AGGREGATED COMBAT MODELS

Operations Research Department
Naval Postgraduate School
Monterey, California

May, 2022

Aggregated Combat Models

Last updated by Alan Washburn

These notes have been created over a period of several decades. Contributors include Bill Caldwell, Jim Hartman, Sam Parry, Al Washburn, and Mark Youngren. They will continue to change as the art of combat modeling keeps pace with computer technology and the changing nature of combat itself.

Table of Contents

CHAPTER 1

AGGREGATION IN COMBAT MODELS

1.1 – The Need for Aggregated Models

Simulation models of large-scale combat contribute useful insights for many military decision problems. The designers of such models attempt to achieve a representation of warfare that is as accurate and believable as possible. For moderate sized forces believability is aided by modeling in high resolution, but larger forces require aggregation to keep the models within the limits of computer size and execution time. This first chapter will concentrate on the idea of aggregation in combat models – what it means, how it is accomplished, and what it implies about model believability.

High-Resolution Versus Aggregated Models

In a high-resolution combat model a detailed view of warfare is achieved by representing individual combatants as separate entities. Each such entity has numerous attributes that define its unique position in the force, its unique perception of the battlefield and the enemy force, its capabilities, and its activity at each moment of simulated battle time.

Combat processes are decomposed into high-resolution sequences of events and activities. Complex timing mechanisms coordinate the event sequences for the numerous combatants so that subtle interaction patterns can be modeled.

The goal throughout a high-resolution simulation is to model each combat phenomenon so that results are traceable (via formulas and logic that we understand and accept as representing combat actions) to specific physical performance data or to specific behavioral assumptions. The existence of such an audit trail is the single most significant advantage of a well constructed high-resolution model. It enables us to evaluate subtle differences in weapons, sensors, or tactics, and to understand why the different inputs yield improved (or degraded) combat performance.

But there are some problems with the high-resolution modeling approach. High-resolution models involve large, complex computer programs that are expensive to develop, maintain, and run. They are also usually stochastic, so replication is needed to obtain answers about simulated battles. As we try to model larger forces at division level or higher, the sheer number of combatants and weapon systems makes it impossible to maintain individual item resolution. Models at these levels have to sacrifice detail for scope. By aggregating individual combatants into larger units, the large-scale combat modeler can decrease the number of simulation entities back to a manageable number.

As we will soon see, aggregating individual combatants into units totally changes the description of some basic combat processes. There are many interesting approaches to modeling combat interactions among these aggregated units. The primary purpose of this book is to examine aggregation procedures and the large-scale combat models that make use of the resulting aggregated units.

1.2 – What Aggregation Involves

At the simplest level, an aggregated combat model is one in which the basic model entities are groups rather than individual combatants. This simple concept has implications that propagate throughout the entire combat model structure.

Aggregate Combatants into Combat Units

Combining individual combatants into groups of combatants is a simple task because the hierarchical military command structure has already defined a spectrum of appropriate groupings. Army combat models generally select their basic unit size from the command hierarchy:

individual combatant,
squad,
platoon,
company,
battalion,
brigade,
division,
corps,
army,
theater.

The simulation entities are then combat and support units of the basic size or larger. Detailed attribute lists are maintained for each simulation entity, so that (supposing the basic unit size to be a battalion) each battalion has its own unique situation, perception of the battlefield, capabilities, and current activity.

Units larger than the basic size, such as higher headquarters, are modeled as individual entities with whatever attributes are required be the model’s description of the command hierarchy.

Individual combatants and groups of combatants that are smaller than the basic size are not carried as separate simulation entities. The detailed attributes of these smaller groups are lost to the simulation. For example, in place of a precise location for each combatant, an aggregated model might maintain a center location and a front width for a battalion. Individual combatants in the battalion are assumed to be somewhere in the battalion region (perhaps uniformly distributed) but the model does not keep track of individual locations.

Limited information about the state and activity of the combatants in an aggregated unit may be kept in the form of average attributes, but the variability across individual systems is lost. Thus, for example, the total remaining ammunition for a tank battalion might be kept as an attribute of the battalion. This, along with the remaining number of tanks, yields an average number of rounds for each individual tank, but an aggregated model would typically not care that some tanks were nearly out of ammunition while others have fired only a few shots.

The first consequence of aggregating individual combatants into groups is that information about individual differences is lost. Aggregated models often deal in average properties and thus tend to smooth out the fluctuations seen in the outputs of high-resolution models.

Aggregate Stochastic Interactions into Process Averages

The second consequence of combining individuals into groups is that we lose track of what each individual is doing at any given time. The complex intertwined stochastic event sequences found in high-resolution models are replaced by average behavior. In a high-resolution model, attrition is simulated by acting out target acquisition, target selection, firing, impact, and assessment as a sequence of discrete events. In an aggregated model we might, instead, compute the rate at which an average tank kills enemy tanks. This attrition rate implicitly includes all of the events that lead up to a target kill, but the stochastic variability is suppressed. In fact, aggregated simulations are often designed as deterministic models.

One of the problems with modeling combat using process averages is that we may lose the audit trail back to engineering level data. Many assumptions and scenario details get wrapped up into the attrition rate number, and it is not easy to untangle their effects. This is particularly true if the model user is relying on some other agency to provide the input rates for his study.

Aggregate Event Sequences into Time Blocks

The third consequence of aggregation is that an aggregated model loses information about event sequencing because it does not keep track of individual actions. The precise times of critical events, such as target kills, are not available. Instead, the process averages are applied over relatively long periods of time (from a few minutes to a day) to compute total casualties for that time period. The process rate data must take into account the fact that the attrition process is two sided, so that not all combatants will survive to continue firing for the entire time block.

Extreme care must be taken to make the underlying assumptions clear in computing rate data for such models. For example, a daily attrition rate for aircraft stationed at an airbase that is under attack may be inconsistent with the day's plan for those aircraft (they may be flying missions and thus not on the ground when their base is attacked). A simulation that assesses attrition on an aggregated daily cycle cannot know whether the airbase attack coincided with aircraft on the ground, so an assumption must be made and figured into the attrition rate data.

Overall, then, the outstanding feature of aggregation in combat models is information loss. As a result, it is necessary to model combat processes using aggregate behavior rather than individual behavior. Although values for the aggregate process rates are generally computed outside the combat model, the complex nature of these rates demands that we devote considerable effort in this text to explaining their derivation.

1.3 – Some Typical Entity Aggregations

The previous section's description of aggregating individual combatants into larger combat units is oversimplified. In practice, aggregated combat simulations use a variety of different aggregation patterns. Even within a single combat simulation, the aggregation used for, say, maneuver control may be different from that used for attrition computations or for logistics accounting.

The distinction between homogeneous and heterogeneous aggregations is particularly important for attrition computations. A homogeneous aggregation is one where the combat power of the unit is combined into a single measure (or perhaps one combat power index for ground weapons and another for aircraft). Attrition computations are then based on the relative power of the two forces in a battle, often by computing the ratio of their combat power indices.

In a heterogeneous aggregated model, the aggregated unit maintains a count of the number of surviving weapon systems of each distinct type. This allows modeling of the differing effectiveness of a firer weapon type against various types of enemy target. The trend in modern models is toward heterogeneous aggregations because they allow more accurate attrition modeling than the homogeneous models. More information is lost in a homogeneous aggregation. We will discuss several attrition models for homogeneous and heterogeneous aggregations in later chapters of the book.

In this section we will briefly discuss the entity aggregations used by several current large-scale combat simulation models. These descriptions are only intended to give an idea of the variety of aggregation patterns in use today. The descriptions concentrate on the basic ideas and ignore some fine points that are important for a thorough understanding of any of these models. Details that are suppressed here will be provided for some of the models in the chapters that discuss aggregated process modeling.

The following list of about a dozen models is representative of the variety found in large-scale combat simulations. The list is by no means exhaustive, but rather represents models with which the author has some familiarity. Throughout the book we will provide specific modeling examples drawn from these simulations to illustrate our discussions of aggregated combat process modeling.

It should be emphasized that the specific modeling examples and combat simulation descriptions in this book are mostly taken from published documentation that cannot always be totally comprehensive. Also, the models are often modified and improved while the documentation may lag. Thus some details of the models may have evolved beyond the descriptions presented here. Nevertheless we feel that it is important to give examples of actual modeling practice as well as the theory (where a theory exists) upon which the models are based.

The ATLAS Model

ATLAS is a deterministic theater level model of conventional ground and air combat developed in the mid 1960s. For the ground war, a homogeneous aggregation method is used with a division sized basic unit. Each division has its ground combat power represented by a single "firepower index" value. A combat sector covers about a corps front, and the firepower indices for the divisions in a sector are added together to give a measure of the corps' combat power.

Aircraft operations are aggregated to the number of aircraft assigned to each of several mission categories within each sector. Individual aircraft or aircraft sorties are not represented.

The CEM Model

CEM (originally CONAF Evaluation Model, currently Combat Evaluation Model) is a deterministic theater model of ground and air combat that has been developed through several versions since about 1970. It is currently maintained by the Center for Army Analysis (CAA).

A partial heterogeneous aggregation system is used with a brigade sized basic unit for blue forces and a division sized unit for red. A brigade entity (or division for red) has attributes to keep track of the kinds of battalions (regiments) it contains.

Firepower for the brigade is maintained in six weapon type categories: heavy armor, light armor, soft systems, artillery, helicopter, and fixed wing aircraft. Within each of the six classes, distinct and rather dissimilar weapon systems may still be aggregated together.

The IDAGAM Model

IDAGAM is a deterministic theater level model of ground and air combat developed in 1974 by the Institute for Defense Analysis. The basic maneuver unit is generally a division or an independent brigade. Each division is represented in a heterogeneous manner; the model keeps track of the surviving number of each distinct weapon system type within each division. A typical division description might contain about a dozen ground weapon system types. In addition, IDAGAM represents the number of people in each division in several categories.

The air aggregation represents the number of aircraft (by aircraft type) at each of several airbases. Later in the model these aircraft are allocated to seven primary and five secondary mission categories. Individual sorties and individual aircraft are not represented.

The VECTOR-2 Model

The VECTOR-2 model was developed in about 1976 by the Vector Research Corporation. It represents deterministic ground and air theater combat among several kinds of units.

Ground maneuver forces are represented by battalion sized basic maneuver units. Within each aggregated maneuver battalion, the VECTOR model keeps track of the number of each distinct weapon system (in eleven categories plus personnel) thus using a heterogeneous aggregation system.

Artillery units, air defense units, fixed wing tactical air units, and helicopter units are represented similarly in terms of the weapon systems they contain.

The FORCEM Model

FORCEM was developed in the early 1980s by the Center for Army. FORCEM represents theater combat with a division sized basic simulation entity. The forces within a division are represented as weapon systems by type.

The COMMANDER Model

COMMANDER is an updated (1980) version of the TALON model that was developed in 1978. The model represents theater combat with a basic ground maneuver unit determined by the user. The ground force part of the model uses a homogeneous aggregation scheme, measuring combat power of a ground unit in "T62 tank equivalents".

The air model contains substantially more detail than the ground portion. COMMANDER simulates individual air sorties in detail. Each sortie is characterized by attributes such as the airbase, air unit, aircraft type, mission category, number of aircraft, specific target, time of attack, and weapon load.

The JTLS Model

JTLS was developed in 1983-84 for a group consisting of the Joint U.S. Readiness Command, the Army War College, and the Center for Army Analysis. JTLS is a stochastic, real time player interactive simulation of joint theater combat.

The basic ground maneuver unit in JTLS is a division or sometimes an independent brigade. Each division maintains a heterogeneous list of combatants by weapon system type.

The air model basic entity is an air mission with a heterogeneous representation of the individual aircraft types involved and a detailed simulation of the progress of the mission against a specific target unit or location.

The ICOR Model

ICOR is a corps level man-in-the-loop ground and air combat simulation developed by the BDM Corporation in the late 1970s. The basic ground simulation units are maneuver battalions and sometimes individual companies, batteries, or sensors. Weapon systems within the unit are represented by a heterogeneous list. Air operations are resolved at the sortie level.

The COSAGE Model

COSAGE is a fairly high-resolution, stochastic model of division level ground combat. It was developed about 1980 for the Center for Army Analysis. The main use of COSAGE is to provide attrition and ammunition usage rates for more aggregated theater models such as CEM and FORCEM.

Maneuver units in COSAGE are resolved to blue platoons and red companies. Within each maneuver unit a heterogeneous list of weapons is maintained. During direct fire engagements, individual weapon systems are arranged in combat formations, interactions between weapon system types are computed, and individual weapons may be stochastically killed.

Close air support of the ground battle by tactical aircraft is resolved to a detailed simulation of the flight of each sortie.

The FOURCE Model

FOURCE is a division level, deterministic combat model developed by the U.S. Army TRADOC Systems Analysis Activity in the mid 1970s. The model emphasizes command and control staff functions and intelligence processing from the battalion to the division level. Its primary use is for the evaluation of proposed automated C3 systems.

Combat forces are aggregated to battalion entities using a heterogeneous aggregation scheme.

1.4 – Aggregated Process Descriptions

The description of combat processes is substantially different in aggregated simulations than in high resolution into sequences of discrete events that occur at precise moments of simulated time. Event outcomes and event times are often modeled using random variables.

Processes for Units Above the Resolution Limit

At echelons above the basic aggregated unit, an aggregated combat model may represent combat processes in a manner that shares features of the high-resolution approach. Units at or above the basic aggregated unit size exist in the model as distinct entities and have their own unique lists of attributes. Thus interactions among them can be represented in detail and different units can be performing distinct combat missions. Thus division, corps, and theater level decisions and tasks are often represented in relatively high resolution in aggregated models. Because these larger units tend to operate with longer time horizons, it is common (but not universal) to model their actions using a time step simulation mechanism.

Processes for Units Below the Resolution Limit

Echelons below the size of the basic aggregated unit are not explicitly represented as simulation entities in an aggregated model. Rather they are represented as attributes of the aggregated unit to which they belong (with substantial loss of information about the properties of the individual combatants).

Combat process descriptions for these smaller echelons are radically different from the high-resolution event sequences. Since the model does not keep track of individual attributes, it cannot know details of what a particular individual is doing at any time. Instead, aggregated models represent the <u>average results</u> of <u>many combatants</u> interacting <u>over a period of time</u> by using the rates at which various process outcomes occur. As a simple example, an attrition equation might compute enemy casualties during a time interval as

$$\text{Y\_CASUALTIES} = \text{X\_FIRERS} * \text{ATTR\_RATE} * \Delta \text{T}$$

where X_FIRERS is the average number of friendly shooters in the battle, ATTR_RATE is the average rate at which a single surviving friendly shooter kills enemy systems, and Δ T is the length of the engagement.

In such an equation it is necessary to use the average number of firers because the aggregation causes us to lose track of the details of event sequencing – we cannot know when a particular firing system becomes a casualty to enemy fire and stops shooting.

Although this simple attrition equation is “obviously” correct, it is not trivial to make it behave realistically. Much of the complexity of a real battle has been concealed inside the attrition rate coefficient. It is not at all clear, for example, that ATTR_RATE is a constant throughout the battle, and if not, what other variables it depends upon. Aggregated models tend to have simple process descriptions that incorporate coefficients whose meanings may be difficult to understand and whose values may be very difficult to compute.

Of all the combat processes that occur below the resolution limit of aggregated models, attrition is the process that has received the most study. Many aggregated attrition model structures have been developed, and later chapters of this work will examine some of them in detail.

Because aggregated combat process descriptions average over many individual engagements and over relatively long time intervals, they are usually treated as deterministic. The random fluctuations of a high-resolution model are assumed to average out yielding fluctuations that are insignificant at levels above the division.

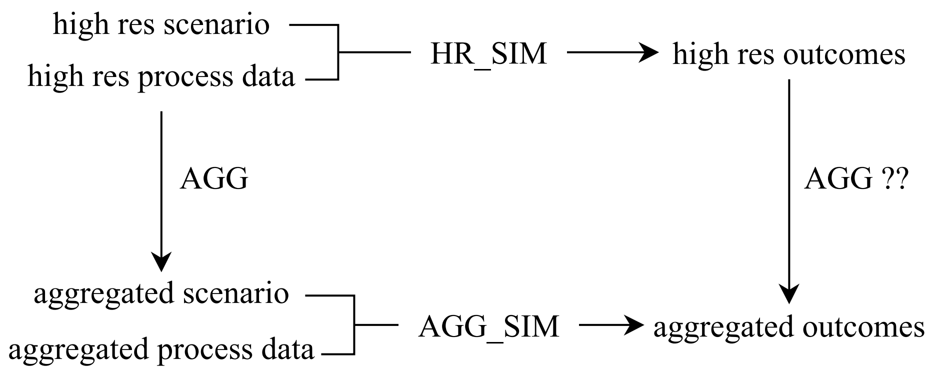
1.5 – Consistency, Estimation, and Calibration

The entire question of whether a combat model is behaving in a way that really represents combat outcomes is a complex and difficult issue. It is not our purpose here to provide an extended discussion of the validation of combat models. Instead we will address the more restricted issue of determining whether an aggregated combat model is consistent with a high-resolution model when both simulate the same scenario. The high-resolution model has some face validity because its detailed process structures follow real actions and events closely. Thus, being consistent with a high-resolution model would be an advantage for an aggregated simulation.

Consistency

From the beginning it should be clear that we cannot verify the consistency of an entire aggregated theater simulation against a high-resolution model because no high-resolution simulations of theater combat exist. Our goal must be more modest. During the execution of a large-scale aggregated simulation, numerous local combat encounters involving small parts of the total force will occur. The aggregated model will resolve the combat outcomes using its aggregated attrition process models. We could (at least in principle) set up the same local scenarios in a small-scale high-resolution simulation and then compare the results from the two models.

The overall procedure can be viewed as in the following diagram:



The high-resolution simulation can be viewed as a mapping, HR_SIM, that converts a high-resolution input data set into combat outcomes for each individual combatant (thus high-resolution outcomes). Similarly, AGG_SIM is the mapping defined by the aggregated simulation program that converts aggregated inputs to aggregated combat outcomes.

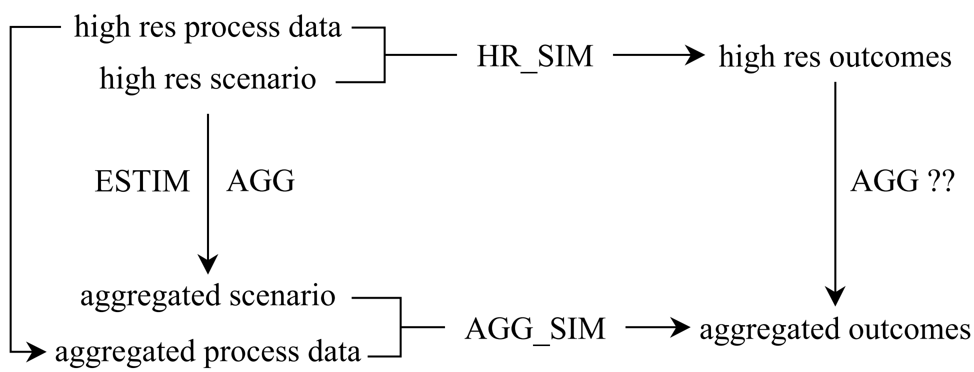
Running both programs on the same scenario is accomplished by aggregating the high-resolution data to provide an aggregated scenario (using the aggregation mapping AGG). If, when we apply the same AGG mapping to the high-resolution outcomes, the results are similar to the outcomes from AGG SIM, then the two models are consistent.

In practice it is not so simple. Describing the AGG mapping for scenario entities is probably straightforward, but aggregating the process data is considerably more difficult because the structure of the aggregated process descriptions is so different from the high-resolution process models.

A direct application of the above diagrammed consistency check would be extremely difficult. Even if it could be accomplished, it would not be surprising to find considerable output differences between two models developed by different modelers, under different circumstances, with different study objectives in mind, and with different implicit assumptions during the modeling process.

Estimation of Process Coefficients

Since the aggregated process descriptions are so different from high-resolution process descriptions, an additional step must be inserted if the above consistency check is to have any hope of succeeding. A separate procedure must be developed for estimating the aggregated process coefficients from the engineering level data base for the high-resolution model. The resulting diagram then becomes:

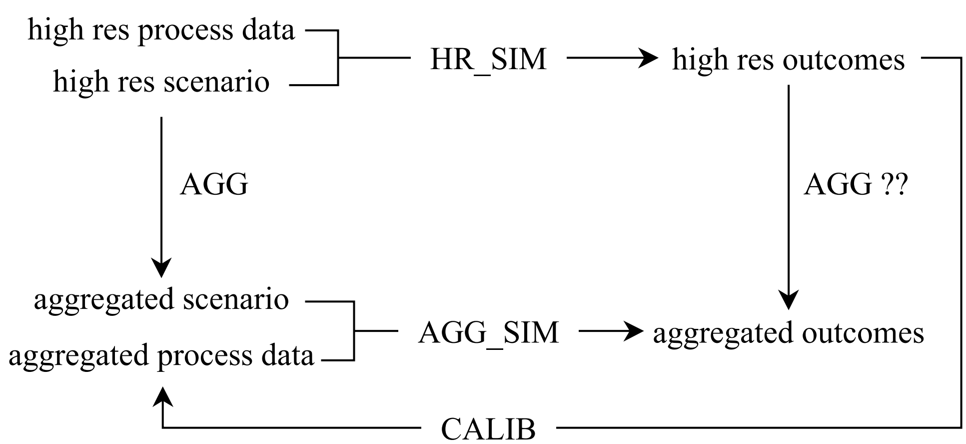


where ESTIM is the aggregated coefficient estimation procedure.

Coefficient estimation procedures are highly developed for the attrition process. They involve complex stand-alone stochastic models that are run as preprocessors to the aggregated simulation run. Details will be presented in later chapters.

Calibration

Another approach to obtaining consistency to use the output from the high-resolution model to determine the aggregated process data in a way that forces consistency between the two models. Such a procedure is called calibration of the aggregated model. It can be diagrammed as follows:



Note that calibration requires that the high-resolution model must exist and be executed in a scenario compatible with the aggregated scenario. Estimation only requires access to high-resolution process data. It can be performed even if a high-resolution model is not available for output comparisons.

A procedure for estimating or calibrating aggregated process coefficients is a particularly important part of any aggregated combat modeling project. The mathematical details of coefficient determination procedures will be presented along with the corresponding aggregated process models in later chapters.

Estimation or calibration procedures are generally not a part of the combat model itself, but are rather executed in a preprocessing mode before the combat model is run. Some agencies maintain libraries of aggregated process coefficients corresponding to various combat scenarios so that appropriate values can be selected without always having to repeat the preprocessing runs (and the corresponding high-resolution simulation runs in the case of calibration). In this case, finding the library entry that most closely matches a particular small unit battle that develops during the aggregated simulation (or determining that none are acceptable matches) becomes an interesting problem.

CHAPTER 2

AN AGGREGATED COMBAT MODEL CHECKLIST

2.1 – Content of a Typical Aggregated Model

This chapter provides a brief survey of the content of a typical large-scale ground and air combat model. We have found this checklist to be useful when trying to learn the structure of a new combat model. A full understanding of such a model requires knowledge of numerous details of unit representation, battlefield representation, and combat process representation.

The chapter contains the following sections corresponding to important parts of large-scale combat models:

- Aggregation of Units Below the Resolution Limit
- Attributes of Units Above the Resolution Limit
- Battlefield Representation
- Time Advance Mechanism
- Command and Control Processes
- Movement Processes
- Intelligence and Target Acquisition Processes
- Engagement Air Attrition Processes
- Air Allocation and Engagement Processes
- Logistics Processes

For some of these areas we will provide examples in this chapter from actual combat simulations. Other areas that require extended discussions will be considered in individual chapters in the remainder of the book.

2.2 – Aggregation of Units Below the Resolution Limit

The most significant feature of an aggregated model is the representation used for combatants and units that are too small to be modeled as individual simulation entities. Small units that are below the resolution limit of the model do not have an independent existence in an aggregated simulation. Generally such small units are modeled by assigning average values to some of the attributes of the aggregated unit that contains them.

As indicated in Chapter 1, aggregation always involves loss of information. More information is lost with a homogeneous aggregation than with a heterogeneous aggregation. Examples of typical entity aggregations were presented in Section 1.3 and will not be repeated here.

When approaching an aggregated model, the following questions provide a guide to understanding its aggregation pattern:

1. What are the largest and smallest units that are explicitly represented as distinct entities in the model?
2. What is the method for representing force structure below the smallest command level that is explicitly modeled?
3. What characteristics of individual combatants are maintained in the attributes of the aggregated units?
4. If a heterogeneous aggregation is used, how many different weapon type categories are represented?

2.3 – Attributes of Units Above the Resolution Limit

Units that are large enough to be represented as individual simulation entities have an independent existence in an aggregated simulation. Each such unit will possess its own vector of attribute variables that describe its own unique status, capabilities, activity, and perception of the battlefield. The list of entity attributes in a simulation is a good first indicator of the functions represented and the degree of detail in the representation.

Some attributes that are commonly found in unit descriptions are:

- Unit Type – Various kinds and sizes of combat and noncombat units,
- Location – Information about the location, orientation, movement path, movement speed, and final objective; for air units also the location of their home airbase,
- Mission – Mission category, combat posture category, readiness state,
- Command organization – Identification of superior and subordinate units for information and order routing,
- Authorized unit composition – List of numbers of combatants in a full strength unit (by weapon system type for a heterogeneous aggregation),
- Actual unit composition – List of surviving numbers of combatants in the unit (by weapon system type for a heterogeneous aggregation),
- Logistics state – Available ammunition, POL, and other supply categories; also identification of the logistics supply depot to be used by this unit,
- State of knowledge – Description of unit’s perception of friendly and enemy forces, terrain, and obstacles.

