$

Then the Q-value of action u at belief state b^k is

$$Q_{H-k}(b^k, u) = g(b^k, u) + E\left(J^*_{H-k-1}(b^{k+1}) | b^k, u\right), \tag{1.1}$$

where $J^*_{H-k-1}(b^{k+1})$ is the optimal value over $H-k-1$ time steps starting at the next belief state b^{k+1}. Bellman's optimality principle for POMDPs states that the minimum expected total cost is given by $J_H(p^0, \pi^*) = \min_u Q_H\left(p^0, u\right)$, and the policy that selects action $\pi^{*k}(b^k) = \arg\min_u \left(Q_{H-k}(b^k, u)\right)$ at step k is optimal. Because the Q-value of an action summarizes the future cost of taking this action, Bellman's principle gives rise to a control approach called "lookahead."

When H is very large, the optimal policy can be assumed to be stationary. In this case, the optimal policy can be approximated by assuming, at each time step, the remaining horizon is still H steps. Hence, the optimal action is given by

$$\pi^*(b^k) = \arg\min_u \left(Q_H(b^k, u)\right). \tag{1.2}$$

This approach is called *receding horizon control*. The resulting optimal policy chooses the action that minimizes the Q-value over a horizon of H steps at any current belief state.

In the sensor-scheduling scenario studied here, there are M sensors distributed in a sensor field to track T targets. The central controller collects and processes data from these sensors, and manages the sensor activation status. Our goal is to select the number and combination of sensors to trade off tracking accuracy and sensor usage. We formulate this POMDP as follows.

1.2.1 System state, action, and state transition law

The system state vector consists of the state S^k of T targets, the state a^k of M sensors, and the state F^k of the filter. At time step k, the system state is written as

$$X^k = [S^k,\ a^k,\ F^k]^\top,$$

$$S^k = [x_1^k,\ \cdots,\ x_T^k]^\top,$$

$$x_i^k = [x_i^k,\ \dot{x}_i^k,\ y_i^k,\ \dot{y}_i^k]^\top, \quad i = 1, \cdots, T,$$

$$a^k = [a_1^k,\ \cdots,\ a_M^k]^\top.$$

Here x_i^k is the state of target i ($i = 1, \cdots, T$), including the target position and velocity in Cartesian coordinates. The vector element a_m^k is the activation status of sensor m ($m = 1, \cdots, M$). If sensor m is activated at time step k, then $a_m^k = 1$; otherwise $a_m^k = 0$. The filter state F^k could describe any filter used in the belief-state estimation. Because the goal of sensor scheduling is to trade off tracking error with sensor cost, we need the state of the filter to estimate the tracking error, and to select sensors for measurement. In the specific setting considered in this paper, the filter is a particle filter, and F^k contains N particles, each being a sample of the target state S^k:

$$F^k = [S_1^k,\ \cdots,\ S_N^k]^\top.$$

An action $u^k = [u_1^k,\ \cdots,\ u_M^k]^\top$ is an M-vector, where $u_m^k = 1$ or $u_m^k = 0$ specifies whether the m-th sensor is active or inactive at time step $k+1$ to generate observation at time $k+1$ based on the system state X^{k+1}.

The state transition law $K(X^{k+1}|X^k, u^k)$ is defined by the state dynamics $X^{k+1} = f(X^k, u^k, v^k)$, where v^k represents the randomness in the state transition. If we assume that targets move independently, then the state dynamics can be decomposed into the state dynamics for each target, state transition of sensors, and state transition of the filter:

$$X^{k+1} = [f^t(x_1^k,\ v_1^k),\ \cdots,\ f^t(x_T^k,\ v_T^k),\ f^a(u^k),\ f^f(F^k)]^\top.$$

The sensor state transition is given by

$$a^{k+1} = f^a(u^k) = u^k.$$

The progression of the filter state is uniquely defined by the particle-filter algorithm given in the subsection 1.3.1.

The target motion model $f^t(x_i^k,\ v_i^k)$, $i = 1, \cdots, T$, used in this specific example is the Nearly Constant Velocity (NCV) model [15] with Gaussian acceleration uncertainty:

$$x_i^{k+1} = f^t(x_i^k,\ v_i^k) = \begin{bmatrix} 1 & T_s & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 1 \end{bmatrix} x_i^k + \begin{bmatrix} \frac{T_s^2}{2} & 0 \\ T_s & 0 \\ 0 & \frac{T_s^2}{2} \\ 0 & T_s \end{bmatrix} \begin{bmatrix} v_{i,x}^k \\ v_{i,y}^k \end{bmatrix}. \tag{1.3}$$

Here, the noise $v_{i,x}^k$ and $v_{i,y}^k$ represent the acceleration uncertainty of target i in the x and y directions, respectively. We assume that they are independent and zero-mean Gaussian.

1.2.2 One-step cost

The one-step cost $r\left(X^k, u^k\right)$ combines the target tracking error and sensor usage cost. In our experiments, we use

$$r\left(X^k, u^k\right) = \alpha \sum_{i=1}^{T} \left[(\widehat{x}_i^k - x_i^k)^2 + (\widehat{y}_i^k - y_i^k)^2\right] + \sum_{m=1}^{M} \left(c_m^{\text{usage}} \cdot a_m^k + c_m^{\text{start}} \cdot (a_m^k - u_m^k)^2\right). \quad (1.4)$$

Here $(\widehat{x}_i^k, \widehat{y}_i^k)$ is the estimated position of target i determined by the filter state, c_m^{usage} and c_m^{start} are the sensor operation cost (per step) and starting/stopping cost of sensor m, respectively, and α is a weight factor to adjust the relative importance of tracking error and sensor cost. We explore the effect of varying α in Section 1.4.

1.2.3 Observation and observation law

The observation law $L(Z^k|X^k, u^{k-1})$ depends on the sensor model. We use the *associated sensor model* [16], which dominates the tracking and data fusion literature. In this model, each sensor measurement consists of a number of observations that can be either valid measurements from established targets, or false alarms from clutter or emerging targets. The overall observation Z^k is the collection of observations from all the active sensors, and each sensor scan may output several observations. Assuming that at time k, the output Z_m^k of sensor m includes N_m observations, the overall observation can be written as

$$Z^k = \left\{Z_m^k : m \text{ satisfies } a_m^k = 1\right\}, \qquad Z_m^k = \left\{z_{m,1}^k,\ \cdots,\ z_{m,N_m}^k\right\}.$$

Associated with each sensor is a coverage area. Within each coverage area, the quality of the measurement is related to the distance between the target and the sensor. Let $r(i,m)$ denote the distance between target i and sensor m. If $r(i,m) < R_{\min}$ or $r(i,m) > R_{\max}$, sensor m does not generate any measurement from target i. Otherwise, sensor m generates at most one measurement based on the state of target i with detection probability P_d.

Each single observation $z_{m,i}^k$ $(j = 1, \cdots, N_m)$, includes range, angle, and range rate of either a target or a false alarm. If $z_{m,j}^k$ is generated from target i, then it depends on the state of target i as follows:

$$z_{m,j}^k = \left[r_{m,j}^k,\ \theta_{m,j}^k,\ \dot{r}_{m,j}^k\right] = h\left(x_i^k, w_j^k(i,m)\right),\ \ j = 1, \cdots, N_m, \quad (1.5)$$

with

$$r_{m,j}^k = \sqrt{(x_i^k - sp_x(m))^2 + (y_i^k - sp_y(m))^2} + w_{j,r}^k(i,m),$$

$$\theta_{m,j}^k = \tan^{-1} \frac{y_i^k - sp_y(m)}{x_i^k - sp_x(m)} + w_{j,\theta}^k(i,m),$$

$$\dot{r}_{m,j}^k = \frac{(x_i^k - sp_x(m))\dot{x}_k + (y_i^k - sp_y(m))\dot{y}_k}{\sqrt{(x_i^k - sp_x(m))^2 + (y_i^k - sp_y(m))^2}} + w_{j,\dot{r}}^k(i,m).$$

Here, $sp_x(m)$ and $sp_y(m)$ are the (x,y) positions of sensor m, and $w_{j,r}^k(i,m)$, $w_{j,\theta}^k(i,m)$, and $w_{j,\dot{r}}^k(i,m)$ represent measurement noise, assumed to be Gaussian with zero mean. Since the measurement variance is proportional to $1/\text{SNR}$, and SNR is proportional to r^{-4} [17], the variances $\sigma_r^2(m,r)$, $\sigma_\theta^2(m,r)$, and $\sigma_{\dot{r}}^2(m,r)$ are proportional to $r^4(i,m)$.

We assume that the number of false alarms is Poisson with parameter (rate) β. Spatially, the false alarms are distributed uniformly throughout the surveillance volume V_c.

1.3 POMDP Solutions

Our sensor management controller applies a policy that approximately solves the sensor-scheduling POMDP problem. At each decision epoch, the controller takes observations from the sensor system and generates an action to control the sensor activation at the next time step. There are two components in the controller: a "tracker" that takes observations and outputs the belief-state estimation, and a "sensor selector" that takes the belief-state estimation and outputs a sensor-scheduling action.

For the tracker, under certain circumstances, the belief state can be derived analytically. For example, if the transition law and the observation law are linear and the noise is Gaussian, Kalman filtering provides an analytical solution of the belief state. In general, however, the linear-Gaussian assumption does not hold, making it impossible to obtain an analytical solution. Particle filtering is a Monte Carlo tracking algorithm that can be applied to many nonlinear non-Gaussian systems to estimate the belief state. If a particle filter is used as the tracker, the best we can do to implement a practically viable policy is to map the particle-filter output (i.e., the filter state) to actions.

For the sensor selector, the policy specifies, at the current filter state, the approximate best sensor activating action according to the objective function. Recall from (1.2) that in the lookahead approach, the optimal action is the one that minimizes the Q-value for a horizon into the future. Therefore, in the action selector, the key task is Q-value estimation. In general, Q-values are difficult to obtain, especially for problems with large state spaces, such as the target tracking problem. This motivates a Monte Carlo (simulation-based) Q-value approximation method.

Because particle filtering and simulation-based Q-value approximation are both Monte Carlo methods, they combine naturally. Particle filtering provides samples as the starting point of the Monte Carlo simulation in the Q-value evaluation, and the Q-value approximation method provides an action for the particle filter to predict new particles.

1.3.1 Particle filtering for belief-state estimation

If we were to ignore the control variable, the belief state can be updated at each time step using a Bayesian filter in two steps: predicting and updating [9, 10]. The special feature in a POMDP is the control variable u in the state transition law and the observation map. With this control variable, the predicting step is given by [18, 19]

$$p(X^k|I^{k-1},u^{k-1}) = \int p(X^k|X^{k-1},u^{k-1}) \cdot b^{k-1}\, dX^{k-1}, \tag{1.6}$$

and the updating step is given by:

$$b^k = \frac{p(X^k|I^{k-1},u^{k-1}) \cdot p(Z^k|X^k,u^{k-1})}{\int p(X^k|I^{k-1},u^{k-1}) \cdot p(Z^k|X^k,u^{k-1})\, dX^k}. \tag{1.7}$$

Particle filtering is an implementation of the above optimal recursive Bayesian filter by Monte Carlo simulation. This method uses a set of N particles (samples) to approximate the belief state. In our problem, only the target state part (S^k) of the system state (X^k) needs to be estimated, so each particle $S_i, i = 1, \cdots, N$, represents the states of targets.

$$p(S|I) \approx \sum_{p=1}^{N} w_p \delta(S - S_p).$$

Here δ is the Dirac delta function, and w_p is the importance weight of particle S_p.

In a particle filter, the filter state is recursively updated when new observations become available. There are many variations of particle-filtering algorithms. We use the Sampling Importance Resampling (SIR) algorithm (also called the standard particle filter). In SIR, filter state update is done in three steps. First, new particles are generated from the previous particles by sampling from the kinematic prior distribution, $p(S^k|S^{k-1},u^{k-1})$. Then, particle weights are updated according to its agreement with the observation:

$$w_p^k = p(Z^k|S_p^k,u^{k-1}).$$

Finally, a resampling step may be used to prevent particle degeneracy.

In our previous work [11, 12, 13], we used the SIR particle-filtering algorithm for single-sensor single-target tracking. For multi-target tracking, there is a problem of data association: to determine which observation is associated with which target. This problem comes from the associated measurement model, where the dependence of an observation on target state is explicitly given only for associated target-observation pairs. In terms of the particle filter, the problem is that $p(Z^k|S_p^k, u^{k-1})$ is not directly available from $h\left(x_i^k, w^k(i,m)\right)$ in the multitarget case. To solve this problem, We use the Joint Probability Data Association (JPDA) algorithm. The probability calculation of the target-observation association hypotheses in JPDA provides a good connection between $p(Z^k|S_p^k, u^{k-1})$ and the observation law for known target-observation pairs [20].

We consider the single-sensor case first (assuming it is sensor m) and will incorporate multisensor data fusion later. Let H_l, $l = 1, \cdots, N_H(m)$, be all the possible association hypotheses. The number of hypotheses, $N_H(m)$, is determined by the number of targets T and the number of observations N_m. Let $H_l(i)$ be the observation index in $\{0, 1, \cdots, N_m\}$ assigned to target i according to association hypothesis H_l, while 0 means that target i is not detected. Observations that are not assigned to any target are considered false alarms. Using the total-probability theorem, $p(Z_m^k|S_p^k, u^{k-1})$ can be calculated as follows:

$$p(Z_m^k|S_p^k, u^{k-1}) = \sum_{l=1}^{N_H(m)} p(Z_m^k, H_l|S_p^k, u^{k-1}). \tag{1.8}$$

In JPDA, the probability $p(Z_m^k, H_l|S_p^k, u^{k-1})$ is calculated using

$$p(Z_m^k, H_l|S_p^k, u^{k-1}) = \beta^{n_f(H_l)} \cdot (1 - P_d)^{T - n_d(H_l)} \cdot P_d^{n_d(H_l)} \prod_{i:H_l(i) \neq 0} e_m(i, H_l(i)). \tag{1.9}$$

Here, $n_f(H_l)$ and $n_d(H_l)$ are the number of false alarms and detected targets in the association H_l, and $e(i, H_l(i))$ is the likelihood of getting observation $H_l(i)$ from target i, determined by the distribution of the measurement noise. (See Chapter 6 in [21] for details and examples.) For the sensor model in Subsection 1.2.3, the likelihood of getting observation j at sensor m from target i is given by

$$e_m(i,j) = \frac{\exp(-d_m(i,j)^2/2)}{\sqrt{(2\pi)^3}\sigma_r(m, \hat{r}_{m,i})\sigma_\theta(m, \hat{r}_{m,i})\sigma_{\dot{r}}(m, \hat{r}_{m,i})}, \tag{1.10}$$

$$d_m(i,j)^2 = \left(\frac{\hat{r}_{m,i} - r_{m,j}}{\sigma_r(m, \hat{r}_{m,i})}\right)^2 + \left(\frac{\hat{\theta}_{m,i} - \theta_{m,j}}{\sigma_\theta(m, \hat{r}_{m,i})}\right)^2 + \left(\frac{\hat{\dot{r}}_{m,i} - \dot{r}_{m,j}}{\sigma_{\dot{r}}(m, \hat{r}_{m,i})}\right)^2.$$

If multiple sensors are activated simultaneously, multisensor fusion should be used to improve the tracking accuracy. The extension from JPDA to parallel-implemented multisensor JPDA (MS-JPDA) is given in [22]. Later, *sequential* MS-JPDA [23] was studied, and was shown to be more efficient computationally and superior in tracking performance than the parallel implementation. In the sequential MS-JPDA, the observations from sensors are processed one sensor at a time. After each sensor reading, JPDA and Kalman filtering are used to compute an intermediate belief-state estimate, and then the observation from the next sensor is used to further improve this intermediate state estimate. Here we combine sequential MS-JPDA with particle filtering, i.e., we update the filter state after each sensor reading using a particle filter, instead of a Kalman filter.

The integrated sequential MS-JPDA and particle filtering algorithm is as follows:

1. Initialization.
   For each particle $p = 1, \cdots, N$, sample S_p^0 from initial distribution, and set $w_p^0 = 1/N$. Set $k = 1$.

2. Prediction.
   For each particle $p = 1, \cdots, N$,

   (a) Predict $x_{p,i}^k$ by $f^t(x_{p,i}^{k-1}, v_{p,i}^{k-1})$ for each target $i = 1, \cdots, T$.

   (b) Construct a new particle using
   $S_p^k = [x_{p,1}^k, \ \cdots, \ x_{p,T}^k]^\top$.

3. Weight update.
   For each particle $p = 1, \cdots, N$,

   (a) For each sensor $m = 1, \cdots, M$, calculate weight $w_{p,m}^k = P(Z_m^k | S_p^k, u^{k-1})$ using (1.8), (1.9), and (1.10).

   (b) Calculate weight $w_p^k = \prod_m w_{p,m}^k$.

   Normalization: for $p = 1, \cdots, N$, $\widetilde{w}_p^k = \frac{w_p^k}{\sum_{p=1}^N w_p^k}$.

4. Resampling. Select N particles from S_p^k based on $\widetilde{w}_p^k$.

5. Let $k = k + 1$, and go to Prediction step.

As the number of particles becomes very large, the particle filter approaches the exact Bayesian update in (1.6) and (1.7). However, constrained by computer memory and computational complexity, the number of particles used in simulations is necessarily limited. This causes a problem when the variance of measurements is much smaller than the variance of the distribution represented by the particles. In this case, most particles have very small weight (sometimes even smaller than the smallest number a computer can handle) except a few ones that are closest to the measurement. This then causes loss of particle diversity at the resampling step. As time goes on, tracks may get lost. To ensure the stability of the tracker, the number of particles must be large enough to match the measurement accuracy. For our sensor model, the variance of measurements is proportional to r^4, varying in a very wide range. When the target is very close to the sensor, the number of particles needed is beyond our computer's ability. Our solution to this problem is to use a fixed number of particles (2000 in our simulations) and change the likelihood calculation $e_m(i,j)$ slightly. If $r(i,m) < R_0$, where R_0 is some predefined threshold value between $R_{\min}$ and $R_{\max}$,we use fixed $\sigma_r(m,R_0)$, $\sigma_\theta(m,R_0)$, and $\sigma_{\dot{r}}(m,R_0)$ in (1.10) (instead of the very small σ_r, σ_θ, and $\sigma_{\dot{r}}$) to artificially increase the particles' weights and the diversity after resampling. With this change, the tracker is much more stable and has much smaller tracking error.

1.3.2 CO-Rollout for Q-value approximation

Recall that in the "lookahead" procedure, the optimal action minimizes the Q-value defined in (1.1). There are several Q-value approximation methods available [7, 24, 25]. We use the "policy rollout" method given in [24, 25]. The basic idea of rollout is to replace $J^*_{H-1}(b^{k+1})$ in (1.1) by $J^{\pi_b}_{H-1}(b^{k+1})$, the objective function value corresponding to a base policy π_b. This base policy should be a reasonable heuristic policy for the problem, and makes $J^{\pi_b}_{H-1}(b')$ relatively easy to compute. Since $J^*_{H-1} = \min_\pi J^\pi_{H-1}$, this approximation gives an upper bound on the true Q-value. The resulting policy from this rollout Q-value approximation algorithm is defined as

$$\pi(b^k) = \arg\min_u \left\{ g(b^k,u) + E\left(J^{\pi_b}_{H-1}(b^{k+1}) | b^k, u \right) \right\}. \tag{1.11}$$

It has been shown that this policy is at least as good as the base policy [25].

We use Monte Carlo simulation to compute $J^{\pi_b}_{H-1}(b^{k+1})$. Because the belief state is approximated by a set of particles, our method can take advantage of these particles in initiating the simulation. Starting from each particle, we run a simulation: apply action u for the first time step, and apply base policy π_b to the remaining time steps. The estimated Q-value for action u is obtained by averaging the cumulative costs from these simulation runs.

In the rollout Q-value approximation method for POMDPs, we need a base policy that maps belief states to actions, and we need to simulate how the system evolves in the future in response to actions resulting from this base policy. Practical implementations of this involve using a base policy that maps *filter* states to actions. In our case, this involves simulating a policy that maps particle-filter states to actions over time (this is called "rolling out" the policy). Keeping track of particle-filter states in a simulation is unduly burdensome, considering that we need many simulation runs for each candidate action. To overcome this problem, we use a base policy that maps underlying *target* states to actions. In this case, the lookahead simulation involves only keeping track of a *completely observable* version of the system. We call this method Completely Observable Rollout (CO-rollout)[7].

Although the real target state is not known to the controller in practice, remember that we are using the target state information only in the lookahead simulation for Q-value approximation, not in the simulation of a real controller. For target tracking applications, a

CO-rollout base policy is naturally available, e.g. the Closest Point Approach (CPA) that choose the closest sensor to the real target position.

Because we use the CO-rollout approximation, and do not keep track of the particles in the rollout simulation, we cannot use (1.4) to calculate future costs. It is necessary to use some cost surrogation [7]. For this purpose we use a very simple tracker in the lookahead simulation. If sensor m is used to generate the measurement of target i, the target-state estimate of this tracker is just the converted Cartesian coordinates from the polar measurement. The tracking error is given by the variances $\sigma_x^2(m)$ and $\sigma_y^2(m)$, which are functions of $r(i,m)$, $\theta(i,m)$, $\sigma_r(m,r)$, and $\sigma_\theta(m,r)$. (This conversion is given in [26].) When multiple sensors are scheduled to take measurements, the target-state estimates from these sensors are combined by a Gaussian-Markov estimator[25], and the variances are given by $\sigma_x^2 = (\sum_m \sigma_x^{-2}(m))^{-1}$ and $\sigma_y^2 = (\sum_m \sigma_y^{-2}(m))^{-1}$.

Because only radial velocity ($\dot{r}$) is measured, the direction of the target movement (call it ϕ) affect the velocity (v) estimate, and therefore the future tracking error. From $\dot{r} = v \cdot \cos(\phi - \theta)$, the resulting tracking error in x and y can be given by $T_s^2\sigma_{\dot{r}}^2/\cos^2(\phi-\theta)$, while T_s is the time interval between measurements.

