2018-06

SEABED WARFARE AND THE XLUUV

Carr, Christopher J.; Franco, Jahdiel; Mierzwa, Cheryl; Shattuck IV,Lewis B.; Suursoo, Melissa A.

Monterey, CA; Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

SYSTEMS ENGINEERING CAPSTONE REPORT

SEABED WARFARE AND THE XLUUV

by

Christopher J. Carr, Jahdiel Franco, Cheryl Mierzwa,
Lewis B. Shattuck IV, and Melissa A. Suursoo

June 2018

Project Advisors: Paul T. Beery
Eugene P. Paulo
Richard D. Williams

THIS PAGE INTENTIONALLY LEFT BLANK

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

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2018	**3. REPORT TYPE AND DATES COVERED** Systems Engineering Capstone Report

4. TITLE AND SUBTITLE
SEABED WARFARE AND THE XLUUV

5. FUNDING NUMBERS

6. AUTHOR(S) Christopher J. Carr, Jahdiel Franco, Cheryl Mierzwa, Lewis B. Shattuck IV, and Melissa A. Suursoo

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)

Seabed warfare is quickly becoming one of the most important new research areas for the U.S. Navy. Defining seabed warfare is a challenge being faced by many as research continues in this new and innovative field. The problem of developing a concept of operations for performing seabed warfare operating in both offensive and defensive environments is becoming increasingly important as the need to complete kill chains without placing high-value assets at risk rises. With the introduction of the new Extra-Large Unmanned Undersea Vehicle (XLUUV), there is an interest to determine the utility of the XLUUV in performing seabed warfare. There are many potential capabilities to unlock within the seabed warfare field. This report takes the concept of kill box, a three-dimensional area used to facilitate the integration of coordinated joint weapons fire, and applies it to a new domain: undersea. The kill box includes seabed sensors; intelligence, surveillance, and reconnaissance (ISR) devices; and effects devices. This paper provides a concept of operations for seabed warfare in an undersea kill box with simulation results to determine the utility of the XLUUV.

14. SUBJECT TERMS seabed warfare, extra-large unmanned underwater vehicle, kill box	**15. NUMBER OF PAGES** 225
	16. PRICE CODE

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

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

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

SEABED WARFARE AND THE XLUUV

Christopher J. Carr, Jahdiel Franco, Cheryl Mierzwa,

Lewis B. Shattuck IV, and Melissa A. Suursoo

Submitted in partial fulfillment of the
requirements for the degrees of

MASTER OF SCIENCE IN ENGINEERING SYSTEMS

and

MASTER OF SCIENCE IN SYSTEMS ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 2018

Lead editor: Cheryl Mierzwa

Reviewed by:
Paul T. Beery
Project Advisor

Eugene P. Paulo
Project Advisor

Richard D. Williams
Project Advisor

Accepted by:
Ronald E. Giachetti
Chair, Department of Systems Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Seabed warfare is quickly becoming one of the most important new research areas for the U.S. Navy. Defining seabed warfare is a challenge being faced by many as research continues in this new and innovative field. The problem of developing a concept of operations for performing seabed warfare operating in both offensive and defensive environments is becoming increasingly important as the need to complete kill chains without placing high-value assets at risk rises. With the introduction of the new Extra-Large Unmanned Undersea Vehicle (XLUUV), there is an interest to determine the utility of the XLUUV in performing seabed warfare. There are many potential capabilities to unlock within the seabed warfare field. This report takes the concept of kill box, a three-dimensional area used to facilitate the integration of coordinated joint weapons fire, and applies it to a new domain: undersea. The kill box includes seabed sensors; intelligence, surveillance, and reconnaissance (ISR) devices; and effects devices. This paper provides a concept of operations for seabed warfare in an undersea kill box with simulation results to determine the utility of the XLUUV.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I. INTRODUCTION....1
A. BACKGROUND....1
B. PURPOSE....4
C. TECHNICAL APPROACH....5
1. Systems Engineering Process....5
2. Department of Defense Architecture Framework....6
D. SUMMARY....7

II. PROBLEM DEFINITION....9
A. PROBLEM BOUNDARIES....9
B. PROBLEM ASSUMPTIONS AND CONSTRAINTS....9
C. PROBLEM SCOPE....10
D. STAKEHOLDER ANALYSIS....10
E. PROBLEM STATEMENT....13
F. SUMMARY....13

III. WARFARE REQUIREMENTS DEFINITION....15
A. SEABED WARFARE DEFINITION....15
1. Operational Analysis....15
2. Concept of Operations....21
3. Operational Activities to Capabilities Map (CV-6)....24
B. MISSION CONTEXT....26
1. Mission Assumptions....26
2. Mission Constraints....26
3. Mission Boundaries....26
4. Mission Interfaces....27
C. REQUIREMENTS ANALYSIS....27
D. WARFARE REQUIREMENTS DEFINITION SUMMARY....28

IV. MISSION DEVELOPMENT....29
A. MISSION ARCHITECTURE AND DESIGN....29
1. Mission 1: Intelligence Preparation of the Operational Environment....29
2. Mission 2: Intelligence, Surveillance, and Reconnaissance....30
3. Mission 3: Effects Field Deployment....32
4. Mission 4: Engage and Strike....34
B. MODELING AND SIMULATION....36

1. Modeling Approach 36
2. Assumptions and Constraints 37
3. Attribute Decomposition 39
4. Simulation Logic 39
C. MISSION DEVELOPMENT SUMMARY 43

V. MISSION ANALYSIS 45
A. ANALYSIS METHODOLOGY 45
1. Data Definition and Collection 45
2. Design of Experiments 47
B. MISSION ANALYSIS 49
1. Intelligence Preparation of the Operational Environment 49
2. Intelligence, Surveillance, and Reconnaissance 68
3. Effects Field Deployment 80
4. Engage and Strike 90
C. SIMULATIONS LIMITATION ANALYSIS 100
D. MISSION ANALYSIS SUMMARY 101

VI. CONCLUSION 103
A. MISSION CONCLUSION 103
B. KILL BOX CONCLUSION 105
C. FUTURE UUV CAPABILTY 105
D. FUTURE WORK 106

APPENDIX A. SEABED WARFARE CAPABILITIES 109

APPENDIX B. MISSION ACTION DIAGRAMS 111

APPENDIX C. MODELING AND SIMULATION 117

APPENDIX D. DESIGN OF EXPERIMENT ANALYSIS 121
A. MISSION 1: INTELLIGENCE PREPARATION OF THE OPERATIONAL AREA 121
B. MISSION 2: INTELLIGENCE, SURVEILLANCE, AND RECONNAISSANCE 125
C. MISSION 3: EFFECTS FIELD DEPLOYMENT 132
D. MISSION 4: ENGAGE AND STRIKE 133

APPENDIX E. SIMULATION SOURCE CODE 135
A. GRAPHICAL USER INTERFACE AND MAIN FUNCTION 135

B. SIMULATION INITIALIZATION 176
C. MISSION SIMULATION MODELS 179
1. Mission 1: Intelligence Preparation of the Operational Environment 179
2. Mission 2: Intelligence, Reconnaissance, and Surveillance 181
3. Mission 3: Effects Field Deployment 185
4. Mission 4: Engage and Strike 189
D. INTERFACE UPDATES 192
E. EXPORT RESULTS 193

LIST OF REFERENCES 197

INITIAL DISTRIBUTION LIST 199

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

Figure 1. Seabed Warfare Overview. Source: General Dynamics Mission Systems (2018).....2

Figure 2. Underwater Great Wall. Source: Lin & Singer (2016). 3

Figure 3. Waterfall Systems Engineering Process 6

Figure 4. Multi-Service Tactics, Techniques, and Procedures (MTTP) Defined Purple Kill Box. Source: Army (2005). 16

Figure 5. Proposed Yellow Kill Box 17

Figure 6. Proposed Orange Kill Box 18

Figure 7. Operational Activity Decomposition Tree of Operation Leviathan 20

Figure 8. Scenario 1: IPOE.....21

Figure 9. Scenario 2: ISR 22

Figure 10. Scenario 3: Effects Field Delivery.....23

Figure 11. Scenario 4: Engage and Strike 24

Figure 12. Mission 1: IPOE Sequence Diagram 29

Figure 13. Mission 2: ISR Sequence Diagram.....31

Figure 14. Mission 3: Effects Field Deployment Sequence Diagram.....33

Figure 15. Mission 4: Engage and Strike Sequence Diagram.....35

Figure 16. Simulation Data Flow 36

Figure 17. IPOE Mission Scatterplot 52

Figure 18. XLUUV Response Distribution for IPOE 54

Figure 19. LDUUV Response Distribution for IPOE 55

Figure 20. MDUUV Response Distribution for IPOE 56

Figure 21. SUUV Response Distributions for IPOE.....57

Figure 22. Kill Box Size vs. IPOE Mission Success.....59

Figure 23. IPOE Mission Success for Small Kill Box 60

Figure 24. IPOE Mission Success for Medium/Large Kill Boxes 60
Figure 25. Likelihood of Identification during IPOE for Small Kill Boxes.....61
Figure 26. Likelihood of Identification during IPOE for Medium/Large Kill Boxes.....62
Figure 27. Number of IPOE Attempts Used for Small Kill Boxes 63
Figure 28. Number of IPOE Attempts Used for Medium/Large Kill Boxes 63
Figure 29. Number of Detections during IPOE for Small Kill Box.....64
Figure 30. Number of Detections during IPOE for Medium/Large Kill Boxes.....64
Figure 31. IPOE Mission Success Comparison for Full Kill Box Range 65
Figure 32. IPOE Mission Success Comparison for Small (left) and Medium/Large (right) Kill Boxes 66
Figure 33. IPOE Mission Success Comparison for 1–100 nm Kill Boxes.....68
Figure 34. ISR Mission Scatterplot.....71
Figure 35. ISR Payload Size.....72
Figure 36. XLUUV Response Distributions for ISR 73
Figure 37. LDUUV Response Distributions for ISR 74
Figure 38. MDUUV Response Distributions for ISR 75
Figure 39. ISR Mission Success Binned by Kill Box 77
Figure 40. Kill Box Size vs. ISR Mission Success/Number of Successful ISR Deployments 78
Figure 41. ISR Mission Success Binned by Number of Undersea Systems 79
Figure 42. ISR Mission Success Comparison for Full Kill Box Range.....80
Figure 43. Effect Field Deployment Mission Results.....82
Figure 44. Kill Box Size vs. EF Mission Success/Number of Successful EF Deployments 83
Figure 45. Effects Payload Size 84
Figure 46. XLUUV Response Distribution for Effects Field Deployment.....85

Figure 47. LDUUV Response Distribution for Effects Field Deployment 86
Figure 48. Number of Successful Deployments Compared to the Probability of Deployment 88
Figure 49. Effects Field Deployment Comparison for Full Kill Box Range 89
Figure 50. Engage and Strike Mission Scatterplot 92
Figure 51. Engage and Strike Mission Target Responses 93
Figure 52. Engage and Strike Mission Responses 93
Figure 53. Detection, Classification and Tracking Capability across Undersea System Type 94
Figure 54. XLUUV Response Distribution for Engage and Strike 95
Figure 55. LDUUV Response Distributions for Engage and Strike 96
Figure 56. MDUUV Response Distribution for Engage and Strike 97
Figure 57. Engage and Strike Factors Effect Size 99
Figure 58. Mission Responses Binned by Kill Box Type 100
Figure 59. Mission Responses Binned by Number of Undersea Systems 100
Figure 60. Kill Box Deployment Success 102
Figure 61. CV-2: Seabed Warfare Capabilities Diagram 110
Figure 62. High-Level Operation Leviathan 112
Figure 63. Mission 1: IPOE 113
Figure 64. Mission 2: Intelligence, Surveillance, and Reconnaissance 114
Figure 65. Mission 3: Effects Field Deployment 115
Figure 66. Mission 4: Engage and Strike 116
Figure 67. Additional IPOE Mission Scatterplots 121
Figure 68. Additional Scatterplots for IPOE Mission Success for Small Kill Box Ranges 122
Figure 69. Additional Scatterplots for IPOE Mission Success for Medium/Large Kill Box Ranges 122

Figure 70. Additional Scatterplots for Identification Likelihood during IPOE for Small Kill Box Ranges....123

Figure 71. Additional Scatterplots for Identification Likelihood during IPOE for Medium/Large Kill Box Ranges....123

Figure 72. Additional Scatterplots for IPOE Attempts Used for Small Kill Box Ranges....124

Figure 73. Additional Scatterplots for IPOE Attempts Used for Medium/Large Kill Box Ranges....124

Figure 74. Additional Scatterplots for Number of Detections during IPOE for Small Kill Box Ranges....125

Figure 75. Additional Scatterplots for Number of Detections during IPOE for Medium/Large Kill Box Ranges....125

Figure 76. Additional ISR Mission Scatterplots....126

Figure 77. Number of Successful ISR Deployments Binned ISR Deployment Probability....127

Figure 78. Number of Detections During ISR Binned by ISR Deployment Probability....128

Figure 79. Number of Successful ISR Deployments Binned by OPFOR Detecting U.S. Probability....129

Figure 80. Likelihood of Identification During ISR Binned by ISR Deployment Probability....130

Figure 81. Likelihood of Identification During ISR Binned by ISR Field Resolution Device Requirement....131

Figure 82. Likelihood of Identification During ISR Binned by OPFOR Detecting U.S. Probability....132

Figure 83. Additional EF Mission Scatterplots....133

Figure 84. Additional Engage and Strike Mission Scatterplots....134

Figure 85. Seabed Warfare Simulation GUI....135

LIST OF TABLES

Table 1. Stakeholder Analysis Table 12

Table 2. Operation Leviathan Mission Scenarios 19

Table 3. CV-6 Capability to Operational Activities Mapping 25

Table 4. Top Level Mission Requirements 28

Table 5. IPOE Mission Criteria and Data Outputs 40

Table 6. ISR Mission Criteria and Data Outputs 41

Table 7. Effects Field Mission Criteria and Data Outputs 42

Table 8. Engage and Strike Mission Criteria and Data Outputs 43

Table 9. DOE Design Factors 48

Table 10. DOE Design Responses 49

Table 11. IPOE Mission Factors 50

Table 12. IPOE Mission Responses 51

Table 13. IPOE Response Screening 58

Table 14. ISR Mission Factors 69

Table 15. ISR Mission Responses 70

Table 16. ISR Response Screening 76

Table 17. Effects Field Deployment Mission Factors 81

Table 18. Effects Field Deployment Mission Responses 81

Table 19. Effects Field Deployment Response Screening 87

Table 20. Engage and Strike Mission Factors 90

Table 21. Engage and Strike Mission Responses 91

Table 22. Engage and Strike Response Screening 98

Table 23. Attribute Decomposition 117

Table 24. Attribute Trace Matrix 119

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

ASW	antisubmarine warfare
ASuW	anti-surface warfare
AUWS	Advanced Undersea Weapon System
AO	area of operation
AoA	analysis of alternatives
CONOPs	concept of operations
CV	Capability Viewpoint
DARPA	Defense Advanced Research Projects Agency
DODAF	Department of Defense Architecture Framework
DOE	design of experiments
FDECO	Forward Deployed Energy and Communications Outpost
EF	effects device
FSCM	fire support coordinating measure
GUI	graphical user interface
HWIL	hardware-in-the-loop
ILS	integrated logistics support
IPOE	intelligence preparation of the environment
ISR	intelligence, surveillance, and reconnaissance
LDUUV	Large Diameter Unmanned Underwater Vehicle
MCM	mine countermeasures
MOE	measure of effectiveness
MOP	measure of performance
MTTP	Multi-Service Tactics, Techniques, and Procedures
MUSE	Modular Undersea Effector
MDUUV	Medium Diameter Unmanned Underwater Vehicle
NSW	Naval Special Warfare
NSWC	Naval Surface Warfare Center
NUWC	Naval Undersea Warfare Center
ONR	Office of Naval Research
OPFOR	opposing forces

OV	Operational Viewpoint
PLA	People’s Liberation Army
PMS	Program Office of the Navy
SADL	situational awareness data link
SWIL	software-in-the-loop
SUUV	Small Unmanned Underwater Vehicle
SV	Systems Viewpoint
TAI	tactical area of interest
U.S.	United States
US	undersea system
USV	unmanned surface vehicle
UUV	unmanned underwater vehicle
XLUUV	Extra Large Unmanned Underwater Vehicle

EXECUTIVE SUMMARY

The research and recommendations provided in this paper detail the current work in the seabed warfare mission area and present future work to define further the framework definition of seabed warfare as a new mission area in the Navy. Seabed warfare is a relatively new term for the United States Navy with limited, although rapidly increasing, research to find innovative and effective warfare solutions for the U.S. Navy. Due to its infancy, the term seabed warfare has yet to become a universally accepted definition. Currently, there is a need for the investigation of a concept of operations of performing seabed warfare operating in both offensive and defensive environments to complete kill chains without placing high value assets at risk. Therefore, the U.S. Navy has a need for preliminary development of an unmanned open mission architecture system capable of performing seabed warfare.

The Autonomous Undersea Vehicle Requirement for 2025 prepared by the Chief of Naval Operations Undersea Warfare Directorate defines missions that are current and projected for 2025. These missions include: intelligence, surveillance, and reconnaissance (ISR); antisubmarine warfare (ASW); anti-surface warfare (ASuW); strike; mine warfare; and naval special warfare (NSW) which will continue to be relevant in the future. Several new areas are also increasing in relevance and importance and may become critical to the U.S. Navy's success in 2025. One of these new areas, seabed warfare, will be vital for disabling, confusing, deceiving, and destroying future military targets according to the Chief of Naval Operations Undersea Warfare Directorate.

The purpose of this project was to assist in the development of a framework definition for seabed warfare, including defining an initial operational concept and a set of requirements for seabed warfare. This research developed an open mission architecture that shows the operational activities and systems associated with seabed warfare and analyzed the Extra-Large Unmanned Undersea Vehicle (XLUUV) for its potential utility as an enabler of seabed warfare. The XLUUV was a good candidate for this analysis due to its large payload and ability potentially to deploy seabed systems unmanned. Lastly, we developed an operational simulation to identify key performance drivers to assist in

creating requirements that are more informed. This project helps broaden the knowledge base of the seabed and the utilization of seabed systems with existing or future systems, as well as identify the critical system performance drivers for seabed warfare.

Team Leviathan defined seabed warfare as "operations that involve undersea networks and systems capable of operating on the seabed, interacting with seabed systems, and taking actions against other systems." While incorporating relevant missions identified by the Chief of Naval Operations Undersea Warfare Directorate, Team Leviathan instituted using a kill box to accomplish a number of those missions using seabed warfare. The project took the concept of a kill box, defined by Joint Publication 3–09 as a three-dimensional area used to facilitate the integration of coordinated joint weapons fire, and applying it where it has never been used: undersea.

The two new kill box types proposed in this project include a yellow kill box and an orange kill box. In yellow kill boxes, systems employed within the kill box area engage at targets entering the kill box environment, while orange kill boxes allow surface systems external to the kill box area to engage with indirect fire into the kill box. The team investigated only the yellow kill box concept in this project and four operational scenarios of increasing complexity. The team modeled, simulated, and analyzed results for learning more about seabed warfare performance and XLUUV utility within these kill box scenarios.

Operation Leviathan encompasses the four mission scenarios created, with the first three missions accomplishing the deployment of an undersea kill box and a final mission that would engage and strike a target entering the kill box in order to obtain a mission kill. The first mission, intelligence preparation of the operational environment (IPOE), involves sending an undersea system into an area of interest to gather environmental information to help assess the suitability of the area for a kill box. Intelligence, surveillance, and reconnaissance is the second mission, and involves sending an undersea system to the area approved during IPOE in order to deploy ISR systems to be affixed to the seafloor and sensing systems anchored to the seabed, both of which are linked to a central command station. The third mission, effects field delivery, involves sending an undersea system again, this time to deploy effect devices that can engage with a target of interest such as

mines, acoustic devices, electromagnetic devices, or even seabed mounted launch tubes. This completes the deployment of the kill box. The final mission, engage and strike, triggers the execution of the kill box with all deployed systems working together. If a system from an opposing force enters the kill box, whether undersea or on the surface, the kill box can detect and classify the system and while maintaining a track on the target, receive the command to strike the target with the available effects devices.

We used MATLAB to simulate the four missions for the kill box as a probabilistic model to keep this project unclassified and to allow for easy adaptation and modification to support different mission scenarios. The team based the data used within the mission analysis on unclassified data, general naval knowledge, and generic device information. Due to this, Team Leviathan based all conclusions made within the subsequent analysis on the specific mission scenario modeled within the design. We performed an analysis of alternatives (AoA) to analyze the XLUUV's utility within these models. This included three other UUV alternates: large-diameter UUV (LDUUV); medium-diameter UUV (MDUUV); and small UUV (SUUV). The analyzed kill box size ranged from 1–1500 nm^2. The team performed a design of experiments (DOE) to obtain a set of scenarios aimed to assess the performance of each undersea system and replicated each scenario 30 times in the simulation for statistical analysis.

The results of the analysis provided performance data for each UUV alternative and investigated the utility of each UUV regarding the overall kill box deployment and effective size of a kill box for each mission. This investigation showed that the XLUUV had, on average, a higher rate of kill box deployment success for kill boxes up to 250 nautical miles squared (nm^2). For the individual missions, researchers assumed that an 80% effectiveness was an acceptable risk and therefore set as a performance threshold. For the IPOE mission, the MDUUV was the only UUV to meet the 80% threshold, at 80% success for kill boxes up to 100 nm^2. For the ISR mission, the MDUUV, LDUUV, and XLUUV had an 80% success rate for a kill box up to 120 nm^2. The effects field deployment mission showed that the LDUUV and XLUUV met the 80% threshold for kill boxes up to 100 nm^2.

The results of the final mission showed that UUVs were not a significant part of mission success. The success of the previous three missions predicates the engage and

strike mission success. The success rate of 70% up to 1500 nm^2 assumes all effects fire at the target and that the target is in range of all effects.

The simulation provided insight into the kill box's performance within five of the seven capabilities of seabed warfare: anti-submarine warfare; anti-surface warfare; intelligence, surveillance, and reconnaissance; military deception; extensions to strike. While two capabilities are not modeled, electromagnetic warfare and mine warfare, they could potentially become one of the effects deployed in the mission. Results indicate that a kill box ranging between 0–250 nm^2 is the most optimal size given the capabilities modeled.

The seabed warfare model created for this project provides a foundation for future work by changing various parameters or adding new variables not modeled such as Integrated Logistics Support (ILS) or integrating tactical decision-making. As it stands with the model simulated, the XLUUV has potential to become a great asset to seabed warfare, but there are a few capability gaps identified throughout this project. If the XLUUV could be equipped with the same side-scan sonar payload capability as the MDUUV to conduct IPOE, it could become a full-mission solution for all missions. The ideal solution for both deployment and execution of the kill box would be an XLUUV with a split payload containing a side scan with full IPOE capability, ISR devices, and effects devices.

References

Chief of Naval Operations. 2016. Autonomous Undersea Vehicle Requirement for 2025. 2016. Washington, DC: Undersea Warfare Directorate. https://news.usni.org/wp-content/uploads/2016/03/18Feb16-Report-to-Congress-Autonomous-Undersea-Vehicle-Requirement-for-2025.pdf.

Joint Chiefs of Staff. 2014. *Joint Fire Support*. Joint Publication 3-09. Washington, DC: Joint Staff.

ACKNOWLEDGMENTS

The team would like to thank Professor Paul Beery, Professor Eugene Paulo, and RDML Richard Williams (Retired) for their guidance and mentorship throughout this project. The team would also like to thank the support from the NAVSEA laboratories, U.S. Naval War College, and Naval Postgraduate School for guidance, support, and input throughout this project.

To our families, friends, and other loved ones, we thank you for providing everlasting support and encouragement.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

The Autonomous Undersea Vehicle Requirement for 2025 prepared by the Chief of Naval Operations Undersea Warfare Directorate defines missions that are current and projected for 2025. Missions such as: intelligence, surveillance, and reconnaissance (ISR); antisubmarine warfare (ASW); anti-surface warfare (ASuW); strike; mine warfare; and naval special warfare (NSW) will all continue to be relevant in the future. These missions will evolve, expand, and push the limits of execution by performing at greater ranges with greater precision. Several new mission areas are increasing in relevance and importance and can become critical for the success of the navy in 2025. These new mission areas include counter-UUV warfare, electromagnetic maneuver warfare, non-lethal sea control, and seabed warfare. Most importantly, seabed infrastructures will be vital for disabling, confusing, deceiving, or destroying future military targets in this new seabed warfare mission (Chief of Naval Operations 2016).

Development of the seabed as a warfare area necessarily requires development of a definition for seabed warfare. Seabed warfare utilizes seabed systems or systems that interact with seabed systems in order to perform missions such as mine countermeasures, ASW, ASuW, ISR, military deception, and strike. The Naval Surface Warfare Center (NSWC) in Panama City, FL, describes seabed warfare as operations to, from, and across the seabed. Items of technological obstacles include unmanned delivery, homing effectors, non-lethal effectors, persistent sensor networks, and remote command and control. This suggests that seabed warfare may utilize a modular approach in which various vehicles or platforms deliver modular sensors, communication nodes, and seabed weapons. All of these systems are adaptable and configurable for different missions, environments, and tactical situations (Everhart 2017).

Figure 1 displays an overview of seabed warfare and the undersea-distributed network as defined by General Dynamics Mission Systems. Seabed sensors on the seabed monitor and perform surveillance tasks, complete challenging maritime missions, and

upload or change UUV mission plans to continue scanning the seabed with side scan sonar, which generates information about the adversary. Seabed systems also include docking and recharging stations that ensure maximum endurance for the UUVs. Buoys on the surface of the water relay information to the surface or collect information. All of these systems working together generate a successful implementation of seabed warfare (General Dynamics Mission Systems 2018).

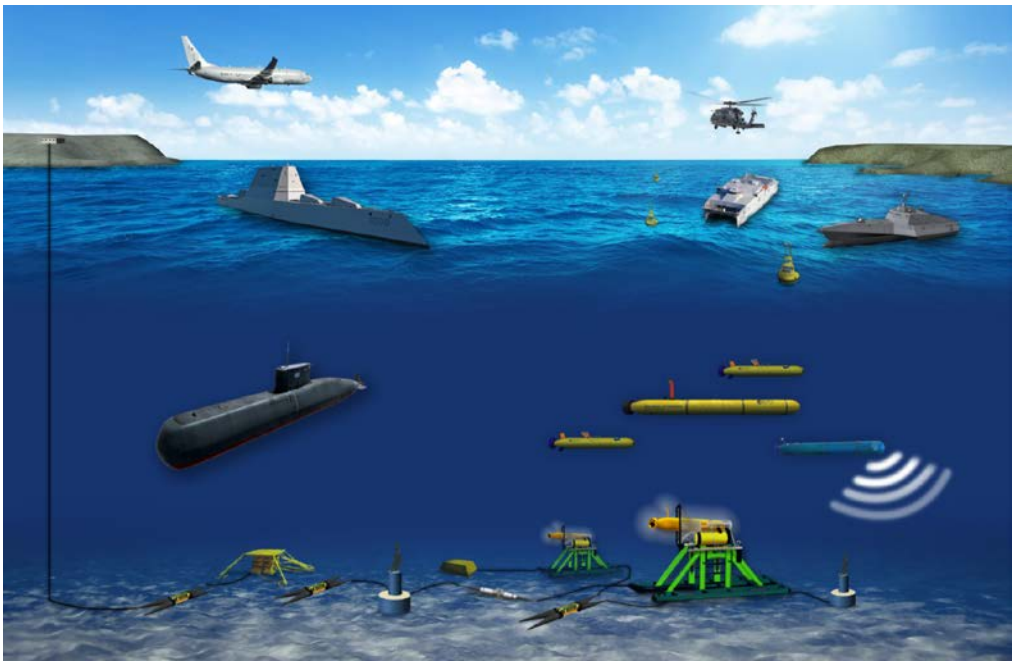


Figure 1. Seabed Warfare Overview. Source: General Dynamics Mission Systems (2018).

Another example of seabed warfare is the planned Underwater Great Wall, shown in Figure 2. The People's Liberation Army (PLA) Navy describes this as a network of stationary sensors on the ocean seabed that listen for enemy submarines and ASW efforts. Not only does the Great Wall contain seabed sensors, but it also consists of all active and passive sensors supporting UUVs and USVs that can autonomously locate and track enemy submarines. As shown in the glass reflection in Figure 2, the PLA Navy also has a large diameter UUV for long endurance missions, hauling large payloads, performing

surveillance, intelligence collection, mine countermeasures, and supporting other UUVs that act as a force multiplier for Chinese submarines (Lin and Singer 2016).

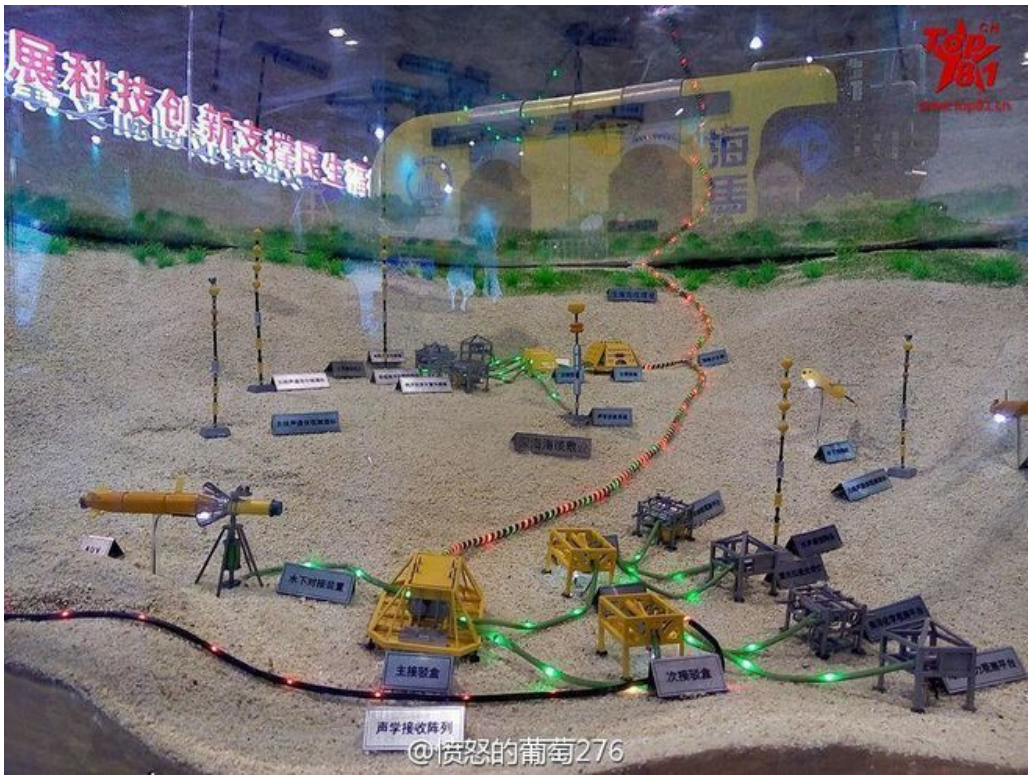


Figure 2. Underwater Great Wall. Source: Lin & Singer (2016).

Unmanned underwater vehicles and UUV technologies are a major area of interest for seabed warfare, but other technologies, such as undersea communication cables, are also relevant to seabed warfare. Countries depend on the undersea cables for implementing national security, coordinating military operations, and conducting intelligence. Voice and internet traffic passes through over 300 fiber-optic cables across the ocean, mounted along the seabed (Clark 2016). This traffic includes military and financial transactions at both the classified and unclassified level (Clark 2016). Improvements with UUV technology and the ocean-floor survey equipment make finding cables faster and easier for adversaries. In a crisis, adversaries can coordinate attacks on undersea cables and deny the military from access to sensor information and intelligence data. Unmanned underwater vehicles are advancing their technology in autonomy and the undersea imaging capability to gain

autonomously more insight on how adversaries damage or attack undersea cables and to deploy payloads on the seafloor successfully to act against enemies (Clark, 2016).

Seabed systems can also improve the limitations of undersea communications and UUV endurance. For example, the Forward Deployed Energy and Communications Outpost (FDECO) is a seabed system that provides the capability for an unmanned vehicle to download data and recharge batteries, enabling UUVs to conduct sustained operations underwater (Clark 2016). Other portable seabed systems, such as the Persistent Water Surveillance system, move to places such as choke points where adversary UUVs are attempting to travel (Clark 2016). In addition, the undersea seabed system sensor network enables a network controlled from a manned submarine or other platform that can have autonomous surveillance or attack operations by multiple UUVs (Clark 2016). These deployed and fixed sensors have UUVs that support submarine capabilities during a conflict (Clark 2016).

Based on the current UUV technology pertaining to seabed systems, the Navy is capable of isolated analysis for the sensing, charging, and strike characteristics of individual systems that may play a role in the use of the seabed as a warfare area. These systems cross a broad range of existing warfare areas and the operational assessment of the performance of those systems align accordingly. Stakeholders such as the United States Naval War College, Office of Naval Research (ONR), Defense Advanced Research Projects Agency (DARPA), and NAVSEA Laboratories all have a definition of seabed warfare and have worked with projects that depend heavily on seabed systems. Before the seabed can be examined similarly to traditional warfare areas (surface, air, mine), there is a need to define an operational framework that captures the requirements and concept of operations for offensive and defensive use of the seabed. This definition will help the Navy identify all capabilities necessary to engage in seabed warfare properly.

B. PURPOSE

The purpose of this project was to assist in the development of a framework definition for seabed warfare. This effort included defining an initial operational concept and a set of requirements for seabed warfare. This project also defined an open mission

architecture that shows the operational activities and systems associated with seabed warfare. Of the associated systems, the team analyzed the XLUUV for its potential utility as an enabler of seabed warfare due to its large payload size, endurance, and seabed system deployment capability. Finally, the team developed an operational simulation with the intention of analyzing it to identify key performance drivers to assist in creating informed requirements for seabed warfare. This project helps broaden the knowledge base of the seabed, how to utilize it with existing or future systems, and identify the critical system performance drivers for seabed warfare.

C. TECHNICAL APPROACH

1. Systems Engineering Process

Team Leviathan chose a tailored system engineering waterfall model as the systems engineering process for this project. The team considered many of the different systems engineering processes available and decided that a waterfall model was an appropriate match for the project's scope, deliverables, and team size. Therefore, the team organized this project as a sequential process, completing each step before the next step starts. Figure 3 displays the waterfall model structure with tailored stages to meet the needs for the project. It depicts this project's steps from beginning to end with the product outputs of each step in the process.

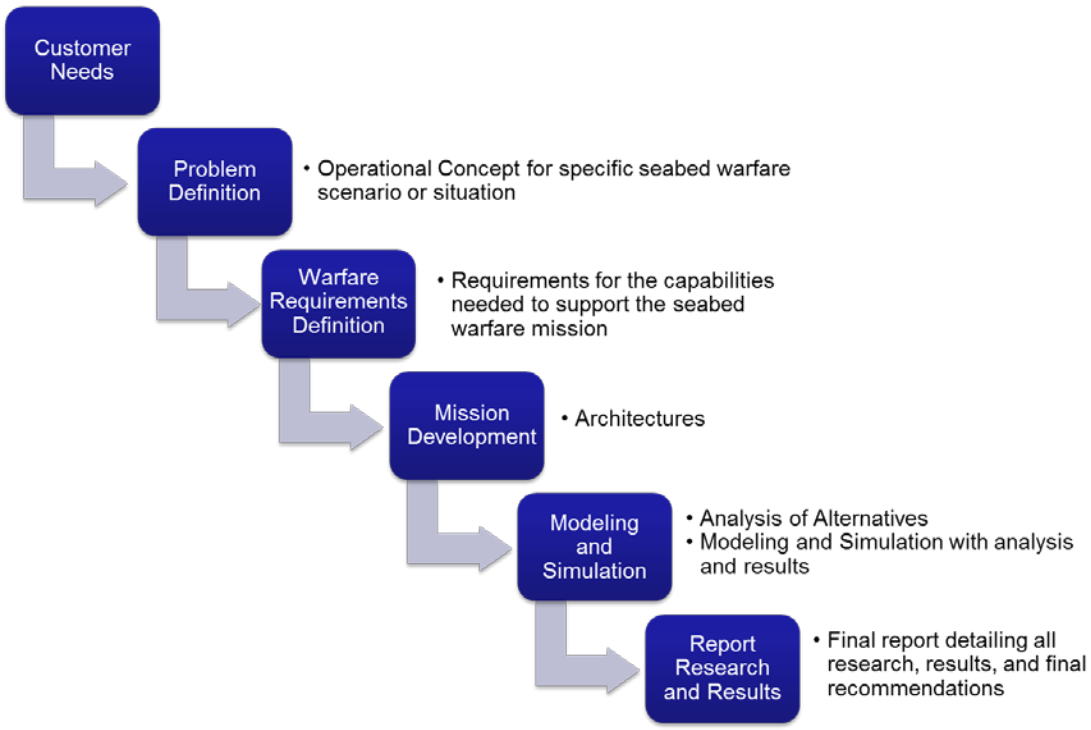


Figure 3. Waterfall Systems Engineering Process

The waterfall model began with analyzing the customer needs. This facilitated problem definition and served as the basis for the development of an operational concept for seabed warfare. Using this concept to define the seabed warfare problem led to the creation of the seabed warfare definition and requirements for the purpose of Project Leviathan. It is important that the requirements are clear, well written, and traceable to the customer needs. With the problem properly defined and requirements created, the team developed an architecture for the mission and performed an analysis of alternatives. Next, this mission architecture was modeled and simulated, from which the results informed the potential utility of the XLUUV in seabed warfare, based on the performance of an XLUUV versus other available technology.

2. Department of Defense Architecture Framework

Team Leviathan chose the Department of Defense Architecture Framework (DoDAF) to provide effective and efficient traceability and organization during the mission architecture creation. DoDAF is an overarching comprehensive architecture framework for

the DoD that presents visualization of requirements, capabilities, operations, and system functions. All major DoD procurements are required to document the system or mission architectures through the view products within DoDAF (Department of Defense n.d.). Utilizing the operation and capability viewpoints provided a structured approach to documenting the design of the seabed warfare mission architecture and requirements.

D. SUMMARY

Chapter I presented all relevant background information pertaining to Project Leviathan. This shows existing research and current capabilities identified up to now, as well as showing the current state of seabed warfare and the XLUUV. The purpose of Project Leviathan was to develop a framework definition for seabed warfare and analyze the potential utility of an XLUUV as it applies to seabed warfare. These parts formed the "why" of this project. The technical approach explained and showed the systems engineering process that we followed throughout the project. The systems engineering process showed the "how" and a plan for accomplishing and completing the purpose of the project. Chapter II covers the first two steps of the tailored waterfall process, customer needs, and problem definition.

THIS PAGE INTENTIONALLY LEFT BLANK

II. PROBLEM DEFINITION

A. PROBLEM BOUNDARIES

Recall that there is not a single accepted definition of seabed warfare. Accordingly, for the purposes of this project, this research team defined seabed warfare as the operations that involve undersea networks and systems capable of operating on the seabed, interacting with seabed systems, and taking actions against other systems. Additionally, the team integrated this definition with existing U.S. Army documentation and definition of a kill box, a term further discussed in Chapter III. In simple terms, a kill box is a three-dimensional area used to enable the integration of joint fires while reducing the coordination required from commanders to fulfill the mission (Army 2005). Additionally, the team focused on conducting an analysis of alternatives (AoA) that involved the various kill box scenarios with the XLUUV discussed in Chapter III. We assessed and compared the utility of the XLUUV within each scenario to other existing systems or devices.

B. PROBLEM ASSUMPTIONS AND CONSTRAINTS

This project includes several assumptions. In addition to the XLUUV, with a diameter between 84" and 120," there were other platforms available for use and assessment for AoA: a large-diameter UUV (LDUUV) with a diameter between 21" and 84," a medium-diameter UUV (MDUUV) with a diameter between 10" and 21," and a small UUV (SUUV) with a diameter between 3" and 10." The team assumed the different platforms have the capability of inter-platform underwater communication by using acoustic communications channels. In order to have mission flexibility and interoperability, the mission capability of the UUVs can change dependent on the type of payload carried.

This project also defined the following constraints. The team assessed the XLUUV's utility only for use within the seabed warfare kill box and limited exposure of high value assets within the engagement areas. The environment chosen for the missions was a GPS-denied environment (both surface as well as subsurface). The problem was further constrained by the seabed composition, depth, and acoustic properties. The

XLUUV, based on the current capabilities of Boeing's Echo Voyager, was constrained to a max depth of 3000 meters with a max speed of 4.5 knots and a range of 13,000 nautical miles. There are also environmental impacts that constrain the seabed sensors or UUV systems chosen. Since Team Leviathan is not designing a UUV for use within seabed warfare, the team used currently existing technology performance specifications as distributions. This includes power generation, energy storage, navigational accuracy, and acoustical communications capabilities.

C. PROBLEM SCOPE

With near peer adversaries placing strategic emphasis on asymmetric capability and capacity in both air and surface warfare, a challenge for the U.S. Navy is to maintain and expand its dominance of the undersea and seabed battlespace to offset the areas challenged by surface and air assets. Considering this, the project scope included defining a concept of operations (CONOPS) for seabed warfare in both offensive and defensive conditions and performing an operational analysis of seabed warfare to aid in the development of functional requirements and needs analysis. An AoA was necessary to determine if the XLUUV was useful, and evaluated its utility in four operational scenarios. With this analysis, Team Leviathan assessed the XLUUV's potential utility as an asset to seabed warfare.

D. STAKEHOLDER ANALYSIS

Key enablers identified by OPNAV N97 that allow the Navy to have enhanced and efficient capabilities includes endurance; autonomy and precision navigation; command, control, and communications; payload and sensors; and platform integration. These enhanced capabilities execute a family of UUVs for the anticipated tasks, with each UUV executing a task based on its size and weight. Submarines launching small and medium sized UUVs integrating with large and extra-large UUVs will serve as a complement for advanced mission sets (N97 2017).

The Office of Naval Research (ONR) has two programs focusing on mine and seabed warfare technology. The Advanced Undersea Weapon System (AUWS) demonstrates shallow water advanced mining capability as an offensive mining system.

The system consists of a weapon system, specifically an LDUUV, with an autonomy and fire control solution, a system for communication within the field as well as off board the system to an operations center, and sensors spread across the minefield. The concept of operations involves a sensor that would be able to specifically detect a target vessel and transmit a message acoustically to the LDUUV, which houses the weapon system. At that point, the LDUUV calculates a fire control solution a lightweight torpedo launches. The Modular Undersea Effector (MUSE) is a smart mine, which focuses on being a defensive/protective mine system and is modular. The effectors can be kinetic or non-kinetic in nature (Everhart 2017).

The Defense Advanced Research Projects Agency (DARPA) has many programs that deal with payload integration on the LDUUV and XLUUV platforms. A program called Hydra is integrating a distributed network of UUVs and UAVs with ISR and MCM payloads to operate for months at a time (Keller 2013). The Hydra program will demonstrate UUVs covertly brought into battle with payloads that can launch, dock, and recharge from the mothership (Keller 2013). DARPA also requested information from the industry about advanced payload delivery systems on UUVs, specifically XLUUVs with the Hunter Program (Keller 2017b). Boeing and Lockheed Martin will be designing the largest unmanned submersibles for long endurance surveillance missions or for undersea cargo vessels to deliver other sensors payloads or other UUVs (Keller 2017a).

Based on technology maturity, Table 1 identifies stakeholders with their needs, goals, and concerns for seabed warfare technology. Each of the stakeholders require the need to formulate their systems under a schema that aides them in operational tactics in a threat environment. The team addressed these needs in several ways, with a framework definition of seabed warfare to begin defining the capabilities and a tactically representative scenario to carry out those designed missions within the framework.

Table 1. Stakeholder Analysis Table

Stakeholder	Type	Need	Goals	Concerns
United States Combatant Commanders	User	Military dominance	Ensure that command and control for XLUUV and Seabed Warfare strategy is timely and accurate	Technology readiness level and maturity for transition, Budget and schedule, safety
Commander Submarine Forces	Requirements Generator	Military dominance Meets requirements	Develop requirements for systems to protect U.S. assets	Technology readiness level and maturity for transition
N97	Decision Maker	A capability of UUVs with maximum Endurance, Autonomy & Precision, high accuracy navigation, high quality Command, Control, & Communications, Payloads & Sensors Integration	Small and medium vehicles contribute to the undersea CONOPS. Large and Extra-large being developed to complement SSNs, not to replace (N97 2017). Protect U.S. assets	Budget and schedule
Future Naval Capabilities Office of Naval Research	Sponsor	Program success Meets requirements Payload maturity	Develop and transition XLUUV/Seabed Warfare technology to acquisition program within three-year timeframe	Technology readiness level and maturity for transition Budget and schedule
Defense Advanced Research Projects Agency (DARPA)	Sponsor	Program success Meets requirements	Develop XLUUV and Seabed Warfare Technology	Innovative Technology
PMS 406 under PEO LCS	Acquisition Agent	Program success Meets requirements Payload maturity	Ensure transition into acquisition	Technology readiness level and maturity for transition Budget and schedule
NAVSEA Laboratories Newport/Keyport/ Panama City	Integrator / Designer	System success Meets requirements	Assist the US Navy to acquire systems necessary for XLUUV/Seabed Warfare CONOPS through research and development and/or contracts	Budget and schedule

Overall, Team Leviathan defined seven high-level mission capabilities to support seabed warfare: anti-submarine warfare; anti-surface warfare; electromagnetic maneuver warfare; military deception; ISR, mine warfare, and strike extensions. The CV-2 displayed in Appendix A is a hierarchy of high-level capabilities within seabed warfare and the first

decomposed level. These capabilities developed and designed the operational activities within the kill box scenarios discussed in Chapter III.

E. PROBLEM STATEMENT

Currently, there is limited research in the investigation of a concept of operations of performing seabed warfare operating in both offensive and defensive environments to complete kill chains while potentially minimizing placing high value assets at risk. The United States (U.S.) Navy has a need for preliminary development of an unmanned open mission architecture system capable of performing seabed warfare.

F. SUMMARY

Chapter II covered the first two steps of the tailored waterfall process, the customer needs and problem definition. This showed the problem boundaries, assumptions, constraints, scope, and stakeholder analysis to bound and further detail the problem definition.

Chapter III goes into detail defining mission operational scenarios of seabed warfare. It also provides four scenarios that describe tactics involving battlespace transparency and execution of tactically relevant operations tied to seabed warfare capabilities. The team analyzed the missions and generated requirements for the capabilities needed to support the seabed warfare mission.

THIS PAGE INTENTIONALLY LEFT BLANK

III. WARFARE REQUIREMENTS DEFINITION

A. SEABED WARFARE DEFINITION

1. Operational Analysis

Naval operations encompass a great deal of information on tactics and naval strategies. In order to analyze how seabed warfare ties into the greater hierarchy of naval warfare, it is important to incorporate and isolate specific tactics and see how seabed warfare is applicable and what systems will be most suitable for the application of that tactic. For this analysis, Team Leviathan used kill box tactics typically used on surface-based targets.

The idea of a kill box originated with development of operational concepts for ambushes against surface targets to emphasize that systems not physically within the area can support operations. In the case of seabed warfare, U.S. Navy has not utilized kill box as a tactic. With recent developments in military and seabed-based technologies, the ability to create battlespace transparency and execute an ambush-based tactic or kill box has only recently become an actionable reality. Considering the potential capabilities provided by the use of seabed warfare for theater commanders, the kill box is a tactic that can provide a platform to show the employment of those seabed warfare capabilities in a controlled space predefined by commanders in the field.

The U.S. military defines a kill box as “a three-dimensional fire support coordinating measure (FSCM) used to facilitate the expeditious air-to-surface lethal attack of targets, which may be augmented by or integrated with surface-to-surface indirect fires” (Army 2005). The kill box has a goal to “reduce the coordination required to fulfill support requirements with maximum flexibility, while preventing fratricide” (Army 2005). The U.S. military typically employs this tactic on land-based or surface targets with air and surface effects. These kill boxes are typically referred to as a blue kill box for air engagement only, or a purple kill box for air and surface engagements. Figure 4 shows a purple kill box. This kill box is organized with FSCM to allow for permissive fires from

surface and air assets (Army 2005). Team Leviathan proposes two new seabed-based kill boxes, yellow and orange, for use by the U.S. Navy that specifically uses seabed assets.

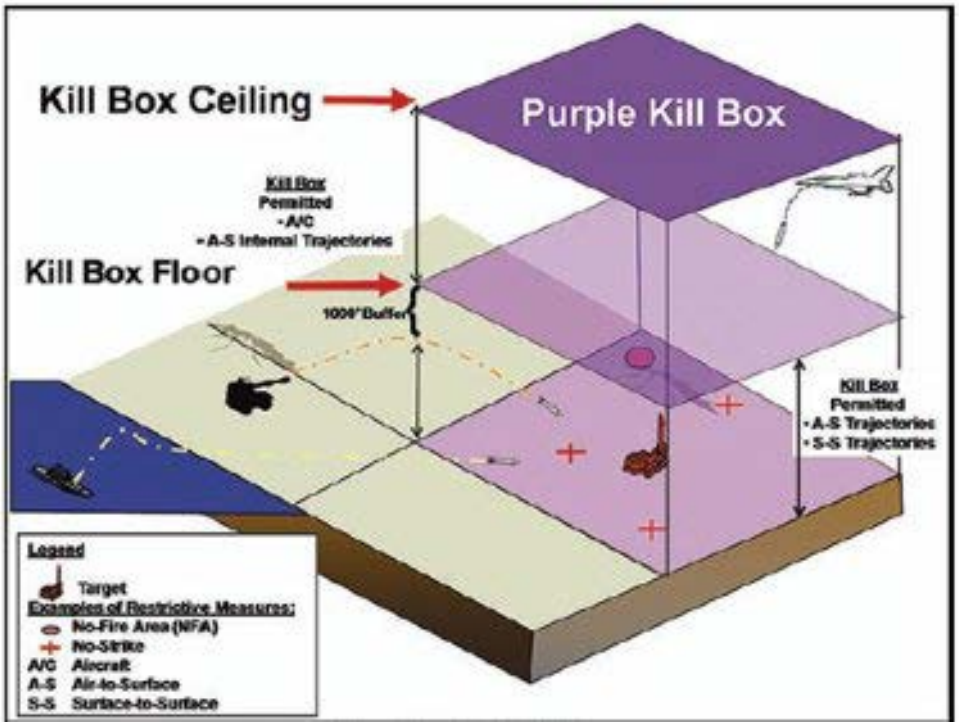


Figure 4. Multi-Service Tactics, Techniques, and Procedures (MTTP) Defined Purple Kill Box. Source: Army (2005).

The two proposed kill boxes follow the same methodology of blue and purple kill boxes from the seabed to the surface. Figure 5 shows the proposed yellow kill box. A yellow kill box permits seabed engagement in the undersea and surface environments without further required coordination with the establishing headquarters. For example, with a deployed yellow kill box, only systems employed in the kill box area can target the opposing forces entering the defined kill box environment. Systems external to the defined area can monitor the engagement but not apply effects into the target area.

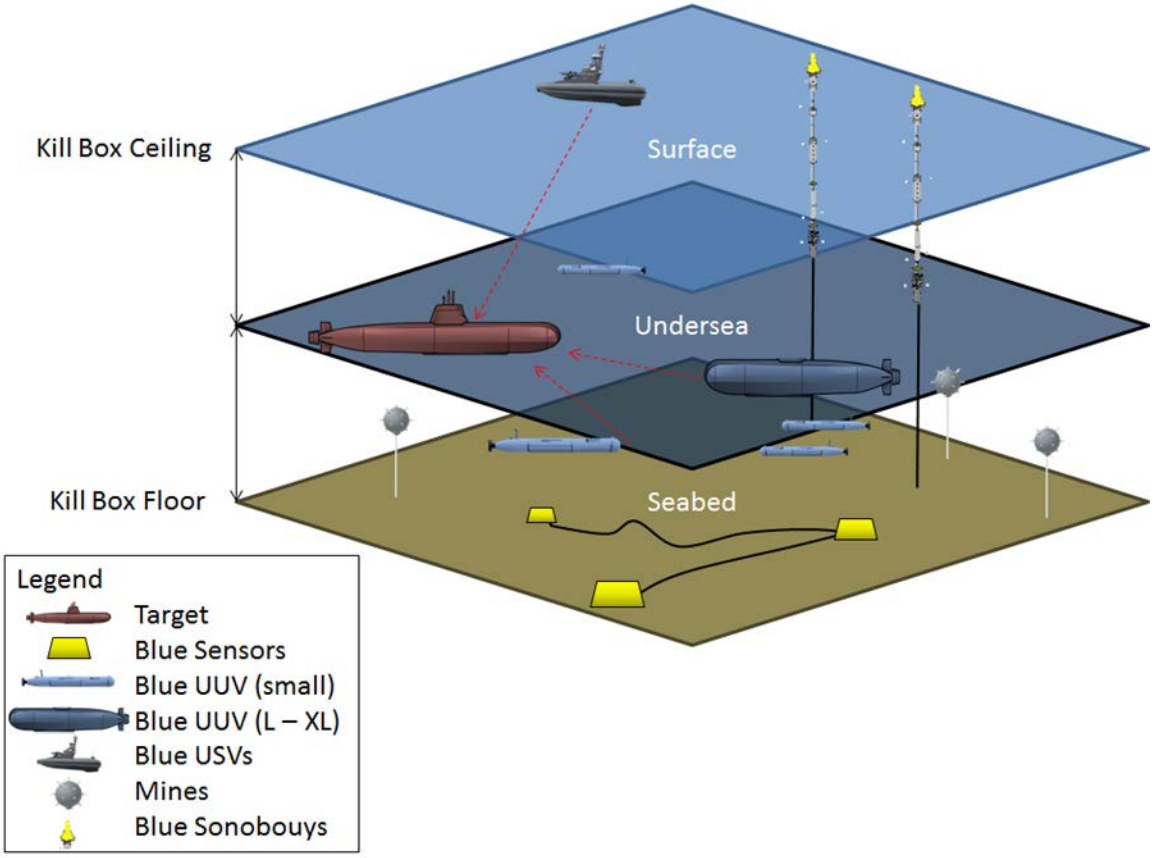


Figure 5. Proposed Yellow Kill Box

Figure 6 shows an orange kill box, which permits seabed engagement in the undersea and surface environments while also allowing surface to undersea indirect fires. In this orange kill box, surface systems external to the kill box area engage with indirect fire into the kill box without the possibility of friendly fire. While the goal of the kill box is to create an area that neutralizes any targets of interest in the area, it can also include no fire areas and sea space coordination with appropriate planning.

