NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

A STATISTICAL MECHANICS MODEL
OF COMBAT

by

Stephen C. Upton

March 1987

Thesis Advisor Lester Ingber

SECURITY CLASSIFICATION OF THIS PAGE

AD-A180483

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; Distribution is unlimited
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b OFFICE SYMBOL (If applicable) 55	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000	7b ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000
8a NAME OF FUNDING/SPONSORING ORGANIZATION 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)
A STATISTICAL MECHANICS MODEL OF COMBAT

12 PERSONAL AUTHOR(S)
UPTON, Stephen C.

13a TYPE OF REPORT	13b TIME COVERED FROM ____ TO ____	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Master's Thesis		1987 March	97

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Statistical mechanics, Simulated annealing, Stochastic models, Combat models, path integral

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

The purpose of this thesis is to model a Combat System utilizing modern methods of nonlinear nonequilibrium statistical mechanics. This initiates development of methods which eventually can be used as a decision aid to the commander in force planning, battle management, budgeting decisions, doctrinal evaluations, and combat analysis. A general method is developed and then applied to a particular battle scenario using the combat wargame JANUS. The method fits empirical data to a functional form (a Lagrangian) which describes the short time probability distribution of a set of order parameters. A maximum likelihood fit is obtained using a simulated annealing optimization

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION	
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	Unclassified	
22a NAME OF RESPONSIBLE INDIVIDUAL Lester Ingber	22b TELEPHONE (Include Area Code) 408-646-2687	22c OFFICE SYMBOL 61Il

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

19. Abstract (cont).

algorithm. The most likely states of the order parameters and the associated risks (variances) of those states ultimately depend on the detailed structure of the Lagrangian. A long time probability distribution of the order parameters can then be found by using the path integral.

Approved for public release; distribution is unlimited.

A Statistical Mechanics Model of Combat

by

Stephen C. Upton
Captain, United States Marine Corps
B.S., University of Idaho, 1978

Submitted in partial fulfillment of the
requirements for the degrees of

MASTER OF SCIENCE IN OPERATIONS RESEARCH
and
MASTER OF SCIENCE IN PHYSICS

from the

NAVAL POSTGRADUATE SCHOOL
March 1987

Author: _____
Stephen C. Upton

Approved by: _____
Lester Ingber, Thesis Advisor

_____
E. Rockower, Second Reader

_____
Peter Purdue
Department of Operations Research

_____
K. E. Woehler, Chairman,
Department of Physics

_____
K. T. Marshall
Dean of Information and Policy Sciences

_____
G. Schacher,
Dean of Science and Engineering

ABSTRACT

The purpose of this thesis is to model a Combat System utilizing modern methods of nonlinear nonequilibrium statistical mechanics. This initiates development of methods which eventually can be used as a decision aid to the commander in force planning, battle management, budgeting decisions, doctrinal evaluations, and combat analysis. A general method is developed and then applied to a particular battle scenario using the combat wargame JANUS. The method fits empirical data to a functional form (a Lagrangian) which describes the short time probability distribution of a set of order parameters. A maximum likelihood fit is obtained using a simulated annealing optimization algorithm. The most likely states of the order parameters and the associated risks (variances) of those states ultimately depend on the detailed structure of the Lagrangian. A long time probability distribution of the order parameters can then be found by using the path integral.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

Imagine you are the commander of a large force faced with the following situation. You have been ordered to defend a key piece of terrain. Intelligence sources estimate that an enemy force, approximately three times as large, is moving towards your position and is expected to arrive within a couple of hours. You now must make a decision on how to allocate your forces on the line of defense in order to repel the enemy's attack. You have several alternatives. You could leave your forces as is on a line defense. But you know the enemy will only attack a small portion of your front and use his overwhelming force to penetrate your position. You could place them in a dispersed defense. You ask your operations officer to develop other alternatives. You must have them quickly so that the defense plan can be promulgated to the units in a timely fashion.

The operations officer and his plans/analysis officer, armed with PIACA (pronounced $\overline{\text{PI}}$-CA), Path Integral Algorithm for Combat Analysis, begin to develop the alternatives. PIACA is a hardware/software package designed to give the commander the most likely results of decisions and the risks associated with that decision. By inputing information about their own forces and those of the enemy, and information relating to the type of mission, PIACA will give them the most probable outcomes of the forces (levels) at the end of some pre-selected time interval. By modifying the scenario slightly as to initial force levels and other parameters, they will then have a good idea of the best alternatives to present to the commander. The commander has now an objective evaluation of his alternatives and is able to make a more informed decision.

There will be occasions when the commander is under a severe time constraint and must make a decision based on incomplete information. He now has PIACA available as a powerful aid to combat planning and analysis. It is the purpose of this thesis to develop a stochastic model of combat and a generalized methodology based on that model for eventual use in PIACA. It is additionally shown how the model and the methodology can be used for a simple scenario based on data from the combat wargame JANUS. PIACA can also be used to evaluate new system hardware,i.e. weapons systems, analyze combat plans, and aid in the analysis of doctrinal changes.

Chapter 2 outlines the Lanchester theory of Combat systems. This chapter is provided as background. Chapter 3 introduces the concept of order parameters and discusses their relation with combat systems. Several possible order parameters for the combat system are presented. Chapter 4 develops the underlying mathematical theory and introduces the reader to the Lagrangian and the associated Path Integral, a mathematical physics approach to C^3 systems developed by Ingber [Ref. 1,2]. Chapter 5 develops PIACA as a generalized methodology for modeling combat systems. Chapter 6 gives several empirical examples. The first example uses a one order-parameter model with simulated data generated from a stochastic Lanchester equation with constant variance. This will be shown to be equivalent to a quadratic Lagrangian with the result that the distribution of the order parameters will be Gaussian with non-stationary means and variances. The second example is a two order parameter model using simulated data from a different stochastic Lanchester equation. The long time conditional distribution will be shown to be non-Gaussian even though the short time distribution is Gaussian with respect to the temporal changes of order parameters. These examples are provided to show the relationship between the stochastic Lanchester representation and the Lagrangian representation. The third example will begin with a Lagrangian representation. Data from the combat wargame JANUS is used to develop the functional form (Lagrangian). Then an analysis of the short time probability distribution of the order parameters using the Lagrangian is given. Chapter 7 concludes the thesis with a summary of the methodology, its importance and utility, and discusses further applications of PIACA and development of the full decision aid.

II. AN INTRODUCTION TO LANCHESTER THEORY

In this chapter, we outline the Lanchester model of warfare, both deterministic and stochastic. For a more detailed development, the interested reader should refer to Taylor [Ref. 3].

A. DETERMINISTIC MODELS

Lanchester originally formulated his model of combat as a set of differential equations, one being,

$$\dot{X} = dX/dt = -aY \quad \text{where } X(t_0) = X_0$$
$$\dot{Y} = dY/dt = -bX \quad \text{where } Y(t_0) = Y_0 \tag{2.1}$$

where X and Y are the number of combatants for each side and a,b are called attrition rate coefficients. This is Lanchester's aimed fire model. The other is

$$\dot{X} = -aXY$$
$$\dot{Y} = -bXY\ , \tag{2.2}$$

where the variables are defined above. Equation 2.1 when integrated yields the so-called "square-law"

$$b(X_0^2 - X^2) = a(Y_0^2 - Y^2) \tag{2.3}$$

which gives the interpretation that the more initial force level a side has the less his casualties. This equation assumes the casualty rate is proportional to the number of combatants. It is also referred to as a state equation. Equation 2.2 is referred to as the state equation for area fire, and assumes fire is distributed uniformly by the combatants. When integrated, equation 2.2 yields

$$b(X_0 - X(t)) = a(Y_0 - Y(t)) \tag{2.4}$$

and is called the Lanchester linear law. Although the above equations model simple combat quite well, they are limited in scope and have several disadvantages [Ref. 3]:

1. Constant rate coefficients
2. No force movement
3. Various aspects of logistics, C^3I, terrain, geographics, etc. are not considered.

This has led to various modifications which attempt to overcome the above shortcomings such as:

1. using variable rate-coefficients
2. modeling breakpoint or battle-termination conditions
3. modeling morale
4. modeling communications [Ref. 4], etc.

The basic disadvantage with using deterministic Lanchester equations is that in reality combat is a severely stochastic system, which leads us to our next topic.

B. STOCHASTIC MODELS

In attempting to define a stochastic model from a deterministic model we must recognize that this definition is not unique, i.e., there is a many-to-one mapping of deterministic systems into stochastic contexts. For example, one might arbitrarily add noise to equation 2.1 in the following manner,

$$\dot{X} = f(X+\eta) \tag{2.5}$$

$$\dot{X} = f(X)+f'(X)\eta + \text{ higher order terms} \tag{2.6}$$

where η has some distribution and zero mean. Another example might be to add noise in an additive way, such as

$$\dot{X} = -aY + g\eta \tag{2.7}$$

$$\dot{Y} = -bX + g\eta \tag{2.8}$$

where g is some constant multiplying the standard deviation of the noise. We could also formulate a model stochastically by developing a set of Kolmogorov equations. Once developed, all models should be able to answer questions concerning the outcome of the battle and other factors such as:

1. What is the probability of win for each side?
2. How does win probability change with initial force levels?
3. What is the expected length of battle?
4. What is the probability distribution of the force levels?

As is evident, there are a number of possibilities of stochastically modeling combat. In this thesis we will develop a stochastic model of combat which, as a side benefit, will incorporate an underlying physical explanation. It will have several advantages:

1. to model the stochastic nature of combat
2. to answer questions such as those above concerning the battle
3. to incorporate non-linearities in the model
4. with the methodology developed, to be able to fit empirical data to the model and thus have the potential of forecasting battle outcomes.

III. ORDER PARAMETERS AND COMBAT

A. INTRODUCTION

We will begin this discussion with assumptions and definitions. A battle will be defined as a combat engagement between two opposing forces constrained to a small geographic area. This will be our system that we will attempt to model. The state of the system will be defined as a collection of variables which, as a set describe the system at any time, t.

This system will be nonlinear, dynamic and stochastic: nonlinear, since the moments of the distribution of the state variables may be described as nonlinear functions of the other state variables; dynamic, since the state variables could be functions of time; and stochastic, because the variables will be random due to inherent noise in the system. This noise will reflect imperfect knowledge of the enemy's forces, weather, equipment failure, etc., and also of the commander's own forces, and may also be a nonlinear function of the state variables.

When modeling this system we have several alternatives. One alternative would be to use generalized stochastic differential equations as our model with the variables denoting the microscopic state of the system. This is a very intuitive approach, but there are mathematical difficulties in solving such large sets of coupled stochastic differential equations, and even more difficulties in interpreting the numerical results. However, there may be alternative sets of variables which define the system appropriately enough for a study of combat at a middle, i.e. intermediate level of aggregation, or "mesoscopic" level. A probability distribution of these new variables would allow us to make predictions of the variables at any time, t. We will call these new variables order parameters [Ref. 5]. The order parameters of the system will contain all the information inherent in the system, relevant to a specific "coarse-grained" scale to be studied, and should be kept at a minimum to allow easy assimilation by the commander essentially defining the appropriate scale of aggregation to be considered. This is one of the problems associated with complex C^3I and combat systems; i.e. we must be careful not to pass on too much information to overburden the command nodes.

B. REPRESENTATIONS OF COMBAT

We could describe this battle in several different, equivalent representations. For example, consider the grid shown in Figure 3.1 as a "coarse" geographic representation of our battle. The T_i^{μ} represent tanks of force μ in cell i where $\mu = 1,2$ and $i = 1\text{-}9$. Similar notation exists for the personnel, P_i^{μ} . The state variables would then incorporate all microscopic information such as velocity, position, ammo, etc. Another could be a time sequence event description, i.e. at each moment in time, a particular event occured and was duly noted in some log book. From a bird's eye view, a description could include movements of the forces geographically, and rates of change of both forces. In a global view, the overall evolution of the battle in time, and the search for underlying patterns in that evolution might be sufficient to describe the battle.

T_1^1 P_1^1 T_1^2 P_1^2	T_2^1 P_2^1 T_2^2 P_2^2	T_3^1 P_3^1 T_3^2 P_3^2
T_4^1 P_4^1 P_4^2 T_4^2	T_5^1 P_5^1 T_5^2 P_5^2	T_6^1 P_6^1 T_6^2 P_6^2
T_7^1 P_7^1 T_7^2 P_7^2	T_8^1 P_8^1 T_8^2 P_8^2	T_9^1 P_9^1 T_9^2 P_9^2

Figure 3.1 Coarse Grid Representing Geographic Position.

At each stage, or level of command, there is a need for a differing view of the battle, since at each level, the information needed is different. At the lowest level, concern might be for resupplies, i.e., ammo, fuel, requests for transportation or support external to the lower organization. At a higher level, concern is for allocating the resources among competing requests and determining the priorities associated with those requests. At still a higher level, only the developing outcome of the battle might be relevant.

We will attempt to describe an intermediate ("mesoscopic") level between the upper (commander's or "macroscopic") level and the lower (units or "microscopic") level which incorporates all the relevant information a commander would need in order to make his decision. In defining this level, a set of order parameters needs to be developed. Order parameters represent this mesoscopic level and are a specific aggregation of the microscopic or state variables. For instance, in the tank example in the previous section, the state variables represent the detailed information about the tank, i.e. velocity, position on the battlefield, training level of the crew, ammo supply, etc. A simple example of an order parameter in this case might just represent the number of tanks in a particular cell of the battlefield. An order parameter model would then develop the necessary interactions among cells, resupply considerations, capability degradation, etc. We will see an application of this through the examples discussed later.

As a first representation, both forces could simply be described as a certain number of personnel. We are then interested in describing this battle, given a set of initial conditions, in terms of force losses per unit time. This is equivalent to a Lanchester approach with noise, alluded to earlier. This is only one of several ways to derive a stochastic Lanchester equation. This is referred to as a Langevin equation in the physics literature. This could be mathematically described as shown:

$$dX/dt = f^X(X,Y) + g^X(X,Y)\eta \ ,$$

$$dY/dt = f^Y(X,Y) + g^Y(X,Y)\eta \ , \qquad (3.1)$$

where X = the number of blue forces available to engage the Y forces, Y = the number of red forces available to engage the X forces, $f^{X,Y}$ = functions relating average

numbers of blue forces and red forces, $g^{X,Y}$ = functions multiplying (square-root) variances of the background noise, a_{ij} = coefficients relating rates of blue force and red force losses, and η = background noise. For example, functional forms might be

$$f^X = a_{11}X + a_{12}Y \ , \tag{3.2}$$

$$f^Y = a_{21}X + a_{22}Y \ , \tag{3.3}$$

with similar equations for $g^{X,Y}$.

To attempt to solve this equation, we could put it on a computer, introduce some random noise (via a Monte Carlo simulation) and the aggregated output of many runs would give us at any time, t, via a probability distribution, the level of blue and red forces and any other variable which is dependent on these, such as force ratio, surviving maneuver force ratio, or some equivalent descriptive variable in which we are interested.

We have selected the form (equation 3.1) because the current state-of-the-art mathematical physics then permits us to develop extremely general nonlinear means and variances into representations suitable for analysis by methods of statistical mechanics.

C. EXAMPLES OF ORDER PARAMETERS

To better understand the concept of order parameters, let us take a physical system as an example, a gas confined to a box in thermal equilibrium. The gas can obviously be defined at a microscopic level by representing it as a collection of atoms with certain velocities, relative positions, collision rate with other atoms and the walls of the box, and some internal energy state. In analogy to the battle, the atoms would represent the individual personnel, their velocities and positions corresponding to their movements and geographical positions on the battlefield, and the collision rate could correspond to the engagement rate with the enemy. The internal energy state could relate to the amount of ammo, firepower, and possibly training level of the individual combatants. However, on a more global level, there is a pressure associated with the gas, a temperature, and a volume. One of the problems with the order parameter concept is to find these global variables associated with combat and relate them to something of use to the commander.

The order parameter concept can be used to describe systems far from equilibrium. Combat is obviously a system far from equilibrium, except possibly in some isolated cases. Since the objective of both commanders is to accomplish some mutually exclusive mission, the system will tend towards a solution which favors one or the other commander. In analogy with our gas in the box, suppose we lower the temperature of the gas. At a certain temperature, the gas becomes a liquid which is called a phase transition to a long range collective order. The question is then: Is there an analogous "phase transition" associated with our forces and how do such collective patterns of information represent themselves? At what "temperatures" related to the size of the $g^{x,y}$ functions does this occur? Is it a unique phenomenon, i.e. does it only occur at this "temperature"? What if we change the volume of the box, is there then some "phase transition" associated with our forces possibly relating to the change in geographic area of the battle? Are the order parameters of our physical system transformable to some similar order parameters of battle?

In answering these questions, we can arrive at some understanding of combat and relate this to our understanding of other physical systems which undergo the same or similar transformations when the state of the system is changed.

As a start, and following Bretnor [Ref. 6], two order parameters that seem likely are the destructive force and the vulnerability of the force. Destructive force is defined as the amount of combat potential which can be delivered to the enemy in order to destroy him. It includes the training, the readiness, the sustainability, the morale, the weapons mix, etc. of the force. It is obvious that these factors do change during the course of battle, and that their level certainly would indicate the success or failure of combat. The vulnerability of a force are those factors which degrade the capability of the force, i.e. position on the battlefield (terrain factors such as line of sight, cover, concealment, protective armor, etc.), lack of morale, discipline, or training, etc. As you can see, the vulnerability of a force is in some ways a degradation of the destructive force, yet they are not totally the opposite of the other. For example, a force in the open would have more vulnerability than one under cover, yet they would have the same destructive force. There are other examples, the point being that they are distinguishable order parameters, although we could effectively model the force using the destructive force and modifying it so that it is somewhat degraded when its vulnerability is increased. Nonlinear functions of the order parameters will be used to model scenarios in which the objective of the commander would be to attack the others vulnerability while exploiting his own destructive force.

IV. MATHEMATICAL FORMALISM OF THE MODEL

A. INTRODUCTION

In this chapter we develop the mathematical formalism of the model, and introduce mathematical objects such as the Lagrangian and the path integral. We will show there exist equivalent representations among the Langevin, Fokker-Planck and Path Integral descriptions of a stochastic system [Ref. 1,7]. We begin with a simple one order-parameter, non-linear model. The linear model is a special case of the non-linear model. We then fully develop the two order-parameter model which is used in the remainder of this thesis to illustrate the path integral method. Generalization to many order-parameters is made and included for completeness of the description. Assumptions of the model and their significance are given. The primary assumption is the requirement for a Gaussian-Markovian system. Finally, the relation to classical deterministic systems and the usefulness to classical statistical systems of the Lagrangian will be discussed.

B. ONE ORDER-PARAMETER, NON-LINEAR MODEL

The one order-parameter (1OP) model is not particularly useful in describing combat, but we present it for completeness and ease of illustration of the general model. The generalization to two or more order parameters can be easily made. Order parameters we could use are the ratio of the forces or the difference of the force levels.

We begin by labeling our order parameter X, for example, the ratio of Blue forces to Red forces. We are interested in how X changes with time. Within a time increment, Δt, we can write

$$X(t+\Delta t) - X(t) = \Delta t\, f(X(t)) \quad , \tag{4.1}$$

where $f(X(t))$ is some function to be fit. If Δt is assumed small and X is assumed to be continuous, we can then write

$$\dot{X} = dX/dt = f(X(t)) \quad . \tag{4.2}$$

In the context of describing combat equation 4.2 is referred to as a Lanchester equation and simply represents the mean or expected path of the order parameter for a large system.

We now want to add a noise term to equation 4.2 so we can model the severely stochastic nature of combat. Hence, the change in X can be written

$$\dot{X} = f(X(t)) + g(X(t))\eta \quad , \tag{4.3}$$

where η is the background noise with zero mean and variance = 1 (assumed) and $g^2(X(t))$ is the variance, which is not necessarily a constant. We also assume that η is Gaussian-Markovian (normally distributed "white noise"). The assumptions will be discussed in Section E. Equation 4.3 is referred to as a Langevin rate equation in the scientific literature, but we will refer to it as a generalized stochastic Lanchester (GSL) equation in the context of describing combat. This is only one way of obtaining a stochastic Lanchester equation. I.e., there is a many-to-one mapping of deterministic systems into stochastic contexts.

Associated with this GSL is a Fokker-Planck equation [Ref. 8] defining a differential equation of the conditional probability distribution, $P(X(t+\Delta t)|X(t))$, given as

$$\partial P/\partial t = -\partial(fP)/\partial X + 1/2\partial^2(g^2P)/\partial X^2 + VP \quad . \tag{4.4}$$

The function f represents a drift term and g^2 is the diffusion term of the probability distribution $P(X(t+\Delta t)|X(t))$. Sometimes a potential term, V, is present, which is often useful to analytically enforce boundary conditions.

Another representation exists for describing $P(X(t+\Delta t)|X(t))$ [Ref. 8]. For small time increments Δt, we assume

$$P(X(t+\Delta t)|X(t)) = 1/(2\pi g^2\Delta t)^{1/2} \exp(-L\Delta t) \tag{4.5}$$

where $L = (\dot{X} - f)^2/2g^2$ is the Lagrangian of the system, i.e. a Gaussian form for this conditional probability, with mean $(X(t)+f\Delta t)$ and variance $g^2\Delta t$, as follows for short times from equation 4.4 . It must be emphasized equation 4.5 is the short time

conditional probability distribution of $P(X(t+\Delta t)|X(t))$. With this representation, it is possible to obtain a long time conditional probability distribution $P(X(t)|X(t_0))$ through a path integral description. This is defined as (more precisely defined in Langouche, et. al. [Ref. 7] and Schulman [Ref. 9])

$$P(X(t)|X(t_0)) = \int\cdots\int dX_{t-\Delta t} dX_{t-2\Delta t} \cdots dX_{t_0+\Delta t} \times P(X(t)|X(t-\Delta t))P(X(t-\Delta t)|X(t-2\Delta t)) \times \cdots \times P(X(t_0+\Delta t)|X(t_0)) \,, \tag{4.6}$$

$$= \int\cdots\int \mathcal{D}X \exp\left(-\sum_{n=0}^{s} \Delta t L_n\right) \,,$$

where

$$\mathcal{D}X = (2\pi g_0^2 \Delta t)^{-1/2} \prod_{n=1}^{s} (2\pi g_n^2 \Delta t)^{-1/2} \, dX_n \,,$$

$t_n = t_0 + n\Delta t$ and $t = t_0 + s\Delta t$ where t_n are the intermediate time increments in the limits $s \to \infty$ and $\Delta t \to 0$. Equation 4.6 is called a path integral and is recognized as simply a Chapman-Kolmogorov equation. With the path integral, given some initial state at t_0, $X(t_0)$, we can determine the probability distribution of X at some later time t. The path integral is discussed in Appendix B.

The purpose of the previous discussion was to show the relationship between the GSL equations and the path integral description of combat. This will also be shown for the two order-parameter and the many order parameter models, but in the actual development of the model all that is required is a functional form of the Lagrangian. This will be discussed further in section F.

C. TWO ORDER PARAMETERS, NON-LINEAR MODEL

We now develop the path integral representation of combat for two order parameters. We begin, as before, with a set of Langevin equations, show the related Fokker-Planck equation, and finally the path integral description. Our emphasis is on

the formulation and the notation of the path integral, whereas the Langevin equations are used to support our intuition.

Suppose we are interested in our own force level and that of the enemy. We will use these as our order parameters and denote their level by X(t) and Y(t) for Blue and Red forces, respectively.

As before in the 1OP model, we will be interested in the change of X, and Y with time according to

$$X(t+\Delta t) - X(t) = \Delta t f^{1}[X(t), Y(t)] ,$$

$$Y(t+\Delta t) - Y(t) = \Delta t f^{2}[X(t),(Y(t)] , \qquad (4.7)$$

where the f^{i}, $i = 1,2$ are some functions to be fit, and Δt is some small time increment. If we assume continuity of the order parameters and for small enough Δt, we can write equations 4.7 as

$$\dot{X} = dX/dt = f^{i} ,$$

$$\dot{Y} = dY/dt = f^{2} , \qquad (4.8)$$

These are the Lanchester equations (deterministic).

We now assume that multiplicative noise (Gaussian-Markovian on $\dot{X}$ and $\dot{Y}$) is present and the order parameters are now modified according to

$$\dot{X} = f^{1} + g^{1}{}_{1}\eta^{1} + g^{1}{}_{2}\eta^{2} ,$$

$$\dot{Y} = f^{2} + g^{2}{}_{1}\eta^{1} + g^{2}{}_{2}\eta^{2} , \qquad (4.9)$$

where the $g^{\mu}{}_{i}$ are functions multiplying the variance of the background noise. If the $g^{\mu}{}_{i}$ were constants, then the η_{i}'s would simply contribute "white noise". The mean of the η_{i}'s are assumed to be zero. We will also assume the number of noise terms is equal to or greater than the number of order parameters [Ref. 10]. Equation 4.9 is our generalized stochastic Lanchester equation (GSL) for two order parameters.

