2020-06

DESIGN OF EXPERIMENTS FOR AIR LAUNCHED EFFECTS UNMANNED AERIAL VEHICLES

Fabijanowicz, Jason R.

Monterey, CA; Naval Postgraduate School

https://hdl.handle.net/10945/65513

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

DESIGN OF EXPERIMENTS FOR AIR LAUNCHED EFFECTS UNMANNED AERIAL VEHICLES

by

Jason R. Fabijanowicz

June 2020

Thesis Advisor: Jeffrey A. Appleget
Second Reader: Thomas W. Lucas

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

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

1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** June 2020	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE DESIGN OF EXPERIMENTS FOR AIR LAUNCHED EFFECTS UNMANNED AERIAL VEHICLES	**5. FUNDING NUMBERS**
6. AUTHOR(S) Jason R. Fabijanowicz	
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)

America needs to rapidly inject advanced technology into its Armed Forces to put pressure on near-peer adversaries in great power competition. In one such effort, the United States Army is working to quickly and efficiently develop a system involving two vehicles, the Future Attack Reconnaissance Aircraft with Air-Launched Effects. This family of vehicles will be critical to establishing dominance in multi-domain operations on the battlefield. These vehicles are in a rapid prototyping initiative with no established Analysis of Alternatives. In order for the program to stay on an ambitious deliverable timeline, a set of low-detailed simulations for these vehicles in the Joint Dynamic Allocation of Fires and Sensors modeling environment are created with a design of experiments to provide narrowed criteria and required performance characteristics for the two vehicles. This data will feed a higher-fidelity model in the Advanced Warfighting Simulation (AWARS) environment. This research will reduce the delivery time of the project by enabling AWARS modelers to focus on a smaller subset of characteristic attributes. This thesis provides the Research and Analysis Center with a Pareto optimal frontier for the vehicles' characteristics, explores characteristic tradeoffs, and determines measures of effectiveness and measures of performance for policymakers and program managers.

14. SUBJECT TERMS FARA, DOE, ALE, AWARS, JDAFS, DAFS	**15. NUMBER OF PAGES** 101
	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

DESIGN OF EXPERIMENTS FOR AIR LAUNCHED EFFECTS UNMANNED AERIAL VEHICLES

Jason R. Fabijanowicz
Major, United States Army
BS, U.S. Military Academy, 2008
MS, Missouri University of Science and Technology, 2013

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
June 2020

Approved by: Jeffrey A. Appleget
Advisor

Thomas W. Lucas
Second Reader

W. Matthew Carlyle
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

America needs to rapidly inject advanced technology into its Armed Forces to put pressure on near-peer adversaries in great power competition. In one such effort, the United States Army is working to quickly and efficiently develop a system involving two vehicles, the Future Attack Reconnaissance Aircraft with Air-Launched Effects. This family of vehicles will be critical to establishing dominance in multi-domain operations on the battlefield. These vehicles are in a rapid prototyping initiative with no established Analysis of Alternatives. In order for the program to stay on an ambitious deliverable timeline, a set of low-detailed simulations for these vehicles in the Joint Dynamic Allocation of Fires and Sensors modeling environment are created with a design of experiments to provide narrowed criteria and required performance characteristics for the two vehicles. This data will feed a higher-fidelity model in the Advanced Warfighting Simulation (AWARS) environment. This research will reduce the delivery time of the project by enabling AWARS modelers to focus on a smaller subset of characteristic attributes. This thesis provides the Research and Analysis Center with a Pareto optimal frontier for the vehicles' characteristics, explores characteristic tradeoffs, and determines measures of effectiveness and measures of performance for policymakers and program managers.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

ADA	air defense artillery
AoA	analysis of alternatives
AWARS	Advanced Warfighting Simulation
ALE	Air Launched Effects
DAFS	Dynamic Allocation of Fires and Sensors
DOE	design of experiments
FARA	Future Attack Reconnaissance Aircraft
IADS	integrated air defense system
JDAFS	Joint Dynamic Allocation of Fires and Sensors
MOE	measurement of effectiveness
NOLH	nearly orthogonal Latin hypercube
TRAC	The Research and Analysis Center
XML	Extensible Markup Language

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

Within the next few years, the United States Army will have a capability gap with ensuring mission success in a wide array of conflicts. The Army may not have the critical ability against near-peer advisories to disrupt or destroy opposition air defense assets in order to provide air corridors deep behind enemy lines. To reduce this capabilities gap, the Army is in the process of developing an autonomous family of aerial vehicles charged with the mission of ensuring friendly force survival through the course of an engagement.

This family of vehicles is called the Future Attack Reconnaissance Aircraft ecosystem. These vehicles are designed to travel close to the forward line of troops and deploy smaller unmanned aerial vehicles. These smaller vehicles are called Air Launched Effects (ALE) and they are being designed to disrupt or destroy enemy vehicles, radars, and launched systems. However, due to the experimental nature of the vehicles and their unproven capability in battle, it is difficult to tell what critical capabilities should be focused on for in-depth analysis and high-resolution modeling. Without an alternative against which the ALEs can be compared, the range of values for characteristics such as speed, lethal range, and radar ranges are limitless. This requires a narrow band to alleviate finite critical resources on classified systems.

To provide recommendations for ALE characteristics and formation configurations, a low-resolution model is required for use as a rapid method of modeling to feed higher-resolution models. In this study, the Joint Dynamic Allocation of Fires and Sensors (JDAFS) model was used. This model was used because it has been validated by the Research and Analysis Center–Monterey and can provide extremely fast results with relatively low manpower compared to higher-resolution models. The purpose of this is to provide initial recommendations to the sponsor about optimal trait mixes for the ALE and suggest further areas for study in higher-resolution models. The results from this study will directly feed the Advanced Warfighting Simulation and COMBATXXI models. These systems require much more resources than JDAFS. From there, the Army will have a picture of what is required for design specifications in the contracting phase.

The methodology required to provide these observations and recommendations required research behind the proper formations and implementation of weapons systems and the capabilities of systems. Then, the characteristics capable of being modeled in JDAFS were used to create a design of experiments. This design provided a range of values for the study, then provided an optimal mix of these configurations to determine what was important to the model. Thousands of configurations were turned into scripts to run through simulations to determine what is important to ALE survival and enemy asset destruction. A snapshot of this methodology is provided in Figure ES-1.

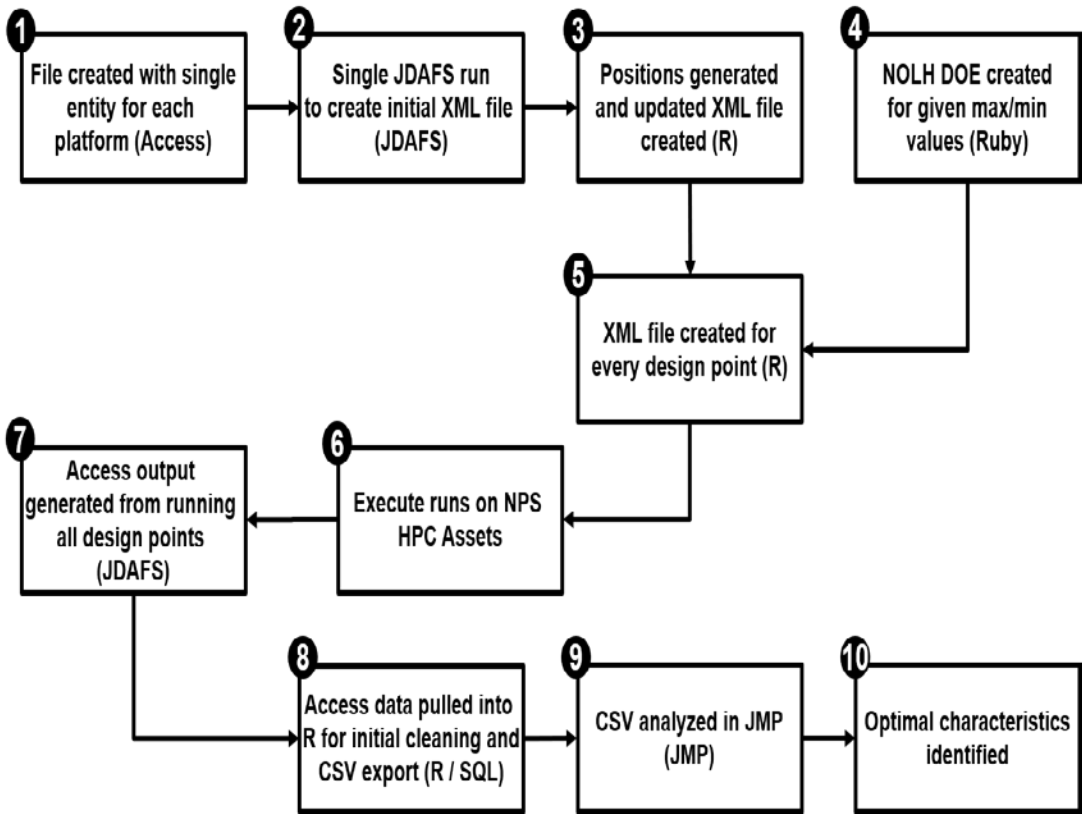


Figure ES-1. Methodology Flowchart

The results from 44,870 simulated battles were used to analyze the performance of the ALE. Two important metrics were created to assess ALE characteristics and configurations. These metrics are ALE survivability and ALE lethality. A linear model was developed for each one of these metrics, or measurements of effectiveness (MOE), to determine variable importance and interactions between predictor variables. Next, random forests were also used to predict these MOEs and provide recommendations based on the splits in the model.

When each design point was aggregated across its number of replications, the linear models provided decent fits for the MOE for survivability (with an R-square value of .791) as well as for the MOE of lethality (with an R-square value of .737). For survivability, the flight configuration of the ALEs and their speed were the most important variables. For lethality, the configuration and speed were important, in addition to the number of rounds each ALE had.

When creating random forests, the cross-validation error for survivability was reduced to .193 and for lethality this value was reduced to .221. Random forests did a much better job of creating a predictive model for this study. The important variables and their importance are in Table ES-1, as well as the best possible configuration in Table ES-2.

Table ES-1. Random Forest Variable Importance

MOE1 Variable	**Variable Importance**	**MOE2 Variable**	**Variable Importance**
# Disrupt ALE	8.1744	Configuration	1.4919
# Kill ALE	4.2467	# Rounds	.8122
Configuration	4.2467	# Disrupt ALE	.7584
# Rounds	2.9141	# Kill ALE	. 7584
Speed	.5340	Speed	.41047

Table ES-2. Optimal Characteristics

Characteristic	**Optimal Value**
# Kill ALE	9
# Disrupt ALE	9
Speed	150knots
Pk(near)	.9
Pk(far)	.16
Radar Distance	7.8 miles
Kill Distance	14.9 miles
Burst Radius	64.136 meters
# Rounds	5
Mean Time to Detection	1.281 seconds
Configuration	5

Based on which variables were important to the model, this study found areas for future research. Most importantly, higher speeds were critical for ALE success, as well as having at least three rounds on each ALE. Areas highlighted in darker green in Figure ES-2 show these areas that should be utilized in future higher-resolution models to directly support the final development stages of the ALE and the FARA ecosystem. With less than 12% of ALEs surviving on average from any replication, the vehicles may be best utilized as a disposable asset. In addition, mixes of kill and disrupt ALEs provide better results than simulations using only one type of variant.

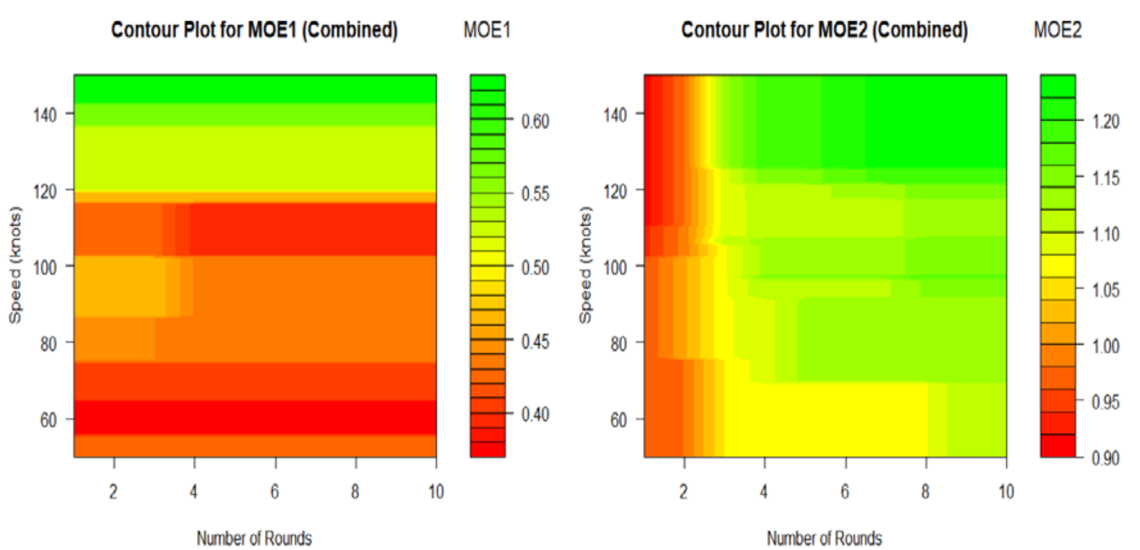


Figure ES-2. R Studio Screenshot of Contour Plot for Summarized MOE Metrics

ACKNOWLEDGMENTS

I would first like to thank MAJ Charles Timm and LCDR Brendon Smeresky for their help and support for the initial phases of this project through its conclusion. Their determination was incredibly inspiring. I would also like to thank Dr. Arnold Buss for all his efforts in keeping JDAFS alive and keeping me sane through the learning process. Additionally, LTC Brian Wade provided fantastic direction and insights into delivering what the sponsor wanted.

My heartfelt appreciation goes out to Dr. Jeffrey Appleget and Dr. Thomas Lucas for their guidance and motivation in guiding me through the dark corridors of this thesis journey. They helped me see the light at the end of the tunnel. I would be remiss if I did not thank the Operations Research teachers as well. The outstanding members of this faculty are experts in education and inspiration.

Finally, I want to thank my parents, my wife, and my dog. I love you all more than the Central Limit Theorem.

THIS PAGE INTENTIONALLY LEFT BLANK

THESIS DISCLAIMER

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

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

This thesis uses simulation and efficient design of experiments (DOE) to determine the optimal design characteristics and support an initial analysis of alternatives for a future aerial reconnaissance and attack group of vehicles called the Future Attack Reconnaissance Aircraft (FARA) ecosystem. The FARA itself is a helicopter-type vehicle that would bridge a current capability gap within the Army and provide targeting capabilities alongside reconnaissance technology in delivering lethal and non-lethal effects (Tate 2020). Two concept photos are provided in Figure 1.

Source:https://www.army.mil/article/234002/future_vertical_lift_army_selects_future_attack_reconnaissance_aircraft_prototype_performers.

Figure 1. FARA Concept Photos

The FARA cannot accomplish this alone. The FARA is currently designed to deploy smaller vehicles to bridge the gap between the enemy front and air defense assets deep behind enemy lines. This smaller vehicle is called air launched effects (ALE). Each FARA is being designed to hold multiple ALEs, capable of relaying information to friendly forces and disrupting adversarial assets on the battlefield. But, until this study, the range of capabilities for these ALEs have been identified, but the ideal values not yet found.

