2019-09

MODELING SUBMARINE ANTI-SHIPPING WARFARE IN THE SOUTH AND EAST CHINA SEAS

McDonough, Bryan P.

Monterey, CA; Naval Postgraduate School

http://hdl.handle.net/10945/63480

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

MODELING SUBMARINE ANTI-SHIPPING WARFARE IN THE SOUTH AND EAST CHINA SEAS

by

Bryan P. McDonough

September 2019

Thesis Advisor:	Hong Zhou
Second Reader:	Bard K. Mansager

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

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

1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** September 2019	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE MODELING SUBMARINE ANTI-SHIPPING WARFARE IN THE SOUTH AND EAST CHINA SEAS	**5. FUNDING NUMBERS**
6. AUTHOR(S) Bryan P. McDonough	
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) N/A	**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** A

13. ABSTRACT (maximum 200 words)

With a strong nuclear arsenal, rapidly expanding Navy, and increasing economic influence, China is quickly turning into a peer adversary that matches the United States' military and economic strength. Strategies must be developed and analyzed that effectively curb Chinese aggression. Jeffrey E. Kline and Wayne P. Hughes, both professors at the Naval Postgraduate School and retired Navy Captains, developed the "War at Sea Strategy," which relies heavily on U.S. submarines creating a maritime exclusion zone in the South and East China Seas. In Zachary P. Schwartz's 2013 thesis, "Using Undersea Assets to Establish a Maritime Exclusion Zone in the South and East China Seas," Schwartz developed the submarine anti-shipping engagement model (SASEM) to analyze the feasibility of the "War at Sea Strategy."

This thesis developed a new model to test the viability of SASEM and build upon its conclusions. The new model uses a different methodology that removes many of SASEM's underlying assumptions and allows for more complicated modeling behaviors, such as changing submarine search and movement patterns. By comparing our results to SASEM's, we found that the SASEM methodology was flawed and produced unreliable results. By testing various search patterns, we found that barrier search is superior when the targets move in predictable paths. Additionally, we found the difference between random and grid search to be small but statistically significant.

14. SUBJECT TERMS combat simulation, random search, directed search, South China Sea, East China Sea, Poisson Scan model, anti-submarine warfare	**15. NUMBER OF PAGES** 117
	16. PRICE CODE

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

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

MODELING SUBMARINE ANTI-SHIPPING WARFARE IN THE SOUTH AND EAST CHINA SEAS

Bryan P. McDonough
Lieutenant, United States Navy
BSEE, University of Washington, 2011

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN APPLIED MATHEMATICS

from the

NAVAL POSTGRADUATE SCHOOL
September 2019

Approved by: Hong Zhou
Advisor

Bard K. Mansager
Second Reader

Wei Kang
Chair, Department of Applied Mathematics

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

With a strong nuclear arsenal, rapidly expanding Navy, and increasing economic influence, China is quickly turning into a peer adversary that matches the United States' military and economic strength. Strategies must be developed and analyzed that effectively curb Chinese aggression. Jeffrey E. Kline and Wayne P. Hughes, both professors at the Naval Postgraduate School and retired Navy Captains, developed the "War at Sea Strategy," which relies heavily on U.S. submarines creating a maritime exclusion zone in the South and East China Seas. In Zachary P. Schwartz's 2013 thesis, "Using Undersea Assets to Establish a Maritime Exclusion Zone in the South and East China Seas," Schwartz developed the submarine anti-shipping engagement model (SASEM) to analyze the feasibility of the "War at Sea Strategy."

This thesis developed a new model to test the viability of SASEM and build upon its conclusions. The new model uses a different methodology that removes many of SASEM's underlying assumptions and allows for more complicated modeling behaviors, such as changing submarine search and movement patterns. By comparing our results to SASEM's, we found that the SASEM methodology was flawed and produced unreliable results. By testing various search patterns, we found that barrier search is superior when the targets move in predictable paths. Additionally, we found the difference between random and grid search to be small but statistically significant.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

ANOVA	analysis of variance
BSP	Blue-Survival-Percent
CDF	cumulative distribution function
ECS	East China Sea
LSM	least squares mean
MOEs	measures of effectiveness
nm	nautical miles
PDF	probability distribution function
PLA	People’s Liberation Army
PLAN	People’s Liberation Army NAVY
PRC	People’s Republic of China
SASEM	Submarine Anti-shipping Engagement Model
SCS	South China Sea
TTF	Time-to-Finish

THIS PAGE INTENTIONALLY LEFT BLANK

Executive Summary

The purpose of this thesis is twofold. First, it determines the effectiveness of the Submarine Anti-shipping Engagement Model (SASEM) [1] in predicting the capability of U.S. submarines in establishing a maritime exclusion zone. Second, it expands upon SASEM and removes a number of its assumptions in order to better estimate the feasibility of the War at Sea Strategy in curbing Chinese aggression [2].

The War at Sea Strategy, developed in 2012 by retired Navy Captains L. Kline and W. Hughes, was developed in response to growing tensions between East Asian countries regarding disputed ownership of islands and land masses in the South China Sea (SCS) and East China Sea (ECS). In the event of open hostilities between the PRC and the U.S. or one of its allies, the strategy suggests avoiding a costly and possibly unwinnable ground war with the PRC by relying extensively on U.S. submarines. By using U.S. submarines to establish a maritime exclusion in the South China Sea (SCS) and East China Sea (ECS), Chinese merchant trade would be disrupted severely enough that the PRC would yield to U.S. demands without the need for total warfare.

The SASEM was developed in 2013 to model the effectiveness and feasibility of the War at Sea Strategy. Statistical analysis was performed to determine the most significant factors that could affect the outcome of the War at Sea Strategy, and Lanchester empirical differential equations were used to simulate the outcome of the strategy. The model predicted that, given a set of ideal initial conditions, the strategy would require 88 to 165 days to be successful and result in minimal U.S. losses.

Today, territorial disputes in the SCS and ECS continue to escalate as the PRC continues to develop man-made islands in order to expand their territorial claims. Additionally, relationships are worsened by growing trade and economic disputes between the U.S. and the PRC. Combined, these factors mean that the U.S. military must now, more than ever, be prepared to intervene to stop or prevent Chinese aggression. Therefore, another look at the War at Sea Strategy is warranted, with a focus on examining the techniques and outcomes of the SASEM. Specifically, this thesis seeks the answers to the following questions:

- How effective was the SASEM at predicting the outcomes of a submarine centric

anti-shipping campaign?
- Can the results of the SASEM be verified using an independent methodology?
- How does changing the SASEM assumptions concerning unit level search and engagement behavior affect the predicted results?
- Does improving upon the assumptions and methodology of SASEM make the War at Sea Strategy more or less feasible?

To answer the previous questions, a new simulation model is used and compared to SASEM. Unlike SASEM, which relies on differential equations to track different force levels, the new model tracks the course, speed, and position of all units involved and uses their predefined search and engagement behavior to determine their actions. The approach is more computationally expensive than the SASEM approach and therefore does not allow for analysis over a wide range of parameter values. However, since each unit is tracked separately and independently, it allows for the analysis of more dynamic force level behavior. For instance, this approach allows for simultaneous engagements, differing force level search patterns, and differing unit level tactical priorities. Therefore, this thesis focuses primarily on the recommended parameters set forth by SASEM, and explores how changes to force behavior affect the outcome. For each set of behavioral settings, the simulation was performed 500-1200 times. The results of these experiments are as follows:

The SASEM results are unrealistic. Decisions made within the SASEM model caused the model's reported time-to-completion to be excessively large. Additionally, the model's reported U.S. attrition was excessively small.

Our model produced a time-to-completion that was one-to-two orders of magnitude smaller than SASEM's model. This increases the feasibility of the War at Sea Strategy. However, our model's U.S. attrition was also larger, which may make the strategy too costly. Though our model does not show the strategy to be infeasible, it also does not offer evidence in support of the strategy. More research is required to determine acceptable U.S. losses and potential methods for minimizing them.

Concerning search techniques, when the PRC force movements are predictable and well-defined, a barrier search outperformed any other search method. Otherwise, placing the U.S. submarines in a grid was slightly preferable to allowing the submarines to move randomly. However, the difference was slight, you using random search in place of grid search in a combat simulation would not result in a serious loss of fidelity.

References

[1] Z. P. Schwartz, “Using undersea assets to establish a maritime exclusion zone in the South and East China Seas,” M.S thesis, Op. Res. Dept., Monterey, CA, USA, 2013.

[2] J. Kline and W. Hughes, “Between peace and war: A war at sea strategy,” *Naval War College Review*, vol. 65, no. 4, pp. 35-41, 2012

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgments

First and foremost I would like to thank my academic advisor Hong Zhou for helping me through this process. Additionally, thank you to my second reader, Bard Mansager, my department chair, Wei Kang, and my program officers, CDR Paula Travis and CDR Clay Herring. Many professors helped me with the education I needed to write this thesis. Special thanks go to Professor James Eagle for teaching me search and detection theory, LCDR Michelle Isenhour for strongly encouraging me to learn JMP, and Professor Ralucca Gera for teaching me the process of writing a research paper.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

Relations between the United States and the PRC have been strained ever since the PRC rose to power in 1949. Differences in ideological views and practices have always been a source of tension, and more recently, territorial water disputes and economic competition have furthered the divide. This tension has resulted in indirect conflict between the two nations, as was the case with the Korean War. Since the mid-1990s, the People's Liberation Army (PLA) has steadily increased its military budget, which rose by about 10 percent from 2017 to 2018 alone [1]. Therefore, it is more important than ever that the United States be prepared for conflict with the PRC. It is necessary not only to identify comprehensive strategies but also to test these strategies using both quantitative and qualitative methods.

This chapter explores the relations between the United States and the PRC in the South China Sea (SCS) and East China Sea (ECS). It describes the area's geographical characteristics and the roles of regional U.S. allies in this U.S.-PRC dynamic. Next, it will provide an brief overview of the War at Sea Strategy, which was developed as a possible U.S. response to Chinese aggression. Finally it will detail the basics of Submarine Anti-shipping Engagement Model (SASEM) [2], a model developed to test the feasibility of the War at Sea Strategy [3], and the primary focus of this thesis.

1.1 The First Island Chain

Though not an official part of either the PRC or U.S. military policy, unofficial doctrine divides Chinese maritime defense into three perimeters defined by a series of islands and archipelagoes. Of chief concern for the War at Sea Strategy, SASEM, and our thesis is the first island chain inside the South China Sea (SCS) and East China Sea (ECS). These are the waters that start at the east coast of Asia and end at an imaginary line which starts at the southern tip of Japan, runs south through the Ryukyu Islands and down the east coast of Taiwan, passes west across the northern coast of Borneo, and reconnects to East Asia at the southern tip of the Malay Peninsula [4]. An approximation of this region is highlighted in Figure 1.1, along with its area and perimeter.

Establishing sea control within the first island chain comes with various geographical and military difficulties. Geographically, access to the SCS and ECS is restricted by the island chain, creating choke points such as at the Luzon Strait. Militarily, much of the SCS and ECS are within operating range of the formidable PRC diesel submarine fleet, which numbered 59 as of 2015. Besides being armed with torpedoes, over half of the PRC's submarines also carry cruise missiles that pose a long-range threat to surface ships [5]. Additionally, surface ships are also vulnerable to the PRC's DF-21, a shore-based anti-ship ballistic missile [6]. Overall, the operating environment is dangerous for any asset operating within its waters, but surface ships would be extremely vulnerable.

1.1.1 U.S.-PRC Relations

U.S.-PRC relations are becoming steadily worse due to a combination of economic conflict and geographical disputes. Economically, disagreements over trade practices led to a trade war, which began to escalate in early 2018 [7], threatening the health of both nations' economies. Geographically, China continues to build and occupy islands in the SCS and uses them as justification for claiming ownership over large swaths of water for navigational, economic, and military use. This practice has led to territorial disputes with many other actors in the region, including Vietnam, Brunei, Malaysia, the Philippines, and Japan [5].

The territorial tension in the region is of particular note due to the large amount of allies the United States has in and around the SCS and ECS. Specifically, the United States has defense treaties with the Philippines, Japan, Vietnam, Malaysia, Brunei, South Korea, and Singapore, including mutual defense treaties with the Philippines, Japan, and South Korea [8]. This means that a conflict between any of these nations and China could be a triggering event that necessitates U.S. intervention, further emphasizing the need for the United States to have well analyzed strategies in place to deter and, if necessary, defeat Chinese aggression.

1.2 SASEM

SASEM was developed in 2012 by Zachary Schwartz [2] in order to evaluate the feasibility of the War at Sea Strategy. The approach uses differential equations and stochastic inputs to determine how quickly United States and PRC force levels attrite. Since one goal of our thesis is to validate the results of SASEM, this section will give an overview of its major components. All information regarding SASEM is taken from Schwartz's master's thesis "Using Undersea Assets to Establish a Maritime Exclusion Zone in the South and East China Seas," and a more complete discussion on SASEM can be found there.

1.2.1 Approach

SASEM tracks the force levels for five types of units: blue submarines, red submarines, red surface ships, red logistic ships, and red merchants. At its core, SASEM uses Lanchester differential equations to determine the expected casualty rate for each unit type. These casualty rates are then used to determine the engagement type that occurs next by determining the time-to-next-encounter for each possible engagement.

If X(t) represents the number of blue submarines at time t, and Y(t) represents the number of red submarines at time t, then the casualty rate of red submarines would be calculated by,

$$\frac{dY}{dt} = A \cdot X(t) \cdot Y(t), \tag{1.1}$$

where A is a constant determined by assumptions concerning unit level exchange ratios, sensor performances, and search techniques. Using this process, a casualty rate is calculated for each unit type.

Using the casualty rates, the time-to-next-encounter for each type of engagement is calculated. This is a random variable with a probability curve determined by the casualty rate and assumptions concerning unit level search behavior. The lowest time-to-next-encounter determines the next simulated engagement. For instance, if the times until a blue submarine encountered a red submarine, red warship, and red merchant were randomly determined to be one second, 3 seconds, and five seconds, respectively, then the next engagement to occur would be blue submarine versus red submarine, and the simulation time would be advanced by one second. One of the two forces involved in that engagement loses a unit. The losing side is determined by a uniformly distributed random variable, which is then compared to an assumed exchange ratio. Finally, the simulation time is advanced by a preset prosecution time. The prosecution time is used to reflect that an engagement does not occur instantaneously.

Table 1.1. SASEM Recommended Parameters

PARAMETER	VALUE
Red Break Point	>0.7
Merchant Prosecution Time	<3hr45min
Total Blue Subs	>24
Merchant Search Radius	>6 nm
Submarine Exchange Ratio	3:1

These are the SASEM recommended values for maximizing the likelihood of a blue force victory.

The above process is repeated until one of the forces reaches a preset breakpoint level, at which point the simulation ends and one of the sides is determined to be the victor. For Schwarz's analysis, the simulation was repeated several times using different combinations of parameter values. Statistical analysis was performed to determine which parameters had the most impact on the simulation outcome and which parameter values maximized the likelihood of a blue victory.

1.2.2 Parameters of Interest

SASEM contains 21 adjustable parameters. However, only the most significant parameters will be discussed here. Some of these parameters have recommended values to increase the probability of a blue victory. Since these values heavily influenced the choices made for our model's parameters, they are summarized in Table 1.1.

Force Levels

These parameters include Total Blue Submarines, Initial Blue Submarines, Initial Red Submarines, Initial Red Merchants, Initial Red Warships, and Initial Red Logistics Ships. Since SASEM allows for blue force reinforcements at preset times within the simulation, Total Blue Submarines and Initial Blue Submarines are not necessarily the same.

Exchange Ratios

These parameters include the Submarine Exchange Ratio and Warship Exchange Ratio. They represent how many of the respective unit type is lost per blue submarine lost, and control the probability of a unit being destroyed in an encounter.

Search Radii

These parameters include the Merchant Submarine Search Radius, Warship Search Radius, Logistics Ship Search Radius, and Merchant Search Radius. This is the range at which a blue submarine detects a red submarine, surface warship, logistics ship, and merchant, respectively. Note that since SASEM assumes simultaneous detection, none of the red force units has its own search radii.

Merchant Breakpoint

After the percentage of red merchants remaining is reduced below this breakpoint level, the blue force wins the simulation.

Merchant Prosecution Time

This is the amount of time added to the simulation every time a blue submarine engages a red merchant.

1.2.3 Assumptions

This section highlights the most important assumptions used by SASEM.

Random Search

All units move according to random search theory. For more on random search theory, see Section 2.5.

Perfect Sensors

If the range from a red unit to a blue submarine is equal to or less than its respective search radius, then the blue submarine will detect it. If a unit is outside its respective search radius, then the blue submarine will not detect it. This is often called a “Cookie Cutter” sensor.

Every Engagement Results in Exactly One Casualty

For every engagement, exactly one unit must be destroyed. The cases where both or neither units are destroyed are not considered.

1.3 Problem Statement

Any military conflict with the PRC will invariably require an advanced force of U.S. submarines to neutralize threats both in and around the SCS and ECS. SASEM's use of the War at Sea Strategy carries this idea further, tasking U.S. submarines with the responsibility of disrupting PRC merchant activity to the point where the PRC is unwilling to engage in further conflict. Since such a large significance is placed on the ability of the U.S. submarines to find and destroy PRC merchants in a timely manner, a thorough analysis should be performed to test the feasibility of the strategy. Though SASEM and its use of differential equations is a good launching point for this analysis, it is unable to model the more dynamic behaviors and outcomes that can occur from havinig multiple units engaging each other simultaneously. A new model is required that builds upon the SASEM model to improve the accuracy of the results.

1.4 Objectives

This thesis will use a new combat simulation model based on the significant parameters obtained from SASEM. The new model will be used to attempt to validate and improve upon SASEM's results. Additionally, it will be used to re-evaluate the feasibility of the War at Sea Strategy. The following questions will be examined:

- How effective was SASEM at predicting the outcomes of a submarine focused anti-shipping campaign?
- Can the results of SASEM be verified using an independent methodology?
- How does changing SASEM assumptions concerning unit level search and engagement behavior affect the predicted results?
- Does improving upon the assumptions and methodology of SASEM make the War at Sea Strategy more or less feasible?

1.5 Methodology

In order to test the validity of SASEM, we have created a new model using a different approach from SASEM. Initially, we will force our model to use most of the assumptions made by SASEM in order to ensure they produce similar results. Then, we will further the analysis of the War at Sea Strategy by removing some of the assumptions made by SASEM.

This section gives a brief overview of our model's construction and the analysis performed on its results. For more information on our model, see Chapter 2, and for more information on our experiment design and data analysis, see Chapters 3 and 4, respectively.

1.5.1 Model and Experiment

Our model does not use equations to predict force levels and unit interactions. Instead, it tracks the position and velocities of individual units. This removes the abstraction caused by using differential equations, and it allows units to interact naturally as they move within detection range of each other. Initially, many of the parameters and behaviors of our simulation will be set to best imitate the restraints and behaviors of SASEM. After comparing these results to SASEM's, we will then begin modifying unit behavior outside the bounds of SASEM's assumptions to determine how that affects the simulation's results.

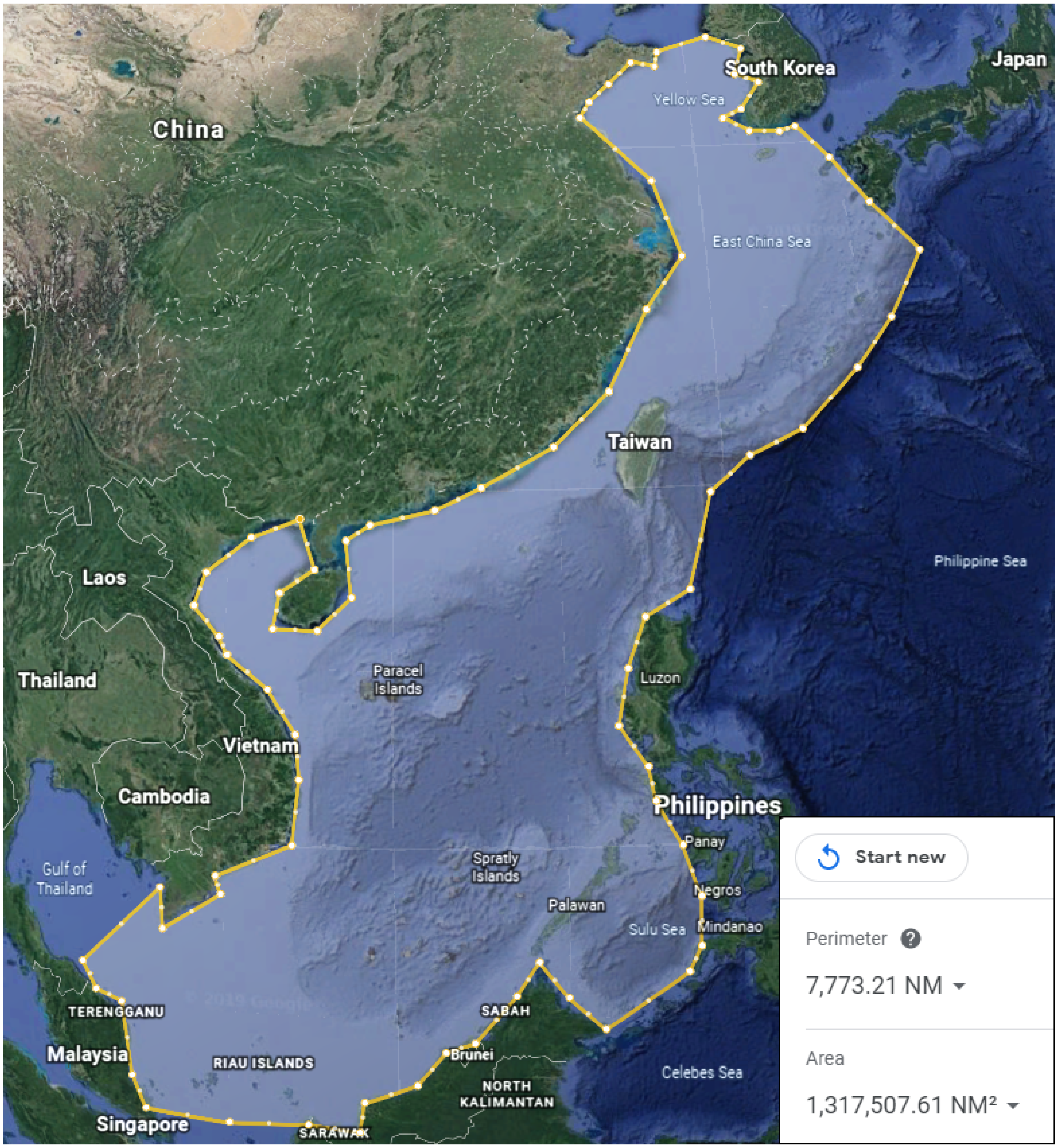
Our aim is to make three determinations. First, we wish to determine if SASEM is a valid model for simulating the War at Sea Strategy. Second, given the improvements we make upon SASEM, we wish to re-evaluate the feasibility of using submarines to create a maritime exclusion zone in the SCS and ECS. Finally, regardless of the feasibility of the strategy, we wish to determine which search technique gives the U.S. force the most successful outcome. The level of U.S. success will be judged based on two measures of effectiveness (MOEs). First, the amount of time required to complete the simulation, which we call Time-to-Finish (TTF). Second, the amount of U.S. submarines that are able to survive the simulation, which we call Blue-Survival-Percent (BSP).

1.5.2 Data Analysis

To perform data analysis, we will compile the variable input parameters, TTF, and BSP for each iteration of the combat model. The variable inputs will consist of U.S. force behavior, PRC force behavior, and the initial number of merchants. The statistical analysis software JMP will be used to perform multi-factor analysis of variance (ANOVA) with the variable inputs representing the different factors. Using this analysis, we will be able to identify statistically significant changes in TTF and BSP for the different levels of force behavior and merchants.

1.6 Thesis Organization

Chapter 1 of this thesis provides background information and defines the problem. Chapter 2 is a full description of our combat model. Chapter 3 describes the design our experiments and the specific parameter values used in each experiment. Chapter 4 presents the results of the experiments and uses statistical analysis to analyze those results. Finally, Chapter 5 contains our conclusions and recommendations.