Obviously the tracking error of this simple tracker is different from that of the particle filter. To keep the relative contribution of the tracking error to the sensor usage cost appropriate, we need to scale the tracking error of this simple tracker (by a factor of γ) so that the effect of the current sensor scheduling action is properly reflected in the Q-value approximation. To summarize, the tracking error term in lookahead is approximated by

$$\gamma\left(\left(\sum_m \sigma_x^{-2}(m)\right)^{-1} + \left(\sum_m \sigma_y^{-2}(m)\right)^{-1} + \frac{T_s^2\sigma_{\dot{r}}^2}{\cos^2(\phi-\theta)}\right).$$

1.4 Simulation Experiments

In our experiments, there are two moving targets and four sensors (radars) in a surveillance area of 40km×40km. The two moving targets are modeled by an NCV model with acceleration uncertainty $\sigma_x = \sigma_y = 7\text{m/s}^2$. Sensor 0 is at the center of the area, the other 3 sensors are 30km from sensor 0, with a 120° separation from each other. All four sensors have the same sensible area with $R_{\min} = 25$km and $R_{\max} = 100$km, and same start/stop cost 0.1. Sensor 0 is more expensive to use than other sensors, and sensor 3 is more accurate than other sensors. A summary of the sensor parameters are as follows:

Sensor 0: at (0, 0)km, cost 0.5/step, $\sigma_r = 500$m, $\sigma_\theta = 2°$, $\sigma_{\dot{r}} = 6$m/s.

Sensor 1: at $(-30, 0)$km, cost 0.1/step, $\sigma_r = 500$m, $\sigma_\theta = 2°$, $\sigma_{\dot{r}} = 6$m/s.

Sensor 2: at $(15, 26)$km, cost 0.1/step, $\sigma_r = 500$m, $\sigma_\theta = 2°$, $\sigma_{\dot{r}} = 6$m/s.

Sensor 3: at $(15, -26)$km, cost 0.1/step, $\sigma_r = 250$m, $\sigma_\theta = 1°$, $\sigma_{\dot{r}} = 3$m/s.

The sampling interval is $T_s = 2$s, and we simulate for 300s. The target detection probability is $P_d = 0.85$, and the false alarm rate is $\beta = 1.0 \times 10^{-8}$, which corresponds to a false alarm probability of $P_f = 0.47$ in our experiment. We use 2000 particles in the particle filter, and use a receding horizon of 8 steps in the rollout Q-value approximation method.

In the single-sensor single-target tracking problem, the CPA policy is commonly used [11, 12]. Since the measurement variance is proportional to r^4, it is natural to select the closest sensor to the target. But CPA is a “greedy” approach because it does not take into account the different sensor costs or their error statistics.

Let us define the effective distance $\tilde{r}(i,m)$ between target i and sensor m as $r(i,m)$ if $r(i,m) > R_{\min}$, and as $R_{\max}$ otherwise. For the case of one target (target 0) and one sensor, CPA chooses sensor $\arg\min_m \tilde{r}(0,m)$ to measure target 0. This policy can be generalized to include multiple targets, i.e., to choose sensor $\arg\min_m \sum_{i=0}^{T-1} \tilde{r}(i,m)$ as the closest sensor to multiple targets. For the multisensor multitarget tracking problem, we construct a Closest Sensors Policy (CSP) based on the CPA policy, which chooses the action with the minimum sum of distances between all targets and their closest active sensors:

$$u^{\text{CSP}} = \arg\min_u \left(\sum_{i=0}^{T-1} \min_{m:u_m=1} \tilde{r}(i,m) \right). \tag{1.12}$$

For activating at most two sensors to track two targets, CSP activates one closest sensor for each target. If the closest sensors to both targets are the same, only that sensor is activated. To have a fair comparison between the base policy and the CO-rollout policy, the maximum number of activated sensors selected by the CO-rollout policy is also limited to two. Note that as long as $\tilde{r}(i,m)$ is calculated from the underlying target state, the CSP base policy has the property of mapping the states of targets to action, as required by the CO-rollout approximation method.

We compare the performance of the CSP policy (1.12) (with a maximum of 2 sensor activations) and the CO-rollout policy (1.11) using CSP as the underlying base policy. Fig. 1.1 illustrates one example of the true trajectories of the targets (shown by solid lines), the estimates of the target positions (shown by marks), and the selected sensors (shown by shapes of marks) using CSP and CO-rollout, respectively. In this example, target 0 starts at $(2.5, 25)$ and moves south at 222m/s, target 1 starts at $(-5, -20)$ and moves east at 120m/s. We notice that our multisensor multitarget tracker works well in this example: the estimated target positions closely follow the target trajectories, and the target crossing is handled correctly. Next, we notice the differences in the sensor selections of the two policies. First, while CSP activates sensor 0 all the time because its location is closest to either target, CO-rollout only actives sensor 0 for a short period of time, considering its high usage cost. Second, CSP turns on sensor 3 after the target crossing, when sensor 3 becomes the closest sensor to target 0, while CO-rollout turns on sensor 3 before the target crossing, because of its high accuracy of measurement. Sometimes CSP select only one sensor when

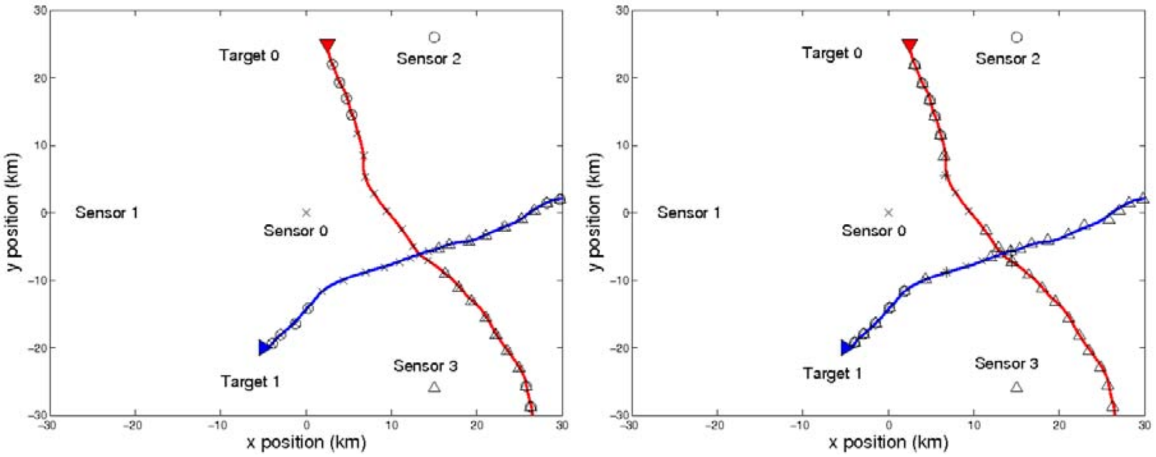


Figure 1.1. Target trajectories and sensor selection of CSP (left) and CO-rollout (right).

it is the closest to both targets, but CO-rollout may find it beneficiary to activate 2 sensors. Yet at other times CSP activates 2 sensors, and CO-rollout decides one is enough.

Fig. 1.2 shows a quantitative comparison of the four policies: CSP with 1 or 2 sensor activations, and CO-rollout with 1 or 2 sensor activations. The tracking error, sensor usage, and total cost are the average values and 90% confidence intervals over 100 random simulations. Here the total cost is defined in (1.4) with $\alpha = 1$. Notice that by activating up to two sensors, the total cost is reduced by 15% in CSP and 30% in CO-rollout. We can also conclude that our CO-rollout policy improves the total cost by 20% for one-sensor activation and by 40% for two-sensor activation.

Obviously there is a trade-off between the tracking error and sensor usage. We study this trade-off in our algorithm and illustrate it in Fig. 1.3. By adjusting the value of α in (1.4), the priority could be put on either the tracking error or the sensor cost. Here combined priority means $\alpha = 1$, tracking priority means $\alpha = 32$, and usage priority means $\alpha = 0.032$. As we can see in Fig. 1.3, as α decreases, tracking error increases, while the usage decreases.

1.5 Conclusion

In this chapter, we formulate the problem of multisensor scheduling for multitarget tracking as a POMDP, and employ a general approach that combines particle filtering and simulation-based Q-value approximation for solving this POMDP. We integrate sequential MS-JPDA and particle filtering for belief state estimation, and use a CO-rollout algorithm based on the Closest Sensors Policy for Q-value approximation. Our simulation experiments involving the selection of multiple sensors for tracking multiple targets illustrate the effectiveness of our approach in trading off tracking error and sensor usage cost.

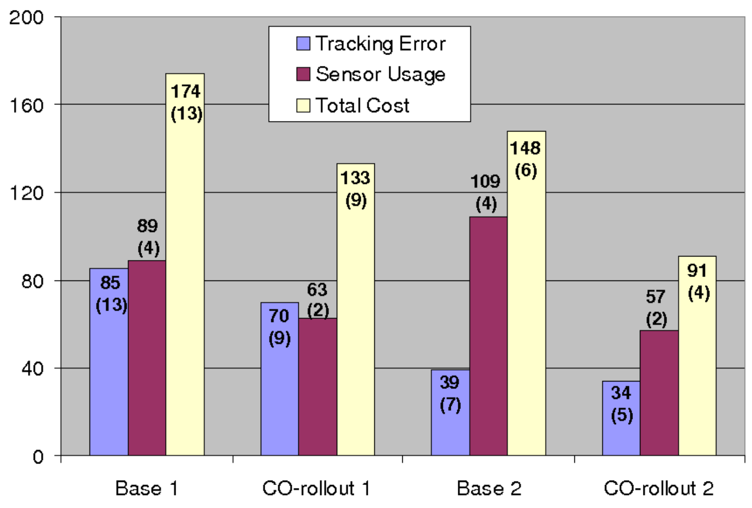


Figure 1.2. Comparison of policies and number of sensors.

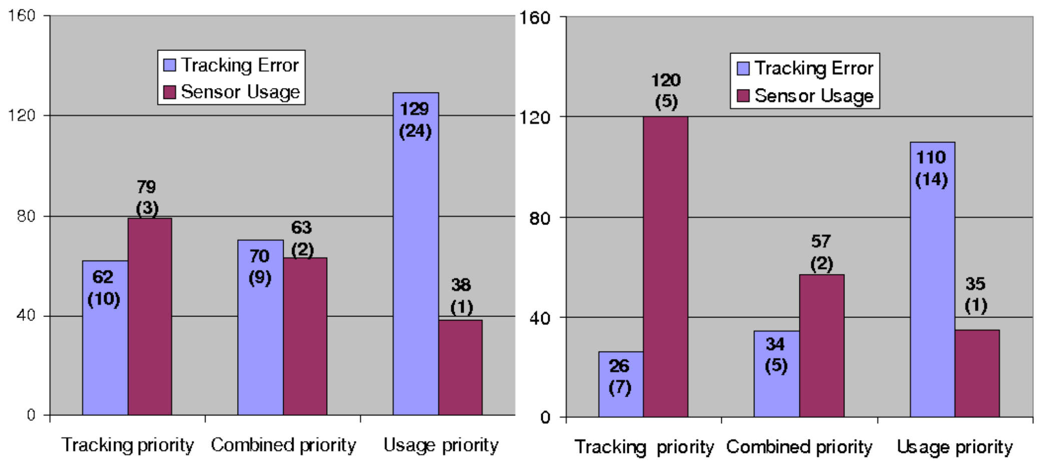


Figure 1.3. Tradeoff between tracking error and sensor usage for CO-rollout 1 (left) and CO-rollout 2 (right).

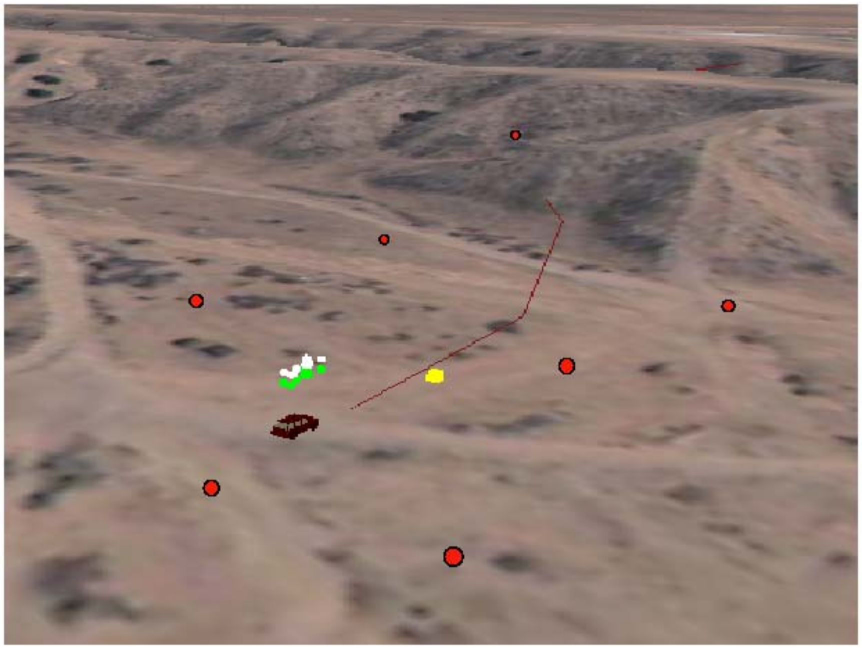


Figure 1.4. Umbra interface

Although we described our simulation-based approach in generic terms, the power of the methodology is realized when it is implemented on a high-fidelity simulator tailored to an application domain. An example of such a simulator is Umbra (see [27] for details). We have implemented our algorithm in Umbra to exploit and leverage the simulation capabilities that have already been embedded into it. Umbra is an agent-based simulation environment developed by Sandia National Laboratories. In this project, sensors, targets, terrains, particles, and POMDP controller, are all agents. Umbra provides a time-stepped execution engine to calculate sequentially the behavior of the agents through time. Since agents in Umbra are composed of modules, simple module insertion and replacement are allowed. Fig. 1.4 shows an example of Umbra output, where a target (a truck in this example) is moving in the terrain along its trajectory (shown by the line). Some sensors (shown by circles) are distributed in the terrain, and one sensor (shown by the square box) is currently active. The particles (shown by dots on top of the target) follow the target using the measurement from the activated sensor.

In the future, we plan to apply our work to a more general sensor management problem, which includes sensor motion control, sensor waveform switching, sensor bandwidth allocation, as well as sensor scheduling. We will also consider applying our approach to a more dynamic environment, like tracking targets in an urban terrain, where a non-myopic approach should significantly improve the tracking performance.

Chapter 2

Percolation and Graph Theory

Percolation theory is concerned with the flow of fluid in random media and was first introduced by Broadbent and Hammersley in 1957 [28]. For a two-dimensional lattice two main types of percolation models are studied: site percolation and bond percolation. Site percolation considers the lattice squares or sites to be the relevant entities while bond percolation considers the edges of the lattice. Under site percolation, a lattice site is open with probability p and closed with probability $1-p$ and fluid flows from open site to open site across the lattice. Under bond percolation, each edge of the lattice is said to be open with probability p and closed with probability $1-p$ and fluid flows through the open edges of the lattice. A cluster is defined as a set of connected entities (edges for bond percolation and sites for site percolation). An infinite cluster is defined as a cluster that reaches from one side of the lattice to the other. One of the more interesting properties observed in percolation models is the phase transition that occurs as p increases and the state changes from a finite number of clusters to one infinite cluster. An example of this phase transition is shown in Figure 2.1. The x-axis is the probability p that a site or edge is open while the y-axis is the probability $\theta(p)$ that any individual site is a member of the infinite cluster. Above a critical probability, p_c, there is a phase transition and the probability of any site belonging to the infinite cluster rapidly changes from a very low probability to a very high probability. For the example shown in Figure 2.1, $p_c \approx 0.4$. Percolation theory studies the existence and value of p_c for which a phase transition occurs, as well as cluster numbers, sizes, and structures. Unfortunately, except for a few distinct cases, there is no general analytical formula for estimating p_c for a given type of lattice structure. Usually p_c is estimated with Monte Carlo simulations.

Graph Theory is a powerful tool for modeling the structure of large scale systems. A graph is represented as $G=(V,E)$ where V is the set of vertices and E is the set of edges. An example of a simple graph is shown in Figure 2.2. For the graph shown in Figure 2.2 $V=\{1,2,3,4,5\}$ and $E=\{(1,2),(2,3),(2,4),(3,5),(4,5)\}$. In a graph with no loops, the degree of a vertex is the number of edges adjacent to that vertex. The diagonal degree matrix D of G is defined as $d_i=|N_i|$ where d_i is the degree of node i. The degree matrix D for the graph shown in Figure 2.2 is

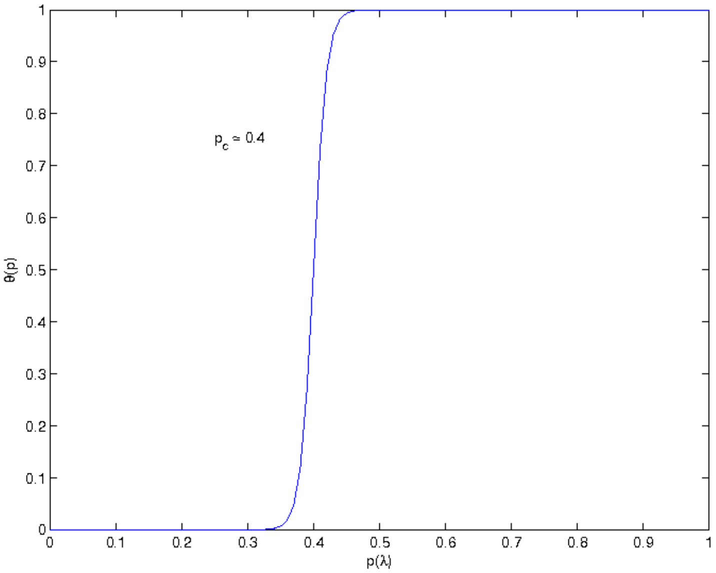


Figure 2.1. Percolation Phase Transition

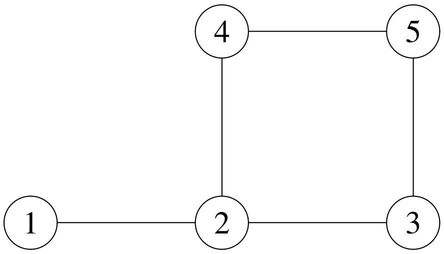


Figure 2.2. Graph Example

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix} \tag{2.1}$$

The adjacency matrix A of a graph G with n vertices is an $n \times n$ matrix with the following structure

$$a_{ij} = \begin{cases} i \neq j & \text{the number of edges joining vertex } i \text{ and vertex } j \\ i = j & \text{the number of loops at vertex } i \end{cases} \tag{2.2}$$

The adjacency matrix A of a graph G is always a symmetric matrix. A complete graph K_n is a graph with n vertices with exactly one edge joining every pair of vertices. The adjacency matrix A for the graph shown in Figure 2.2 is

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \tag{2.3}$$

The Laplacian matrix $L(G)$ of a graph G is defined as $L = D - A$ where D is the degree matrix and A is the adjacency matrix. The Laplacian matrix always has a zero eigenvalue $\lambda_1 = 0$. If G is connected, the second smallest eigenvalue λ_2 is greater than zero. This eigenvalue is named the algebraic connectivity of the graph [29] because it serves as a lower bound on the degree of robustness of the graph to node and edge failures. This follows from the following inequality [29]

$$\lambda_2(G) \leq \nu(G) \leq \eta(G) \tag{2.4}$$

where $\nu(G)$ is the node-connectivity and $\eta(G)$ is the edge-connectivity of a graph. Therefore, a network with high algebraic connectivity is robust to both node and edge failures.

The edge-connectivity and node-connectivity may also be calculated directly from the graph. For a graph $G = (V, E)$ and a set of edges denoted by F , the graph $G - F$ represents the graph obtained from G by deleting all of the edges in F. If a connected graph G becomes disconnected after removing the set of edges F, the set F is called a disconnecting set. A graph is *k-edge connected* if every disconnecting set has at least k edges. The edge-connectivity number $\lambda(G)$ is the minimum size of the disconnecting set in G. For a complete graph with N vertices, $\lambda(G) = N - 1$. Node or vertex connectivity is similarly defined. A set of vertices W in a graph $G = (V, E)$ is defined as a separating set if $G - W$ has more than one component. The connectivity number $\kappa(G)$ of a graph G is defined as the minimum size of the separating set. A graph is *k-connected* if $\kappa(G) \geq k$. In other words, k is the minimum number of vertices that must be removed in order to break a connected graph into two or more components.

The edge-connectivity number $\lambda(G)$ for graph shown in Figure 2.2 is $\lambda(G) = 1$. By removing the edge between nodes 1 and 2 the graph becomes disconnected. The connectivity number $\kappa(G)$ is $\kappa(G) = 1$. By removing node 2 the graph becomes disconnected. The Algebraic Connectivity for the same graph is $\lambda_2 = 0.83$, which satisfies the inequality.

2.1 Literature Review

Percolation theory was originally developed to model the flow of fluids through random media, e.g. water flowing through sedimentary rocks. Recent applications have focused on communications and epidemiology applications [30, 31, 32, 33]. Because percolation theory is concerned with the connections within a lattice, the connectivity of a communications network is a natural application of percolation theory. Areas of research include the reachability of nodes within a network and the probability of flooding a network with a command message.

Graph theory was originally developed by the famous mathematician Leonhard Euler in 1736. The problem that motivated Euler was whether it was possible to walk a route that crosses each of the seven bridges in Königsberg, Prussia exactly once and return to the starting point. Using graph theory, Euler proved that no such path exists. Because graph theory looks at pairwise relationships between objects in a collection, graph theory is well suited for modeling and analyzing different types of networks. An overview of complex networks appears in [34]. A description of the world-wide web's scale-free characteristics appears in [35].

In [36] Watts and Strogatz look at the phase transitions that occur between regular and random graphs. They show that the small-world phenomenon (popularly known as six degrees of separation) can occur in sparse networks with many vertices. They show this effect by starting with regular ring lattice networks with n vertices and k edges per vertex and then rewire each edge with probability p. The effects of random rewiring of links for regular ring lattice networks are shown in Figure 2.3. Olfati-Saber then showed that it is possible to greatly increase the algebraic connectivity in regular complex networks without adding new links or nodes by using the same type of random rewiring [37]. They use this result to imply that the consensus problem can be solved more quickly on certain small-world networks. He also claims that this increase in algebraic connectivity results in an increase in node and edge connectivity. In this project we have shown that the contrary is true. While there is an increase in algebraic connectivity, there is a decrease in node and edge connectivity (but Fiedler's inequality still holds, it is just a conservative bound). Research has also been conducted on maximizing the algebraic connectivity for a given graph [38]. A reference on Laplacian matrices of graphs appears in [39].

