2002-03

An exploratory analysis on the effects of information superiority on battle outcomes

Pee, Eng Yau.

Monterey, California. Naval Postgraduate School

https://hdl.handle.net/10945/6035

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

AN EXPLORATORY ANALYSIS ON THE EFFECTS OF INFORMATION SUPERIORITY ON BATTLE OUTCOMES

by

Pee Eng Yau

March 2002

Thesis Advisor: Thomas W. Lucas
Second Reader: Walter L. Perry

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

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

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** March 2002	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE: Title (Mix case letters) An Exploratory Analysis on the Effects of Information Superiority on Battle Outcomes.	**5. FUNDING NUMBERS**
6. AUTHOR(S) Pee Eng Yau	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) RAND. 1200 South Hayes Street, Arlington, VA 22202-5050	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

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

13. ABSTRACT *(maximum 200 words)*

Visions of future warfighting, such as Joint Vision 2020, emphasize using new technologies to obtain and exploit information advantages to achieve new levels of effectiveness in joint warfighting. Unfortunately, our warfighting models are notoriously poor at capturing the effects of information on battle outcomes. Moreover, traditional measures of effectiveness (MOEs) usually ignore the effects of information and decision making on battle outcomes. The Department of the Navy and other DoD organizations have tasked RAND to create a framework for developing measures and metrics to assess the impact of C4ISR systems and procedures on battle outcomes. In order to quantify the effects of information and decision making on battle outcomes, RAND built a deterministic model and hypothesized a scenario involving the search for, and destruction of, a time-critical target (TCT). This thesis extends their work by making the simulation stochastic and exploring practical issues such as: (i) the effects of improved C4ISR systems and procedures on battle outcomes; (ii) which messaging and data processing delay reductions give the greatest improvements in kill probability; (iii) which command and control architecture provides the highest kill probability.

14. SUBJECT TERMS C4ISR, Exploratory Analysis, Stochastic Simulation, Information Superiority, Network-Centric Warfare, Combat Modeling	**15. NUMBER OF PAGES** 124
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

AN EXPLORATORY ANALYSIS ON THE EFFECTS OF INFORMATION SUPERIORITY ON BATTLE OUTCOMES

Pee Eng Yau
Singapore Defence Science and Technology Agency
B.Eng., National University of Singapore, 1996

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
March 2002

Author: Pee Eng Yau

Approved by: Thomas W. Lucas, Thesis Advisor

Walter L. Perry, Second Reader

James N. Eagle, Chairman
Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Visions of future warfighting, such as Joint Vision 2020, emphasize using new technologies to obtain and exploit information advantages to achieve new levels of effectiveness in joint warfighting. Unfortunately, our warfighting models are notoriously poor at capturing the effects of information on battle outcomes. Moreover, traditional measures of effectiveness (MOEs) usually ignore the effects of information and decision making on battle outcomes. The Department of the Navy and other DoD organizations have tasked RAND to create a framework for developing measures and metrics to assess the impact of C4ISR systems and procedures on battle outcomes. In order to quantify the effects of information and decision making on battle outcomes, RAND built a deterministic model and hypothesized a scenario involving the search for, and destruction of, a time-critical target (TCT). This thesis extends their work by making the simulation stochastic and exploring practical issues such as: (i) the effects of improved C4ISR systems and procedures on battle outcomes; (ii) which messaging and data processing delay reductions give the greatest improvements in kill probability; (iii) which command and control architecture provides the highest kill probability.

THIS PAGE INTENTIONALLY LEFT BLANK

THESIS DISCLAIMER

The reader is cautioned that computer programs developed in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational and logic errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I want to thank my thesis advisor, Professor Tom Lucas for his guidance, support and patience throughout this endeavor. I would like to express my appreciation to Dr. Walter Perry and Tom Sullivan for their advice and suggestions on how best to implement the stochastic simulation model. I would also like to thank Professor Samuel Buttrey and CDR Matthew Boensel for their enthusiasm in answering my endless questions on Clementine and MS Excel respectively. They made my job so much easier. Lastly, I want to thank my wife for always being there to support me, to feed me when I am hungry, and cheer me up when I am feeling down.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF SYMBOLS, ACRONYMS AND/OR ABBREVIATIONS

AOU	Area of Uncertainty
CG	Cruiser
CJTF	Commander U.S. Joint Task Force
CVBG	Carrier Battle Group
C4ISR	Command, Control, Communication, Computer, Intelligence, Surveillance and Reconnaissance
DDG	Destroyer
EDA	Exploratory Data Analysis
FCW	Future Network-Centric Warfare
ISR	Intelligence, Surveillance and Reconnaissance
LHS	Latin Hypercube Sampling
NCW	Network-Centric Warfare
NDRI	National Defense Research Institute
NN	Neural Network
ONR	Office of Naval Research
PCS	Position, Course and Speed
PCW	Platform-Centric Warfare
Pk	Kill Probability
SLAM-ER	Stand-off Land Attack Missile – Extended Response
SS	Submarine
SSN	Nuclear Submarine
SubGroup	Submarine Group
TCT	Time-Critical Target
UCAV	Unmanned Combat Air Vehicle
VBA	Visual Basic for Applications

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

Visions of future warfighting, such as Joint Vision 2020, emphasize using new technologies to obtain and exploit information advantages to achieve new levels of effectiveness in joint warfighting. Unfortunately, our warfighting models are notoriously poor at capturing the effects of information on battle outcomes. Moreover, traditional measures of effectiveness (MOEs) usually ignore the effects of information and decision making on battle outcomes. To address this shortcoming, the Department of the Navy and other DoD organizations have tasked RAND to create a framework for developing measures and metrics to assess the impact of Command, Control, Communication, Computer, Intelligence, Surveillance and Reconnaissance (C4ISR) systems and procedures on battle outcomes.

In order to quantify the effects of information and decision making on battle outcomes, RAND hypothesized a conflict scenario and built a deterministic model based on it. The conflict scenario involves a small island country facing a large hostile neighboring country determined to annex the island. A vignette developed by RAND, based on the conflict, is selected for examination: An operation consisting of a search for and the destruction of a time-critical target (TCT), specifically an enemy KILO submarine. A TCT is a target with a limited window of vulnerability or engagement opportunity, during which it must be found, identified, targeted, and engaged. The measure of performance (MOP) for RAND's TCT vignette is the effective time remaining to conduct the search and detection mission of the KILO submarine, and the measure of effectiveness (MOE) is the kill probability (Pk) of the KILO submarine.

Three alternative operating procedures are developed to analyze the TCT vignette. They are, in the order of increasing network connectivity, better C4ISR and weapon systems, (i) Platform-Centric Warfare (PCW), (ii) Network-Centric Warfare (NCW), and (iii) Future Network-Centric Warfare (FCW) operations.

This thesis extends RAND's work by developing a stochastic simulation model for the TCT vignette, benchmarking it against the existing deterministic model, and utilizing it to explore practical issues such as: (i) the effects of improved C4ISR systems and procedures on battle outcomes, specifically Pk in the TCT vignette; (ii) which messaging and data processing delay reductions give the greatest improvements in Pk; (iii) which command and control architecture provides the highest Pk.

A. BENCHMARKING

Six sets of inputs are supplied to both the deterministic and stochastic model, and the results are compared. The developed stochastic simulation model generally produces consistent results with the deterministic model, i.e., low Pk (MOE) in the stochastic model goes with low Pk in the deterministic model, and vice versa. Having said that, the mean of the stochastic outputs should not be expected to match up exactly to the deterministic output—this is a consequence of the nonlinear transfer function from RAND's framework of measures and metrics.

For any set of search and detection parameters, Pk rises rapidly from zero to close to one within a small range of effective time remaining (0 hour to some "threshold" value). When the mean effective time remaining is significantly higher than the "threshold" value, both the deterministic and stochastic models produce consistently high Pks. The deterministic and stochastic Pks start to deviate when the mean effective time remaining drops near, or even below the "threshold". In general, deterministic and stochastic models produce the same results only when the results are clear.

B. NETWORK CENTRICITY COMPARISON

A key objective of this thesis is to assess the effects of improved C4ISR systems and procedures on battle outcomes. What this translates to in the TCT vignette case study is, based on RAND's framework of measures and metrics, do Future Network-Centric systems and procedures produce higher kill probability (Pk) than Platform-Centric or Network-Centric systems and procedures?

A variant of Latin Hypercube Sampling (LHS) is used to generate the input sets for comparing the three operating procedures. The stochastic simulation results (Figure 1) show that Future Network-Centric systems and procedures produce significantly higher Pks than the Platform-Centric and Network-Centric cases. The results confirm the potential of RAND's framework of measures and metrics in modeling the general effects of C4ISR systems and procedures on battle outcomes. What remains to be done is the calibration and validation of the framework, i.e., fine-tuning the framework to achieve results that are consistent with the real world.

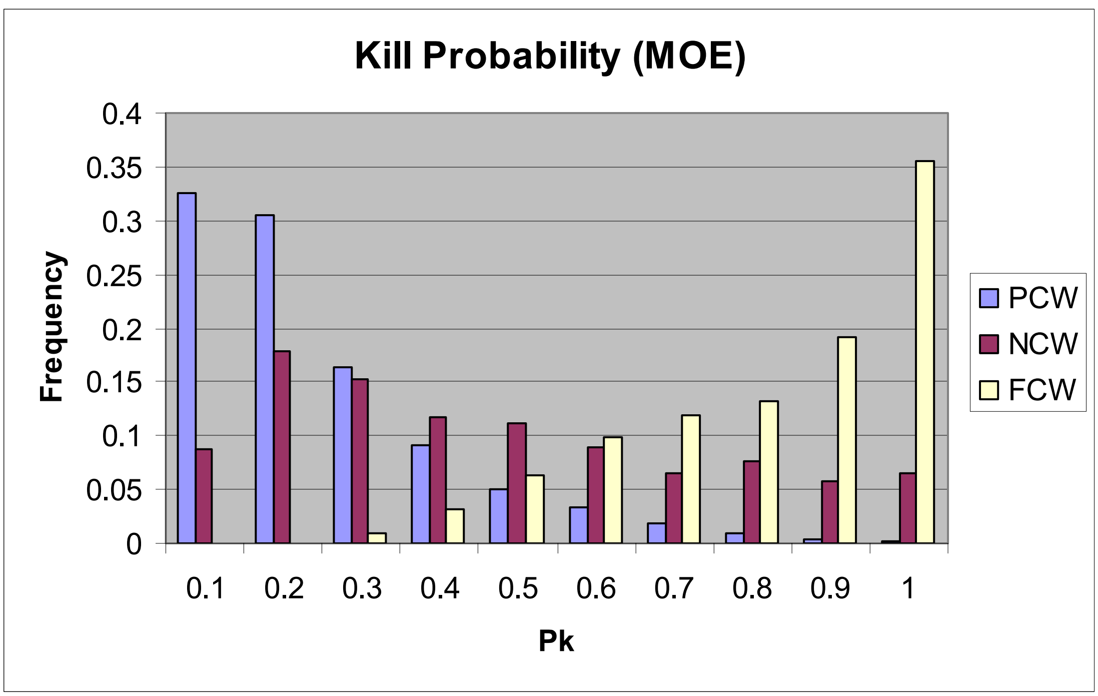

Figure 1. Network Comparison in Kill Probability. The Future Network-Centric (FCW) systems and procedures produce significantly higher Pks than the Platform-Centric (PCW) and Network-Centric (NCW) cases.

C. CRITICAL INPUT VARIABLES

Which messaging and data processing delay reductions give the greatest improvements in kill probability (Pk) for RAND's TCT vignette? Three data mining models are used to determine the variables that have the greatest impact on Pk, and to extract any interesting patterns/relationships from the stochastic simulation data. Data mining offers a strategic approach to finding useful relationships in large data sets. All three data mining models arrive at the same conclusion, specifically the critical variables in the time-critical target vignette, Future Network-Centric system, are the Strike/UCAV latency, initial SSN report latency, DDG latency, and enemy submarine submerge time. One of the interesting patterns extracted from the simulation results is shown in Figure 2. As stated earlier, Strike/UCAV latency and the initial SSN report latency are critical variables that have a great impact on Pk. However, what is implied in Figure 2 is a stronger statement, i.e., if the Strike/UCAV and initial SSN report latencies lie within the triangle shown, **regardless** of the values (within the bounds defined) of the other input variables, Pk ≥ 0.8.

D. POLLING OPTIONS FOR FCW

How should platforms be assigned to launch the Unmanned Combat Air Vehicle (UCAV) in the Future Network-Centric system? This is essentially a command and control question that addresses the way the richly-connected network is utilized to support combat operations. There are three alternative polling options, and each requires different times for collaboration and UCAV fly out in the TCT vignette. Analysis on the simulation results shows no significant differences between the three.

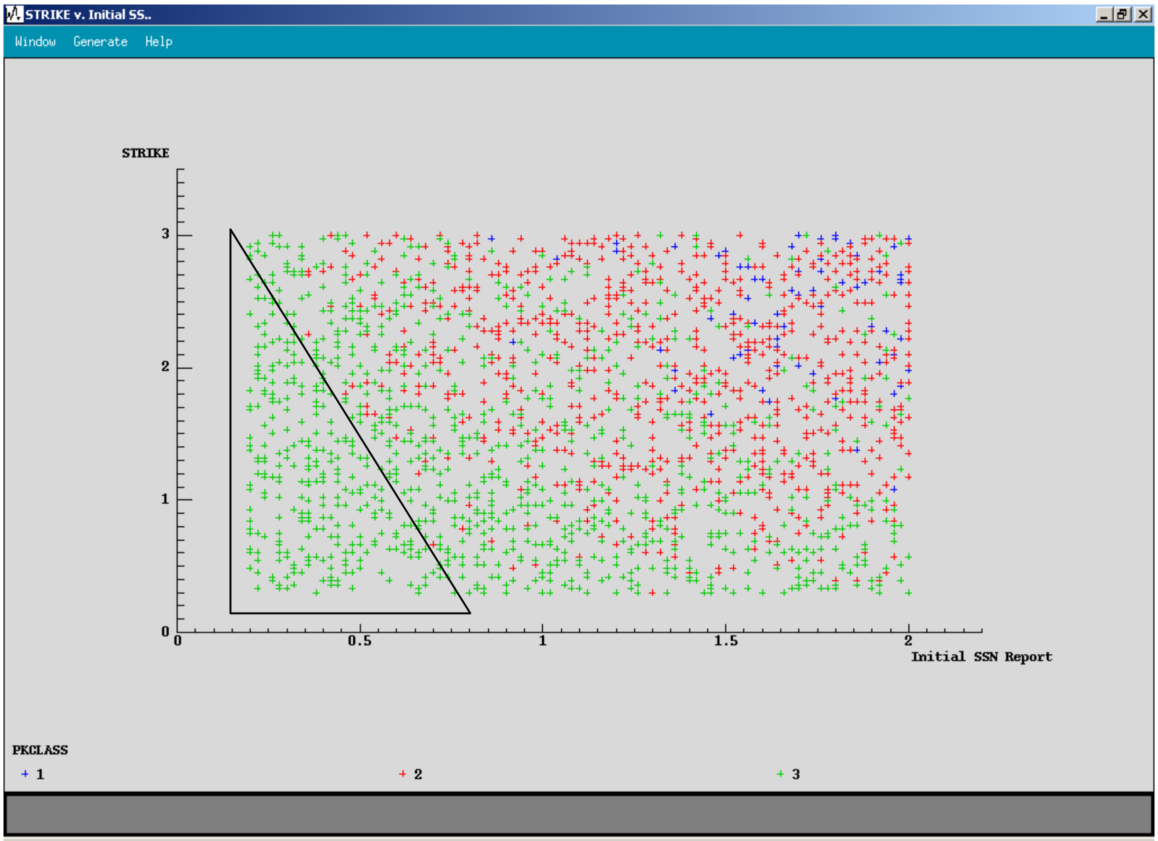

Figure 2. Strike/UCAV vs. Initial SSN Report Plot. As long as the Strike/UCAV and initial SSN report latencies lie within the triangle shown, **regardless** of the values of the other input variables, Pk ≥ 0.8.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A key element of Joint Vision 2020 is "decision superiority"—translating information superiority into better decisions arrived at and implemented faster than an enemy can react. To that end, the National Defense Research Institute (NDRI) at RAND has been tasked by the Assistant for Strategic Planning (N6C), Department of the Navy, and Office of the Chief of Naval Operations (CNO), to create a framework for developing measures and metrics to assess the impact of Command, Control, Communication, Computer, Intelligence, Surveillance and Reconnaissance (C4ISR) systems and procedures on information superiority; and more importantly, battle outcomes. This is a first attempt to create such a link between C4ISR systems and procedures and battle outcomes for the Navy.

A. BACKGROUND

The primary objective of RAND's work is to create a framework for developing measures and metrics that adequately assess the impact of improved (or degraded) C4ISR systems and procedures on battle outcomes. In the process, example measures and metrics are suggested that purport to achieve this goal. These are presented with the idea of generating dialog in the Naval and C4ISR communities concerning the framework and the measures and metrics suggested.

Although measures are simply bases or standards of comparison, and can therefore, be described qualitatively, metrics must be mathematical expressions that allow us to evaluate, not only the relative effect of alternative C4ISR systems on battle outcomes, but also the degree to which one is better or worse than another. This argues for strict mathematical formulations that produce the expected results. It is important to note however, that the process suggested by RAND is deductive; i.e., none of the equations are based on experimental or operational data. Validation remains an essential task for future work.

Traditional measures of effectiveness (MOEs) usually ignore the effects of information and decision-making on battle outcomes (Reference 1). C4ISR operations have been analyzed separately, and their effects on battle outcomes have usually been inferred rather than directly assessed. For RAND's study, an important part of their work is to create an appropriate naval warfare scenario, whereby the effects of information and decision making on battle outcomes can be quantified.

The conflict scenario hypothesized involves a small island country facing a large hostile neighboring country determined to annex the island. The conflict is set 10 years into the future to provide time to implement emerging C4ISR systems and procedures, as well as emerging Navy systems. The fact that the primary attack routes are over water implies a significant naval component. The U.S. role in the conflict is to enhance the island's defensive capabilities against enemy missile attacks by attacking enemy launchers and intercepting their missiles in flight. There is no desire for the U.S. to attack the enemy's territory. Two carrier battle groups (CVBGs) are dispatched, one to the north, and another to the south end of the island. Cruisers working in pairs are assigned to ballistic missile defense duty off the island's two major ports, and nuclear submarines (SSNs) are assigned to attack enemy interdiction submarines. One of RAND's vignettes based on the conflict scenario is selected for detailed study: An operation consisting of a search for and the destruction of a time-critical target (TCT).

A TCT is a target with a limited window of vulnerability or engagement opportunity, during which it must be found, identified, targeted, and engaged. The focus of the TCT analysis is on the development of mathematical relationships that link Network-Centric operations, command and control, combat operations, and battle outcomes. The first two focuses on the measure of performance (MOP), effective time on target, and the latter two focuses on the MOE, kill probability. In developing the combined metric:

a. Graph theory is used to assess the network connectivity, i.e., determine the number of nodes and connections in the command, control and communications network supporting the mission. More nodes in the TCT network may lead to the positive effects of collaboration, or the negative effects of complexity. Collaboration enhances the degree of shared awareness in the network, whereas complexity is the result of too much information being made available to the Task Force nodes resulting in what is generally referred to as “information overload”.

b. Information theory is used to quantify the degree of knowledge present and how it affects kill probabilities.

c. Search theory is used to determine the detection probability of the TCT.

RAND suggests a three-step exploratory data analysis method for evaluating the MOP and MOE from the TCT vignette:

a. **Phase 1 – An introductory visual exploration:** This allows all inputs to occur with equal probability.

b. **Phase 2 – A focused analysis:** The objective is to restrict the exploration to ranges of input variables that are more likely to occur.

c. **Phase 3 – A full-scale stochastic simulation:** The simulation does not use the expected values of known distributions, but randomly draws from the distributions at each simulation iteration.

Exploratory data analysis (EDA) is an approach developed by John Tukey (1977). EDA takes an open-minded, exploratory attitude towards data, employing graphical techniques to find useful relationships and patterns within the data. EDA differs from traditional analysis in the way the model is used. In exploratory analysis, the model is run many times with varying input levels, as opposed to the traditional approach of running the best-estimate case followed by sensitivity analysis.

The RAND EDA tool is implemented in an Excel spreadsheet. The spreadsheet model enables the analyst to generate hundreds of alternatives based on varying operating procedures. Prior to this thesis, RAND’s EDA tool supported only Phases 1 and 2 of the EDA process.

B. OBJECTIVE AND SCOPE

The purpose of this thesis is to assess the effects of improved C4ISR systems and procedures on battle outcomes, using stochastic simulation (Phase 3 of the EDA process).

To do this, the RAND EDA tool is extended to include stochastic simulation capabilities, and the TCT vignette is used as a case study for the assessment.

The stochastic simulation model developed is then used to answer three questions that RAND and their Navy sponsors are interested in:

a. Does improved C4ISR systems and procedures produce a quantifiable improvement in the battle outcome, i.e., does kill probability increase in the TCT vignette?

b. Which are the critical processing and messaging delay times that impact kill probability the most?

c. How should platforms be assigned to launch the UCAV in the Future Network-Centric system?

With the new stochastic simulation portion of the EDA tool, three important areas of concern that could not be addressed previously now can be:

a. Real-world outcomes—Each input and MOE should belong to a finite set of possible real-world outcomes, e.g., we either manage to kill the target or we do not. That is, we do not kill fractional targets as is done in deterministic models.

b. Variability—The current EDA tool uses expected values for the stochastic input variables, which produces a single output for the effective time on target (MOP) and kill probability (MOE). The use of expected values for the stochastic input variables, instead of their true distribution will often generate biased outcomes, which might lead to poor decision-making.

c. Extreme values analysis—In an analysis, extreme outcomes often provide answers to our questions. For example, what causes a failure? Are there simple but effective ways to push the marginal failure cases into the pass region? This analysis is sometimes impossible using expected values.

C. ORGANIZATION OF THESIS

Chapter I introduces the thesis, it provides the background of the thesis and the work done so far by RAND. The objective and scope of the thesis are clearly indicated in the chapter. In Chapter II, the TCT vignette hypothesized for the analysis is fully described. The basic theories behind the formulas used in the development of mathematical relationships that link Network-Centric operations, command and control, combat operations, and battle outcomes are provided. Chapter III focuses on the developmental process of the simulation portion of RAND's EDA tool. The formulas implemented in the Excel spreadsheet are documented and explained. The simulation portion of RAND's EDA tool is benchmarked against the deterministic portion. In Chapter IV, the EDA results/findings from the stochastic simulation are discussed. The last chapter, Chapter V concludes by highlighting the important findings of the thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

II. TCT VIGNETTE AND FORMULAS OVERVIEW

Part of RAND's study to quantify the effects of information and decision making on battle outcomes was to create an appropriate naval warfare scenario. The description of the conflict scenario and the vignette chosen for detailed analysis constitutes the first half of this chapter. The second half of the chapter lays down the theories behind the formulas used in developing the mathematical relationships between C4ISR systems and procedures, and battle outcomes. Most of the materials presented in this chapter are extracted from the RAND study report (Reference 1).

A. TCT VIGNETTE

The conflict scenario hypothesized involves a small island country facing a large hostile neighboring country determined to annex the island. A vignette developed by RAND, based on the conflict, is selected for examination: An operation consisting of a search for, and the destruction of a time-critical target (TCT). This thesis focuses on the TCT vignette, particularly the development of mathematical relationships that link Network-Centric operations, command and control, combat operations, and battle outcomes.

A TCT is a target with a limited window of vulnerability or engagement opportunity, during which it must be found, identified, targeted, and engaged. RAND's TCT vignette (Reference 1) starts on day D+6, with a U.S. Virginia class nuclear submarine (SSN) beginning a previously planned Intelligence, Surveillance and Reconnaissance (ISR) mission off the enemy's coast. On D+10, the ISR SSN detects an enemy KILO submarine leaving port, and it starts tracking the KILO. The U.S. plan is to kill the KILO on the surface as it emerges from the port without revealing the ISR submarine or disrupting its mission. A surfaced submarine is highly vulnerable. Submerging increases the difficulty of detecting, classifying, localizing, and killing it. When the SSN report gets through the network, an F/A-18 fighter attack aircraft is vectored to the KILO and will try to kill it using a SLAM-ER (Stand-Off Land Attack Missile – Extended Response) missile.