The Fokker-Planck equation describing the evolution of the conditional probability distribution $P(\mathbf{X}(t+\Delta t)|\mathbf{X}(t))$ where $\mathbf{X}(t+\Delta t) \equiv \{X(t+\Delta t), Y(t+\Delta t)\}$ is

$$\partial P/\partial t = VP + \partial(-g^{\mu} P)/\partial M^{\mu} + 1/2\partial^2(g^{\mu\nu}P)/\partial M^{\nu}\partial M^{\mu} , \tag{4.10}$$

where $M^1 \equiv X$, $M^2 \equiv Y$ and V is a potential used to add constraints on the order parameters or to simulate boundary conditions. The indices $\mu,\nu = 1,\ldots, N$ where N is the number of order parameters (2 in this model). The g^{μ} and $g^{\mu\nu}$ are different functions from the g^{μ}_i in the GSL and are defined as

$$g^{\mu} = f^{\mu} + 1/2g^{\nu}_i \partial g^{\mu}_i/\partial M^{\nu} , \tag{4.11}$$

$$g^{\mu\nu} = g^{\mu}_i g^{\nu}_i . \tag{4.12}$$

We are now using the Einstein summation convention, whereby repeated indices in any term imply summation over those indices. In the 2OP model these are, for example when $\mu = 1$,

$$g^1 = f^1 + 1/2g^1_1 \frac{\partial g^1_1}{\partial X} + 1/2g^1_2 \frac{\partial g^1_2}{\partial X} + 1/2g^2_1 \frac{\partial g^1_1}{\partial Y} + 1/2g^2_2 \frac{\partial g^1_2}{\partial Y} , \tag{4.13}$$

(The compactness of the Einstein summation convention is evident here) and for $\mu = 1$, $\nu = 2$,

$$g^{12} = g^1_1 g^2_1 + g^1_2 g^2_2 . \tag{4.14}$$

Note the g^{μ}_i which are the variances of the microscopic noise sources are summed over and contribution from individual sources need not be fit in the path integral description. The path integral description of equation 4.9 is

$$P(X(t)|X(t_0) = \int\cdots\int \mathcal{D}X \exp(-\sum_{n=0}^{\infty} \Delta t L_n) ,$$

$$\underline{D}M = g_0^{1/2}(2\pi\Delta t)^{-1/2}\prod_{n=1}^{s} g_n^{1/2}\prod_{\mu=1}^{N}(2\pi\Delta t)^{-1/2}\,dM^{\mu}{}_n$$

$$\int dM^{\mu}{}_n \rightarrow \sum \Delta M^{\mu}{}_{in}\,;\; M^{\mu}{}_0 = M^{\mu}{}_{i0}\,,\; M^{\mu}{}_{s+1} = M^{\mu}{}_f\,,$$

$$L = 1/2(\dot{M}^{\mu} - g^{\mu})g_{\mu\nu}(\dot{M}^{\nu} - g^{\nu}) - V\,,$$

$$g_{\mu\nu} = (g^{\mu\nu})^{-1}\,, \qquad (4.15)$$

$$g_n = \det(g_{\mu\nu})_n\,.$$

This is the long time conditional probability distribution of our order parameters. The short time conditional distribution is

$$P(X(t+\Delta t)|X(t)) = g^{1/2}(2\pi\Delta t)^{-1/2}\exp(-L\Delta t) \qquad (4.16)$$

where L and g are as defined above.

This description is correct as long as we adopt an Ito or pre-point discretization of our order parameter, i.e.

$$M^{\mu}(t_n) = M_n{}^{\mu}\,,$$

$$\dot{M}^{\mu}(t_n) = (M^{\mu}{}_{n+1} - M^{\mu}{}_n)/(t_{n+1} - t_n)\,. \qquad (4.17)$$

and $t_n = t_0 + r\Delta t$. This affords us the luxury of a relatively simple Lagrangian. There exists a mid-point or Stratonovich discretization of the order parameter given by

$$M^{\mu}(t_n) = 1/2(M^{\mu}{}_{n+1} + M^{\mu}{}_n)\,,$$

$$\dot{M}^{\mu}(t_n) = (M^{\mu}{}_{n+1} - M^{\mu}{}_n)/(t_{n+1} - t_n)\,. \qquad (4.18)$$

This induces a curved or Riemannian space on the order parameters with the subsequent requirement of additional terms being added to the Lagrangian. The presence of the noise actually induces the non-Euclidean geometry of the μ-space and the variance $g^{\mu\nu}$ is the inverse of the μ-space metric, $g_{\mu\nu}$. The benefit of having a

mid-point discretized Lagrangian is that the associated Euler-Lagrange equations determine a variational principle. This allows us to derive a most likely path of the order parameter, without doing a full calculation of the long-time probability distribution [Ref. 1,7,11].

The preceding discussion was not meant to be rigorous, but to point out the subtleties in actually evaluating any of the functional forms. For simplicity we will use the pre-point discretization and not carry any of the Riemanian terms. An example of the two order-parameter model with an explicit form for the Lagrangian is given in Chapter 6.

Although we have developed a Lagrangian from the GSL equations in the preceding discussion, it is not necessary to do this in general, i.e. we can begin our model by assuming a functional form for the Lagrangian without having first written down the associated GSL equations. This is an important point. There is an algebraic relationship between the Lagrangian representation and the GSL representation (under the assumptions) and one could, in principle, derive the GSL from the Lagrangian and vice versa. There exists a large body of literature on combat modeling with Lanchester equations and thus the experience gained using that approach can be transformed to the Lagrangian approach. There also exists a large body of literature dealing with the applications of the Lagrangian approach to other large, complex, physical systems which can then be directly used to provide physical insight into the problems of modeling combat.

D. MANY ORDER PARAMETERS, NON-LINEAR MODEL

The extension to many variables can be made [Ref. 1,12]. Suppose now we are interested in modeling the spatial-temporal patterns of the order parameters and not simply the temporal patterns as before. To extend the 2OP, where $\mu = 1,2$, we now let $\mu = 1, \ldots, N$, where N is the number of order parameters we want to model. For one example of a many parameter model, suppose we divide the battlefield into distinct cells, labeled by $\alpha = 1, \ldots, m$. For example, in each cell we would examine the Blue and Red force levels as composed of tanks and personnel and as shown in Figure 4.1. The $f^{\mu,\alpha}$, where $\mu\alpha$ forms an enlarged index of the $\mu \times \alpha$ variable-space, and V can now incorporate NN (nearest-neighbor) interactions and N^2N (next-nearest-neighbor) interactions to account for external forces, such as higher level constraints, resupplies from adjacent units, actual movement of forces from cell to cell, etc. The model would be as follows,

$$P(X(t)|X(t_0) = \int \cdots \int \underline{D}X \exp(- \sum_{n=0}^{s} \Delta t L_n) \quad ,$$

$$\underline{D}M = g_0^{1/2} (2\pi\Delta t)^{-1/2} \prod_{n=1}^{s} g_n^{1/2} \prod_{\mu}^{N} \prod_{\alpha}^{m} (2\pi\Delta t)^{-1/2} dM^{\mu,\alpha}{}_n$$

$$g_n = \det (g_{\mu\nu,\alpha\beta}) \tag{4.19}$$

$$L = 1/2(\dot{M}^{\mu,\alpha} - f^{\mu,\alpha}) g_{\mu\nu,\alpha\beta} (\dot{M}^{\nu,\beta} - f^{\nu,\beta}) \tag{4.20}$$

$$M = \{M_n^{\mu,\alpha} | \mu = 1,\ldots,N \quad \alpha = 1,\ldots,m \quad n = 1,\ldots,s\} \tag{4.21}$$

It must be emphasized that we are only assuming a Gaussian distribution of the rate of change of the variables in time, and that the spatial distribution could be non-Gaussian. This is a modeling consideration when deciding on a functional form of the Lagrangian. It should also be emphasized the distribution is only Gaussian in the short time, and only in the post-point value of the variables, whereas the long time distribution could be any distribution.

E. ASSUMPTIONS

The primary assumption of the general model is that the system to be modeled is a Gaussian-Markovian system in the rate of change of the variables. This means there must be sufficient order parameters available to describe the system as Markovian, i.e. that the future state of the system only depends on the present state. This assumption comes into play in describing the short time conditional probability distribution of the order parameter. The Gaussian assumption states the short time conditional probability distribution is Gaussian in the post-point variables. This is a standard assumption made when dealing with many stochastic models. It is hoped the Lagrangian or path integral representation is very robust with respect to these assumptions. This means if the noise is non-Gaussian or there are not enough order parameters to ensure a Markovian description, then we can still obtain a reasonable path integral representation with these assumptions.

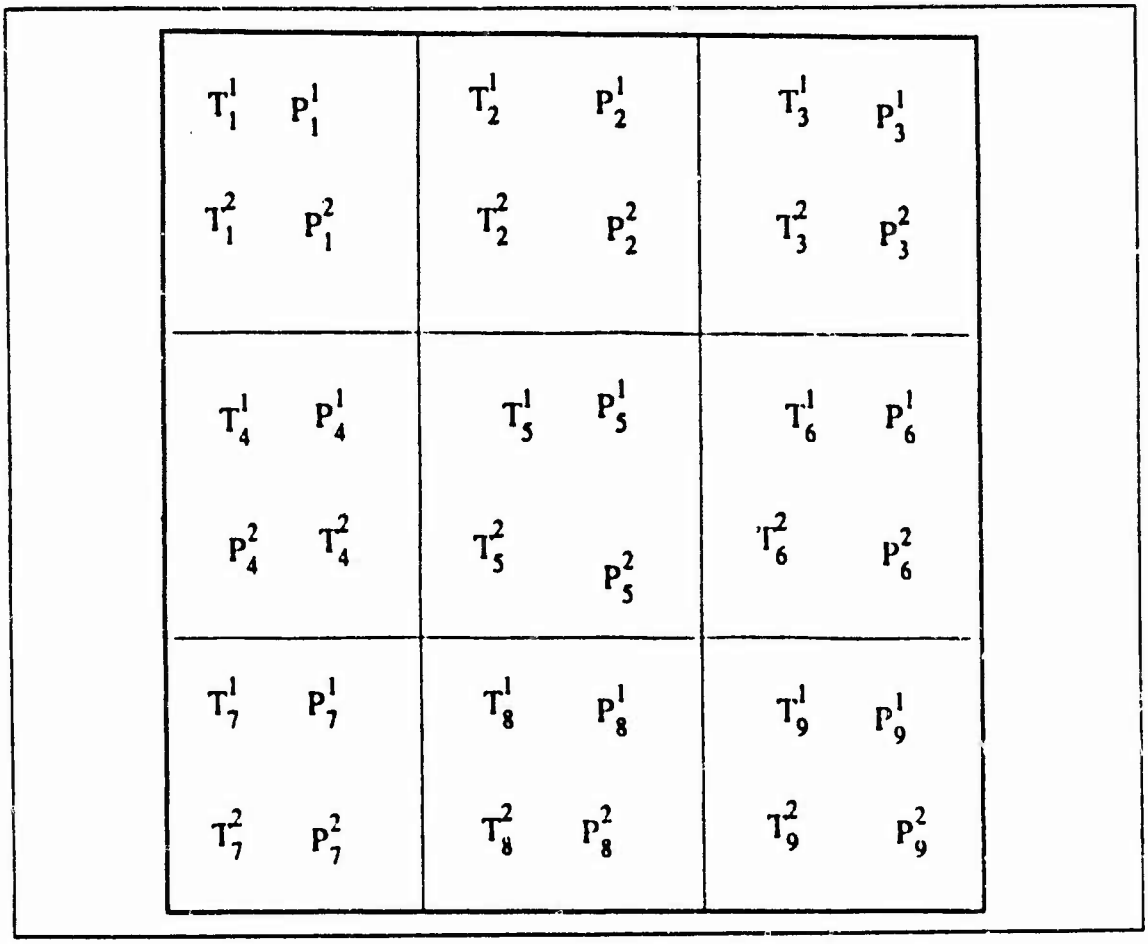


Figure 4.1 Battle Grid Showing Blue and Red Force Parameters.

F. INTERPRETATION OF THE LAGRANGIAN

Suppose we have a functional form of the Lagrangian which can be plotted as in Figure 4.2. We now show how the Lagrangian contains information about the system: most likely states of the system; a measure of the risk associated with that state; and a measure of the transition probability between most likely states.

The minima of the Lagrangian correspond to the most likely states of the system. We assume we are looking only at the short time probability distribution, P^s. The Lagrangian contains a widely varying expression containing factors of the difference between the actual state and the average or mean state, i.e. $L \propto [X_{t+1} - (X_t + f\Delta t)]^2$ where the term in the parenthesis is the past state corrected for the drift. Therefore, if the actual state is much different from the average state, then L will be a relatively large value compared to the other terms, and the corresponding value of the probability distribution will be correspondingly exponentially small.

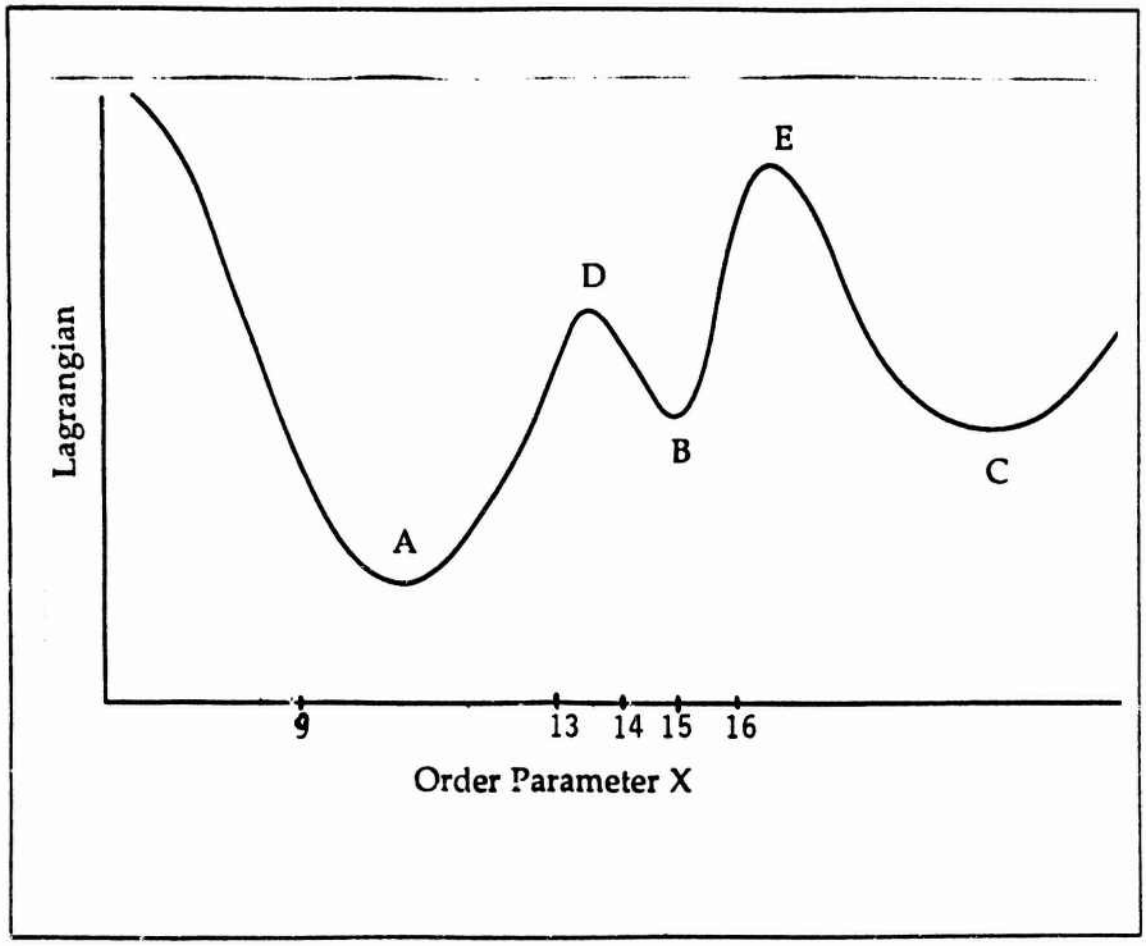


Figure 4.2 Plot of Lagrangian vs Order parameter.

The minima are shown as points A,B and C in Figure 4.2 and represent the most likely states of the system at time t.

The Lagrangian also contains the $g_{\mu\nu}$ term which is the metric of the order parameter space, i.e. it is a measure of the distance in this space. This space is curved for all metrics which are not constants, and we then have a short time Gaussian distribution in curved space.

The $g_{\mu\nu}$ term is also related to the variance $g^{\mu\nu} = (g_{\mu\nu})^{-1}$, of the underlying microscopic sources of noise which is a measure of the "width" of the minima. The minima width of a most likely state can be seen as a "degree of risk" measure associated with that state, i.e. the wider the minima the larger the confidence interval is for that state. For example, look at the minima at point A. If we were trying to forecast the value of the order parameter X, then we could only say it is between X = 9

and X=13, with any degree of confidence (to be defined). At point B we have a minima which is more sharply defined. We have most likely states of X=15, with a confidence interval of X=(14,16) and the $g^{\mu\nu}$ would be a much smaller term. The smaller the $g^{\mu\nu}$ the sharper the width of the minima. A measure of the transition probability between two most likely states is the relative height of the "peak" between the two "valleys" of the minima. For example, point D is much smaller than point E, and if the state of the system was at point B, then a transition to state A would be more likely than a transition to state C.

Granted the above discussion is very qualitative, but quantitative results can be obtained. However, a graphical device portraying the information contained in Figure 4.2 would be more useful as a qualitative tool than a quantitative one.

V. DESCRIPTION OF THE METHOD

A. INTRODUCTION

We will now present, in outline form, the generalized procedure for obtaining a path integral representation of combat systems.

- Select, derive or develop the order parameters of the system thereby defining the independent variables
- Obtain sufficient empirical data from the system you wish to model
- Functional forms of the independent variables are developed in terms of theoretical parameters/coefficients to be fit to data, to model means and variances
- Perform a maximum likelihood fit of the short time probability distribution, fitting coefficients of the functional form
- Using the path integral technique, a probability distribution of the order parameters is found for long times.
- Perform sensitivity analysis
- With the probability distribution, you can then use the method to
  - Analyze budget decisions in terms of hardware/software purchases
  - Perform combat analysis for use in battle management
  - Determine the effect of proposed doctrinal changes
  - Perform "What if" scenarios for use in combat planning
  - Have additional input to your decision making cycle

The method is an iterative process. We will collect some data, look for order parameters, attempt a fit, and if not very successful, try a new functional form of the Lagrangian. This process will continue until we have decided our assumptions are satisfied and we have a reasonably good fit to the data. Of course, after examining the structure of the Lagrangian and discovering that the model gives results which are not correct, then we must go back to the beginning or try a different functional form for the Lagrangian. We will now cover each of the steps in detail.

B. ASSUMPTIONS OF THE METHOD

Before we begin describing the method in detail, it would be appropriate to look at the assumptions. These assumptions will guide our development, and selection of the order parameters. Selection of the order parameters will then guide our data collection efforts.

As stated before in Chapter 4, our primary assumption is that the system to be modeled is Gaussian-Markovian. This assumption was necessary in developing the path integral. This assumption has further consequences in defining the amount of data required in order to sufficiently model our complex combat system.

There are several items which need to be addressed:

- number of elements in the battle; these can be personnel, vehicles, aircraft, ships, etc. If they are to be used as an order parameter, then there must be enough individual elements to ensure approximate continuity.
- number of runs of the experiment/war game/simulation; this is required in order to provide sufficient statistics for a good fit of the Lagrangian to the data, mainly in estimating the parameters of the $g^{\mu\nu}$, i.e. the variance, and the means g^{μ}.
- number of order parameters; this is required to ensure the system is Markovian. If not enough order parameters are used, then even if the true system is Markovian, our model of it may not be Markovian, which could result in a bad fit of the Lagrangian. If the model is robust, then a good fit could still be obtained if the number of order parameters used is not too different from the actual number of underlying order parameters. There is a subtle but important feature of our modeling which helps to create a robust fit: when care is taken to handle all nonlinearities, e.g., including Riemannian terms in the midpoint discretized Lagrangian, equation 4.18, then the probability distribution is invariant under nonlinear transformations of the variables. (This is what induces the Riemannian geometry [Ref. 10].) Thus we are really fitting a wide class of functional forms whenever we do one generic fit to the data.
- "uncertainty principle" $0 < \tau = \Delta t < 1/L$, $L \sim <(\Delta x)>^2 / (<(\Delta x)^2> \Delta t)$ where $\sim$ means on the order of. This places a requirement on the amount of change in the order parameter in a particular time increment. This is also necessary to calculate the path integral. This states that in a time increment τ, the average drift of the order parameter must be less than (or on the order of) the variance [Ref. 13]. Obviously, when considering actual data, i.e. from operational exercises or combat, then Δt will become part of the data and then we can only do the best fit we can with the data available.

- Aggregation (co-location) of capabilities; it is possible this type of model may not be adequate if the system is composed of only a few large distinct entities which have several capabilities. For example, a naval battle group may have large numbers of personnel and aircraft, but they are aggregated into a relatively few number of ships. Destruction of one ship's capability may have a large impact on the outcome of the battle. It still may be possible, however, to model some aspect of the naval battle, for example the group's outer air battle, which has sufficient numbers of elements i.e. aircraft. There is more interesting work that could be done here but is beyond the scope of this thesis.

In summary, we need a combat system which has a large number of elements to ensure approximate continuity, a sufficient number of order parameters, and a sufficient number of experiments to provide for a good fit of the Lagrangian. Once a Lagrangian is developed and coupled with the path integral, we will have a "propagator" to describe the time evolution of the system from any initial time t_0 to any final time, t.

C. SELECTION OF ORDER PARAMETERS

The development of the order parameters is first dependent upon the system you wish to study. For example, if you were interested in the length of a battle as defined by some cutoff strength for either side, then the order parameters used might be just the force levels of each side. If you were interested in the relation of C^3 to the outcome of a battle, then you might use some MOE of C^3 of either side, together with the force levels.

Second, the order parameters used must satisfy the aforementioned assumptions. Obviously, an order parameter must be something which changes value during combat and cannot be a constant or a very slowly changing variable. Otherwise there would be no need to model that particular order parameter.

Some examples of order parameters (per side) are:

1. number of vehicles (tanks, trucks, aircraft, etc.)
2. number of aircraft
3. number of elements firing (artillery/tanks/aircraft)
4. number of shots fired
5. number of tanks (armor study)
6. supply/logistics availability
7. troop carrying capacity (helicopters/tactical troop carriers)

8. geographical position on the battlefield (appropriate for motorized infantry)
9. number of aircraft/sector (outer naval air battle)
10. log(bits) of information used in communications

Through the appropriate use of the order parameters and the path integral other questions such as the outcome of the battle and duration of combat can be answered.

D. DATA COLLECTION

The first step in the analysis is to obtain empirical data. This could be the result of simulations, war game results, field exercise data or data from actual combat. Each of the data sources has its advantages and disadvantages. If the data were available, the best source would be from actual combat. Obviously, there have been no large scale wars recently, but that does not prevent us from doing analysis on past wars. This will not be the approach here. There is data available from field exercises, but it is not in a suitable form for analysis at the present time. This includes data from CAX's (Combined Arms Exercises) which are held at Twentynine Palms, California by the Marines, or exercises conducted at Fort Irwin by the Army on their calibrated range. This could be done at a later time. War games are the next best place to obtain data. Several War Games available at NPS are JANUS (after the mythological two-faced god) and IBGTT (Interactive Battle Group Tactical Trainer). TWSEAS (Tactical Warfare Simulation, Evaluation and Analysis System) is a war game available at the Marine Corps Development Center in Quantico, Virginia. These simulations are discussed in the Appendix C. Other simulations available are CARMONETTE, SOTACA, and FOURCE to name a few. A brief description of each is also included in Appendix C.

For the purposes of constructing a statistical mechanics model of combat, many trajectories of the order parameters are needed. What do we mean by trajectories of the order parameters? In the space defined by the order parameters, a point represents one possible state of the order parameters. A path which connects the initial state of the system to some final state is called a trajectory. The trajectory represents one possible realization of combat. For a good fit of the model to the data many trajectories, and therefore, many stochastic experiments are needed. This will naturally leads us to select a war game or simulation as a source of data. In these cases, many experiments can be completed with variations in the noise.