The program used to determine the optimal characteristics for the FARA ecosystem is the Joint Dynamic Allocation of Fires and Sensors (JDAFS) model. This simulation, developed in 2002 at the Naval Postgraduate School, is a highly capable low-fidelity model approved for use by The Research and Analysis Center - Monterey (TRAC-Monterey) within the scope of identifying the best possible vehicle characteristics (Havens 2002). Creating an optimal vehicle design or a Pareto optimal frontier of characteristics through modeling in a low-resolution simulation enables future American operations across a spectrum of multi-domain operations.

A. BACKGROUND

The United States Army has a need for a near-term capability to successfully conduct multi-domain operations. As an accredited model accepted by TRAC-Monterey, the low-resolution JDAFS model is the first step into determining what parameters should be used in the Advanced Warfighting Simulation (AWARS) scenario development. JDAFS is a low-resolution model that can be implemented to determine initial results and provide data exploration, saving time on classified systems and reducing the sample space required on a more computationally restricted cluster of computers. Using an unclassified generic enemy threat scenario, varying elements relevant to the FARA ecosystem in JDAFS determines what characteristics are the most important contributing factors to mission success. This is critically important considering the nature of an experimental vehicle. Without models such as JDAFS or AWARS, it would be difficult to determine the most critical aspects of the FARA vehicles and their optimal characteristics. The necessary condition for success is a FARA ecosystem that overcomes enemy air defense artillery (ADA) and integrated air defense systems (IADS).

B. THESIS OBJECTIVE

This research creates and explores a basic unclassified friendly formation and a generic enemy threat scenario as the backdrop for the experiment within JDAFS. Utilizing a sponsor-approved range of values for vehicle characteristics, a data farming approach using the Naval Postgraduate School's Hamming supercomputer provides the ability to conduct output analysis and determine optimal characteristics. This generated output also provides the opportunity for tradeoff analysis, giving the sponsor and other decision makers the opportunity to determine how having a vehicle that is not capable of the optimal design characteristics may affect mission success.

The purpose of this study is to provide guidance on the optimal design mix for the Air-Launched Effects (ALE) portion of the FARA ecosystem. The study determines this by varying as many system characteristics as possible within JDAFS. The DOE is created from an efficient nearly orthogonal Latin hypercube (NOLH) to provide the greatest possible fidelity in the results. Output analysis then provides the tools and equations necessary to provide informed input to higher-resolution simulations.

C. RESEARCH QUESTIONS

This research provides insights into the following questions necessary to determine how ALEs can best enable ground forces and follow-on aerial assets:

- What ALE characteristics provide the greatest benefit to enable friendly force survival?
- What ALE characteristics enable the suppression of enemy air defenses?
- What is the tradeoff of friendly force survivability if certain ALE characteristics cannot be achieved?
- What is the narrow band of characteristics that should be explored in higher resolution models?
- Should the ALE platform have long-term survivability or be designed to be disposable?
- Are some ALE characteristics unimportant to the success of the platform environment and ground forces?

- To where should additional time and resources be allocated during in-depth characteristic studies in modern models and simulations?
- Are certain vehicle characteristics dependent upon each other for success?

D. THESIS SCOPE

The scope of this thesis is as follows:

1. Identify the vehicles, formations, and characteristics necessary to create a model for simulations within JDAFS.
2. Create a DOE for ALE characteristics to be simulated within JDAFS.
3. Implement the DOE and scenarios within the JDAFS environment.
4. Conduct analysis with simulation output data to determine the optimal ALE characteristics for the scenario to support higher level modeling and analysis.
5. Conduct measurement of performance modeling to support characteristic tradeoff analysis.

E. THESIS FLOW

The format of this thesis is as follows:

- Chapter II provides background to the FARA ecosystem, background for creating the DOE, a review of prior JDAFS and Dynamic Allocation of Fires (DAFS) use cases, and scenario development.
- Chapter III covers the methodology for data farming, model creation, data-output processing, and tradeoff analysis.
- Chapter IV provides the results from the model.
- Chapter V contains a summary of the study, conclusions, and recommendations for follow-on research.

II. BACKGROUND

This chapter provides background on portions of the FARA ecosystem and the basis for the NOLH design used to determine characteristics used within the simulation.

A. DESIGN OF EXPERIMENTS ANALYSIS SUPPORT

In mid-2019, TRAC – Monterey was tasked with providing support to TRAC – Fort Leavenworth on an experimental aerial vehicle that would enable ground forces as well as follow-on air assets. TRAC – Fort Leavenworth is currently supporting the Future Vertical Lift Cross Functional Team's modernization priority to design, build, and test FARA vehicles to fill an existing capability gap with the retirement of the OH-58 Kiowa helicopter. The goal is to have a multi-purpose vehicle (either manned or unmanned) that could deploy smaller vehicles to enable communications to long-range precision fires while simultaneously destroying or disrupting enemy efforts. As the vehicle is experimental, and no realistic alternatives exist, the state-space for characteristics on vehicles within the FARA ecosystem is limitless. The FARA ecosystem encompasses multiple vehicles and configurations, but for the purposes of this thesis, and in support of TRAC – Monterey's objectives, the FARA and the ALE assets are the basis of the study.

1. Design of Experiments Characteristic Development

Before any portion of JDAFS was fit for modeling, it was first important to determine what characteristics are most important and the range values for those characteristics. Since JDAFS is a low-resolution model, it is not capable of addressing every characteristic critical to an aviation asset. However, it does create an environment capable of closely examining a few specifications. In conjunction with both TRAC – Monterey and TRAC – Fort Leavenworth, a list of possible attributes capable of implementation within the model were proposed. Table 1 provides the list of characteristics modeled within this study as well as their abbreviated notation for use within this thesis.

Table 1. Study Characteristics

Abbreviation	Characteristic and Unit of Measure
Speed	Speed of the vehicle (knots)
Pk(near)	Probability of kill at nearest range (miles)
Pk(far)	Probability of kill at farthest range (miles)
Rad D	Radar detection range (miles)
Kill D	Kill distance (miles)
B Rad	Burst radius (meters)
Rounds	Armament round count
Mean Det	Mean time to detection (seconds)

The second step was to create a range of values for each of these characteristics that is generic enough for unclassified distribution but also realistic enough to properly address concerns with the model. Table 2 provides the ranges of data used for each of the characteristics listed in Table 1, as well as the number of significant digits used after the decimal point.

Table 2. Characteristic Value Thresholds

Characteristic	Lower Value	Upper Value	Decimals
Speed	50 knots	150 knots	1
Pk(near)	.75	.95	2
Pk(far)	.1	.4	2
Rad D	2 miles	11 miles	1
Kill D	12 miles	20 miles	1
B Rad	3.31 feet	165.3 feet	1
Rounds	1	10	0
Mean Det	1 second	10 seconds	8

2. Additional Design Consideration

While the characteristic's threshold values are of critical importance to the model, there is another consideration that is important to decision makers and to those analysts that will use the results of this study to inform higher-level modeling in AWARS. This major consideration involves the different types of vehicles that TRAC – Fort Leavenworth wanted to study. With the ALE, TRAC – Fort Leavenworth wanted to study a vehicle capable of killing enemy SEAD assets as well as disrupting enemy SEAD assets. These are the kill and disrupt variants of the platform. For this thesis, characteristics needed to be varied in order to study what is most important as well as different configurations and mixes of the two types of vehicles. This added another dimension to the DOE, but one that required more of a systematic approach in evaluating what is possible on a battlefield and what may be feasible in exploring the state-space. Table 3 describes the seven configurations of ALE explored in this study.

Table 3. ALE Configurations

Configuration Number	Percentage of Kill ALEs	Percentage of Disrupt ALEs
1	100%	0%
2	0%	100%
3	50% (every other)	50% (every other)
4	50% (all on flanks)	50% (all on inside)
5	50% (all on inside)	50% (all on flanks)
6	66.6%	33.3%
7	33.3%	66.6%

3. Design of Experiments Limitations in JDAFS

Due to some of the limitations in JDAFS, a common-sense approach to determining the characteristics in Table 2 had to be adapted. For example, model entities, particularly ones that move, have a difficult, if not impossible, time functioning properly if their radar detection distance is greater than their kill distance. Applying this to a modern combat example, a bullet from a rifle or a tank round can very well go much further than the eye or a radar can detect. While this provides constraint on the range of certain parameters and design points within the DOE, it still passes common weapon conventions. Thus, in determining characteristics, for the model to perform properly, the kill and radar distances had to be layered in such a way that would not break the simulation.

A second limitation that needed to be addressed before a proper DOE could be implemented was the difference in units of measure. The characteristics within the model required different units between distances. These units are also represented differently within JDAFS. Metrics were converted from the JDAFS proprietary units of measure in order to help inform the DOE and further confirm that these characteristics are possible to study. These conversions are provided in Table 4.

Table 4. JDAFS Unit Conversion Table

Standard Unit of Measure	JDAFS Unit of Measure
1 knot	30.63212 speed units
1 mile per hour	26.61857 speed units
1 mile	1597.114 distance units
1 foot	.30248 distance units

A third limitation is the inability of the model to handle an effective modeling situation for a vehicle launching smaller vehicles and subsequently providing mission support. Originally, TRAC – Fort Leavenworth wanted a model where ALEs would be able to move forward and relay positional data back to their parent FARA. Then, each FARA could provide a target handoff to long-range precision fires. The intent for this would be to keep critical assets safe while simultaneously allowing them to fire on targets. However, JDAFS does not allow for multi-stage target passing. The ALEs can be modeled to pass off target data, but can only do it once. A system of relays is not possible within the model. Thus, the study evolved into a version looking almost exclusively at the capabilities of the ALEs.

This was not entirely unrealistic. Thoughts behind this scoping focus was that FARA vehicles would maintain a safe distance way from enemy air defenses anyway. Thus, modeling them into the system was just another thing the enemy could shoot down in the simulation, but they would not be able to do as easily in real life. As a feasible alternative, the ALEs were given a longer radar and detections distance, which also enables them to communicate further with units in the rear.

Within the JDAFS graphical user interface, the ALEs can be grouped together and then branch out to accomplish their mission after a certain line of departure. This has a similar effect to being launched from a FARA, as a single round impacting any one of the

ALEs would have a burst radius large enough to destroy all the ALEs housed within that vehicle. Once these terms were accepted by TRAC, the design space became the next important step in generating the model design.

B. NEARLY ORTHOGONAL LATIN HYPERCUBES

With the characteristics, units, and conversions established, an intelligent method was required to sample the nine-dimensional states-space to provide an efficient method of identifying an optimal configuration for the ALE. A quick and efficient way of doing this was in creating a design based on NOLHs.

1. NOLH Background

Former U.S. Army Colonel Thomas M. Cioppa and Dr. Thomas W. Lucas explored the possibilities of efficient high-dimensional DOEs with their research on NOLHs (Cioppa et al. 2007). The basic idea of their work explores how to create random Latin hypercubes with the best possible orthogonality and space-filling properties. In their research, they developed an algorithm for creating NOLHs that provides minimal correlation between columns in the design matrix, which are factors within this study. These designs also sample as uniformly as possible in the design space and provide broad analytic flexibility, allowing analysts to fit a diverse set of metamodels on multiple model outputs. These designs sample the state-space extremely well and, considering the number of characteristics attributed to the design of the ALEs, the design is a good initial exploration for the range of possible values. In addition, this method is quick in terms of its implementation within the model and can be utilized on characteristics with and without decimal values.

2. NOLH Implementation (Sanchez)

A NOLH design was implemented on the first eight characteristics. Then, the same NOLH was used on each of the seven ALE configurations. To do this efficiently, the Ruby programming language (Ruby 2019) was accessed through the command line to create the design. This programming language uses an efficient syntax to generate requested output. The exact DOE was created using a Ruby Gem that was built by Dr. Paul J. Sanchez

(Sanchez 2018). This Ruby Gem provides a text file of maximum and minimum values with the requisite number of decimal places in a NOLH. The Ruby script, called *stack_nolhs.rb* is a component of the *Datafarming Ruby Gem*. This text file can then be directly used to vary parameter values within the experiment. 641 design points were created for each ALE configuration. 65 levels were used for each characteristic. This means that 65 evenly distributed values were chosen between the maximum and minimum values for each ALE characteristic, with decimal places also strategically assigned. Certain values did not lend themselves to non-whole numbers (such as number of rounds) and others had a minimum and maximum value so high that it did not provide for a very large increase in information for the increase data generated by longer strings of numbers. The design was also stacked ten times, providing rotation in the design-space and further decreasing correlation while increasing the degrees of freedom. This number was also chosen strategically, because of memory allocation issues with replication.

The final number that is critical to the design is the number of repetitions required for each design point. Initially, 30 repetitions for each design point was the target value. However, this was not possible for two major reasons. The first is that the Java Virtual Machine for the JDAFS program had memory allocation issues with storing so much information before being able to write out to a file. Simulation runs with greater than ten repetitions caused the simulation to freeze, even on cluster computing. The second reason dealt with the ability to download the information from NPS' Hamming supercomputer. Due to the state of the nation and required quarantine in March through April 2020, output data must be downloaded through a virtual private network. The gigabytes of data took approximately eight days to download and increasing the amount of data through repetition was not a linear increase in the time required. Even though using rotational stacking is not the same as increasing repetitions, it is an additional method of sampling the state-space. Doing additional runs of the same design points was also not possible since it is impossible to set the random number seed outside of the JDAFS graphical user interface, and running the design point again ran the risk of generating the same output instead of providing new data. In summary, a total of 44,870 simulated battles were run (641 design points × 7 configurations × 10 replications). Without an efficient design, a full factorial design would have resulted in over 3.16×10^{15} replications. Any reduction in the amount of levels within this replication (65 levels) would have affected the precision of the model. On a single personal computer, this would have resulted in a run time of approximately 1.2 billion centuries.

3. JDAFS and DAFS Background

The nexus for JDAFS began with the development and publication of DAFS in 2002. The purpose of then Lieutenant Michael Haven’s study was to create a low-resolution model that could export Extensible Markup Language (XML) files from a discrete event simulation to enable higher-fidelity models and Army systems (Havens 2002). The real benefit of DAFS is its ability to dynamically use linear programming to solve engagement optimizations and prioritize fires across platforms.

In an effort to improve this system’s modeling capabilities, DAFS was upgraded to JDAFS and validation in this improved model was demonstrated in work conducted by Jeffrey T. Freye (2007). This work generated further improvements to the communication abilities of the model and a proof of concept through use as a way of data farming information through a DOE and a NOLH to determine measures of effectiveness and provide tradeoff analysis for surveillance assets. Data farming is a method of finding insights into data, particularly in areas where very little or no data exists at all. Additional information on utilizing this method first in research done by Lieutenant Ben L. Anderson (2011).

This study builds upon the amazing work done before it to establish JDAFS as a capable model with a reputation credible enough to provide real-world support to higher-fidelity models. Being a low-resolution model does not undermine its critical input into other modeling systems, such as AWARS or COMBATXXI. More information can be found on AWARS (2020a) and more information on COMBATXXI (2020b) can be found on the TRAC website. The results generated from JDAFS helps narrow the range of design points necessary in future studies. The system also provides quick results. While many combat simulations require a team of personnel working over the course of months, the whole JDAFS system can be taught, implemented, and results analyzed by one person in less than four months. The ability to provide such rapid analysis helps cut down on required personnel, time, and funding—as well as to speed up the decision cycle.