Three alternative operating procedures are developed to analyze this problem. They are, in the order of increasing network connectivity, better C4ISR and weapon systems, (i) Platform-Centric Warfare (PCW), (ii) Network-Centric Warfare (NCW), and (iii) Future Network-Centric Warfare (FCW) operations.

In the Platform-Centric case (Figure 3), the ISR SSN will report up the chain of command to the Submarine Group (SubGroup) commander, who will then alert the CVBGs that a threat submarine has left port. A previously designated F/A-18 on one of the two carriers, CV and nuclear CV (CVN) flies out to attack the KILO from outside of the enemy's surface-to-air missile (SAM) envelope using a SLAM-ER missile. The ISR SSN will continue to provide updates on the KILO's position, course and speed (PCS). Command and control in this Platform-Centric case is split awkwardly between the SSN and Air Operations on the carrier, and there is no direct communication between the two.

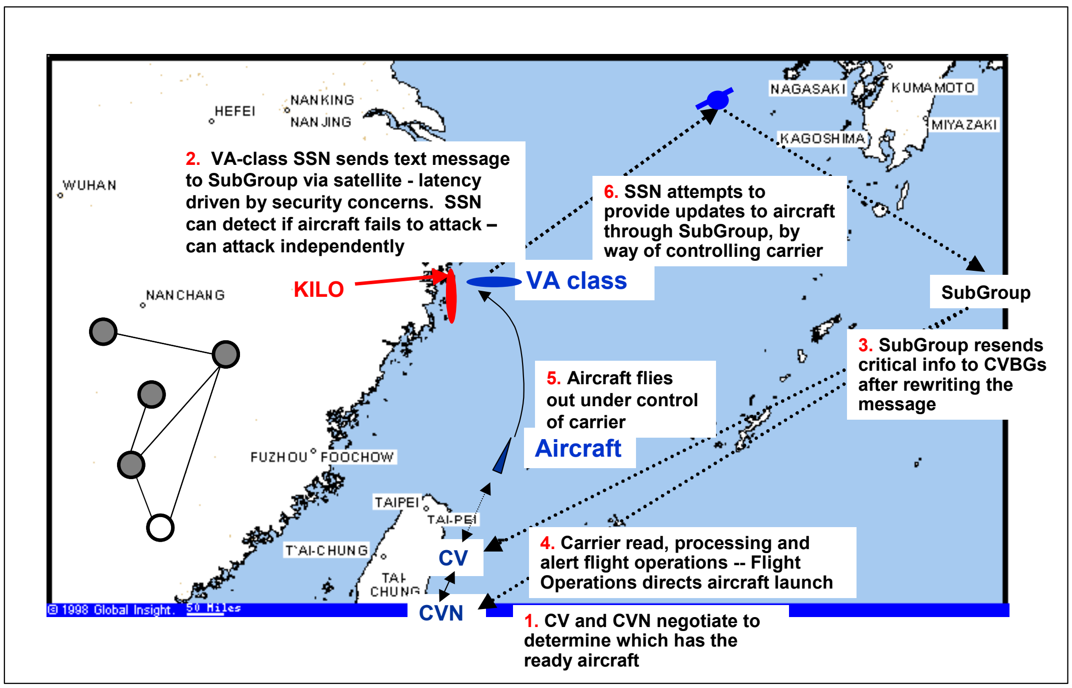

Figure 3. Platform-Centric Operations. The key disadvantage with the Platform-Centric case is the long messaging delays between the ISR submarine and the F/A-18.

In the Network-Centric case (Figure 4), the connectivity among the participants is richer. The ISR SSN has two-way communications to the carriers and the deploying aircraft. This removes the delay time for the SubGroup to relay messages. The F/A-18 receives periodic target updates directly from the ISR submarine. The command and control architecture has the same division as the Platform-Centric case, i.e., the F/A-18 is still under the command and control of the CVBG, and the ISR SSN still reports to the SubGroup commander, however, with the direct communication link between the ISR SSN and the F/A-18, the messaging delay time can be reduced.

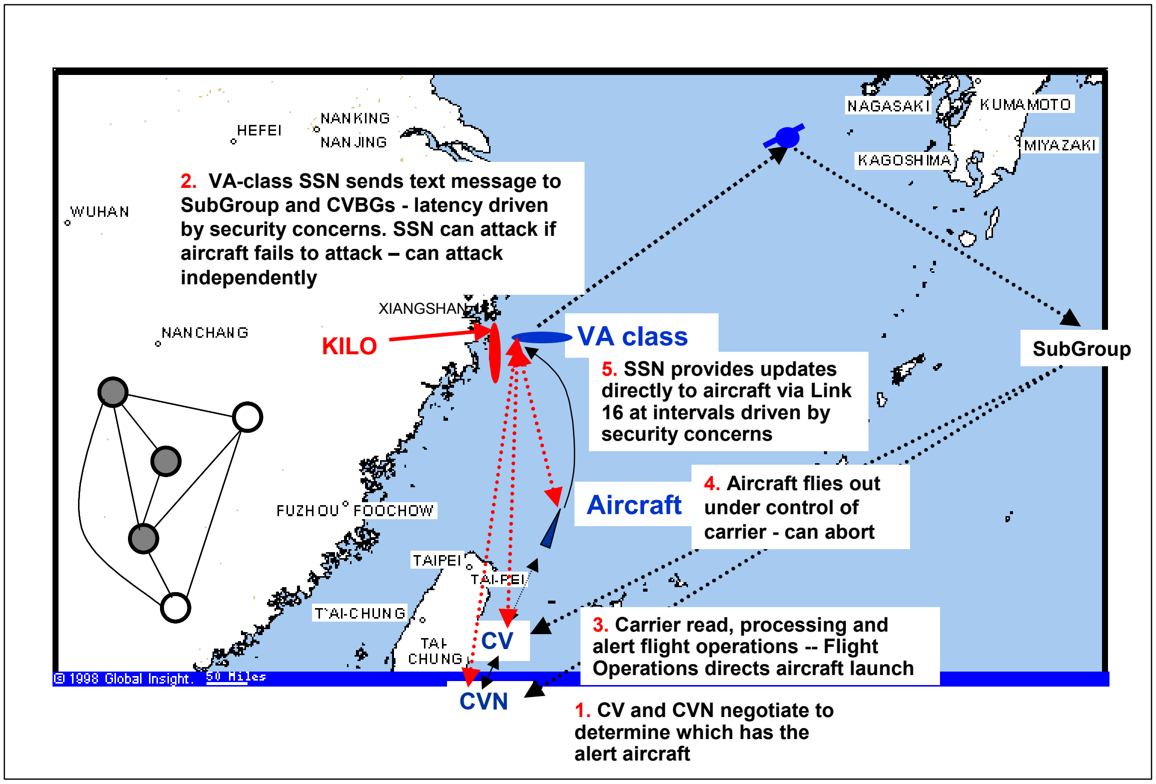

Figure 4. Network-Centric Operations. With a direct communication link between the ISR SSN and the F/A-18, the messaging delay time can be reduced.

In the Future Network-Centric case (Figure 5), an Unmanned Combat Air Vehicle (UCAV) replaces the F/A-18. UCAVs are designed to be launched from a variety of surface combatants. When the ISR submarine detects the KILO, it alerts all potential UCAV launch ships. Command and control procedural questions that need to be addressed include: Who determines which combatants are candidates to launch the UCAVs? Who makes the final selection of which ship to launch the UCAV, etc? The ships receiving the message negotiate to determine which can get a UCAV to the KILO first. A UCAV is then launched and begins its flyout to the KILO area of uncertainty (AOU). The ISR submarine takes over control of the UCAV, including weapon release.

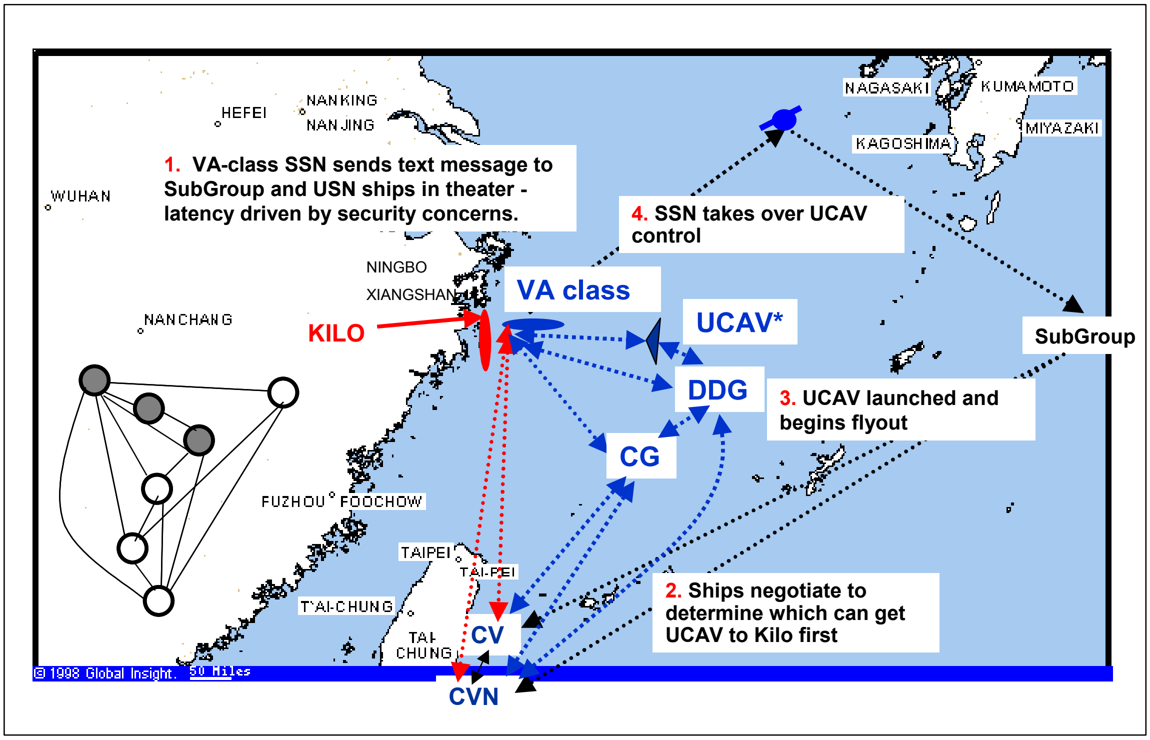

Figure 5. Future Network-Centric Operations. Unmanned Combat Air Vehicle (UCAV) replaces the F/A-18.

B. FORMULAS OVERVIEW

The measure of performance (MOP) is the expected amount of time the F/A-18 or UCAV will have to detect, acquire and destroy the target. The measure of effectiveness (MOE) is the probability that the weapon will kill the target given the amount of time to search and acquire it. The derivation of the formulas used to determine the MOP and MOE constitutes the rest of this chapter.

1. Graph Theory

We begin by describing the command, control and communications network supporting the operation as an abstraction of an undirected graph. Consider a notional network that consists of n nodes, with m connections.[1] Of the n nodes in the network, however, only τ are involved in the current operation. For example, Figure 6 illustrates a network with 10 nodes but only 13 connections. The shaded nodes represent those

[1] By connection we mean that the "connected" nodes are able to communicate to each other directly. This does not necessarily mean that there is a physical connection between the two, only that a communication channel exists. Whether it is a direct link or a relayed link is immaterial.

involved in the operation. This is typically the structure of operational networks. Not all potential operational elements are connected and not all are involved in the current operation. Some interesting relationships arise from this topology however.

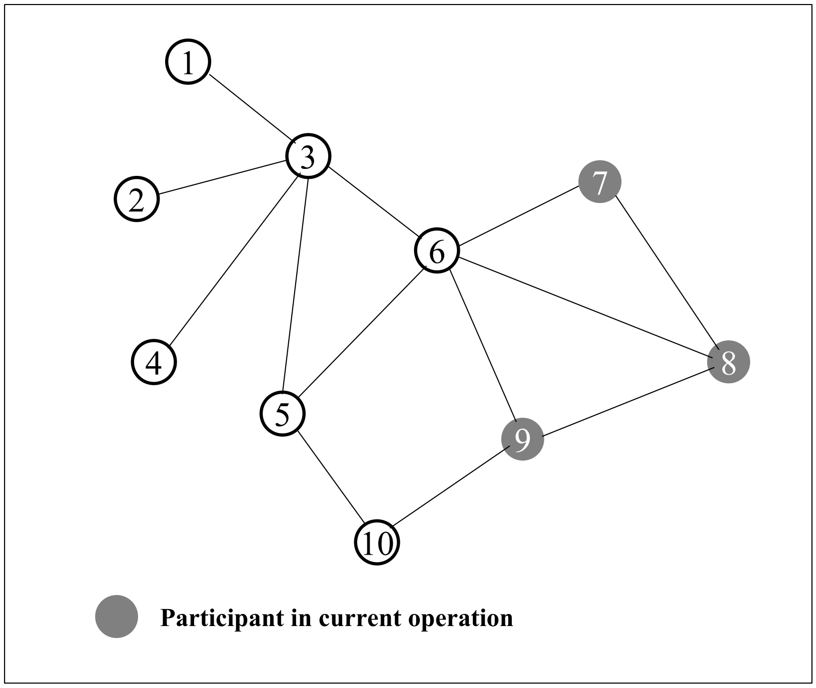

Figure 6. Notional Operating Network. The maximum number of connections in the network is $\binom{10}{2} = 45$.

First, we note that the maximum number of connections in a network with n nodes is:

$$\binom{n}{2} = \frac{n(n-1)}{2} \tag{1}$$

Thus $m \leq \frac{n(n-1)}{2}$. In Figure 6, we have a maximum of 45 possible connections. If all were connected, the graph representing the network would be complete.

Secondly, it is important to analyze the role of connected facilities not directly involved in the operation. For example, nodes 6 and 10 are connected to node 9. If node 9 were the Commander of the U.S. Joint Task Force (CJTF) controlling the operation, then 6 and 10 might be information sources (fusion centers on board or remotely located, national intelligence centers, etc.) available to the CJTF. These connections allow the participants to collaborate in arriving at a decision. Collaboration in this case may improve the quality (accuracy, timeliness, and completeness) of the decision and is therefore, an attribute of the command, control and communications process that needs to be factored into the overall metric. On the other hand, there is always a possibility that too much information is made available to the Task Force nodes resulting in what is generally referred to as "information overload". This is the complexity effect and it has the opposite effect of collaboration.

2. A Probability Model of Knowledge

The uncertainties addressed in this thesis in the TCT problem center on the time required getting ordnance on target. The intermediate times used to collect, process, and disseminate information, all of which are also uncertain, contribute to this time. Because they are uncertain, all are considered to be random variables. The most common distribution assumed for the intermediate times is the exponential[2] distribution. Let's consider the time, t, required to complete one of the tasks in the TCT problem, where t is an exponential random variable with density function:

$$f(t:\lambda) = \lambda e^{-\lambda t} \text{ for } t \geq 0 \tag{2}$$

The expected time required to complete the task is $1/\lambda$. The uncertainty in this and the other times comprising the overall TCT problem can be taken to reflect a lack of knowledge. Knowing exactly how long each task takes facilitates planning and execution, a lack of knowledge can result in poor planning and possibly, mission failure.

3. Information Entropy

To assess the degree of knowledge present in the density functions used in the TCT problem, we employ the concept of information or Shannon entropy. Information entropy is a measure of the average amount of information in a probability distribution and is defined as:

[2]The only other distribution assumed for the intermediate times is the gamma distribution, for the initial SSN report delay. Only the exponential distribution is discussed in this section. The same formulas apply to the gamma case, with details provided in the simulation development chapter.

$$H(t) = -\mathrm{E}[\ln(f(t))] = -\int_{t=-\infty}^{\infty} \ln[f(t)] f(t) dt \tag{3}$$

Information entropy is the negative expected value of the logarithm of the probability density function. Information entropy is based on the notion that the amount of information in the occurrence of an event is inversely proportional to the likelihood that the event will occur.

Applying the formula to the exponential distribution, we get:

$$\begin{aligned} H(t) &= -\mathrm{E}\left[\ln \lambda e^{-\lambda t}\right] = -\mathrm{E}[\ln \lambda - \lambda t] = -\{\ln \lambda - \lambda \mathrm{E}(t)\} = -\left\{\ln \lambda - \lambda\left(\frac{1}{\lambda}\right)\right\} \\ &= 1 - \ln \lambda = \ln\left(\frac{e}{\lambda}\right) \end{aligned} \tag{4}$$

Note that entropy varies with the variance of the distribution, as should be expected. As the variance $1/\lambda^2$ increases, $H(t)$ also increases. Note that entropy is unbounded for this distribution.[3]

RAND uses the entropy function to develop a measure of knowledge by first assessing the "certainty" in the density function. This requires an approximate upper bound be assigned to $H(t)$, the equivalent to assigning a maximum expected time to complete a given task. This should not be too difficult to do for most tasks associated with the TCT problem. If we let $\lambda_{\min}$ represents the minimum rate that corresponds to the maximum expected time, then a measure of certainty or knowledge can be written as:

$$K(t) = \ln\left(\frac{e}{\lambda_{\min}}\right) - \ln\left(\frac{e}{\lambda}\right) = \ln\left(\frac{\lambda}{\lambda_{\min}}\right) \tag{5}$$

Note that this quantity is dimensionless and therefore, can be used directly to influence combat measures of effectiveness. It is desirable however, for the measure of knowledge to be normalized. This can be accomplished by noting that when $\lambda = \lambda_{\min}$, $K(t) = \ln(1) = 0$ and when $\lambda / \lambda_{\min} = e, K(t) = \ln(e) = 1$. Using this logic, RAND uses the following definition for knowledge:

[3] This is true for all continuous distributions.

$$K(t)=\begin{cases}0 & \text{if } \lambda<\lambda_{\min}\\ \ln(\lambda/\lambda_{\min}) & \text{if } \lambda_{\min}\le\lambda<e\lambda_{\min}\\ 1 & \text{if } \lambda\ge e\lambda_{\min}\end{cases} \tag{6}$$

One problem with this formulation is the condition for "perfect" knowledge. This occurs when $K(t)=1$, or when the expected time to complete a task, $1/\lambda$, is approximately one-third the maximum expected time to complete the task. Figure 7 illustrates the knowledge function for $\lambda_{\min}=0.5$ completions per hour or a maximum time of 2 hours to complete a task.[4]

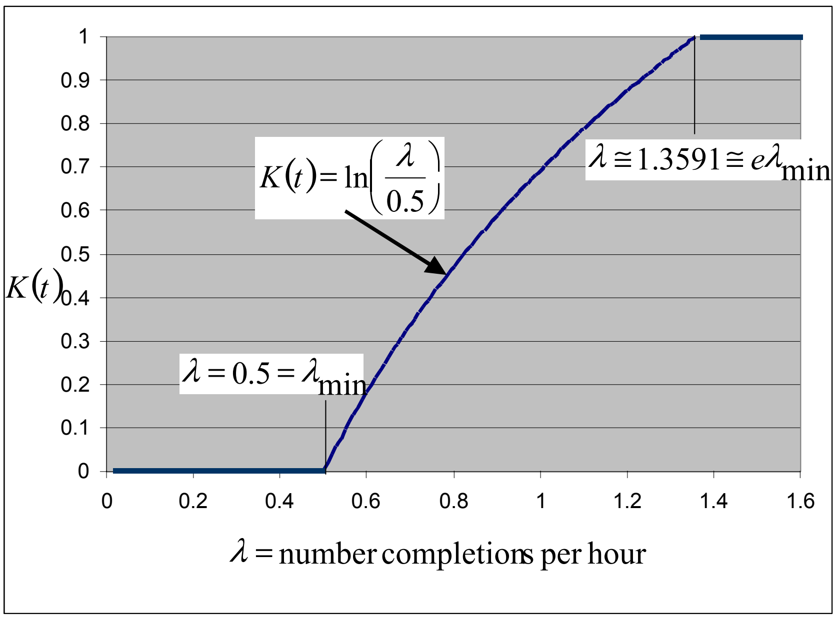

Figure 7. Knowledge Function for Exponential Distribution. $\lambda_{\min}$ represents the minimum rate that corresponds to the maximum expected time to complete a task.

It may be desirable in some cases to employ more stringent conditions on "perfect" knowledge. This can be done by casting the probability distribution in terms of $M>e$:

$$K(t)=\begin{cases}0 & \text{if } \lambda<\lambda_{\min}\\ \dfrac{\ln\lambda-\ln\lambda_{\min}}{\ln M}=\dfrac{\ln(\lambda/\lambda_{\min})}{\ln M} & \text{if } \lambda_{\min}\le\lambda<M\lambda_{\min}\\ 1 & \text{if } \lambda\ge M\lambda_{\min}\end{cases} \tag{7}$$

[4] For additional information on the use of information entropy as a measure of knowledge, see W. Perry and J. Moffat, *"Measuring the Effects of Knowledge in Military Campaigns"*, in *"The Journal of the Operational Research Society"*, (1997) 48, No. 10, pp 965-972.

4. Latencies

For each of the three cases (Platform-Centric, Network-Centric, and Future Network-Centric) studied, the time required to perform the required tasks is central to computing the latency MOP necessary to evaluate the effectiveness of the TCT operations. Table 1 lists the expected (mean) times/latencies required, as assessed by Navy personnel (see RAND's report, Reference 1), to complete the tasks listed along with a reasonable upper bound (the lower bound is, of course, zero).

Tasks	**Platform-Centric**		**Network-Centric**		**Future Network-Centric**	
	Mean	**Maximum**	**Mean**	**Maximum**	**Mean**	**Maximum**
ISR SSN alert	15	60	15	60	15	60
SubGroup processing	20	45	20	45	20	45
CV reads, processes, alerts flight operations	10	20	5	10	-	-
CV directs aircraft	2	5	-	-	-	-
Select launch platform	-	-	-	-	2	5
Aircraft preparation and launch	5	10	5	10	-	-
UCAV launch	-	-	-	-	5	10
UCAV fly out	-	-	-	-	5	10
F/A-18 fly out	15	30	15	30	-	-
SLAM-ER fly out	15	20	15	20	15	20
SSN update	15	60	15	60	-	-

All times in minutes

Table 1. Expected and Maximum Latencies for the Three Networks.

Although not the complete story, the time required to get a weapon on target is an important part of the time-on-target metric. In general, there are $\tau \leq n$ nodes involved in the operation. We will refer to these nodes as the Task Force. Not all nodes need to be combat elements; some may be sensors, information processing facilities, etc. The only criterion is that they be directly involved in the mission. The time required for each to perform its assigned tasks contributes directly to latency. Note that we are not concerned about "how well" they perform their task at this point, just how long it takes. It is also possible that the elements of the Task Force perform their tasks in parallel, sequentially or some combination of both.

For node i, the time, t, required to perform all of its tasks in support of the operation is taken to be an exponential random variable:

$$f(t : \lambda_i) = \lambda_i e^{-\lambda_i t} \tag{8}$$

where:

$\frac{1}{\lambda_i}$ is the mean time to complete all tasks at node i. Assuming that all nodes act sequentially, we then get a total expected latency of,

$$L = \sum_{i=1}^{\tau} \frac{1}{\lambda_i} \tag{9}$$

Other operating concepts are possible. For example, Figure 8 depicts two different concepts, both of which have sequential and parallel processing components. The expected latency for the first concept is:

$$\begin{aligned} L_1 &= \max\left\{ \frac{1}{\lambda_6} + \frac{1}{\lambda_7} + \frac{1}{\lambda_8}, \frac{1}{\lambda_6} + \frac{1}{\lambda_9} + \frac{1}{\lambda_8}, \frac{1}{\lambda_6} + \frac{1}{\lambda_5} + \frac{1}{\lambda_8} \right\} \\ L_2 &= \max\left\{ \frac{1}{\lambda_6} + \frac{1}{\lambda_7} + \frac{1}{\lambda_8} + \frac{1}{\lambda_9}, \frac{1}{\lambda_6} + \frac{1}{\lambda_5} + \frac{1}{\lambda_9} \right\}. \end{aligned} \tag{10}$$

Note that only the path nodes are assessed, not the transit time between the nodes. The reason is that we are assessing the delay at the nodes only: the communication time between nodes is taken to be practically instantaneous.