E. DEVELOPMENT OF THE LAGRANGIAN

The next step in the analysis is to determine a functional form of the Lagrangian. The simplest and most versatile form is a ratio of polynomials. This defines a Padé approximant form and is suitable for approximating many functional forms. The Lagrangian is

$$L = 1/2(\dot{M}^{\mu} - f^{\mu})g_{\mu\nu}(\dot{M}^{\nu} - f^{\nu}) \tag{5.1}$$

where we need to assume functional forms for the f^{μ} , $\mu = 1,\ldots, N$ (N is the number of order parameters) and the variance $g^{\mu\nu}$. Since $g_{\mu\nu}$ is a metric it must be positive definite, i.e. det $(g_{\mu\nu})>0$. Except for this one condition, the Lagrangian can be of any form. Obviously, we would like to keep the form simple, yet model the data accurately. This might require several iterations of:

1. select a functional form of the Lagrangian
2. performing a maximum likelihood fit to determine the unknown coefficients,
3. testing the fit of the Lagrangian with the data
4. and if not satisfactory, go back to 1.

If, after several iterations, a good fit has not been attained, then we must look at our data to ensure we have satisfied our assumptions. This could be one test to see if we satisfy our assumptions.

F. MAXIMUM LIKELIHOOD FIT

After deciding on a set of order parameters defining independent variables and a functional form of the Lagrangian, we are now in a position to estimate the parameters/coefficients of our Lagrangian. This will be accomplished by using a maximum likelihood fit.

The short time conditional likelihood function, M, is defined to be

$$M \equiv (2\pi\Delta t)^{-1/2}\, g^{1/2} \exp(-L\Delta t) = P(\mathbf{X}(t+\Delta t)|\mathbf{X}(t)) \ , \tag{5.2}$$

$$L = 1/2(\dot{M}^{\mu} - f^{\mu})g_{\mu\nu}(\dot{M}^{\nu} - f^{\nu}) \tag{5.3}$$

where the other variables are as previously defined. Our data is in the form of I runs for $J\Delta t$ time periods, where one run is one realization of combat and J is the duration of combat. Our maximum likelihood function for many runs now becomes

$$M' = \prod_{i=1}^{I} \prod_{j=1}^{J} (2\pi\Delta t)^{-1/2} g_{ij}^{1/2} \exp(-L_{ij}\Delta t)$$

$$= \prod_{i=1}^{I} P(X_i(t_0+j\Delta t)|X_i(t_0+(j-1)\Delta t) \times \ldots \times P(X_i(t_0+\Delta t)|X_i(t_0)) \quad , \qquad (5.4)$$

where $X_i(t_0+j\Delta t) = \{X_i(t_0+j\Delta t), Y_i(t_0+j\Delta t)\}$,

and where the data is used to calculate specific values of the X's. We now wish to find the parameters in the Lagrangian which maximizes this function. To do this we first take the logarithm of M' to accommodate computer requirements on acceptable ranges of numbers which can be processed. Since the logarithm is a positive monotone function, the maximum of $\ln M'$ will be in the same location as that of M'. Therefore we now wish to maximize

$$\ln M' = \sum_{i=1}^{I} \sum_{j=1}^{J} [-1/2\ln(2\pi\Delta t) + 1/2\ln g_{ij} - L_{ij}\Delta t \,] \qquad (5.5)$$

$$= - \sum_{i=1}^{I} \sum_{j=1}^{J} [-1/2\ln(2\pi) + 1/2\ln(\Delta t) - 1/2\ln g_{ij} + L_{ij}\Delta t \,]$$

The maximization of $\ln M'$ is equivalent to a minimization of $-\ln M'$. Also the constant $1/2\ln(2\pi)$ can be deleted from $\ln M'$ since it will not affect the location of the minimum. In general, Δt will be part of the data and thus we will not drop this term. Therefore our problem is to locate the minimum (global, if possible) of

$$N = \sum\sum [1/2\ln(\Delta t) + L_{ij}\Delta t - 1/2\ln g_{ij} \,] \qquad (5.6)$$

Current algorithms for solving non-linear minimization problems are deterministic and only guarantee local minima. Therefore we have developed a version of the simulated annealing algorithm which guarantees convergence to the global

minima if certain conditions are satisfied. The algorithm uses a random sampling rule to select points from the parameter/coefficient space, and a criteria on which to base acceptance of that point as the new state of the fit. We use a Cauchy distribution tied into a "temperature" on which to sample the space, and as the temperature is lowered, the search is more localized. The Cauchy distribution (a long tailed, ∞ variance distribution) is used so that the localized search does not get trapped in a local minima, and there is some probability of leaving to find better optima. The Boltzmann distribution (from the Metropolis algorithm) is used as a criteria on which to accept the point. This distribution is particularly useful in our case, because the cost function is in the form of a Boltzmann distribution and thus there is a more efficient mapping to the parameter space. A description of the algorithm, the necessary conditions, and a FORTRAN program of the simulated annealing algorithm are contained in Appendix A.

There are a few subtle points to mention concerning data, variables, parameters and spatial dimensions. The paths which are generated from a source of data are equivalent to a set of likely paths which can be sampled from the theoretical distribution of the variables. This is usually what is meant when discussing sampling statistics. The data are a sample from some theoretical distribution which is at present unknown. The parameters are the coefficients in the functional form of the Lagrangian and the $g_{\mu\nu}$ metric and are to be estimated from the data. At the time we are fitting the data, M becomes a function of the parameters with the variables being the data. Once the Lagrangian is fit to some set of the data, it then becomes the fitted distribution of the underlying order-parameter variables which attempts to mimic the unknown underlying theoretical distribution. The "dimension" of the space of variables is determined by the number of order parameters being considered. If two or three spatial dimensions are being considered, then typically separate cells within this space would contain independent order parameters, which would be functionally tied to the order parameters in other cells by the functional forms used to define the multivariate drifts and diffusions.

It should be stressed that fitting the Lagrangian does not mean separately fitting the means and diffusions to the same accuracy. That is, we are fitting the functional form of the Lagrangian to the data and NOT finding a set of parameters/coefficients in the drift and diffusion terms of the data. Thus there may be a "wide" discrepancy between parameters/coefficients using similar data, but it is the resultant fit of the Lagrangian to the data that is most important.

G. PATH INTEGRAL TECHNIQUE

The most important part of the analysis is now complete, i. e. developing a functional form of the Lagrangian. With the Lagrangian in hand we are in a position to calculate the long time conditional probability distribution P^l defined as

$$P^l \equiv P(X(t)|X(t_0) = \int \cdots \int \underline{D}X \exp(-L\Delta t) \ , \tag{5.7}$$

where $t_0 < t$ and t can be any time chosen in the future. This is the path integral and acts as a propagator of the system, i.e. once the path integral is known, given any initial state, the probability distribution of the order parameters at any other time can be calculated. For simple systems, Monte Carlo techniques are used for multi-dimensional systems. However, there is only one method, recently developed, that has proved to be accurate for a wide range of nonlinear, nonstationary problems, such as those we expect to be present in combat systems. At the present time, using this method, the path integral has been calculated in one dimension [Ref. 13,14,15] for many highly nonlinear systems, and it has recently been extended to two dimensions [Ref. 16]. Work is ongoing at Lawrence Livermore and Sandia National Labs to develop an algorithm for the many dimensional case. However, meaningful results can still be obtained by examining the static Lagrangian where $X \sim 0$. This gives some indication of the characteristics of the system before doing long calculations.

H. VALIDATION AND SENSITIVITY ANALYSIS

Now we are in a position to check the sensitivity of the model to changes in the data. The Lagrangian essentially contains all the elements of the model. This is why it is so important to obtain a good fit. This analysis could be done in two ways:

1. With many runs of the data, we can separate (randomly) a set of runs to do our fit and then validate the fit using the remaining set of runs. It is not clear at the present time how to determine a good fit but a method that seems reasonable is determining deviations from the most likely path in the following manner. First, at each time increment, calculate the sample variance of the data. This will be our weighting factor. Next, calculate the distance between each of the data points and the most likely state calculated from the Lagrangian. This will be our "width". We then form the following statistic:

$$1/(n_e(n_t-1)) \sum(\Delta w_{it})^2/s^2_t = 1/(n_e(n_t-1))\sum(x_{it}-\tilde{x}_t)^2/s^2_t \tag{5.8}$$

where $s^2_t = 1/(n_t-1)\sum(x_{it}-x_t)^2$, x_t is the most likely value at time t, x_t is the sample mean at time t, n is the total number of data points, n_e is the number of experiments and n_t is the number of data points time t. The distributional properties under our assumptions of this statistic remain to be seen. One property which is evident is if the long time conditional distribution is symmetric then x_t and x_t are equal and our statistic will be unity. Another is if the data were to fall within one standard deviation s of the most likely path for all time, then the statistic will be close to 1. In this case a good fit would be evident if the statistic were "close" to 1.

2. Using a set of runs of a particular scenario, fit a Lagrangian. Now change a parameter of the scenario, for example, starting force levels. Next use a goodness of fit test such as described above to check the sensitivity of the Lagrangian to the change in the scenario parameter. This could be done for several parameters or several iterations of the same parameter.

Only after a full sensitivity analysis has been completed and the Lagrangian can be shown to be robust, can we say the Lagrangian accurately models the scenario(s).

I. OPTIONAL USES

We now present several generic uses of the path integral method and show some of its usefulness and versatility as a combat model.

1. Procurement Decisions

Suppose you are a commander in charge of procurement decisions. You are faced daily with comparing C^3 systems to tanks to aircraft to field artillery pieces and up to now have relied mainly on subjective or qualitative models. With PIACA, the comparison between different items of equipment weapons is summarized into comparing probability distributions and actually compare the items value in changing the outcome of combat.

For example, suppose we are interested in comparing the relative worth of purchasing a new C^3 system or purchasing more tanks. A study could be conducted as follows:

1. Obtain a sufficient physical model of the C^3 system which can be simulated or used in a war game. A similar requirement exists for the tanks. Although this seems somewhat complicated, this step is usually completed for most comparisons.

2. Once a sufficient physical model has been obtained, conduct many runs of a simulation/war game using first one system, then the other using various appropriate scenarios.
3. We now fit a Lagrangian to both data sets and develop a short time conditional distribution and a long time propagator or path integral.
4. We now have common objects, the Lagrangian and the path integral in which to compare the two different systems. We might then compare the most likely states of the two Lagrangians and determine if they are satisfactory. Or we might compare the long time distributions or develop some MOE which combines features of the Lagrangian and the path integral.

The are several advantages to this approach. First, fewer simulations are required since the information present in the scenario is captured by the Lagrangian within a relatively small number of runs. Second, once the fit has been obtained it is possible to perform additional sensitivity studies using the Lagrangian without requiring more, expensive simulations. This leads to a quantitative and objective tool which can be used by procurement managers.

2. "What If" Scenarios in Combat Planning

Now that we have seen how to use the method in procurement decisions, it is not much of a jump to the use in combat planning. There is a common thread to the method, the comparison of similar quantities, the Lagrangian and the path integral. For illustrative purposes we present an example of the method in combat planning.

We now suppose we want to examine the effectiveness of several combat plans. These are not as dissimilar objects as before, but we now have an additional objective evaluation which we may use. A study would follow similar lines as before:

1. Perform many runs of simulations for each combat plan. This alone would aid in understanding the significance or utility of each plan.
2. Fit a Lagrangian to each data set. Perform any required/desired sensitivity analysis.
3. Calculate the long time distribution via the path integral.
4. Develop any MOE's or compare the distributions in a qualitative manner.

Another advantage of the method is the user becomes more involved and is able to see the consequences of each plan more fully. Examining the structure of the Lagrangian enables him to see transition points in the developing outcome of the battle which might be critical points. With this information, he can then make more informed decisions regarding the evolution of the battle.

3. Doctrinal Evaluations

This use again follows the same procedure and we will suggest several examples. The procedure followed is the same as above.

One example could be to examine the effect of a change in armor tactics due to a small change in the weapon system. A large change in the weapon system might require a new study and therefore a new Lagrangian.

Another example is to look at a change in defensive tactics such as the difference between using a line defense and using a dispersed defense.

We could look at a change in helicopter tactics as a final example.

I have attempted to give the flavor of the possibilities of the method. It is extremely rich in applications and much interesting work can be accomplished.

J. SUMMARY

We have provided a general methodology for developing a statistical mechanics model of combat in this chapter. It is as follows:

- Derive or develop the order parameters of the system
- Obtain sufficient empirical data from the system you wish to model
- Functional forms of the order parameters are developed to model means and variances
- Perform a maximum likelihood fit of the short time probability distribution, fitting coefficients of the functional form
- Using the path integral technique, a probability distribution of the order parameters is found for long times.

We have suggested several possibilities of the method for use in procurement, combat planning, and doctrinal evaluations. The utility of the method is being able to extract the essence of a combat system and representing it by a functional form, i.e. the Lagrangian. Once the Lagrangian has been found, it then becomes possible to predict future outcomes with some degree of statistical (un)certainty.

We have also argued that although the Lagrangian is specific for a particular scenario, it may be robust enough to small changes in the scenario. A collection of Lagrangians is possible for example, to describe differing scenarios such as cold weather, desert, or jungle environments, defensive versus offensive tactics/strategies, differing force structures and differing advantage factors, i.e. either for or against the enemy. This collection after suitable testing can then be incorporated into the decision aid PIACA, for use by the commander and his staff.

VI. THREE EXAMPLES OF THE METHOD

A. INTRODUCTION

In this chapter we look at three examples of the use of the path integral or Lagrangian representation of combat. The first is a one order-parameter model with constant variance. This will lead to a quadratic Lagrangian which will imply a Gaussian distribution at each time step with means and variances being functions of the order parameters. The second example is a two order-parameter model with muitiplicative noise. The fit becomes more difficult but results can be obtained. For both of these examples, simulated data from the associated GSL is used and it will be shown that the coefficients used in the GSL can be obtained from the fit using the Lagrangian representation. These two examples are provided to illustrate the relationship between the GSL and the Lagrangian representation, and as a test of the computer code of the maximum likelihood fit program. It must be emphasized that we will be fitting a Lagrangian to the data and not merely to find coefficients which are exactly the same as those in the GSL. Thus we will compare the theoretical Lagrangian (derived from the GSL) and the empirical Lagrangian which is fit from the data. In the third example we show how to proceed with the method when given a set of empirical data which was taken from the war game JANUS. Due to time constraints it was not possible to perform the necessary fit to the data.

B. ONE ORDER-PARAMETER MODEL

Although as stated before, the one order-parameter model may not be a good model for combat, we present it here as a simple exposition of the mathematics and the methodology.

1. Data Collection and Order Parameter Used

The data was generated from a generalized stochastic Lanchester equation of the form

$$\dot{X} = aX + g\eta \,. \tag{6.1}$$

The η are distributed N(0,1). (Note: This is, in fact, an Ornstein-Uhlenbeck process.) The APL program used to generate the data for this example is given in Appendix D.

The order parameter represents, for our example, the Blue force level, which is being attrited through some process such as an essentially constant Red force level, i.e. $a = \alpha Y$. The data consists of 20 runs of the simulation for total time increments of $s = 20$, and $\Delta t = 0.1$. Therefore the total time of one simulation represents, $s\Delta t$, or say 2 minutes. We are assuming there is no explicit time dependence in the Lagrangian, and any time dependence is implicit in the variable X. Our GSL for the data is then

$$\dot{X} = -.1X + \eta \text{ where } \eta \sim N(0,1) \; . \tag{6.2}$$

For generating data we take the form

$$X(t+(j+1)\Delta t) = X(t + j\Delta t)+(-0.1X(t + j\Delta t)+\eta)\Delta t \tag{6.3}$$

Sample trajectories for 20 runs are shown in Figure 6.1. It should be noted here that all figures in this this chapter used the powerful statistical and graphics capability of GRAFSTAT. Figure 6.2 plots the sample distribution for each time increment and also connects the means of the distributions. It is obvious here that the expected value path is the solution of the deterministic equation, i.e. $<\dot{X} - aX = \eta> = <\dot{X}> - a<X> = 0 \rightarrow <\dot{X}> = a<X>$ since $<\eta> = 0$. The notation <argument> signifies the expected value of the argument. This will be seen to be the case when the Lagrangian is quadratic.

We now pick the short time conditional distribution representation to fit this data:

$$P(X(t+\Delta t)|X(t)) = 1/(2\pi g^2\Delta t)^{1/2} \exp(-L\Delta t) \tag{6.4}$$

where $L = (\dot{X} - f)^2/2g^2 = (\dot{X}-\hat{a}_1X)/2\hat{a}_2^2$ and where the $\hat{a}_i$'s are parameters to be estimated. For this example,

$$P(X(t+\Delta t)|X(t)) = (2\pi\Delta t)^{-1/2} \exp\{-1/2[X(t+\Delta t)-X(t) \\ -(\hat{a}_1X(t))\Delta t]^2/\hat{a}_2^2\Delta t\} \tag{6.5}$$

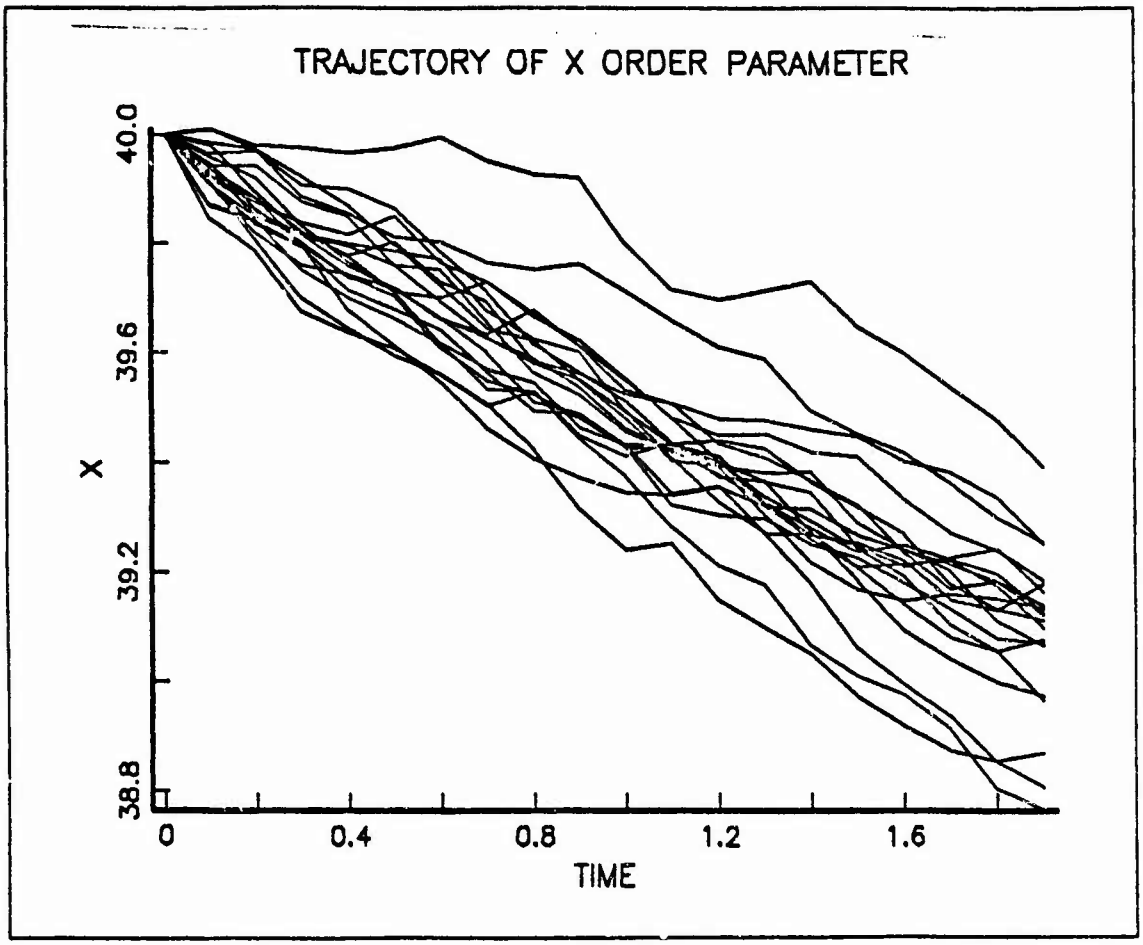


Figure 6.1 Trajectory of Order Parameter X.

and we can see this is in a Gaussian form with mean $\mu = X+(\hat{a}_1 X)\Delta t$, and variance $\hat{a}_2^2\Delta t$. We also see the Lagrangian in equation 6.5 is in quadratic form.

2. **Maximum Likelihood Fit of the Lagrangian**

We will use a maximum likelihood fit to estimate the parameters $\hat{a}_1$ and $\hat{a}_2$ in the Lagrangian. Therefore we want to maximize

$$M = \prod_{i=1}^{I} \prod_{j=1}^{J} P(X_i(t+(j+1)\Delta t)|X_i(t+j\Delta t)) \qquad (6.6)$$

or equivalently to maximize ln M,

$$\ln M = \sum_i \sum_j \ln P(X_i(t+(j+1)\Delta t)|X_i(t+j\Delta t))$$

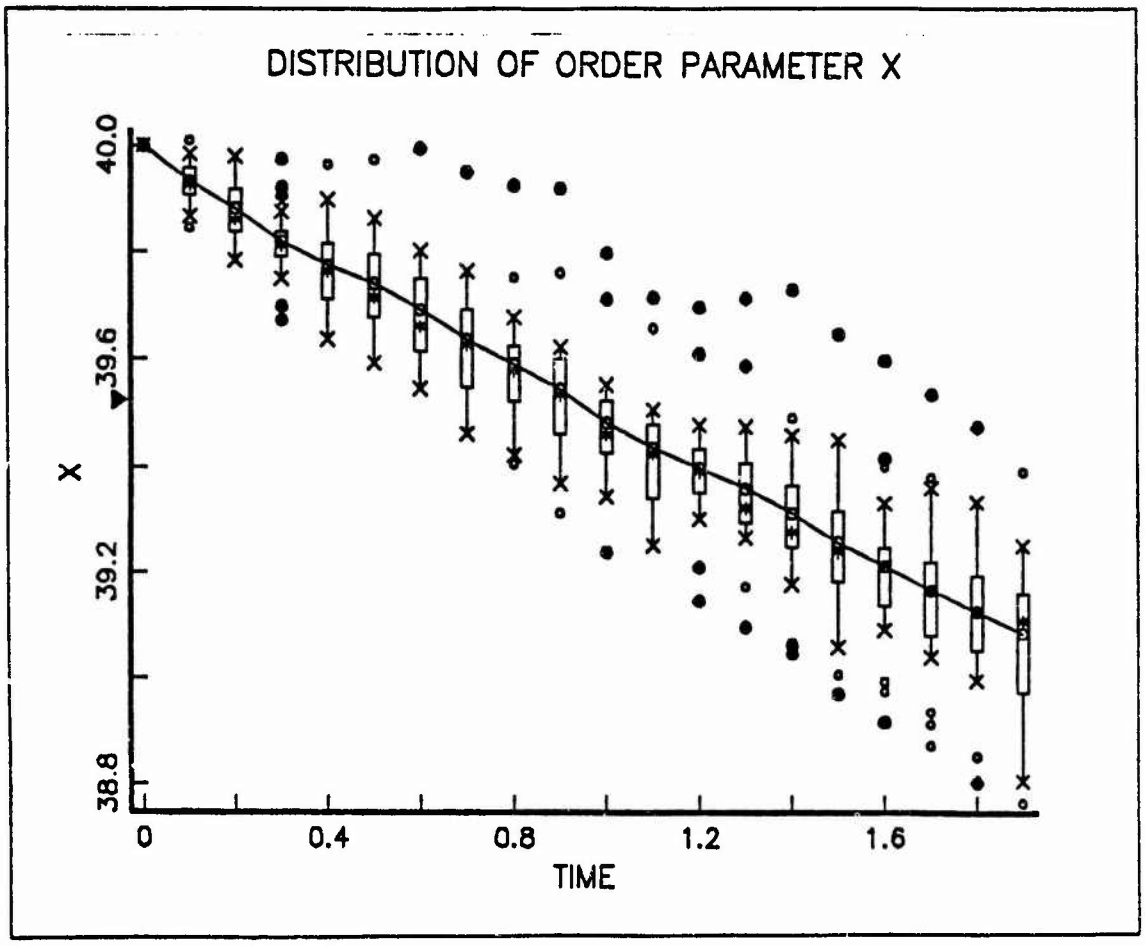


Figure 6.2 Distribution of Order Parameter X.

$$= \sum_i \sum_j (1/2\ln(2\pi\Delta t) + 1/2\ln g^2 + L_{ij}\Delta t) \quad .$$

To use the simulated annealing algorithm described in Appendix A, we need our cost function (likelihood) in terms of a minimization. Therefore, we want to minimize

$$Q = -\ln M = \sum\sum (1/2\ln(2\pi\Delta t) + 1/2\ln g^2 + L_{ij}\Delta t)$$

$$= \sum\sum \{1/2\ln(2\pi\Delta t) + 1/2\ln \hat{a}_2^2 + 1/2\Delta t[X(t+(j+1)\Delta t) - X(t+j\Delta t) - (\hat{a}_1 X(t+j\Delta t))\Delta t]^2\}$$

There is an algebraic relationship between the GSL and the Lagrangian. The coefficients/parameters in the Lagrangian correspond to the coefficients in the GSL, if

defined properly. Therefore, a good fit would be obtained if the estimated parameters of the Lagrangian were "close" to the coefficients used to generate the GSL. However, it must be stressed again that we are fitting a Lagrangian to the data and not merely attempting to match the coefficients. We will compare the theoretical Lagrangian (derived from the GSL) to the Lagrangian fit from the data.