Finally, with efforts being conducted elsewhere across the Army to create scenarios and determine what factors are most important, JDAFS was used in such a rapid and effective manner that the results of this study can be used to support planning and implementation of these scenarios in higher-resolution simulations. With computational times reduced on classified systems, there is potential that Soldiers on the ground could see the final product sooner than later.

4. Scenario Development

The last big research element came from friendly allocation of forces documents and enemy threat templates. The first step was to determine which kinds of fighting and sensing force entities were necessary in the model. To find the number and types of equipment within a combat unit, data was pulled from the Army database for unit allocations located on FMSWeb (2020). Critical equipment types and quantities were annotated from this database for a friendly armored regiment. U.S. Army Acquisition Support Center published a guide in 2018 with unclassified system characteristics (U.S. Army Acquisition Support Center 2018).

Unclassified threat documents were found on the Army Training Network website (2019) to determine the vehicle types and quantities for a generic enemy force. The enemy force size of an armored brigade and its assets were identified and non-critical assets to the model were dropped. Two final documents were required to find enemy asset characteristics. The first document also came from the Army Training Network site in the form of the Worldwide Equipment Guide for 2016 (2016). This guide has generic weapon and senor data for various equipment across the world. However, this guide is missing certain enemy air defense characteristics. The data for enemy air defense assets came from work done by Dr. Lester W. Grau and Charles K. Bartles on certain modern anti-air weapons systems (2016). With both friendly and enemy forces, for weapon and sensor characteristics that could not be found, inferences were made using similar vehicles to complete the model. It was impossible to find the probabilities of kill for any vehicles or assets in an unclassified medium.

Since JDAFS lacks an ability for entities to survive a certain number of rounds or to explicitly incorporate armor thickness, infantry forces were removed from the study. Additionally, assault vehicles fighting against tanks would not be modeled properly as an assault vehicle armament would have little to no affect against a tank and would mostly only be able to provide scouting capabilities. Therefore, the model was reduced to friendly tanks, artillery, long-range rockets, radar units, and decoys. On the enemy side, the entities included tanks, artillery, surface-to-air missiles, anti-aircraft guns, radar units, and integrated air defense systems (IADS). Table 5 shows the units and their common nomenclature used throughout the modeling and results process.

Table 5. Entity Table

Common Name	Entity Type	Friendly / Enemy / Both
Friendly Tank	M1A2	Friendly
MLRS	Multiple Launch Rocket System	Friendly
Paladin	Paladin Artillery	Friendly
Decoy	Decoy Aerial Vehicle	Friendly
Apache	Apache Helicopter	Friendly
Radar	Generic Radar System	Both
Enemy Tank	T90s	Enemy
Enemy Howitzer	2S19M1	Enemy
MANPAD	SA-24	Enemy
AA Gun	2S6M	Enemy
SAM	9K37	Enemy
IADS	Integrated Enemy Air Defense System	Enemy

Certain threat documents do not show this exact configuration. However, wherever possible, the most effective enemy vehicle of that type was researched and chosen in an effort to model future combat systems in the mid-2020s. For example, generic threat documents do not always show enemy units having both T90s tanks and 2S6M MANPADS within the same unit, but they were both chosen as the most effective pieces of equipment for that type with unclassified and reputable weapons characteristics.

C. CONSTRAINTS, LIMITATIONS, AND ASSUMPTIONS

The biggest constraint in this study was the requirement to use JDAFS as the modeling environment. JDAFS was selected because TRAC – Monterey required a model in which they were confident and had a certain degree of accreditation. In addition, there were few modeling alternatives that could be learned as quickly with such few personnel working on it. The next biggest constraint was that the scenario had to be unclassified. This created issues in finding unclassified weapons characteristics that could be put within JDAFS. This created a possibility for error in that more assumptions were required to complete the inputs within the scenarios.

Entity quantities were a large limitation to this study. Exact quantities of what would be an operational-level engagement was not possible due to memory allocation and optimization sequences. Entity numbers on both sides were reduced to get the model to run on the Hamming supercomputer. Other limitations to this study revolve around the ability of JDAFS to model certain aspects of the simulated engagements. For example, assets on both sides can only be hit by one round before they are destroyed. Non-critical hits are not possible. Similarly, units could not be reconstituted or be temporarily disabled.

These constraints and limitations led to the development of certain assumptions. The first assumption is that units and engagements can be scaled up or down to conduct analysis on strategic level engagements all the way to tactical level scenarios. Assumptions were also made to the way an IADS system was disrupted. Essentially, disruption of this system in the model is equivalent to it being destroyed, but was handled in a manner that provided recommendations as if it were disrupted. Assumptions were also made on friendly and enemy weapon’s characteristics. These are likely the most loosely held assumptions, but due to the nature of having the model on an unclassified system, there was very little that could be done to validate these assumptions.

III. METHODOLOGY

This chapter describes the steps required to use the JDAFS simulation to determine optimal characteristics for the ALE design. In order to achieve this goal, multiple different coding languages were required to streamline the effort and achieve the objective in a manageable time. Without the steps described in this section, this research would have a duration orders of magnitude longer than that conducted in this report. As a quick reference, the methodology for this work is portrayed in the flowchart described in Figure 2.

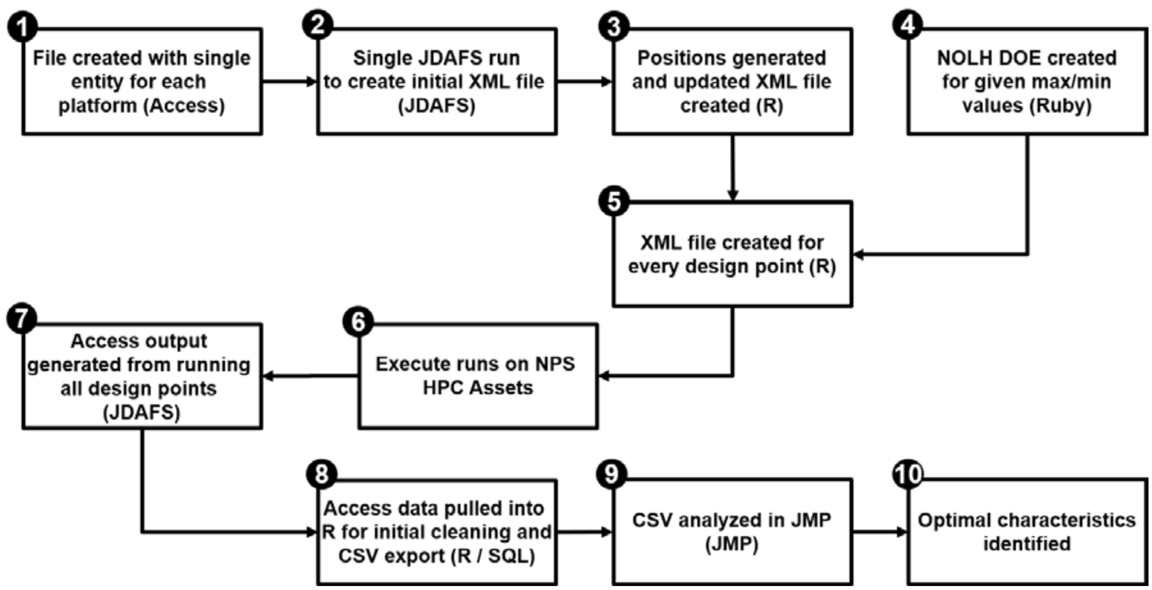


Figure 2. Methodology Flowchart

A. JDAFS SETUP

JDAFS is a delicate environment and generating a sizeable force for two different factions can take an incredible amount of time. Additionally, without an organized and methodical process, it would be near impossible to conduct a full-scale DOE without automating changes. While JDAFS is a system that mostly relies on hand changes in its database to run, creating a base file, changing certain parameters, then running the simulations is a far more streamlined process.

1. Model Creation

The first step in this process was to create a base JDAFS file with the approved entities represented with their given combat characteristics. Tables 6 and 7 show the friendly and enemy entities created in the model and their characteristics. The tables also show the number of entities for each type that are eventually entered for each simulation run but were then added at a later step. If an asterisk accompanied a number, that characteristic or quantity came from a noted reference, allocation table, or United States Army provided generic threat assessment.

Table 6. Friendly Entity Characteristics

Entity	Quantity	Speed	Pk(near)	Pk(far)	Rad D	Kill D	B Rad	Rounds
M1A2	42	30 mph*	.9	.5	4000 m	3500 m	60 m	40*
MLRS	12	65 kph*	.9	.45	N/A	70 km*	100 m	12
Paladin	18*	28 mph	.9	.35	N/A	30 km*	80 m	95*
Radar	13*	31 mph*	N/A	N/A	30 km	N/A	N/A	N/A
Decoy	10	90 knots	N/A	N/A	N/A	N/A	N/A	N/A
Apache	20	50 mph	N/A	N/A	N/A	N/A	N/A	N/A

Table 7. Enemy Entity Characteristics

Entity	Quantity	Speed	Pk(near)	Pk(far)	Rad D	Kill D	B Rad	Rounds
T90s	71	N/A	.9	.45	3000 m	2000 m	60 m	37*
Radar	5*	N/A	N/A	N/A	80 km*	N/A	N/A	N/A
152MM Howitzer	14	N/A	.85	.1	N/A	24.7 km*	80 m	60
MANPAD	42*	N/A	.85	.1	6 km	3 km*	15 m	1
AA Gun	12*	N/A	.5	.1	18 km*	4 km*	1 m	500
SAM	4	N/A	.85	.1	80 km*	40 km*	10 m	3
IADS	20	N/A	.75	.25	113 mi	119 mi	50 m	6
Notional Radar	24	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Based on the scenario, the enemy entities are created as stationary units that do not move. The rational is that the enemy would be conducting a defense and it would logically be safer for them to maintain defensive and protective positions against an approach march instead of breaking position and moving to contact. Therefore, research on speeds for these vehicles were noted but not used in this scenario.

Using conversions provided in Table 4, these values were converted to the JDAFS standard units of measure and stored. In order to input the converted data for simulations, entries were made into a Microsoft Database file (mdb). Within JDAFS 0.7.3, there were certain tables required for entity creation. An example of one of these tables is provided in Figure 3.

name	type	qty	assignment	affiliation	xLoc	yLoc	MaxSpeed
AA_GUN1	AA_GUN	1	combined	Red	415932.2	5063006	0
ALE_D1	ALE_D	1	combined	Blue	223647	4975704	1531.606
ALE_K1	ALE_K	1	combined	Blue	223647	4943762	1531.606
APACHE1	APACHE	1	sensor	Blue	30000	4850606	1331
BLUE_RADAR1	RADAR	1	sensor	Blue	252975.4	4968718	798.5571
BLUE_TANK1	TANK	1	combined	Blue	250000	4963700	798.5571
DECOY1	DECOY	1	sensor	Blue	252675.4	4959733	2756.885
IADS1	IADS	1	combined	Red	517499.3	4850606	0
MANPAD1	MANPAD	1	combined	Red	459596.6	4879864	0
MLRS1	MLRS	1	fires	Blue	194101	4963700	1065
NOTIONAL_RADAR1	NOTIONAL_RADAR	1	sensor	Red	517499.3	4850356	0
PALADIN1	PALADIN	1	fires	Blue	251983.6	4968079	825.1757
RED_HOWITZER1	HOWITZER	1	fires	Red	372818.1	4970637	0
RED_RADAR1	RADAR	1	sensor	Red	362520.4	4961984	0
RED_TANK1	TANK	1	combined	Red	365000	4960000	0
SAM1	SAM	1	fires	Red	389584	4953057	0

Figure 3. Screenshot from the Mover Table

The first step in creating the table was to manipulate the PlatformType Table. In this table, one entity of each type is put into the model. Next, each entity is entered into the Mover Table. Its side and characteristics are entered at its stationary starting location. After this step, waypoints were created to give direction and speed to entities that move. For this scenario, most of the entities are given direct lines of movement, from their start-point directly to their end-point. One exception to this is the movement of the ALEs. In an effort for some representation of the FARA, the ALEs do not move in a straight line. They are grouped in three sets of six, and then branch out at the friendly forward line of troops to their final locations. The effect of this is for real-world representation. If a FARA is shot down before it can deploy its ALEs, all of its ALEs would be destroyed. Mapping the ALEs over each other has the same effect in JDAFS. Enabling these entities to work within JDAFS, the MoverManger Table is used. This table just tells each vehicle or system when to start after the model begins its run, if at all.

An additional aspect of the PlatforrmType Table is critical to the model and required careful consideration before entry. Within this table is a place to assign values to each entity. This value has two important functions. The first function is that it provides a score for each replication that can be used to determine a winner or measure the effectiveness of a battle. This is not used in this study because the survival or destruction of the entities are pulled directly from the results and calculated based on what the sponsors deem important. The second but most critical function of these values are for model optimization. Fires are allocated and optimized based on this value. Given two values are detected, a firing system will engage the target with the highest value from this column. The values for these entities are in Table 8.

Table 8. Entity Optimization Value

Entity	Value	Entity	Value
Anti-Aircraft Gun	700	Apache	900
Kill ALE	1000	Disrupt ALE	1000
Decoy	150	Howitzer	1000
IADS	950	MANPAD	300
MLRS	650	Notional Radar	100
Paladin	700	Radar	500
SAM	900	Tank	800

Entities were placed in anticipation of where the rest of the model entities would fall in on the formation. A sample friendly and enemy formation was initially created and vetted with TRAC–Monterey and Fort Leavenworth. These were created based on common-sense and prior military experience with maneuver forces because of the classification level. The first entity of each type is created in the database, but the rest of the units are generated through XML. A screenshot of the formation generated in JDAFS is displayed in Figure 4.

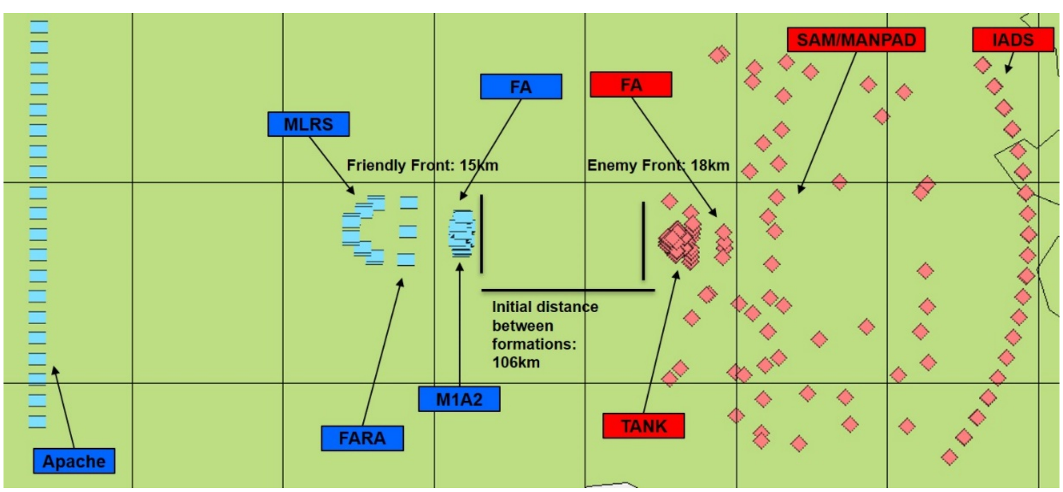


Figure 4. Screenshot of JDAFS Friendly and Enemy Formations

The MunitionType Table is then changed in order to assign weapons to the entities. Similarly, the Munition Table assigns the designed weapons to each required entity. A sample of the MunitionType Table (after all units were properly converted) is shown in Figure 5.