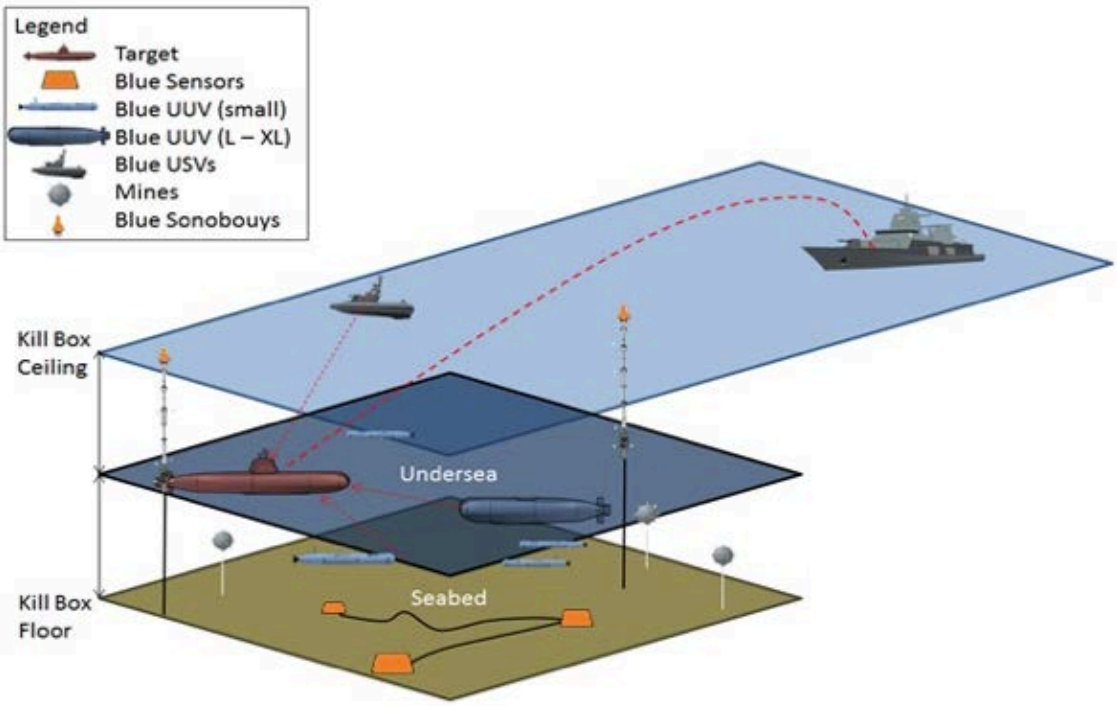


Figure 6. Proposed Orange Kill Box

Team Leviathan conducted a study investigating seabed warfare and the application of an XLUUV into the tactics and strategies. Operation Leviathan employed a simulated kill box made up of four individual missions that can be performed separately or in conjunction with each other to create scenarios that are more complex. Table 2 lists these scenarios.

Table 2. Operation Leviathan Mission Scenarios

Operation Leviathan			
	#	***Mission***	***Description***
Kill Box Deployment	1	Intelligence preparation of operational environment (IPOE)	Determination of FSCM attributes (priority, location, and time) of the tactical area of interest (TAI) will be developed
	2	Intelligence, Surveillance, and Reconnaissance (ISR)	Determination of target and effect priorities, target area clearance, risk assessment, and Situational Awareness Data Link (SADL)
	3	Effects Field Delivery	Integration and synchronization of maneuvers, fires, and determination of effects employment in the area of operation (AO)
Kill Box Execution	4	Engage and Strike	ID target entering the operational area, maintain track of the target, determine options, order engagement, strike, assess, report results

Figure 7 depicts an operational activity decomposition tree (OV-5a) of Operation Leviathan in the context of seabed warfare operations. Figure 7 decomposes the four missions presented in Table 2 into operational activities to maintain consistency with DoDAF standard terminology. These operational scenarios provide additional detail regarding each mission and to serve as guidance for the development of operational simulations.

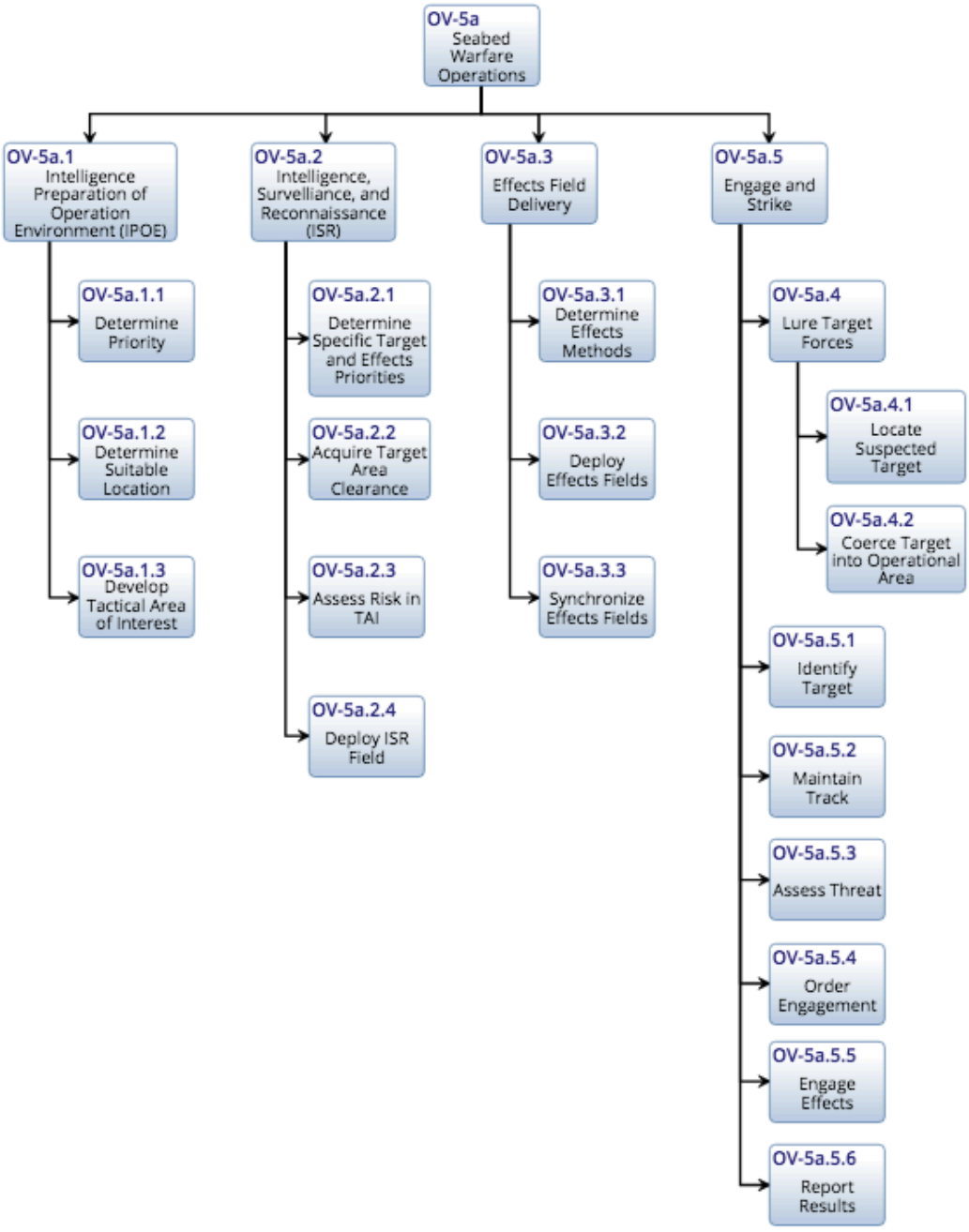


Figure 7. Operational Activity Decomposition Tree of Operation Leviathan

2. Concept of Operations

a) Scenario 1– Intelligence Preparation of the Operational Environment

Figure 8 focuses on an undersea system sent to an area of interest. Once the vehicle enters the area of interest, it begins the process of gathering intelligence about the environment. This information would include the seabed characteristics such as depth, surface topography, soil type, water and acoustic properties of the area. The vehicle sends this information back to a command for analysis to determine how effective the area is for further missions and possibly converting it into an undersea kill box.

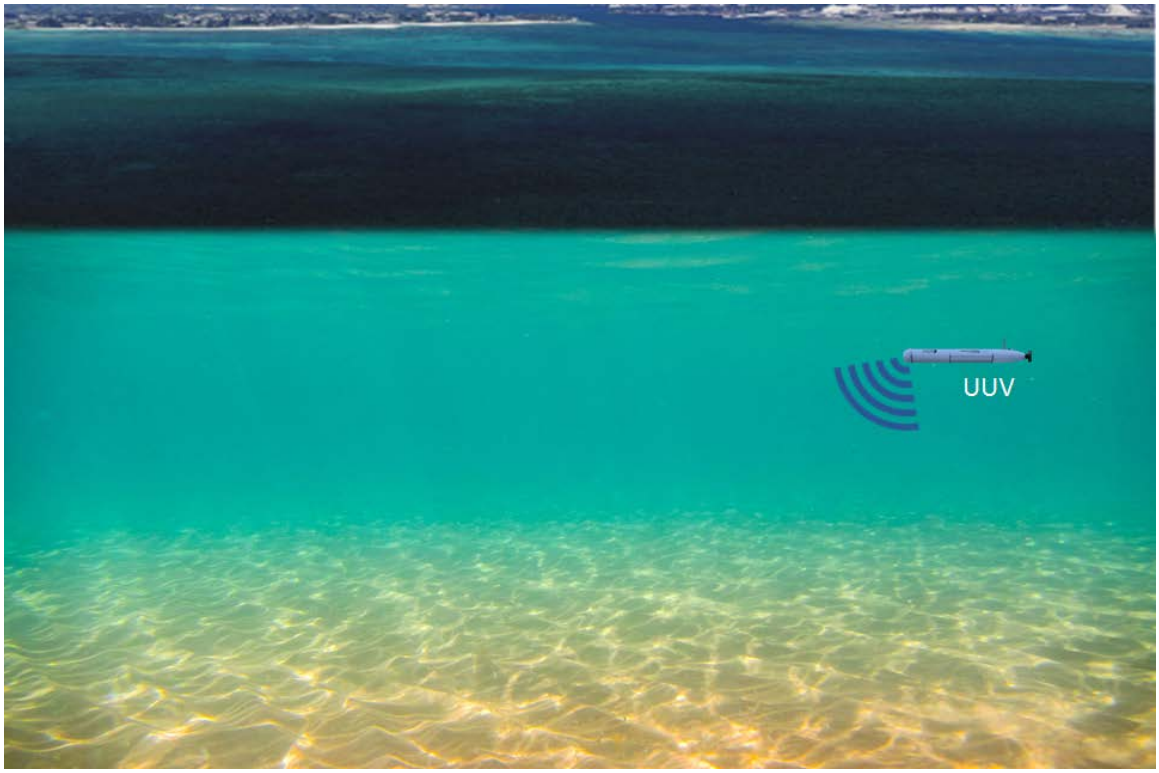


Figure 8. Scenario 1: IPOE

b) Scenario 2 – Intelligence, Surveillance, and Reconnaissance

After analyzing the IPOE data and approving the area of interest for further missions, there is a need to alter this area into a seabed warfare mission space. This includes installing ISR systems, represented by the seabed systems affixed to the sea floor and the

sensing systems anchored to the seabed. The central command station links to these systems. In addition to the permanent systems, one or more unmanned platforms may be patrolling the area to support ISR.

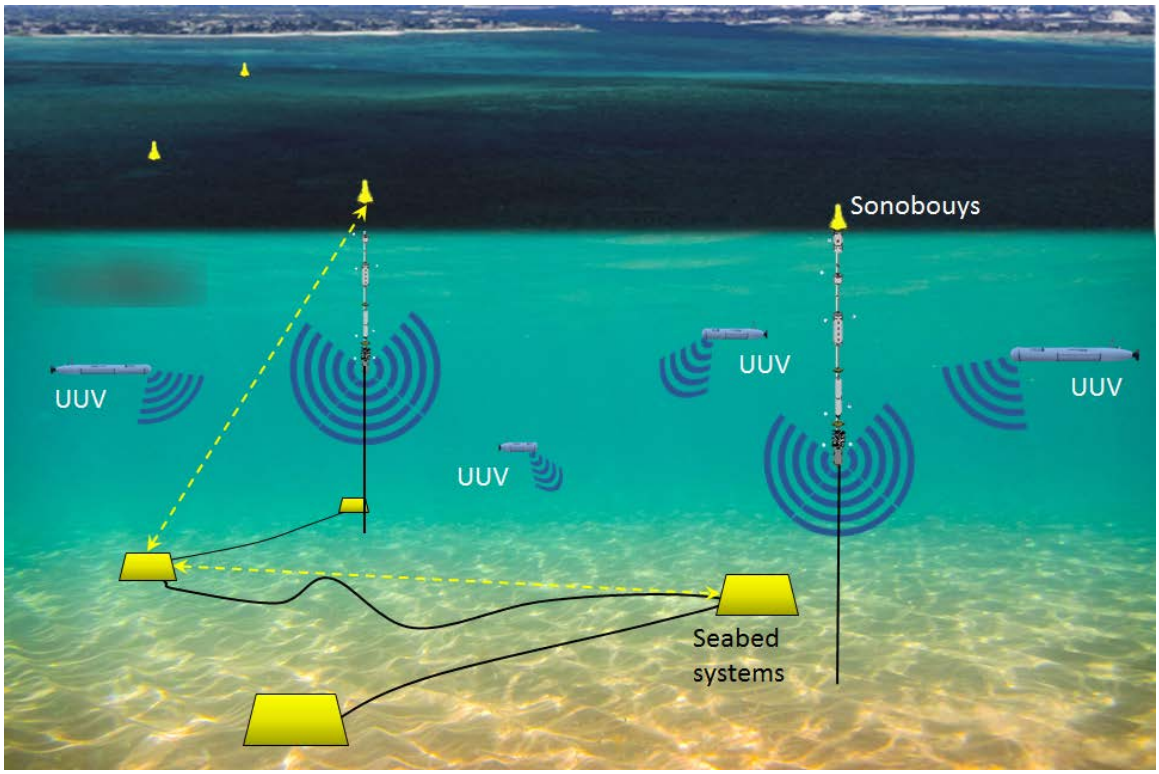


Figure 9. Scenario 2: ISR

c) Scenario 3 – Effects Field Delivery

Figure 10 displays the installation or activation of various effect devices that could engage with a target of interest. There are a variety of effects devices that could be used in this mission such as mines, acoustic devices, electromagnetic devices, or even seabed mounted launch tubes. For this scenario, Team Leviathan assumes that one or more undersea systems will be placing the effects device on the seabed to intercept opposing forces that may enter the kill box.

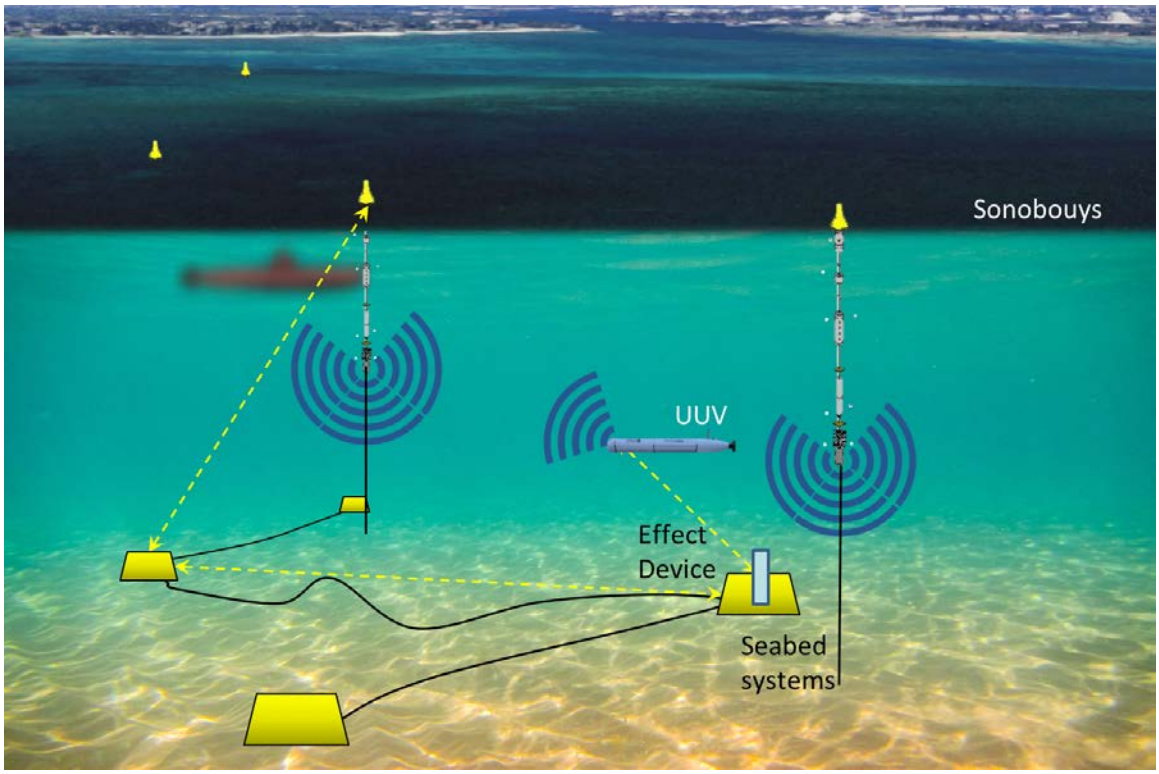


Figure 10. Scenario 3: Effects Field Delivery

d) Scenario 4– Engage and Strike

Figure 11 assumes that the target of interest has entered the kill box on its own or by coercion. Coercion may involve using the deployed systems to attempt to deceive the target and draw it closer to an "effects device" such as mines, acoustic devices, electromagnetic devices, or seabed mounted launch tubes. By this time, the seabed systems have communicated with each other and with any unmanned platforms in the area. Once the kill box system has verified the location of the target and proximity to an effects device, the command center issues a strike order for the target. This mission requires the successful coordination of all of the platforms in the kill box to aim and strike the target accurately and effectively. Once the systems have carried out the strike, ISR will reassess the target to determine its state and communicate with the command center while waiting for the next order.

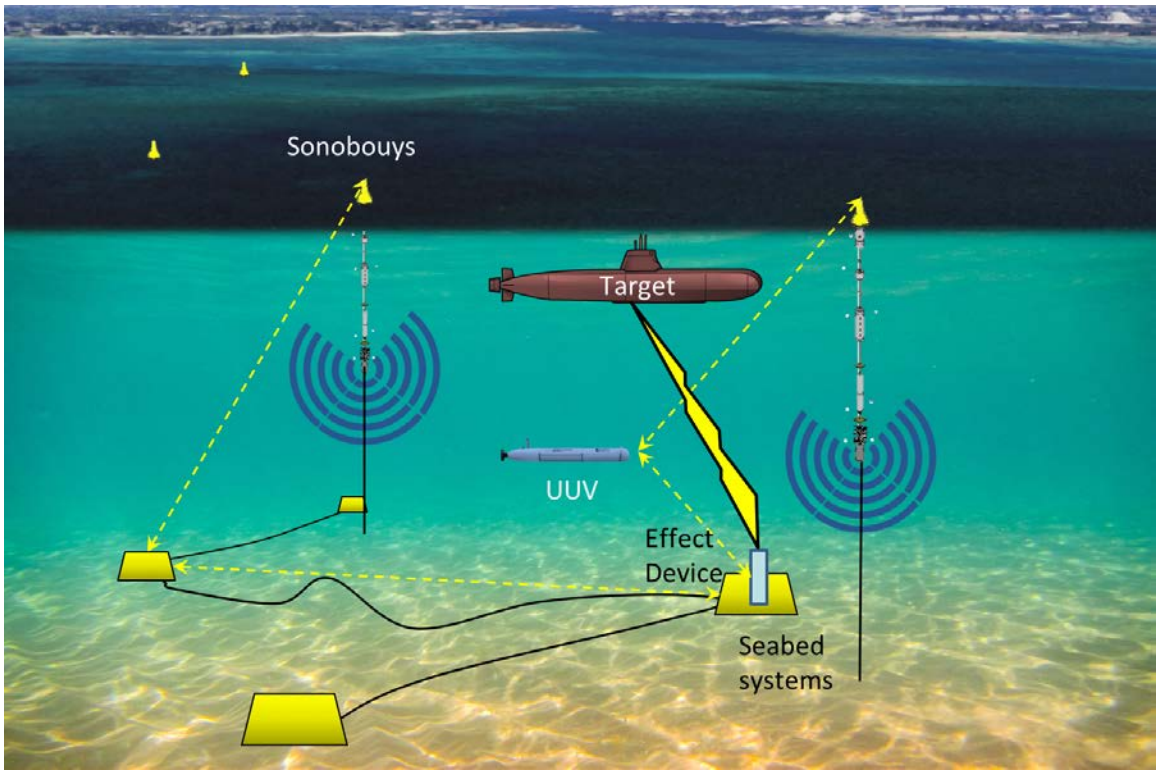


Figure 11. Scenario 4: Engage and Strike

3. Operational Activities to Capabilities Map (CV-6)

Team Leviathan developed a CV-6 diagram was developed to map operational activities to the previously stated CV-2. Through the course of this study, the team reviewed the CV-6 to maintain awareness of this mapping and ensure fidelity of the developed scenarios and application of each capability to an operational activity covered in Operation Leviathan. Table 3 displays the CV-6 matrix.

Table 3. CV-6 Capability to Operational Activities Mapping

CV-6 Capability to Operational Activities Mapping			Operational Activities																	
			IPOE			ISR				Effects Field Delivery			Engage and Strike							
			OV-5a.1.1 Determine Priority	OV-5a.1.2 Determine Suitable Location	OV-5a.1.3 Develop Tactical Area of Interest (TAI)	OV-5a.2.1 Determine Specific Target and Effects	OV-5a.2.2 Acquire Target Area Clearance	OV-5a.2.3 Assess Risk in	OV-5a.2.4 Deploy ISR Field	OV-5a.3.1 Determine Effects Methods	OV-5a.3.2 Deploy Effects Fields	OV-5a.3.3 Synchronize Effects Fields	OV-5a.4.1.1 Locate Suspected Target	OV-5a.4.1.2 Coerce Target into Operational Area	OV-5a.4.2 Identify Target	OV-5a.4.3 Maintain Track	OV-5a.4.4 Assess Threat	OV-5a.4.5 Order Engagement	OV-5a.4.6 Engage Effects	OV-5a.4.7 Report Results
Capability	ASW	CV-2.1.1 Submarine Tracking				X	X	X	X	X			X	X	X	X	X			
		CV-2.1.2 Submarine Identification & Classification				X	X	X	X	X			X	X	X	X	X			
		CV-2.1.3 Submarine Deterrence							X				X	X	X	X	X	X	X	X
		CV-2.1.4 Submarine Neutralization													X	X	X	X	X	X
		CV-2.1.5 Submarine Communication Disruption	X										X	X						
	EMW	CV-2.2.1 Platform Defense							X				X		X	X	X	X	X	X
		CV-2.2.2 Strike Countermeasures													X	X	X	X	X	X
		CV-2.2.3 Platform Neutralization													X	X	X	X	X	X
		CV-2.2.4 Communication Disruption											X	X				X	X	
	ISR	CV-2.3.1 Intelligence Gathering	X	X	X															X
		CV-2.3.2 Multi-system Data Analysis	X	X	X															
		CV-2.3.3 IPOE	X	X	X															
	Military Deception	CV-2.4.1 Adversarial Deterrence											X	X	X			X	X	
		CV-2.4.2 Information Interception											X	X						
	ASuW	CV-2.5.1 Surface Ship Tracking				X	X	X	X	X			X	X	X	X	X			
		CV-2.5.2 Surface Ship Identification & Classification				X	X	X	X	X			X	X	X	X	X			
		CV-2.5.3 Surface Ship Deterrence											X	X	X	X	X	X	X	X
		CV-2.5.4 Surface Ship Neutralization													X	X	X	X	X	X
		CV-2.5.5 Surface Ship Communication Disruption											X	X						
	MW	CV-2.6.1 Mine Countermeasures	X	X	X															
		CV-2.6.2 Mine Delivery	X	X	X						X	X								
	Strike	CV-2.7.1 Extend Strike Range								X	X	X								

B. MISSION CONTEXT

1. Mission Assumptions

For the kill box and its deployment, the team assumed full deployment success is only after the completion of IPOE, ISR, and the effects field delivery missions. Success in kill box execution is when engage and strike was successfully completed. The definition of overall success of Operation Leviathan is when the systems obtain mission kill on the primary target of interest.

2. Mission Constraints

The Operation Leviathan kill box size deployed during the mission was constrained by the performance of the payload sensor technology (both ISR and effects devices). The undersea system performance within the missions was constrained by the endurance of the vehicle and range it can travel.

3. Mission Boundaries

Operation Leviathan is a seabed-based kill box that is under the domain of seabed warfare. The kill box boundaries are limited to the definition of seabed warfare stated in Chapter I. In the case for Operation Leviathan, the operation began with the surveying of an area for suitability to deploy a kill box, which is an interaction with the seabed. The following mission of the operation involves the monitoring and prosecuting of other systems in the area. The third and final mission in the deployment phase involves expanding the capability of the seabed kill box with additional systems and effects. The extensions to engage and strike missions allow for the attack of other systems in the area, which is a seabed system that engaged an undersea or surface target. This matches the definition of seabed warfare. For this specific operation, any tactics that reach for things outside of the kill box are not within the mission boundary. Any system that is not a seabed system or interacts with the seabed system is also out of the mission boundary.

4. Mission Interfaces

The Operation Leviathan mission interfaces with both the command center and the environment. The undersea system collects environmental data such as depth, bottom composition, and salinity. This data is a determining factor for whether or not to deploy and operate the systems. For example, the salinity, bathymetry, and bottom composition all play a part in the ability for deploying an effect or an ISR device. The more adverse the elements are, the more difficult it will be for deploying the system. The command center interfaces with the mission by receiving and transmitting data, sending acknowledgements, making decisions, and verifying that all seabed systems are functional, linked, and communicating together.

C. REQUIREMENTS ANALYSIS

The requirements shown below in Table 4 indicate an initial set of operational requirements that will be necessary to execute the seabed warfare mission, Operation Leviathan, successfully. We derived these top-level mission requirements from the operational activities that trace back to the needs of the seabed warfare stakeholders and seabed warfare capabilities.

Table 4. Top Level Mission Requirements

Req. #	Requirement Text	OV-5a Traceability
1	The kill box shall be mountable to the seabed	OV-5a.2.4 Deploy ISR field; OV-5a.3.2 Deploy effects fields
2	The kill box shall be able to communicate to all supporting systems	OV-5a.4.7 Report Results; OV-5a.4.5 Order Engagement; OV-5a.3.3 Synchronize effects fields
3	The kill box shall be able to engage target within its range	OV-5a.4.6 Engage Effects
4	The kill box shall be able to detect, classify, and track objects in its area	OV-5a.5 Engage and Strike (OV-5a.4.2 and OV-5a.4.3)
5	The kill box shall be able to monitor its environment	OV-5a.1.2 Determine Suitable Location, OV-5a.1.3 Develop TAI
6	The kill box shall be able to disseminate its collected data in real time to a command center	OV-5a.4.7 Report Results; OV-5a.1.2 Determine Suitable Location; OV-5a.1.1 Determine Priority; OV-5a.3.1 Determine effects methods
7	The kill box shall be able to lure and coerce the target into the operational area.	OV-5a.4.1 Lure Target Forces
8	The kill box shall be able to receive sensing data	OV-5a.2.1 Determine specific target and effects priorities; OV-5a.2.3 Access risk in TAI; OV-5a.5.3 Assess threat.

D. WARFARE REQUIREMENTS DEFINITION SUMMARY

This chapter discussed the operational analysis and concept of operations for performing seabed warfare using a seabed-based kill box. The DoDAF framework ensured that the kill box's operational activities mapped to the capabilities that concern seabed warfare. Operation Leviathan consists of four missions. A mission to identify the characteristic of an area of interest, IPOE. The second mission to deploy a sensor suite in the area, ISR. The third mission to complete the deployment of the kill box, a series of effects devices deployed in the area. Lastly, the fourth and final mission, engage and strike, where the deployed kill box will engage targets in its range. The results of this analysis produced an initial series of requirements for a seabed-based system employing seabed warfare.

IV. MISSION DEVELOPMENT

A. MISSION ARCHITECTURE AND DESIGN

Chapter III, Section A.2 displayed the concept of operations for the four scenarios described in this report. These high-level operational concepts provided a potential application of seabed warfare. The following four sections will explain in more detail the activities taking place during these missions, the sequence of these activities, and the entities involved in the performance of these missions.

1. Mission 1: Intelligence Preparation of the Operational Environment

There are three primary operational activities executed in the IPOE mission shown in the OV-5a in Figure 7. The operational goals of the missions are to determine if the location is a priority for a kill box, to determine if the location is suitable to execute a kill box, and to develop tactical information of the tactical area of interest. The IPOE mission involves an undersea system that is sent to an area of interest to begin gathering intelligence about the environment and completing the IPOE operational activities. There are three entities involved in the performance of this mission: the command center, environment, and undersea system.

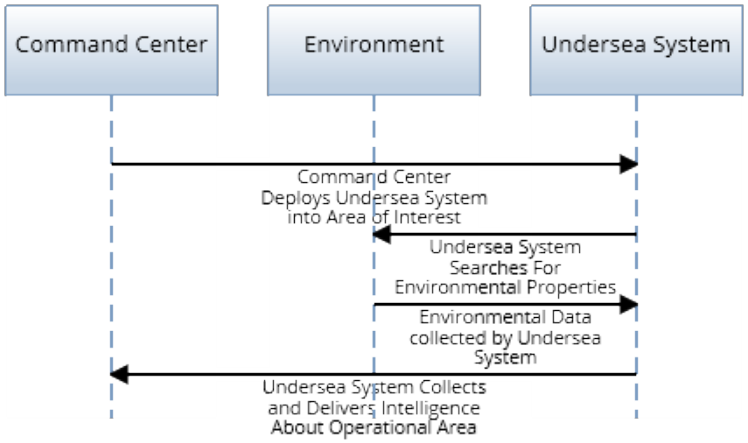


Figure 12. Mission 1: IPOE Sequence Diagram

The mission begins with the command center giving the order to deploy an undersea system into an area of interest. The deployed undersea system searches for and collects environmental properties of the seabed, including depth, surface topography, soil type, and acoustic properties. The undersea system delivers the collected data to the command center in preparation for the next mission, ISR.

2. Mission 2: Intelligence, Surveillance, and Reconnaissance

The second mission consists primarily of operational interactions between the seabed, undersea system, ISR, and seabed system. As traced back to the OV-5a, there are four primary efforts undertaken in mission two. The first effort determines the specific target and priority the command center executes when deploying the ISR system. The command center, based on the operational theater of war, will consider the suspected targets and deploy the necessary ISR systems. The command center also executes acquire target area clearance and assessing risk in the TAI while sending the command to the vehicles to deploy the ISR systems internal to its own system functions. Lastly, the undersea systems throughout the TAI execute the operation of ISR field deployment.

The ISR mission takes place if the environmental data collected during the IPOE mission returns a suitable area to continue. This mission involves deploying systems capable of ISR, as well as a suite of seabed sensors anchored to the seabed. Together, these systems form the boundaries of the effective sensing area of the kill box. The entities involved in this mission are the command center, environment, ISR system, seabed system, and undersea system.

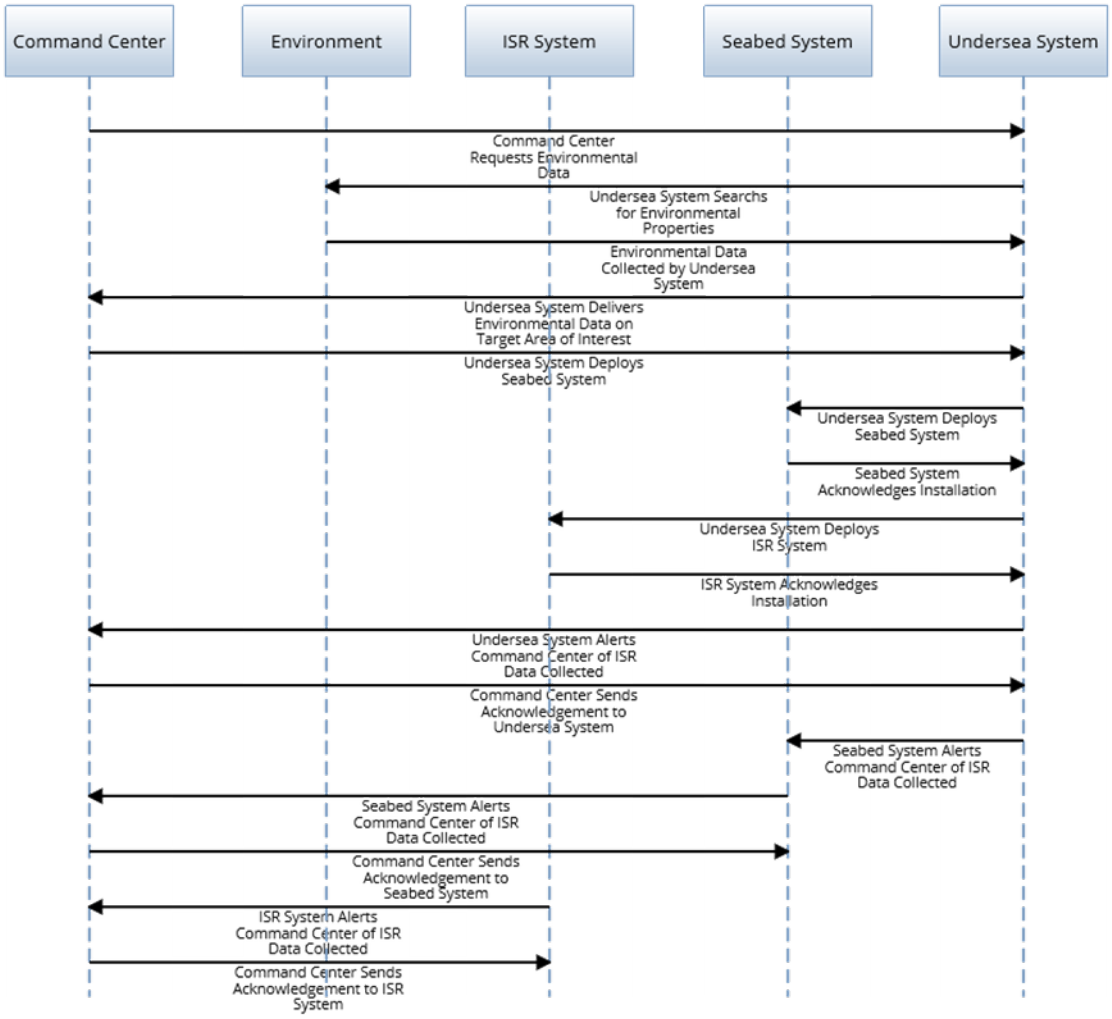


Figure 13. Mission 2: ISR Sequence Diagram

The ISR mission begins with a request from the command center to collect environmental data from the area of interest. The undersea system performs this task, collects the requested data, and delivers it back to the command center. Once complete, the command center decides whether it is suitable to deploy the seabed system. Assuming the environment is suitable, the undersea system then deploys the seabed system and verifies successful deployment. This is followed by the deployment and verification of the ISR system. The undersea system then alerts the command center of collected ISR data. The

command center acknowledges the receipt of the data. The command center also receives and acknowledges ISR data collected by the seabed system and ISR system.

The command center links to the ISR system, seabed systems, and undersea system. At this point, a target entering the kill box would trigger an alert to the command center through the deployed sensory components. In addition to the permanent systems, one or more unmanned platforms may be patrolling the area to support ISR. However, the kill box has no way of responding to a threat until the next mission is completed, effects field deployment.

3. Mission 3: Effects Field Deployment

Upon successful completion of the deployment of the ISR field, the third mission executes the three operational activities shown in the OV-5a in Figure 7. The undersea system deploys the effects devices to implement the effects field; with the specific devices chosen by the command center's operational need in the TAI. Lastly, the deployed effects fields synchronize and become an integrated effects field. This mission is what gives the kill box its ability to respond to threats that enter it. This mission also involves deploying effects devices into the kill box and installing the various effects devices that could engage a target of interest. There are many types of effects devices that can be used as discussed in the concept of operations and the devices used will depend highly on the goal of the kill box mission, whether target neutralization, communications disruption, or other types of mission kills. The entities involved in this mission are the command center, effects device, environment, ISR system, seabed system, and undersea system.

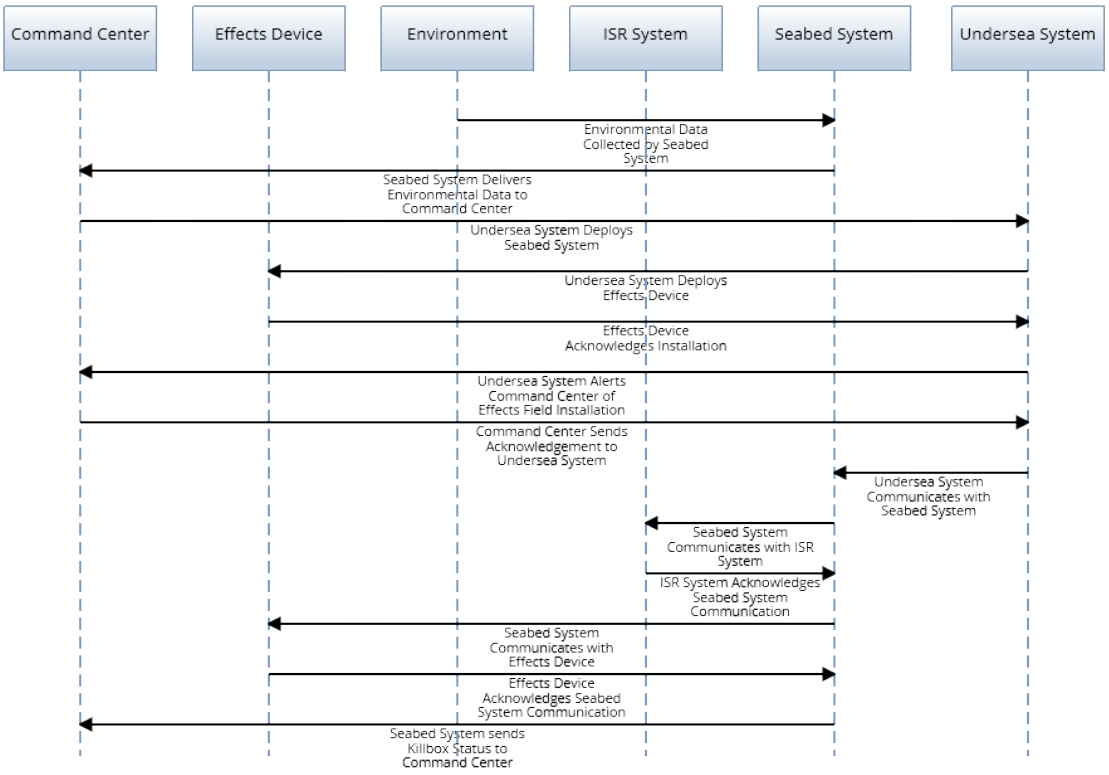


Figure 14. Mission 3: Effects Field Deployment Sequence Diagram

In order to begin the deployment of the effects field, the seabed system collects environmental data and sends the collected data to the command center. The command center then decides whether it is suitable to deploy the desired effects device. If environment is suitable, the undersea system deploys the effects device(s) and verifies successful deployment. The undersea system then communicates with the seabed system, and the seabed system communicates with both the ISR system and effects device. The seabed system receives an acknowledgement from those systems and once the command center receives verification that there is a working connection between all systems, the kill box is now fully deployed and functional.

With the kill box deployed and operating, the U.S. Navy is ready to handle potential threats. If a target enters the kill box and get detected, the U.S. Navy can take appropriate action. This is what the next mission is about, engage and strike.

4. Mission 4: Engage and Strike

The final step of Operation Leviathan is the most intricate and complicated of the four missions. In this mission there are seven primary operational activities executed by the full system of systems deployed on the seabed. Depending on the mission, offensive measures can utilize tactics such as target luring or the kill box could simply be protecting an area from incoming threats in a more defensive measure. With the kill box now operational at this stage, the system is able to wait and "listen" for potential threats. As systems lure targets, either externally or internally to the kill box, the ISR system will work towards identifying and then tracking the target. Once tracked, the command center will assess the threat and order the engagement. The effects field will release effects while the ISR system analyzes and reports the results to the command center. The process will continue until a successful mission kill is achieved or the targeted opposing force escapes. The entities involved in this mission are the command center, effects device, environment, seabed ISR system, target, and undersea system.

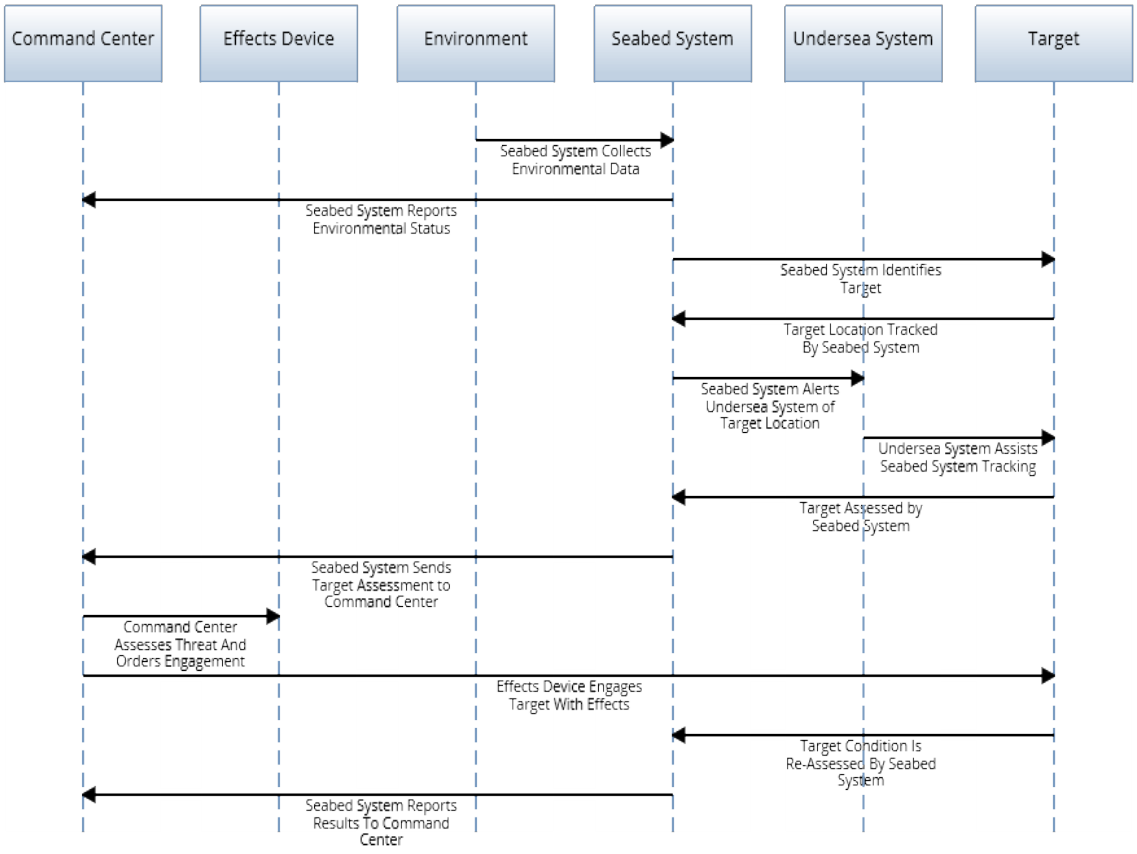


Figure 15. Mission 4: Engage and Strike Sequence Diagram

This mission begins with the kill box sensors listening, collecting environmental data, and reporting the data to the command center. A target then either enters the kill box through luring or on its own. The seabed system identifies the target and begins tracking the target. The target location is sent to the undersea system and the undersea system assists in tracking the target. The seabed ISR system classifies the target and information sent to the command center. The command center assesses the threat and orders engagement if appropriate. The effects device then engages the target with desired available effects. The seabed system assesses the target condition and sends the results to the command center.

Assuming there are enough effects resources available, the kill box can continue monitoring the environment and continue to assess threats as needed. For the purposes of this project, the mission ends here. However, there is potential to deploy additional effects

devices over time or add systems to the kill box for better performance depending on the ever-changing needs various missions.

B. MODELING AND SIMULATION

1. Modeling Approach

To perform the AoA between different undersea systems, specifically the utility of the XLUUV in each of the missions, the team used a combination of MATLAB and Excel to develop probabilistic simulations for each mission individually. Probabilistic or stochastic simulations rely on distributions and ranges, both discrete and continuous, to influence the simulation. The decision to proceed with the probabilistic models versus a tactical model such as hardware-in-the-loop (HWIL) or software-in-the-loop (SWIL) was to keep the project unclassified and to allow for easy adaptation and modification of all models to support different mission scenarios. The progression of data is shown in Figure 16. System and environment attributes along with mission criteria acted as inputs into the design of experiments (DOE). Using DOE, a set of scenarios, aimed to assess the performance of each undersea system, was constructed. Those scenarios were processed through the MATLAB simulation where pre-defined measures of performance (MOPs) and measures of effective (MOEs) were gathered. From these outputs, the team analyzed the the overall performance of each system.

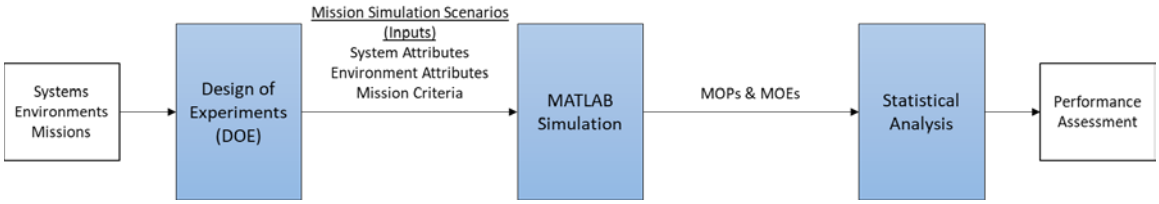


Figure 16. Simulation Data Flow

Team Leviathan designed the input tables within Excel to allow for easy modification without affecting the underlying structure of the simulation. This table contains critical attributes of the systems modeled. These systems include the ISR devices, the effects devices, the undersea systems, and the opposing forces target systems. There

are additional look-up tables that include crucial mission criteria and environmental factors that affect mission performance.

To simplify the computations performed and allow for modification and alterations to the mission and system attributes, the Excel look-up table combines multiple attributes into single values. To illustrate, simulating the ability of an undersea system to evade an adversary, system attributes such as speed, endurance, and acoustic signature are components that comprise the probability range that represents the undersea systems ability to evade successfully. The MATLAB simulation utilizes these probabilities for that mission or sequence of missions. The major advantage of this approach is the ability to modify the probabilities and the factors that they are comprised of with little or no impact on the actual MATLAB simulation. For instance, if a specific system of interest had known attributes, they could be entered into look up table so that the simulation could evaluate using that system, thus transitioning the probabilistic simulation into a deterministic simulation.

Only a successful mission kill of the opposing forces achieves overall success of Operation Leviathan. This does not necessarily have to be the full destruction of the threat.

2. Assumptions and Constraints

The decision to use probabilistic models as opposed to tactical models caused additional assumptions and constraints to be defined. To determine a deployment failure on the mission success it was necessary to develop ISR and effects field minimum resolutions that define the number of required devices for the mission to be successful. For the calculation of the field resolution, Team Leviathan presumed the kill box to be rectangular while ISR and effects systems are circular. To ensure no gaps occur within the kill box, the team halved the ISR and effects ranges so that the largest number of those ranges that can fit in the area of the simulated kill box is the required resolution for that mission. The equation below calculates the number of devices n required for the device field resolution for both ISR and effects devices. The kill box area A_{kb} and device circular area A_d are in squared nautical miles.

$$n = \frac{A_{kb}}{(A_d / 2)}$$

Using this methodology for resolution the simulation can ensure that no gaps occur within the kill box and no target can sneak through the kill box without an effect able to reach it. The required number of devices, combined with the environment, and listening/detection range of the systems involved, determines the standard number of systems that the undersea system(s) must deploy during the missions. The team also assumed that undersea systems would only deploy the number of subsystems required to achieve the resolution set within the mission criteria.

To reduce complexity, all simulated systems operated at their most efficient speed and maximum range. The size of the kill box was fixed and not dependent on undersea system capabilities to create smaller or larger kill boxes. For example, the XLUUV can travel a maximum of 6500 nautical miles at a speed of 2.5 knots. The simulation assumes that the XLUUV will travel at this speed during the entire mission. Additionally, the team assumed that all intelligence information gathered during a run of the IPOE mission remains constant throughout that mission and any other missions performed in sequence with that mission.

To reduce the complexity of the simulation and to allow for easy modification, the simulation used single value probabilities that could be comprised of multiple factors. Since there cannot be a human-in-the-loop, all tactical decisions made either by the command center or by the opposing forces are out of scope. Team Leviathan is composed of system, acoustic, modeling and simulation, and mechanical subject matter experts. Without any background in military tactical decision-making, it was not within the capability of the team to determine tactical decisions within the simulated model. The simulation presented herein is adaptable to any military based tactical, logistical, and administrative decision making as a future capability.

For the final mission, the opposing forces or targets were already within the kill box, either by luring or by random chance. If the target entered by luring, it was assumed the probability of detection would be higher due to the opposing force being on alert. In

the case of the targets identifying the blue forces the opposing forces undersea systems have a known value to be able to detect undersea systems, while there is limited data in the capability of surface vessels to detect the vehicles, the simulation will model it as half of the capability of undersea opposing forces. This is due to surface vessels being much louder at sea than undersea vessels, which make it harder to detect undersea systems.

3. Attribute Decomposition

The MATLAB simulation uses singular values comprised of multiple system attributes. In order to obtain those values for utilization by the simulation, Team Leviathan decomposed the major attributes down into the factors. The full decomposition of each attribute along with a matrix mapping each attribute back to the mission scenarios that use it are in Appendix B. It should be noted that these are not all of the factors that could be encompassed within each major system attribute or mission threshold. As the simulation is used within future research and matures, the decomposition and functional groupings can be further refined.

4. Simulation Logic

a) Mission 1: Intelligence Preparation of the Operational Environment

Chapter III discussed that the kill box simulation begins with the attempt to conduct the intelligence preparation of the tactical area of interest. The number of attempts that the undersea systems have to conduct the operational function of IPOE is an integer value based on multiple factors. For each undersea system, the number of IPOE attempts x per undersea system is determined by the equation below:

$$x = \frac{E * S}{R}$$

where E is the endurance of the undersea system, S is the survey speed, and R is the range of the undersea system.

The system continually loops through the simulation logic until either the system was successful in IPOE, all attempts are depleted, or the opposing forces have identified the system.

After each IPOE attempt, there is an opportunity for identification of the undersea system, meaning opposing forces can detect and classify the undersea system. If identification has occurred, subsequently causing a mission failure, the simulation determines if the undersea system was able to evade successfully. If the opposing forces detected the undersea system, the simulation will enter the "detection avoidance" logic. Within this logic, the system will forego the following IPOE attempt, if it had not been successful, to avoid being further detected. This will allow the system the ability to reattempt IPOE. This loop will continue until the exit criteria for the simulation is completed. Once it exits, simulation will determine the overall mission success and collect various system data shown in Table 5.

Table 5. IPOE Mission Criteria and Data Outputs

Mission Success Criteria	All undersea systems successfully conducted IPOE No systems were identified by opposing forces
MOPS & MOEs	# of successful IPOE missions # of undersea systems detected # of undersea systems detected & classified (identified) # of successful evasions given the undersea system was identified

b) Mission 2: Intelligence, Surveillance, and Reconnaissance

The second mission begins with an attempt to deploy an ISR device. The mission objective is to deploy enough systems to meet the field resolution set within the mission criteria. Similar to first mission, this mission is essentially a repetitive loop that continues until either the field resolution is met, all possible devices are deployed or a system is identified by the opposing forces.

After the simulation checks to see if the deployment had been successful, the simulation will check the undersea system and any previously deployed ISR devices for

detection and classification. Once again, both systems can be detected and classified. If opposing forces just detected the undersea system, the simulation enters the "detection avoidance" logic. Within this logic, the system will forego the following deployment attempt to avoid detection. If the opposing forces identified the undersea system, potentially causing a mission failure, the simulation determines if the undersea system was able to evade successfully. If unable to evade the ISR field successfully, it is also abandoned. If the undersea system successfully evades, this will allow the system the ability to continue with deployment of the field. If the opposing forces identified the ISR field, the undersea system will abandon the field to avoid being caught. The loop of ISR device deployment continues until the exit criteria is completed. Once the exit criteria for the simulation has been hit, the simulation determines the overall mission success and collect various system data shown in Table 6.

Table 6. ISR Mission Criteria and Data Outputs

Mission Success Criteria	ISR field resolution meets the required field resolution No systems were identified by opposing forces
MOPS & MOEs	# of successful ISR device deployments # of successful ISR field deployments # of undersea systems detected # of undersea systems detected & classified (identified) # of ISR devices detected & classified (identified) # of successful evasions given the undersea system was identified

c) Mission 3: Effects Field Deployment

The third mission begins with an attempt to deploy an effects device. The mission objective is to deploy enough systems to meet the field resolution set within the mission criteria. The logic within the mission flows similarly to the ISR deployment mission. Similar to Mission 2, this mission is essentially a large loop that continues until meeting the field resolution, deploying all possible devices or identifying a system by the opposing forces.

After the simulation checks to see if the deployment had been successful, the undersea system, ISR field and any previously deployed ISR devices will be assessed for detection and classification. Once again, the opposing forces have the opportunity to detect and classify all systems. If detection occurs, the simulation will enter the "detection avoidance" logic. Within this logic, the system foregoes the following deployment attempt to avoid detection. If the opposing force identified the undersea system, subsequently causing a mission failure, the simulation determines if the undersea system was able to evade successfully. If this happens, the undersea system will also abandon the ISR field and effects field. If the system successfully evades, the system will have the ability to continue with deployment of the field. Similarly, if the opposing forces detect either fields, the undersea system abandons the kill box deployment to avoid being caught. The loop of device deployment continues until the exit criteria is completed. Once the exit criteria for the simulation has been hit, the simulation determines the overall mission success and collect various system data shown in Table 7. If this mission was successful, along with the previous two missions, it signifies that the successful deployment of the kill box and that the system is ready to engage any targets that enter.

Table 7. Effects Field Mission Criteria and Data Outputs

Mission Success Criteria	Effects field resolution meets the required field resolution No systems were identified by opposing forces Kill box deployment success
MOPS & MOEs	# of successful Effects device deployments # of successful Effects field deployments # of undersea systems detected # of undersea systems detected & classified (identified) # of ISR fields detected & classified (identified) # of successful evasions given the undersea system was identified # of successful kill box deployments

d) Mission 4: Engage and Strike

The final mission is by far the most complex mission and is predicated on the success of the previous missions; meaning that the kill box has been successfully deployed.

Since there are two ways that the opposing forces can enter the kill box arena, one assumes that if the threat is lured in that the detectability of the system of systems that was deployed would have a higher detectability since the forces would be on alert.

The undersea system along with the two fields that it had deployed act as a system of systems. Together, they will detect, classify, and track the opposing threat. Once the kill box has a lock on a target, the systems will proceed with the strike and attempt to acquire the mission kill. While this chain of events is happening, there is parallel logic that determines if the target has identified any of the systems within the kill box. Similarly, the undersea system receives the opportunity to evade the compromised kill box. Table 8 shows all mission criteria and data outputs.

Table 8. Engage and Strike Mission Criteria and Data Outputs

Mission Success Criteria	Mission kill of target was obtained No systems were identified by opposing forces before mission kill was obtained Success kill box execution
MOPS & MOEs	# of successful target detection # of successful target classification given target detected # of successful target tracking given target detected & classified (identified) # of successful mission kills # of undersea systems identified # of ISR fields identified # of Effects fields identified # of successful evasions given the undersea system was identified & mission kill was not obtained # of successful kill box executions

C. MISSION DEVELOPMENT SUMMARY

This chapter discussed the sequencing of the events that occur within each mission and the modeling and simulation development process. Team Leviathan mapped the simulation logic directly back into the system event traces and linked to the operation decomposition tree. Team Leviathan used the simulation explained within this chapter to

conduct the AoA on the XLUUV's utility within seabed warfare operations. The next chapter examines the evolution and process of this step.

V. MISSION ANALYSIS

A. ANALYSIS METHODOLOGY

1. Data Definition and Collection

Before presenting the simulation results, it is useful to define some of the attributes used within the simulation. Due to the data classification, project timelines, and simulation limitations, the data used within the mission analysis was unclassified, general naval knowledge, and generic device information. Team Leviathan based the subsequent analysis only on the specific mission scenario modeled within the DOE.