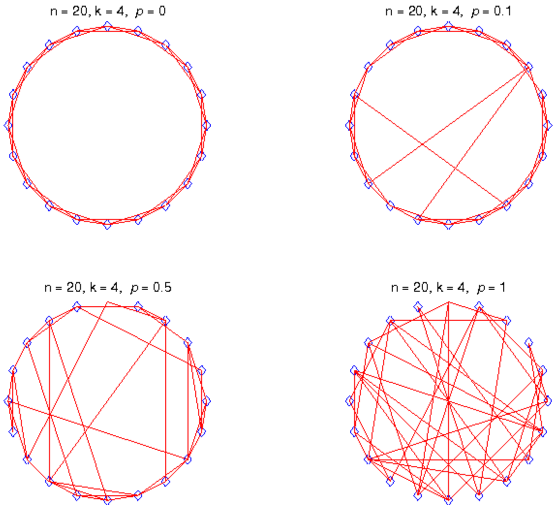


Figure 2.3. Random ring lattice graph $G = C(n,k)$ with $n = 20$, $k = 4$, for $p = 0, 0.1, 0.5, 1$

A $k-separator$ or $k-shredder$ of a $k-$node connected graph is defined as the set of k nodes whose removal results in an unconnected graph. When analyzing network robustness, it is important to identify the nodes that are most vulnerable to bringing down network connectivity. Papers that discuss algorithms for $k-shredders$ include [40, 41].

2.2 Percolation of Network Broadcast Messages

When a broadcast message is presented to a network, a common approach is to have every node repeat the broadcast message once to make sure that every node receives the message. For a system with N nodes and a broadcast message of length T, the minimum length of time required to flood the network with the broadcast message is NT. By applying percolation theory, every node rebroadcasts the message with probability p. The communications range of each node and the probability p must be selected so that every node still hears the broadcast message. However, since p can be much less than one, the minimum time required to flood the network becomes pNT which is much less than NT. Simulation results are presented for a 30 by 30 mesh network (900 nodes) with a communication radius of 4 and a rebroadcast probability $p = 0.25$. The nodes that receive the initial broadcast from the center node $(15, 15)$ are shown in Figure 2.4. Red nodes have not yet received the broadcast message. Yellow nodes are the nodes that will rebroadcast the message. They are randomly chosen with probability $p = 0.25$. Each round of transmissions is illustrated in Figures 2.5-2.11. A summary of the nodes that "talked" is shown in Figure 2.12. Of the 900 nodes, only 223 rebroadcast the message in this simulation. This corresponds well with $p = 0.25$ and is a significant improvement in the time required to flood the network

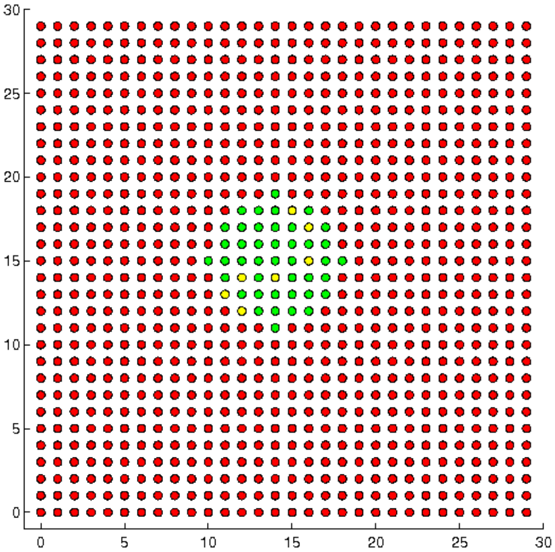


Figure 2.4. Initial Broadcast by One Node

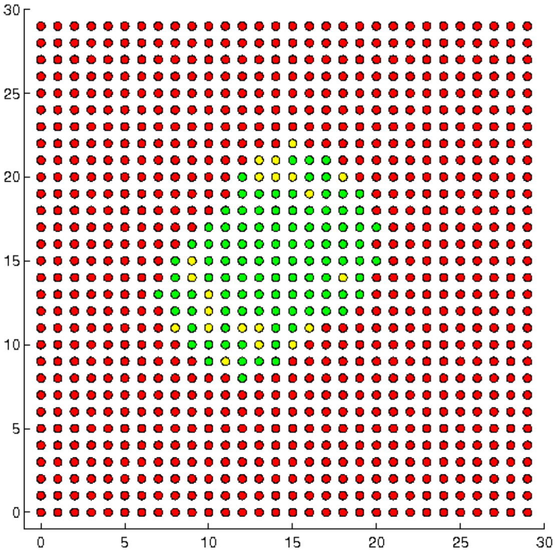


Figure 2.5. Second Round of Transmissions

over the traditional approach of every node rebroadcasting.

2.3 Percolation Applied to Distributed Sensor Networks

One concept for applying percolation theory to distributed sensor networks is summarized by the following hypothesis: the probability of detection for a distributed sensor network will undergo a "phase transition" as the number of nodes is increased. Each sensor node has a cost associated with it. If there were a steep phase transition in the probability of detection curve, there would be minimal benefit in using more sensors than the number necessary to get past the steep part of the curve. This concept was tested using Monte Carlo simulations of a hypothetical distributed sensor network. The parameters for the simulations are summarized in Table 2.1. Each sensor was designed to have a probability of detection $P_D = 0.9$ for each footstep taken within the sensor range of 20 meters. An intruder performs a random walk through the sensor field in each simulation run. Fifty simulations were conducted as the number of randomly placed sensors was increased from 1 to 50. The probability of detection for each run was calculated using

$$P_{detection} = 1 - (1 - P_D)^{\text{(number of sensor hits)}} \tag{2.5}$$

where P_D is the probability of detection of each sensor (0.9 within a range of 20 meters for this simulation). The results of a typical test run are shown in Figure 2.13 (left). The overall probability of detection curve versus the number of sensors is shown in Figure 2.13 (right). Unfortunately, for smaller numbers of sensors there is a roughly linear relationship

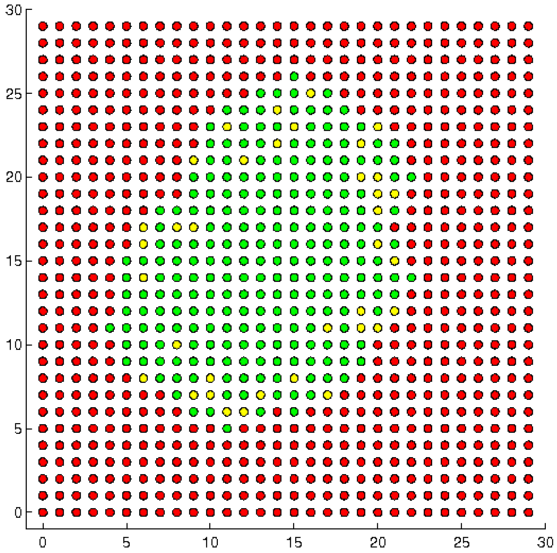


Figure 2.6. Third Round of Transmissions

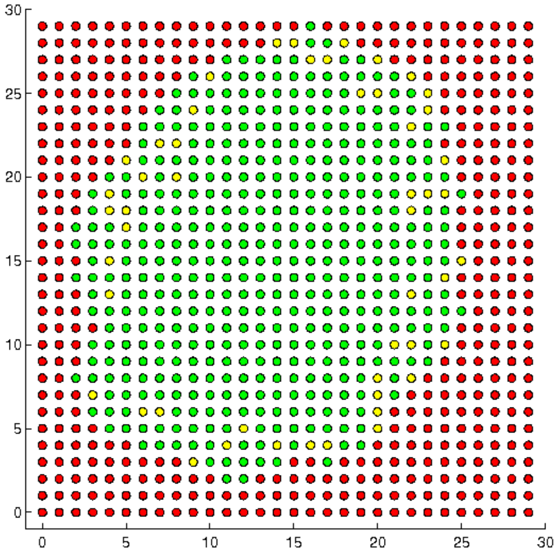


Figure 2.7. Fourth Round of Transmissions

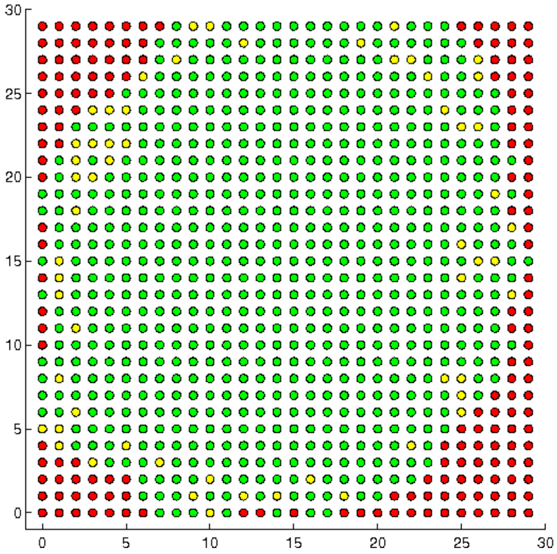


Figure 2.8. Fifth Round of Transmissions

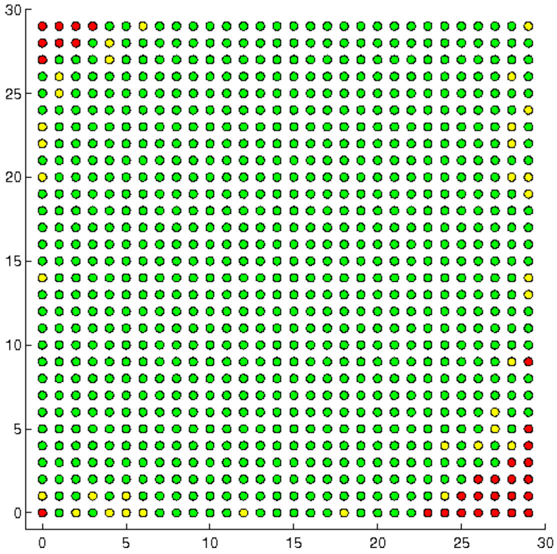


Figure 2.9. Sixth Round of Transmissions

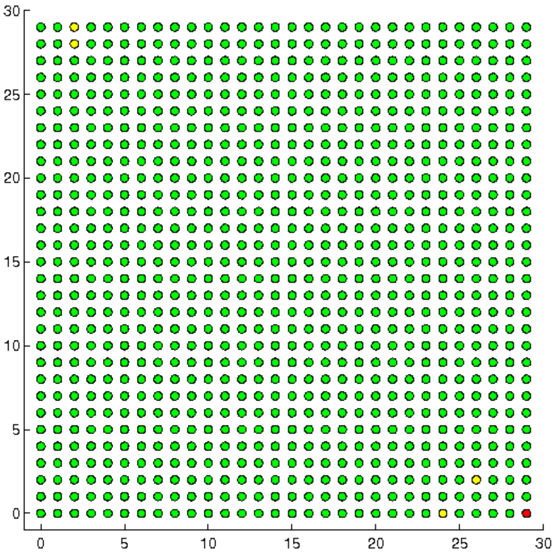


Figure 2.10. Seventh Round of Transmissions

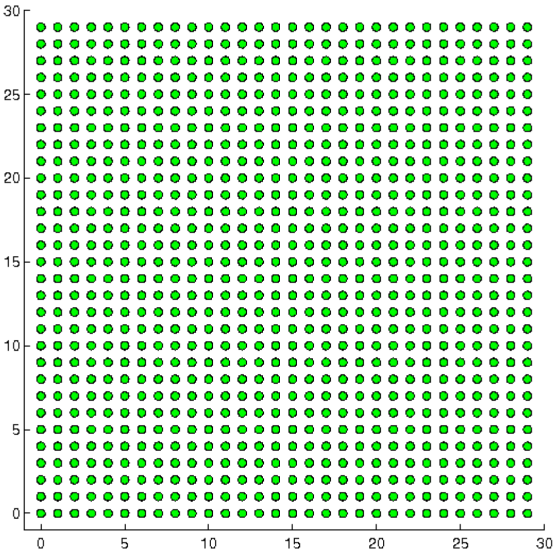


Figure 2.11. Eighth Round of Transmissions

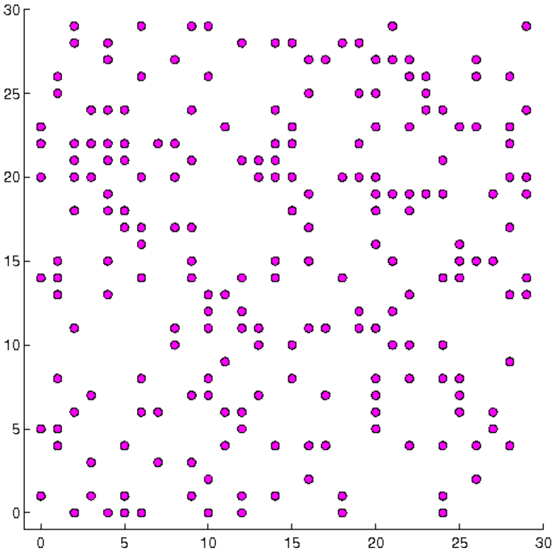


Figure 2.12. 223 of 900 Nodes that Transmit the Broadcast Message

Parameter	Value
Sensor range	20 *m*
Sensor P_D	0.9
Number of sensors	1-50
Test field	1000 *m* by 1000 *m*
Random walk step size	-1 to 1 *m* x-component
Random walk step size	0 to 2 *m* y-component
Monte Carlo runs per sensor number	50

Table 2.1. Simulation Parameters

between the number of sensors and the overall probability of detection and therefore, the phase transition is not as steep as it would be for larger numbers of sensors.

2.4 Algebraic Connectivity and Network Robustness

A graph G that consists of a set V of vertices (or nodes) and a collection of E edges is said to be connected if and only if there is a path between every pair of vertices in it. The node-connectivity number $\nu(G)$ of a graph G is defined as the minimum size of a separating set, or in other words, the minimum number of nodes that may be removed to separate the graph into more than one component. Similarly, the edge-connectivity number $\eta(G)$ is defined as the minimum number of edges that may be removed to separate the graph into more than one component. Algebraic connectivity is of great interest because of the following inequality developed by Fiedler:

$$\lambda_2(G) \leq \nu(G) \leq \eta(G) \tag{2.6}$$

which states that the Algebraic Connectivity of a graph G (defined as the second smallest eigenvalue $\lambda_2(G)$ of the Laplacian) is less than or equal to the node-connectivity which is less than or equal to the edge-connectivity [29]. Although increasing the Algebraic Connectivity increases the lower bound on node-connectivity, our simulation results show that for circular and mesh lattice graphs an increase in Algebraic Connectivity often corresponds to a decrease in node-connectivity and edge-connectivity.

The *small-world network* introduced by Watts and Strogatz [36] was based on a one-dimensional lattice on a ring with n nodes where each node is connected to its k nearest neighbors. They showed that random rewiring of nodes with a small probability p greatly reduces the characteristic path length resulting in a small-world network. Figure 2.14 (left) shows the effects of random rewiring for a network with 20 nodes and $k = 4$. Olfati-Saber

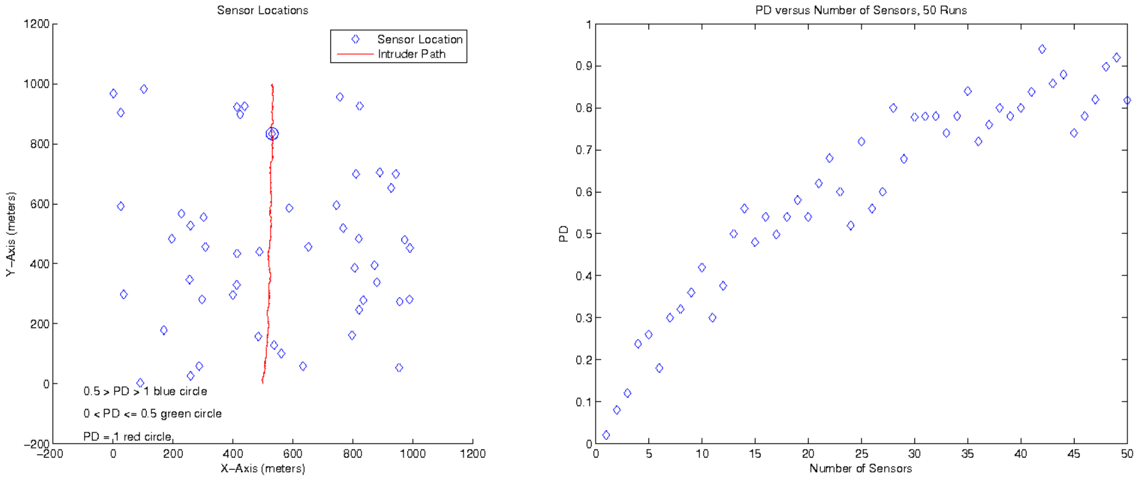


Figure 2.13. Typical Test Run Result, 50 sensors (left) and Probability of Detection versus Number of Sensors (right)

then showed that this random re-wiring also results in a large increase in Algebraic Connectivity for ring lattices [37].

Unfortunately large increases in Algebraic Connectivity for certain types of networks often correspond to a decrease in node-connectivity and edge-connectivity. As an example, the results for a circular random graph with 100 nodes are shown in Figure 2.15 (left). For this case, we start with a ring lattice with $n = 100$ vertices and $k = 4$ edges per vertex and then rewire each edge at random with a probability p. As p increases from 0 to 0.9 there is a large increase in algebraic connectivity and a decrease in the mean path length of the network. However, the node-connectivity and edge-connectivity of the network decrease as the probability p increases. Similar results can be shown for a regular mesh lattice like the one shown in Figure 2.14 (right) where there are 100 nodes and each node has a communication radius $R = 1$. The results for this mesh lattice are summarized in Figure 2.15 (right).

In a system where nodes are redundant or dispensable, improving Algebraic Connectivity can improve the overall robustness of the network by reducing the characteristic path length. However, in systems where each node is critical, concepts like node-connectivity and edge-connectivity are important parameters for assessing robustness. This highlights the fact that there are often tradeoffs when assessing robustness to different parameters. In addition there are computational tradeoffs. Computing Algebraic Connectivity is much quicker than computing node-connectivity or edge-connectivity for large networks.

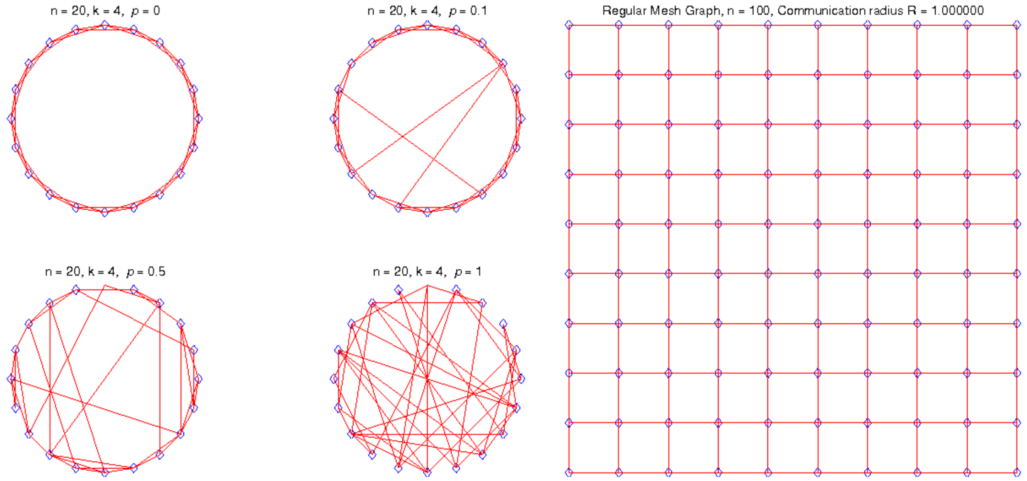


Figure 2.14. Random ring lattice graph $G = C(n,k)$ with $n = 20$, $k = 4$, for $p = 0, 0.1, 0.5, 1$, (left - 4 circles). Regular mesh lattice graph, N=100, Communication radius R=1 (right - square).

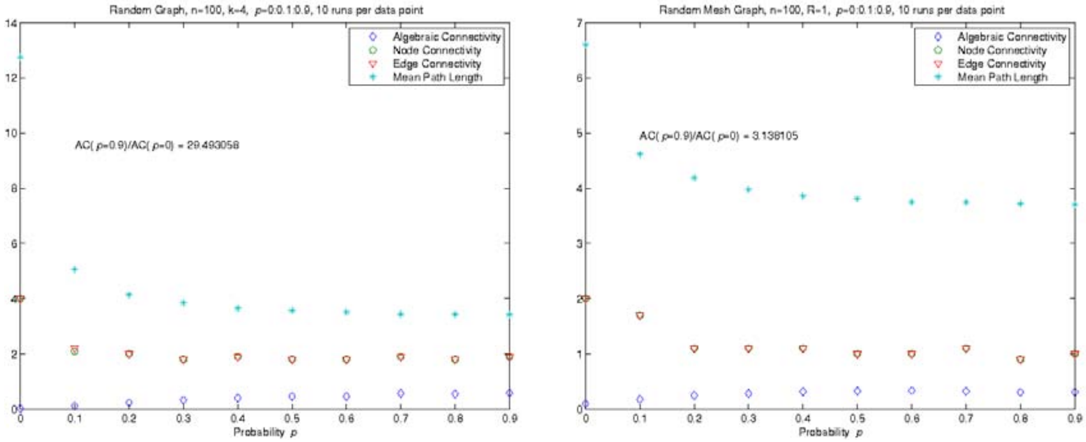


Figure 2.15. Results for a ring lattice random graph, N=100, k=4 (left). Results for a mesh lattice graph, N=100, R=1 (right).

2.5 Summary

Percolation and graph theory are complimentary techniques for analyzing the connectivity of large distributed systems. Using percolation theory, we have presented a simple but powerful approach for greatly reducing the time required to flood a network with a broadcast message. We also explored the relationship between probability of detection and the number of nodes in a distributed sensor network. Unfortunately, this relationship turns out to be roughly linear. A steep phase transition would have identified an ”optimal” number of sensor nodes. Additional nodes would add to the system cost but provide little improvement in the probability of detection. Our results on algebraic connectivity and graph robustness are very interesting. Algebraic connectivity is more closely correlated with characteristic path length and we have shown that node/edge connectivity can actually decrease for random circular and mesh networks even as algebraic connectivity increases.

Chapter 3

Exergy/Entropy Theory

This chapter develops a novel control system design methodology that uniquely combines: concepts from thermodynamic exergy and entropy; Hamiltonian systems; Lyapunov's direct method and Lyapunov optimal analysis; electric AC power concepts; and power flow analysis. Relationships are derived between exergy/entropy and Lyapunov optimal functions for Hamiltonian systems. The methodology is demonstrated with two fundamental numerical simulation examples: 1) a Duffing oscillator/Coulomb friction nonlinear model that employs PID regulator control and 2) a van der Pol nonlinear oscillator system. The control system performances and/or appropriately identified terms are partitioned and evaluated based on exergy generation and exergy dissipation terms. This novel nonlinear control methodology results in both necessary and sufficient conditions for stability of nonlinear systems.