2.4 – Battlefield Representation

The major function of a battlefield terrain model in an aggregated simulation is to influence the maneuver plan and the mobility of ground units. The detailed terrain profile representation and line-of-sight computations of high-resolution simulations have no place in an aggregated model, but terrain characteristics may influence the aggregated target acquisition process model. The weather state is particularly important because of its influence on the planning and outcomes of air missions.

Chapter 3 of this text is devoted to the examination of battlefield representations and their influence on combat process models, so in this section we will only list some questions that can help in the understanding of an aggregated battlefield model:

1. What terrain characteristics are represented and how are they modeled?
2. How is the battlefield partitioned (if at all) to represent the sectors of responsibility for various units?
3. Does the model have a FEBA orientation?
4. How does the terrain influence combat processes such as choice of movement path, movement along a path, target acquisition, and attrition?
5. Do combat processes influence the terrain (for example, can a bridge or a road be destroyed by artillery fire)?
6. How are weather, visibility, and obscurants represented, and how do they influence combat processes?
7. How are rear areas, deep interdiction targets, and remote airfields represented?

2.5 – Time Advance Mechanism

Large-scale aggregated combat simulations are usually dynamic models that explicitly keep track of a simulated clock. State variables of the model are updated to represent passage of battle time as the simulated clock advances. One important characteristic of a combat model is the simulation mechanism that is used to represent and to advance the battle clock.

Fixed Time Step

The fixed time step method for managing the simulation clock is particularly appropriate for highly aggregated large-scale combat models. Since aggregation causes us to lose track of time-critical event sequences, little is lost by stepping through time with rather large increments.

A daily update cycle is often used in theater level models. Several time step models are listed below with the time step increment indicated. Although the time step increment is a user input in some of these models, the available process data usually determines the increment that is used. For such models we list the typical increment.

The ATLAS theater model has a daily update cycle.

The IDAGAM theater model typically uses a daily update cycle, but longer time steps can be specified by the input data.

The FORCEM theater model is a fixed time step simulation with a time interval of twelve hours.

Multiple Nested Time Steps

Several large-scale combat models use a refinement of the fixed time step method in which several nested simulation clocks are maintained. Each higher level clock has a time interval that is an integer multiple of the next lower clock, so the timing mechanism can be implemented on the computer by a simple nested loop structure.

Each state variable update in such a model is assigned to one of the simulation clocks. This assignment determines how frequently that update is performed.

The FOURCE division level model has two clock frequencies. Movement and direct fire attrition are updated every ten seconds, and all other model processes are updated every minute of simulated battle time.

The CEM theater model uses four update intervals, one for each echelon above brigade. Division updates occur every twelve hours; corps – every day; army – every two days; theater – every four days.

The VECTOR-2 theater combat model maintains eight simulation clocks in a nested loop structure. The time step interval for the outermost clock is typically 24 hours. This clock is used to update theater planning and force allocations. The fastest clock in the nested loops has a typical time step interval of 3.75 minutes and is used for updating air movement and air combat attrition. The remaining clocks have intermediate time step intervals that are used to time combat functions such as intelligence processing, fire support allocation, situation assessment, unengaged force movement, and maneuver unit combat outcomes.

Event Scheduling

Several modern large-scale simulations have taken advantage of the timing flexibility made possible in the event scheduling approach to simulation. Within the event scheduling mechanism it is not unusual to find some combat process descriptions for which the updates are scheduled to occur at fixed time intervals thus emulating the fixed time step approach for these processes.

The COSAGE division model is an event sequenced simulation using numerous event routines as well as process oriented control structures.

The ICOR corps model operates using an event sequenced timing mechanism. Within the event scheduling mechanism, some combat processes are described using fixed time increments. For example, ground attrition is updated every five minutes of battle time.

ICOR requires man-in-the-loop decision making by human players. The combat model is stopped while orders are being formulated and input. This typically occurs every one to two hours of battle time. After orders are received, the event sequence is restarted.

The COMMANDER theater model is also event driven with the capability for man-in-the-loop command inputs.

Synchronized to Real Time

Combat simulations that attempt to create a decision making environment for human players are often synchronized to some multiple of real time. A good example is the JTLS theater simulation that involves multiple decision makers in a human computer interactive combat simulation.

The combat model advances the simulation clock time at some multiple of real time (chosen by the game director). The players must input orders to their combat units to set missions, objectives, and activities. While the players are formulating new orders the combat simulation continues to run executing the most recently received commands. A fast clock rate will place considerable time pressure on the players. When enemy units encounter each other, the simulation resolves the combat outcomes and provides situation reports back to the players.

2.6 – Command and Control Processes

Modeling of command and control processes is particularly important for large-scale combat models. The higher echelon headquarters represented in the large-scale models are responsible for intelligence collection and fusion, force allocation, and planning activities that are not simulated in a typical high-resolution task force model.

There are two main approaches to the command and control decision function in aggregated combat models:

1. Automated decision models perform the command and control functions by computer subroutines completely within the simulation program.
2. Man-in-the-loop decision models insert human players into the simulation process to make command and control decisions.

Within each category there are numerous variations, and some models combine the two approaches – using human decision makers for some decisions and leaving others to the computer.

Automated decision models are not simple to develop. There is a long history of automated combat simulations that make stupid tactical blunders. While the man-in-the-loop approach hopefully improves the quality of decisions made, it also markedly increases the cost and duration of the simulation runs. In addition, the use of human players inserts a high variance stochastic element into a simulation that might otherwise be deterministic.

We will consider command and control process models in greater detail in a later chapter. The following list of questions is useful in examining the command and control process models of a large-scale simulation.

1. What decisions are represented at each echelon of the force?
2. What general methodology is used to simulate decision making?
3. Does the decision process try to model the staff information processing and planning procedures by which real C3 decisions are made, or only the resulting decisions?
4. For each decision process, what situation status variables are input factors and what is the influence of the decision on the continuing execution of the model?
5. Are the situation status variables considered in a decision fixed by the computer program or are they determined by user input data?
6. How flexible is the process of developing and changing decision rules both before and during the execution of the simulation?
7. For human interactive decisions, does the computer simulation stop while the decisions are formulated, or does it continue to execute thus placing time pressure on the players?
8. What communications processes are represented at each echelon in the model?
9. How are communications processes modeled (errors, congestion, delays, electronic warfare, and the results of either receiving or not receiving a message)?

2.7 – Movement Processes

Movement processes are part of the basic bookkeeping in any combat model. For an aggregated model the entities that move are the aggregated units. Two aspects of the movement process are particularly interesting for the combat modeler:

1. Where the unit should go – The choice of movement objective and the choice of a route to the objective are generally either preplanned as a part of the scenario or else an output from the command and control model.
2. How fast the unit can move – The unit speed is typically determined as a function of the terrain conditions and of the combat situation.

Both of these aspects of the movement process will be discussed in the chapter on battlefield representation since they are so closely tied to the terrain characteristics.

Some questions useful in examining a model's movement processes are:

1. What are the model entities that are moved?
2. How are the destination and the movement path determined?
3. What factors are considered in selecting movement paths-terrain? Known or suspected enemy units?
4. Is movement in arbitrary directions allowed or is the movement path constrained to be within FEBA oriented battlefield sectors?
5. When is a unit's location updated?
6. How is movement speed determined, both for units in combat and for units that are not engaged?

2.8 – Intelligence and Target Acquisition Processes

The acquisition of individual targets for the direct fire main battle is generally combined with the aggregated attrition process model in large-scale combat simulations. We will discuss it in the chapters that consider these attrition models.

Echelons of division and higher are also interested in long range acquisition of unengaged enemy units and other critical intelligence information. Large-scale combat simulations model the acquisition and processing of intelligence information as an input to the command and control module and as a source of interdiction targets for long range strike forces.

The following questions are useful in examining the intelligence processes of a large-scale combat model:

1. What sources of intelligence data are represented?
2. Are collection assets modeled explicitly; Are they subject to attrition?
3. How does the model represent the content of a piece of intelligence information?
4. Is the collected intelligence subject to error, incompleteness, or time delay?
5. How is fusion of raw intelligence data represented?
6. How does the model represent the perceived view of the battle situation; at what unit levels do such perceptions exist?

7. How does processed intelligence data affect the perception of the battle situation?
8. How is dissemination of intelligence data represented; who receives reports?

2.9 – Engagement and Attrition Processes

Engagement and attrition processes are at the heart of any combat model. For large-scale aggregated models there are two different categories of attrition representations:

1. For maneuver unit combat, both direct and indirect fire, the firers and targets are so numerous on the large-scale battlefield that they must be aggregated into larger simulated units. Engagement and attrition processes must be represented at the unit-vs-unit level using aggregated process descriptions.
2. For significant rare assets (such as a cruise missile installation) that are represented explicitly as separate simulation entities, the modeling of engagements and attrition can be similar to that for high-resolution models.

Attrition processes are the most developed of the aggregated models, and several interesting mathematical approaches have been used in large-scale combat simulations. We will devote several chapters to the description of aggregated attrition assessment models.

Some questions useful when investigating the engagement and attrition processes of a large-scale model are:

1. How does the model initiate engagements and determine which units will be involved in an engagement?
2. What is the mechanism that terminates an engagement?
3. What basic techniques are used to assess the results of an engagement (for each of the categories – ground to ground direct fire, ground to ground indirect fire, ground to air, air to ground, air to air)?
4. How is fire distribution among the individual systems in an aggregated unit modeled?
5. What kill-categories are represented?
6. What firing effects are represented on non-combatant systems and targets such as airfields, bridges, and command posts?
7. Are chemical and nuclear weapons represented?
8. Is the firing process limited by ammunition constraints?

2.10 – Air Allocation and Engagement Processes

The modeling of tactical air operations is another aspect of large-scale combat modeling not found in high-resolution task force models. The smaller-scale high-resolution models may play close air support sorties, but the process that generates and allocates those sorties is above the task force echelon.

The large-scale combat models considered in this text have been developed primarily within the ground combat community. Their representations of air combat show a high degree of variability both in the amount of detail and in what aspects of air combat are considered important enough to include. Only one of the models we have listed

(COMMANDER) has an air combat representation that is significantly better than its ground combat model.

In the chapter on air combat modeling we will investigate some of the details of the air components in large-scale combat simulations. The remainder of this section lists some questions useful in examining an air combat simulation:

1. What air bases, air units, aircraft, and air weapons are represented; what is the aggregation level?
2. What air mission categories are represented?
3. What is the decision logic for allocating air assets to air missions?
4. Are missions flown against individual targets or just allocated to target classes?
5. Are individual air sorties simulated in detail, or are they aggregated into mission categories?
6. How are the outcomes of air missions evaluated (air to ground, ground to air, and air to air)?
7. How are aircraft maintenance, turnaround, and RAM failures modeled?

2.11 – Logistics Processes

Logistics processes are particularly important in the long duration campaigns that are simulated in large-scale theater combat models. High-resolution task force battles seldom last long enough for significant logistics activity; thus high-resolution models often ignore logistics except for counting ammunition consumed. Large-scale models are the appropriate place to consider questions of resupply, recovery, repair, transportation, and long term sustainability of the combat effort.

Logistics activities are relatively simple to model because they are mostly one-sided. Logistics installations may be targets for enemy action, but they are passive targets; any combat capabilities they may possess are treated by other parts of the simulation such as an air defense module.

The allocation of available supplies to combat units is an important aspect of the logistics activity that we will consider along with other allocations in the command and control chapter.

The following checklist may be helpful in investigating a new logistics model:

1. What logistics units are represented; how far back from the combat zone?
2. What classes of supplies are represented?
3. At what unit level is each class of supplies monitored?
4. At what levels is consumption monitored and how is it computed?
5. At what unit level is resupply modeled; what triggers a resupply?
6. Are resupply activities explicitly modeled; are they subject to attrition?
7. How are supply vehicle reload time and RAM factors modeled?
8. Are recovery and repair processes modeled?
9. What logistics decision processes are represented; how are the decisions made?
10. What is the effect on the rest of the model of shortages in each of the supply classes represented?

CHAPTER 3

BATTLEFIELD AND MOVEMENT PROCESS MODELS

3.1 – Battlefield Representation for Aggregated Models

Battlefield representations for large-scale aggregated models are considerably different from those for high-resolution simulations. The huge battlefield areas considered in theater level models make a detailed representation of terrain contours impractical; the 100 meter terrain resolution typical in task force models would yield an unmanageably large terrain data set.

Fortunately, there is no need for such terrain detail in aggregated combat models. Since individual combatants are not represented, there is no need to compute line of sight between them, so the terrain contours are not needed.

The primary purpose of the terrain model in a large-scale aggregated combat simulation is to represent trafficability for the movement process model. Terrain features determine where avenues of advance will be located, where units can safely defend, and how fast units can move across the battlefield.

Aggregated models also use features of the terrain to influence the aggregated process models for target acquisition and combat attrition, but the features are represented as averages over large areas of the battlefield rather than actual values along an observer to target line of sight or at a specific target location.

Regular Grid and Combat Sector Terrain Models

There are two major types of battlefield representation that are widely used in aggregated combat simulations:

1. Regular grid terrain models overlay a map of the battlefield area with a regular grid of terrain cells. The cells are usually squares or hexagons (in one case, rectangles). Each terrain cell in the regular array has attributes that describe the characteristics of the terrain within the cell.
2. Combat sector terrain models divide the battlefield into sectors of responsibility for the aggregated combat units. These sectors are irregularly shaped and of variable size; the size and shape of the sectors depends on the forces involved and on the battle plan for using those forces. Within each sector, the model stores terrain attributes for that part of the battlefield.

Combat sector terrain models combine the combat scenario with the terrain representation. Once a unit is allocated into a sector, it is typically constrained to remain within the sector and its movement is along the sector axis, perpendicular to the front line or FEBA. Sector terrain models are most useful for simulating European style FEBA battles. To simulate a different scenario on the same terrain might require that the sectors be redefined (thus changing the entire terrain data set) or might only require that different forces be allocated within the existing sector boundaries.

Regular grid terrain models are scenario-free. The terrain organization does not constrain the choice of movement paths.

This is an advantage since more complex forms of battle can be represented such as penetrations, encirclements, and sparsely populated battlefields. However, there is a corresponding disadvantage. Since the battle plan is not preplanned in the terrain representation, it must be determined in some other way. It is not easy to write computer routines that will take advantage of terrain features in forming a coherent battle maneuver plan and adapting it to changing circumstances as the battle progresses.

In the next two sections of this chapter we will examine terrain models of each of the two basic types that have been used in actual large-scale combat simulations.

Statistical Terrain Models

Another sort of terrain model, which is used in the COSAGE division level combat model, is a statistical terrain model. COSAGE does not contain a terrain map of the battlefield area, but rather has parameter sets that describe several typical types of terrain. Each terrain type defines parameters for line of sight probabilities, movement rate, and cover from weapons effects.

At various points in the simulation process (for example, when a small unit ground battle is initiated, when artillery effects are assessed, or when a unit moves) a random draw is used to determine which of the terrain types to use. The corresponding parameters then influence the combat process routines for the evaluation at that point in the simulation.

Network Models

A final type of battlefield model that is particularly useful for modeling the movement of forces and supplies from the rear areas to the front lines is based on a network. Nodes of the network represent physical locations on the battlefield map. Arcs of the network represent possible movement paths between nodes such as roads, trails, or likely cross country routes.

Each node and arc of the network has attributes that determine its transport capacity and the speed of combat units moving through that node or along that arc.

A major advantage of the network model is that it allows efficient network optimization algorithms to be applied to the problem of route selection for moving combat units.

An example of a model that uses a network structure is the VECTOR-3 theater combat simulation. This model represents the battlefield using a combination of a combat sector model (derived from VECTOR-2) with a superimposed transportation network.

3.2 – Regular Grid Terrain Models

Regular grid terrain models for aggregated simulations have the same battlefield geometry as high-resolution grid models. However, the grid sizes are much larger (1 to 20 kilometers) and the information stored as grid attributes is highly aggregated. The terrain cell size is frequently chosen to correspond to the approximate size of the basic aggregated combat unit in the model.

Since the grid terrain representations are scenario-free, maneuver instructions for the aggregated units must be provided in other input data or from the command and control module. This terrain form is particularly suited to human interactive players.

In this section we will examine the terrain attributes of several actual combat simulations that use regular grid terrain.

The FOURCE Model

FOURCE represents combat among battalions on a division sized battlefield. The terrain cells are 1km by 3km rectangles with the long dimension parallel to the axis of advance. This shape was chosen because lateral movement to position forces for the main attack is an important aspect of FOURCE's typical battle plan.

Within each grid rectangle the terrain is classified by:

1. relief – several categories of terrain roughness,
2. vegetation – the fraction of the cell covered by forests,
3. axial roads – the kinds of roads running the length of the rectangle, and
4. lateral roads – the kinds of roads across the rectangle.

These terrain attributes influence unit movement speed, intelligence collection, and probabilities of line of sight for the direct fire battle.

The ICOR Model

ICOR represents combat among battalions on a corps sized battlefield. Terrain grid cells are hexagonal with a 3.57km diameter.

Within each hexagonal cell (hex) ICOR stores attributes for:

1. terrain roughness,
2. urbanization, and
3. forestation.

Each hex side has trafficability attributes that represent obstacles such as rivers along the side and roads that cross the side into an adjacent hex.

The ICOR terrain attributes influence movement speed, target acquisition, and combat attrition. Maneuver objectives are determined by the players who interact with the simulation.

The JTLS Model

JTLS represents combat among divisions on a theater sized battlefield. The terrain cells are hexagonal with a typical diameter of about 20km. Each terrain hex has attributes to represent:

1. terrain elevation,
2. trafficability within the hex,
3. trafficability at the boundaries between hexes.

The terrain attributes influence the unit movement speed. Terrain features such as bridges between hexes may be destroyed by air or artillery attack, thus changing the trafficability until repairs are made. The terrain elevation limits areas in which helicopters can operate.

3.3 – Combat Sector Terrain Models

Combat sector terrain models organize the battlefield according to a predetermined battle plan. They are therefore scenario dependent; successful simulation of a scenario is highly dependent on the proper choice of sectors and the proper allocation of forces to the sectors.

Combat sector models divide the battlefield into a number of roughly parallel terrain strips usually called sectors. Each sector extends from the blue rear area, across the direct fire battle area, and to the red rear area. When drawn on a map, the sectors may have varying width and curved side boundaries, but the models measure distance along a (curved) coordinate parallel to the sector, and most unit movements are along this coordinate. Each sector is crossed by a pair of lines denoting the forward line of troops (FLOT or FEBA for "forward edge of the battle area") for each side where the enemy forces meet in battle.

Terrain within a sector is described by dividing the sector terrain strip into short segments or arenas that have the same width as the sector and variable length. Within each such segment the terrain attributes are considered homogeneous.

Once the combat sectors are defined, the tactical command and control of the force is relatively simple. Allocations of units and supplies flow from the rear areas into their chosen sector and gradually move along the sector to the combat front. Combat outcomes may cause the location of the front line troops (and thus of the FEBA) to move in the direction of a successful attack. In most sector models unit interactions across sector boundaries are very limited; typically combat units cannot move or fight across a sector boundary when they are close to the FEBA.

Within this general outline, there are variations in terminology, in the details of how the sectors are laid out, and in the details of how sector boundaries influence combat processes, In the rest of this section we will examine a few typical sector battlefield models.

The ATLAS Model

ATLAS represents combat among divisions on a theater battlefield. The sectors that define avenues of advance may vary in width from a division to a corps front. Combat activities within a sector are completely independent of any other sector (once forces are allocated to the sectors). There are no interactions across sector boundaries, no considerations of flank exposure, and no logic to maintain FEBA integrity from sector to sector.

Each sector is divided into terrain segments (see Figure 3.3.1 for the geometry). The terrain within a segment is described by three terrain types:

1. open rolling terrain, good armor battlefields,
2. marginal armor terrain, and
3. mountainous or thickly wooded terrain.

For each of the three types, the terrain is coded for the presence or absence of manmade barriers or defensive positions. Thus there are six different kinds of terrain in ATLAS. These influence combat posture, combat attrition, and movement rates for forces in combat.

Within each sector, running from an entry port in the rear area to the front lines, are supply lines of communication (LOC) that connect supply point nodes spaced about a day's travel apart. Airbases and SAM sites are also placed at these nodes.

The IDAGAM Model

IDAGAM represents combat among divisions on a theater-sized battlefield. The combat sector, FEBA, and terrain segment structure is essentially the same as in ATLAS. Each terrain segment (called "intervals" in IDAGAM) is characterized by one of three trafficability codes (as in ATLAS) and by a combat posture code describing the type of defensive posture appropriate within the segment.

In addition to the sector structure, IDAGAM adds a more detailed representation of the rear areas of the battlefield. Sectors are grouped into regions away from the FEBA, and the regions are combined into one theater communications zone at the back of the battlefield (see Figure 3.3.2 for the geometry). Airbases are located in the front and rear of the regions and in the communications zone. Allocations are made from the communications zone to the regions and then from the regions to the combat sectors.

The combat and FEBA movement processes in IDAGAM are computed independently sector by sector. Then the FEBA location is adjusted to represent defender withdrawal in any sector where the front-to-flank exposure ratio cannot be supported.

The VECTOR-2 Model

VECTOR-2 represents combat among battalions on a theater battlefield. The battlefield representation consists of roughly parallel sectors as in the other models considered here. Since the combat resolution is at battalion level, each sector can be further subdivided into corridors that have boundaries roughly parallel to the sector boundaries. A corridor has width appropriate for a battalion defense. Then each corridor is divided into a sequence of combat arenas (segments) within which the battalion units are located.

The terrain in each arena is characterized by six intervisibility levels and six trafficability levels, plus codes for special features such as rivers or urban areas. In addition, the arena description includes the number of defensive positions that will be occupied in succession by defending forces as they move back through the arena.

Rear areas are defined by up to four bands of zones that maintain a constant distance from the FEBA. Rear area assets of various sorts are located in particular zones to maintain their location relative to the front line battalions.