Perimeter

7,773.21 NM

Area

1,317,507.61 NM^2

Figure 1.1. Waters within the First Island Chain

The highlighted portion of this map roughly represents the waters defined by the First Island Chain. The area of the highlighted portion will be used to determine the area of our combat model. Source: [9]

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2:
Model Description

This chapter will explain the methodology and features of our combat simulation model. The model was developed solely using MATLAB and its code can be referenced in Appendix C. First is an overview of the combat model, including its assumptions, algorithm, and parameters. Then the basics of random search theory are explained, along with how this model replicates these random search conditions. This last portion is important since accurately creating random search patterns is essential for comparing this model with SASEM. For ease of labeling and to conform with standard modeling convention, the U.S. force is designated as the blue force and the PRC force is designated as the red force.

2.1 Building upon SASEM

The results obtained using SASEM provide a useful set of key parameters and values that can be used as a starting point for the new model. However, in order to validate and improve upon SASEM, a different approach to combat modeling will be used. The new model will still track the force levels of U.S. submarines, PRC submarines, PRC surface warships, and PRC merchants as the simulation progresses in time. Whereas SASEM advanced time using random increments calculated with stochastic equations, this model will advance time in set increments. At each new time increment, the position of each individual unit will be tracked in the X-Y plane, and encounters will occur organically as opposing forces move within detection range of each other. Randomness will be introduced through unit starting positions, unit course changes, and battle outcomes. Once a force level is reduced to a breakpoint value, the simulation is ended. Since the model tracks the course and speed of every unit, various behaviors can be simulated beyond random search. Behaviors of interest will include random search from random starting locations, random search from uniformly distributed starting locations, and a barrier search technique.

2.2 Assumptions

Though this model removes several assumptions required by SASEM, other key assumptions still remain. First, a list of the assumptions key to this model will be explained. Then, the assumptions from SASEM that are no longer applicable will be explained.

2.2.1 New Model Assumptions

The following is a list of major assumptions used in this model.

Perfect Sensors

In order to improve computational efficiency, all sensors are assumed to have 100 percent detection capability up to a predefined range, after which they will have no detection capability. No sensor degradation is modeled based on range or environment. The predefined sensor range will vary depending on the pairing of detecting unit and detectable unit. For instance, the detection range for a blue submarine is different against red submarines and red surface ships, and the range a blue submarine can detect a red submarine is different from the range a red submarine can detect a blue submarine.

Unit Types

Though an operating environment will contain many types of assets from multiple organizations, this model assumes only four types of units exist: blue submarines, red submarines, red surface warships, and red merchants. Additionally, though in reality much variability exists within each of those classes, this model assumes all units of the same type are identical. Of note, SASEM also includes red logistics ships. However, since SASEM did not allow for logistics ships to detect or kill blue submarines, their contribution to the model is assumed to be insignificant.

Combatants

Of the unit types discussed above, only submarines and surface warships are considered combatants capable of attacking enemy forces.

Every Detection Event Results in an Engagement

The scenario where a combatant unit ignores all detected enemies is not considered. In real life, a vessel may avoid combat due to possible ongoing repairs, tactical or strategic priorities, or armament inventory. Temporary damage and armament levels are not tracked in this model. If a combatant gains multiple enemy detection at the same time, then only one of the enemies will be pursued for engagement. If multiple types of units are detected at the same time, then the type of enemy pursued depends on preset behavior programming.

Prosecution Times

When a combatant detects an enemy unit, a predefined prosecution time must pass before the combatant can attempt to kill the enemy. During this prosecution time, the combatant can neither detect nor engage other units. The prosecution time is specific to the detected enemy, with submarines having a longer prosecution time than surface ships.

2.2.2 SASEM Improvements

The following is a list of assumptions that existed in SASEM that do not exist in this model.

Random Search

With SASEM, the predicted time until next detection is derived using the probability distribution expected from random blue force movement. Though our model can simulate random search as one of its behavioral options, it is not restricted to it.

Every Engagement Results in Exactly One Vessel Being Destroyed

SASEM attrites forces by removing exactly one unit per engagement. The unit removed is determined randomly using a predefined exchange ratio. For our model, both units have a chance to survive an engagement based on predefined survival rates. Likewise, both units have a chance to be destroyed in an engagement. The survival rates used are determined from the unit types involved in the engagement and which side shoots first.

Simultaneous Detections

Our model does not require simultaneous detections. If units have different detection ranges relative to each other, or if one unit is already engaged with a different unit, then a one-way detection will occur. This can result in a unit surprising an enemy, which will increase its likelihood of killing the enemy.

2.3 Algorithm Description

At its core, our model tracks the positions, speeds, and courses of all units, advances them throughout an X-Y plane at a specified time increment, and causes units to engage enemies that are within their detection range. Below is a more detailed step-by-step description of the algorithm. The full algorithm can be viewed in Appendix C.

1. Set force levels, movement behaviors, and combat behaviors based on specified parameters.
2. Start a master loop that determines how many times the simulation will be performed.
3. Determine unit starting positions, courses, and speeds based on specified parameters.
4. Start the time loop which will advance the simulation at a specified time increment (typically six minutes per iteration).
5. Calculate the distance between actively searching units and their enemies.
6. For each searching unit, determine which of its enemies are within its detection range. Begin tracking the detected unit with the highest priority level.
7. Determine which units have been tracking an enemy for the required prosecution time, and have them attack their enemy.
8. Use randomly generated numbers to determine if units survive combat based on the specifics of the encounter and the specified parameters.
9. End the simulation if any force is reduced to a breakpoint level.
10. Determine if any unit changes course or speed based on specified parameters.
11. Advance all units to their new positions based on their current positions, velocities, and the time increment used.
12. Reiterate the time loop until a breakpoint is reached.
13. Save battle results.
14. Reiterate the master loop for the specified number of iterations.

2.4 Model Parameters

Our model has a total of 44 parameters. This includes continuous parameters that can be chosen from a range of values, Boolean parameters that are either TRUE or FALSE, and categorical parameters that can be set to one of several available options. A list and brief description of each parameter is given in Tables 2.1, 2.2, and 2.3 for miscellaneous, movement, and combat related parameters, respectively. Some common abbreviations used

Table 2.1. Miscellaneous Parameters

PARAMETER	TYPE	DESCRIPTION
Breakpoint	N	Merchant breakpoint percentage
Iterations	N	Number of times simulation is performed
Initial-b	N	Initial number of blue submarines
Initial-rs	N	Initial number of red submarines
Initial-rw	N	Initial number of red surf. warships
Initial-rm	N	Initial number of red merchants
Width	N	Width of combat box (x-direction)
Length	N	Length of combat box (y-direction)

This table is a summary of all the miscellaneous model parameters, including their value type. Value types can be (N)umerical, (C)ategorical, or (B)oolean.

throughout the parameter names are as follows:

- b - blue force
- r - red force
- rs - red submarine
- rw - red surface warship
- rm - red merchant

Parameters types are defined as (N)umerical, (C)ategorical, or (B)oolean.

2.4.1 Breakpoint

This value controls when the simulation ends. It can take values between zero and one. Once the percentage of the red merchant fleet remaining falls below the breakpoint level, the simulation will end. It is at this level that it is assumed the PRC will be unwilling to continue combat operations.

2.4.2 Initial Force Levels

These values control the starting number of blue submarines, red submarines, red surface warships, and red merchants. Our model does not include reinforcements of any units, so this is also the maximum number of each unit type for a given simulation.

Table 2.2. Movement and Positioning Parameters

PARAMETER	TYPE	DESCRIPTION
Waterspace	C	Sets the navigational boundaries of blue submarines
Movement-b	C	Controls blue movement behavior
Movement-r	C	Controls red movement behavior
Lambda	C	Rate that randomly moving units change course
Vb-search	N	Blue search speed
Vb-sub	N	Blue attack speed against submarines
Vb-surf	N	Blue attack speed against surf. warships
Vb-m	N	Blue attack speed against merchants
Vrs-search	N	Red submarine search speed
Vrs-sub	N	Red submarine attack speed
Vrw-search	N	Red surf. warship search speed
Vrw-sub	N	Red surf. warship attack speed
Vrm	N	Red merchant cruising speed

This table is a summary of all the movement and positioning related model parameters, including their value type. Value types can be (N)umerical, (C)ategorical, or (B)oolean.

2.4.3 Width and Length

The Width and Length define the boundaries of the combat area. The area is a rectangular box with x-values from zero to Width and y-values from zero to Length. If any unit moves outside these dimensions, it will alter its course to return to the combat area. Since SASEM used an area of 1,350,000 nm^2, our model uses a Width and Length of 900nm x 1500nm. These dimensions result in the same area and geographically approximate the dimensions of the combined SCS and ECS.

2.4.4 Waterspace

This categorical parameter determines blue force navigational boundaries, in addition to those already defined by Length and Width. Options for this categorical parameter include Random, Grid, and Barrier. Each option is discussed in detail below, and pictorial representation of Grid and Barrier can be seen from Figures 2.1 and 2.2, respectively. In these figures, the downward facing blue triangles represent blue submarines, and the bolder black lines represent their navigational boundaries. The light grey lines only correspond to x-axis

Table 2.3. Combat Parameters

PARAMETER	TYPE	DESCRIPTION
Pros-sub	N	Prosecution time against a submarine
Pros-surf	N	Prosecution time against a surface ship
Rec-time	N	Recovery time after attacking or being attacked
Cooperation	B	Minimizes the prosecution times of red units
SASEM	B	Restricts model to cause it to act more like SASEM
Priority-ws	B	Blue prioritizes attacking warships over merchants
b-rs-b	N	Blue prob. of surviving a red submarine, blue shoots first
rs-b-b	N	Red submarine prob. of surviving, blue shoots first
b-rw-b	N	Blue prob. of surviving surf. warship, blue shoots first
rw-b-b	N	surf. warship prob. of surviving, blue shoots first
rm-b-b	N	Merchant prob. of surviving a blue attack
b-rs-rs	N	Blue prob. of surviving a red submarine, red shoots first
rs-b-rs	N	Red submarine prob. of surviving, red shoots first
b-rw-rw	N	Blue prob. of surviving a surf. warship, red shoots first
rw-b-rw	N	surf. warship prob. of surviving, red shoots first
b-rs-s	N	Blue prob. of surviving a red submarine, simultaneous fires
rs-b-s	N	Red submarine prob. of surviving, simultaneous fires
b-rw-s	N	Blue prob. of surviving a surf. warship, simultaneous fires
rw-b-s	N	surf. warship prob. of surviving, simultaneous fires
R-b-rs	N	Range that blue can detect red submarines
R-b-rw	N	Range that blue can detect surf. warships
R-b-rm	N	Range that blue can detect merchants
R-rs-b	N	Range that red submarines can detect blue
R-rw-b	N	Range that surf. warships can detect blue

This table is a summary of all the combat related model parameters, including their value type. Value types can be (N)umerical, (C)ategorical, or (B)oolean.

and y-axis tick marks. Also note that the density of red units is not representative of the experiments presented later in this thesis.

Random

All blue forces will start at a random point within the operational area. No further navigational boundaries exist.

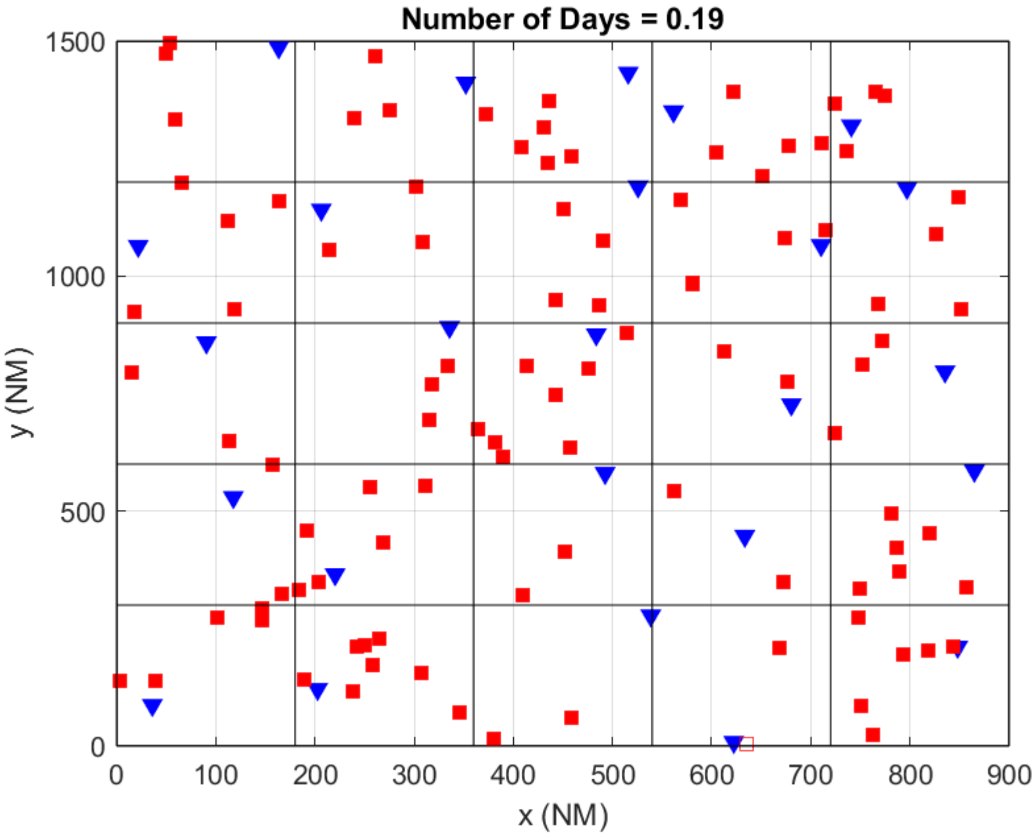


Figure 2.1. Grid Waterspace Starting Positions

This is an example of the blue submarines' starting positions when Waterspace = Grid. The blue force is represented by blue downward facing triangles. The red units plotted are not representative of the actual number used in the experiments.

Grid

The operational area will be subdivided into a number of smaller boxes equal to the initial number of blue submarines. Each submarine will be assigned one of the boxes and placed randomly within its assigned box. The dimension and placement of the boxes will be as uniform as possible. If a blue submarine dies, the boxes of neighboring submarines will expand to encompass the waterspace of the dead submarine. Navigationally, if a blue submarine moves outside its assigned box, it will correct its course to return to its own waterspace.

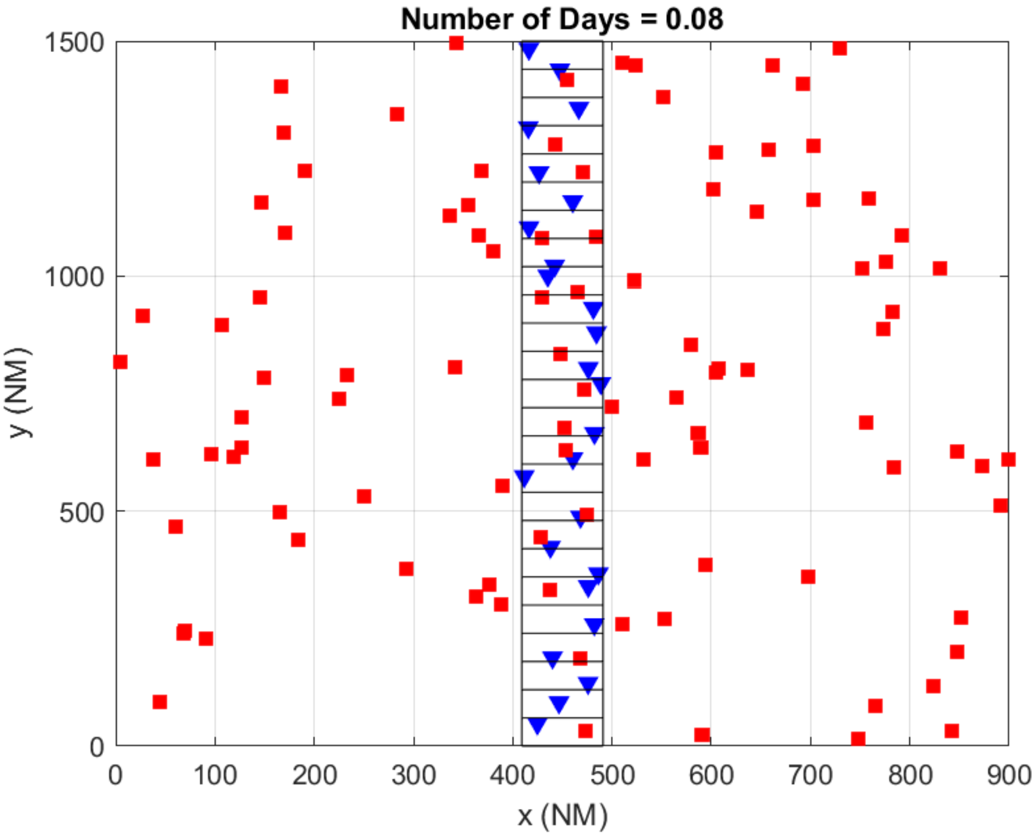


Figure 2.2. Barrier Waterspace Starting Positions

This is an example of the blue submarines' starting positions when Waterspace = Barrier. The blue force is represented by blue downward facing triangles. The red units plotted are not representative of the actual number used in the experiments.

Barrier

Similar to Grid, each submarine will be assigned its own navigational box. However, all the boxes will be placed in one column centered at half the Width of the combat area, creating a barrier of closely positioned blue submarines that is hard to pass through undetected.

2.4.5 Movement

Movement-b and Movement-r are categorical parameters which determine the blue and red force movement behaviors, respectively. Movement-b can take the values of Random and Barrier. Movement-r can take the values of Random and Straight.

Random Movement
This option causes forces to change direction at random times. Unless the unit is engaging an enemy or outside its navigational area, it will change directions with a probability determined by Lambda, where Lambda is the average amount of times a unit will change course per hour.

Straight Movement
This option is available to the red force. Unless engaged with an enemy, each red unit will always move from west to east or east to west. When a unit passes a navigational boundary of the combat area it turns around to proceed in the opposite direction.

Barrier Movement
This option is available to the blue force. Unless engaged with an enemy or outside its navigational area, each blue unit will move up and down between the northern and southern boundaries of its navigational box.

2.4.6 Lambda

See Random Movement under Section 2.4.5.

2.4.7 Unit Speeds

These are the unit travel speeds in Knots. The red merchants have only one speed, but each combatant has a different speed for searching and for attacking. The attack speed is used to close distance with a detected enemy prior to the prosecution time expiring. Of note, if an attacking unit becomes sufficiently close to its target, it will instead attempt to match the course and speed of its target until the prosecution time expires.

2.4.8 Prosecution Times

These numbers represent how much time must pass after a unit detects an enemy before it can begin searching for additional enemies. This is intended to simulate the time required for a unit to determine its enemy's course, speed, and position with enough certainty to make an accurate shot. Since submarines are harder to detect and track, it will typically take longer to prosecute a submarine than a surface ship.

2.4.9 Recovery Time

This number represents the amount of time a unit must take to recover after attacking or being attacked. If the unit is the attacking unit, then Recovery Time is included as part of Prosecution Time. Recovery Time is intended to simulate a unit taking evasive action and going through the process of preparing additional shots after attacking or counter-attacking an enemy. While a unit is in recovery, it may not detect, attack, or counter-attack any enemy units.

2.4.10 Cooperation

This Boolean determines the extend of red force cooperation. When TRUE, if multiple red units engage the same target, the prosecution time for all red units will be reduced to whichever red unit has the minimum prosecution time. This is intended to represent the combined efforts and communications of the red units attempting to determine the blue unit's position. This will allow all red units to fire upon the blue unit at the same time, and if Recovery Time is greater than zero, the blue unit will only get to counter-attack against one of them. Since U.S. submarines do not train to operate in the same water space, there is no option for blue forces to cooperate.

If Cooperation is set to FALSE, multiple red units may still engage the same blue unit, but they will each have an individual prosecution time. In general, this Boolean will be set to TRUE, but may be set to FALSE to cause our model to more closely match the behavior of SASEM.

2.4.11 SASEM Mode

To distinguish the parameter from the model, quotation marks are used for the "SASEM" parameter. "SASEM" is a Boolean which is used to make our model assumptions and behaviors more closely match the model SASEM. When this Boolean is set to TRUE, only one unit is allowed to detect and engage another unit at any given time. This not only means that simultaneous engagements cannot happen at the same time, but it also means that two units cannot detect and engage each other and that multiple units cannot detect and engage the same enemy at the same time. This last stipulation makes the Cooperation Boolean moot. In general, this Boolean will be set to FALSE, but may be set to TRUE to cause our model to more closely match the behavior of SASEM.

2.4.12 Engagement Priority

Priority-ws is a Boolean that, when set to TRUE, will cause blue units to prioritize attacking warships over merchants. The specific priority order would be red submarine, red surface ship, and then red merchant. When set to FALSE, the priority order would be reversed. Of note, an actively searching blue unit cannot decline to engage a red unit if at least one has been detected. These priority orders only apply when multiple types of red units are detected at the same time. Once a blue unit has made the decision to engage a red unit, it will not start looking for a new enemy to attack until the engagement is complete.

2.4.13 Survival Rates

These parameters are probabilities between zero and one and are named in an X-Y-Z format. Their value is the probability that an X unit will survive a Y unit if Z unit fires first. For instance, the value assigned to b-rs-b is the probability that a blue submarine will survive a red submarine if the blue submarine fires first. Simultaneous shots are also considered. For instance, b-rs-s is the probability that a blue submarine will survive a red submarine if they both fire simultaneously. Note that the parameters which represent a unit's probability of surviving a counterattack, such as b-rs-b or rs-b-rs, will only apply if the attacked unit is not in a state of recovery. See Section 2.4.9 for more on recovery.

2.4.14 Detection Ranges

Detection ranges are numerical values and are named in an R-X-Y format. Their values are the maximum ranges in nautical miles (nm) that unit X can detect unit Y. For instance, R-b-rs is the maximum range a blue submarine can detect a red submarine.

2.5 Simulating Random Search

SASEM uses random search theory to stochastically determine the time-to-next-encounter for each possible type of engagement. Since one goal of this thesis is to determine the validity of SASEM, it was important to develop search behavior that mimicked the theoretical random search. In this section, a brief overview of random search theory will be given and it will be shown how our model effectively simulates it.

2.5.1 Random Search Theory

Random search theory predicts the probability that a single searcher will detect a single target over time if the searcher's course is random. For any given time t, the searcher has a detection rate γ(t), which is determined by the searcher's speed (V), the target's speed (U), sensor performance, and the search area (A). If the detection rate is constant, then γ(t) = γ, and the probability that the searcher will detect the target at time t is given by the following probability distribution function (PDF):

$$f(t) = \gamma \cdot e^{-\gamma \cdot t} \tag{2.1}$$

Integrating f(t) gives the probability that the searcher will have found the target by time t, which is given by the following cumulative distribution function (CDF):

$$F(t) = 1 - e^{-\gamma \cdot t} \tag{2.2}$$

Next, we develop a random search model for a simple case where the target is stationary [10] and for a more complex case where the target is moving [11]. For the simple case, the following assumptions are made:

- The searcher is moving randomly with a constant speed V.
- The target is stationary within the search area, and therefore U = 0.
- The searcher has a sensor with 100 percent detection probability up to range R and zero percent detection probability at ranges greater than R.

At any given time during the search, the searcher will have a sweep width of 2R and a forward speed of V. This means over a small period of time Δ, the searcher will cover a swath of area equal to 2RVΔ. The probability of detecting the target over this small period of time is the percentage of the total area which has been searched over Δ, and so the probability of detection is 2RVΔ/A. Dividing by the time component gives the following detection rate:

$$\gamma(t) = 2RV/A \tag{2.3}$$

Note that since R, V, and A are all constant, then γ(t) is also constant and γ(t)=γ. Therefore, by substituting γ into Equation 2.2, the CDF is given by

$$F(t) = 1 - e^{-2RVt/A} \tag{2.4}$$

Next, we wish to further our model by allowing the target to move. For this model, the following assumptions are made:

- The searcher is moving randomly with a constant speed V.
- The target is moving randomly with a constant speed U.
- The searcher has a sensor with 100 percent detection probability up to range R and zero percent detection probability at ranges greater than R.