MUNITION	WEIGHT	MER	MINRANGE	MAXRANGE	LOAD	SPEED	ALGORITHM	BURST_SIZE	SUBMUNITION_COUN
AA_GUN	835	80	10	3969.595	500	324000	OLD_DAFS	50	1
ALE_D	835	50	50	2000	8	324000	OLD_DAFS	1	1
ALE_K	835	50	50	2000	1	324000	OLD_DAFS	1	1
IADS	835	50	150	224050.1	6	324000	OLD_DAFS	1	1
MANPAD	835	15	100	2977.196	1	324000	OLD_DAFS	1	1
MLRS	835	100	300	69468.07	20	324000	OLD_DAFS	1	1
PALADIN	835	80	100	29771.96	95	324000	OLD_DAFS	1	1
RED_HOWITZER	835	80	50	24512.29	60	324000	OLD_DAFS	1	1
SAM	835	10	100	39695.95	3	324000	OLD_DAFS	1	1
TANK_CANNON_BLUE	835	60	50	2200	40	324000	OLD_DAFS	1	1
TANK_CANNON_RED	835	60	50	1900	37	324000	OLD_DAFS	1	1

Figure 5. Screenshot from the MunitionType Table

Similar additions and changes were made to the SensorType and Sensor tables. Sensors were added to entities that required them and then assigned to the movers. Not all movers needed both weapons and sensors. Entities such as radar systems provide information for all friendly assets within their radar range. As such, entities that only had a firing capability relied on these sensors to provide distance and direction for fires. Within the SensorType Table, the mean time to detection is also changed for the ALEs to provide a unit of measure on how quickly the ALE sensor can detect enemy units.

The last few tables are essential for any model to run, regardless of the entity types. The Mediator table required modification in order to add a mean time to detection element for the ALEs. The DFASScenario Table changes the number of replications for each run. For this study, ten replications was used because of the memory requirement in the JDAFS Java Virtual Machine. Finally, the Constrained Value Optimizer Table holds the optimization tolerance for the simulation. The implementation interval had to be increased because the simulation had an excessive amount of sensors within a close space and the simulation would time out. Increasing the time between firing and sensing optimizations reduced the memory and time required to execute every design point.

The result of this effort is the creation of seven different base scenarios. The reason for seven scenarios instead of one is because of the need for the different type of ALE configurations. Kill variants of the ALE can only hold one munition. Disrupt ALEs have a changing number of rounds based on the DOE. For scenarios with both configurations, the JDAFS databased required configuring to make sure that one of each type of ALE was starting in the proper place. Figure 6 shows a screenshot of the output for one of the base configurations.

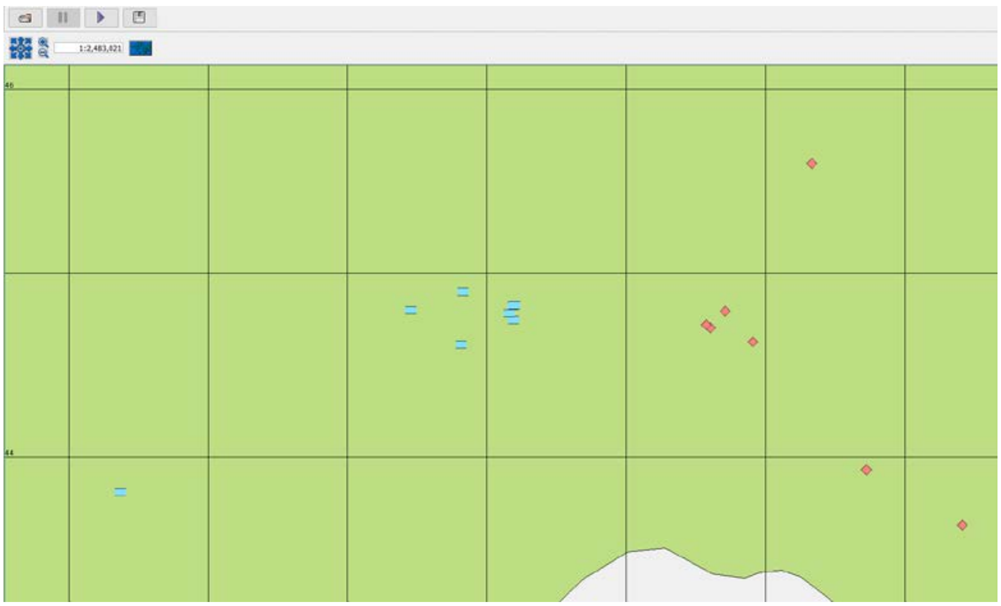


Figure 6. Base Configuration JDAFS Output

2. Additional Model Considerations

There were certain elements that needed adjustment for the model to more accurately depict a real-world scenario. The biggest challenge was to model how a disrupt ALE could have an effect on the battlefield without destroying an enemy asset. A kill variant ALE can only have one munition, so the challenge was to find a solution possible within JDAFS. The solution for this was to create notional enemy radars around enemy air defenses. These radars have the same characteristics as the air defense counterparts and are spaced far enough away from to avoid blast damage from the ALE. Therefore, kill ALEs target air defense entities and disrupt ALEs target notional radars. A destroyed notional radar renders the defense useless without targeting assistance.

There is an interpretability issue behind this methodology. A real-world scenario would allow for these radars or defense capabilities to merely be temporarily disrupted, but without a reconstitution capability in the model, this is just not possible. This work-around is merely a way of attempting to show differences in ALE configurations and would require further exploration in a higher-resolution model.

The second adaptation of the model was the late addition of Apache Helicopters to the model. The inclusion of these vehicles was an additional way of measuring the effectiveness of the ALEs. Their ability to clear an aerial corridor is critical to ensuring the survival of follow on air assets. This provides a measure of ALE capability and a possible metric to help distinguish the optimal ALE characteristics.

Another way to add distinguishability between design points was the inclusion of MANPAD entities on the battlefield. A box was identified around the enemy area of operations. From there, a seed was set in R and 42 random numbers were generated for both the x and y coordinates within JDAFS. With these coordinates established, they were maintained for every design point across every configuration. This provided an added amount of noise to determine whether they affect the model and should studied in future iterations. Full randomization of these positions was unnecessary and inappropriate considering the low number of replications allowed for each design point, given the memory and optimization issues.

There was one big correction that needed to be made after doing some initial testing and small-scale runs. Due to the incredible quantity of sensors and weapons in such a small location, the model would not run, even after relaxing the optimization parameters. In order to get the model to run, even after expanding the amount of memory allocated to the Java Virtual Machine, a small number of entities needed to be removed. This was done in a symmetric manner in that the friendly and enemy forces were not skewed towards one side or another. Table 9 shows the initial number of entities pulled from background research as well as the resulting number of entities that the model can handle.

Table 9. Entity Quantity Corrections in JDAFS

Entity	Initial Quantity	Resulting Quantity
Friendly Tank	58	42
Friendly MLRS	16	12
Enemy Tank	95	71
Enemy Howitzer	18	14

B. XML EXECUTION

Once a shell database file is created, JDAFS can run that scenario. This scenario does not mean much for the process. However, every database that is processed by JDAFS produces an XML file. This file is a replication of the Microsoft Database file, but in an alternate and more compact form. This XML file is the building block for the DOE.

1. XML Basics

The idea behind utilizing XML to conduct the DOE is that the files are small enough to quickly generate and move to the Hamming supercomputer for the simulation runs. Because the DOE consisted of 65 levels, with a stack of ten, across seven configurations, 4,487 different XML files needed to be generated. Each file would be encoded to conduct ten replications in order to generate more fidelity with the data. This base file represents every model entity except for the additional ALEs, which needed to be created using a different file. The easiest method for doing this was to implement the process through the R Studio coding environment (R Core Team 2020).

Using the “xml2” package within R Studio created by Hadley et al. (2019), model entities were easier to find and manipulate. The reason this became important is because of the necessity to potentially change and shift formations or locations of units. Changing these characteristics in Microsoft Access would take a significant amount of time to keep track of and move. Through R Studio, entities are added, modified, and moved in a rapid fashion. This is the reason it was important to create one of each type of mover. That mover is copied as many times as necessary and a function changes the location and destination (if necessary) without having to manipulate the original file. Every entity was given a function to place its additional identical units into a new XML file. Figure 7 provides a screenshot of an entity's function within R.

```
add_single_paladin <- function(center_point_x, center_point_y, end_point_x, end_point_y){
  i <- 1
  while (i < 2){
    c <- length(xml_find_all(b, './/Mover'))
    xml_add_child(xml_child(b, 1), xml_child(xml_child(b, 1), paladin_ref), .where = (c), .copy = TRUE)
    d <- 'PALADIN'
    e <- c + 1
    f <- paste(d, e, sep = '')
    xml_set_attr(xml_child(xml_child(b, 1), e), 'name', f)
    g <- toString(e)
    xml_set_attr(xml_child(xml_child(xml_child(b, 1), e), 5), 'ID', g)
    xml_set_attr(xml_child(xml_child(xml_child(b, 1), e), 5), 'mover', f)
    xml_set_attr(xml_child(xml_child(xml_child(xml_child(b, 1), e), 4), 1), 'mover', f)
    if (i == 1){
      g_x <- toString(center_point_x)
      g_y <- toString(center_point_y)
      xml_set_attrs(xml_child(xml_child(xml_child(xml_child(b, 1), e), 3)), c('xLoc' = g_x, 'yLoc' = g_y))
      g_x2 <- toString(end_point_x)
      g_y2 <- toString(end_point_y)
      xml_set_attrs(xml_child(xml_child(xml_child(xml_child(b, 1), e), 4), 1), c('xLoc' = g_x2, 'yLoc' = g_y2))
    }
    i <- i + 1
  }
}
```

Figure 7. Sample R Studio Code for Entity Creation Function (Paladin Vehicle)

These functions were used to map the units properly across the JDAFS scenario. An additional function utilizing the equation of a circle was created in order to place the IADS in a circular manner away from the front of the enemy defenses. Putting all the elements together, an XML base file was created for every one of the seven scenarios. This effort, while creating a large up-front cost in time to create, likely saved over 48 hours of additional work in having to type every one of these unit entries and changing their locations every time the scenario was modified prior to final execution.

2. DOE Manipulation

With a base file for every scenario, the DOE was executed within R. R read in the XML file as well as the list of design points created from the NOLH Ruby Gem. Functions converted the necessary units required for JDAFS and returned a dataframe for the final DOE. The "xml2" package then used a created function to change every one of the attributes studied in this thesis for the ALE. For every ALE that was a kill variant, the number of rounds were subsequently reduced to a single round. A screenshot of this function is in Figure 8.

```
doe_function <- function(x1, x2, x3, x4, x5, x6, x7, x8, x9){
  a1 <- as.character(x1)
  xml_set_text(xml_child(xml_child(xml_child(b, 1), ale_ref), 1), a1)
  xml_set_attr(xml_child(xml_child(xml_child(xml_child(b, 1), ale_ref), 4), 1), 'speed', a1)
  xml_set_attr(xml_child(xml_child(xml_child(xml_child(b, 1), ale_ref), 4), 2), 'speed', a1)
  # Change max range PK and Kill distance
  a2 <- as.character(x2)
  a3 <- as.character(x3)
  a5 <- as.character(x5)
  xml_set_attr(xml_child(xml_child(xml_child(b, 4), ale_ref), 1), 'minRangePK', a2)
  xml_set_attr(xml_child(xml_child(xml_child(b, 4), ale_ref), 1), 'maxRangePK', a3)
  xml_set_attr(xml_child(xml_child(xml_child(b, 4), ale_ref), 1), 'maxRange', a5)
  # Change radar
  a4 <- as.character(x4)
  xml_set_attr(xml_child(xml_child(b, 5), ale_ref), 'maxRange', a4)
  # Change number of rounds and burst size
  a6 <- as.character(x6)
  xml_set_text(xml_child(xml_child(xml_child(b, 3), ale_ref), 9), a6)
  a7 <- as.character(x7)
  xml_set_text(xml_child(xml_child(xml_child(b, 3), ale_ref), 6), a7)
  # Change mean time to detection
  a8 <- as.character(x8)
  xml_set_attr(xml_child(xml_child(b, 5), 2), 'meanTimeToDetect', a8)
  a9 <- as.character(x9)
}
```

Figure 8. Sample R Studio Code for ALE Attribute Modification Function

This attribute modification function was then nested within the ALE creation function. This led to the final step in file output creation. Each design point created an XML file (641 for each configuration in all), but special care had to be paid in order to name the file properly and capture design characteristics.

3. Grep Utilization

Each file output required a naming convention for output analysis. In order to do post-processing analysis, the input DOE characteristics need to follow along with the results. Therefore, every design carried these variables within its name. An example of this file-naming convention is provided in Figure 9.

Design_12_6_1531.60589923_0.8_0.34_8304.9928_27310.6494_31.6_6_0.00269966_6_

Figure 9. Sample XML File Name

The example above shows that within this simulation run, there are 12 kill ALEs, six disrupt ALEs, a speed of 1,531.6 speed units, a Pk(near) of 0.34, a Pk(far) of 0.34, a radar distance of 8,305.0 distance units, a weapons distance of 27,310.6 distance units, a burst radius of 31.6 distance units, six rounds for all disrupt ALEs, a mean time to detection of .0027 time units, and is part of the sixth configuration. Grep is then used in the post-processing R Studio script in order to capture which characteristics produced which results. Without context of what generate them, the results would have been meaningless. Enabling this process is the "stringr" package through R Studio (Hadley 2019). This package is an efficient way of extracting string data out of given parameters. For this use, it extracts the data necessary from the file name to turn into proper variables for analysis.

4. Hamming Supercomputer

The next step for this thesis was to transfer and run all 4,487 XML files on the NPS Hamming supercomputer. This process was necessary in order to accomplish this study in a reasonable amount of time for multiple reasons. The biggest reason was because of the number of runs required for the DOE. The Java Virtual Machine requires a dedicated core to run. On a personal computer with only six cores, running constantly, it would have taken over 312 hours to accomplish. With ten replications, it takes approximately 20 to 25 minutes to conduct all runs and produce an output file.

The second biggest reason for using Hamming was the memory required for the output. Each design point generated over 40 megabytes of output. The result was over 178 gigabytes of memory in output. Using Hamming allowed for runs to be conducted while simultaneously downloading finished data and moved to a storage space large enough to accommodate the data locally.

Three scripts were generated in order to run all the design points. Additional support for creating efficient batch simulations came from the Hamming help guide produced by Naval Postgraduate School (Naval Postgraduate School 2018). The first script called the requisite amount of memory, a minimum number of computer nodes, and initiated a batch run. The second script properly arrayed the number of runs, directed where all the input XML files were located and where to send the output along with how to name it. The final script dictated to run JDAFS on the input file and to create a Microsoft Access file with the output. Each configuration was batched into two groups in order to offer continuous data processing and initial data analysis.