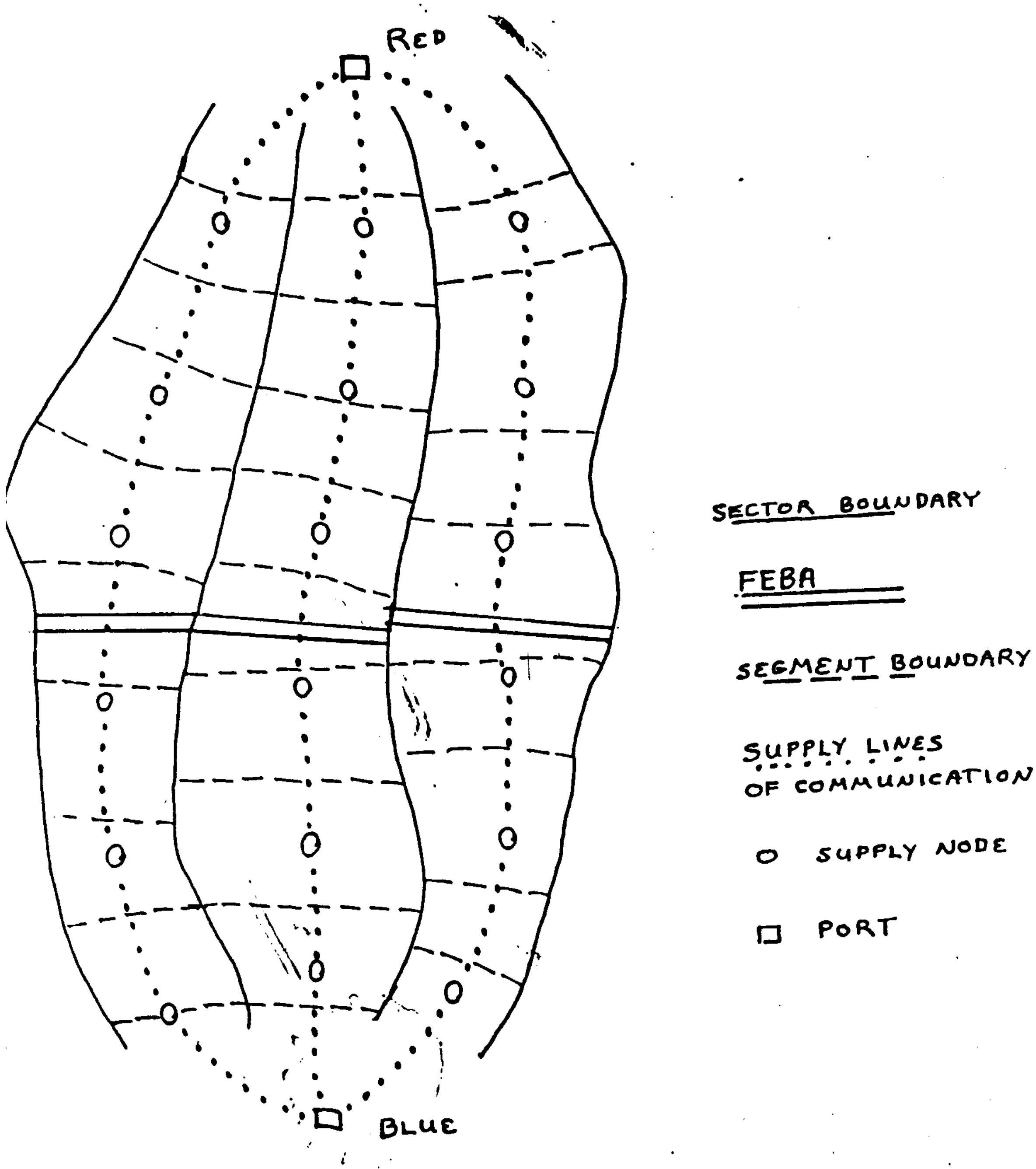


Figure 3.3.1 – ATLAS Battlefield Structure

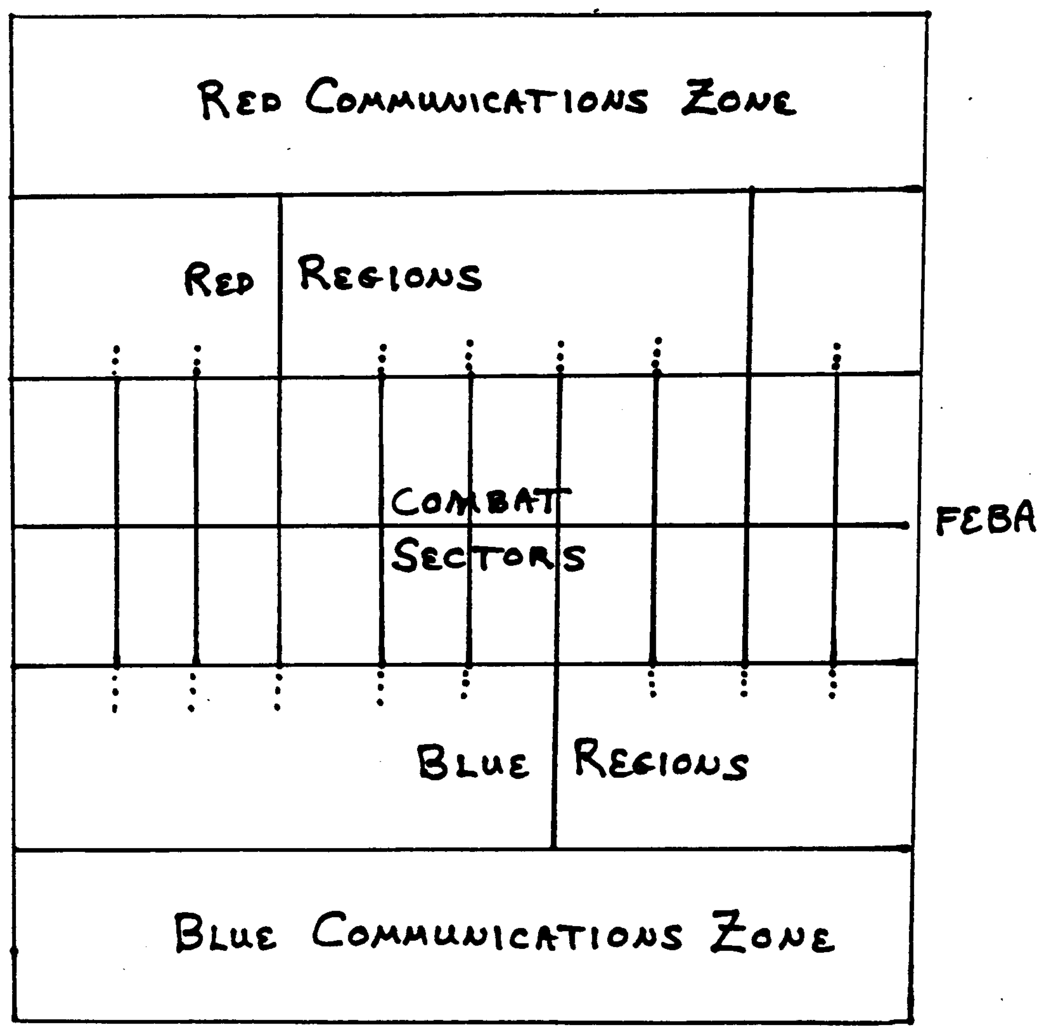


Figure 3.3.2 – IDAGAM Battlefield Structure

3.4 – Movement Models for Aggregated Units

Movement process models for large-scale aggregated combat simulations are responsible for determining three aspects of unit movement:

1. the movement destination for each moving unit,
2. the route that will be traveled to the destination, and
3. the unit speed at each point along the movement route.

Movement Destination and Route

The choice of unit destinations is largely determined by the battle scenario. Scenario information is transmitted to the simulation in several ways.

1. Simulations that use a combat sector terrain model have the scenario built into the battlefield representation. Each unit is assigned to a particular combat sector. Then the destination for maneuver units is the FEBA within their sector, and the destination for other units (such as artillery) is offset from the FEBA in the sector rear. Routes to the destination are generally straight line paths through the combat sector. At the FEBA, model decision logic determines whether a unit advances or withdraws according to its combat posture assignment and combat outcomes.
2. Other models have the movement destinations and routes for each unit as a part of the model input data. The movement process then reduces to a simple matter of updating unit position as time passes.

Both of these methods of determining destinations and routes have the disadvantage that they are not very responsive to changing battle conditions. Users frequently have to redefine the input data sets several times before they can achieve realistic looking maneuvers in the battle outcomes.

3. Human interactive models require the player to provide movement orders. Reasonable outcomes are easier to achieve since the player can give new orders to any unit at any time.
4. The most ambitious movement models are those that determine movement objectives during the simulated battle as part of a simulated command and control procedure. Our current models for automated command and control of forces are not very good, but a considerable amount of effort is being devoted to improving them. Further discussion of C3 models will be presented in a later chapter.

Unit Movement Speed

All large-scale models compute unit movement speeds using a variation of one basic simplistic idea. The user must provide unit movement speeds under a variety of conditions as part of the model input data. The simulation then determines the conditions for each movement increment and looks up the appropriate speed.

Implementations of this idea vary in the parameters that are allowed to influence speed and in which parameters are independent of others. Typical movement speed arrays have the following sorts of indices:

1. unit type,
2. terrain trafficability,
3. obstacles and minefields,
4. whether in combat or in an administrative move,
5. combat posture (if in combat), and
6. opposing combat strengths often measured by a force ratio (if in combat).

In the rest of this section we will discuss the movement process implementations in several actual large-scale combat models.

The ATLAS Model

All movement destinations and routes are implicit in the ATLAS combat sector battlefield and in the allocation rules that assign units and supplies to combat sectors. Administrative moves of reinforcing units and movement of supplies in ATLAS rear areas are along the sector lines of communication. The ground movement speed is one supply node per day, and air transport can also be used if it is available.

Movement at the FEBA, if any, is in the direction of the attacking force. The rate of FEBA movement is determined as a function of:

1. attacker to defender force ratio (10 values from 0.5 to 0.8),
2. terrain type including manmade barriers (6 types),
3. defender combat posture (7 types), and
4. attacker mobility (infantry or armor).

Movement rates to cover all possible combinations of these factors have been derived from real division battles in WWII and Korea. The data has been smoothed, augmented to represent new postures, and modified for various purposes over the years. The 1971 Army Model Review Committee report has an interesting discussion of the movement rate data sources and transformations.

Movement speed within a terrain segment is determined as a weighted average of the infantry and armor speeds with weights corresponding to the relative composition of the attacking forces. When a day's combat extends over several terrain segments, ATLAS evaluates FEBA movement (and casualties) one segment at a time.

Air movement in ATLAS is not explicitly modeled. Only the results of aggregated classes of air missions are computed and not the detailed progress of individual missions.

The IDAGAM Model

IDAGAM uses essentially the same movement rate representation as ATLAS for FEBA movement as a function of force ratio, terrain type, and defender posture.

The treatment of attacker mobility is extended to allow the user to choose from several options for each battle:

1. independent of attacker division types,
2. use speed of slowest attacking division in sector,
3. use speed of fastest attacking division in sector,
4. average speeds of attacking divisions in sector.

IDAGAM also adds a factor to account for how air power is used in support of the ground battle. There are three categories:

1. attacker's air advantage greater than his ground advantage,
2. attacker's air advantage less than his ground advantage,
3. attacker does not have ground advantage.

The JTLS Model

The real-time, human interactive JTLS simulation takes its unit destinations from player orders. Movement routes are optimized automatically by the simulation to achieve the minimum travel time. The optimization considers only the speed of movement so the player must stay alert to cases where movement paths will encounter enemy forces. The route selection can also be bypassed in favor of straight line paths.

Movement in JTLS is only possible into terrain hexes that do not contain enemy units. A unit that tries to move into a hex controlled by the enemy must stop and fight until the enemy is forced to withdraw. When movement is possible, the movement speed is determined by multiplying the unit's basic speed by a speed factor that depends on the terrain in the unit's hex.

Air movement in JTLS is explicitly modeled for each individual air mission. Objectives are chosen by player orders, and the routes can be optimized for minimum time subject to avoiding air defenses. Air speed is an attribute of the aircraft flying the mission.

The FOURCE Model

The division level FOURCE model is oriented around two major maneuver alternatives: a main attack by second echelon forces in the southern sector or in the northern sector. Maneuver objectives for the attacking force are determined by the C3 module to implement one of these alternatives. The main emphasis of the entire model is to determine whether the defender C3 system can detect the location of the main attack in time to defend against it.

Thus the movement objectives for both sides are determined by the command and control routines of the model. The highly simplified nature of the scenario contributes to making the maneuver decision rules feasible in automated form.

Maneuver routes are always straight line paths to the chosen objectives.

Unit (battalion) movement speed is determined as:

```
SPEED = BASIC_UNIT_SPEED   * (Factor for terrain relief and vegetation)
                           * (Factor for roads in the terrain cell)
                           * (Factor for being engaged by air or artillery)
                           * (Factor if defender is in delay posture)
                           * (Factor for combat power ratio)
```

Values of the factors are part of the model's input data set. In addition, constraints are imposed on the movement speed to account for operating in enemy (thus unfamiliar) territory and for ensuring unit integrity.

Unit location is updated every 18 seconds for maneuver battalions and every minute for other units such as artillery and higher headquarters units.

3.5 – Environmental Models

The treatment of weather and other environmental effects is extremely varied in large-scale combat models. Older models such as ATLAS, IDAGAM, and CEM represent essentially no environmental effects. For such models, the only way weather, visibility, cloud cover, or obscuration can affect the battle is if the aggregated combat process data sets are derived from observations under adverse battle conditions. Even then the conditions are uniform over the entire battlefield and over the duration of the battle.

Since both ATLAS and IDAGAM operate on a daily update cycle, they are also incapable of representing the difference between day and night fighting. CEM also seems not to represent the difference between day and night even though its 12 hour division cycle and 12 hour air assessment cycle would make such a distinction possible.

Several other models explicitly represent different environmental states and use them to influence the computation of combat process outcomes.

The COSAGE Model

COSAGE allows the user to input two separate process data sets for day versus night operations. The model selects the proper data set to use based on the simulation clock. Activities influenced by the day/night time period include target detection Probability, target location, lethality, use of smoke or illumination rounds, and movement rates.

COSAGE also models visibility conditions in considerable detail. Target acquisition times in the direct fire battle are influenced by smoke, dust, illumination, and weather conditions through the use of the Night Vision and Electro-Optical Laboratory target detection model. Weather conditions are assumed to influence the entire division level battlefield and can be changed by user input as battle time advances.

The VECTOR-2 Model

The theater level VECTOR-2 model allows for environmental conditions to be varied through user input for each sector and each hour of the battle. The conditions defined are:

1. four ground to ground visibility categories,
2. four ground to air and air to ground visibility categories,
3. four air to air visibility categories,
4. four ground trafficability categories (representing surface conditions such as mud or snow), and
5. four air trafficability categories (representing wind speed and direction).

These environment codes are combined with the battlefield terrain codes in the model to influence combat processes such as movement and target acquisition.

As is often the case in combat models, the values of the environmental codes are not assigned any meanings by the computer program, but are rather used as indices into arrays of user supplied process coefficients. The user can assign any desired meanings to the various code values and must make sure that consistent meanings are maintained in all data arrays that use these codes as subscripts.

The COMMANDER Model

The treatment of the environment in COMMANDER is interesting because of the resolution with which the air battle is modeled.

COMMANDER models the weather over the battlefield using a three dimensional data base. For each 25nm square grid of battlefield area, fifteen altitude bands are represented.

Each cell of the three dimensional array contains weather parameters that influence the process models for air to ground target acquisition in air strike and air reconnaissance missions.

CHAPTER 4

FORCE RATIO ATTRITION MODELS

4.1 – Introduction to Attrition Process Models

Combat attrition is the single aspect of combat modeling that has received the most attention over the years. It is the only combat process for which well developed mathematical "theories" might be said to exist.

In this and the next two chapters we will investigate the treatment of attrition process models for the ground battle in large-scale aggregated combat simulations (air battle attrition algorithms will be considered separately in the chapter on air models). Our presentation of the theory is biased toward those results that are required to understand the applications in actual computer simulations. For further mathematical developments the reader is referred to the references, especially the comprehensive treatment of Lanchester models in reference [4.1].

The Scope of Aggregated Attrition Models

Aggregated attrition models describe the results of engagements among aggregated combat units. Since individual combatants are not represented in these units, details of one-on-one engagements are not simulated, but rather the attrition process models consider average results.

Individual combatants are aggregated into combat units ranging in size from companies to divisions. The contributions of the individuals are averaged together over the entire unit (for homogeneous models) or over weapon system classes within the unit (for heterogeneous models).

Discrete activities such as target acquisition, fire allocation, and lethality assessment are aggregated together into a single process called attrition. Direct fire, indirect fire, and close air support are sometimes further aggregated together, but are considered separately in other models. The attrition model is also often the natural place to compute movement of the FEBA since it depends on the strength of the forces and the casualties suffered by both sides in the battle.

Attrition is also averaged over periods of time. The models considered in this book have attrition update intervals ranging from 10 seconds (for the division level FOURCE model) to 24 hours (for several theater level models).

Two Basic Types of Aggregated Attrition Models

Aggregated attrition process models can be categorized into two basic types that correspond to the two basic entity aggregation patterns – homogeneous and heterogeneous. Older combat simulations tend to use the homogeneous models while more recent simulations apply – heterogeneous attrition models.

1. In a homogeneous attrition model, combat attrition is assessed against a scalar measure of the unit's combat power. Sometimes this scalar measure is defined as "personnel" and in other cases it is a more abstract combat power measure such as COMMANDER's "T62 tank equivalents".

Most homogeneous attrition models determine the amount of attrition by computing attacker to defender force ratios. We will consider two such models in detail in this chapter (ATLAS and IDAGAM). Another approach sometimes used applies homogeneous Lanchester equations. We will discuss Lanchester equations in chapter 5, but will not examine any actual combat models that use the homogeneous version of the equations.

2. A heterogeneous attrition model assesses combat attrition caused by weapon system classes against enemy weapon system classes within the combat units. Such models have the advantage of being able to represent characteristics of firer-target weapon type pairings, and thus provide a better view of the attrition process.

In chapter 5 we will consider heterogeneous attrition models that are based on the Lanchester equations of combat. Chapter 6 is devoted to a heterogeneous exponential attrition model, ATCAL, which is specially designed to allow calibration from high-resolution model output.

Engagement Initiation and Termination

The attrition process models in a large-scale aggregated simulation are used to compute combat outcomes for localized battles involving small parts of the total force. Before attrition can be computed, the model must determine where localized battles will be fought and which forces will be engaged in each battle.

Battles among aggregated units generally start whenever the units move within weapons range of each other. For combat sector battlefield representations the battle will include all combat units in a sector that are in the terrain segment closest to the FEBA (with some forces possibly withheld for flank protection). On regular grid battlefields, combat normally starts when enemy units occupy adjacent grids or when a unit tries to move into a grid occupied by the enemy.

The attrition process model is responsible for determining what fraction of the weapon systems in the engaged units are within range of enemy targets, what fraction of the systems acquire enemy targets, and thus what fraction of the systems in a unit actually participate in the battle.

Simulated battles continue until one side or the other withdraws from the battle area. Combat simulations typically make the decision to withdraw based on user input attrition thresholds. In human-interactive simulations, of course, the players may try to disengage at any time.

Mission and Combat Posture

Localized battles are generally categorized by mission and combat posture into several different battle types. The attrition models have separate sets of attrition coefficients corresponding to each battle type.
Unit missions are set by the scenario or by the command and control module. The combat posture (usually for the defender) is determined as a combination of the defender mission and of the defensive positions offered by the terrain.
A typical list of the resulting battle types is:

1. defense of a fortified zone,
2. defense of a prepared position,
3. defense of a hasty position,
4. meeting engagement,
5. delaying action,
6. orderly withdrawal,
7. disorganized retreat, and
8. static situation (no attacker).

As combat units reach casualty thresholds, or as the command and control model dictates, their missions may change resulting in a change to a different battle type. Thus a single battle may evolve through several battle types before it ends.

4.2 – Firepower Scores and Force Ratios

The basic idea of homogeneous force ratio attrition models is to aggregate all of the individual combatants in a unit into a single scalar measure of the unit's combat power. Then the ratio of attacker to defender combat power is used to determine the amount of combat power destroyed by the enemy in a battle.

Firepower Score, Firepower Index, and Force Ratio

The firepower score approach measures the combat power of a unit by summing the combat power values for each weapon system in the unit. Suppose that there are n different types of weapon system in a combat unit and that:

X_i = number of weapons of type i in the unit (i = 1,2,…,n). Define the firepower score for each weapon of type i to be
S_i = score value representing combat power for each type i weapon, and define the firepower index of the aggregated unit to be

$$FPI = \sum_{i=1}^{n} X_i * S_i \qquad (4.2.1)$$

Finally, suppose FPI(A) is the firepower index of the total attacking force in a battle and FPI(D) is the firepower index of the defenders. Then the force ratio is defined as:

$$FR = FPI(A) / FPI(D). \qquad (4.2.2)$$

The force ratio gives a measure of relative combat power in the battle.

There are many different terminologies for the firepower scores and indices corresponding to different ways of determining the combat power. For simplicity we will use the above definitions throughout except where we discuss the different ways of determining the scores.

The firepower index is used in homogeneous aggregated large-scale combat simulations as the primary description of what a combat unit contains. Force ratios in these models are used for several purposes such as:

determining missions and combat postures,
computing casualties for both sides in a battle,
computing FEBA movement,
determining priorities for air and artillery support,
determining priorities for resupply and reinforcement, and
determining mission success at the end of engagements.

The details are quite different from model to model, and not all simulations do all of the above computations. Our main interest here is in the attrition computations and the resulting FEBA movement. Details of how the force ratio is used in ATLAS and IDAGAM will be presented in later sections of this chapter.

Characteristics of the General Approach

There are several important limitations of the firepower score approach to attrition modeling. These derive both from the theoretical properties of the aggregation equation and also from the practical aspects of determining the data values to use.

1. The firepower index equation is additive across weapon system types. Thus it cannot represent interactions among the weapon systems on one side of the battle. Synergistic effects, where the presence of one weapon type makes another weapon type more effective, will not appear in models that base their calculations on firepower indices. Such models should not be used to assess force mix or force balance issues since 100 points of combat power is 100 points whether it derives from a balanced force or from an unbalanced force.
2. The firepower index equation is linear in the number of weapons (X_i) of each type. Thus it cannot represent the minimum unit size required for effectiveness, nor can it show diminishing returns as the number of any single system becomes unmanageably large.
3. The firepower index equation loses track of the types of weapons in the aggregated combat units. Thus the user must be careful not to set up ridiculous battles involving unnatural combinations of enemies.

4. As we will see in the next section, there is no general agreement on how to determine the firepower score values (S_i). There is also considerable uneasiness about the data for the process models that relate the force ratios to attrition and FEBA movement.

Nevertheless, combat simulations that rely on the force ratio approach are used regularly for important analyses.

<u>Determining Firepower Score Values</u>

Several approaches to determining the numeric values for firepower scores have been tried over the years. Proposals for new score computation methods have suggested numerous weapon parameters to be folded into the score values. None of the methods are entirely satisfactory because simple addition of scalar scores cannot capture the variety of characteristics and interactions in a complex combat unit.

More extensive discussion of some of the earlier scoring systems and the reasoning that produced them can be found in reference [4.2].

1. <u>Measures of perceived combat value</u> – Systems of weighting weapon systems according to perceived combat power have their historical base in procedures that have used for situation assessment and planning for many years. The score values used were derived from military experience and judgement.
2. <u>Measures of historical combat performance</u> – Some portion of the score values in the 1950s was derived from WWII and Korean War data. For example, data about the number of casualties caused by small arms fire and by artillery fire was used to relate the scores for these systems. It is difficult is determine how much data was available and how it was manipulated.
3. <u>Measures of weapon firepower</u> – In the 1960s, in connection with the development of the ATLAS model, a weapon's score was determined as a measure of "firepower". For area weapons (artillery) the firepower score was defined as:

S_i = (daily ammunition expenditure) * (lethal area per round)

For point fire weapons the score was similarly defined as:

S_i = (daily ammunition expenditure) * (probability of kill) but it was not easy to relate the two classes of definitions.

The expected ammunition expenditures were highly situation dependent and the resulting firepower score values still needed judgmental adjustments.

4. <u>Measures of mission dependent firepower</u> – Problems with the firepower measures led to including some of the situational factors. In particular, separate sets of S_i values were developed for attack versus defense missions.
5. <u>Measures of multiple characteristics of the weapon system</u> – In the late 1960s and early 1970s more elaborate scoring systems were developed. They involved combining a weapon's firepower with other system characteristics such as mobility,

vulnerability, and reliability. The factors were combined using linear weighting schemes with weights determined by Delphi analysis. Acronyms for the resulting scores and indices were WEI (for weapon effectiveness index – the score for a weapon) and WUV (for weighted unit value – the aggregated unit index).

6. Measures of what a weapon can kill – A different approach developed in the early 1970s determined the score for a weapon by observing what it could accomplish in a battle. In particular, the score for a weapon was defined as proportional to the total of the scores for all the enemy systems it kills. This definition leads to a circular system of eigenvalue equations that can be solved for the weapon score values (reference 4.4]). The resulting scores are highly situation dependent and must be evaluated in the context of a battle scenario. The computational procedure is called the Potential Anti-Potential Method, The mathematics of this method will be developed in Section 4-4 and its application in the IDAGAM combat model will be discussed in Section 4-5.

Static Evaluation Models Using Force Ratios

Although we will not have the space to consider them in detail, it should be noted that some of these methods of computing a unit's combat power or value were developed primarily for static force comparisons rather than for use in dynamic combat simulations. In such a static analysis, the opposing forces are defined, scores are determined for each weapon type, the aggregated index values are summed, and the analysis ends with the force ratio computation.

The resulting force comparison is a very rough estimate of combat capabilities, but it is certainly better than just counting divisions on each side. Such static evaluations are attractive because they are extremely simple to perform and to explain (if we assume that the score values are given and accepted).

4.3 – The ATLAS Ground Attrition Model

The evaluation of ground combat outcomes in the ATLAS theater level simulation uses a straight forward force ratio method. The simplicity of its structure is one of the main attractions of the ATLAS model that keep it in the inventory of actively used simulations in spite of other model limitations.

Combat outcomes are assessed once each simulated day in ATLAS. The computations are performed independently for each combat sector. The primary assessment objectives are to compute casualties to each side and movement of the FEBA within the sector. In the remainder of this section we will describe the ground combat assessment for one day's update in one combat sector (see references [4.5] and [4.6] for further details).

Initialization

At the beginning of a simulation run the firepower index for each combat unit (division) is computed. ATLAS calls this index the Index of Combat Effectiveness (ICE) and computes it in the standard why by summing the firepower score for each weapon in the unit. The full strength ICE is a constant for each unit throughout the entire battle.

Degradation of Unit Effectiveness Due to Casualties

For each combat unit in the combat sector, ATLAS maintains an attrition level measured as the percent of personnel casualties. New casualties are assessed at the end of each day's attrition computations. Replacement personnel allocated to the unit will decrease its attrition percentage.

Given the casualty percentage at the start of a day's update, ATLAS computes the percentage effectiveness due to casualties for each unit in the combat sector using a function graphed in Figure 4.3.1.

The model's behavior is very sensitive to this effectiveness curve (and to the similar curve for logistics). While the function shown in Figure 4.3.1 is plausible, it is difficult to why whether it is "correct" or whether it is appropriate to apply the some curve to all kinds of combat divisions or to different combat scenarios.

Degradation of Unit Effectiveness Due to Logistics

ATLAS also maintains the supply state of each combat unit measured as the number of days of supplies on hand. The number decreases by one each day and is increased when new supplies arrive from rear areas along the supply lines of communication.

Given the days of supply on hand, ATLAS computes the percent effectiveness due to logistics for each unit in the combat sector using the function graphed in Figure 4.3.2. This function is also plausible, but not obviously correct.