In this model, the V in Equation 2.4 is replaced with an enhanced average searcher speed ($\tilde{V}$). Though it is not explained here, it can be shown that [11]

$$\tilde{V} = \frac{2}{\pi}(V + U)E(K)/\pi, \tag{2.5}$$

where

$$K = 2\sqrt{UV}/(U + V),$$

and

$$E(K) = \int_0^{2\pi} \sqrt{1 - K^2 \sin^2(\phi)d\phi}$$

$\tilde{V}$ is an enhancement to the searcher's speed due to the speed of the target. For a given V, $\tilde{V}$ will be equal to or greater than V.

2.5.2 Modeling Random Movement

In our model, we control exactly when and if a unit changes course. In order to simulate random search, a unit must change course at random times. In order to accomplish this, we define Lambda as the average number of course changes per hour. For every iteration of the time loop, we generate a random number p for each unit, where p is a uniformly distributed random variable between zero and one. Over the time period Δ, a unit has a probability of changing course equal to Lambda·Δ. Therefore, if p < Lambda·Δ, then the unit will change course. The new course is random and uniformly distributed over the full 2π radians of possible courses.

In order to keep all the different units within the combat area, we will also have a unit change course if it leaves the combat area. This will result in units being temporarily outside the boundaries of the combat area, but is is assumed the effect is insignificant. To ensure the unit rapidly returns to and stays within the combat area, we first set the new course so that it is perpendicular to the violated area boundary. To introduce some randomness, we then adjust the course using a standard normal distribution with a standard deviation of 0.3. For example, if a unit were to exit the search area by crossing the southern boundary, its new course (in radians) would be selected using the probability distribution shown in Figure 2.3.

2.5.3 Validating our Model

To validate our random search model, a simple simulation was performed to determine how long it takes a single searcher to detect a single target. For this simulation, the search area was set at 60 nm x 60 nm, sensor range R was set at 5 nm, and both searcher and target speeds were set at 5 knots. Next, the probability of a unit changing courses (Lambda) needed to be set. The selection of Lambda was not arbitrary. Choosing Lambda to be too large relative to the search area would result in a unit becoming nearly stationary, and choosing λ to be too small relative to the search area would cause most course changes to occur due to boundary violations. For the search area selected, we set Lambda = 0.2.

The simulation was run 500 times and the time of detection was recorded for each iteration. For any given time t, let D(t) be the number of detections that occurred at or before t. By dividing D(t) by the total number of iterations, we obtained the probability of a detection occurring at or before time t. These values represent the CDF of our model.

Finally, to ensure our data approximated a theoretical random search, we used our data to calculate the detection rate that would give a best fit random search CDF. Both the empirical data and the best fit curve are plotted on Figure 2.4. As can be seen, the empirical data closely matches the theoretical curve, giving us confidence that our model provides a suitable tool for evaluating the results of SASEM. The code that was used to generate this test is found in Appendix B.

The only change from this test to our larger model is the frequency of random course

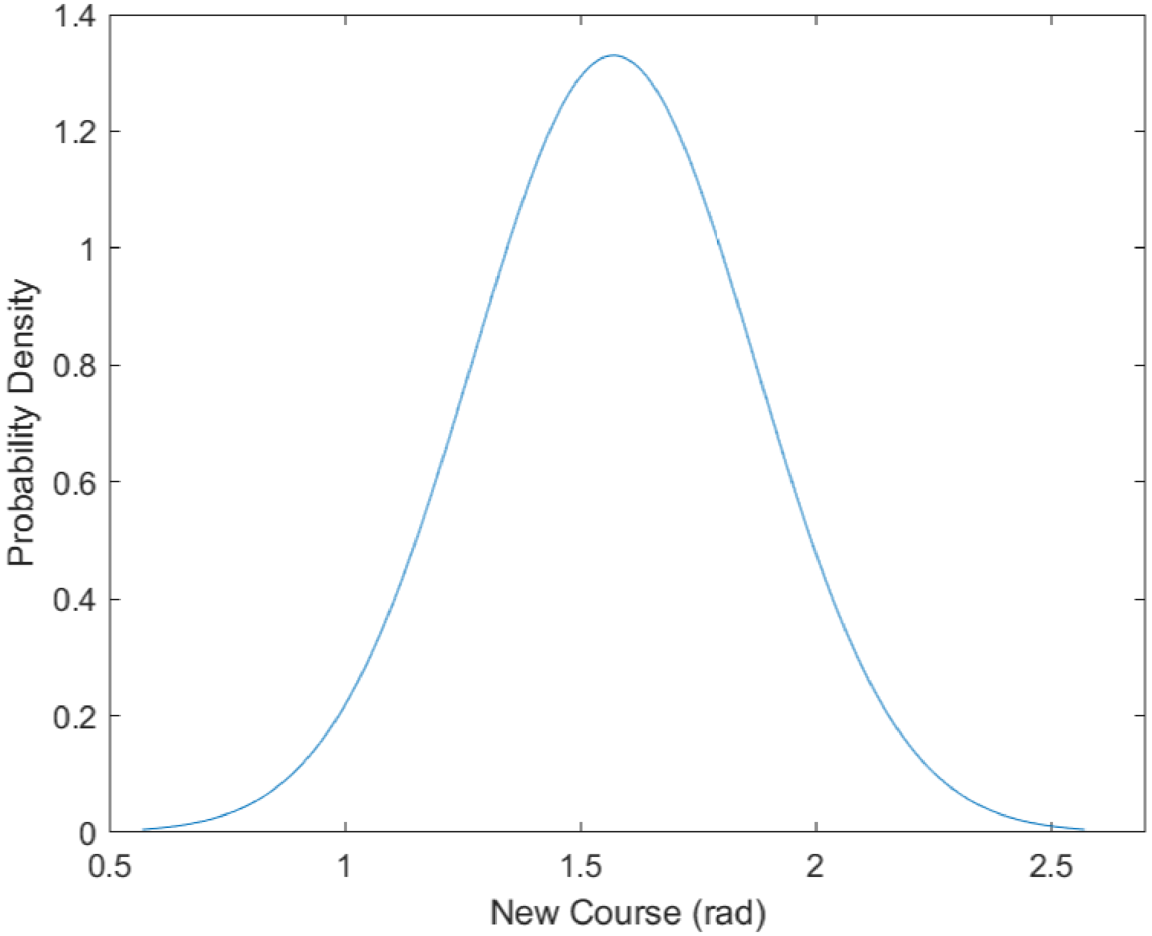


Figure 2.3. New Course Selection PDF

This is the probability distribution for the new course selected if a unit exited the search area to the south. New course is given in radians.

changes. Since our combat model uses an area that is 1500 nm x 900 nm, we needed to decrease the likelihood of random course changes. The dimensions of the combat model are 25 and 15 times larger than this test, so we decreased the course change probability by a factor of 20. Therefore, in the combat model, Lambda = 0.01.

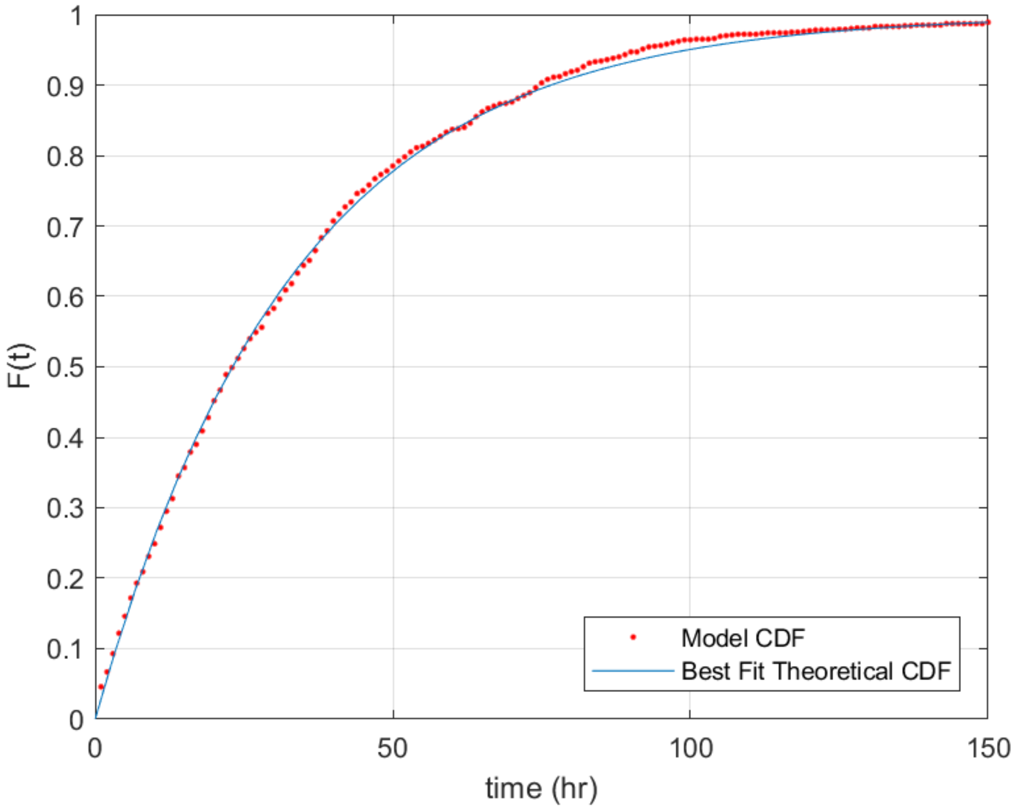


Figure 2.4. Empirical versus Theoretical Random Search CDFs

This is a plot of the empirically generated CDF from our model compared to a best fit theoretical CDF from random search theory.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3:
Experiment Design

Our experiments are designed in order to answer the following three questions:

1. Is SASEM a valid model for simulating the War at Sea Strategy?
2. Given the improvements we have made upon SASEM, is it still feasible to use submarines to create a maritime exclusion zone in the SCS and ECS?
3. Regardless of the feasibility of the strategy, which search technique gives the U.S. force the most successful outcome?

To help answer these questions, we perform four experiments. Experiment 1 tests if our model is suitable for judging the performance of SASEM. To do this, we create behaviors and assumptions within our model that most closely reflect SASEM. We expect that the results of these simulations will be close to SASEM's in both time required and blue force attrition. Experiment 2 then removes many of SASEM's behavioral assumptions. It will allow for multiple engagements and for red force movement, but otherwise it keeps most parameters as unchanged from experiment 1, including random search. Experiment 2 can be viewed as a stepping stone to experiments 3 and 4. By only changing parameters that affect core assumptions, we can analyze how changing from SASEM's assumption set to our assumption set performance.

Next, in experiments 3 and 4, we test the affect of varying red and blue force behaviors. Additionally, we adjust some of the performance parameters to values we believe are more realistic. Experiment 3 will focus on relevant cases with the red force exhibiting random movement. The blue force will be varied between two behaviors: random search and grid search. Experiment 4 will focus on relevant cases with the red force exhibiting predictable, straight movement. The blue force will be varied between three behaviors: random search, grid search, and barrier search. Therefore, in both these experiments, the red force behavior is held constant and blue force behavior is treated as an independent variable. Additionally, the number of initial red merchants will be varied from 250 to 3000. This way, we can also determine if merchant density causes a significant change to any search technique effect that may be present.

By comparing the results of experiment 1 to those of experiment 2, we attempt to help answer the first and second questions above. If the results are significantly different, this will suggest that SASEM may be unreliable due to overly restrictive and unrealistic assumptions. Additionally, since experiment 2 uses more intelligent unit behavior than SASEM, the results should give us more insight into the feasibility of the War at Sea Strategy. By analyzing the results of experiment 3 and experiment 4, we attempt to help answer the second and third questions above. Our goal is to perform enough simulations that a superior search technique can be determined with at least 95 percent confidence for both types of red force behavior.

What follows in this chapter is a description of the parameter values used in each experiment and a justification for those values. First, the parameter values that are constant to all experiments will be discussed. Then, each individual experiment and its unique parameter values will be discussed. For all experiments, results and analysis will be focused on the two MOEs, TTF and BSP. Further information regarding results and analysis is presented in Chapter 4.

3.1 Recommended SASEM Parameters

Since the work of this thesis is building upon SASEM, many of our parameter values adhere to SASEM's recommended parameter value set. These are parameter values that will help minimize both mission accomplishment time and blue force attrition. From [2], these recommended values are as follows:

- Breakpoint > 0.7
- Merchant Prosecution (Pros-surf) <3hr45min
- Blue Submarines (Initial-b) > 24
- Merchant Search Radius (R-b-rm) > 6 nm
- Submarine Exchange Ratio > 3:1

Only Submarine Exchange Ratio does not have a direct counterpart in our model. Instead, this parameter is replaced by engagement survival rates, as is explained in Section 3.2. For experiments 1 and 2, we select values for all these parameters that are near the SASEM recommended value. For experiments 3 and 4, we change R-b-rm and Pros-surf to values we believe are more realistic. Therefore, further discussion on R-b-rm and Pros-surf can be found within the description of each individual experiment, whereas further discussion on Breakpoint, Initial-b, and engagement survival probabilities can be found in Section 3.2.

3.2 Universal Parameters

This section discusses the parameter values that are the same for all experiments. For quick reference, a list of all constant parameter values is provided in Table 3.1.

3.2.1 Breakpoint and Initial-b

These two values were set based on the SASEM recommended values given in Section 3.1.

3.2.2 Survival Probabilities

In Section 3.1, the recommended submarine exchange ratio (>3:1) has no direct counterpart in our model. Instead, our model uses survival probabilities for each type of encounter. In order to ensure we honored this recommended value, we first set the survival probabilities for a blue submarine versus red submarine engagement with the blue submarine shooting first. We assume that a PRC submarine would not be proficient at counter-firing if they were surprised, so we set the chance that the blue submarine survives the encounter, b-rs-b, equal to 0.8. This gives the blue submarine a 20 percent chance of being destroyed in this encounter. To ensure a 3:1 exchange ratio, the red submarine must have a 60 percent chance of being destroyed, which makes the chance of a red submarine surviving this encounter, rs-b-b, equal to 0.4.

All other survival probabilities were set based on these two initial numbers. In general, we set a blue submarine as more lethal than a red submarine, and a red submarine as more lethal than a red surface warship. Additionally, the chance that a unit survives an encounter is greatest when they shoot first. A unit shooting simultaneously with its enemy has the next best survival rate, and a unit surprised by its enemy has the worst survival rate. The exchange ratio for blue submarines to red surface warships ends up being 16:1 (when blue shoots first), which falls with the range of 5-20 set within SASEM.

3.2.3 Length and Width

With Length = 1500 nm and Width = 900 nm, the simulation area A = 900 nm x 1500 nm = 1,350,000 nm^2. This is the same area used in SASEM. A representative waterspace that has approximately this much area can be seen in Figure 1.1. For simplicity, we made our simulation area into a simple rectangle, and although we made the y-direction longer than the x-direction, the perimeter of our area does not match the shaded area of Figure 1.1.

3.2.4 Lambda

Lambda determines how often our units will randomly change course. When we verified the validity of our random search model in Section 2.5, we set Lambda = 0.2. Since that test used a 60 nm x 60 nm test area and our combat model uses a 900 nm x 1500 nm combat area, we reduce Lambda to 0.01 to ensure our units do not turn too often.

3.2.5 Blue Submarine Speeds

SASEM has a blue submarine search speed range of 3 kts - 7 kts. As a submarine moves faster through the water, it creates more ownship noise, making it harder to detect other units. However, we believe that 7 kts does not significantly inhibit a U.S. submarine's ability to search, and so we set Vb-search = 7 kts.

Our model also allows a blue submarine to increase speed when it is engaging an enemy, allowing it to close distance to a detected unit. This is important because, at the time of fire, if the target is too far away it will affect kill accuracy. We limit the attack speed of the blue submarine based on how quiet its target is. We assume that red submarines make less noise than red surface warships, and that red surface warships make less noise than red merchants. Therefore, blue force attack speeds are more limited against red submarines and least limited against red merchants.

Unlike SASEM, our model also includes speeds for all red force units as well. Since a majority of PRC submarines are diesel submarines, which operate at slower speeds than nuclear powered submarines, we set Vrs-search = 3 kts and Vrs-sub = 6 kts, where Vrs-sub is the red submarine attack speed. The values for red surface ships were similarly set based on expected performance. Since it is assumed that red merchants cannot detect blue submarines, they are only given one cruising speed.

3.2.6 Detection Ranges

As explained in Section 3.1, the detection range for a blue submarine detecting a red merchant, R-b-rm, should be greater than 6 nm. All other values were based on this initial value, assuming that a blue submarine has better search capability than a red submarine and that a red submarine has better search capability than a red surface warship.

3.3 Experiment 1

The purpose of this experiment is to restrict the operation of our model to more closely represent SASEM and verify we produce similar results. Even though SASEM uses Lanchester differential equations and our model uses unit level position tracking, the mission completion time and blue force attrition should be comparable if we force our model to use the same assumptions as SASEM. This will ensure our model is operating correctly and that it can be used to evaluate and build upon SASEM's results. All parameter values specific to experiment 1 are shown in Table 3.2.

3.3.1 Setting SASEM Engagement Behavior

To start, we set the Boolean "SASEM" to TRUE. When "SASEM" is TRUE and one unit detects an enemy, all other units are prevented from searching for enemies. This means there will only be one active engagement at a time, and these engagements will only be one-way detections. Since blue submarines have longer detection ranges than both red submarines and red surface warships, and since the blue force searches for enemies prior to the red force in the algorithm's time loop, it will always be true that blue submarines shoot first in engagements. Additionally, in this mode, unit survival rates will not be affected by distance at the time of firing. These effects all result in exchange ratios comparable to SASEM's values, as explained in Section 3.1.

Next, since the SASEM recommended parameters have Merchant Prosecution < 3hr45min (see Section 3.1), we set Pros-surf = 3hr30min. This parameter gives the prosecution time for both red merchants and red surface warships. Prosecution times are primarily meant to represent the time required for a unit to exactly locate its target after initial detection has occurred. Since it is assumed a red submarine will be harder to track and locate than a surface ship, we create a longer prosecution time and set Pros-surf = 5hrs.

Finally, Cooperation and Priority-ws are set to FALSE, and Rec-time is set to zero. Cooperation and Rec-time actually have no effect on the simulation when "SASEM" is set to TRUE, so their values here are meaningless. Setting Priority-ws to FALSE causes blue submarines to prioritize attacking merchants when they detect multiple types of units. We do not expect this to have a significant impact on the simulation, but we prioritize merchants since the value of red combatants is severely diminished when only one engagement can be active at any given time.

3.3.2 Setting Random Search

In order to simulate random search, we set Waterspace, Movement-b, and Movement-r to Random and set the speed of all red force units to zero. Setting the categoricals to random will cause our model to simulate the random search theory used in SASEM (see Section 2.5). Since SASEM does not model red force movement, we must keep the red force stationary. Note that, if we allowed the red force to move, we would still be simulating random search. However, it would be equivalent to a random search process that uses the enhanced search speed $\tilde{V}$, whereas SASEM only uses the blue search speed V to calculate its detection rate.

As a point of comparison between the two types of search theory, we consider the case where blue submarine speed V = 5 kts and red merchant speed U = 20 knots. For this example, we set area A = 3,600 nm^2 and detection radius R = 7 nm. For the simple case where the red merchant is assumed stationary, the detection rate for a single blue submarine $\gamma = 2RV/A = 0.0272$ detections / hr. On the other hand, when we account for the red merchant speed, we get $\tilde{V} = \frac{2}{\pi}(V + U)E(K) = 20.6173$ kts. This results in $\gamma = 2R\tilde{V}/A = 0.0802$ detections / hr. As can be seen, when the enhanced search velocity is calculated the merchant speed becomes dominant to the submarine speed and results in a much higher detection rate. Therefore, at the very least, we expect experiments 2 and 3 will be faster due to the enhancement to search speed caused by red force movement.

3.3.3 Variable Parameters

In this experiment, only Initial-rm is variable. We split the experiment into two trials.. For the first trial, Initial-rm = 1500 merchants, and for the second trial, Initial-rm = 3000 merchants. These values correspond to the allowable SASEM Initial-rm range of 1500-3000. For each trial, we will run the simulation through 300 iterations, for a total of 600 iterations for experiment 1.

3.4 Experiment 2

Experiment 2 is designed to test the viability of the SASEM model. Assuming the results of experiment 1 are similar to SASEM, we can use it as a representation of expected results given SASEM's assumptions apply. In experiment 2, we remove many of those assumptions and observe how significantly that affects the simulations results. The parameters that are held constant between trials and the parameters that vary between trials are discussed separately, below. All parameter values specific to experiment 2 are shown in Table 3.2.

3.4.1 Constant Parameters

For experiment 2, we remove many of the restrictions set by SASEM. Now we allow multiple engagements to occur simultaneously, and we allow multiple units to engage the same target. We also allow red units to work together in order to minimize their prosecution time. These effects are accomplished by setting 'Cooperation' and 'Priority-ws' to TRUE and setting "SASEM" to FALSE. Finally, red force combatants are given search and attack speeds and red force merchants are given a cruising speed.

To add additional realism to the combatants' behavior, Rec-time is set to 30 min. This will cause units to fire with 30 min remaining in their prosecution time, but will not let them begin searching for new targets until their recovery time has ended. It also causes units to enter a 30 min recovery period after being attacked.

All other constant parameters match their value from experiment 1.

3.4.2 Variable Parameters

As in experiment 1, Initial-rm is varied between 1500 merchants and 3000 merchants for a total of two trials. Each trial will be run through 300 iterations, for a total of 600 iterations for experiment 2.

3.5 Experiment 3

Experiment 3 is designed to test if different blue submarine search behaviors create different performance results given that the red force exhibits random movement behavior. The parameters that are held constant between trials and the parameters that vary between trials are discussed separately, below. All parameter values specific to experiment 3 are shown in Table 3.3.

3.5.1 Constant Parameters

As in experiment 2, we wish to allow for more realistic combat behaviors, so we set Cooperation = TRUE, Priority-ws = TRUE, “SASEM” = FALSE, and Rec-time = 30 min. See Section 3.4 for a description of their effects. Additionally, we adjust prosecution times and blue submarine detection ranges to values we believe are more realistic but still conservative. Pros-surf and Pros-sub are each reduced to 2 hrs and 3hrs30min, respectively. R-b-rm, R-b-rw, and R-b-rs are increased to 9 nm, 6nm, and 5 nm, respectively. To create random red force movement, we set Movement-r = Random. Finally, red force speeds are left unchanged from experiment 2.

3.5.2 Variable Parameters

The two variables for this experiment are blue force movement behavior and red merchant density. The blue force is varied between two different behaviors: random search with random distribution (Movement-b = Random, Waterspace = Random) and random search with a uniform distribution (Movement-b = Random, Waterspace = Grid). The number of red merchants, Initial-rm, is varied from 250-3000 merchants at intervals of 250 merchants (250, 500, 750, etc.). Therefore, there are two factors. The blue force behavior factor has two treatment levels and the Initial-rm factor has 12 treatment levels, for a total of 2 x 12 = 24 different trials. Each trial is repeated through 300 iterations, for a total of 7,200 iterations in experiment 3.

3.6 Experiment 4

Experiment 4 is designed to test if different blue submarine search behaviors create different performance results given that the red force exhibits straight, predictable movement behavior. The parameters that are held constant between trials and the parameters that vary between trials are discussed separately, below. All parameter values specific to experiment 4 are shown in Table 3.3.

3.6.1 Constant Parameters

To adjust the red force movement behavior, we set Movement-r = Straight. All other constant parameters and their values are identical to experiment 3.