3. **Results of the Fit**

In addition to the above example, two other fits to data for different GSL's were performed to truly test the simulated annealing computer code. The results are listed in Table 1. In defining the model we will use the terms linear or non-linear to describe the functional form of the drift and additive or multiplicative to describe the form of the diffusion or noise terms. As is evident in the table, the fitted coefficients were "close" to the generated coefficients. The true test however, was how well the generated Lagrangian and the fitted Lagrangian agreed.

TABLE 1

ONE ORDER PARAMETER RESULTS

Model	GSL	Generated Coefficients	Fitted Coefficients
I Linear Additive	$\dot{X} = aX + g\eta$	$a = -0.008$	$\hat{a}_1 = -0.0121649$
		$g = 0.2$	$\hat{a}_2 = 0.104862$
II Non-linear Additive	$\dot{X} = aX + bX^2 + g\eta$	$a = -0.008$	$\hat{a}_1 = -0.0385419$
		$b = -0.001$	$\hat{a}_2 = -0.000818763$
		$g = 0.2$	$\hat{a}_3 = 0.104391$
III Linear Multiplicative	$\dot{X} = aX + gX\eta$	$a = -0.008$	$\hat{a}_1 = -0.0121295$
		$g = 0.01$	$\hat{a}_2 = 0.00482471$

Before comparing, the Lagrangians had to be renormalized. Due to the underlying noise the Lagrangians could only be fit within an arbitrary constant. This constant was chosen so that the renormalized Lagrangians were unitless (this was arbitrary). A first choice was to use as a constant the Lagrangian evaluated at L_N ($\dot{X} = 0$, $X = X_r$) where X_r is the value of X which L is a global minimum. However, $L_N(0,X_r)$ will more than likely be close to zero and this could cause numerical and

mathematical difficulties. Therefore, we chose to evaluate $L_N = L(0,X_L)$ where $X_L = X_r + g^2(X_r)\sqrt{t}$. $g^2(X_r)$ is the variance evaluated at X_r. This seemed appropriate since we would like to incorporate into the normalization factor a measure of the curvature of the order parameter space and yet be "close" to the global minimum. Therefore

$$L_R = L(\dot{X},X)/L_N(0,X_L) . \tag{6.7}$$

is our renormalized Lagrangian. These are plotted in Figures 6.3, 6.4, and 6.5. It can be seen that the agreement in the drift terms measured by the location of the minimum, are very good. The agreement in the variance measured by the shallowness of the well was fairly good. This was for 20 runs of the simulation from time 0 to time 2 at increments of 0.1. The estimated coefficients were determined by having the simulated annealing program run for 10000 generated points and selecting the minimum value obtained. More work is being done here in running the program for 1,000,000 points and also performing the fit with more data, i.e. more runs and more time increments to see the effect of these modifications on the location of the minimum.

These simple examples were meant to illustrate the principles of the method and thus simple Lagrangians were obtained, i.e. those with only one minima. However, more complicated Lagrangians are admissible and this method is only limited by the ingenuity of the modeler.

C. TWO ORDER-PARAMETER EXAMPLE

The two order-parameter example is now given. The Lagrangian for this example involves a ratio of polynomials because of multiplicative noise. We again show the coefficients used to generate the data can be estimated quite well using the two order-parameter Lagrangian. This example tests our computer code, so that if the functions used in the Lagrangian are derived in a special way from the Langevin equations, then the short time conditional probability distribution will correspond directly to the probability distribution of the variables generated by the Langevin equation.

1. Data Collection and Order Parameters Used

The data are generated from a GSL of the form

$$\dot{X} = a_{11}Y + a_{12}XY + a_{13}\eta_1 + a_{14}Y\eta_2$$

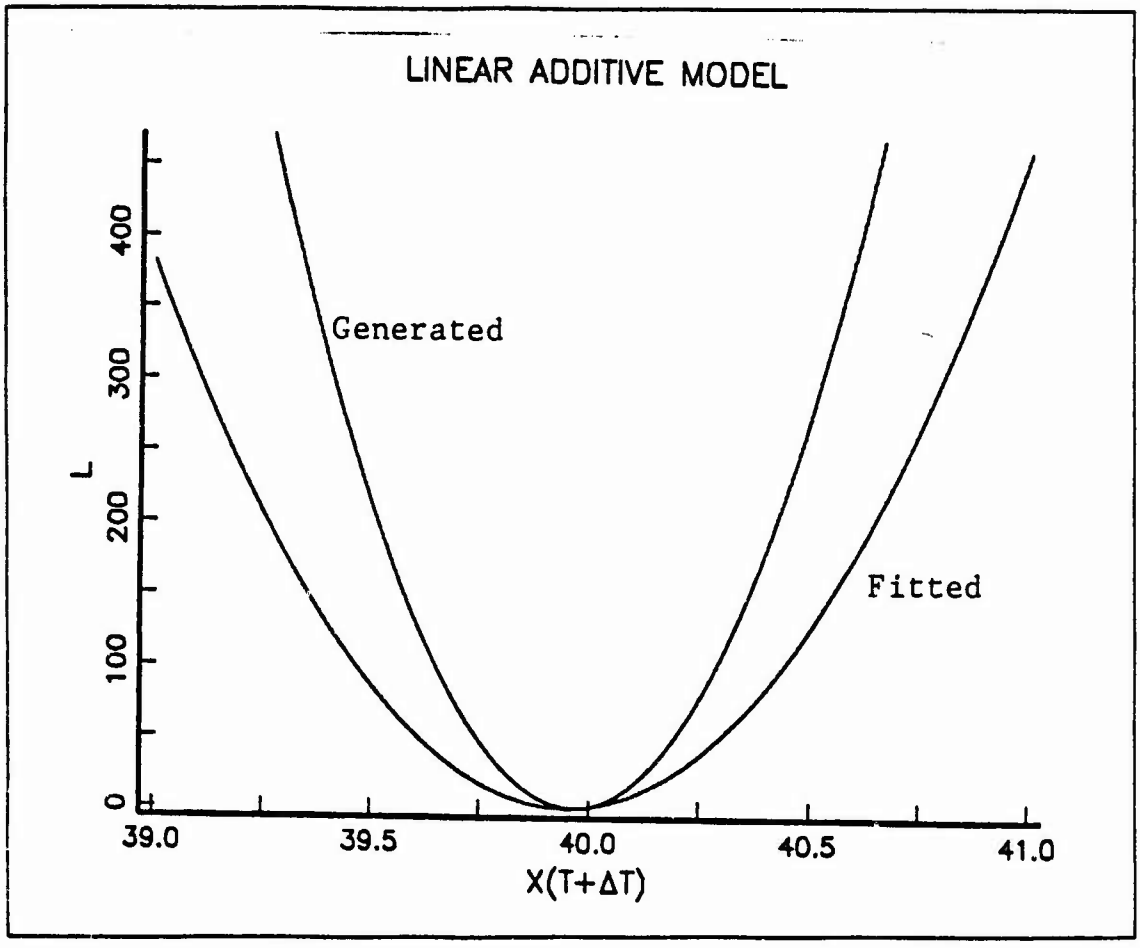


Figure 6.3 Generated and Fitted Lagrangians for Linear Additive Model.

$$\dot{Y} = a_{21}X + a_{22}XY + a_{23}X\eta_1 + a_{24}\eta_2 \ . \qquad (6.8)$$

where cross terms are present in the drift (mean) and the diffusion (variance). The multiplicative noise is present in the coefficients of the η's. The APL program LANCHESTER described in Appendix D was used to generate the data. Sample trajectories of X and Y are shown in Figures 6.6 and 6.7. Figures 6.8 and 6.9 plots the sample distributions.

We now derive the form of the Lagrangian corresponding to the GSL in equation 6.8 .

The theoretical Lagrangian, L, to be fit to the data is defined in equation 6.9.

$$L = 1/2(\dot{M}^{\mu} - g^{\mu})g_{\mu\nu}(\dot{M}^{\nu} - g^{\nu}) \ ,$$

$$g_{\mu\nu} = (g^{\mu\nu})^{-1} \ ,$$

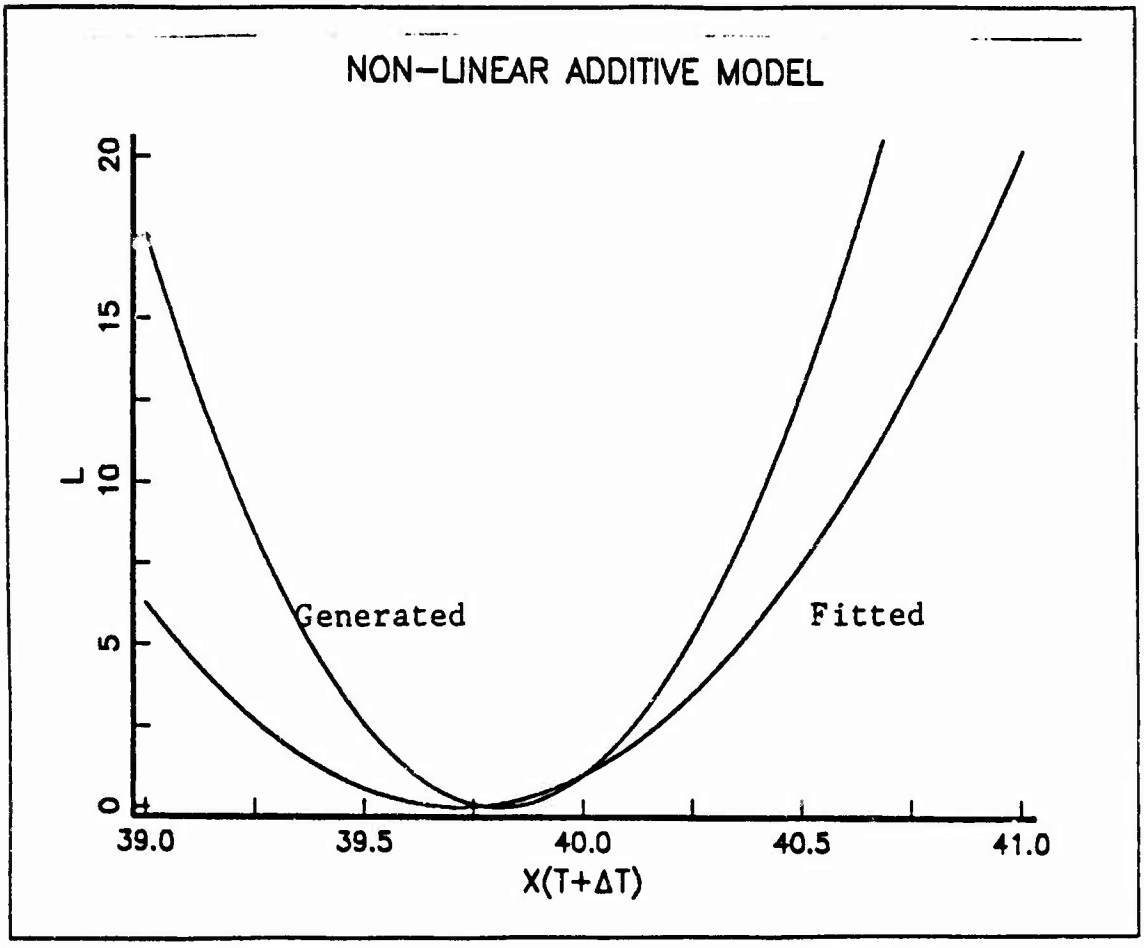


Figure 6.4 Generated and Fitted Lagrangians for Non-Linear Additive Model.

$$g^{\mu} = f^{\mu} + 1/2 g^{\nu}_{i}\, \partial g^{\mu}_{i} / \partial M^{\nu} \ ,$$

$$g^{\mu\nu} = g^{\mu}_{i}\, g^{\nu}_{i} \ . \qquad (6.9)$$

where we have assumed $V = 0$ and $f^{1} = a_{11}Y + a_{12}XY$. Therefore, g^{1} is given by equation 6.10 .

$$g^{1} = f^{1} + 1/2 g^{1}_{1} \frac{\partial g^{1}_{1}}{\partial X} + 1/2 g^{1}_{2} \frac{\partial g^{1}_{2}}{\partial X} + 1/2 g^{2}_{1} \frac{\partial g^{1}_{1}}{\partial Y} + 1/2 g^{2}_{2} \frac{\partial g^{i}_{2}}{\partial Y} \ , \qquad (6.10)$$

$$g^{1}_{1} = a_{13} \ , \ g^{1}_{2} = a_{14}Y \ , \ g^{2}_{1} = a_{23}X \ , \ g^{2}_{2} = a_{24} \ , \qquad (6.11)$$

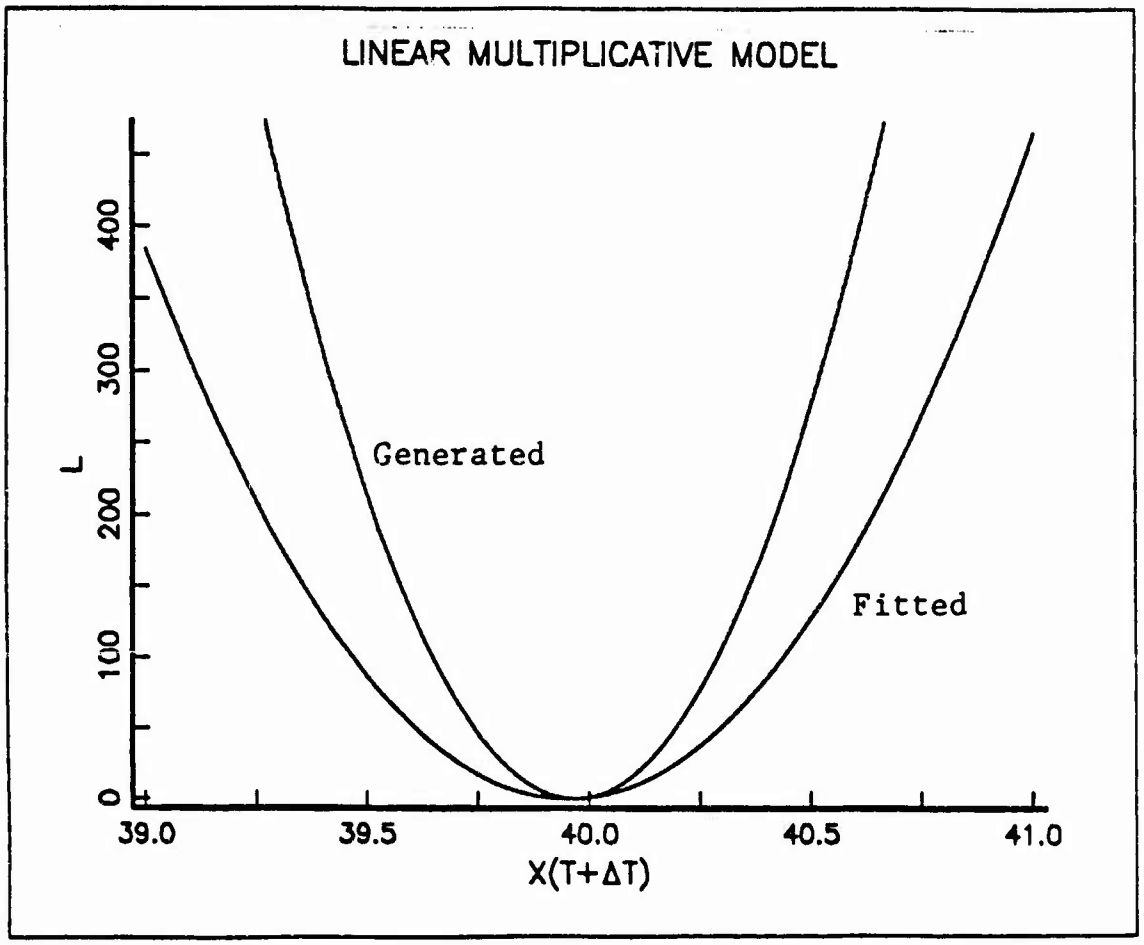


Figure 6.5 Generated and Fitted Lagrangians for Linear Multiplicative Model.

$$g^1 = a_{11}Y + a_{12}XY + 1/2a_{14}a_{24} \tag{6.12}$$

For g^2 we have

$$g^2 = f^2 + 1/2g^1{}_1 \frac{\partial g^2_1}{\partial X} + 1/2g^1{}_2 \frac{\partial g^2_2}{\partial X} + 1/2g^2_1 \frac{\partial g^2_1}{\partial Y} + 1/2g^2{}_2 \frac{\partial g^2_2}{\partial Y}\ , \tag{6.13}$$

and $f^2 = a_{21}X + a_{22}XY$. Therefore,

$$g^2 = a_{21}X + a_{22}XY + 1/2a_{13}a_{23} \tag{6.14}$$

Next we calculate the $g^{\mu\nu}$ terms.

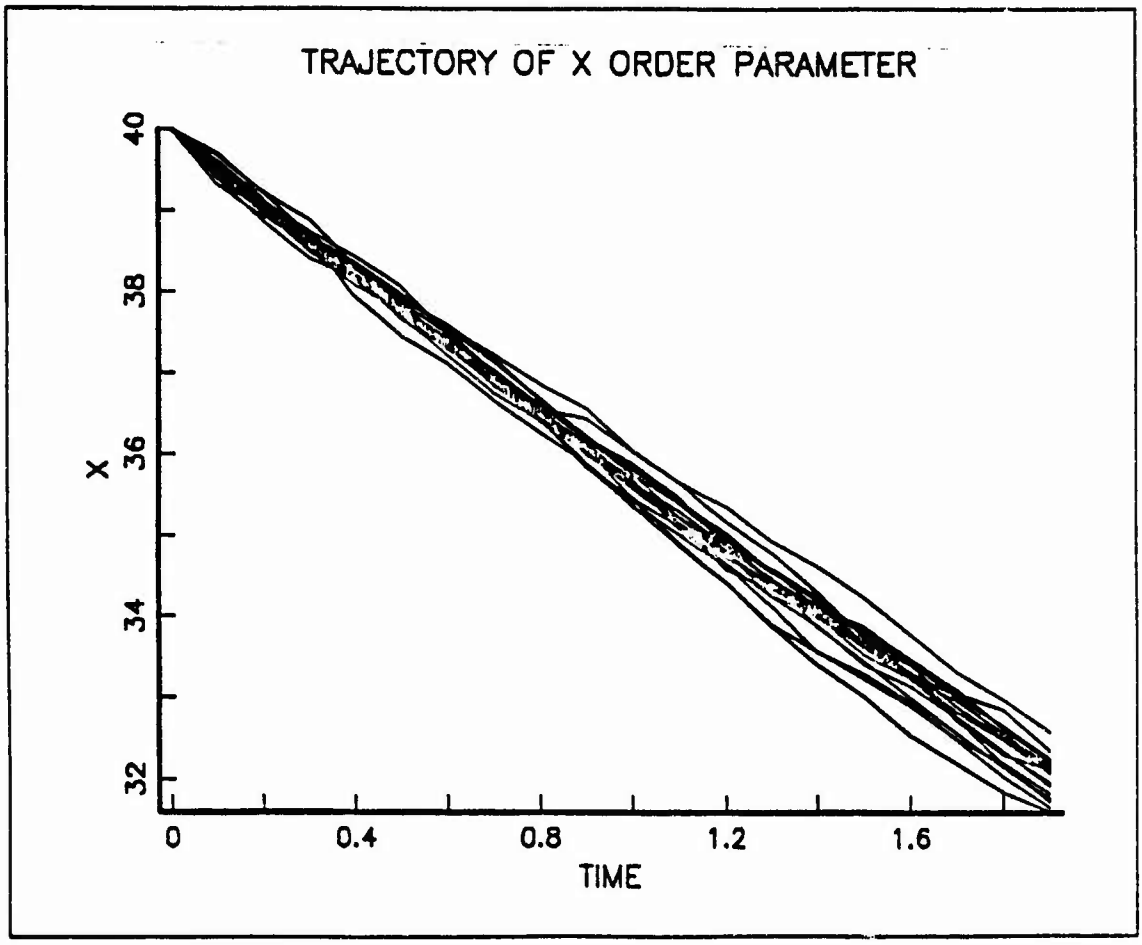


Figure 6.6 Trajectory of Order Parameter X.

$$g^{11} = g^1{}_1 g^1{}_1 + g^1{}_2 g^1{}_2 = a_{13}{}^2 + a_{14}{}^2 Y^2 \tag{6.15}$$

$$g^{12} = g^1{}_1 g^2{}_1 + g^1{}_2 g^2{}_2 = a_{13}a_{14}X + a_{14}a_{24}Y \tag{6.16}$$

$$g^{21} = g^2{}_1 g^1{}_1 + g^2{}_2 g^1{}_2 = g^{12} \tag{6.17}$$

$$g^{22} = g^2{}_1 g^2{}_1 + g^2{}_2 g^2{}_2 = a_{23}X^2 + a_{24}{}^2 \tag{6.18}$$

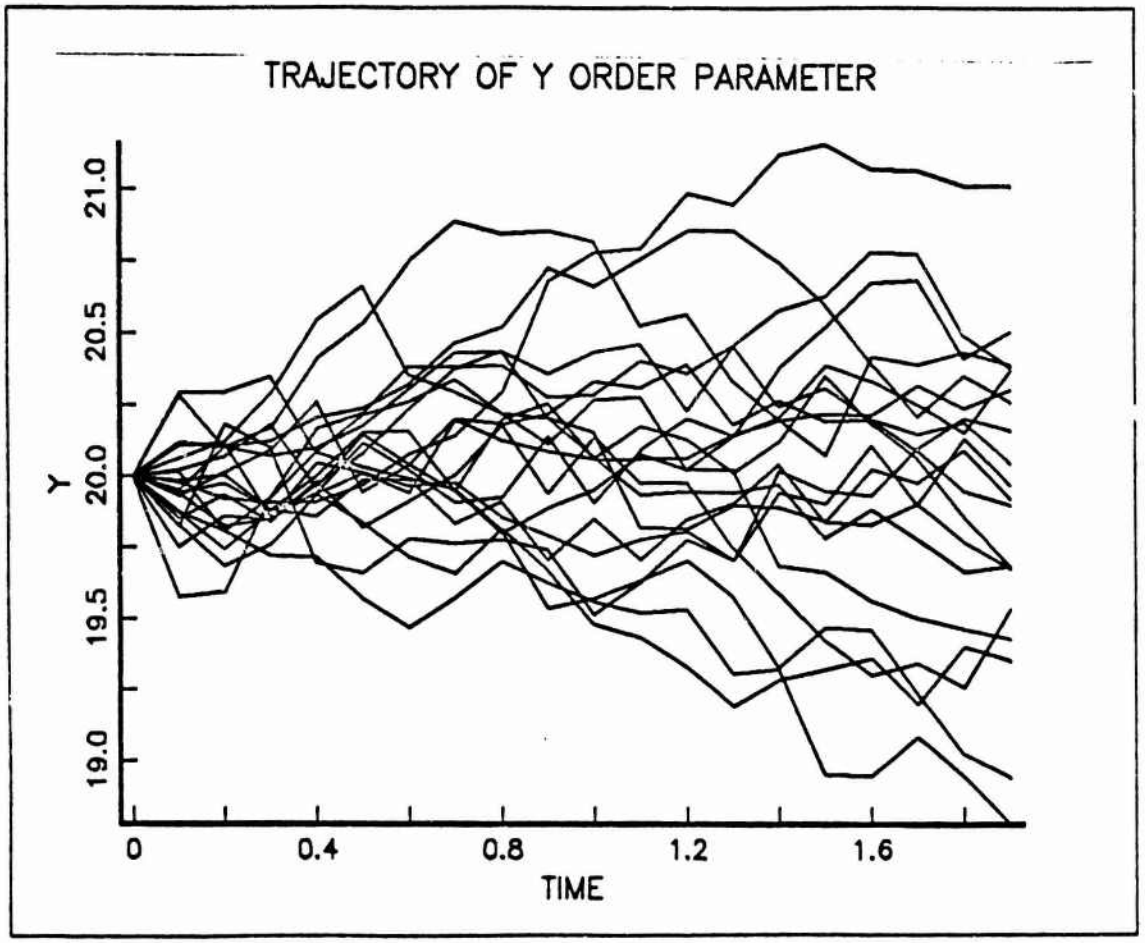


Figure 6.7 Trajectory of Order Parameter Y.