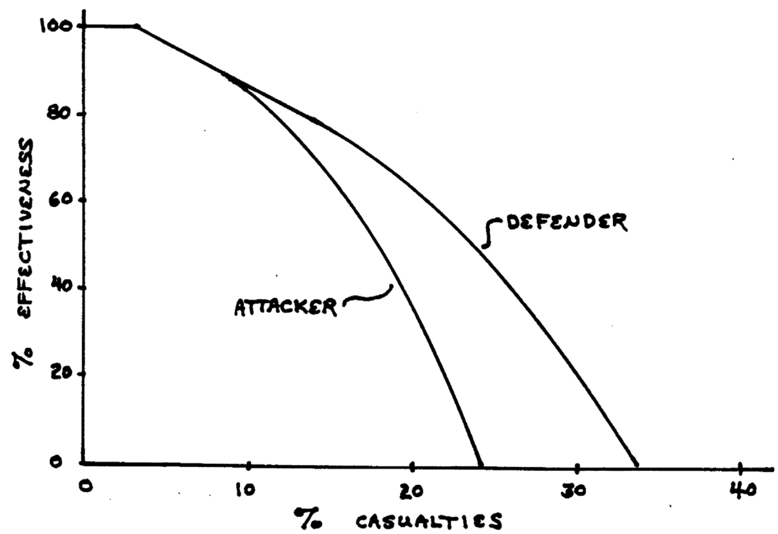


Figure 4.3.1 – ATLAS Percent Effectiveness Due to Casualties

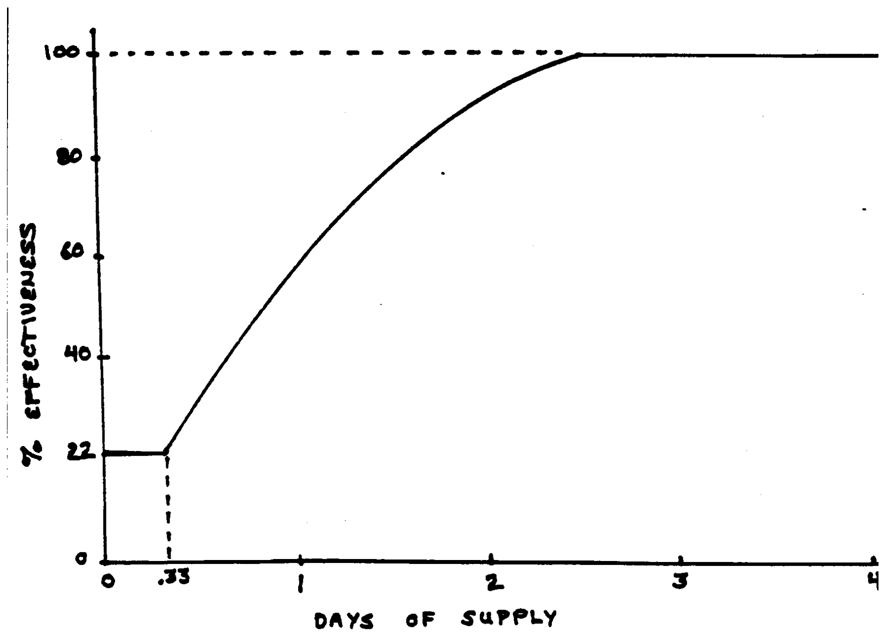


Figure 4.3.2 – ATLAS Percent Effectiveness Due to Logistics

Aggregation Over Units in the Combat Sector

For each unit in the sector, the unit percent effectiveness is computed as the minimum of the percent effectiveness due to casualties and due to logistics. Then the total combat power in the sector is computed as:

$$FPI(A) = \sum_{units} (\text{unit ICE}) * (\text{unit pct effective})$$

where the sum is over all attacker units at the front lines in the combat sector. A similar computation gives the sector total FPI(D) for the defender. The force ratio is also computed:

$$FR = FPI(A) / FPI(D). \quad (4.3.1)$$

Determination of Engagement Type

ATLAS then determines the defender combat posture and, as a result, the type of engagement to be simulated for the day's combat. The choice depends on the relative effectiveness percentages for the attacker and the defender along with defensible positions available in the current terrain segment. We will skip the details.

Movement of the FEBA

FEBA movement, if any, is always in the direction of the attack. If the combat posture is "holding" then the FEBA will not move. Otherwise, the rate of FEBA movement is a table lookup with subscripts for:

1. force ratio (10 categories),
2. terrain type including manmade barriers (6 categories),
3. engagement type (7 categories), and
4. attacker mobility (infantry or armor).

The actual rate used is a weighted average between the infantry rate and the armor rate with weights corresponding to the composition of the attacking force. A typical FEBA movement rate array is given in Figure 4.3.3.

If the computed FEBA movement distance extends past the boundary of the current terrain segment, then the FEBA movement and attrition computation will be done in several parts, one for each segment crossed during the day.

Computation of Casualty Rate

Casualty rates for the day's battle are determined as a table lookup with subscripts for:

1. force ratio (10 categories),
2. engagement type (7 categories plus "holding"), and
3. attacker or defender.

The rates are measured as percent personnel casualties per day for the division sized units.

The original casualty rate date used in ATLAS was derived from data on 37 division level engagements in WWII and Korea. The values were smoothed and enriched by procedures whose details are not known to provide the 160 values required in the data tables. The resulting casualty rate data curves are shown in Figure 4.3.4. There are severe doubts about the relevance of these values to modern combat and to various scenarios. The model user is free, of course, to modify the input values, but such changes are usually judgmental and probably cannot be validated except perhaps by reference to higher resolution division battle models.

ARMORED DIVISION MOVEMENT RATES
(Terrain Type A, No Barriers)
(In Miles per 24 Hours)

FORCE RATIO (ATK / DEF)

	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0
Fortified Zone	–	–	–	0.23	0.56	0.9	1.3	1.6	2.0	2.4
Prepared Position	–	–	–	0.86	2.0	3.1	4.0	4.7	5.35	6.0
Hasty Position	–	–	0.66	2.7	4.0	5.5	6.4	7.8	8.65	9.5
Meeting Engagement	–	0.0	2.9	5.6	7.3	8.9	10.3	11.6	12.4	13.6
Delaying Action	–	5.2	8.3	11.0	13.0	14.0	15.0	16.0	16.5	17.0
Orderly Retirement	0.0	13.0	16.0	17.4	18.5	18.9	19.2	19.5	19.8	20.0
Disorganized Retreat	0.0	15.0	18.1	20.0	20.8	21.5	21.9	22.2	22.4	22.6

Figure 4.3.3 – Typical ATLAS FEBA Movement Rates

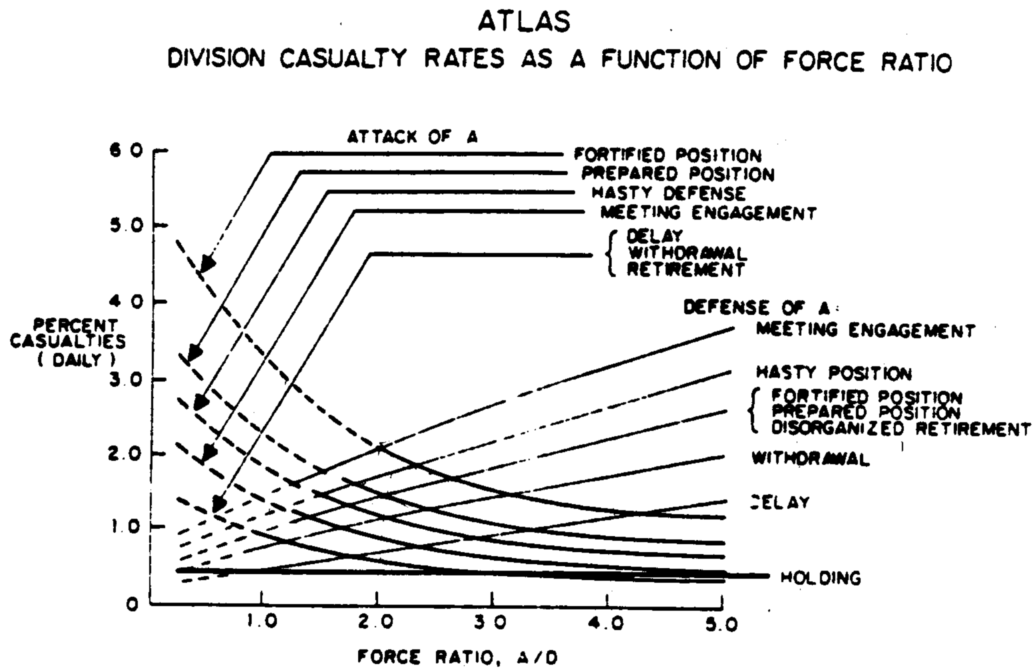


Figure 4.3.4 – Typical ATLAS FEBA Rate Curves

Casualty Assessment

The day's personnel casualty percent for the attacker (defender) is added to the casualty percent attribute for each attacker (defender) unit in the sector. One day of supplies is subtracted from each unit's supply attribute. If a unit's effectiveness falls to zero, then that unit is withdrawn from the battle until reinforcements or new supplies arrive.

This completes the description of the ATLAS daily update cycle for the ground battle. The description is slightly incomplete because it ignores the effects of close air support sorties. These are included by assigning on ICE value to CAS sorties and adding it to the sector total ground ICE.

Evaluation of the ATLAS Attrition Model

The attrition structure described above is extremely highly aggregated. Once the initial ICE is computed, the model does not know anything about the weapons that make up its combat divisions. The ICE is static throughout the simulated battle and thus the ability of ATLAS to respond to changing battle conditions is limited.

The combat sectors in ATLAS are strictly independent so only traditional FEBA battles can be represented. Coordination between sectors is not modeled, so ridiculous FEBA lines with extreme flank exposure may sometimes result.

There is very little that could be called a "theory" in the attrition computations. Rather, the model simply reads back user input data values at the appropriate times.

ATLAS was developed with an initial data base, but without an accompanying estimation or calibration procedure. Thus the derivation and justification of new aggregated process data input values is somewhat arbitrary. It is not easy to modify the data base to represent modern combat conditions.

ATLAS is a simple model, with a small input data set and fast computer execution times. It is frequently used.

4.4 – Eigenvalue Force Ratio Computations

The Potential Anti-Potential method (or eigenvalue method) for computing weapon scores is significantly different from all the earlier methods. The earlier score computation formulas all depended entirely on the characteristics of the weapon itself to yield a score value that was (hopefully) useful independent of the enemy being faced and of the particular scenario. This goal was too ambitious, and thus there was a continual effort to change and improve the score definitions.

The eigenvalue method depends on how the weapon capabilities interact with enemy vulnerabilities in a particular combat scenario. The computations include elements of the heterogeneous approach to aggregation, but eventually yield scores, indices, and force ratios for a homogeneous representation of unit combat power (references [4.4] and [4.7] discuss the eigenvalue method).

The Basic Principle

The Potential Anti-Potential method for computing weapon system scores is defined by the following basic principle:

> The value (score) of a weapon system is directly proportional to the rate at which it destroys the value of opposing enemy weapon systems.

Thus the value of a system depends on its kill rates and on the value of the enemy systems it kills. Conversely, the enemy system values depend on the values of the friendly systems that they kill. Thus the value definitions are circular.

Notation and Definitions

Consider two opposing forces (called X and Y) made up of heterogeneous weapon systems. Suppose that the X force contains m different weapon system types and that the Y force contains n types. Let

X_i = the number of weapons of type i in the X force for i = 1,2,…,m, and let
Y_j = the number of weapons of type j in the Y force for j = 1,2,…,n.

Define the weapon values (or scores) to be

SX_i = the value of one type i weapon in the X force, and
SY_j = the value of one type j weapon in the Y force.

Finally, define the kill rates:

K_{ij} = the rate at which one X_i system kills Y_j systems, and

L_{ji} = the rate at which one Y_j system kills X_i systems, for all possible combinations of the weapon indices i = 1,2,…,m and j = 1,2,…,n.

The kill rates are assumed to have known non-negative numeric values, and we will solve for the score values. Kill rate values can be obtained from the killer-victim scoreboard output from a high-resolution simulation model. Note that they will implicitly depend on scenario details such as the composition of both forces, the force missions, target acquisition conditions, target selection rules, and the outcomes of one-on-one engagements.

In terms of these definitions, the basic valuation principle can be written as a system of equations. The value for a system of type i in the X force is given by totaling the values of every enemy system that it kills:

$$CX * SX_i = \sum_j K_{ij} * SY_j \qquad (4.4.1)$$

and similarly the value of i system of type j in the Y force is

$$CY * SY_j = \sum_i L_{ji} * SX_i \qquad (4.4.2)$$

where CX and CY are the proportionality constants for the two forces.

Combining the equations gives a system of m + n linear equations in the m + n unknowns SX_i and SY_j for any given values of the proportionality constants. However, we will allow the mathematics to determine the values of CX and CY also because then we can guarantee a solution in which all the scores are non-negative.

The value equations can be expressed more compactly in matrix notation. Define

$\underline{X}$ = $(X_1, X_2,\ldots,X_m)$, an m-vector,
$\underline{Y}$ = $(Y_1, Y_2,\ldots,Y_n)$, an n-vector,
$\underline{SX}$ = $(SX_1, SX_2,\ldots,SX_m)$, an m-vector,
$\underline{SY}$ = $(SY_1, SY_2,\ldots,SY_n)$, an n-vector,
$\underline{K}$ = the m by n matrix whose elements are the K_{ij}, and
$\underline{L}$ = the n by m matrix whose elements are the L_{ji}.

Then the value equations can be expressed as

$$CX * \underline{SX} = K * \underline{SY}, \text{ and} \qquad (4.4.3)$$

$$CY * \underline{SY} = L * \underline{SX}. \qquad (4.4.4)$$

The Eigenvalue Solution

To solve the value equations, substitute the expression for $\underline{SY}$ from equation (4.4.4) into equation (4.4.3) yielding

$$CX * CY * \underline{SX} = \underline{K} * \underline{L} * \underline{SX}. \qquad (4.4.5)$$

Similarly substituting for $\underline{SX}$ in equation (4.4.3) yields

$$CX * CY * \underline{SY} = \underline{L} * \underline{K} * \underline{SY}. \qquad (4.4.6)$$

If we define E = CX * CY, then the above become

$$E * \underline{SX} =(\underline{K}*\underline{L}) * \underline{SX}, \text{ and} \qquad (4.4.7)$$

$$E * \underline{SY} =(L*K) * \underline{SY} \qquad (4.4.8)$$

which can be recognized as a pair of eigenvalue problems for the non-negative matrices $\underline{K}$ * $\underline{L}$ (m by m) and L * K (n by n). The eigenvalue is E and the eigenvectors are $\underline{SX}$ and $\underline{SY}$.

Although we cannot present the details here, the Frobenius Theorem guarantees that

1. there exists a real, non-negative, largest eigenvalue E (the same for both equation systems), and
2. there exist non-negative eigenvectors $\underline{SX}$ and $\underline{SY}$ (unique up to a scale factor) that satisfy the equations of the eigenvalue problem.

The resulting solutions are consistent with the original basic principle for valuing weapon systems, so they can be used for score values. Using the scores SX_i and SY_j we can compute aggregated unit index values and force ratios in the ordinary fashion.

Scaling the Score Values

Solutions $\underline{SX}$ and $\underline{SY}$ can easily be computed by standard eigenvalue programs. Unfortunately, the solutions are not unique. By examining the original equations

$$CX * \underline{SX} = K * \underline{SY}. \text{ and} \qquad (4.4.9)$$

$$CY * \underline{SY} = L * \underline{SX}, \qquad (4.4.10)$$

we can see that if $\underline{SX}$ and $\underline{SY}$ solve the equations then so will the scalar multiples MX * $\underline{SX}$ and MY * $\underline{SY}$ for any arbitrary scale factor multipliers MX and MY. In the new solution, the values of the proportionality constants CX and CY will adjust to absorb the scale changes.

If a solution vector $\underline{SX}$ is multiplied by the scalar MX, then the relative value between two different X force weapons will remain unchanged:

$$SX_1 / SX_2 = (MX * SX_1) / (MX * SX_2). \qquad (4.4.11)$$

Thus the scale factors do not affect weapon comparisons within the same force.

However, weapon comparisons between the X force and the Y force are clearly changed when the scores are scaled:

$$SX_1 / SY_2 \neq (MX * SX_1) / (MY * SY_2) \qquad (4.4.12)$$

if MX ≠ MY. The same result is true for the aggregate force ratio; scaling the scores changes the force ratio value by a factor of MX / MY. Thus the method chosen to scale the score vectors is extremely important, Several scaling methods have been proposed, but we will concentrate on the scaling method used in the IDAGAM attrition structure.

Suppose SX_i and SY_j are the scores that come out of the eigenvalue method before scaling and that we want to scale them to obtain new scores

$$NSX_i = MX * SX_i, \text{ and} \tag{4.4.13}$$

$$NSY_j = MY * SY_j. \tag{4.4.14}$$

Select some major weapon system from the X force, one that will engage numerous Y force system types (say we choose X system 1). Scale the X force score vector so that the new $NSX_1 = 1.0$ thus determining new values for all the other NSX_i. This can be accomplished by setting the scale factor to

$$MX = 1.0 / SX_1. \tag{4.4.15}$$

Consider the value equation that gives the value of X_1 in term of the systems attacked by the chosen weapon X_1,

$$CX * NSX_1 = \Sigma_j K_{1j} * NSY_j. \tag{4.4.16}$$

Scale SY so that the right hand side of this equation gives

$$\Sigma_j K_{1j} * NSY_j = \sqrt{E}, \tag{4.4.17}$$

Thus fixing all of the NSY_j values. This can be accomplished by choosing the scale factor

$$MY = \frac{\sqrt{E}}{\sum_j K_{ij} SY_j} \tag{4.4.18}$$

As a result of this scaling, $CX * NSX_1 = CX * 1.0 = \sqrt{E}$. Also, recall that $E = CX * CY$, so we have

$$CX = CY = \sqrt{E} \tag{4.4.19}$$

for this scaling method. There is some intuitive appeal to having the proportionality constants the same for both sides. In the discussions that follow we will drop the prefix "N" on the new scaled score vectors and just assume that the scaling has already been done as a part of the eigenvalue computations that yield SX_i and SY_j.

Evaluation of the Eigenvalue Method

The score values that result from the eigenvalue solutions are scenario dependent because they depend on the kill rates A_{ji} and B_{ij}. The circular nature of the value equations makes the relationships among the scores complex. The kill rates also depend implicitly on the number of weapon systems in the forces since that influences target engagement opportunities.

Typically, any change in any of the kill rates will cause all of the score values for both sides to change in ways that are hard to predict.

Thus we should not consider the eigenvalue scores to be a measure of long term inherent value of a weapon system, but rather only of transient value in a specific situation. Indeed, the IDAGAM theater combat simulation, which was the first simulation to use eigenvalue scores, reevaluates the score values for each day of combat.

The eigenvector score values sometimes change in ways that are hard to explain and that have been called paradoxical by some. For example, a shift in fire distribution to increase the kill rates of a higher value enemy target can sometimes reduce the total value index of the firing force. Whether this is, in fact, paradoxical depends on how deeply the relationships are followed. If there are few of the high value targets, then shifting fire away from a lower value but more numerous enemy system might very well result in a lower total value being killed by the firing force.

Other anomalies, however, are harder to explain. The numeric values of the scores are sometimes oversensitive to small changes in the input kill rate matrices. Zero score values sometimes occur for major weapons. Also, the method sometimes splits an engagement into two disconnected separate engagements (reference [4.8]). The cause of these problems seems to be that the eigenvalue equations are linear while the combat process that they are attempting to model is nonlinear. In chapter 6 we will examine a similar method that uses nonlinear value equations.

Force ratios computed from the eigenvalue scores are used for casualty assessment and FEBA movement in models like IDAGAM. Although the details of the algorithms have changed, essentially the same historical data base is used as in the ATLAS model. Since this data base was developed with the "firepower score" force ratios in mind, it is not clear that it applies equally to the force ratios that result from the eigenvalue computations.

Finally, the eigenvalue method has been criticized because it removes the element of military judgement from the scoring process. Judgement is needed for the kill rate evaluations, but the remaining computations are strictly mathematical. It is interesting that the earlier firepower scores were often criticized because their evaluation required too much military judgement.

4.5 – The IDAGAM Ground Attrition Model

The attrition structure in the theater level IDAGAM model was built with the goal of improving on the firepower score, force ratio method of the earlier ATLAS model. The model is complex, including many aspects of attrition that were not modeled in ATLAS. IDAGAM's attrition process model can be viewed as combining features of the homogeneous force ratio approach with the heterogeneous Lanchester equation approach. We classify it with the force ratio models because the force ratio is used to determine the casualty levels in the simulation.

It is important, however, not to consider IDAGAM to be just a more complicated version of ATLAS. The force ratios in IDAGAM do not use simple "firepower scores". Instead, the Potential Anti-Potential score computation method is applied and the scores are recomputed on each day of the war to account for changing conditions. Thus IDAGAM overcomes many of the criticisms that are made of the static homogeneous firepower scores used in ATLAS.

For clarity in the presentation we have chosen to include only some of the detailed aspects of the IDAGAM attrition process model. The omitted details do not change the general overall attrition structure. We have also adjusted the notation to be consistent with the rest of the book. For complete details, see the IDAGAM model documentation in reference [4.7].

Setting for the Attrition Calculations

IDAGAM computes attrition for the divisions in a combat sector with a daily update cycle. Before the attrition computation begins, the model has moved any forces that are entering or leaving the combat sector, allocated reinforcements, determined who is the attacker in the sector, determined the defender posture for the day, and computed the results of the day's air battle. As a part of the air battle evaluation, the model computes the number of close air sorties in the combat sector for both the attacker and the defender. The close air support sorties will be used during the ground combat update.

Throughout this section we will display the formulas for half of the attrition computations, for the X force firing at the Y force. Computations for Y firing at X are exactly equivalent. For concreteness, assume that X is the attacker in the sector and that Y's defensive posture is determined. Many of the data items in the calculations are input separately for attack versus defense and for different postures.

Computation of Kill Rate Matrices

Define the following variables:

$\hat{F}_{ij}$ = fire allocation against standard force = the fraction of X_i's firing that is directed against enemy targets of type Y_j when X engages a "standard enemy force" (user input for attack and defense),

P_{ij} = the potential number of Y_j weapons killed each day by one type X_i weapon if all of its firing is directed against type j targets (user input for attack vs. defense, and for each posture),

$\hat{Y}_j$ = the number of type j weapons in a standard enemy force (user input), and

Y_j = the actual number of type j weapons in today's sector battle (the result of yesterday's combat, reinforcements, and withdrawals).

Then we can calculate the actual fire allocation to be used by X against the enemy in today's sector battle as:

$$F_{ij} = \frac{\hat{F}_{ij}\dfrac{Y_j}{\hat{Y}_j}}{\sum_k \left(F_{ik}\dfrac{Y_k}{\hat{Y}_k} \right)} \tag{4.5.1}$$

This formula changes the fire allocations proportionally to account for enemy target types that are more or less numerous than in the "standard enemy force".

The "standard enemy force" gives a baseline for which we can describe how the X force would fight. As the combat scenario progresses, however, the composition of the actual enemy Y force will change. As a result, the X force may become more effective or less effective, depending on the match between X force capabilities and Y force vulnerabilities. Simple firepower score methods cannot capture this heterogeneous X_i vs. Y_j pairing, and thus cannot react to changing force structure (only to changing total firepower).

Finally define the potential kill rates against today's actual enemy force. For the X force, K_{ij} is the potential number of Y_j targets killed in today's battle by one X_i firer who allocates his fires according to F_{ij}.

$$K_{ij} = F_{ij} * P_{ij}. \tag{4.5.2}$$

Similarly we can compute a fire allocation for the Y force and then the potential kill rates for Y firing at X, which we call

L_{ji} = potential number of X_i targets killed in today's battle by one Y_j firer who allocates his fires.

The K_{ij} and L_{ji} are called "potential" kills because they will only be used to gauge relative casualty levels and not used directly for casualty computation. Data sources available at the time of IDAGAM's development were not felt to be adequate to support absolute casualty rates as a function of firer and target weapon system types. As we will see in the next chapter, several other large-scale models use rates like the K_{ij} to compute the casualties directly and provide estimation or calibration procedures for determining their numeric values.

Eigenvalue Score Computation

Let $\underline{K}$ and $\underline{L}$ be the matrices formed from the K_{ij} and L_{ji} respectively, and apply the eigenvalue score computation method to determine scores (IDAGAM calls them "values") for each weapon on each side in today's battle,

S_{X_i} = value for type i weapon in the X force, and
S_{Y_j} = value for type j weapon in the Y force.

Then compute the unit combat value index for each division in the sector using its current number of weapons X_i or Y_j,

$$TX = \sum_i S_{X_i} X_i \text{ and} \tag{4.5.3}$$

$$TY = \sum_j S_{Y_j} Y_j \tag{4.5.4}$$

Note that the score values, and thus the division totals are dependent on the force structure for the day through the scores as well as the numbers of surviving weapons since the scores are derived from the K_{ij} and L_{ji} which are recomputed for each day.

Degrade for Logistics and Personnel Shortages

The division indices TX and TY are degraded for personnel shortages for each division, degraded for logistics shortages, and summed over all divisions in the sector for each side, yielding

VGX = total effective X ground combat value in sector, and
VGY = total effective Y ground combat value in sector.

Because of the interpretation that X value equals Y value killed, these can also be interpreted as the total potential casualties that a force can inflict on the enemy considering the actual weapons, actual personnel, and actual logistics state of the units in the sector.

The computations are more intricate than in ATLAS, and we suppress the details. Essentially they are driven by user input effectiveness degradation curves for personnel and logistics shortages. The curves can be different for different division types. The computations also consider the time required for a division to reorganize and to incorporate replacements.

Compute the Air Combat Value in the Sector

The total close air support values in the sector for each force are computed from the numbers of sorties generated in the air combat model,

VAX = total effective X air combat value in sector, and
VAY = total effective Y air combat value in sector.

We will not develop the details.

Force Ratios and Personnel Casualties for the Day

IDAGAM computes two force ratios, one for the attacker and one for the defender. The force ratio for computing percent casualties to the defender (assumed to be the Y force) is

$$FRd = (VGX + VAX) / VGY, \tag{4.5.5}$$

and the force ratio for the attacker's casualty percent computation (assuming X attacks) is:

$$FRa = VGX / (VGY + VAY). \tag{4.5.6}$$

These force ratios are then used, along with the combat posture, to compute the percent personnel casualties for each side using the same historically derived casualty rate curves as in the ATLAS model. It is at this point in the model that the absolute amount of attrition is determined for each division in the combat sector:

$$XCAS = (X\ pers\ in\ division) * (X\ pct\ cas),\ and \quad (4.5.7)$$

$$YCAS = (Y\ pers\ in\ division) * (Y\ pct\ cas). \quad (4.5.8)$$

The personnel in each division are adjusted to account for supporting personnel and for the supply status before the percentages are applied (details skipped). The same percent is applied to each division in the combat sector.