3.6.2 Variable Parameters

The variables for this experiment are blue force movement behavior and red merchant density. The blue force is varied between three different behaviors: random search with random distribution (Movement-b = Random, Waterspace = Random), random search with a uniform distribution (Movement-b = Random, Waterspace = Grid), and barrier search (Movement-b = Barrier, Waterspace = Barrier). The number of red merchants, Initial-rm, is varied from 250-3000 merchants at intervals of 250 merchants (250, 500, 750, etc.). Therefore, there are two factors. The blue force behavior factor has three treatment levels and the Initial-rm factor has 12 treatment levels, for a total of 3 x 12 = 36 different trials. Each trial is repeated through 300 iterations, for a total of 10,800 iterations in experiment 4.

Table 3.1. Parameters: Universal

PARAMETER	VALUE
Breakpoint	0.7
Initial-b	25
Initial-rs	60
Initial-rw	60
Width	900 nm
Length	1500 nm
Lambda	0.01
Vb-search	7 kts
Vb-sub	10 kts
Vb-surf	13 kts
Vb-m	15 kts
b-rs-b	0.8
rs-b-b	0.4
b-rw-b	0.95
rw-b-b	0.2
rm-b-b	0.1
b-rs-rs	0.5
rs-b-rs	0.7
b-rw-rw	0.6
rw-b-rw	0.6
b-rs-s	0.6
rs-b-s	0.5
b-rw-s	0.7
rw-b-s	0.5
R-rs-b	3 nm
R-rw-b	2 nm

This is a list of parameter values that will remain constant throughout all experiments.

Table 3.2. Parameters: Experiments 1 and 2

PARAMETER	EXP 1	EXP 2
Iterations	500/trial	500/trial
Initial-rm	1500/3000	1500/3000
Waterspace	random	random
Movement-b	random	random
Movement-r	random	random
Pros-sub	5 hrs	5 hrs
Pros-surf	3hrs30min	3hrs30min
Rec-time	0	30 min
Cooperation	FALSE	TRUE
Priority-ws	FALSE	TRUE
SASEM	TRUE	FALSE
Vrs-search	0 kts	3 kts
Vrs-sub	0 kts	6 kts
Vrw-search	0 kts	15 kts
Vrw-sub	0 kts	25 kts
Vrm	0 kts	20 kts
R-b-rs	4 nm	4 nm
R-b-rw	5 nm	5 nm
R-b-rm	7 nm	7 nm

These are the parameter values specific to experiment 1 and 2.

Table 3.3. Parameters: Experiments 3 and 4

PARAMETER	EXP 3	EXP 4
Iterations	300/trial	300/trial
Initial-rm	250-3000	250-3000
Waterspace	Random/Grid	Random/Grid/Barrier
Movement-b	Random	Random/Barrier
Movement-r	Random	Straight
Pros-sub	3hrs30min	3hrs30min
Pros-surf	2 hrs	2 hrs
Rec-time	30 min	30 min
Cooperation	TRUE	TRUE
Priority-ws	TRUE	TRUE
SASEM	FALSE	FALSE
Vrs-search	3 kts	3 kts
Vrs-sub	6 kts	6 kts
Vrw-search	15 kts	15 kts
Vrw-sub	25 kts	25 kts
Vrm	20 kts	20 kts
R-b-rs	5 nm	5 nm
R-b-rw	6 nm	6 nm
R-b-rm	9 nm	9 nm

These are the parameter values specific to experiments 3 and 4.

CHAPTER 4: Results and Analysis

This chapter analyzes the results of the experiments described in Chapter 3. The results will be presented in four different sections. First, the results of experiment 1 will be compared to the expected SASEM results to verify the validity of our model. Second, the results of experiments 1 and 2 will be compared to determine the reliability of the SASEM model. Third, the results of experiments 3 and 4 will be independently analyzed to determine the effects of varying blue force behavior and red force merchant density. Forth, the TTF data will be compared to the BSP data to determine if there is a correlation.

For experiments 1 and 2, no special analysis technique will be performed. These are merely surface level checks to help identify any major inconsistencies or defects among the two models. For experiments 3 and 4, the JMP software will be utilized to perform statistical analysis. Primarily, ANOVA will be performed to identify any statistically significant effects due to blue force behavior, red merchant density, or the interaction between the two. JMP will also be used to compare TTF to BSP. For this comparison, we use a linear regression model to determine if there is a statistically significant correlation between the two MOEs.

Though each of the following sections will explore the results in more detail, a summary of mean TTFs and BSPs is given in Appendix A for all experiments.

4.1 Experiment 1 versus SASEM

Here we compare our model's results to SASEM's results to ensure compatibility. In Schwartz's thesis [2], he reports an estimated TTF of between 88-165 days and an estimated BSP of 89%, assuming the recommended parameter values are used. Since the recommended parameters do not include an Initial_rm (initial number of red merchants) value, we ran our simulation once at the low end of SASEM's merchant range (1500 merchants) and once at the high end of SASEM's merchant range (3000 merchants).

The TTF versus Initial_rm and BSP versus Initial_rm are plotted in Figures 4.1 and 4.2, respectively. The 95% confidence intervals and means for TTF and BSP are as follows:

- Initial_rm = 1500: 78.07-78.47 (78.27) days, 88.3-89.8 (89.1) percent
- Initial_rm = 3000: 150.38-150.77 (150.58) days, 88.2-89.7 (89.0) percent

For both values of Initial_rm, the mean BSP nearly matches the value predicted by SASEM. The range of mean TTF, 78.27-150.58 days, is close to, but slightly under, the values predicted by SASEM. We can contribute two factors to this discrepancy. First, though our modeling of random search behavior is close to the predicted results (see Section 2.5), some deviation may exist. Second, and more importantly, the SASEM recommended values are not specific values but instead limits on the parameter values. Therefore, we do not know the specific SASEM parameters used to create the 88-165 days estimated range of TTF. Given these uncertainties, we believe the results of experiment 1 show that our model is able to replicate the conditions created using SASEM.

Before proceeding to experiment 2, it is worth noting an unusual behavior produced by SASEM conditions. Even though the TTF approximately doubled when the number of red merchants was double, the BSP remained the same. In fact, an ANOVA analysis of the BSP factor shows no statistically significant difference in BSP between the two levels of Initial_rm. One would expect that a longer mission time would result in more blue force casualties. This behavior results due to a discrepancy in SASEM. When Initial_rm is doubled, more engagements are required to reach the merchant Breakpoint level, giving more opportunities for a blue submarine to be destroyed by a red combatant. However, the higher merchant density reduces the probability that a blue submarine will encounter a red combatant for each engagement. Since only one engagement is allowed to occur at any given time, the two effects offset each other, resulting in the same BSP. To see how allowing multiple engagements affects TTF and BSP, proceed to Section 4.2.

4.2 Experiment 2 versus Experiment 1

Here we compare the results of experiment 2 to the results of experiment 1. The major difference between the two experiments is that experiment 2 allows for multiple simultaneous engagements and also allows for more than two units to participate in single engagements. The results for experiment 1 are presented in Section 4.1, above. Like in experiment 1, experiment 2 contains one trial with Initial_rm = 1500 merchants and one trial with Initial_rm = 3000 merchants. The TTF versus Initial_rm and BSP versus Initial_rm are

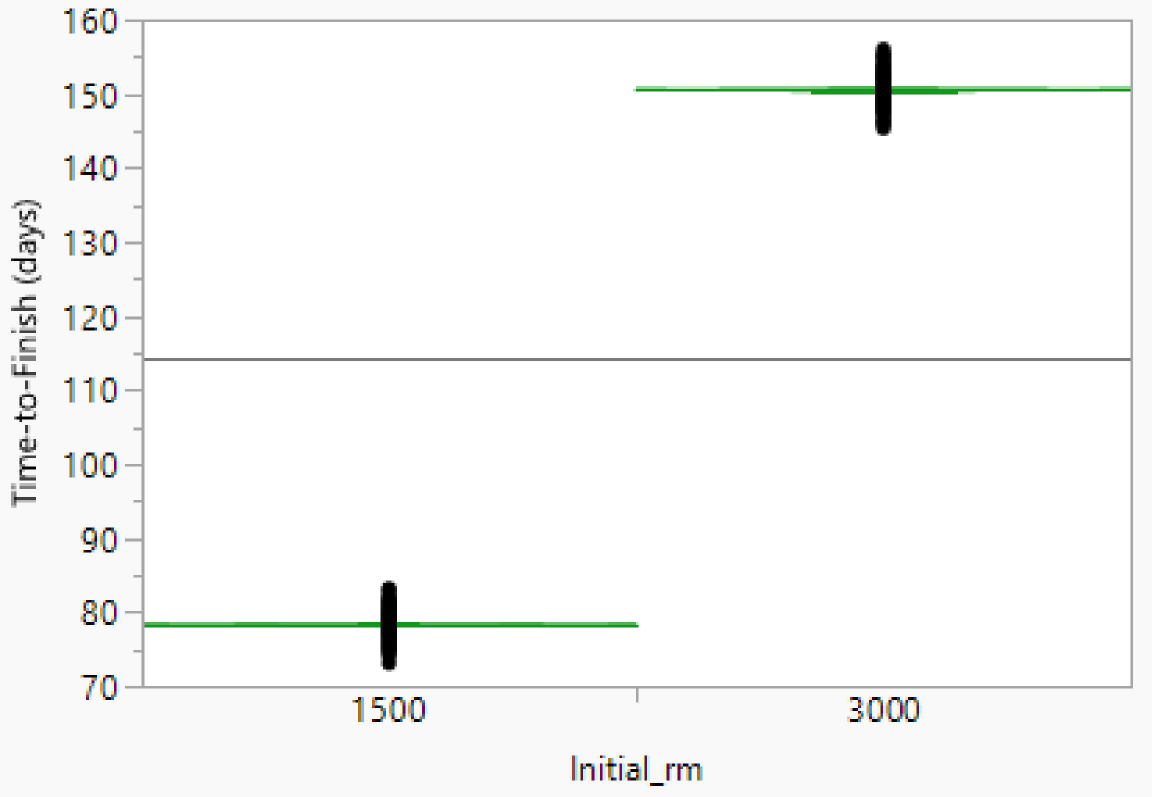


Figure 4.1. Experiment 1: TTF versus Initial_rm

This is a plot of TTF versus Initial_rm for experiment 1. The green lines represent the mean value for each level of Initial_rm. The grey line represents the overall mean.

plotted in Figures 4.3 and 4.4, respectively. The 95% confidence intervals and means for TTF and BSP are as follows:

- Initial_rm = 1500: 11.72-12.22 (11.97) days, 68.3-71.0 (69.6) percent
- Initial_rm = 3000: 19.24-19.75 (19.49) days, 56.4-59.2 (57.8) percent

There are several differences between the results of experiments 1 and 2. Some worth pointing out are as follows:

- TTF is significantly less in experiment 2.
- BSP is significantly less in experiment 2.
- In experiment 2, BSP appears to have an inverse relationship with TTF, whereas in experiment 1 it was constant.

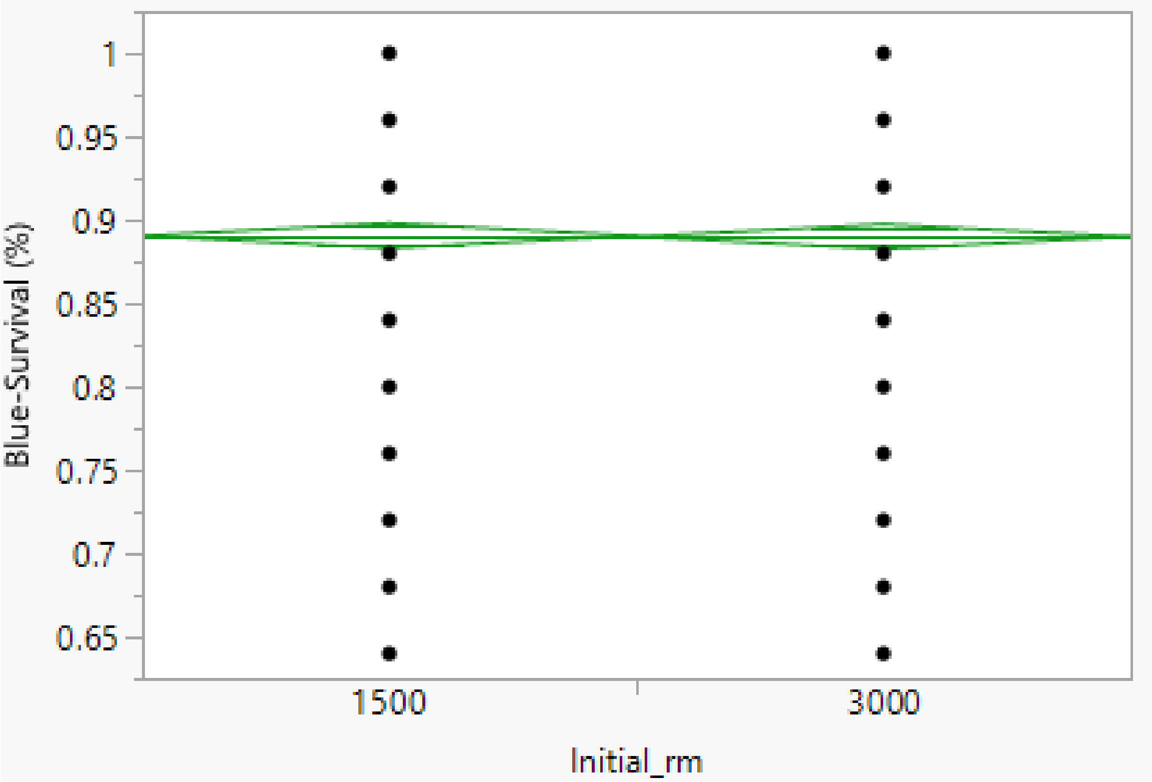


Figure 4.2. Experiment 1: BSP versus Initial_rm

This is a plot of BSP versus Initial_rm for experiment 1. The green diamonds represent the mean value and 95% confidence interval for each level of Initial_rm.

All these differences can be explained by a flaw in the SASEM approach. SASEM does not allow for multiple engagements simultaneously. If engagements were resolved instantaneously, this could have been an acceptable assumption for the model. However, when combined with the inserted prosecution time for each encounter, it made the model unrealistic. The purpose of the prosecution time was to simulate two units being unavailable during the course of an engagement, but since no other unit could act until the engagement ended, it resulted in all units being unavailable during the course of an engagement. Put another way, every time an engagement would occur, the simulation clock would be pushed forward by the prosecution time without any actual combat simulation occurring.

The net effect results in a TTF that depends mostly on the prosecution time and the amount of engagements required to reduce the red merchant force to its Breakpoint value. Take the case where merchant prosecution time (Pros-surf) is 3hrs30min, Initial_rm is 1500

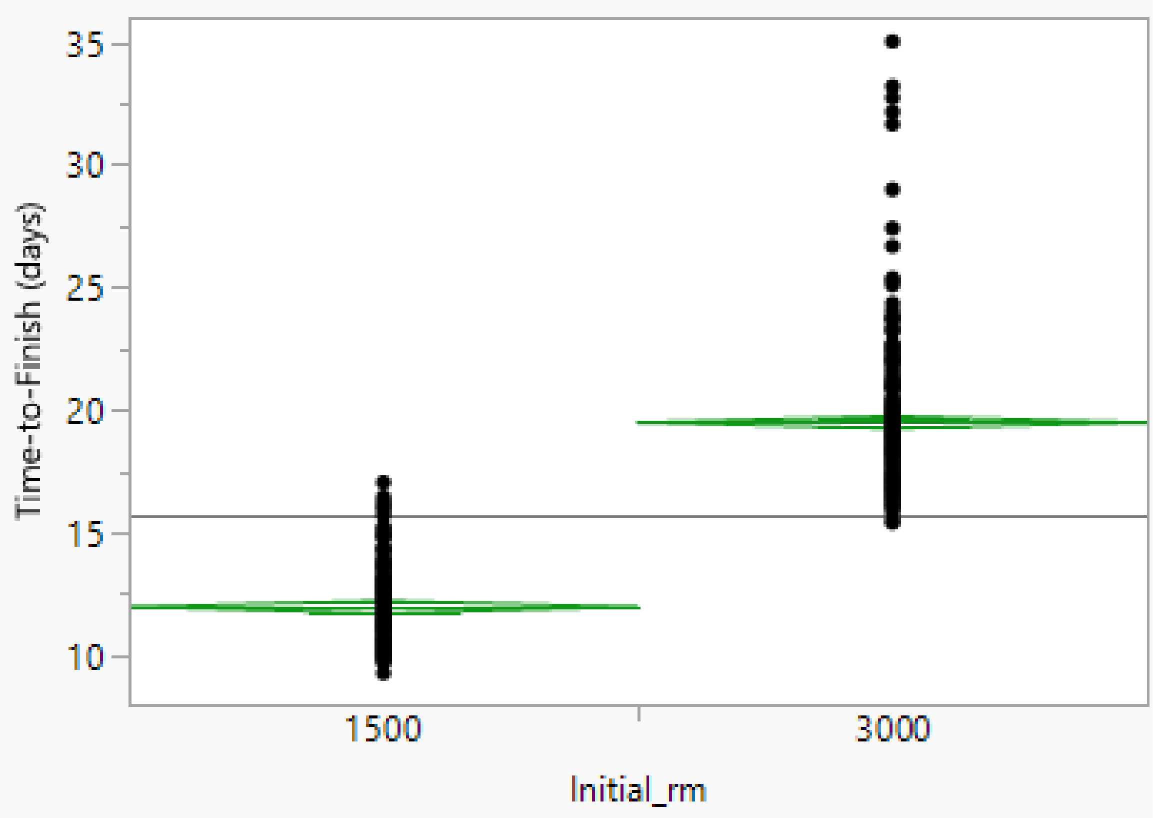


Figure 4.3. Experiment 2: TTF versus Initial_rm

This is a plot of TTF versus Initial_rm for experiment 2. The green lines represent the mean value for each level of Initial_rm. The grey line represents the overall mean.

merchants, and Breakpoint = 70%. In this scenario, the number of merchants that must be destroyed is (1-0.7)*1500 = 450 merchants. Therefore, the total amount of prosecution time that passes while engaging merchants is 3.5*450 = 1,575 hrs = 65.63 days. For these parameters in experiment 1, the mean TTF was 78.27 days. That's only about 13 days the simulation was not paused for merchant engagements. For most of that 13 days, the simulation would be paused for combatant engagements, and a very small portion of it would represent search time between engagements.

As was discussed in Section 4.1, the prosecution time dynamic also negatively affected BSP in SASEM. With SASEM conditions, the BSP was independent of TTF. In experiment 2, we can see the BSP declines as TTF increases. The difference in BSP between different levels

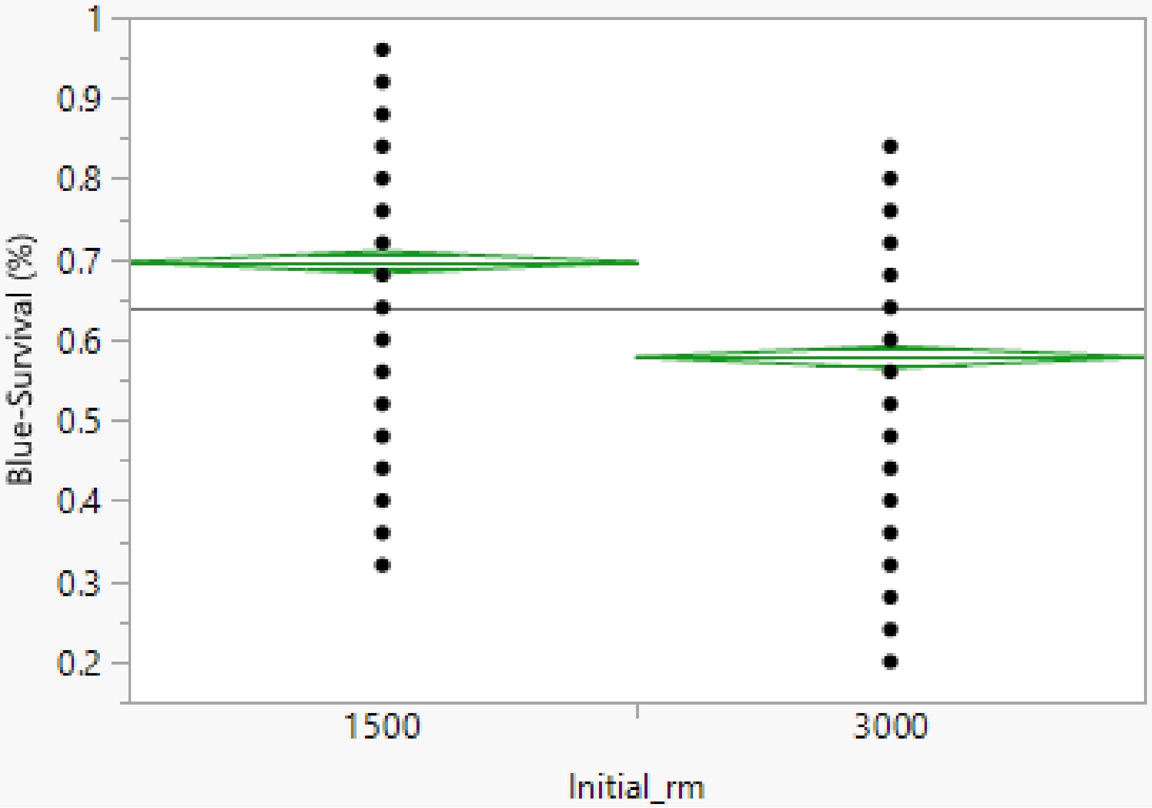
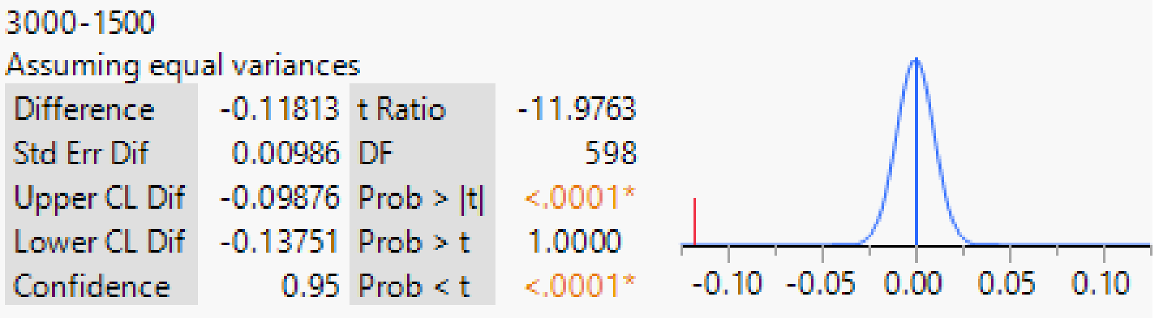


Figure 4.4. Experiment 2: BSP versus Initial_rm

This is a plot of BSP versus Initial_rm for experiment 2. The green diamonds represent the mean value and 95% confidence interval for each level of Initial_rm.

of Initial_rm is visually statistically significant in Figure 4.4, since the 95% confidence intervals do not overlap. This fact can be further validated by the t-test data presented in Figure 4.5, where it can be seen that the statistical significance of the difference in means is greater than 99%.

In addition to BSP changing with TTF, we can also observe that it is much lower in experiment 2. This result is not surprising. With SASEM conditions, the relatively high density of red merchants provided a shielding effect for the blue submarines. As long as a blue submarine could find a red merchant before a red combatant found a blue submarine, then the blue submarines remained safe. When simultaneous engagements are allowed, red combatants can attack blue submarines even while the blue submarines are attacking red merchants. This also allows the red combatants to fire first in many of the engagements,



3000-1500

Assuming equal variances

Difference	-0.11813	t Ratio	-11.9763
Std Err Dif	0.00986	DF	598
Upper CL Dif	-0.09876	Prob > |t|	<.0001*
Lower CL Dif	-0.13751	Prob > t	1.0000
Confidence	0.95	Prob < t	<.0001*

Figure 4.5. Experiment 2: BSP t-test

This is a t-test comparing mean BSP values at different levels of Initial_rm. The data shows that the difference in BSP mean values at Initial_rm = 1500 versus Initial = 3000 is statistically significant with greater than 99% confidence.

which reduces the Red-to-Blue exchange ratio. The exchange ratios were not tracked for any of the experiments described in Chapter 3. However, 200 additional simulations were conducted in order to obtain a rough estimate. In the first 100 simulations, red surface warships were removed, and the resulting mean (red submarine):(blue submarine) exchange ratio was 6:5. In the second 100 simulations, red submarines were removed, and the resulting mean (red surface ship):(blue submarine) exchange ratio was approximately 5:1. As a reminder, with SASEM mode enabled, the exchange ratio against red submarines and red surface ships was 3:1 and 16:1, respectively.