The Excel sheet of attributes structures the input data for the simulation and translates system design characteristics, presented in the architecture products, to event probabilities used in the simulation. The Excel lookup spreadsheet contains data on the undersea system, ISR device, effects device, and opposing force. The undersea system contains data on the ability to detect, classify, and track the targeted opposing force; lethality; ability to evade; number of ISR devices; and number of effects devices. Additional data presented in the Excel sheet is information necessary about the system to fill in data for the necessary input fields. These fields include reliability, endurance, range, speed, signature, sonar capability, situational awareness capability, and maximum payload size. The reliability of each UUV was determined from either UUV system specifications or data on fielded systems. Vehicle specification sheets provided the diameter, endurance, range, speed, and depth.

Since current evasion capabilities for the assessed undersea systems do not exist, the ability to evade was set to zero across all vehicles. Similarly, there is not a current lethality capability within the undersea systems, thus the probability of lethality associated to each vehicle is set to zero. The ability for a XLUUV to detect the targeted submarines was set to a probability range of 0.85 and 0.9. This value increased slightly for the surface ships due to surface wake and higher levels of noise. The ability of the LDUUV to detect the opposing forces decreased slightly compared to the XLUUV, and the MDUUV decreased more due to the size limits with passive sensing technology. The SUUV does

not have sensing capability to detect, classify, or track the targeted opposing forces. For classification, we assumed that the XLUUV and LDUUV had the same classification algorithm onboard, and the MDUUV had limited classification capability due to the size limits with technology. The XLUUV had a higher capability to track the targeted opposing force due to the eight-knot sprint speed of the XLUUV. The number of ISR and effects devices that could fit into the payload of each UUV was determined based on the size of the payload space. The size of the ISR device used within the simulation is one foot in diameter and three feet in length. The size of the effects device is one foot in diameter and ten feet in length. Current technology does not exist within the assessed UUVs to identify that a threat is pursuing them. This capability will not be used in initial modeling, but can be executed later to identify the benefit of a capability of this kind.

Team Leviathan based the ISR device's ability to detect, classify, and track the targeted opposing force on probability data for a sea mounted ISR device. The ability for the ISR device to classify and track were estimated based on current technology. The ISR device deployment assumes parallel probability since the UUV deploys the device and then the ISR device completes the deployment. The reliability of the UUV and the ISR device reliability were multiplied to acquire the ISR device deployment probability. The team acquired the additional data on the ISR device from system specifications and knowledge about ISR sea-mounted devices.

The derivation of the effects device deployment probability was similar to the ISR device. The lethality of the effects device assumes a reduced probability of the device when launched from other platforms. The team acquired additional data on the effects device from system specifications and knowledge about effects devices.

The opposing forces utilized in the simulation were the surface ship and submarine; modeled after U.S. vessels. The ability for each force to detect the blue systems (undersea system, ISR device, and effects device) was based on the fact that the XLUUV has low detectability. The detectability of other UUVs was scaled down from the XLUUV since they are quieter. Since actual values of the opposing force detecting the ISR and effects device is unknown, the simulation utilized an estimated probability. The probabilities associated with the surface ship's ability to detect and classify UUVs assumes half of the

submarine's ability. Team Leviathan assumed high evasion probability for the submarine and surface ship since there are crews onboard.

Due to the difficulty in obtaining an accurate representation of an environmental model for the simulation, the simulation used the concept of an "isolated system." This means that the simulation operated independently without the consideration of the environmental factors known to exist within the real world.

2. Design of Experiments

Team Leviathan used DOE to evaluate the XLUUV's utility within the deployment and execution of the kill box. Using JMP, the various inputs were entered as factors to create a run set of mission scenarios for usage within the simulation. Since the probabilities and values change among the undersea systems, the team created separate DOE scenarios for each system using a base design. Since the data is a mixture of continuous and categorical inputs that can variate per mission, JMP used a space filling design to generate the scenarios. Space-filling designs are "useful in situations where run-to-run variability is of far less concern than the form of the model" (SAS Institute 2018). It allows the ability to maximize the distance clustering and spaces the points in each factor uniformly to ensure full coverage of each factor.

To evaluate the utility of the XLUUV, the team designed eight sets of DOE scenarios using the four undersea systems and two types of opposing forces, surface and sub-surface. Table 9 and Table 10 show the factors and responses used to create the DOE mission scenarios. Within Table 9, the asterisks denote the values that were dependent on the undersea system or target type. Appendix B presents the connections between these variables and system characteristics.

Table 9. DOE Design Factors

Factor	Type	Values
Kill Box Size	Continuous	1-1500
P_{US}(Detection)	Continuous	*
P_{US}(Classification)	Continuous	*
P_{US}(Tracking)	Continuous	*
P_{US}(Lethality)	Continuous	*
P_{US}(Evasion)	Continuous	*
P_{US}(IPOE)	Continuous	*
P_{US}(Deployment ISR)	Continuous	*
P_{US}(Deployment EF)	Continuous	*
P_{OPFOR}(Detection US)	Continuous	*
P_{OPFOR}(Classification US)	Continuous	*
P_{ISR} (Detection)	Continuous	*
P_{ISR}(Classification)	Continuous	*
P_{ISR}(Tracking)	Continuous	*
P_{OPFOR}(Detection ISR)	Continuous	*
P_{ISR}(Tracking)	Continuous	*
P_{OPFOR}(Detection ISR)	Continuous	*
P_{OPFOR}(Classification ISR)	Continuous	*
P_{EF}(Lethality)	Continuous	*
P_{OPFOR}(Detection EF)	Continuous	*
P_{OPFOR}(Classification EF)	Continuous	*
Target Lured	Categorical	TRUE FALSE
P_{TARGET}(Detection US)	Continuous	*
P_{TARGET}(Detection ISR)	Continuous	*
P_{TARGET}(Detection EF)	Continuous	*
P_{TARGET}(Classification US)	Continuous	*
P_{TARGET}(Classification ISR)	Continuous	*
P_{TARGET}(Classification EF)	Continuous	*
P_{TARGET}(Evasion)	Continuous	*

Table 10. DOE Design Responses

Response	Goal
Kill Box Type	NA
Undersea System Type	NA
Number of Undersea Systems	NA
IPOE Mission Success	Maximize
Number of Attempts Used in IPOE	Minimize
Number of Detections During IPOE	Minimize
Number of Systems Identified During IPOE	Minimize
Number of Systems Evaded During IPOE	Maximize
ISR Mission Success	Maximize
Number of Successful Deployments During ISR	Maximize
Number of Detections During ISR	Minimize
Number of Systems Identified During ISR	Minimize
Number of Systems Evaded During ISR	Maximize
EF Mission Success	Maximize
Number of Successful Deployments During EF	Maximize
Number of Detections During EF	Minimize
Number of Systems Identified During EF	Minimize
Number of Systems Evaded During EF	Maximize
Strike Success	Maximize
Number of Times Target Tracked by Kill box	Maximize
Undersea System Identified During Strike	Minimize
Undersea System Evaded During Strike	Maximize
Number of Systems Identified During ISR	Minimize
Number of Systems Evaded During ISR	Maximize
EF Mission Success	Maximize
Number of Successful Deployments During EF	Maximize

B. MISSION ANALYSIS

1. Intelligence Preparation of the Operational Environment

The IPOE mission is the first of the three missions required to complete for successful kill box deployment. This mission involves sending an undersea system into a tactical area of interest to gather data on the operational environment.

Table 11 shows the factors used in this mission. The asterisks within the value column denote the factors that variate dependent on the undersea system evaluated.

Table 11. IPOE Mission Factors

Factor	Values
Kill Box Size	1-1500
P_{US}(IPOE)	*
P_{OPFOR}(Detection US)	*
P_{OPFOR}(Classification US)	*
Number of Undersea Systems	*
Number of IPOE Attempts	*

In order to complete the IPOE mission, the UUV needs to survey the entire kill box size. Subsequently, the amount of area a single undersea system can cover may not be enough to survey the entire operational environment. For this reason, to sufficiently survey the tactical area, some of the undersea system types required multiple systems. The number of IPOE attempts shows the maximum number of attempts that each undersea system has available to complete the mission. If the undersea system fails to successfully complete the IPOE mission, the simulation allows the system to reattempt. From the mission, the simulation collected certain MOEs in order to assess performance, shown by order of importance below in Table 12.

Table 12. IPOE Mission Responses

Response
IPOE Mission Success
Likelihood of Identification During IPOE
Number of Attempts Used in IPOE
Number of Detections During IPOE

Team Leviathan conducted trend analysis to evaluate all mission factors against the response variables. In the scatterplot shown in Figure 17, it appears that the kill box size is a major driver for the mission success. All systems decline in mission success as the kill box size increases, with the XLUUV more loosely correlated than the other UUVs. At a high level, the only kill box size that is possible for all four UUVs is the small kill box, which covers a 1 to 499 nm^2 area. The XLUUV and MDUUV are able to have some success with medium and large kill boxes; investigated later within the analysis of IPOE. Additional scatterplots are included in Appendix D; however, the team developed no actionable analysis based on these figures.

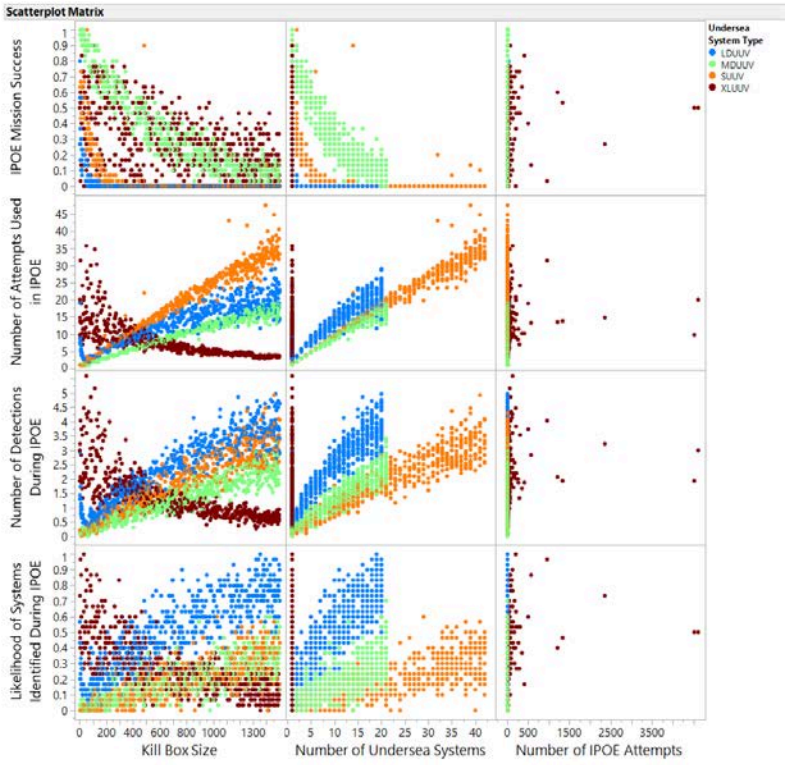


Figure 17. IPOE Mission Scatterplot

Looking across the responses against the kill box size, the spread of the data increases as the kill box size increases for the number of attempts used in IPOE to the number of detections during IPOE to the likelihood of system identification during IPOE. This is because each attempt gives the opposing force another opportunity to detect, and it follows that a detected system gives the opposing force another opportunity to identify the UUV performing the mission. In addition, all systems except the XLUUV follow positive non-linear trends as the kill box size increases. The XLUUV has a large enough endurance that the mission deployed only a singular system even for the largest kill box. Thus, the number of attempts available for the XLUUV decreases as the size of the kill box increases. The smaller kill box allows the system to have the ability to attempt IPOE several times even when the probability of completing IPOE is near zero for a given run. All other systems modeled during the AoA usually had multiples deployed for the mission, each with their own independent number of attempts and success rates.

Outside of the kill box size factor, the opposing force's capabilities to detect and classify the undersea systems have little effect on the response, as the factors are more or less constant through the dataset. As expected, the number of IPOE attempts, detections and likelihood of identification all increase as the number of undersea systems deployed increases. IPOE mission success declines as the number of systems increases but this is most likely due to smaller kill box sizes requiring less undersea systems to survey for IPOE.

The probability of IPOE, P_{US}(IPOE), and the number of IPOE attempts needs to be analyzed in detail in order to see if there any impacts. One note is that XLUUV has a few outliers that go into the thousands for the number of IPOE attempts. These cases are due to a very small kill box, approximately 1 nm^2 in area. Next, the distributions of the different UUVs will be assessed.

Shown in Figure 18 are the distributions of the different responses, based on the full range of the kill box, for the XLUUV. The average probability of mission success is about 28%. The XLUVV typically attempts IPOE 7.58 times on average while being detected 1.35 times. The likelihood of identification of the XLUUV by an opposing force is about 27%. There are potentially interesting outliers on all responses except IPOE mission success. This is again due to the times when the XLUUV is performing IPOE on a very small kill box. Since XLUUV endurance is very large but the probability of IPOE is small, the XLUUV had many attempts at performing IPOE and failing, with each attempt giving the opposing force a chance to detect and classify the XLUUV. Additionally, the spread of the interquartile range across the IPOE mission success response is (0.13, 0.43), with the densest amount of data found within this range. All response variable data is negatively skewed, but with a heavy left tail due to the wide spread of data regardless of kill box size. An interesting thing to note here is that number of attempts, detections, and likelihood of being identified trend downward as the kill box size increases. For the XLUUV, this data trends in the opposite direction of all other UUVs again due to kill box size and the effect it has on the number of attempts for an XLUUV. Comparing the XLUUV endurance to the other three UUVs, the XLUUV's endurance is about 88 times that of single LDUUV or MDUUV, and 180 times more than an SUUV.

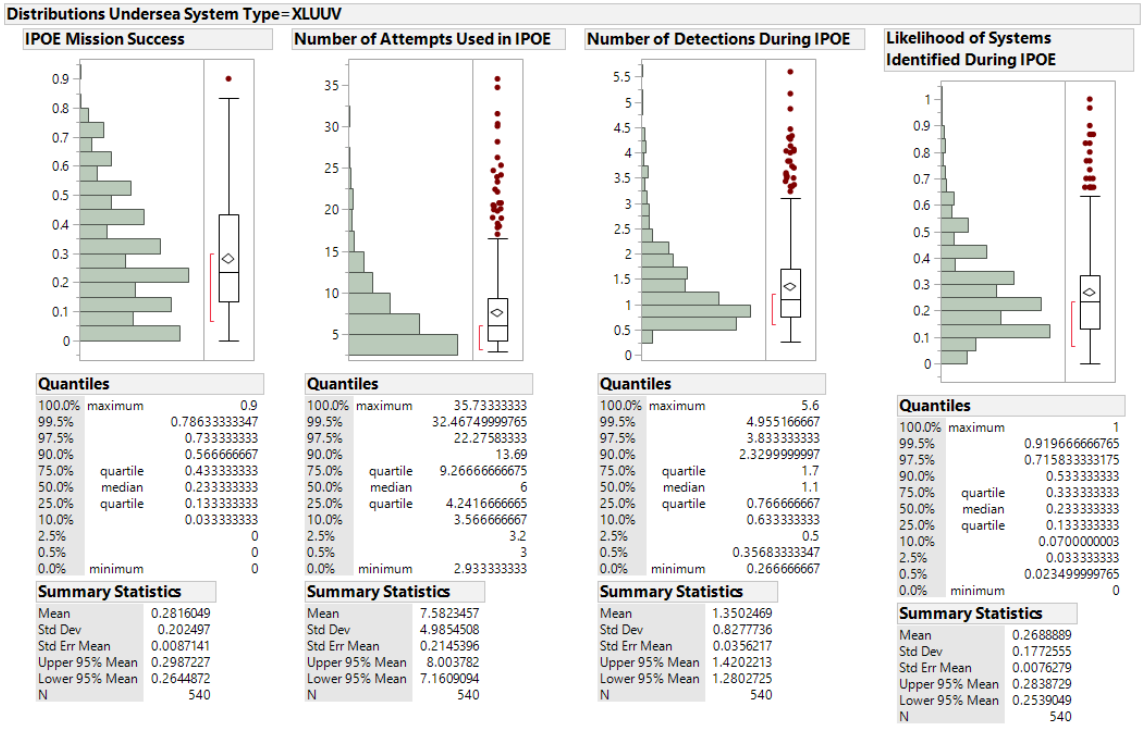


Figure 18. XLUUV Response Distribution for IPOE

Shown in Figure 19 are the distributions of the different responses for the LDUUV. The probability of mission success falls much lower, about 1%. The LDUUV attempts IPOE 13.08 times while being detected about 2.44 times on average. The likelihood of identification for the LDUUV is about 51%. For the LDUUV, only the IPOE mission success had potentially interesting outliers. Team Leviathan attributes these outliers to the fact that the LDUUV has some success at very small kill box sizes of $100nm^2$ but anything above this range was mostly 0%. The spread of the data across all other variables is moderately wide with the density of the data trending positively, though the metrics for these other variables are desired to be lower rather than higher.

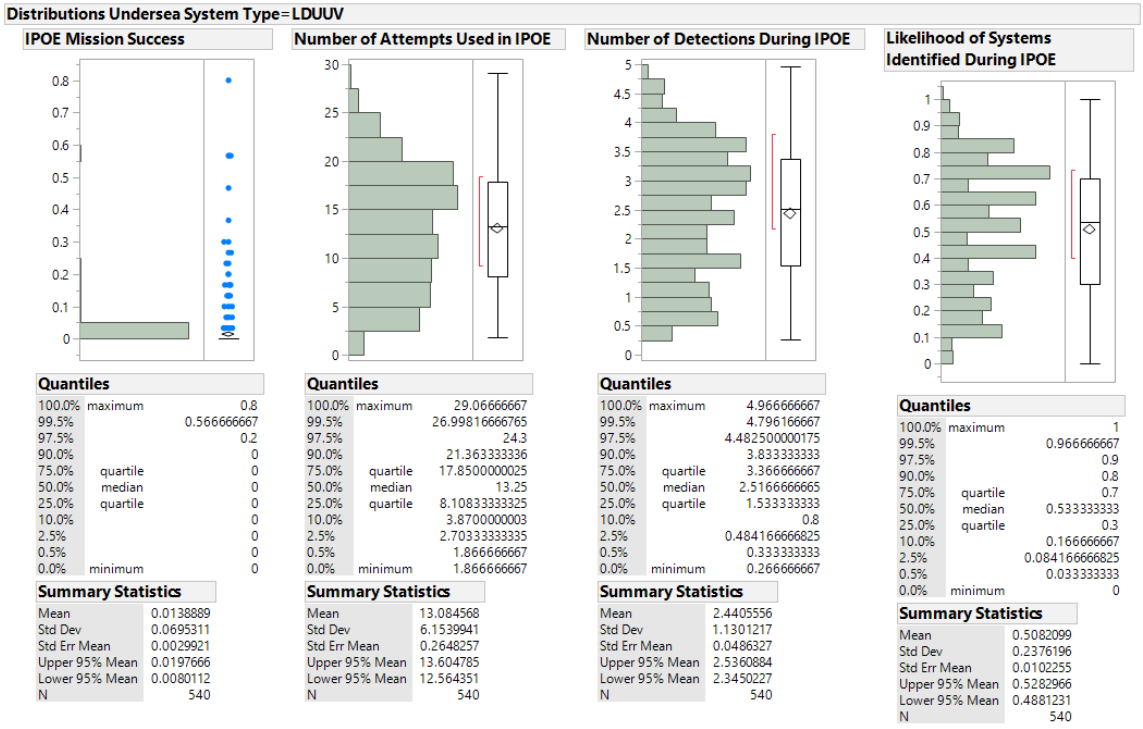


Figure 19. LDUUV Response Distribution for IPOE

Moving onto the MDUUV's response distributions shown in Figure 20, the MDUUV performs better than the XLUUV and the LDUUV. The probability of IPOE mission success for the MDUUV is about 40%, which still is not very high. The MDUUV attempts IPOE on average 9.51 times while being detected 1.31 times. The likelihood that the opposing force would identify the MDUUV is about 19%. The interquartile range for IPOE mission success of the MDUUV is (0.13, 0.57) with the densest region of data found within the range. The data is negatively skewed with a heavy left tail, and this can be seen as the MDUUV performance starts highest but drops steadily up to a kill box size of 1000 nm^2, after which is evens out for kill boxes sized larger. The other responses are positively skewed except for likelihood of systems identified during IPOE, which is balanced around 0.16 with the outliers observed at the largest kill box size.

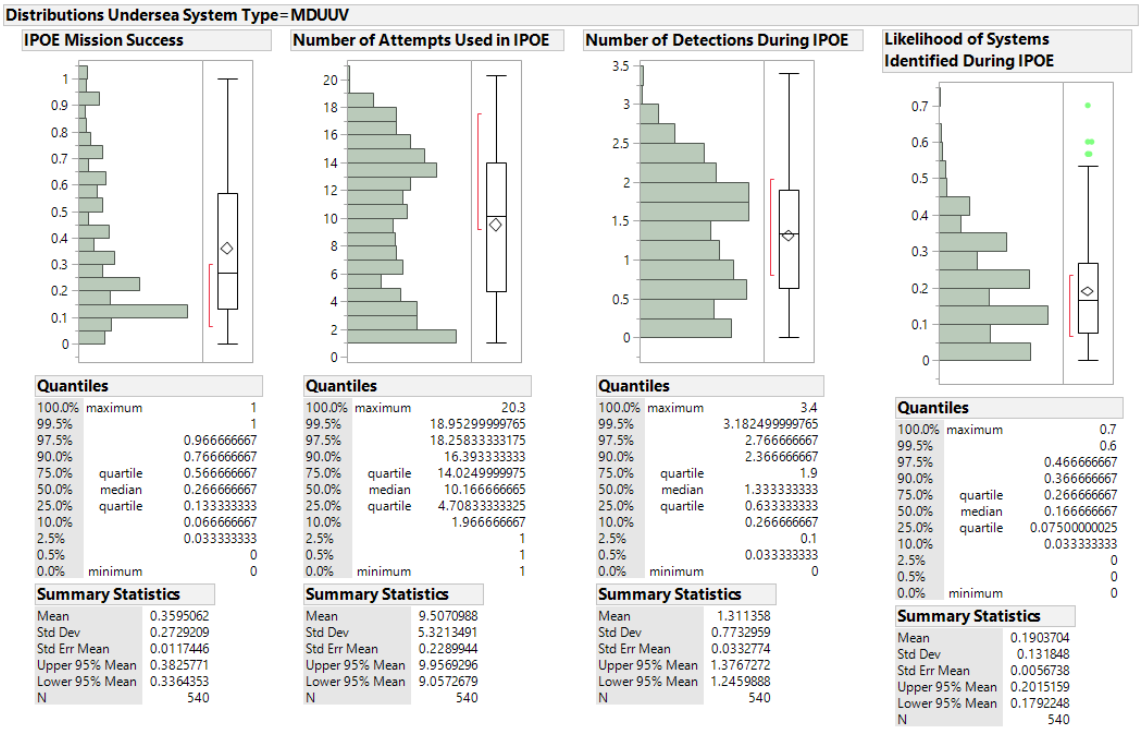


Figure 20. MDUUV Response Distribution for IPOE

Finally, Figure 21, the SUUV has the second lowest success rate, approximately 6%. The system attempts IPOE approximately 18.74 on average while being detected 1.85 times. The likelihood of identification for the SUUV by opposing forces is about 19%. There is a similarity between the distributions of IPOE mission success between the SUUV and the LDUUV, with the interquartile range centered near zero for both systems. Both systems have many outliers but where they differ is that the SUUV has many more as it performs much better for kill boxes sized 100 nm^2 or less but then joins the LDUUV performance as it rapidly declines to meet the LDUUV at 0% success for larger kill boxes. The SUUV does have the lowest likelihood of identification, and the data for number of attempts used in IPOE and number of detections skewed positively with a heavy left tail. The means and medians are very close in these responses, with very similar looking distributions.

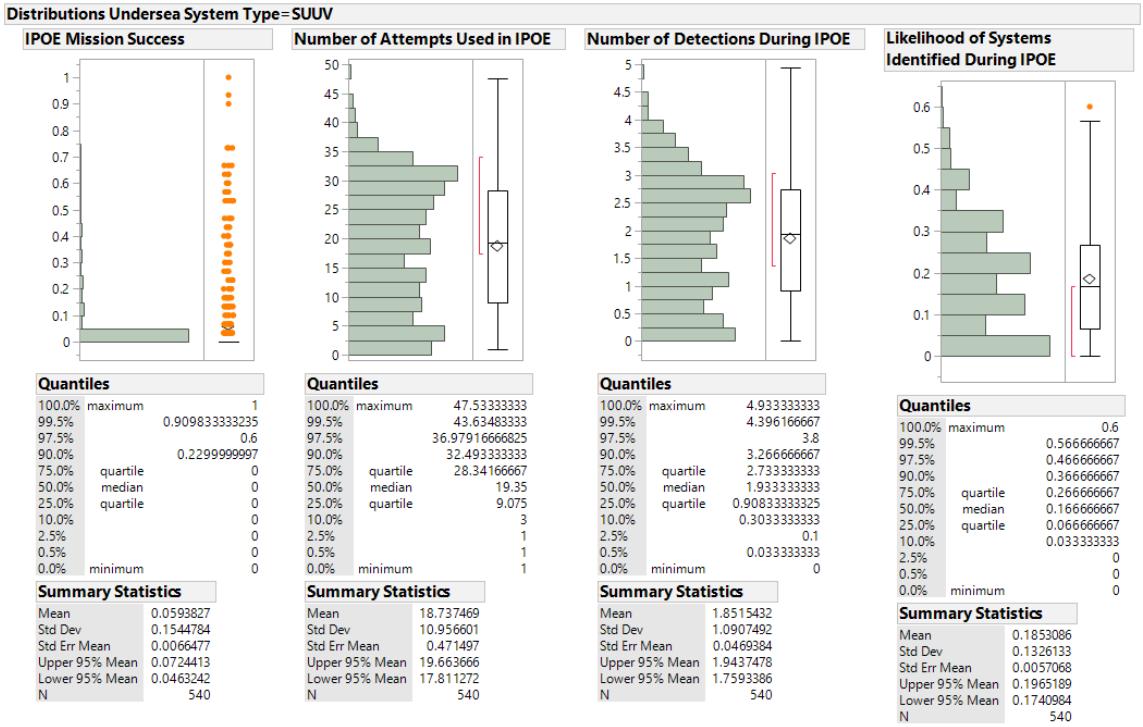


Figure 21. SUUV Response Distributions for IPOE

The distributions of the responses show the overall performance of the different undersea systems but none perform particularly well when looking across the entire range of the kill box. It is clear from the initial scatterplot shown in Figure 17 that no UUVs do particularly well with medium and large kill boxes. For this reason, the small kill box merited more in depth analysis. Prior to going further, an IPOE response screening finds which factors were not statistically significant for the responses. Table 13 shows the results of this screening.

Table 13. IPOE Response Screening

Response	Factor	P-Value	R-Squared
IPOE Mission Success	Kill Box Size	0	0.233
	P US(IPOE)	0	0.070
	P OPFOR(Detection US)	0.667	0
	P OPFOR(Classification US)	0.517	0
	Number of Undersea Systems	0	0.240
	Number of IPOE Attempts	0	0.008
Likelihood of Identification During IPOE	Kill Box Size	0	0.103
	P US(IPOE)	0	0.194
	P OPFOR(Detection US)	0	0.193
	P OPFOR(Classification US)	0	0.203
	Number of Undersea Systems	0	0.061
	Number of IPOE Attempts	0	0.007
Number of Attempts Used in IPOE	Kill Box Size	0	0.283
	P US(IPOE)	0	0.010
	P OPFOR(Detection US)	0	0.105
	P OPFOR(Classification US)	0	0.107
	Number of Undersea Systems	0	0.807
	Number of IPOE Attempts	0.334	0
Number of Detections During IPOE	Kill Box Size	0	0.253
	P US(IPOE)	0	0.039
	P OPFOR(Detection US)	0	0.014
	P OPFOR(Classification US)	0.015	0.003
	Number of Undersea Systems	0	0.410
	Number of IPOE Attempts	0.011	0.003

The most important response is IPOE mission success, and the response screening shows that the opposing force capabilities are not statistically significant factors. All factors have some statistical value when looking at the likelihood of identification during IPOE. For the number of attempts used in IPOE, the number of IPOE attempts which shows the maximum number of attempts for the UUV type is statistically insignificant. Number of detections during IPOE has two responses with p-values that are below the traditional threshold of 0.05, P_{OPFOR}(Classification US) and number of IPOE attempts, but the R^2 values are so low that they were removed from subsequent analysis as they explained almost zero of the variability in the model. For the purpose of this analysis and in order to find which UUV is best suited for the IPOE Mission, the analysis will be broken down into

looking at the small kill box, as highlighted in yellow in Figure 22, and the medium and large kill boxes will be grouped together.

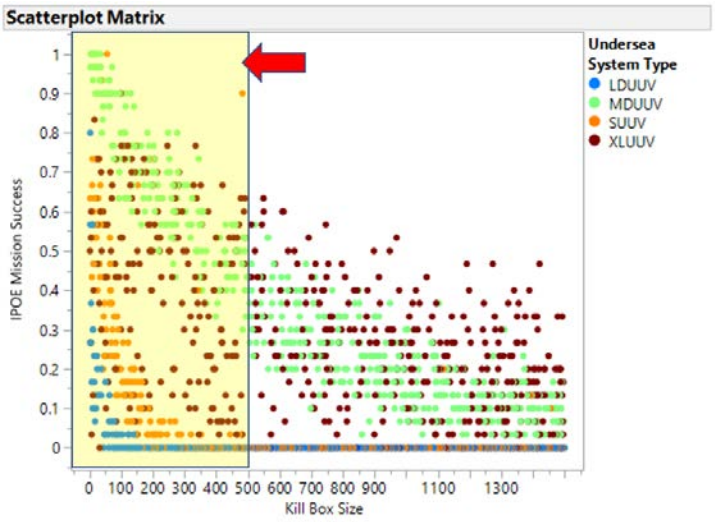


Figure 22. Kill Box Size vs. IPOE Mission Success

As seen in Figure 22, all UUVs show capability to perform IPOE for small kill boxes, but only the XLUUV and the MDUUV have a chance at successfully performing IPOE for large kill boxes. Figure 23 and Figure 24 present scatterplots detailing the relationship between both kill box size and number of undersea systems and the probability of IPOE mission success.

Looking at Figure 23 and Figure 24, for both the small and large kill boxes, the MDUUV is more predictable in its performance than the XLUUV. This is due to the MDUUV's superior equipment for performing IPOE. Of course, more MDUUVs are required in order to do what one XLUUV can do, but the XLUUV does not have a reliable performance. This is largely due to the XLUUV not having the same technology as the MDUUV for performing IPOE.

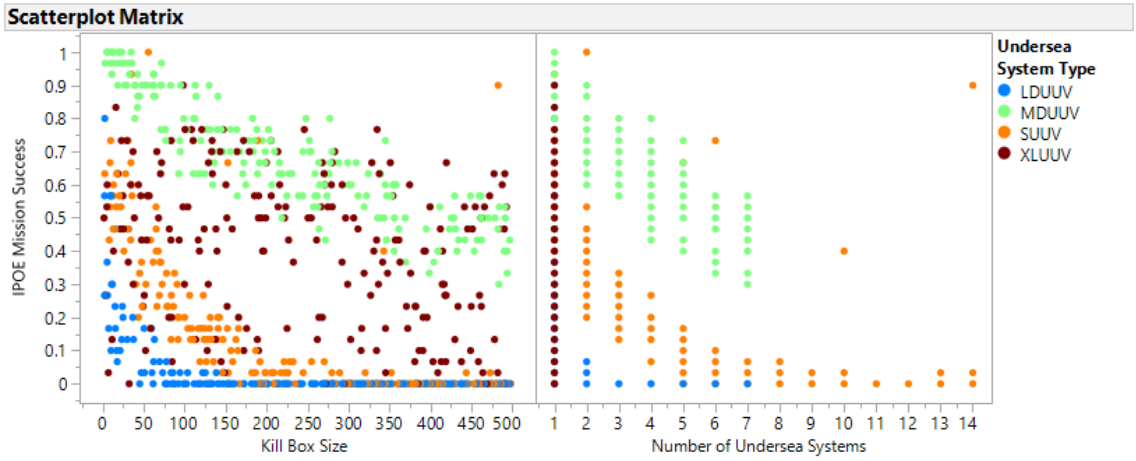


Figure 23. IPOE Mission Success for Small Kill Box

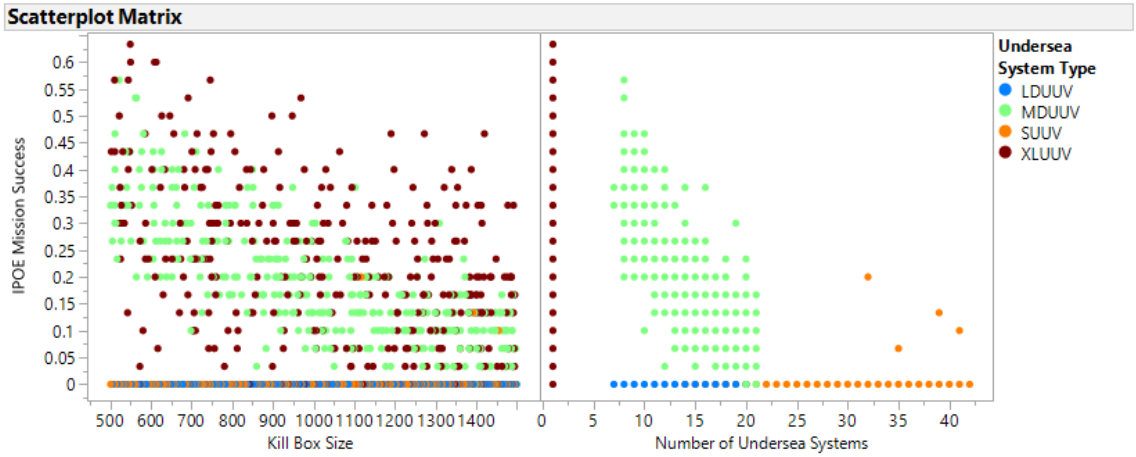


Figure 24. IPOE Mission Success for Medium/Large Kill Boxes

A medium/large kill box has at best under 70% chance of IPOE success with that data largely coming from the medium-ranged kill boxes. For large kill boxes, the likelihood drops to below 50%. More systems only make success worse as there are more failure points and probability of detection rises. This is true across for all kill box sizes. With small kill boxes, it seems the MDUUV would be the best suited for ranges between 1 and 100 nm^2 in size. This gives a more reliable range of success between 80–100%. If the

XLUUV's current capability of IPOE was to be enhanced by more advanced technology, it would become the best option.

Figure 25 and Figure 26 displays the likelihood of identification during IPOE. The downward trend of the XLUUV likelihood of identification as the kill boxes size increases is largely due to the low probability of IPOE success but a large endurance to attempt IPOE many times. This is the opposite of all other undersea systems. Each attempt is the possibility of an opposing force detecting the XLUUV. Longer missions for larger kill boxes means fewer attempts even if the XLUUV is performing the mission for a similar amount of time. This shows another limitation of the probabilistic model, as this does not accurately reflect actual time spent in the mission equating to probability of detection and classification. Within these scatterplots for the small kill boxes, the likelihood of identification is higher for the XLUUVs than LDUUVs whereas for medium and large kill boxes, the opposing forces are more likely to identify the LDUUV.

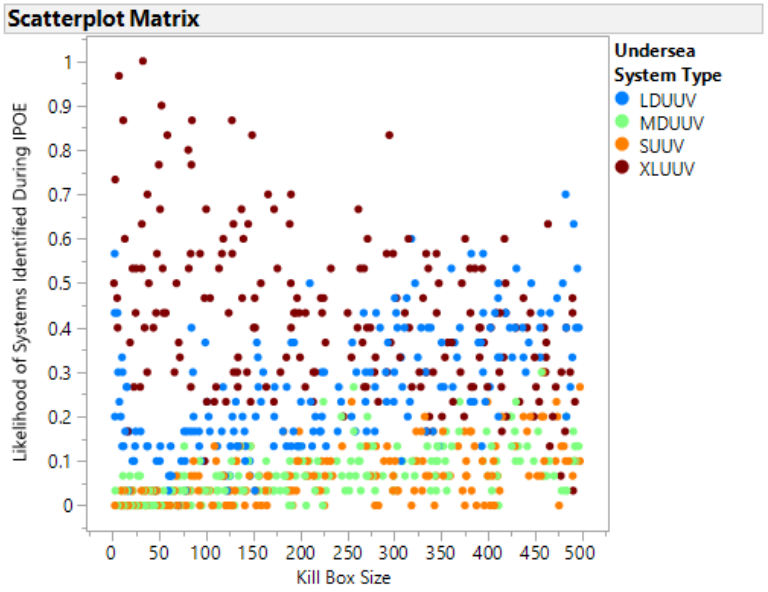


Figure 25. Likelihood of Identification during IPOE for Small Kill Boxes

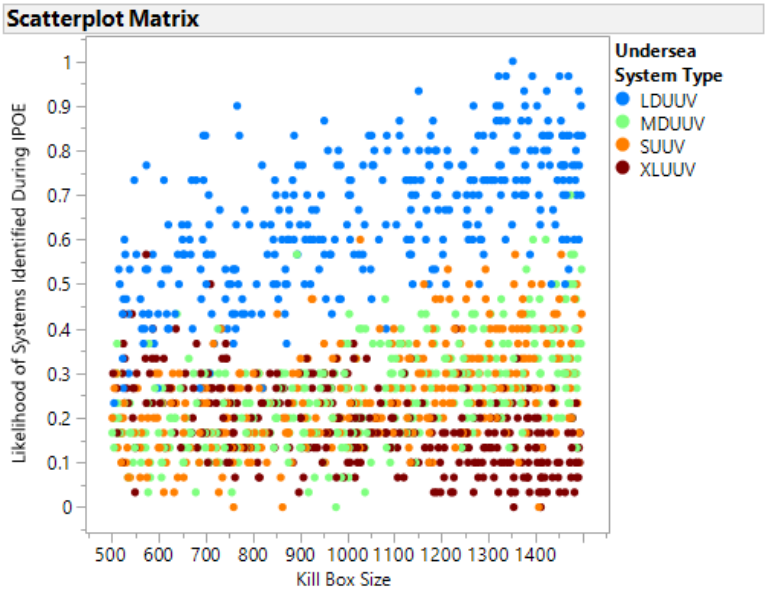


Figure 26. Likelihood of Identification during IPOE for Medium/Large Kill Boxes

Looking at a similar binning for the number of IPOE attempts used, shown in Figure 27 and Figure 28, the reasoning behind the XLUUV and LDUUV likelihood of identification swapping can further be investigated. The LDUUV attempts IPOE more than the XLUUV for the larger kill boxes, whereas the opposite is true for the small kill box. The size trend is the same here as the identification probability since they are related and the number of IPOE attempts used increases as the number of systems increase since each system contributes at least one attempt, and potentially more.

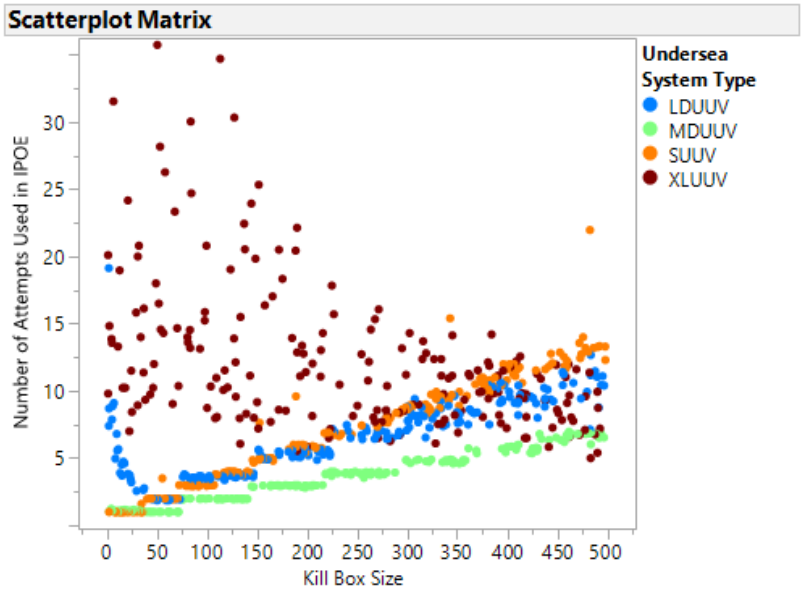


Figure 27. Number of IPOE Attempts Used for Small Kill Boxes

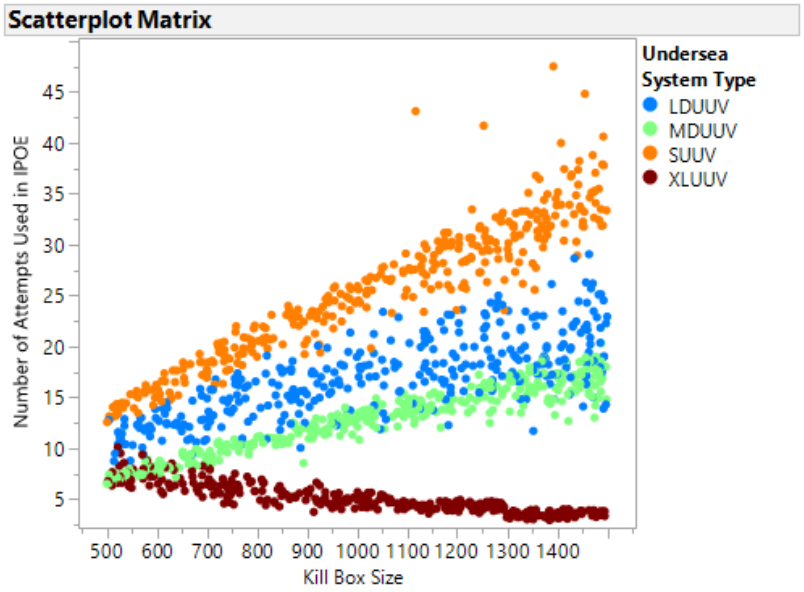


Figure 28. Number of IPOE Attempts Used for Medium/Large Kill Boxes

The final response, number of detections during IPOE shown in Figure 29 and Figure 30, continues to repeat the same trends discussed for the previous two responses. This is because attempts increase the detections and detection is the first step to

identification. Next, a comparison of the systems shows the best performance for each type of kill box.

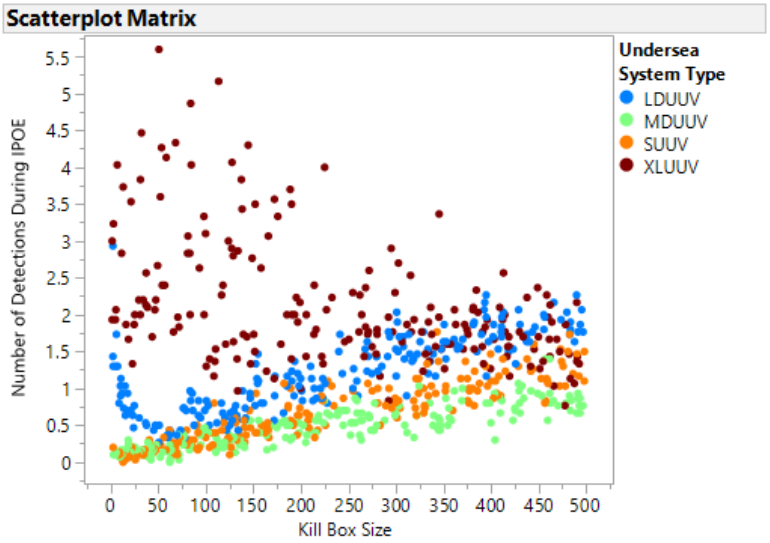


Figure 29. Number of Detections during IPOE for Small Kill Box

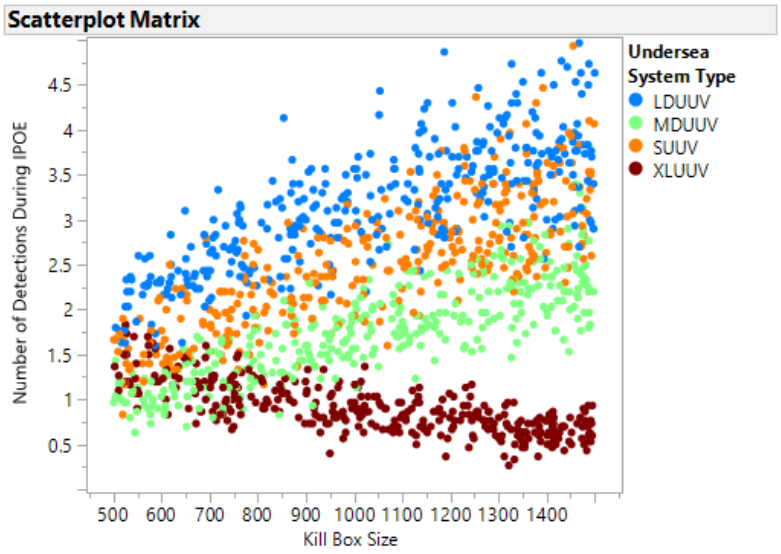
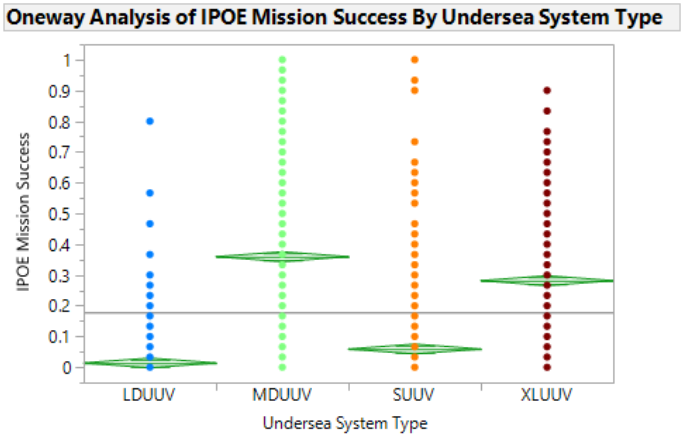


Figure 30. Number of Detections during IPOE for Medium/Large Kill Boxes

Figure 31 shows that if only one undersea system type is to perform IPOE in any size kill box between 1 and 1500 nm^2 in size, the MDUUV is best suited, although it will

only succeed about 40% of the time on average. The team recognizes that the data output may create the impression of false precision. Team Leviathan does not mean for this analysis to suggest that the probability of success can be analyzed to multiple decimal places of accuracy. Rather, the numbers used in the narrative provide enough decimal places for the reader quickly to connect the narrative text to the output of the statistics program used to support the analysis. The overall probability, for the analyzed UUVs, is only about 18%, which means that there is a large capability gap when it comes to finding a singular system that could complete the mission for any kill box size. This shows that a better approach is to analyze the UUVs against each kill box type as opposed to across the entire range. As stated earlier in analysis, the team combined the medium and large kill boxes due to the scatterplot results.



Oneway Anova

Summary of Fit

Rsquare	0.370424
Adj Rsquare	0.369548
Root Mean Square Error	0.189861
Mean of Response	0.178596
Observations (or Sum Wgts)	2160

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Undersea System Type	3	45.72697	15.2423	422.8428	<.0001*
Error	2156	77.71788	0.0360		
C. Total	2159	123.44485			

Means for Oneway Anova

Level	Number	Mean	Std Error	Lower 95%	Upper 95%
LDUUV	540	0.013889	0.00817	-0.0021	0.02991
MDUUV	540	0.359506	0.00817	0.3435	0.37553
SUUV	540	0.059383	0.00817	0.0434	0.07541
XLUUV	540	0.281605	0.00817	0.2656	0.29763

Std Error uses a pooled estimate of error variance

Figure 31. IPOE Mission Success Comparison for Full Kill Box Range

Figure 32 shows a side-by-side comparison of kill box performance. This reveals that at small kill box sizes, the MDUUV again is the best performer with an approximately 67% success rate on average. This is based on a kill box sized 1 to 500 nm^2 in area. This is not an acceptable number by any means but it is the best out of the selected UUVs. The XLUUV success rate is about 41%, the SUUV success rate is about 16%, and the LDUUV success rate is about 4%.

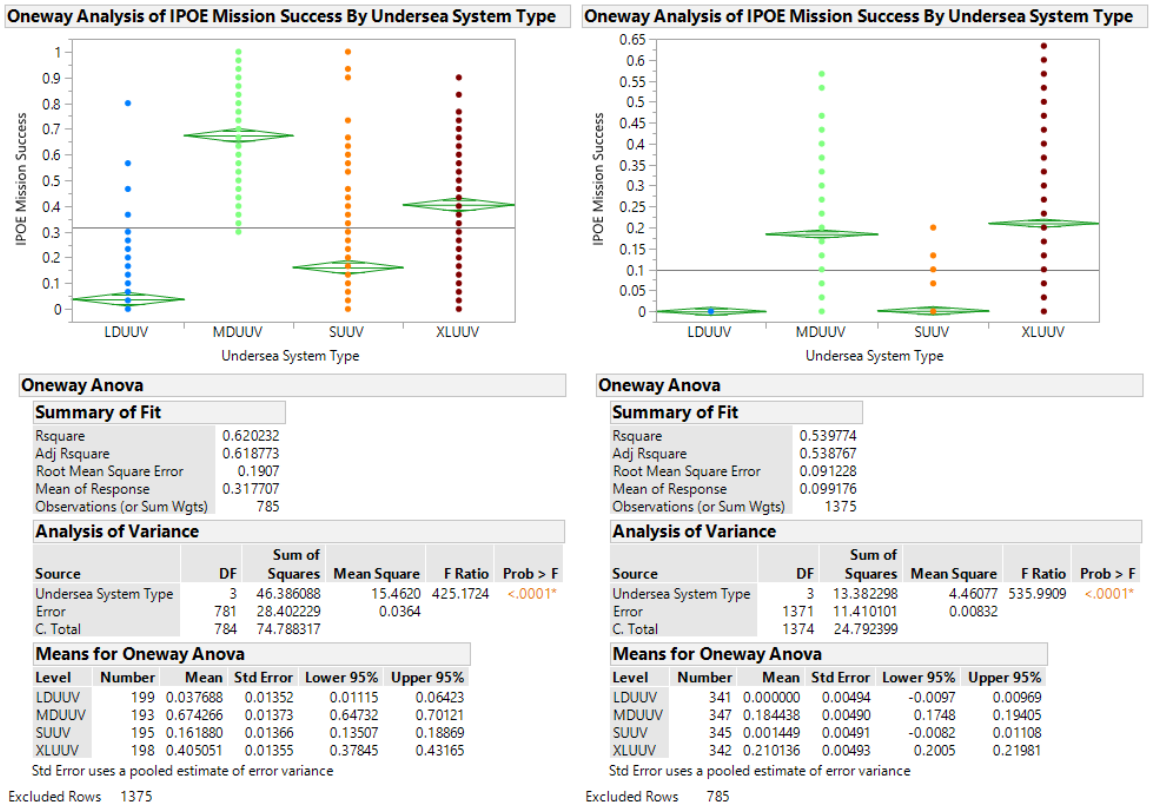
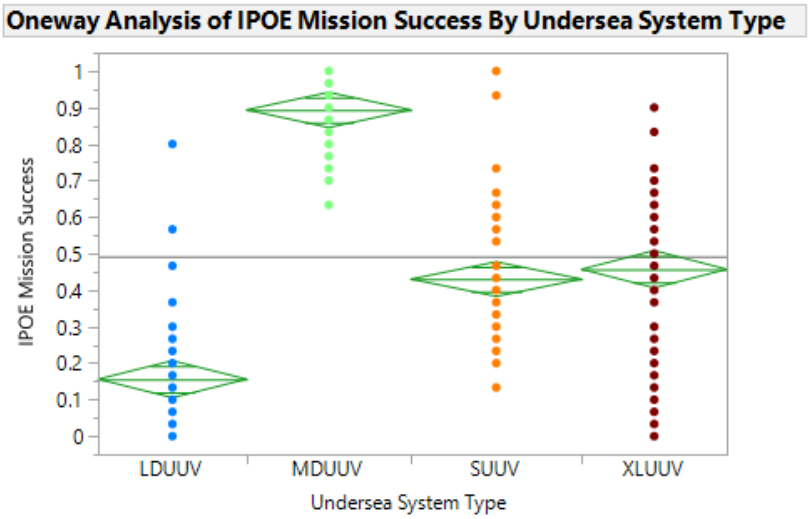


Figure 32. IPOE Mission Success Comparison for Small (left) and Medium/Large (right) Kill Boxes

For the medium and large kill box however, the XLUUV now becomes the best performer, with a success rate of about 21%. The MDUUV success rate is about 18% and the LDUUV and SUUV are not capable with rates of 0% and 0.14% respectively. In both kill box sizes, the LDUUV is the worst performer due to its poor IPOE capability and high detectability.

Since neither kill box size range is feasible with current technology, Figure 33 shows one final comparison based on all of the results up until now. At a range of 1 to 100 nm^2 for the kill box, the MDUUV now has a success rate of approximately 89%. The XLUUV and SUUV, although not great, are much closer in performance at this range, with success rates of about 46% and 43% respectively. This also further proves that the LDUUV has limited utility in this scenario. When looking at the utility of the XLUUV for seabed warfare within the context of a kill box, there is some potential in medium and large kill boxes, but if all capabilities stay the same, the MDUUV is better all-around, and especially in the smaller kill boxes. If the XLUUV is able to inherit some of the MDUUVs IPOE capabilities, it could become very useful.



Oneway Anova

Summary of Fit

Rsquare	0.697653
Adj Rsquare	0.693002
Root Mean Square Error	0.176095
Mean of Response	0.492797
Observations (or Sum Wgts)	199

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Undersea System Type	3	13.952840	4.65095	149.9850	<.0001*
Error	195	6.046836	0.03101		
C. Total	198	19.999676			

Means for Oneway Anova

Level	Number	Mean	Std Error	Lower 95%	Upper 95%
LDUUV	47	0.156028	0.02569	0.10537	0.20669
MDUUV	52	0.893590	0.02442	0.84543	0.94175
SUUV	54	0.430247	0.02396	0.38299	0.47751
XLUUV	46	0.457246	0.02596	0.40604	0.50845

Std Error uses a pooled estimate of error variance

Excluded Rows 1961

Figure 33. IPOE Mission Success Comparison for 1–100 nm Kill Boxes

2. Intelligence, Surveillance, and Reconnaissance

The ISR mission is the second of the three missions required for successful kill box deployment. This mission involves sending an undersea system into the tactical area to deploy a set number of ISR devices in order to achieve full field resolution of the desired kill box.

Table 14 shows the factors used in this mission analysis. The asterisks within the value column denote the factors that variate dependent on the undersea system evaluated. Since the AoA is on the undersea system, there will be no investigation into the factors that are unrelated to the performance of the undersea system within the mission; these factors are shown in grey within the table.

Table 14. ISR Mission Factors

Factor	Values
Kill Box Size	1-1500
PUS(ISR Deployed)	*
POPFOR(Detection US)	*
POPFOR(Classification US)	*
POPFOR(Detection ISR)	*
POPFOR(Classification ISR)	*
Number of Undersea Systems	*
Number of ISR	*
Number of ISR Required	*
Endurance	*

In order to complete the ISR mission, the undersea systems must deploy the number of required ISR devices for the field resolution. From the mission, the simulation collected certain MOEs in order to assess performance, shown by order of importance below in Table 15.