Note that in each of the force ratios, the air combat value is only added for the firing side, and not for the targets. This is because it is felt that the addition of close air support will increase overall casualty levels, even if both sides employ CAS. If the air value was added in both the numerator and the denominator, then there would be a tendency for them to cancel each other out resulting in no net increase of casualty levels. Since the casualty effects are only being computed for ground targets in this part of the model, it is appropriate to use total firers (ground + air) versus ground targets in the force ratios.

If we were using the force ratio for a static overall force comparison, then the ordinary force ratio

$$FR = (VGX + VAX) / (VGY + VAY) \quad (4.5.9)$$

would be the appropriate ratio.

Several other ways of combining ground and air power for the numerator and denominator of a force ratio are used when IDAGAM does its FEBA movement computations for the day. Since we have touched on the FEBA movement process in Chapter 3, it will not be discussed further here.

Disaggregate Personnel Casualties to Weapon Casualties

Since IDAGAM keeps track of a heterogeneous aggregation of each division, it needs to know how many weapons of each type were destroyed when the above computed personnel casualties were incurred. The procedure used to disaggregate personnel casualties into weapon system casualties is a proportional disaggregation.

Define the following variables:

TYCAS = total personnel casualties for all Y force divisions in the sector today (totaled from the division personnel casualties above),

DY_j = number of weapons of type Y_j destroyed in the sector today (to be computed),

C_j = number of people killed for each Y_j weapon destroyed (computed as a weighted average of several inputs – details suppressed), and

PDY_j = potential number of weapons of type Y_j destroyed in the sector today.

The potential weapons destroyed for type Y_j is computed as

$$PDY_j = \Sigma_i (X_i * K_{ij}) + (\text{term for AIR}), \quad (4.5.10)$$

where the summation is over all the X force weapons that might shoot at Y_j weapons and where X_i is the number of firers of type i in the entire combat sector today. The close air support term is similar, but we suppress the details to avoid having to define any more notation.

Assume that actual weapon system casualties are proportional to potential casualties. This assumption defines the proportional disaggregation method to be used. In mathematical notation, assume that:

$$DY_j = Q * PDY_j \tag{4.5.11}$$

where the same proportionality constant, Q, is used for all weapon types, j. For consistency of the definition of C_j we must have

$$TYCAS = \sum_j C_j \; DY_j \tag{4.5.12}$$

Substituting for DYj from equation (4.5.11) yields

$$TYCAS = \sum_j C_j \; Q \; PDY_j \tag{4.5.13}$$

so we can solve for Q as follows,

$$Q = \frac{TYCAS}{\sum_j C_j \; PDY_j} \tag{4.5.14}$$

and thus compute $DY_j = Q * PDY_j$ for each weapon type j in the Y force. These weapon casualties for the sector are then divided among the divisions in the sector.

The equation for the potential casualties, PDY_j, is essentially a Lanchester Square Law equation. In this model it is being used only to compute the relative number of casualties and not the absolute numbers.

This completes our description of the IDAGAM ground attrition computation for one day in one combat sector. In addition to the attrition, IDAGAM computes daily FEBA movement in each sector and adjusts the FEBA to account for front-to-flank exposures.

The significant features of the IDAGAM attrition structure are:

1. the force ratio computation is based on battle conditions in the sector for the day through the eigenvalue method, and not on static firepower scores, and
2. the model maintains a heterogeneous aggregation and thus can assess X_i versus Y_j weapon type interactions, even though the absolute casualty level is computed from a homogeneous force ratio using the ATLAS casualty rate data.

Problems for Chapter 4

(to be determined later)

References for Chapter 4

[4.1] Taylor, J.G., *Lanchester Type Models of Warfare*, Operations Research Society of America, Vols 1 and 2, 1983.

[4.2] Stockfisch, J.A., "Models, Data, and War: A Critique of the Study of Conventional Forces," Report R-1526-PR, RAND Corporation, March 1975.

[4.4] Howes, D.R., and R.M. Thrall, "A Theory of Ideal Linear Weights for Heterogeneous Combat forces," Naval Research Logistics Quarterly, Vol. 28, pp. 645-659, 1973.

[4.5] Kerlin, E.P., and R.H. Cole, "ATLAS: A Tactical, Logistical, and Air Simulation," Technical Paper RAC-TP-338, Research Analysis Corporation, April 1969.

[4.6] Man Kwon Nam, "An Examination of the Conceptual Basis of ATLAS," MSOR Thesis, Naval Postgraduate School, March 1980.

[4.7] Anderson, L.B. et al., "IDA Ground-Air Model I (IDAGAM I)," Report R-199, Institute for Defense Analysis, October 1974.

[4.8] "ATCAL: An Attrition Model Using Calibrated Parameters," Technical Paper CAA-TP-83-3, Center for Army Analysis, August 1983.

CHAPTER 5

LANCHESTER-TYPE DIFFERENTIAL ATTRITION MODELS

5.1 – Introduction

Lanchester-type attrition models refer to the set of differential equation models that describe changes, over time, in the force levels of combatants and other significant variables that describe the combat process. (Ref 1, Taylor p.28) Subsequently, Lanchester-type models express casualties/attrition in terms of force size, and other associated variables and how they change over time. They may be simple models with closed form solutions capable of being solved through simple mathematics or they may be large, highly complex models requiring a variety of analytical and simulation techniques. Such models are used to answer such basic questions as who wins the battle or more complex operational questions pertaining to force mix or tactics.

Lanchester differential equation models have gained importance through their ability to provide insight into the dynamics of combat and their applicability to almost the entire hierarchy of combat operations (e.g. battalion through theater-level). In cases where simple models are utilized, explicit analytical functional forms may be derived and answers readily provided to the client/user. Further, these differential equation models provide a basis for developing quantitative insights into combat dynamics. The simple equations form the base for model enrichment that provides the means to simulate combat and address more critical operational problems.

While there exists a wide variety of Lanchester-type differential models based on size and complexity, there are several underlying factors that appear common to the model development process. These concepts are:

- attrition to a force is a function of force size and other associated parameters (i.e. casualty rate = f(force size; other possible parameters))
- force size is a function of time, and the continuous real time variables x(t), y(t) and t are approximations to the discrete combat units in a real force.
- if we consider two opposing forces X, Y and let
  x(t) = size of the X force as a function of time
  y(t) = size of the Y force as a function of time
  the casualty rates can be written as a simple pair of differential equations

$$\frac{dx}{dy} = f(x, y) \qquad \frac{dy}{dt} = f(x, y)$$

- The solution to any such system of differential equations is a pair of functions giving x(t) and y(t) as a function of time.

While the use and study of Lanchester-type differential attrition models spans over 70 years since the initial equations formulated by F.W. Lanchester, the remainder of this chapter will examine Lanchester's original models, the mathematics of the Lanchester Square and Linear Laws, other functional forms, enrichment to Lanchester-type operational models, attrition rate coefficients, and Lanchester-type models currently in use. In no way is the treatment of these topics meant to be exhaustive but rather to provide the reader with a basis to understand Lanchester-type differential models. For a detailed study of Lanchester-type differential attrition models, we recommend the reader consult the various works of James G. Taylor as a source for a more comprehensive and in depth analysis.

5.2 – Lanchester's Original Models

Origins of the Lanchester Models

In 1914 F.W. Lanchester, a British engineer and inventor, formulated two differential models for attrition under specific conditions of war. His purpose was to quantitatively justifying the principle of concentration of forces under the then conditions of modem warfare. Lanchester hypothesized that in "ancient warfare", a battle was simply a collection of one-on-one duels, with the casualty rate being independent of the number of units on the opposing side. Under "modern" conditions, he contended that the firepower/lethality of weapons widely dispersed across the battlefield can be concentrated on surviving targets and a many-against-one situation could exist. Therefore, the casualty rates should be proportional to the size of the opposing force. Lanchester formulated some models based on ordinary differential equations to translate these hypotheses into mathematical terms.

Conditions of Ancient Warfare

Based on the hypothesized model for one-on-one duels, Lanchester argued that two forces of equal strength and fighting ability should intuitively be expected to lose about the same number of men. Further, under this one-on-one condition, any forces not engaged with an opponent must wait until an enemy soldier became available before joining combat. This implies that regardless of how large the X force is, it cannot engage the opposing Y force with more men than Y puts forth on the battlefield. Therefore under the condition of "ancient warfare" there should be no advantage in concentrating forces.

While never explicitly formulated, Lanchester's ancient warfare equations reflect a combat attrition process where attrition rates are independent of force size; that is

$$\frac{dx}{dt} = -a \qquad \text{and} \qquad \frac{dy}{dt} = -b \tag{5.2.1}$$

The individual X unit is superior to the individual Y unit if and only if b > a. Both sides decrease gradually in any case until one or the other becomes 0, at which point battle stops and 5.2.1 no longer holds. The relationship between x and y can be found from

$$\frac{dy}{dx} = \frac{\frac{dy}{dt}}{\frac{dx}{dt}} = \frac{-b}{-a} = \frac{b}{a} \qquad (5.2.2)$$

Since the slope of y with respect to x is constant, x and y must at all times be related by

$$b[x_0 - x] = a[y_0 - y], \qquad (5.2.3)$$

where x_0 and y_0 are the initial values.

Examination of this simple model provides information on the dynamics of combat under Lanchester's hypothesized conditions of ancient warfare. Specifically, there is no advantage from concentrating forces when such conditions exist.

Example 5.1. Given two forces X and Y with initial strengths of $x_o = 100$, $y_o > 100$ and a casualty exchange ratio of one (i.e. a = b), consider a fight to the finish where the X force is totally destroyed ($x_f = 0$). From 5.2.3, the surviving number of Y units must be $y_f = y_0 - 100$. Regardless of the initial Y force size (provided that $y_o > 100$, the necessary condition for a Y victory) there will always be 100 Y force casualties when engaging an X force of 100. See Table 5.1, which numerically illustrates this relationship.

Table 5.1 – Force Sizes Under Conditions of Ancient Warfare

y_o	y_f	Y Casualties
100	0	100
200	100	100
300	200	100
500	400	100

Conditions of Modern Warfare

As previously noted, Lanchester defined the principal condition for modern warfare as the ability of many firers to engage a single target. He based this condition on the advent of modern weapons that allowed multiple engagement possibilities and concentration of fires from weapons widely dispersed on the battlefield.

Considering the nature of modern weapons and how the concentration of fires could be achieved, Lanchester examined two general cases of combat, aimed fire and area fire. The first, aimed fire, assumes that individual targets are identified and attacked by any number of opposing systems/firers. The second case, area fire, considers the situation where a force concentrates its fires over a general area occupied by the enemy and not at any particular enemy target.

Under aimed fire conditions, Lanchester stated that the attrition rate of x depends on how many y's are shooting at him, and likewise for y. In mathematical terms

$$\frac{dx}{dt} = -ay \qquad \frac{dy}{dt} = -bx \tag{5.2.4}$$

Here a is an attrition rate coefficients expressed in terms of (X casualties)/(Y firer) × (time), and similarly for b.

As will be shown below, it follows from 5.2.4 that x and y are related by

$$b\left[x_o^2 - x^2\right] = a\left[y_o^2 - y^2\right]. \tag{5.2.5}$$

Using the information from Example 5.1, it is possible to illustrate the significant difference that occurs under the conditions of modem warfare and the advantages of concentrating forces.

Example 5.2. When a = b, according to 5.2.5, the number of Y survivors when X has been annihilated is $y_f = \sqrt{y_0^2 - x_0^2}$. Enumeration for several values of y_0 indicates that there is a marked advantage in concentrating forces. Numerical results are shown in Table 5.2.

Table 5.2 – Force Sizes Under Conditions of Modern Warfare

y_o	y_f	Y Casualties
100	0	100
200	173	27
300	283	17
500	490	10

Lanchester's original purpose was to develop a mathematical argument to support the general tactical principle of concentration. The values of the attrition coefficients (a and b) did not really matter in his qualitative arguments. However today, in addition to a qualitative use, we demand that combat models give reasonable quantitative results. Thus the numerical coefficients do matter, and the specific nature of combat is important – not just its overall nature.

Through the use of Lanchester-type combat models, it is possible to answer a variety of questions about combat between two forces. Taylor (Ref 1, pg. 65-66) lists seven general questions for which the answers can be extracted from Lanchester models. These questions are:

1. Who will win the battle; or which force will be annihilated?
2. What force ratio is required to guarantee victory?
3. How many survivors will the winner have?
4. How long will the battle last?
5. How do the force levels change over time?
6. How do changes in the parameters {e.g. initial force levels (x_o and y_o) or attrition coefficients (a and b)} affect the outcome of the battle?
7. Is concentration of forces a good tactic?

While the terms found in these questions are subject to various interpretations, more specific questions can be answered based on the complexity of the model and the number of parameters incorporated. As additional parameters are added to a model, more questions may be posed. However, for our purpose, discussion will be limited to how Lanchester-type models are developed and how they yield answers to the seven basic questions listed above.

5.3 – Mathematics of the Lanchester Square Law (aimed fire)

Lanchester originally hypothesized that combat between two homogeneous forces under the conditions of modem warfare could be modeled as:

$$\frac{dx}{dy} = -ay \qquad \text{where } x(0) = x_o$$

$$\frac{dy}{dt} = -bx \qquad \text{where } x(0) = y_o$$

The equations hold only as long as both x(t) and y(t) are positive. Battle stops when either number becomes zero, if not before. Based on the hypothesized differential equations, it is possible to derive the equations that will allow us to determine who wins the battle, force size and time to battle termination.

Derivation of the State Equation

Using the equations for force casualty rates, it is possible to derive an expression for the instantaneous casualty-exchange ratio as follows:

$$\frac{\frac{dx}{dt}}{\frac{dy}{dt}} = \frac{ay}{bx} = \frac{dx}{dy}.$$

Separating the variables,

$$bx\,dx = ay\,dy$$

and integrating both sides, we discover that $bx^2 + c_1 = ay^2 + c_2$. Given the initial conditions, the constants must be such that at all times

$$b\left(x_o^2 - x^2\right) = a\left(y_o^2 - y^2\right). \qquad (5.3.1)$$

Therefore given a value for either x or y, it is possible to solve for the other. However, it is important to note that we do not get any information about *when* any particular force level is achieved.

Force Levels as a Function of Time

The pair of ordinary differential equations that determines x(t) and y(t) can be solved using standard methods. The solution is

$$x(t) = \frac{1}{2}\left(\left(x_o - \sqrt{\frac{a}{b}}\,y_o\right)e^{\sqrt{ab}t} + \left(x_o + \sqrt{\frac{a}{b}}\,y_o\right)e^{-\sqrt{ab}t}\right) \qquad (5.3.2)$$

$$y(t) = \frac{1}{2}\left(\left(y_o - \sqrt{\frac{b}{a}}\,x_o\right)e^{\sqrt{ab}\,t} + \left(y_o + \sqrt{\frac{b}{a}}\,x_o\right)e^{-\sqrt{ab}\,t}\right). \qquad (5.3.3)$$

The verity of these equations can be established by observing that x(0) and y(0) have the required values, and that the two differential equations are satisfied. Of course it should be understood that 5.3.2 and 5.3.3 hold only as long as both x(t) and y(t) are nonnegative. Figure 5.3.1 shows x(t) and y(t) for the case a = .01, b = .02, x_0 = 20, and y_0 = 40. Note that the outnumbered X loses even though his firepower rate b is twice that of his opponent. Figure 5.3.2 is the same battle except that b is increased from .02 to .05.

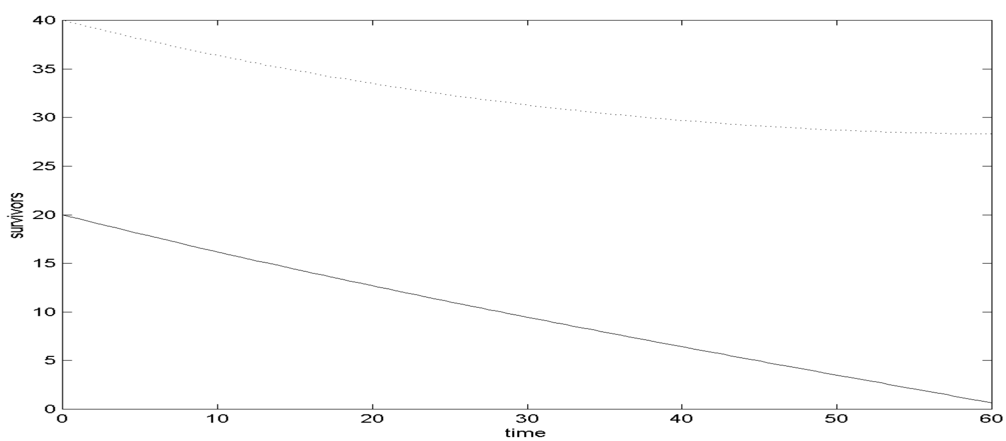


Figure 5.3.1: A Lanchester Square Law battle where x (solid line) loses to y (dotted line).

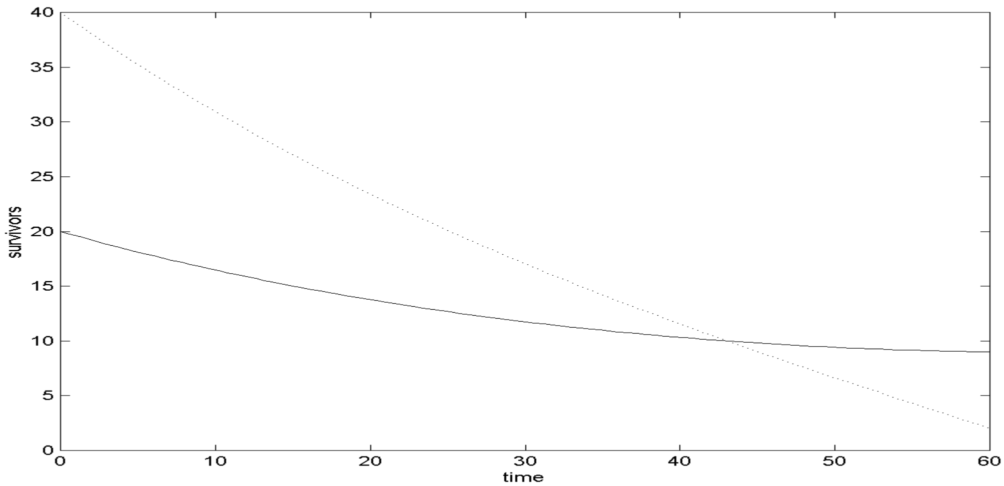


Figure 5.3.2: A Lanchester Square Law battle where y (dotted line) loses to x (solid line).

Battle Outcome and Duration

To determine who will win the battle it is necessary to specify some condition that will cause the battle to terminate. Assume that the x-side will surrender or break off fighting in some other way if x(t) ever shrinks to x_{BP}, where of course $x_{BP} < x_0$, and similarly for the y-side and y_{BP}. At the terminal time, either $x(t) = x_{BP}$ and $y(t) > y_{BP}$, in which case the y-side is the winner, or $y(t) = y_{BP}$, $x(t) > x_{BP}$, and the x-side is the winner.

Case Y wins:

Since the X loses, the number of y-survivors y_f can be obtained by solving 5.3.1 with $x = x_{BP}$.

$$b\left(x_o^2 - x_{BP}^2\right) = a\left(y_o^2 - y_f^2\right)$$

$$y_f = \sqrt{y_o^2 - \frac{b}{a}\left(x_o^2 - x_{BP}^2\right)}, \qquad (5.3.4)$$

assuming that $y_f > y_{BP}$. The criterion for this to be true; that is, the criterion for the y-side to win the battle, is

$$b\left(x_o^2 - x_{BP}^2\right) < a\left(y_o^2 - y_{BP}^2\right). \qquad (5.3.5)$$

The left and right-hand sides of 5.3.5 might be called the "fighting strengths" of the two sides, since the comparison determines the winner. Note that the number of participants on each side is squared, whereas the firepower rate coefficient is not, hence the term "Square Law". In a Square Law battle, it is more important to have lots of units than it is to have powerful units. Intuitively, adding one more unit to a square law battle serves two purposes: it fires at the enemy, and in addition it dilutes the enemy's fire against existing units. Increasing a firepower rate coefficient only serves the first purpose.

The length of the battle can be determined by solving 5.3.2 for t when $x(t) = x_{BP}$. Let $z = \exp\left(-t\sqrt{ab}\right)$. Since $\exp\left(t\sqrt{ab}\right) = 1/z$, 5.3.2 is a quadratic equation in z. The only solution for which $0 < z \le 1$ is

$$z = \frac{\sqrt{b\left(x_{BP}^2 - x_0^2\right) + ay_0^2} + \sqrt{b}x_{BP}}{\sqrt{b}x_0 + \sqrt{a}y_0}. \tag{5.3.6}$$

The time t at which x(t) is x_{BP} is therefore

$$t = -\ln(z)/\sqrt{ab}. \tag{5.3.7}$$

For example, suppose a =.01/day, b =.02/day, $x_0 = 20$, and $y_0 = 40$, with $x_{BP} = y_{BP} = 0$. Then z =.4142, t = 62.32 days and $y_f = 28.28$. In spite of having inferior units (a < b), Y wins with most of his forces intact.

Case X wins:

If
$$b\left(x_o^2 - x_{BP}^2\right) > a\left(y_o^2 - y_{BP}^2\right), \tag{5.3.8}$$

then y(t) will become y_{BP} before $x(t) = x_{BP}$; that is, X wins. The number of x-survivors is

$$x_f = \sqrt{x_o^2 - \frac{a}{b}\left(y_o^2 - y_{BP}^2\right)}. \tag{5.3.9}$$

Solving the quadratic equation 5.3.3 with $y(t) = y_{BP}$ for z as above, the solution is

$$z = \frac{\sqrt{a\left(y_{BP}^2 - y_0^2\right) + bx_0^2} + \sqrt{a}y_{BP}}{\sqrt{b}x_0 + \sqrt{a}y_0}, \tag{5.3.10}$$

with 5.3.7 still determining the time of battle termination.

If neither inequality 5.3.5 nor 5.3.8 holds because the two fighting strengths are equal, then the battle is a tie. Either 5.3.6 or 5.3.10 can be used to determine z, since both equations have the same solution.

5.4 – Mathematics of the Lanchester Linear Law (area fire)

The basic hypothesis of the Linear Law is

$$\frac{dx}{dt} = -axy \qquad \text{and} \qquad \frac{dy}{dt} = -bxy .$$

While a and b are still referred to as attrition coefficients, they differ from those coefficients used in the Square Law. Specifically, the attrition coefficients are measured in units of (casualties/((time) × (firers) × (targets)). Any comparison of attrition coefficients between laws is a comparison of applies with oranges, since the units are different. While the Linear Law is usually assumed to apply to area fire weapons such as artillery, any other assumptions that lead to the conclusion that attrition rates should be proportional to the number of targets, as well as the number of firers, would do as well.

Derivation of the State Equation

The instantaneous casualty exchange ratio for the Linear Law can be expressed as:

$$\frac{dy}{dx} = \frac{\frac{dy}{dt}}{\frac{dx}{dt}} = \frac{-bxy}{-axy} = \frac{a}{b} . \tag{5.4.1}$$

In other words, the rate of change of y with respect to x is a constant, as in ancient warfare. Therefore x and y must always be related by

$$b\left[x_o - x\right] = a\left[y_o - y\right], \tag{5.4.2}$$

where x_0 and y_0 are the initial values of x and y. If the battle breakpoints are x_{BP} and y_{BP}, the fighting strengths of the two sides are now $b(x_0 - x_{BP})$ and $a(y_0 - y_{BP})$, respectively. The Linear Law derives its name from these formulas, since fighting strength is linear in the number of combatants. The winner is still the side with the larger fighting strength. Note that 5.4.2 is the same state equation that holds in ancient warfare; the dynamics change drastically under the Linear Law, but not the final outcome.

The analytic solution of x(t) and y(t) as functions of time is more complex than for the Square Law, so we omit the formulas (see Taylor). However, one feature of the solution is worth noting: neither side is ever annihilated even in a fight-to-the-finish. This should make intuitive sense, since the effectiveness of area fire diminishes as the density of targets becomes small.

5.5 – Other Functional Forms of Lanchester's Original Models

We have shown how Lanchester's differential equations can be used to answer some basic questions on combat under the conditions of aimed and area fire homogeneous combat.

However, as previously noted, combat is rarely homogeneous and as such the original Lanchester models while useful have many shortcomings. Some of these shortcomings are:

- considers only constant attrition rate coefficients
- no force movement during battle
- battle termination is not modeled
- tactical decision processes are not considered
- C^3 is not considered
- no logistical aspects are portrayed
- suppressive effects of weapons are not considered
- target prioritization/fire allocation not explicitly considered
- noncombat losses are not considered.

Obtaining a full list of shortcomings will probably never be possible as new doctrine, weapons and system interaction are modeled, and new operational questions arise. This does not say that the original equations/models are inadequate but rather they require refinement to account for the various aspects of concern to the decision maker. To this end, we will examine several refinements or extensions that have been developed.

The first operational refinement of the Lanchester equations that we will consider is to the basic function form. However, before we examine these modifications, it is convenient to introduce a shorthand method, developed by Taylor, for classifying models in accordance with these functional forms. The methodology incorporates a two part X|Y descriptor, which defines the attrition rate for the X and Y forces, respectively. Because X and Y acquire their value based on the terms in their respective attrition rate, the classification scheme uses these terms as a descriptor. We have seen that attrition may be dependent on the number of firers (denoted **F**) or the number of targets (denoted as **T**). If the attrition rate is constant than it is independent of either F or T, and is designated with the letter **C**. Using these three characters it is possible to classify the functional form of the combat processes within a given model. For example, we would designate Lanchester's Linear Law, where firers and targets are key inputs to the attrition process, as **FT|FT**. Multiple attrition processes are expressed using a plus (+) to separate the various component terms. The various functional forms, their associated equations and shorthand designators are provided in the explanations below.