Next we need $g_{\mu\nu}$. The $g^{\mu\nu}$ form a 2 by 2 matrix whose inverse is $g_{\mu\nu}$. Thus

$$g_{\mu\nu} = (\det g^{\mu\nu})^{-1} \begin{pmatrix} g^{22} & -g^{12} \\ -g^{21} & g^{11} \end{pmatrix} , \tag{6.19}$$

where $\det g^{\mu\nu} = g^{11}g^{22} - g^{21}g^{12} = (\det g_{\mu\nu})^{-1}$. The Lagrangian is

$$L = 1/2(\dot{X}-g^1)^2 g_{11} + 1/2(\dot{X}-g^1)g_{12}(\dot{Y}-g^2) + 1/2(\dot{Y}-g^2)g_{21}(\dot{X}-g^1) + 1/2(\dot{Y}-g^2)^2 g_{22} \tag{6.20}$$

$$= 1/2(\det g^{\mu\nu})^{-1}[(\dot{X}-g^1)^2 g^{21} - 2(\dot{X}-g^1)(\dot{Y}-g^2)g^{12} + (\dot{Y}-g^2)^2 g^{11}] \tag{6.21}$$

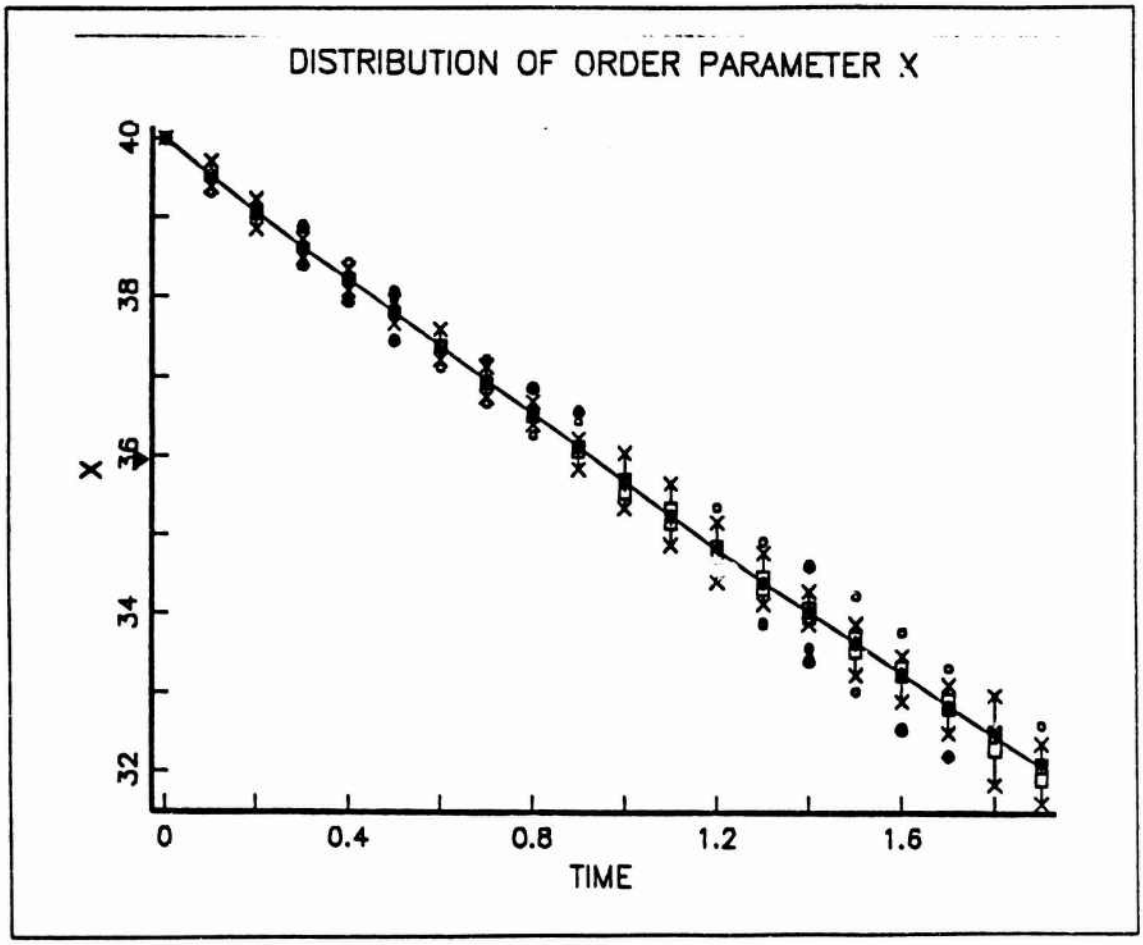


Figure 6.8 Distribution of Order Parameter X.

since all the terms contain the $(\det g_{\mu\nu})^{-1}$ factor and since $g^{12} = g^{21}$.

The short time conditional probability distribution $P^s_n \equiv P(X(t+\Delta t)|X(t+(n-1)\Delta t))$ is

$$P^s_n = g_n^{1/2} (2\pi\Delta t)^{-1/2} \exp(-L_n \Delta t) \tag{6.22}$$

where

$$g_n = \det (g_{\mu\nu})_n \ .$$

$$L_n = (1/2\Delta t)(M^{\mu}(t+(n+1)\Delta t) - M^{\mu}(t+n\Delta t) - g^1\Delta t) \times g_{\mu\nu}(M^{\nu}(t+(n+1)\Delta t) - M^{\nu}(t+n\Delta t) - g^2\Delta t) \tag{6.23}$$

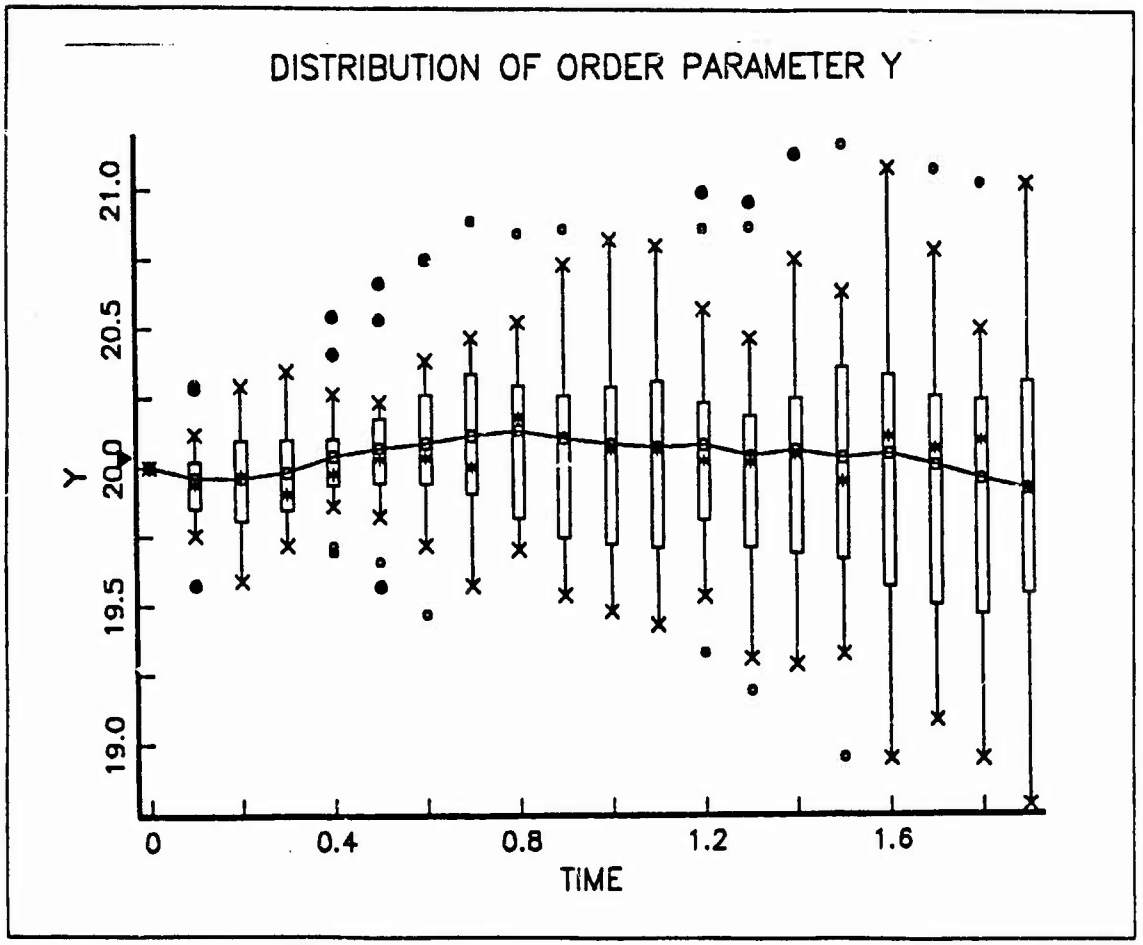


Figure 6.9 Distribution of Order Parameter Y.

and specifically,

$$L_n = 1/2\Delta t(g^{11}g^{22}-g^{12}g^{21})^{-1}\,[(X_n^{+} - X_n - g^1\Delta t)^2 g^{22}$$
$$-2g^{12}(X_n^{+} - X_n - g^1\Delta t)(Y_n^{+} - Y_n - g^2\Delta t) + (Y_n^{+} - Y_n - g^2\Delta t)^2 g^{11}]$$

where $X_n^{+} = X_n(t+(n+1)\Delta t)$, $X_n = X_n(t+n\Delta t)$, and where similar equations exist for Y.

2. Maximum Likelihood Fit of the Lagrangian

After a functional form of the Lagrangian has been derived (or guessed), the next step is to estimate the parameters in the Lagrangian. This is done using a maximum likelihood fit, that is, we wish to maximize

$$M = \prod_{i=1}^{I} \prod_{j=1}^{J} P^{s}_{ij}$$

where $P^{s}_{ij} = F^{s}(X_i(t+(j+1)\Delta t)|X_i(t + j\Delta t))$ is the short time conditional probability distribution for experiment i, at time $t+(j+1)\Delta t$ (post-point). As before, we will take the logarithm and pull the minus sign out front, and then minimize the resulting function. This becomes

$$\ln M = \sum \sum \ln P^{s}_{ij}$$

$$= \sum_{i} \sum_{j} \{1/2\ln (2\pi\Delta t) - 1/2 \ln g_{ij} + L_{ij}\Delta t\} = -N$$

We minimize N with respect to the parameters/coefficients of the Lagrangian form. This will be a minimization problem of 8 parameters for our example. N will likely contain many minima but we are only interested in the best fit, i.e. the global minimum. Therefore we will use the simulated annealing algorithm developed for this purpose of performing the best fit. Again we are looking for the best fit of the Lagrangian to the theoretical Lagrangian and not attempting to extract any particular parameters/coefficients. However, any constraints or conditions imposed on the parameters/coefficients based on phenomenological reasons should and must be included to obtain the best fit.

3. **Results of the Fit**

In addition to the above example, one other fit to data obtained from a different GSL is given to test the code and to examine the similarities between the Lagrangians. The results are given in Tables 2 and 3. Again we will normalize the Lagrangians. For the two dimensional case we will use $X_L = X_r + g^{XX}(X_r)$ where g^{XX} are the diagonal terms of the metric and are a measure of the curvature in the order parameter space. The renormalized Lagrangian becomes

$$L_R = L(\dot{X},X)/L_N(0,X_L) \quad . \tag{6.24}$$

The generated Lagrangians are plotted in Figures 6.10 and 6.12. The fitted Lagrangians are plotted in Figures 6.11 and 6.13. Again we find good agreement in the drift terms or the location of the minimum of the Lagrangian. These are located near

TABLE 2

TWO ORDER PARAMETER MODELS

Model	GSL
I Linear Additive	$\dot{X} = a_{11}Y + a_{12}\eta_1 + a_{13}\eta_2$ $\dot{Y} = a_{21}X + a_{22}\eta_1 + a_{23}\eta_2$
II Non-Linear Multiplicative	$\dot{X} = a_{11}Y + a_{12}XY + a_{13}\eta_1 + a_{14}Y\eta_2$ $\dot{Y} = a_{21}X + a_{22}XY + a_{23}X\eta_1 + a_{24}\eta_2$

TABLE 3

TWO ORDER PARAMETER MODEL RESULTS

Model	Generated Coefficients	Fitted Coefficients
I	$a_{11} = -0.008$	$\hat{a}_{11} = -0.0121839$
	$a_{12} = 0.3$	$\hat{a}_{12} = 0.0787236$
	$a_{13} = 0.1$	$\hat{a}_{13} = 0.130564$
	$a_{21} = -0.004$	$\hat{a}_{21} = -0.00655852$
	$a_{22} = 0.1$	$\hat{a}_{22} = 0.152455$
	$a_{23} = 0.3$	$\hat{a}_{23} = 0.0209012$
II	$a_{11} = 0.01$	$\hat{a}_{11} = 0.0550272$
	$a_{12} = -0.004$	$\hat{a}_{12} = -0.0073603$
	$a_{13} = 0.5$	$\hat{a}_{13} = 0.246917$
	$a_{14} = 0.01$	$\hat{a}_{14} = 0.00235102$
	$a_{21} = 0.02$	$\hat{a}_{21} = 0.0855577$
	$a_{22} = -0.001$	$\hat{a}_{22} = -0.00449816$
	$a_{23} = 0.015$	$\hat{a}_{23} = 0.00857855$
	$a_{24} = 0.7$	$\hat{a}_{24} = 0.281460$

the contour labeled 0.01 in the plots. The contour level spread is a measure of the variance and it is evident that there is more spread in the levels in both of the fitted Lagrangians. This is somewhat to be expected when dealing with a small sample from a probability distribution. More work is being done here to extend the results to more complicated Lagrangians, using more data, and using the simulated annealing program to find a better minimum (by including more points in the search). These early results have been encouraging.

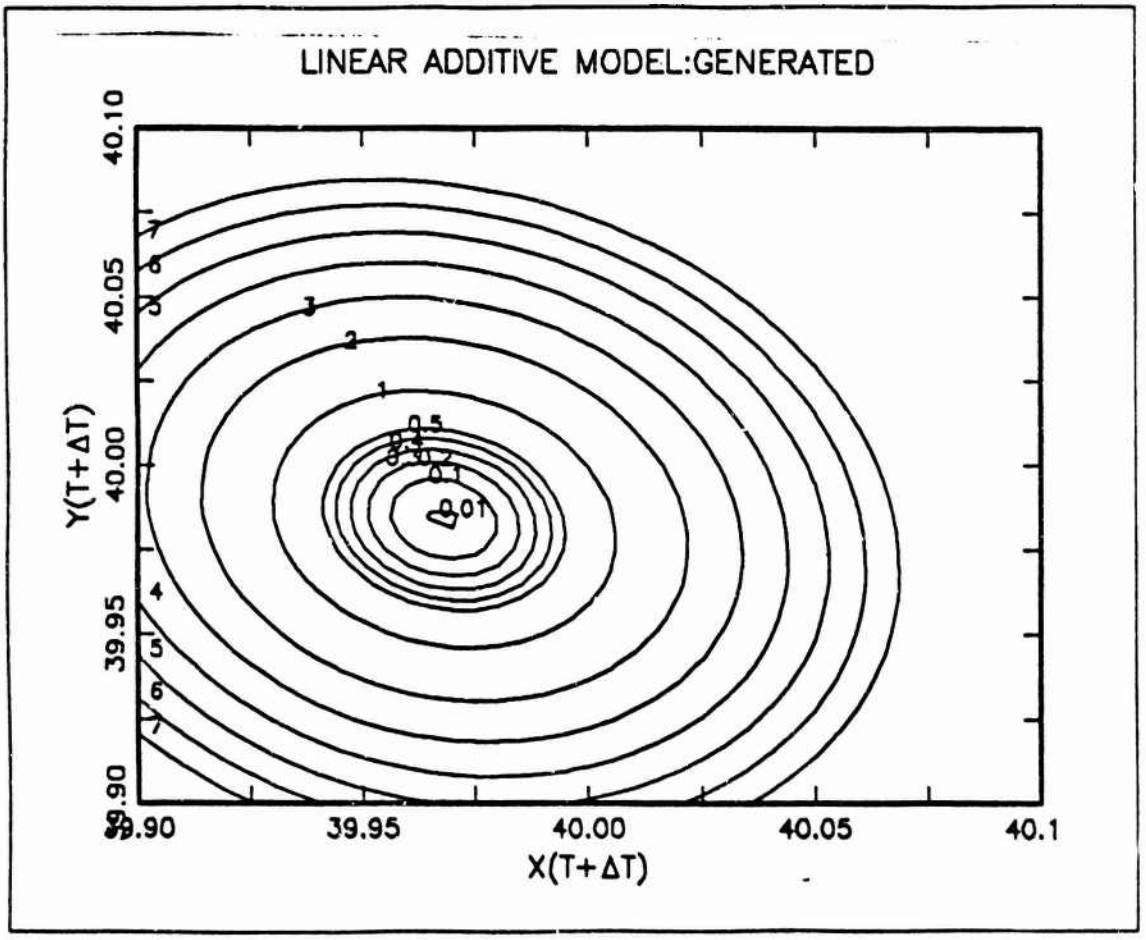


Figure 6.10 Generated Lagrangian for the Linear Additive Model.

D. TWO ORDER-PARAMETER MODEL USING JANUS DATA

1. Selection of Order Parameters

To ensure a simple description we will assume here that our order parameters are the numbers of personnel in each force and that we will look at the attrition, similar to a Lanchester approach.

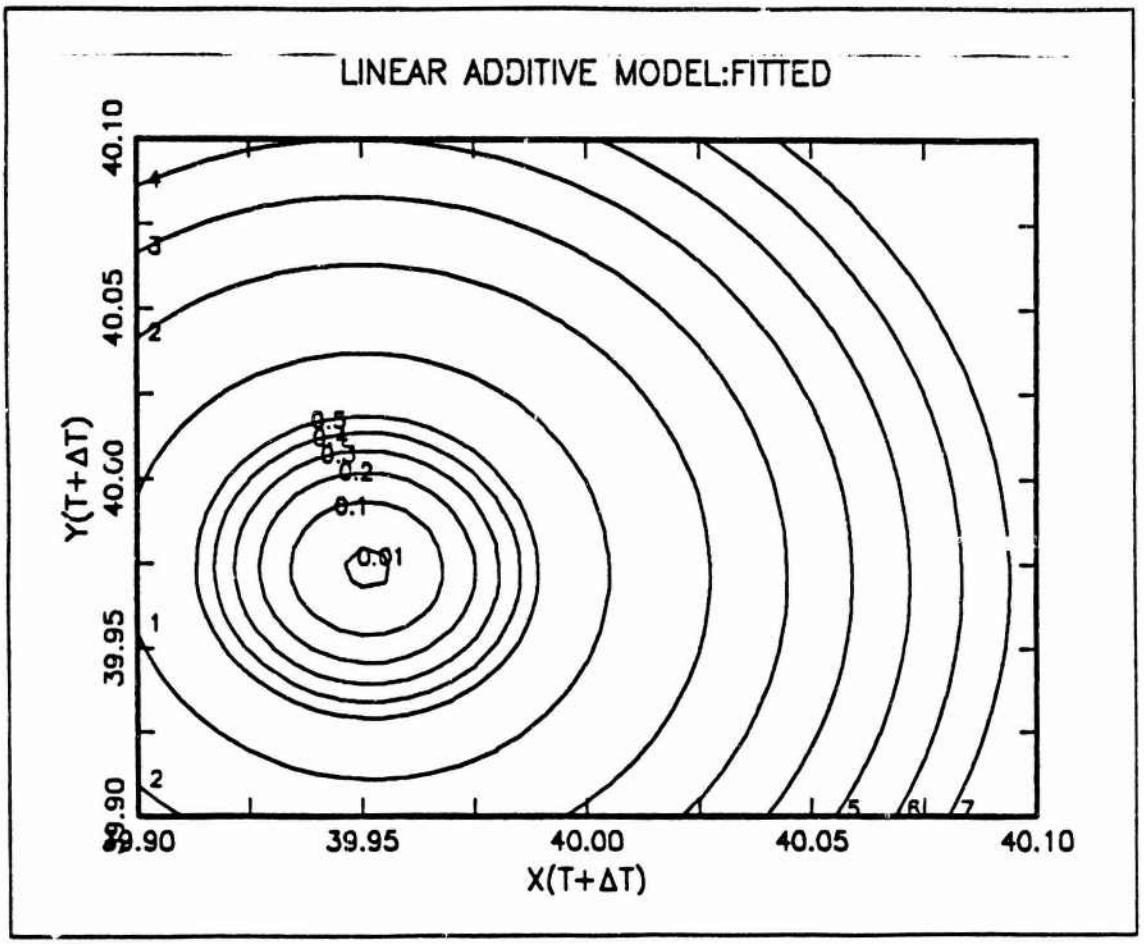


Figure 6.11 Fitted Lagrangian for the Linear Additive Model.

2. Data Collection

JANUS was chosen as a source of data because of its easy accessibility (present at NPS on a number of computers) and the combat data is in an easily reducible form for analysis. The high degree of graphics support also gives us a broad overview of the scenario we are dealing with. A Blue defense versus a Red Offense scenario was used. Terrain was fictitious and generated using the graphics capability of JANUS. It was flat with no foliage nor any built-up areas (cities). The defender had the capability to perform pop-up maneuvers, i. e. moving from full defilade to partial defilade. This was a separate condition from the terrain. The attacker moved across the terrain exposed. Blue was composed of 3 task forces of 12 M60A3 tanks and 4 M901 TOW weapons each and was in a line defense. Red was composed of 3 task forces of 30 T72 tanks and 9 BMP troop carriers each and was in the attack. Red

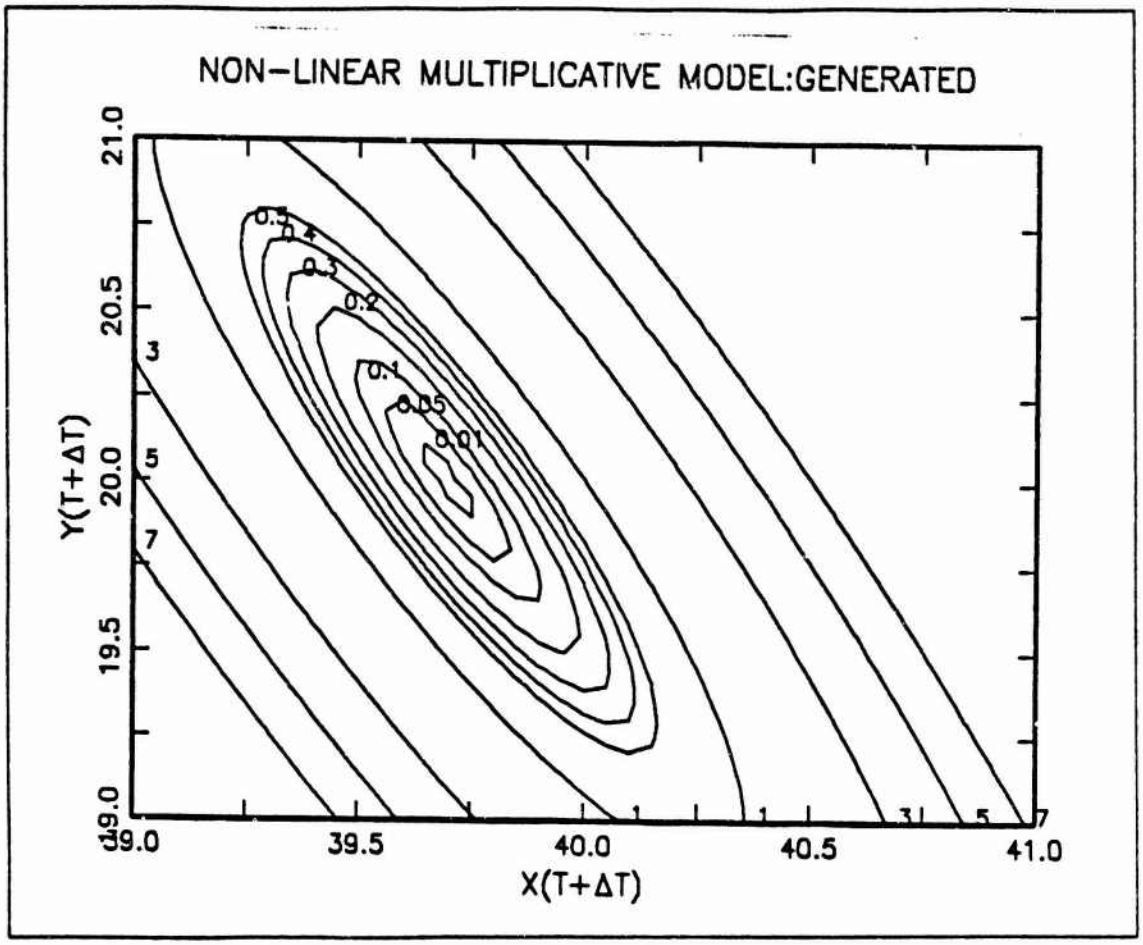


Figure 6.12 Generated Lagrangian for the Non-Linear Multiplicative Model.

objective was to penetrate Blue's defense, and Blue's objective was to repel Red's attack (mutually exclusive missions).