Table 15. ISR Mission Responses

Response
ISR Mission Success
Number of Systems Successfully Deployed during ISR
Number of Detections During ISR
Likelihood of Identification During ISR

Team Leviathan used trend analysis to evaluate all mission factors against the response variables. In the scatterplot shown in Figure 34, it appears that the kill box size is a major driver for the mission success. Another major point is that the SUUV is not effective in performing the mission. For the kill box size, the results show there is minimal success within a large kill box but the success increases as the size decreases. ISR Deployment also plays a role in mission success but does not exhibit a trend. The remaining three responses follow similar trends to each other. Subsequently number of detections and identification increases as the kill box size increases since the need for more ISR systems rises. The data conveys this linear trend for the three responses in kill box size. For probability of ISR deployment, the data is significant as a whole without variability or trends in the results in the responses.

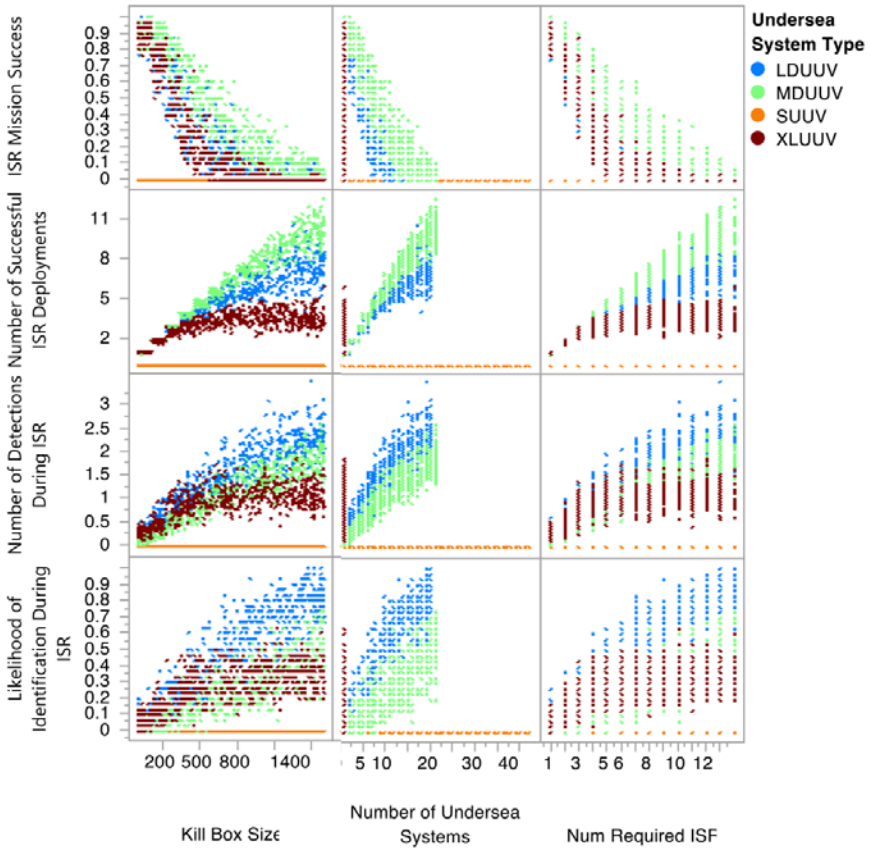


Figure 34. ISR Mission Scatterplot

The scatterplot also indicates that the probability of opposing forces detecting or classifying the undersea systems does not have a significant impact on system performance. The number of undersea systems affects mission success within a negative non-linear trend. In this case, the kill box size drives the responses. The number of ISR systems required also trends downward for success and trends up for the remaining three responses. The size of the kill box predicts the need for additional ISRs.

The simulation predefined the number of undersea systems needed for the mission by the IPOE mission and carried throughout all missions sequentially; thus, only a singular XLUUV performs each mission. The scatterplot shown in Figure 35, shows the number of ISR device each undersea system is capable of carrying. Based on the graph, it is clear that the SUUV is unable to carry any ISR devices and the team has omitted the system from the analysis of this mission.

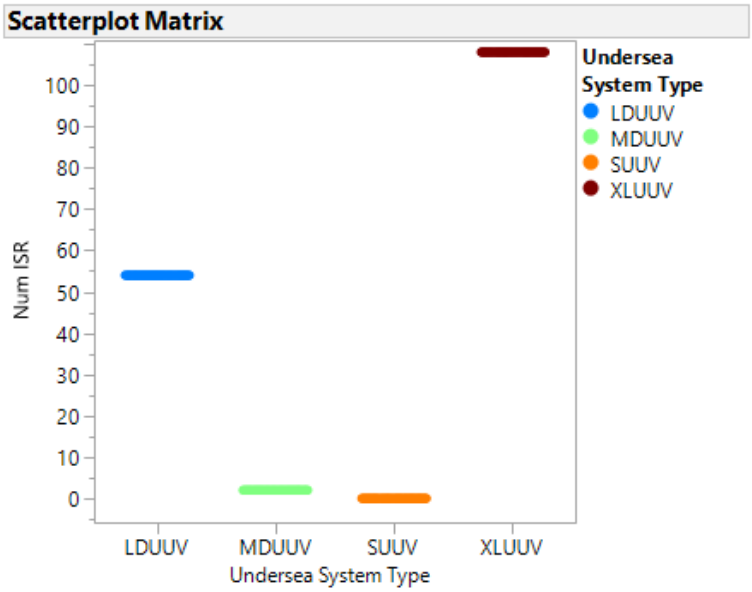


Figure 35. ISR Payload Size

Following the overview of the responses against the various factors, analysis of the distributions obtains a preliminary understanding of the summary statistics associated with each response.

Shown in Figure 36 are the distributions of the different responses, based on the full range of the kill box, for the XLUUV. The overall mean probability of mission success is about 23% with an interquartile range of (0, 0.4). In this responses case the mean lies significantly above the median with a negative skewedness and a heavy right tail. The XLUVV typically successfully deploys a median of 3.16 ISR devices on average with an interquartile range of (2.48, 3.66) while being detected 0.93 times with an interquartile range of (0.7, 1.13). The mean of ISR deployments and median are similar being at 3.00 and 3.16 successful deployments respectively. Similarly, the mean and median of detections followed the same similarities being at .91 and .93 respectively. The median likelihood of identification for the XLUUV is about 30% with an interquartile range of (0.36, 0.1). Similar to the previous two responses the medians and mean share comparative values at about 30% and 27% respectively. Each of the three responses do not show significant skew one way or the other.

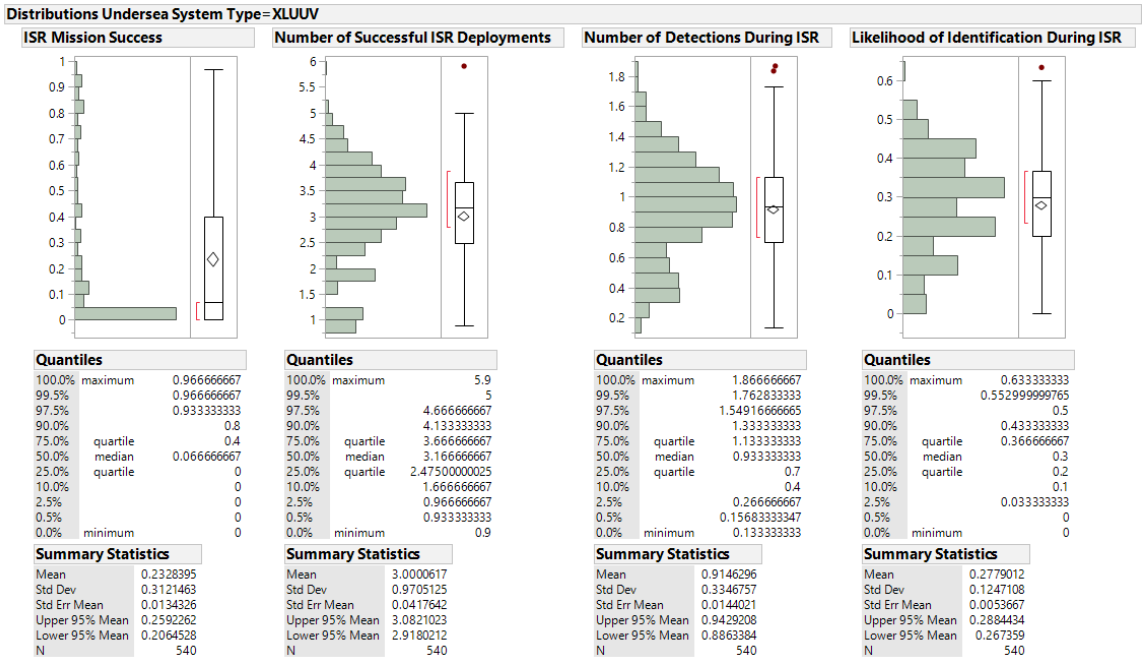


Figure 36. XLUUV Response Distributions for ISR

Shown in Figure 37 are the distributions of the different responses, based on the full range of the kill box, for the LDUUV. The overall mean probability of mission success is about 29% with an interquartile range of (0, 0.56). In this responses case the mean lies significantly above the median with a negative skewedness and a heavy right tail. The LDUUV typically successfully deploys a median of 5.05 ISR devices on average with an interquartile range of (2.84, 6.56) while being detected 1.56 times with an interquartile range of (0.9, 2.1). The mean of ISR deployments and median are similar being at 4.83 and 5.05 successful deployments respectively. Similarly, the mean and median of detections followed the same similarities being at 1.48 and 1.56 respectively. The response of successful ISR deployments displays a negative skew with a progressive right tail. While the LDUUV detections maintains a negative steady skew, the median likelihood of the LDUUV being identified by an opposing force is about 50% with an interquartile range of (0.21, 0.7). Similar to the previous two responses the medians and mean share comparative values at about 50% and 47% respectively. The skew of identifications of LDUUVs displays a slight positive skew.

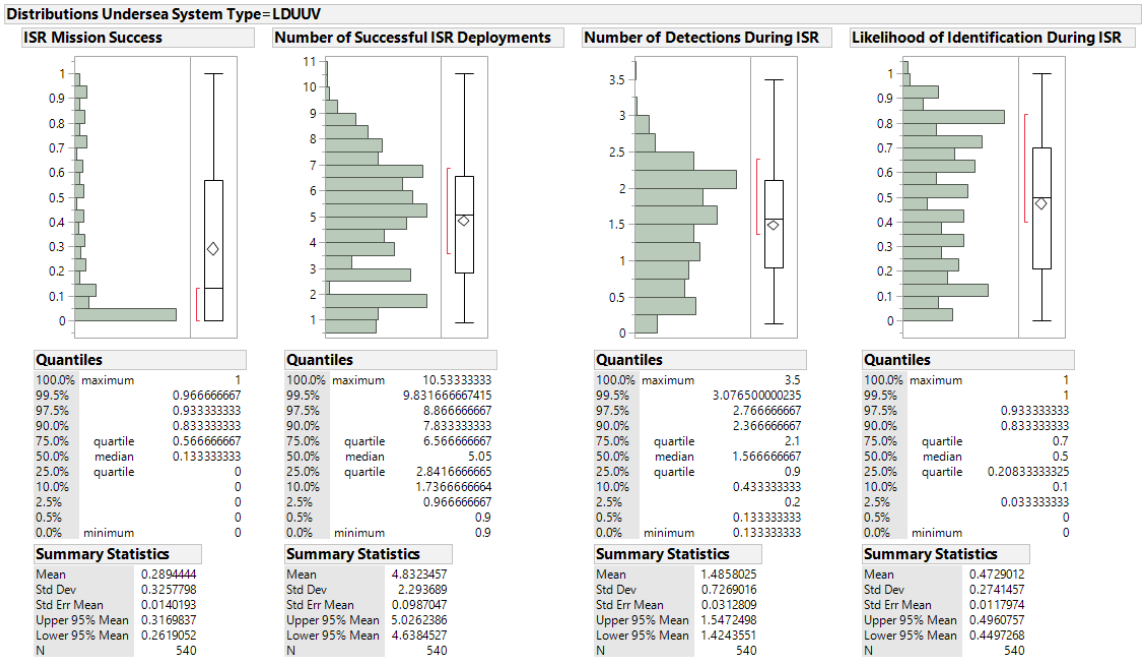


Figure 37. LDUUV Response Distributions for ISR

Shown in Figure 38 are the distributions of the different responses, based on the full range of the kill box, for the MDUUV. The overall mean probability of mission success is about 40% with an interquartile range of (0.1, 0.73). In this responses case, the mean lies significantly above the median with a negative skewedness and a heavy right tail. The MDUUVs typically successfully deploy a median of 6.3 ISR devices on average with an interquartile range of (3, 8.63) while being detected 1.1 times with an interquartile range of (0.57, 1.57). The mean of ISR deployments and median are similar being at 6 and 6.3 successful deployments. Similarly, the mean and median of detections followed the same similarities being at 1.08 and 1.1 respectively. Successful ISR deployments display a consistent negative skew. While the average number of detections on the MDUUV has a negative skew with a leftward tail. The median likelihood of identification of the MDUUV by an opposing force is about 23% with an interquartile range of (0.1, 0.4). Similar to the previous two responses the medians and mean share comparative values at about 23% and 25% respectively. The skew of identifications of MDUUVs displays a negative skew with a progressive tail.

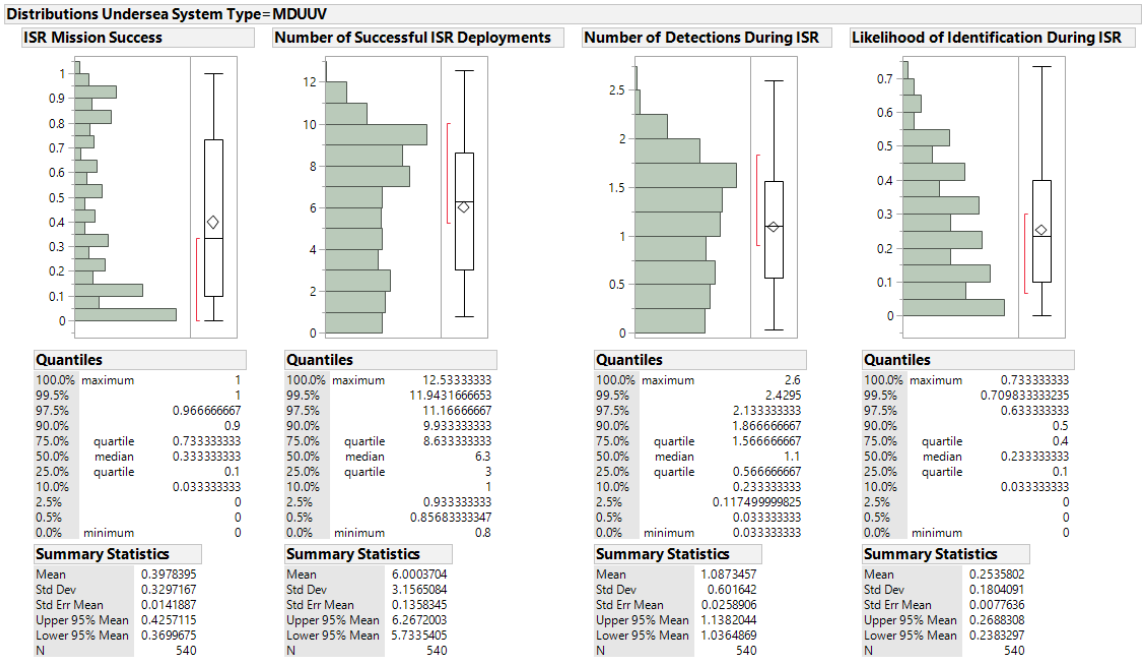


Figure 38. MDUUV Response Distributions for ISR

The most important response is ISR mission success, and the response screening each factor has some impact on the response with kill box size, number of ISR devices required, and number of undersea systems being the most important. The reason why the number of ISR systems required for full resolution is important to each response variables is because it is directly correlated to the kill box size. For the number of detections during ISR, the probability of the opposing forces classifying the undersea system was determined to be statistically insignificant. Next, the team used a response screening to investigate the major factors. Note that this response screening does not include the SUUV in the screening due to its inability to perform the mission and degrades the true values in the response screening.

Table 16. ISR Response Screening

Response	Factor	P-Value	R-Squared
ISR Mission Success	Kill Box Size	0	0.471
	P_US(ISR Deployment)	0	0.189
	P_OPFOR(Detection US)	0	0.0439
	P_OPFOR(Classification US)	0	0.047
	Number of Undersea Systems	0	0.216
	Num ISR	0.001	0.005
	Num Required ISR	0	0.470
	Endurance	0.725	0
Number of Successful ISR Deployments	Kill Box Size	0	0.238
	P_US(ISR Deployment)	0	0.464
	P_OPFOR(Detection US)	0	0.103
	P_OPFOR(Classification US)	0	0.111
	Number of Undersea Systems	0.664	0
	Num ISR	0.010	0.003
	Num Required ISR	0	0.237
	Endurance	0	0.007
Number of Detections During ISR	Kill Box Size	0	0.230
	P_US(ISR Deployment)	0	0.407
	P_OPFOR(Detection US)	0	0.340
	P_OPFOR(Classification US)	0	0.282
	Number of Undersea Systems	0.003	0.004
	Num ISR	0	0.850
	Num Required ISR	0	0.230
	Endurance	0.083	0.001
Likelihood of Identification During ISR	Kill Box Size	0	0.240
	P_US(ISR Deployment)	0	0.283
	P_OPFOR(Detection US)	0	0.332
	P_OPFOR(Classification US)	0	0.333
	Number of Undersea Systems	0.045	0.002
	Num ISR	0	0.119
	Num Required ISR	0	0.240
	Endurance	0.002	0.005

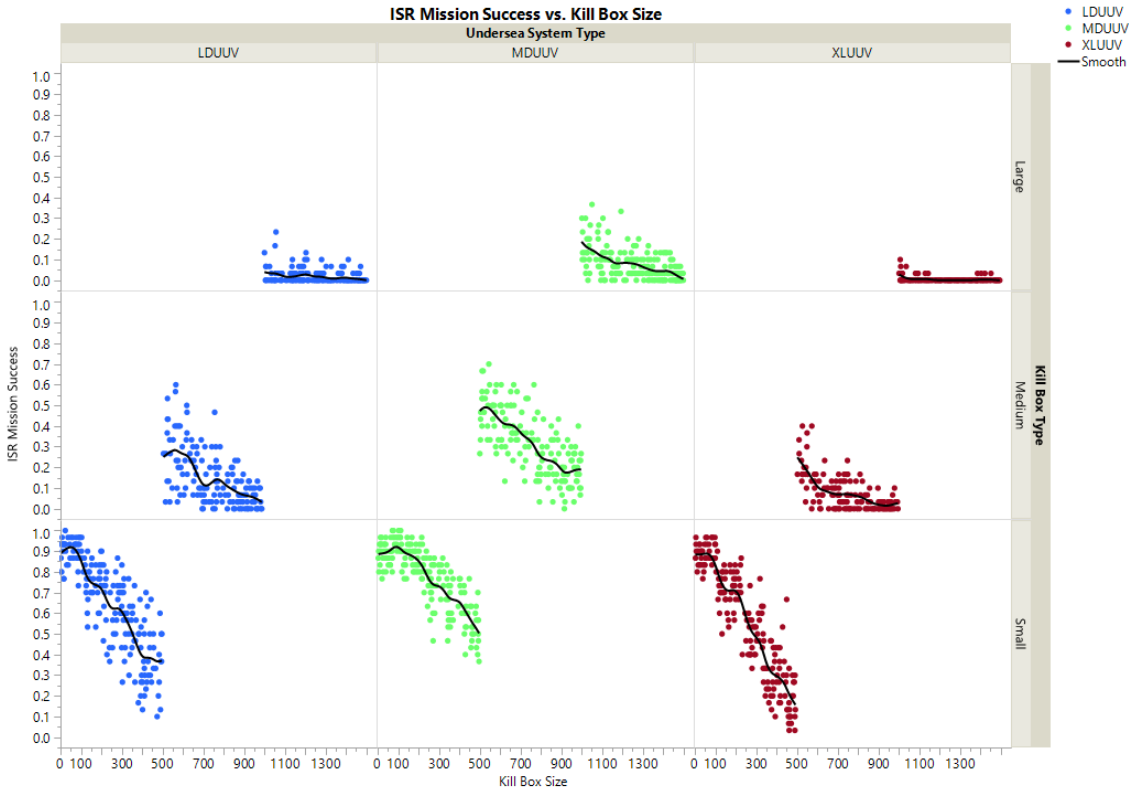


Figure 39. ISR Mission Success Binned by Kill Box

The largest driver for mission success is the kill box size. The MDUUV is capable of achieving ISR mission success in the largest kill box size followed by the LDUUV and the XLUUV. The MDUUVs require a great deal of systems to be able to achieve success in those large kill boxes, similar to the LDUUV at a lower success rate. The XLUUV alone was at times able to achieve success in a large kill box close to the largest kill box size predefined by the simulation inputs. The UUVs achieved the highest ISR mission success rates in the small kill boxes, classified as 500 nm^2 or smaller. For a high success rate, the kill box will need to be a least 250 nm^2 area. The scatterplot in Figure 40 shows the interrelation of the scope of the mission by kill box size and the difficulty in achieving success due to the mission's scope increase.

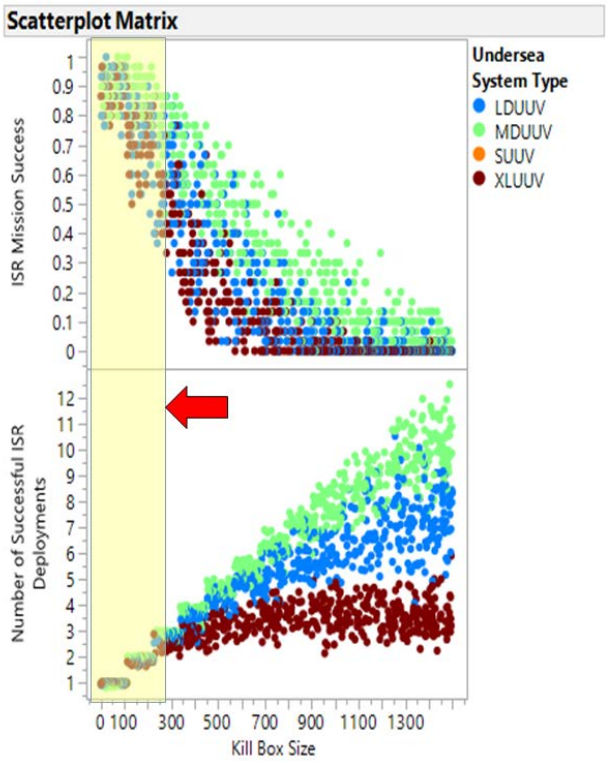


Figure 40. Kill Box Size vs. ISR Mission Success/Number of Successful ISR Deployments

The number of undersea systems plays an important role in Operation Leviathan, and that is not lost on the ISR mission. The value remains constant throughout each of the four missions. As shown in Figure 41, the more required systems, the higher probability of detection. In the case of the ISR mission, if one undersea system was able to perform the mission, that mission will be more likely to succeed.

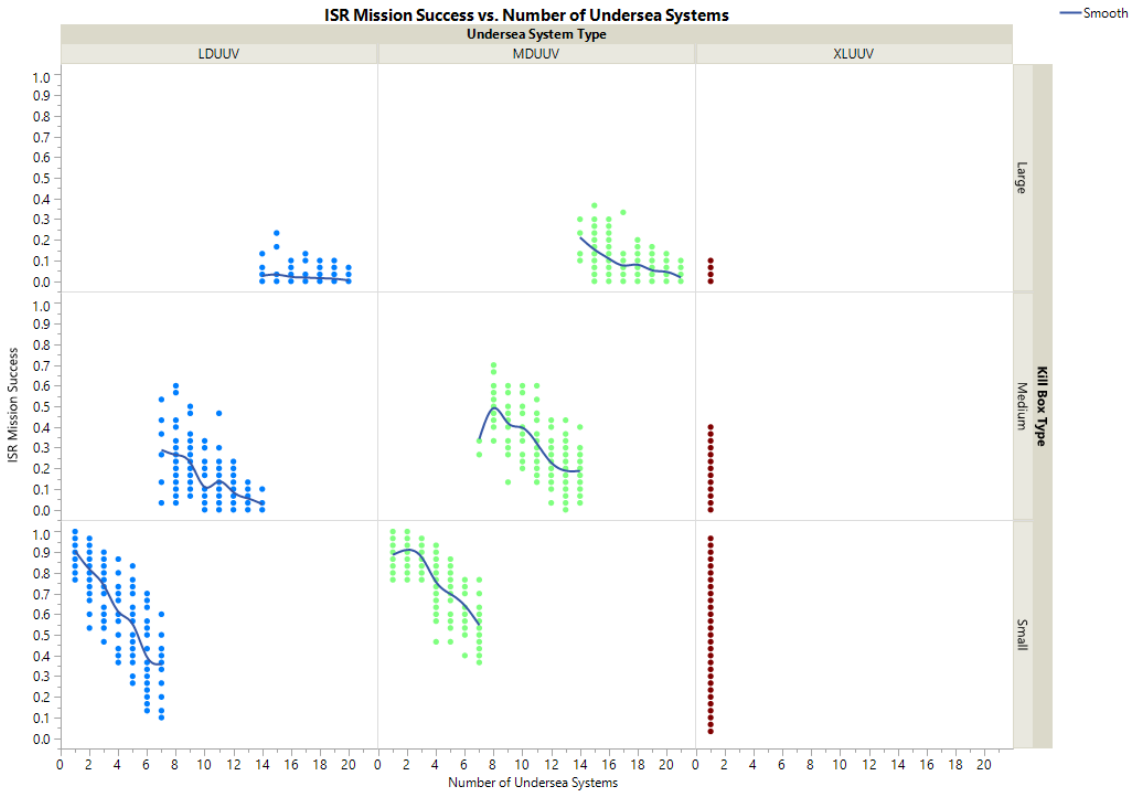


Figure 41. ISR Mission Success Binned by Number of Undersea Systems

The results of the ANOVA in Figure 42 indicate that the system is outputting expected results and confirming suspected results that the SUUV has no ability to perform the mission. Without an ISR payload, it is incapable of performing the ISR deployments necessary to achieve mission success, while the other three undersea systems are each capable of performing the mission to some degree with some success. The MDUUV is capable of performing the mission with an average of about 40% success rate, next highest successor is the LDUUV, about 29%, and lastly the XLUUV, about 23%.

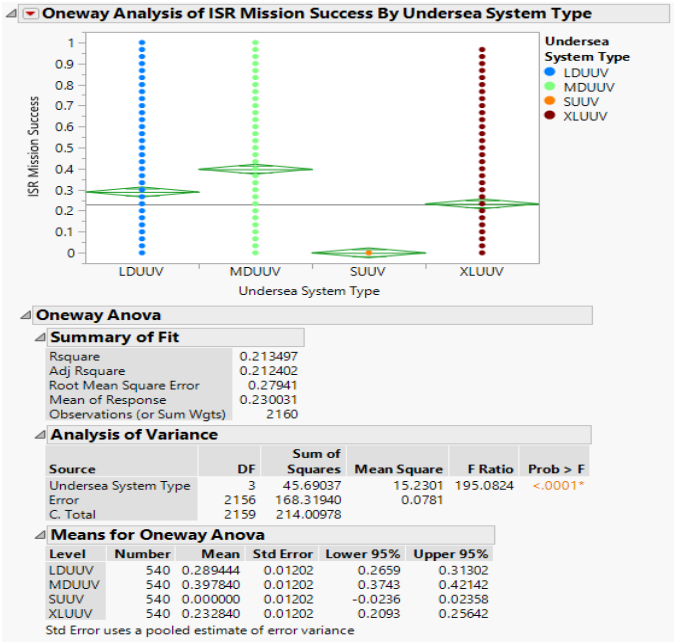


Figure 42. ISR Mission Success Comparison for Full Kill Box Range

3. Effects Field Deployment

The effects field deployment mission is the final of the three mission required for successful kill box deployment. This mission involves sending an undersea system to provide the installation or activation of various effect devices that could engage with a target of interest.

There are a variety of effects devices that could be used in this mission such as mines, acoustic devices, electromagnetic devices, or even seabed mounted launch tubes. For this scenario, one or more UUVs will be placing the effects device on the seabed to intercept opposing forces that may enter the kill box.

Table 17 shows the factors used in this mission. The asterisks within the value column denote the factors that variate dependent on the undersea system evaluated.

Table 17. Effects Field Deployment Mission Factors

Factor	Values
Kill Box Size	1-1500
PUS(EF Deployed)	*
POPFOR(Detection US)	*
POPFOR(Classification US)	*
POPFOR(Detection ISR)	*
POPFOR(Classification ISR)	*
POPFOR(Detection EF)	*
POPFOR(Classification EF)	*
Number of Undersea Systems	*
Number of ISR	*
Number of EF	*
Number of EF Required	*
Endurance	*

From the mission, the simulation collected certain MOEs in order to assess performance, shown by order of importance below in Table 18.

Table 18. Effects Field Deployment Mission Responses

Response
EF Mission Success
Number of Systems Successfully Deployed during EF
Number of Detections During EF
Likelihood of Identification During EF

Team Leviathan used trend analysis to evaluate all mission factors against the response variables. The scatterplot shown in Figure 43 depicts a high-level comparison of all the factors with the responses with color coded for the different undersea system types. Note that the opposing forces detected the XLUUV far less than the LDUUV during the mission unless the kill box was over 300 nm^2. In addition, the LDUUV and XLUUV were not able to succeed in a kill box with a size greater than ~250 nm^2 as shown in Figure 44. Additional scatterplots are included in Appendix D; however, the team developed no actionable insights based on these figures.

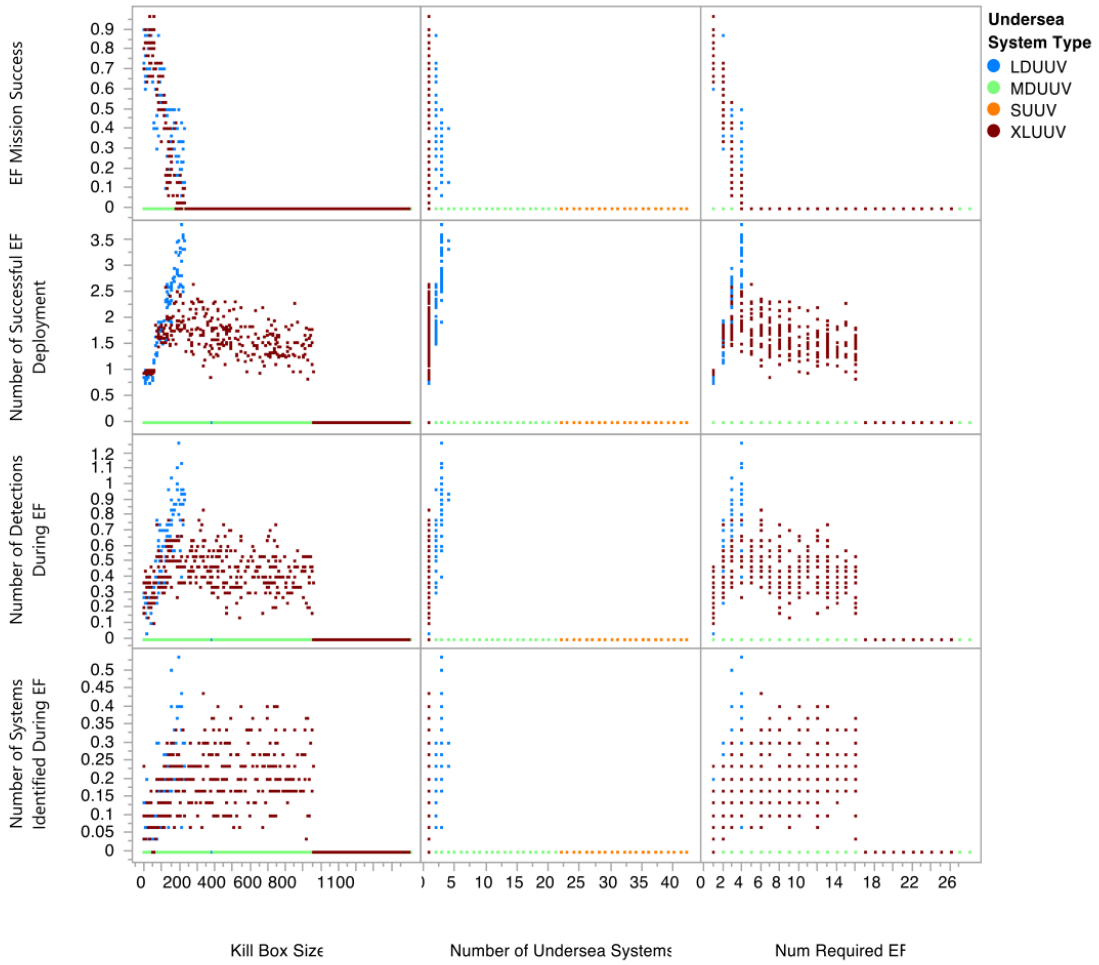


Figure 43. Effect Field Deployment Mission Results

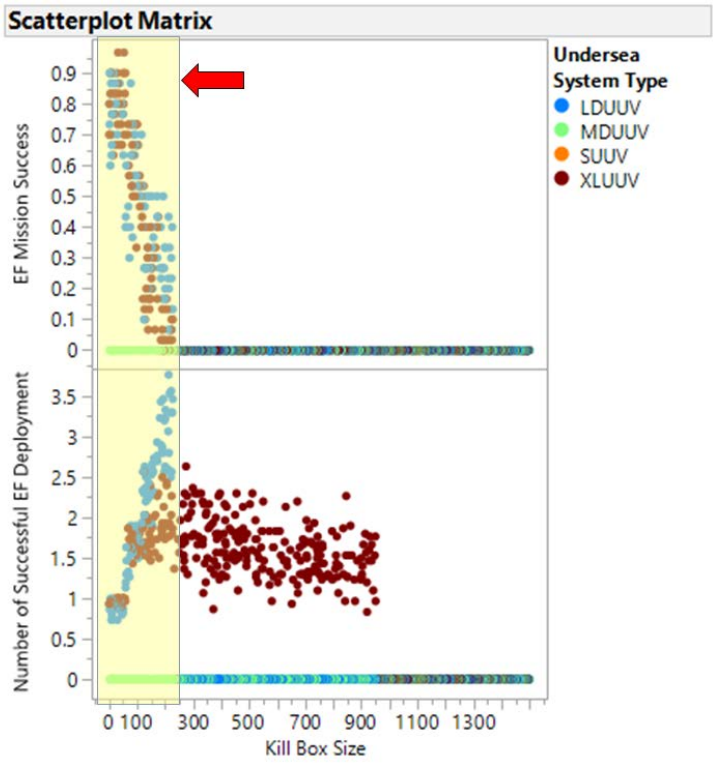


Figure 44. Kill Box Size vs. EF Mission Success/Number of Successful EF Deployments

Figure 45 shows the number of effects device each undersea system is capable of carrying. Based on the graph it is clear that the MDUUV and SUUV cannot carry any effects devices and the team has omitted the system from the analysis of this mission.

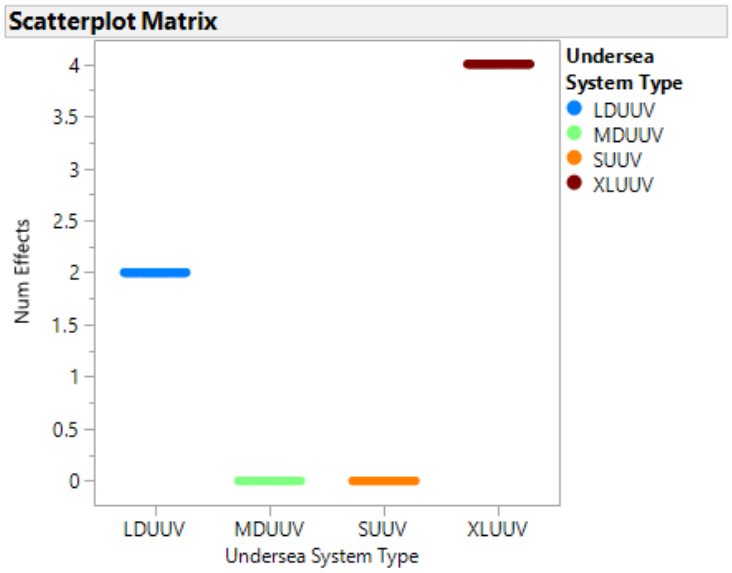


Figure 45. Effects Payload Size

Following the overview of the responses against the various factors, analysis of the distributions obtains a preliminary understanding of the summary statistics associated with each response.

Shown in Figure 46 are the distributions of the different responses for the XLUUV. The average probability of mission success for the XLUUV falls approximately at 8%. This is primarily due to the high identification rate. The likelihood of identification about 12%, attributed to the size of the UUV. However, the XLUUV did deploy more devices on average, which is why the mission success rate is similar to the LDUUV. The XLUUV successfully deployed 0.96 effects devices on average while being detected 0.26 times. There are a large number of outliers within the mission success for the XLUUV with the data very near to zero. The interquartile range for the data is (0, 0) meaning that the outliers were enough to skew the mean value to the 8% seen.

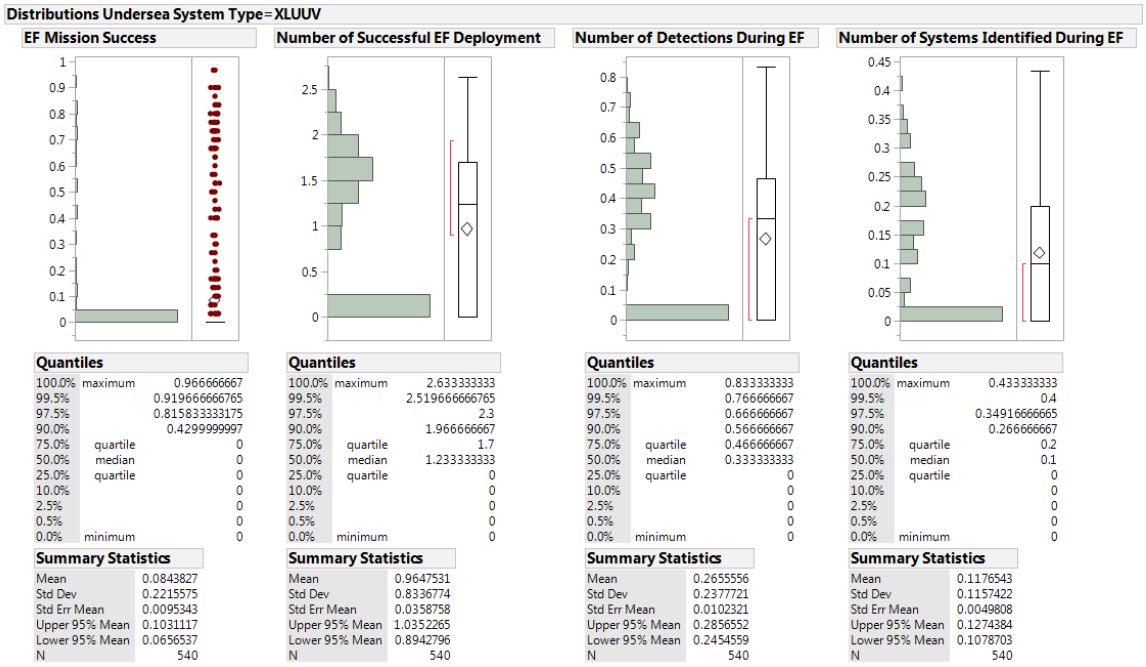


Figure 46. XLUUV Response Distribution for Effects Field Deployment

Shown in Figure 47 are the distributions of the different responses for the LDUUV. The overall mean probability of mission success falls slightly higher than the XLUUV, approximately 9%. This is primarily because the identification of the LDUUV was not as often; 3% of the time. The LDUUV successfully deployed 0.37 effects devices on average while being detected 0.11 times. Once again, across all response variables, there are an extreme number of outliers. This causes the data to skew negatively causing the mean to be 0.0937 while the interquartile range is (0, 0). Both UUVs show heavy left tails meaning that the densest amount of data is collected around 0.

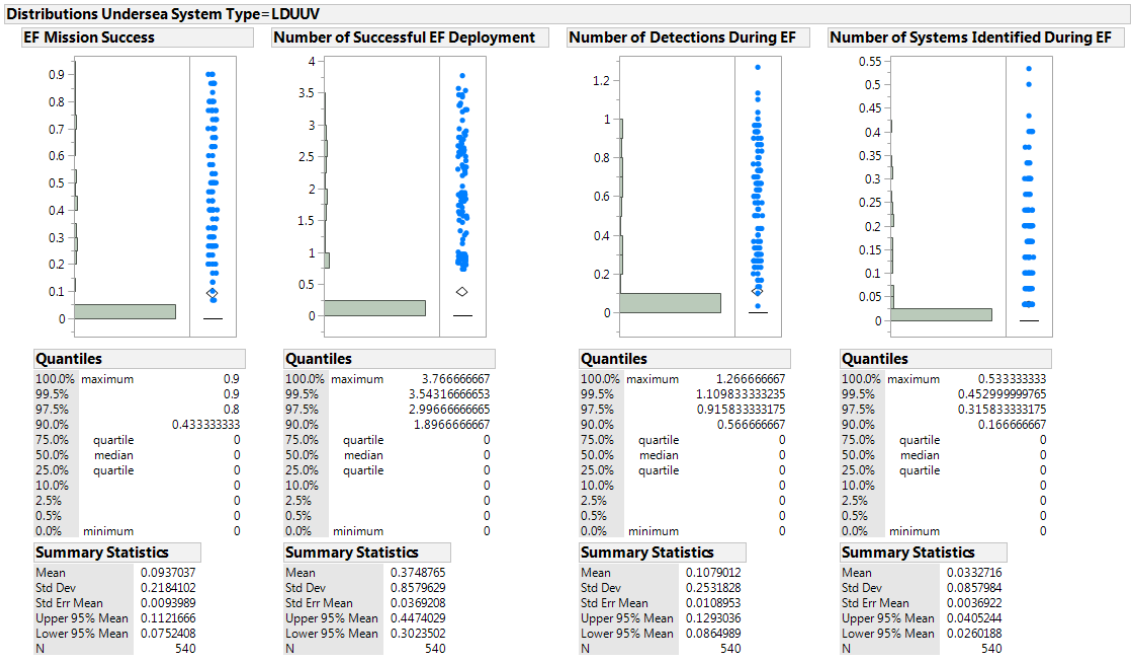


Figure 47. LDUUV Response Distribution for Effects Field Deployment

The two distributions provide a clear picture that neither system performed the mission particularly well. The XLUUV was able to deploy the effects devices more reliably but neither system was very successful in completing the full mission for the larger kill box sizes.

This highlights the comment made earlier; both the XLUUV and LDUUV were unable to complete the mission successfully if the kill box was above 250 nm^2. This was due to the number of systems used within the mission are the same number of systems used during IPOE. This causes the simulation to deploy only a single XLUUV. This, however, conflicts with the number of effects devices needed to obtain full resolution on the kill box. If the decision of the number of systems had taken into account the field resolution requirements, the number of systems deployed would have increased. However, because Team Leviathan did not incorporate tactical decisions into the simulation, the concept of increasing the number of assets was not addressed for this initial analysis of the utility of the XLUUV.

Prior to going further, the team performed a response screening to find which factors were not statistically significant for the responses. Table 19 shows the results of this screening.

Table 19. Effects Field Deployment Response Screening

Response	Factor	P-Value	R-Squared
EF Mission Success	Kill Box Size	0	0.345
	P_US(EF Deployment)	0.216	0.002
	P_OPFOR(Detection US)	0.711	0
	P_OPFOR(Classification US)	0.808	5.50E-05
	Number of Undersea Systems	0	0.074
	Number of Effects	0.687	0
Number of Successful EF Deployment	Kill Box Size	0	0.395
	P_US(EF Deployment)	0.315	0.001
	P_OPFOR(Detection US)	0.703	0
	P_OPFOR(Classification US)	0.791	0
	Number of Undersea Systems	0	0.237
	Number of Effects	0	0.094
Number of Detections During EF	Kill Box Size	0	0.259
	P_US(EF Deployment)	0.087424	0.004
	P_OPFOR(Detection US)	0.739	0
	P_OPFOR(Classification US)	0.701	0
	Number of Undersea Systems	0	0.220
	Number of Effects	0	0.147
Likelihood of Identification During EF	Kill Box Size	0	0.420
	P_US(EF Deployment)	0	0.018
	P_OPFOR(Detection US)	0.101	0.003
	P_OPFOR(Classification US)	0.808	0
	Number of Undersea Systems	0	0.255
	Number of Effects	0	0.109

The factors highlighted were determined to be insignificant based on the p-values and R^2 values. The most important response for this mission was the mission success and

the most significant factors for this mission based on p-values less than 0.05 would be the kill box size and number of undersea systems. This result makes sense based on the results shown in Figure 44, due to the need of more undersea systems to provide the required effects devices. To illustrate the larger the kill box the more effects devices are required to support the kill box size thus more undersea systems are required. In regards to the other responses the number of effects also came into play because the more systems deployed causes more opportunities for system identification.

The two graphs in Figure 44 and Figure 48 display the number of successful effects deployments compared to both the probability of the undersea system to deploy the effects and the size of the kill box. Based on these graphs it is clear that the XLUUV consistently deployed between 1 and 2 effects devices through all scenarios whereas the LDUUV while it did manage to deploy more effects devices it was less consistent and more often deployed less than 1 device. In addition, the XLUUV was able to deploy effects devices across larger kill boxes. For any kill box larger than 250 nm^2 the LDUUV was unable to deploy any effects devices whereas the XLUUV was successful in deploying effects devices in kill boxes up to 900 nm^2.

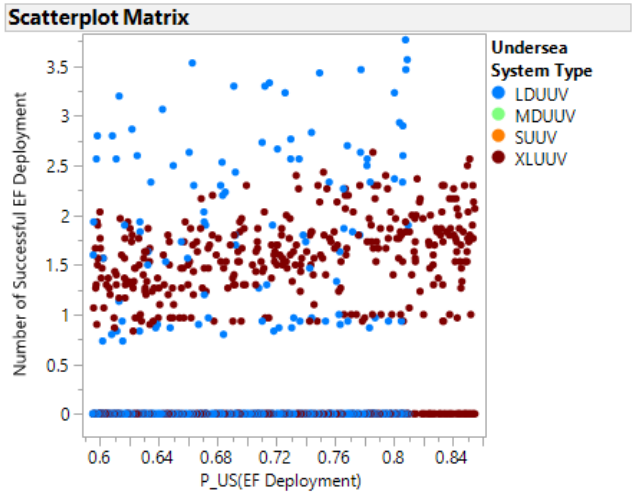


Figure 48. Number of Successful Deployments Compared to the Probability of Deployment

Figure 49 highlights the total utility of both the XLUUV and LDUUV in the success of the effects deployment mission. Based on the ANOVA, if only one undersea system could be selected for a kill box from 0 – 1500 nm^2, the LDUUV is slightly better with a mean mission success rate of 0.093, just 0.009 higher than the XLUUV, suggesting that there is no statistically significant difference between the performance of the systems. This data would likely change however if more undersea systems had been deployed thus increasing the possible kill box sizes available for both the XLUUV and LDUUV.

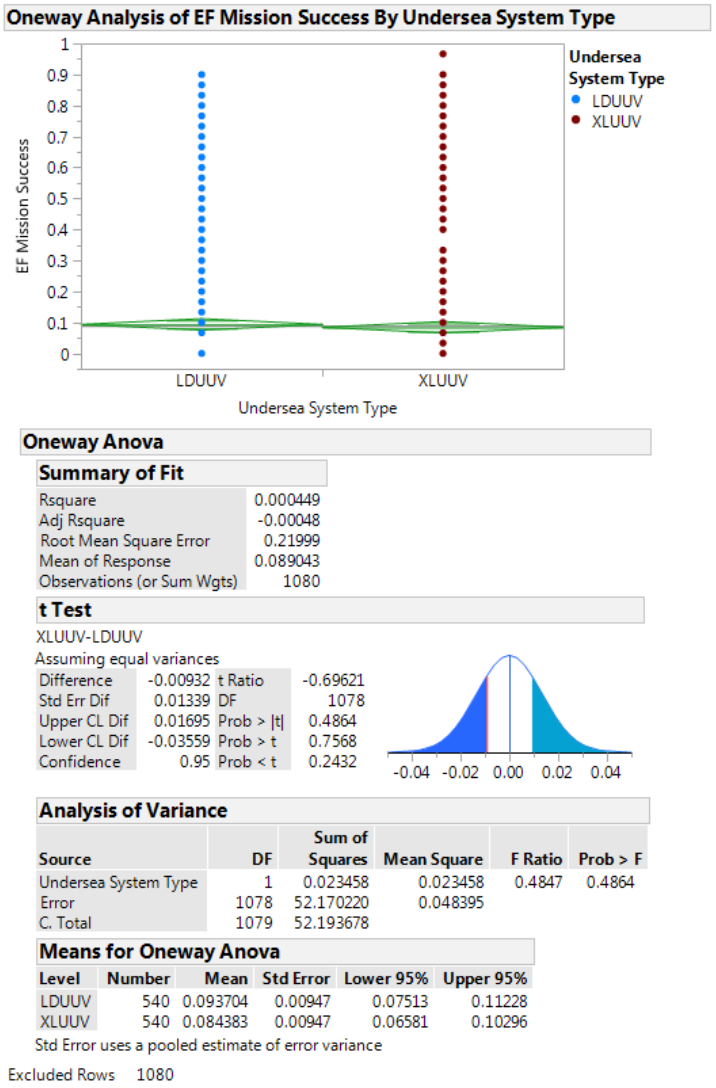


Figure 49. Effects Field Deployment Comparison for Full Kill Box Range

The XLUUV was more consistent in deploying devices but the LDUUV could deploy more due to the increased number of LDUUVs required to support larger kill boxes. It is also important to note that neither 20 LDUUVs nor 1 XLUUV could perform this mission in a kill box larger than 250 nm^2.

4. Engage and Strike

Kill box execution is conducted through the final mission, engage and strike. Since the undersea systems evaluated within the AoA do not have the capabilities to launch effects, the undersea system only assists with the detection, classification, and tracking of the targeted opposing forces within this mission.

Table 20 shows the factors used in this mission. The execution of the kill box is the most complex mission and has the most factors associated with it. The asterisks within the value column denote the factors that variate dependent on the undersea system evaluated. There will be no investigation into the factors that are unrelated to the performance of the undersea system within the mission, displayed in grey within the table.

Table 20. Engage and Strike Mission Factors

Factor	Values
Kill Box Size	1-1500
P_{US}(Detection)	*
P_{US}(Classification)	*
P_{US}(Tracking)	*
P_{ISR}(Detection)	*
P_{ISR}(Classification)	*
P_{ISR}(Tracking)	*
P_{EF}(Lethality)	*
Target Lured	TRUE FALSE
Target Type	ASUW ASW
P_{TARGET}(Detection US)	*

Factor	Values
PTARGET(Detection ISR)	*
PTARGET(Detection EF)	*
PTARGET(Classification US)	*
PTARGET(Classification ISR)	*
PTARGET(Classification EF)	*
PTARGET(Evasion)	*

From the mission, the simulation collected certain MOEs to assess performance, shown below in Table 21.

Table 21. Engage and Strike Mission Responses

Response
Strike Success
Number of Times Target Tracked by Kill box
Likelihood of Identification During Strike

Scatterplots can provide initial trend analysis among all factors against the mission response variables. Figure 50 compares the undersea system factors against the MOEs with each undersea system represented with a different color. The kill box size and number of undersea systems shows positive non-linear trends within each response. The spread of data within each response grows larger as both kill box size and number of systems increase. It is hard to determine if the XLUUV performed better or worse than the other UUVs when comparing across all undersea systems. It is likely that the extreme variance in system performance is a resultant of number of undersea systems within the kill box. Unlike the previous three missions, there is a slight positive relationship between strike success and kill box size. This is due to the number of effects deployed is directly related to the size of the kill box. The larger the kill box, the larger the number of effects that must

be deployed in order to obtain full coverage of the kill box. Thus, there are more engagement attempts available. Similarly, the number of undersea systems increases as the kill box size increases which directly correlates to the increase in the likelihood of identification. The detection, classification, tracking probabilities for both the undersea system and the targeted opposing forces do not have an immediate correlation but provide an impact on the number of times the target is tracked within the kill box. Additional scatterplots are included in Appendix D; however, the team developed no actionable insights based on these figures.

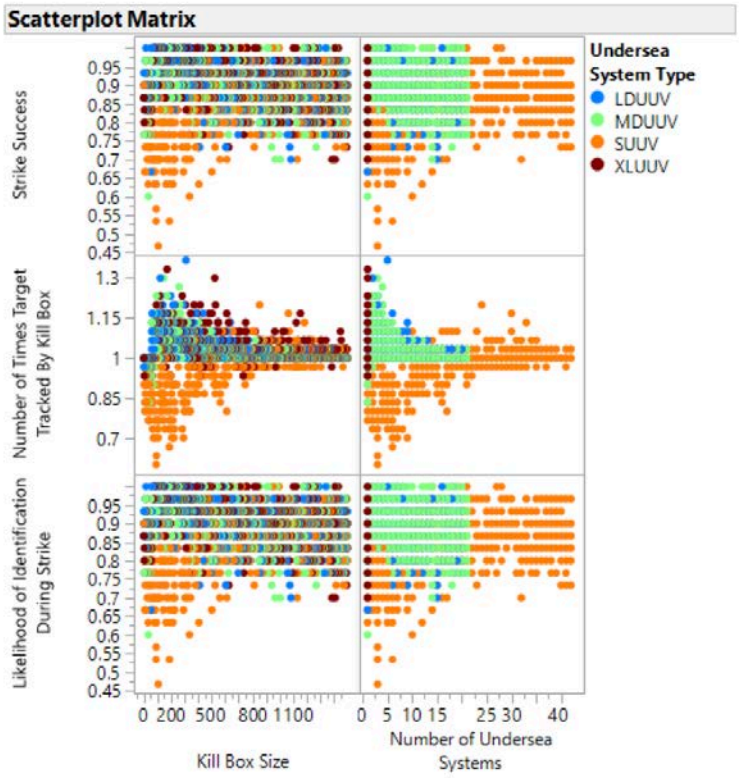


Figure 50. Engage and Strike Mission Scatterplot

The response variables can be evaluated further under the unique categorical types associated with the final mission: target type and way of entrance into the kill box. Figure 51 and Figure 52 categorize all responses by target type, entrance method, and undersea system type. The undersea systems are equally effective against both surface and sub-

surface targets. Additionally, the way of entrance into the kill box also does not appear to have a correlation with mission success.

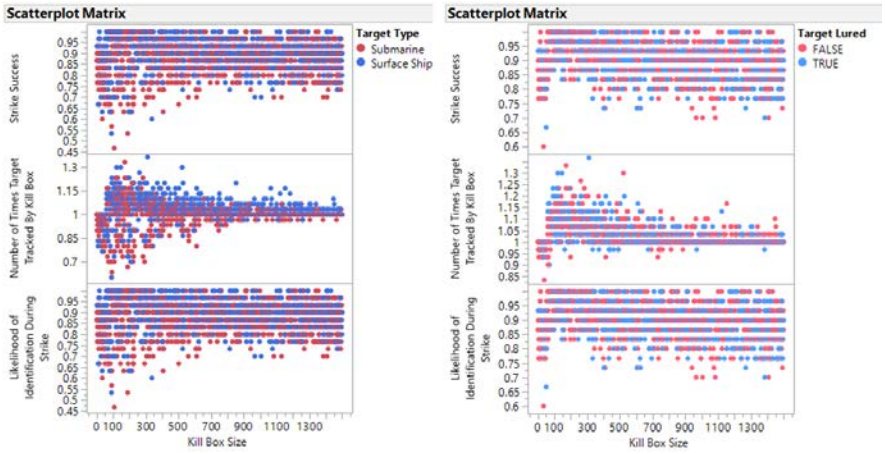


Figure 51. Engage and Strike Mission Target Responses

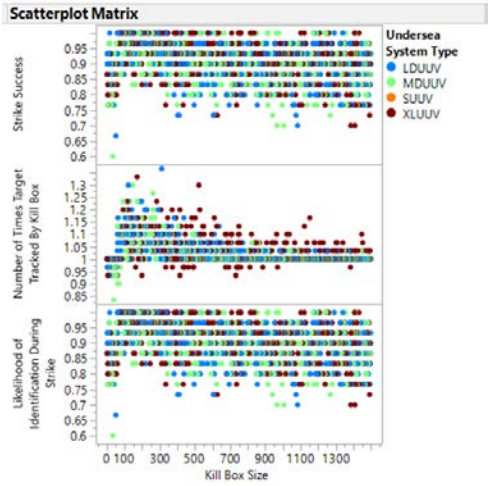


Figure 52. Engage and Strike Mission Responses

The graphic in Figure 53 indicates that the SUUV does not have the capability to aid in the detection, classification, and tracking of the target. Due to this, Team Leviathan will not use the SUUV further in analysis.