Mixed Combat

Up to this point we have developed the necessary functional forms for the aimed fire (or Square Law) and the area fire/Linear Law equations for homogeneous force combat. From here it is possible to describe various forms of combat that are combinations of these two forms. The most obvious form is the one for **mixed** combat. Specifically where one force uses aimed fires and the opposing force uses area fires, denoted F|FT.

This situation is analogous to the X force attacking the Y force in a prepared defensive position. While both sides use aimed fires, it is important to remember that the time to acquire a target for an X firer dominates the attacking force actions and therefore the Linear Law applies. The same situation was shown to apply for insurgency operation models where one force ambushes another force. If X is in the open and Y ambushes X, then the Y firers use aimed fires but the X force firers must use area fire since they do not know the exact positions of their attackers. In these cases, the state equation can be developed exactly as in the linear and Square Law to yield:

$$\frac{b}{2}\left(x_o^2 - x^2\right) = a\left(y_o - y\right). \tag{5.5.1}$$

Logarithmic Law (noncombat losses)

A second extension of the Lanchester models hypothesizes that the initial states of a small unit engagement can be models as a T|T attrition process or

$$\frac{dx}{dt} = -ax \qquad \text{and} \qquad \frac{dy}{dt} = -by. \tag{5.5.2}$$

This process is referred to as the logarithmic law from its state equation

$$b \ln\frac{x_o}{x} = a \ln\frac{y_o}{y}. \tag{5.5.3}$$

The logarithmic law is almost silly as a "combat" model because each side decreases asymptotically to zero independent of the number of combatants on the other side. "We have met the enemy, and he is us!" But the logarithmic law makes more sense than might appear at first sight, since there are many sources of attrition other than hostile fire that must be accounted for (disease, desertion,…). The logarithmic law is not the whole story, but including terms such as –ax in the expression for dx/dt can still be used to model such phenomena in a larger situation.

Helmbold Equations

A general form for homogeneous force attrition rate that yields the square, linear, and logarithmic laws as special cases was postulated by R. Helmbold in 1965. He stated that *the relative fire effectiveness is influenced by the force ratio in the sense that if x/y is extremely large, then X cannot effectively bring all his weapons to bear on the Y force.* His reasoning was based on the perception that limitations of space, terrain masking, and the target engagement opportunities would prevent a large force from using its full firepower.

In conjunction with this hypothesis, Helmbold suggested that the following Lanchester-type differential equations would be more appropriate

$$\frac{dx}{dt} = -a\left(\frac{x}{y}\right)^{1-\omega} y \quad \text{and} \quad \frac{dy}{dt} = -b\left(\frac{y}{x}\right)^{1-\omega} x \tag{5.5.4}$$

where ω is a measure of efficiency with which the large force can be brought to bear on the small force. The alert reader will immediately see that these equations are the aimed fire equations with a force ratio modifier added in.

In order to illustrate the range of situations that the Helmbold equations cover, we need only to assign ω the values of 0, 1, and ½ and evaluate the resulting differential equations. When

- $\omega = 0$, the Logarithmic Law.
- $\omega = 1$, the Square Law.
- $\omega = ½$, the Linear Law, at least for the state equation.

In the last case, Helmbold's equations share with ancient warfare the property that a force can be annihilated in finite time. This is not true for the Linear Law, but nonetheless the same state equation holds.

5.6 – Enrichment to Lanchester-type Models

The preceding section dealt with modeling the attrition processes for various combat situations in terms of force characteristics for homogeneous force combat. While these equations can cover a broad range of combat scenarios, they do not account for many factors that can affect the outcome of a battle. Since we recognize that the dynamics of battle entails a myriad of factors, we must introduce them into our earlier equations if we wish to accurately portray combat. In this section we will consider the following enrichments of the Lanchester-type models:

1. replacements, reinforcements, and/or withdrawal,
2. range dependency,
3. heterogeneous forces, and
4. stochastic Lanchester models.

Replacements, Reinforcements, and Withdrawals

Each of these three options is a reality on the battlefield and in practice, a critical decision problem faced by a commander. While each alternative may occur under various conditions and in different form, we will only consider the simple and direct changes. For our purpose, we define two models:

- continuous replacement/withdrawal, and
- unit reinforcement.

The *continuous model* simply adds a constant to each equation to represent replacement or withdrawal at a specified rate. Morse and Kimball (1950) define P and Q to be the reinforcement rates for the two sides and consider a model with both Square Law and Logarithmic Law attrition:

$$dx/dt = P – ay – \beta x \text{ and } dy/dt = Q – bx – \alpha y. \quad (5.6.1)$$

They are able to obtain a complicated analytic solution. We will not discuss it further, except to note that x(t) or y(t) can now be an increasing function of time on account of the reinforcements. Indeed, the equations were motivated by prior work in biological systems where increases are natural.

In the case of *unit reinforcement*, instantaneous changes in x(t) or y(t) occur at a particular time (t_r) which is the reinforcement/withdrawal time. Unlike the continuous model, this process occurs outside the basic attrition equation. This essentially requires us to stop the equation at t_r and then resume the battle with a different force structure. Figure 5.6.1 illustrates the unit replacement process over time.

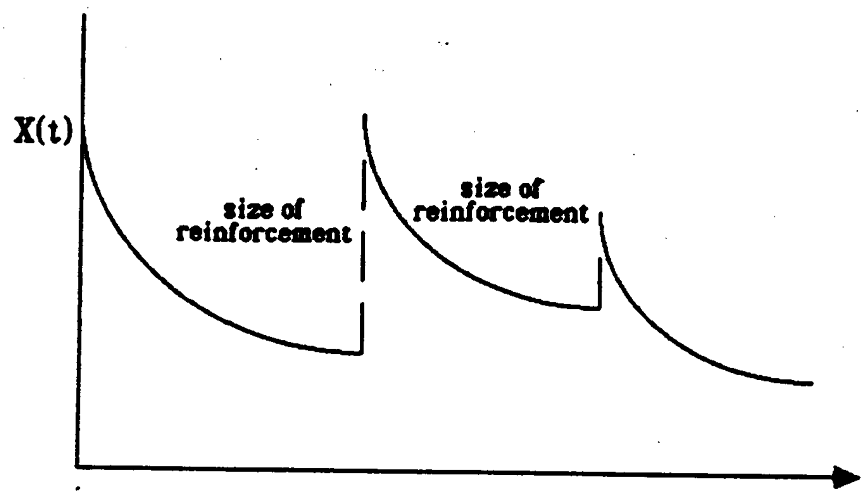


Figure 5.6.1 – Unit Reinforcement in Lanchester Models

Engel (1954) is an application of this idea to the WWII battle for Iwo Jima. Engel estimated that the individual Japanese soldier fighting from prepared positions was about five times as effective (a/b = 5.1) as a US soldier. In spite of this all 20,000 Japanese troops initially on the island were killed. Marines landed over a period of several days, numbering about 75,000 at the peak. The number of US casualties was also about 20,000, but 15,000 of those survived.

Range Dependency

Early models failed to consider in detail the effect of range on the attrition process. Practical experience indicated that attrition is affected by range and should be considered depending on the resolution level of the model. Under range dependency, the attrition coefficients are functions of range and the differential equations are of the form

$$\frac{dx}{dt} = -a(r)y \quad \text{and} \quad \frac{dy}{dt} = -b(r)\,x\,.$$

The dependency of the attrition coefficient on range was first studied by Bonder for a constant speed attack and various forms for a(r) and b(r). Based on his studies, Bonder suggested that constant attrition coefficients could be replaced by

$$a(r) = a_o\left(1 - \frac{r}{r_{max}}\right)^{\mu} \qquad \text{for } 0 \le r \le r_{max}$$

$$a(r) = 0 \qquad \text{for } r \ge r_{max} \qquad (5.6.2)$$

where r_{max} = maximum range of the weapon system, and a_o = maximum attrition rate.

Plotting the attrition coefficient as a function of range (Figure 5.6.2) we see how different values for μ can affect the outcomes.

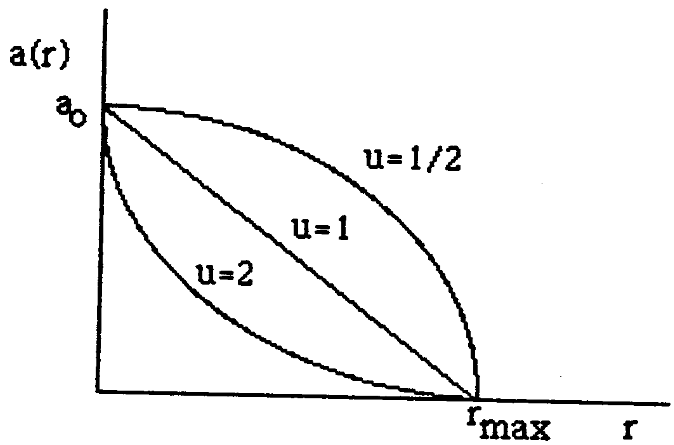


Figure 5.6.2 – Bonder Range Dependent Attrition Coefficient Plots

When we consider the constant speed model, we can express the range as a function of time.

$$r(t) = r_o - \nu t$$

where r_o = initial range and ν = the closing velocity.

The differential equations then become functions of t only

$$dx/dt = -a[r(t)]y$$

and can be analyzed.

The point here is that *as soon as we consider real situations the constant coefficient models are no longer valid*. Some analytical work on differential equations with nonconstant coefficients has been done (Bonder), but the majority of work has been done numerically because of the sensitivity of such models to the form of range dependency used.

Heterogeneous Forces

Up to now we have assumed that individual elements of the X force have identical characteristics. Thus only the total number of combatants X(t) is the driving factor for attrition assessment. A schematic of homogeneous combat is shown in Figure 5.6.3.

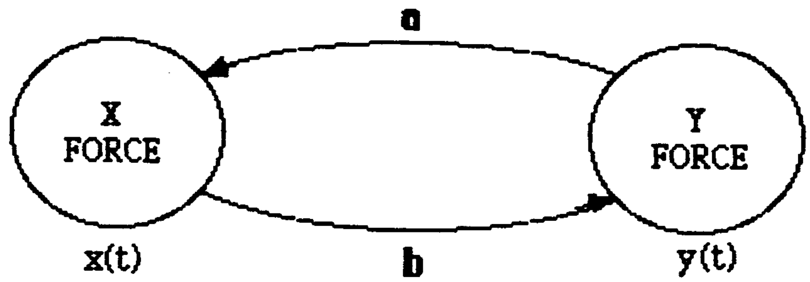


Figure 5.6.3 – Homogeneous Combat Model

Now let us consider a combined arms force:

$X = [x_1(t), x_2(t), \ldots, x_m(t)]$

$Y = [y_1(t), y_2(t), \ldots, y_n(t)]$

where $x_i(t)$ = number of X survivors of weapon system i at time t.

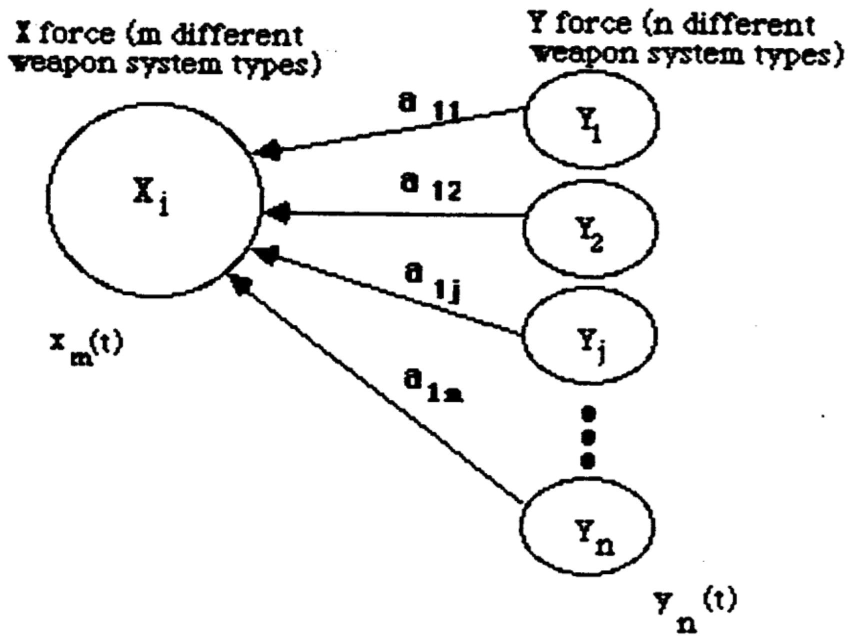


Figure 5.6.4 – Heterogeneous Combat x_i System vs. Y Force

Lanchester-type model for attrition assessment will involve m + n *differential equations* – assessing the casualty rate for each x_i, y_i separately. If we select a single x_i system and consider attrition to that system it would resemble Figure 5.6.4.

Therefore, we can assess the attrition to a single system as:

$$\frac{dx_i}{dt} = \sum_{j=1}^{n}\left(\text{attrition of } x_i \text{ systems caused by } y_j \text{ systems}\right).$$

(**Note:** the attrition may be 0 for some j if y_j does not kill x_i.)

For the heterogeneous model to function we have to make two assumptions about additivity and proportionality. The first assumption, *additivity* says that there is no direct synergism. Simply stated the only way any antitank systems can contribute to the effectiveness of tank systems is by killing enemy tank systems. Consequently, their presence or absence in a force does not enhance the killing potential of a tank system. Hence synergism does not exist if attrition depends only on y_j. If attrition depends on y_j and y_k, for $k \neq j$ then synergism exists. To model synergistic effects is a complex task however it is not a problem here as the additivity assumption has eliminated the possibility of such effects.

The second assumption, *proportionality* says that *the loss rate* of x_i *caused by* y_j is *proportional to the number* of y_j that engage x_i. To better understand this assumption let us define ψ_{ij} as the fraction of y_i fires allocated to targets of type x_i, where $\left(\sum_i \psi_{ij} = 1\right)$. Then on the average we can say that:

$$y_{ij} = \psi_{ij} y_j$$

is the number of y_j's that engage x_i. For example, if $y_j = 100$ and $\psi_{ij} = 0.25$ then:

$$y_j = 25.$$

This does not say that only 25 Y firers shoot at x_i, but rather averaged over the y_j force, ¼ of the time is spent engaging x_i targets.

Now if we let a_{ij} represent the attrition rate of one y_j system shooting at x_i, then if all the y_j firers are allocated against x_i systems

$$a_{ij}\psi_{ij}y_j = a_{ij}y_{ij}.$$

Defining the combination of the attrition term (a_{ij}) and the allocation term (ψ_{ij}), we get:

$$A_{ij}y_j = a_{ij}y_{ij}.$$

Since we now can represent one system within the force, it is a simple step to model the complete system. Hence the complete heterogeneous system is

$$\frac{dx_i}{dt} = -\sum_{j=1}^{n} A_{ij}y_j \quad i = 1, \ldots, m$$

$$\frac{dy_j}{dt} = -\sum_{i=1}^{m} B_{ji}x_i \quad j = 1, \ldots, n$$

with initial conditions $x_i(0) = x_i^o$; $y_j(0) = y_j^0$ and with the understanding that A_{ij} becomes zero if either $x_i(t) = 0$ or $y_j(t) = 0$.

Once we have written this system of equations, we can go no further analytically. We are at a point very similar to the original Square Law solution but the answer gives little insight into the combat dynamics. In short, the equations are too complex and there are too many coefficients. This leaves us with the problem of application to real world models.

If we hold A_{ij} and B_{ji} constant, the equations are essentially the Lanchester Square Law equations. However, we are not bound by any one particular law when we model heterogeneous force combat. In most operational combat models using Lanchester-type attrition processes, the heterogeneous equations shown above are either explicitly or implicitly changed to correspond to the nature of particular system interactions. Thus in a series of ij engagements there may be any permutation of the homogeneous laws previously discussed. The implicit option occurs when the form of the basic equation is changed by letting the coefficients be variable and letting A_{ij}/B_{ji} be functions of the number of x_i's and y_j's. In either case, we are forced to use numerical solutions or some method for coefficient estimation.

While heterogeneous force combat appears to be a nearly impossible task to model, it is quickly placed in perspective if one remembers that the same techniques used to model homogeneous combat can be used to model the subcomponents of heterogeneous force combat. Thus we may state, simply, that heterogeneous force combat is just the summation of a series of homogeneous force battles.

Numerical Solutions to Differential Equations of Combat Attrition

As we have seen, many differential equations are intractable to solve analytically particularly in the cases of the more enriched Lanchester formulations. Accordingly, applied mathematicians have developed some very sophisticated procedures for numerically solving these systems. One reason why numerical integration is a viable approach lies in the fact that the Lanchester-type equations are well behaved when compared to many other equations used in science and engineering. Specifically, the Lanchester-like equations are:

- monotone
- have no singularities
- have stable solutions

Thus the simplest of the numerical methods (Euler-Cauchy) normally works quite well. This is fortunate because the method corresponds to a simple time step simulation that interfaces easily with combat model technology.

Consider the combat model

$$\frac{dx}{dt} = -A(x, y, t, \text{ other factors})$$

$$\frac{dy}{dt} = -B(x, y, t, \text{ other factors})$$

where *other factors* may include such variables as range, weather, unit posture, etc. and $x(0) = x_0$, $y(0) = y_0$, with battle termination conditions set at x_{BP}, y_{BP}. The *Euler-Cauchy* method is based on the simple finite difference approximation to dx/dt. Then:

$$\frac{dx}{dt} \cong \frac{x(t + \Delta t) - x(t)}{\Delta t} = -A(x, y, \ldots)$$

$$\frac{dy}{dt} \cong \frac{y(t + \Delta t) - y(t)}{\Delta t} = -B(x, y, \ldots)$$

or

$$x(t + \Delta t) - x(t) = -A(x, y, \ldots)\Delta t$$

$$y(t + \Delta t) - y(t) = -B(x, y, \ldots)\Delta t$$

resulting in a simple pair of difference equations. From here it is a straightforward procedure of

1. initialize all values
2. select a Δt based on a well defined characteristic such as firing cycle times
3. Compute x, y, and t. In calculating A, B we must use the previous values x(t), y(t) and t as the starting point of the interval $(t + \Delta t)$. Since Δt is assumed to be small enough that no major inaccuracy occurs (i.e. combat conditions or force size do not change radically within the time interval) this should not be a problem.
4. Finally, test for break point conditions. If $x \le x_{BP}$ or $y \le y_{BP}$, then the battle stops. Otherwise we repeat the previous step until termination conditions exist.

Stochastic Lanchester Models

Everything we have done so far with Lanchester equation models has been deterministic and continuous, whereas actual battles are fought by discrete units very much subject to luck. Intuitively, the deterministic models should be at their best when the numbers of units involved are large, but what does "large" mean, and exactly what do we mean by statements like "the Blue side has 36.78 units left"? Does the statement mean that the average number of units left is 36.78? To some extent, those questions can be answered by simply reinterpreting the Lanchester right-hand-side as the transition rates of a continuous time Markov chain whose state (m,n) is the number of survivors on the two sides. Instead of saying that the Blue side will lose $\Delta A(x,y,t)$ units out of x in the next Δ time units, we say that the probability of losing one unit out of m is $\Delta A(m,n,t)$ (we will replace (x,y) by (m,n) in this section to emphasize that the state is composed of integers). The average number of units lost in time Δ is the same with either interpretation, but the Markov interpretation has the desired properties of having both sides composed of integral quantities and subject to luck. The deterministic model can be thought of as being derived form the Markov model by Expected Value Analysis (EVA), which is just a name for the practice of replacing all random variables by their expected values. EVA generally results in simpler models, but with a sacrifice in accuracy.

There is a computational price to pay for employing the stochastic model. In general, one must generate the time history of (m,n) in a time-step Monte Carlo simulation that increments time until some termination condition is reached. Lanchester's differential equations only need to be solved once, but the stochastic simulation must be run many times to get an idea of the variability of casualties and other combat results. Unless some kind of structure is imposed on A() and B(), the process of studying the impact of a change in initial numbers or coefficients could be time consuming.

One useful kind of structure is the homogeneous case where A(m,n) and B(m,n) do not depend on time. In that case, the transition rate out of state (m,n) is A(m,n) + B(m,n) and the time spent in state (m,n) is an exponential random variable whose mean is the reciprocal of that rate (Ross,1997). The probability that the next transition is to state (m,n–1) is the ratio b(m,n) ≡ B(m,n)/(A(m,n) + B(m,n)), and, since the only other possibility is that the m-side loses a unit, the probability that the next state is (m–1,n) is a(m,n) ≡ A(m,n)/(A(m,n) + B(m,n)) = 1 – b(m,n). A simple event-step Monte Carlo simulation is thus possible: remain in state(m,n) for an exponentially distributed time, choose one of the two possible succeeding states at random according to known probabilities, then choose another exponential random variable, etc. This will be more efficient than the time-step method.

There is one particular kind of computation that does not require simulation in the homogeneous case. Let (M,N) be the state at battle termination, and suppose that MOE(M,N) is some measure of effectiveness that depends only on the terminal state. The expected value e(m,n)≡E(MOE(M,N)) can be computed iteratively by employing the conditional expectation theorem. The intention here is to compute the expected value under the condition that the state starts in (m,n) (or passes through (m,n)—it makes no difference in a Markov model). For example, suppose that the battle proceeds until one side or the other has 0 units left, and let MOE(m,0) = 1 for m > 0, or MOE(0,n) = 0 for n > 0 (state (0,0) is impossible in such a battle). In other words, MOE(M,N) simply indicates whether the m-side wins the battle. Since the timing of transitions is unimportant, only the dimensionless probabilities a(m,n) and b(m,n) are relevant, and e(m,n) must satisfy the equations

$$e(m,n) = \begin{cases} 0 \text{ if } m = 0 \\ 1 \text{ if } n = 0 \\ a(m,n)e(m-1,n) + b(m,n)e(m,n-1) \text{ if } m > 0, n > 0 \end{cases}$$

Note that the first two equations define e(m,n) = MOE(m,n) on the terminal boundary, and the last is the conditional expectation theorem. The equations are used iteratively by beginning on the boundary and working into the interior in such a manner that the right-hand-side is always known whenever the left-hand-side needs to be computed. Eventually one will arrive at the starting state (m_0,n_0). For example, e(1,1) = b(1,1), and once e(1,1) is known, e(1,2) and e(2,1) can be computed, etc. A single pass is all that is required, even though the unknown function e() is involved on both sides of the conditional expectation.

It may help to imagine an (m,n) grid as in Figure 5.6.5, which shows the transition probabilities for Lanchester's Linear Law where A(m,n) = Gmn and B(m,n) = Hmn, where G + H = 1. In that case a(m,n) = G and b(m,n) = H for all (m,n), so each leftward transition has probability G and each downward transition has probability H. The arrows in that diagram can be interpreted in two ways. In terms of a simulation, they show the possible transitions out of a state. In terms of computations, the function e(m,n) can be calculated at any state if it is already known at both of the possible succeeding states.

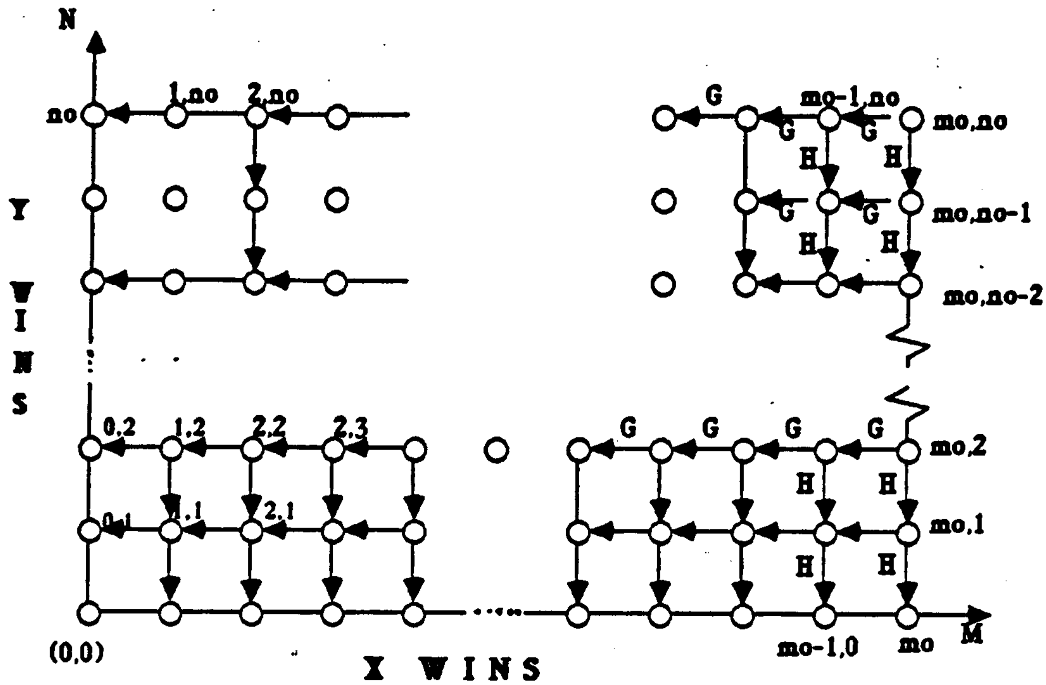


Figure 5.6.5 – P(win) Computation Grid

The same method can be used for other measures of interest by simply changing the values on the 0-boundaries to be MOE(0,n) or MOE(m,0), as appropriate. To make e(m,n) represent the average number of m-type survivors, for example, let e(m,0) = m and e(0,n) = 0. To make e(m,n) represent the probability that the terminal state is (3,0), let e(3,0) = 1 and make all other values 0. The same conditional expectation formula applies regardless of the MOE() function. The process is easily automated in a spreadsheet. The necessity of computing e(m,n) a total of m_0n_0 times when the goal is merely to compute e(m_0,n_0) might even be thought a virtue, since it provides an automatic sensitivity study as far as initial numbers are concerned.