Exergy is the elixir of life. Exergy is that portion of energy available to do work. Elixir is defined as a substance held capable of prolonging life indefinitely, which implies sustainability of life. In terms of mathematics and engineering, exergy sustainability is defined as the continuous compensation of irreversible entropy production in an open system with an impedance and capacity-matched persistent exergy source. Irreversible and nonequilibrium thermodynamic concepts are combined with self-organizing systems theories as well as nonlinear control and stability analyses to explain this definition. In particular, this paper provides a missing link in the analysis of self-organizing systems: a tie between irreversible thermodynamics and Hamiltonian systems. As a result of this work , the concept of "on the edge of chaos" is formulated as a set of necessary and sufficient conditions for stability and performance of sustainable systems. In addition, exergy is shown to be a fundamental driver and necessary input for sustainable systems, since exergy input in the form of power is a single point of failure for self-organizing, adaptable systems. Further developments of exergy sustainability for complex systems are in reference [42].

Collective systems are typically defined as a group of agents (physical and/or cyber) that work together to produce a collective behavior with a value greater than the sum of the individual parts. This amplification or synergy can be harnessed by solving an inverse problem via an information-flow/communications grid: given a desired macroscopic/collective behavior find the required microscopic/individual behavior of each agent and the required communications grid. The goal of this report is to describe the fundamental nature of the Hamiltonian function in the design of collective systems and the connections between and values of physical and information exergies intrinsic to collective systems. In particular, physical and information exergies are shown to be equivalent based on thermodynamics and Hamiltonian mechanics. Further developments of exergy/entropy concepts applied to collective systems for both physical and information exergies are given in reference [43].

3.1 Introduction

Today's engineering systems sustain desirable performance by using well-designed control systems based on fundamental principles and mathematics. Many engineering breakthroughs and improvements in sensing and computation have helped to advance the field. Control systems currently play critical roles in many areas, including automation, manufacturing, electronics, communications, transportation, computers, and networks, as well as many commercial and military systems [44]. Traditionally, almost all modern control design is based on forcing the nonlinear systems to perform and behave like linear systems, thus limiting its maximum potential. In this paper a novel nonlinear control design methodology is introduced that overcomes this limitation.

Several of the popular advanced nonlinear control system approaches are based in passivity and dissipative control theories. Initially, Moylan [45] discussed the implications of passivity for a broad class of nonlinear systems, a connection is established between the input-output property of passivity and a set of constraints on the state equations for the system. Later, Wyatt, et.al. [46, 47] clarified the meaning of passivity and losslessness as understood in nonlinear circuit theory, and their counterparts in classical physics. Most recently, Ortega, Jiang, and Hill [48] reviewed recent results on the stabilization of nonlinear systems using a passivity approach. Passivity properties play a vital role in designing asymptotically stabilizing controllers for nonlinear systems where the nonlinear versions of the Kalman-Yacubovitch-Popov lemma are used as key testing tools. The dissipative characteristics of dynamical systems has its origins in work by Willems [49] with further specifics given by Hill and Moylan [50]. In [50], a technique is introduced for generating Lyapunov functions for a broad class of nonlinear systems represented by state equations. The system, for which a Lyapunov function is required, is assumed to have a property called dissipativeness. In other words, the system absorbs more energy from the external world than it supplies. Different types of dissipativeness can be considered depending on how the "power input" is selected. Dissipativeness is shown to be characterized by the existence of a computable function which can be interpreted as the "stored energy" of the system. Under certain conditions, this energy function is a Lyapunov function which establishes stability, and in some cases asymptotic stability, of the isolated system. It was shown that for a certain class of nonlinear systems, that an "energy" approach was useful in analyzing stability. Kokotovic and Arcak [51] provide a recent discussion about the historical perspective of constructive nonlinear control theories. Structural properties of nonlinear systems and passivation-based designs exploit the connections between passivity and inverse optimality, and between Lyapunov functions and optimal value functions. Recursive design procedures, such as backstepping and forwarding, achieve certain optimal properties for important classes of nonlinear systems. Some of the more popular nonlinear control system designs [52, 53, 54] have their fundamental foundations built upon these concepts.

In other engineering disciplines, Alonso and Ydstie [55] connect thermodynamics and the passivity theory of nonlinear control. The storage function is derived from the convexity of the entropy and is closely related to thermodynamic availability. Dissipation is related to positive entropy production. In this form the supply function is a product of force and flow variation variables. Results are discussed in relationship to heat conduction and reaction diffusion equation problems. Anthony [56] suggests that non-equilibrium thermodynamics of irreversible processes may be included into the framework of a Lagrangian formalism. This formalism presents a unified method for reversible and irreversible processes. A straightforward procedure allows for the incorporation of both the first and second laws of thermodynamics into the Lagrangian. The theory is illustrated in three representative examples which include; material flow, heat conduction, diffusion and chemical reactions.

The main contribution of this chapter is to present a novel nonlinear control design methodology that is based on thermodynamic exergy and irreversible entropy production concepts. Relationships are developed between exergy, irreversible entropy production, Hamiltonian systems, Lyapunov optimal functions, electric AC power concepts, and power flow, for control system design. Both necessary and sufficient conditions for stability are determined for nonlinear systems. By combining the first and second laws of thermodynamics, an exergy analysis approach is developed to construct Lyapunov optimal functions for Hamiltonian systems. The first time derivative of the Lyapunov functions, based on exergy, irreversible entropy production rate, and power flow is partitioned into either exergy dissipative or exergy generative terms.

This chapter is divided into eight sections. Sections 3.2 and 3.3 provide the preliminary thermodynamics and Hamiltonian mechanics definitions. Section 3.4 develops the relationships and connections between thermodynamics and Hamiltonian mechanics. Section 3.5 defines the necessary and sufficient conditions for stability of nonlinear systems. Section 3.6 shows how, with simplifications to this novel control theory, conventional Lyapunov optimal and passivity control design methodologies are recovered. Section 3.7 presents regulator control design examples that include; 1) a PID control regulator for a nonlinear Duffing oscillator/Coulomb friction dynamic system and 2) a van der Pol nonlinear oscillator system. Numerical simulations resulted in the demonstration of both performance and stability criteria. Finally, Section 3.8 summarizes the results with concluding remarks.

3.2 Thermodynamic Concepts

In this section the first and second laws of thermodynamics are used to define exergy. One interpretation of the first law of thermodynamics states energy is conserved (see Fig. 3.1-

left). The second law of thermodynamics implies that the entropy of the universe always increases. The first law is a conservation equation while the second law is an inequality. Mathematically, a result of the first law can be written in terms of its time derivatives or energy rate for a system [57] as

$$\dot{\mathcal{E}} = \sum_i \dot{\mathcal{Q}}_i + \sum_j \dot{\mathcal{W}}_j + \sum_k \dot{m}_k \left(h_k + ke_k + pe_k + \ldots \right). \tag{3.1}$$

The term on the left represents the rate at which energy is changing within the system. The heat entering or leaving the system is given by $\dot{\mathcal{Q}}_i$ and the work entering or leaving the system is given by $\dot{\mathcal{W}}_j$. Next, material can enter or leave the system by $\dot{m}_k$ that includes enthalpy, h, kinetic and potential energies, ke, pe, etc. In addition, each term is "summed" over an arbitrary number of entry and exit locations i, j, k.

The second law or entropy rate equation for a system [57] is given as

$$\dot{\mathcal{S}} = \sum_i \frac{\dot{\mathcal{Q}}_i}{T_i} + \sum_k \dot{m}_k s_k + \dot{\mathcal{S}}_i = \dot{\mathcal{S}}_e + \dot{\mathcal{S}}_i. \tag{3.2}$$

Here the left hand term is the rate entropy changes within the system and the right hand terms represent, in order, the rate heat conducts entropy to and from the system and the rate material carries it in or out. These two terms can be combined into one term $\dot{\mathcal{S}}_e$, the entropy exchanged (either positive or negative) with the environment and $\dot{\mathcal{S}}_i$ is the irreversible entropy production rate within the system. Figure 3.1 (right) shows the entropy exchanges and production within the system [58].

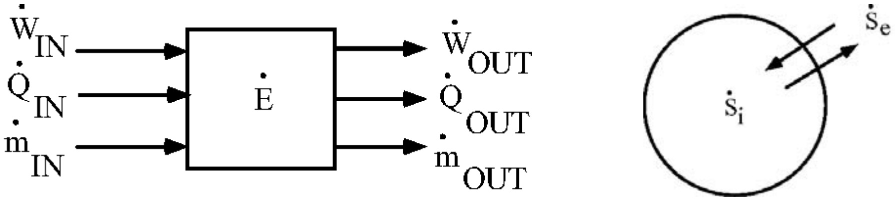


Figure 3.1. Energy flow control volume (left), second law entropy with flux exchange system (right)

The irreversible entropy production rate can be written as the sum of the thermodynamic forces and the thermodynamic flows [58, 59]

$$\dot{\mathcal{S}}_i = \sum_k \mathcal{F}_k \dot{\mathcal{X}}_k \geq 0 \tag{3.3}$$

where the entropy change is the sum of all the changes due to the irreversible flows $\dot{\mathcal{X}}_k$ with respect to each corresponding thermodynamic force $\mathcal{F}_k$.

Next, for systems with a constant environmental temperature, a thermodynamic quantity called the availability function which has the same form as the Helmholtz free energy function is defined as [58]

$$\Xi = \mathcal{E} - T_o\mathcal{S} \tag{3.4}$$

where T_o is the reference environmental temperature. The availability function is described as the maximum theoretically available energy that can do work which we call exergy. Exergy is also known as negative-entropy [57, 60]. By taking the time derivative of the availability function (3.4) and substituting in the expressions for (3.1) and (3.2) results in the exergy rate equation

$$\begin{aligned} \dot{\Xi} &= \textstyle\sum_i \left(1 - \frac{T_o}{T_i}\right) \dot{\mathcal{Q}}_i \\ &\quad + \textstyle\sum_j \left(\dot{\mathcal{W}}_j - p_o \frac{d\bar{V}}{dt}\right) + \sum_k \dot{m}_k \zeta_k^{flow} - T_o\dot{\mathcal{S}}_i. \end{aligned} \tag{3.5}$$

Where $\dot{\Xi}$ is the rate at which exergy stored within the system is changing. The terms on the right, in order, define the rate exergy is carried in/out by; i) heat, ii) work (less any work the system does on the environment at constant environmental pressure p_o if the system volume $\bar{V}$ changes), and iii) by the material (or quantity known as flow exergy). The final term, $T_o\dot{\mathcal{S}}_i$, is the rate exergy is destroyed within the system.

3.3 Hamiltonian Mechanics

The derivation of the Hamiltonian [61] begins with the Lagrangian for a system defined as

$$\mathcal{L} = \mathcal{T}(q, \dot{q}, t) - \mathcal{V}(q, t) \tag{3.6}$$

	t	=	time explicitly
	q	=	N-dimensional generalized coordinate vector
where	$\dot{q}$	=	N-dimensional generalized velocity vector
	$\mathcal{T}$	=	Kinetic energy, and
	$\mathcal{V}$	=	Potential energy.

The Hamiltonian is defined in terms of the Lagrangian as

$$\mathcal{H} \equiv \sum_{i=1}^{n} \frac{\partial \mathcal{L}}{\partial \dot{q}_i}\dot{q}_i - \mathcal{L}(q, \dot{q}, t) = \mathcal{H}(q, \dot{q}, t). \tag{3.7}$$

The Hamiltonian in terms of the canonical coordinates (q, p) is

$$\mathcal{H}(q,p,t) = \sum_{i=1}^{n} p_i\dot{q}_i - \mathcal{L}(q,\dot{q},t) \tag{3.8}$$

where the canonical momentum is defined as

$$p_i = \frac{\partial \mathcal{L}}{\partial \dot{q}_i}. \tag{3.9}$$

Then Hamilton's canonical equations of motion become

$$\begin{array}{rcl} \dot{q}_i & = & \frac{\partial \mathcal{H}}{\partial p_i} \\ \dot{p}_i & = & -\frac{\partial \mathcal{H}}{\partial q_i} + Q_i \end{array} \tag{3.10}$$

where Q_i is the generalized force vector. Next taking the time derivative of (3.8) gives

$$\dot{\mathcal{H}} = \sum_{i=1}^{n} \left(\dot{p}_i\dot{q}_i + p_i\ddot{q}_i - \frac{\partial \mathcal{L}}{\partial t} - \frac{\partial L}{\partial q_i}\dot{q}_i - \frac{\partial \mathcal{L}}{\partial \dot{q}_i}\ddot{q}_i \right). \tag{3.11}$$

Then substitute (3.10) into (3.11) and simplifying gives

$$\dot{\mathcal{H}} = \sum_{i=1}^{n} Q_i\dot{q}_i - \frac{\partial \mathcal{L}}{\partial t}. \tag{3.12}$$

Hamiltonians for most natural systems are not explicit functions of time (or $\partial \mathcal{L}/\partial t = 0$). Then for

$$\mathcal{L} = \mathcal{L}(q,\dot{q}) \tag{3.13}$$

the power (work/energy) equation becomes

$$\dot{\mathcal{H}}(q,p) = \sum_{i=1}^{n} Q_i\dot{q}_i. \tag{3.14}$$

3.4 Thermo-Mechanical Relationships

3.4.1 Conservative Mechanical Systems

A system is conservative if

$$\dot{\mathcal{H}} = 0 \qquad \text{and} \qquad \mathcal{H} = \text{constant}.$$

A force is conservative if

$$\oint F \cdot dx = \oint F \cdot v dt = \oint Q_j \dot{q}_j dt = 0$$

where F is the force, dx the displacement, and v the velocity. Basically, all of the forces can be modeled as potential force fields which are storage devices.

3.4.2 Reversible Thermodynamic Systems

A thermodynamic system is reversible if

$$\begin{array}{rcl} d\mathcal{S} & = & \frac{dQ}{T} \\ \oint d\mathcal{S} & = & \oint \frac{dQ}{T} = 0 \\ \oint d\mathcal{S} & = & \oint [d\mathcal{S}_i + d\mathcal{S}_e] = \oint \left[\dot{\mathcal{S}}_i + \dot{\mathcal{S}}_e\right] dt = 0 \end{array}$$

which implies that $\dot{\mathcal{S}}_e = \dot{Q}/T$ since by definition the second law gives $\dot{\mathcal{S}}_i = 0$.

3.4.3 Irreversible Thermodynamic Systems

For

$$\oint d\mathcal{S} = \oint \left[\dot{\mathcal{S}}_i + \dot{\mathcal{S}}_e\right] dt = 0$$

then $\dot{\mathcal{S}}_e \leq 0$ and $\dot{\mathcal{S}}_i \geq 0$.

3.4.4 Analogies and Connections

Now the connections between thermodynamics and Hamiltonian mechanics are investigated.

1. The irreversible entropy production rate can be expressed as

$$\dot{\mathcal{S}}_i = \sum_k \mathcal{F}_k \dot{\mathcal{X}}_k = \frac{1}{T_o} \sum_k \mathcal{Q}_k \dot{q}_k \geq 0. \tag{3.15}$$

2. The time derivative of the Hamiltonian is equivalent to the exergy rate

$$\begin{array}{rcl} \dot{\mathcal{H}} & = & \sum_k \mathcal{Q}_k \dot{q}_k \\ \dot{\Xi} & = & \dot{W} - T_o \dot{\mathcal{S}}_i = \sum_{j=1}^{N} \mathcal{Q}_j \dot{q}_j - \sum_{l=N+1}^{M+N} \mathcal{Q}_l \dot{q}_l \end{array} \tag{3.16}$$

   Where N is the number of generators, M the number of dissipators, and let $\dot{W} = \sum_j \dot{\mathcal{W}}_j$. The following assumptions apply when utilizing the exergy rate equation (3.5) for *Hamiltonian systems*:

   (a) No substantial heat flow:

   $$\dot{Q}_i \approx 0.$$

   (b) No substantial exergy flow or assume T_i is only slightly greater than T_o:

   $$1 - \frac{T_o}{T_i} \approx 0.$$

   (c) No $p_o \bar{V}$ work on the environment:

   $$p_o \frac{d\bar{V}}{dt} = 0.$$

   (d) No mass flow rate:

   $$\sum_k \dot{m}_k \zeta_k^{flow} = 0.$$

   (e) Then define:

   $$\begin{array}{lcll} \dot{W} & \geq & 0 & \text{power input/generated} \\ T_o \dot{\mathcal{S}}_i & \geq & 0 & \text{power dissipated.} \end{array}$$

3. A conservative system is equivalent to a reversible system when

$$\dot{\mathcal{H}} = 0 \quad \text{and} \quad \dot{\mathcal{S}}_e = 0$$

   then

$$\dot{\mathcal{S}}_i = 0 \quad \text{and} \quad \dot{W} = 0.$$

4. For a system that "appears to be conservative", but is not reversible is defined as:

$$\begin{aligned}\dot{\mathcal{H}}_{ave} &= P_{ave}(\text{over a cycle}) = 0 \\ &= \tfrac{1}{\tau}\oint[\dot{W} - T_o\dot{S}_i]dt \\ &= (\dot{W})_{ave} - (T_o\dot{S}_i)_{ave} \\ &= \tfrac{1}{\tau}\oint[\textstyle\sum_{j=1}^{N} Q_j\dot{q}_j - \sum_{l=N+1}^{M+N} Q_l\dot{q}_l]dt\end{aligned}$$

where τ is the period of the cycle. To be more specific about the average power calculations, the AC power factor [62] provides an excellent example. For the general case of alternating current supplied to a complex impedance the voltage and current differ in phase by an angle θ. For

$$\begin{aligned}\dot{W} &= P = Q\dot{q} = v\,i = \sqrt{2}\bar{v}\cos(\omega t + \theta)\cdot\sqrt{2}\bar{i}\cos\omega t \\ &= \bar{v}\bar{i}[\cos\theta + \cos(2\omega t + \theta)]\end{aligned}$$

where P is power, v is voltage ($\bar{v}$), i is current ($\bar{i}$), θ is the phase angle, and ω is the frequency. Integrating over a cycle gives

$$(\dot{W})_{ave} = \bar{v}\bar{i}\cos\theta$$

where for the second term

$$\oint \cos(2\omega t + \theta)dt = 0.$$

This is an important set of conditions that will be used in the next section to find the generalized stability boundary.

5. Finally, the power terms are sorted into three categories:

   (a) $(\dot{W})_{ave}$ - power generators; $(Q_j\dot{q}_j)_{ave} > 0$,

   (b) $(T_o\dot{S}_i)_{ave}$ - power dissipators; $(Q_l\dot{q}_l)_{ave} < 0$,

   (c) $(T_o\dot{S}_{rev})_{ave}$ - reversible/conservative exergy storage terms; $(Q_k\dot{q}_k)_{ave} = 0$.

These three categories are fundamental terms in the following design procedures.

3.5 Necessary and Sufficient Conditions for Stability

The Lyapunov function is defined as the total energy which for most mechanical systems is equivalent to an appropriate Hamiltonian function

$$V = \mathcal{H} \tag{3.17}$$

which is positive definite. The time derivative is

$$\begin{aligned} \dot{V} &= \dot{\mathcal{H}} = \textstyle\sum_k Q_k \dot{q}_k = \sum_{j+1}^{N} Q_j \dot{q}_j - \sum_{l=N+1}^{M+N} Q_l \dot{q}_l \\ &= \dot{W} - T_o \dot{\mathcal{S}}_i. \end{aligned} \tag{3.18}$$

3.5.1 Stability and Instability Theorems

To describe a nonlinear system's behavior two theorems [63] help to characterize the essential features of its motion. In addition, by bounding the Lyapunov function between these Theorems, both necessary and sufficient conditions are a result of the transition of the time derivative of the Lyapunov function from stable to unstable.

1. **Lyapunov Theorem for Stability** Assume that there exists a scalar function V of the state x, with continuous first order derivatives such that

   $$\begin{array}{ll} V(x) & \text{is positive definite} \\ \dot{V}(x) & \text{is negative definite} \\ V(x) & \rightarrow \infty \quad \text{as} \quad \|x\| \rightarrow \infty \end{array}$$

   Then the equilibrium at the origin is globally asymptotically stable.

2. **Chetaev Theorem for Instability** Considering the equations of disturbed motion, let V be zero on the boundary of a region R which has the origin as a boundary point, and let both V and $\dot{V}$ be positive-definite in R; then the undisturbed motion is unstable at the origin.

3.5.2 Stability Lemma for Nonlinear Systems

Based on the relationship between thermodynamic exergy and Hamiltonian systems a fundamental stability Lemma can be formulated.

Fundamental Stability Lemma for Hamiltonian Systems The stability of Hamiltonian systems is bounded between Theorems 1 and 2. Given the Lyapunov derivative as a decomposition and sum of exergy generation rate and exergy dissipation rate then:

$$\dot{V} = \dot{W} - T_o \dot{\mathcal{S}}_i = \sum_{j=1}^{N} Q_j \dot{q}_j - \sum_{l=N+1}^{M+N} Q_l \dot{q}_l \tag{3.19}$$

that is subject to the following general necessary and sufficient conditions:

$$\begin{array}{lll} T_o \dot{\mathcal{S}}_i & \geq 0 & \text{Positive semi-definite, always true} \\ \dot{W} & \geq 0 & \text{Positive semi-definite; exergy pumped in.} \end{array}$$

The following corollaries encompass both stability and instability for Hamiltonian systems which utilize AC power concepts [62]:

Cor 1: For $(T_o\dot{S}_i)_{ave} = 0$ and $(\dot{W})_{ave} = 0$ then $\dot{V} = 0$ the Hamiltonian system is neutrally stable, conservative and reversible.

Cor 2: For $(T_o\dot{S}_i)_{ave} = 0$ and $(\dot{W})_{ave} > 0$ then $\dot{V} > 0$ the Hamiltonian system is unstable.

Cor 3: For $(T_o\dot{S}_i)_{ave} > 0$ and $(\dot{W})_{ave} = 0$ then $\dot{V} < 0$ the Hamiltonian system is asymptotically stable and a passive system in the general sense (passivity controllers).

Cor 4: Given apriori $(T_o\dot{S}_i)_{ave} > 0$ and $(\dot{W})_{ave} > 0$ then the Hamiltonian system is further subdivided into:

4.1: For $\left(T_o\dot{S}_i\right)_{ave} > \left(\dot{W}\right)_{ave}$ with $\dot{V} < 0$ yields asymptotic stability.

4.2: For $\left(T_o\dot{S}_i\right)_{ave} = \left(\dot{W}\right)_{ave}$ with $\dot{V} = 0$ yields neutral stability.

4.3: For $\left(T_o\dot{S}_i\right)_{ave} < \left(\dot{W}\right)_{ave}$ with $\dot{V} > 0$ yields an unstable system.

The bottom line is that stability is defined in terms of power flow which determines whether the system is moving toward or away from its minimum energy and maximum entropy state.