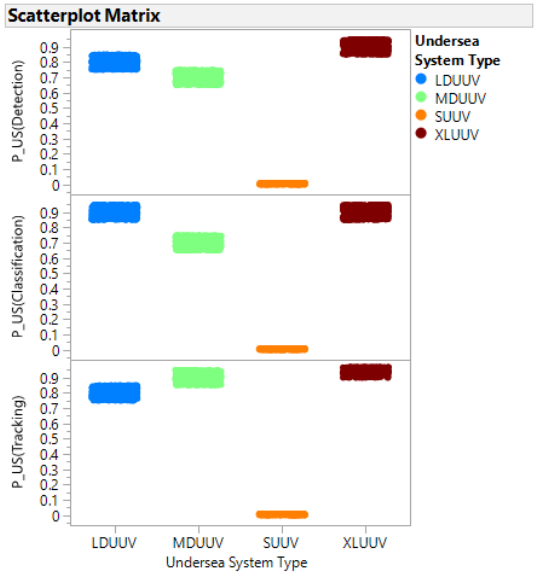


Figure 53. Detection, Classification and Tracking Capability across Undersea System Type

Each response factor can be decomposed based on undersea system type in order to view the distribution of the results. Shown in Figure 54 are the distributions across the XLUUV. The overall mean probability of mission success is about 91%. The kill box is able to track the target 1.03 times successfully; this means that on average, the kill box was able to obtain mission kill with only tracking the target once. Despite this high success rate, the XLUUV's likelihood of system identification is about 0.9 meaning that identification by the targeted opposing force is likely. The interquartile range for strike success is (0.87, 0.97) with the densest region of data found within the range. The data is negatively skewed with a heavy right tail.

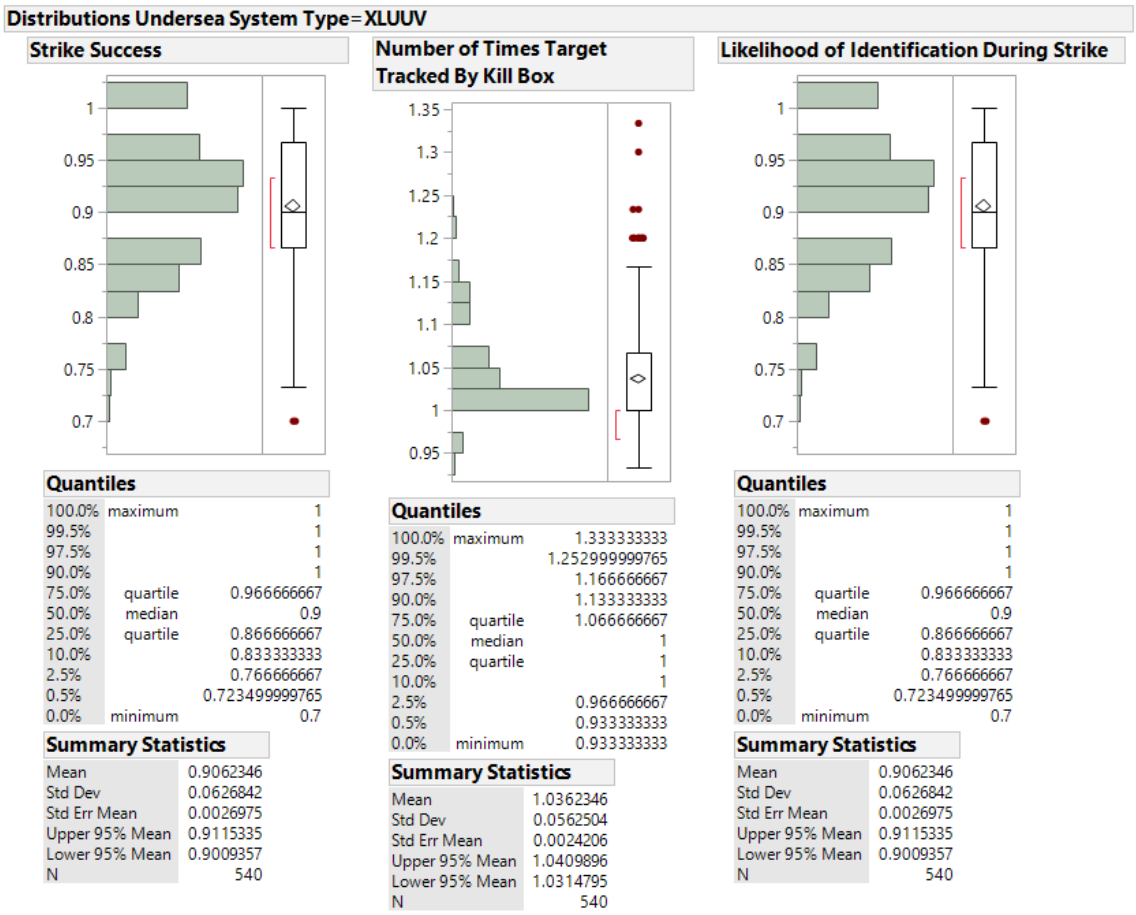


Figure 54. XLUUV Response Distribution for Engage and Strike

Shown in Figure 55 are the distributions of the different responses for the LDUUV. The average probability of mission success is marginally lower than the XLUUV, at about 90%. The kill box is able to track the target 1.03 times again. Since this value is similar to the XLUUV, it may be possible that the ISR field is doing the majority of the tracking within the simulation. Once again, the LDUUV's likelihood of system identification is about 0.9 meaning that the undersea system is very likely to be seen by the targeted opposing force. The interquartile range for strike success is (0.87, 0.97) with the densest region of data found within the range. The data is negatively skewed with a heavy right tail.

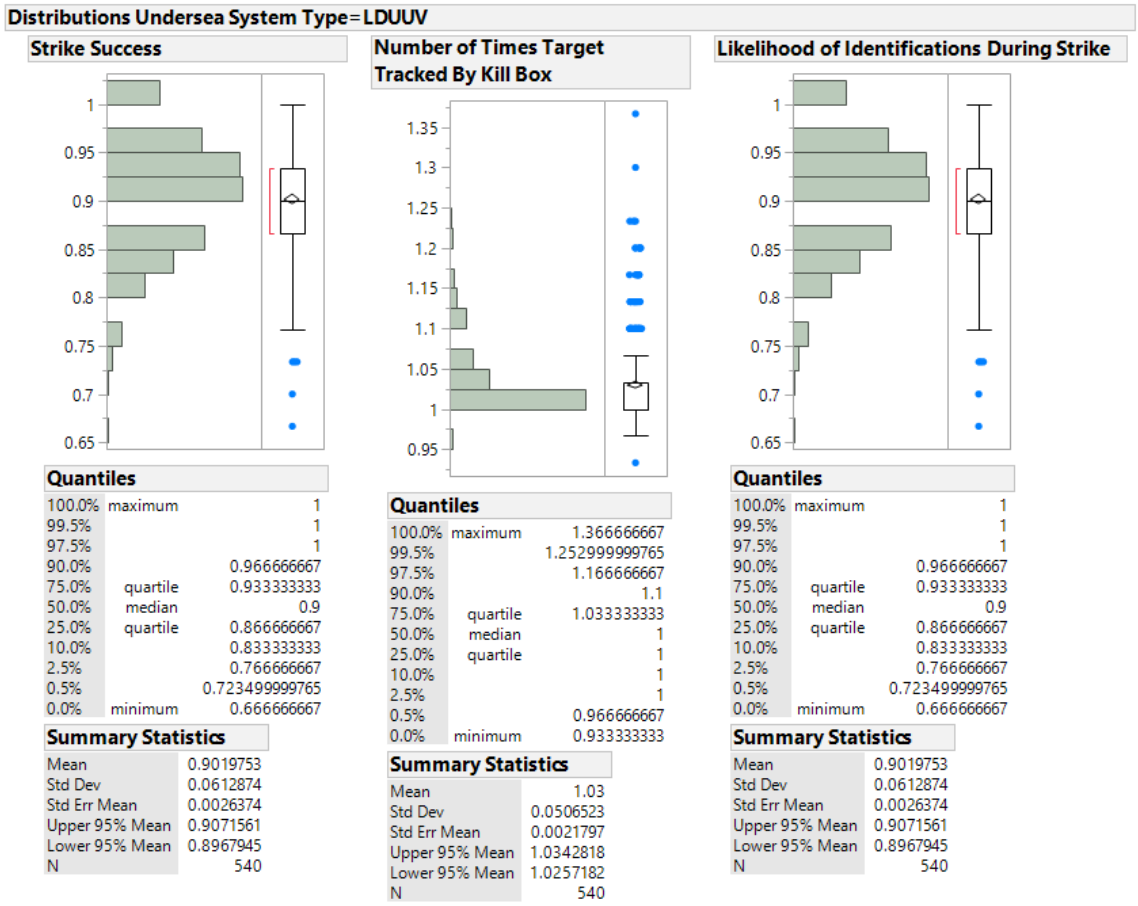


Figure 55. LDUUV Response Distributions for Engage and Strike

Shown in Figure 56 are the distributions of the different responses for the MDUUV. The overall mean probability of mission success is the lowest out of the three systems (excluding the SUUV), at about 90%. The kill box is able to track the target 1.03 times successfully. Since this value is almost identical to the other two systems, it may be possible that the ISR field is doing the majority of the tracking. This high success rate, the LDUUV's likelihood of system identification is about 0.89 meaning that identification by the targeted opposing force is likely. The interquartile range for strike success is (0.87, 0.97) with the densest region of data found within the range. The data is negatively skewed with a heavy right tail. Since the interquartile range of the data across all three systems is identical, only the mean of the data is falling lower.

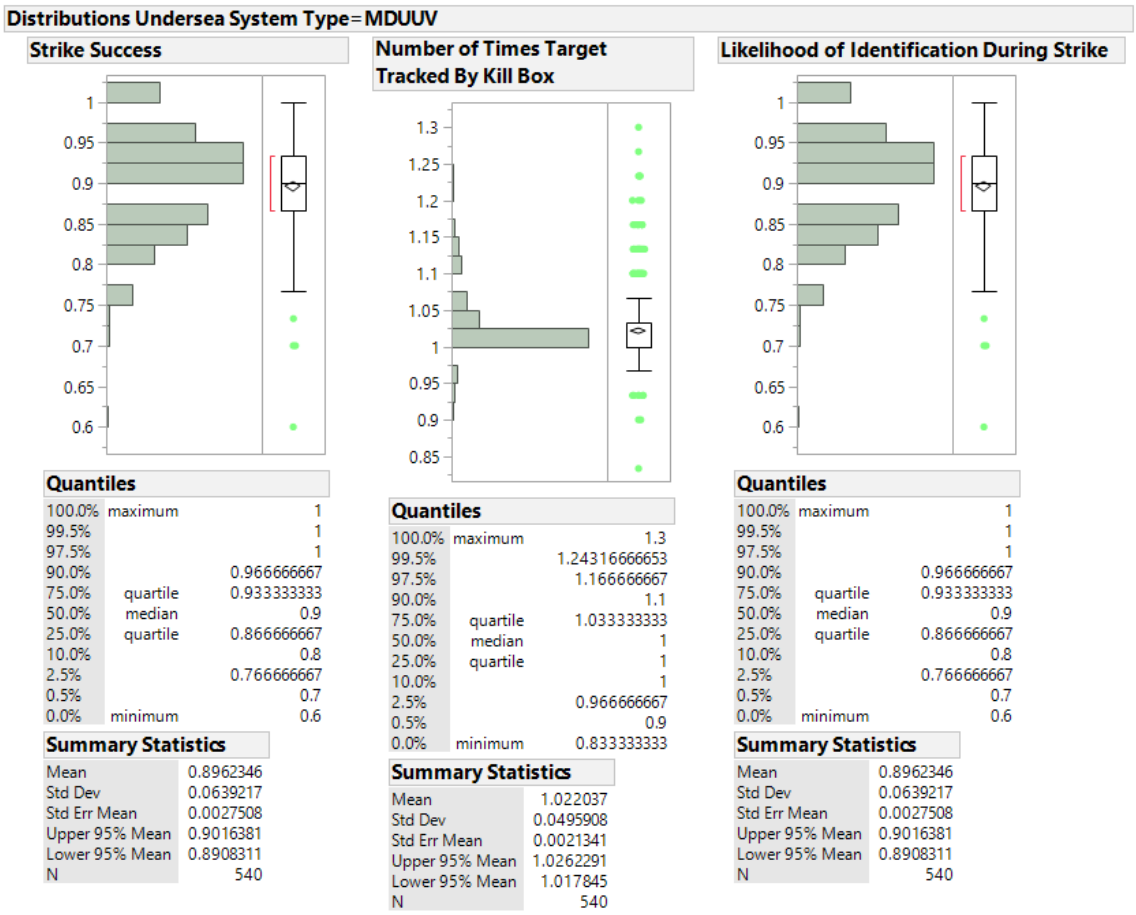


Figure 56. MDUUV Response Distribution for Engage and Strike

Prior to going further, the team performed a response screening to find which factors were not statistically significant for the responses. Table 22 shows the results of this screening. The colored cells correlate to the circled relationships within Table 22; these factors show the highest R^2 value and have a large impact on the system performance.

Table 22. Engage and Strike Response Screening

Response	Factor	P-Value	R-Square
Strike Success	Kill Box Size	0	0.080
	P_US(Detection)	0	0.011
	P_US(Classification)	0	0.009
	P_US(Tracking)	0.085	0.002
	Target Lured	0.732	0
	P_TARGET(Detection US)	0	0.013
	P_TARGET(Classification US)	0	0.010
	Target Type	0	0.022
	Number of Undersea Systems	0	0.043
Number of Times Target Tracked By Kill Box	Kill Box Size	0	0.187
	P_US(Detection)	0	0.026
	P_US(Classification)	0	0.021
	P_US(Tracking)	0.003	0.006
	Target Lured	0.571	0
	P_TARGET(Detection US)	0	0.016
	P_TARGET(Classification US)	0	0.025
	Target Type	0	0.038
	Number of Undersea Systems	0	0.115
Likelihood of Identification During Strike	Kill Box Size	0	0.080
	P_US(Detection)	0	0.011
	P_US(Classification)	0	0.009
	P_US(Tracking)	0.085	0.002
	Target Lured	0.732	0
	P_TARGET(Detection US)	0	0.013
	P_TARGET(Classification US)	0	0.010
	Target Type	0	0.022
	Number of Undersea Systems	0	0.043

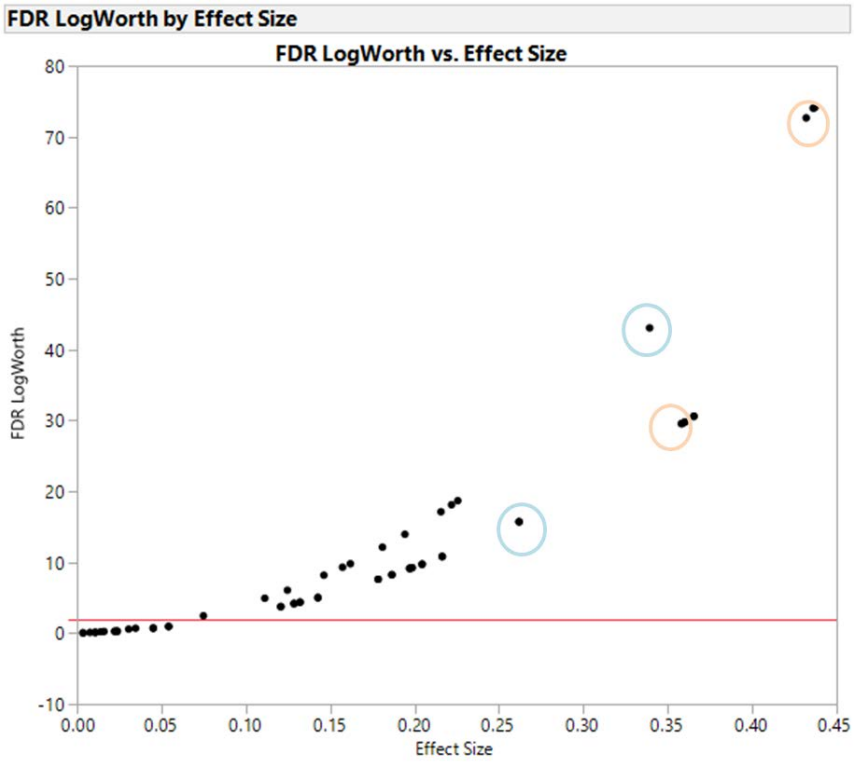


Figure 57. Engage and Strike Factors Effect Size

The two most important factors were the kill box size and number of undersea systems. Despite having moderately large effects sizes, the R^2 value for the factor versus the responses small. This means that while the factors are statistically significant, their operational significance may be minimal. Shown in Figure 58 and Figure 59 are the two factors of the engage and strike mission binned against each independent undersea system type and kill box type. Looking at both graphics, performance within the mission is almost equal across each undersea system. This corroborates the conclusion made from the distributions that the undersea system assisting with the detection, classification, and tracking of the target provides no added or enhanced performance within kill box execution. Overall, since the likelihood of system identification is high, this mission puts a high value asset at risk with no added benefit.

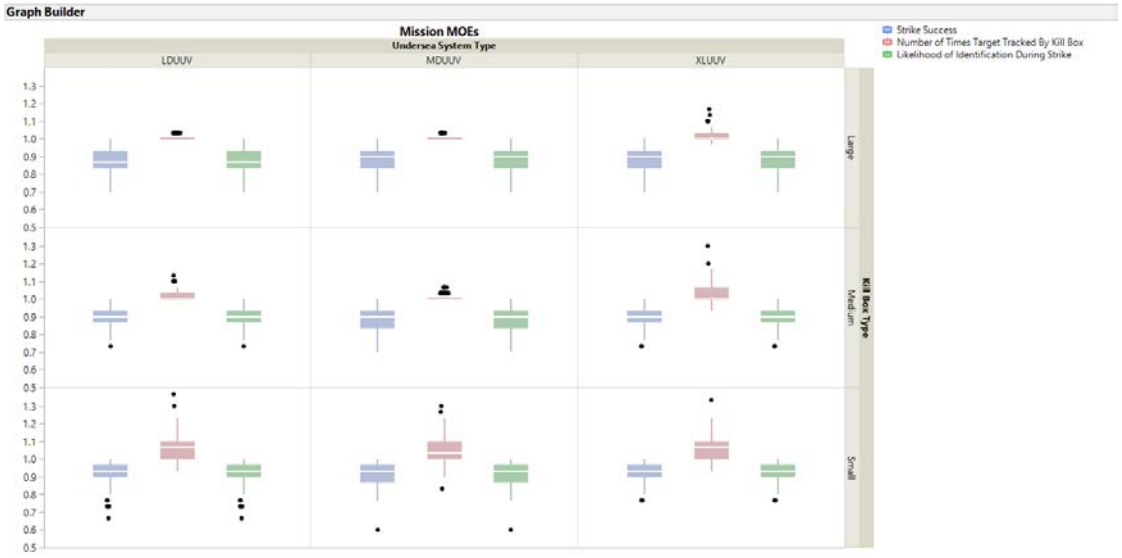


Figure 58. Mission Responses Binned by Kill Box Type

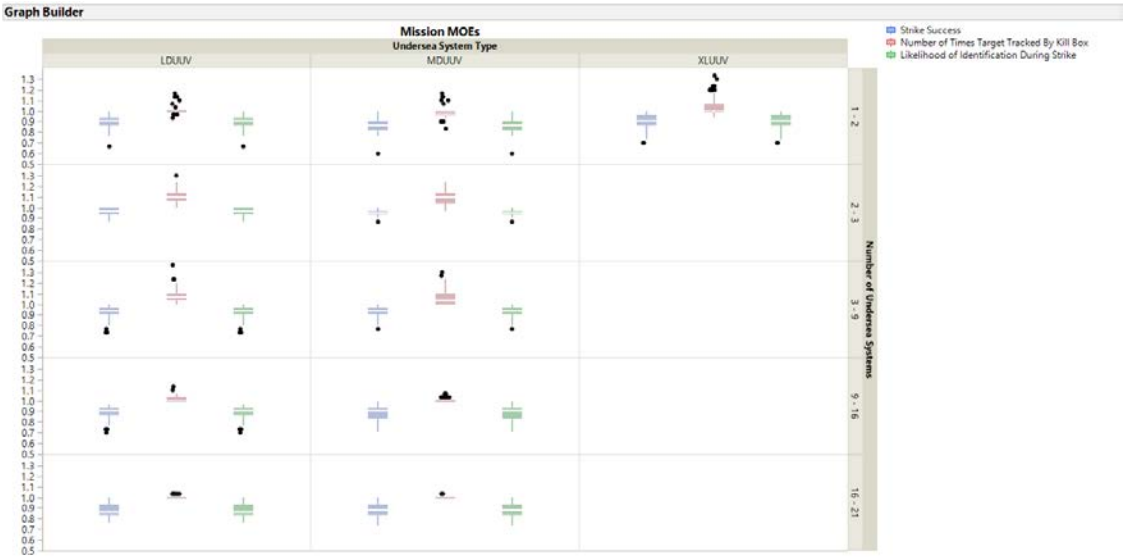


Figure 59. Mission Responses Binned by Number of Undersea Systems

C. SIMULATIONS LIMITATION ANALYSIS

Team Leviathan found some simulation limitations from the investigation into the XLUUV's utility within the kill box execution and deployment.

Since the initial mission scenario set up determined the number of required undersea systems based on the IPOE mission alone, there were mission failures within the deployment of the ISR devices and the effects devices since the total number of devices across all payloads did not meet the number required for the field resolution. The team remedied this by taking into account the minimum number of systems required to complete all missions. Note that if the deployment of a kill box was executed in real life, the number of systems deployed is entirely dependent on the command center and system availability.

Additionally, for every time a system conducted a portion of each mission, it had the opportunity for detection and classification by the opposing forces. After analysis into the DOE mission scenarios, the probabilities assigned to the opposing forces may have been higher than what one would see in reality.

Within the final engage and strike mission, the simulation gives the target the ability to evade. In a real world operational scenario, it is unlikely that a target would attempt to exit the kill box prior to the strike engagement.

D. MISSION ANALYSIS SUMMARY

This simulation provided performance data for each undersea system type for each mission and the kill box execution. The AoA analyzed this performance data for each mission independently; however, the analysis described below was used to determine the utility of each undersea system regarding the overall kill box deployment and effective size of a kill box for each mission.

Based on Figure 60 it is clear that the XLUUV had on average a higher rate of deployment success for kill boxes up to 250 nm^2.

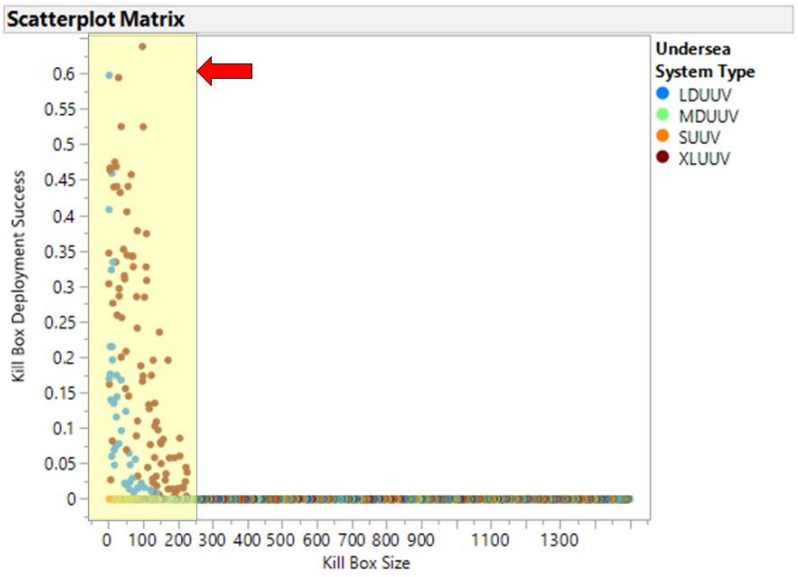


Figure 60. Kill Box Deployment Success

Each of the different missions resulted in varying ranges of effectiveness, presuming that 80% effectiveness is acceptable mission risk. In the IPOE mission, the only successful undersea system to meet that 80% threshold for success was the MDUUV; this was for a 100 nm^2 kill box. For the ISR mission the MDUUV, LDUUV, and XLUUV were capable of deploying a 120 nm^2 with an average success rate of 80%. Only the LDUUV and XLUUV performed the effects deployment mission, both of which were limited to a kill box just under 100 nm^2 based on the required 80% success rate. The undersea systems did not directly influence the success rate of the engage and strike mission; however, this mission did rely on the success of the previous three missions. Based on the results of the current simulation the only undersea systems capable of performing missions one through three successfully are the LDUUV, and XLUUV. While the MDUUV and SUUV have some value for the IPOE mission, they should not be considered for any of the other missions. Additionally, it is important to note that due to limitations of the current simulation data the kill box size has been restricted for the LDUUV and XLUUV during ISR and effects deployment. This limitation may not have an effect on the results of the current AoA, but the team recommends reevaluation in the future.

VI. CONCLUSION

A. MISSION CONCLUSION

Team Leviathan's goal of this project was to assist in the development of a framework definition for seabed warfare by defining an initial operational concept, a set of requirements for seabed warfare, and an open mission architecture for seabed warfare. The AoA analyzed the XLUUV for its potential utility as an enabler of seabed warfare through a developed simulation, with the intention of identifying key performance drivers to assist in creating more informed seabed warfare requirements. Based on those research objectives, conclusions were determined in three areas: mission conclusions, kill box conclusions, and UUV conclusions.

Based on the results of the simulation and the analysis performed on that data, the team made the following conclusions for each mission. Due to the data classification, project timelines, and simulation limitations, the data used within the mission analysis is based on unclassified data, general naval knowledge, and generic device information. Due to this, Team Leviathan based their conclusions entirely on the mission scenarios analyzed. The missions were evaluated independently so that the utility of each undersea system could be identified for each mission without being influenced by the outcome of the previous missions.

For the first mission of kill box deployment, IPOE, within a medium (500-999 nm^2) to large (1000-1500 nm^2) kill box, the highest performing undersea system was the XLUUV. However, it is important to note that the XLUUV had an average success rate of 45%, meaning that even the top performing undersea system was below an average success rate of even 50%. The medium and large kill boxes were far more difficult to conduct IPOE in than the small kill box (0 to 499 nm^2) even for a platform such as the XLUUV. On average, the other three undersea systems evaluated within the AoA performed as well if not better than the XLUUV for small kill box.

For the second mission of kill box deployment, deployment of ISR devices, the analysis excluded the SUUV because it was unable to carry any ISR devices. The analysis

for the mission demonstrated that the MDUUV was the most capable followed by the LDUUV and the XLUUV. The MDUUV is the ideal asset for ISR based on the average success rate and the fact that it had the lowest rate of detection. The best size that the single XLUUV was able to deploy was a 104 nm^2 kill box, with a success rate of 80% and was able to deploy 1.5 devices on average. Since the simulation only deployed one XLUUV, the MDUUV and LDUUV dominated the number of ISR devices it could deploy.

The third mission and final mission for kill box, the analysis evaluated only the XLUUV and LDUUV due to the MDUUV's and SUUV's inability to carry an effects device payload. This is partially due to the predetermined size of effects device simulated. In future research, the two systems may prove to be capable of the mission if the simulation utilized other types of effects. The analysis for the mission demonstrated that the LDUUV was only slightly favored over the XLUUV, and this was due to the average number of successful deployments by the LDUUV. While the kill box size limited the LDUUV, only ever succeeding in a kill box 250 nm^2 or smaller, the success rate was nearly double that of the XLUUV. The LDUUV also had a higher rate of overall mission success than the XLUUV, initially making it the clear winner. However, based on a limitation found within the simulation, only one XLUUV was every deployed. With more XLUUVs present the LDUUV may not have been the top performing undersea system; the trade-off would be on risk versus reward of sending additional high valued assets into the field. As it stands now, the LDUUV is still the optimal choice for this mission based on the current results of the simulation and assuming the kill box is smaller than 250 nm^2.

Finally, the last mission evaluated was engage and strike. While the analysis assessed all undersea systems for this mission, it was clear that none of the undersea systems provided any value to the mission. This was especially true in the case of the XLUUV because it simply became a target for the opposing force and became a high value asset unnecessarily put at risk. The only benefit in using an undersea system would be to use it to lure a target into the kill box; however, again, the XLUUV would not be ideal for this.

B. KILL BOX CONCLUSION

Team Leviathan's proposed seabed warfare capabilities are comprised of seven different capability areas found within the CV-2: anti-submarine warfare; anti-surface warfare; mine warfare; electromagnetic maneuver warfare; intelligence, surveillance and reconnaissance; military deception; and extensions to strike. Recall that the definition of a kill box is a three-dimensional area used to enable the integration of joint fires while reducing the coordination required from commanders to fulfil the mission (Army 2005). This involves four key missions: intelligence preparation of the operational area; intelligence surveillance, and reconnaissance, effects field deployment; and engage and strike. Team Leviathan evaluated each of these missions for effectiveness under a series of presumed mission constraints and assumptions for the mission itself while under simulation. The results of the simulation allowed for the determination of the effective size of a seabed-based kill box in each of the different mission areas.

The simulation has directly displayed five of the seven capabilities of seabed warfare: anti-submarine warfare; anti-surface warfare; intelligence surveillance, and reconnaissance; military deception; and extensions to strike. The two capabilities not directly modeled by this simulation, mine warfare and electromagnetic maneuver warfare, could be implemented into the effects deployment and engage and strike missions with little impact. However, this addition may change the classification for this simulation. Team Leviathan's results indicate that a kill box ranging between 0–200 nm^2 is the most optimal size for Operation Leviathan's kill box.

C. FUTURE UUV CAPABILTY

Throughout the generation of the seabed kill box simulation and data input table, the team assumed current UUV technology. Future technology of UUVs can greatly improve the outcome of the simulation and the utilization of UUVs within the kill box. For example, if a UUV developed the technology to evade, or avoid a threat that detected it, the probability of mission success would increase. The ISR mission simulated assumes sea-mounted sensing devices. If the U.S. Navy retrofitted an ISR device into a payload of an SUUV, it can be deployed by a surface ship or from an XLUUV to be successful at the ISR

mission. If the XLUUV has the same side-scan sonar payload capability as the MDUUV to conduct IPOE, then it can provide an attempt at the IPOE mission and a full mission solution for all missions. In addition, if future capability allows a UUV to release an effects device, then the third mission is not necessary, and this allows for execution of the kill box without placing high value assets at risk. The XLUUV with split payload of a side scanner with full IPOE capability, ISR devices, and effects devices would provide an ideal system of systems for both deployment and execution of the kill box.

D. FUTURE WORK

There can be multiple different functionalities integrated into the simulation to enhance its ability to evaluate the performance of different undersea systems. While this AoA was evaluated independent of environmental factors, it would be highly beneficial to develop a model that can create dynamic (changes per mission within each scenario) and static (remains constant throughout the entire scenario) environment thresholds. This would consider various factors about the tactical area of interest such as seabed composition, salinity, depth, temperature, and others.

Another feature could be the implementation of varying field resolutions for the ISR and effects devices. This would allow the investigation into system performance if the kill box has reduced capabilities due to deployment failures. Additionally, integrating an AoA for the ISR and effects device would reveal the optimum performance for the seabed mounted sensors as well as the optimum pairing of devices for certain scenarios. Specific kill box tactics or actual system parameters would to allow for the evaluation of actual performance given scenarios of interest.

An important factor of UUV operations is the Integrated Logistics Support (ILS) Plan for maintaining and deploying UUVs. For example, the amount of time needed to recharge batteries and the maintenance down time in-between missions can be integrated into the simulation. Some UUVs require the full vehicle to download data and recharge batteries while others have field-replaceable batteries to reduce downtime between missions. If the UUVs lack the field-replaceable batteries, this would entail the systems that need to be recharged to head back to the command center as opposed to staying out in

the field for the duration of the kill box deployment. Since the simulation did not incorporate logistics, this could be paired with the potential for changes in the intelligence gathered during IPOE.

Team Leviathan's research and development from this seabed warfare work included the following topic areas: mission deployment/threat evaluation, employment/tactical research, system specification development, and payload development. We propose focusing future research in a classified environment to add in tactics, threat evaluations, and classified values of detectability, classification, lethality, etc., for each UUV and target will greatly increase the value of the simulation results. Future research can focus on using the simulation to define the ideal UUV capability needed for seabed warfare to develop a system specification. For instance, if a UUV is receiving upgrades to a classification algorithm onboard, this simulation can aid in determining what required probability to make the classification algorithm successful. Generating a system matrix for better assessment of a system's utility within the kill box or seabed warfare scenario can fully address all capabilities of the CV-2 and determine where capability gaps are. As a next step, examining the simulation from a payload integration perspective can generate a list of payload options for seabed warfare with an AoA on the payload and vehicle architectures to determine the ideal system of systems.

Team Leviathan's research and recommendations provided in this paper detail the current work in the seabed warfare mission area and present future work to further define the framework definition of seabed warfare as a new mission area in the Navy.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. SEABED WARFARE CAPABILITIES

This appendix presents the capability taxonomy, CV-2, developed in Innoslate to describe the capability hierarchy within seabed warfare. The coloring of the capabilities indicates whether the team considers the capability offensive (purple), defensive (red) or dual (green).

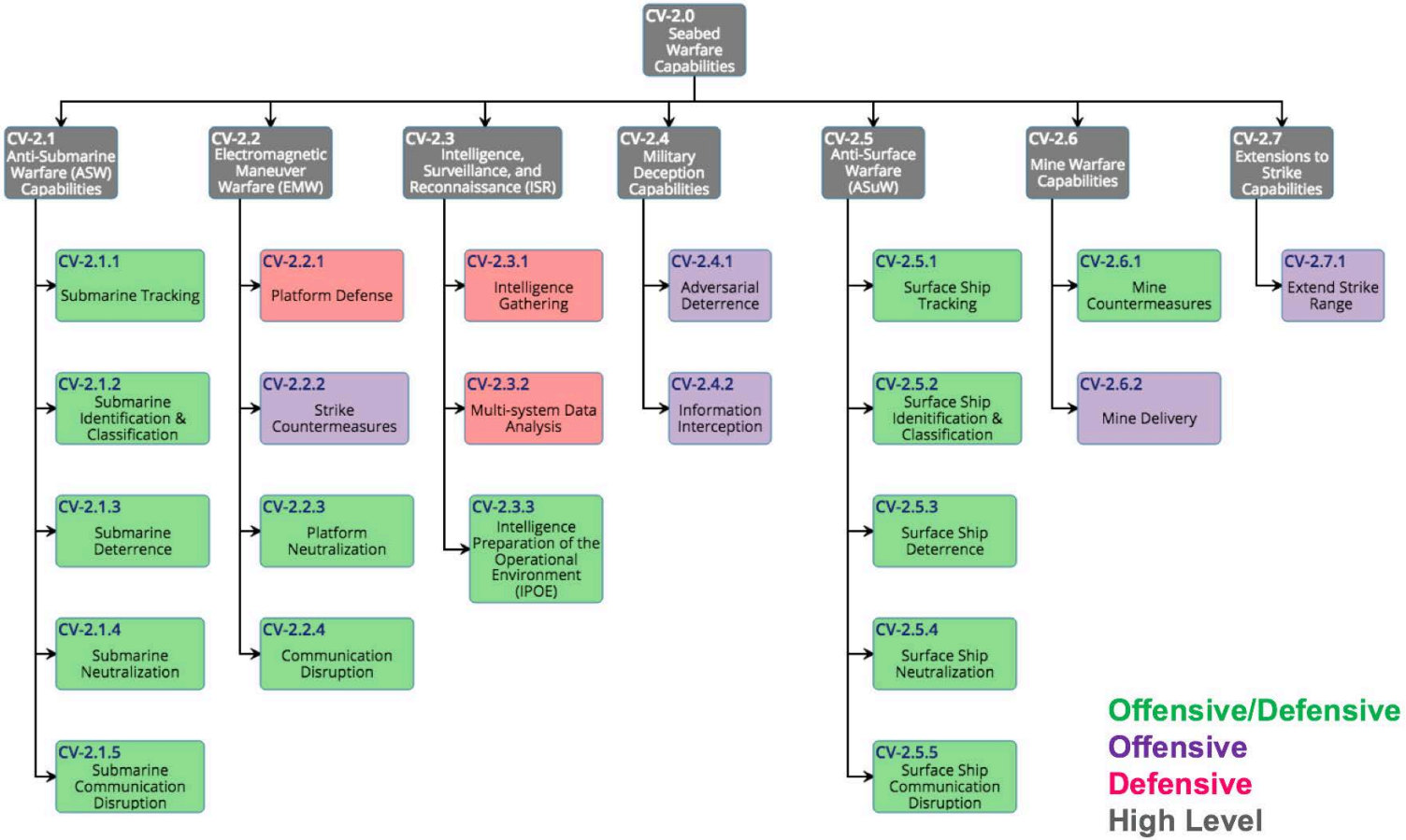


Figure 61. CV-2: Seabed Warfare Capabilities Diagram

APPENDIX B. MISSION ACTION DIAGRAMS

This appendix presents the detailed action diagrams developed in Innoslate to describe the operational activities associated with the execution of each seabed warfare mission.

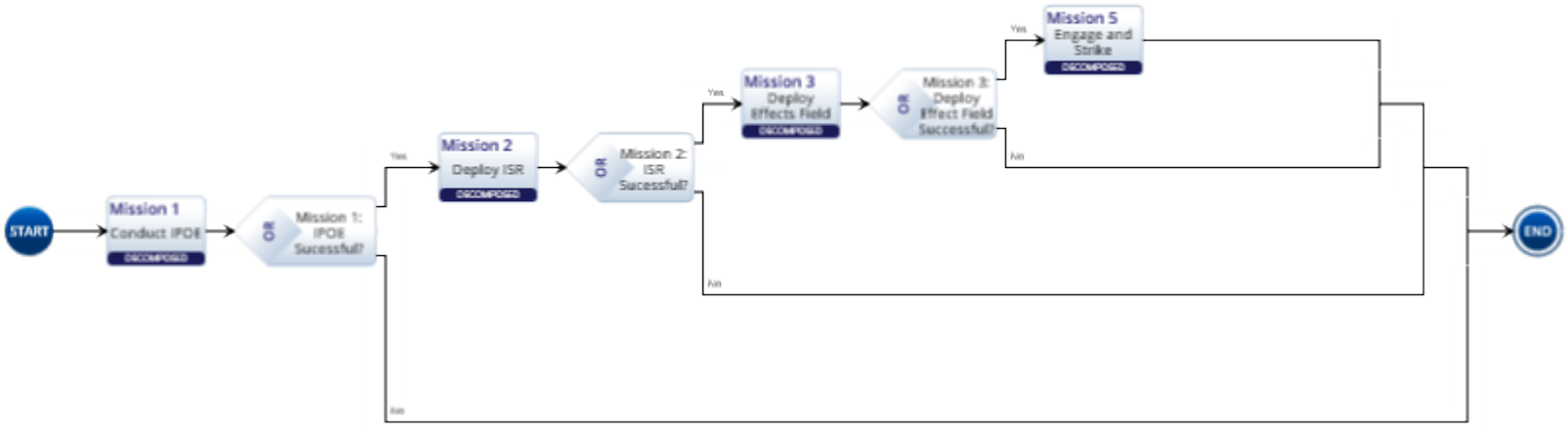


Figure 62. High-Level Operation Leviathan

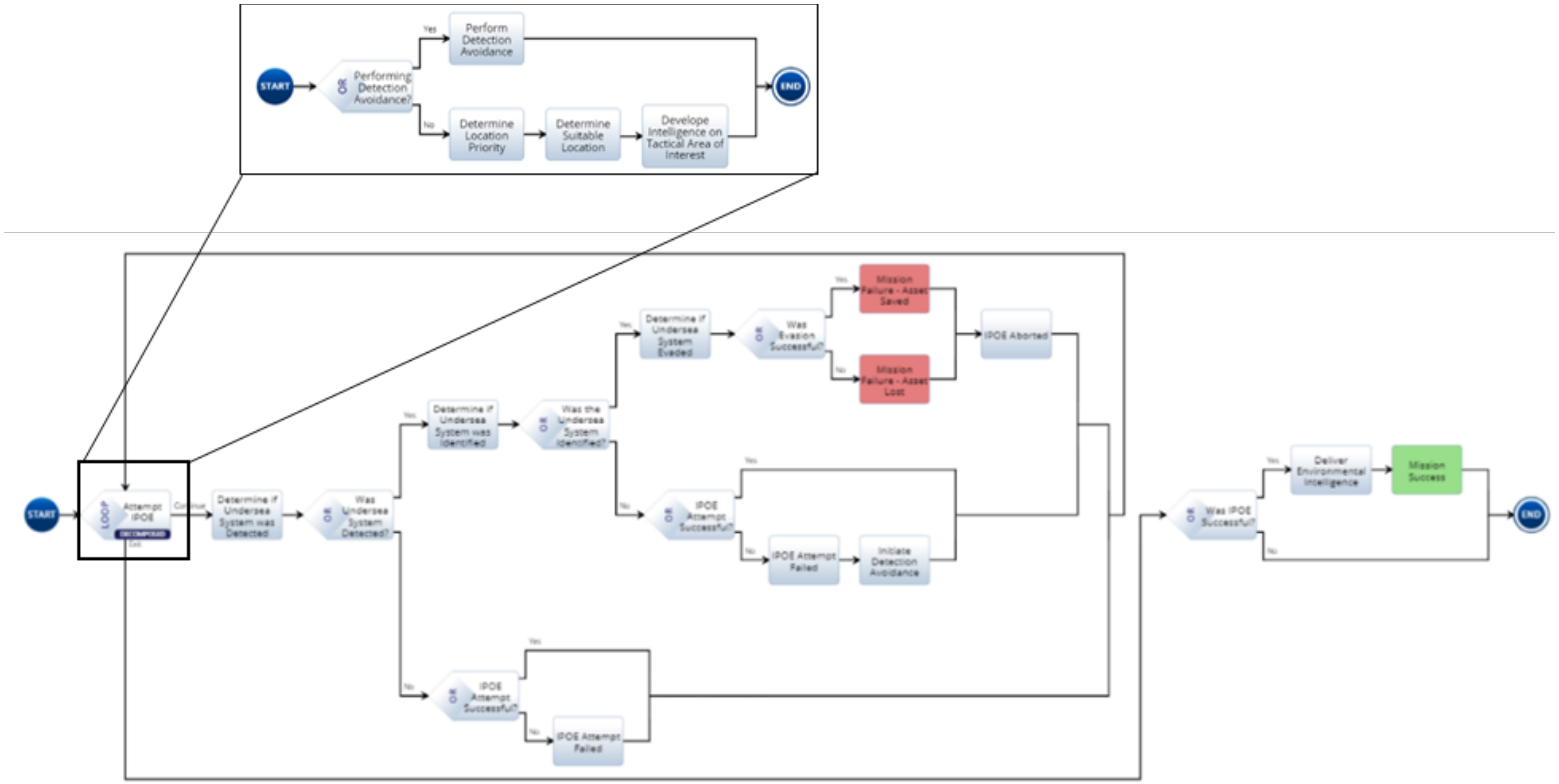


Figure 63. Mission 1: IPOE

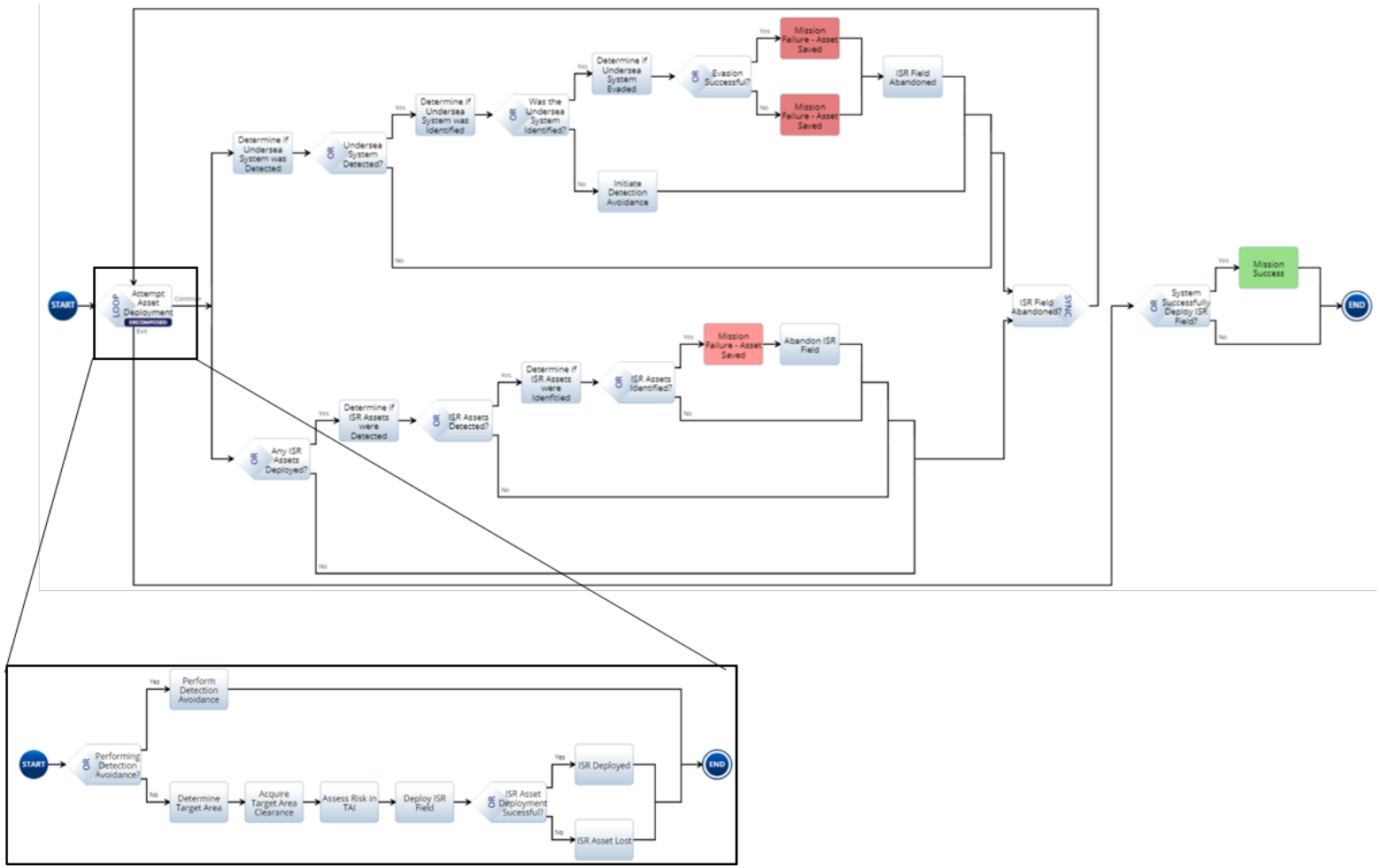


Figure 64. Mission 2: Intelligence, Surveillance, and Reconnaissance

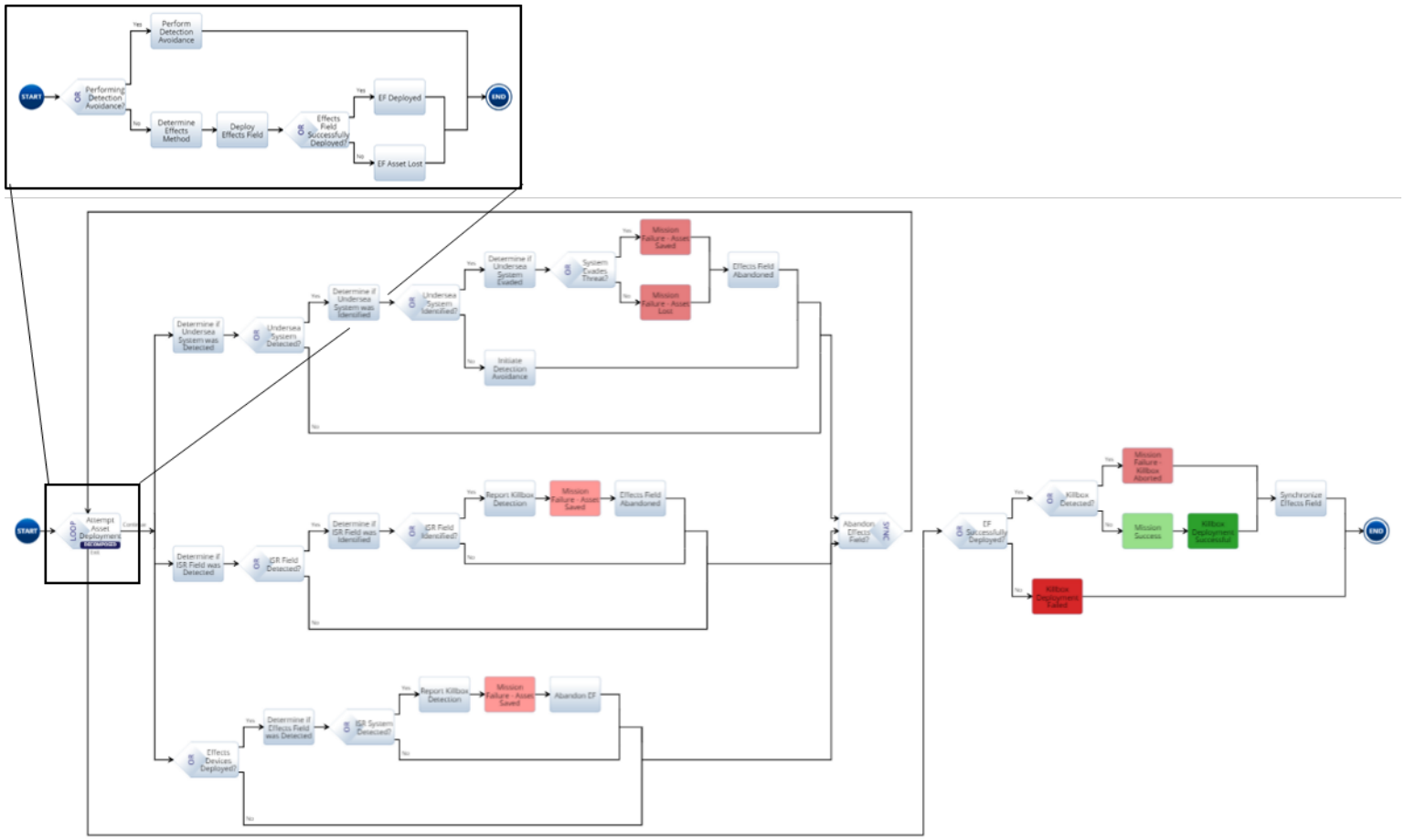


Figure 65. Mission 3: Effects Field Deployment

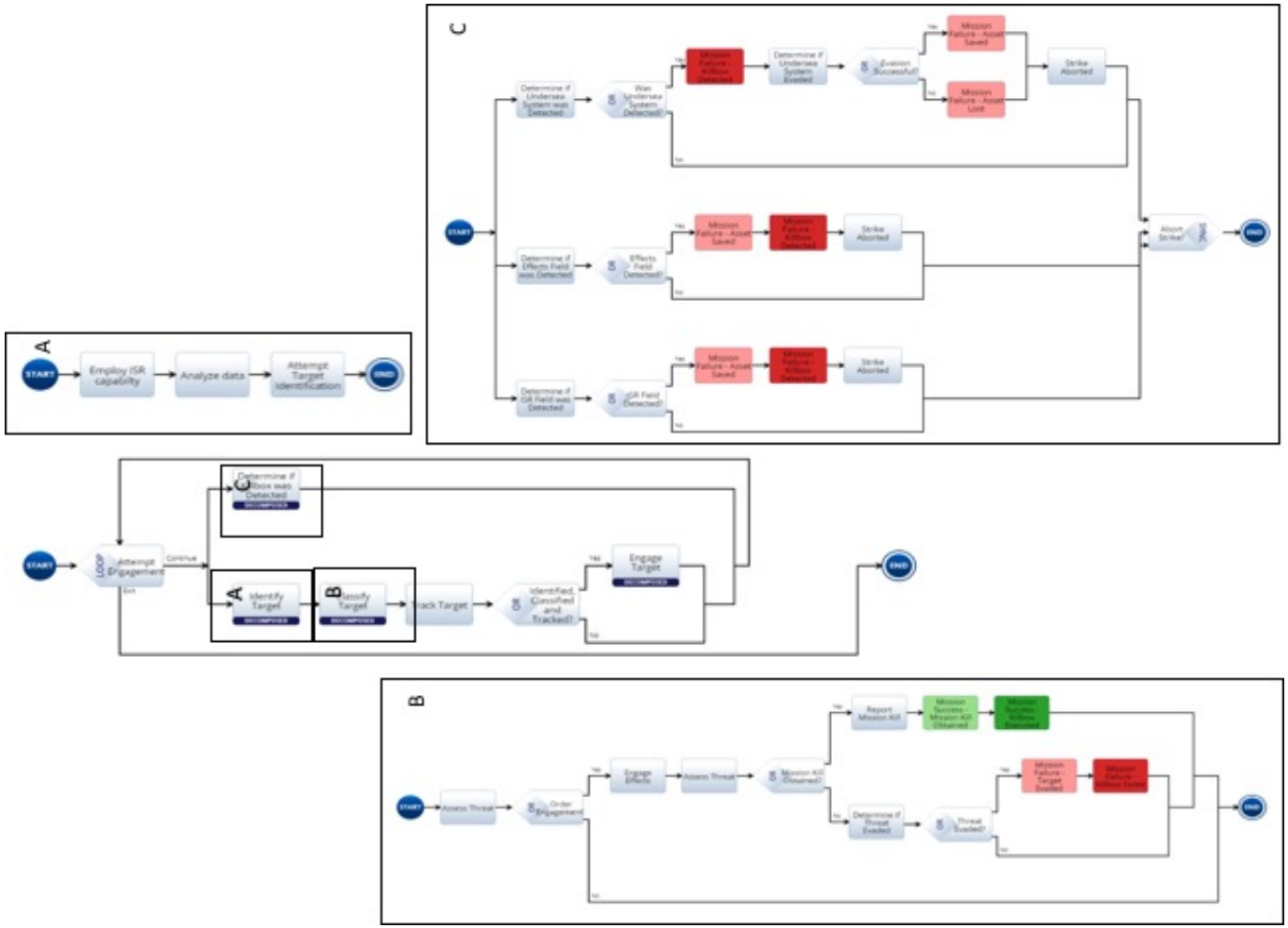


Figure 66. Mission 4: Engage and Strike

APPENDIX C. MODELING AND SIMULATION

This appendix presents a decomposition of the attributes included in the operational simulations, to include the attribute type, definition, related factors, related systems, and associated missions.