Overall, allowing multiple, simultaneous engagements reduced TTF and BSP to such a significant degree that SASEM cannot be considered a reliable model for either making combat predictions or determining key parameter values. It should be noted that the reduction in TTF occurred despite a significantly greater merchant survival percent in experiment 2. In the original SASEM model, a detected merchant was always destroyed. In experiment 1, a merchant had a 10% chance of survival if detected. In experiment 2, the combatants' accuracy was allowed to degrade with distance, which resulted in a merchant having a 47% mean chance of survival if detected. It should also be noted that SASEM's results would have been more useful if the model removed the prosecution time and allowed engagements to occur simultaneously.

Since the recommended SASEM values cannot be considered reliable, it is hard to make a determination concerning the feasibility of the War at Sea Strategy by restricting the simulation to those values. The significant drop in TTF certainly makes time less of a restraint, increasing the strategies feasibility. However, the large blue force attrition rate makes the strategy more problematic. Of course, some of the additional time gained in the simulation could be sacrificed to allow for more cautious blue force behavior, which would help manage the amount of blue force casualties. Additional information is required to determine how realistic it is for blue submarines to avoid red combatants while they prosecute red merchants, so no determination will be made concerning the War at Sea Strategy using these results. Regardless, we will still analyze the results of experiments 3 and 4 to determine how changing red and blue force behaviors affects TTF and BSP.

4.3 Experiment 3 and Experiment 4 Results

This section analyzes the results of experiments 3 and 4. Each experiment has two factors: blue force behavior (Waterspace) and Initial_rm. The key focus of this analysis is determining how changing blue force behavior affects TTF. Additionally, we determine if the effect on TTF due to blue behavior is dependent on Initial_rm.

4.3.1 Experiment 3 - Random red force movement

For every trial in this experiment, the red force units moved randomly, as described in Section 2.5. Initial_rm was divided into 12 treatment levels ranging from 250-3000 merchants in 250 merchant increments. Waterspace was divided into two treatment levels: random and grid. Additional information on the experimental setup is given in section 3.5.

Two-way ANOVA was used to analyze the significance of the factors as well as their interaction. First, the least squares mean (LSM) plots for TTF versus Initial_rm and TTF versus Waterspace are given in Figures 4.6 and 4.7, respectively. As can be seen, TTF increases steadily and significantly as Initial_rm increases. TTF is also higher for random search behavior versus grid behavior, though only slightly. Regardless, both factors are shown to be statistically significant in the ANOVA model, as can be seen in Figure 4.8.

Also from Figure 4.8, note that the interaction Waterspace*Initial_rm is not significant. Therefore, even though changing Waterspace type has a statistically significant effect on TTF, the size of that effect will not change for different values of Initial_rm. This can also be seen in the interaction plots presented in Figure 4.9. Note that in both plots the slopes for the different effect levels are the same. If an interaction effect was present, you would expect different slopes at different factor levels.

Overall, while both Initial_rm and Waterspace had statistically significant effects on TTF, only Initial_rm was practically significant. The effect size of Waterspace was less than a fifth of a day for this experiment. You can also note that the Initial_rm F-ratio reported in Figure 4.8 is about 25 times larger than the Waterspace F-ratio, which means most of the model's variability is explained by Initial_rm. The small effect size of grid versus random behavior is an important take-away. Actual U.S. submarines would be more likely to operate in a grid to minimize the chances of friendly forces colliding with or attacking each other. However, random behavior is easier to model mathematically. Therefore, we can see from this experiment that, when creating models, random search locations can be used as an approximate substitute for a uniform grid of units.

4.3.2 Experiment 4 - Straight red force movement

For every trial in this experiment, the red force units moved in straight lines either directly west or directly east, as described in Section 2.5. Initial_rm was divided into 12 treatment levels ranging from 250-3000 merchants in 250 merchant increments. Waterspace was divided into three treatment levels: random, grid, and barrier. Additional information on the experimental setup is given in section 3.6.

Two-way ANOVA was used to analyze the significance of the factors as well as their interaction. First, the LSM plots for TTF versus Initial_rm and TTF versus Waterspace are given in Figures 4.10 and 4.11, respectively. As can be seen, TTF increases steadily and significantly as Initial_rm increases. For Waterspace, Barrier mode minimizes TTF and Random maximizes it. None of the 95% confidence intervals overlap, meaning each level's mean is statistically different from each other level. Though the results are not given, a pair-wise t-test was performed to confirm this result, and the metrics for each effect can be seen in Figure 4.12. Note that for experiment 4, the F-ratio for the Initial_rm effect is only twice the F-ratio of the Waterspace effect, meaning that the Waterspace factor accounts for a much larger fraction of the model's variability.

Unlike in experiment 3, the Inital_rm*Waterspace effect is statistically significant for experiment 4. This can be best seen in the interaction plots presented in Figure 4.13. Note

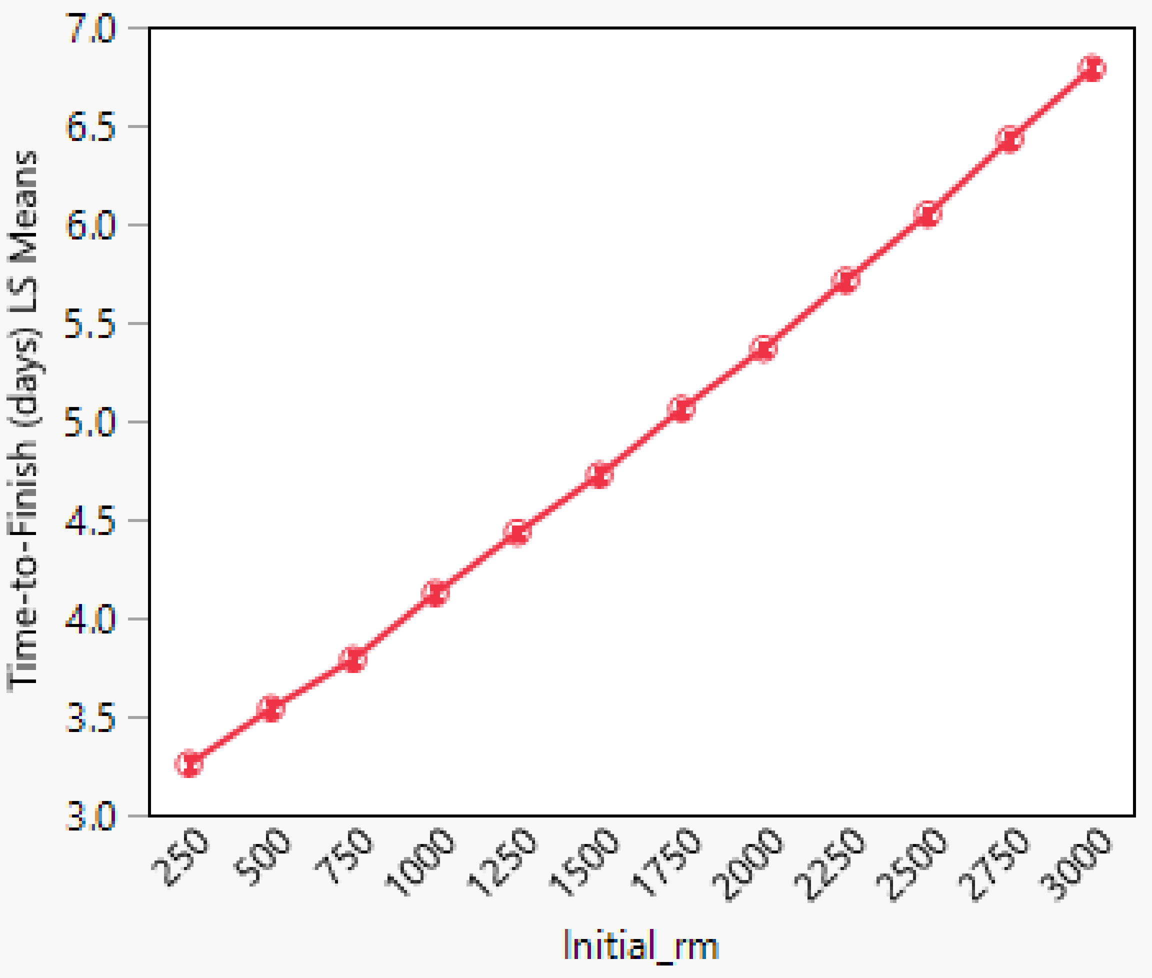


Figure 4.6. Experiment 3: LSM of TTF versus Initial_rm

that in the bottom-left plot, the drop in TTF when Waterspace = Barrier is slightly greater when Initial_rm = 250 merchants vice Initial_rm = 3000 merchants. You can also note that, in the top-right plot, the plots of TTF versus Initial_rm for each Waterspace begin to converge as Initial_rm increases. The effect size, however, is small. The F-ratio for the interaction effect is only 4.08, compared to F-ratios of 3028.38 and 6049.27 for the other two effects. If this effect was removed from the model, the model would still be viable. It is likely that further decreasing or increasing the value of Initial_rm would increase the Initial_rm*Waterspace effect. However, since these levels of Initial_rm would be outside the bounds of reality, such an experiment would be purely theoretical.

Overall, changing red force movement from Random to Straight increased the significance

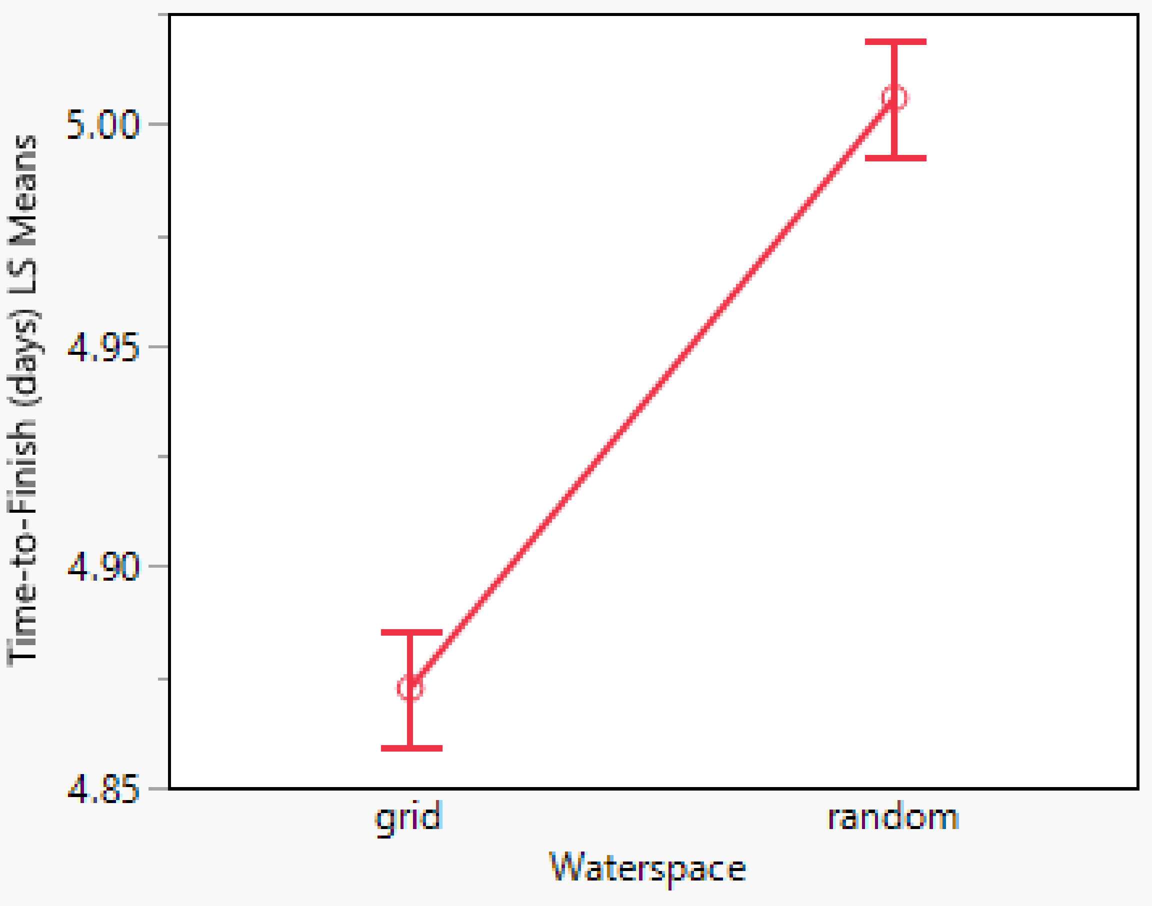
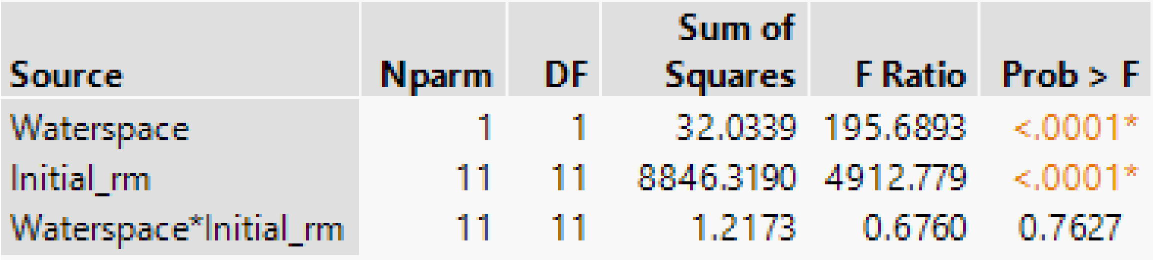


Figure 4.7. Experiment 3: LSM of TTF versus Waterspace

of blue force behavior. Interestingly, the Grid and Random blue force behaviors produced mean TTFs that were slightly higher in experiment 4 than in experiment 3. However, the addition of the Barrier behavior produced the smallest TTF among all behavior modes in both experiments. The advantage of setting Waterspace = Barrier was amplified when Initial_rm was small, though only slightly. Theoretically, we expect that blue force search behavior becomes insignificant when target density is increased to a certain threshold, but for the values of Initial_rm used in our experiments, that threshold was not reached.

4.4 TTF versus BSP

As previously stated, the two MOEs for our experiments are TTF and BSP. The analysis for our experiments has primarily focused on the effect of different factors on TTF. The reason



Source	Nparm	DF	Sum of Squares	F Ratio	Prob > F
Waterspace	1	1	32.0339	195.6893	<.0001*
Initial_rm	11	11	8846.3190	4912.779	<.0001*
Waterspace*Initial_rm	11	11	1.2173	0.6760	0.7627

Figure 4.8. Experiment 3: ANOVA Effects

These are the effect metrics for the experiment 3 ANOVA model. The effects include Initial_rm, Waterspace, and the Initial_rm*Waterspace interaction.

is that BSP directly follows from TTF. An ANOVA analysis of BSP versus Waterspace and Initial_rm reveals that both factors are significant in the determination of BSP. However, this is expected since both factors are significant in the determination of TTF. Of more interest is whether there are significant interaction effects between TTF*Waterspace and TTF*Initial_rm.

Two BSP ANOVAs were conducted. One used TTF and Waterspace as factors, and the other used TTF and Initial_rm as factors. Their respective parameter effects are shown in Figures 4.16 and 4.17. Note that statistically significant interaction effects do exist. However, as seen by the F-ratio values, these interaction effects are slight compared to the TTF effect.

The TTF*Waterspace interaction plot is shown in Figure 4.14. In general, given a constant TTF, a more effective Waterspace level, such as Barrier, leads to a slightly lower BSP than a less effective Waterspace level, such as Random. This is probably because the advantage gained from smarter search patterns is amplified against slower targets, such as the red force submarines. Since the red force submarines are the most lethal red force unit, the effectiveness of the barrier search leads to a slightly higher casualties for the blue force.

The TTF*Initial_rm interaction plot is shown in Figure 4.15. Here the plots of BSP versus TTF at two different levels of Initial_rm will cross. At low TTFs, a smaller Initial_rm yields a smaller BSP. This is simply because, at higher values of Initial_rm, a low TTF is not possible unless nearly all the blue submarines stay alive for the entirety of the simulation. At high TTFs, a smaller Initial_rm yields a higher BSP. We suspect this is due to the fact that with higher red merchant density a blue submarine is more likely to be engaged with a

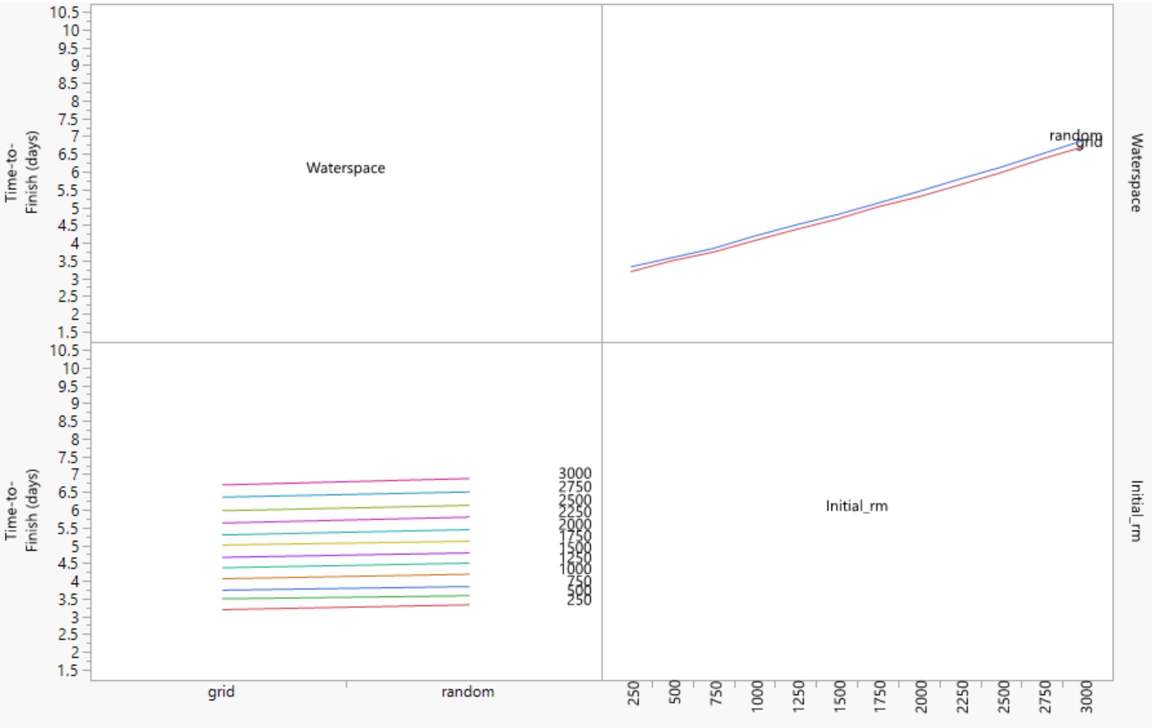


Figure 4.9. Experiment 3: Interaction Plots

These are the interaction plots for the experiment 3 ANOVA model. The bottom-left is a plot of TTF versus Waterspace at each level of Initial_rm. The top-right is a plot of TTF versus Initial_rm at each level of Waterspace.

merchant at any given time, which makes them more vulnerable to being ambushed by red combatants.

As was previously stated, even though interaction effects do exist, they are relatively small compared to the overall TTF effect. Therefore, a reasonable prediction of BSP can be made using TTF data alone. Figure 4.18 is a plot of BSP versus TTF using all data collected in experiments 3 and 4. The solid line is a degree-4 prediction model.