In either case, the critical path times constitute the expected latency. If we let $\rho \le \tau$ represent the nodes on the critical path, the expected latency then is:

$$L = \sum_{i=1}^{\rho} \frac{1}{\lambda_i} \tag{11}$$

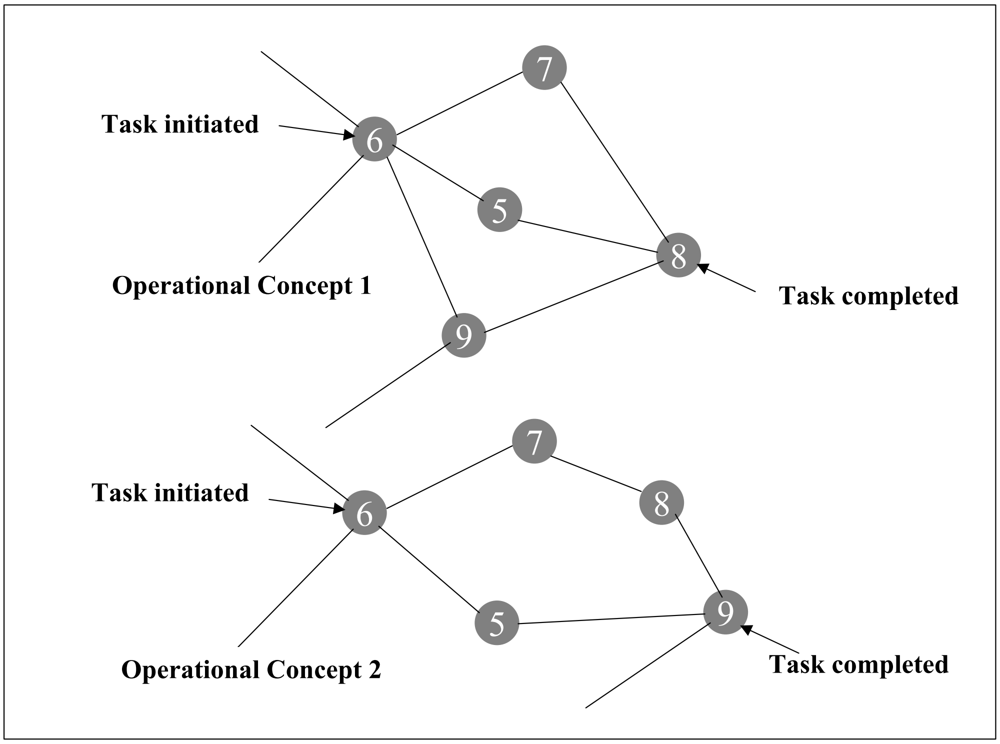

Figure 8. Alternative Operating Concepts. Only the latencies of those nodes on the critical path constitute the expected latency.

5. Quality

In RAND's example, there are several ways the quality of the information regarding the location of the enemy submarine may be influenced by the command, control and communications system. First, the equipment and procedures in place at each of the nodes that contribute to the operation affect the accuracy of the intermediate products produced at that node. For example, the fusion facilities on board the cueing system determine, in part, how well the enemy submarine is tracked. Secondly, the degree to which the Task Force is able to collaborate to inform decisions increases the confidence that a correct (accurate) decision is taken. Thirdly, the ability of the Task Force to access other nodes in the network to complete the operational picture helps ensure nothing is missed. Finally, the amount of training and level of experience of the crews and the length of time they have operated as a team affects the speed with which they are able to accomplish their assigned task—to locate and engage the enemy submarine.

A suitable measure of quality in the TCT problem is therefore, the amount of knowledge available about the expected times required to complete the tasks. The quality of the processes and equipment in place at each node, *i*, in the Task Force is calculated as the knowledge function, and therefore, RAND uses a metric, $0 \le K_i(t) \le 1$. A value of $K_i(t)$ close to 1.0 implies high quality whereas one nearer to zero implies low quality. In addition to the nodes in the Task Force, RAND assumes that the quality of the products produced by other nodes in the network can also be measured in the same way.

6. Collaboration

Collaboration is a process in which a team of individuals work together to achieve a common goal. It is important because collaboration enhances the degree of shared awareness in the group focused on solving a specific problem or arriving at an agreed decision. There are several reasons why collaboration might be expected to improve the degree of shared awareness, including the potential for increased sharing of information and experience, as well as synergy of inference. However, there are other factors that can degrade performance, such as disruptive interactions, misunderstandings or over-valuing a particular point of view due to the persuasiveness or authoritarian role of an individual team member. For this reason, the opportunity to collaborate can both add to and detract from effective combat operations. This section treats the contributions only. The detractions[5] are addressed later

We now assess the contribution of collaboration to the task of locating and engaging the enemy submarine. But first, we need the definition of the degree of a node (or vertex) from graph theory:

Degree: The degree of a node or vertex in an undirected graph is the number of edges emanating from it, with loops counted twice.[6]

The network graphs in Figure 6 and Figure 8 are undirected graphs in that the connection is two-way. Note that node 6 in Figure 6, for example, has degree 5.

[5] For a fuller discussion of collaboration and shared awareness, see W. Perry, D. Signori and J. Boon, *"Exploring Information Superiority: A Methodology for Measuring the Quality of Information and its Impact on Shared Awareness"*, RAND DRR-2389-OSD, 2001.

[6] Taken from B. Jackson and D. Thoro, *"Applied Combinatorics with Problem Solving"*, Addison-Wesley, 1990.

The opportunity for collaboration depends upon the number of Task Force and other nodes each Task Force node is connected to, or the degree of the node. Letting n_i be the degree of node i, then the contribution of collaboration to the quality of node i's operation is expressed by RAND, as the product:

$$\prod_{j=1}^{n_i} \left(1 - K_j(t)\right)^{\omega_j} \tag{12}$$

where:

$$\omega_j = \begin{cases} 0.5 \text{ if node } j \text{ is not in the Task Force} \\ 1.0 \text{ if node } j \text{ is in the Task Force} \end{cases}$$

If the quality of the interaction between nodes i and j is "good", i.e., $K_j(t)$ is close to 1, then $1 - K_j(t)$ will be small—thus reducing the overall product. RAND uses this effect to define the expected latency accounting for collaboration as:

$$L(c) = \sum_{i=1}^{\tau} \prod_{j=1}^{n_i} \left[\left(1 - K_j(t)\right)^{\omega_j}\right] \frac{1}{\lambda_i} \tag{13}$$

The effect of collaboration is to reduce the expected time required to complete the mission and "good" collaboration reduces it further.

7. Complexity

A well-connected network is necessary for effective command and control, but it is not sufficient. For this reason, RAND refers to the network as the potential energy in a command and control system. The sufficient condition that must be added is the command and control process that operates over the network. This is the kinetic energy of the command and control system and to be effective, it must produce quality information that is reflected in good combat outcomes; it is always possible to misuse a well-connected network and to effectively use one that is not well connected.

In a well-connected network there is always the possibility that too much information is made available to the Task Force nodes resulting in what is generally referred to as "information overload." This can have the opposite effect of collaboration. Instead of speeding the time required to complete tasks, it can slow the time as staff and commanders sift through the information for what is required. RAND refers to this effect as complexity, and asserts that every command and control system exhibit this effect to some degree.

Complexity is defined by RAND as a function of the total number of connections to the Task Force nodes, or the total degree of the operation. Therefore, complexity focuses on the potential misuse of the network, whereas collaboration focuses on the effective use of the network. Letting *C* represent operational complexity, then

$$C = \sum_{i=1}^{\rho} n_i \tag{14}$$

For small values of *C*, the complexity effect is negligible and for some range it increases rapidly, leveling off at what might be referred to as the information overload point, i.e., when the information arriving from the multiple connections is so great as to practically shut down operations. This suggests a logistic or S-curve relation between *C* and the complexity factor to be introduced into the expected latency metric or[7]:

$$g(C) = \frac{e^{a+bC}}{1+e^{a+bC}} \tag{15}$$

The parameters *a* and *b* determine both the region of minimal impact and the size of the region of rapidly increasing impact. Figure 9 illustrates a typical complexity function for the zero to 45 possible connections for the network depicted in Figure 6.

[7] This curve is sometimes referred to as the logistics response function or the growth curve. See J. Neter and W. Wasserman, *"Applied Linear Statistical Models"*, R.D. Irwin, 1974.

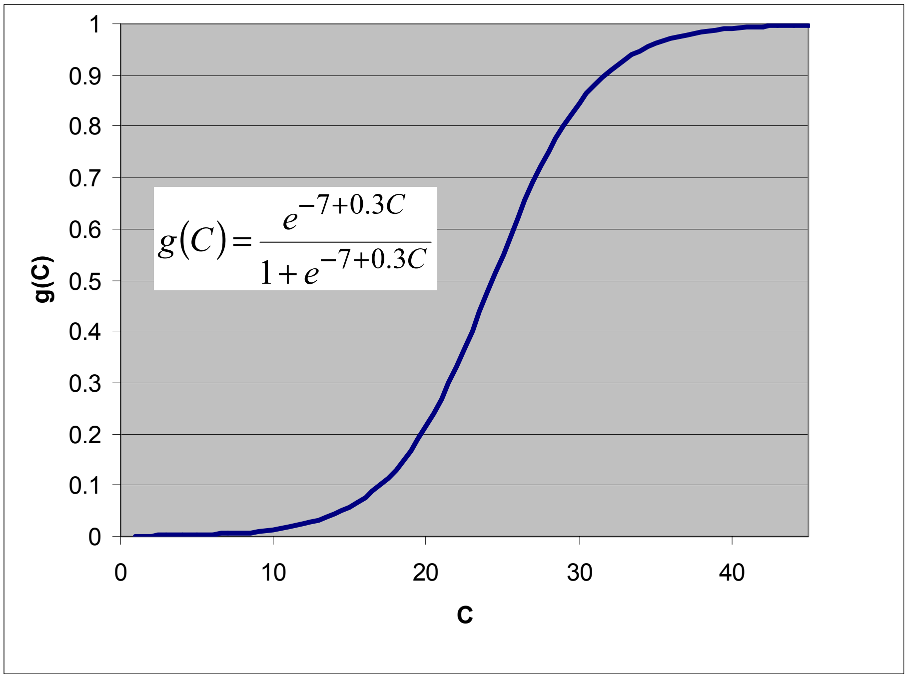

Figure 9. Complexity Factor. The parameters *a* (-7) and *b* (0.3) determine both the region of minimal impact and the size of the region of rapidly increasing impact.

Including complexity in the calculation of the expected latency, yields:

$$L(c,C) = \frac{1}{1-g(C)} \sum_{i=1}^{\tau} \prod_{j=1}^{n_i} \left[(1 - K_j(t))^{\omega_j} \right] \frac{1}{\lambda_i} \tag{16}$$

When the number of connections is low, the complexity effect on latency is minimal. Between approximately 15 and 35 connections, the complexity effect rises sharply, leveling off to nearly paralysis at 45 connections.

Equation (16) reflects the balance between the positive effects of collaboration and the negative effects of complexity. If the effects of complexity are negligible, i.e., there are few connections in the network, and the effects of collaboration are considerable, i.e., the knowledge function for most distributions is high, then it is possible for the expected latency to be much lower than the sum of the critical path latencies. What this means is that the positive effects of collaboration have compensated for the time required to perform all operational tasks. The converse is also true in a richly connected network where the knowledge functions are rather small. That is, the effective latency can exceed the critical path latency. For this reason, $L(c, C)$ is called the "effective expected latency".

8. Detection and Target Acquisition

The measure of TCT effectiveness is the probability that the target can be attacked during the window of opportunity. For the case of the surfaced threat submarine, it is the probability that the aircraft can detect, classify, and place ordnance on the submarine before it submerges. This probability of detection depends upon time on target, the quality (accuracy, timeliness and frequency) of the location and speed estimates of the enemy submarine, and the characteristics of the attack weapon. For the purpose of illustration, it is assumed that the aircraft will attack using a missile with an electro-optical system capable of detecting and classifying the threat submarine on the surface. The aircraft is not expected to detect the submarine directly. Instead, the pilot uses the cockpit display from the missile to detect and classify the target. The pilot then locks the missile onto the target submarine. For simplicity, the aircraft is assumed as searching the KILO area of uncertainty (AOU), with the missile employed as a remote sensor. RAND also assumes a sea-skimming missile with an accordingly short acquisition range, and that once the missile has acquired the submarine it will be killed quickly. In other words, the time of flight over the acquisition range and weapon reliability is not considered.

If S is the time that elapsed between the moment the submarine leaves port and submerges (in hours), then $T = S - L(c, C)$. If $T \leq 0$, the aircraft fails to engage the target. If $T > 0$, the cumulative probability that the aircraft detects and acquires[8] the target depends upon the length of time it has to search the AOU.

[8] For purposes of this analysis, we are concerned with both detection and acquisition. However, for ease of exposition, we refer to both as simply "detection".

Letting s denote the sweep width in nautical miles, v denote missile speed in knots, and A the AOU in square nautical miles, the probability of detection $P_d(T)$[9] as a function of search time T is:

$$P_d(T) = 1 - e^{-\gamma T} \tag{17}$$

where:

$$\gamma = \frac{sv}{A}$$

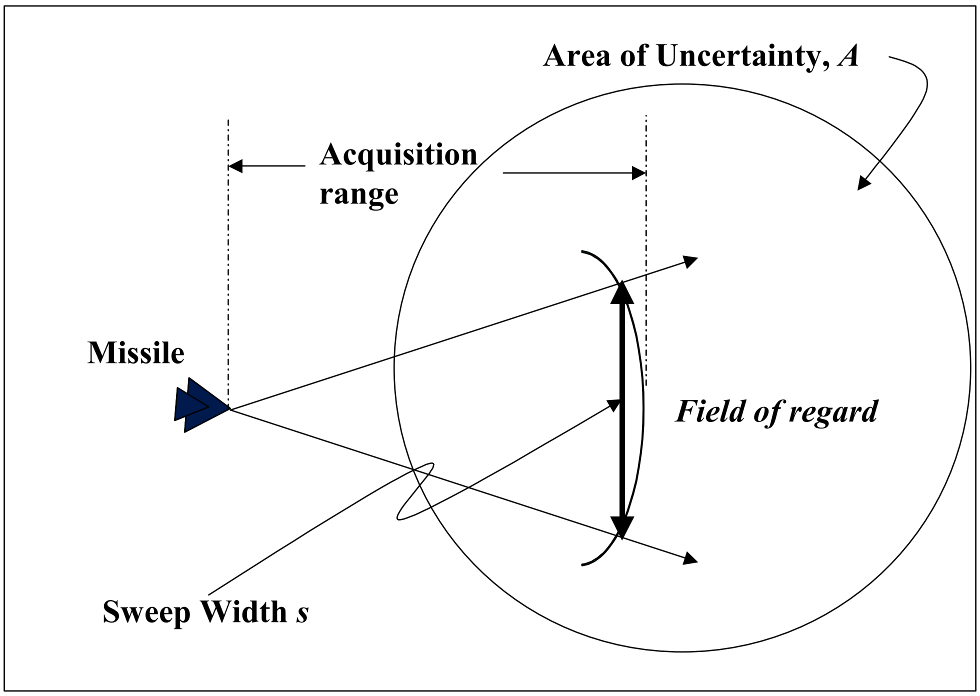

Figure 10. Search Operations. The actual shape of the area of uncertainty (AOU) depends upon what the friendly force knows about the enemy submarine's mission.

As depicted in Figure 10, A is taken to be the area of a circular region. However, the actual shape of the region depends upon what the friendly force knows about the enemy submarine's mission. The effect of knowledge is to reduce the size of the AOU by restricting the search to a fraction of the circle coincident with the direction of the submarine, which has the same effect as reducing the radius of search.

9 See B. Koopman, *"Search and Screening: General Principles with Historical Applications"*, Pergamon Press, Inc., 1980.

The radius of the AOU depends upon the elapsed time, t_u, since the last update and upon the speed of the surfaced submarine; $A = \pi\left(\frac{r}{k}\right)^2 = \pi\left(\frac{wt_u}{k}\right)^2$, where $0 < \frac{1}{k^2} \leq 1$ is the fraction of the circle that must be searched based on the prior knowledge of the submarine's route of advance. For simplicity, AOU growth is not considered during the search. Similarly, the possibility of updating target data during the search is not addressed. Now, the cumulative detection probability function becomes:

$$P_d(T) = 1 - e^{-\frac{svk^2}{\pi(wt_u)^2}T} \tag{18}$$

Although the friendly commander has no control over target speed *w*, improved equipment and procedures can greatly affect *s*, *v*, *T*, and intelligence information can affect *k*.

Figure 11 illustrates the increase in detection probability for two cases: (i) when the AOU is 20 square nautical miles and (ii) when the AOU is only 1 square nautical mile. In both cases, the speed of the missile is 450 knots and the sweep width is 0.25 nautical miles. If we assume that the speed of the target submarine is constant (or in any case not under the friendly commander's control), and then the radius of the AOU is dependent on solely the time elapsed since the last update on the target submarine's location. Note the dramatic difference in the results. For the 1 square nautical mile case, detection probability "approaches one" within two or three minutes of searching whereas the detection probability for the 20 square nautical mile case has still not peaked after 30 minutes of searching.

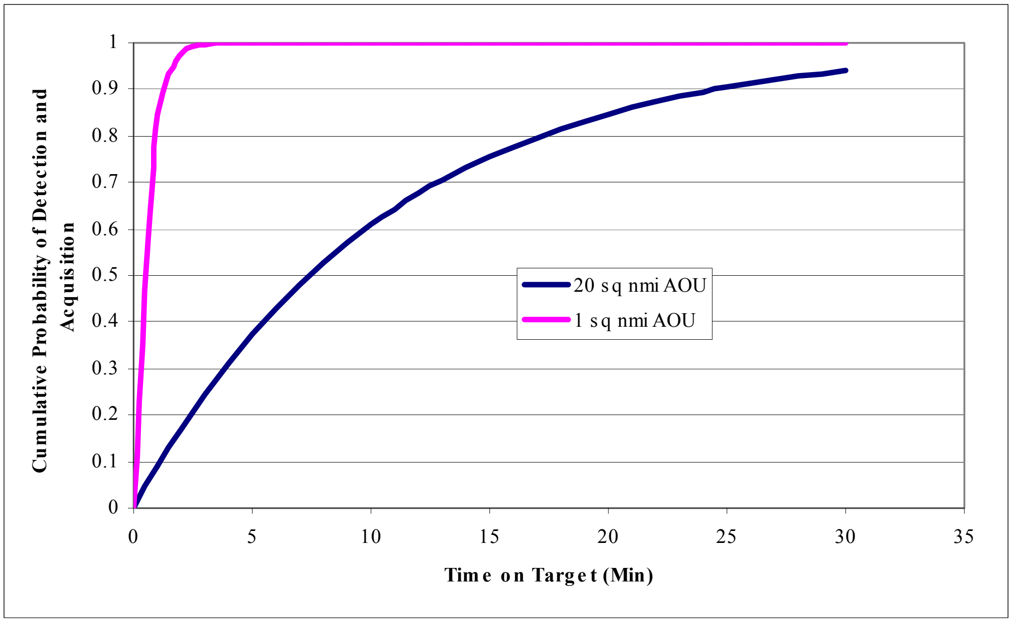

Figure 11. AOU Effects on Detection Probability. Note the dramatic difference in the results. For the 1 square nautical mile case, detection probability "approaches one" within two or three minutes of searching, whereas the detection probability for the 20 square nautical mile case has still not peaked after 30 minutes of searching.

The probability, $P_d(T)$, is the probability that the target will be detected by time T. This is the cumulative probability distribution for the probability density function:

$$f_d(T) = \gamma e^{-\gamma T} \tag{19}$$

This function has a mean $\frac{1}{\gamma} = \frac{\pi(wt_u)^2}{svk^2}$. This is the expected time required to detect the target. As with the times required to collect, process, and disseminate information, a maximum expected time can be determined and therefore, the knowledge resident in the detection time density $f_d(T)$ is assessed by RAND to be:

$$K(T) = \begin{cases} 0 & \text{if } \gamma < \gamma_{\min} \\ \ln(\gamma/\gamma_{\min}) & \text{if } \gamma_{\min} \le \gamma < e\gamma_{\min} \\ 1 & \text{if } \gamma \ge e\gamma_{\min} \end{cases} \tag{20}$$

This can be used to reflect the quality of the target location estimate, and it will influence the probability of detection.

In general, if *K(T)* is large, i.e., the uncertainty of the search time is small, we would expect a search more effectively matched to the time available, which has the effect of reducing the search area. The effective search area E_A is:

$$E_A = [1 - K(T)]\pi\left(\frac{wt_u}{k}\right)^2 \tag{21}$$

Applying this to the detection probability equation, the adjusted detection probability is:

$$P_d^*(T) = 1 - e^{-\frac{svk^2}{[1-K(T)]\pi(wt_u)^2}T} \tag{22}$$

If we let $P^*_{K|T}$ be the knowledge-enhanced probability of kill, then in the case where detection is equivalent to a kill with probability one, $P^*_{K|T} = P^*_d(T)$.

III. SIMULATION DEVELOPMENT AND BENCHMARKING

The first section of this chapter documents the developmental steps to implement the stochastic simulation model. In the second section, the conclusions from the benchmarking exercise of the stochastic simulation model against the existing deterministic model are discussed.

A. SIMULATION DEVELOPMENT

The RAND EDA tool, which was a purely deterministic model, is extended to include stochastic simulation capabilities, with the TCT vignette used as a case study. The stochastic simulation tackles three issues that could not be addressed using a deterministic model: real-world outcomes, variability, and extreme values analysis.

The main developmental steps in implementing the simulation model are:

a. Determine the appropriate distributions to represent the various latencies and the search and detection variables, e.g., sweep width of the SLAM-ER missile depends on factors like the weather conditions. Thus, sweep width has a certain minimum and maximum value, and a value for a "typical weather" day. The beta distribution with parameters minimum, maximum and mode are used to fit the sweep width variable.

b. Design and develop a data entry form to elicit parameters of the various latencies and search and detection distributions. Data validation checks are incorporated in the data entry form to make it user-friendly, i.e., the simulation model automatically checks that the data that the user has entered are logical, e.g., minimum ≤ average.

c. Implement a process for utilizing the stochastic simulation to analyze the TCT vignette. Adopting the framework of measures and metrics created by RAND, compute the effective time remaining (MOP) and kill probability (MOE) for each simulation replication. The simulation is repeated for a user-specified number of times, and the user-specified confidence intervals of the MOP and MOE are calculated from the simulation results, and the MOP and MOE histograms are drawn.

The details of the simulation development are documented in Appendix A.

B. BENCHMARKING AGAINST DETERMINISTIC MODEL

In this section, the stochastic simulation model developed is benchmarked against the deterministic model. Six pairs (stochastic vs. deterministic) of results are compared to provide some assurance that the stochastic model produces logical and consistent results with the deterministic model:

a. Pair 1: Network centricity is set to Future Network-Centric. All inputs are deterministically set to their average values.

b. Pair 2: Network centricity is set to Network-Centric. All inputs are deterministically set to their average values.

c. Pair 3: Network centricity is set to Platform-Centric. All inputs are deterministically set to their average values. The first three pairs (second pair uses the same inputs as the first pair except the network centricity is changed to Network-Centric, and the third pair is for Platform-Centric) of results are based on the same inputs so that the performance of each network centricity can be gauged.

d. Pair 4-Pair 6: Network centricity and inputs are set randomly, in an effort to add credibility to the benchmarking exercise.