Table 23. Attribute Decomposition

Attribute Decomposition					
Attribute	**Type**	**Definition**	**FACTORS**	**System(s)**	**Mission(s)**
Detection	Probability	Probability that the presence of the system is discovered by another system (the act of being detected)	Size (diameter), Signature (Acoustic and non-acoustic), Speed	Blue Undersea System	All
				ISR Systems	
				Effects Systems	
				OPFOR Naval Systems	
		Probability that the system discovers the presence of another system (the act of detecting)	Reliability, situational awareness capabilities	Blue Undersea System	All
				ISR Systems	
				OPFOR Naval Systems	
Classification	Probability	Probabilty that the system is recognized by another system (the act of being identified)	Size, Signature (Acoustic and non-acoustic)	Blue Undersea System	All
				ISR Systems	
				Effects Systems	
				OPFOR Naval Systems	
		Probability that the system recognizes another system (the act of identifying)	Reliability, situational awareness capabilities	Blue Undersea System	Engage & Strike
				ISR Systems	
				OPFOR Naval Systems	
Tracking	Probability	Probability of the system maintain knowledge of another system's position	Speed, Reliability, situational awareness capabilities	Blue Undersea System	Engage & Strike
				ISR Systems	

Ability to Evade	Probability	Probability of the system escaping another system	Speed, Endurance, Depth	Blue Undersea System OPFOR Naval Systems	All
Lethality	Probability	Probability of obtaining a mission kill on another system	Reliability of the effects devices, range/coverage of effects device	Blue Undersea System Effects Systems	Engage & Strike
Deployment	Probability	Probability of successful device placement within the region	Reliability of system being deployed, Reliability of system deploying the device	ISR Systems	ISR
				Effects Systems	Effects Field Delivery
IPOE	Probability	Probability of the system conducting valid IPOE on the tactical area of interest	Reliability, Acoustic sonar capability	Blue Undersea System	IPOE
Number of IPOE Attempts	Integer	Number of attempts available within the mission	Energy, Endurance, Survey Speed	Blue Undersea System	IPOE
Number of ISR Devices	Integer	Undersea system payload capacity for ISR systems	Size of the payload, size of ISR System	Blue Undersea System ISR Systems	ISR
		Number of devices required for field resolution	Range of system, Field Resolution (Mission Criteria), Kill box size	ISR Systems	ISR
		Total Deployed	Reliability of IRS System, Reliability of system deploying the device	ISR Systems	ISR Engage & Strike
Number of Effects	Integer	Undersea system payload capacity of Effect devices	Size of the payload, size of effects device	Blue Undersea System Effects Field	Effects Field Delivery
		Number of devices required for field resolution	Range of system, Field Resolution (Mission Criteria), Kill box size	Effects Field	Effects Field Delivery
		Total Deployed	Reliability of IRS System, Reliability of system deploying the device	Effects Field	Effects Field Delivery Engage & Strike
Number of Engagements	Integer	Number of effects that can be launched at target	Number of effects available	Blue Undersea System Effects Field	Engage & Strike
Environment Difficulty	Threshold	Cumulative difficulty factor set for each mission	Static: bottom composition, salinity, depth;	Blue Undersea System	All

			Dynamic: sea state, temperature	ISR Systems Effects Systems	
Kill Box Size	Integer	Square nautical mileage of the kill box	Environmental characteristics, Mission criteria from command center	Blue Undersea System ISR Systems Effects Systems	All

Table 24. Attribute Trace Matrix

Factor	Mission 1: IPOE	Mission 2: ISR	Mission 3: Effects Field Delivery	Mission 4: Engage & Strike	Kill Box Deployment	Kill Box Execution
Undersea System Type	X	X	X	X	X	X
Kill Box Type	X	X	X	X	X	X
Target Type				X		X
Kill Box Size	X	X	X	X	X	X
P_US(Detection)				X		X
P_US(Classification)				X		X
P_US(Tracking)				X		X
P_US(Lethality)				X		X
P_US(IPOE)	X				X	X
P_US(ISR Deployment)		X			X	X
P_US(EF Deployment)			X		X	X
P_OPFOR(Detection US)	X	X	X		X	X
P_OPFOR(Classification US)	X	X	X		X	X
P_ISR(Detection)				X		X
P_ISR(Classification)				X		X
P_ISR(Tracking)				X		X
P_OPFOR(Detection ISR)		X	X		X	X
P_OPFOR(Classification ISR)		X	X		X	X
P_EF(Lethality)				X		X
P_OPFOR(Detection EF)			X		X	X
P_OPFOR(Classification EF)			X		X	X
Target Lured				X		X
P_TARGET(Detection US)				X		X

P_TARGET(Detection ISR)				X		X
P_TARGET(Detection EF)				X		X
P_TARGET(Classification US)				X		X
P_TARGET(Classification ISR)				X		X
P_TARGET(Classification EF)				X		X
P_TARGET(Evasion)				X		X
Number of Undersea Systems	X	X	X	X	X	X
Number of IPOE Attempts	X				X	X
Num ISR		X			X	X
Num Effects			X		X	X
Num Required ISR		X	X	X	X	X
Num Required EF			X	X	X	X
Endurance	X	X	X		X	X

APPENDIX D. DESIGN OF EXPERIMENT ANALYSIS

This appendix presents additional mission analysis in support of Chapter V; the team developed no actionable insights based on these figures.

A. MISSION 1: INTELLIGENCE PREPARATION OF THE OPERATIONAL AREA

The scatterplot below compares additional undersea system factors against the four responses from the IPOE mission. The data shown appears to be strictly noise with no obvious trends or correlations.

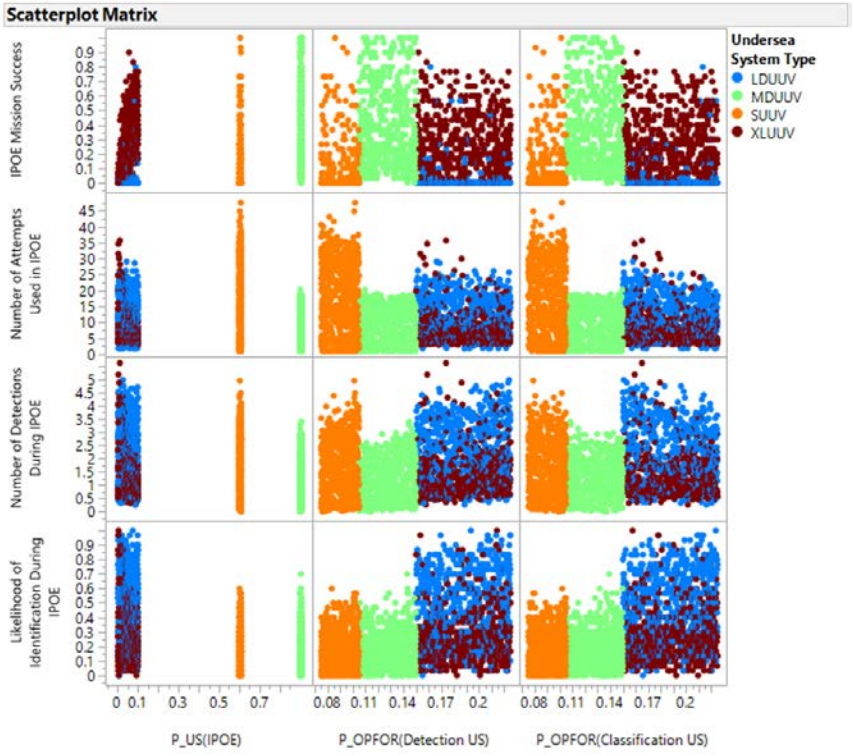


Figure 67. Additional IPOE Mission Scatterplots

The next two figures below show additional system factors against the main response variable, IPOE mission success, for the two kill box groupings. This corroborates the fact that more MDUUV's are required in order to complete the mission.

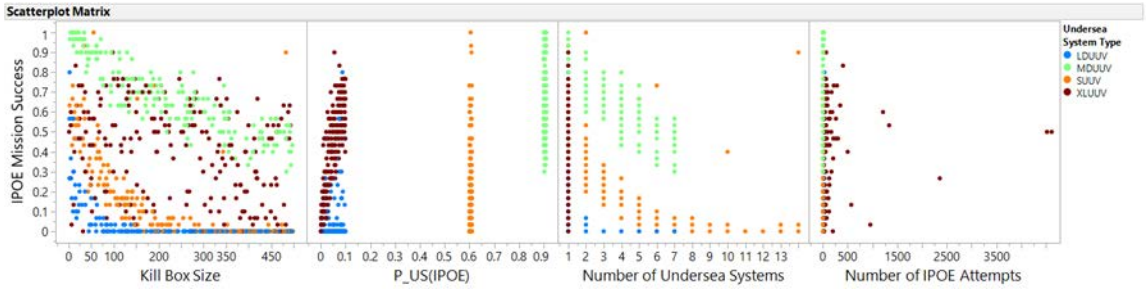


Figure 68. Additional Scatterplots for IPOE Mission Success for Small Kill Box Ranges

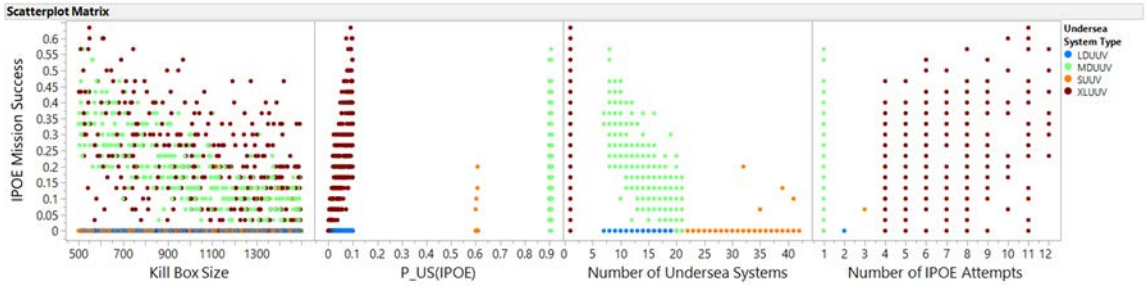


Figure 69. Additional Scatterplots for IPOE Mission Success for Medium/Large Kill Box Ranges

The next two figures below show additional system factors against the likelihood of identification for the two kill box groupings. The majority of these scatterplots show mainly noise with no obvious trends.

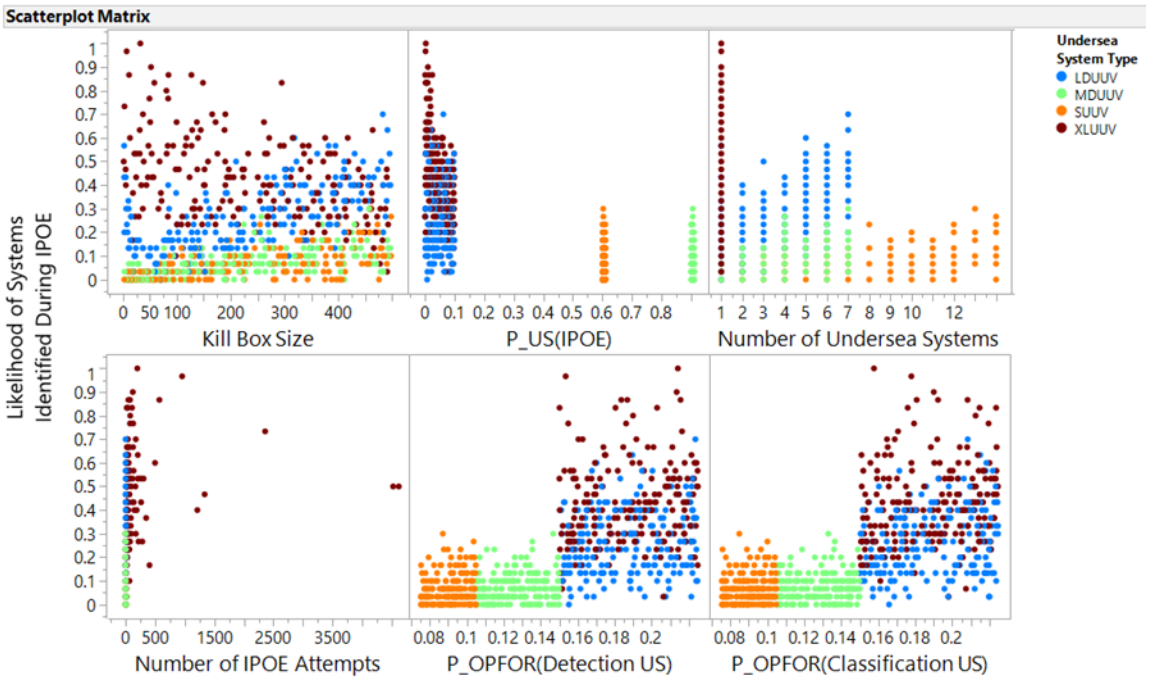


Figure 70. Additional Scatterplots for Identification Likelihood during IPOE for Small Kill Box Ranges

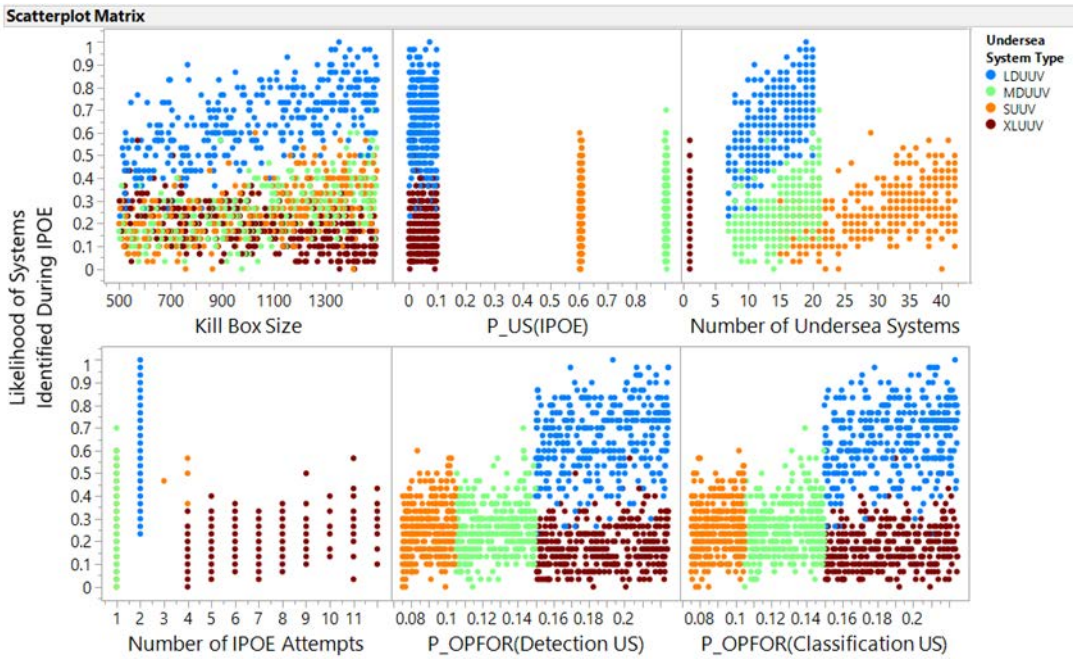


Figure 71. Additional Scatterplots for Identification Likelihood during IPOE for Medium/Large Kill Box Ranges

Looking at a similar binning for number of IPOE attempts used, shown in the next two figures below, the two trends identified are within kill box size and number of undersea systems used. The three middle scatterplots in both graphics appear to be strictly randomized noise.

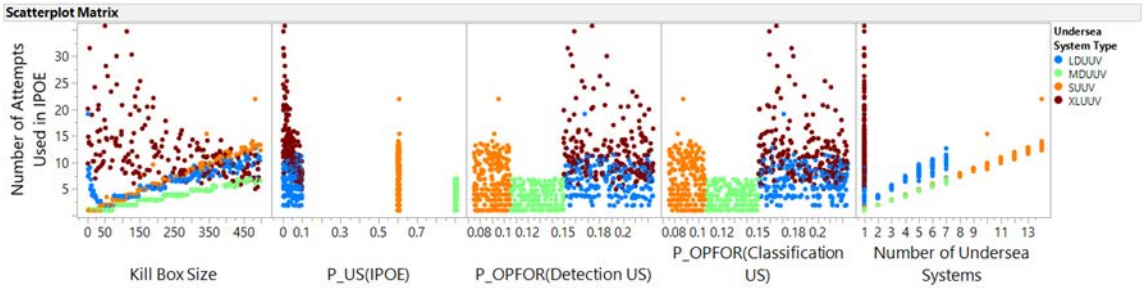


Figure 72. Additional Scatterplots for IPOE Attempts Used for Small Kill Box Ranges

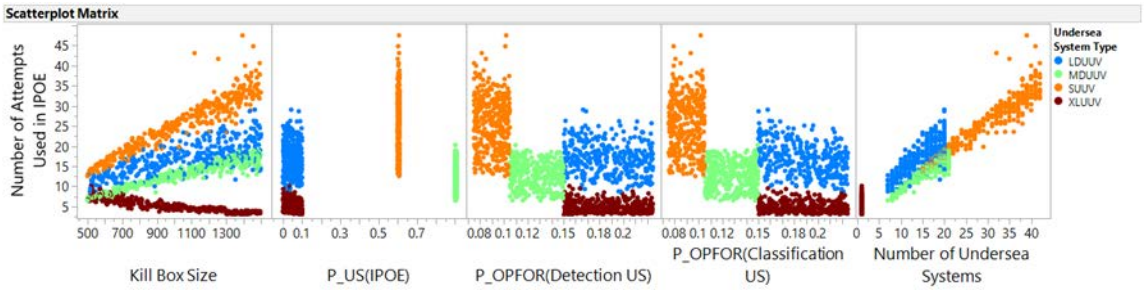


Figure 73. Additional Scatterplots for IPOE Attempts Used for Medium/Large Kill Box Ranges

The final response, number of detections during IPOE shown in the next two figures, continues to repeat the same trends discussed for the previous responses.

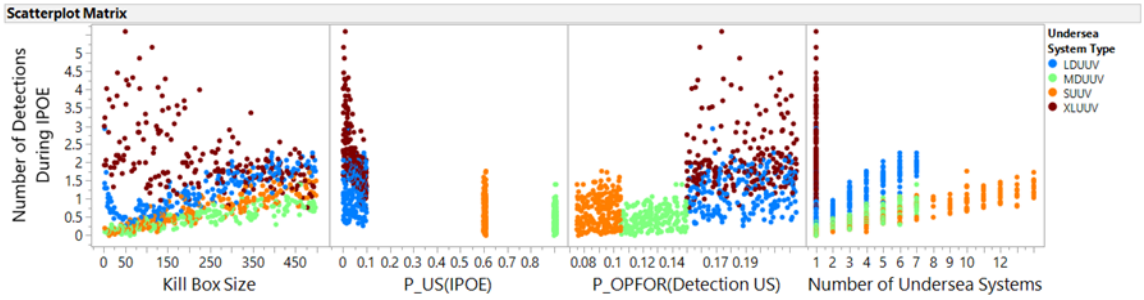


Figure 74. Additional Scatterplots for Number of Detections during IPOE for Small Kill Box Ranges

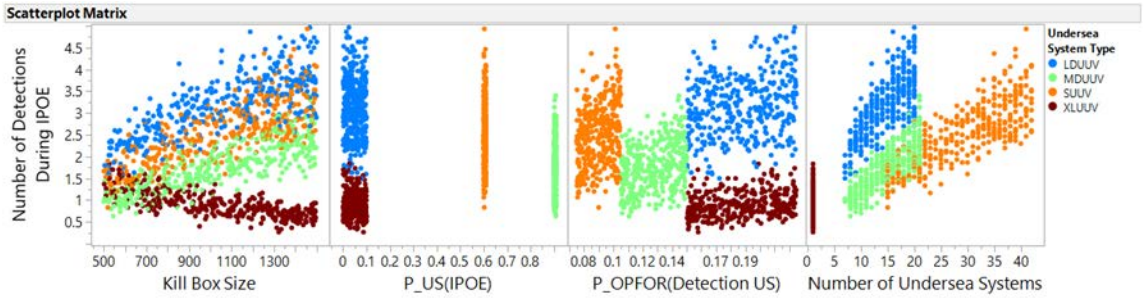


Figure 75. Additional Scatterplots for Number of Detections during IPOE for Medium/Large Kill Box Ranges

B. MISSION 2: INTELLIGENCE, SURVEILLANCE, AND RECONNAISSANCE

The scatterplot below compares additional undersea system factors against the four responses from the ISR deployment mission. The data shown appears to be strictly noise with no obvious trends or correlations.

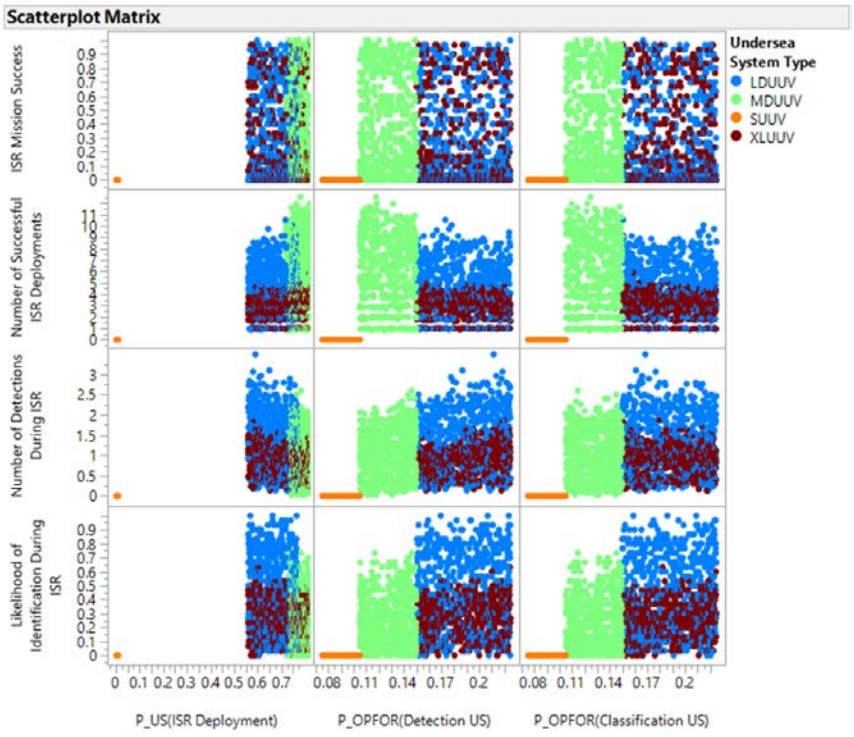


Figure 76. Additional ISR Mission Scatterplots

The response of successful ISR deployments in the figure below is predicated on the probability of successful ISR deployment. Analyzing this data directly indicated a linear trend that this data does in fact play a role in successful deployments. This is an expected trend in the results as the factor and the response go hand in hand. What is notable in this data is that the smaller kill boxes were less affected by P_{US}(ISR Deployment) than the medium and large kill boxes.

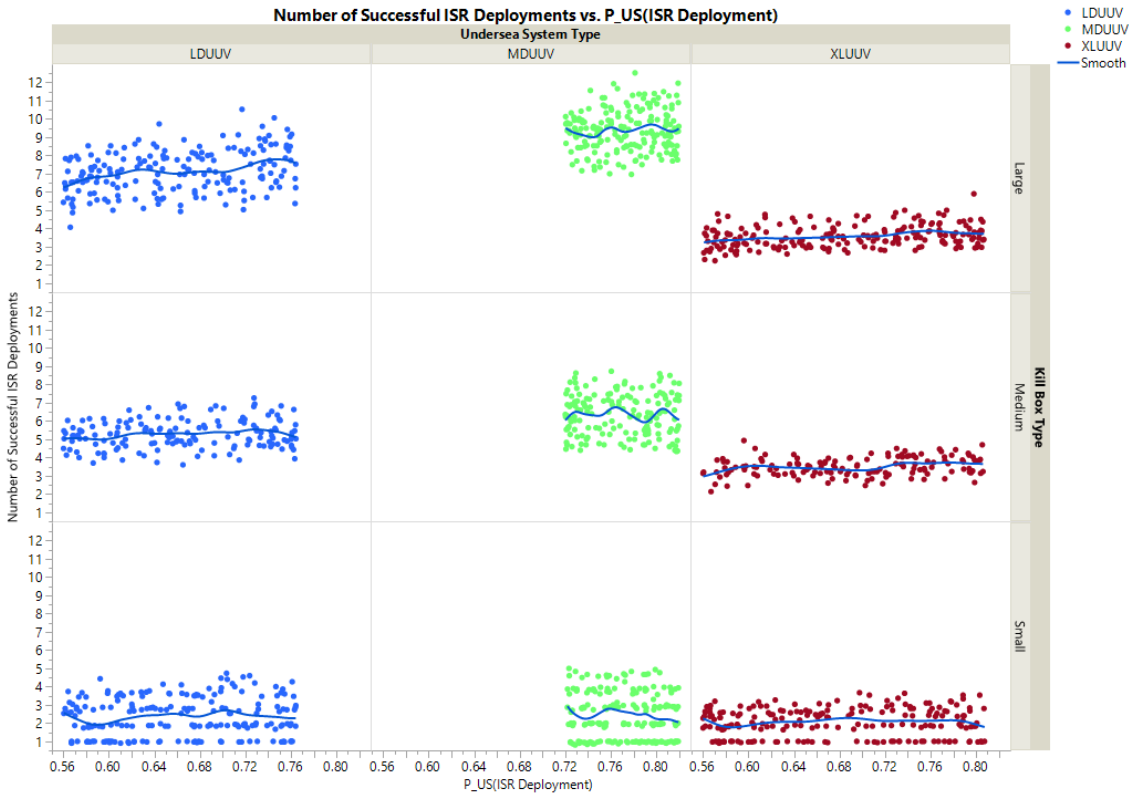


Figure 77. Number of Successful ISR Deployments Binned ISR Deployment Probability

The factor of ISR deployment probability in Figure 78 has a linear depression with the probability of ISR deployment. It is important to note that linear depression in the case of this response is a beneficial result of the increase in P_{US}(ISR Deployment). With the increase in this factor, there are less detections of undersea systems. This is directly because the more successful the system is at deploying ISR systems, the less number of times the undersea systems need to reattempt a deployment.

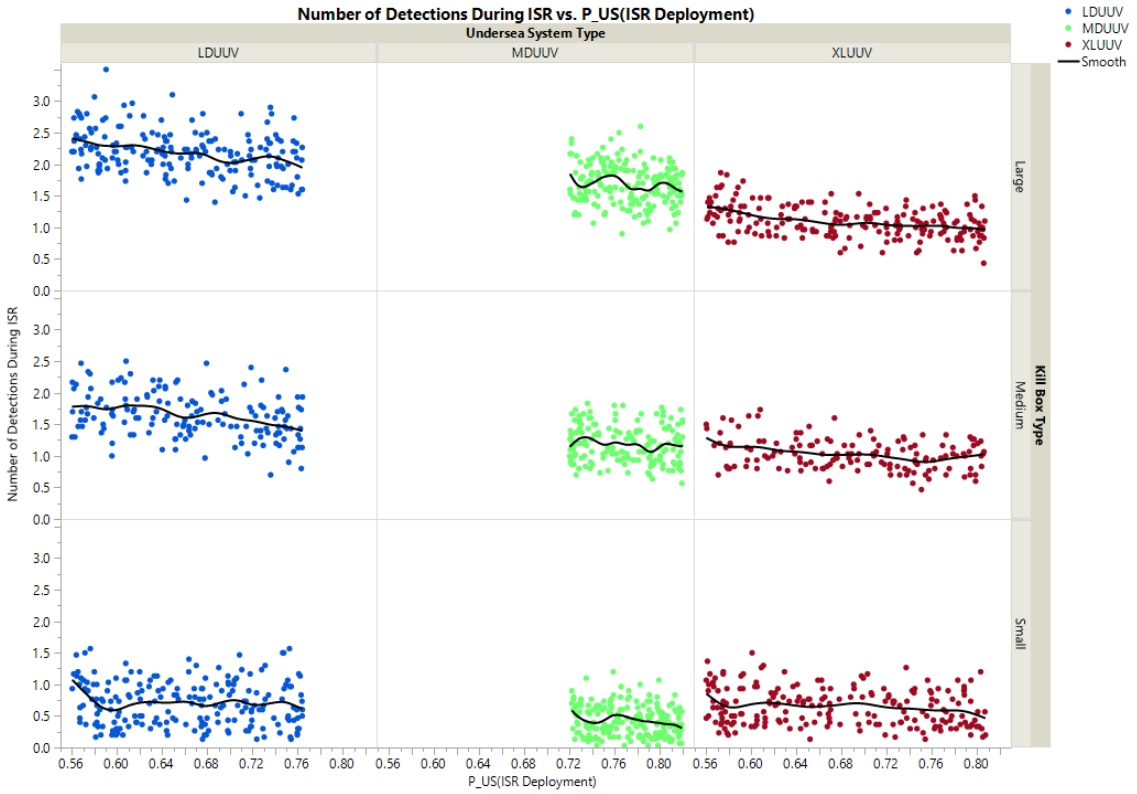


Figure 78. Number of Detections During ISR Binned by ISR Deployment Probability

There is a linear trend in the increase in opposing force's capabilities to the reduction of the number of successful ISR deployments, though only for the LDUUV and XLUUV, as the MDUUV does not show much correlation. While it is understood that the chance of a threat being present to detect and classify an unmanned system in the area, there is a significant likelihood that they will be able to prevent the successful deployments necessary for the execution of ISR deployment and that of the effects field in the next mission.

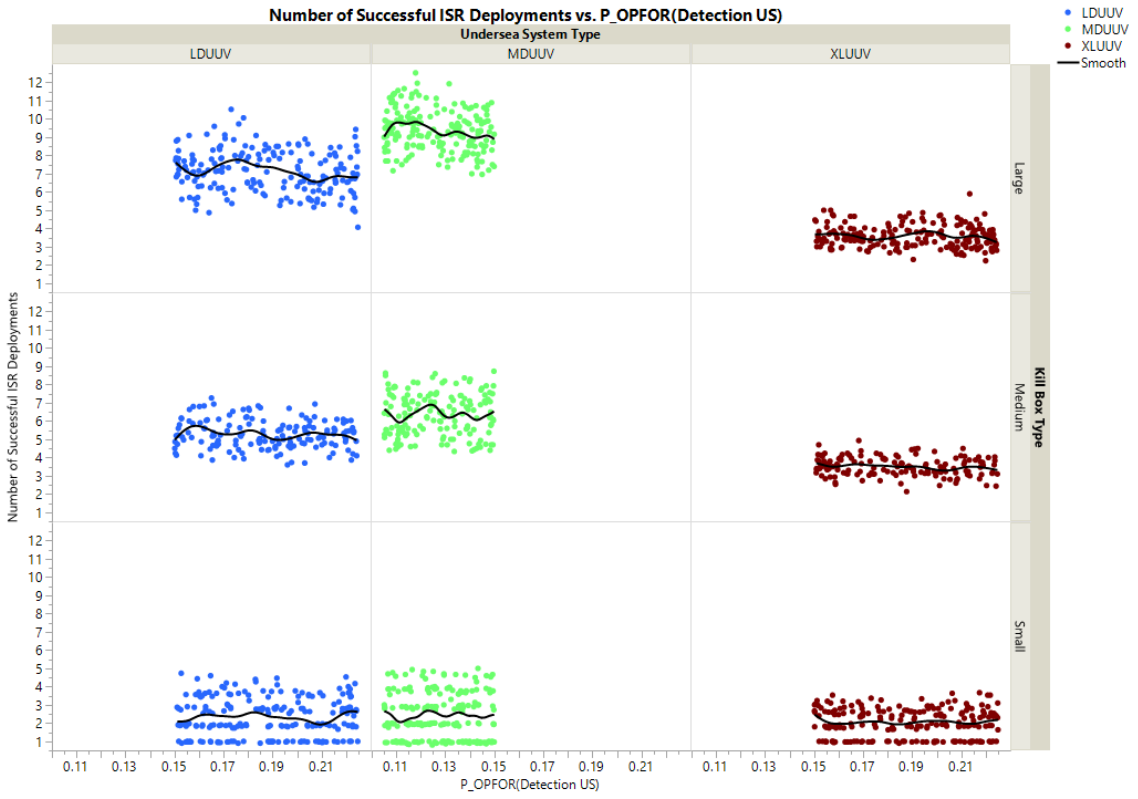


Figure 79. Number of Successful ISR Deployments Binned by OPFOR Detecting U.S. Probability

Identified systems during the ISR mission have a linear trend against the higher probability of ISR deployment probabilities. With each larger kill box the average number of systems identified increases significantly; with LDUUV being the worst offender followed by XLUUV and lastly MDUUV. The opposing forces are least likely to identify MDUUV during the mission as ISR deployment probabilities increased, but as the larger kill boxes were employed in the simulation it was ultimately the XLUUV that was capable of remaining unidentified the most while it followed the MDUUVs trend closely overtaking it in the end.

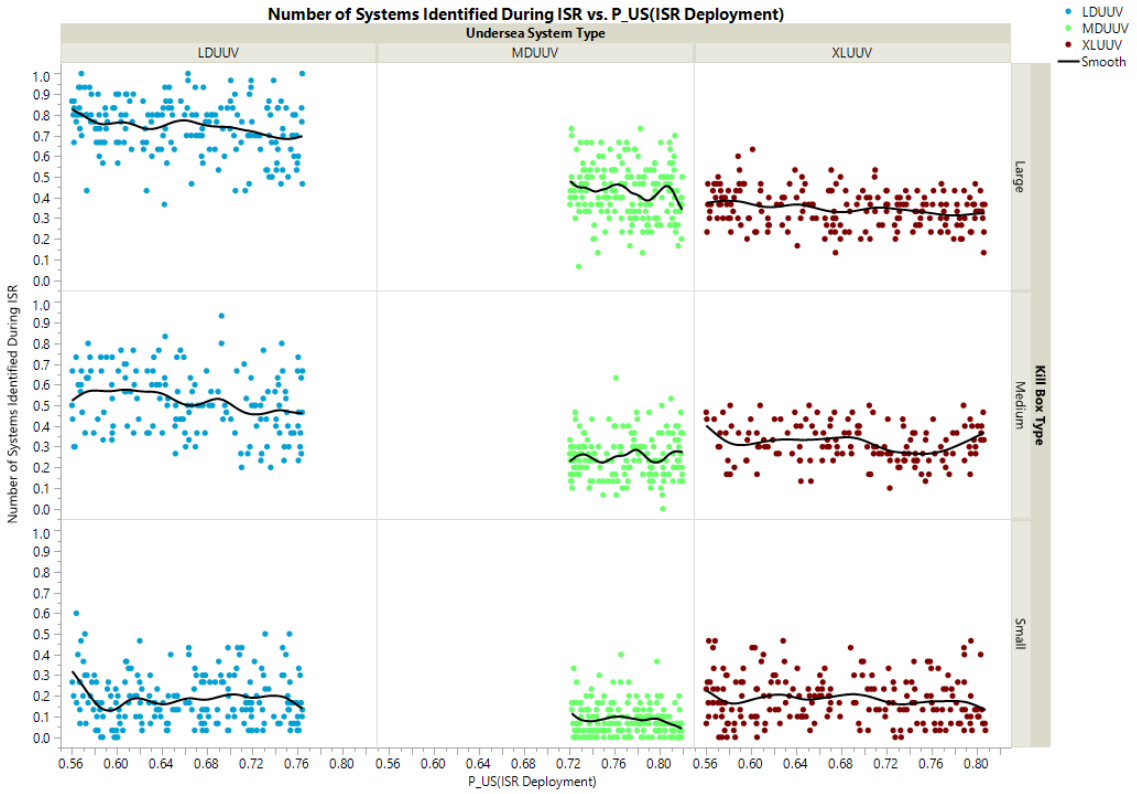


Figure 80. Likelihood of Identification During ISR Binned by ISR Deployment Probability

For the likelihood of identification during ISR shown in Figure 81, a big factor in its response is the number of required ISR to complete that mission. The need for more ISR systems dictates the simulation to run checks on detection more times. This results in higher averages of systems detected, detections for all systems climb linearly significantly fast indicating that the less number of ISR systems needed to perform the mission the better for the success of the mission. As for the undersea system capability of performing this mission the LDUUV was the most effected by the number of required ISR to cover the kill box. The MDUUV was most capable in performing small and medium sized kill boxes but when it came to the large kill boxes, the XLUUV was most successful at remaining undetected.

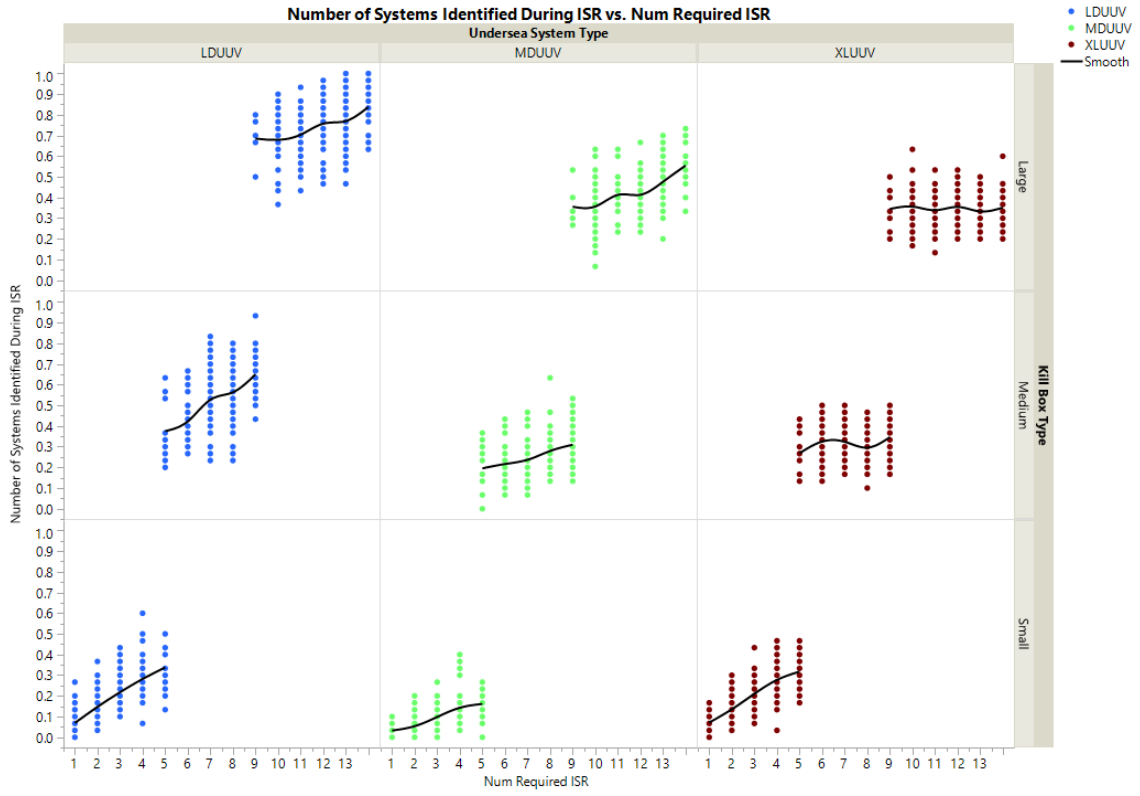


Figure 81. Likelihood of Identification During ISR Binned by ISR Field Resolution Device Requirement

With the small variations of the opposing force's capability to detect the undersea systems, it still had a significant impact on the ability for the systems to perform their mission. The LDUUV is the most identified system according to Figure 82, followed by the MDUUV and the XLUUV. Similar to results in the previous factors the MDUUV was capable of remaining unidentified during the small and medium kill boxes, but was overtaken by the LXUUV when it came to large kill box sizes.

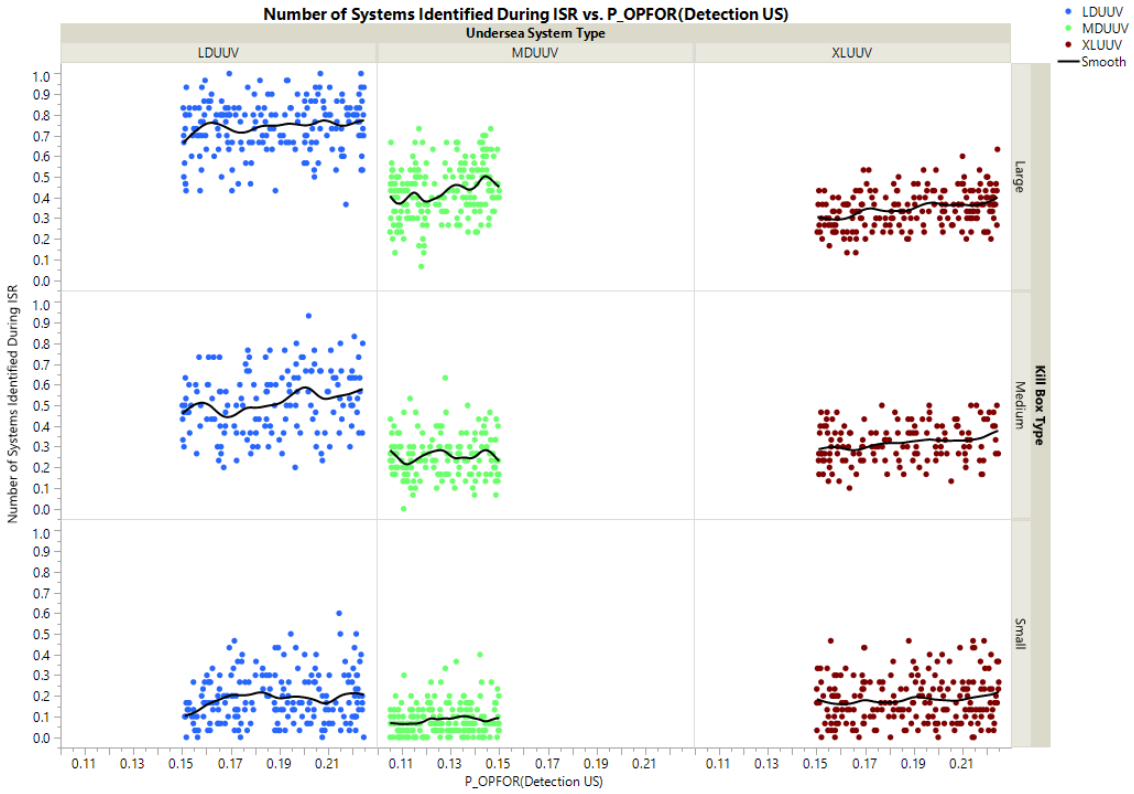


Figure 82. Likelihood of Identification During ISR Binned by OPFOR Detecting U.S. Probability

C. MISSION 3: EFFECTS FIELD DEPLOYMENT

The scatterplot below compares additional undersea system factors against the four responses from the effects field deployment mission. The data shown appears to be strictly noise with no obvious trends or correlations.

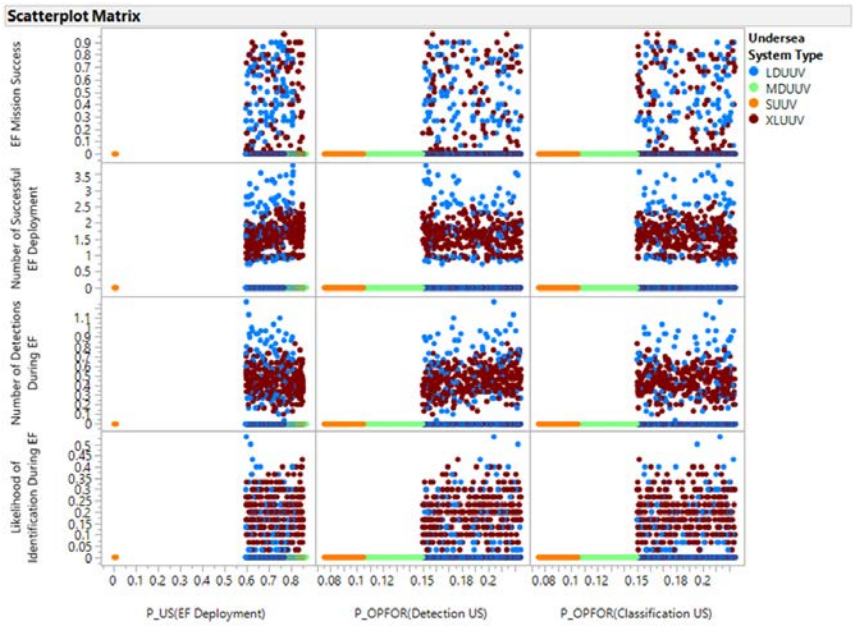


Figure 83. Additional EF Mission Scatterplots

D. MISSION 4: ENGAGE AND STRIKE

The scatterplot below compares additional undersea system factors against the three responses from the final mission. The data shown appears to be strictly noise with no obvious trends or correlations.

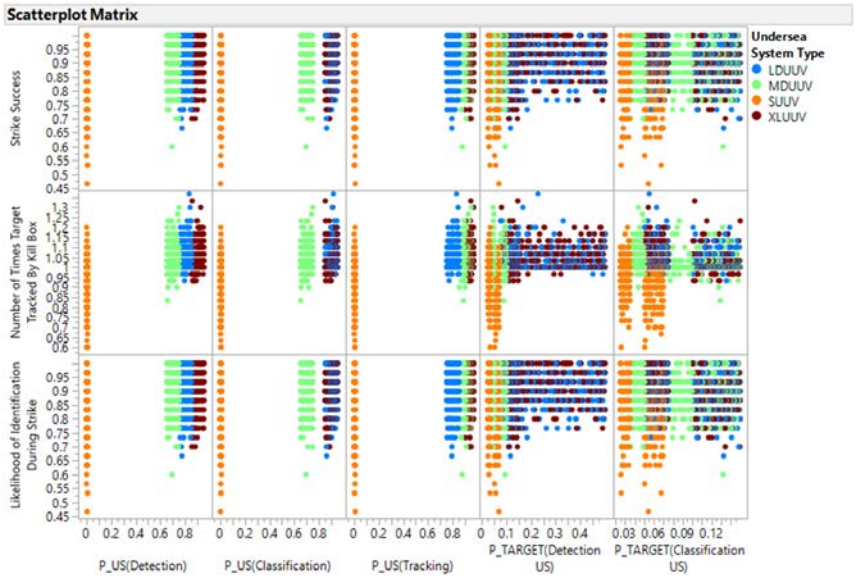


Figure 84. Additional Engage and Strike Mission Scatterplots

APPENDIX E. SIMULATION SOURCE CODE

This appendix presents the source code of the seabed warfare operational simulation; developed out of MATLAB 2018a. The following sections decompose the simulation into its various components.