Using Hamming was not without its complications. Based on the time this study was written, there was an increase in workload across the campus and the number of cores that were dedicated to the study varied wildly between 42 cores and 100. Additionally, due to the nature of quarantined telecomputing during the methodology and analysis periods, downloading data off the Hamming supercomputer through a virtual private network was very slow. Because of this, each configuration took approximately 17 hours to run and download.

5. Merging the Data

With all the data returned and downloaded off Hamming, the final step in the methodology was to determine what information was most important to save and getting that data on a useful platform. JDAFS outputs four tables into each output database with indications for each one of the ten replications. The first table describes when sensors detect and stop detecting enemy units. The second table describes what happens with each shot of a weapon system. The third table reports the end state of every entity in the simulation. The final table records the time and absolute directory path of the simulation.

An important consideration was to determine what data was useful for this study. The fourth table did not provide any relevant information and was immediately deemed unnecessary for the study. The first table provided interesting data, but since the study did not focus on sensor systems, it was also discarded. The second table was useful for initial troubleshooting and to determine whether the model was behaving properly, but was not entirely useful. In discussions with the sponsor and interested parties, the third table was the most important. To be precise, the most important aspects were what entities were alive at the end of a replication. This table was deemed the only one necessary for analysis in this study.

JMP is a powerful data analysis tool that can be utilized to help interpret results (JMP 2020), and it was chosen to make sense of the output data. JMP can input Access data, however, because of the tabular nature of the output, a user interface is required to select the correct data. R Studio has similar issues with importing data due to multiple tables. Administrative permissions were changed on a personal computer and a blank database was created in order to link admin privileges between the computer and R Studio to use the "RODBC" (Ripley et al. 2019) and "odbc" (Hester et al. 2020) packages in R Studio. These packages allowed for a specific table to be extracted from an Access database in an iterative manner.

From this selected table, a few numbers were pulled for final analysis. Friendly ALE and Apache numbers were selected from every replication, as well as friendly missile launchers. Paladin systems and tanks which were grouped together into a friendly vehicle category. Enemy tanks, anti-aircraft guns, and Howitzers were grouped into an enemy vehicle category. Finally, enemy air defense assets and notional radars were selected for final data analysis. Utilizing Grep in R Studio, all of these items were brought into a final database along with the corresponding ALE characteristics for every design point and replication. The resulting dataframe is a by 44,870 by 17 sized element.

There were certain instances where runs timed out on Hamming or not all the data was captured within the Access database. There are many possible reasons for this. One reason could be that the duration set for reserving the nodes on Hamming was not long enough and a run was cut off before it finished. Internet connection may have also been an issue and interrupted the downloading process. Using R Studio, these problematic runs were identified and sent back to Hamming.

There were additional complications in transferring the data from output to R Studio, then subsequently into JMP. For R Studio to read Microsoft Access files, the ODBC Data Source Administrator application on the personal computer needed to be reverted to its 2010 build. The idea for this came from a Microsoft forum (BananaRepublic 2015). This provided the proper operating conditions for the computer to connect the data to a processing program. At this point, an empty database was required to connect permissions for other programs to read Access databases. The instructions for creating this blank database is provided on the "R-bloggers" website by the user "Programming on nielsenmark.us (2018).

IV. ANALYSIS AND RESULTS

This chapter describes the analysis and results stemming gleaned from the data generated by the methodology described in the previous chapter. With a considerable amount of output data, intelligent methods were required to provide actionable recommendations and conclusions from the data. To find the aspects of a successful ALE, the input data was first reviewed. From there, the output data was then analyzed. After these two steps were complete, the final step was to use the analysis to provide answers to the thesis research questions.

The analysis started with determining four measurements of effectiveness (MOEs). These metrics are important measures in determining the success or lack thereof for each configuration. These MOEs fit into the categories of friendly asset survivability and friendly asset lethality. However, after binning the results into certain MOEs, two of these metrics did not provide any insights into the performance of certain characteristics. Because there was no signal, two metrics were dropped, which ended up helping analysis.

A. INPUT ANALYSIS

To provide any recommendations to the sponsor or answer any of the deliverables for this study, the first step was to make sure that everything entered into the model was in accordance with the DOE. This helped to frame follow-on actions and provide context for results.

The first check for making sure that all the inputs were correct in the model was to look at the distributions of the input variables. Figure 10 shows a JMP screen capture of the distributions of these variables that went into the DOE.

Figure 10. JMP Screenshot of Input Distributions

These distributions seem to be reasonable for 65 evenly spaces data points between most of the variables. The initial inclination for a DOE would be to see uniform distributions across each of the variables, but this was not possible each input for multiple reasons. For the number of ALEs, there were more configurations where there were nine kill and nine disrupt ALEs. There was also only one configuration were ALE counts were at their extremes, thus producing discrete almost normal looking distributions. Additionally, with the distributions for round counts, kill ALEs can only have a single round, so the distribution is skewed because the kill ALEs had to be forced to have only one round. Furthermore, because of the intelligent design of NOLHs, the DOE is created to minimize correlation while trying to evenly space points within a range of values for each characteristic. Finally, the discrete nature of certain variables can cause issues with the sample space. This issue is most clearly demonstrated in Figure 11. This screenshot of a scatterplot matrix for the input variables from JMP shows some of the gaps in the design created by discrete variables.

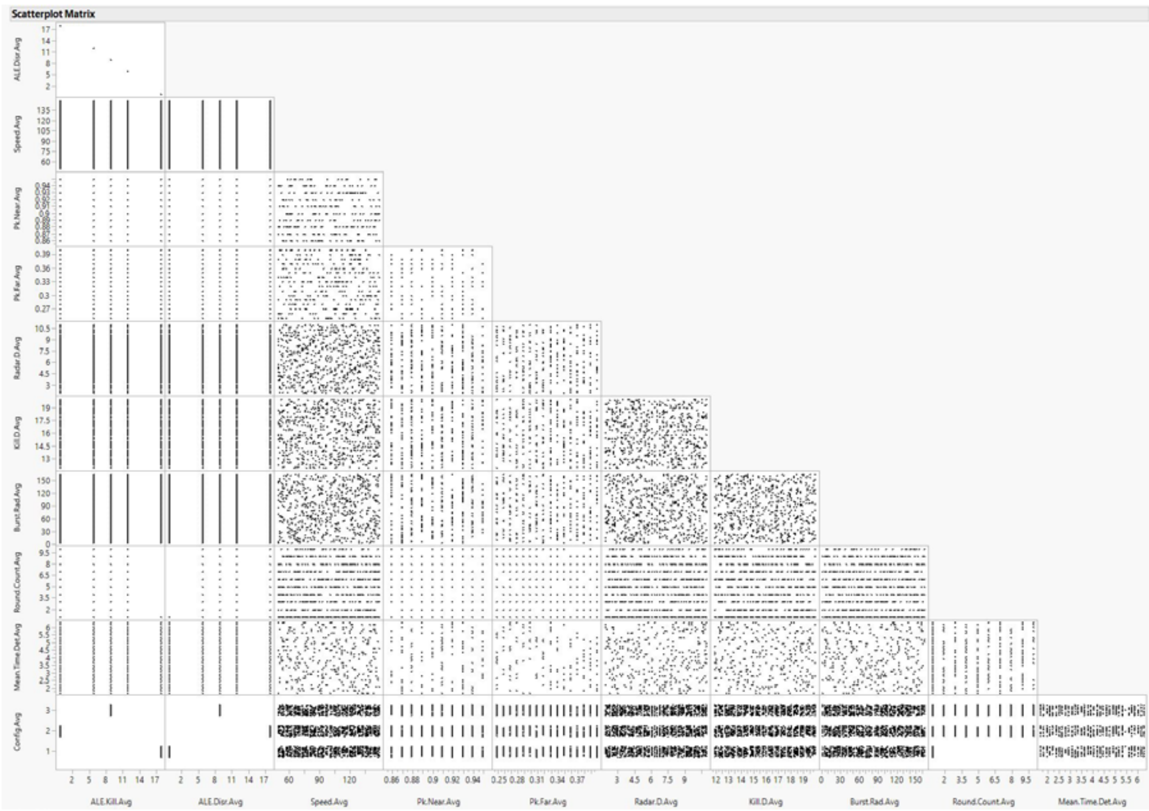


Figure 11. JMP Screenshot of a Scatterplot Matrix for Input Variables

As evident from Figures 10 and 11, this design appeared to be reasonable and valid for analysis. Because of this, there did not appear to be a need to create a second DOE in order to sample within design gaps and the data was sufficient to begin analysis. For reference, the correlations that were not zero between the variables are provided in Table 10.

Table 10. Correlation Coefficients

Variables	**ALE Kill**	**ALE Disrupt**	**Radar Dist.**	**Round Count**
Round Count	-.39	.39	.01	N/A
Configuration	-.17	.17	N/A	.33
Radar Dist.	N/A	N/A	N/A	.01

B. INITIAL OUTPUT ANALYSIS

1. Master File Creation

With the data merged into a single dataframe, two types of analysis were required. The first analysis done was on the unaltered raw data. R Studio was used to combine the results into a single dataframe for each configuration. These files had the configuration parameters and the vehicle status at the end of every design point and replication. R Studio was used to summarize the amount of each vehicle type alive or killed, depending on if they were friendly or enemy assets, by replication. Additionally, vehicles were grouped into categories to better understand the data. Table 11 shows these groupings and the maximum optimal value for each of these groups in terms of survival or killed. As a reminder, the only way within JDAFS to simulate a disruption was to kill a radar system supporting an enemy air defense asset, annotated as a notional radar system.

Table 11. Summarized Vehicle Categories Table

Group Category	**Vehicles in Group**	**Maximum Survive / Dead**
ALE	ALE	**18**
Apache	Apache	**20**
Blue	Tank / MLRS / Paladin	**72**
Red	Tank / Howitzer / AA Gun	**97**
ADA Killed	IADS / SAM	**48**
ADA Disrupt	Notional Radar	**48**

Each configuration resulted in a dataframe with 6410 entries that were eventually merged into one master file. This created a file containing each replication's design characteristics and the value for the number survived or killed for the above groups.

2. Metric Determination

The initial intent was to look at the most important metrics a commander would use to assess an engagement: friendly ALEs surviving, friendly Apaches surviving, friendly vehicles surviving, enemy vehicles killed, and enemy air defenses killed or disrupted. For ease of analysis, these MOEs were turned into percentages, and ground vehicles for friendly and enemy sides were grouped together. Table 12 shows the four MOEs for initial consideration.

Table 12. MOE Formulas

MOE	Formula
1	$\frac{ALE\ Surviving}{Total\ ALE}$
2	$\frac{Killed\ E.Air\ Defense + (.6)Disrupted\ E.Air\ Defense}{Total\ E.Air\ Defense + (.6)Total\ E.Air\ Defense}$
3	$\frac{Apaches\ Surviving}{Total\ Apaches}$
4	$\frac{Surviving\ F.Ground\ Forces + (.5)Killed\ E.Ground\ Forces}{Total\ F.Ground\ Forces + (.5)Total\ E.Ground\ Forces}$

These MOEs were confirmed with the sponsor as appropriate for analysis. The rationale behind this decision is that for MOE2, a disrupted enemy air defense system is worth 60% of the value of a killed enemy air defense system. Similarly, for MOE4, a surviving friendly vehicle was deemed twice as important as killing an enemy ground force unit. Figure 12 shows the histograms for MOE1 and MOE2 over every battle within the simulation. There is enough variability within this data to provide reasonable confidence in the possibility to conduct modeling in support of characteristic analysis.

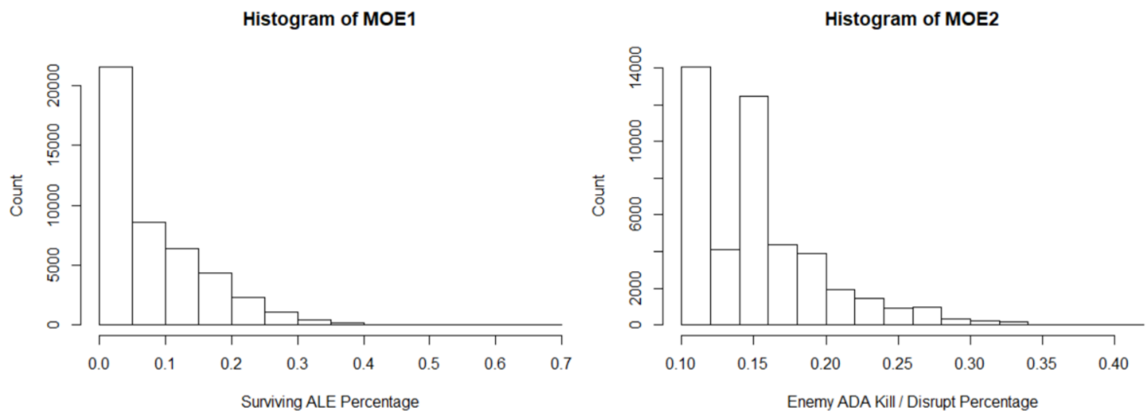


Figure 12. Histogram Graphs for MOE1 and MOE2

A reduction in the number of MOEs was required due to a lack of variance and an inability to properly model these metrics. This showed that varying ALE parameters did nothing to affect the outcome of these parameters. As visual confirmation of this, Figure 13 shows the two histograms of these MOEs. They show very little variance and the inability to fit any descriptive model for these MOEs with reasonable confidence made them of little interest to the sponsor.

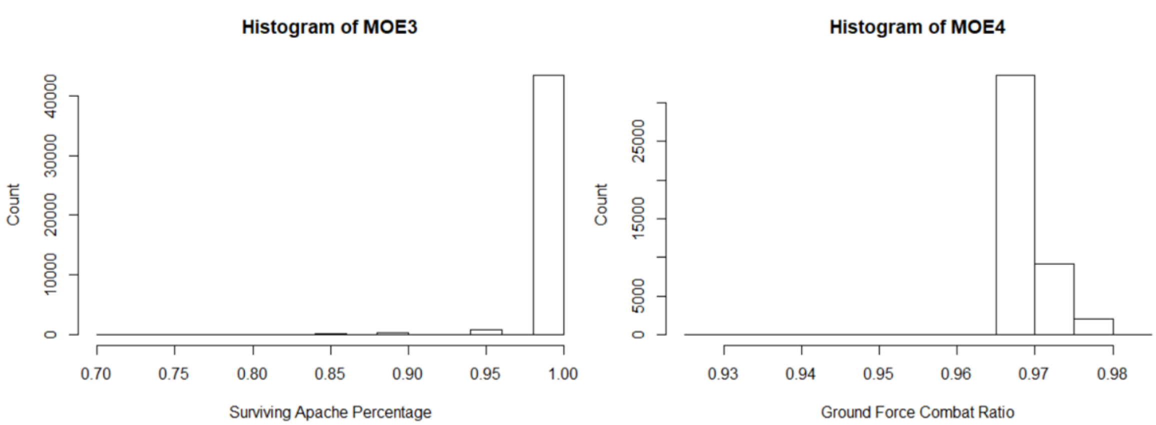


Figure 13. Histogram Graphs for MOE3 and MOE4

An additional benefit of using these histogram graphics in initial analysis was that outliers were easy to determine. 11 design points lied well outside the reasonable range of values. In-depth research of these points showed that the runs for these 11 design points did not run properly and output data was cut off when ran on the Hamming Supercomputer. For these runs, the commands sent to the Hamming supercomputer timed-out and the full output data was incomplete in areas describing whether vehicles were alive or dead at the end of each replication. Without full vehicle survivability data, the data points were useless. These points were identified, run again, and incorporated into the plots above.