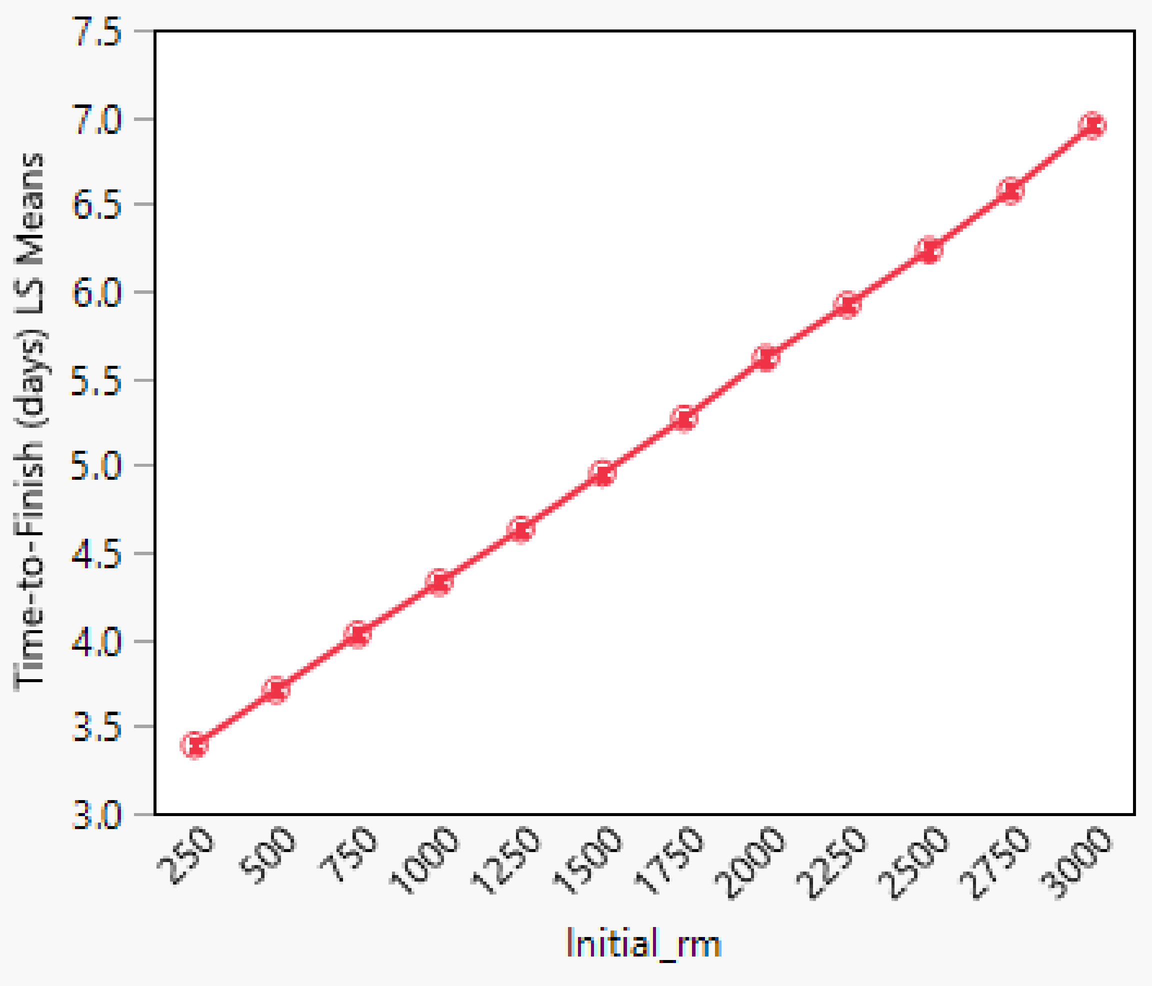


Figure 4.10. Experiment 4: LSM of TTF versus Initial_rm

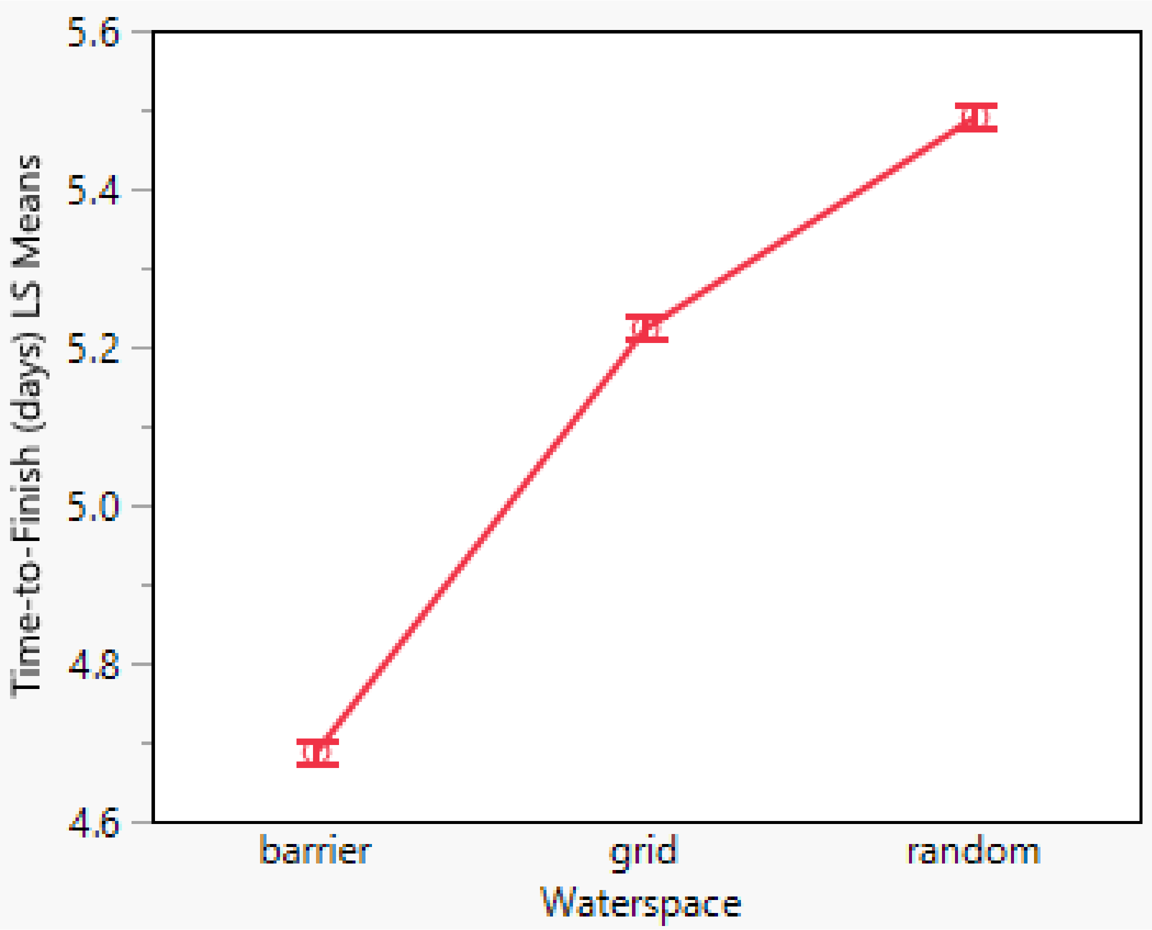
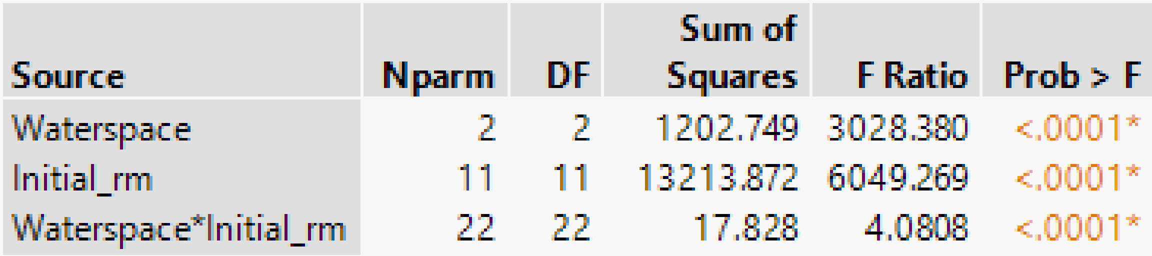


Figure 4.11. Experiment 4: LSM of TTF versus Waterspace



Source	Nparm	DF	Sum of Squares	F Ratio	Prob > F
Waterspace	2	2	1202.749	3028.380	<.0001*
Initial_rm	11	11	13213.872	6049.269	<.0001*
Waterspace*Initial_rm	22	22	17.828	4.0808	<.0001*

Figure 4.12. Experiment 4: ANOVA Effects

These are the effect metrics for the experiment 4 ANOVA model. The effects include Initial_rm, Waterspace, and the Initial_rm*Waterspace interaction.

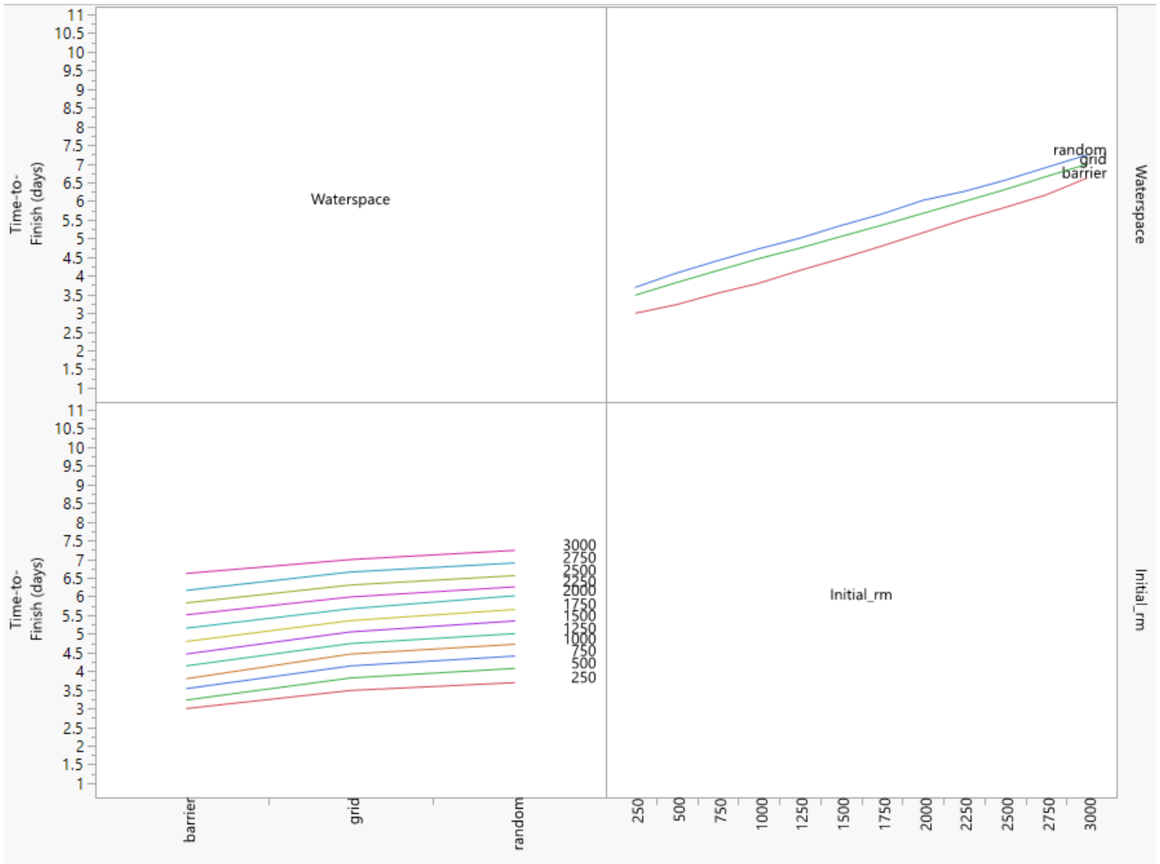


Figure 4.13. Experiment 4: Interaction Plots

These are the interaction plots for the experiment 4 ANOVA model. The bottom-left is a plot of TTF versus Waterspace at each level of Initial_rm. The top-right is a plot of TTF versus Initial_rm at each level of Waterspace.

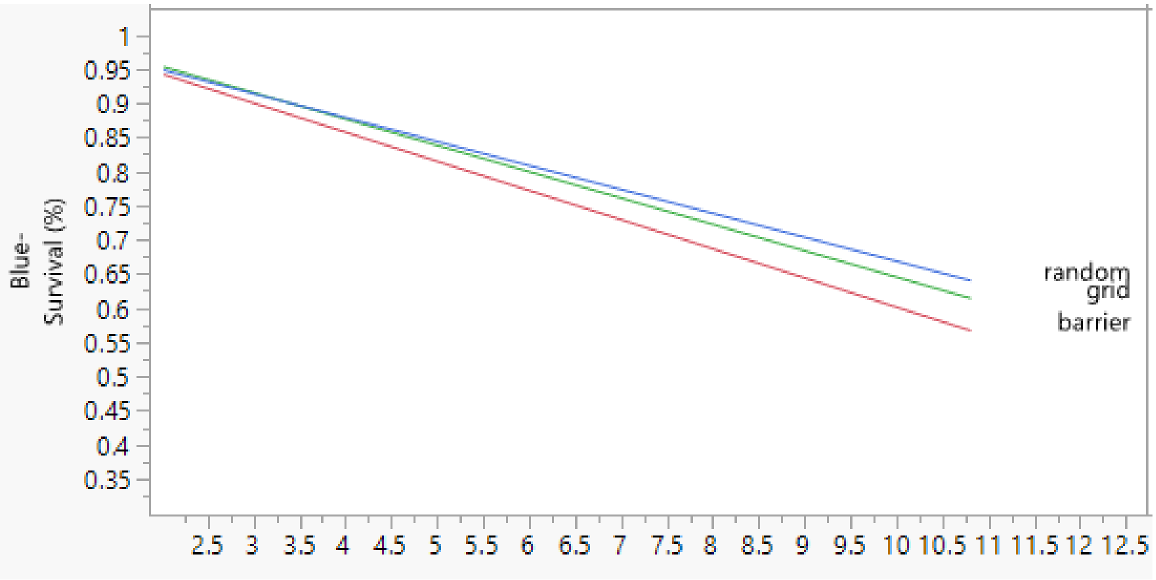


Figure 4.14. TTF*Waterspace effect on BSP

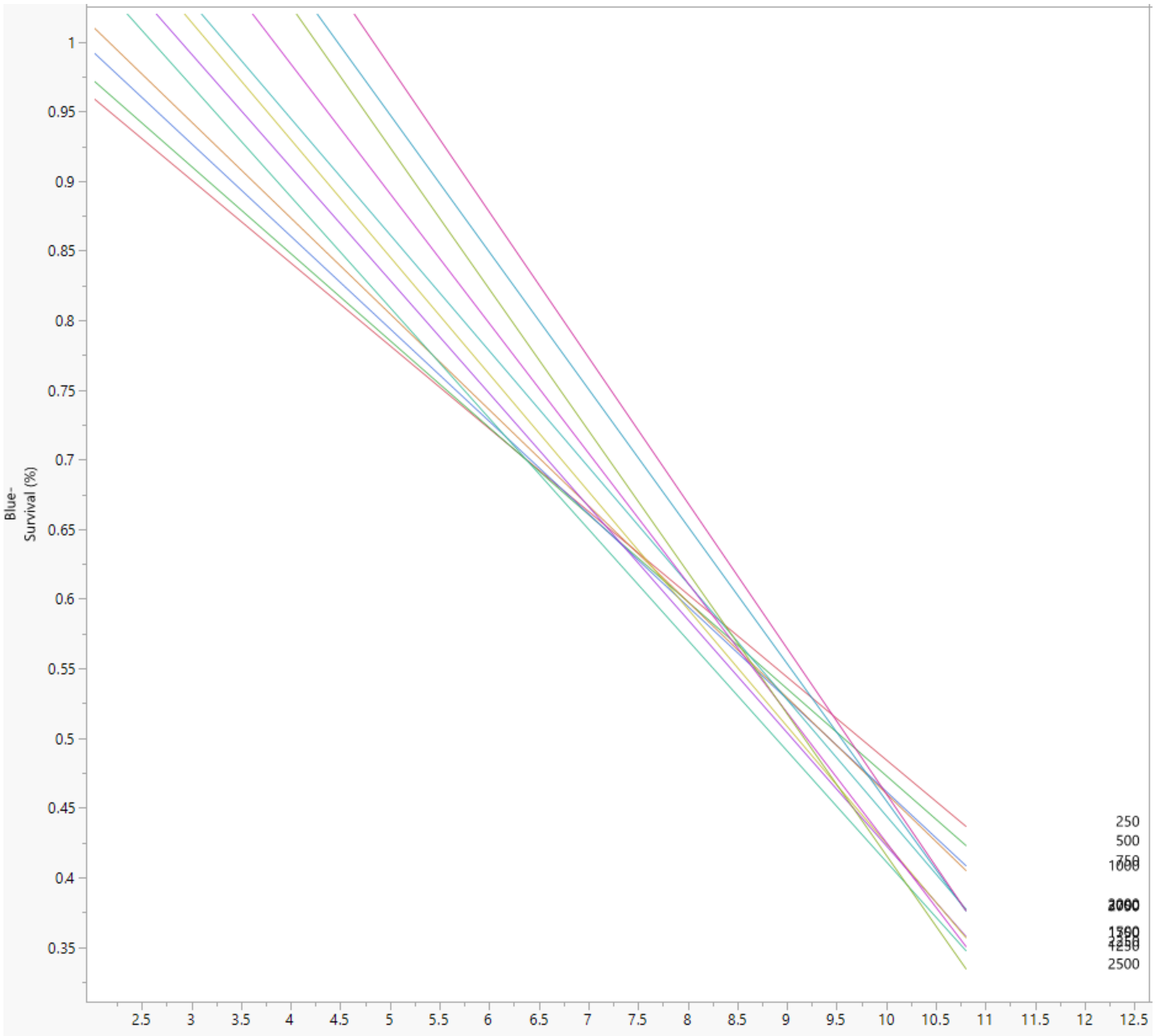
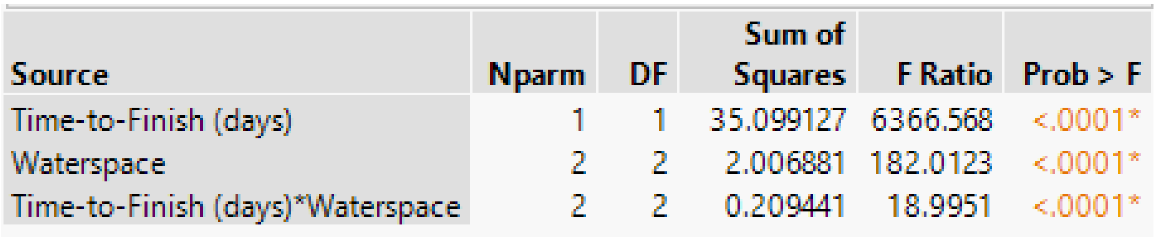
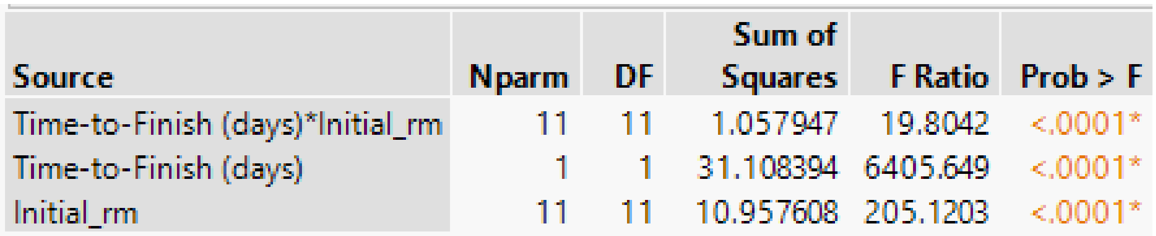


Figure 4.15. TTF*Initial_rm effect on BSP



Source	Nparm	DF	Sum of Squares	F Ratio	Prob > F
Time-to-Finish (days)	1	1	35.099127	6366.568	<.0001*
Waterspace	2	2	2.006881	182.0123	<.0001*
Time-to-Finish (days)*Waterspace	2	2	0.209441	18.9951	<.0001*

Figure 4.16. BSP ANOVA with Waterspace and TTF factors



Source	Nparm	DF	Sum of Squares	F Ratio	Prob > F
Time-to-Finish (days)*Initial_rm	11	11	1.057947	19.8042	<.0001*
Time-to-Finish (days)	1	1	31.108394	6405.649	<.0001*
Initial_rm	11	11	10.957608	205.1203	<.0001*

Figure 4.17. BSP ANOVA with Initial_rm and TTF factors

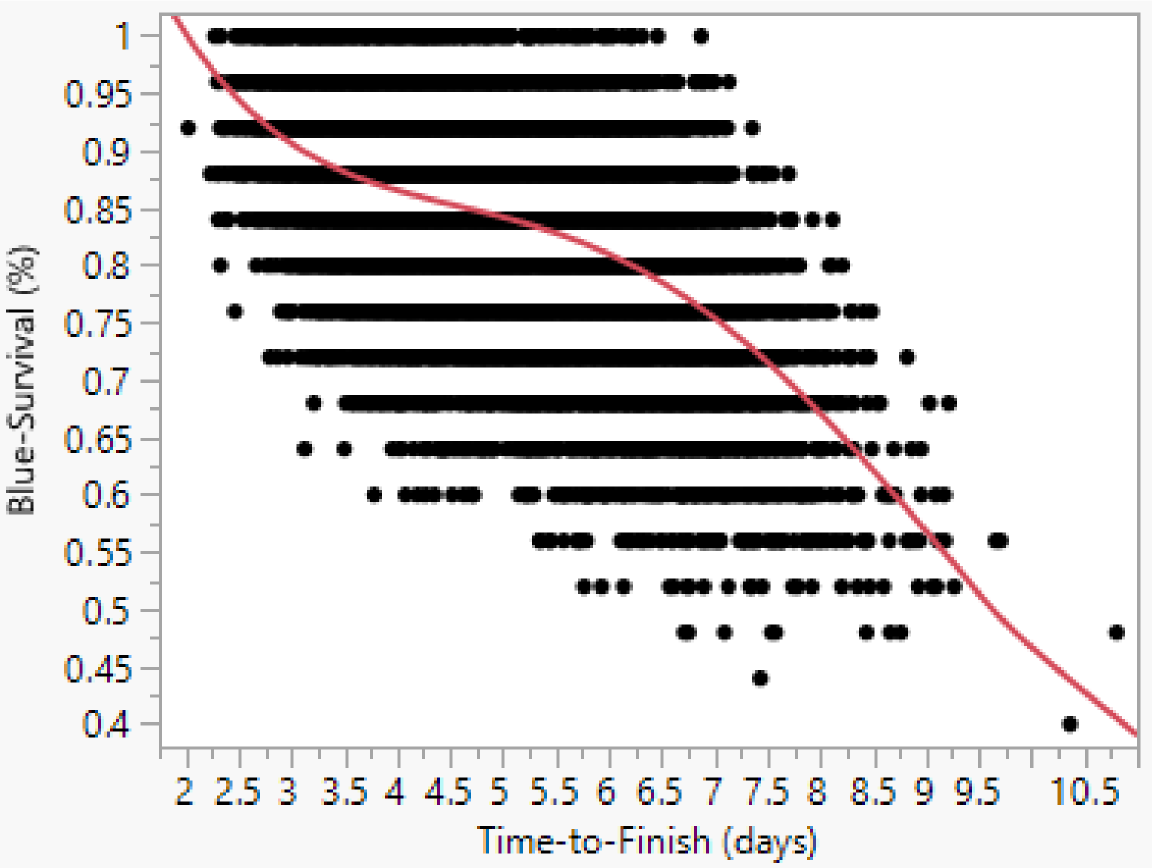


Figure 4.18. BSP versus TTF

This is a plot of BSP versus TTF for all data collected from experiments 3 and 4. The solid line is a degree-4 best-fit polynomial with the following equation: BSP $= 0.9666553 - 0.0248921 * TTF - 0.0053815 * (TTF - 5.05586)^2 - 0.0033247 * (TTF - 5.05586)^3 + 0.0004695 * (TTF - 5.05586)^4$

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5:
Conclusion

Heightening political and economic tensions with China have re-emphasized the need for well sought out strategies in the SCS and ECS. Due to the vulnerability of surface ships and air assets near mainland China, the undersea domain must play a pivotal role in these strategies. This thesis sought to further our understanding of U.S. submarine capabilities in the SCS and ECS by building upon the results of SASEM. By creating an independent model, we were able to further explore the viability of SASEM and the War at Sea Strategy.

5.1 Research Questions

The goal of this thesis was to address the following questions:

1. Is SASEM a valid model for simulating the War at Sea Strategy?
2. Given the improvements we have made upon SASEM, is it still feasible to use submarines to create a maritime exclusion zone in the SCS and ECS?
3. Regardless of the feasibility of the strategy, which search technique gives the U.S. force the most successful outcome?

To help answer these questions, four experiments were conducted and their results were analyzed based on two MOEs: Time-to-Finish (TTF) and Blue-Survival-Percent (BSP). In the following sections, each question is addressed separately.

5.1.1 SASEM Viability

Since SASEM and our model use different methodologies, we first attempted to replicate SASEM conditions in experiment 1 to ensure we produced similar results. We found our BSP was identical to SASEMs. TTF was slightly lower with a range of 78.27-150.58 days versus SASEM's 88-165 days. However, since the exact parameter values that produced SASEM's range for TTF are unknown, we conclude that our results are close enough to SASEM's results that we believe we successfully reproduced SASEM conditions using our methodology.

After validating that we could reproduce SASEM's results, we removed many of SASEM's assumptions from our model to see how significantly the model was affected. Most importantly, we removed the SASEM restraint that only allowed one engagement to occur at a time. We found that mean TTF significantly dropped to a new range of 11.97-19.49 days. Additionally, BSP dropped from a constant 89% to a range of 57.8-69.6%. The reason for these large deviations was determined to be a flaw in the SASEM methodology. The prosecution time used by SASEM effectively paused all the units in the simulation while an engagement was in progress, which meant most units spent the majority of the simulation as inactive. Additionally, by not allowing simultaneous engagements, the blue force submarines were protected from red force combatants by the relatively high density of red merchants. Considering both of the previously described deficiencies in SASEM, we conclude that SASEM is not a valid model for providing combat simulations. Therefore, we also conclude that the recommended values provided by Schwarz's thesis [2] are meaningless.

5.1.2 Viability of the War at Sea Strategy

The original design of this thesis was to gain further insight into the War at Sea Strategy by combining the original recommended parameters from SASEM with the more intelligent behavioral modeling of our combat simulation. However, since we no longer view SASEM as a viable model, its recommended values have no bearing on the War at Sea Strategy, which makes it difficult to judge the feasibility of the strategy based on our results.

In experiment 2, when we removed SASEM's limitations but otherwise used the recommended parameters, we saw that TTF and BSP both significantly decreased. Therefore, TTF is less of a restraint, which makes the strategy more viable. However, the level of blue force casualties could be considered unacceptable, which makes the strategy less viable. It would be useful to fix the deficiencies with SASEM and re-perform its analysis to find a new set of recommended key parameters. However, that is outside the scope of this thesis, and so we conclude that the War at Sea Strategy is neither more nor less viable based on our results.

5.1.3 Search Technique Analysis

Despite our results concerning SASEM, we still proceeded with experiments 3 and 4 to determine the effects of using different search and movement patterns. This was accomplished by analyzing red force patterns independently of each other and determining how TTF and BSP changed at different levels of blue force behavior. Experiment 3 focused on random red force movement, and experiment 4 focused on straight, predictable red force movement.