A. GRAPHICAL USER INTERFACE AND MAIN FUNCTION

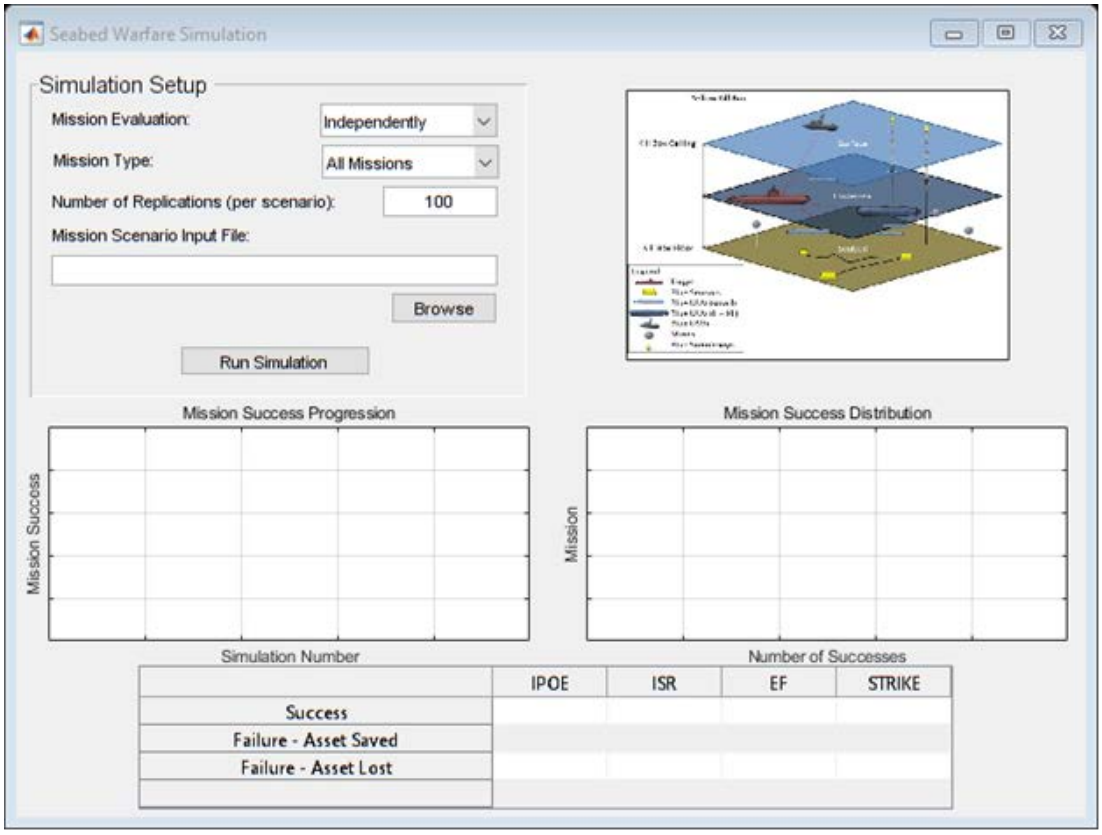


Figure 85. Seabed Warfare Simulation GUI

```
function varargout = PerformSeabedWarfare(varargin)
%%-------------------------------------------------------------------------
% function varargout = PerformSeabedWarfare(varargin)
%   Initialization function of the Seabed Warfare Simulation Graphical User
%   Interface (GUI).
```

```
%%-------------------------------------------------------------------------

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
    'gui_Singleton',  gui_Singleton, ...
    'gui_OpeningFcn', @SimulateKillbox_OpeningFcn, ...
    'gui_OutputFcn',  @SimulateKillbox_OutputFcn, ...
    'gui_LayoutFcn',  @PerformSeabedWarfare_LayoutFcn, ...
    'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State,varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
end

%-------------------------------------------------------------------------
function SimulateKillbox_OpeningFcn(hObject,eventdata,handles,varargin)
%%-------------------------------------------------------------------------
% function SimulateKillbox_OpeningFcn(hObject,eventdata,handles,varargin)
%   This function executes prior to the interface becoming visible. Allows
%   additional modifications to the interface without having to go through
%   updating the figure in GUIDE.
%%-------------------------------------------------------------------------

%Choose default command line output for SimulateKillbox
handles.output = hObject;

%Update the graphics to include titles, labels, and grids.
axes(handles.SimulationOverview);
% hold(handles.SimulationOverview,'on');
title('Mission Success Progression','FontWeight','normal','FontSize',8);
xlabel('Simulation Number','FontWeight','normal','FontSize',8);
yLab = ylabel('Mission Success','FontWeight','normal','FontSize',8);
grid on;

axes(handles.SimulationHist);
title('Mission Success Distribution','FontWeight','normal','FontSize',8);
xlabel('Number of Successes','FontWeight','normal','FontSize',8);
ylabel('Mission','FontWeight','normal','FontSize',8);
grid on;

kbFile = which('Kill Box Graphic.jpg');
img = imread(kbFile);
imgHan = image(handles.killboxGraphics,img);
set(handles.killboxGraphics,'XTick',[]);
set(handles.killboxGraphics,'YTick',[]);
```

```
%Update handles structure.
guidata(hObject,handles);
end

%--------------------------------------------------------------------------
function varargout = SimulateKillbox_OutputFcn(hObject,eventdata,handles)
%%------------------------------------------------------------------------
% function varargout = SimulateKillbox_OutputFcn(hObject,eventdata,handles)
%   Returns the requested outputs from the interface.
%%------------------------------------------------------------------------

%Get default command line output from handles structure
varargout{1} = handles.output;
end

%--------------------------------------------------------------------------
function RunSimulation(hObject,eventdata,handles)
%%------------------------------------------------------------------------
% function RunSimulation
%   Main function of the Seabed Warfare Simulation. Takes the simulation
%   setup provided by the user and conducts the probabilistic simulation.
%%------------------------------------------------------------------------

global numSimulations numMissions simCount misNum

%Determine the number of runs to be made through the simulation.
numSimulations = get(handles.NumSimulation,'String');
numSimulations = str2double(numSimulations);
if isnan(numSimulations)
    msgbox('Invalid Number Entry...Aborting Kill Box Simulation',...
        'Unable to Process','error');
    return;
end

%Grab the scenario file.
scenarioFile = get(handles.ScenarioFile,'String');
if strcmpi(scenarioFile,'') || isempty(scenarioFile)
    msgbox('No Mission Scenario File Selected...Aborting Kill Box Simulation',...
        'Unable to Process','error');
    return;
end

%Pull the mission evalation type for processing.
evalVal = get(handles.MissionEvaluationType,'Value');
missionEvalTypes = get(handles.MissionEvaluationType,'String');

runSuccessive = 0;
SIMPARAMS = struct('IPOE',1,'ISR',1,'EF',1,'ENG',1);
if strcmpi(missionEvalTypes{evalVal},'Successively')
    runSuccessive = 1;
else
    %Pull the mission evalation type for processing.
```

```
    misVal = get(handles.MissionType,'Value');
    misTypes = get(handles.MissionType,'String');
    if ~strcmpi(misTypes{misVal},'All Missions')
        simFields = fieldnames(SIMPARAMS);
        for fld = 1:length(simFields)
            if fld ~= misVal-1
                SIMPARAMS.(simFields{fld}) = 0;
            end
        end
    end
end
%Find the number of missions.
try
    misIndex = xlsread(scenarioFile,'A:A');
    numMissions = length(misIndex);
catch
    msgbox('Invalid Mission Scenario File Selected...Aborting Kill Box Simulation',...
        'Unable to Process','error');
    return;
end
%Initialize the mission parameters
try
    InitializeSimulation(scenarioFile);
catch
    msgbox('Invalid Mission Scenario File Selected...Aborting Kill Box Simulation',...
        'Unable to Process','error');
    return;
end

%Intialize the waitbar
waitHan = waitbar(0,'Processing Missions');

for misNum = 1:numMissions
    for simCount = 1:numSimulations
        if ishandle(waitHan)
            waitbar((misNum*simCount)/(numMissions*numSimulations),waitHan,...
                sprintf('Processing Scenario %i (%i/%i)
...',misNum,simCount,numSimulations));
        else
            msgbox('Processing Cancelled...Aborting Kill Box Simulation',...
                'User Cancelled','error');
            clear all
            return;
        end

        %Pull the mission parameters that will be used.
        MISPARAMS = InitializeSimulation;

        %Initialize the mission results structure.
        MISRESULTS(simCount) = InitializeMOPs;

        %Iterate through the missions
```

```
        while true
            % MISSION 1: Conduct IPOE
            %Attempt the mission.
            if SIMPARAMS.IPOE
                MISRESULTS(simCount).IPOE = ConductIPOE(MISPARAMS);
            end

            %If running successively, if unsuccessful, abort the remainder
            %of the missions.
            if ~MISRESULTS(simCount).IPOE.missionSuccess && runSuccessive
                break
            end

            % MISSION 2: Deploy ISR
            %Attempt the mission.
            if SIMPARAMS.ISR
                MISRESULTS(simCount).ISR = DeployISR(MISPARAMS);
            end

            %If running successively, if unsuccessful, abort the remainder
            %of the missions.
            if ~MISRESULTS(simCount).ISR.missionSuccess && runSuccessive
                break
            end

            % MISSION 3: Deploy EF
            %Update the mission parameters with the total number of deployed
            %ISR systems.
            if runSuccessive
                MISPARAMS.ISRSystems.totalDeployed =
MISRESULTS(simCount).ISRSystems.totalDeployed;
            else
                MISPARAMS.ISRSystems.totalDeployed = MISPARAMS.ISRSystems.numRequired;
            end

            %Attempt mission.
            if SIMPARAMS.EF
                MISRESULTS(simCount).EF = DeployEF(MISPARAMS);
            end

            %If running successively, if unsuccessful, abort the remainder
            %of the missions.
            if ~MISRESULTS(simCount).EF.missionSuccess && runSuccessive
                break
            end

            %Set success of killbox deployment to TRUE
            MISRESULTS(simCount).KillBoxDeployed = 1;

            % MISSION 4: Engage and Strike
            %Update the mission parameters with the total number of deployed
            %ISR systems.
```

```
            if runSuccessive
                MISPARAMS.EFSystems.totalDeployed =
MISRESULTS(simCount).EFSystems.totalDeployed;
            else
                MISPARAMS.EFSystems.totalDeployed = MISPARAMS.EFSystems.numRequired;
            end

            %Attempt the mission
            if SIMPARAMS.ENG
                MISRESULTS(simCount).ENG = EngageAndStrike(MISPARAMS);
            end

            if ~MISRESULTS(simCount).ENG.missionSuccess
                break
            end

            %Set the success of the killbox execution to TRUE.
            MISRESULTS(simCount).KillBoxExecution = 1;
            break
        end

        %Update the overview graphics on the interface
        UpdateOverview(handles,MISRESULTS(simCount));

        %Refresh the user interface.
        pause(0.001);
    end

    %Update the results CSV
    WriteSimResults(scenarioFile,MISRESULTS,MISPARAMS);
end

%Remove all global and persistent variables.
close(waitHan);
clear all
end

%--------------------------------------------------------------------------
function MISSIONRESULTS = InitializeMOPs
%%------------------------------------------------------------------------
% function MISSIONRESULTS = InitializeMOPs
%   This function intializes the return structure of mission results.
%%------------------------------------------------------------------------

MISSIONRESULTS.IPOE = struct('missionSuccess',false,...
    'numIdentified',0,...
    'NumDetections',0,...
    'NumAttempts',0,...
    'numEvaded',0);

MISSIONRESULTS.ISR = struct('missionSuccess',0,...
    'numIdentified',0,...
```

```
    'NumDetections',0,...
    'AssetsDeployed',0,...
    'numEvaded',0,...
    'totalDeployed',0,...
    'fieldDetected',0);

MISSIONRESULTS.EF = struct('missionSuccess',0,...
    'numIdentified',0,...
    'NumDetections',0,...
    'AssetsDeployed',0,...
    'numEvaded',0,...
    'totalDeployed',0,...
    'fieldDetected',0,...
    'isrDetected',0);

MISSIONRESULTS.ENG = struct(...
    'missionSuccess',0,...
    'numEffectsFired',0,...
    'numTgtTracked',0,...
    'numIdentified',0,...
    'isrDetected',0,...
    'efDetected',0,...
    'usEvaded',0 ...
    );

MISSIONRESULTS.KillBoxDeployed = 0;
MISSIONRESULTS.KillBoxExecution = 0;
end

%--------------------------------------------------------------------------
function UploadScenarioFile(hObject, eventdata, handles)
%%-------------------------------------------------------------------------
% function UploadScenarioFile(hObject, eventdata, handles)
%   Allows user to browse for the scenario file.
%%-------------------------------------------------------------------------

%Prompt the user for the file
[scenFile, scenPath] = uigetfile('.xlsx','Select Mission Scenario Input Excel');

if isequal(scenFile,0) || isequal(scenPath,0)
    return
end

set(handles.ScenarioFile,'String',[scenPath scenFile]);
end

%--------------------------------------------------------------------------
function MissionEvaluationType_Callback(hObject, eventdata, handles)

typeVal = get(hObject,'Value');
typeStr = get(hObject,'String');
if strcmpi(typeStr{typeVal},'Successively')
```

```
    set(handles.MissionType,'Enable','off');
else
    set(handles.MissionType,'Enable','on');
end
end


% --- Creates and returns a handle to the GUI figure.
function h1 = PerformSeabedWarfare_LayoutFcn(policy)
% policy - create a new figure or use a singleton. 'new' or 'reuse'.

persistent hsingleton;
if strcmpi(policy, 'reuse') & ishandle(hsingleton)
    h1 = hsingleton;
    return;
end

appdata = [];
appdata.GUIDEOptions = struct(...
    'active_h', [], ...
    'taginfo', struct(...
    'figure', 2, ...
    'text', 6, ...
    'uipanel', 4, ...
    'axes', 4, ...
    'uitable', 2, ...
    'pushbutton', 3, ...
    'edit', 5, ...
    'uibuttongroup', 2, ...
    'radiobutton', 3, ...
    'popupmenu', 2), ...
    'override', 0, ...
    'release', [], ...
    'resize', 'none', ...
    'accessibility', 'callback', ...
    'mfile', 1, ...
    'callbacks', 1, ...
    'singleton', 1, ...
    'syscolorfig', 1, ...
    'blocking', 0);
appdata.lastValidTag = 'figure1';
appdata.GUIDELayoutEditor = [];
appdata.initTags = struct(...
    'handle', [], ...
    'tag', 'figure1');

h1 = figure(...
'Units',get(0,'defaultfigureUnits'),...
'Position',[100 100 700 500],...
'Visible',get(0,'defaultfigureVisible'),...
'Color',get(0,'defaultfigureColor'),...
'CurrentAxesMode','manual',...
```

```
'IntegerHandle','off',...
'MenuBar','none',...
'Name','Seabed Warfare Simulation',...
'NumberTitle','off',...
'Tag','figure1',...
'Resize','off',...
'PaperPosition',get(0,'defaultfigurePaperPosition'),...
'ScreenPixelsPerInchMode','manual',...
'HandleVisibility','callback',...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );

appdata = [];
appdata.lastValidTag = 'SimulationOverview';

h2 = axes(...
'Parent',h1,...
'FontUnits',get(0,'defaultaxesFontUnits'),...
'Units','pixels',...
'View',get(0,'defaultaxesView'),...
'CameraPosition',[0.5 0.5 9.16025403784439],...
'CameraPositionMode',get(0,'defaultaxesCameraPositionMode'),...
'CameraTarget',[0.5 0.5 0.5],...
'CameraTargetMode',get(0,'defaultaxesCameraTargetMode'),...
'CameraViewAngle',6.60861036031192,...
'CameraViewAngleMode',get(0,'defaultaxesCameraViewAngleMode'),...
'Projection',get(0,'defaultaxesProjection'),...
'LabelFontSizeMultiplier',get(0,'defaultaxesLabelFontSizeMultiplier'),...
'AmbientLightColor',get(0,'defaultaxesAmbientLightColor'),...
'PlotBoxAspectRatio',[1 0.444444444444444 0.444444444444444],...
'PlotBoxAspectRatioMode',get(0,'defaultaxesPlotBoxAspectRatioMode'),...
'FontName',get(0,'defaultaxesFontName'),...
'FontAngle',get(0,'defaultaxesFontAngle'),...
'FontWeight',get(0,'defaultaxesFontWeight'),...
'FontSmoothing',get(0,'defaultaxesFontSmoothing'),...
'TickLabelInterpreter',get(0,'defaultaxesTickLabelInterpreter'),...
'XDir',get(0,'defaultaxesXDir'),...
'YDir',get(0,'defaultaxesYDir'),...
'ZDir',get(0,'defaultaxesZDir'),...
'ColormapMode',get(0,'defaultaxesColormapMode'),...
'AlphamapMode',get(0,'defaultaxesAlphamapMode'),...
'Layer',get(0,'defaultaxesLayer'),...
'TickLength',get(0,'defaultaxesTickLength'),...
'GridLineStyle',get(0,'defaultaxesGridLineStyle'),...
'MinorGridLineStyle',get(0,'defaultaxesMinorGridLineStyle'),...
'XAxisLocation',get(0,'defaultaxesXAxisLocation'),...
'XTick',[0 0.2 0.4 0.6 0.8 1],...
'XTickMode',get(0,'defaultaxesXTickMode'),...
'XTickLabelRotation',get(0,'defaultaxesXTickLabelRotation'),...
'XScale',get(0,'defaultaxesXScale'),...
'XTickLabel',blanks(0),...
'XMinorTick',get(0,'defaultaxesXMinorTick'),...
'YAxisLocation',get(0,'defaultaxesYAxisLocation'),...
```

```
'YTick',[0 0.2 0.4 0.6 0.8 1],...
'YTickMode',get(0,'defaultaxesYTickMode'),...
'YTickLabelRotation',get(0,'defaultaxesYTickLabelRotation'),...
'YScale',get(0,'defaultaxesYScale'),...
'YTickLabel',blanks(0),...
'YMinorTick',get(0,'defaultaxesYMinorTick'),...
'ZTickLabelRotation',get(0,'defaultaxesZTickLabelRotation'),...
'ZScale',get(0,'defaultaxesZScale'),...
'ZMinorTick',get(0,'defaultaxesZMinorTick'),...
'BoxStyle',get(0,'defaultaxesBoxStyle'),...
'LineWidth',get(0,'defaultaxesLineWidth'),...
'Color',get(0,'defaultaxesColor'),...
'ClippingStyle',get(0,'defaultaxesClippingStyle'),...
'CameraMode',get(0,'defaultaxesCameraMode'),...
'DataSpaceMode',get(0,'defaultaxesDataSpaceMode'),...
'ColorSpaceMode',get(0,'defaultaxesColorSpaceMode'),...
'DecorationContainerMode',get(0,'defaultaxesDecorationContainerMode'),...
'ChildContainerMode',get(0,'defaultaxesChildContainerMode'),...
'BoxFrameMode',get(0,'defaultaxesBoxFrameMode'),...
'XAxisMode','manual',...
'YRulerMode',get(0,'defaultaxesYRulerMode'),...
'ZAxisMode','manual',...
'AmbientLightSourceMode',get(0,'defaultaxesAmbientLightSourceMode'),...
'XGrid',get(0,'defaultaxesXGrid'),...
'XMinorGrid',get(0,'defaultaxesXMinorGrid'),...
'YGrid',get(0,'defaultaxesYGrid'),...
'YMinorGrid',get(0,'defaultaxesYMinorGrid'),...
'ZGrid',get(0,'defaultaxesZGrid'),...
'ZMinorGrid',get(0,'defaultaxesZMinorGrid'),...
'ButtonDownFcn',blanks(0),...
'Interruptible','off',...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} ,...
'DeleteFcn',blanks(0),...
'Tag','SimulationOverview',...
'UserData',[],...
'HitTest','off',...
'PickableParts',get(0,'defaultaxesPickableParts'),...
'Position',[22 116 315 140],...
'ActivePositionProperty','position',...
'LooseInset',[92.82 60.61 67.83 41.325],...
'LooseInsetMode',get(0,'defaultaxesLooseInsetMode'),...
'ColorOrderIndex',get(0,'defaultaxesColorOrderIndex'),...
'LineStyleOrder',get(0,'defaultaxesLineStyleOrder'),...
'LineStyleOrderIndex',get(0,'defaultaxesLineStyleOrderIndex'),...
'FontSize',8.5,...
'FontSizeMode',get(0,'defaultaxesFontSizeMode'),...
'TitleFontWeight',get(0,'defaultaxesTitleFontWeight'),...
'TitleFontSizeMultiplier',get(0,'defaultaxesTitleFontSizeMultiplier'),...
'SortMethod','childorder',...
'Clipping',get(0,'defaultaxesClipping'),...
'NextPlot','replacechildren',...
'Box','on',...
```

```
'ChildrenMode','manual',...
'Visible',get(0,'defaultaxesVisible'),...
'HandleVisibility',get(0,'defaultaxesHandleVisibility'));

h3 = get(h2,'title');

set(h3,...
'Parent',h2,...
'Units','data',...
'FontUnits','points',...
'DecorationContainer',[],...
'DecorationContainerMode','auto',...
'Color',[0 0 0],...
'ColorMode','auto',...
'Position',[0.500000543442984 1.01669642857143 0.500000000000007],...
'PositionMode','auto',...
'Interpreter','tex',...
'InterpreterMode','auto',...
'Rotation',0,...
'RotationMode','auto',...
'FontName','Helvetica',...
'FontNameMode','auto',...
'FontSize',9.35,...
'FontSizeMode','auto',...
'FontAngle','normal',...
'FontAngleMode','auto',...
'FontWeight','bold',...
'FontWeightMode','auto',...
'HorizontalAlignment','center',...
'HorizontalAlignmentMode','auto',...
'VerticalAlignment','bottom',...
'VerticalAlignmentMode','auto',...
'EdgeColor','none',...
'EdgeColorMode','auto',...
'LineStyle','-',...
'LineStyleMode','auto',...
'LineWidth',0.5,...
'LineWidthMode','auto',...
'BackgroundColor','none',...
'BackgroundColorMode','auto',...
'Margin',3,...
'MarginMode','auto',...
'Clipping','off',...
'ClippingMode','auto',...
'Layer','middle',...
'LayerMode','auto',...
'FontSmoothing','on',...
'FontSmoothingMode','auto',...
'IncludeRenderer','on',...
'IsContainer','off',...
'IsContainerMode','auto',...
'HG1EraseMode','auto',...
```

```
'BusyAction','queue',...
'Interruptible','on',...
'HitTest','on',...
'HitTestMode','auto',...
'PickableParts','visible',...
'PickablePartsMode','auto',...
'DimensionNames',{  'X' 'Y' 'Z' },...
'DimensionNamesMode','auto',...
'XLimInclude','on',...
'XLimIncludeMode','auto',...
'YLimInclude','on',...
'YLimIncludeMode','auto',...
'ZLimInclude','on',...
'ZLimIncludeMode','auto',...
'CLimInclude','on',...
'CLimIncludeMode','auto',...
'ALimInclude','on',...
'ALimIncludeMode','auto',...
'Description','Axes Title',...
'DescriptionMode','auto',...
'Visible','on',...
'VisibleMode','auto',...
'Serializable','on',...
'SerializableMode','auto',...
'HandleVisibility','off',...
'HandleVisibilityMode','auto',...
'TransformForPrintFcnImplicitInvoke','on',...
'TransformForPrintFcnImplicitInvokeMode','auto');

h4 = get(h2,'xlabel');

set(h4,...
'Parent',h2,...
'Units','data',...
'FontUnits','points',...
'DecorationContainer',[],...
'DecorationContainerMode','auto',...
'Color',[0.15 0.15 0.15],...
'ColorMode','auto',...
'Position',[0.500000476837158 -0.0209523809523809 7.105427357601e-15],...
'PositionMode','auto',...
'Interpreter','tex',...
'InterpreterMode','auto',...
'Rotation',0,...
'RotationMode','auto',...
'FontName','Helvetica',...
'FontNameMode','auto',...
'FontSize',9.35,...
'FontSizeMode','auto',...
'FontAngle','normal',...
'FontAngleMode','auto',...
'FontWeight','normal',...
```

```
'FontWeightMode','auto',...
'HorizontalAlignment','center',...
'HorizontalAlignmentMode','auto',...
'VerticalAlignment','top',...
'VerticalAlignmentMode','auto',...
'EdgeColor','none',...
'EdgeColorMode','auto',...
'LineStyle','-',...
'LineStyleMode','auto',...
'LineWidth',0.5,...
'LineWidthMode','auto',...
'BackgroundColor','none',...
'BackgroundColorMode','auto',...
'Margin',3,...
'MarginMode','auto',...
'Clipping','off',...
'ClippingMode','auto',...
'Layer','back',...
'LayerMode','auto',...
'FontSmoothing','on',...
'FontSmoothingMode','auto',...
'IncludeRenderer','on',...
'IsContainer','off',...
'IsContainerMode','auto',...
'HG1EraseMode','auto',...
'BusyAction','queue',...
'Interruptible','on',...
'HitTest','on',...
'HitTestMode','auto',...
'PickableParts','visible',...
'PickablePartsMode','auto',...
'DimensionNames',{  'X' 'Y' 'Z' },...
'DimensionNamesMode','auto',...
'XLimInclude','on',...
'XLimIncludeMode','auto',...
'YLimInclude','on',...
'YLimIncludeMode','auto',...
'ZLimInclude','on',...
'ZLimIncludeMode','auto',...
'CLimInclude','on',...
'CLimIncludeMode','auto',...
'ALimInclude','on',...
'ALimIncludeMode','auto',...
'Description','AxisRulerBase Label',...
'DescriptionMode','auto',...
'Visible','on',...
'VisibleMode','auto',...
'Serializable','on',...
'SerializableMode','auto',...
'HandleVisibility','off',...
'HandleVisibilityMode','auto',...
'TransformForPrintFcnImplicitInvoke','on',...
```

```
'TransformForPrintFcnImplicitInvokeMode','auto');

h5 = get(h2,'ylabel');

set(h5,...
'Parent',h2,...
'Units','data',...
'FontUnits','points',...
'DecorationContainer',[],...
'DecorationContainerMode','auto',...
'Color',[0.15 0.15 0.15],...
'ColorMode','auto',...
'Position',[-0.00931216931216931 0.500000476837158 7.105427357601e-15],...
'PositionMode','auto',...
'Interpreter','tex',...
'InterpreterMode','auto',...
'Rotation',90,...
'RotationMode','auto',...
'FontName','Helvetica',...
'FontNameMode','auto',...
'FontSize',9.35,...
'FontSizeMode','auto',...
'FontAngle','normal',...
'FontAngleMode','auto',...
'FontWeight','normal',...
'FontWeightMode','auto',...
'HorizontalAlignment','center',...
'HorizontalAlignmentMode','auto',...
'VerticalAlignment','bottom',...
'VerticalAlignmentMode','auto',...
'EdgeColor','none',...
'EdgeColorMode','auto',...
'LineStyle','-',...
'LineStyleMode','auto',...
'LineWidth',0.5,...
'LineWidthMode','auto',...
'BackgroundColor','none',...
'BackgroundColorMode','auto',...
'Margin',3,...
'MarginMode','auto',...
'Clipping','off',...
'ClippingMode','auto',...
'Layer','back',...
'LayerMode','auto',...
'FontSmoothing','on',...
'FontSmoothingMode','auto',...
'IncludeRenderer','on',...
'IsContainer','off',...
'IsContainerMode','auto',...
'HG1EraseMode','auto',...
'BusyAction','queue',...
'Interruptible','on',...
```

```
'HitTest','on',...
'HitTestMode','auto',...
'PickableParts','visible',...
'PickablePartsMode','auto',...
'DimensionNames',{  'X' 'Y' 'Z' },...
'DimensionNamesMode','auto',...
'XLimInclude','on',...
'XLimIncludeMode','auto',...
'YLimInclude','on',...
'YLimIncludeMode','auto',...
'ZLimInclude','on',...
'ZLimIncludeMode','auto',...
'CLimInclude','on',...
'CLimIncludeMode','auto',...
'ALimInclude','on',...
'ALimIncludeMode','auto',...
'Description','AxisRulerBase Label',...
'DescriptionMode','auto',...
'Visible','on',...
'VisibleMode','auto',...
'Serializable','on',...
'SerializableMode','auto',...
'HandleVisibility','off',...
'HandleVisibilityMode','auto',...
'TransformForPrintFcnImplicitInvoke','on',...
'TransformForPrintFcnImplicitInvokeMode','auto');

h6 = get(h2,'zlabel');

set(h6,...
'Parent',h2,...
'Units','data',...
'FontUnits','points',...
'DecorationContainer',[],...
'DecorationContainerMode','auto',...
'Color',[0.15 0.15 0.15],...
'ColorMode','auto',...
'Position',[0 0 0],...
'PositionMode','auto',...
'Interpreter','tex',...
'InterpreterMode','auto',...
'Rotation',0,...
'RotationMode','auto',...
'FontName','Helvetica',...
'FontNameMode','auto',...
'FontSize',10,...
'FontSizeMode','auto',...
'FontAngle','normal',...
'FontAngleMode','auto',...
'FontWeight','normal',...
'FontWeightMode','auto',...
'HorizontalAlignment','left',...
```

```
'HorizontalAlignmentMode','auto',...
'VerticalAlignment','middle',...
'VerticalAlignmentMode','auto',...
'EdgeColor','none',...
'EdgeColorMode','auto',...
'LineStyle','-',...
'LineStyleMode','auto',...
'LineWidth',0.5,...
'LineWidthMode','auto',...
'BackgroundColor','none',...
'BackgroundColorMode','auto',...
'Margin',3,...
'MarginMode','auto',...
'Clipping','off',...
'ClippingMode','auto',...
'Layer','middle',...
'LayerMode','auto',...
'FontSmoothing','on',...
'FontSmoothingMode','auto',...
'IncludeRenderer','on',...
'IsContainer','off',...
'IsContainerMode','auto',...
'HG1EraseMode','auto',...
'BusyAction','queue',...
'Interruptible','on',...
'HitTest','on',...
'HitTestMode','auto',...
'PickableParts','visible',...
'PickablePartsMode','auto',...
'DimensionNames',{  'X' 'Y' 'Z' },...
'DimensionNamesMode','auto',...
'XLimInclude','on',...
'XLimIncludeMode','auto',...
'YLimInclude','on',...
'YLimIncludeMode','auto',...
'ZLimInclude','on',...
'ZLimIncludeMode','auto',...
'CLimInclude','on',...
'CLimIncludeMode','auto',...
'ALimInclude','on',...
'ALimIncludeMode','auto',...
'Description','AxisRulerBase Label',...
'DescriptionMode','auto',...
'Visible','off',...
'VisibleMode','auto',...
'Serializable','on',...
'SerializableMode','auto',...
'HandleVisibility','off',...
'HandleVisibilityMode','auto',...
'TransformForPrintFcnImplicitInvoke','on',...
'TransformForPrintFcnImplicitInvokeMode','auto');
```

```
appdata = [];
appdata.lastValidTag = 'SimulationHist';

h7 = axes(...
'Parent',h1,...
'FontUnits',get(0,'defaultaxesFontUnits'),...
'Units','pixels',...
'View',get(0,'defaultaxesView'),...
'CameraPosition',[0.5 0.5 9.16025403784439],...
'CameraPositionMode',get(0,'defaultaxesCameraPositionMode'),...
'CameraTarget',[0.5 0.5 0.5],...
'CameraTargetMode',get(0,'defaultaxesCameraTargetMode'),...
'CameraViewAngle',6.60861036031192,...
'CameraViewAngleMode',get(0,'defaultaxesCameraViewAngleMode'),...
'Projection',get(0,'defaultaxesProjection'),...
'LabelFontSizeMultiplier',get(0,'defaultaxesLabelFontSizeMultiplier'),...
'AmbientLightColor',get(0,'defaultaxesAmbientLightColor'),...
'PlotBoxAspectRatio',[1 0.444444444444444 0.444444444444444],...
'PlotBoxAspectRatioMode',get(0,'defaultaxesPlotBoxAspectRatioMode'),...
'FontName',get(0,'defaultaxesFontName'),...
'FontAngle',get(0,'defaultaxesFontAngle'),...
'FontWeight',get(0,'defaultaxesFontWeight'),...
'FontSmoothing',get(0,'defaultaxesFontSmoothing'),...
'TickLabelInterpreter',get(0,'defaultaxesTickLabelInterpreter'),...
'XDir',get(0,'defaultaxesXDir'),...
'YDir',get(0,'defaultaxesYDir'),...
'ZDir',get(0,'defaultaxesZDir'),...
'ColormapMode',get(0,'defaultaxesColormapMode'),...
'AlphamapMode',get(0,'defaultaxesAlphamapMode'),...
'Layer',get(0,'defaultaxesLayer'),...
'TickLength',get(0,'defaultaxesTickLength'),...
'GridLineStyle',get(0,'defaultaxesGridLineStyle'),...
'MinorGridLineStyle',get(0,'defaultaxesMinorGridLineStyle'),...
'XAxisLocation',get(0,'defaultaxesXAxisLocation'),...
'XTick',[0 0.2 0.4 0.6 0.8 1],...
'XTickMode',get(0,'defaultaxesXTickMode'),...
'XTickLabelRotation',get(0,'defaultaxesXTickLabelRotation'),...
'XScale',get(0,'defaultaxesXScale'),...
'XTickLabel',blanks(0),...
'XMinorTick',get(0,'defaultaxesXMinorTick'),...
'YAxisLocation',get(0,'defaultaxesYAxisLocation'),...
'YTick',[0 0.2 0.4 0.6 0.8 1],...
'YTickMode',get(0,'defaultaxesYTickMode'),...
'YTickLabelRotation',get(0,'defaultaxesYTickLabelRotation'),...
'YScale',get(0,'defaultaxesYScale'),...
'YTickLabel',blanks(0),...
'YMinorTick',get(0,'defaultaxesYMinorTick'),...
'ZTickLabelRotation',get(0,'defaultaxesZTickLabelRotation'),...
'ZScale',get(0,'defaultaxesZScale'),...
'ZMinorTick',get(0,'defaultaxesZMinorTick'),...
'BoxStyle',get(0,'defaultaxesBoxStyle'),...
'LineWidth',get(0,'defaultaxesLineWidth'),...
```

```
'Color',get(0,'defaultaxesColor'),...
'ClippingStyle',get(0,'defaultaxesClippingStyle'),...
'CameraMode',get(0,'defaultaxesCameraMode'),...
'DataSpaceMode',get(0,'defaultaxesDataSpaceMode'),...
'ColorSpaceMode',get(0,'defaultaxesColorSpaceMode'),...
'DecorationContainerMode',get(0,'defaultaxesDecorationContainerMode'),...
'ChildContainerMode',get(0,'defaultaxesChildContainerMode'),...
'BoxFrameMode',get(0,'defaultaxesBoxFrameMode'),...
'XAxisMode','manual',...
'YRulerMode',get(0,'defaultaxesYRulerMode'),...
'ZAxisMode','manual',...
'AmbientLightSourceMode',get(0,'defaultaxesAmbientLightSourceMode'),...
'XGrid',get(0,'defaultaxesXGrid'),...
'XMinorGrid',get(0,'defaultaxesXMinorGrid'),...
'YGrid',get(0,'defaultaxesYGrid'),...
'YMinorGrid',get(0,'defaultaxesYMinorGrid'),...
'ZGrid',get(0,'defaultaxesZGrid'),...
'ZMinorGrid',get(0,'defaultaxesZMinorGrid'),...
'ButtonDownFcn',blanks(0),...
'Interruptible','off',...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} ,...
'DeleteFcn',blanks(0),...
'Tag','SimulationHist',...
'UserData',[],...
'HitTest','off',...
'PickableParts',get(0,'defaultaxesPickableParts'),...
'Position',[374 116 315 140],...
'ActivePositionProperty','position',...
'LooseInset',[92.82 60.61 67.83 41.325],...
'LooseInsetMode',get(0,'defaultaxesLooseInsetMode'),...
'ColorOrderIndex',get(0,'defaultaxesColorOrderIndex'),...
'LineStyleOrder',get(0,'defaultaxesLineStyleOrder'),...
'LineStyleOrderIndex',get(0,'defaultaxesLineStyleOrderIndex'),...
'FontSize',8.5,...
'FontSizeMode',get(0,'defaultaxesFontSizeMode'),...
'TitleFontWeight',get(0,'defaultaxesTitleFontWeight'),...
'TitleFontSizeMultiplier',get(0,'defaultaxesTitleFontSizeMultiplier'),...
'SortMethod','childorder',...
'Clipping',get(0,'defaultaxesClipping'),...
'NextPlot','replacechildren',...
'Box','on',...
'ChildrenMode','manual',...
'Visible',get(0,'defaultaxesVisible'),...
'HandleVisibility',get(0,'defaultaxesHandleVisibility'));

h8 = get(h7,'title');

set(h8,...
'Parent',h7,...
'Units','data',...
'FontUnits','points',...
'DecorationContainer',[],...
```

```
'DecorationContainerMode','auto',...
'Color',[0 0 0],...
'ColorMode','auto',...
'Position',[0.500001609136188 1.01669642857143 0.500000000000007],...
'PositionMode','auto',...
'Interpreter','tex',...
'InterpreterMode','auto',...
'Rotation',0,...
'RotationMode','auto',...
'FontName','Helvetica',...
'FontNameMode','auto',...
'FontSize',9.35,...
'FontSizeMode','auto',...
'FontAngle','normal',...
'FontAngleMode','auto',...
'FontWeight','bold',...
'FontWeightMode','auto',...
'HorizontalAlignment','center',...
'HorizontalAlignmentMode','auto',...
'VerticalAlignment','bottom',...
'VerticalAlignmentMode','auto',...
'EdgeColor','none',...
'EdgeColorMode','auto',...
'LineStyle','-',...
'LineStyleMode','auto',...
'LineWidth',0.5,...
'LineWidthMode','auto',...
'BackgroundColor','none',...
'BackgroundColorMode','auto',...
'Margin',3,...
'MarginMode','auto',...
'Clipping','off',...
'ClippingMode','auto',...
'Layer','middle',...
'LayerMode','auto',...
'FontSmoothing','on',...
'FontSmoothingMode','auto',...
'IncludeRenderer','on',...
'IsContainer','off',...
'IsContainerMode','auto',...
'HG1EraseMode','auto',...
'BusyAction','queue',...
'Interruptible','on',...
'HitTest','on',...
'HitTestMode','auto',...
'PickableParts','visible',...
'PickablePartsMode','auto',...
'DimensionNames',{  'X' 'Y' 'Z' },...
'DimensionNamesMode','auto',...
'XLimInclude','on',...
'XLimIncludeMode','auto',...
'YLimInclude','on',...
```

```
'YLimIncludeMode','auto',...
'ZLimInclude','on',...
'ZLimIncludeMode','auto',...
'CLimInclude','on',...
'CLimIncludeMode','auto',...
'ALimInclude','on',...
'ALimIncludeMode','auto',...
'Description','Axes Title',...
'DescriptionMode','auto',...
'Visible','on',...
'VisibleMode','auto',...
'Serializable','on',...
'SerializableMode','auto',...
'HandleVisibility','off',...
'HandleVisibilityMode','auto',...
'TransformForPrintFcnImplicitInvoke','on',...
'TransformForPrintFcnImplicitInvokeMode','auto');

h9 = get(h7,'xlabel');

set(h9,...
'Parent',h7,...
'Units','data',...
'FontUnits','points',...
'DecorationContainer',[],...
'DecorationContainerMode','auto',...
'Color',[0.15 0.15 0.15],...
'ColorMode','auto',...
'Position',[0.500000476837158 -0.0209523809523809 7.105427357601e-15],...
'PositionMode','auto',...
'Interpreter','tex',...
'InterpreterMode','auto',...
'Rotation',0,...
'RotationMode','auto',...
'FontName','Helvetica',...
'FontNameMode','auto',...
'FontSize',9.35,...
'FontSizeMode','auto',...
'FontAngle','normal',...
'FontAngleMode','auto',...
'FontWeight','normal',...
'FontWeightMode','auto',...
'HorizontalAlignment','center',...
'HorizontalAlignmentMode','auto',...
'VerticalAlignment','top',...
'VerticalAlignmentMode','auto',...
'EdgeColor','none',...
'EdgeColorMode','auto',...
'LineStyle','-',...
'LineStyleMode','auto',...
'LineWidth',0.5,...
'LineWidthMode','auto',...
```

```
‘BackgroundColor’,’none’,...
‘BackgroundColorMode’,’auto’,...
‘Margin’,3,...
‘MarginMode’,’auto’,...
‘Clipping’,’off’,...
‘ClippingMode’,’auto’,...
‘Layer’,’back’,...
‘LayerMode’,’auto’,...
‘FontSmoothing’,’on’,...
‘FontSmoothingMode’,’auto’,...
‘IncludeRenderer’,’on’,...
‘IsContainer’,’off’,...
‘IsContainerMode’,’auto’,...
‘HG1EraseMode’,’auto’,...
‘BusyAction’,’queue’,...
‘Interruptible’,’on’,...
‘HitTest’,’on’,...
‘HitTestMode’,’auto’,...
‘PickableParts’,’visible’,...
‘PickablePartsMode’,’auto’,...
‘DimensionNames’,{  ‘X’ ‘Y’ ‘Z’ },...
‘DimensionNamesMode’,’auto’,...
‘XLimInclude’,’on’,...
‘XLimIncludeMode’,’auto’,...
‘YLimInclude’,’on’,...
‘YLimIncludeMode’,’auto’,...
‘ZLimInclude’,’on’,...
‘ZLimIncludeMode’,’auto’,...
‘CLimInclude’,’on’,...
‘CLimIncludeMode’,’auto’,...
‘ALimInclude’,’on’,...
‘ALimIncludeMode’,’auto’,...
‘Description’,’AxisRulerBase Label’,...
‘DescriptionMode’,’auto’,...
‘Visible’,’on’,...
‘VisibleMode’,’auto’,...
‘Serializable’,’on’,...
‘SerializableMode’,’auto’,...
‘HandleVisibility’,’off’,...
‘HandleVisibilityMode’,’auto’,...
‘TransformForPrintFcnImplicitInvoke’,’on’,...
‘TransformForPrintFcnImplicitInvokeMode’,’auto’);

h10 = get(h7,’ylabel’);

set(h10,...
‘Parent’,h7,...
‘Units’,’data’,...
‘FontUnits’,’points’,...
‘DecorationContainer’,[],...
‘DecorationContainerMode’,’auto’,...
‘Color’,[0.15 0.15 0.15],...
```

```
'ColorMode','auto',...
'Position',[-0.00931216931216916 0.500000476837158 7.105427357601e-15],...
'PositionMode','auto',...
'Interpreter','tex',...
'InterpreterMode','auto',...
'Rotation',90,...
'RotationMode','auto',...
'FontName','Helvetica',...
'FontNameMode','auto',...
'FontSize',9.35,...
'FontSizeMode','auto',...
'FontAngle','normal',...
'FontAngleMode','auto',...
'FontWeight','normal',...
'FontWeightMode','auto',...
'HorizontalAlignment','center',...
'HorizontalAlignmentMode','auto',...
'VerticalAlignment','bottom',...
'VerticalAlignmentMode','auto',...
'EdgeColor','none',...
'EdgeColorMode','auto',...
'LineStyle','-',...
'LineStyleMode','auto',...
'LineWidth',0.5,...
'LineWidthMode','auto',...
'BackgroundColor','none',...
'BackgroundColorMode','auto',...
'Margin',3,...
'MarginMode','auto',...
'Clipping','off',...
'ClippingMode','auto',...
'Layer','back',...
'LayerMode','auto',...
'FontSmoothing','on',...
'FontSmoothingMode','auto',...
'IncludeRenderer','on',...
'IsContainer','off',...
'IsContainerMode','auto',...
'HG1EraseMode','auto',...
'BusyAction','queue',...
'Interruptible','on',...
'HitTest','on',...
'HitTestMode','auto',...
'PickableParts','visible',...
'PickablePartsMode','auto',...
'DimensionNames',{  'X' 'Y' 'Z' },...
'DimensionNamesMode','auto',...
'XLimInclude','on',...
'XLimIncludeMode','auto',...
'YLimInclude','on',...
'YLimIncludeMode','auto',...
'ZLimInclude','on',...
```

```
'ZLimIncludeMode','auto',...
'CLimInclude','on',...
'CLimIncludeMode','auto',...
'ALimInclude','on',...
'ALimIncludeMode','auto',...
'Description','AxisRulerBase Label',...
'DescriptionMode','auto',...
'Visible','on',...
'VisibleMode','auto',...
'Serializable','on',...
'SerializableMode','auto',...
'HandleVisibility','off',...
'HandleVisibilityMode','auto',...
'TransformForPrintFcnImplicitInvoke','on',...
'TransformForPrintFcnImplicitInvokeMode','auto');

h11 = get(h7,'zlabel');

set(h11,...
'Parent',h7,...
'Units','data',...
'FontUnits','points',...
'DecorationContainer',[],...
'DecorationContainerMode','auto',...
'Color',[0.15 0.15 0.15],...
'ColorMode','auto',...
'Position',[0 0 0],...
'PositionMode','auto',...
'Interpreter','tex',...
'InterpreterMode','auto',...
'Rotation',0,...
'RotationMode','auto',...
'FontName','Helvetica',...
'FontNameMode','auto',...
'FontSize',10,...
'FontSizeMode','auto',...
'FontAngle','normal',...
'FontAngleMode','auto',...
'FontWeight','normal',...
'FontWeightMode','auto',...
'HorizontalAlignment','left',...
'HorizontalAlignmentMode','auto',...
'VerticalAlignment','middle',...
'VerticalAlignmentMode','auto',...
'EdgeColor','none',...
'EdgeColorMode','auto',...
'LineStyle','-',...
'LineStyleMode','auto',...
'LineWidth',0.5,...
'LineWidthMode','auto',...
'BackgroundColor','none',...
'BackgroundColorMode','auto',...
```

```
'Margin',3,...
'MarginMode','auto',...
'Clipping','off',...
'ClippingMode','auto',...
'Layer','middle',...
'LayerMode','auto',...
'FontSmoothing','on',...
'FontSmoothingMode','auto',...
'IncludeRenderer','on',...
'IsContainer','off',...
'IsContainerMode','auto',...
'HG1EraseMode','auto',...
'BusyAction','queue',...
'Interruptible','on',...
'HitTest','on',...
'HitTestMode','auto',...
'PickableParts','visible',...
'PickablePartsMode','auto',...
'DimensionNames',{  'X' 'Y' 'Z' },...
'DimensionNamesMode','auto',...
'XLimInclude','on',...
'XLimIncludeMode','auto',...
'YLimInclude','on',...
'YLimIncludeMode','auto',...
'ZLimInclude','on',...
'ZLimIncludeMode','auto',...
'CLimInclude','on',...
'CLimIncludeMode','auto',...
'ALimInclude','on',...
'ALimIncludeMode','auto',...
'Description','AxisRulerBase Label',...
'DescriptionMode','auto',...
'Visible','off',...
'VisibleMode','auto',...
'Serializable','on',...
'SerializableMode','auto',...
'HandleVisibility','off',...
'HandleVisibilityMode','auto',...
'TransformForPrintFcnImplicitInvoke','on',...
'TransformForPrintFcnImplicitInvokeMode','auto');

appdata = [];
appdata.lastValidTag = 'killboxGraphics';

h12 = axes(...
'Parent',h1,...
'FontUnits',get(0,'defaultaxesFontUnits'),...
'Units','characters',...
'CameraPosition',[0.5 0.5 9.16025403784439],...
'CameraPositionMode',get(0,'defaultaxesCameraPositionMode'),...
'CameraTarget',[0.5 0.5 0.5],...
'CameraTargetMode',get(0,'defaultaxesCameraTargetMode'),...
```

```
'CameraViewAngle',6.60861036031192,...
'CameraViewAngleMode',get(0,'defaultaxesCameraViewAngleMode'),...
'PlotBoxAspectRatio',[1 0.705179282868526 0.705179282868526],...
'PlotBoxAspectRatioMode',get(0,'defaultaxesPlotBoxAspectRatioMode'),...
'ColormapMode',get(0,'defaultaxesColormapMode'),...
'AlphamapMode',get(0,'defaultaxesAlphamapMode'),...
'XTick',[0 0.2 0.4 0.6 0.8 1],...
'XTickLabel',blanks(0),...
'YTick',[0 0.2 0.4 0.6 0.8 1],...
'YTickLabel',blanks(0),...
'CameraMode',get(0,'defaultaxesCameraMode'),...
'DataSpaceMode',get(0,'defaultaxesDataSpaceMode'),...
'ColorSpaceMode',get(0,'defaultaxesColorSpaceMode'),...
'DecorationContainerMode',get(0,'defaultaxesDecorationContainerMode'),...
'ChildContainerMode',get(0,'defaultaxesChildContainerMode'),...
'XRulerMode',get(0,'defaultaxesXRulerMode'),...
'YRulerMode',get(0,'defaultaxesYRulerMode'),...
'ZRulerMode',get(0,'defaultaxesZRulerMode'),...
'AmbientLightSourceMode',get(0,'defaultaxesAmbientLightSourceMode'),...
'Tag','killboxGraphics',...
'Position',[79.8 22.9230769230769 50.2 13.6153846153846],...
'ActivePositionProperty','position',...
'ActivePositionPropertyMode',get(0,'defaultaxesActivePositionPropertyMode'),...
'LooseInset',[18.2 4.23076923076923 13.3 2.88461538461538],...
'LooseInsetMode',get(0,'defaultaxesLooseInsetMode'),...
'FontSize',9,...
'FontSizeMode',get(0,'defaultaxesFontSizeMode'),...
'SortMethod','childorder',...
'SortMethodMode',get(0,'defaultaxesSortMethodMode'),...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );

h13 = get(h12,'title');

set(h13,...
'Parent',h12,...
'Units','data',...
'FontUnits','points',...
'DecorationContainer',[],...
'DecorationContainerMode','auto',...
'Color',[0 0 0],...
'ColorMode','auto',...
'Position',[0.500001996636866 1.01398305084746 0.5],...
'PositionMode','auto',...
'Interpreter','tex',...
'InterpreterMode','auto',...
'Rotation',0,...
'RotationMode','auto',...
'FontName','Helvetica',...
'FontNameMode','auto',...
'FontUnitsMode','auto',...
'FontSize',9.9,...
'FontSizeMode','auto',...
```

```
'FontAngle','normal',...
'FontAngleMode','auto',...
'FontWeight','bold',...
'FontWeightMode','auto',...
'HorizontalAlignment','center',...
'HorizontalAlignmentMode','auto',...
'VerticalAlignment','bottom',...
'VerticalAlignmentMode','auto',...
'EdgeColor','none',...
'EdgeColorMode','auto',...
'LineStyle','-',...
'LineStyleMode','auto',...
'LineWidth',0.5,...
'LineWidthMode','auto',...
'BackgroundColor','none',...
'BackgroundColorMode','auto',...
'Margin',3,...
'MarginMode','auto',...
'Clipping','off',...
'ClippingMode','auto',...
'Layer','middle',...
'LayerMode','auto',...
'FontSmoothing','on',...
'FontSmoothingMode','auto',...
'UnitsMode','auto',...
'IncludeRenderer','on',...
'IsContainer','off',...
'IsContainerMode','auto',...
'HG1EraseMode','auto',...
'BusyAction','queue',...
'Interruptible','on',...
'HitTest','on',...
'HitTestMode','auto',...
'PickableParts','visible',...
'PickablePartsMode','auto',...
'DimensionNames',{  'X' 'Y' 'Z' },...
'DimensionNamesMode','auto',...
'XLimInclude','on',...
'XLimIncludeMode','auto',...
'YLimInclude','on',...
'YLimIncludeMode','auto',...
'ZLimInclude','on',...
'ZLimIncludeMode','auto',...
'CLimInclude','on',...
'CLimIncludeMode','auto',...
'ALimInclude','on',...
'ALimIncludeMode','auto',...
'Description','Axes Title',...
'DescriptionMode','auto',...
'Visible','on',...
'VisibleMode','auto',...
'Serializable','on',...
```

```
'SerializableMode','auto',...
'HandleVisibility','off',...
'HandleVisibilityMode','auto',...
'TransformForPrintFcnImplicitInvoke','on',...
'TransformForPrintFcnImplicitInvokeMode','auto');

h14 = get(h12,'xlabel');

set(h14,...
'Parent',h12,...
'Units','data',...
'FontUnits','points',...
'DecorationContainer',[],...
'DecorationContainerMode','auto',...
'Color',[0.15 0.15 0.15],...
'ColorMode','auto',...
'Position',[0.500000476837158 -0.0210922787193972 0],...
'PositionMode','auto',...
'Interpreter','tex',...
'InterpreterMode','auto',...
'Rotation',0,...
'RotationMode','auto',...
'FontName','Helvetica',...
'FontNameMode','auto',...
'FontUnitsMode','auto',...
'FontSize',9.9,...
'FontSizeMode','auto',...
'FontAngle','normal',...
'FontAngleMode','auto',...
'FontWeight','normal',...
'FontWeightMode','auto',...
'HorizontalAlignment','center',...
'HorizontalAlignmentMode','auto',...
'VerticalAlignment','top',...
'VerticalAlignmentMode','auto',...
'EdgeColor','none',...
'EdgeColorMode','auto',...
'LineStyle','-',...
'LineStyleMode','auto',...
'LineWidth',0.5,...
'LineWidthMode','auto',...
'BackgroundColor','none',...
'BackgroundColorMode','auto',...
'Margin',3,...
'MarginMode','auto',...
'Clipping','off',...
'ClippingMode','auto',...
'Layer','back',...
'LayerMode','auto',...
'FontSmoothing','on',...
'FontSmoothingMode','auto',...
'UnitsMode','auto',...
```

```
'IncludeRenderer','on',...
'IsContainer','off',...
'IsContainerMode','auto',...
'HG1EraseMode','auto',...
'BusyAction','queue',...
'Interruptible','on',...
'HitTest','on',...
'HitTestMode','auto',...
'PickableParts','visible',...
'PickablePartsMode','auto',...
'DimensionNames',{  'X' 'Y' 'Z' },...
'DimensionNamesMode','auto',...
'XLimInclude','on',...
'XLimIncludeMode','auto',...
'YLimInclude','on',...
'YLimIncludeMode','auto',...
'ZLimInclude','on',...
'ZLimIncludeMode','auto',...
'CLimInclude','on',...
'CLimIncludeMode','auto',...
'ALimInclude','on',...
'ALimIncludeMode','auto',...
'Description','AxisRulerBase Label',...
'DescriptionMode','auto',...
'Visible','on',...
'VisibleMode','auto',...
'Serializable','on',...
'SerializableMode','auto',...
'HandleVisibility','off',...
'HandleVisibilityMode','auto',...
'TransformForPrintFcnImplicitInvoke','on',...
'TransformForPrintFcnImplicitInvokeMode','auto');

h15 = get(h12,'ylabel');

set(h15,...
'Parent',h12,...
'Units','data',...
'FontUnits','points',...
'DecorationContainer',[],...
'DecorationContainerMode','auto',...
'Color',[0.15 0.15 0.15],...
'ColorMode','auto',...
'Position',[-0.0148738379814077 0.500000476837158 0],...
'PositionMode','auto',...
'Interpreter','tex',...
'InterpreterMode','auto',...
'Rotation',90,...
'RotationMode','auto',...
'FontName','Helvetica',...
'FontNameMode','auto',...
'FontUnitsMode','auto',...
```

```
'FontSize',9.9,...
'FontSizeMode','auto',...
'FontAngle','normal',...
'FontAngleMode','auto',...
'FontWeight','normal',...
'FontWeightMode','auto',...
'HorizontalAlignment','center',...
'HorizontalAlignmentMode','auto',...
'VerticalAlignment','bottom',...
'VerticalAlignmentMode','auto',...
'EdgeColor','none',...
'EdgeColorMode','auto',...
'LineStyle','-',...
'LineStyleMode','auto',...
'LineWidth',0.5,...
'LineWidthMode','auto',...
'BackgroundColor','none',...
'BackgroundColorMode','auto',...
'Margin',3,...
'MarginMode','auto',...
'Clipping','off',...
'ClippingMode','auto',...
'Layer','back',...
'LayerMode','auto',...
'FontSmoothing','on',...
'FontSmoothingMode','auto',...
'UnitsMode','auto',...
'IncludeRenderer','on',...
'IsContainer','off',...
'IsContainerMode','auto',...
'HG1EraseMode','auto',...
'BusyAction','queue',...
'Interruptible','on',...
'HitTest','on',...
'HitTestMode','auto',...
'PickableParts','visible',...
'PickablePartsMode','auto',...
'DimensionNames',{  'X' 'Y' 'Z' },...
'DimensionNamesMode','auto',...
'XLimInclude','on',...
'XLimIncludeMode','auto',...
'YLimInclude','on',...
'YLimIncludeMode','auto',...
'ZLimInclude','on',...
'ZLimIncludeMode','auto',...
'CLimInclude','on',...
'CLimIncludeMode','auto',...
'ALimInclude','on',...
'ALimIncludeMode','auto',...
'Description','AxisRulerBase Label',...
'DescriptionMode','auto',...
'Visible','on',...
```

```
'VisibleMode','auto',...
'Serializable','on',...
'SerializableMode','auto',...
'HandleVisibility','off',...
'HandleVisibilityMode','auto',...
'TransformForPrintFcnImplicitInvoke','on',...
'TransformForPrintFcnImplicitInvokeMode','auto');

h16 = get(h12,'zlabel');

set(h16,...
'Parent',h12,...
'Units','data',...
'FontUnits','points',...
'DecorationContainer',[],...
'DecorationContainerMode','auto',...
'Color',[0.15 0.15 0.15],...
'ColorMode','auto',...
'Position',[0 0 0],...
'PositionMode','auto',...
'Interpreter','tex',...
'InterpreterMode','auto',...
'Rotation',0,...
'RotationMode','auto',...
'FontName','Helvetica',...
'FontNameMode','auto',...
'FontUnitsMode','auto',...
'FontSize',10,...
'FontSizeMode','auto',...
'FontAngle','normal',...
'FontAngleMode','auto',...
'FontWeight','normal',...
'FontWeightMode','auto',...
'HorizontalAlignment','left',...
'HorizontalAlignmentMode','auto',...
'VerticalAlignment','middle',...
'VerticalAlignmentMode','auto',...
'EdgeColor','none',...
'EdgeColorMode','auto',...
'LineStyle','-',...
'LineStyleMode','auto',...
'LineWidth',0.5,...
'LineWidthMode','auto',...
'BackgroundColor','none',...
'BackgroundColorMode','auto',...
'Margin',3,...
'MarginMode','auto',...
'Clipping','off',...
'ClippingMode','auto',...
'Layer','middle',...
'LayerMode','auto',...
'FontSmoothing','on',...
```

```
'FontSmoothingMode','auto',...
'UnitsMode','auto',...
'IncludeRenderer','on',...
'IsContainer','off',...
'IsContainerMode','auto',...
'HG1EraseMode','auto',...
'BusyAction','queue',...
'Interruptible','on',...
'HitTest','on',...
'HitTestMode','auto',...
'PickableParts','visible',...
'PickablePartsMode','auto',...
'DimensionNames',{  'X' 'Y' 'Z' },...
'DimensionNamesMode','auto',...
'XLimInclude','on',...
'XLimIncludeMode','auto',...
'YLimInclude','on',...
'YLimIncludeMode','auto',...
'ZLimInclude','on',...
'ZLimIncludeMode','auto',...
'CLimInclude','on',...
'CLimIncludeMode','auto',...
'ALimInclude','on',...
'ALimIncludeMode','auto',...
'Description','AxisRulerBase Label',...
'DescriptionMode','auto',...
'Visible','off',...
'VisibleMode','auto',...
'Serializable','on',...
'SerializableMode','auto',...
'HandleVisibility','off',...
'HandleVisibilityMode','auto',...
'TransformForPrintFcnImplicitInvoke','on',...
'TransformForPrintFcnImplicitInvokeMode','auto');

appdata = [];
appdata.lastValidTag = 'SetupPanel';

h17 = uipanel(...
'Parent',h1,...
'FontUnits',get(0,'defaultuipanelFontUnits'),...
'Units','pixels',...
'BorderType','beveledin',...
'Title','Simulation Setup',...
'Tag','SetupPanel',...
'Position',[10 275 325 215],...
'FontSize',11,...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );

appdata = [];
appdata.lastValidTag = 'RunSim';
```

```
h18 = uicontrol(...
'Parent',h17,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'String','Run Simulation',...
'Style',get(0,'defaultuicontrolStyle'),...
'Position',[98 14 125 20],...
'Callback',@(hObject,eventdata)PerformSeabedWarfare('RunSimulation',hObject,eventdata,gui
data(hObject)),...
'Children',[],...
'Tag','RunSim',...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );

appdata = [];
appdata.lastValidTag = 'NumSimulationTxt';

h19 = uicontrol(...
'Parent',h17,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'HorizontalAlignment','left',...
'String','Number of Replications (per scenario):',...
'Style','text',...
'Position',[14 115 203 20],...
'Children',[],...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} ,...
'Tag','NumSimulationTxt',...
'FontSize',get(0,'defaultuicontrolFontSize'));

appdata = [];
appdata.lastValidTag = 'NumSimulation';

h20 = uicontrol(...
'Parent',h17,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'String','100',...
'Style','edit',...
'Position',[231 118 75 20],...
'BackgroundColor',[1 1 1],...
'Callback',blanks(0),...
'Children',[],...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} ,...
'Tag','NumSimulation');

appdata = [];
appdata.lastValidTag = 'ScenarioFile';

h21 = uicontrol(...
'Parent',h17,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
```

```
'String',blanks(0),...
'Style','edit',...
'Position',[14 73 292 20],...
'BackgroundColor',[1 1 1],...
'Callback',blanks(0),...
'Children',[],...
'ButtonDownFcn',blanks(0),...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} ,...
'DeleteFcn',blanks(0),...
'Tag','ScenarioFile',...
'KeyPressFcn',blanks(0));

appdata = [];
appdata.lastValidTag = 'FileBrowser';

h22 = uicontrol(...
'Parent',h17,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units','characters',...
'String','Browse',...
'Style',get(0,'defaultuicontrolStyle'),...
'Position',[47 3.53846153846154 14 1.69230769230769],...
'Callback',@(hObject,eventdata)PerformSeabedWarfare('UploadScenarioFile',hObject,eventdat
a,guidata(hObject)),...
'Children',[],...
'Tag','FileBrowser',...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );

appdata = [];
appdata.lastValidTag = 'text3';

h23 = uicontrol(...
'Parent',h17,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'HorizontalAlignment','left',...
'String','Mission Scenario Input File:',...
'Style','text',...
'Position',[14 93 169 20],...
'Children',[],...
'ButtonDownFcn',blanks(0),...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} ,...
'DeleteFcn',blanks(0),...
'Tag','text3',...
'FontSize',get(0,'defaultuicontrolFontSize'));

appdata = [];
appdata.lastValidTag = 'MissionEvaluationTxt';

h24 = uicontrol(...
'Parent',h17,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
```

```
'Units',get(0,'defaultuicontrolUnits'),...
'HorizontalAlignment','left',...
'String','Mission Evaluation:',...
'Style','text',...
'Position',[14 169 203 20],...
'Children',[],...
'ButtonDownFcn',blanks(0),...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} ,...
'DeleteFcn',blanks(0),...
'Tag','MissionEvaluationTxt',...
'FontSize',get(0,'defaultuicontrolFontSize'));

appdata = [];
appdata.lastValidTag = 'MissionEvaluationType';

h25 = uicontrol(...
'Parent',h17,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'String',{  'Independently'; 'Successively' },...
'Style','popupmenu',...
'Value',1,...
'ValueMode',get(0,'defaultuicontrolValueMode'),...
'Position',[190 172 116 20],...
'BackgroundColor',[1 1 1],...
'Callback',@(hObject,eventdata)PerformSeabedWarfare('MissionEvaluationType_Callback',hObj
ect,eventdata,guidata(hObject)),...
'Children',[],...
'ButtonDownFcn',blanks(0),...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} ,...
'DeleteFcn',blanks(0),...
'Tag','MissionEvaluationType',...
'KeyPressFcn',blanks(0));

appdata = [];
appdata.lastValidTag = 'MissionTypeTxt';

h26 = uicontrol(...
'Parent',h17,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'HorizontalAlignment','left',...
'String','Mission Type:',...
'Style','text',...
'Position',[15 142 203 20],...
'Children',[],...
'ButtonDownFcn',blanks(0),...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} ,...
'DeleteFcn',blanks(0),...
'Tag','MissionTypeTxt',...
'FontSize',get(0,'defaultuicontrolFontSize'));
```

```
appdata = [];
appdata.lastValidTag = 'MissionType';

h27 = uicontrol(...
'Parent',h17,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'String',{  'All Missions'; 'IPOE'; 'ISR Field Deployment'; 'Effects Field Deployment';
'Engage & Strike' },...
'Style','popupmenu',...
'Value',1,...
'ValueMode',get(0,'defaultuicontrolValueMode'),...
'Position',[191 145 116 20],...
'BackgroundColor',[1 1 1],...
'Callback',blanks(0),...
'Children',[],...
'ButtonDownFcn',blanks(0),...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} ,...
'DeleteFcn',blanks(0),...
'Tag','MissionType',...
'KeyPressFcn',blanks(0));

appdata = [];
appdata.lastValidTag = 'SimulationTable';

h28 = uitable(...
'Parent',h1,...
'FontUnits',get(0,'defaultuitableFontUnits'),...
'Units',get(0,'defaultuitableUnits'),...
'BackgroundColor',[1 1 1;0.941176470588235 0.941176470588235 0.941176470588235],...
'ColumnName',{  'IPOE'; 'ISR'; 'EF'; 'STRIKE' },...
'ColumnWidth',{  'auto' 'auto' 'auto' 'auto' },...
'Data',{  blanks(0) blanks(0) blanks(0) blanks(0); blanks(0) blanks(0) blanks(0)
blanks(0); blanks(0) blanks(0) blanks(0) blanks(0); blanks(0) blanks(0) blanks(0)
blanks(0) },...
'RowName',{  'Success'; 'Failure - Asset Saved'; 'Failure - Asset Lost' },...
'Position',[81 6 533 94],...
'ColumnEditable',[false false false false],...
'ColumnFormat',{  [] [] [] [] },...
'RearrangeableColumns',get(0,'defaultuitableRearrangeableColumns'),...
'RowStriping',get(0,'defaultuitableRowStriping'),...
'CellEditCallback',blanks(0),...
'CellSelectionCallback',blanks(0),...
'Children',[],...
'ForegroundColor',get(0,'defaultuitableForegroundColor'),...
'Enable',get(0,'defaultuitableEnable'),...
'TooltipString',blanks(0),...
'Visible',get(0,'defaultuitableVisible'),...
'ButtonDownFcn',blanks(0),...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} ,...
'DeleteFcn',blanks(0),...
'Tag','SimulationTable',...
```

```
'UserData',[],...
'KeyPressFcn',blanks(0),...
'KeyReleaseFcn',blanks(0),...
'HandleVisibility',get(0,'defaultuitableHandleVisibility'),...
'FontSize',get(0,'defaultuitableFontSize'),...
'FontName',get(0,'defaultuitableFontName'),...
'FontAngle',get(0,'defaultuitableFontAngle'),...
'FontWeight',get(0,'defaultuitableFontWeight'));


hsingleton = h1;
end

% --- Set application data first then calling the CreateFcn.
function local_CreateFcn(hObject, eventdata, createfcn, appdata)

if ~isempty(appdata)
   names = fieldnames(appdata);
   for i=1:length(names)
       name = char(names(i));
       setappdata(hObject, name, getfield(appdata,name));
   end
end

if ~isempty(createfcn)
   if isa(createfcn,'function_handle')
       createfcn(hObject, eventdata);
   else
       eval(createfcn);
   end
end
end

% --- Handles default GUIDE GUI creation and callback dispatch
function varargout = gui_mainfcn(gui_State, varargin)

gui_StateFields =  {'gui_Name'
    'gui_Singleton'
    'gui_OpeningFcn'
    'gui_OutputFcn'
    'gui_LayoutFcn'
    'gui_Callback'};
gui_Mfile = '';
for i=1:length(gui_StateFields)
    if ~isfield(gui_State, gui_StateFields{i})
        error(message('MATLAB:guide:StateFieldNotFound', gui_StateFields{ i },
gui_Mfile));
    elseif isequal(gui_StateFields{i}, 'gui_Name')
        gui_Mfile = [gui_State.(gui_StateFields{i}), '.m'];
    end
end
```

```
numargin = length(varargin);

if numargin == 0
    % PerformSeabedWarfare
    % create the GUI only if we are not in the process of loading it
    % already
    gui_Create = true;
elseif local_isInvokeActiveXCallback(gui_State, varargin{:})
    % PerformSeabedWarfare(ACTIVEX,...)
    vin{1} = gui_State.gui_Name;
    vin{2} = [get(varargin{1}.Peer, 'Tag'), '_', varargin{end}];
    vin{3} = varargin{1};
    vin{4} = varargin{end-1};
    vin{5} = guidata(varargin{1}.Peer);
    feval(vin{:});
    return;
elseif local_isInvokeHGCallback(gui_State, varargin{:})
    % PerformSeabedWarfare('CALLBACK',hObject,eventData,handles,...)
    gui_Create = false;
else
    % PerformSeabedWarfare(...)
    % create the GUI and hand varargin to the openingfcn
    gui_Create = true;
end

if ~gui_Create
    % In design time, we need to mark all components possibly created in
    % the coming callback evaluation as non-serializable. This way, they
    % will not be brought into GUIDE and not be saved in the figure file
    % when running/saving the GUI from GUIDE.
    designEval = false;
    if (numargin>1 && ishghandle(varargin{2}))
        fig = varargin{2};
        while ~isempty(fig) && ~ishghandle(fig,'figure')
            fig = get(fig,'parent');
        end

        designEval = isappdata(0,'CreatingGUIDEFigure') ||
(isscalar(fig)&&isprop(fig,'GUIDEFigure'));
    end

    if designEval
        beforeChildren = findall(fig);
    end

    % evaluate the callback now
    varargin{1} = gui_State.gui_Callback;
    if nargout
        [varargout{1:nargout}] = feval(varargin{:});
    else
        feval(varargin{:});
    end
```

```
    % Set serializable of objects created in the above callback to off in
    % design time. Need to check whether figure handle is still valid in
    % case the figure is deleted during the callback dispatching.
    if designEval && ishghandle(fig)
        set(setdiff(findall(fig),beforeChildren), 'Serializable','off');
    end
else
    if gui_State.gui_Singleton
        gui_SingletonOpt = 'reuse';
    else
        gui_SingletonOpt = 'new';
    end

    % Check user passing 'visible' P/V pair first so that its value can be
    % used by oepnfig to prevent flickering
    gui_Visible = 'auto';
    gui_VisibleInput = '';
    for index=1:2:length(varargin)
        if length(varargin) == index || ~ischar(varargin{index})
            break;
        end

        % Recognize 'visible' P/V pair
        len1 = min(length('visible'),length(varargin{index}));
        len2 = min(length('off'),length(varargin{index+1}));
        if ischar(varargin{index+1}) && strncmpi(varargin{index},'visible',len1) && len2
> 1
            if strncmpi(varargin{index+1},'off',len2)
                gui_Visible = 'invisible';
                gui_VisibleInput = 'off';
            elseif strncmpi(varargin{index+1},'on',len2)
                gui_Visible = 'visible';
                gui_VisibleInput = 'on';
            end
        end
    end

    % Open fig file with stored settings. Note: This executes all component
    % specific CreateFunctions with an empty HANDLES structure.


    % Do feval on layout code in m-file if it exists
    gui_Exported = ~isempty(gui_State.gui_LayoutFcn);
    % this application data is used to indicate the running mode of a GUIDE
    % GUI to distinguish it from the design mode of the GUI in GUIDE. it is
    % only used by actxproxy at this time.
    setappdata(0,genvarname(['OpenGuiWhenRunning_', gui_State.gui_Name]),1);
    if gui_Exported
        gui_hFigure = feval(gui_State.gui_LayoutFcn, gui_SingletonOpt);

        % make figure invisible here so that the visibility of figure is
```

```
        % consistent in OpeningFcn in the exported GUI case
        if isempty(gui_VisibleInput)
            gui_VisibleInput = get(gui_hFigure,'Visible');
        end
        set(gui_hFigure,'Visible','off')

        % openfig (called by local_openfig below) does this for guis without
        % the LayoutFcn. Be sure to do it here so guis show up on screen.
        movegui(gui_hFigure,'onscreen');
    else
        gui_hFigure = local_openfig(gui_State.gui_Name, gui_SingletonOpt, gui_Visible);
        % If the figure has InGUIInitialization it was not completely created
        % on the last pass. Delete this handle and try again.
        if isappdata(gui_hFigure, 'InGUIInitialization')
            delete(gui_hFigure);
            gui_hFigure = local_openfig(gui_State.gui_Name, gui_SingletonOpt,
gui_Visible);
        end
    end
    if isappdata(0, genvarname(['OpenGuiWhenRunning_', gui_State.gui_Name]))
        rmappdata(0,genvarname(['OpenGuiWhenRunning_', gui_State.gui_Name]));
    end

    % Set flag to indicate starting GUI initialization
    setappdata(gui_hFigure,'InGUIInitialization',1);

    % Fetch GUIDE Application options
    gui_Options = getappdata(gui_hFigure,'GUIDEOptions');
    % Singleton setting in the GUI MATLAB code file takes priority if different
    gui_Options.singleton = gui_State.gui_Singleton;

    if ~isappdata(gui_hFigure,'GUIOnScreen')
        % Adjust background color
        if gui_Options.syscolorfig
            set(gui_hFigure,'Color', get(0,'DefaultUicontrolBackgroundColor'));
        end

        % Generate HANDLES structure and store with GUIDATA. If there is
        % user set GUI data already, keep that also.
        data = guidata(gui_hFigure);
        handles = guihandles(gui_hFigure);
        if ~isempty(handles)
            if isempty(data)
                data = handles;
            else
                names = fieldnames(handles);
                for k=1:length(names)
                    data.(char(names(k)))=handles.(char(names(k)));
                end
            end
        end
        guidata(gui_hFigure, data);
```

```
    end

    % Apply input P/V pairs other than 'visible'
    for index=1:2:length(varargin)
        if length(varargin) == index || ~ischar(varargin{index})
            break;
        end

        len1 = min(length('visible'),length(varargin{index}));
        if ~strncmpi(varargin{index},'visible',len1)
            try set(gui_hFigure, varargin{index}, varargin{index+1}), catch break, end
        end
    end

    % If handle visibility is set to 'callback', turn it on until finished
    % with OpeningFcn
    gui_HandleVisibility = get(gui_hFigure,'HandleVisibility');
    if strcmp(gui_HandleVisibility, 'callback')
        set(gui_hFigure,'HandleVisibility', 'on');
    end

    feval(gui_State.gui_OpeningFcn, gui_hFigure, [], guidata(gui_hFigure), varargin{:});

    if isscalar(gui_hFigure) && ishghandle(gui_hFigure)
        % Handle the default callbacks of predefined toolbar tools in this
        % GUI, if any
        guidemfile('restoreToolbarToolPredefinedCallback',gui_hFigure);

        % Update handle visibility
        set(gui_hFigure,'HandleVisibility', gui_HandleVisibility);

        % Call openfig again to pick up the saved visibility or apply the
        % one passed in from the P/V pairs
        if ~gui_Exported
            gui_hFigure = local_openfig(gui_State.gui_Name, 'reuse',gui_Visible);
        elseif ~isempty(gui_VisibleInput)
            set(gui_hFigure,'Visible',gui_VisibleInput);
        end
        if strcmpi(get(gui_hFigure, 'Visible'), 'on')
            figure(gui_hFigure);

            if gui_Options.singleton
                setappdata(gui_hFigure,'GUIOnScreen', 1);
            end
        end

        % Done with GUI initialization
        if isappdata(gui_hFigure,'InGUIInitialization')
            rmappdata(gui_hFigure,'InGUIInitialization');
        end

        % If handle visibility is set to 'callback', turn it on until
```

```
        % finished with OutputFcn
        gui_HandleVisibility = get(gui_hFigure,'HandleVisibility');
        if strcmp(gui_HandleVisibility, 'callback')
            set(gui_hFigure,'HandleVisibility', 'on');
        end
        gui_Handles = guidata(gui_hFigure);
    else
        gui_Handles = [];
    end

    if nargout
        [varargout{1:nargout}] = feval(gui_State.gui_OutputFcn, gui_hFigure, [],
gui_Handles);
    else
        feval(gui_State.gui_OutputFcn, gui_hFigure, [], gui_Handles);
    end

    if isscalar(gui_hFigure) && ishghandle(gui_hFigure)
        set(gui_hFigure,'HandleVisibility', gui_HandleVisibility);
    end
end
end

function gui_hFigure = local_openfig(name, singleton, visible)

% openfig with three arguments was new from R13. Try to call that first, if
% failed, try the old openfig.
if nargin('openfig') == 2
    % OPENFIG did not accept 3rd input argument until R13,
    % toggle default figure visible to prevent the figure
    % from showing up too soon.
    gui_OldDefaultVisible = get(0,'defaultFigureVisible');
    set(0,'defaultFigureVisible','off');
    gui_hFigure = matlab.hg.internal.openfigLegacy(name, singleton);
    set(0,'defaultFigureVisible',gui_OldDefaultVisible);
else
    % Call version of openfig that accepts 'auto' option"
    gui_hFigure = matlab.hg.internal.openfigLegacy(name, singleton, visible);
%     %workaround for CreateFcn not called to create ActiveX
%
peers=findobj(findall(allchild(gui_hFigure)),'type','uicontrol','style','text');
%         for i=1:length(peers)
%             if isappdata(peers(i),'Control')
%                 actxproxy(peers(i));
%             end
%         end
end
end

function result = local_isInvokeActiveXCallback(gui_State, varargin)

try
```

```
    result = ispc && iscom(varargin{1}) ...
             && isequal(varargin{1},gcbo);
catch
    result = false;
end
end

function result = local_isInvokeHGCallback(gui_State, varargin)

try
    fhandle = functions(gui_State.gui_Callback);
    result = ~isempty(findstr(gui_State.gui_Name,fhandle.file)) || ...
             (ischar(varargin{1}) ...
             && isequal(ishghandle(varargin{2}), 1) ...
             && (~isempty(strfind(varargin{1},[get(varargin{2}, 'Tag'), '_'])) || ...
                ~isempty(strfind(varargin{1}, '_CreateFcn'))) );
catch
    result = false;
end
end
```