3.6 Lyapunov Optimal and Passivity Control

Present day robotic and aerospace applications use feedback controller designs that are *Lyapunov Optimal* [64]. A control law is *Lyapunov Optimal* if it minimizes the first time derivative of the Lyapunov function over a space of admissible controls. In general, a set of feedback gains are optimized by minimizing the regulating and/or tracking error of the feedback controller while regulating to zero and/or tracking a desired reference input. The Lyapunov function is the total error energy which for most mechanical systems is equivalent to an appropriate Hamiltonian function

$$V = \mathcal{H}. \tag{3.20}$$

Then the concept of Lyapunov Optimal [64] follows directly from setting $\dot{W} = 0$ in (3.19) and maximizing $T_o\dot{S}_i$ for which the time derivative of the Lyapunov function (Hamiltonian) or the modified power (work/energy) equation is written as

$$\dot{V} = \dot{\mathcal{H}} = -T_o\dot{S}_i = -\sum_{j=1}^{N} Q_j\dot{q}_j = -\sum_{j=1}^{N} F_j\dot{R}_j \tag{3.21}$$

which is independent of system dynamics and is a *kinematic quantity* that applies to any system. Note that F_j denotes a set of forces acting on a mechanical system and $\dot{R}_j$ denotes the inertial linear velocity of the point where F_j is applied.

Passivity control [52] for robotic systems follows directly from setting $\dot{W} = 0$ in (3.19).

3.7 Regulator Control Design Examples

Two nonlinear dynamic systems are investigated to demonstrate exergy/entropy control design analogies for control design theory and to provide unique insights as well. These examples are based on 1) a PID regulator control for nonlinear Duffing oscillator/Coulomb friction dynamic system and 2) a van der Pol nonlinear system.

3.7.1 Duffing Oscillator/Coulomb Friction with PID Control System

This example is the design of a control law for a single degree of freedom nonlinear oscillator. The Duffing oscillator/Coulomb friction dynamic model (see Fig. 3.2) is defined as

$$M\ddot{x} + C\dot{x} + C_{NL}\,\mathrm{sign}(\dot{x}) + Kx + K_{NL}x^3 = u \tag{3.22}$$

where M, C, K, and u are the mass, damper, stiffness coefficients and external force input terms, respectively. The nonlinear stiffness and Coulomb friction coefficients are K_{NL} and C_{NL}, respectively.

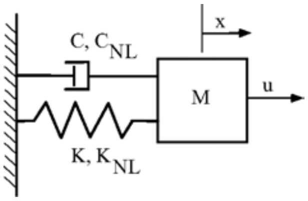


Figure 3.2. Duffing oscillator/Coulomb friction system

The PID controller is defined as

$$u = -K_P x - K_I \int_0^t x d\tau - K_D \dot{x} \tag{3.23}$$

where K_P, K_I, and K_D are the proportional, integral and derivative controller gains, respectively.

Initially, the nonlinear Duffing oscillator is investigated as a neutrally stable, reversible conservative system or

$$M\ddot{x} + Kx + K_{NL}x^3 = -K_P x$$

subject to the initial condition $x(0) = x_o = 1.0$. Now apply exergy/entropy control design and the derivative of the Lyapunov function/Hamiltonian becomes

$$\dot{V} = \dot{\mathcal{H}} = \dot{W} - T_o\dot{\mathcal{S}}_i = \sum_{j=1}^{N} Q_j\dot{q}_j - \sum_{l=N+1}^{M+N} Q_l\dot{q}_l$$

which yields

$$\begin{aligned} T_o\dot{\mathcal{S}}_i &= 0 \\ \dot{W} &= 0 \\ (T_o\dot{\mathcal{S}}_{rev})_{ave} &= (M\ddot{x}\cdot\dot{x} + (K+K_P)x\cdot\dot{x} + K_{NL}x^3\cdot\dot{x})_{ave} \\ &= 0. \end{aligned}$$

Numerical simulations are performed with the numerical values listed in Table 3.1. Note that for all cases that $M = 10.0$ kg, $K = 10.0$ N/m, and $K_{NL} = 100.0$ N/m^3. For this initial Case 1 the phase plane plot (left) and the potential and kinetic energy rate plots (right) are shown in Fig. 3.3. This run demonstrates Corollary 1 and a stable orbit for the nonlinear system with offsetting potential and kinetic energy rates responses.

Table 3.1. Duffing oscillator/Coulomb friction model and PID control system numerical values

.

Case	K_P	K_I	K_D	C	C_{NL}
No.	(kg/s^2)	(kg/s^3)	(kg/s)	(kg/s)	(N)
1	10.0	0.0	0.0	0.0	0.0
2	10.0	20.0	2.0	0.1	5.0
3	10.0	40.05	2.0	0.1	5.0
4	10.0	80.0	2.0	0.1	5.0

Next, consider the additional PID, linear, and Coulomb friction effects applied to the Duffing oscillator and partition into exergy generation and exergy dissipation terms. Now apply

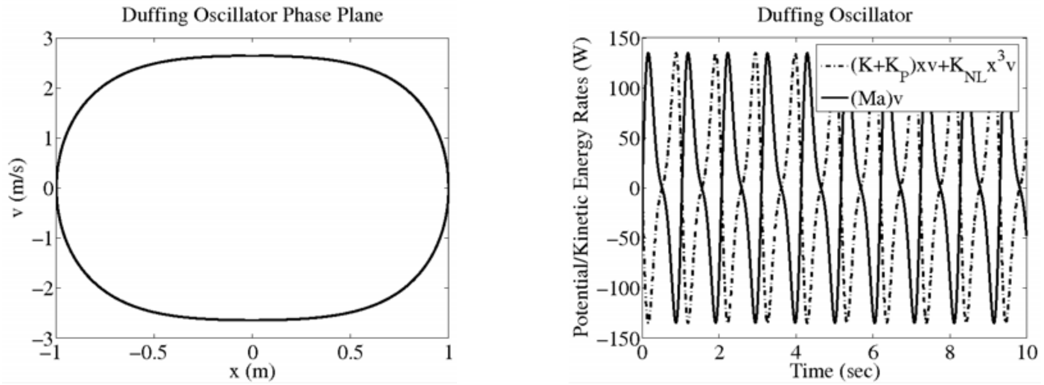


Figure 3.3. Cases 1: Duffing oscillator/Coulomb friction with PID control numerical results - phase plane and energy rates

the exergy/entropy control design and the derivative of the Lyapunov function/Hamiltonian becomes

$$\dot{V} = \dot{\mathcal{H}} = \dot{W} - T_o\dot{\mathcal{S}}_i = \sum_{j=1}^{N} Q_j\dot{q}_j - \sum_{l=N+1}^{M+N} Q_l\dot{q}_l$$

which yields

$$\begin{array}{ll} T_o\dot{\mathcal{S}}_i & = (C+K_D)\dot{x}\cdot\dot{x} + C_{NL}\,\mathrm{sign}(\dot{x})\cdot\dot{x} \\ \dot{W} & = -K_I\int_o^t x d\tau\cdot\dot{x} \\ (T_o\dot{\mathcal{S}}_{rev})_{ave} & = (M\ddot{x}\cdot\dot{x} + (K+K_P)x\cdot\dot{x} + K_{NL}x^3\cdot\dot{x})_{ave} \\ & = 0. \end{array}$$

To determine the nonlinear stability boundary from the exergy/entropy control design

$$\dot{V} = \dot{\mathcal{H}} = \dot{W} - T_o\dot{\mathcal{S}}_i$$

which gives

$$(\dot{W})_{ave} = (T_o\dot{\mathcal{S}}_i)_{ave}.$$

Substituting the actual terms yields the following:

$$\left[-K_I\int_o^t x d\tau\cdot\dot{x}\right]_{ave} = [(C+K_D)\dot{x}\cdot\dot{x} + C_{NL}\,\mathrm{sign}(\dot{x})\cdot\dot{x}]_{ave} \tag{3.24}$$

which is the *nonlinear stability boundary*. To best understand how the boundary is determined, concepts and analogies from electric AC power have been introduced earlier. Essentially, when the average power$_{in}$ is equivalent to the average power$_{dissipated}$ over a cycle, then the system is operating at the stability boundary. Later, in the exergy and exergy rate responses for the nonlinear system, one may observe that the area under the curves for the exergy rate generation and the exergy rate dissipation are equivalent and for the corresponding exergy responses the slopes will be equal and opposite. This helps to explain why PID control works well for nonlinear systems.

Numerical simulations are performed to demonstrate where the nonlinear stability boundary lies for the Duffing oscillator/Coulomb friction dynamic model subject to PID control. Three separate cases are conducted with the numerical values listed in Table 3.1. The nonlinear system is subject to an initial condition of $x_0 = 1.0$. For Case 2 the integral of position, position, velocity, and acceleration responses along with the exergy and exergy rate responses are plotted in Fig. 3.4. For this case, the dissipative term is greater than the generative term. This is observed from the decaying system responses. In Case 3 the system responses along with the exergy and exergy rate responses are shown in Fig. 3.5.

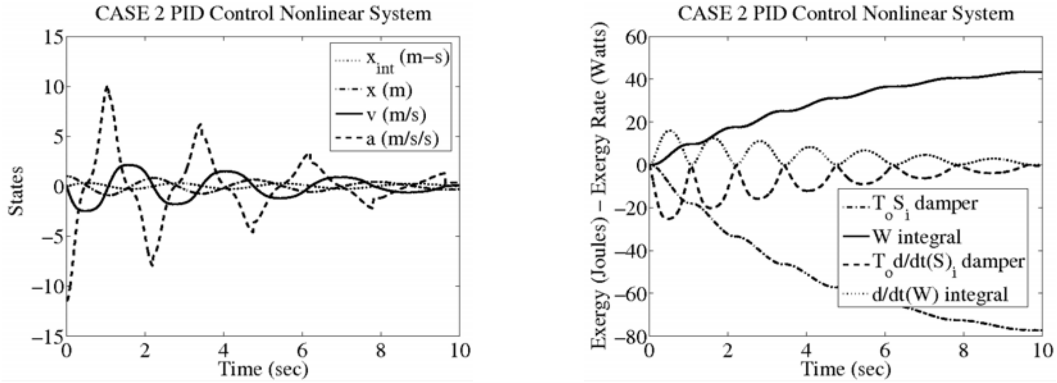


Figure 3.4. Cases 2: Duffing oscillator/Coulomb friction with PID control numerical results

In this case, the average exergy slopes and integrated power areas for the dissipative and generative terms are equivalent which demonstrates (3.24). This results in system responses that do not decay, displaying constant nonlinear oscillatory behavior. In final Case 4, the system responses along with the exergy and exergy rate responses are shown in Fig. 3.6. In this case, the dissipative term is less than the generative term which results in a system response with increasing nonlinear oscillatory behavior. In conclusion, Fig. 3.7 shows the responses for the total exergy with respect to each case along with the phase plane plot for the nonlinear system. For Case 3 the nonlinear stability boundary (or neutral stability) is characteristic of an average zero output for the total exergy response or validation of (3.24). For the phase plane plot, Case 2 demonstrates an asymptotically stable decaying response, Case 3 a neutrally stable orbital response, and Case 4 an asymptotically unstable increasing orbit response.

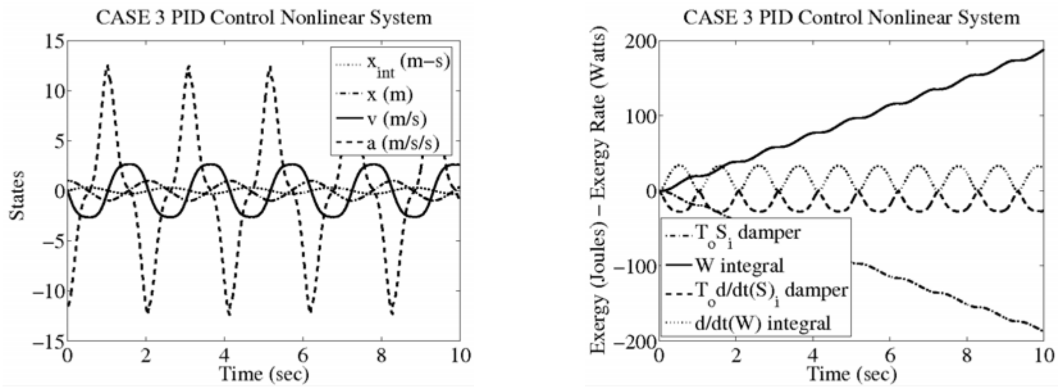


Figure 3.5. Cases 3: Duffing oscillator/Coulomb friction with PID control numerical results

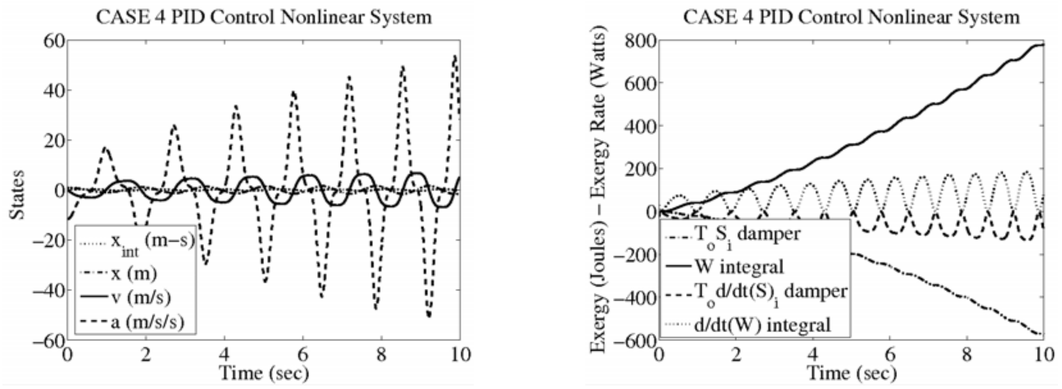


Figure 3.6. Cases 4: Duffing oscillator/Coulomb friction with PID control numerical results

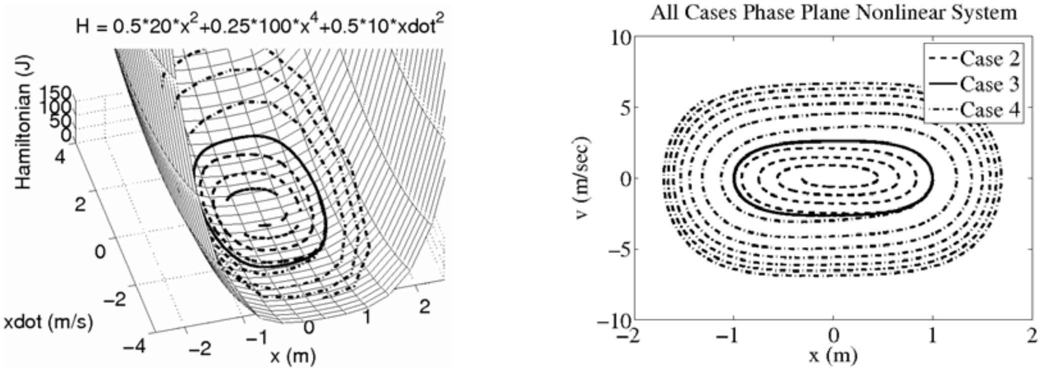


Figure 3.7. Cases 2:4 - Duffing oscillator/Coulomb friction numerical results 3D Hamiltonian (left) and phase plane plot (right)

The last three cases for the PID control regulator Duffing oscillator/Coulomb friction dynamic system demonstrates the three subcases for Corollary 4: Given apriori $(T_o\dot{S}_i)_{ave} > 0$ and $(\dot{W})_{ave} > 0$ then the nonlinear system showed the following:

i. Case 2 yielded $(T_o\dot{S}_i)_{ave} > (\dot{W})_{ave}$; asymptotic stability; damped stable nonlinear response and demonstration of Corollary 4.1.

ii. Case 3 yielded $(T_o\dot{S}_i)_{ave} = (\dot{W})_{ave}$; neutral stability; and demonstration of Corollary 4.2. This case is the nonlinear stability boundary where dissipation and generation terms cancel each other out on the average.

iii. Case 4 yielded $(T_o\dot{S}_i)_{ave} < (\dot{W})_{ave}$; increasingly unstable towards another orbit; and demonstration of Corollary 4.3.

Notice that the trajectories are constrained to move along the Hamiltonian surface. The specific trajectory path can be influenced as shown above, by either control integral action $K_I \int x d\tau$ (generator) or control derivative action $K_D\dot{x}$, respectively. The control proportional action, $K_P x$ changes the storage/potential function or the Hamiltonian surface. For example, to determine the effect that the proportional controller gain K_P has on the system, Hamiltonian phase plane plots are generated. By investigating a system with negative stiffness and by adding enough K_P to result in an overall positive net stiffness, changes the shape of the Hamiltonian surface from a saddle point surface (see Fig. 3.8 -top pair) to a positive bowl surface (see Fig. 3.8- bottom pair). A two-dimensional cross-section of the Hamiltonian versus the position shows the characteristics of the overall storage or potential functions. The operating point at $(H,\dot{x},x) = (0,0,0)$ changes from being unstable to stable, for small values of $|x| > 0$, when enough additional K_P is added, a net positive stiffness for the system results.

3.7.2 Van der Pol Nonlinear System

The classic van der Pol's equation [65] is analyzed using the techniques of this section. Originally, the "van der Pol equation" is credited to van der Pol, and is a model of an electronic circuit for early radio vacuum tubes of a triode electronic oscillator [65]. The tube acts like a normal resistor when the current is high, but acts as a negative resistor if the current is low. The main feature is that electrical circuits that contain these elements pump up small oscillations due to a negative resistance when currents are small, but drag down large amplitude oscillations due to positive resistance when the currents are large. This behavior is known as a *relaxation oscillation*, as each period of the oscillation consists of a slow buildup of energy ('stress phase') followed by a phase in which energy is discharged ('relaxation phase'). This particular system has played a large role in nonlinear dynamics and has been used to study limit cycles and self-sustained oscillatory phenomena in nonlinear systems.

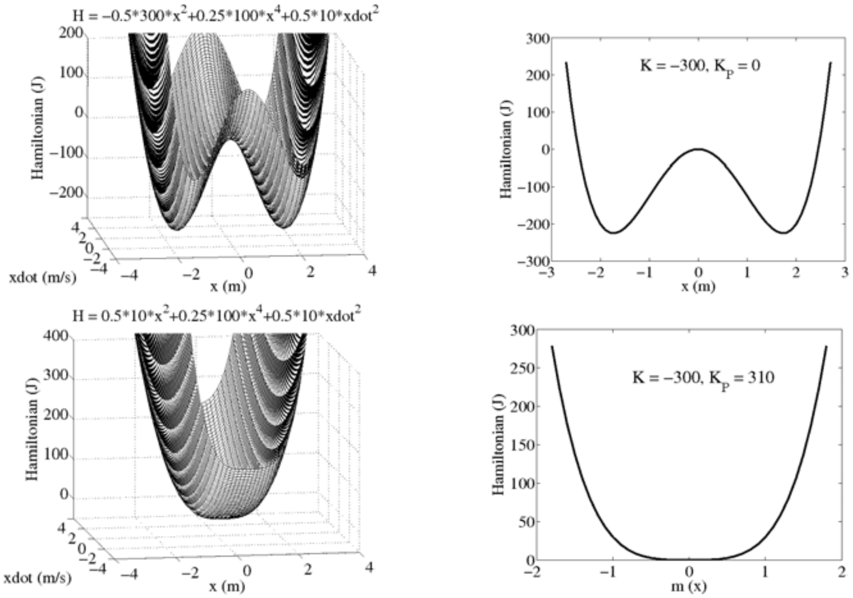


Figure 3.8. Three dimensional (top-left) Hamiltonian phase plane plot negative stiffness produces a saddle surface. The two-dimensional cross-section plot (top-right) is at $\dot{x} = 0$. Three dimensional (bottom-left) Hamiltonian phase plane plot where the net positive stiffness produces a positive bowl surface. The two-dimensional cross-section plot (bottom-right) is at $\dot{x} = 0$.

Consider the van der Pol equation with mass (m) and stiffness (k) values other than unity and a nonlinear damping term (μ) to be defined as:

$$m\ddot{x} - \mu(1 - x^2)\dot{x} + kx = 0.$$

The appropriate Hamiltonian/Lyapunov function is defined as:

$$\mathcal{H} = V = \frac{1}{2}m\dot{x}^2 + \frac{1}{2}kx^2 > 0.$$

Then the corresponding time derivative of the Lyapunov function/Hamiltonian becomes

$$\begin{aligned} \dot{V} &= [m\ddot{x} + kx]\,\dot{x} \\ &= \left[\mu\dot{x}(1 - x^2)\right]\dot{x} \\ &= \mu\dot{x}^2 - \mu x^2\dot{x}^2. \end{aligned}$$

Next identifying the generator and dissipator terms yields

$$\begin{aligned} T_o\dot{S}_i &= \mu x^2\dot{x}^2 \\ \dot{W} &= \mu\dot{x}^2 \\ (T_o\dot{S}_{rev})_{ave} &= (m\ddot{x}\cdot\dot{x}+kx\cdot\dot{x})_{ave}=0. \end{aligned}$$

The *nonlinear stability boundary* can be determined as

$$\begin{aligned} \left[\dot{W}\right]_{ave} &= \left[T_o\dot{S}_i\right]_{ave} \\ \left[\mu\dot{x}^2\right]_{ave} &= \left[\mu x^2\dot{x}^2\right]_{ave} \end{aligned}$$

By investigating several initial conditions both inside, on, and outside the limit cycle then three separate regions can be observed. Figure 3.9 shows these conditions with the corresponding numerical values given in Table 3.2.

Table 3.2. van der Pol model numerical values

Case	x_o	$\dot{x}_o$	μ	m	k
	(m)	(m/s)	(kg/s)	(kg)	(kg/s^2)
generate	0.1	−0.1	1.5	1.0	1.0
neutral	1.0	−1.0	1.5	1.0	1.0
dissipate	2.0	−2.0	1.5	1.0	1.0

The responses are plotted on the Hamiltonian 3D surface (top) with the projection onto the phase plane shown on the 2D plot (middle). For the case outside the limit cycle, the dissipator term dominates and for the case inside the limit cycle the generator term dominates. For both cases inside and outside the limit cycle, the system migrates back to the stability boundary. For the case already on the limit cycle then the system is already at neutral stability. The neutral exergy-rate and exergy plots are shown in Fig. 3.9 (bottom). The cycle is defined at approximately $\tau = 3.5$ seconds. For the neutral pair the terms cancel each other out at the end of the cycle or $[\dot{W}]_{ave} = [T_o\dot{S}_i]_{ave}$. For the generator case then $[\dot{W}]_{ave} > [T_o\dot{S}_i]_{ave}$ and for the dissipator case then $[\dot{W}]_{ave} < [T_o\dot{S}_i]_{ave}$, respectively. Eventually, given enough cycles both the generator and dissipator cases will converge to the neutral case.