1. Pair 1 Comparison (FCW)

The deterministic inputs for the first pair of results:

a. Future Network-Centric (same as the Futuristic Network in Figure 10).

b. All input parameters to the deterministic model are set at their average values, i.e., the mid slider bar positions (see Figure 12), except for submerge time and UCAV. These two inputs are set at values that ensure non-zero outputs (to make sure useful insights can be gained from the comparisons).

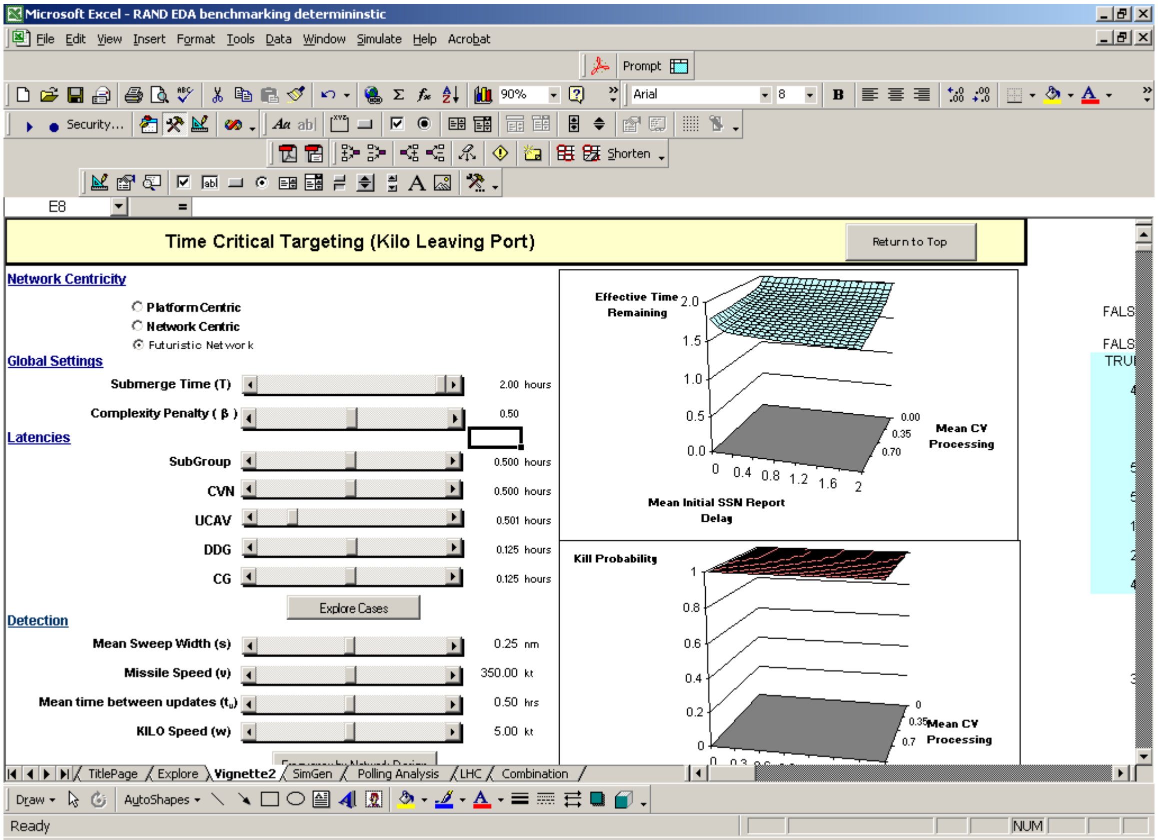

Figure 12. RAND EDA Tool for TCT Vignette. The left portion of the screen shows the input variables, and the right portion shows the effective time remaining and kill probability output surfaces.

Note that the output surfaces for the effective time remaining and kill probability have 441 (21×21) outputs. All 441 results have their network centricity set to Future Network-Centric, the submerge time set to 2 hours, etc. What differentiate them are the values of the initial SSN report delay and the mean CV processing delay. The initial SSN report delay is varied from zero to two hours in steps of 0.1 hour (21 values), and the mean CV processing delay is varied from zero to one hour in steps of 0.05 hour (21 values).

The only result from the 441 cases that are used in the deterministic/stochastic comparison is that with initial SSN report delay of one hour (midpoint of zero and two hours), and the mean CV processing delay of 0.5 hours (midpoint of zero and one hour).

The required result shown below is extracted from the data used to construct the effective time remaining and Pk output surfaces:

Effective time remaining = 1.72 hours

Pk = 1.00

Input Variables	**Deterministic**	**Stochastic Distribution**	**Stochastic Parameter 1**	**Stochastic Parameter 2**	**Stochastic Parameter 3**
Network Centricity	Futuristic Network	NA	Futuristic Network	NA	NA
Number of Runs	NA	NA	1000	NA	NA
Submerge Time	2 hrs	Beta	1.999 hrs (min)	2.001 hrs (max)	2 hrs (mode)
Complexity Penalty	0.5	Constant	0.5 (constant)	NA	NA
Initial SSN	1 hr	Gamma	0 min (min)	60 mins (mean)	NA
CV	0.5 hr	Exponential	30 mins (mean)	NA	NA
SubGroup	0.5 hr	Exponential	30 mins (mean)	NA	NA
CVN	0.5 hr	Exponential	30 mins (mean)	NA	NA
UCAV	0.5 hr	Exponential	30 mins (mean)	NA	NA
DDG	0.125 hr	Exponential	7.5 mins (mean)	NA	NA
CG	0.125 hr	Exponential	7.5 mins (mean)	NA	NA
Sweep Width	0.25 nm	Beta	0 nm (min)	0.5 nm (max)	0.25 nm (mode)
Missile Speed	350 kts	Beta	200 kts (min)	500 kts (max)	350 kts (mode)
Time b/w Updates	0.5 hr	Exponential	0.5 hrs (mean)	NA	NA
KILO Speed	5 kts	Beta	0 kt (min)	10 kts (max)	5 kts (mode)

Table 2. Inputs Setup for Pair 1 (FCW). Network centricity set to Future Network-Centric, all input variables are set to their average values, except for submerge time and UCAV/Strike latency.

The parameters of the stochastic input variables are chosen such that their distributions' means agree with the deterministic values. See Table 2 for the inputs setup of Pair 1. Note that because the data entry form for the stochastic model is designed to facilitate ease of use by the analyst, some input variables have different units, e.g., the CVN latency is stated in hours for the deterministic model but minutes for the stochastic model. However, the point to note in the comparison is that the inputs are set to the same values (0.5 hours = 30 minutes).

See Figure 13 and Figure 14 for the outputs[10] from the stochastic model. Note that probability (y-axis label) for both histograms refer to the proportion of the 1000 (in this case) replications with those values on the x-axis. The number of replications for the stochastic simulation is fixed at 1000 for all the runs in this thesis, and that produces stochastic means estimates with halfwidths of less than 1.5 minutes for the effective time remaining, and 2.5 percent for Pk, in all the results stated in this report.

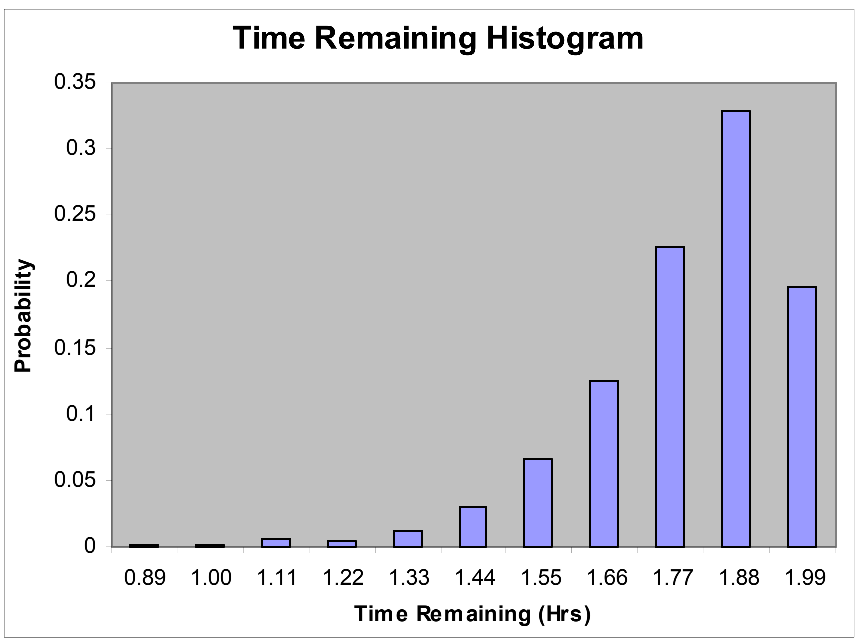

Figure 13. Stochastic Effective Time Remaining (MOP) for Pair 1 (FCW). The mean stochastic effective time remaining is 1.75 hours, as opposed to the 1.72 hours from the deterministic model. Note the spread of the effective time remaining that is not evident from the single value of 1.72 hours obtained from the deterministic model.

[10] All the histograms in this report should be interpreted with the general rule, the smallest value on the x-axis shows the minimum value from the simulation run, and the largest value shows the maximum value. The rightmost histogram bin is for data that lies between the second rightmost to the rightmost value.

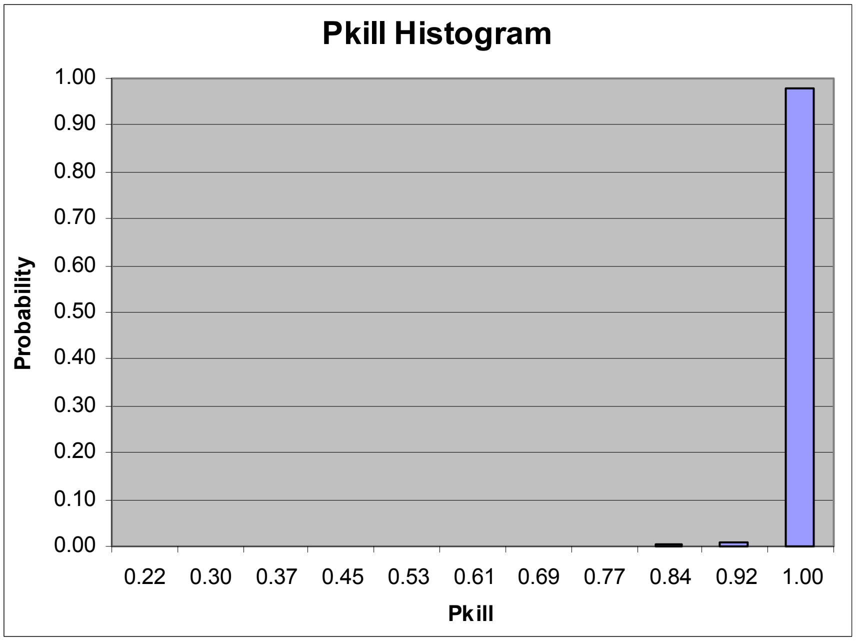

Figure 14. Stochastic Kill Probability (MOE) for Pair 1 (FCW). Probability on the y-axis refers to the proportion of the 1000 replications with kill probability (Pk) shown on the x-axis. Over 950 replications have Pks between 0.92 and 1.00.

The means of the effective time remaining and kill probability are 1.75 hours and 0.99 respectively. Testing the null hypothesis:

H_0: The mean of the stochastic outputs is equal to the deterministic output

For effective time remaining:

$$t_{1000} = \frac{\overline{X}(1000) - 1.72}{S/\sqrt{1000}} = \frac{1.75 - 1.72}{0.16/\sqrt{1000}} = 5.8$$

where:

$\overline{X}$ = mean of the stochastic outputs

S = standard deviation of the stochastic outputs

For Pk:

$$t_{1000} = \frac{\overline{X}(1000) - 1}{S/\sqrt{1000}} = \frac{0.99 - 1}{0.052/\sqrt{1000}} = -6.1$$

Both hypothesis tests have p-values << 0.01, which means we reject H_0 at $\alpha = 0.01$. Although the mean of the stochastic outputs is not statistically equal to the deterministic output (according to the hypothesis tests), the stochastic results can still be considered to be consistent to the deterministic results, based on the minimal absolute deviation between the deterministic and stochastic results.

Out of the 1000 replications, there are 22 cases where Pk < 0.9 (0.9 is an arbitrary choice). The lowest Pk is 0.22, however, it is not visible in the histogram (Figure 14) due to the scale of the y-axis. A clear pattern from these 22 cases is, low sweep width and high time between updates from the ISR submarine. For the deterministic case, the Pk is guaranteed to be at 100 percent, as the effective time remaining for the search and detection effort is high at 1.72 hours, and with the search and detection parameters at their expected values, Pk is 100 percent. The element of variance is missing from the deterministic case, which provides as much information as the means. Note that in this case, we have little difficulty in sinking the KILO. In a more difficult situation, the variance could cause a divergence between the stochastic simulation Pk and the deterministic one.

2. Pair 2 Comparison (NCW)

The Pair 2 comparison is exactly the same as Pair 1, except that the network centricity is changed to Network-Centric. The deterministic result:

Effective time remaining = 1.10 hours

Pk = 1.00

The stochastic outputs are in Figure 15 and Figure 16, with their means of 1.17 hours and Pk of 0.94 respectively. Hypothesis tests similar to the one conducted for Pair 1 have been conducted for Pair 2, as well as the remaining four pairs of deterministic/stochastic comparisons, and their *t*-statistics are at least 4.0, and p-values much smaller than 0.01, implying that the deterministic and stochastic means are not statistically equal. The detailed computations of the *t*-statistics for the hypothesis tests are left out from the report, as no additional insights can be gained from them.

Similar to the conclusions on Pair 1, the Pair 2 stochastic results are consistent with the deterministic results, based on the minimal absolute deviation between the deterministic and stochastic results. However, there are 90 cases where $Pk < 0.9$. The general pattern from these 90 cases is the low time remaining, only averaging 0.53 hours (as opposed to the 1.17 hours average for the entire 1000 cases).

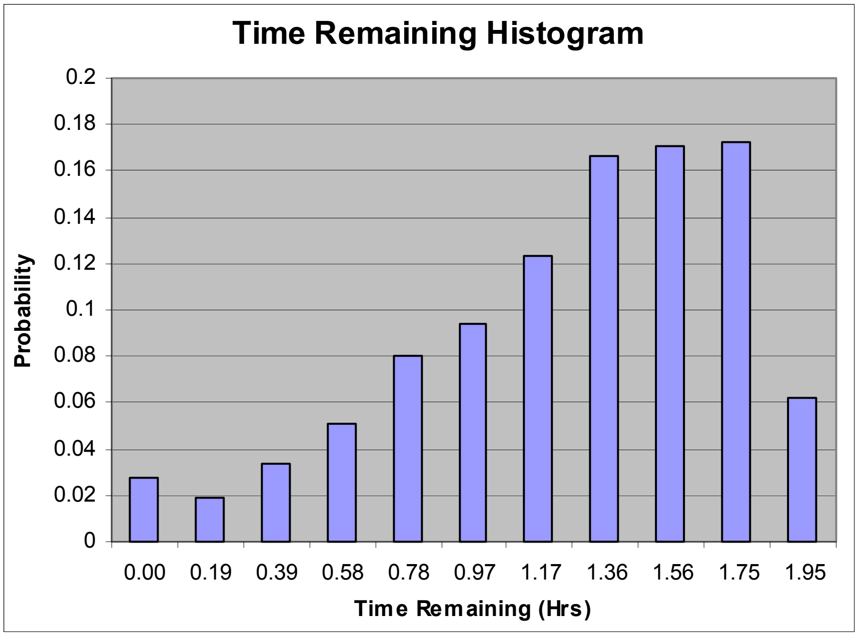

Figure 15. Stochastic Effective Time Remaining (MOP) for Pair 2 (NCW). Unlike the FCW case in Figure 13, there are close to three percent with effective time remaining of zero hour, i.e., no chance of mission success.

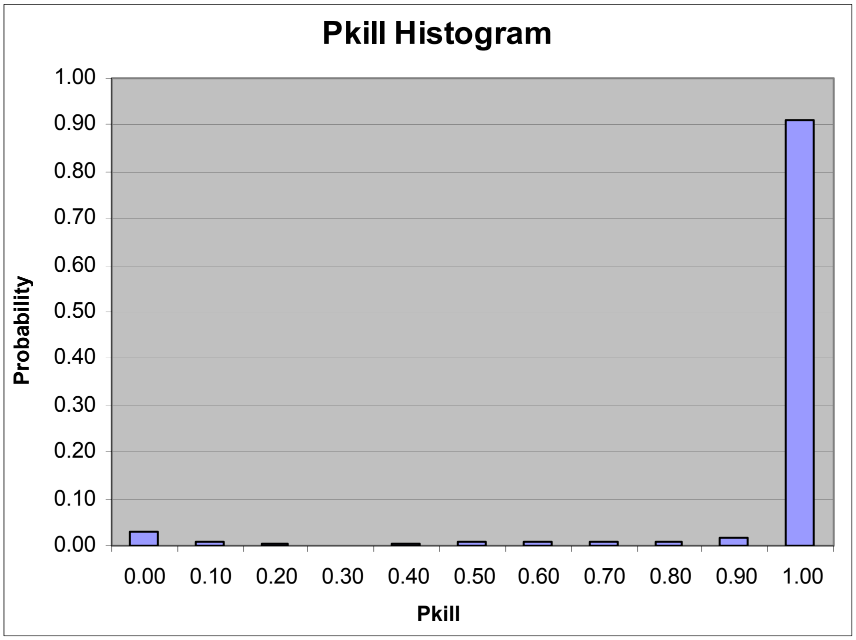

Figure 16. Stochastic Kill Probability (MOE) for Pair 2 (NCW). The spread in Pk is nothing like the spread for the equivalent effective time remaining (Figure 15). This is due to the greatly nonlinear transfer function of the search and detection mission.

3. Pair 3 Comparison (PCW)

Pair 3 comparison is exactly the same as Pair 1, except that the network centricity is changed to Platform-Centric. The deterministic result:

Effective time remaining = 0.50 hour

Pk = 1.00

The stochastic outputs are in Figure 17 and Figure 18, with their means of 0.71 hour and Pk of 0.68 respectively. Note that the deterministic model performs poorly, i.e., the deterministic means deviates significantly from the stochastic means. The stochastic simulation model produces 223 cases (out of the 1000 replications) with zero Pk. This is vastly inconsistent with the 100 percent Pk derived in the deterministic model. The 223 cases have zero Pk because there is no effective time remaining to conduct the search and detection mission. The latencies (messaging and processing delays) in these 223 cases add up to more time than it takes for the enemy KILO submarine to submerge. This, of course, never happens in a deterministic model.

Having said that, it should be noted that deterministic models could produce results close to stochastic simulation models. This occurs when the results are clear, e.g. in another combat context, two opposing sides (blue-to-red) with 100-to-1 ratio, and similar combat effectiveness, will produce similar results from both deterministic and stochastic simulation model, 100 percent win for the blue force. However, when it becomes a 1.1-to-1 ratio, the deterministic model will still predict a 100 percent win for the blue force, while the stochastic simulation model will likely produce the more realistic result that blue force may not always win.

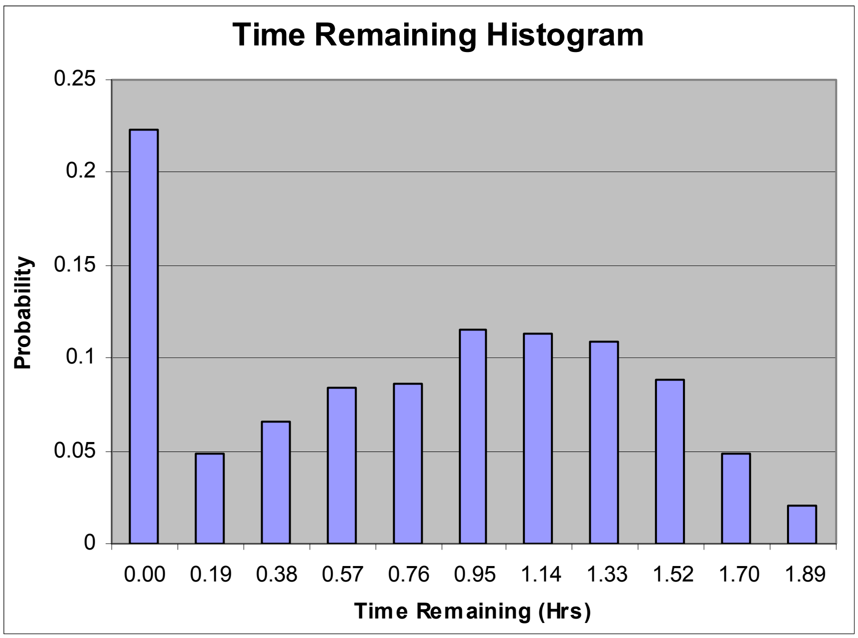

Figure 17. Stochastic Effective Time Remaining (MOP) for Pair 3 (PCW). The "spike" at zero hour is an accumulation of zero as well as negative time remaining (total latencies > submerge time, therefore time remaining = submerge time – total latencies = negative value).

An abnormality observed from Figure 17 is the "spike" at zero hour. This is due to the fact that zero hour is an accumulation of zero as well as negative time remaining (total latencies > submerge time, therefore, time remaining = submerge time – total latencies = negative value). Total latencies is the sum of several individual latencies, and as long as one of the individual latencies gets a big number (which happens not so infrequently) in the stochastic replication, the time remaining will become negative, or practically no time remaining for the search and detection mission.

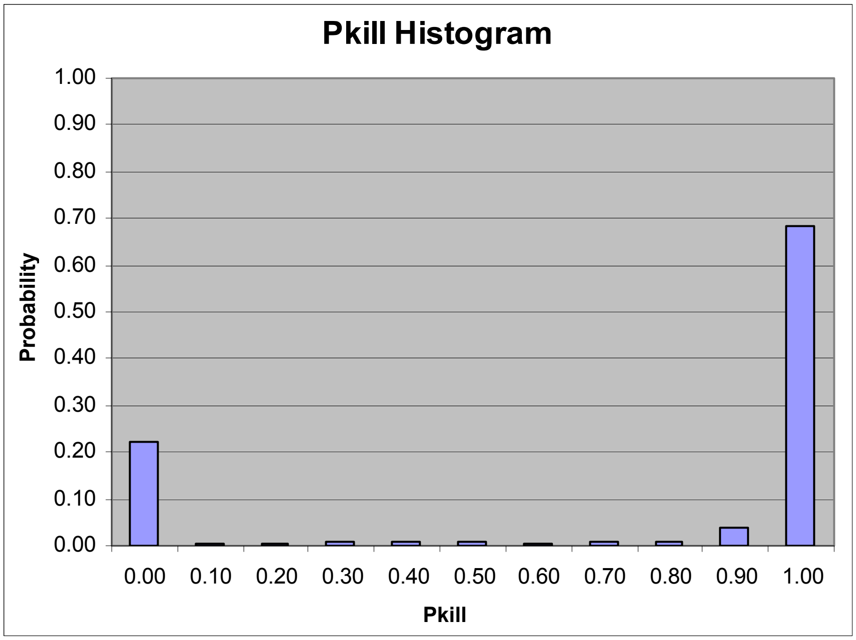

Figure 18. Stochastic Kill Probability (MOE) for Pair 3 (PCW). Note that not many cases fall within the middle bins. This is due to the greatly nonlinear transfer function of the search and detection mission. For any set of search and detection parameters, Pk rises rapidly from zero to close to one within a small range of effective time remaining (zero hour to some "threshold" value). As long as the effective time remaining for the search and detection mission exceeds the "threshold" value, Pk is "pushed" towards one.