B. SIMULATION INITIALIZATION

```
function MISPARAMS = InitializeSimulation(scenarioFile)
%%-------------------------------------------------------------------------
% function MISPARAMS = InitializeSimulation
%   This initializes the simulation parameters utilizes the lookup table
%   stored.
%%-------------------------------------------------------------------------

global misNum numMissions scenRaw
persistent SIMPARAMS

%Avoid performance lag, parse the excel only once.
if isempty(SIMPARAMS)
    [scenNum,~,scenRaw] = xlsread(scenarioFile);
    SIMPARAMS.allMissionNumber = [1:numMissions]';
    SIMPARAMS.allEnvirDifficulty = nan(length(SIMPARAMS.allMissionNumber),1);
    SIMPARAMS.allSizes = scenNum(:,5);
    SIMPARAMS.allTargetLured = scenNum(:,23);

    SIMPARAMS.allUnderseaSys = [];
    SIMPARAMS.allUnderseaSys.numSystems = scenNum(:,31);
    SIMPARAMS.allUnderseaSys.DetProbability = scenNum(:,13);
    SIMPARAMS.allUnderseaSys.ClsProbability = scenNum(:,14);
    SIMPARAMS.allUnderseaSys.OpDetProbability = scenNum(:,6);
    SIMPARAMS.allUnderseaSys.OpClsProbability = scenNum(:,7);
    SIMPARAMS.allUnderseaSys.OpTrkProbability = scenNum(:,8);
    SIMPARAMS.allUnderseaSys.EvsProbability = zeros(length(SIMPARAMS.allMissionNumber));
    SIMPARAMS.allUnderseaSys.IPOEProbability = scenNum(:,10);
```

```
    SIMPARAMS.allUnderseaSys.NumIPOEAttempts = scenNum(:,32);
    SIMPARAMS.allUnderseaSys.ISRDepProbability = scenNum(:,11);
    SIMPARAMS.allUnderseaSys.EFDepProbability = scenNum(:,12);
    SIMPARAMS.allUnderseaSys.EnvirThreshold = nan(length(SIMPARAMS.allMissionNumber),1);
    SIMPARAMS.allUnderseaSys.Endurance = scenNum(:,37);

    SIMPARAMS.allISRSys = [];
    SIMPARAMS.allISRSys.numSystems = scenNum(:,33);
    SIMPARAMS.allISRSys.KillboxResolution = nan(length(SIMPARAMS.allMissionNumber),1);
    SIMPARAMS.allISRSys.numRequired = scenNum(:,35);
    SIMPARAMS.allISRSys.OpDetProbability = scenNum(:,15);
    SIMPARAMS.allISRSys.OpClsProbability = scenNum(:,16);
    SIMPARAMS.allISRSys.OpTrkProbability = scenNum(:,17);
    SIMPARAMS.allISRSys.DetProbability = scenNum(:,18);
    SIMPARAMS.allISRSys.ClsProbability = scenNum(:,19);
    SIMPARAMS.allISRSys.EnvirThreshold = nan(length(SIMPARAMS.allMissionNumber),1);

    SIMPARAMS.allEFSys = [];
    SIMPARAMS.allEFSys.numSystems = scenNum(:,34);
    SIMPARAMS.allEFSys.KillboxResolution = nan(length(SIMPARAMS.allMissionNumber),1);
    SIMPARAMS.allEFSys.numRequired = scenNum(:,36);
    SIMPARAMS.allEFSys.DetProbability = scenNum(:,21);
    SIMPARAMS.allEFSys.ClsProbability = scenNum(:,22);
    SIMPARAMS.allEFSys.Lethality = scenNum(:,20);
    SIMPARAMS.allEFSys.EnvirThreshold = nan(length(SIMPARAMS.allMissionNumber),1);

    SIMPARAMS.allOPFOR = [];
    SIMPARAMS.allOPFOR.EvsProbability = scenNum(:,30);
    SIMPARAMS.allOPFOR.DetProbability = scenNum(:,24:26);
    SIMPARAMS.allOPFOR.ClsProbability = scenNum(:,27:29);
    return
end

MISPARAMS.MissionScenario = SIMPARAMS.allMissionNumber(misNum);
MISPARAMS.EnvirDifficulty = SIMPARAMS.allEnvirDifficulty(misNum);
MISPARAMS.KillboxSize = SIMPARAMS.allSizes(misNum);
MISPARAMS.TargetLured = SIMPARAMS.allTargetLured(misNum);

MISPARAMS.UnderseaSystems = [];
MISPARAMS.UnderseaSystems.numSystems = SIMPARAMS.allUnderseaSys.numSystems(misNum);
MISPARAMS.UnderseaSystems.DetProbability =
SIMPARAMS.allUnderseaSys.DetProbability(misNum);
MISPARAMS.UnderseaSystems.ClsProbability =
SIMPARAMS.allUnderseaSys.ClsProbability(misNum);
MISPARAMS.UnderseaSystems.OpDetProbability =
SIMPARAMS.allUnderseaSys.OpDetProbability(misNum);
MISPARAMS.UnderseaSystems.OpClsProbability =
SIMPARAMS.allUnderseaSys.OpClsProbability(misNum);
MISPARAMS.UnderseaSystems.OpTrkProbability =
SIMPARAMS.allUnderseaSys.OpTrkProbability(misNum);
MISPARAMS.UnderseaSystems.ClsProbability =
SIMPARAMS.allUnderseaSys.ClsProbability(misNum);
```

```
MISPARAMS.UnderseaSystems.EvsProbability =
SIMPARAMS.allUnderseaSys.EvsProbability(misNum);
MISPARAMS.UnderseaSystems.IPOEProbability =
SIMPARAMS.allUnderseaSys.IPOEProbability(misNum);
MISPARAMS.UnderseaSystems.NumIPOEAttempts =
SIMPARAMS.allUnderseaSys.NumIPOEAttempts(misNum);
MISPARAMS.UnderseaSystems.ISRDepProbability =
SIMPARAMS.allUnderseaSys.ISRDepProbability(misNum);
MISPARAMS.UnderseaSystems.EFDepProbability =
SIMPARAMS.allUnderseaSys.EFDepProbability(misNum);
MISPARAMS.UnderseaSystems.EnvirThreshold =
SIMPARAMS.allUnderseaSys.EnvirThreshold(misNum);
MISPARAMS.UnderseaSystems.Endurance = SIMPARAMS.allUnderseaSys.Endurance(misNum);

MISPARAMS.ISRSystems = [];
MISPARAMS.ISRSystems.numSystems = SIMPARAMS.allISRSys.numSystems(misNum);
MISPARAMS.ISRSystems.KillboxResolution = SIMPARAMS.allISRSys.KillboxResolution(misNum);
MISPARAMS.ISRSystems.numRequired = SIMPARAMS.allISRSys.numRequired(misNum);
MISPARAMS.ISRSystems.DetProbability = SIMPARAMS.allISRSys.DetProbability(misNum);
MISPARAMS.ISRSystems.ClsProbability = SIMPARAMS.allISRSys.ClsProbability(misNum);
MISPARAMS.ISRSystems.OpDetProbability = SIMPARAMS.allISRSys.OpDetProbability(misNum);
MISPARAMS.ISRSystems.OpClsProbability = SIMPARAMS.allISRSys.OpClsProbability(misNum);
MISPARAMS.ISRSystems.OpTrkProbability = SIMPARAMS.allISRSys.OpTrkProbability(misNum);
MISPARAMS.ISRSystems.EnvirThreshold = SIMPARAMS.allISRSys.EnvirThreshold(misNum);


MISPARAMS.EFSystems = [];
MISPARAMS.EFSystems.numSystems = SIMPARAMS.allEFSys.numSystems(misNum);
MISPARAMS.EFSystems.KillboxResolution = SIMPARAMS.allEFSys.KillboxResolution(misNum);
MISPARAMS.EFSystems.numRequired = SIMPARAMS.allEFSys.numRequired(misNum);
MISPARAMS.EFSystems.DetProbability = SIMPARAMS.allEFSys.DetProbability(misNum);
MISPARAMS.EFSystems.ClsProbability = SIMPARAMS.allEFSys.ClsProbability(misNum);
MISPARAMS.EFSystems.Lethality = SIMPARAMS.allEFSys.Lethality(misNum);
MISPARAMS.EFSystems.EnvirThreshold = SIMPARAMS.allEFSys.EnvirThreshold(misNum);

MISPARAMS.OpSystems = [];
MISPARAMS.OpSystems.EvsProbability = SIMPARAMS.allOPFOR.EvsProbability(misNum);
MISPARAMS.OpSystems.DetProbability = SIMPARAMS.allOPFOR.DetProbability(misNum,:);
MISPARAMS.OpSystems.ClsProbability = SIMPARAMS.allOPFOR.ClsProbability(misNum,:);

%Environments are treated ISO right now, set environment threhsold
%evaluation to FALSE.
MISPARAMS.ISOEnvir = 1;

%Boost detectability by 20% if target was lured
if MISPARAMS.TargetLured
    MISPARAMS.UnderseaSystems.DetProbability =
min(MISPARAMS.UnderseaSystems.DetProbability*1.2,1);
    MISPARAMS.UnderseaSystems.ClsProbability =
min(MISPARAMS.UnderseaSystems.ClsProbability*1.2,1);
    MISPARAMS.ISRSystems.DetProbability = min(MISPARAMS.ISRSystems.DetProbability*1.2,1);
    MISPARAMS.ISRSystems.ClsProbability = min(MISPARAMS.ISRSystems.ClsProbability*1.2,1);
```

```
    MISPARAMS.EFSystems.DetProbability = min(MISPARAMS.EFSystems.DetProbability*1.2,1);
    MISPARAMS.EFSystems.ClsProbability = min(MISPARAMS.EFSystems.ClsProbability*1.2,1);
end
end
```

C. MISSION SIMULATION MODELS

1. Mission 1: Intelligence Preparation of the Operational Environment

```
function MISRESULTS = ConductIPOE(MISPARAMS)
%%---------------------------------------------------------------------
% INTELLIGENCE PREPARATION OF THE OPERATIONAL ENVIRONMENT (IPOE) MISSION
%   This mission is focused on an undersea system that is sent into an area
%   of interest. Once the system enters the area of interest, it begins
%   the process of intelligence gathering to determine characteristics
%   about the environment. This information would include (but is not
%   limited to) seabed characteristics, acoustics, environment, and aquatic
%   life. This information is sent back to the Command Center for anlaysis
%   to determine how affable the area is for the deployment of a killbox.
%%---------------------------------------------------------------------

%Initialize the result structures.
[USSYS, MISRESULTS] = InitializeMOPs;

%Break out the system parameters from the missionParams structure.
USPARAMS = MISPARAMS.UnderseaSystems;

%If not evaluating in an ISO environment, Mission is only conducted if the
%environment difficulty is lower than the environment difficulty threshold.
if ~MISPARAMS.ISOEnvir && USPARAMS.EnvirThreshold < MISPARAMS.EnvirDifficulty
    return
end

%Iterate through undersea systems to evaluate individual performances.
for us = 1:USPARAMS.numSystems
    %Initialize exit criteria.
    numAttempts = USPARAMS.NumIPOEAttempts;
    ipoeSuccessful = 0;
    ipoeAbort = 0;

    %Initialize the detection counter and flag.
    wasDetected = 0;
    numDetected = 0;

    %The undersea system will conduct IPOE until either it was success,
    %runs out of attempts, or aborts the mission. If mission abort occurs,
    %all undersea systems will abort.
    while numAttempts >0 && ~ipoeSuccessful && ~ipoeAbort
        %The system will only attempt IPOE if it has not been detected and
```

```
        %identified. If it was only detected, the undersea system will
        %perform “detection avoidance” and wait for a set time frame (1 attempt)
        %before reattempting IPOE.
        if ~wasDetected
            %Determine if the IPOE attempt was successful. The IPOE
            %probability is comprised of multiple factors that contribute to
            %the operational activities associated with the mission.
            ipoeSuccessful = binornd(1,USPARAMS.IPOEProbability);

            %IPOE is only aborted if the undersea system is both detected
            %and classified. Otherwise, it will perform “detection
            %avoidance.”
            wasDetected = binornd(1,USPARAMS.DetProbability);
            if wasDetected
                numDetected = numDetected+1;
                wasIdentified = binornd(1,USPARAMS.ClsProbability);
                if wasIdentified
                    ipoeSuccessful = 0;
                    ipoeAbort = 1;
                end
            end
        else
            wasDetected = 0;
        end

        %Reduce the number of attempts.
        numAttempts = numAttempts-1;
    end

    %Determine if the undersea system successfully gathered the required data.
    USSYS(us).ipoeSuccessful = ipoeSuccessful;
    USSYS(us).numDetected = numDetected;
    USSYS(us).attemptsUsed = USPARAMS.NumIPOEAttempts - numAttempts;

    %If the undersea system aborted IPOE due to being detected, determine
    %if the system successfully evaded.
    if ipoeAbort
        USSYS(us).wasIdentified = 1;
        USSYS(us).evasionSuccessful = binornd(1,USPARAMS.EvsProbability);
        break
    end
end

%All systems need to have collected the data for mission success.
MISRESULTS.missionSuccess = all([USSYS.ipoeSuccessful]);

%Find all systems that were identified and the number of detections/attempts.
MISRESULTS.NumDetections = sum([USSYS.numDetected]);
MISRESULTS.NumAttempts = sum([USSYS.attemptsUsed]);
MISRESULTS.numIdentified = sum([USSYS.wasIdentified]);

%Find all systems that successfully evaded (given detection).
```

```
MISRESULTS.numEvaded = sum([USSYS.evasionSuccessful]);

%Assign the undersea sysetm into the mission results for statistical
%analysis.
MISRESULTS.UnderseaSystem = USSYS;
end

%--------------------------------------------------------------------------
function [USRES, MISRES] = InitializeMOPs
%%------------------------------------------------------------------------
% function [USRES, MISRES] = InitializeMOPs
%   This function intializes the return structure of mission results.
%%------------------------------------------------------------------------

USRES = struct(...
    'attemptsUsed',0,...
    'ipoeSuccessful',0,...
    'wasIdentified',0,...
    'evasionSuccessful',0,...
    'numDetected',0 ...
    );

MISRES = struct(...
    'missionSuccess',0,...
    'NumAttempts',0,...
    'NumDetections',0,...
    'numIdentified',0,...
    'numEvaded',0 ...
    );
end
```

2. Mission 2: Intelligence, Reconnaissance, and Surveillance

```
function MISRESULTS = DeployISR(MISPARAMS)
%%------------------------------------------------------------------------
%  INTELLIGENCE, SURVELLIANCE, AND RECONNAISANCE (ISR) MISSION
%   This mission is focused on the deployment of the ISR systems into the
%   Kill box. This mission can only occur after a sucessful IPOE mission.
%   The ISR systems to be deployed will provide information reguarding the
%   status of the Kill box and its enviroment to the command center.
%%------------------------------------------------------------------------

%Deployment times/hiding times.
deploymentTime = 2; %hours
detavoidTime = 4; %hours

%Initialize the result structures.
[USSYS, MISRESULTS] = InitializeMOPs;
isrfieldDetected = 0;
isrfieldAbandoned = 0;
```

```
%Break out the system parameters from the missionParams structure.
USPARAMS = MISPARAMS.UnderseaSystems;
ISRPARAMS = MISPARAMS.ISRSystems;

%If not running in an ISO environment, mission is only conducted if the
%environment difficulty is lower than the environment difficulty threshold.
if ~MISPARAMS.ISOEnvir && USPARAMS.EnvirThreshold < MISPARAMS.EnvirDifficulty
    return
end

%For ISR to succeed, the undersea systems have to deploy a certain amount
%of systems so that the resolution threshold is acheived.
numRequired = ISRPARAMS.numRequired;

if numRequired > USPARAMS.numSystems*ISRPARAMS.numSystems
    return
end

%Set the total deployed (equivalent to the number of ISRs required for
%now).
totalDeployed = 0;

%Iterate through undersea systems to evaluate individual performances. The
%systems will only deploy the minimum number of ISR systems required for
%the field to meet the designated resolution.
for us = 1:USPARAMS.numSystems
    %Initialize the exit criteria
    numSystems = ISRPARAMS.numSystems;
    enduranceLeft = USPARAMS.Endurance;

    fieldDetected = 0;
    isrAbort = 0;

    %Initialize the detection counter and flag.
    wasDetected = 0;
    numDetected = 0;
    assetDeployed = 0;

    %The undersea system will deploy ISR devices until the desired resolution
    % (coverage) is achieved, the entire payload is expended, or aborts
    % mission. If the mission is aborted (i.e, the ISR field is identified
    % by opposing forces, all undersea systems in the tactical area of
    % interest will abort.
    while numRequired ~= 0 && numSystems > 0 && ~isrAbort && enduranceLeft ~= 0
        %The system will only attempt to deploy the ISR devices if it has
        %not been detected and identified. The system will perform
        %"detection avoidance" if detected and wait for a set time frame
        %before reattempting additional deployments.
        if ~wasDetected
            %Determine if the asset deployment was successful.
            depSuccess = binornd(1,USPARAMS.ISRDepProbability);
```

```
            enduranceLeft = max(enduranceLeft - deploymentTime,0);

            %If successful, reduce the number required for ISR success.
            if depSuccess
                numRequired = numRequired-1;

                %Count the number of successes.
                assetDeployed = assetDeployed+1;
                totalDeployed = totalDeployed+1;
            end

            %Reduce the amount of ISR systems within the payload.
            numSystems = numSystems-1;

            %ISR deployment is only aborted if either systems were detected
            %and classified.
            wasDetected = binornd(1,USPARAMS.DetProbability);
            if wasDetected
                numDetected = numDetected+1;
                wasIdentified = binornd(1,USPARAMS.ClsProbability);
                if wasIdentified
                    isrAbort = 1;
                end
            end
        else
            enduranceLeft = max(enduranceLeft - detavoidTime,0);
            wasDetected = 0;
        end

        %Determine if the previously deployed ISR systems have been
        %detected and classified.
        for f = 1:totalDeployed
            devDetected = binornd(1,ISRPARAMS.DetProbability);
            if devDetected
                devIdentified = binornd(1,ISRPARAMS.ClsProbability);
                if devIdentified
                    fieldDetected = 1;
                    isrAbort = 1;
                end
            end
        end
    end

    %If the ISR system was detected and abandoned.
    isrfieldDetected = isrfieldDetected | fieldDetected;
    isrfieldAbandoned = isrfieldAbandoned | isrAbort;

    %Assign in number of detections/deployments.
    USSYS(us).numDetected = numDetected;
    USSYS(us).assetsDeployed = assetDeployed;

    %If the undersea system aborted ISR due to being detected, determine
```

```
    %if the system successfully evaded.
    if isrAbort && wasDetected
        USSYS(us).wasIdentified = 1;
        USSYS(us).evasionSuccessful = binornd(1,USPARAMS.EvsProbability);
        break
    end

    %If the resolution has been obtained, no more devices are deployed.
    if numRequired <= 0
        break
    end
end

%The number of systems required for the resolution have to be deployed and
%the field must not be abandoned.
MISRESULTS.missionSuccess = ~isrfieldAbandoned & (numRequired == 0);

%Find all the systems that were identified and the average number of
%detections/deployments. Add in whether or not the ISR field was detected.
MISRESULTS.NumDetections = sum([USSYS.numDetected]);
MISRESULTS.AssetsDeployed = sum([USSYS.assetsDeployed]);
MISRESULTS.totalDeployed = totalDeployed;
MISRESULTS.fieldDetected = fieldDetected;
MISRESULTS.numIdentified = sum([USSYS.wasIdentified]);

%Find all systems that successfully evaded (given detection)
MISRESULTS.numEvaded = sum([USSYS.evasionSuccessful]);

%Assign the systems into the mission results for statistical
%analysis.
MISRESULTS.underseaSystem = USSYS;
end

%--------------------------------------------------------------------------
function [USRES, MISRES] = InitializeMOPs
%%------------------------------------------------------------------------
% function [USRES, MISRES] = InitializeMOPs
%   This function intializes the return structure of mission results.
%%------------------------------------------------------------------------

USRES = struct(...
    'isrDeployed',0,...
    'wasIdentified',0,...
    'evasionSuccessful',0,...
    'numDetected',0, ...
    'assetsDeployed',0 ...
    );

MISRES = struct(...
    'missionSuccess',0,...
    'AssetsDeployed',0,...
    'NumDetections',0,...
```

```
    'numIdentified',0,...
    'numEvaded',0,...
    'totalDeployed',0,...
    'fieldDetected',0 ...
    );
end
```

3. Mission 3: Effects Field Deployment

```
function MISRESULTS = DeployEF(MISPARAMS)
%%--------------------------------------------------------------------------
% EFFECT FIELD (EF) DELIVERY MISSION
%   This mission is focused on the deployment of the Effects Feild into the
%   Kill box. This mission can only occur after a sucessful ISR mission.
%   The Undersea system will attempt to deploy the effects field, if it is
%   sucsessful in deploying all of the EF then it can be assumed that the
%   Kill box is fully deployed and that the mission was a sucsess.
%%--------------------------------------------------------------------------

%Deployment times/hiding times.
deploymentTime = 10; %hours
detavoidTime = 4; %hours

%Initialize the result structures.
[USSYS, MISRESULTS] = InitializeMOPs;
isrfieldDetected = 0;
effieldDetected = 0;
effieldAbandoned = 0;

%Break out the system parameters from the missionParams structure.
USPARAMS = MISPARAMS.UnderseaSystems;
ISRPARAMS = MISPARAMS.ISRSystems;
EFPARAMS = MISPARAMS.EFSystems;

%if not running in an ISO environment, mission is only conducted if the
%environment difficulty is lower than the environment difficulty threshold.
if ~MISPARAMS.ISOEnvir && USPARAMS.EnvirThreshold < MISPARAMS.EnvirDifficulty
    return
end

%For EF to succeed, the undersea systems have to deploy a certain amount
%of systems so that the resolution threshold is acheived.
numRequired = EFPARAMS.numRequired;

if numRequired > EFPARAMS.numSystems*EFPARAMS.numSystems
    return
end

totalDeployed = 0;
```

```
%Iterate through undersea systems to evaluate individual performances. The
%systems will only deploy the minimum number of EF systems required for the
%field to meet the designated resolution.
for us = 1:USPARAMS.numSystems
    %Initialize the exit criteria
    numSystems = EFPARAMS.numSystems;
    enduranceLeft = USPARAMS.Endurance;

    fieldDetected = 0;
    isrDetected = 0;
    efAbort = 0;

    %Initalize the detection counter and flag.
    wasDetected = 0;
    numDetected = 0;
    assetDeployed = 0;

    %The undersea system will deploy EF devices until the desired resolution
    % (coverage) is achieved, the entire payload is expended, or aborts
    % mission. If the mission is aborted (i.e, the EF field is identified
    % by opposing forces, all undersea systems in the tactical area of
    % interest will abort.
    while numRequired ~= 0 && numSystems > 0 && ~efAbort && enduranceLeft ~= 0
        %The system will only attempt to deploy the EF devices if it has
        %not been detected and identified. The system will perform
        %"detection avoidance" if detected and wait for a set time frame
        %before reattempting additional deployments.
        if ~wasDetected
            %Determine if the asset deployment was successful.
            depSuccess = binornd(1,USPARAMS.EFDepProbability);
            enduranceLeft = max(enduranceLeft-deploymentTime,0);

            %If successful, reduce the number required for the EF success.
            if depSuccess
                numRequired = numRequired-1;

                %Count the number of successes.
                assetDeployed = assetDeployed+1;
                totalDeployed = totalDeployed+1;
            end

            %Reduce the amount of EF systems within the payload.
            numSystems = numSystems-1;

            %EF deployment is only aborted if any systems were detected and
            %classified.
            wasDetected = binornd(1,USPARAMS.DetProbability);
            if wasDetected
                numDetected = numDetected+1;
                wasIdentified = binornd(1,USPARAMS.ClsProbability);
                if wasIdentified
                    efAbort = 1;
```

```
                end
            end
        else
            wasDetected = 0;
            enduranceLeft = max(enduranceLeft-detavoidTime,0);
        end

        %Determine if the previously deployed ISR systems have been
        %detected and classified.
        for f = 1:ISRPARAMS.totalDeployed
            devDetected = binornd(1,ISRPARAMS.DetProbability);
            if devDetected
                devIdentified = binornd(1,ISRPARAMS.ClsProbability);
                if devIdentified
                    isrDetected = 1;
                    efAbort = 1;
                end
            end
        end

        %Determine if the previously deployed EF systems have been detected
        %and classified.
        for f = 1:totalDeployed
            devDetected = binornd(1,EFPARAMS.DetProbability);
            if devDetected
                devIdentified = binornd(1,EFPARAMS.ClsProbability);
                if devIdentified
                    fieldDetected = 1;
                    efAbort = 1;
                end
            end
        end
    end

    %If the EF system was detected and abandoned.
    effieldDetected = effieldDetected | fieldDetected;
    effieldAbandoned = effieldAbandoned | efAbort;

    %See if the ISR was detected as well.
    isrfieldDetected = isrfieldDetected | isrDetected;

    %Assign in number of detections/deployments.
    USSYS(us).numDetected = numDetected;
    USSYS(us).assetsDeployed = assetDeployed;

    %If the undersea system aborted EF due to being detected, determine
    %if the system successfully evaded.
    if efAbort && wasDetected
        USSYS(us).wasIdentified = 1;
        USSYS(us).evasionSuccessful = binornd(1,USPARAMS.EvsProbability);
        break
    end
```

```
    %If the resolution has been obtained, no more devices are deployed.
    if numRequired <= 0
        break
    end
end

%The number of systems required for the resolution have to be deployed and
%the field must not be abandoned.
MISRESULTS.missionSuccess = ~effieldAbandoned & (numRequired == 0);

%Find all the systems that were identified and the average number of
%detections/deployments. Add in whether or not the ISR field was detected.
MISRESULTS.NumDetections = sum([USSYS.numDetected]);
MISRESULTS.AssetsDeployed = sum([USSYS.assetsDeployed]);
MISRESULTS.totalDeployed = totalDeployed;
MISRESULTS.isrDetected = isrfieldDetected;
MISRESULTS.fieldDetected = effieldDetected;
MISRESULTS.numIdentified = sum([USSYS.wasIdentified]);

%Find all systems that successfully evaded (given detection)
MISRESULTS.numEvaded = sum([USSYS.evasionSuccessful]);

%Assign the systems into the mission results for statistical
%analysis.
MISRESULTS.underseaSystem = USSYS;
end

%--------------------------------------------------------------------------
function [USRES, MISRES] = InitializeMOPs
%%------------------------------------------------------------------------
% function [USRES, MISRES] = InitializeMOPs
%   This function intializes the return structure of mission results.
%%------------------------------------------------------------------------

USRES = struct(...
    'efDeployed',0,...
    'wasIdentified',0,...
    'evasionSuccessful',0, ...
    'numDetected',0,...
    'assetsDeployed',0 ...
    );

MISRES = struct(...
    'missionSuccess',0,...
    'AssetsDeployed',0,...
    'NumDetections',0,...
    'numIdentified',0,...
    'numEvaded',0,...
    'totalDeployed',0,...
    'fieldDetected',0,...
    'isrDetected',0 ...
```

```
    );
end
```

4. Mission 4: Engage and Strike

```
function MISRESULTS = EngageAndStrike(MISPARAMS)
%%------------------------------------------------------------------------
% ENGAGE AND STRIKE MISSION
%   The mission assumes that the target of interest is not only within the
%       kill box but that it is within the vicinity? of an effect device
%       and that the command base has issued an order of engagement. By this
%       time, all friendly systems within the area have communicated, the
%       target location has been identified/verified and the command center has
%       issued a strike order for the target. This mission requires the successful
%   coordination of all of the platforms in the kill box, in order to
%   accurately and effectively aim and strike the target. Once the strike
%   has been carried out, ISR will reassess the target to determine its
%   state and communicate with the command center and wait for the next order.?
%
%   Once the target is in the killbox, the ISR system will identify and
%   lock the target for effects targeting. The effects will then engage the
%   target and attempt a “mission kill.”  The ISR system will reassess the
%   the target and determine if additional effects will be necessary to
%   re-engage the target.
%%------------------------------------------------------------------------

%Initialize the result structures.
[USSYS, MISRESULTS] = InitializeMOPs;
killboxAbandoned = 0;
strikeAbort = 0;

%Break out the system parameters from the missionParams structure.
%Break out the system parameters from the missionParams structure.
USPARAMS = MISPARAMS.UnderseaSystems;
ISRPARAMS = MISPARAMS.ISRSystems;
EFPARAMS = MISPARAMS.EFSystems;
TGTPARAMS = MISPARAMS.OpSystems;

%If not running in an ISO environment, mission is only conducted if the
%environment difficulty is lower than the environment difficulty threshold.
if ~MISPARAMS.ISOEnvir && USPARAMS.EnvirThreshold < MISPARAMS.EnvirDifficulty
    return
end

%For the strike to success, the seabed system of systems (SoS) have to
%obtain a mission kill.
missionKill = 0;

%Initialize the exit criteria.
numEffects = EFPARAMS.totalDeployed;
```

```
numEngAttempts = EFPARAMS.totalDeployed;
efIdentified = 0;
isrIdentified = 0;
engAbort = 0;

%Initialize the detection counter and flag.
numTgtTracked = 0;
effectFired = 0;

%Prior to successfully engaging the target, the seabed SoS must detect,
%identify, and track the opposing force. The SoS will continue to
%maintain the track and attempt engagement until either the mission
%kill is obtained, entire payload of effects engaged, or strike is
%aborted.
while ~missionKill && numEffects > 0 && ~engAbort  && numEngAttempts > 0
    tgtIdentified = 0;

    %Determine if the seabed SoS was able to fully identify and track
    %the target of interest.
    detSuccess = binornd(USPARAMS.numSystems,USPARAMS.OpDetProbability) || ...
        binornd(ISRPARAMS.totalDeployed,ISRPARAMS.OpDetProbability);
    clsSuccess = binornd(USPARAMS.numSystems,USPARAMS.OpClsProbability) || ...
        binornd(ISRPARAMS.totalDeployed,ISRPARAMS.OpClsProbability);
    trkSuccess = binornd(USPARAMS.numSystems,USPARAMS.OpTrkProbability) || ...
        binornd(ISRPARAMS.totalDeployed,ISRPARAMS.OpTrkProbability);
    if detSuccess && clsSuccess && trkSuccess
        tgtIdentified = 1;
        numTgtTracked = numTgtTracked+1;
    end

    %For now, assume that the command always calls for the strike.
    strikeOrdered = 1;
    if tgtIdentified && strikeOrdered
        missionKill = binornd(1,EFPARAMS.Lethality);

        %Iterate the counters.
        numEffects = numEffects-1;
        effectFired = effectFired+1;

        %If the mission kill was not obtained, see if the target evaded.
        targetEvaded = 0;
        if ~missionKill
            targetEvaded = binornd(1,TGTPARAMS.EvsProbability);
        end
    end

    %Determine if the any of the systems in the area have been
    %detected.
    usEvaded = 0;
    usIdentified = binornd(USPARAMS.numSystems,TGTPARAMS.DetProbability(1)) && ...
        binornd(USPARAMS.numSystems,TGTPARAMS.ClsProbability(1));
    efIdentified = binornd(EFPARAMS.totalDeployed,TGTPARAMS.DetProbability(3)) && ...
```

```
        binornd(EFPARAMS.totalDeployed,TGTPARAMS.ClsProbability(3));
    isrIdentified = binornd(ISRPARAMS.totalDeployed,TGTPARAMS.DetProbability(2)) && ...
        binornd(ISRPARAMS.totalDeployed,TGTPARAMS.ClsProbability(2));
    if (any(usIdentified) || any(efIdentified) || any(isrIdentified)) && ~missionKill
        engAbort = 1;

        if any(usIdentified)
            usEvaded = binornd(1,USPARAMS.EvsProbability);
        end
    end
end

%The mission kill must have been obtained.
MISRESULTS.missionSuccess = missionKill;

%Find all the systems that were identified and the average number of
%detections/deployments. Add in whether or not the ISR field was detected.
MISRESULTS.numEffectsFired = numEffects;
MISRESULTS.numTgtTracked = numTgtTracked;
MISRESULTS.numIdentified = usIdentified;
MISRESULTS.isrDetected = any(isrIdentified);
MISRESULTS.efDetected = any(efIdentified);
MISRESULTS.usEvaded = usEvaded;
end

%--------------------------------------------------------------------------
function [USRES, MISRES] = InitializeMOPs
%%------------------------------------------------------------------------
% function [USRES, MISRES] = InitializeMOPs
%   This function intializes the return structure of mission results.
%%------------------------------------------------------------------------

USRES = struct(...
    'wasIdentified',0,...
    'evasionSuccessful',0,...
    'numDetected',0 ...
    );

MISRES = struct(...
    'missionSuccess',0,...
    'numEffectsFired',0,...
    'numTgtTracked',0,...
    'numIdentified',0,...
    'isrDetected',0,...
    'efDetected',0,...
    'usEvaded',0 ...
    );
end
```

D. INTERFACE UPDATES

```
function UpdateOverview(handles,MISRES)
%%-------------------------------------------------------------------------
% function UpdateOverview(handles,MISRES)
%   Updates the overview graphics/tables in the Seabed Warfare user
%   interface.
%%-------------------------------------------------------------------------
global numSimulations simCount
persistent simArray resArray overView

%Initialize the simCount, resArray and overView
misFlds = {'IPOE' 'ISR' 'EF' 'ENG'};
numMissions = length(misFlds);
simArray(:,end+1) = nan(numMissions,1);
resArray(:,end+1) = nan(numMissions,1);

if isempty(overView)
    overView = zeros(4,length(misFlds));
end

for mis = 1:length(misFlds)
    if MISRES.(misFlds{mis}).missionSuccess
        simArray(mis,end) = simCount;
        resArray(mis,end) = mis;
        overView(1,mis) = overView(1,mis)+1; %Add to the success count
    elseif ~MISRES.(misFlds{mis}).missionSuccess
        if (isfield(MISRES.(misFlds{mis}),'numEvaded') && MISRES.(misFlds{mis}).numEvaded
~= 0) || ...
                (isfield(MISRES.(misFlds{mis}),'usEvaded') &&
MISRES.(misFlds{mis}).usEvaded)
            overView(2,mis) = overView(2,mis)+1; %Add to the mission failures asset saved
        else
            overView(3,mis) = overView(3,mis)+1; %Add to the mission failures asset lost
        end
    end
end

%Update the success rate for each mission.
overView(end,:) = overView(1,:)./(sum(overView(1:3,:)));
overView(isnan(overView)) = 0;

%Update Simulation Overview.
plot(handles.SimulationOverview,simArray,resArray,'Color','k','Marker','s','MarkerSize',3
,'LineStyle','none');
set(handles.SimulationOverview,'YTick',1:4);
set(handles.SimulationOverview,'YTickLabel',{'1' '2' '3' '4'});
set(handles.SimulationOverview,'YLim',[0 5]);
set(handles.SimulationOverview,'XLim',[1 numSimulations]);
set(handles.SimulationOverview,'XTick',round(linspace(1,numSimulations,4)));
set(handles.SimulationOverview,'XTickLabel',round(linspace(1,numSimulations,4)));
```

```
%Update the Simulation Histogram.
[m,n] = size(resArray);
unbatchedArray = reshape(resArray,1,m*n);
histogram(handles.SimulationHist,unbatchedArray,'Orientation','horizontal');
set(handles.SimulationHist,'YTick',1:4);
set(handles.SimulationHist,'YTickLabel',{'1' '2' '3' '4'});
set(handles.SimulationHist,'YLim',[0 5]);
set(handles.SimulationHist,'XLim',[1 numSimulations]);
set(handles.SimulationHist,'XTick',round(linspace(1,numSimulations,4)));
set(handles.SimulationHist,'XTickLabel',round(linspace(1,numSimulations,4)));

%Update the Simulation Overview Table.
set(handles.SimulationTable,'Data',overView);

if numSimulations == simCount
    overView = zeros(4,length(misFlds));
    simArray = [];
    resArray = [];
end
end
```

E. EXPORT RESULTS

```
function WriteSimResults(scenarioFile,MISRES,MISPARAMS)
%%-------------------------------------------------------------------------
% function WriteSimResults(MISRES,MISPARAMS)
%   Writes simulation setup, criteria, and results to output CSV for later
%   use.
%%-------------------------------------------------------------------------
global simCount misNum numMissions numSimulations scenRaw
persistent simFile scenFull overMat

if isempty(scenFull)
    [filePath,fileName,fileExt] = fileparts(scenarioFile);
    scenFull = [filePath filesep fileName '_Ran' fileExt];
    copyfile(scenarioFile,scenFull,'f');
end

%Write mission overview results.
overView = struct('IPOEMissionSuccess',0,...
        'NumberAttemptsUsedIPOE',0,...
        'NumberDetectionsIPOE',0,...
        'NumberIdentifiedIPOE',0,...
        'ISRMissionSuccess',0,...
        'NumberDeploymentsISR',0,...
        'NumberDetectionsISR',0,...
        'NumberIdentifiedISR',0,...
        'EFMissionSuccess',0,...
        'NumberDeploymentsEF',0,...
```

```
        'NumberDetectionsEF',0,...
        'NumberIdentifiedEF',0,...
        'KillBoxDeployment',0,...
        'StrikeSuccess',0,...
        'TimesTargetTracked',0,...
        'NumberIdentifiedStrike',0,...
        'KillBoxExecution',0);
for simCount = 1:numSimulations
    overView.IPOEMissionSuccess =
overView.IPOEMissionSuccess+MISRES(simCount).IPOE.missionSuccess;
    overView.NumberAttemptsUsedIPOE =
overView.NumberAttemptsUsedIPOE+MISRES(simCount).IPOE.NumAttempts;
    overView.NumberDetectionsIPOE =
overView.NumberDetectionsIPOE+MISRES(simCount).IPOE.NumDetections;
    overView.NumberIdentifiedIPOE =
overView.NumberIdentifiedIPOE+MISRES(simCount).IPOE.numIdentified;
    overView.ISRMissionSuccess =
overView.ISRMissionSuccess+MISRES(simCount).ISR.missionSuccess;
    overView.NumberDeploymentsISR =
overView.NumberDeploymentsISR+MISRES(simCount).ISR.totalDeployed;
    overView.NumberDetectionsISR =
overView.NumberDetectionsISR+MISRES(simCount).ISR.NumDetections;
    overView.NumberIdentifiedISR =
overView.NumberIdentifiedISR+MISRES(simCount).ISR.numIdentified;
    overView.EFMissionSuccess =
overView.EFMissionSuccess+MISRES(simCount).EF.missionSuccess;
    overView.NumberDeploymentsEF =
overView.NumberDeploymentsEF+MISRES(simCount).EF.totalDeployed;
    overView.NumberDetectionsEF =
overView.NumberDetectionsEF+MISRES(simCount).EF.NumDetections;
    overView.NumberIdentifiedEF =
overView.NumberIdentifiedEF+MISRES(simCount).EF.numIdentified;
    overView.StrikeSuccess = overView.StrikeSuccess+MISRES(simCount).ENG.missionSuccess;
    overView.TimesTargetTracked =
overView.TimesTargetTracked+MISRES(simCount).ENG.numTgtTracked;
    overView.NumberIdentifiedStrike =
overView.NumberIdentifiedStrike+MISRES(simCount).ENG.numIdentified;
    overView.KillBoxExecution =
overView.KillBoxExecution+MISRES(simCount).KillBoxExecution;
end

overView = structfun(@(x){(x/numSimulations)},overView,'UniformOutput',false);
overView.KillBoxDeployment{1} = overView.IPOEMissionSuccess{1}*...
    overView.ISRMissionSuccess{1}*...
    overView.EFMissionSuccess{1};
overMat = vertcat(overMat,struct2array(overView));

if misNum == numMissions
    for i = 1:size(overMat,1)
        for j = 1:size(overMat,2)
            scenRaw{i+1,37+j} = overMat{i,j};
        end
```

```
    end
    xlswrite(scenFull,scenRaw);
end
end
```

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Army, Marine Corps, Navy, Air Force. 2005. Kill Box: Multi-Service Tactics, Techniques, and Procedures for Kill Box Employment. FM3-09.34 Air Sea Land Application Center. https://info.publicintelligence.net/fm3_09x34.pdf.

Chief of Naval Operations. 2016. Autonomous Undersea Vehicle Requirement for 2025. 2016. Washington, DC: Undersea Warfare Directorate. https://news.usni.org/wp-content/uploads/2016/03/18Feb16-Report-to-Congress-Autonomous-Undersea-Vehicle-Requirement-for-2025.pdf

Clark, Bryan. 2016. "Undersea cables and the future of submarine competition." Bulletin of the Atomic Scientists 72, no. 4, 234–237, http://dx.doi.org/10.1080/00963402.2016.1195636

Department of Defense. n.d. "The DoDAF Architecture Framework Version 2.02." Accessed January 15, 2018. http://dodcio.defense.gov/Library/DoD-Architecture-Framework/.

Everhart, Dave. 2017. "Seabed Warfare Modules." Presentation at Naval Undersea Warfare Center, Newport, RI, September 25, 2017.

General Dynamics Mission Systems, 2018. "Undersea Distributed Network (UDN)." Accessed October 20, 2017. https://gdmissionsystems.com/intelligence-systems/signals-intelligence/undersea-distributed-network

Keller, John. 2017a. "Boeing and Lockheed Martin to build prototype extra-large UUVs for long-endurance military missions." October 2, 2017. http://www.militaryaerospace.com/articles/2017/10/uuvs-long-endurance-extra-large.html.

———. 2017b. "DARPA asks industry for large unmanned undersea vehicle advanced payload delivery system." March 28 2017. http://www.militaryaerospace.com/articles/2017/03/payload-deliver-unmanned-undersea.html.

———. 2013. "DARPA releases BAA for unmanned submersible mothership able to launch UAVs and UUVs." August 27 2013. http://www.militaryaerospace.com/articles/2013/08/darpa-hydra-solicitation.html.

Lin, Jeffrey, and P.W Singer. 2016. "The Great Underwater Wall of Robots: Chinese Exhibit Shows off Sea Drones." Popular Science, June 22 2016. https://www.popsci.com/great-underwater-wall-robots-chinese-exhibit-shows-off-sea-drones.

OPNAV N97. 2017. “CNO UUV Day: Family of UUVs.” Presentation at the Naval Undersea Warfare Center, Newport, RI, September 25 2017.

SAS Institute, Inc. 2018. “Design of Experiments Guide: Space-Filling Designs.” Last modified April 13, 2018. https://www.jmp.com/support/help/14/space-filling-designs.shtml.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California