Twenty runs were collected using the batch mode of JANUS. Data collected was driven by the attrition process, and each attrition was classified as an event. At each event, clock time of the simulation and attrited side was recorded. Data consisted of a collection of clock times and status of forces at clock time.

3. Development of the Lagrangian

The first step in the development of the Lagrangian is to examine the sample paths of the data. This might suggest an appropriate model, such as Model I or Model II to begin with. It is probably best to begin with a simple model and move on to more highly non-linear models unles the data is known to be complicated. These functional forms can be as complicated as you wish, combining trigonometric,

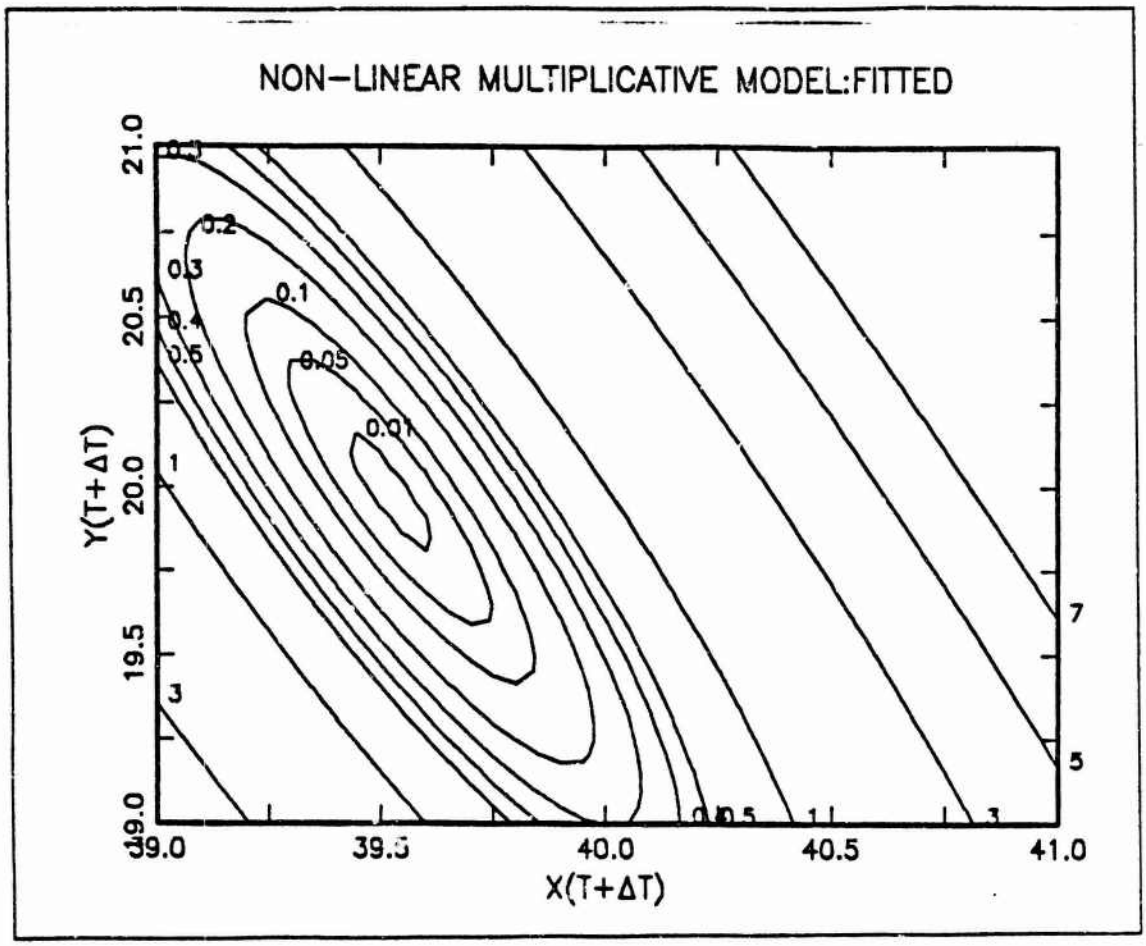


Figure 6.13 Fitted Lagrangian for the Non-Linear Multiplicative Model.

exponential, or polynomial terms. However, as mentioned before, a Padé approximant can approximate many functional forms and thus a ratio of polynomials would be your best first guess.

4. **Performing the Maximum Likelihood Fit**

Having selected a functional form we must now estimate the coefficients/parameters in the form. This is done by using a maximum likelihood fit as described in the two earlier examples and in Chapter 5. This requires the user, in order to use the simulated annealing program, to write a subroutine for his cost function, i.e. the function he wishes to maximize/minimize. This should be written in terms of a minimization for the program to work. The user must also decide how many points he wants to use in the search, starting temperatures, any constraints on the parameters/coefficients he wants to impose, and a starting point. These are relatively

easy to incorporate into the program. One rule of thumb that has been found is to start with a small number of points in order to check your subroutine and then increase to the amount of points you want. The minimum suggested number of points for one dimensional problem is 10000. This usually allows the simulated annealing algorithm sufficient time to find a good fit. Of course, the more dimensions/coefficients you have, the more points you will need in order to get a good fit. The minimum cost and the point associated with that cost is the final output. These are your fitted coefficients for your Lagrangian.

With the Lagrangian, we can now test the fit using similarly obtained data, i.e. data from the same scenario. If the fit does not seem satisfactory based on some pre-selected conditions, then we must return to the development step and try a different functional form. This iterative process continues until we are satisfied we have a good fit.

5. **Path Integral Representation**

With a satisfactory Lagrangian we can now examine the long term behavior of the system using recently developed code for the path integral (not available at the time of this thesis). By using the path integral code we can numerically determine the probability distribution of our order parameters at any time t. For example, an item of interest may be when a particular transition point in a battle might occur. We would use the path integral to calculate the probability distribution at each subsequent time step and then look for evolving trends of that distribution. For example, imagine we had a bistable probability distribution, i.e. one with two most likely states, and wanted to know how the system evolved from one minimum state to the other. We would then keep stepping through the calculation of the path integral until we found a noticeable change in the distribution.

6. **Sensitivity Analysis**

By using sensitivity analysis we can test the robustness of our fit and of our functional form. This might proceed as follows. First obtain sufficient data from some scenario you wish to study, e.g. our JANUS data. Our particular order parameters are the number of each vehicle on each side, so we have four order parameters. We decide on a functional form and perform the maximum likelihood fit. After having satisfied certain conditions, we are satisfied with our fit. We now return to our scenario and make a change in a microscopic variable, for example the pop-up rate of the tanks in the defense. Using the same functional form of the Lagrangian, we perform a

maximum likelihood fit using the newly obtained data. Remembering that we are fitting the Lagrangians and not the individual coefficients/parameters, we are not concerned with agreement here. We plot the Lagrangians for both the original data and the modified data and determine if there is any significant change. We might continue this process for differing values of the pop-up rate or turn to a different variable, e. g. the initial force level of each side, max range of the weapons, manuevering speed, change in the terrain, weather, etc. If the Lagrangian has not changed appreciably we can feel confident that our model is appropriate.

E. SUMMARY

In this chapter we presented three examples of the method using the one order-parameter and two order-parameter models. In each case we have assumed no explicit time dependence in the functional form. This was done to keep the examples simple. If the multiplicative noise is small compared to a constant noise term, then the Lagrangian is approximately quadratic in the post-point variable, and all distributions are Gaussian. However if the multiplicative noise is significant, if the mean is higher order than linear, then the long time distribution will be non-Gaussian even though the short time distribution is Gaussian. This must be emphasized. Otherwise the model would mimic a Gaussian distribution and be of limited use.

We have also presented the use of the simulated annealing algorithm to perform the maximum likelihood fit. A relatively new algorithm, simulated annealing seems to be particularly useful, especially when little is known about the system.

VII. CONCLUSIONS

A. INTRODUCTION

We have presented in this thesis an alternative to modeling combat which incorporates its severely stochastic nature. We have found it is a promising model for different types of combat if certain assumptions are satisfied. A relatively large combat system is necessary to satisfy this assumption. This should be of battalion size or larger for land battles and a carrier battle group or larger to describe the outer air battle of naval engagements.

A methodology has been developed which will allow the combat analyst to derive specific functional forms for use in the decision aid PIACA. The necessary mathematical theory has been developed and provides the foundation for the methodology. A simulated annealing algorithm to perform the maximum likelihood fits was developed. A FORTRAN program was written using the alogrithm and is in Appendix A. Three examples using the methodolgy and theory have been given with mixed results. These results are discussed next. Following that discussion, an outline for the development of the decision aid is given. Finally, the significance of the model is discussed.

B. RESULTS

The "truly-nonlinear" path integral method has been used sucessfully to describe such large scale systems as financial markets, the brain, and in nuclear physics [Ref. 8,17,18,19,20,21,22,23,24,25,26,27]. This is the first attempt to actually numerically calculate the nature of large scale combat using these methods [Ref. 1,2,12]. In the one order parameter example with constant variance, a Lagrangian was fit to the data with excellent results. In this case the path of the expected value of the order parameter followed the associated deterministic Lanchester equation which has been shown to model combat quite well under certain conditions. Other features of the one order parameter model was the incorporation of a drift and diffusion term of the order parameters. This is an improvement over the simple Lanchester approach.

The two order parameter example was the first look at incorporating multiplicative noise into the model. This is a significant improvement over the

Lanchester approach, stochastic or deterministic. Finally, using as our order parameters the level of the Blue and Red forces, a two order parameter model was developed for a JANUS simulation.

Given the time constraints of the NPS program, the purpose of this thesis was to lay the formulation for future theses to build upon. The Lagrangian we have formulated can be further investigated using this approach, e.g., using the simulated annealing program to fit a Lagrangian to a specific scenario.

C. DEVELOPMENT OF THE DECISION AID

We will now outline below the development of the decision aid PIACA and suggest some design requirements.

First, to be useful as a decision aid under severe time constraints, PIACA should be graphically and qualitatively oriented. The Lagrangian should be presented in a form similarly developed in Chapter 4, i.e. a graphical portrayal of most likely states and the risks associated with those states. This will allow easy assimilation by the commander and his staff. Of course, human factors should also be incorporated in this design.

Second, PIACA should enable the user to develop his own form of the Lagrangian by using data from simulations he has selected. This would allow for Lagrangians to be developed which are terrain, scenario, or commander dependent.

Third, once a Lagrangian is developed, the long time probability distribution, or path integral should be calculated easily, i.e. in real time. This is a heavy requirement since at present the path integral is not easy to calculate on supercomputers much less something which can be brought to the field.

To summarize, to have the full decision aid will require:

- The development of efficient algorithms to solve the path integral. This will most likely be a joint improvement in software and hardware.
- Graphical devices to portray the evolving long time probability distribution
- User-friendly software to interface with the user to develop Lagrangians of scenarios he has selected with the capability of pre-selecting Lagrangians from standard scenarios.
- Graphical depiction of alternatives. This is a must requirement for any decision aid.
- Ability to link to other users. This will then provide an easy information exchange in the meta-language of the model, i.e. passing values of the order parameters, scenarios, etc.

Obviously, there is much research to be completed in all these areas. One purpose of this thesis was to lay the groundwork for the development of the full decision aid as well as provide a unified means of describing combat using physical concepts.

D. SIGNIFICANCE OF THE MODEL

It has been shown that this model is extremely rich in applications. We have kept the examples simple to illustrate the use of the method. We have assumed only implicit time dependence and a free particle Lagrangian, i.e. $V=0$. The significance of the model could be increased with the addition of these terms. For example, it would then be possible to:

- incorporate boundary conditions, i.e. constraints on the forces either geographically or from higher levels of command, by the addition of the potential term, V. Such a model might be useful to describe relatively isolated combat.
- The possibility to examine "phase transitions", such as during nuclear, biological, and chemical exchanges which drastically alter the evolving battle. Other relatively small "phase transitions" could be examined such as when a commander has reached a critical point in his force level and how he reacts. These "phase transitions" could be modeled by matching at the critical point the two Lagrangians, i.e. one from the left and one from the right of the critical point.
- The full power of mathematical tools such as stability analysis and calculation of first passage times are available for examining the structure of the Lagrangian. These tools have been used by physicists and others to examine large and complex systems and many results could be applied to combat systems.

Obviously, there is much work to be done in this area that was beyond the scope of this thesis. The author hopes more work is done and this thesis layed the appropriate groundwork for others.

APPENDIX A
THE SIMULATED ANNEALING ALGORITHM

Simulated Annealing is a stochastic optimization algorithm for finding global extrema. First, the algorithm is described, and a stochastic model, specifically the Markov process, is used to show that the algorithm converges to the global optimum. Second, the algorithm is used on several unconstrained minimization problems.

1. BACKGROUND

We will begin this discussion by first asking the question, what is simulated annealing? In order to answer this question, we first need to know what is meant by annealing. Annealing is a process whereby a metal or crystal substance is first melted, then subsequently cooled to freezing temperature. At intermediate temperatures the substance is allowed to come to equilibrium. The purpose of an annealing experiment is to determine the ground or lowest energy state for that particular substance. The sequence of temperatures at which the substance is allowed to come to equilibrium is refered to as the annealing or "cooling" schedule. If this cooling schedule is too rapid, then the state of the substance will most likely be trapped in a metastable state and will not reach the ground state and it is said to have been "quenched".

Simulated Annealing is then a simulation of this process using a computer. We generate an initial configuration of the system and calculate its energy, E_0. Then the state of the system is changed, and the new energy, E_1, is calculated. If $\Delta E = E_1 - E_0 \leq 0$, then the state of the system becomes this new state. If $\Delta E > 0$, then the state of the system becomes the new state with probability $= e^{-\Delta E/T}$, where T is the temperature. Otherwise, it remains in the old state, and the process is repeated until $T=0$. In this case, we are interested in a cooling schedule which minimizes the total energy. The above procedure is refered to as the Metropolis algorithm after Metropolis, et. al. [Ref. 28] who developed it to perform equation of state calculations for substances at an equilibrium temperature. The algorithm is a useful one for performing simulation in statistical mechanics. In large physical systems, one is typically interested in the average properties of systems in equilibrium since these are the ones that are directly measureable. For example, the procedure has been used to simulate ferromagnetism of an Ising model [Ref. 29] and used to calculate $< U >$, the

average energy, and M, the magnetic moment. Also, the entropy of the amorphous magnetic state can be calculated using an algorithm similar to the Metropolis algorithm [Ref. 30].

Kirkpatrick, et. al. [Ref. 31] reformulated the algorithm to be used in the minimization of a general class of functions, analogous to the minimization of energy in the statistical mechanics system. They used the algorithm to calculate the optimal placement of integrated circuit chips on a computer circuit board, and also obtained a solution to the N-city traveling salesmen problem. Since that time, the simulated annealing algorithm has been used in various forms for problems in circuit design [Ref. 31], image reconstruction [Ref. 32,33,34], target tracking, [Ref. 35] layer assignment [Ref. 36], speech recognition [Ref. 37], and others [Ref. 38,39,40].

In section 2, we discuss the general class of problems associated with non-convex optimization. In section 3, the general simulated annealing algorithm is discussed with a description of the Markov chain model of simulated annealing. We show that if there exists a generation function, and an acceptance function that satisfy certain conditions which impose an irreducible, aperiodic Markov chain, then the algorithm will converge to the global optimum. In section 4, we give the results of an experiment which tested various combinations of generating and acceptance functions on the minimization of three different cost functions where the global optimum is known. Preliminary results of some modifications to the algorithm are also given. We conclude our results in section 5.

2. NON-CONVEX OPTIMIZATION (NCO)

Non-convex functions are functions which have multiple extrema. In NCO, we are typically interested in obtaining the global minimum or maximum. Algorithms which attempt to find these extrema can be classified as being either deterministic, stochastic or mixed in origin. Some deterministic algorithms such as quasi-Newton (BFGS) or steepest descent methods typically locate only local extrema and are only guaranteed to find the global when the function to be optimized is convex. There are other deterministic methods which have achieved good performance such as Levy and Montalvo [Ref. 41] for locating multiple extrema. Stochastic methods include the class of algorithms associated with simulated annealing, and which are purely stochastic in nature. Mixed algorithms are typically of the multiple starting point, iterative improvement variety. An example of this is the IMSL routine ZXMWD which uses a

quasi-Newton method with multiple starting points and selects the lowest value. As the number of starting points is increased, the probability the global minimum has been found is also increased.

The question now becomes, why use simulated annealing? Obviously, if there are algorithms which exploit the structure/properties of a class of problems, then these should be used. For example, there exist heuristics for the traveling salesmen problem which exploit the characteristics of the problem and probably would perform better than using simulated annealing. Simulated annealing becomes very useful if the problem you are interested in lacks any special structure or for which there are no heuristics available. Simulated annealing is easy to implement and gives good insight into the problem, if you can identify the cost function to be used. This will become more apparent when we describe the algorithm.

3. GENERIC SIMULATED ANNEALING ALGORITHM

The algorithm is very simple and is stated as follows [Ref. 42].

STEP 0

- Pick a starting point x_0. This could be at random or set to some initial guess. If you have some idea of the space to be sampled, the algorithm will run much quicker if it is confined to a smaller space.
- Set the initial temperature, T_0. Again if the sample space is confined, then T_0 can be set at a lower temperature. If not, then set T_0 to some high value. x_0 and T_0 are dependent upon the function to be minimized and are considered control parameters.
- Select a cooling schedule. This will be dependent upon g_T discussed in step 1 below. This is equivalent to setting $T(t) = f(t)$ where t is the step counter. For example, set $T(t) = T_0/(1+t)$ (inverse linear cooling).

STEP 1

- Pick a new point, x_1. This new point will be picked according to some generating function g_T which could be a function of the temperature, T. Some examples of generating functions are given in section D.

STEP 2

- Calculate $\Delta C \equiv C_1 - C_0$ where $C_i = f(x_i)$. This is where the dynamics of the cost function enter into the algorithm.
- Generate a uniform (0,1) random variable, U.

- If the acceptance function (discussed below) $a_T(\Delta C,T) > U$, then accept the point x_1 as the new state of the system and let $x_0 = x_1$, $C_0 = C_1$.
- If $C_0 < C_{min}$, $C_{min} = C_0$, $x_{min} = x_0$. (This keeps track of the lowest value obtained so far)
- Otherwise, keep x_0.
- Repeat step 1.

Another control parameter is the number of iterations/steps to complete. This is dependent upon the starting temperature and the number of variables in the function to be optimized. Generally speaking, if the sample space is confined to a small region, the number of steps needed will be reduced.

It is clear from the above, how we can model the algorithm as a Markov process and use those results to show global convergence. First, we consider the sample space as our Markov state space. We can consider this to be a finite state space if we take into account the resolution of our computer. Intuitively, it can be seen that if the generating function is capable of generating any point in the space, and the acceptance function has a finite probability of accepting the point, then every point could be visited/generated an inifinite number of times. Mathematically, this is equivalent to stating that our Markov chain is aperiodic and irreducible. For this to be true, the generating function must satisfy four conditions:

1. g_T be a bonafide probability distribution
2. $g_T(x_0, x_1) = g_T(x_1, x_0)$ symmetry. This condition is needed for aperiodicity of the Markov chain, i.e. that the generating function is symmetric and no cycles are present.
3. $g_T(x_0, x_1) > 0$ for all x_0,x_1 S (sample space). This could be generated as a path from x_0 to x_1. This says that every state can be reached by any other state, although not necessarily in one step.
4. lim $g_T(x_0,x_1)$ must exist.

The three conditions that the acceptance function must satisfy are:

1. $a_T(x_0,x_1)/a_T(x_1,x_0)$ be a monotone positive function.
2. If $C_0 < C_1$, then $a_T(C_0,C_1) > 0$ i.e. there must be a positive probability of accepting a decrease in the cost function.
3. lim $a_T(x_0,x_1)$ must exist.

If the aforementioned conditions hold, and if it can be shown that every state can be generated an infinite number of times, then the algorithm will converge to the global optimum. The preceding results are stated here for completeness of the paper. The results are proven in the literature [Ref. 42,43].

It can be seen from the above that the simulated annealing algorithm is actually a whole class of algorithms depending on your choice of the generating and acceptance functions. For example, for the Metropolis algorithm and many other algorithms in the literature, g_T is a uniform distribution over a neighborhood of fixed size. I.e. the next point generated is some fixed step distance from the previous point akin to a Random Walk process. A degenerate case would be to let g_T equal some constant, i.e. the uniform distribution over the whole space. This would be equivalent to a random sampling algorithm, keeping the state with the least cost. In our experiments, we use generating functions which sample the whole space with a bias towards the current point. This is equivalent to a random "hop" process, where the process can "hop" around the state space, and hops farther at higher temperatures than at lower temperatures. For example, suppose we have some function with multiple hills and valleys (extrema). Now suppose we have a ball with a lot of energy (high temperature) bouncing about this surface of hills and valleys. The ball loses energy according to some friction function, a_T, which is dependent on the energy of the ball. As the ball loses its energy, it will gradually fall into a valley corresponding to a minima. How the ball jumps around and how fast the ball loses energy will determine whether the valley reached is the lowest valley.

4. EXPERIMENTAL RESULTS

Experiments consisted of testing various combinations of generating and accepting functions on three cost functions. The purpose was to determine the most effective combination in terms of a measure of performance (MOP). One MOP that could be used and has intuitive appeal is CPU time. Another MOP is the acceptance ratio which is the ratio of the number of accepted points divided by the total number of generated points. These MOP's are directly related. This can be seen if you consider that the times to generate a point for each generating function are essentially equal. Since the number of accepted points is equal to the number of steps (step counter is incremented when a new point is accepted), and this was held constant, the only variable in the ratio is the number of points generated. Therefore, the only

difference in the runs was dependent on the number of points generated and so the acceptance ratio was used as a MOP. Obviously, this MOP could only be used effectively if the algorithm was actually able to locate the global minimum.

The cost functions used are as follows:

$C^1 = x^4 - 16x^2 + 5x$.

This has 2 minima at $x=2.90$, $C=-78.33$ (global) and $x=2.74$, $C=-50.06$ (local) [Ref. &szu2].

$C^2 = x^2 - 2y^2 - .3\cos(3\pi x) - .4\cos(4\pi y) + 0.7$.

This has multiple minima with global at $x=y=0.0$, $C=0.0$ [Ref. 39].

$C^3 = x^2 + 2y^2 - .3(\cos(3\pi x)\cos(4\pi y)) + 0.3$.

This has multiple minima with global at $x=y=0.0$, $C=0.0$ [Ref. 39].

The generating functions used are as follows:

$g_T^1 = (1/\pi)\, T/(T + x^2)$ Cauchy

$g_T^2 = 1/\sqrt{2\pi\sigma^2}\, e^{-x^2/2\sigma^2}$ Normal $(0, \sigma^2 = 1)$

$g_T^3 = 1/\sqrt{2\pi\sigma^2}\, e^{-x^2/2\sigma^2}$ Normal $(0, \sigma^2 = 4)$

The Cauchy was chosen because of its infinite variance (wide tails) which should indicate a good sampling of the space and not have a tendency to get trapped in a local minima early. The normals were chosen to see if the variance had any effect on trapping in local minimas. If it did, then that would indicate further research is needed to determine if the variance could be used as a control parameter, or if faster cooling could be used.

The acceptance functions used are as follows:

$h_T^1 = 1/(1 + e^{+\Delta C/T'})$ Heat Bath

$h_T^2 = e^{-\Delta C/T'}$ Boltzmann

The Boltzmann function is used in the Metropolis algorithm and is a fundamental function in statistical mechanics. This form arises quite naturally in our problem, as our cost function for the parameters is the Lagrangian of the probability

distribution of the independent variables. One modification to the above acceptance functions for better performance is to link T′, the control parameter, to the relative success of the acceptance function, instead of retaining the global temperature, T. That is, the temperature that the acceptance function uses would be a different temperature than the one tied to the number of steps or generation of points. We will call this the 2-Temperature generic simulated annealing algorithm. This was developed for this thesis in an attempt to obtain better results in higher dimensions of the parameter/coefficient space. At this time it is only an ad hoc procedure, but it does satisfy the above conditions for the acceptance function. The two temperatures allow for a small degree of independence in sampling the space. The global temperature, T, controls the sampling of the space by scaling the amount of the jumps in the space. The control temperature or acceptance temperature, T′, controls the amount of cost difference we are willing to accept at each step, and is subsequently connected to the state of the system since the state changes each time we accept a point. The FORTRAN program is contained in Figures A.1a-A.1h.