Insights gained from the first three pairs of deterministic/stochastic results:

a. FCW performs best under the specified inputs setup. This conclusion is generalized to the entire input domain space in the next chapter.

b. The difference in the deterministic and stochastic means should be expected, as the framework of metrics and measures recommended by RAND is nonlinear, i.e., the transfer function (which is determined by the framework) that takes the inputs and generates the output is nonlinear, i.e., mathematically:

$$f(\overline{x_1}, \overline{x_2}, \ldots) \neq \overline{f(x_1, x_2, \ldots)}$$

where:

f is the transfer function, the underlying framework of metrics and measures on which the deterministic and stochastic models are developed

x_1, x_2, … are the input variables, such as submerge time and missile speed

What is implied in the mathematical form is, the deterministic model (to the left of ≠) that takes in the expected values of the input variables need not produce the same result as the mean of the stochastic outputs, which use the input distributions, unless f is linear. Of course, our simulation has many nonlinear components.

c. A pattern that is apparent from the Pk histograms is, there is always a big proportion of data with Pk = 1, and the other data are divided without any obvious pattern amongst the other Pks. The reason for that lies in the greatly nonlinear transfer function of the search and detection mission. For any set of search and detection parameters, Pk rises rapidly from zero to close to one within a small range of effective time remaining (zero hour to some "threshold" value). As long as the effective time remaining for the search and detection mission exceeds the "threshold" value (different "thresholds" for different search and detection parameters), Pk is "pushed" towards one. When there's no effective time remaining for the search and detection mission, obviously Pk = 0, and for effective time remaining between zero hour and the "threshold", Pk is distributed from zero to one.

4. Pair 4 Comparison (Random Inputs Set 1)

To add credibility to the benchmarking exercise, the next three pairs of results are based on random inputs. To elaborate what is meant by random inputs, see Table 3 column "Excel Implementation". An Excel spreadsheet is developed with those formulas in Table 3 column "Excel Implementation", and run three separate times[11] to generate the random input sets shown in Table 3 column "Random Set 1", "Random Set 2", and "Random Set 3". Note that the random numbers in Table 3 have been rounded to the appropriate decimal places according to the deterministic model input requirements. Also, the units for the input variables in Table 3 are consistent with the units for the

[11] Note that each time the spreadsheet is run (press F9 key); the "RAND()" function in Excel will generate a uniform random number between zero and one.

deterministic model, e.g., the units for the CVN processing latency is hours (instead of minutes).

Input Variables	Excel Implementation	Random Set 1	Random Set 2	Random Set 3
Network Centricity	=if(A1<0.333, “PCW”, if(A1<0.666, “NCW”, “FCW”) where Cell A1: =RAND()	PCW	NCW	FCW
Submerge Time	=2*RAND()	0.39	1.52	1.06
Complexity Penalty	=RAND()	0.14	0.69	0.71
Initial SSN	=2*RAND()	0.40	2.00	0.60
CV	=RAND()	0.60	0.85	0.40
SubGroup	=RAND()	0.83	0.20	0.63
CVN	=RAND()	0.27	0.64	0.71
Strike/UCAV	=3*RAND()	0.25	1.31	0.98
DDG	=0.25*RAND()	N/A	N/A	0.09
CG	=0.25*RAND()	N/A	N/A	0.22
Sweep Width	=0.5*RAND()	0.35	0.44	0.22
Missile Speed	=200+300*RAND()	333	296	485
Time b/w Updates	=RAND()	0.82	0.29	0.25
KILO Speed	=10*RAND()	4.4	3.3	8.7

Table 3. Random Inputs for Benchmarking. The second column shows the Excel formulas, where the values in the remaining columns are generated randomly.

The deterministic result for Pair 4:

Effective time remaining = 0 hour

Pk = 0

The reason for zero Pk is the quick submerge time of the KILO submarine, which leads to zero time for the search and detection mission.

The stochastic outputs are shown in Figure 19 and Figure 20, with means of 0.02 hour and Pk of 0.09 respectively. There are 62 cases (6.2 percent of the 1000 replications) with Pk > 0.9. An analysis of the inputs (random realizations of the replications rather than the input parameters, which are fixed for all 1000 replications) for these 62 cases show a strong pattern, that all 62 cases have initial SSN report delay that is less than 0.35 hour (mean of the initial SSN report delay is 0.40 hour), and Strike latency that is less than 0.32 hour (mean of the Strike latency is 0.25 hour).

The practical interpretation of this pattern is, if the enemy submarine is expected to submerge within a short time (mean of 0.39 hour in this Pair 4 comparison), all efforts must be put into achieving a low (< 0.35 hour) initial SSN report delay and low (< 0.32 hour) Strike latency to have a good (> 0.9) Pk. This implies the importance of initial SSN report and Strike latency in achieving a high Pk.

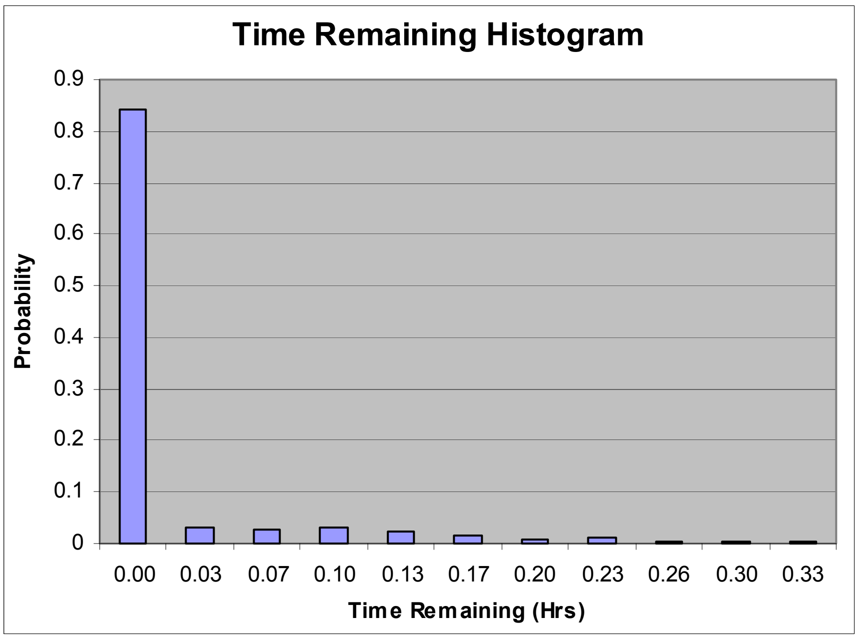

Figure 19. Stochastic Effective Time Remaining (MOP) for Pair 4 (Random Inputs). Due to the relatively quick submerge time of 0.39 hour, most of the replications have zero effective time remaining.

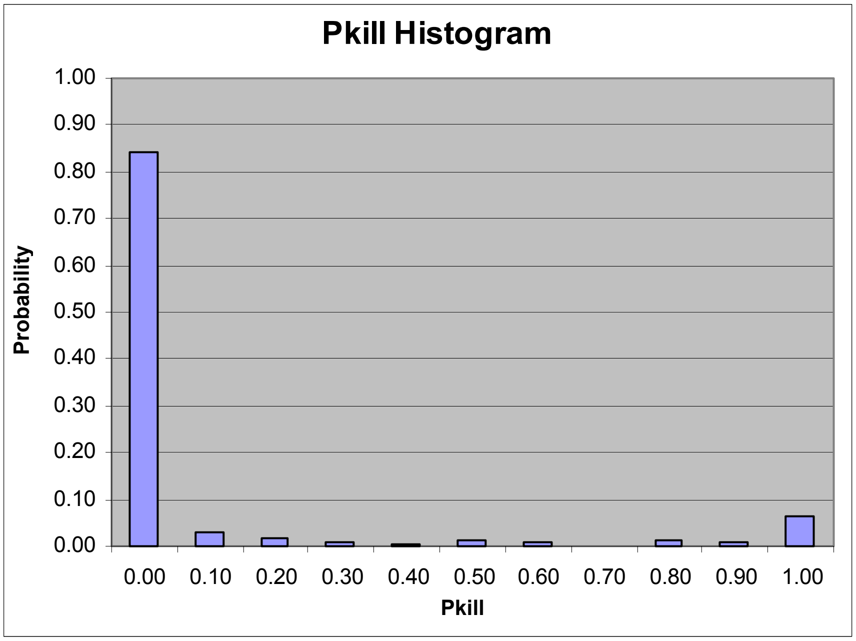

Figure 20. Stochastic Kill Probability (MOE) for Pair 4 (Random Inputs). The same 800+ replications with zero effective time remaining also have zero Pk.

5. Pair 5 Comparison (Random Inputs Set 2)

Based on the input settings in Table 3 for Random Inputs Set 2, the effective time remaining and Pk for the deterministic model are both zero. The stochastic outputs are shown in Figure 21 and Figure 22, with means of 0.12 hour and Pk of 0.23 respectively. No other interesting patterns can be extracted from this pair of results.

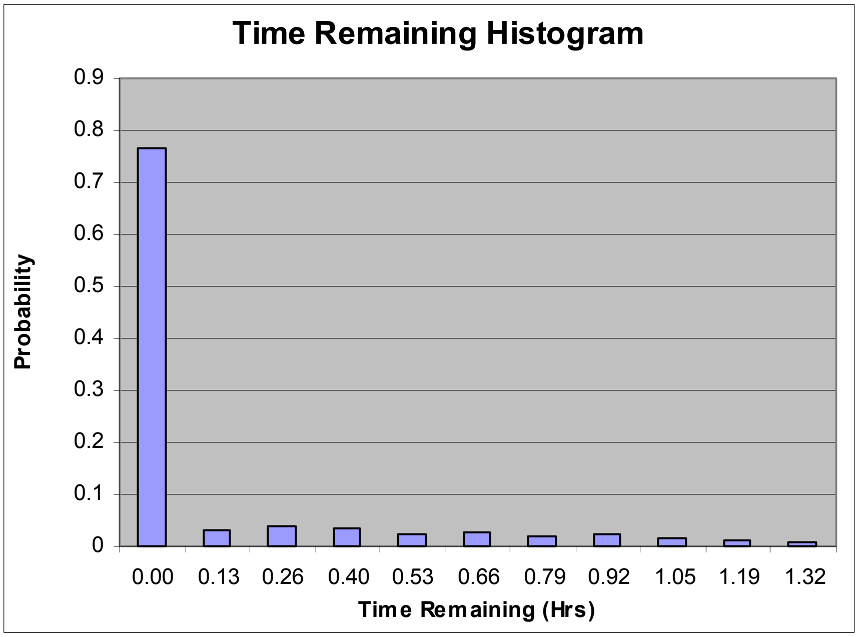

Figure 21. Stochastic Effective Time Remaining (MOP) for Pair 5 (Random Inputs). The mean stochastic effective time remaining is 0.12 hour as compared to zero hour for the deterministic result. Note that some replications even go as high as 1.32 hours.

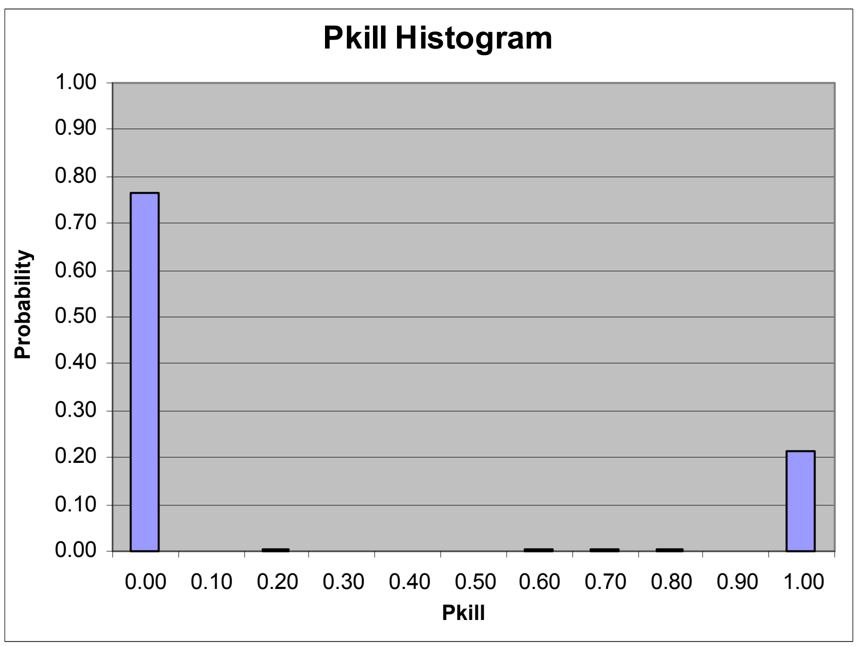

Figure 22. Stochastic Kill Probability (MOE) for Pair 5 (Random Inputs). Note the "spike" at the rightmost bin, which is the accumulation of those replications with effective time remaining greater than some "threshold" value in Figure 21.

6. Pair 6 Comparison (Random Inputs Set 3)

The deterministic result for Pair 6:

Effective time remaining = 0.73 hour

Pk = 1.00

The stochastic outputs are shown in Figure 23 and Figure 24, with means of 0.80 hour and Pk of 0.99 respectively. There are only 21 cases with Pk < 0.9. The obvious pattern from these cases is the relatively high UCAV (note that UCAV latency for FCW is the equivalent of Strike latency for PCW and NCW) latencies, with an average of 2.5 hours (mean of UCAV distribution for Pair 6 is only 0.98 hour). This reinforces the conclusion from Pair 4, i.e., Strike/UCAV latency is a critical factor influencing effective time remaining, and subsequently Pk.

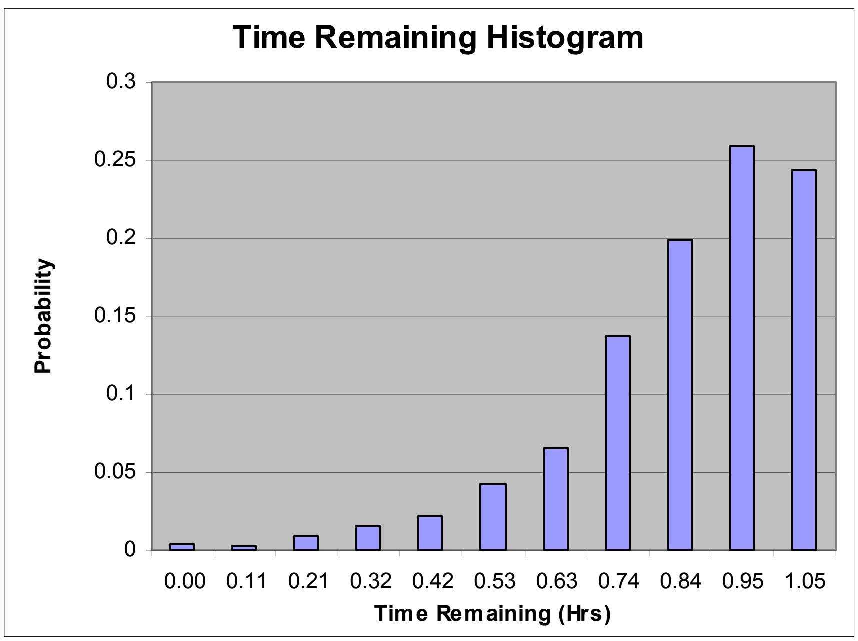

Figure 23. Stochastic Effective Time Remaining (MOP) for Pair 6 (Random Inputs). The mean stochastic effective time remaining is 0.73 hour, as opposed to the 0.80 hour from the deterministic model.

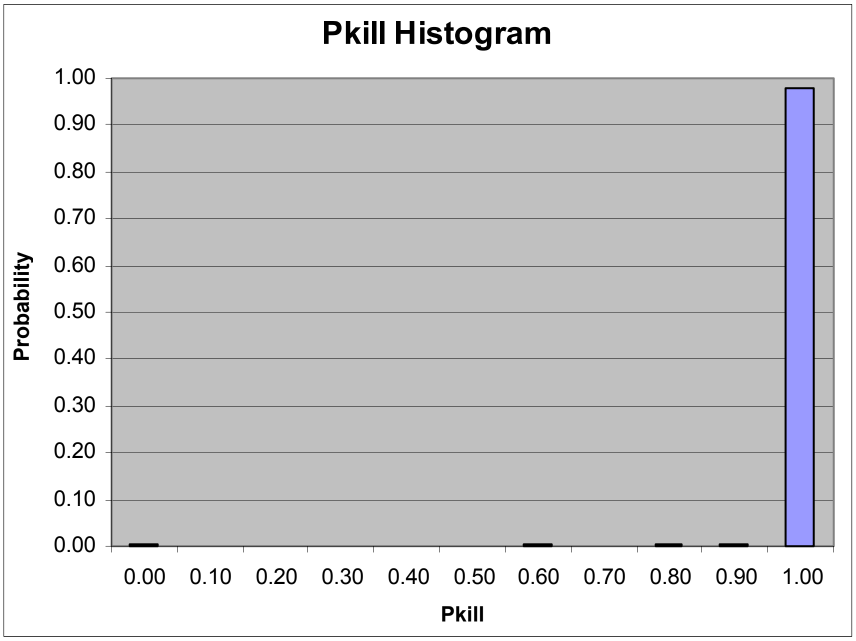

Figure 24. Stochastic Kill Probability (MOE) for Pair 6 (Random Inputs). Although the deterministic model produces a 100 percent Pk result, 2.1 percent of the 1000 stochastic replications have Pk < 0.9.

Conclusions from the benchmarking exercise:

a. In general, the mean of the stochastic outputs should not be expected to match up exactly to the deterministic output, and that is a consequence of the nonlinear transfer function from RAND framework of metrics and measures. Hypothesis tests with null hypothesis that the means of the stochastic outputs are equal to the deterministic outputs produce t-statistics > 4 (note that 2-tailed 99 percent confidence interval has t-statistic of only 2.3), for all 6 pairs of deterministic/stochastic results. This further confirms that the mean of the stochastic outputs do not statistically match up exactly to the deterministic output.

Having said that, the six pairs of deterministic/stochastic results show logical agreement, i.e., a poor (low effective time remaining and Pk) deterministic result has a poor stochastic result, and a good (high effective time remaining and Pk) deterministic result has a good stochastic result.

For any set of search and detection parameters, Pk rises rapidly from zero to close to one within a small range of effective time remaining (zero hour to some “threshold” value). When the mean effective time remaining is significantly higher than the “threshold” value, both deterministic and stochastic models produce consistently high Pks. The deterministic and stochastic Pks start to deviate when the mean effective time remaining drops near, or even below, the “threshold”. In general, deterministic and stochastic models produce the same results only when the results are clear.

b. If the total latency is longer than the submerge time of the enemy KILO submarine, it does not matter how strong the friendly assets’ capability in search and detection is, the Pk is still zero. This reiterates the **importance of C4ISR systems and procedures** in coming up with timely decisions, before any of the physical assets can be effectively put into combat.

c. The initial SSN report delay and the Strike latency shows up as critical factors determining effective time remaining (MOP) and Pk (MOE). This observation **confirms the potential** of RAND’s framework of metrics and measures, which models the importance of the initial SSN report delay and Strike latency through Equation (12) (as part of RAND’s framework), with their w_j set to 1.

d. All the patterns observed/discussed in this section are not possible (or more difficult) without the stochastic simulation model.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. ANALYSIS

In this chapter, the stochastic simulation model developed is used to answer three questions that RAND, and their sponsors are interested in:

a. Does improved C4ISR systems and procedures produce a quantifiable improvement in the battle outcome, i.e., does kill probability increase in the TCT vignette?

b. Which are the critical processing and messaging delay times that impact kill probability the most?

c. How should platforms be assigned to launch the UCAV in the Future Network-Centric system?

A. NETWORK CENTRICITY COMPARISON

A key objective of this thesis is to assess the effects of improved C4ISR systems and procedures on battle outcomes. What it translates to in the TCT vignette case study is, based on RAND's framework of measures and metrics, do Future Network-Centric systems and procedures produce higher kill probability (Pk) than Platform-Centric or Network-Centric systems and procedures? This is the question to be answered in this section. The procedure used to compare the three networks is:

a. Generate *m* sets of inputs (same inputs for all three networks) to be fed to the three networks.

b. Determine the stochastic outputs for the three networks.

c. Compare the three sets of outputs.

1. LHS Variant

An easy way to generate the required *m* sets of inputs is to adopt the method outlined in Table 3 (will be referred to as Simple Random method), which is to randomly (within the bounds stated in the deterministic model) generate the various input variables to make up one set of inputs. Repeat the procedure *m* times.

The other method, which is the preferred method, is a variant of Latin Hypercube Sampling (LHS), and it will be called the LHS variant in this report. The procedure of the LHS variant is:

a. Divide each input variable (all continuous in our case) into *n* equal intervals. The bounds of the input variables are shown in Table 4. Note that the units for all the time variables have been changed (from the original simulation model) to be in hours. This is because it is easier to analyze the results with a common time unit.

Input Variables	**Lower Bound**	**Upper Bound**
Submerge Time (hrs)	0.2	2
Complexity Penalty	0.1	1
Initial SSN (hrs)	0.2	2
CV (hrs)	0.1	1
SubGroup (hrs)	0.1	1
CVN (hrs)	0.1	1
Strike/UCAV (hrs)	0.3	3
DDG (hrs)	0.025	0.25
CG (hrs)	0.025	0.25
Sweep Width (nm)	0.05	0.5
Missile Speed (kts)	200	500
Time b/w Updates (hrs)	0.1	1
KILO Speed (kts)	1	10

Table 4. Inputs' Bounds for LHS Variant. In general, the lower bound is set to 10 percent of the upper bound.

The upper bounds of the input variables are the same as those used in the deterministic model. Most of the lower bounds in the deterministic model are very close to zero, except the missile speed with lower bound of 200 kts. For this analysis, since the bounds are used to define the range whereby the means of the input variables are varied, it is logical for the lower bounds to be non-zero. The lower bounds are set at 10 percent of their upper bounds.

b. Given that 10 percent has been "lopped off", the remaining 90 percent is used to generate 90 equal intervals, i.e., $n = 90$. That means for CVN latency, there are 91 endpoints to the 90 intervals, 0.1 hour, 0.11 hour, 0.12 hour, …, 1 hour. This process of generating 91 endpoints is repeated for all input variables.

c. The next step involves the random selection (without replacement) of an endpoint value from each variable to make up a set of inputs. There are a total of 91 sets of inputs. The S+ codes to generate the 91 sets of inputs, given the bounds and *n*, are attached in Appendix B. A sample (only the first few input variables are shown) of the inputs generated is also attached in Appendix B, Table 12. The advantage of the LHS variant over the Simple Random method is the improvement in coverage of the input space. In addition, LHS has been shown to be efficient under a large range of conditions (Reference 2).

d. A total of 2002 (22 sets of 91) sets of inputs are generated. This is a high number, chosen to enable the comparison of the networks to be conducted with a high confidence level.

2. Outputs from the Three Networks

The outputs (MOP and MOE) generated from the 2002 sets of inputs for the three networks are shown in Figure 25 and Figure 26.

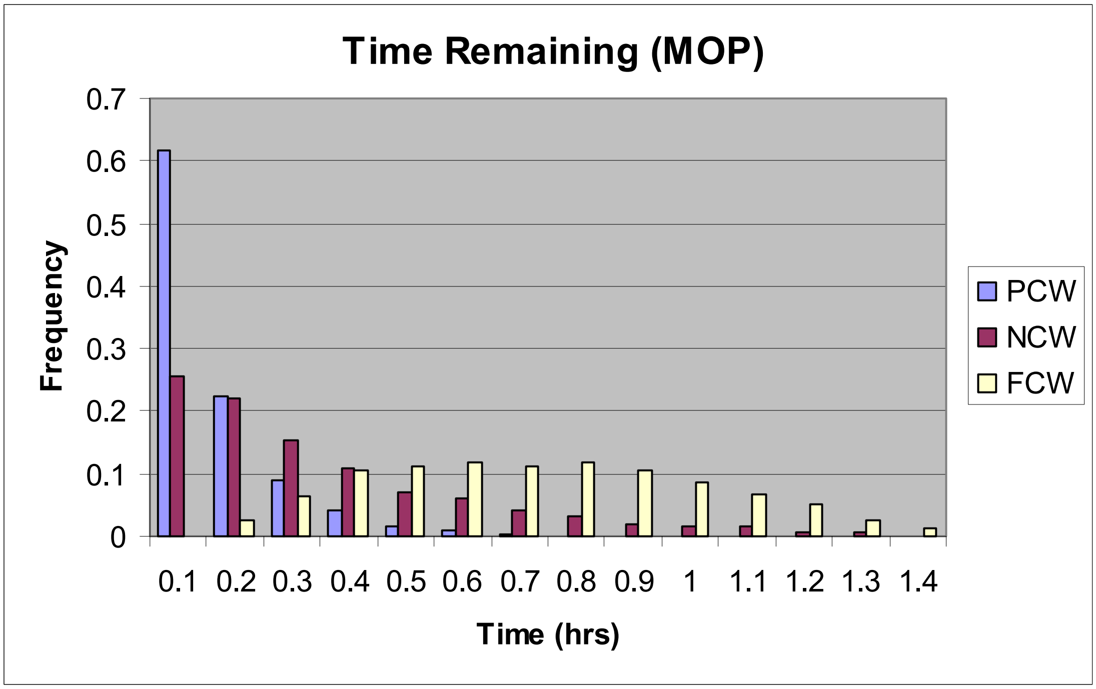

Figure 25. Network Comparison in Effective Time Remaining (MOP). The Future Network-Centric (FCW, mean of 0.68 hour) systems and procedures produce significantly higher effective time remaining than the Platform-Centric (PCW, mean of 0.11 hour) and Network-Centric (NCW, mean of 0.30 hour) cases.