C. LINEAR REGRESSION

The next step in this study was to find a method of modeling that could be used in tradeoff analysis. In particular, the most important deliverable in creating a model is determining how much a metric of interest is reduced or improves by varying a particular parameter or combinations of parameters. With the analysis reduced to two MOEs, the first natural step was to see how well a linear model describes the results. From Julian Faraway's text on linear models, he describes this type of modeling as having the goals of predicting future responses and assessing the relationships between predictor variables while having a single response variable (Faraway 2015, p. 7-8). Because each MOE is a single response, and initially assumed to be independent, using linear regression was a quick way to provide cursory analysis on the importance of predictor variables and their possible relationships.

1. Linear Model Using the Raw Data

Using the total dataset, a linear model with two-way interactions and polynomials to the second degree was created in JMP. A stepwise regression using a least-squares fit was generated using the data. Models were created to predict the two remaining MOEs using every design characteristic as an independent variable and the configuration as a factor. Figure 14 shows the results of linear regression on the two MOEs.

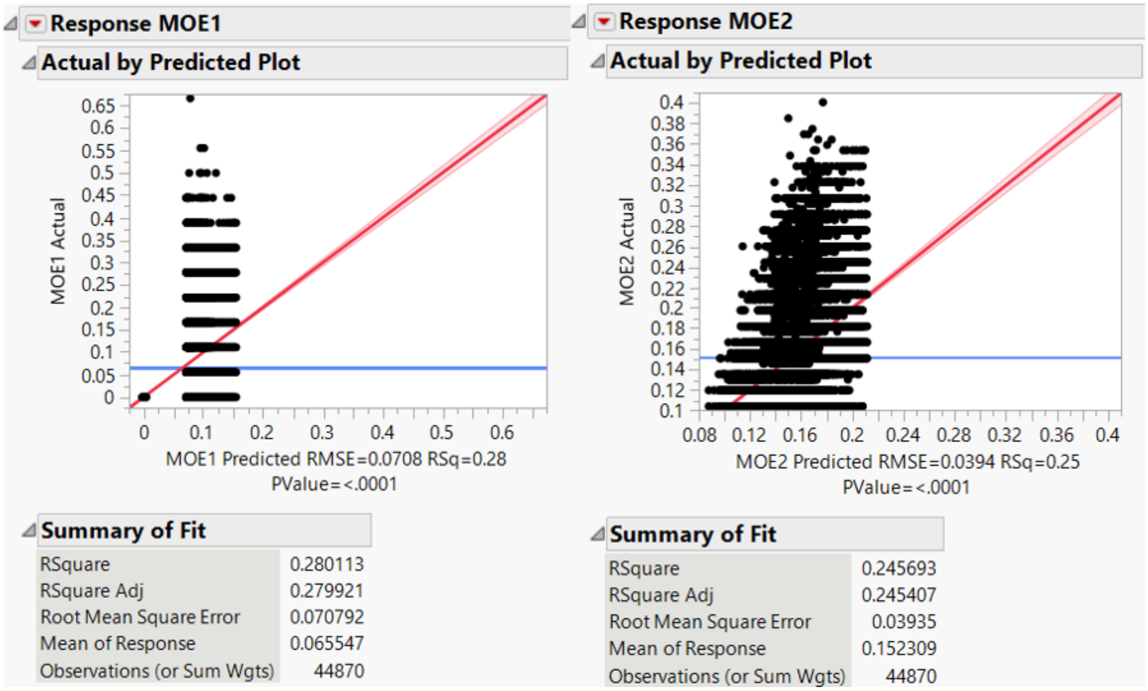


Figure 14. JMP Screenshots of Initial MOE Linear Regression

These two linear models have extremely poor fits. The R-square values as well as the graphical representations show that there is very little evidence to conclude that modeling the MOEs in this fashion is a reasonable representation.

Based on the histograms in Figure 12, the assumption that the residuals for both MOEs are normal is also not a valid one. However, to improve predictability, the data was averaged across each of the ten replications for each design point. Histograms in Figure 15 shows the improvement in the normalization of the data.

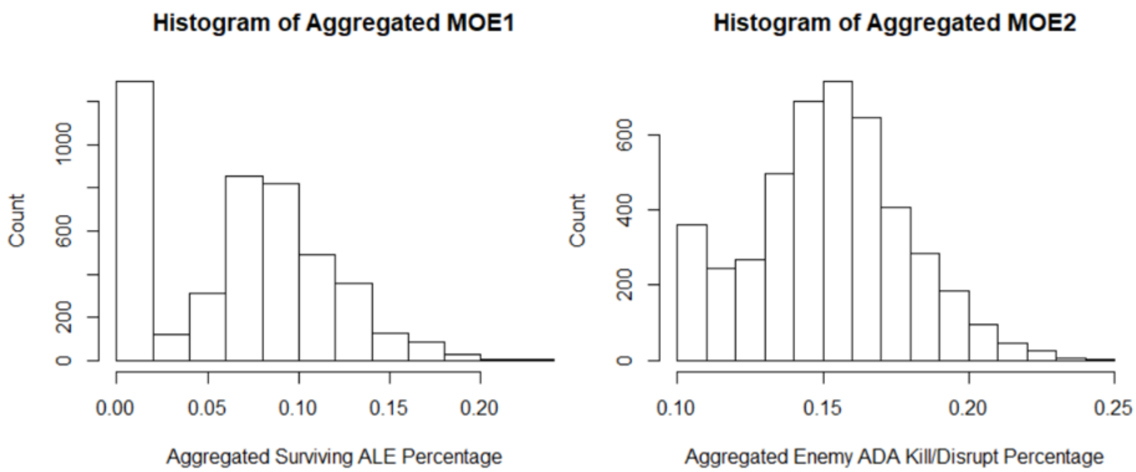


Figure 15. Histogram Graphs for Aggregated MOE1 and MOE2

2. Linear Model Using the Aggregated Data

The improvements through aggregating across the replications go beyond the normalization of the MOEs. The linear model fit increases a substantial amount and is graphically much more representative of the data. Figure 16 shows the improved fit with a stepwise regression using a least-squares fit with two-way interactions and polynomials to the second degree. Three-way interactions and polynomials to the third degree were checked but they provided a miniscule benefit to the data that was not deemed worthy of the loss of parsimony and interpretability.

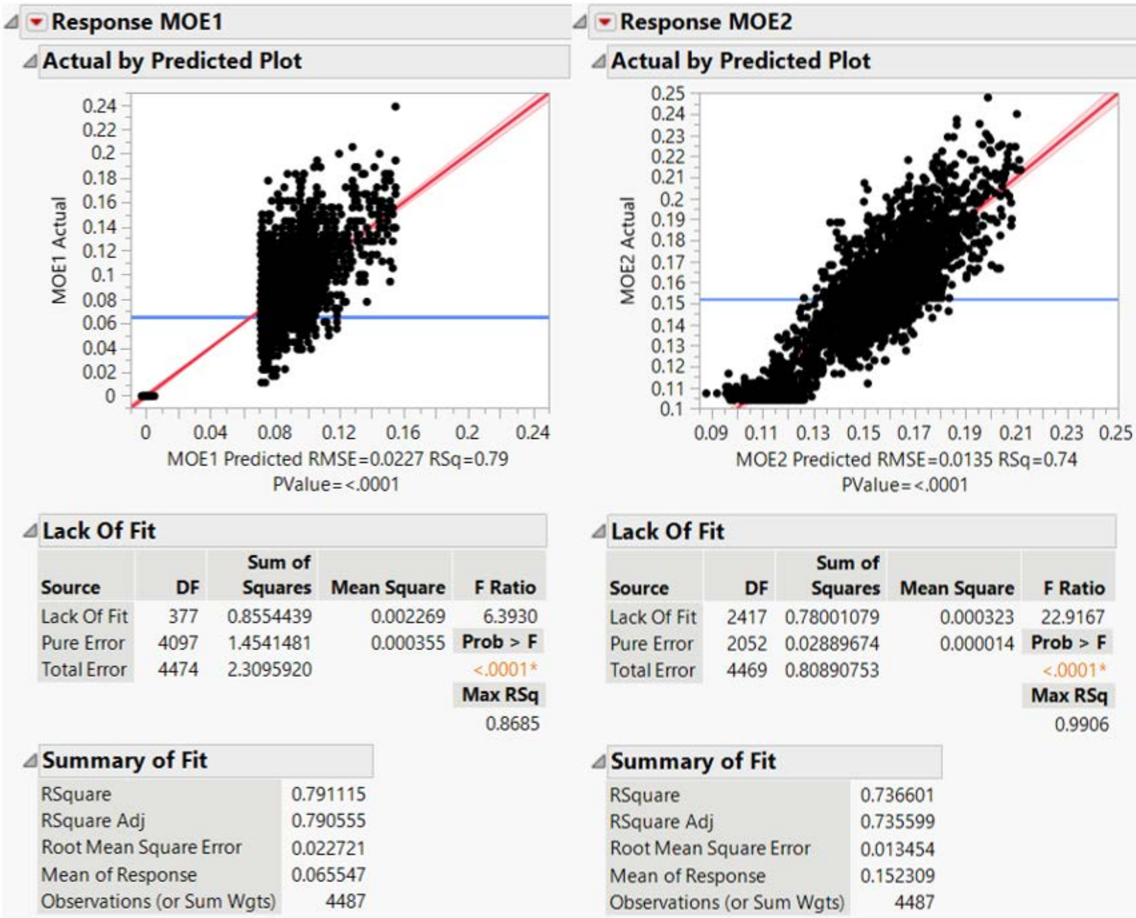
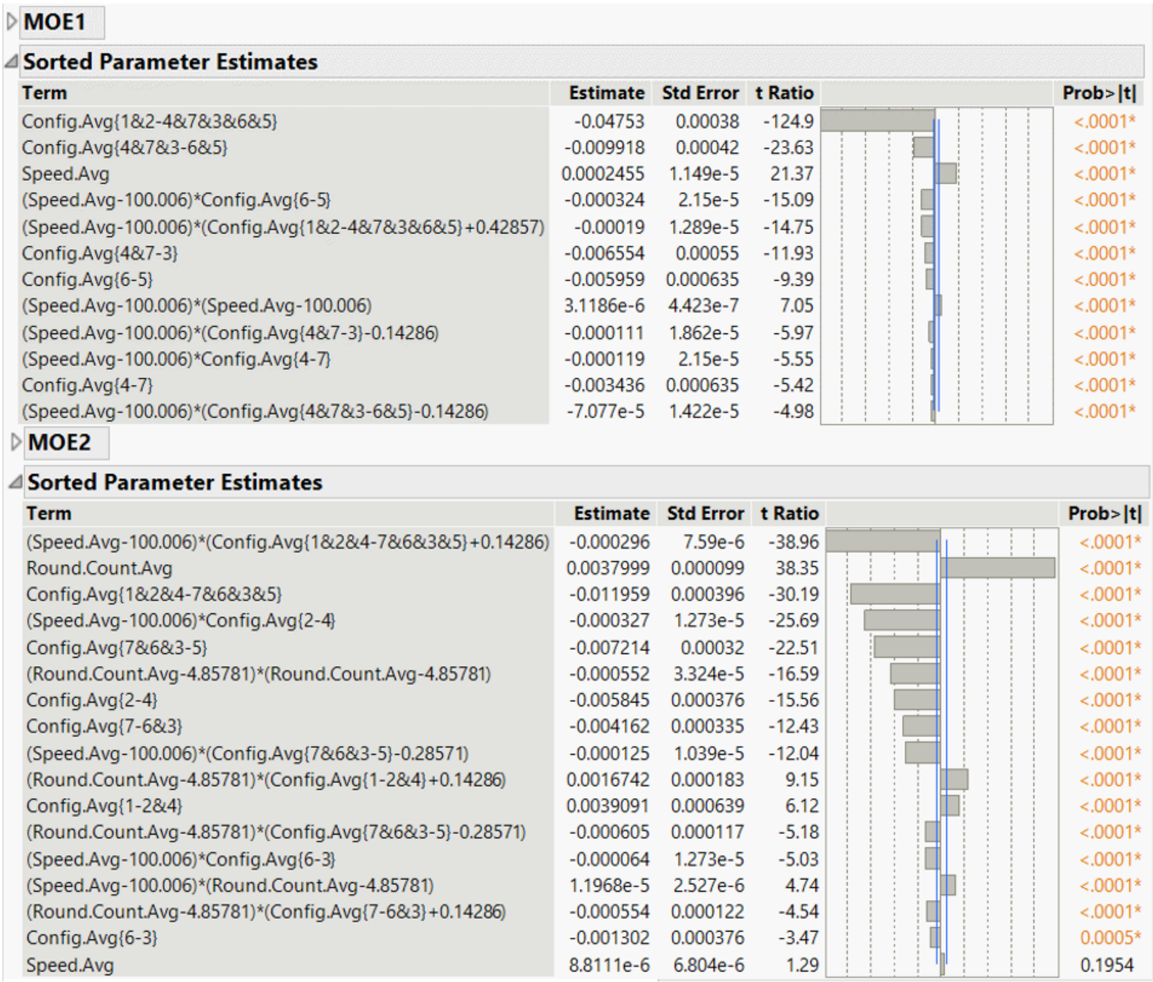


Figure 16. JMP Screenshots of Aggregated MOE Linear Regression

In addition, a better fitting model provides additional confidence in the variables and interactions actually important in the model. But these models still are not what one would like to see in predictive analysis. With R-square values below .90, this shows the complexity of the simulations. The p-values for these regressions will not be due to the lack of normality in the residuals, but the unbiased metamodels are still informative using a least-squares fit. Good insights can be gathered from these MOE responses being explained by important variables. Figure 17 shows these critical parameters of the two models in order of their statistical significance. Note that MOE1 is dominated by ALE configuration categories while MOE2 depends more on multiple variables. This is likely due to the nature of the MOEs. The focus of MOE1 on survivability is much less complex than MOE2, which is focused on lethality and then to a much lesser extent on survivability.

But for MOE2, survivability is still important because a vehicle cannot be lethal if it does not survive long enough to fire on the enemy.