For experiment 3, we analyzed two blue force behaviors: random search and grid search. We found that, regardless of the amount of initial red force merchants, grid search produced TTFs that were in the range of 0.1-0.2 days lower than random search. We conclude that, even though the effect was statistically significant, it was small enough that very little fidelity would be lost in a model by replacing grid search with random search. Also, this statement remains true regardless of the initial target density.

For experiment 4, we analyzed three blue force behaviors: random search, grid search, and barrier search. Once again grid search produced lower TTFs than random search. This time the difference was approximately 0.2-0.3 days. Barrier search produced the lowest TTFs with results that were 0.6-0.7 days lower than random search. Though the advantage gained by barrier search began to decrease at higher numbers of merchants, the effect was not practically consequential for the merchant densities we examined. We conclude that barrier search is the superior search behavior when the targets' paths and trajectories are well defined. Additionally, the advantage gained by barrier search was significant enough that it should not be replaced by random or grid search for modeling purposes.

BSP was examined separately from TTF using all data from experiments 3 and 4. The relationship between TTF and BSP was analyzed for different levels of red force behavior, blue force behavior, and initial number of merchants. Overall, we found that TTF was the most significant predictor of BSP. The greatest interaction effect occurred with blue force behavior. At a constant TTF, mean BSP was highest for random search and lowest for barrier search. However, this interaction effect produced around 350 times less variability than the TTF effect. Therefore, we conclude that TTF alone is a reliable indicator for BSP.

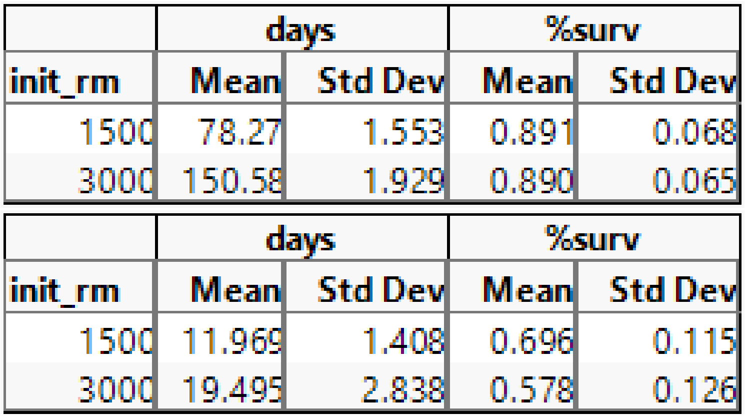
5.2 Follow-on Research

The following is a list of possible follow-on research topics related to this thesis:

- SASEM recommended parameters: This thesis was originally designed to build upon SASEM's results for further insight into the War at Sea Strategy. Since SASEM's results were found to be unreliable, the model should be fixed and its analysis concerning recommended parameters should be re-performed.
- Sensors and environment: Adjusting sensor performance based on environmental analysis would remove one of the simplifying assumptions of this thesis.
- Intelligent PRC response: Though it is an improvement from SASEM, the behaviors in this model are still simplistic. Most notably, more intelligent PRC strategies, such as coordinated maneuvers and conveys, are required.
- Expanding the War at Sea Strategy: Submarine use is only one aspect of the War at Sea Strategy. Implementing other components in the SCS and ECS would further our understanding of the strategy's feasibility.
- Using classified data: The parameters used in this thesis are rough estimates and not based on actual military performance parameters or intelligence. Classifying the thesis would allow for more accurate estimates of performance.
- Faster computation time: Increasing the performance speed of this thesis' model would make it easier to analyze a larger range of parameter values.

APPENDIX A:
Summary of Experiment Results

This appendix gives result summaries for all experiments described in Chapter 3. All figures presented are taken from the JMP software package. The independent variables are the different levels of initial red force merchant count (init_rm) and blue force behavior (Waterspace). The results experiments 3 and 4 have been split into separate tables for each value of Waterspace. The dependent variables are the days required to reduce the red merchant force to the breakpoint level (days) and the percent of blue force submarines remaining (%surv). The dependent variables' means and standard deviations are given for each combination of independent variables.

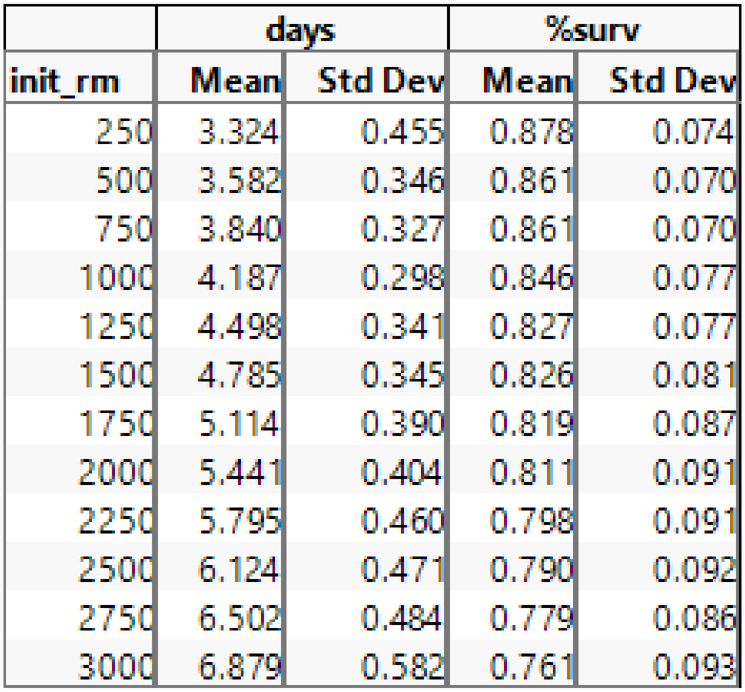


	days		%surv	
init_rm	Mean	Std Dev	Mean	Std Dev
1500	78.27	1.553	0.891	0.068
3000	150.58	1.929	0.890	0.065

	days		%surv	
init_rm	Mean	Std Dev	Mean	Std Dev
1500	11.969	1.408	0.696	0.115
3000	19.495	2.838	0.578	0.126

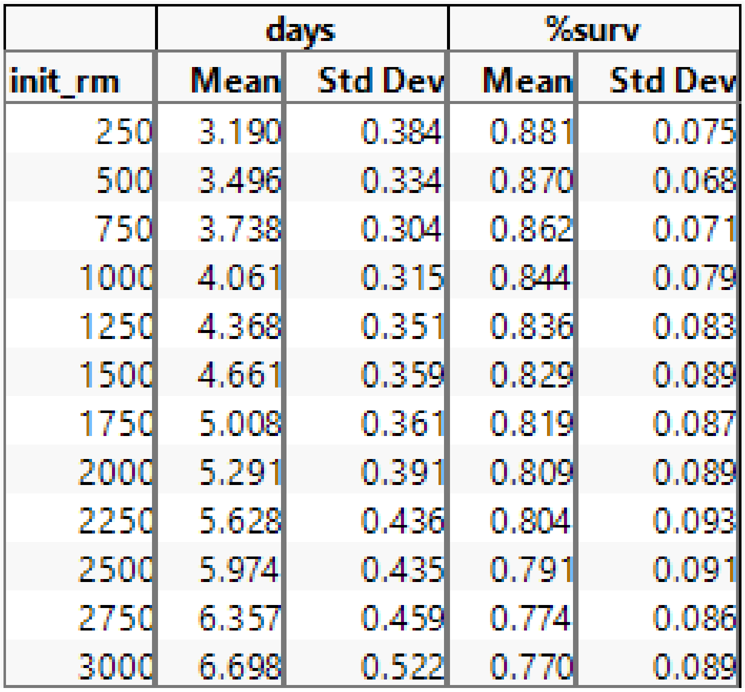
Figure A.1. Experiments 1 and 2 Results Summary

Top: Experiment 1. Bottom: Experiment 2



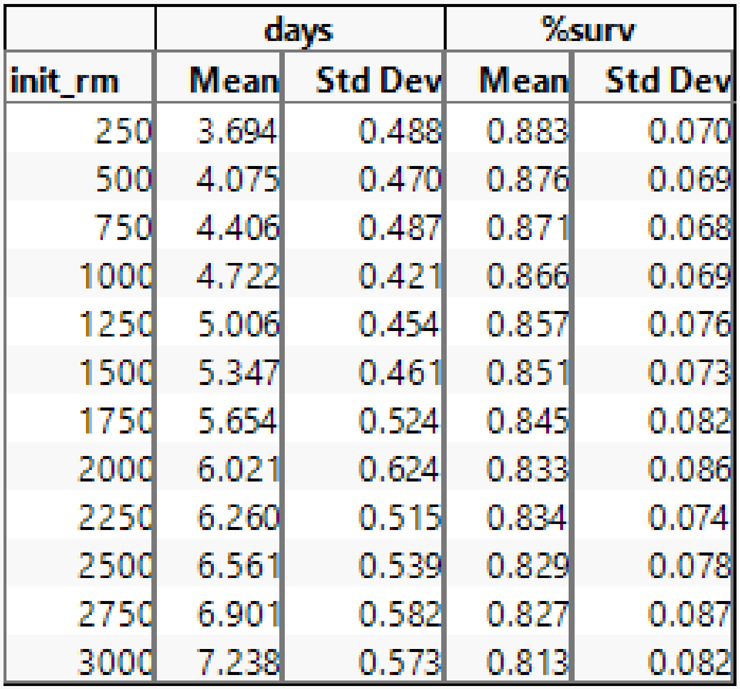
	days		%surv	
init_rm	Mean	Std Dev	Mean	Std Dev
250	3.324	0.455	0.878	0.074
500	3.582	0.346	0.861	0.070
750	3.840	0.327	0.861	0.070
1000	4.187	0.298	0.846	0.077
1250	4.498	0.341	0.827	0.077
1500	4.785	0.345	0.826	0.081
1750	5.114	0.390	0.819	0.087
2000	5.441	0.404	0.811	0.091
2250	5.795	0.460	0.798	0.091
2500	6.124	0.471	0.790	0.092
2750	6.502	0.484	0.779	0.086
3000	6.879	0.582	0.761	0.093

Figure A.2. Experiment 3 Results Summary: Waterspace = Random



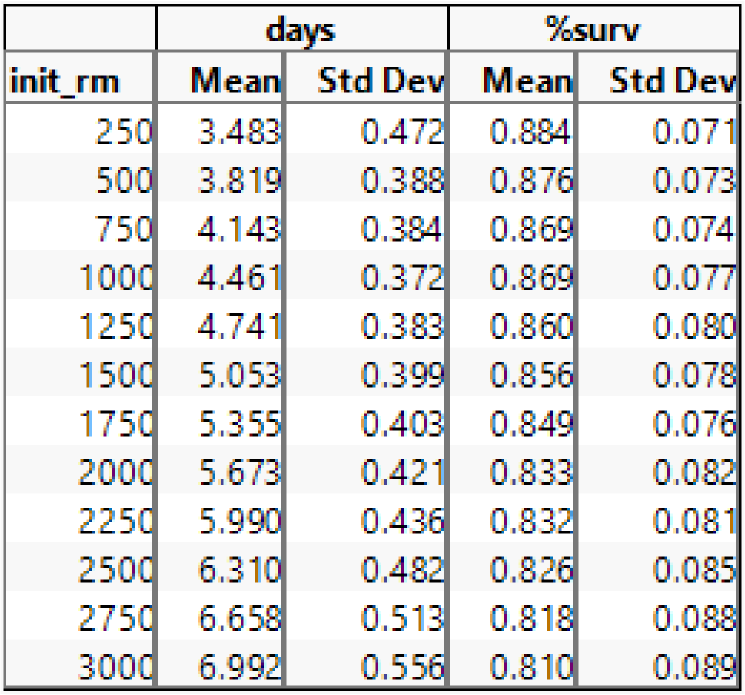
	days		%surv	
init_rm	Mean	Std Dev	Mean	Std Dev
250	3.190	0.384	0.881	0.075
500	3.496	0.334	0.870	0.068
750	3.738	0.304	0.862	0.071
1000	4.061	0.315	0.844	0.079
1250	4.368	0.351	0.836	0.083
1500	4.661	0.359	0.829	0.089
1750	5.008	0.361	0.819	0.087
2000	5.291	0.391	0.809	0.089
2250	5.628	0.436	0.804	0.093
2500	5.974	0.435	0.791	0.091
2750	6.357	0.459	0.774	0.086
3000	6.698	0.522	0.770	0.089

Figure A.3. Experiment 3 Results Summary: Waterspace = Grid



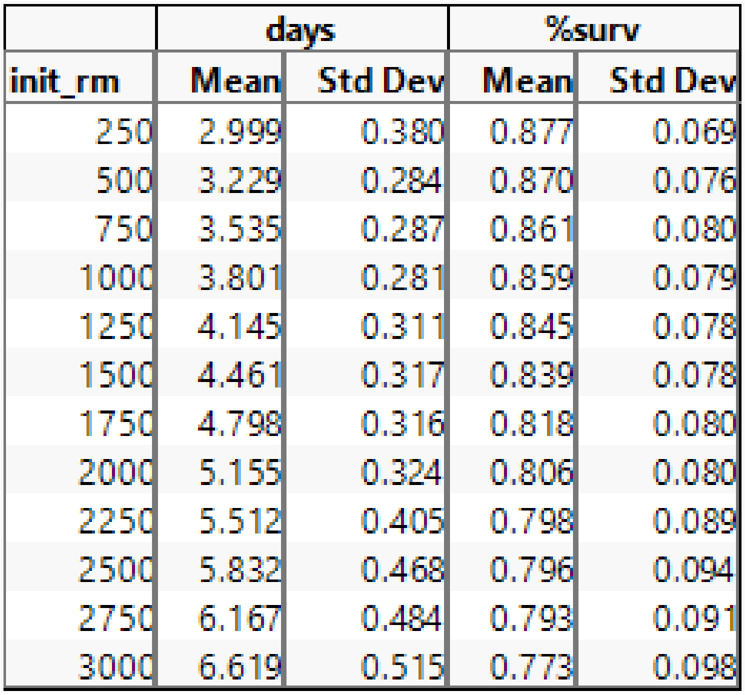
	days		%surv	
init_rm	Mean	Std Dev	Mean	Std Dev
250	3.694	0.488	0.883	0.070
500	4.075	0.470	0.876	0.069
750	4.406	0.487	0.871	0.068
1000	4.722	0.421	0.866	0.069
1250	5.006	0.454	0.857	0.076
1500	5.347	0.461	0.851	0.073
1750	5.654	0.524	0.845	0.082
2000	6.021	0.624	0.833	0.086
2250	6.260	0.515	0.834	0.074
2500	6.561	0.539	0.829	0.078
2750	6.901	0.582	0.827	0.087
3000	7.238	0.573	0.813	0.082

Figure A.4. Experiment 4 Results Summary: Waterspace = Random



	days		%surv	
init_rm	Mean	Std Dev	Mean	Std Dev
250	3.483	0.472	0.884	0.071
500	3.819	0.388	0.876	0.073
750	4.143	0.384	0.869	0.074
1000	4.461	0.372	0.869	0.077
1250	4.741	0.383	0.860	0.080
1500	5.053	0.399	0.856	0.078
1750	5.355	0.403	0.849	0.076
2000	5.673	0.421	0.833	0.082
2250	5.990	0.436	0.832	0.081
2500	6.310	0.482	0.826	0.085
2750	6.658	0.513	0.818	0.088
3000	6.992	0.556	0.810	0.089

Figure A.5. Experiment 4 Results Summary: Waterspace = Grid



	days		%surv	
init_rm	Mean	Std Dev	Mean	Std Dev
250	2.999	0.380	0.877	0.069
500	3.229	0.284	0.870	0.076
750	3.535	0.287	0.861	0.080
1000	3.801	0.281	0.859	0.079
1250	4.145	0.311	0.845	0.078
1500	4.461	0.317	0.839	0.078
1750	4.798	0.316	0.818	0.080
2000	5.155	0.324	0.806	0.080
2250	5.512	0.405	0.798	0.089
2500	5.832	0.468	0.796	0.094
2750	6.167	0.484	0.793	0.091
3000	6.619	0.515	0.773	0.098

Figure A.6. Experiment 4 Results Summary: Waterspace = Barrier

APPENDIX B:
MATLAB Code for Random Search Test

```
close all
clc
%% set parameter
A = 3600;  % Area = 60 * 60
V = 10;   % Searcher speed
U = 5;    % Target speed
lams = .2; % Searcher's mean course change rate of .2/hr
lamt = .2; % Target's mean course change rate of .2/hr
lam = 15;  % Poisson Scan Model with ? = 15 independent looks/hr
sig = 5; % Sigma of signal excess
rep = 1000; % Iteration time
R = 6; %detection range

%% Start the main iteration loop
time = NaN(1,rep);
rng = NaN(1,rep);
for i = 1:rep

    %% Initialize x and y positions and courses
    % Searcher
    sx = rand(1)*60;
    sy = rand(1)*60;
    sc = rand(1)*2*pi;
    % Target
    tx = rand(1)*60;
    ty = rand(1)*60;
    tc = rand(1)*2*pi;

    %% Time loop
    dt = .1;
    for t = 0:dt:150
        r = sqrt((sx-tx)^2 + (sy-ty)^2);

```

```

        % If dectection occurs, then save detection time
        % and range, and break out of the Time Loop
        if r <= R
            time(i) = t;
            rng(i) = r;
            break
        end

        % Course changes of Searcher
        if sx <= 0
            sc = normrnd(0, .3);
        elseif sx >= 60
            sc = pi + normrnd(0, .3);
        elseif sy <= 0
            sc = pi/2 + normrnd(0, .3);
        elseif sy >= 60
            sc = -pi/2 + normrnd(0, .3);
        elseif rand(1) <= lams * dt
            sc = rand(1)*2*pi;
        else
            sc = sc;
        end

        % Course changes of Target
        if tx <= 0
            tc = normrnd(0, .3);
        elseif tx >= 60
            tc = pi + normrnd(0, .3);
        elseif ty <= 0
            tc = pi/2 + normrnd(0, .3);
        elseif ty >= 60
            tc = -pi/2 + normrnd(0, .3);
        elseif rand(1) <= lams * dt
            tc = rand(1)*2*pi;
        else
            tc = tc;
        end

        % Advance searcher and target position
```

```
        sx = sx + cos(sc)*V*dt;
        sy = sy + sin(sc)*V*dt;
        tx = tx + cos(tc)*U*dt;
        ty = ty + sin(tc)*U*dt;
    end
end

%% Compute CDF for time of initial detection
cdf(1) = length(find(time<=1 & time>=0)) / rep;
for t = 2:150
    cdf(t) = cdf(t-1) + length(find(time<=t & time>t-1)) / rep;
end

%% Figure of CDF
t = 1:150;
figure
plot(t, cdf, 'r.')
grid on
hold on

%% Calculate F(t)
lam_1 = -log(1-cdf(150))/150;  % best fit detection rate
% relative speed of V, U
Vdet = 1/pi*integral(@(theta) ...
sqrt(V^2 + U^2 -2*U*V*cos(theta)), 0, pi);
R = lam_1*A/(Vdet*2) % best fit detection range
Ft = @(t) 1 - exp(-lam_1*t);
fplot(Ft, [0, 150])

%% Plot
xlabel('time (hr)')
ylabel('F(t)')
legend('Model CDF','Best Fit Theoretical CDF', ...
'Location','SouthEast')
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C:
MATLAB Code for Combat Model

```
%Code used to simulate submarine combat in SCS and ECS.
%Relevant parameters are explained at the start of the code.
%Indenting has been removed do to page size constraints.

%Author: Bryan McDonough

%Last Updated: 8/30/19

clear
close all

%% choose data options
plot_sim = 0; %plots the simulation and records movie if true
movie_length = 0; %length of movie in days (plot_sim must be true)
movie_name = 'sim_movie2.avi'; %name for movie .avi file
save_name = '412.mat'; %name for data .mat file
plot_hist = 0; %plots course history at end of simulation if true
iter = 300; %number of combat simulations performed

%% choose behavioral options
waterspace = "random"; %random,grid,barrier-affects blue force locations
movement_b = "random"; %random,barrier-affects blue force movement
movement_r = "random"; %random,straight-affects red force movement
cooperation = 1; %determines if red forces fire simultaneously
SASEM = 0; %when true, makes simulation more like SASEM
priority_warship = 1; %if true, blue prioritizes attacking warships

%% set breakpoints
bp_rm = 0.7; % percent merchants remaining

%% set misc parameters
initial_rm = 3000; %number of red merchants
initial_b = 25; %number of blue submarines
```

```
initial_rs = 60; %number of red submarines
initial_rw = 60; %number of red surface warships
Width = 900; %length of x-dir 900
Length = 1500; %length of y-dir 1500
A = Length*Width; % Area for submarines to search in
Vb_search = 7; % Blue search speed
Vb_sub = 10; %prosecuting submarine
Vb_ws = 13; %prosecuting warship
Vb_m = 15; %prosecuting merchant
Vrs_search = 3; % Red sub search speed - 3
Vrs_sub = 6; % Red sub attack speed - 6
Vrw_search = 15; % Red warship search speed - 15
Vrw_sub = 25; % Red warship attack speed - 25
Vrm = 20; % Red merchant speed - 20
lambda = .01; % Searcher's mean course change rate of occurances/hr
lamt = lambda; % Target's mean course change rate of occurances/hr
rec_time = 5; %time required to recover after shooting or being shot at
pros_sub = 50; %time to prosecute a submarine (includes recovery time)
pros_surf = 35; %time to prosecute surface ship (includes recover time)

%% Set encounter survival rates: UnitInQuestion_EnemyUnit_FirstToShoot
%blue shoots first against red sub - 3:1 exchange ratio
b_rs_b = .8; %.8
rs_b_b = .4; %.4
%blue shoots first against red warship - 8:1 exchange ratio
b_rw_b = .95;
rw_b_b = .2;
%blue shoots against red merchant
rm_b_b = .1;
%red sub shoots first against blue
b_rs_rs = .5;
rs_b_rs = .7;
%red warship shoots first against blue
b_rw_rw = .6;
rw_b_rw = .6;
%red and blue subs shoot at each other simultaneously
b_rs_s = .6;
rs_b_s = .5;
%red warship and blue sub shoot at each other simultaneously
b_rw_s = .7;
```

```
rw_b_s = .5;

%% Set sensor parameters
%cookie-cutter model - r_DetectingUnit_DetectedUnit (in nm)
R_b_rm = 7;
R_b_rw = 5;
R_b_rs = 4;
R_rs_b = 3;
R_rw_b = 2;
det_c_b = @(r,unit_type) (r<=R_b_rs).*(unit_type==1)...
+(r<=R_b_rw).*(unit_type==2)+(r<=R_b_rm).*(unit_type==3);
det_c_rs = @(r) r<=R_rs_b;
det_c_rw = @(r) r<=R_rw_b;

%% Start the main iteration loop
%preallocate data vectors
days = zeros(iter,1);
PERCREM_B = days;
PERCREM_RS = days;
PERCREM_RM = days;
PERCREM_RW = days;
RM_DETECTS = days;
KILLperDETECT_RM = days;
end_movie = 0;
for m = 1:iter
if plot_sim == 1
figure
if movie_length > 0
vidobj = VideoWriter(movie_name);
vidobj.FrameRate = 240;
open(vidobj)
end_movie = 1;
end
end
%% Initialize blue and red initial velocities
%blue subs
vb = ones(1,initial_b)*Vb_search;
%red forces
vr_default = [ones(1,initial_rs)*Vrs_search,...
ones(1,initial_rw)*Vrw_search,ones(1,initial_rm)*Vrm];
```

```
vr = vr_default;