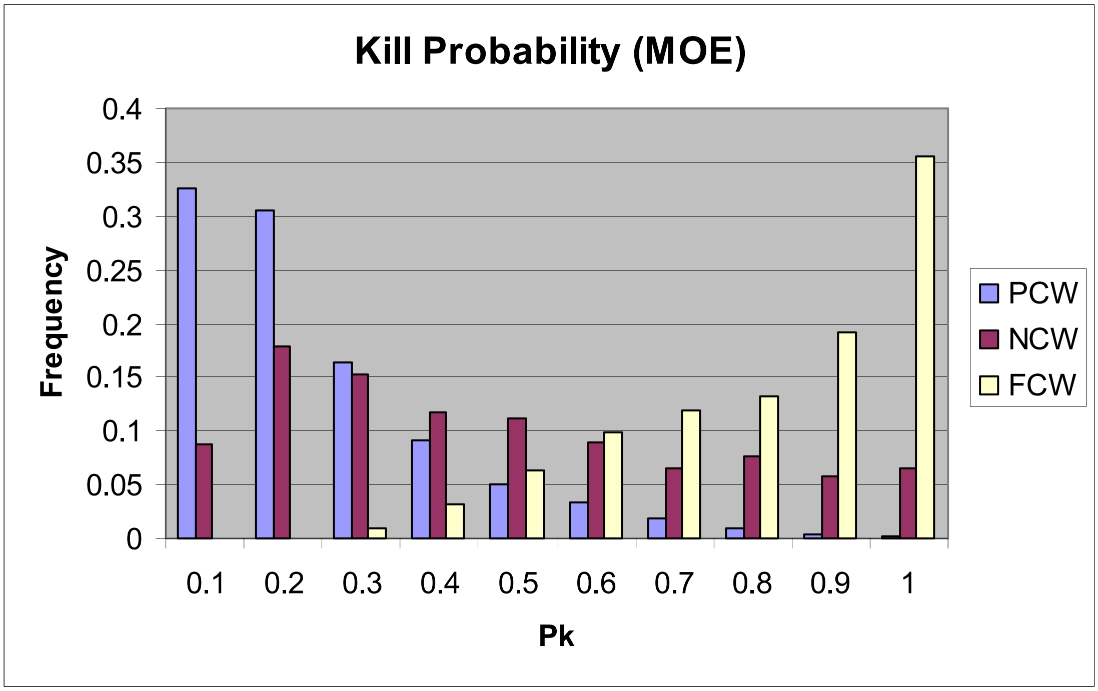

Figure 26. Network Comparison in Kill Probability (MOE). The Future Network-Centric (FCW, mean Pk of 0.78) systems and procedures produce significantly higher Pk than the Platform-Centric (PCW, mean Pk of 0.20) and Network-Centric (NCW, mean Pk of 0.42) cases.

3. Comparison of Networks

The means of the MOP and MOE outputs are listed in Table 5. As mentioned earlier, all the simulation runs in this thesis produce stochastic means estimates with halfwidths of less than 1.5 minutes for the effective time remaining, and 2.5 percent for Pk. This implies that Future Network-Centric (FCW) systems and procedures produce statistically (and practically) superior battle outcomes than Platform-Centric (PCW) and Network-Centric (NCW) cases.

The results confirm the potential of RAND's framework of measures and metrics in modeling the general effects of C4ISR systems and procedures on battle outcomes. What remains to be done is the validation and calibration of the framework, i.e., fine-tuning the framework to achieve results that are consistent with the real world.

Network Centricity	Effective Time Remaining (hrs)	Pk
PCW	0.11	0.20
NCW	0.30	0.42
FCW	0.68	0.78

Table 5. Network Comparison of MOP and MOE. The Future Network-Centric (FCW) systems and procedures performs significantly better than the Platform-Centric (PCW) and Network-Centric (NCW) cases.

B. CRITICAL INPUT VARIABLES

This section answers the question: Which variables affect Pk significantly? The FCW network is used for this analysis as it includes all the input variables, specifically the destroyer (DDG) and cruiser (CG) polling latencies that are applicable only to FCW.

The 2002 input sets used in the previous section are re-used in this analysis. Several models within Clementine[12] are used to determine the critical variables that affect Pk, and extract interesting patterns/relationships within the data. Clementine is a data mining application. Data mining offers a strategic approach to finding useful relationships in large data sets. The main reasons for using Clementine for the data analysis effort are that it's easy to use, and easy to interpret the results generated. In contrast to more traditional statistical methods, the analyst does not necessarily need to know what they are looking for when they start the exploration. The analyst can explore the data, fitting different models and investigating different relationships, until useful information is found.

The Clementine Desktop (Figure 27) makes data exploration easy. The interface uses an approach called visual programming. Various nodes in the workspace represent different objects and actions. The analyst connects the nodes to form streams, which, when executed, enable the analyst to visualize relationships and draw conclusions. Streams are like scripts: which can be saved and reused with different data files.

[12] Clementine is the software used for the data mining course taught at NPS OR department. Interested readers can visit the official website at "http://www.spss.com/spssbi/clementine/".

The Clementine Desktop consists of:

a. Stream pane: The stream pane is the largest area of the Clementine desktop, and is where you build and manipulate data streams.

b. Palettes: The palettes are located across the bottom of the desktop. Each palette contains a related group of nodes that are available to add to the data stream. For example, the Sources palette contains nodes that you can use to read data into your model, and the Graphs palette contains nodes that you can use to explore your data visually.

c. Generated Models palette: The Generated Models palette is located to the right of the stream pane, and it contains the results of machine learning and modeling that you have done.

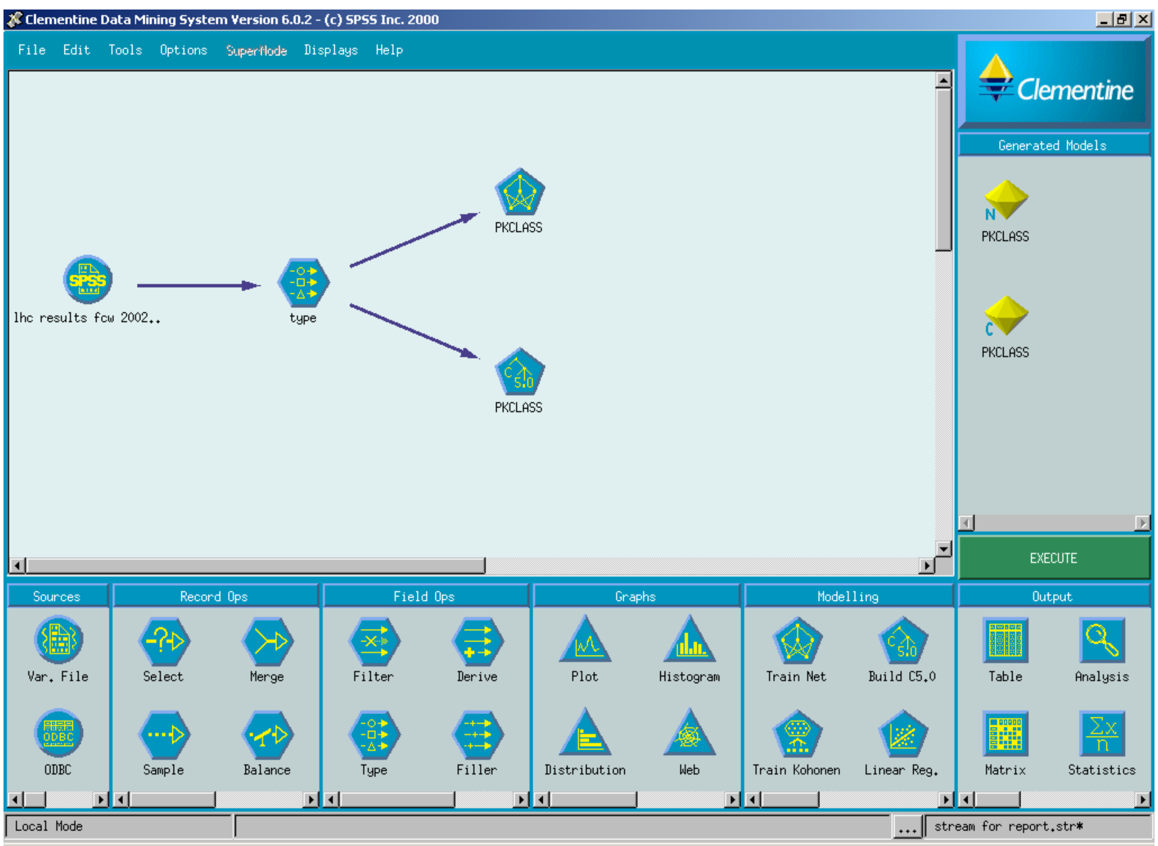

Figure 27. Clementine Desktop. The user drags-and-drop icons from the palettes located across the bottom of the DeskTop, build and manipulate data streams on the Stream Pane (drawing board), and obtain the models' outputs from the Generated Models Palette.

Before the raw data is fed into Clementine, it needs some "touch-up" to maximize the power of the data mining models to be employed. The rule-generating model in Clementine that is used, called C5.0 (see Appendix C for a brief description on C5.0) requires that the output (Pk) analyzed be of type set (e.g., true-false, high-medium-low), i.e., outputs that can be classified into countable classes. In addition to C5.0, there are other rule-generating models that accept continuous outputs, however, in my opinion, they do not produce rulesets as informative as the one produced by C5.0 for the data set that we are working with.

As such, Pk has been divided into three classes[13] (see Table 6) to facilitate the analysis:

Pk Range	**PkClass**
< 0.4	1 (low)
$0.4 \leq Pk < 0.8$	2 (medium)
≥ 0.8	3 (high)

Table 6. PkClass Definition. The choices on the number of Pk classes and the definition of the range for each class are made to separate those cases with high likelihood (PkClass 3, high) of killing the KILO submarine from those with a good chance of mission failure (PkClass 1, low) and those cases in between (PkClass 2, medium).

The distribution of PkClass in the 2002 FCW data set is shown in Figure 28. About 55 percent of the 2002 cases have $Pk \geq 0.8$ (PkClass 3), and there's only a small percentage of cases with $Pk < 0.4$ (PkClass 1).

Note that a common practice to derive better-quality rules (rules that apply to a significant proportion of the cases, and predicts accurately) is to ensure that the classes contain almost equal number of cases. This method has been tried on the current data set, with the Pk range bounds set at 0.7 and 0.9. No improvement is achieved in the quality of the rules/patterns/relationships extracted from the data set. Thus, the PkClass definition is fixed as that stated in Table 6, which at least provides logical definitions for the PkClass.

[13] The choices on the number of Pk classes and the definition of the range for each class are made to separate those cases with high likelihood (PkClass 3, high) of killing the KILO submarine from those with a good chance of mission failure (PkClass 1, low) and those cases in between (PkClass 2, medium).

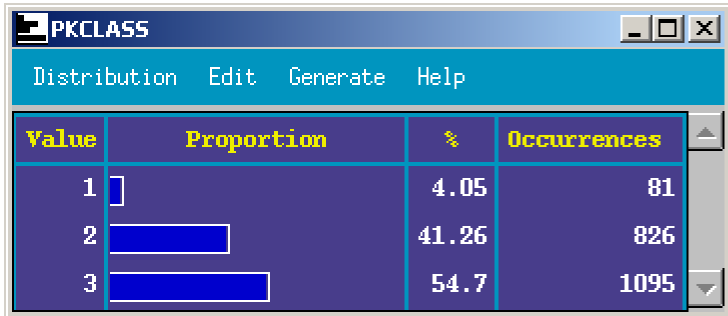

Figure 28. PkClass Distribution. Note the small proportion (4.05 percent) of replications with Pk < 0.4 (PkClass 1). The numbers in the third column add up to 100 percent, and the fourth column adds up to 2002 replications.

1. Neural Network

The first data mining model from Clementine to be used is the neural network (Reference 3) model. See below for the neural network (NN) model generated from the 2002 cases for FCW.

Neural Network "PKCLASS" architecture

Input Layer : 13 neurons
Hidden Layer #1 : 6 neurons
Output Layer : 3 neurons

Predicted Accuracy : 94.60%

Relative Importance of Inputs
STRIKE : 0.52444
Initial SSN Report : 0.51148
DDG : 0.50546
Submerge Time (T) : 0.31567
Mean time between updates (tu) : 0.19936
CG : 0.07770
Complexity Penalty (b) : 0.06990
KILO Speed (w) : 0.06732
Mean Sweep Width (s) : 0.06287
CVN : 0.05975
CV : 0.04829
Missile Speed (v) : 0.03489
SUBGROUP : 0.01900

See the notes below for interpretation of the NN model.

Architecture: The architecture or topology of the network is described. For each layer in the network, the number of units in that layer is listed.

Predicted Accuracy: This is an index of the accuracy of the predictions. For symbolic outputs, this is simply the percentage of records for which the predicted value is correct. For numeric targets, the calculation is based on the differences between the predicted values and the actual values in the training data.

Relative Importance of Inputs: The input variables are listed in order of importance, from most important to least important. The value listed for each input is a measure of its relative importance, varying between zero (a variable that has no effect on the prediction) and 1.0 (a variable that completely determines the prediction).

Note that it is common practice in data mining analysis to split the data set equally into a training set and a test set. The training set is used to develop the models and the test set is then used to evaluate the quality of the models developed. This practice works well if the objective of the analysis is to develop a predictive model, and it guards against overfitting. However, this practice is not adopted for the current analysis as our main objective is to develop a better feel of how the input variables affect the battle outcome, rather than trying to predict the battle outcome from the input variables, since we already know how to do that deterministically.

The interesting portion of the NN model output is the "Relative Importance of Inputs" section, which shows that the three most critical factors that determine PkClass are the Strike/UCAV, initial SSN report, and DDG latencies. These three nodes happen to be the only three nodes in the FCW Task Force. This observation confirms the potential of RAND's framework of metrics and measures which models the importance of the three factors through Equation (12) (as part of RAND's framework), with their w_j set to 1. The other nodes that are not in the FCW Task Force have their w_j set to 0.5. See Table 7 for the w_j settings for the three different network centricity. Note that the w_j for DDG and CG are zero for the PCW and NCW systems, as they are not part of the PCW and NCW systems.

Latency	PCW w_j	NCW w_j	FCW w_j
Initial SSN	0.5	0.5	0.5
CV	1	1	1
SubGroup	1	0.5	0.5
CVN	1	1	0.5
Strike/UCAV	1	1	1
DDG	0	0	1
CG	0	0	0.5

Table 7. w_j for Different Network Centricity. Different nodes make up the Task Force under different network centricity.

The importance of the Strike/UCAV and initial SSN report latencies is reinforced by looking at the plot (Figure 29) of the two latencies, with points of different PkClass in different color. The obvious pattern (not so obvious without color) from the plot is, there are no cases with Pk < 0.8 (PkClass 1 and 2) in the lower left corner (shape of a triangle) of the plot. This observation does not provide information unexpected by the analyst; low latencies lead to high effective time remaining to conduct the search and detection mission, which leads to high Pk. However, what is important about this observation is, regardless of the values (within the bounds defined) of the other input variables in the system, as long as the Strike/UCAV and initial SSN report latencies can be kept within the triangle defined by the plot, we are assured of a high Pk.

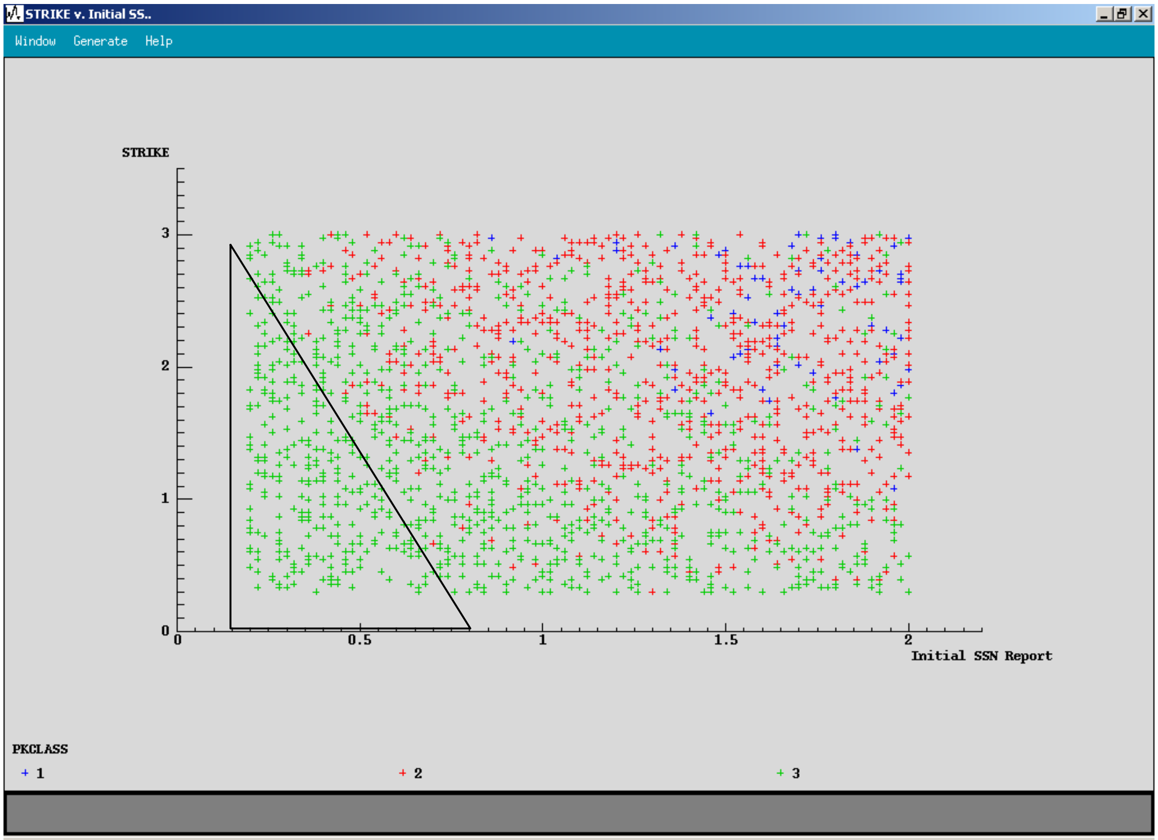

Figure 29. Strike/UCAV vs. Initial SSN Report Plot. As long as the Strike/UCAV and initial SSN report latencies lie within the triangle shown, **regardless** of the values (within the bounds defined) of the other input variables, Pk ≥ 0.8.

In order to confirm that it is indeed the working of RAND's framework of metrics and measures that causes the order of the latencies appearing on the NN model, the PCW results are analyzed using the NN model as well.

The results:

```
Neural Network "PKCLASS" architecture

Input Layer     : 11 neurons
Hidden Layer #1 : 6 neurons
Output Layer    : 3 neurons

Predicted Accuracy :  96.13%

Relative Importance of Inputs
Strike                : 0.31170
Initial SSN Report   : 0.26339
SUBGROUP              : 0.16012
Submerge Time (T)    : 0.12244
CV                    : 0.10718
Mean time between updates (tu) : 0.06753
CVN                   : 0.04239
KILO Speed (w)       : 0.02589
Complexity Penalty ( b ) : 0.01282
Missile Speed (v)    : 0.00790
Mean Sweep Width (s) : 0.00532
```

Note that SubGroup, which is in the PCW Task Force (w_j = 1 for PCW), has "jumped" to near the front of the list, while being last in the list for FCW. Another reason that may explain the importance of an input variable is its range, i.e., the Strike/UCAV latency varies from 0.3 to 3 hours, while CV varies from 0.1 to 1 hour, given that they have the same w_j settings in the PCW network, the latencies with the bigger numbers will have more "weight" in determining Pk.

2. C5.0 Rulesets

The C5.0 model (see Appendix C for a brief description) in Clementine can produce two kinds of models, decision tree or rulesets. The rulesets are the ones that are used in this analysis, as any patterns/relationships in the data are easy to extract and interpret from the rulesets. The rulesets (for output PkClass) generated from the FCW data are shown in Appendix D.

Note that there is a pair of numbers accompanying each rulesets. These numbers show information on the number of cases to which the rule applies (instances) and the proportion of those cases for which the rule is true (confidence).

Note that confidence is calculated as:

(1 + number of cases where rule is correct) / (2 + number of cases to which rule applies)

This calculation of the confidence estimate adjusts for the process of generalizing rules from a decision tree (which is what C5.0 does when it creates a ruleset).

A good rule is one that has:

a. High number of instances: The rule applies to a large proportion of the data set.

b. High confidence: For those cases that satisfy the conditions of the rule, the rule predicts the correct PkClass most, if not all of the time.

Note that the main objective in using C5.0 is not to predict PkClass from the various input variables, since we know exactly how to calculate Pk from the input variables. Rather, we aim to gain a better feel of the weightage of the various input variables in the overall combat picture.

An example of how the rules should be interpreted: Rule #1 for PkClass 3 (Pk >= 0.8): If the mean submerge time of the enemy submarine > 1.2 hours, and the mean initial SSN report latency <= 1.4 hours, and mean DDG latency <= 0.09 hour, then there is a 98.2 percent chance that Pk >= 0.8, **regardless** of the other input variables.

A few interesting rules generated from the C5.0 model are highlighted for discussion.

Rule#1 for PkClass 1 (Pk < 0.4):

```
if  Submerge Time (T) <= 0.92
and  Initial SSN Report > 1.3
and  Strike > 1.77
and  DDG > 0.142
and  CG > 0.067
then -> 1 (56, 0.845)
```

With a low submerge time and high latencies; Pk has a high chance of being low (< 0.4). There are 9 cases out of the 56 cases that satisfy the conditions with PkClass 2, and the maximum Pk from these 9 cases is 0.54, with 6 cases below 0.50.

Rule #1 for PkClass 3 (Pk >= 0.8):

if Submerge Time (T) > 1.2
and Initial SSN Report <= 1.4
and DDG <= 0.09
then -> 3 (166, 0.982)

Analyzing the data shows that of the 166 cases that satisfy the condition, there are two cases that are PkClass 2, instead of the predicted PkClass 3. However, these two case are exceptionally high PkClass 2, and their Pks are 0.7981, 0.7999, that is, they can almost be considered PkClass 3.

Rule #2 for PkClass 3 (Pk >= 0.8):

if Strike <= 1.56
and DDG <= 0.09
and Mean time between updates (tu) <= 0.6
then -> 3 (158, 0.981)

Analyzing the data shows that of the 158 cases that satisfy the condition, there are two cases that are PkClass 2, instead of the predicted PkClass 3. However, these two case are high PkClass 2, with Pks of 0.77 and 0.78.