MOE1

Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Config.Avg{1&2-4&7&3&6&5}	-0.04753	0.00038	-124.9	<.0001*
Config.Avg{4&7&3-6&5}	-0.009918	0.00042	-23.63	<.0001*
Speed.Avg	0.0002455	1.149e-5	21.37	<.0001*
(Speed.Avg-100.006)*Config.Avg{6-5}	-0.000324	2.15e-5	-15.09	<.0001*
(Speed.Avg-100.006)*(Config.Avg{1&2-4&7&3&6&5}+0.42857)	-0.00019	1.289e-5	-14.75	<.0001*
Config.Avg{4&7-3}	-0.006554	0.00055	-11.93	<.0001*
Config.Avg{6-5}	-0.005959	0.000635	-9.39	<.0001*
(Speed.Avg-100.006)*(Speed.Avg-100.006)	3.1186e-6	4.423e-7	7.05	<.0001*
(Speed.Avg-100.006)*(Config.Avg{4&7-3}-0.14286)	-0.000111	1.862e-5	-5.97	<.0001*
(Speed.Avg-100.006)*Config.Avg{4-7}	-0.000119	2.15e-5	-5.55	<.0001*
Config.Avg{4-7}	-0.003436	0.000635	-5.42	<.0001*
(Speed.Avg-100.006)*(Config.Avg{4&7&3-6&5}-0.14286)	-7.077e-5	1.422e-5	-4.98	<.0001*

MOE2

Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
(Speed.Avg-100.006)*(Config.Avg{1&2&4-7&6&3&5}+0.14286)	-0.000296	7.59e-6	-38.96	<.0001*
Round.Count.Avg	0.0037999	0.000099	38.35	<.0001*
Config.Avg{1&2&4-7&6&3&5}	-0.011959	0.000396	-30.19	<.0001*
(Speed.Avg-100.006)*Config.Avg{2-4}	-0.000327	1.273e-5	-25.69	<.0001*
Config.Avg{7&6&3-5}	-0.007214	0.00032	-22.51	<.0001*
(Round.Count.Avg-4.85781)*(Round.Count.Avg-4.85781)	-0.000552	3.324e-5	-16.59	<.0001*
Config.Avg{2-4}	-0.005845	0.000376	-15.56	<.0001*
Config.Avg{7-6&3}	-0.004162	0.000335	-12.43	<.0001*
(Speed.Avg-100.006)*(Config.Avg{7&6&3-5}-0.28571)	-0.000125	1.039e-5	-12.04	<.0001*
(Round.Count.Avg-4.85781)*(Config.Avg{1-2&4}+0.14286)	0.0016742	0.000183	9.15	<.0001*
Config.Avg{1-2&4}	0.0039091	0.000639	6.12	<.0001*
(Round.Count.Avg-4.85781)*(Config.Avg{7&6&3-5}-0.28571)	-0.000605	0.000117	-5.18	<.0001*
(Speed.Avg-100.006)*Config.Avg{6-3}	-0.000064	1.273e-5	-5.03	<.0001*
(Speed.Avg-100.006)*(Round.Count.Avg-4.85781)	1.1968e-5	2.527e-6	4.74	<.0001*
(Round.Count.Avg-4.85781)*(Config.Avg{7-6&3}+0.14286)	-0.000554	0.000122	-4.54	<.0001*
Config.Avg{6-3}	-0.001302	0.000376	-3.47	0.0005*
Speed.Avg	8.8111e-6	6.804e-6	1.29	0.1954

Figure 17. JMP Screenshots of Variable Importance

It is important to look at correlation within the datapoints before conducting initial analysis on variables that should be further studied in-depth. The DOE was structured in an NOLH format in order to minimize correlation. This is evident from a screenshot from R Studio of the correlation table from Figure 18.

	ALE.Kill.Avg	ALE.Disr.Avg	Speed.Avg	Pk.Near.Avg	Pk.Far.Avg	Radar.D.Avg
ALE.Kill.Avg	1.00	-1.00	0	0	0	0.00
ALE.Disr.Avg	-1.00	1.00	0	0	0	0.00
Speed.Avg	0.00	0.00	1	0	0	0.00
Pk.Near.Avg	0.00	0.00	0	1	0	0.00
Pk.Far.Avg	0.00	0.00	0	0	1	0.00
Radar.D.Avg	0.00	0.00	0	0	0	1.00
Kill.D.Avg	0.00	0.00	0	0	0	0.00
Burst.Rad.Avg	0.00	0.00	0	0	0	0.00
Round.Count.Avg	-0.39	0.39	0	0	0	0.01
Mean.Time.Det.Avg	0.00	0.00	0	0	0	0.00
Config.Avg	-0.17	0.17	0	0	0	0.00

	Kill.D.Avg	Burst.Rad.Avg	Round.Count.Avg	Mean.Time.Det.Avg	Config.Avg
ALE.Kill.Avg	0	0	-0.39	0	-0.17
ALE.Disr.Avg	0	0	0.39	0	0.17
Speed.Avg	0	0	0.00	0	0.00
Pk.Near.Avg	0	0	0.00	0	0.00
Pk.Far.Avg	0	0	0.00	0	0.00
Radar.D.Avg	0	0	0.01	0	0.00
Kill.D.Avg	1	0	0.00	0	0.00
Burst.Rad.Avg	0	1	0.00	0	0.00
Round.Count.Avg	0	0	1.00	0	0.33
Mean.Time.Det.Avg	0	0	0.00	1	0.00
Config.Avg	0	0	0.33	0	1.00

Figure 18. R Studio Screenshot of Aggregated Correlation Table

In predicting MOE1, the configuration type, speed, and their interactions are the most important factors in a linear regression. The biggest takeaway is that speed is very important to the ALEs success with respect to this MOE. The only positive coefficients in the model involve speed. Different configurations then make the MOE slightly worse or significantly worse. What is interesting in this model is the lack of the number of rounds as an important parameter. This means that, in terms of its own survival, the number of rounds an ALE has does not determine its survivability.

In predicting MOE2, the configuration type, speed, and round count are the most important factors in a linear regression. This regression passes the common-sense test in that, in determining how well the ALE does in a metric focused on killing or disrupting the enemy, the number of rounds the vehicle has would be important. It also makes sense that speed is important to this MOE as well. As with MOE1, an ALE can only influence the enemy if it survives, and speed is important to survival.

In looking at three dimensional plots with linear regression, there are some optimal value ranges that can be inferred from the model. In JMP, these visualizations were created by varying the parameters important to each of the MOEs while keeping non-critical parameters at their mean. The surface plot for MOE1 is in Figure 19. It is important to note that configuration is a qualitative variable.

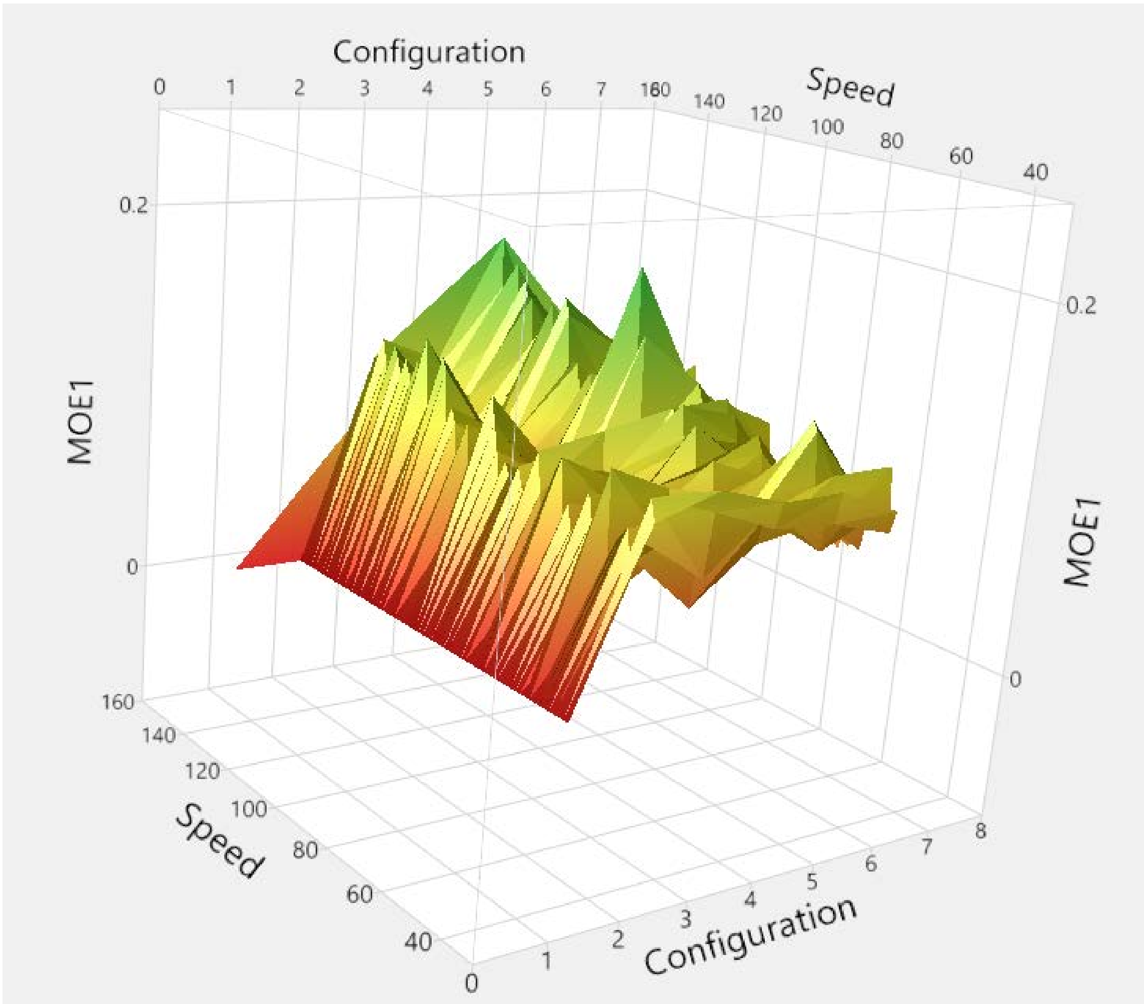


Figure 19. JMP Screenshot of Surface Plot for MOE1 (Speed versus Configuration versus MOE1)

This figure shows that certain characteristics help improve survivability at the same rate or better than others. This is important in the event that a certain type of ALE, whether kill or disrupt, may cost more. Varying configurations may help reduce future costs instead of relying solely on the more expensive type. Additionally, it generally appears that a higher speed improves survivability. But, there is a region at approximately 100 knots where the ALE performs just as good, if not better than when flying at 150 knots. So long as the fifth configuration is used, speed can be sacrificed with no increase in destroyed vehicles. This could provide huge savings when programming for the final ALE design; or, resources can be diverted to other portions of the vehicle.

Three-dimensional plots were also created for MOE2in JMP. Once again, parameters not important to the MOE were held constant at their average. Two plots were necessary because there are three parameters important to this MOE. Figures 20 and 21 show the results of this effort.

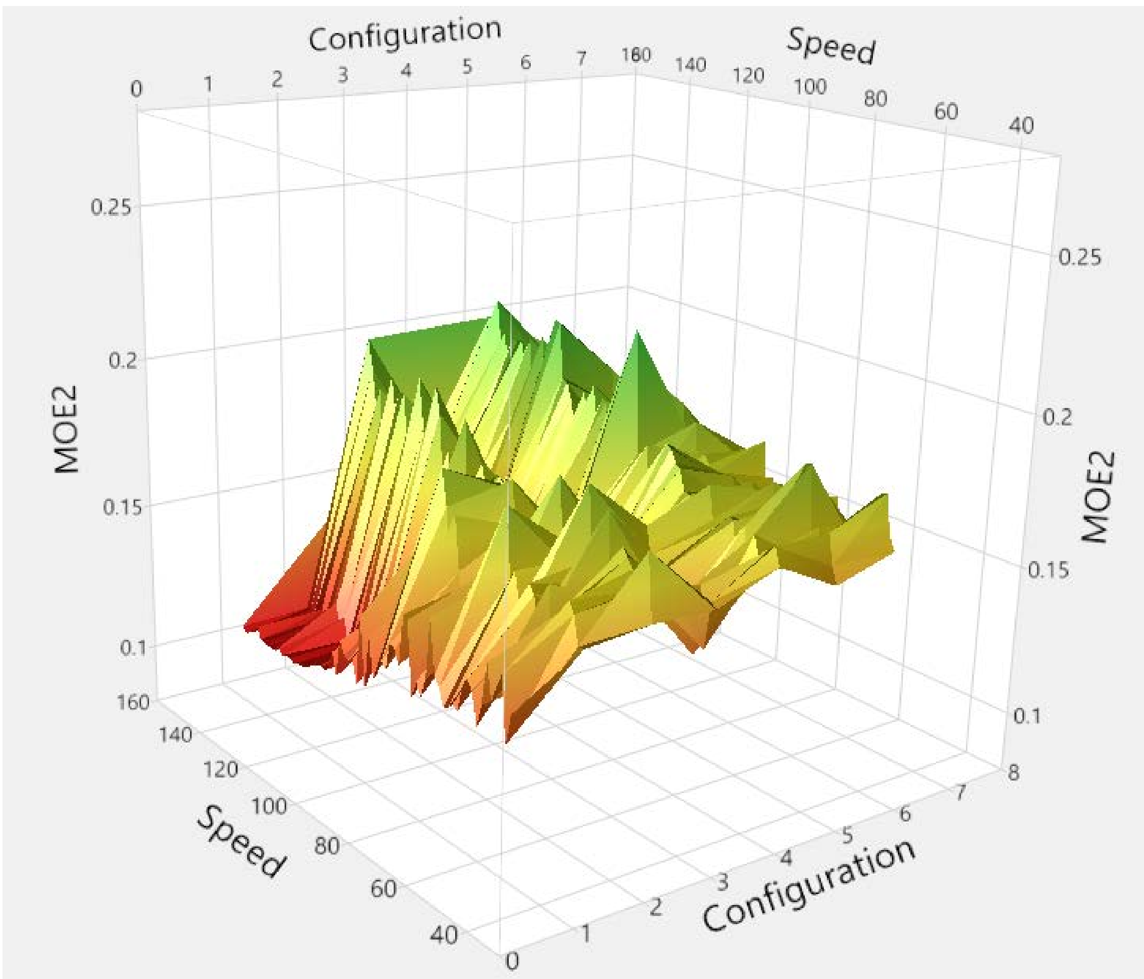


Figure 20. JMP Screenshot of Surface Plot for MOE2 (Speed versus Configuration versus MOE2)