For each run, $T_0 = 5$, $x_0 = (10,10)$ (2Dim), and $T(t) = 1/(1 + t)$ (inverse linear cooling) was used. (For 1-dimensional problem, $x_0 = 10$). Table 4 gives results listed by generating function. Table 5 gives results listed by acceptance function.

TABLE 4
RESULTS BY GENERATING FUNCTION

Gen Fcn	Acc Fcn	Cost Fcn	Global Loc	Acc. Ratio
1	1	1	Y	.41
1	1	2	Y	.37
1	1	3	Y	.38
1	2	1	Y	.74
1	2	2	Y	.64
1	2	3	Y	.66
2	1	1	N	.48
2	1	2	N	.47
2	1	3	N	.48
2	2	1	N	.89
2	2	2	N	.85
2	2	3	N	.87
3	1	1	Y	.35
3	1	2	Y	.28
3	1	3	Y	.31
3	2	1	Y	.60
3	2	2	Y	.44
3	2	3	Y	.49

TABLE 5
RESULTS BY ACCEPTANCE FUNCTION

Acc Fcn	Gen Fcn	Cost Fcn	Global Loc	Acc. Ratio
1	1	1	Y	.41
1	1	2	Y	.37
1	1	3	Y	.38
1	2	1	N	.48
1	2	2	N	.47
1	2	3	N	.48
1	3	1	Y	.35
1	3	2	Y	.28
1	3	3	Y	.31
2	1	1	Y	.74
2	1	2	Y	.64
2	1	3	Y	.66
2	2	1	N	.89
2	2	2	N	.85
2	2	3	N	.87
2	3	1	Y	.60
2	3	2	Y	.44
2	3	3	Y	.49

First we compare the performance of the generating functions. We notice that g_T^2 did not converge to global optimum in any of the runs, so we discard it. For g_T^1 and g_T^3 we note that for every run, g_T^1 outperformed g_T^3. Obviously, this is not conclusive since it is based on a small number of runs. Many different runs with different starting points and initial temperatures would be needed for more definitive conclusions. As to the acceptance functions, we can see that h_T^2 outperformed h_T^1 in every case. Therefore, based on these results, it would seem that the best combination is to use the Cauchy as the generating function, and the Boltzmann as the acceptance function.

5. CONCLUSIONS

This experiment was intended to introduce the reader to simulated annealing and to show how it can be used for optimization problems. Further research is needed in the area of different generating and acceptance functions, applying the algorithm to different cost functions, and extending it to higher dimensions. Preliminary results with a normal generating function, with variance as a function of temperature, indicate there is more interesting work to be done in this area.

6. FORTRAN PROGRAM

Contained in Figures A.1a-A.1h below, is the 2-Temperature version of the Simulated Annealing Algorithm FORTRAN program code. The subroutine COSTFN is for the 2OP model. User must input the number of dimensions (corresponding to the number of parameters/coefficients to be fit and NOT to the number of order parameters), number of steps or total number of accepted points he wants generated, the starting temperature for the generating function, the starting temperature for the acceptance function, the starting point, number of accepted points to print, total number of generated points (controling the amount of time the program will run), the time increment of the data, the number of runs of the data, and the number of time increments. This information must be included as the first line of the data and can be unformatted.

```
      PROGRAM SA
*************************************************************
*THIS PROGRAM IS A TWO TEMPERATURE VERSION OF SA
*************************************************************
      INTEGER NDIM,ACC,NACC,NTOT,INACC,INTOT,STEPS,TI,T2
      DOUBLE PRECISION  T0,T0A,X0(8),X1(8),XMIN(8),
     1TEMPG,TEMPA,C1,C0,DELTAC,H,CMIN,
     1PERACC,IPERAC,Z1(1000,21),Z2(1000,21),DELTAT
      COMMON /COMI/ NDIM,ACC,NACC,NTOT,INACC,INTOT,STEPS,TI,T2
      COMMON /COMR/ C0,C1,TEMPG,TEMPA,DELTAC,H,PERACC,IPERAC,T0A,
     1X0,X1,XMIN,T0,CMIN
      COMMON /LAG/ Z1,Z2,DELTAT,NUMRUN,NTIME
      COMMON /A1/ NMOD
      COMMON /SA1/ MAXTOT
      READ (2,*)NDIM,STEPS,T0,T0A,(X0(N),N=1,8),NMOD,MAXTOT,DELTAT,
     1NUMRUN,NTIME
      DO 5 I=1,NUMRUN
 5    READ (2,*) (Z1(I,N),N=1,NTIME+1),(Z2(I,N),N=1,NTIME+1)
C     WRITE (3,*) (Z1(I,N),N=1,NTIME+1)
C5    WRITE (3,*) (Z2(I,N),N=1,NTIME+1)
      WRITE (3,*) 'NDIM  STEPS  T0  T0A  STARTING POINTS  NMOD  MAXTOT
     1  DELTAT    NUMRUN  NTIME'
      WRITE(3,*)NDIM,STEPS,T0,T0A,(X0(N),N=1,8),NMOD,MAXTOT,DELTAT,
     1NUMRUN,NTIME
      DO 23 JN= 1,NDIM
          XMIN(JN) = X0(JN)
 23    X1(JN) = X0(JN)
      CALL SIMANN
      STOP
      END
```

Figure A.1a FORTRAN Program for 2-Temperature Simulated Annealing Algorithm.

```
*****************************************************************
        SUBROUTINE SIMANN
        INTEGER NDIM,ACC,NACC,NTOT,INACC,INTOT,STEPS,TI,T2
        DOUBLE PRECISION  TO,TOA,X0(8),X1(8),XMIN(8),
       1TEMPG,TEMPA,C1,C0,DELTAC,H,CMIN,
       1PERACC,IPERAC,Z1(100,21),Z2(100,21),DELTAT
        COMMON /COMI/ NDIM,ACC,NACC,NTOT,INACC,INTOT,STEPS,TI,T2
        COMMON /COMR/ C0,C1,TEMPG,TEMPA,DELTAC,H,PERACC,IPERAC,TOA,
       1X0,X1,XMIN,T0,CMIN
        COMMON /SA1/ MAXTOT
        CALL COSTFN(X0,C0)
        CALL INIT
        DO 10 TI = 1, STEPS
        TEMPG = T0/(DFLOAT(TI))
 20   IF (NTOT.GE.MAXTOT) GOTO 11
        CALL GT
        CALL COSTFN (X1,C1)
        CALL HT
        CALL PICKPT
        IF (ACC.EQ.1) THEN
              CALL ACCEPT
        ELSE
              GO TO 20
        ENDIF
 10   CONTINUE
 11   PERACC = NACC/REAL(NTOT)
        WRITE (3,100) NTOT,NACC,PERACC,CMIN,(XMIN(NN),NN=1,NDIM)
 100  FORMAT (1X,'TOTAL GENERATED ',I8,3X,'NUMBER ACCEPTED ',I8,3X,
      1'PERCENTAGE ACCEPTED ',F4.2,3X//1X,'MIN COST ',E12.6,3X,
      1/1X,'MIN POINT ',4(E12.6,1X)/1X,4(E12.6,1X))
        WRITE (3,*) 'TEMPG',TEMPG,'TEMPA',TEMPA
        RETURN
        END
```

Figure A.1b FORTRAN Program for 2-Temperature Simulated Annealing Algorithm (cont).

```
*****************************************************************
      SUBROUTINE COSTFN(P,COST)
      DOUBLE PRECISION P(NDIM),COST,X(1000,21),Y(1000,21)
     1,L1,L2,L3,LD,LX,LY,DELTAX,DELTAY
      INTEGER NDIM,ACC,NACC,NTOT,INACC,INTOT,STEPS,TI,T2
      DOUBLE PRECISION  T0,T0A,X0(8),X1(8),XMIN(8),
     1TEMPG,TEMPA,C1,C0,DELTAC,H,CMIN,
     1PERACC,IPERAC,Z1(100,21),Z2(100,21),DELTAT
      COMMON /COMI/ NDIM,ACC,NACC,NTOT,INACC,INTOT,STEPS,TI,T2
      COMMON /COMR/ C0,C1,TEMPG,TEMPA,DELTAC,H,PERACC,IPERAC,T0A,
     1X0,X1,XMIN,T0,CMIN
      COMMON /LAG/ X,Y,DELTAT,NUMRUN,NTIME
      COST = 0.0
      DO 10 I = 1,NUMRUN
      DO 20 N = 1,NTIME
      DELTAX = X(I,N+1)-X(I,N)
      DELTAY = Y(I,N+1)-Y(I,N)
      H1=P(1)*Y(I,N)+P(2)*X(I,N)*Y(I,N)+P(4)*P(8)/2.0
      G11=P(3)
      G12=P(4)*Y(I,N)
      H2=P(5)*X(I,N)+P(6)*X(I,N)*Y(I,N)+P(3)*P(7)/2.0
      G21=P(7)*X(I,N)
      G22=P(8)
      Q11=G11**2+G12**2
      Q12=G11*G21+G12*G22
      Q22=G21**2+G22**2
      DEN=Q11*Q22-Q12**2
      IF (DEN.LE.0.0) THEN
        COST=1.0E25
        GO TO 99
      ENDIF
      DETG=1.0/DEN
      LX=DELTAX-H1*DELTAT
      LY=DELTAY-H2*DELTAT
      L1=(LX**2)*Q22/2.0
      L2=-1.D*(LX*LY)*Q12
      L3=(LY**2)*Q11/2.0
      LD=DETG*(L1+L2+L3)/DELTAT
C     IF (DEN.LE.0.0) THEN
C     ENDIF
      TERM1=-1.0*DLOG(DETG/DELTAT)/2.0
      COST=COST+TERM1+LD
C       WRITE (3,*) 'A',P(1),'B',P(2),'C',P(3),'D',P(4)
C       WRITE (3,*) 'E',P(5),'F',P(6),'G',P(7),'H',P(8)
C       WRITE (3,*) 'EXPERIMENT NUMBER',I,'TIME STEP',N
C       WRITE (3,*) 'DEN OF DETG',DEN,'Q11',Q11,'Q12',Q12,'Q22',Q22
C       WRITE (3,*) 'LX',LX,'H1',H1,'G11',G11,'G12',G12
C       WRITE (3,*) 'LY',LY,'H2',H2,'G21',G21,'G22',G22
C       WRITE (3,*) 'L1',L1,'L2',L2,'L3',L3
C       WRITE (3,*) 'LD',LD,'TERM1',TERM1,'COST',COST
  20   CONTINUE
  10   CONTINUE
C     WRITE (3,*) NTOT,'GENERATED POINT ',(P(I),I=1,NDIM),COST
  99   RETURN
       END
```

Figure A.1c FORTRAN Program for 2-Temperature Simulated Annealing Algorithm (cont).

```
*****************************************************************
      SUBROUTINE INIT
      INTEGER NDIM,ACC,NACC,NTOT,INACC,INTOT,STEPS,TI,T2
      DOUBLE PRECISION  TO,TOA,X0(8),X1(8),XMIN(8),
     1TEMPG,TEMPA,C1,C0,DELTAC,H,CMIN,
     1PERACC,IPERAC,Z1(100,21),Z2(100,21),DELTAT
      COMMON /COMI/ NDIM,ACC,NACC,NTOT,INACC,INTOT,STEPS,TI,T2
      COMMON /COMR/ C0,C1,TEMPG,TEMPA,DELTAC,H,PERACC,IPERAC,TOA,
     1X0,X1,XMIN,T0,CMIN
      COMMON /A1/ NMOD
      INACC = 0
      INTOT = 0
      NACC = 0
      NTOT = 0
      CMIN = C0
      TEMPA=TOA
      T2=0
      RETURN
      END
```

Figure A.1d FORTRAN Program for 2-Temperature Simulated Annealing Algorithm (cont).

```
*****************************************************************
      SUBROUTINE GT
      INTEGER NDIM,ACC,NACC,NTOT,INACC,INTOT,STEPS,TI,T2
      DOUBLE PRECISION  TO,TOA,X0(8),X1(8),XMIN(8),
     1TEMPG,TEMPA,C1,C0,DELTAC,H,CMIN,
     1PERACC,IPERAC,Z1(100,21),Z2(100,21),DELTAT
      COMMON /COMI/ NDIM,ACC,NACC,NTOT,INACC,INTOT,STEPS,TI,T2
      COMMON /COMR/ C0,C1,TEMPG,TEMPA,DELTAC,H,PERACC,IPERAC,TOA,
     1X0,X1,XMIN,T0,CMIN
      DOUBLE PRECISION U,HOP
      REAL WK(3)
      DOUBLE PRECISION DSEED/5959/
      DO 10 I = 1,NDIM
 5    CALL GGCAY(DSEED,1,WK,U)
      HOP=TEMPG*U
      X1(I) = X0(I) + HOP
      IF (X1(I).GT.2.0.OR.X1(I).LT.-2.0) GO TO 5
      IF (X1(3).LT.0.0.OR.X1(4).LT.0.0.OR.X1(7).LT.0.0.OR.X1(8).LT.0.0)
     1GO TO 5
C     WRITE (3,100) HOP,I,X1(I),X0(I),U
C100  FORMAT(1X,'AMT OF HOP',5X,E12.6,5X,'I',2X,I2/1X,'NEWPT',5X,E12.6,
C    15X,'OLDPT',5X,E12.6,1X,'CAUCHY RANDOM NO',E12.6)
 10   CONTINUE
      RETURN
      END
```

Figure A.1e FORTRAN Program for 2-Temperature Simulated Annealing Algorithm (cont).

```
**********************************************************************
      SUBROUTINE HT
      INTEGER NDIM,ACC,NACC,NTOT,INACC,INTOT,STEPS,TI,T2
      DOUBLE PRECISION  TO,TOA,XO(8),X1(8),XMIN(8),
     1TEMPG,TEMPA,C1,CO,DELTAC,H,CMIN,
     1PERACC,IPERAC,Z1(100,21),Z2(100,21),DELTAT
      COMMON /COMI/ NDIM,ACC,NACC,NTOT,INACC,INTOT,STEPS,TI,T2
      COMMON /COMR/ CO,C1,TEMPG,TEMPA,DELTAC,H,PERACC,IPERAC,TOA,
     1XO,X1,XMIN,TO,CMIN
      REAL*8 Y1
      DELTAC = C1 - CO
      Y1 = DELTAC/TEMPA
      IF (Y1.GT. 20.0) THEN
          H = 0.0
          GO TO 99
      ENDIF
      IF (Y1.LT.-15.0) THEN
          H = 1.0
          GO TO 99
      ENDIF
      H = DEXP (-Y1)
C99   WRITE (3,*) 'DELTAC',DELTAC,'TEMPA',TEMPA,'TEMPG',TEMPG,
C     1'Y1',Y1,'H',H,'NTOT',NTOT
 99   RETURN
      END
```

Figure A.1f FORTRAN Program for 2-Temperature Simulated Annealing Algorithm (cont).

```
**********************************************************************
      SUBROUTINE PICKPT
      INTEGER NDIM,ACC,NACC,NTOT,INACC,INTOT,STEPS,TI,T2
      DOUBLE PRECISION  TO,TOA,XO(8),X1(8),XMIN(8),
     1TEMPG,TEMPA,C1,CO,DELTAC,H,CMIN,
     1PERACC,IPERAC,Z1(100,21),Z2(100,21),DELTAT
      COMMON /COMI/ NDIM,ACC,NACC,NTOT,INACC,INTOT,STEPS,TI,T2
      COMMON /COMR/ CO,C1,TEMPG,TEMPA,DELTAC,H,PERACC,IPERAC,TOA,
     1XO,X1,XMIN,TO,CMIN
      REAL U
      DOUBLE PRECISION DSEED/6969/
      CALL GGUBS(DSEED,1,U)
      ACC = 0
      IF (H.GT.U) ACC = 1
      NTOT = NTOT + 1
      INTOT = INTOT + 1
      RETURN
      END
```

Figure A.1g FORTRAN Program for 2-Temperature Simulated Annealing Algorithm (cont).

```
*****************************************************************
        SUBROUTINE ACCEPT
        INTEGER NDIM,ACC,NACC,NTOT,INACC,INTOT,STEPS,TI,T2
        DOUBLE PRECISION  T0,T0A,X0(8),X1(8),XMIN(8),
       1TEMPG,TEMPA,C1,C0,DELTAC,H,CMIN,
       1PERACC,IPERAC,Z1(100,21),Z2(100,21),DELTAT
        COMMON /COMI/ NDIM,ACC,NACC,NTOT,INACC,INTOT,STEPS,TI,T2
        COMMON /COMR/ C0,C1,TEMPG,TEMPA,DELTAC,H,PERACC,IPERAC,T0A,
       1X0,X1,XMIN,T0,CMIN
        COMMON /A1/ NMOD
        DO 15 J = 1,NDIM
 15      X0(J) = X1(J)
        C0 = C1
        NACC = NACC + 1
        INACC = INACC + 1
        IPERAC = REAL(INACC)/REAL(INTOT)
        IF(IPERAC.GT.0.5) THEN
             T2 = T2 + 1
             TEMPA = T0A/(REAL(T2))
        ENDIF
        IF (MOD(NACC,NMOD).EQ.0) THEN
          WRITE (3,*) 'ACCEPT A POINT'
          WRITE (3,101) INTOT,INACC,IPERAC,NTOT,NACC,
       1TEMPG,TEMPA,(X1(N),N=1,NDIM),C1
 101   FORMAT (1X,I2,1X,I2,1X,F4.2,1X,I7,1X,I7,3X,2(E9.3,3X)/
       11X,4(E9.3,2X)/1X,4(E9.3,2X),E12.6)
        ENDIF
        IF (C0.LT.CMIN) THEN
             CMIN = C0
             DO 17 JJ=1,NDIM
 17           XMIN(JJ) = X1(JJ)
        ENDIF
        INACC = 0
        INTOT = 0
        RETURN
        END
```

Figure A.1h FORTRAN Program for 2-Temperature Simulated Annealing Algorithm (cont).

APPENDIX B
THE PATH INTEGRAL

1. INTUITIVE DESCRIPTION

a. Sum over Paths

We begin by using our previous definition of the path integral, and interpreting the integral as a specific sum over paths. The path integral is defined as

$$P(X(t)|X(t_0) = \int \cdots \int \underline{D}X \exp(-\sum_{n=0}^{s} \Delta t L_n) \quad ,$$

$$\underline{D}X = (2\pi g_0{}^2 \Delta t)^{-1/2} \prod_{n=1}^{s} (2\pi g_n{}^2 \Delta t)^{-1/2} \, dX_n$$

which is a long time conditional probability distribution of a variable X at some time t, given its intial position at t_0 . How is this a sum over paths? We will look at the one dimensional case but the intuitive extension to higher dimensions can be easily made.

To easily see the path integral description we first look at the random walk problem. Suppose we have had a lot of drinks at Tun Tavern (a famous Marine Corps establishment of the late 1700's). Although we do not want to leave, we have an inspection tommorrow and need to get home. Tun Tavern is located as X_0 and home is X(t). Being a true drunk, we would have a 50-50 chance of stepping out a certain distance in a time increment Δt. There will be associated with our walk a probability of never reaching home (depending on how many drinks we have had!) and thus there is a probability distribution of getting home by time t. Suppose we did this every night, then each night we would follow a different path home. This is an example of Brownian motion. The short time conditional distribution, i.e. the probability of being across the street at (z) at time $t + \Delta t$ given you were at Tun Tavern (x) is
P(across the street,$t + \Delta t$|Tun Tavern,0) =

$$P(z,t_0 + \Delta t|x,t_0) = (4\pi D\Delta t)^{-1/2} \exp(-(x-z)^2/4D\Delta t) \qquad (B.1)$$

where D is the diffusion coefficient or a measure of your drunken state.
The long time distribution (being at home at time t given you started

at Tun Tavern) can be calculated and is

$$P(\text{home},t|\text{Tun Tavern},0) = n/\sqrt{4\pi D}\ \exp[(-x^2/4Dt)/\sqrt{t}] \ . \qquad (B.2)$$

Suppose again we are fairly drunk but now we have someone pushing us according to some rule. Again there is a chance of not making it home, and we want to look at our probability of making it home. We will assume our short time conditional distribution is given by

$$P(X(t_0+\Delta t|X(t_0) = (2\pi\Delta t g^2)^{-1/2} \exp(-L\Delta t) \qquad (B.3)$$

where L is the rule used by this person. We again have paths over which we travel going from Tun Tavern to home. There are certain probabilities associated with each path and we will sum over all the probabilities of the paths to determine our chance of making it home. In the limit as our step size becomes continuous and $\Delta t \rightarrow 0$, we will need to integrate over all positions and over all paths and find that our chance of making it home is

$$P(X(t)|X(t_0) = \int\cdots\int \underline{D}X \exp\left(-\int L dt\right) \qquad (B.4)$$

which is our path integral.

b. **What does the Path Integral say?**

We can now easily see what the path integral gives us. If a particle (or drunk) follows a path which is determined by the Lagrangian, we sum over all possible paths (which have been weighted by the Lagrangian) and arrive at a probability distribution of the particle's position at some later time.

The Lagrangian from classical mechanics is T-V where T is the kinetic energy and V is the potential. In classical (deterministic) systems there is no path integral since there is only one path which is followed with certainty. This is the so-called classical trajectory. In classical statistical mechanics the Lagrangian is given by

$$L = (\dot{X} - f)^2/2g^2 \qquad (B.5)$$

where f is the drift and g^2 the diffusion.

2. MATHEMATICAL DESCRIPTION

a. Quantum Mechanics

The utility of the path integral is in its ability to arrive at classical mechanics as a special case of quantum mechanics in which Planck's constant $\hbar << 1$ represents the noise of the system. In quantum mechanics the path integral is defined as

$$K = \int \cdots \int \exp(iS/\hbar) \underline{D}X \tag{B.6}$$

$$S = \int_{t_a}^{t_b} L dt \tag{B.7}$$

where S is the classical action.

In contrast to the statistical mechanical classical sum over probability paths, in quantum mechanics we compute the sum over the probability amplitudes of the paths. There are no paths that are followed with certain probabilities.

For more information on path integrals in quantum mechanics see Feynman and Hibbs [Ref. 44].

b. Statistical Mechanics

Statistical mechanics is a branch of physics which attempts to describe the relationship between the microscopic properties and the macroscopic behavior. It is thus characterized by the investigation of large scale, physical, chemical, and biological systems and the search for underlying similarities. A subfield of statistical mechanics is concerned with modeling nonlinear systems using the Fokker-Planck equation. This equation is the reduced master equation of nonlinear systems and seems to be fundamental to many physical systems. Thus its solution would greatly enhance the understanding of such systems, including combat systems.

It is in this context that we have developed the path integral specifically as an application of statistical mechanics. For other applications of the path integral in statistical mechanics and applications in general see Schulman [Ref. 9] and Wiegel [Ref. 45].

APPENDIX C
SIMULATIONS/WAR GAMES DESCRIPTION

1. JANUS

JANUS is a computer simulation of combat developed by Lawrence Livermore National Labratory. It is an event-driven, stochastic simulation written in FORTRAN and work is underway of an ADA version. It models individual weapons, such as tanks, vehicles, helicopters, fixed wing aircraft and personnel as distinguishable entities. JANUS currently runs on Digital Equipment Corporation (DEC) mini-computers and has Tektronix 4125 color graphics workstations [Ref. 46,47]. JANUS is run from one or more workstations composed of a high resolution graphics terminal, 1 or 2 graphics input tablets with mice, and a DEC VT-100 terminal to communicate with the operating system.

JANUS can be run either in an interactive gaming mode or in batch mode. The graphics terminals display the terrain, location of all combat systems under control of that workstation and any enemy units which have been acquired by those combat systems. In the interactive gaming mode, a player first plans his operation by placing his forces where he wants them to start and can give subsequent movement orders. Once the game begins, the player interacts by giving orders to his forces by using the mouse and the menu on the graphics terminal.

In the batch mode, a particular scenario is chosen including an initial plan and the computer simulates the combat with no player interaction. Results can then be captured via appropriate commands given at the beginning of the run.

JANUS has a tremendous amount of flexibility in designing any partiucular scenario. User has complete control over graphics symbols, weapon system parameters, weapon platform parameters, terrain and weather parameters.

Currently JANUS is not an optimum simulation to study C^3 in batch mode because of relatively unsophisticated rules for decision-making, e. g., for a line of tanks to go around a lead tank stuck in a ditch.

Currently JANUS 3.2 is installed at TRAC (TRADOC Analysis Center) Monterey at their Conflict Simulation Lab (TCSL). They have 4 workstations located in 2 adjacent rooms to simulate the Blue and Red forces.