%% Initialize blue and red initial x and y positions and courses
%Blue Forces
positions = zeros(25,25);
switch waterspace
case "grid"
s = floor(sqrt(initial_b));
excess = initial_b-s^2;
grd = ones(1,s)*s;
for i = 1:excess
col = mod(i-1,s)+1;
grd(col) = grd(col) + 1;
end
maxgrd = max(grd);
positions = zeros(maxgrd,s);
unit = 1;
for j = 1:s
if mod(j,2)~= 0
order = 1:maxgrd;
else
order = maxgrd:-1:1;
end
for i = order
if i > maxgrd-grd(j)
positions(i,j) = unit;
unit = unit+1;
end
end
end
bx = zeros(1,initial_b);
by = bx;
row_subs = length(grd);
for i = 1:initial_b
[I,J] = find(positions==i);
col_subs = grd(J);
I = 1+maxgrd-I;
bx(i) = (J-1)/row_subs*Width + rand(1)/row_subs*Width;
by(i) = (I-1)/col_subs*Length + rand(1)/col_subs*Length;
end
```

```
case "barrier"
grd = [0,0,0,0,0,initial_b,0,0,0,0,0];
maxgrd = initial_b;
positions = zeros(initial_b,11);
positions(:,6) = (1:initial_b)';
unit = 1;
bx = zeros(1,initial_b);
by = bx;
row_subs = length(grd);
for i = 1:initial_b
[I,J] = find(positions==i);
col_subs = grd(J);
I = 1+maxgrd-I;
bx(i) = (J-1)/row_subs*Width + rand(1)/row_subs*Width;
by(i) = (I-1)/col_subs*Length + rand(1)/col_subs*Length;
end
case "random"
bx = rand(1,initial_b)*Width;
by = rand(1,initial_b)*Length;
end
switch movement_b
case "random"
bc = rand(1,initial_b)*2*pi;
case "barrier"
ran = rand(1,initial_b);
bc = round(ran)*pi/2 - ~round(ran)*pi/2;
end
% Red Forces
initial_r = initial_rs+initial_rm+initial_rw;
rx = rand(1,initial_r)*Width;
ry = rand(1,initial_r)*Length;
if movement_r == "random"
rc = rand(1,initial_r)*2*pi;

elseif movement_r == "straight"
ran = rand(1,initial_r);
rc = 0*(ran>.5)+pi*(ran<=.5);
end
%form engagement matrix
eng_trckr_b = zeros(initial_b,2);
```

```
eng_trckr_r = zeros(initial_rs+initial_rw,2);
status_b = ones(1,initial_b);
status_r = zeros(1,initial_r);
status_r(1:initial_rs+initial_rw) = 1;


%% Start Time Loop
%preallocate position matrices
end_time = 10000;
dt = .1;
time_steps = floor(end_time/dt)+1;
bx_1 = zeros(time_steps,initial_b);
by_1 = bx_1;
rx_1 = zeros(time_steps,initial_r);
ry_1 = rx_1;
%set other initial values
k = 1;
alive_r = 1:initial_r;
alive_b = 1:initial_b;
rm_detects = 0; %tracks total number of merchant detections
engagement = 0; %tracks if engagement is in progress
for t = 0:dt:end_time

%% PERFORM PLOTTING
%save current positions for history plot
if plot_hist == 1
bx_1(k,:) = bx;
by_1(k,:) = by;
rx_1(k,:) = rx;
ry_1(k,:) = ry;
end
%Plot Simulation
if plot_sim == 1
dr = [.5,0,0]; %dark red color
%red merchants
hold off
alive_rm = alive_r(alive_r>(initial_rs+initial_rw));
plot(rx(alive_rm), ry(alive_rm), 'rs','MarkerFaceColor','r')
hold on
plot(rx(initial_rs+initial_rw+1:end),...
```

```
ry(initial_rs+initial_rw+1:end), 'rs')
%red subs
alive_rs = alive_r(alive_r<=initial_rs);
plot(rx(alive_rs), ry(alive_rs), 'rv','MarkerFaceColor',dr,...
'MarkerEdgeColor',dr)
plot(rx(1:initial_rs), ry(1:initial_rs), 'rv',...
'MarkerEdgeColor',dr)
%red warships
alive_rw =...
alive_r(alive_r>initial_rs&alive_r<=initial_rw+initial_rs);
plot(rx(alive_rw), ry(alive_rw), '^','MarkerFaceColor',dr,...
'MarkerEdgeColor',dr)
plot(rx(initial_rs+1:initial_rs+initial_rw),...
ry(initial_rs+1:initial_rs+initial_rw), 'r^',...
'MarkerEdgeColor',dr)
%blue subs
plot(bx(alive_b), by(alive_b), 'bv','MarkerFaceColor','b')
plot(bx(:), by(:), 'bv')
%plot options
axis([0,Width,0,Length])
axis square
grid on
xlabel('x (nm)')
ylabel('y (nm)')
ttl = sprintf('Number of Days = %.2f', t/24);
title(ttl)
%Plot Waterspace Grid
switch waterspace
case "grid"
longs = 0:Width/row_subs:Width;
X = [longs',longs'];
Y = [zeros(length(longs),1),...
ones(length(longs),1)*Length];
plot(X',Y','k')
for i = 1:row_subs
lats = 0:Length/grd(i):Length;
x = [longs(i),longs(i+1)];
[X,Y] = meshgrid(x,lats);
X = X';
Y = Y';
```

```
plot(X,Y,'k')
end
case "barrier"
long1 = floor(row_subs/2);
long2 = ceil(row_subs/2);
longs = [long1,long2]/row_subs*Width;
X = [longs',longs'];
Y = [zeros(length(longs),1),...
ones(length(longs),1)*Length];
plot(X',Y','k')
lats = 0:Length/length(alive_b):Length;
[X,Y] = meshgrid(longs,lats);
X = X';
Y = Y';
plot(X,Y,'k')
end
if t/24 < movie_length
Movie = getframe(gcf);
writeVideo(vidobj,Movie)
elseif end_movie
close(vidobj)
end_movie = 0;
end
shg %show current figure
end

%% DETERMINE CONTACT GAINS
%determine distance between living red and blue forces
[Rx,Bx] = meshgrid(rx(alive_r),bx(alive_b));
[Ry,By] = meshgrid(ry(alive_r),by(alive_b));
r = sqrt((Bx-Rx).^2 + (By-Ry).^2); %range matrix
%determine if blue submarines gain any contacts
searching_of_alive = find(status_b(alive_b)==1);
pd = zeros(size(r));
for i = 1:length(alive_r)
red_unit = alive_r(i);
if red_unit<=initial_rs
unit_type = 1;
elseif red_unit<=initial_rw
unit_type = 2;
```

```matlab
else
unit_type = 3;
end
pd(:,i) = det_c_b(r(:,i),unit_type);
end
for  i = searching_of_alive
if  SASEM && engagement
break %only one engagement allowed in SASEM mode
end
blue_unit = alive_b(i);
detected = find(pd(i,:)>=1);
if ~isempty(detected)
engagement = 1;
if priority_warship
red_unit = alive_r(detected(1));
else
red_unit = alive_r(detected(end));
end
status_b(blue_unit) = 0;
eng_trckr_b(blue_unit,1) = red_unit;
if red_unit <= initial_rs
eng_trckr_b(blue_unit,2) = pros_sub;
else
if red_unit > initial_rs+initial_rw
rm_detects = rm_detects+1;
end
eng_trckr_b(blue_unit,2) = pros_surf;
end
end
end
%determine if red forces detect any blue forces
searching_of_alive = find(status_r(alive_r)==1);
for  j = searching_of_alive
if  SASEM && engagement
break %only one engagement allowed in SASEM mode
end
red_unit = alive_r(j);
if red_unit <= initial_rs
pd(:,j) = det_c_rs(r(:,j));
else
```

```
pd(:,j) = det_c_rw(r(:,j));
end

detected = find(pd(:,j)>=1);
if ~isempty(detected)
engagement = 1;
blue_unit = alive_b(detected(1));
status_r(red_unit) = 0;
prior = find(eng_trckr_r(:,1)==blue_unit);
eng_trckr_r(red_unit,1) = blue_unit;
if ~isempty(prior) && cooperation
eng_trckr_r(red_unit,2) = eng_trckr_r(prior(1),2);
else
eng_trckr_r(red_unit,2) = pros_sub;
end
end
end

%% RETURN FORCES TO ACTIVELY SEARCHING
%red forces
engaged_r = eng_trckr_r(:,2)~=0;
eng_trckr_r(:,2) = eng_trckr_r(:,2)-round(10*dt)*engaged_r;
resume_searching = (eng_trckr_r(:,2)~=0)~=engaged_r;
status_r(resume_searching) = 1;
%blue forces
engaged_b = eng_trckr_b(:,2)~=0;
eng_trckr_b(:,2) = eng_trckr_b(:,2)-round(10*dt)*engaged_b;
resume_searching = (eng_trckr_b(:,2)~=0)~=engaged_b;
status_b(resume_searching) = 1;

%% RESOLVE COMBAT
%resolve blue subs firing
attacking_b =...
find(eng_trckr_b(:,2)==rec_time & eng_trckr_b(:,1)~=0);
attacked_r = eng_trckr_b(attacking_b,1);
for i = 1:length(attacking_b)
engagement = 0;
incoming_b = rand(1);
incoming_r = rand(1);
blue_unit = attacking_b(i);
```

```
red_unit = attacked_r(i);
if ~SASEM
%reduce firing accuracy with range
pa = max(1,2/5*r(alive_b==blue_unit,alive_r==red_unit)-1);
else
pa = 1;
end
%ensure red target still lives
if status_r(red_unit) ~= -1
%blue attacking red combatant
if red_unit <= initial_rs+initial_rw
%determine if red is allowed to counterfire
if ~(eng_trckr_r(red_unit,1) == 0 && ...
(eng_trckr_r(red_unit,1)<=rec_time && ...
eng_trckr_r(red_unit,1)>0))
%simultaneous fire
if eng_trckr_r(red_unit,1) == blue_unit &&...
eng_trckr_r(red_unit,2) <= ceil(1.5*rec_time)
%determine if blue dies
if (red_unit<=initial_rs && ...
incoming_b>b_rs_s*pa) || ...
(red_unit>initial_rs && incoming_b>b_rw_s*pa)
status_b(blue_unit) = -1;
eng_trckr_b(blue_unit,2) = 0;
positions(positions==blue_unit) = 0;
end
%determine if red dies
if (red_unit>initial_rs && ...
incoming_r>rw_b_s*pa) || ...
(red_unit<=initial_rs && incoming_r>rs_b_s*pa)
status_r(red_unit) = -1;
eng_trckr_r(red_unit,2) = 0;
else
eng_trckr_r(red_unit,2) = rec_time;
end
%blue fires first
else
%determine if blue dies
if (red_unit<=initial_rs && ...
 incoming_b>b_rs_b*pa) || ...
```

```
 (red_unit>initial_rs && incoming_b>b_rw_b)
status_b(blue_unit) = -1;
eng_trckr_b(blue_unit,2) = 0;
positions(positions==blue_unit) = 0;
end
%determine if red dies
if (red_unit>initial_rs && ...
incoming_r>rw_b_b*pa) || ...
(red_unit<=initial_rs && incoming_r>rs_b_b*pa)
status_r(red_unit) = -1;
eng_trckr_r(red_unit,2) = 0;
else
status_r(red_unit) = 0;
eng_trckr_r(red_unit,2) = rec_time;
end
end
%red cannot counterfire
%red dies
elseif (red_unit>initial_rs && ...
incoming_r>rw_b_b*pa) || ...
(red_unit<=initial_rs && incoming_r>rs_b_b*pa)
status_r(red_unit) = -1;
eng_trckr_r(red_unit,2) = 0;
%red lives
else
status_r(red_unit) = 0;
eng_trckr_r(red_unit,2) = rec_time;
end
eng_trckr_r(red_unit,1) = 0;
%blue attacking red merchant
elseif incoming_r > rm_b_b*pa
status_r(red_unit) = -1;
end
end
end
eng_trckr_b(attacking_b,1) = 0;

%resolve red firing
attacking_r = ...
find(eng_trckr_r(:,2)==rec_time & eng_trckr_r(:,1)~=0);
```

```
attacked_b = eng_trckr_r(attacking_r,1);
for i = 1:length(attacking_r)
engagement = 0;
red_unit = attacking_r(i);
blue_unit = attacked_b(i);
incoming_b = rand(1);
incoming_r = rand(1);
if ~SASEM
%reducing firing accuracy with range
pa = max(1,2/5*r(alive_b==blue_unit,alive_r==red_unit)-1);
else
par = 1;
end
%ensure blue target still lives
if status_b(blue_unit) ~= -1
%determine if blue is allowed to counterfire
if ~(eng_trckr_b(blue_unit,1) == 0 && ...
(eng_trckr_b(blue_unit,1)<=rec_time ...
&& eng_trckr_b(blue_unit,1)>0))
%simultaneous fire
if  eng_trckr_b(blue_unit,1) == red_unit &&...
eng_trckr_b(blue_unit,2) <= ceil(1.5*rec_time)
%determine if blue dies
if (red_unit<=initial_rs && ...
incoming_b>b_rs_s*pa) || ...
(red_unit>initial_rs && incoming_b>b_rw_s*pa)
status_b(blue_unit) = -1;
eng_trckr_b(blue_unit,2) = 0;
positions(positions==blue_unit) = 0;
else
eng_trckr_b(blue_unit,2) = rec_time;
end
%determine if red dies
if  (red_unit>initial_rs && incoming_r>rw_b_s*pa) || ...
(red_unit<=initial_rs && incoming_r>rs_b_s*pa)
status_r(red_unit) = -1;
eng_trckr_r(red_unit,2) = 0;
end
%red fires first
else
```

```
%determine if blue dies
if (red_unit<=initial_rs && ...
incoming_b>b_rs_rs*pa) || ...
(red_unit>initial_rs && incoming_b>b_rw_rw*pa)
status_b(blue_unit) = -1;
eng_trckr_b(blue_unit,2) = 0;
positions(positions==blue_unit) = 0;
else
eng_trckr_b(blue_unit,2) = rec_time;
end
%determine if red dies
if (red_unit>initial_rs && ...
incoming_r>rw_b_rw*pa) || ...
(red_unit<=initial_rs && incoming_r>rs_b_rs*pa)
status_r(red_unit) = -1;
eng_trckr_r(red_unit,2) = 0;
end
end
%blue cannot counterfire - determine if blue dies
elseif (red_unit<=initial_rs && ...
incoming_b>b_rs_rs*pa) || ...
(red_unit>initial_rs && incoming_b>b_rw_rw*pa)
status_b(blue_unit) = -1;
eng_trckr_b(blue_unit,2) = 0;
positions(positions==blue_unit) = 0;
else
status_b(blue_unit) = 0;
eng_trckr_b(blue_unit,2) = rec_time;
end
end
end
eng_trckr_r(attacking_r,1) = 0;
eng_trckr_b(attacked_b,1) = 0;

%% CHECK BREAKPOINT CONDITIONS
remain_rm = sum(alive_r>(initial_rs+initial_rw));
prev_alive_b = alive_b;
alive_b = find(status_b~=-1);
if remain_rm/initial_rm<bp_rm || isempty(alive_b)
break
```

```
end

%% COURSE AND SPEED CHANGES
% Course and speed changes of Blue
vb(1:initial_b) = Vb_search;
for i = 1:length(alive_b)
blue_unit = alive_b(i);
track = eng_trckr_b(blue_unit,1);
%if tracking enemy, get close to enemy
if track~=0
if r(i,alive_r==track)>3.5
dx = rx(track)-bx(blue_unit);
dy = ry(track)-by(blue_unit);
bc(blue_unit) = atan2(dy,dx);
vb(blue_unit) = Vb_sub*(track<=initial_rs)+Vb_ws*...
(track<=(initial_rw+initial_rs)&track>initial_rs)...
+Vb_m*(track>(initial_rw+initial_rs));
else
bc(blue_unit) = rc(track);
vb(blue_unit) = min(vr(track),Vb_m);
end
%if not tracking use preset default behavior
else
%set boundries
switch waterspace
case "barrier"
[~,J] = find(positions==blue_unit);
col = positions(:,J);
col(col==0) = [];
I = find(col==blue_unit);
col_subs = length(col);
grd(J) = col_subs;
I = 1+col_subs-I;

x_low = (J-1)/row_subs*Width;
x_high = J/row_subs*Width;
y_low = (I-1)/col_subs*Length;
y_high = I/col_subs*Length;
case "grid"
[~,J] = find(positions==blue_unit);
```

```
col = positions(:,J);
col(col==0) = [];
I = find(col==blue_unit);
col_subs = length(col);
grd(J) = col_subs;
I = 1+col_subs-I;

x_low = (J-1)/row_subs*Width;
x_high = J/row_subs*Width;
y_low = (I-1)/col_subs*Length;
y_high = I/col_subs*Length;
case "random"
x_low = 0;
x_high = Width;
y_low = 0;
y_high = Length;
end
%determine movement type
switch movement_b
case "random"
if bx(blue_unit) <= x_low
bc(blue_unit) = normrnd(0, .3);
elseif bx(blue_unit) >= x_high
bc(blue_unit) = pi + normrnd(0, .3);
elseif by(blue_unit) <= y_low
bc(blue_unit) = pi/2 + normrnd(0, .3);
elseif by(blue_unit) >= y_high
bc(blue_unit) = -pi/2 + normrnd(0, .3);
elseif rand(1) <= lambda * dt
bc(blue_unit) = rand(1)*2*pi;
end
case "barrier"
if bx(blue_unit) <= x_low
bc(blue_unit) = 0;
elseif bx(blue_unit) >= x_high
bc(blue_unit) = pi;
elseif by(blue_unit) <= y_low
bc(blue_unit) = pi/2;
elseif by(blue_unit) >= y_high
bc(blue_unit) = -pi/2;
```

```
elseif bc(blue_unit)~=pi/2 && bc(blue_unit)~=-pi/2
ran = rand(1);
bc(blue_unit)=round(ran)*pi/2-~round(ran)*pi/2;
end
end
end
end
% Course and speed changes of Red
vr = vr_default;
alive_r = find(status_r ~= -1);
red_units = alive_r;
for i = 1:length(red_units)
red_unit = red_units(i);
%if combatant, determine if it is actively tracking
if red_unit <= initial_rs+initial_rw
track = eng_trckr_r(red_unit,1);
else
track = 0;
end
%if tracking, get close to enemy and match course and speed
if track~=0
%close distance
if r(prev_alive_b==track,i) > 2
dx = bx(track)-rx(red_unit);
dy = by(track)-ry(red_unit);
rc(red_unit) = atan2(dy,dx);
%Red unit increases speed
if red_unit <= initial_rs
vr(red_unit) = Vrs_sub;
else
vr(red_unit) = Vrw_sub;
end
%match course and speed
else
rc(red_unit) = bc(track);
if red_unit <= initial_rs
vr(red_unit) = min(vb(track),Vrs_sub);
else
vr(red_unit) = vb(track);
end
```

```
end
%if not tracking use preset default behavior
elseif rx(red_unit) <= 0
if movement_r == "straight"
rc(red_unit) = 0;
else
rc(red_unit) = normrnd(0, .3);
end
elseif rx(red_unit) >= Width
if movement_r == "straight"
rc(red_unit) = pi;
else
rc(red_unit) = pi + normrnd(0, .3);
end
elseif ry(red_unit) <= 0
rc(red_unit) = pi/2 + normrnd(0, .3);
elseif ry(red_unit) >= Length
rc(red_unit) = -pi/2 + normrnd(0, .3);
elseif movement_r == "random" && rand(1) <= lambda * dt
rc(red_unit) = rand(1)*2*pi;
elseif movement_r == "straight" && ...
(rc(red_unit)~=0 && rc(red_unit)~=pi)
ran = rand(1);
rc(red_unit) = 0*(ran<.5)+pi*(ran>=.5);
end
end

%% ADVANCE RED AND BLUE FORCE POSITIONS
bx(alive_b) = bx(alive_b) + cos(bc(alive_b)).*vb(alive_b)*dt;
by(alive_b) = by(alive_b) + sin(bc(alive_b)).*vb(alive_b)*dt;
rx(alive_r) = rx(alive_r) + cos(rc(alive_r)).*vr(alive_r)*dt;
ry(alive_r) = ry(alive_r) + sin(rc(alive_r)).*vr(alive_r)*dt;
k = k +1;


end
%plot course history
if plot_hist == 1
figure
plot(bx_1(1:k,:), by_1(1:k,:), 'b')
```

```
hold on
plot(rx_1(1:k,:), ry_1(1:k,:), 'r')
hold on
plot(bx_1(1,:), by_1(1,:), 'bo')
hold on
plot(rx_1(1,:), ry_1(1,:), 'ro')
hold on
plot(bx_1(k,:), by_1(k,:), 'bs')
hold on
plot(rx_1(k,:), ry_1(k,:), 'rs')
grid on
xlabel('x')
ylabel('y')
end
%save performance data
days(m,1) = t/24;
PERCREM_B(m,1) = length(alive_b)/initial_b; %remaining blue forces
PERCREM_RS(m,1) = sum(alive_r<=initial_rs)/initial_rs;
PERCREM_RM(m,1) = remain_rm/initial_rm;
PERCREM_RW(m,1) = sum(alive_r>initial_rs&alive_r<= ...
(initial_rs+initial_rw))/initial_rw;
RM_DETECTS(m,1) = rm_detects;
KILLperDETECT_RM(m,1) = (1-PERCREM_RM(m))*initial_rm/rm_detects;
end
t_res = table(days,PERCREM_B,PERCREM_RS,PERCREM_RW,...
PERCREM_RM,KILLperDETECT_RM);
display(t_res)
parameters = table(waterspace,sensor_type,movement_b,movement_r,...
movement_r,cooperation,SASEM,priority_warship,initial_b,initial_rs,...
initial_rm,initial_rw,lambda,lamt,rec_time,pros_sub,...
pros_surf,b_rs_b,rs_b_b,b_rw_b,rw_b_b,rm_b_b,b_rs_rs,rs_b_rs,...
b_rw_rw,rw_b_rw,b_rs_s,rs_b_s,b_rw_s,rw_b_s,R_b_rm,R_b_rw,...
R_b_rs,R_rs_b,R_rw_b);
save(save_name,'t_res','parameters')
```

THIS PAGE INTENTIONALLY LEFT BLANK

List of References

[1] "Military expenditure (current USD)." (2019, Aug). The World Bank. [Online]. Available: https://data.worldbank.org/indicator/MS.MIL.XPND.CD?locations=CN

[2] Z. P. Schwartz, "Using undersea assets to establish a maritime exclusion zone in the south and east china seas," M.S thesis, Op. Res. Dept., Monterey, CA, USA, 2013.

[3] J. Kline and W. Hughes, "Between peace and war: A war at sea strategy," *Naval War College Review*, vol. 65, no. 4, pp. 35–41, 2012.

[4] J. Pike. (2011). "People's Liberation Navy - Offshore defense." [Online]. Available: https://www.globalsecurity.org/military/world/china/plan-doctrine-offshore.htm

[5] "The PLA NAVY: New capabilities and missions for the 21^{st} century." (2015, Mar. 4). Office of Naval Intelligence. [Online]. Available: https://news.usni.org/2015/04/09/document-office-of-naval-intelligence-2015-assessment-of-chinese-peoples-liberation-army-navy

[6] "Ballistic and Cruise Missile Threat." (2017, Jun). National Air and Space Intelligence Center. [Online]. Available: https://fas.org/wp-content/uploads/media/NASIC2017.pdf

[7] J. Pramuk and J. Schoen. (2019, Jun. 29). "US and China agree to continue tariff talks. Here's a timeline of how the trade war started." [Online]. Available: https: //www.cnbc.com/2019/06/29/us-china-trade-talks-at-g-20-timeline-of-how-the-tariff-war-started.html

[8] "Treaties in force: A list of treaties and other international agreements of the United States in force on January 1, 2019." (2019, Jan 1). U.S. State Department. [Online]. Available: https://www.state.gov/wp-content/uploads/2019/06/2019-TIF-Bilaterals-6.13.2019-web-version.pdf

[9] Google Earth Ver. 9.2.91.0. [Online]. Available: https://www.google.com/earth/. Accessed August 16, 2019. Camera alt 6919 Km, 17°00'00"N, 115°00'00"E.

[10] A. Washburn, *Search and Detection*, 5th ed. Self-published, 2014, ch. 2, p. 21.

[11] A. Washburn, *Search and Detection*, 5th ed. Self-published, 2014, ch. 6, pp. 89–90.

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California