If G = H =.5 in Figure 5.6.5, the calculation of e(3,2) might proceed as e(1,1) = 1, e(2,1) = 3/4, e(1,2) = 1/4, e(2,2) = 1/2, e(3,1) = 7/8, and finally e(3,2) = 11/16. The deterministic equivalent would have the side with the larger initial numbers always winning, so the change from deterministic to stochastic has essentially changed the (3,2) win probability from 1 to 11/16. The deterministic and stochastic models are not equivalent here, nor would they be if the MOE were the number of m-units surviving or practically any other quantity. The custom of interpreting the deterministic numbers as the expected values of the stochastic numbers is therefore questionable, although there are some limiting results showing the interpretation is satisfactory in lopsided battles where m and n are large (see Taylor, 1985). In practice "large" is often taken to mean that the initial numbers for both sides exceed 20. This roughly distinguishes land battles from naval battles, with the deterministic version being used for land battles and the stochastic version for naval battles. However, modern computers are easily able to deal with Markov battles involving hundreds or even thousands of units on each side. Although the error involved in using the deterministic approximation in such battles may be small, the additional cost in making the stochastic computations is likewise small.

Lanchester analyses sometimes employ more than two entities. The same Markov interpretation can be made regardless of the number of entities, with the homogeneous case still being advantageous in permitting the use of an event-step simulation. However, although the conditional expectation theorem still applies, computations involving MOE(m,n,...) quickly become cumbersome as the number of arguments of MOE( ).

Probability Maps

Let P(m,n,t) be the probability that the state is (m,n) at time t. A probability map simply shows the probability for all states at some specified time. Probability maps can be constructed by taking advantage of the fact that P(m,n,t) must satisfy the Chapman-Kolmogorov equations (ref to Ross):

$$\frac{dP(m,n,t)}{dt} = A(m+1,n,t)P(m+1,n,t) + B(m,n+1,t)P(m,n+1,t)$$

$$-(A(m,n,t) + B(m,n,t))P(m,n,t)$$

where A() and B() are the transition rates. If the battle starts in state (m_0,n_0), then m and n can be confined to $0 \le m \le m_0$ and $0 \le n \le n_0$, with P(m,n,t) being 0 otherwise. Since the state (0,0) is also impossible, there are a total of $m_0n_0 + m_0 + n_0$ simultaneous differential equations that must be solved. Figures 5.6.6-8 show some examples. The figures apply to a Square Law battle where A(m,n,t) =.01n, B(m,n,t) =.02m, m_0 = 20 and n_0 = 40. The deterministic version would have the y-side winning at time 62.32 with 28.28 survivors. The figures show probability maps at times 20, 40, and 60 for the stochastic version. The battle may actually be over by time 40, since at that time some probability has already accumulated in the states where m = 0. By time 60, most of the probability is in those states. In a less lopsided battle, there might also be some probability in the states where n = 0, but in this case the m-side has essentially no chance of winning. Note that the deterministic version gets the winner right, and (by eyeball) the average number of survivors, but Figure 5.6.8 makes it clear that the number of n-survivors can vary quite a bit from its average.

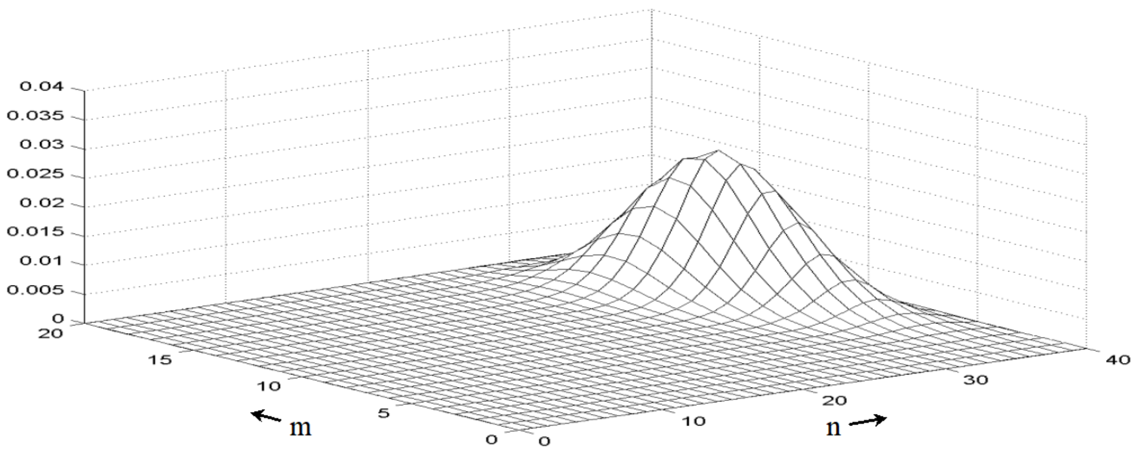


Figure 5.6.6 – Probability Map at Time 20

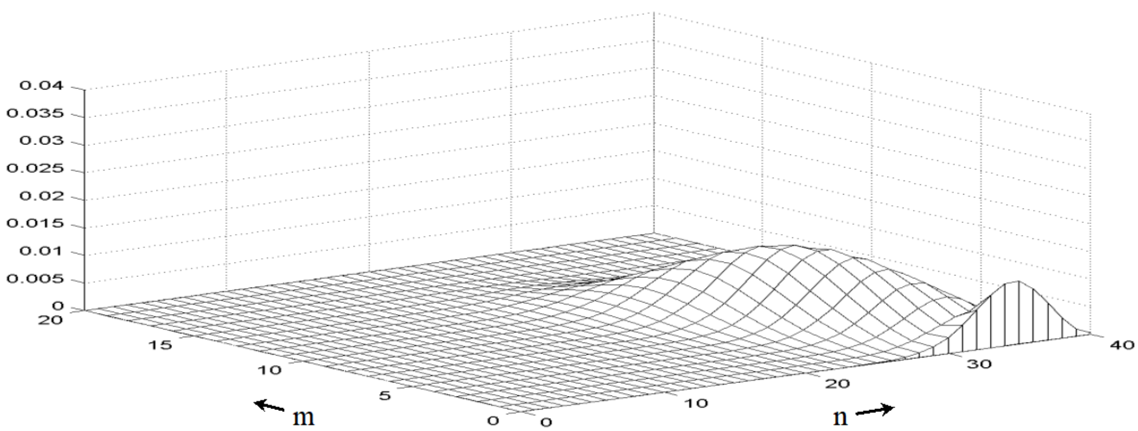


Figure 5.6.7 – Probability Map at Time 40

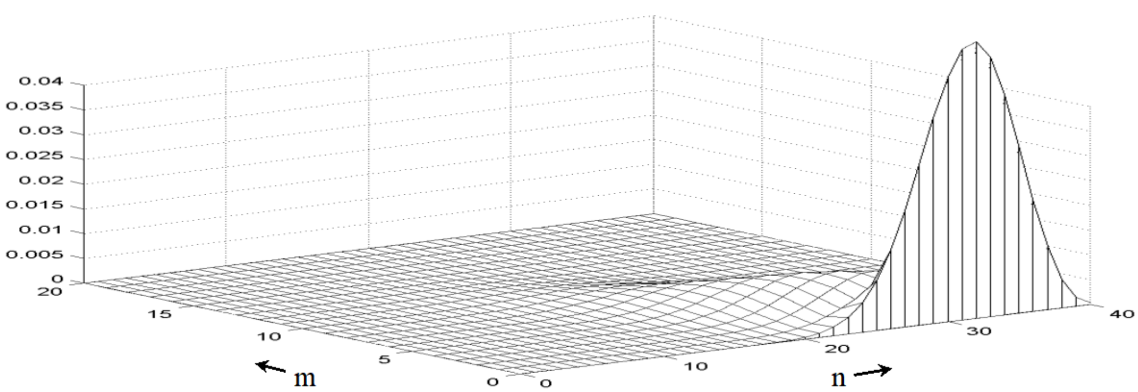


Figure 5.6.8 – Probability Map at Time 60

Exercises for Section 5.6

1. Consider a Square Law battle where A(m,n) = an and B(m,n) = bm, where a =.01/hr and b =.02/hr., with the usual termination condition. Starting from state (2,4), show that
   1. The length of time until the state changes is 33.33 hours, on the average.
   2. The probability that the m-side wins is .26.
   3. The expected number of m-survivors is .41
   4. The expected number of m-survivors, given that m wins, is 1.58

2. Using a spreadsheet, alter problem 1 so that the initial state is (20,40), and show that the answers to the four parts are 3.33 hours, .013, .092, and 6.96, respectively. The "scaled up" battle is more definite in that the m-side hardly ever wins. When he does win, however, he averages almost 7 survivors. The spreadsheet also makes it easy to study the impact of changing the rates a and b.

3. Change problem 1 so that the last unit on the n-side simply surrenders once he realizes that he is alone, thus terminating the battle, while the m-side behaves as before. Show that the probability that the m-side wins increases to .31.

5.7 – Attrition Coefficient Estimation for Lanchester Models

Throughout the chapter we have referred to casualty rates or attrition coefficients while providing only simple dimensional definitions. Additionally we have assigned values to the coefficients for purposes of examining trends and combat processes but never have we shown how we got the values for a and b. For illustrative purposes let us consider a simple F|F deterministic combat process. By definition we know that:

$$\frac{dx}{dt} = -ay \quad \text{and} \quad \frac{dy}{dt} = -bx$$

where a = X casualties/ unit time/ Y firer and b = Y casualties/ unit time/ X firer. And we can say that the attrition coefficients a and b are a function of some unspecified attrition factors (a = f(attrition factors)).

Inherent in these hypotheses is the total X casualties per unit of time is proportional to the number of Y firers. Intuitively we know that many other factors influence attrition. This raises the question of how to capture these other factors into the attrition rate coefficients, a and b. If we are modeling a battle in which any of these factors change with t (e.g. range) then we must let a = a(t), a nonconstant. while our first reaction to this is to say that this will lead to increasingly complex and intractable equations, recall that by using numerical solution techniques our task will not become any harder since Δt should be sufficiently small for a(t) to be considered constant within the interval. Therefore incorporating time dependent factors into attrition coefficients need not be avoided for fear of complexity.

As indicated above, the prime consideration for the modeler is defining the time unit to be used. For example, if we let one time step equal one day (as is the usual practice in highly aggregated firepower score models) then a is measured in casualties/day/enemy. But combat is not a uniform process over an entire day. Thus we somehow have to average attrition over various battle phases including parts of the day when non-direct combat engagements are occurring. On the other hand if we let the time step equal one minute then we need 24x60 or 1440 time steps to make-up a single day. In each time step we can compute a, b to reflect the *essentially instantaneous combat conditions.* Concurrently, the model simulation then sets up the conditions or situation between opposing forces and from the situation we can compute new values for a and b for use in the next time step. Therefore we can relate a, b directly to weapon systems parameters such as P_k, firing rate, basic loads, etc. through a series of look-up tables.

Operational models currently in use tend toward the second case, using time steps for ground casualty assessment in the range of 0.1 to 15 minutes. For our purposes, coefficient estimates will concentrate on small Δts so we can assume

1. a,b are essentially constant over the interval $(t + \Delta t)$.
2. Δt is small enough that the battlefield conditions (including force size) at the beginning of the interval are representative of the entire interval.

As with any attempt to model real world phenomenon it is logical to start with a simple representation and then enrich or embellish as necessary. Using this approach will allow us to build the necessary foundation for more sophisticated techniques without losing sight of our purpose of how to estimate attrition coefficients. With the direction for the examination of coefficient estimation set, let us first look at the basic technique using a deterministic model.

Naive Estimate

In the naive estimate we consider the casualty rate, a, for point fire to be:

$$a = (\text{firing rate}) \times (\text{prob. of a casualty per shot})$$

or

$$a = \nu_f \times P_{ssk}$$

where the maximum value for ν_f is based on engineering parameters while the average ν_f is almost always less due to battlefield conditions developed essentially from behavioral data. The P_{ssk} is a single shot kill probability based primarily on engineering data and dependent upon factors such as range, target type, and firer posture. For aimed fire we consider P_{ssk} to be constant and it firing dominates the target acquisition process then ν_f is also constant. Thus in the aimed fire case, we get

$$\frac{dx}{dt} = -ay \qquad \text{where } a = \nu_f \; P_{ssk} \text{ is a constant}.$$

In the case of area fire, ν_f being constant is a reasonable assumption. The probability of a single shot kill is usually determined by comparing the lethal area of a round to that of the target area. Then

$$P_{ssk} = \text{expected number of targets killed by one round.}$$

Subsequently the probability of a kill for a single shot can be expressed as:

P_{ssk} = lethal area of one round times the target density

$$p_{ssk} = a_l \frac{x}{A_{tgt}}$$

$$= \frac{a_l}{A_{tgt}} x$$

where a_l = lethal area of one round
A_{tgt} = total target area
and x = number of targets.

Thus

$$\frac{dx}{dt} = \nu_f \frac{a_l}{A_{tgt}} xy$$

yielding the expression for the Linear Law since a depends upon x.

<u>Poisson (Markov) Assumption</u>

Recalling from the earlier discussion of stochastic attrition models that we were able to determine the outcome of battle based on the time between casualties and several other factors. During this investigation we noted that the casualty rate could be expressed as the reciprocal of the expected time between casualties at any time during the battle. Therefore since we can express the attrition coefficient as

$$a = \frac{1}{E[T_{XY}]}$$

where the denominator is the expected time for one Y firer to kill one X target, we can estimate the attrition rate coefficients throughout a battle if we can develop a model for the expected time to kill a target.

Such an approach is preferred because we can easily incorporate the various factors that are relevant to the weapon firing cycle that were merely averaged together in the naive estimate. Analogously, for heterogeneous Lanchester models we can compute attrition rates as:

$$a_{ij} = \frac{1}{E[T_{ij}]}$$

where T_{ij} is the time (a random variable) for one Y_j firer to kill one passive X_i target in an engagement where Y_j concentrates on X_i. The level of concentration can then be modified by a fire allocation factor (ψ_{ij}) based on acquisition priorities. This specific process will be examined in greater detail when we discuss the Bonder methodology for attrition coefficient generation.

Current Estimation Methodologies

There are two methodologies currently in use for estimating attrition coefficients, COMAN and Bonder-Farrell. The COMAN approach, developed by G. Clark, is a fitted parameter model that takes a time series of casualty times and computes the maximum likelihood estimates of the mean time between casualties. The Bonder-Farrell technique is used in independent analytical models (they do not depend on outside models for input). In this methodology, a stochastic process model of a single Y_j firing at type x_i targets is built and then $E[t_{ij}]$ values are determined.

Before discussing the details of these methodologies, it is to our best interests to briefly discuss the distinctions between the two approaches. The COMAN model assumes:

1. that a Lanchester process is occurring, and
2. whatever assumptions are implicit in the data source models (which are generally high-resolution small unit combat models).

The details of the assumptions being made are not apparent in the COMAN model output, since and the COMAN approach tends to hide them. The Bonder approach assumes:

1. a Lanchester process is occurring, and
2. whatever explicit assumptions get made in the i–j independent engagement model.

When comparing the two, we see that the assumptions required for the Bonder technique are generally more restrictive since the in-depth engagement model is analytic and in turn suppresses detail. The assumptions are explicit and up-front, which makes them easier to criticize. Finally, there typically is no possibility for synergistic effects to occur in the Bonder approach.

In regards to model data sources both COMAN and Bonder are data intensive but in different ways. In a COMAN model high-resolution simulations are run to develop the necessary data sets for various combat situations and scenarios. Ultimately the coefficients are based on a multitude of situational P_k type data. On the other hand, the Bonder approach uses engineering data directly in its calculations, which requires extensive data on each weapon system. Comparatively, the hardest thing in the COMAN methodology is to be sure that the high-resolution situation is consistent with the original scenario. Since the time and cost of running high-resolution simulations at each time step is prohibitive, COMAN relies on large libraries of a_{ij}'s and selects the particular value that corresponds most closely to the current situation. The Bonder method uses closed form equations for a_{ij} as a function of assumed parameters. As such we can afford to re-evaluate the coefficients at each time step by simply recalling and recomputing the necessary input values.

Under the U.S. Army's Model Improvement Program it has been proposed to link the new generation models

- CASTFOREM (TRACWSMR)
- CORDIVEM (FT. LEAVENWORTH)

- FORCEM (CAA)

into a true model hierarchy. At this point in time, the exact nature of these models has not been determined and details of how to extend COMAN are uncertain.

The Bonder-Farrell methodology is the basis for the VECTOR, Bonder/IUA, and BLDM small unit deterministic Lanchester engagement models. The focus of these models is at battalion-level. The DIVOPS, VECTOR I, VECTOR II, and VIC models use the Bonder equations for their ground combat assessment routines. While the specific models using the technique have fallen under some criticism, the coefficient generation methodology is gaining wide acceptance.

Summarizing this brief aside, the question of coefficient generation for Lanchester attrition rates brings us right to the heart of a current controversy in Army theater-level modeling. This controversy lies in whether such models should be structured as:

- a hierarchy of separate models with external (COMAN-like) linkages, or
- a self-contained architecture.

What we have now is essentially three levels of models at three agencies with at best minimal linkages (Figure 5.7.1). There is currently no way to ensure that these models are in any way consistent with each other.

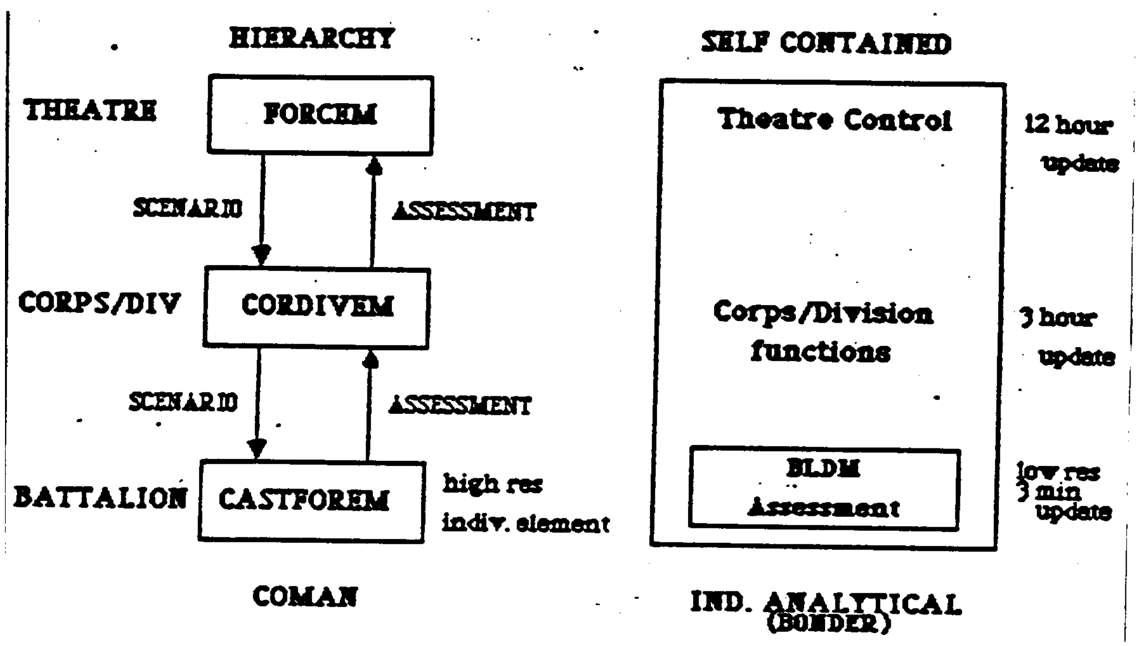


Figure 5.7.1 – Proposed Theater-level Model Structures

Derivation of the COMAN Maximum Likelihood Function

Consider a homogeneous F|F combat. It can be modeled either deterministically

$$\frac{dx}{dt} = -ay \quad \text{and} \quad \frac{dy}{dt} = -bx$$

or stochastically

$P(x \text{ casualties in } \Delta t) = an\Delta t$
$P(y \text{ casualties in } \Delta t) = bm\Delta t$

where casualties occur randomly in accordance with a memoryless Markov process. Now suppose a casualty has just occurred that placed the system into state (m,n) and let S_x, S_y be random variables for the times until the next x or y casualties to occur. Then the probability density functions can be written

$$f_{SX}(s) = an \exp[-(an + bm)s]$$

$$f_{SY}(s) = bm \exp[-(bm + an)s]$$

Given a detailed casualty history of a battle (i.e. high-resolution simulation casualty data) with K equal to the total casualties to both sides, we then define

t_k = time of occurrence of the k^{th} casualty

$$C_k^X = \begin{cases} 0 & \text{otherwise} \\ 1 & k^{th} \text{ casualty to X} \end{cases}$$

$$C_k^Y = \begin{cases} 0 & \text{otherwise} \\ 1 & k^{th} \text{ casualty to Y} \end{cases}$$

and

m_k = size of X force after the k^{th} casualty
n_k = size of Y force after the k^{th} casualty.

It follows that:

m_o = X force size after 0 casualties (starting strength)
n_o = Y force size after 0 casualties (starting strength).

From this information we want to derive $\hat{a}$, $\hat{b}$ the maximum likelihood estimators for a and b.

By the memoryless property of the Markov process, we get the likelihood function as the simple product of the likelihoods for each of the independent kill time events. More specifically, we can say that the contribution of the kth casualty likelihood function equals the probability that it used the recorded amount of time to occur. In other words, if the kth casualty to X occurs, it then contributes $f_{SX}(t_k - t_{k-1})$ to the total casualty function, or

$$f_{SX}(t_k - t_{k-1}) = an_{k-1} \; \exp[-(an_{k-1} + bm_{k-1})(t_k - t_{k-1})]$$

and the kth casualty is to Y then

$$f_{SY}(t_k - t_{k-1}) = bm_{k-1} \; \exp[-(bm_{k-1} + an_{k-1})(t_k - t_{k-1})]$$

We can express the likelihood function for the kth casualty as

$$l_k = (an_{k-1})^{C_k^X} (bm_{k-1})^{C_k^Y} \exp[(an_{k-1} + bm_{k-1})(t_k - t_{k-1})]$$

and for the whole battle as

$$L(a,b) = \prod_{k=1}^{K} l_k \, .$$

We can now find the MLE for a and b as we would any other MLE. From above

$$L(a,b) = \sum_{k=1}^{K} C_k^X \ln(an_{k-1}) + \sum_{k=1}^{K} C_k^Y \ln(bm_{k-1}) - \sum_{k=1}^{K} (an_{k-1} + bm_{k-1})(t_k - t_{k-1}) \, .$$

taking the partial derivatives with respect to a and b; setting each to zero and solving for a and b yields

$$\hat{a} = \frac{C_T^X}{\sum_{k=1}^{K} m_{k-1}(t_k - t_{k-1})}$$

and

$$\hat{b} = \frac{C_T^Y}{\sum_{k=1}^{K} m_{k-1}(t_k - t_{k-1})} \, .$$

Dimensional analysis of the estimators indicates that:

$$\hat{a} = \frac{\text{total X casualties}}{\text{total enemy firer time units against X}}$$

Comparing this to our standard definition for an F|F attrition coefficient

$$a = \frac{\text{number of X casualties}}{(\text{firer}) \text{ x } (\text{time})}$$

indicates that the MLE proves to be a true estimator for attrition coefficients and not just a surrogate value such as developed using firepower scores.

Derivation of the Bonder-Farrell Attrition Coefficients

The Bonder-Farrell methodology was developed as an independent analytical model for attrition rate estimation. The basic procedure in this approach is to develop an analytical model for the time to kill for a single weapon system engaging a passive target. This is accomplished by considering the time to accomplish the various processes that contribute to the engagement. Then the attrition rate coefficient is computed simply as:

$$\hat{a}_{ij} = \frac{1}{E[T]}.$$

The critical task is how to model an engagement of a single firer engaging a passive target. For purposes of our examination we will consider a Y firer engaging an X target and define

T_{XY} = time for Y to kill X.

In order to represent the various firer-target situations it is necessary to develop submodels that reflect the more specific nature of the engagement. A general form for the submodel is

$$T_{XY} = T_{aXY} + T^{1}_{XY}$$

where T_a represents the time to acquire the target and T^{1}_{XY} is the time to kill the target given the target has been acquired. Since we are concerned with only one Y firer and X target, we will for convenience drop the XY subscripts in the equations for the various submodels that will be discussed.

Simple Independent Repeated Shot Model

In this model the firer shoots at a fixed rate of fire until the target is killed. Each shot is considered to be totally independent. Then

t_s = time for each shot (a constant and therefore deterministic)
P_s = probability of a kill for each shot = $P_h \times P_{K|H}$

Letting

$$T = t_s \times N$$

where N equals the number of shots required to get a kill, we then see that

$$E[T] = t_s \times E[N].$$

If f(n) is the probability that a kill occurs exactly on the nth shot (i.e. geometric) then

$$f(n) = (1 - P_s)^{n-1} P_s .$$

It follows that:

$$E[N] = \sum_{n=1}^{\infty} n(1 - P_s)^{n-1} P_s = \frac{1}{P_s}$$

and

$$E[T] = \frac{t_s}{P_s}$$

which is the same as our earlier hypothesized result and concurrently defines a set of circumstances under that the model holds, that is $\hat{a} = \nu_s P_s$, if

1. target acquisition times are negligible
2. statistical independence among shots
3. uniform rate of fire
4. P_s is constant (i.e. range and other environmental changes over the period of a single engagement are negligible).