This rule is more useful than the previous rule in that it says, if the friendly forces can keep the mean Strike/UCAV and DDG latencies below certain times, and get timely updates, there is a high chance of having a high Pk, **regardless** of how soon the enemy submerges, or how other input variables vary. This rule is important in that it sets target levels that the friendly forces can work towards.

Rule #3 for PkClass 3 (Pk >= 0.8):

if DDG <= 0.045
then -> 3 (197, 0.98)

If the friendly forces can achieve a mean destroyer (DDG) latency of <= 0.045 hour (2.7 minutes), then there's a 98 percent chance that Pk >= 0.8, **regardless** of the other input variables. Analyzing the data shows that of the 197 cases that satisfy the condition, there are three cases that are PkClass 2, instead of the predicted PkClass 3. However, these three cases are high PkClass 2, and their Pks are 0.74, 0.77, and 0.77, very close to the PkClass 3 range.

The default rule says that if none of the rules apply to a case, assign PkClass 3 to the case. This is a direct result of PkClass 3 being the majority class

Observations/Conclusions from the C5.0 model:

a. The input variables that show up most in the rules are the same ones "leading" the list for the neural network model developed in the previous section, i.e., Strike/UCAV, initial SSN, DDG, and submerge time.

b. There are fewer rules for PkClass 1 because only about 4 percent of the 2002 cases are of PkClass 1 (see Figure 28).

3. Linear Regression

The Clementine linear regression model estimates the best fitting linear equation for predicting the output based on the input variables. The regression equation represents a straight line or plane that minimizes the squared differences between predicted and actual output values. For this analysis, all 13 input variables are used to fit an equation for Pk (not PkClass as it will not be logical). The resultant linear regression equation:

-0.004307 * KILO Speed (w) +
-0.132599 * Mean time between updates (tu) +
0.000006 * Missile Speed (v) +
0.068051 * Mean Sweep Width (s) +
-0.183708 * CG +
-1.29667 * DDG +
-0.109762 * Strike +
-0.025286 * CVN +
-0.0106 * SUBGROUP +
-0.016936 * CV +
-0.186999 * Initial SSN Report +
-0.012818 * Complexity Penalty (b) +
0.109087 * Submerge Time (T) +
1.361112

Similar conclusions from the neural network and C5.0 ruleset models are obtained, i.e., critical inputs have relatively bigger coefficients than those unimportant inputs. Note that another factor that may affect the size of the coefficients are the ranges of the variables, i.e., missile speed with a range between 200 and 500 kts will generally have a lower coefficient that complexity penalty that ranges from zero to one, although both these variables may be as insignificant in affecting Pk.

As mentioned, the main objective of this analysis is to obtain a better feel of the importance of each input variable in the final battle outcome, and not to build a model to predict Pk from the inputs, since the exact formulas for calculating Pk from the inputs are known. Therefore, no further exploration or analysis of the linear regression model is conducted.

In this section, three data mining models have been used to determine the variables that have the greatest impact on the kill probability. All three models arrive at the same conclusion that the critical variables to the time-critical target vignette, Future Network-Centric system, are the Strike/UCAV latency, initial SSN report latency, DDG latency, and enemy submarine submerge time. The two general factors that determine the impact of an input variable on kill probability are: (i) whether the system is part of the Task Force and (ii) the range of the input variable.

C. POLLING OPTIONS FOR FCW

The question to be answered in this section is: How should platforms be assigned to launch the UCAV in the Future Network-Centric system? This is essentially a command and control question that addresses the way the richly connected network is utilized to support combat operations. The three options (see Table 8)[14] require different times for collaboration and UCAV fly out.

[14] Table extracted from Reference 1.

Option	Process	Impact on Operations
Case 1: Complete Polling at execution time	Poll all potential combatants with UCAVs and select the one that can get to the target quickest	Large cost in collaboration time Most reliable solution Fastest fly out time for UCAV
Case 2: Periodic selection of a subset of combatants with UCAVs	Poll a select subset of combatants with UCAVs considered to be in the best position to respond. Repeat this process periodically	Less cost in collaboration time Least reliable solution Moderate increase in fly out time
Case 3: Periodic complete polling of combatants with UCAVs	Poll all combatants with UCAVs periodically and designate one as the "duty" launcher	Moderate cost in collaboration time Less reliable solution Possibly greatest fly out time for the UCAV

Table 8. Polling Options for FCW. Different polling options have different effects on collaboration and UCAV fly out.

1. Case 1: Complete Polling at Execution Time

Although the most reliable method (in the sense that the target's location is known and therefore, distances to the target are known), considerable time is absorbed by collaborating to arrive at an "optimal" selection based on distance to the target. Calculating the distances to the target from the candidate platforms at execution time means that the time required to fly to the release point for the SLAM-ER is minimized.

2. Case 2: Periodic Polling of a Subset at Execution Time

In this case, a periodically selected subset of the platforms with the UCAV is polled at execution time. Because the number of platforms polled is reduced, the collaboration time required at execution is not as great. The fact that the pre-selection is time consuming has little impact on the delay at execution time. The reliability of the pre-selected choice in terms of the time required to reach the target is however, reduced. There are two reasons for this: (i) the selection of the subset of platforms was based on conditions that may not be prevalent at execution time, and (ii) closely related is the fact that the platform that is selected to execute may be sub-optimal when compared to the entire set of platforms with the UCAV. The impact on fly out time is that it will likely be extended.

3. Case 3: Periodic Complete Polling

In this case, the entire set of platforms with the UCAV is polled periodically. The fact that polling takes place prior to the operation means that little time is spent deciding which platform will launch the UCAV at execution time. The reliability of the pre-selected choice however, is less reliable than selection at execution time. In this case, the fact that all platforms are polled mitigates the deficiency somewhat. The impact on fly out time for the UCAV is greater than the first case, but not as long as the second.

4. Analysis of the Polling Options

Table 9 lists the mean times associated with the three options discussed. Note that only the times that are likely to vary based on the conditions described are listed. The procedure adopted to compare the effectiveness of the polling options is:

a. Using the same 2002 input sets generated earlier, replace the latencies for those input variables stated in Table 9 with the values for Case 1.

b. Run the stochastic simulation under FCW.

c. Repeat the above steps for Case 2 and 3.

Option	**DDG Polling**	**CG Polling**	**CVN Polling**	**CV Polling**	**UCAV Fly out**	**Total**
Case 1: Complete Polling at execution time	15	10	17	17	5	64
Case 2: Periodic selection from a subset of UCAV platforms	8	7	-	-	20	35
Case 3: Periodic complete polling of UCAV platforms	8	7	9	9	10	43

All times in minutes.

Table 9. Time Estimates for Polling Options. Different polling options require different times for collaboration and UCAV fly out.

The MOP and MOE histograms are shown in Figure 30 and Figure 31 respectively.

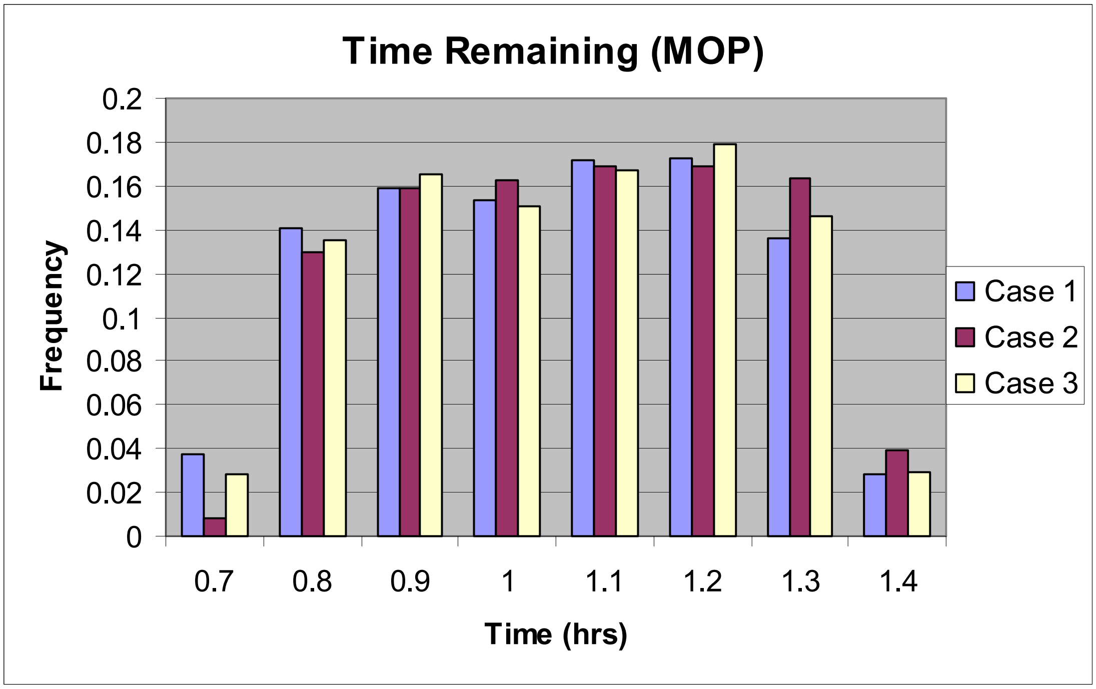

Figure 30. Polling Options Comparison in Effective Time Remaining (MOP). Periodic selection from a subset of UCAV platforms (polling option Case 2) produces slightly higher effective time remaining than the other two polling options.

The means of the MOP and MOE are stated in Table 10. Note that the results in the current section show superior performance (higher Pk) compared to the previous sections. This is because the latencies used in the three polling options have means significantly lower than those of previous sections, which leads to higher effective time remaining and subsequently, higher Pk.

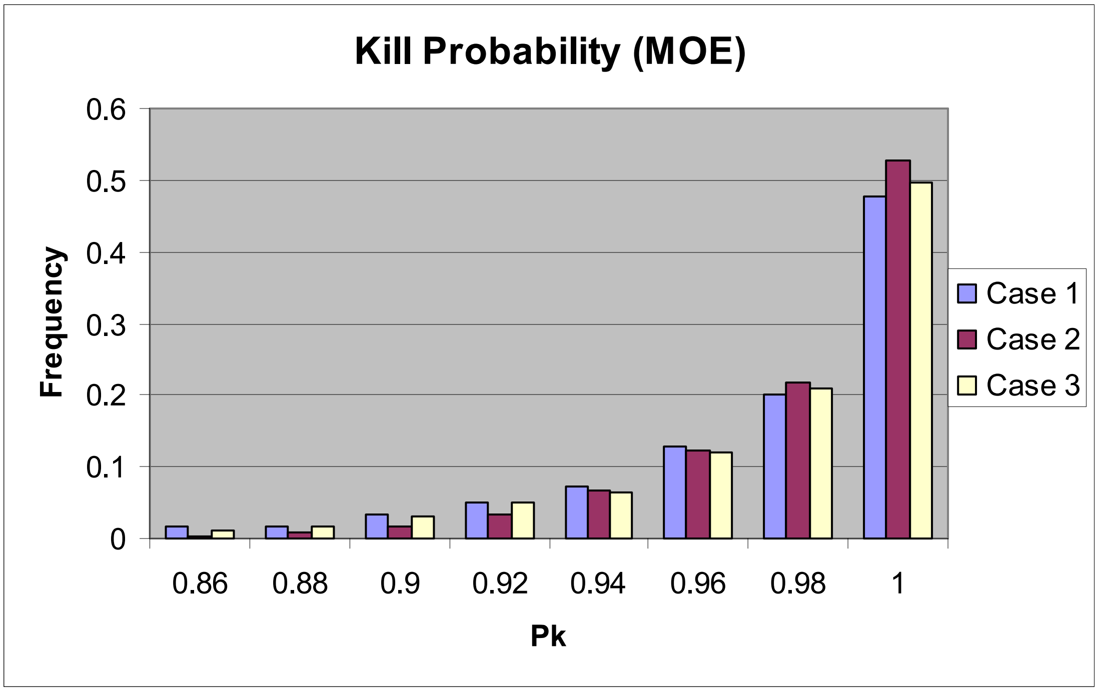

Figure 31. Polling Options Comparison in Kill Probability (MOE). Periodic selection from a subset of UCAV platforms (polling option Case 2) produces slightly higher kill probability (Pk) than the other two polling options.

Polling Option	Effective Time Remaining (hrs)	Pk
Case 1	0.99	0.966
Case 2	1.04	0.973
Case 3	1.00	0.969

Table 10. Polling Options Comparison of MOP and MOE. No significant differences between the three polling options.

The results show that Case 2 is slightly more effective than the other two polling options, but does that constitute a significant practical difference? The analysis from the stochastic simulation model shows that there are no significant practical differences between the three polling options. However, if this conclusion is inconsistent with the real-world situation, there is a need to review the framework of measures and metrics. Have the positive effects of collaboration been overly "exaggerated", so much so as to "squeeze" the effects of latencies, that a difference of 29 minutes (Table 9 Case 1 total latency of 64 minutes vs. Case 2 total latency of 35 minutes) in latencies when passed through the framework produce outputs that are insignificantly different. Or are there other reasons? This will be part of the future validation required on the framework of measures and metrics.

V. CONCLUSIONS

Based on RAND's framework of measures and metrics to assess the impact of C4ISR systems and procedures on battle outcomes, a stochastic simulation model has been developed, benchmarked, and utilized to analyze issues that are important to RAND's research for the U.S. Navy.

A. BENCHMARKING

The developed simulation model is benchmarked against RAND's existing deterministic model, and it produces consistent results with the deterministic model, i.e., low kill probability (MOE) in the stochastic model generally goes with low kill probability in the deterministic model, and vice versa. Having said that, the mean of the stochastic outputs should not be expected to match up exactly to the deterministic output, and that is a consequence of the nonlinear transfer function from RAND's framework of metrics and measures.

For any set of search and detection parameters, Pk rises rapidly from zero to close to one within a small range of effective time remaining (zero hour to some "threshold" value) to conduct the search and detection mission. When the mean effective time remaining is significantly higher than the "threshold" value, both the deterministic and stochastic models produce consistently high Pks. The deterministic and stochastic Pks start to deviate when the mean effective time remaining drops near, or even below the "threshold".

In general, deterministic and stochastic models produce the same results only when the results are clear, e.g. in another combat context, two opposing sides (blue-to-red) with 100-to-1 ratio, and similar combat effectiveness, will produce similar results from both deterministic and stochastic simulation model, a 100 percent win for the blue force. However, when it becomes a 1.1-to-1 ratio, the deterministic model will still predict a 100 percent win for the blue force, while the stochastic simulation model will produce a more realistic result that blue force may not always win.

B. NETWORK CENTRICITY COMPARISON

The stochastic simulation results show that Future Network-Centric systems and procedures produce significantly higher kill probabilities than the Platform-Centric and Network-Centric case. The results confirm the potential of RAND's framework of measures and metrics in modeling the general effects of C4ISR systems and procedures on battle outcomes. What remains to be done is the validation of the framework, i.e., fine-tuning the framework to achieve results that are consistent with the real world.

C. CRITICAL INPUT VARIABLES

Three data mining models have been used to determine the variables that have the greatest impact on the kill probability. All three models arrive at the same conclusion that the critical variables to the time-critical target vignette, Future Network-Centric system are the Strike/UCAV latency, initial SSN report latency, DDG latency, and enemy submarine submerge time. The two general factors that determine the impact of an input variable on kill probability are: (i) whether the system is part of the Task Force and (ii) the range of the input variable.

D. POLLING OPTIONS FOR FCW

There are no significant differences between the three polling options to assign the platform for launching the UCAV in the Future Network-Centric system. If this conclusion is inconsistent with what we expect in real-world situations, there is a need to review the framework of measures and metrics.

APPENDIX A. SIMULATION DEVELOPMENT

A. VARIABLE DISTRIBUTIONS

Table 11 documents the distributions used to represent the various latencies and the search and detection variables. The distributions have been discussed and agreed with RAND (and through them, their Navy sponsors.)

Input Variable	**Distribution**
Submerge Time	Beta
Complexity Penalty	Constant
Initial SSN Report	Gamma
CV	Exponential
SubGroup	
CVN	
UCAV	
DDG	
CG	
Mean Sweep Width	Beta
Missile Speed	Beta
Mean Time b/w Updates	Exponential
KILO Speed	Beta

Table 11. Variable Distributions. The distributions have been discussed and agreed with RAND (and through them, their Navy sponsors).

B. DATA ENTRY FORM

The data entry form (Figure 32) is created using Visual Basic for Applications (VBA) in Excel. It is activated by clicking the "Simulation EDA Tool" command button in two locations, “Vignette2” worksheet cell D23, and “SimGen” worksheet cell C3.

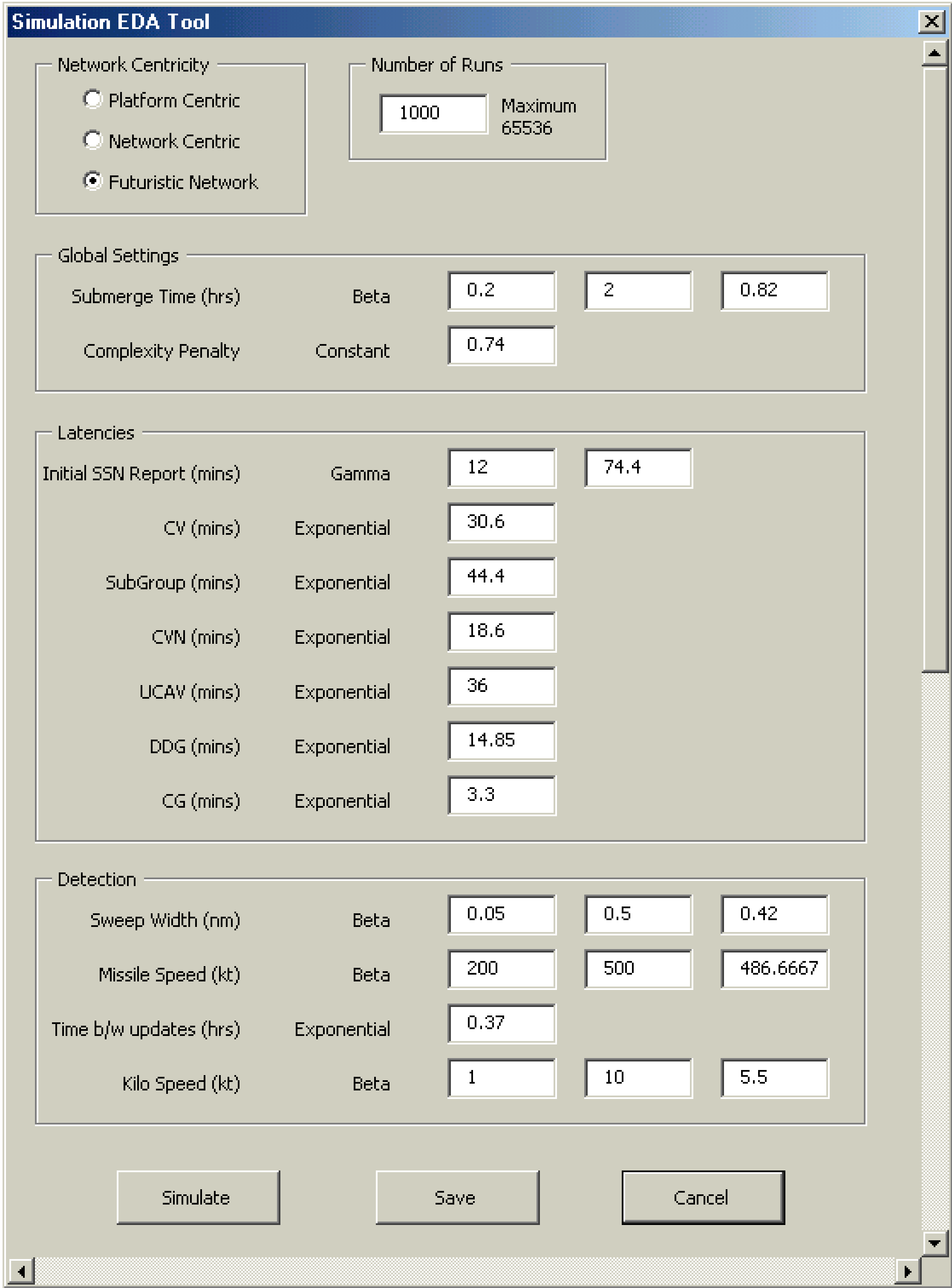

Figure 32. Data Entry Form. The stochastic simulation model requires parameters for 13 input variables, segregated into three frames, “Global Settings”, “Latencies”, and “Detection”.

Features Description (in order of top-down, left-to-right on the form)

a. Network Centricity: Select the network centricity to be analyzed.

b. Number of Runs: Enter the number of runs/replications for the simulation. Estimated run time is approximately 50 seconds for 1000 runs on a Pentium III, 667 Mhz PC with 128 Mb RAM. In the current model design, the number of rows in a single worksheet restricts the maximum number of replications, which at 65,536 is more than sufficient for the purposes of this study.

c. Global Settings

   i. Submerge Time (hrs): The distribution is beta with three parameters, minimum, maximum and mode, from left to right.

   ii. Complexity Penalty: This variable is a constant between zero and one. It is used as a multiplying factor to adjust *b* in Equation (15).

d. Latencies

   Note that the units of time for submerge time and the other time-related latencies are different. This is a direct result of the fact that latencies are usually much shorter than submerge time, and so it's easier for the users to provide the latencies in minutes rather than in hours.

   i. Initial SSN Report (minutes): The distribution is gamma with two parameters, minimum and mean. However, there need to be 3 parameters to pin down a gamma distribution. I have currently assumed that the parameters alpha = beta.

   ii.. CV to CG (minutes): The distributions are exponential with mean as the only parameter.

e. Detection

   i. Sweep Width (nm), Missile Speed (kts), and KILO Speed (kts): The distributions are beta with three parameters, minimum, maximum and mode, from left to right.

ii. Time Between Updates (hrs): The distribution is exponential with mean as the only parameter.

f. Command Buttons (at bottom of the form)

i. Simulate: The inputs entered on the data entry form are saved in the spreadsheet, and the simulation will start. After the simulation ends, the data entry form is closed and the results (MOP and MOE histograms and confidence intervals) are presented on the spreadsheet.

ii. Save: The inputs entered on the data entry form are saved in the spreadsheet. The data entry form will remain open. This is useful when you need to verify certain data in the midst of a data entry session, and you want to save the portion of the data that are already entered.

iii. Cancel: The changes made on the current data entry form are ignored, and the data entry form will be closed.

iv. Close button at the top right hand corner: Same effect as Cancel.

g. Entering Inputs: The user can enter data sequentially using the "Enter" or "Tab" keys. If the user needs only to change a few parameters, it may be easier to use the mouse to highlight the input cells that require change.

h. Tool Tip: All the input cells provide the user with the type of parameters required, i.e., when you move the mouse over an input cell, the screen will show “minimum”, “maximum”, “mode”, or “mean”.

i. Data Verification: The spreadsheet automatically verifies the data that the user has entered before saving or simulating, i.e., the spreadsheet prompts the user to re-enter values if, e.g., minimum > maximum for one of the beta distributions.

j. Distribution Parameters: Other than saving the raw data on the form to the spreadsheet when the “Save” button is clicked, the alphas and betas of the distribution are calculated from the minimums, means, etc., and saved in the worksheet.

C. MS EXCEL IMPLEMENTATION

The main benefit of implementing the RAND EDA tool stochastic simulation in MS Excel is its widespread availability in DoD organizations. It provides a universal platform that users of all levels are comfortable with, and thus reduces any unnecessary startup time to familiarize with the application's user environment.

Furthermore, since the original RAND EDA tool is implemented in Excel, it makes sense to "attach" the stochastic simulation model to the original tool as long as the limitations of the Excel applications does not overly restrict the analysis capability required of the study, which is the case here.

The formulas and assumptions modeled into MS Excel are documented below in the order of the developmental process, i.e., generation of random variables from the user-defined parameters, calculation of the effects of collaboration and complexity to the total latencies, calculation of the confidence interval of the effective remaining time (MOP) and kill probability (MOE), and their histograms. The entire stochastic simulation model is coded in the **"SimGen"** worksheet within the RAND EDA tool.