3.8 Summary and Conclusions

A novel control system design methodology was developed that uniquely combined: concepts from thermodynamic exergy and entropy; Hamiltonian systems; Lyapunov's direct

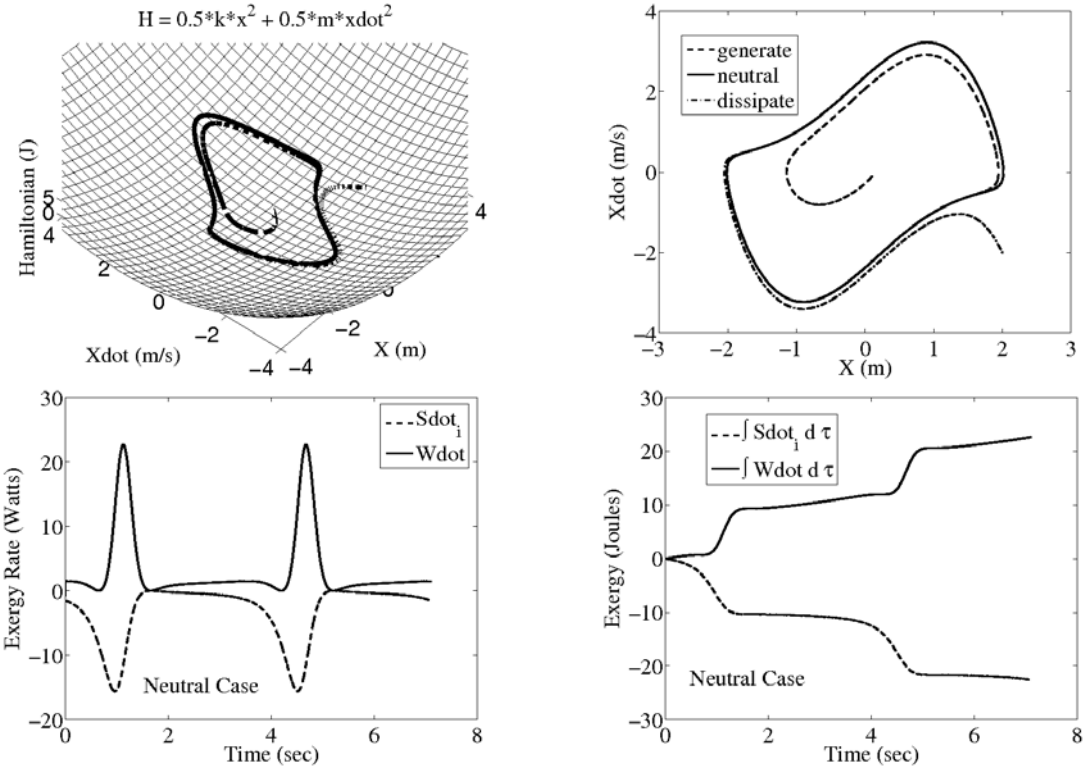
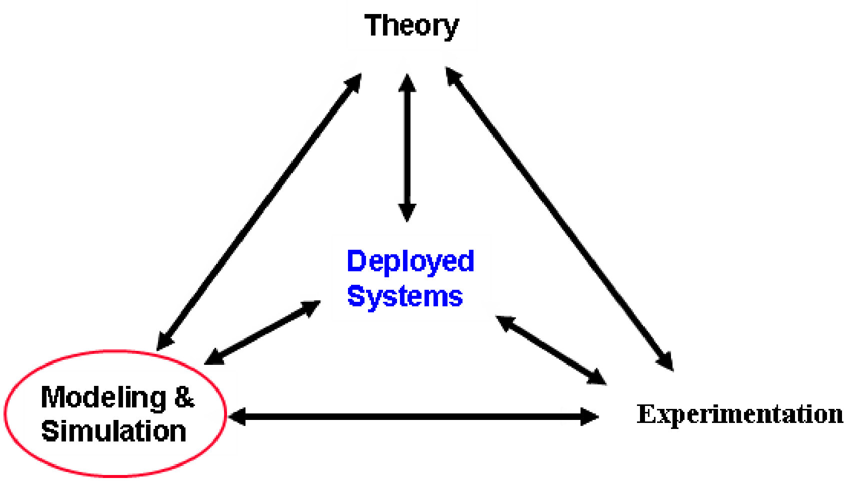


Figure 3.9. van der Pol responses - 3D Hamiltonian, phase plane plot (top), and exergy-rate and exergy plots (bottom)

method and Lyapunov optimal analysis; electric AC power concepts; and power flow analysis. Relationships were derived between exergy/entropy and Lyapunov optimal functions for Hamiltonian systems. The methodology is demonstrated with two fundamental numerical simulation examples: 1) a Duffing oscillator/Coulomb friction nonlinear model that employs PID regulator control and 2) a van der Pol nonlinear oscillator system. The control system performance results and/or appropriately identified terms were partitioned and evaluated based on exergy generation and exergy dissipation terms. These numerical results showed the stability boundaries for each nonlinear system. This novel nonlinear control methodology resulted in both necessary and sufficient conditions for stability of nonlinear systems. In the near future, this novel control system design methodology will be extended to tracking and adaptive control of multi-input/multi-output nonlinear systems. This methodoloy is applicable to a large class of nonlinear systems.

Part II

Modeling & Simulation


Theory
Deployed Systems
Modeling & Simulation
Experimentation

Chapter 4

Simulation Environment

4.1 Umbra

Umbra is a powerful asset to employ for the Q-value approximation. The principle of an agent based simulation program once again melds seamlessly with our continuous notion of "particles" as a representation of the targets' belief-state space. Each particle chosen from the Particle Filters final distribution is acquired by Umbra and translated into the simulations as an actual physical entity (or agent).

Sandia National Laboratories built Umbra to be a simulation environment in which disparate entities could be modeled, and their interactions observed, such that complex system behavior can be analyzed [66]. It implements agents as a collection of "modules," each of which corresponds to different components of that agent. Umbra also implements "worlds" that monitor and influence the behavior of similar modules spread across several agents. As an example, in this project, each of the sensors, the terrain, the vehicles and pedestrians being tracked, the particles and the POMDP are each composed of multiple modules. These modules are given time and spatial constraints as per each agent they represent. The sensors, vehicles and pedestrian are referenced by a world module to process which target is being detected by which sensor at each time step. Additionally, the POMDP is a world to the particles, in that it monitors their position, advances and replays some of them during the Monte-Carlo analysis and filters them appropriately.

In this hierarchical organizational structure, where agents are composed of an assembly of modules, and where world modules can supply phenomenological modeling, Umbra provides a time-stepped execution engine to sequentially calculate the behaviors of the agents through time. Because of the modular composition of the agents, variable degrees of model fidelity can be integrated for different portions of the simulation. For example, because the particles are massless, they do not need a physics-based representation, but because the vehicles and pedestrians do have mass, their agents can have low, medium, or high fidelity physics models associated with them. Umbra allows for simple module insertion and replacement, changing fidelity can be done even during execution of the simulation. Deciding the degree of the fidelity of the model depends on the aspects of the simulation that are of the highest importance. This experiment required good models for sensing, communication and particle dispersal, so it is in these areas where the highest fidelity modules were written.

A large part of the simulation environment was leveraged from former work done at Sandia National Laboratories. Entities like the terrain, vehicles, pedestrians and communications systems had already been developed, and were used without modification. However, new capabilities were written to create the paths for the pedestrians, the particles, the particle motion models and the POMDP. Other capabilities had to be extended for this projects use. Specifically, a new model of radar sensors had to be implemented based on a larger sensor suite that had already been developed for Umbra.

The simulation was organized such that sensor reports were provided to the POMDP through a communication simulation including radio modems and relays. This radio simulation builds an ad-hoc network based on message addressing and route requests. The POMDP does its execution of building targets along with their particle clouds based on these sensor reports. The POMDP, in turn, sends messages back to the sensors using this same radio network. The communication and sensor models both take into account ground effects and line-of-sight in their calculations. One of the needs of the POMDP algorithm is accuracy in the motion model for the particles. The modular architecture and high fidelity of the terrain used in this simulation make it an excellent development and test environment for motion model algorithms. By replacing the module type in the startup script, the different motion model will be instanced and executed for the POMDP in subsequent runs. Different algorithms, such as slope-based costing (Dijkstra), weighted random-walk, etc., can be easily implemented and evaluated within the simulation developed for this study.

In addition to integrating this simulation with the many other capabilities built in Umbra, it is easy to integrate physical hardware systems such as the Wireless Sensor Network into the simulation as well. Since this simulation is time stepped (as the vast majority of Umbra simulations are), by synchronizing the simulation clock with real-time, external data coming from the WSN Annunciator can be fed in directly to the POMDP module. (The WSN Annunciator has a visualization aspect that was developed in Umbra that takes advantage of the real-time execution of Umbra.) By using Umbra as a real-time execution environment for the POMDP, we are able to use any of a suite of real and virtual sensors, in any combination to exercise the POMDP. Additionally, we are able to bring the benefits of the POMDP to a fielded sensor network.

4.2 Matlab/Simulink

The Mathworks Matlab/Simulink environment was used to develop and simulate the various exergy/entropy control architecture evaluations. An S-function was developed for both the controller and dynamic plant, respectively. The S-function environment consists of a predefined general nonlinear state-space model which may include either or both discrete and continuous modules. The control law accepts reference inputs and generates control outputs that are sent into the dynamic plant. Both the control and dynamic modules are compiled as a MEX function and easily integrated into the remaining control architecture components. The particular S-functions used were programmed in the C-language.

Simulations were performed using the simulink suite of integrators (where the 4^{th}-order Runge-Kutta was the primary integrator employed). Gains and pertinent system parameters were created with the constant block, in the simulink model for ease of changing values and re-running each case study. The numerical simulation results were saved in the Matlab environment and post plotted with a custom m-file. A representative exergy/entropy simulation scenario is shown in Figure 4.1.

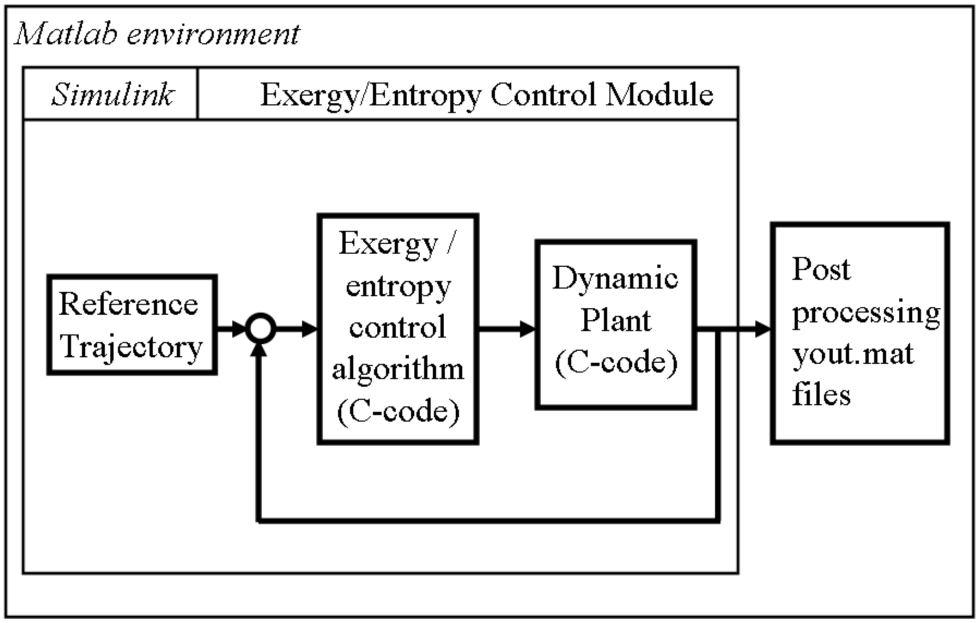
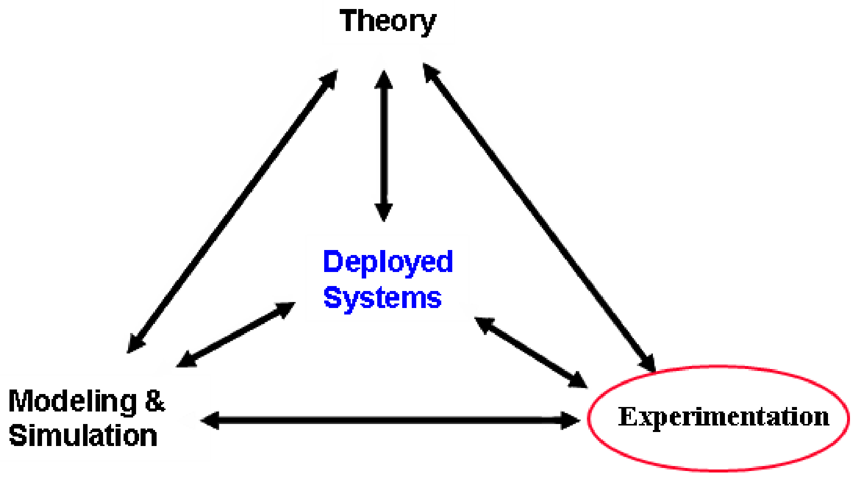

Matlab environment
Simulink
Exergy/Entropy Control Module
Reference Trajectory
Exergy / entropy control algorithm (C-code)
Dynamic Plant (C-code)
Post processing yout.mat files

Figure 4.1. Matlab/Simulink modeling environment - representative exergy/entropy control scenario

Part III

Physical Experiments


Theory
Deployed Systems
Modeling & Simulation
Experimentation

Chapter 5

Wireless Sensor Network

To validate the previously discussed algorithms, as well as other surveillance algorithms, a one-hundred node wireless sensor network was constructed and fielded at Sandia National Laboratories (Kirtland Air Force Base). A single sensor node is shown in Fig. 5.1 (all nodes are identical). The node consists of two primary components: a pyroelectric infrared (PIR) sensor (colored tan) and a networked radio (white box on ground). The primary design criteria for the system were low cost, low power, and RF range over a mile.

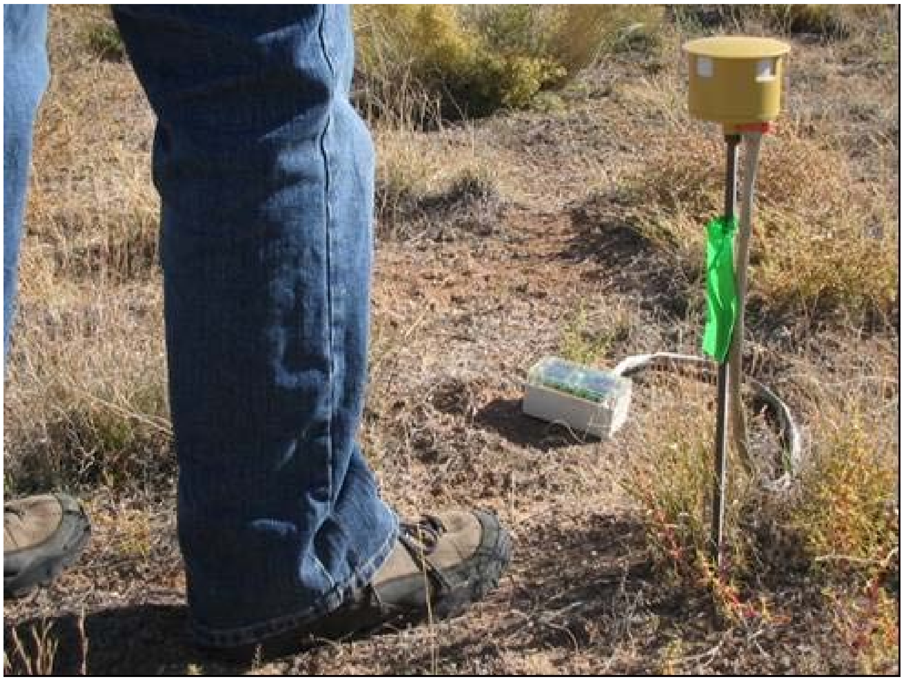


Figure 5.1. Physical Sensor

The PIR node (see Figure 5.2) employs four C172 dual pyroelectric sensors oriented 90 degrees apart. Cone optics mounted with the sensor provide 90 degrees horizontal and 30 degrees vertical field of view. Infrared transmissive plastic film, 0.11mm thick, covers four windows in the custom built plastic housing. Together the sensor is sensitive in the 7-14 micrometer wavelengths and has a range of approximately 15 meters. The analog sensor signal is amplified and digitized by a custom build circuit board utilizing an Atmel Atmega

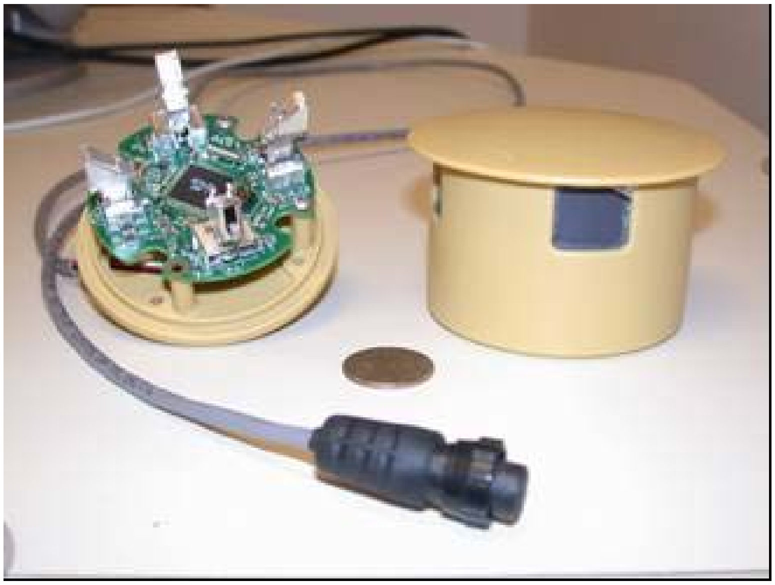


Figure 5.2. Wireless sensor network hardware

128 processor. Communication to the radio is conducted via an RS 485 cable.

The radio (see Figure 5.3) consists of the following components: a modified Xecom XE900S-500 radio transceiver, antenna, solar cells, batteries, geophone, and a custom board for power regulation/charging, signal conditioning and RS 485 interface. They are mounted together in a commercial weather-resistant box with a clear lid.

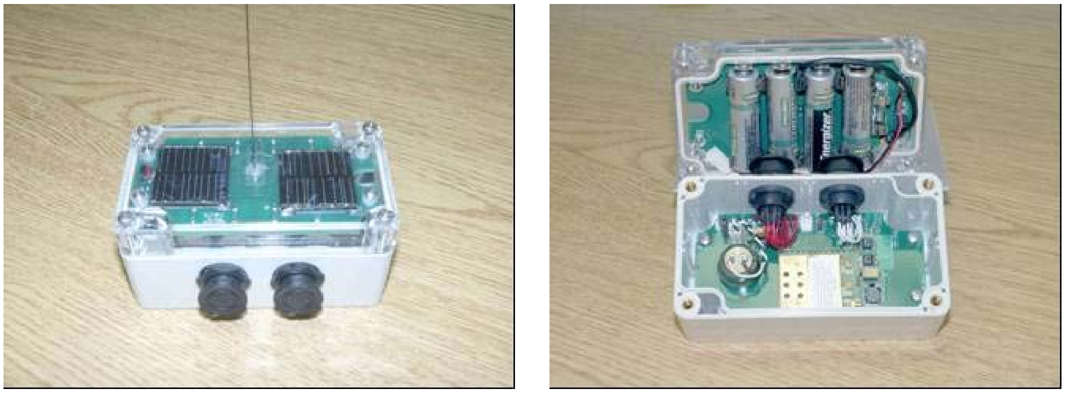


Figure 5.3. Wireless sensor network radio hardware

The Xecom transceiver has adjustable output power from 1mW to 0.5W and operates in 902-928MHz range (frequency hopping). Sandia redesigned the radios firmware under a non-disclosure agreement with Xecom to meet the requirements of WSN and FCC part 15 regulations. Data rate was boosted to 152kbps at a range of several miles. Power consumption was reduced to 130 uA average (two year stand alone operation without solar). An algorithm to process geophone signals for classification (pedestrian, vehicle, tampering) was also added.

The WSN is capable of interfacing with over 1000 sensor nodes arranged in 32 clusters with 32 sensor nodes per cluster. This topology is illustrated in Fig. 5.4.

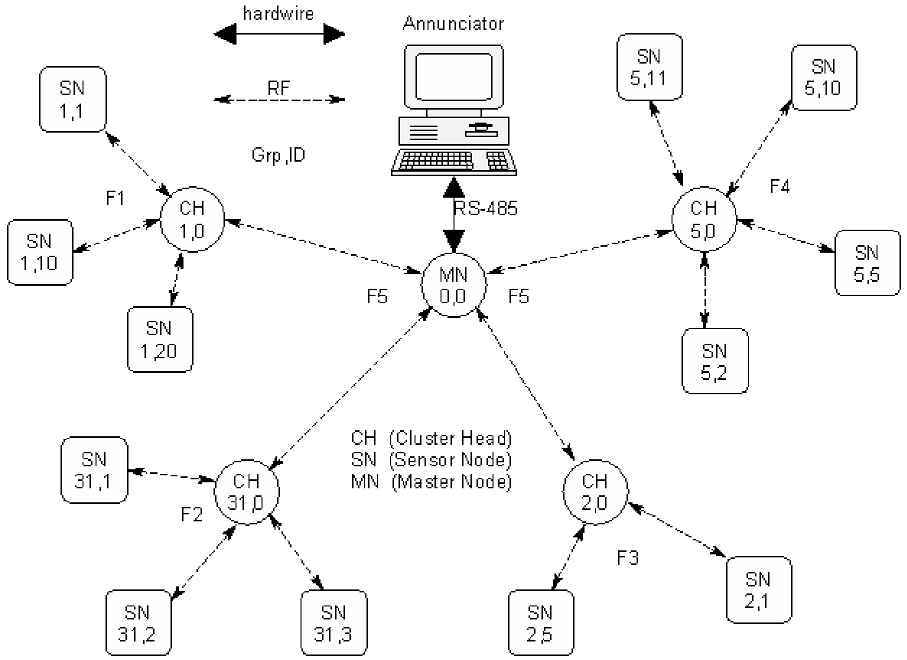
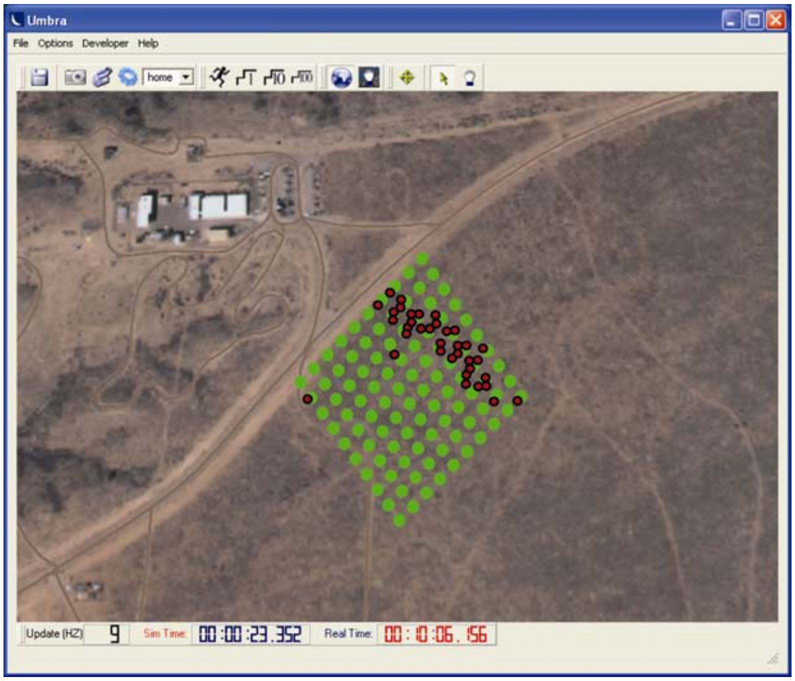


Figure 5.4. Sensor node communication topology

Alarm and status information is funneled through the cluster heads back to a master node that connects with an annunciator display computer using an RS-485 interface (Fig. 5.5). Configuration data from the annunciator can also be sent to sensor nodes for the purpose of adjusting sensor performance.