Finally, if the target acquisition time has a mean, t_a, then

$$E[T^1] = t_a + \frac{t_s}{P_s}. \quad (5.7.1)$$

Markov Dependent Fire Model

It is immediately apparent that the above model is extremely simplistic and that the assumption of total independence among shots is an easily challenged point. Therefore let us consider a second model where the shots are not independent but rather a Markovian situation exists in which the results of each shot are dependent on the previous shot. Such a situation can be described as a *shoot-look-shoot* firing sequence with sensing of a miss or a hit occurring after each round fired. Further, appropriate aim adjustment based on the previous shot results will occur during this process.

We define three states:

1. the first round of the engagement is about to be fired.
2. the previous round fired resulted in a hit.
3. the previous round fired resulted in a miss.

The state transition probabilities are determined by four probabilities associated with the firing process:

- P_1 = probability of a first round hit.
- P_2 = probability of a hit on the current shot given a hit on previous shot.

- P_3 = probability of a hit on current shot given a miss on previous shot.
- P_K = probability of a kill given a hit.

Each state is occupied for a certain amount of time before a transition to the next state occurs, and these times are crucial in determining the expected total time that it takes to kill a target. Let

- t_a = time to acquire a new target.
- t_l = time to fire the first round after target acquisition.
- t_h = time to fire a round following a hit.
- t_m = time to fire a round following a miss.
- t_f = projectile flight time to target.

Then the occupation times in the three states are

- $\tau_1 = t_a + t_l + t_f$
- $\tau_2 = t_h + t_f$
- $\tau_3 = t_m + t_f$

The state transition diagram for the Markov chain is shown in Figure 5.7.2, with the states being represented by circles and the occupation times written inside.

Let x_i be the mean time to get from state i to state 1 in this Markov chain, with x_1 being interpreted as the mean time to *return* to state 1. x_1 is also the mean time between target kills, the main object of the analysis. In each state i, the target spends an amount of time τ_i before either going to state 1 (the goal) or to one of the two transient states, which will require further transitions. Since the transition probabilities are all known, by the conditional expectation theorem,

$$\begin{aligned} x_1 &= \tau_1 + P_1(1 - P_K)\, x_2 + (1 - P_1)\, x_3 \\ x_2 &= \tau_2 + P_2(1 - P_K)\, x_2 + (1 - P_2)\, x_3 \\ x_3 &= \tau_3 + P_3(1 - P_K)\, x_2 + (1 - P_3)\, x_3 \end{aligned} \tag{5.7.2}$$

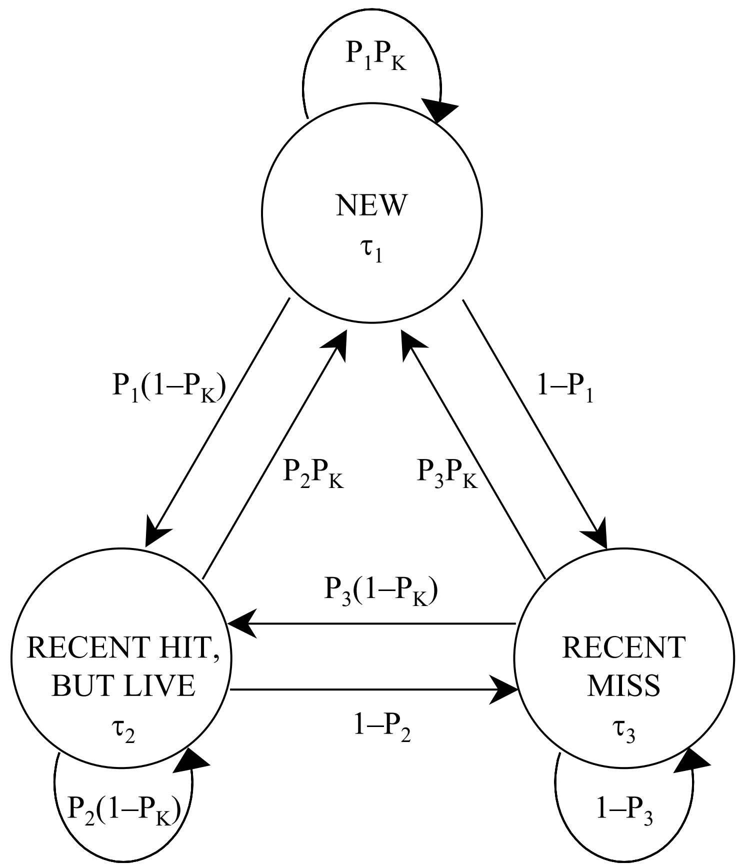


Figure 5.7.2 – Markov Dependent Fire Model

This is a system of three linear equations in three unknowns. It can be solved by first solving for x_2 and x_3, since x_1 is not involved in those equations. The solution is

$$x_2 = \frac{\tau_2 + \tau_3(1-P_2)/P_3}{P_K} \tag{5.7.3}$$

$$x_3 = \frac{\tau_3}{P_3} + (1-P_K)x_2 \tag{5.7.4}$$

After substituting x_2 and x_3 into the equation for x_1 and simplifying, the result is

$$x_1 = \tau_1 - \tau_2 + \frac{\tau_2}{P_K} + \frac{\tau_3}{P_3}\left(\frac{1-P_2}{P_K} + P_2 - P_1\right). \tag{5.7.5}$$

This is the Bonder-Farrell result (Bonder 1970)) expressing the mean time to the next target kill after allowing for acquisition and all of the other activities that must be performed and possibly re-performed along the way. The reciprocal of x_1 can be used in Lanchester models as a kill rate per shooter.

If $t_1 = t_h = t_m$, let the common value be t_s. If $P_1 = P_2 = P_3$, let the common value be P_s. If these common values, as well as $t_f = 0$, are substituted into 5.7.5, equation 5.7.1 results, so 5.7.5 is a generalization of 5.7.1.

Extensions of the Bonder/Farrell equation: The basic equation derived by Bonder and Farrell has been enriched (from its origin as a model for the MBT70 studies) to handle a variety of firing doctrines. In each, the interactions with target acquisition and target selection have been analyzed using a similar approach as above and then applied in the different Vector models. Areas that have been enriched within the basic model include lethality mechanism and firing doctrine. A breakdown of the various aspects found within these topic areas is listed in Table 5.3.

Table 5.3 – Extensions to Bonder/ Farrell Attrition Models

Lethality Mechanism

1. Area
2. Impact

Fire Doctrine

1. Repeated single shot
    a. Without feedback control of aim point
    b. With feedback on immediately preceding round
    c. With complex feedback
2. Burst Fire
    a. Without aim change or drift between bursts
    b. With aim in burst, aim returns to original aim point
    c. With aim drift, re-aim between bursts
3. Multiple-tube firing
    a. Volley fire (with same feedback loops in 1(a,b,c))
4. Mixed mode firing
    a. Adjustment followed by multiple-tube fire
    b. Adjustment followed by burst fire

Also included in the enrichment applications are probabilistic models for line of sight (LOS) and target acquisition process. LOS considers the interactions between the firer-target pairs and terrain. It is modeled as an alternating process of LOS/non-LOS where the length of each time segment is a random variable. The target acquisition or target selection process is delineated as either serial or parallel. Under serial acquisition, the firer selects a new target whenever the previously selected target is killed or LOS is lost. This corresponds with the Markov dependent fire model where t_a is manifested for each entry into state S_1. In a parallel acquisition process the firing system continues to acquire new targets while engaging a given target. Therefore after a kill or loss of sight the system immediately shifts fire to a new target (if available). In such a case $t_a = 0$ and the acquisition process does not dominate the attrition process.

Evaluation of the Bonder/Farrell Methodology: The major advantages of this approach to attrition coefficient generation are that the computation is based on what is going on in the battle at the given moment. This precludes problems of library matching and variance caused with inaccurate fit. Additionally, since the data used is basically measurable this approach is more credible to the product user. Finally, although tentative, the results achieved using this methodology provides a more consistent and believable results in both Lanchester and Potential-Antipotential models than such methods as firepower scores and analytical attrition coefficient generation methodologies. Finally, the main disadvantage is that the attrition mechanism must be fitted into one of the stylized stochastic model structures where interactions between systems are not explicitly represented within these structures.

Problems for Chapter 5

(to be determined later)

References for Chapter 5

[5.1] Bonder, S., "The Mean Lanchester Attrition Rate," *Operations Research*, 18, #1, pp. 179-181, 1970.

[5.2] Engel, J., "A verification of Lanchester's law," *Operations Researc*h, 1954, pp. 163-171.

[5.3] Morse, P. and Kimball, G., *Methods of Operations Research*, MIT Press, ch. 4, 1950.

[5.4] Ross, S., *Introduction to Probability Models*, Academic Press, ch. 6, 1997.

[5.5] Taylor, J., *Lanchester Models of Warfare*, Operations Research Society of America, Vols. 1 and 2, 1983.

CHAPTER 6

THE ATCAL ATTRITION MODEL

6.1 – Introduction to the ATCAL Attrition Model

ATCAL is an aggregated attrition model developed in the early 1980's by the Center for Army Analysis for use in the CEM and FORCEM theater level simulations. ATCAL consists of a number of equations that can be used to compute attrition (if values for several input parameters are known). The same equations can be used, "backwards" to determine values for the parameters from the output of a higher resolution division level model such as COSAGE. Thus ATCAL contains both an attrition model and a consistent calibration procedure.

The attrition equations of ATCAL are heterogeneous, computing casualties for firer-target pairings of weapon system types. There are two basic attrition equations in the methodology, one for "point fire" weapons effects and one for "area fire". In this discussion we will concentrate only on the point fire equation. Details of the area fire model can be found in [Ref 6.1].

The ATCAL model does not step through time, but rather computes the casualties for an entire division level engagement.

6.2 – Parameter Definitions For ATCAL

The following variables will be used in the presentation of the ATCAL attrition model:

N_k = the **initial number of combat vehicles** of type k at the beginning of the battle. Combat vehicles are the basic killable element in the attrition computation. To avoid two sets of notation for the two sides in the battle, we let the subscript, k range over all vehicle types for both sides and will assume that the model knows which are from each force. The numeric values for the N_k are available from the combat simulation at the start of each battle.

DN_k = the total **number of casualties to vehicles** of type k during the entire battle. This is the primary output of the attrition computation.

AN_k = the **average number** of type k vehicles alive during the battle. Since ATCAL does not compute in time steps, but rather assesses the entire engagement at once, we need the averages, AN_k, to determine the number of firing type k vehicles throughout the battle. ATCAL assumes an exponential decrease in weapon count during the battle.

$$N_T = N \exp^{-RT} \qquad (6.2.1)$$

for some decrease rate, R (ATCAL never needs to actually compute **R** or T). For such an exponential decrease, the average number of surviving vehicles satisfies the equation

$$AN_k = \frac{-DN_k}{\ln\left(1-\frac{DN_k}{N_k}\right)} \qquad (6.2.2)$$

where ln is the natural logarithm (see Problem 6.1). The attrition model will compute values for AN_k and DN_k that satisfy this equation. Since the equation is nonlinear, the computation will have to be iterative.

IM_k = the **importance** of a vehicle of type k. The importance of a vehicle is something like the IDAGAM score or value. It is used in ATCAL to help determine target priorities for fire allocation. The importance is computed from the killer-victim scoreboard during the ATCAL attrition computation using a set of value equations that are nonlinear (in contrast to IDAGAM's linear eigenvalue equations). The importance can be used to compute force ratios as an auxiliary output from ATCAL: the force ratio is not needed for the ATCAL computations proper. See Section 6.5.

P_{ijk} = **kills per round** when a vehicle of type i uses its weapon of type j to engage a target of vehicle type k. This lethality measure is an input parameter to the attrition computation. We will see later how it is computed during the calibration process. Throughout the ATCAL description, engagements will be characterized by the above three subscripts: i for the firing vehicle, j for the weapon type (vehicle i may have more than one weapon), and k for the target vehicle.

Q_{ijk} = **target priority** for targets of type k. Each firing system has its targets prioritized for purposes of fire allocation. The target priority is computed during the attrition calculation as a combination of importance and vulnerability of the target to the firing system:

$$Q_{ijk} = P_{ijk} * IM_k. \qquad (6.3)$$

A_{ijk} = the **target availability** is the fraction of the time that a single particular target of type k can be fired upon by a particular type i vehicle using weapon j. This availability, which can be considered to be averaged over all targets of type **k** and firers of type i, is a model input whose values are determined by the calibration procedure.

$RATE_{ij}$ = the **total number of shots** that a single type j weapon on a type i vehicle is capable of firing during the entire duration of the battle (assuming targets are always available). The value should be considered as an average over all firers of type i,j, computed so that $AN_i * RATE_{ij}$ = the total firing capability of all type i,j firers in the engagement. This is computed outside the simulation by a rather complex process involving several high-resolution simulation runs. We will not consider the details, but will just assume that $RATE_{ij}$ values are input.

F_{ijk} = the **number of shots fired** by weapons of type **j** on type i vehicles at type k target vehicles. This is computed by the attrition model as the result of its fire allocation equation.

(DN_k) = the **killer victim scoreboard** is the detailed result of the ATCAL attrition equation. It shows casualties to type k vehicles by the systems (i,j) that caused them. The casualties are computed as:

$$(DN_k)_{ij} = F_{ijk} \quad P_{ijk} \tag{6.2.4}$$

$(DN_k)_{ij}$ can be summed over i and j to yield the total casualties, DN_k.

6.3 – Computing Attrition with the ATCAL Model

The ATCAL attrition process model contains two main attrition equations. In this section we will develop the point fire attrition equation and show how it fits into the overall attrition computation process. The corresponding area fire attrition equation is used in roughly the same way (although the equation itself is different); we will not discuss its details.

Development of the Point Fire Attrition Equation

Suppose that we have values for the target availability fractions, A_{ijk}, and the average number of targets, AN_k.

1. *Assume that availability of each of the AN_k targets is independent of availability for the other type targets.* Then the fraction of the time that no type k target is available for a particular firer of type i,j is

$$(1 - A_{ijk})^{AN_k} \tag{6.3.1}$$

and thus the fraction of the time that a type k target is available is

$$1 - (1 - A_{ijk})^{AN_k} \tag{6.3.2}$$

2. *Assume that a firing weapon will shoot at the highest priority target available.* Then, if type k is the highest priority for i,j, we can compute the number of shots fired at vehicles of type k by all weapons of type j on vehicles of type i as

$$F_{ijk} = AN_i \quad RATE_{ij} \; [1 - (1 - A_{ijk})^{AN_k}]. \tag{6.3.3}$$

Now consider the case where type k is not necessarily the highest priority target.

3. *Assume that availability of targets of type k is independent of availability of any other target type k'.* Then, if firing at type k targets goes on only during the fraction of the time when no higher priority target is available, we can compute the general form for

$$F_{ijk} = AN_i \quad RATE_{ij} \; [1 - (1 - A_{ijk})^{AN_k}] \prod_{\text{all } k'} (1 - A_{i'jk})^{AN_{k'}} \tag{6.3.4}$$

where the subscript k' for the product ranges over all target types whose priority is higher than the priority of target type k.

Finally, the attrition to type k targets caused by all firers of type i,j is given by

$$(DN_k)_{ij} = P_{ijk} \; F_{ijk}$$

$$= P_{ijk} \; AN_i \; RATE_{ij} \; [1 - (1 - A_{ijk})^{AN_k}] \prod_{\text{all } k'} (1 - A_{i'jk})^{AN_{k'}} \qquad (6.3.5)$$

This is the basic point fire attrition equation for the ATCAL attrition model.

The attrition equation combines features of both the Lanchester Square Law and the Lanchester Linear Law. If type k targets are scarce, (either A_{ijk} or AN_k small) then the attrition from equation (6.3.5) is proportional to $AN_i * AN_k$, thus showing Lanchester linear behavior. When type k targets are plentiful, the attrition is proportional to AN_i, the number of firers, thus displaying the behavior of a Lanchester Square Law equation (see Problem 6.2).

<u>The ATCAL Attrition Computation Procedure</u>

The attrition computation in ATCAL is an iterative procedure because the basic equation requires the average number of combatants, AN_i, to compute casualties, but AN_i depends on the number of casualties. Similarly, the computation process needs the vehicle importance, IM_k, to compute target priorities for the attrition equation, but the importance values depend on the attrition because they are computed from the killer-victim scoreboard. The steps of the iterative procedure are listed below.

1. **Initialization** – Set the initial guess for average numbers of vehicles at $AN_i = N_i$ and make an initial guess for the importance values IM_i. Both of these will change at each iteration of the method.

2. **Availabilities** – Scale the input A_{ijk} values to account for the width of the engagement front. See Section 6.4 for further discussion of this scaling.

3. **Priorities** – Compute the target priorities for this iteration as $Q_{ijk} = P_{ijk} * IM_k$.

4. **Attrition** – For each firing weapon, apply the attrition equation to all of its targets in priority order to compute the casualties $(DN_k)_{ij}$. Either the point fire Equation (6.3.5) as developed above or the similar area fire equation will be used. As attrition is computed, the number of rounds fired is also accumulated. If a weapon's ammunition stockpile is exhausted, then targets at the bottom of its priority list will not be attacked.

5. **Update Average Numbers** – When all the weapons on a side have been processed, then the total casualties for each target weapon type are compared with the total number of targets to eliminate overkill, and the average number of vehicles AN_k is updated to be consistent with the new casualty figures for the vehicle type. Equation (6.2.2) is used for this update, and the computations are organized to smooth the convergence of the entire algorithm.

6. **Update Importance Values** – Each time a side is processed, a new killer-victim scoreboard is produced. This can be used to update the importance values, IM_i, for the firing side using the nonlinear value equations that will be explained in Section 6.5.

7. **Check for Convergence** – When all the weapons on both sides have been processed, check for convergence of the iteration by comparing the new AN_k values with those from the previous iteration. If all AN_k, are within a tolerance of their previous values, then the attrition computations are completed, otherwise return to Step 3 for the next iteration.

6.4 – Calibrating the ATCAL Parameters

One of the attractive features of the ATCAL attrition model is that it has a calibration procedure that uses the same equations as the attrition algorithm. Thus the calibration of the algorithm is likely to provide consistent results.

The ATCAL calibration procedure works on the output of a high-resolution division level simulation like COSAGE to produce numeric values for some of the parameters of the attrition equations. A successful calibration will allow the aggregated attrition equations to produce consistent output if they are applied to the same calibrating scenario.

In addition, the ATCAL attrition algorithm is designed so that the same set of attrition parameters can be used for other scenarios in which the force sizes or the force mix are different from the calibrating scenario. The deviations from the calibrating scenario cannot, however, be too severe. For example, introducing a weapon or vehicle that was not present in the calibrating force would not be acceptable. Similarly, conducting a battle under different terrain or visibility conditions or with a different combat posture would require a new set of calibrated parameters since all of these factors are implicitly present in the attrition parameters.

Output of the High-Resolution Simulation

The following output data is required from the high-resolution combat simulation:

N_k = the initial **number of combat vehicles** of type k at the beginning of the battle.

$(DN_k)_{ij}$ = the **killer victim scoreboard** is the total number of casualties to type k vehicles during the entire battle that were caused by all firers of type (i,j).

F_{ijk} = the **number of shots fired** by weapons of type j on type i vehicles at type k target vehicles during the entire engagement.

$RATE_{ij}$ = the total number of shots that a single type j weapon on a type i vehicle is capable of firing during the entire duration of the battle (assuming targets are always available). $RATE_{ij}$ is not actually a direct output from the high-resolution model, but is computed outside the simulation by a rather complex process involving several high-resolution simulation runs.

$RANGE_{ij}$ = the **average engagement range** for weapons of type (i,j).

WIDTH = the **width of the combat front** for the engagement.

Calibration Computations

In this section we will describe the calibration computations for the point fire attrition equation.

1. Compute total casualties to vehicles of type k by summing the killer-victim scoreboard values:

$$DN_k = \sum_{\text{all } i,j} (DN_k)_{ij} . \qquad (6.4.1)$$

2. Compute **average number of vehicles** from Equation (6.2.2):

$$AN_k = \frac{-DN_k}{\ln\left(1-\frac{DN_k}{N_k}\right)}. \quad (6.4.2)$$

3. Compute the **kills** per **round** as:

$$P_{ijk} = \frac{(DN_k)_{ij}}{F_{ijk}}. \quad (6.4.3)$$

4. Compute **vehicle importance**, IM_k, from N_k and $(DN_k)_{ij}$ using an iterative solution of the nonlinear value equations as described in Section 6.5.

5. For each shooter type (i,j), compute and rank order the **target priorities**

$$Q_{ijk} = P_{ijk} \quad IM_k. \quad (6.4.5)$$

6. For each shooter type (i,j), compute **target availabilities**, A_{ijk}, in priority order using the following formula which is consistent with the availability assumptions in Section 6.3.

$$A_{ijk} = 1 - \left[1-\frac{F_{ijk}}{\left(AN_i \quad RATE_{ij} - \sum_k\right)}\right]^{\left(\frac{1}{AN_k}\right)} \quad (6.4.6)$$

where $\sum_k$, is the total number of rounds fired by weapon j on all vehicles of type i at all targets whose priority is higher than that of target vehicle k.

7. Scale the availabilities into a **frontage independent** form by dividing each A_{ijk} by the factor

$$1-\exp^{\left(-\frac{RANGE_{ij}}{WIDTH}\right)}. \quad (6.4.7)$$

The same factor is used as a multiplier (with possibly a different front **WIDTH)** before the A_{ijk} are used in the attrition computations.

The Resulting Aggregated Attrition Parameters

The following calibrated attrition parameters are carried over into the aggregated attrition model:

1. The **target availability parameters,** A_{ijk}, frontage independent.
2. The **engagement ranges** $RANGE_{ij}$, for scaling the target availability parameters.
3. The **kills per round,** P_{ijk}.
4. The **firing capability** parameters, $RATE_{ij}$.

All other attrition model variables are computed during the attrition iterations as indicated in Section 6.3.

6.5 – Nonlinear Equations for Weapon Importance

The weapon importance values in ATCAL are derived using the same sort of circular reasoning as is used in the IDAGAM eigenvalue scoring method. The equations used by ATCAL are nonlinear equations developed as an improvement of the linear eigenvalue equations of IDAGAM with the express purpose of eliminating the observed anomalies of the eigenvalue solutions. It is felt that the problems with the eigenvalue method stem from imposing a linear set of equations on a nonlinear combat process.

ATCAL importance values are derived from the killer-victim scoreboard $(DN_k)_{ij}$. Before the equations are set up, we modify the form of the killer-victim entries by computing

$$DN_{ik} = \sum_{\text{all } j} (DN_k)_{ij} \tag{6.5.1}$$

thus obtaining vehicle to vehicle casualties, and

$$C_{ik} = \frac{DN_{ik}}{(N_k)(DURATION)} \tag{6.5.2}$$

where DURATION is the total battle time. The resulting C_{ik} values are fractional kills per unit time or **fractional casualty rates** for each combination of a type i firing vehicle and a type k target vehicle.

Let X_i be defined as the importance of all shooters of type i at the start of the battle. The nonlinear importance equation system defines the value of each firing vehicle type as:

$$X_i^3 = \sum_{\text{all } k} \frac{C_{ik}^3 \quad X_k}{\sum_{\text{all } i} C_{ik}} . \tag{6.5.3}$$

Then we can compute the importance of each vehicle of type i as:

$$IM_i = \frac{X_i}{N_i} \tag{6.5.4}$$

and the average importance of vehicle type i in the battle as:

$$AX_i = \frac{X_i \quad AN_i}{N_i} . \qquad (6.5.5)$$

The nonlinear importance Equation (6.5.3) can be solved iteratively. An arbitrary positive initial guess for the X_k, for a side can be plugged into the right hand side (RHS) of the equation to provide an updated estimate of X_i for the other side. This estimate can, in turn be plugged into the RHS to produce new estimates for the first side. Repeating the process should eventually converge to stable X_i for all vehicles on both sides.

Force ratios for the battle can be computed simply as the ratio of total value for the two sides:

$$TI_a = \sum_{\substack{\text{all i on the} \\ \text{attacker side}}} X_i \qquad (6.5.6)$$

$$TI_d = \sum_{\substack{\text{all i on the} \\ \text{defender side}}} X_i \qquad (6.5.7)$$

$$\text{INITIAL-FR} = \frac{TI_a}{TI_d} \qquad (6.5.8)$$

since X_i gives the importance for all type i vehicles together. Similarly, an average force ratio for the battle will result if the AX_i are summed. The ATCAL attrition model does not use these force ratios, but they might be useful for making decisions elsewhere in the combat simulation.

The particular form of the nonlinear importance equations can be explained in terms of eliminating enemy importance. The discussion is rather intricate and can be found in [Ref. 6.2].

Problems for Chapter 6

6.1 Assume an exponential decrease in the number of vehicles of type k as in Equation (6.2.1). Show that the average number of vehicles satisfies Equation (6.2.2).

6.2 Show mathematically that the point fire attrition Equation (6.3.5) behaves like a Lanchester linear equation when targets are scarce and like a Lanchester square equation when targets are plentiful. Hint – consider a binomial expansion of $(1–A)^N$.

References for Chapter 6

[6.1] Kerlin, E. P., and R. H. Cole, ATLAS: A Tactical, Logistical, and AIR Simulation, Technical Paper RAC-TP-338, Research Analysis Corporation, April 1969.

[6.2] ATCAL: An Attrition Model Using Calibrated Parameters, Technical Paper CAA-TP-83-3, Center for Army Analysis, August 1983.