1. Random Variables Generation

The cells A1-L18 on the "SimGen" worksheet are used to generate the random variables.

a. Beta (min, max, mode): The following algorithm/pseudo code is implemented to compute the alpha and beta parameters from the parameters that the user provides. As mentioned in the previous section, the alphas and betas of the distribution are calculated and saved in the worksheet when the "Save" button is clicked.

$$\text{mean} = \frac{\min + \max + \text{mode}}{3}$$

$$\text{variance} = \frac{\min^2 + \max^2 + \text{mode}^2 - \min \times \max - \min \times \text{mode} - \max \times \text{mode}}{18}$$

$$\text{mean} = \frac{\text{mean} - \min}{\max - \min}$$

$$\text{variance} = \frac{\text{variance}}{(\max - \min)^2}$$

$$\text{temp} = \frac{\text{mean}}{1 - \text{mean}}$$

$$\text{beta} = \frac{\frac{\text{temp}}{\text{variance}} - (\text{temp} + 1)^2}{(\text{temp} + 1)^3}$$

$$\text{alpha} = \text{temp} \times \text{beta}$$

The underlying principles for the above algorithm comes from the fact that the input variables that are fitted with a beta distribution are those with obvious minimum and maximum bounds, and a nominal value, similar to a triangular distribution. Thus, the minimum, maximum and mode of the triangular distribution are transformed to derive the alpha and beta parameters of a beta distribution. The means and variances (Reference 4) of a triangular distribution are matched up with that of a beta distribution to derive the parameters of the beta distribution.

For the triangular distribution:

$$\text{mean} = \frac{\min + \max + \text{mode}}{3}$$

$$\text{variance} = \frac{\min^2 + \max^2 + \text{mode}^2 - \min \times \max - \min \times \text{mode} - \max \times \text{mode}}{18}$$

For the beta distribution:

$$\text{mean} = \frac{\alpha}{\alpha + \beta}$$

$$\text{variance} = \frac{\alpha\beta}{(\alpha + \beta)^2 (\alpha + \beta + 1)}$$

With the alpha and beta parameters, the Excel implementation for the beta random variable is: “=BETAINV(RAND(), alpha, beta, min, max)”, where RAND() is the Excel function to get a Uniform(0,1) random variable, and BETAINV is the inverse beta function.

b. Gamma (min, mean): The following algorithm/pseudo code is implemented to compute the alpha and beta parameters from the user-provided minimum and mean. The assumption of the algorithm is, parameter alpha = beta. This assumption is necessary, as 3 parameters are required to “pin down” a gamma distribution. Another way to resolve this problem is to ask the user to provide the third parameter other than minimum and mean, either the variance or the mode of the distribution, which may not be easy for the user.

$$\begin{aligned}&\text{mean} = \text{mean} - \text{min} \qquad \text{'comments: normalize the mean}\\&\text{alpha} = \sqrt{\text{mean}}\\&\text{beta} = \text{alpha}\end{aligned}$$

For the gamma distribution (Reference 4):

$$\begin{aligned}\text{mean} &= \alpha\beta\\\text{variance} &= \alpha\beta^2\end{aligned}$$

The Excel implementation is: “=GAMMAINV(RAND(), alpha, beta) + min”, where GAMMAINV is the inverse gamma function. Due to Excel’s characteristics, when the mean provided by the user is very close to the min, the result from GAMMAINV(RAND(), alpha, beta) may sometimes be smaller than the smallest number presentable in Excel, and Excel will output “#NUM!” in the cell. This causes error in the simulation output.

Two additional cells B35-B36 are used to solve this problem. B35 has the formula “=GAMMAINV(RAND(), alpha, beta) + min”, and B36 checks if the result from B35 is so small that “#NUM!” is the output. The resultant gamma random variable generated in cell C8 is the result of an “if-else” statement based on B35-B36.

c. Exponential (mean): The Excel implementation is:

"=-mean*LN(RAND())"

where:

LN is the natural logarithm.

2. Collaboration

Collaboration acts to reduce the expected time to complete the mission. The effects of collaboration on each node are different, depending on the knowledge of those nodes that they are connected to. The mathematical form (in the framework of metrics and measures recommended by RAND) of the contribution of collaboration to node *i*'s effective latency is expressed as the product in Equation (12), repeated below:

$$\prod_{j=1}^{n_i}\left(1-K_j(t)\right)^{\omega_j}$$

where:

$K_j(t)$ is the knowledge function of node *j*, it represents the quality of the processes and equipment at node *j*, 1.0 represents high quality, and 0.0 implies low quality

n_i is the degree of node *i*

$$\omega_j = \begin{cases} 0.5 & \text{if node } j \text{ is not in the Task Force} \\ 1.0 & \text{if node } j \text{ is in the Task Force} \end{cases}$$

Cells X26-BF52 calculate the collaboration contributions for each node, under each network centricity. The intermediate results:

a. Original Latencies (X35-AD38): These are the random numbers generated from the distributions. These numbers change for each replication.

b. Information Entropy (X39-AD42): All latencies, except the initial SSN report latency have an exponential distribution. The mean latencies of the exponential distributions provide the λ parameter required to calculate the knowledge function. The computation of the knowledge function for the gamma distributed initial SSN report is different from the exponential distribution, and it is explained next.

c. Knowledge Function (AE39-AK42)

i. Exponential Distribution (mean $1/\lambda$): The formula to calculate the knowledge function for the exponential distribution is stated in Equation (7), repeated below:

$$K(t) = \begin{cases} 0 & \text{if } \lambda < \lambda_{\min} \\ \dfrac{\ln\lambda - \ln\lambda_{\min}}{\ln M} = \dfrac{\ln(\lambda/\lambda_{\min})}{\ln M} & \text{if } \lambda_{\min} \le \lambda < M\lambda_{\min} \\ 1 & \text{if } \lambda \ge M\lambda_{\min} \end{cases}$$

where:

λ_{min} represents the minimum rate that corresponds to the maximum expected time, λ_{min}[15] is chosen to be 0.5, implying that the maximum expected latency is 2.0 hours. *M* is chosen to be 40, implying perfect knowledge if the expected latency is ≤ 1/20 hour.

ii. Gamma Distribution (α and β): The information entropy of a gamma distribution (Reference 5) is:

$$H(d) = \ln[\beta\Gamma(\alpha)] - (1-\alpha)\psi(\alpha) + \alpha$$

where:

$\psi(\alpha)$ is the first derivative of Euler's gamma, $\psi(\alpha) = \dfrac{d}{d\alpha}\Gamma(\alpha)$

The following code (Reference 6) can compute an approximation to $\psi(\alpha)$ accurate to 10 decimal places:

[15] Note that although information entropy is a universally accepted theory, the knowledge function is part of the framework of measures and metrics recommended by RAND, with the choices of λ_{min} and *M* based on educated guesses.

```
function psi(x)
  x = x + 6;
  p = 1 / x^2;
  p = 0.004166666666667 p^4 - 0.003968253986254 p^3 +
     0.008333333333333 p^2 - 0.83333333333333 p;
  p = p + ln(x) - (0.5 / x) - 1/(x - 1) - 1/(x - 2) - 1/(x - 3) -
     1/(x - 4) - 1/(x - 5) - 1/(x - 6);
  p = -p;
return(p);
end;
```

An appropriate knowledge function (Reference 6) is:

$$K(d) = 1 - e^{\phi[H(d) - H_{\max}(d)]}$$

When $H(d) = H_{\max}(d)$, knowledge K(d) is zero. Therefore, we associate minimum knowledge with maximum entropy as desired. As $H(d)$ gets smaller, knowledge improves.

d. Knowledge Functions $\left(1 - K_j(t)\right)^{w_j}$ (AE35-AK38): Under different network centricity, the Task Force consists of different nodes, which means different w_j for the nodes.

e. Product of Knowledge Functions for Different Network Centricity $\prod_{j=1}^{n_i} \left(1 - K_j(t)\right)^{\omega_j}$ (AL35-BF38): Under different network centricity, each node is connected to a different set of nodes, i.e., it collaborates with a different set of nodes.

f. Latencies for Different Network Centricity (BG35-CD38): The three sets of effective latencies are calculated from the product of the knowledge functions and the original latencies.

g. Collaboration-Induced Latency (CE35-CE38): The total effective latency, considering the positive effects of collaboration, for the network centricity chosen.

3. Complexity

The number of connections within the TCT network increases from Platform-Centric (4) to Network-Centric (8) to Future Network-Centric (12)[16]. The complexity factor to be introduced into the expected latency metric is:

$$g(C)=\frac{e^{-7+\frac{14}{45}\beta C}}{1+e^{-7+\frac{14}{45}\beta C}}$$

where:

β is the user-provided complexity penalty, between zero and one

C is the number of connections

Figure 9 illustrates a typical complexity function for zero to 45 possible connections of the TCT network.

The complexity/collaboration-induced latency is calculated (CF32-CK38) by:

$$\text{Complexity/Collaboration induced latency}=\frac{1}{1-\mathrm{g}(\mathrm{C})}\times\text{Collaboration induced latency}$$

4. Effective Time Remaining (MOP)

The effective time remaining (MOP) is calculated (CM35-CM38) by subtracting the complexity/collaboration-induced latency from the submerge time of the KILO submarine.

5. Kill Probability (MOE)

The kill probability formula as stated in Equation (22) is:

$$P_k(T)=1-e^{-\frac{svk^2}{[1-K(T)]\pi(wt_u)^2}T}$$

The formula is implemented in cells A23-A33.

[16] As with all other aspects of the framework of measures and metrics, the number of connections are based on educated guesses, validation on the number of connections remains a future task.

6. Replicating the Simulation

With the user-provided parameters, a VBA Excel macro will automatically replicate the simulation, drawing different random numbers for each replication. As a rough guide, the estimated run time for 1000 replications is 50 seconds on a Pentium III, 667 Mhz Pc with 128 Mb RAM.

7. Outputs

The confidence intervals (user can define the confidence level) for the effective time remaining (MOP) and kill probability (MOE) are calculated in cells EF1-EH11, using conventional statistics formula. The histograms for the MOP and MOE are also plotted.

APPENDIX B. LHS S+CODES

The original version of the codes below has been provided by Thomas W. Lucas. Slight modifications to it have been made for this analysis.

```
LHC <- function(theMatrix, npoints)
{
      f <- function(m,n)
            {
                        lb <- m[1]
                        ub <- m[2]
                        i <- (m[2]-m[1])/(n-1)
                        return(seq(m[1],m[2],i))
            }
            hyper.design.temp <- apply(theMatrix, 1, f, npoints)
            hyper.design <- apply(hyper.design.temp, 2, sample)
            return(hyper.design)
}

temp <-
matrix(c(0.2,0.1,0.2,rep(0.1,3),0.3,rep(0.025,2),0.05,200,0.1,1,2,1,2,1
,1,1,3,0.25,0.25,0.5,500,1,10), ncol=2)
temp
npoints <- 91
out.design <- LHC(temp, npoints)
out.design
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C. LHS INPUT SETS

Replication #	Submerge Time (T)	Complexity Penalty (b)	Initial SSN Report	CV	SubGroup	CVN
1	0.6	0.2	0.62	0.14	0.71	0.66
2	0.7	0.88	1.28	0.48	0.66	0.86
3	0.94	0.65	0.22	0.36	0.52	0.49
4	0.64	1	0.78	0.64	0.88	0.28
5	0.52	0.14	0.98	0.84	0.22	0.89
6	1.96	0.15	1.54	0.93	0.63	0.31
7	1.52	0.39	1.88	0.74	0.72	0.47
8	1.6	0.57	1.76	0.29	0.65	1
9	0.38	0.54	1.16	0.68	0.34	0.48
10	0.96	0.12	1.66	0.82	0.84	0.61
11	0.66	0.23	0.56	1	0.87	0.8
12	1.46	0.3	1.36	0.8	0.67	0.83
13	1.94	0.9	0.94	0.25	0.96	0.87
14	1.18	0.84	0.6	0.12	1	0.23
15	1.9	0.95	1.52	0.94	0.4	0.14
16	1.8	0.74	1.84	0.59	0.83	0.24
17	1.76	0.34	1.92	0.98	0.25	0.12
18	2	0.81	1.02	0.9	0.56	0.92
19	1.12	0.56	1.42	0.17	0.99	0.7
20	1.2	0.5	2	0.7	0.64	0.13
21	1.44	0.85	0.26	0.42	0.15	0.56
22	0.74	0.55	0.2	0.11	0.2	0.15
23	0.82	0.59	0.84	0.51	0.42	0.46
24	0.72	0.69	1.06	0.21	0.91	0.63
25	0.9	0.17	1.32	0.6	0.98	0.35
26	0.22	0.16	1.68	0.24	0.89	0.79
27	1.32	0.42	1.72	0.46	0.85	0.55
28	0.36	0.52	1.3	0.1	0.46	0.54
29	1.06	0.6	1.04	0.41	0.39	0.1
30	1.34	0.43	1.78	0.81	0.32	0.21
31	0.44	0.64	0.66	0.96	0.29	0.71
32	1.84	0.94	1.1	0.71	0.48	0.17
33	0.32	0.38	0.8	0.61	0.41	0.36
34	0.84	0.89	0.44	0.85	0.37	0.29
35	0.34	0.19	0.48	0.45	0.28	0.99
36	1.42	0.78	0.24	0.35	0.82	0.73
37	1	0.76	0.96	0.4	0.51	0.34
38	1.02	0.75	1.12	0.32	0.23	0.9
39	1.16	0.71	0.88	0.16	0.93	0.33
40	0.76	0.32	0.36	0.49	0.94	0.76
41	0.28	0.7	0.28	0.22	0.5	0.72
42	1.36	0.36	0.54	0.37	0.45	0.74
43	1.68	0.41	1.94	0.95	0.97	0.95
44	1.62	0.48	1.34	0.52	0.75	0.97
45	1.1	0.24	0.72	0.89	0.43	0.6
46	0.98	0.11	0.58	0.44	0.47	0.44
47	0.56	0.79	1.48	0.34	0.31	0.43

48	1.26	0.31	1.26	0.53	0.27	0.81
49	0.54	0.8	0.46	0.99	0.58	0.67
50	0.88	0.63	1.8	0.56	0.74	0.85
51	1.28	0.91	0.74	0.13	0.19	0.64
52	0.78	0.73	1.4	0.18	0.62	0.3
53	0.24	0.29	1.96	0.26	0.9	0.52
54	1.3	0.53	1.98	0.77	0.12	0.96
55	1.08	0.21	0.92	0.23	0.3	0.84
56	0.8	0.4	1.24	0.87	0.38	0.38
57	1.54	0.72	1.44	0.79	0.17	0.5
58	1.22	0.99	1.58	0.88	0.8	0.98
59	1.56	0.22	1.9	0.75	0.16	0.65
60	0.46	0.13	1.5	0.47	0.95	0.51
61	1.5	0.82	0.9	0.86	0.35	0.26
62	1.58	0.83	0.5	0.5	0.86	0.4
63	0.68	0.45	1.86	0.15	0.76	0.91
64	1.98	0.68	1.7	0.63	0.44	0.58
65	1.38	0.67	1.2	0.67	0.7	0.37
66	0.4	0.98	1.64	0.73	0.36	0.77
67	0.86	0.44	0.82	0.27	0.57	0.78
68	1.72	0.49	0.64	0.55	0.68	0.62
69	0.26	0.1	0.38	0.91	0.33	0.57
70	1.78	0.58	1.6	0.97	0.21	0.45
71	1.74	0.25	1	0.65	0.53	0.93
72	1.24	0.46	1.46	0.33	0.78	0.94
73	0.5	0.51	0.68	0.62	0.79	0.11
74	0.42	0.33	1.62	0.72	0.54	0.2
75	0.92	0.77	1.18	0.19	0.69	0.88
76	1.82	0.96	1.38	0.39	0.92	0.41
77	1.86	0.86	0.42	0.66	0.24	0.27
78	0.3	0.92	0.4	0.3	0.14	0.32
79	0.58	0.62	0.3	0.54	0.6	0.16
80	1.64	0.18	0.52	0.28	0.61	0.53
81	1.92	0.27	0.86	0.31	0.13	0.69
82	0.62	0.35	0.76	0.78	0.77	0.68
83	1.04	0.28	1.82	0.38	0.18	0.39
84	0.48	0.97	1.08	0.57	0.55	0.59
85	1.4	0.37	0.7	0.69	0.81	0.18
86	1.66	0.61	0.32	0.92	0.49	0.19
87	1.7	0.26	1.74	0.83	0.73	0.75
88	1.88	0.87	1.14	0.76	0.26	0.82
89	1.48	0.47	0.34	0.2	0.59	0.22
90	0.2	0.66	1.56	0.43	0.11	0.25
91	1.14	0.93	1.22	0.58	0.1	0.42

Table 12. Latin Hypercube Sampling Input Sets Sample. Note that not all input variables are shown in this sample. Each variable is divided into 90 equal intervals (giving 91 endpoints).

APPENDIX D. C5.0 DESCRIPTION

A C5.0 model works by splitting the sample based on the variable that provides the maximum expected reduction in information entropy. Each subsample defined by the first split is then split again, usually based on a different variable, and the process repeats until the subsamples cannot be split any further. Finally, the lowest level splits are re-examined, and those that do not contribute significantly to the value of the model are removed or pruned.

C5.0 can produce two kinds of models. A decision tree is a straightforward description of the splits found by the algorithm. Each terminal or "leaf" node describes a particular subset of the training data, and each case in the training data belongs to exactly one terminal node in the tree. In other words, exactly one prediction is possible for any particular data record presented to a decision tree.

In contrast, a ruleset is a set of rules that tries to make predictions for individual records. Rulesets are derived from decision trees, and in a way represent a simplified or distilled version of the information found in the decision tree. Rulesets can often retain most of the important information from a full decision tree, but with a less complex model. Because of the way rulesets work, they do not have the same properties as decision trees. The most important difference is that with a ruleset, more than one rule may apply for any particular record, or no rules at all may apply. If multiple rules apply, each rule gets a weighted "vote" based on the confidence associated with that rule, and the final prediction is decided by combining the weighted votes of all the rules that apply to the record in question. If no rule applies, a default prediction is assigned to the record.

C5.0 models are quite robust in the presence of problems such as missing data and large numbers of variables. They usually do not require long training times to estimate. In addition, C5.0 models tend to be easier to understand than some other model types, since the rules derived from the model have a very straightforward interpretation.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX E. C5.0 RULESETS

Rules for 1:

Rule #1 for 1:

if Submerge Time (T) <= 0.92
and Initial SSN Report > 1.3
and Strike > 1.77
and DDG > 0.142
and CG > 0.067
then -> 1 (56, 0.845)

Rule #2 for 1:

if Initial SSN Report > 1.18
and Strike > 1.59
and DDG > 0.21
and Mean time between updates (tu) > 0.68
then -> 1 (31, 0.758)

Rules for 2:

Rule #1 for 2:

if Submerge Time (T) <= 1.2
and Initial SSN Report > 0.84
and Strike > 1.56
and DDG > 0.045
and DDG <= 0.09
and Missile Speed (v) <= 313.333
then -> 2 (34, 0.944)

Rule #2 for 2:

if Initial SSN Report > 1.26
and Strike > 0.78
and Strike <= 1.05
and DDG > 0.125
and Mean time between updates (tu) > 0.57
then -> 2 (21, 0.913)

Rule #3 for 2:

if Submerge Time (T) > 0.94
and Initial SSN Report > 0.44

and Strike > 2.52
and DDG > 0.14
and DDG <= 0.22
then -> 2 (59, 0.885)

Rule #4 for 2:

if Submerge Time (T) <= 0.94
and Initial SSN Report > 0.44
and Initial SSN Report <= 0.84
and Strike > 1.44
and DDG > 0.08
then -> 2 (78, 0.863)

Rule #5 for 2:

if Submerge Time (T) <= 0.68
and Initial SSN Report > 0.84
and Strike > 0.45
and Strike <= 1.05
and DDG > 0.125
then -> 2 (37, 0.846)

Rule #6 for 2:

if Submerge Time (T) > 0.94
and Initial SSN Report > 0.44
and Strike > 1.44
and DDG > 0.22
then -> 2 (79, 0.84)

Rule #7 for 2:

if Submerge Time (T) > 0.68
and Initial SSN Report > 1.56
and Strike > 0.63
and DDG > 0.125
and Mean time between updates (tu) <= 0.57
then -> 2 (76, 0.833)

Rule #8 for 2:

if Initial SSN Report > 1.26
and Strike <= 1.05
and DDG > 0.18
and Missile Speed (v) <= 346.667
and Mean time between updates (tu) > 0.57

then -> 2 (20, 0.773)

Rule #9 for 2:

if Initial SSN Report > 0.84

then -> 2 (1272, 0.547)

<u>**Rules for 3:**</u>

Rule #1 for 3:

if Submerge Time (T) > 1.2

and Initial SSN Report <= 1.4

and DDG <= 0.09

then -> 3 (166, 0.982)

Rule #2 for 3:

if Strike <= 1.56

and DDG <= 0.09

and Mean time between updates (tu) <= 0.6

then -> 3 (158, 0.981)

Rule #3 for 3:

if DDG <= 0.045

then -> 3 (197, 0.98)

Rule #4 for 3:

if Submerge Time (T) > 1.2

and Strike <= 1.95

and DDG <= 0.09

then -> 3 (161, 0.975)

Rule #5 for 3:

if Initial SSN Report <= 1.36

and DDG <= 0.09

and Mean time between updates (tu) <= 0.6

then -> 3 (213, 0.953)

Rule #6 for 3:

if Submerge Time (T) > 1.52

and Strike <= 1.59

and DDG <= 0.158

then -> 3 (142, 0.938)

Rule #7 for 3:

if Submerge Time (T) > 0.92

and Initial SSN Report <= 1.18

and Mean time between updates (tu) <= 0.24

then -> 3 (114, 0.922)

Rule #8 for 3:

if Strike <= 1.05

then -> 3 (572, 0.838)

Rule #9 for 3:

if Initial SSN Report <= 0.84

then -> 3 (730, 0.821)

Default : -> 3

LIST OF REFERENCES

1. Walter Perry, Robert Button, Jerome Bracken, Thomas Sullivan, Jonathan Mitchell, *"Measures of Effectiveness for the Information-Age Navy: The Effects of Network-Centric Operations on Combat Outcomes"*, MR-1449-NAVY, RAND, 2002.

2. McKay, M. D., Conover, W. J., and Beckman, R. J., *"A Comparison of Three Methods for Selecting Values of Inputs Variables in the Analysis of Output from a Computer Code"*, Technometrics, Vol. 21, 1979.

3. Laurene V. Fausett, *"Fundamentals of Neural Networks, Architectures, Algorithms, and Applications"*, Prentice Hall, 1994.

4. Averill M. Law and W. David Kelton, *"Simulation Modeling and Analysis"*, McGraw-Hill, 3rd Ed, 2000.

5. T. Cover and J. Thomas, *"Elements of Information Theory"*, Wiley, 1991.

6. Walter Perry, *"Information Quality and the COFM Measure"*, RAND, Nov 2001.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, VA

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, CA

3. Professor Thomas W. Lucas
   Department of Operations Research
   Naval Postgraduate School
   Monterey, CA

4. Dr. Walter L. Perry
   RAND
   Arlington, VA

5. Tom Sullivan
   RAND
   Arlington, VA

6. Thomas Choinski
   CNO Strategics Studies Group
   Naval War College
   Newport, RI

7. Dr. Michael I. Bell
   Naval Research Laboratory
   Washington, DC

8. Dr. Larry Wiener
   OCNO Strategic Planning Office (N6C)
   Arlington, VA

9. Professor Dan Boger
   Department of Operations Research
   Naval Postgraduate School
   Monterey, CA

10. Pee Eng Yau
    Defence Science and Technology Agency
    1 Depot Rd., #15-06
    Defence Technology Tower A
    Singapore 109679

11. Loke Yim Peng/Cheryl Tang
Appraisal and Promotion Branch
1 Depot Rd, #03-01J
Defence Technology Tower A
Singapore 109679