Sandia developed a unique form of network control for WSN called Hybrid Division Multiple Access (HDMA) that combines several forms of common access control strategies to minimize contention within the network. Carrier Sense Multiple Access (CSMA) is used at the sensor node level. When a sensor node needs to communicate a spontaneous alarm message to a cluster head, it waits if it senses another node transmitting, thus avoiding a collision. This technique works well because most sensor nodes within a cluster are within radio range of one another.

Routine status messages within a cluster are scheduled over time to preclude simultaneous communications. This technique is called Time Division Multiple Access (TDMA) or



Update (HZ) 9 Sim Time: 00:00:23.352 Real Time: 00:10:06.156

Figure 5.5. WSN Graphic interface through Umbra

slotting.

Frequency Division Multiple Access (FDMA) is used to separate adjacent clusters from interfering with each other. Since every cluster operates on a different frequency at any given time, there is no possibility of interference.

Cluster heads communicate with the master node on yet a different frequency than the cluster frequencies (FDMA). Contention between cluster heads is avoided using TDMA instead of CSMA because clusters may not be able to sense one another due to their remoteness. The entire network is synchronized and is constantly changing frequencies to avoid interference with other radio systems that might be operating within the general area.

Supervisory state of health messages are sent from each node every 8 seconds and a node's voltage and temperature is communicated to the master node and annunciator every 30 minutes. Alarm messages have a maximum latency of 250ms and are time stamped to an accuracy of 8ms. Synchronized time within the entire network is maintained to about 100μs. Configuration commands from the annunciator require up to 8 seconds to be received by a sensor node because these units are dormant most of the time to conserve power. This communication scheme is illustrated in Fig. 5.6.

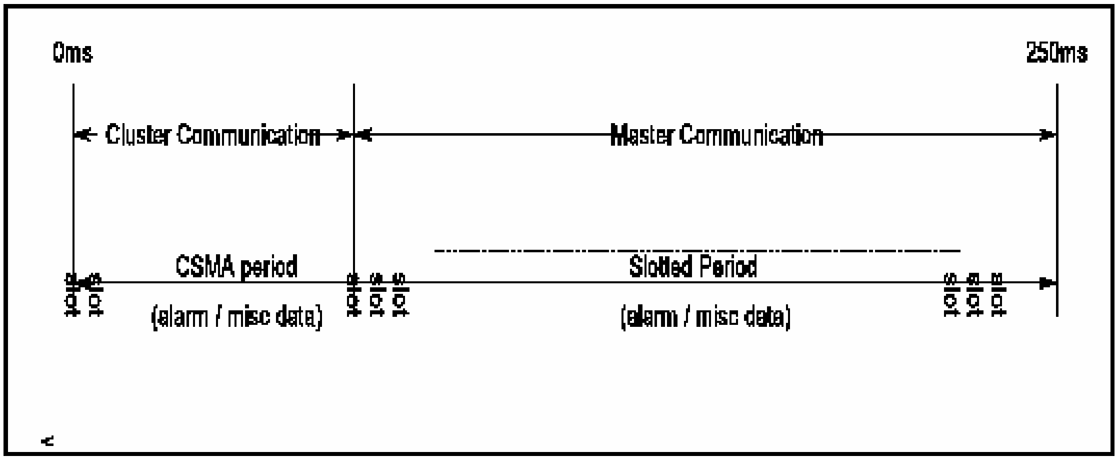


Figure 5.6. TDMA communication diagram

Chapter 6

Unmanned Ground Vehicles

An unmanned ground vehicle, or UGV, has many uses in a dynamic security system. It can act as a constantly moving sensor platform, thus denying an intruder the advantage of knowing were all the sensors are. It can move to investigate sensor trips, and if an intruder is detected, it can intercept and challenge him. The intruder's response will determine his "intent." If properly equipped (with sticky foam, flash bangs, lethal weapons, etc.) the vehicle can effectively delay an intruder. Finally, a UGV can be used to test other sensors in the field. From the perspective of complexity, a UGV's mobility provides a type of dynamic system interaction that static sensors cannot provide.

Because of its capability to carry large payloads (weapons, sensors, etc.), the Hagar vehicle was chosen as our UGV platform (See figure 6.1). Developed under a previous LDRD, the vehicle used the DOS operating system and a primitive communication strategy with its base station. The vehicle was not equipped with an inertial measurement system and could thus be neutralized by jamming GPS. Finally the vehicle used a crude *sonar system*

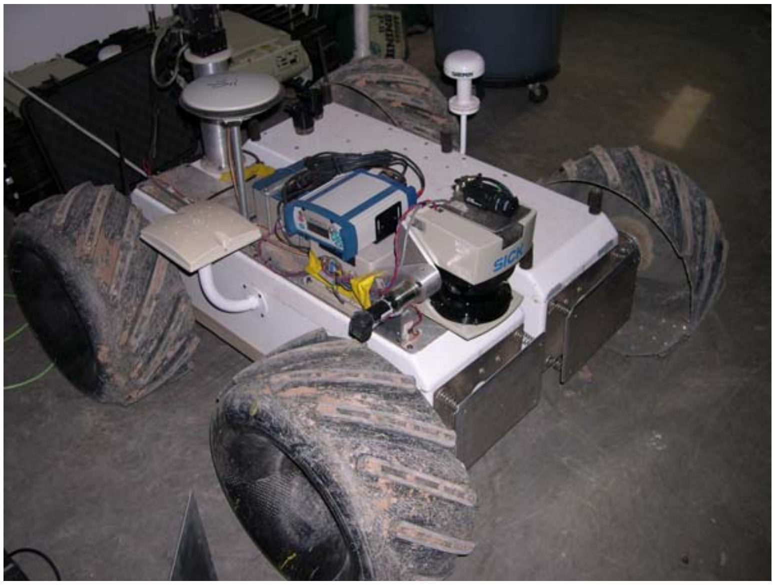


Figure 6.1. UGV Hagar

for collision avoidance. Our first action was to upgrade the vehicle with state of the art capabilities. The remainder of this section describes the vehicle's capabilities.

Hagar's main computer is an Ampro Littleboard 700 single board computer with an Intel 400MHz ULV Celeron processor, 256 Mbyte memory, and 1 Gbyte compact flash disk. It interfaces to a PC104 stack. The interface to the four wheels is provided by a Galil Motion Control DMC-1200 4-axis board.

Communication between the base station and Hagar is either by a Cisco Aironet BR500 wireless bridge, which provides high bandwidth at close range, or by a Freewave DGR-115H data radio, which can provide communication up to 20 miles. A separate California Microwave transmitter is used for the video feed.

The robot pose is determined by a Garmin GPS17N gps receiver, a new Novatel IMU, and a Crossbow Technology CXTILT02E roll, pitch, and yaw sensor. Proximity to obstacles is determined by a new Sick AG model LMS-291 laser rangefinder , which was mounted on an aluminum nodding table to enable 3 dimensional measurements (See figure 6.2). The nodder had its own Galil DMC-1200 motion control board. Video was provided by a Sony CCD color video camera Directed Perception pan-tilt unit.

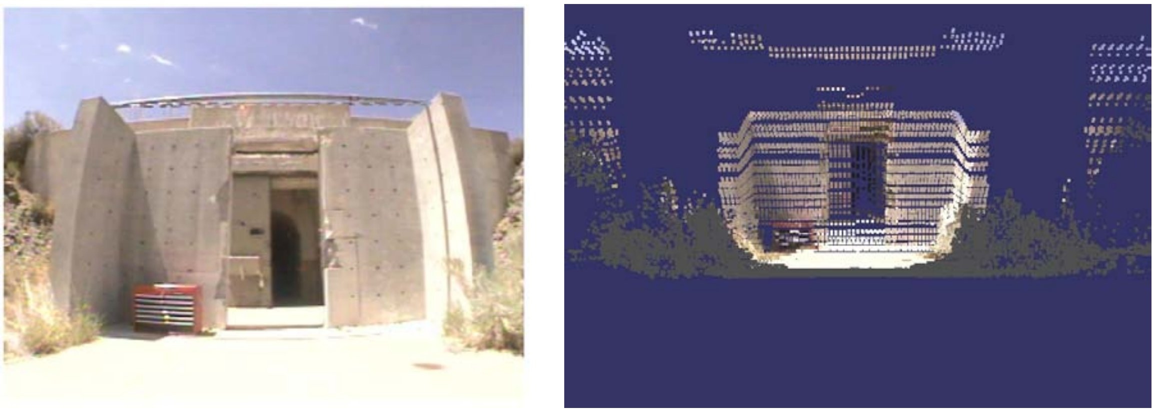


Figure 6.2. UGV target and mapping

The computer on Hagar runs a version of Debian Linux. This seemed optimal for a small computer. The kernel was 2.4.27 customized for low latency, and with modules added to handle the video and motion control boards, and up to 12 serial ports. The Swarm vehicle code was used to manage sensor input and wheel control.

We used the Swarm basestation code on a Dell Latitude C600 laptop computer running Windows 2000. This allowed us to teleoperate the robot, or to send it to designated GPS waypoints. All telemetry data was recorded on the basestation for future analysis.

Experiments were carried out at the RVR to determine the accuracy of the Novatel IMU, and to assess the capability for autonomous waypoint navigation. Given the complex course at the RVR, the robot was quite successful in waypoint navigation. The result of the IMU mapping of the RVR motocross track is shown in Figure 6.3. It was usually able to navigate about 1/3 of the course at a time, using up to 12 waypoints. After that, it appeared that drift in either the GPS or the compass would send the robot off the track, almost always in the counterclockwise direction.

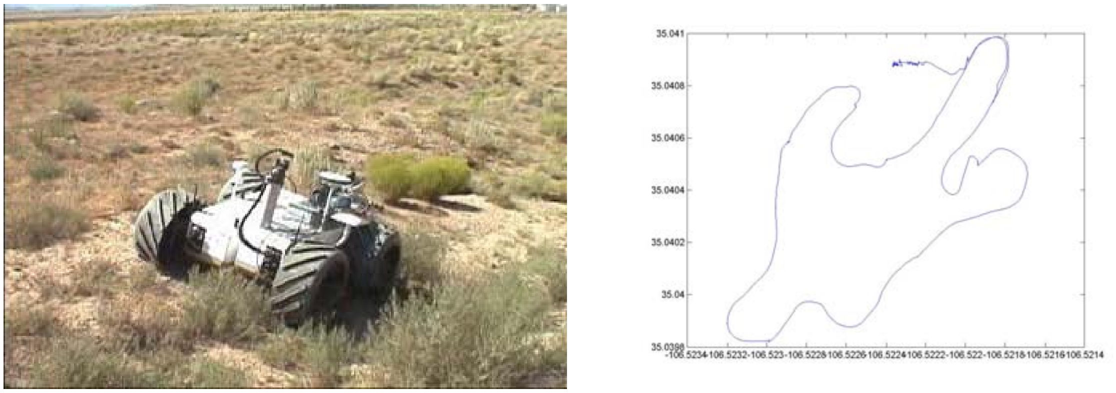


Figure 6.3. Mapping of RVR facility with IMU

We did not explore using the Hagar vehicle for SLAM (Simultaneous Localization and Mapping). However, Linux drivers were developed for the DMC-1200 motion control boards under another LDRD (Enhanced Perception), which would enable us to run the Carmen code from Carnegie Mellon University. This would provide SLAM capability on Hagar, should that be needed.

Chapter 7

Unmanned Aerial Vehicles

An unmanned aerial vehicle, or UAV, has many uses in a dynamic security system. Its high speed and relatively simple collision avoidance algorithms allows it to respond to alarms in extremely remote areas, whether night or day. Its high vantage point gives it an unprecedented view of the area, allowing large swaths of sensor coverage or detailed images of a small area.

Until recently, UAVs have been very expensive. This has prevented Sandia from delving into this area in the past. Driven by interest from DoD and Homeland Security, UAV components are now affordable, especially components for smaller UAVs. Our goal for this project was to evaluate the performance of off the shelf UAV components and determine whether an inexpensive security UAV was practical.

Figure 7.1 displays our UAV platform. Based on a "Roadrunner" airframe, designed for remote control hobbyists, this platform provided suitable room, lift and duration to test the components of interest. Costing around a $1000, it is also very inexpensive in the case of a

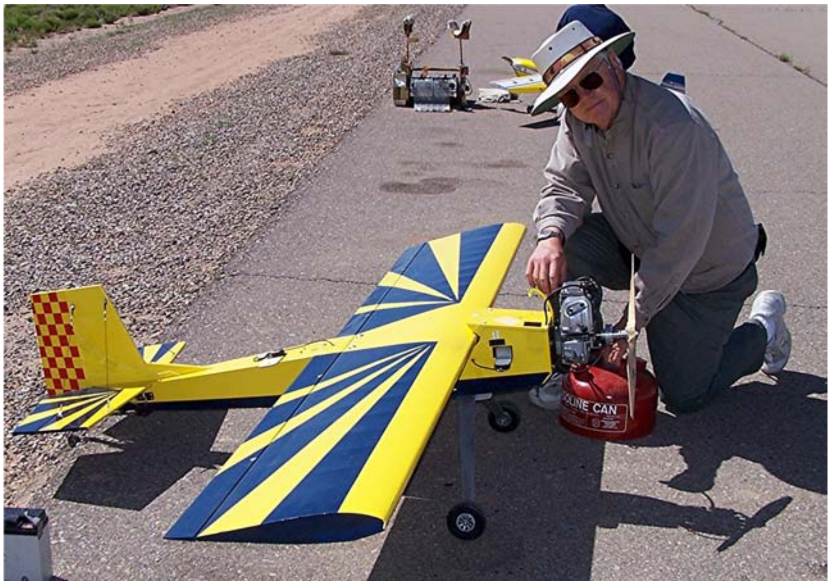


Figure 7.1. UAV Mark 1 aircraft

crash.

The UAV was equipped with ground surveillance video, shown in Figure 7.2. A Sony CCD color camera provided video while the data was relayed to the ground through a California Microwave transmitter. Autopilot functions were provided by a Micro-pilot MP2008 unit and associated ground control software. A display of the ground control laptop is given in Figure 7.3. The autopilot communicated to the base station via a Freewave radio. Telemetry was downloaded and waypoints could be changed while the vehicle was in the air.

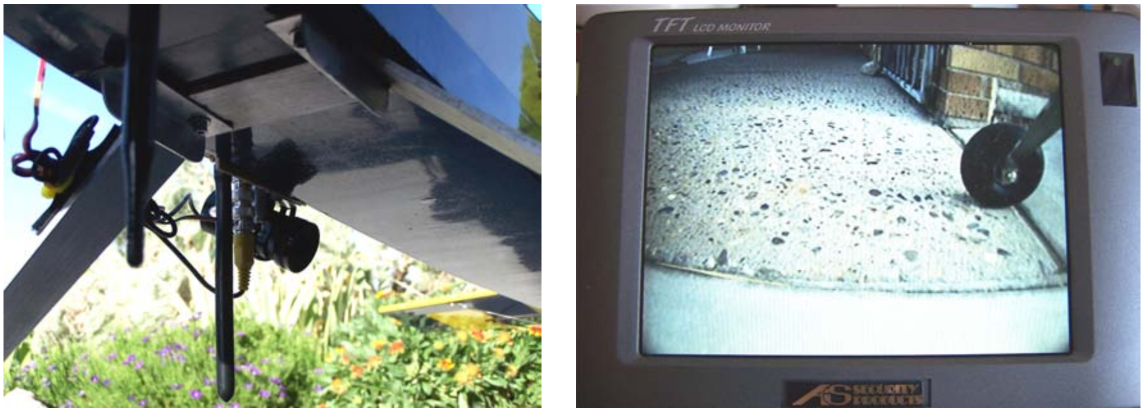


Figure 7.2. UAV equipped with ground surveillance video

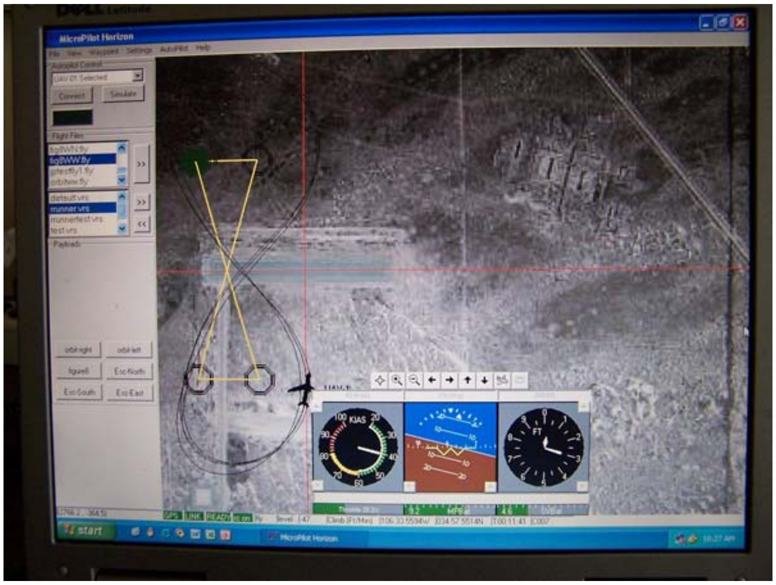


Figure 7.3. Display of UAV ground control station

Flying on base is not a trivial matter. Bob Bickerstaff worked out an arrangement between the Air Force and the FAA to allow UAV operations in the RVR area.

A great deal of UAV experience was gained in this portion of the project. The work contributed to two follow on projects: one to create a UAV roadmap for Sandia management, and the other project was classified.

Chapter 8

Conclusions and Recommendations

The goal of this LDRD was to “Develop the core competencies needed to design and deploy complex physical (security) systems that are robust, limited in behavior and not brittle.” Our discussions with staff in the physical security area revealed that they were indeed fielding more and more complex systems (wireless sensor networks, remotely operated weapon systems, barricades, etc.) and that managing these systems-of-systems was becoming problematic. We believe that when components of a security system begin to number in the hundreds or thousands manual control of security assets through an annunciator, as it is done presently, will no longer be effective. Some type of man-in-the-loop autonomous control will be necessary to manage the complexity.

Four tools for developing and controlling complex physical security systems have been presented in this paper. How they collectively contribute to a security system is illustrated in figure 8.1. This figure shows the major components of a security system. At the lowest level, sensor components (wireless sensor networks, UAVs, UGVs, patrol personnel, etc.) must efficiently communicate with one another so that information on intruders can be forwarded to decision makers. Percolation and Graph Theory provide the tools to design this communication network. At the next level up, sensor information needs to be combined so as to produce an accurate track of the intruders’ paths. POMDP achieves this task of multi-target tracking while simultaneously minimizing power consumption in power limited assets. Finally, tracking information for multiple targets is passed to a fire control system which optimally assigns defense assets to individual targets taking into account their position, required orientation change, collision free paths, etc. Entropy principles are utilized to determine the number of weapons and their performance so as to defeat the design basis threat.

Finally, complexity experts note that certain insects (ants, termites, bees, etc.) are capable of amazing group behaviors: building the hive, collecting food, attacking an intruder, etc. What is more amazing is that they achieve it not through centralized control but through the dynamics of individuals interacting with one another while using a small set of behaviors. This is termed “emergent behavior” and it is a form of complexity. This capability could be of great benefit to many system-of-systems because of the robustness it provides and because of the simplicity of the algorithm on each system. Unfortunately, it has proven to be difficult to start from a desired group behavior and work backwards to the behaviors of the individual system or agent.