2. TWSEAS

The Tactical Warfare Simulation Evaluation, and Analysis System (TWSEAS) is used primarily by the Marine Corps as an instructional tool at the Command and Staff College, Quantico, Virginia [Ref. 48]. The computer simulation is designed mainly as an aid to assist in the war gaming by providing casualty, intelligence, movement and supply reports. It also calculates positions of all forces both enemy and friendly. Controllers act as mediator between the computer and the players by inputing tactical commands from the players and then reporting to the players results of their commands. This can be done over real or simulated communications nets.

The simulation is stochastic and all results are printed on high speed or PC dot matrix printers. This limits the capability of this simulation as an analysis tool. Current work is underway to correct this situation and provide an analytic as well as training tool.

3. SOTACA

The State of the Art Contigency Analysis (SOTACA) is a high resolution graphics device combined with powerful decision rule software to provide the commander a tool to select contigency alternatives. The software is written in FORTRAN and the hardware used is a DEC VAX. It is deterministic and uses as a selection process a series of decision rules which are entered by the user. The primary means of unit movement are through arcs which are placed by the user. The program then selects an optimal routing based on the decision rules and the characteristics of the arcs. Although very useful as a planning aid, it was not suitable as a source of data because of its non-stochastic nature [Ref. 49].

SOTACA was developed for use by the Pentagon and the CINC's (Commander in Chief's, Atlantic and Pacific Forces).

4. BGTT

BGTT is the Battle Group Tactical Trainer and is primarily used as a computer simulation of the naval warfare environments. Its configuration consists of one control and three player stations. At NPS, the stations are subdivided physically in the C^3 Lab by partitions. BGTT runs on a VAX-11/780 with RAMTEK Graphics Display Stations and has a high speed printer for paper output of game results. BGTT is used primarily as a staff trainer and as an analysis tool for students at NPS [Ref. 50].

5. OTHERS

a. CARMONETTE

CARMONETTE is a high resolution computer simulation of combat using small unit combined forces. It is an event driven stochastic simultion which provides for intermediate and terminal results, and is used for feasibility studies of alternative weapon systems and tactics over varying scenarios [Ref. 51].

b. FOURCE

FOURCE is a deterministic simulation of division level force-on-force combat. It is used primarily as a measure of command and control effectiveness in combat [Ref. 51].

APPENDIX D
APL SIMULATION PROGRAMS

This appendix contains the APL programs LANGEVIN and LANCHESTER. They were used to generate data from the Langevin equation and a GSL as described below and in the text.

1. PROGRAM LANGEVIN

The program LANGEVIN (shown in Figure D.1) generated data from the Langevin equation

$$\dot{X} = -.1X + g\eta \qquad (D.1)$$

where $\eta \sim N(0,1)$ and $g = 1$ (constant variance). The user inputs two vectors, INIT and P which are described in the program. The output consists of a matrix where a row corresponds to one simulated trajectory of the Langevin equation. The columns are the value of the variable (X in this case) at each time increment $(t+j\Delta t)$ where $j = 1,\ldots,T$.This data was then used by the simulated annealing algorithm described in Appendix A to perform the maximum likelihood fit.

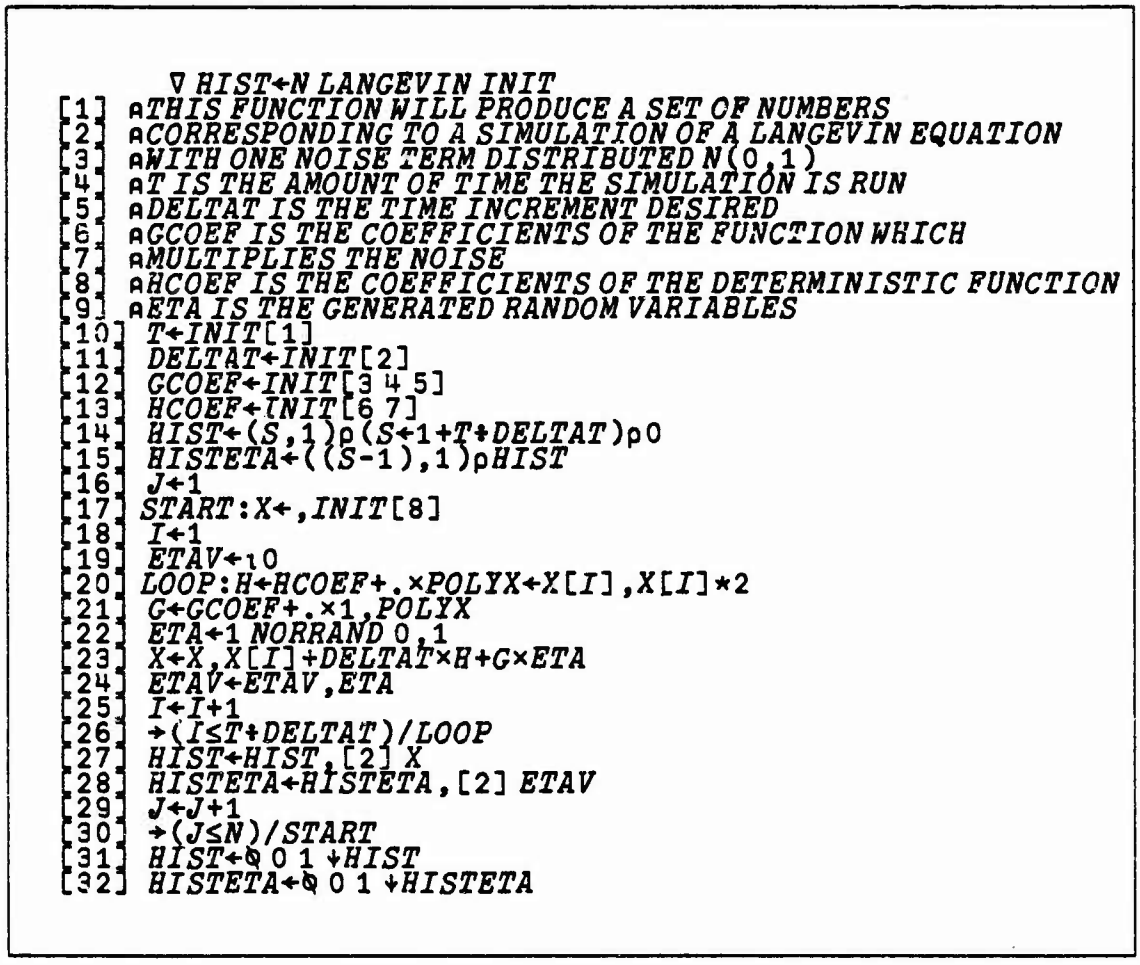
2. PROGRAM LANCHESTER

The APL program LANCHESTER (shown in Figure D.2) was used to generated data for the 2OP example described in chapter 6. Data was generated from the following equation:

$$\dot{X} = a_{11}Y + a_{12}XY + a_{13}\eta_1 + a_{14}Y\eta_2$$

$$\dot{Y} = a_{21}X + a_{22}XY + a_{23}X\eta_1 + a_{24}\eta_2$$

The user inputs, as before, two vectors, INPUT and P, where the vector INPUT contains the initial values of X and Y, time increment Δt, number of experiments, and number of time increments to use. Output is in the form of 3 matrices, HISTX, HISTY, and EXP. The rows of HISTX and HISTY are the trajectories of the X and Y



```
      ∇ HIST←N LANGEVIN INIT
[1]   ⍝THIS FUNCTION WILL PRODUCE A SET OF NUMBERS
[2]   ⍝CORRESPONDING TO A SIMULATION OF A LANGEVIN EQUATION
[3]   ⍝WITH ONE NOISE TERM DISTRIBUTED N(0,1)
[4]   ⍝T IS THE AMOUNT OF TIME THE SIMULATION IS RUN
[5]   ⍝DELTAT IS THE TIME INCREMENT DESIRED
[6]   ⍝GCOEF IS THE COEFFICIENTS OF THE FUNCTION WHICH
[7]   ⍝MULTIPLIES THE NOISE
[8]   ⍝HCOEF IS THE COEFFICIENTS OF THE DETERMINISTIC FUNCTION
[9]   ⍝ETA IS THE GENERATED RANDOM VARIABLES
[10]  T←INIT[1]
[11]  DELTAT←INIT[2]
[12]  GCOEF←INIT[3 4 5]
[13]  HCOEF←INIT[6 7]
[14]  HIST←(S,1)⍴(S←1+T÷DELTAT)⍴0
[15]  HISTETA←((S-1),1)⍴HIST
[16]  J←1
[17]  START:X←,INIT[8]
[18]  I←1
[19]  ETAV←⍳0
[20]  LOOP:H←HCOEF+.×POLYX←X[I],X[I]*2
[21]  G←GCOEF+.×1,POLYX
[22]  ETA←1 NORRAND 0,1
[23]  X←X,X[I]+DELTAT×H+G×ETA
[24]  ETAV←ETAV,ETA
[25]  I←I+1
[26]  →(I≤T÷DELTAT)/LOOP
[27]  HIST←HIST,[2] X
[28]  HISTETA←HISTETA,[2] ETAV
[29]  J←J+1
[30]  →(J≤N)/START
[31]  HIST←⍉0 1 ⍉HIST
[32]  HISTETA←⍉0 1 ⍉HISTETA
```

Figure D.1 LANGEVIN Program.

order parameters, respectively. The columns correspond to the values of the variables at the time, $t + j\Delta t$, where $j = 1, \ldots, T$ where $T = NEXP \times DELTAT$. The matrix EXP is a concatenation of the HISTX and HISTY matrices and was used as input for the simulated annealing program of Appendix A.

```
     ∇ INIT LANCHESTER P;INIT;P
[1]  ⍝THIS FUNCTION WILL PRODUCE A SET OF NUMBERS
[2]  ⍝CORRESPONDING TO A SIMULATION OF A SET OF LANGEVIN
[3]  ⍝EQUATIONS WITH TWO NOISE TERMS DISTRIBUTED N(0,1)
[4]  ⍝NUMRUN IS THE AMOUNT OF TIME THE SIMULATION IS RUN
[5]  ⍝DELTAT IS THE TIME INCREMENT DESIRED
[6]  ⍝GCOEF1 IS THE COEFFICIENTS OF THE FUNCTION WHICH
[7]  ⍝MULTIPLIES THE NOISE OF THE X VARIABLE
[8]  ⍝HCOEF1 IS THE COEFFICIENTS OF THE DETERMINISTIC
[9]  ⍝FUNCTION FOR X
[10] ⍝ETA1 IS THE GENERATED RANDOM VARIABLES FOR X VARIABLE
[11] ⍝SIMILAR VARIABLES EXIST FOR Y WITH A NUMBER 2 SUFFIX
[12] HISTX←HISTY←⍳0
[13] NUMRUN←INIT[1]
[14] DELTAT←INIT[2]
[15] NEXP←INIT[5]
[16] EXP←((2×NEXP),NUMRUN+1)⍴0
[17] HCOEF1←P[⍳4]
[18] GCOEF1←P[4+⍳4]
[19] HCOEF2←P[8+⍳4]
[20] GCOEF2←P[12+⍳4]
[21] V1←GCOEF1>0
[22] V2←GCOEF2>0
[23] J←1
[24] BEGIN:X←1⍴INIT[3]
[25] Y←1⍴INIT[4]
[26] I←1
[27] LOOP:POLY←1,X[I],Y[I],X[I]×Y[I]
[28] ETA←2 NORRAND 0 1
[29] H1←HCOEF1+.×POLY
[30] H2←HCOEF2+.×POLY
[31] G1←V1/GCOEF1×POLY
[32] G2←V2/GCOEF2×POLY
[33] X←X,X[I]+DELTAT×H1+G1+.×ETA
[34] Y←Y,Y[I]+DELTAT×H2+G2+.×⌽ETA
[35] I←I+1
[36] →(I≤NUMRUN)/LOOP
[37] HISTX←HISTX,X
[38] HISTY←HISTY,Y
[39] EXP[J;]←X
[40] EXP[J+1;]←Y
[41] J←J+2
[42] →(J<2×NEXP)/BEGIN
[43] HISTX←(NEXP,NUMRUN+1)⍴HISTX
[44] HISTY←(NEXP,NUMRUN+1)⍴HISTY
```

Figure D.2 LANCHESTER Program.

LIST OF REFERENCES

1. L. Ingber, "Nonlinear nonequilibrium statistical mechanics approach to C^3 systems," in *9th MIT/ONR Workshop on C^3 Systems*, (MIT, Cambridge, MA, 1986), p. 237-244.

2. L. Ingber, "Applications of biological intelligence to command, control and communciations," in *Computer Simulation in Brain Science*, ed. by R. Cotterill, (Cambridge U.,London, 1987), to be published.

3. J.G. Taylor, *Lanchester Models of Warfare*, Vols. I and II, (Operations Research Society of America, Arlington, VA, 1983).

4. D. P. Gaver, "Models of conflict, with explicit representation of command and control capabilities and vulnerabilities," (Naval Postgraduate School, CA, February 1981).

5. H. Haken, *Synergetics, An Introduction*, 3rd edition, (Springer-Verlag, 1983).

6. R. Bretnor, *Decisive Warfare - A Study in Military Theory*, (Stackpole Books, 1969).

7. F. Langouche, D. Roekaerts, and E. Tirapegui, *Functional Integration and Semiclassical Expansions*, (Reidel, Dordrecht, 1982).

8. L. Ingber, "Statistical mechanics of nonlinear nonequilibrium financial markets," *Math. Modelling*, Volume 5, 343-361 (1984).

9. L.S. Schulman, *Techniques and Applications of Path Integration*, (J. Wiley & Sons, New York, 1981).

10. R. Graham, "Covariant formulation of non-equilibrium statistical thermodynamics," *Z. Physik*, Volume B26, 397-405 (1977).

11. F. Langouche, D. Roekaerts and E. Tirapegui, "On the most probable path for diffusion processes," *J. Phys. A:Math. Gen.*, Volume 11, L263-L268 (1978).

12. L. Ingber, "Statistical mechanics algorithm for response to targets (SMART)," in *Workshop on Uncertainty and Probability in Artificial Intelligence*, (AAAI-RCA, Menlo Park, CA, 1985), p. 258-264.

13. M.F. Wehner and W.G. Wolfer, "Numerical evaluation of path-integral solutions to Fokker-Planck equations I.," *Phys. Rev. A*, Volume 27, 2663-2670 (1983).

14. M.F. Wehner and W.G. Wolfer, "Number evaluation of path-integral solutions to Fokker-Planck equations II. Restricted stochastic processes," *Phys. Rev. A*, Volume 28, 3003-3011 (1983).

15. M.F. Wehner and W.G. Wolfer, "Numerical evaluation of path integral solutions to Fokker-Planck Equations III. Time and functionally dependent coefficients," *Phys. Rev. A*, to be published (1987).

16. J. Adams, U. Wisconsin and Sandia Lab, private communication (1987).

17. L. Ingber, "Statistical mechanics of neocortical interactions. I. Basic formulation," *Physica D*, Volume 5, 83-107 (1981).

18. L. Ingber, "Statistical mechanics of neocortical interactions. Dynamics of synaptic modification," *Phys. Rev. A.*, Volume 28, 395-416 (1983).

19. L. Ingber, "Statistical mechanics of neocortical interactions. Derivation of short-term-memory capacity," *Phys. Rev. A*, Volume 29, 3346-3358 (1984).

20. L. Ingber, "Statistical mechanics of neocortical interactions. EEG dispersion relations," *IEEE Trans. Biomed. Eng.*, Volume 32, 91-94 (1985).

21. L. Ingber, "Statistical mechanics of neocortical interactions: Stability and duration of the 7 ± 2 rule of short-term-memory capacity," *Phys. Rev. A*, Volume 31, 1183-1186 (1985).

22. L. Ingber, "Towards clinical applications of statistical mechanics of neocortical interactions," *Innov. Tech. Biol. Med.*, Volume 6, 753-758 (1985).

23. L. Ingber, "Towards a unified brain theory," *J. Social Biol. Struct.*, Volume 4, 211-224 (1981).

24. L. Ingber, "Nuclear forces," *Phys. Rev.*, Volume 174, 1250-1263 (1968).

25. L. Ingber, "Riemannian corrections to velocity-dependent nuclear forces," *Phys. Rev. C*, Volume 28, 2536-2539 (1983).

26. L. Ingber, "Path-integral Riemannian contributions to nuclear Schrödinger equation," *Phys. Rev. D*, Volume 29, 1171-1174 (1984).

27. L. Ingber, "Riemannian contributions to short-ranged velocity-dependent nucleon-nucleon interactions," *Phys. Rev. D*, Volume 33, 3781-3784 (1986).

28. N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *Journal of Chemical Physics*, Volume 21, 1087-1092 (1953).

29. S.K. Ma, *Statistical Mechanics*, (World Scientific,Philadelphia, 1985).

30. S.K. Ma and M. Payne, "Entropy of a spin glass model with long range interactions," *Physical Review B*, Volume 24, 3984-3990 (1982).

31. Kirkpatrick, S.,Gelatt, C. D.Jr.,and Vechhi, M. P., "Optimization by simulated annealing," *Science*, Volume 220, 671-680 (1983).

32. W. E. Smith, H. H. Barett, R.G. Paxman, "Reconstruction of objects from coded images by simulated annealing," *Optics Letters*, Volume 8, 199-201 (1983).

33. W. E. Smith, R. G. Paxman, H. H. Barrett, "Image reconstruction from coded data: I. Reconstruction algorithms and experimental results," *Journal of the Optical Society of America A*, Volume 2, 491-500 (1985).

34. J.W. Greene and K. J. Supowit, "Simulated annealing without rejected moves," *Proceedings, IEEE International Conference on Computer Design: VLSI in Computers*, (Port Chester, NY,1984), p. 658-663.

35. Y. Bar-Shalom, "Tracking methods in a multitarget environment," *IEEE Transactions on Automatic Control*, Volume 23, 618-626 (1978).

36. P. Molitor, "Layer assignment by simulated annealing," *Microprocessing and Microprogramming*, Volume 16, 345-349 (1985).

37. D.B. Paul, "Training of HMM recognizers by simulated annealing (speech recognition)," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Volume 1, 13-16 (1985).

38. E. Bonomi, J.L. Lutton, "The N-city travelling salesman problem:Statistical mechanics and the Metropolis algorithm," *SIAM Review*, Volume 26, 551-568 (1984).

39. I. O. Bohachevsky,M. E. Johnson, and M. L. Stein, "Generalized simulated annelealing for function optimization," *Technometrics*, Volume 28, 209-217 (1986).

40. L.T. Wille and J. Vennik, "Electrostatic energy minimization by simulated annealing," *Journal Physics A (GB)*, Volume 18, no. 17, L1113-17 (1985).

41. A.U. Levy and A. Montalvo, "The tunneling algorithm for the global minimization of functions," *SIAM Journal of Scientific and Statistical Computing*, Volume 6, 15-29 (1985).

42. R. Hartley and H. Szu, "Generalization and analysis of the simulated annealing algorithm," to be published.

43. S. Geman, D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Transactions, on Pattern Analysis and Machine Intelligence*, Volume 6, 721-741 (1984).

44. R.P. Feynman and A.R. Hibbs, *Quantum Mechanics and Path Integrals*, (McGraw-Hill, New York, 1965).

45. F.W. Wiegel, "Path integral methods in statistical mechanics," *Phys. Rep. Phys. Lett. C (Netherlands)*, Volume 16, 57-114 (1975).

46. Lawrence Livermore National Laboratory, Conflict Simulation Center, *The JANUS Manual Version, 3.02*, (Lawrence Livermore National Laboratory, CA, November 1986).

47. Lawrence Livermore National Laboratory, Conflict Simulation Center, *JANUS Algorithm Document Version 3.01*, (Lawrence Livermore National Laboratory, CA, September 1986).

48. U.S. Marine Corps, *Introduction to TWSEAS, AN/TSQ-T9(V)* , (U.S. Marine Corps, MCDEC, Quantico, VA, not dated).

49. *State of the Art Contingency Analysis (SOTACA) Analyst's Guide to Theory (Preliminary Draft)*, (Joint Analysis Directorate, Organization of the Joint Chiefs of Staff, Washington, D.C., March 1986).

50. N. E. Hardee, *An Assessment of the Ability of the Headquarters Assessment Tool (HEAT) to quantify the C² system effectiveness of a simued U.S. Navy Tactical-Level Headquarters under periods of Communications Stress*, (Masters Thesis, Naval Postgraduate School, Monterey, CA, September 1985)

51. Studies and Analysis Activity, Headquarters TRADOC *Inventory of Models* (Headquarters, Training and Doctrine Command, Fort Monroe, VA, June 1985).

BIBLIOGRAPHY

Baker, G.A. Jr., and P. R. Graves-Morris, *Padé Approximants, Part I:Basic Theory*, (Addison-Wesley, Reading, MA, 1981).

Baker, G.A. Jr., and P. R. Graves-Morris, *Padé Approximants, Part II:Extensions and Applications*, (Addison-Wesley, Reading, MA, 1981).

Dekker, H., "Quantization in curved spaces," in *Functional Integration: Theory and Applications*, ed. by J.P. Antoine and E. Tirapegui, (Plenum, New York, 1980), pp. 207-224.

Gardiner, C. W., *Handbook of Stochastic Methods for Physics, Chemistry and the Natural Sciences*, (Springer-Verlag, Berlin, Germany, 1983).

Goldstein, H., *Classical Mechanics*, (Addison-Wesley, Reading, MA. 1959).

Graham, R., "Lagrangian for diffusion in curved space," *Phys. Rev. Lett.*, Volume 38, 51-53 (1977).

Mathews, J., and R. L. Walker, *Mathematical Methods of Physics*, (W. A. Benjamin, Menlo Park, CA, 1970).

Szu, H., "Three layers of vector outer product neural networks for optical pattern recognition," to be published.

van Kampen, N. G., *Stochatic processes in physics and chemistry*, (North-Holland, Amsterdam, 1981).

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Professor Lester Ingber, Code 66IL Naval Postgraduate School Monterey, CA 93943-5004	30
4.	Professor Ed Rockower, Code 55RF Naval Postgraduate School Monterey, CA 93943-5004	3
5.	Peter Purdue, Code 55 Chairman, Operations Research Department Naval Postgraduate School Monterey, CA 93943-5004	2
6.	K. E. Woehler, Code 61 Chairman, Physics Department Naval Postgraduate School Monterey, CA 93943-5004	2
7.	Mike Sovereign, Code 74 Chairman, C3 Academic Group Naval Postgraduate School Monterey, CA 93943-5004	5
8.	Professor Don Gaver, Code 55GV Naval Postgraduate School Monterey, CA 93943-5004	2
9.	Maj. Bernie Galing TRADOC Monterey Box 8692 Monterey, CA 93943-0692	2
10.	Professor Gerry Brown, Code 55BW Naval Postgraduate School Monterey, CA 93943-5004	2
11.	Professor R. R. Read, Code 55RE Naval Postgraduate School Monterey, CA 93943-5004	1
12.	S. Christian Simonson III Conflict Simulation Center Evaluation and Planning Divison Lawrence Livermore National Laboratory Livermore, CA 94550	2

13. Col. Wilson
Code C4
Headquarters, U.S. Marine Corps
Washington, D.C. 20380 — 1

14. J. B. Adams
Sandia Laboratories
Livermore, CA 94550 — 1

15. I. R. Goodman
Naval Ocean Systems Center (NOSC)
San Diego, CA 92152 — 1

16. Mike Madarian
Naval Ocean Systems Center (NOSC)
San Diego, CA 92152 — 1

17. Don Blumenthal
Conflict Simulation Center
Lawrence Livermore National Laboratory
Livermore, CA 94550 — 1

18. G. E. Schacher, Code 06
Dean of Science and Engineering
Naval Postgraduate School
Monterey, CA 93943-5004 — 1

19. Prof. Don E. Harrison, Code 61HX
Naval Postgraduate School
Monterey, CA 93943-5004 — 1

20. Mike F. Wehner
Lawrence Livermore National Laboratory
Livermore, CA 94550 — 1

21. William G. Wolfer
Sandia National Laboratory
Livermore, CA 94550 — 1

22. Prof. Aram Mekjian
The State University of New Jersey
Rutgers
Department of Physics and Astronomy
P.O. Box 849
Piscataway, NJ 08854 — 1

23. Adrian S. Yano
2124 Kittredge Street
Berkeley, CA 94704 — 1

24. Dr. Israel Mayk
CECOM
Fort Monmouth, NJ 07703 — 1

25. Captain Stephen C. Upton
Research, Development and Studies Division
Headquarters, U.S. Marine Corps
Washington, D.C. 20380 — 5

26. Maj. Russ Wimberly
TRADOC Monterey
Box 8692
Monterey, CA 93943-0692 — 2