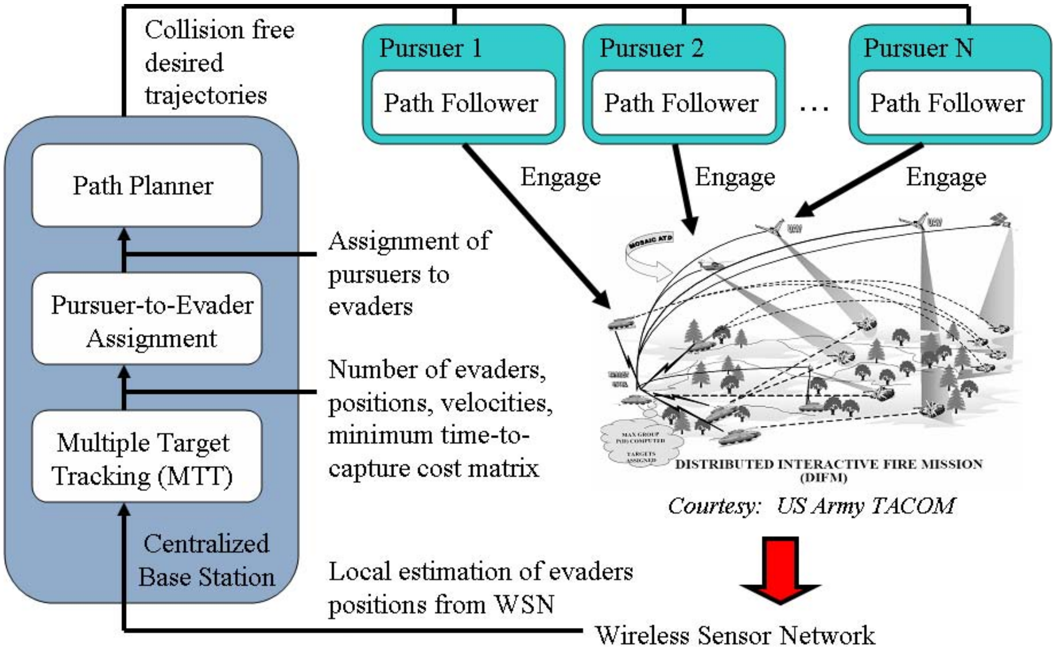

Collision free desired trajectories
Pursuer 1
Path Follower
Pursuer 2
Path Follower
...
Pursuer N
Path Follower
Engage
Engage
Engage
Path Planner
Assignment of pursuers to evaders
Pursuer-to-Evader Assignment
Number of evaders, positions, velocities, minimum time-to-capture cost matrix
Multiple Target Tracking (MTT)
Centralized Base Station
MOSAIC ATD
DISTRIBUTED INTERACTIVE FIRE MISSION (DIFM)
Courtesy: US Army TACOM
Local estimation of evaders positions from WSN
Wireless Sensor Network

Figure 8.1. Heterogeneous remote security systems scenario

The techniques presented in this paper also produce some amazing group behaviors: optimizing target tracking error and power utilization in a large sensor network, determining under what conditions a group of defenses will be overwhelmed by an attacking force and optimizing information flow in a communications network. However they do so utilizing centralized strategies. We submit that for physical security applications centralized control is not a limitation. In these applications, a centrally located highly defended asset is the prize. Once it has been seized, the game is over. Therefore, control of the security system should be located with or near the asset. For other applications, coordinating the Future Combat System for example, robustness to controller destruction is paramount. However, centralized control can be made robust by distributing redundant controllers throughout the force. When one is destroyed, another takes over, just as is done with personnel in the military hierarchy.

References

[1] Y. Oshman, “Optimal sensor selection strategy for discrete-time state estimators,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 30, no. 2, pp. 307–314, April 1994.

[2] A. Logothetis and A. Isaksson, “On sensor scheduling via information theoretic criteria,” in *Proceedings of the American Control Conference*, San Diego, CA, pp. 2402–2406, 1999.

[3] C. Kreucher, K. Kastella, and A. O. Hero III, “Multi-target sensor management using alpha-divergence measures,” Lecture Notes in Computer Science, *Proceedings of the 2nd International Conference on Information Processing in Sensor Networks (IPSN)*, no. 2634, pp. 209–222, April 2003.

[4] D. Castañon, “Approximate dynamic programming for sensor management,” in *Proceedings of the IEEE Conference on Decision and Control*, San Diego, CA, pp. 1202–1207, 1997.

[5] V. Krishnamurthy, “Algorithms for optimal scheduling of hidden Markov model sensors,” *IEEE Trans. Signal Processing*, vol. 50, no. 6, pp. 1382–1397, 2002.

[6] J. Evans and V. Krishnamurthy, “Optimal sensor scheduling for hidden Markov model state estimation,” *International Journal of Control*, vol. 74, no. 18, pp. 1737–1742, 2001.

[7] E. K. P. Chong, C. Kreucher, and A. O. Hero III, “POMDP approximation methods based on heuristics and simulation, ” Chapter 8 in *Foundations and Applications of Sensor Management*, Springer, to appear 2007.

[8] O. Hernández-Lerma, *Adaptive Markov Control Processes*, Springer-Verlag, New York, 2001.

[9] A. Doucet, N. de Freitas, and G. Gordon, *Sequential Monte Carlo Methods in Practice,* Springer-Verlag, New York, 2001.

[10] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking,” *IEEE Trans. Signal Processing,* vol. 50, no. 2, pp. 174–188, Feb. 2002.

[11] Y. He and E. K. P. Chong, “Sensor scheduling for target tracking: A Monte Carlo sampling approach,” *Digital Signal Processing*, to appear.

[12] Y. He and E. K. P. Chong, “Sensor scheduling for target tracking in sensor networks,” in *Proceedings of the 43rd IEEE Conference on Decision and Control (CDC’04)*, Paradise Island, Bahamas, pp. 743–748, Dec. 2004.

[13] Y. He and E. K. P. Chong, "Sensor scheduling for target tracking: A Monte Carlo sampling approach," in *Proceedings of the 2005 Workshop on Defense Applications of Signal Processing (DASP'05)*, Midway, Utah, March 2005.

[14] Y. Li, L. W. Krakow, E. K. P. Chong, and K. N. Groom, "Dynamic sensor management for multisensor multitarget tracking," *Proceedings of the 40th Annual Conference on Information Sciences and Systems,* Princeton, New Jersey, March 22–24, 2006, pp. 1397–1402.

[15] Y. Bar-Shalom and X. Li, *Estimation with Application to Tracking*, Artech house, 2001.

[16] C. Kreucher, K. Kastella, and A. O. Hero III, "Particle filtering for multitarget detection and tracking," in *Proceedings of The Twenty Sixth Annual IEEE Aerospace Conference*, March 2005.

[17] P. Z. Peebles, Jr., *Radar Principles*, John Wiley & Sons, Inc., 1998.

[18] C. Kwok, D. Fox, and M. Meila, "Real-time particle filters," in *Proceedings of the IEEE*, vol. 92, no. 3, pp. 469–484, 2004.

[19] S. Thrun, "Particle filters in robotics," in *Proceedings of the 17th Annual Conference on Uncertainty in AI*, 2002.

[20] R. Karlsson and F. Gustafsson, "Monte Carlo data association for multiple target tracking," in *Proceedings of the IEE Workshop on Target Tracking: Algorithms and Applications*, vol. 1, pp. 13/1–13/5, Oct. 2001.

[21] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems,* Artech House, Boston, 1999.

[22] L. Y. Pao and S. D. O'Neil, "Multisensor fusion algorithms for tracking," in *Proceedings of the American Control Conference*, San Francisco, CA, pp. 859–863, June 1993.

[23] L. Y. Pao and C. W. Frei, "A comparison of parallel and sequential implementations of a multisensor multitarget tracking algorithm," in *Proceedings of the American Control Conference*, Seattle, WA, vol. 3, pp. 1683–1687, June 1995.

[24] D. P. Bertsekas and D. A. Castañon, "Rollout algorithms for stochastic scheduling problems," *Journal of Heuristics*, vol. 5, pp. 89–108, 1999.

[25] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, third edition, Athena Scientific, 2005.

[26] D. Lerro and Y. Bar-Shalom, "Tracking with debiased consistent converted measurements versus EKF," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 29, no. 3, July 1993.

[27] L. W. Krakow, Y. Li, E. K. P. Chong, K. N. Groom, J. Harrington, and B. Rigdon, "Control of perimeter surveillance wireless sensor networks via partially observable Markov decision process," *Proceedings of the 2006 IEEE International Carnahan Conference on Security Technology (ICCST),* Lexington, Kentucky, October 17–20, 2006.

[28] S. R. Broadbent and J. M. Hammersley, *Percolation Processes I. Crystals and Mazes*, Proceedings of the Cambridge Philosophical Society, Vol. 53, pp. 629-641, 1957.

[29] M. Fiedler, *Algebraic Connectivity of Graphs*, Czechoslovak Mathematical Journal, Vol. 23, No. 98, pp. 298-305, 1973.

[30] Y. Sasson, D. Cavin, and A. Schiper, *Probabilistic Broadcast for Flooding in Wireless Mobile Ad Hoc Networks*, Proceedings of the IEEE Wireless Communications and Networking Conference, pp. 1124-1130, March 2003.

[31] O. Dousse, Patrick Thiran, and M. Hasler, *Connectivity in Ad-Hoc and Hybrid Networks*, in INFOCOM. Annual Joint Conference of the IEEE Computer and Communications Societies, New York, pp. 1079-1088, June 2002.

[32] O. Dousse, F. Baccelli, and Patrick Thiran, *Impact of Interferences on Connectivity in Ad Hoc Networks*, in INFOCOM. Annual Joint Conference of the IEEE Computer and Communications Societies, pp. 1724-1733, March 2003.

[33] Zhi-Jie Tan, Chun Long, Xian-Wu Zou, Wei Zhang, and Zhun-Zhi Jin, *Epidemic Spreading in Percolation Worlds*, Physics Letters A, Vol. 300, pp. 317-323, July 2002.

[34] S. H. Strogatz, *Exploring Complex Networks*, Nature, Vol. 410, pp. 268-276, March 2001.

[35] A. Barabási, R. Albert, and H. Jeong, *Scale-Free Characteristics of Random Networks: The Topology of the World-Wide Web*, Physica A, Vol. 281, pp. 69-77, 2000.

[36] D. J. Watts and S. H. Strogatz, *Collective Dynamics of 'Small-World' networks*, Nature, Vol. 393, pp. 440-442, June 1998.

[37] Reza Olfati-Saber, *Ultrafast Consensus in Small World Networks*, Proceedings of the 2005 American Control Conference, Portland, OR, pp. 2371-2378, June 2005.

[38] Yoonsoo Kim and Mehran Mesbani, *On Maximizing the Second Smallest Eigenvalue of a State-dependent Graph Laplacian*, Proceedings of the 2005 American Control Conference, Portland, OR, pp. 99-103, June 2005.

[39] R. Merris, *Laplacian Matrices of Graphs: A Survey*, Linear Algebra and Its Applications, Vol. 197, No. 198, pp. 143-176, 1994.

[40] J. Cheriyan, *Fast Algorithms for k-Shredders and k-Node Connectivity Augmentation*, Journal of Algorithms, Vol. 33, No. 1, pp. 15-50, October 1999.

[41] Rajneesh Hegde, *Finding 3-Shredders Efficiently*, citeseer.ist.psu.edu/hegde02finding.html.

[42] R.D. Robinett, III, D.G. Wilson, and A.W. Reed, *Exergy Sustainability*, Sandia National Laboratories, SAND2006-2759, May 2006.

[43] R.D. Robinett, III and D.G. Wilson *Collective Systems: Physical and Information Exergies*, Sandia National Laboratories, draft SAND report, October 2006.

[44] R.M. Murray, Ed., *Control in an Information Rich World: Report of the Panel on Future Directions in Control, Dynamics, and Systems*, SIAM, Philadelphia, PA, 2003.

[45] P.J. Moylan, *Implications of Passivity in a Class of Nonlinear Systems*, IEEE Transactions of Automatic Control, Vol. AC-19, No. 4, August 1974, pp. 373-381.

[46] J.L. Wyatt, Jr., L.O. Chua, J.W. Gannett, I.C. Goknar, and D.N. Green, *Energy Concepts in the State-Space Theory of Nonlinear n-Ports: Part I - Passivity*, IEEE Transactions on Circuits and Systems, Vol. CAS-28, No. 1, January 1981, pp. 48-61.

[47] J.L. Wyatt, Jr., L.O. Chua, J.W. Gannett, I.C. Goknar, and D.N. Green, *Energy Concepts in the State-Space Theory of Nonlinear n-Ports: Part II - Losslessness*, IEEE Transactions on Circuits and Systems, Vol. CAS-29, No. 7, July 1982, pp. 417-430.

[48] R. Ortega, Z.P. Jiang, and D.J. Hill, *Passivity-Based Control of Nonlinear systems: A Tutorial*, Proceedings of the American Control Conference, Albuquerque, NM, June 1997, pp. 2633-237.

[49] J.C. Willems, *Dissipative Dynamical Systems Part I: General Theory; Part II: Linear Systems with Quadratic Supply Rates*, Archive for Rational Mechanics and Analysis, Vol. 45, pp. 321-393, 1972.

[50] D. Hill and P.J. Moylan, *The Stability of Nonlinear Dissipative Systems*, IEEE Transactions on Automatic Control, October 1976, pp. 708-710.

[51] P. Kokotovic and M. Arcak, *Constructive Nonlinear Control: A Historical Perspective*, Preprint submitted to Elsevier, August 2000.

[52] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*, Prentice Hall, Inc., N.J., 1991.

[53] M. Kristic, I. Kanellakopoulos, and P. Kokotovic, *Nonlinear and Adaptive Control Design*, John Wiley & Sons, Inc., New York, 1995.

[54] P.A. Ioannou and J. Sun, *Robust Adaptive Control*, Prentice Hall, Englewood Cliffs, N.J., 1996.

[55] A.A. Alonso and B.E. Ydstie, *Stabilization of Distributed Systems Using Irreversible Thermodynamics*, Automatica, Vol. 37, 2k1, pp. 1739-55.

[56] K.-H. Anthony, *Hamilton's Action Principle and Thermodynamics of Irreversible Processes - A Unifying Procedure for Reversible and Irreversible Processes*, J. Non-Newtonian Fluid Mechanics, Vol. 96, 2001, pp. 291-339.

[57] D.S. Scott, *Links and Lies*, International Journal of Hydrogen Energy, Vol. 28, 2003, pp. 473-476.

[58] D. Kondepudi and I. Prigogine, *Modern Thermodynamics: From Heat Engines to Dissipative Structures*, John Wiley & Sons, N.Y., N.Y., 1999.

[59] A. Greven, G. Keller, and G. Warnecke, *Entropy*, Princeton University Press, New Jersey, 2003.

[60] D.S. Scott, *Exergy*, International Journal of Hydrogen Energy, Vol. 28, 2003, pp. 369-375.

[61] L. Meirovitch, *Methods of Analytical Dynamics*, McGraw-Hill, New York, 1970.

[62] R.J. Smith, *Circuits, Devices, and Systems: A First Course in Electrical Engineering*, John Wiley & Sons, Third Edition, 1976.

[63] T.L. Saaty and J. Bram, *Nonlinear Mathematics*, McGraw-Hill, New York, 1964.

[64] R.D. Robinett, III, G.G. Parker, H. Schaub, and J.L. Junkins, *Lyapunov Optimal Saturated Control for Nonlinear Systems*, AIAA Journal of Guidance, Control, and Dynamics, Vol. 20, No. 6, Nov-Dec 1997, pp. 1083-1088.

[65] B. van der Pol, *Radio Rev.* **1**, 704-754, 1920 and B. van der Pol, *Phil. Mag.*, **3**, 65, 1927.

[66] E. J. Gottlieb, M. J. McDonald, F. J. Oppel III, B. J. Rigdon, P. G. Xavier, "The Umbra Simulation Framework as Applied to Building HLA Federates" *Proceedings of the 2002 Winter Simulation Conference*, San Diego, CA , Dec. 2002.

Appendix A

Land Agreement

To demonstrate the complex adaptive technologies developed by this LDRD, experiments were planned that involved a 100 node wireless sensor network, an unmanned ground vehicle, an unmanned aerial vehicle and simulated remotely/robotically operated weapon systems. A large tract of land would be needed to field this testbed. The Robotic Vehicle Range controls 220 acres of land, see Figure A.1 and the area marked in blue. However this area was mostly flat and could easily be monitored with video and infrared cameras. What was needed to properly demonstrate the capabilities of a large complex adaptive heterogeneous interlinked system was rugged terrain. Many Sandia controlled sites were inspected but none were found to be adequate. The best alternative was to obtain a ”Land Use Permit” from Kirtland Air Force base for land they controlled. A 93 acre area to the west of the RVR, shown in Figure A.1 (area marked in red) and in Figure A.2, was deemed to be suitable. This LDRD and the VPS project combined resources to raise the $14K needed to apply for the Land Use Permit. Additionally, the RVR provided personnel to assist with the application and to administer the permit.

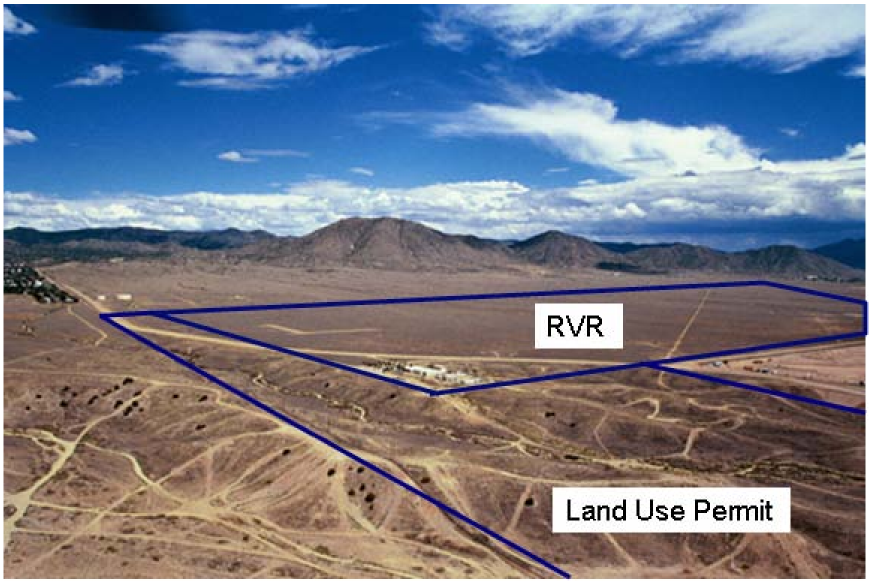


Figure A.1. Land use permit at the RVR

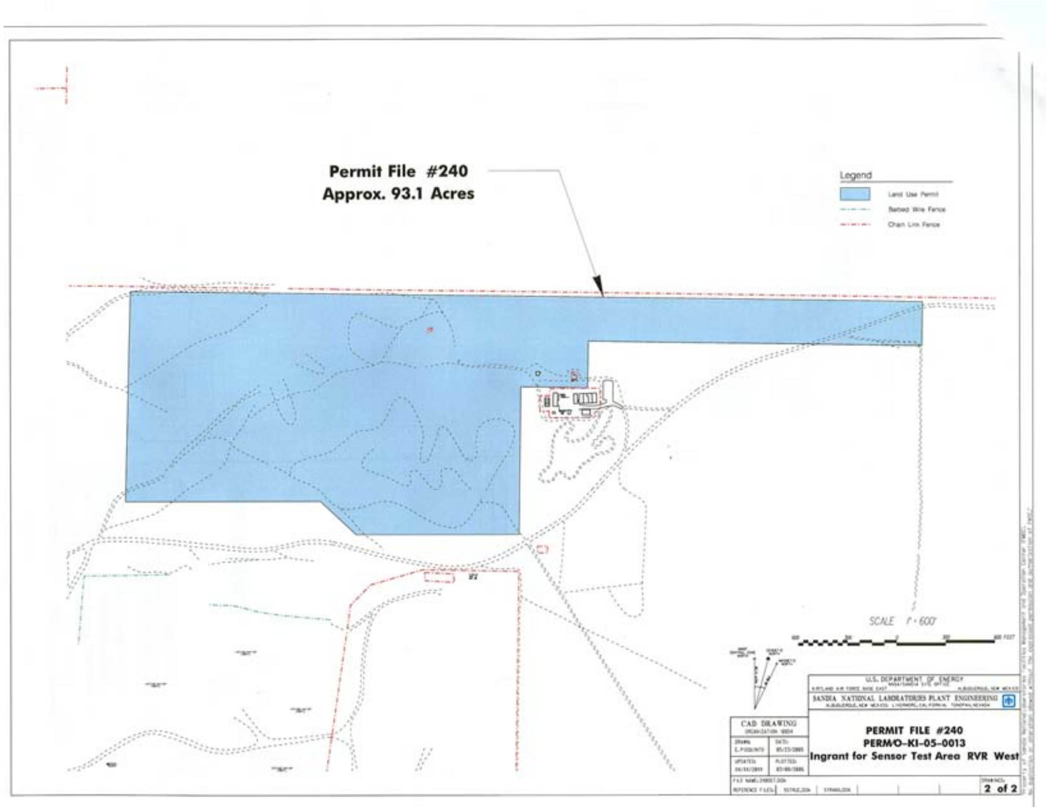

Permit File #240
Approx. 93.1 Acres
Legend
Land Use Permit
Barbed Wire Fence
Chain Link Fence
SCALE 1" = 600'
FEET
U.S. DEPARTMENT OF ENERGY
KIRTLAND AIR FORCE BASE EAST
ALBUQUERQUE, NEW MEXICO
SANDIA NATIONAL LABORATORIES PLANT ENGINEERING
ALBUQUERQUE, NEW MEXICO; LIVERMORE, CALIFORNIA; TONOPAH, NEVADA
CAD DRAWING
PERMIT FILE #240
PERM/O-KI-05-0013
Ingrant for Sensor Test Area RVR West
2 of 2

Figure A.2. Land use permit map

The permit took over a year to move through Air Force channels. Its duration is from January 1, 2006 to December 2010. To extend the permit, a renewal must be filed one year before the expiration date.

Both an Environmental Impact Analysis and an Environmental Baseline Survey were required as part of the permit. And the Air Force stipulates that the land be returned in the same condition as when it was given. No alterations of the area are allowed without prior Air Force approval. The permit allows for the placing of wireless sensor nodes anywhere in the area. UGVs, however, are restricted to existing roads and pathways.

Further information can be found in the Land Use Permit and supporting documents. Copies can be obtained from Kenneth Groom, Dan Pritchard, or Dan Puetz.

DISTRIBUTION:

1 MS 0769
Dennis S. Miyoshi, 06400

1 MS 1002
Stephen C. Roehrig, 06300

3 MS 1002
Philip Heermann, 06470

1 MS 1005
Russell D. Skocypec, 06340

1 MS 0762
Rebbecca D. Horton, 06420

1 MS 1161
Daniel M. Rondeau, 05430

3 MS 1003
John T. Feddema, 06473

1 MS 1007
Larry R. Shipers, 06471

1 MS 1125
Phil C. Bennett, 06472

1 MS 1009
Charles Duus, 06301

1 MS 0780
Stephen Ortiz, 06424

1 MS 1131
Regan Stinnett, 06429

1 MS 0780
Daniel A. Pritchard, 06428

1 MS 1161
Hung D. Nguyen, 05432

3 MS 1003
Kenneth N. Groom, 06473

1 MS 1003
Raymond H. Byrne, 06473

1 MS 1108
David G. Wilson, 06332

1 MS 1104
Rush D. Robinett, III, 06330

1 MS 1125
John J. Harrington, 06472

1 MS 1004
Brian J. Rigdon, 06344

1 MS 1010
Brandon R. Rohrer, 06473

1 MS 1003
Glen A. Laguna, 06473

1 MS 1133
Robert Bickerstaff, 06472

2 MS 9018
Central Technical Files, 8944

2 MS 0899
Technical Library, 4536

1 MS 0161
Legal Intellectual Property, 11500

1 MS 0123
D. Chavez, LDRD Office, 1011