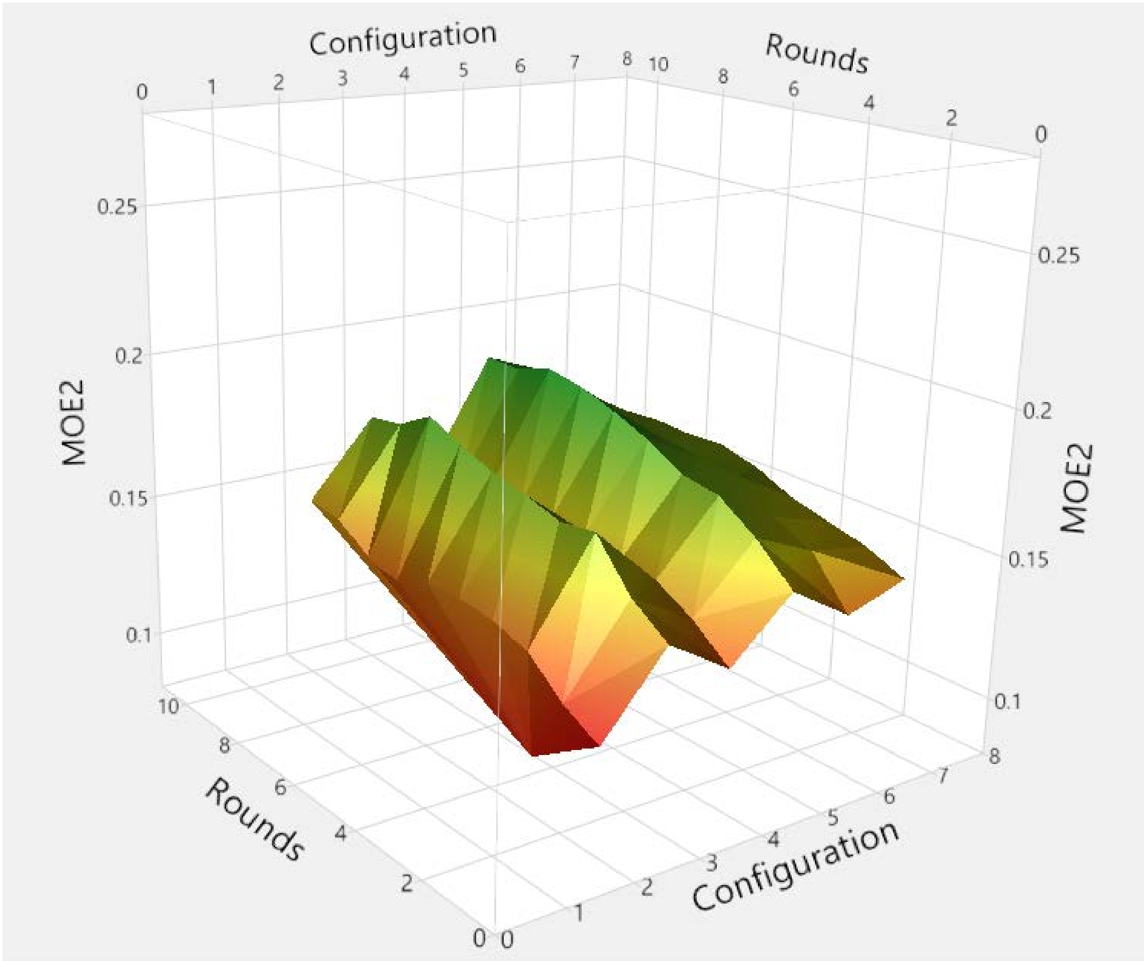


Figure 21. JMP Screenshot of Surface Plot for MOE2 (Rounds versus Configuration versus MOE2)

In terms of lethality, the fifth configuration is also important with respect to MOE2. 100 knots as a speed is also as important to MOE2 as it is to MOE1. The reason for this is likely because at speeds higher than 100 knots, the ALE is likely going too fast and is gets too close to the decoys deployed ahead of them, giving the IADS a choice between targets instead of spending rounds primarily on decoys. Eight rounds also seem to be a good configuration for the ALE, as there are marginal returns to arming the ALE with anything more. Table 13 provides recommendations for areas of study for certain characteristics in future modeling efforts.

Table 13. Areas for Further Study Based on Linear Regression

Parameter	Area for Future Study
Speed	95-105, 120–125, 150 + (knots)
Number of Rounds	5-8 (rounds)
Configurations	3, 5

D. RANDOM FOREST

The benefit of using linear regression is the interpretability of the model and it is easy to see whether parameters have a positive or negative influence on an MOE and what their interactions may be. However, a random forest was able to provide a better fitting model and conduct predictive analysis. Based on Dr. Robert Koyak's class lectures, the idea of using random forests for this type of analysis is appropriate for a few reasons (2019). First, the model uses binary splitting to fit the data. This means that, based on simple questions on the characteristics, counts of the number of times a specific value comes up for an MOE can be traced back to different bins it falls into through the different classifications. Additionally, they can be easily read to provide recommendations without the complex interaction terms or transformations.

1. Random Forest Creation

The "rpart" package was used in R Studio to create two separate models for the MOEs with a complexity parameter of .0001 for each (Therneau et al. 2019). This could lead to overfitting, particularly for future prediction capabilities. In an effort to combat overfitting, the forest splits were pruned back within one standard deviation of the best splits. This method of pruning random forests came from an advanced data analysis class lecture by Dr. Robert Koyak (2019). Once again, the aggregated data was used to create these models. The aggregated data again performed considerably better than models generated with the raw dataset.

For the random forest for MOE1, the optimal number of splits after this methodology was 17 splits. This produced a relative error of .1932 and a cross-validation error of .1981. For the random forest for MOE2, the optimal number of splits was 36 splits.

This random forest produced a relative error of .2208 and a cross-validation error of .2402. This shows how complex it was to model MOE2.

For both MOEs, Table 14 shows the important variables to each random forest and the variable importance for those factors. Variables left out of this table were orders of magnitude less important than any of the others in the table.

Table 14. Random Forest Variable Importance

MOE1 Variable	Variable Importance	MOE2 Variable	Variable Importance
# Disrupt ALE	8.1744	Configuration	1.4919
# Kill ALE	4.2467	# Rounds	.8122
Configuration	4.2467	# Disrupt ALE	.7584
# Rounds	2.9141	# Kill ALE	. 7584
Speed	.5340	Speed	.41047

The "rattle" package was used in R Studio to provide some visualization to these two random forests (Williams 2011). This package provides improved controls on random forest visualization though increased user controls. Although difficult to see because of the complexities of the trees, they are displayed in Figure 22 and Figure 23 for MOE1 and MOE2, respectively.

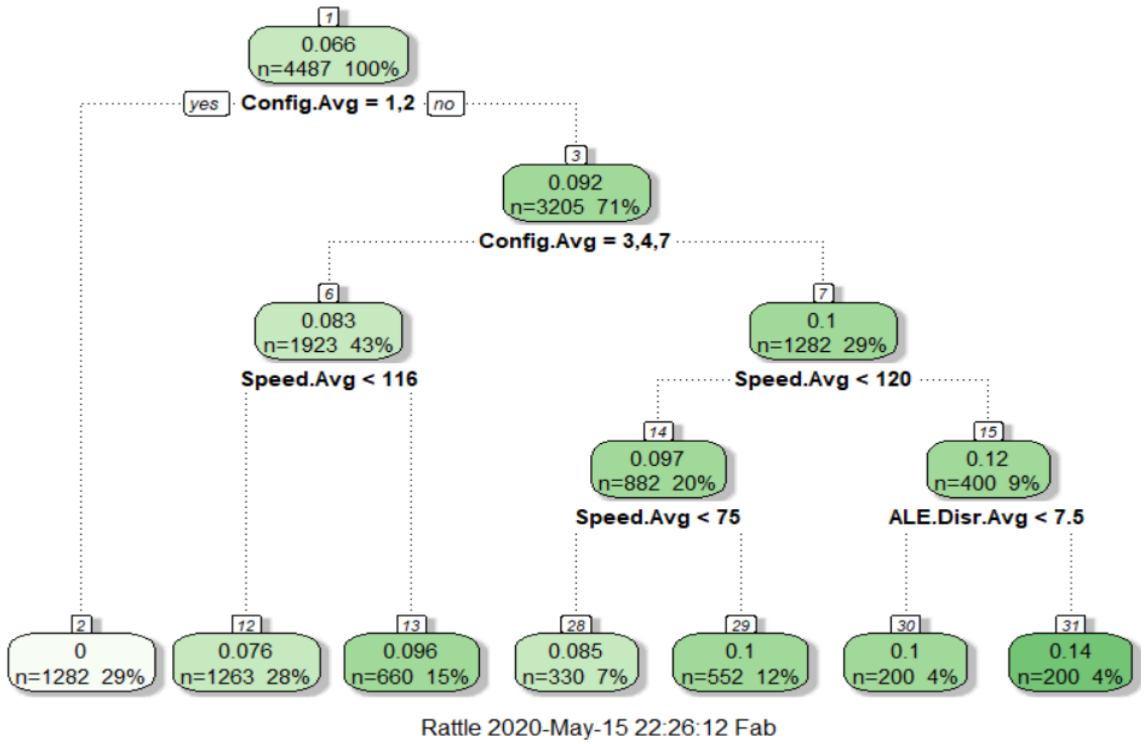


Figure 22. R Studio Minimal Split "rpart" Random Forests for MOE1

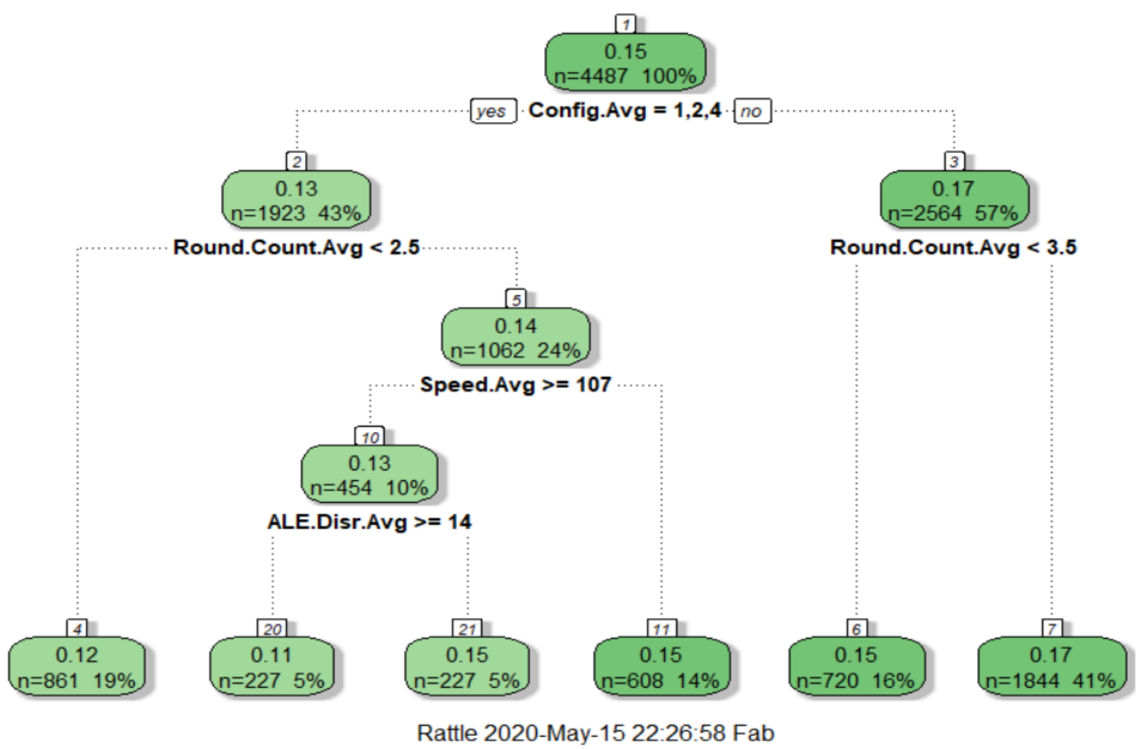


Figure 23. R Studio Minimal Split "rpart" Random Forest for MOE2

These random forests are important because they provide good fits while providing relaxations on some of the assumptions. Because they have no distributional assumptions, correlation and other statistical issues that plague linear regression are not required for model validation. In addition, interactions occur naturally. Particularly with visualization, splits show a very simple way of determining what is critical to providing success for different MOEs.

For both random forests, the first and most important factor in the fit is the configuration. For MOE1, after configuration splits, the random forest splits on the speed of the ALE, followed by another split between speed and the number of disrupt ALEs. This shows how important these very few variables are to the fit. There are additional splits that fall within one standard deviation of the best number of splits, but for legibility, these plots for both MOE1 and MOE2 are in the appendix, Figures 29 and 30. For MOE2, configuration is still the most important factor for the splits in the random forest. Next, the round count becomes the most important variable to the model fit. After that, speed and the number of disrupt ALEs become important.

An interesting insight of these two trees is the values for the characteristics on the initial splits. For the maximum values for each of the MOEs, the paths of the trees do not immediately go to endpoint or maximum values. Rather, the trees take more than a few splits to get to levels where the random forests split near these maximum values. This shows that, based on this scenario, the ALEs do not need to be at their maximum values for every characteristic to have optimal performance.

2. Improved Random Forest Visualization

The decision trees for the random forests provide a cursory glance at the complexity of the random forest and the importance of certain characteristics based on their placement within the plot, but it does little for in-depth tradeoff analysis. The next step was to create a visualization method to enhance the ability to determine where in the DOE the critical characteristics provided the greatest benefit to the modeled FARA ecosystem.

Because there are 11 dimensions varied through the design, visualization is particularly difficult. To alleviate this issue, the variable importance for both MOEs were taken into consideration and only the five most important variables were brought in for visualization and tradeoff analysis. Because variables after the top five most critical were so much less important, changes in their values produces such minimal deviations that they would not provide much information for follow-on analysis. Within the most important characteristics, located in Table 13, ALE configuration is a method of representation for the number of kill and disrupt ALEs. Although not extremely correlated, with the correlation between kill ALE and configuration to be –0.17 and the correlation between disrupt ALE as .17, they are the most highly correlated variables in the DOE. An assumption that they are correlated enough was a bold assumption, but necessary to reduce dimensionality for visualization. Pairs plots were generated to predict both MOEs using their respective random forests, to include smoothers, but there were very little interactions between the variables. The pairs plots for the most critical factors to MOE1 and MOE2 are provided in Figure 24 and Figure 25, respectively. The full pairs plots for all variables are located in the appendix, Figure 31 and Figure 32, respectively.

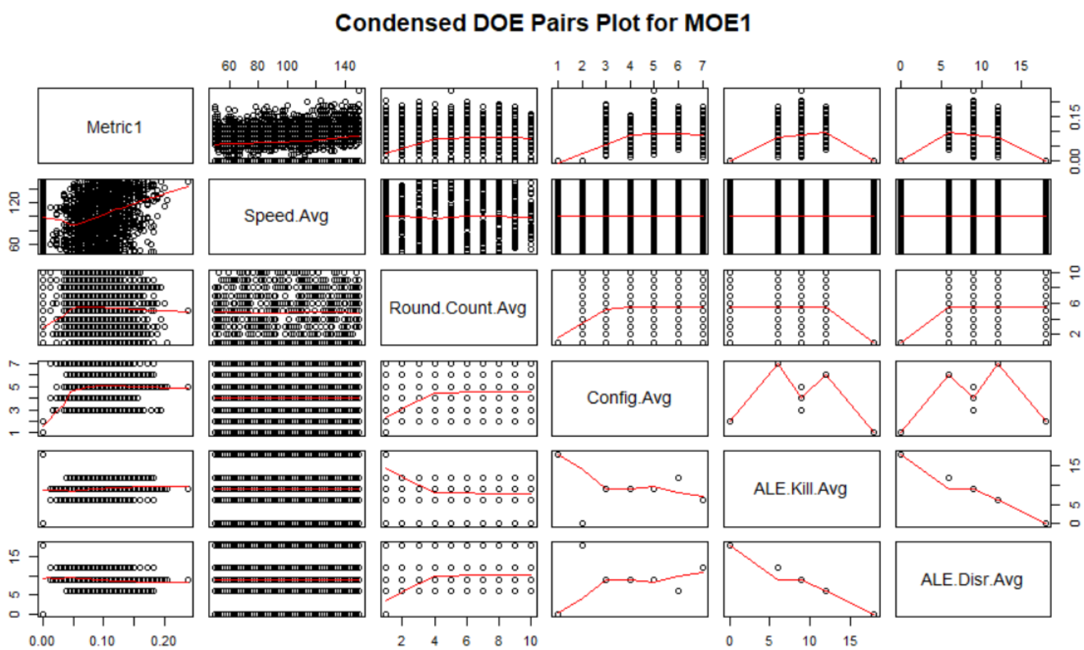


Figure 24. R Studio Screenshot of MOE 1 Pairs Plot with Smoothers

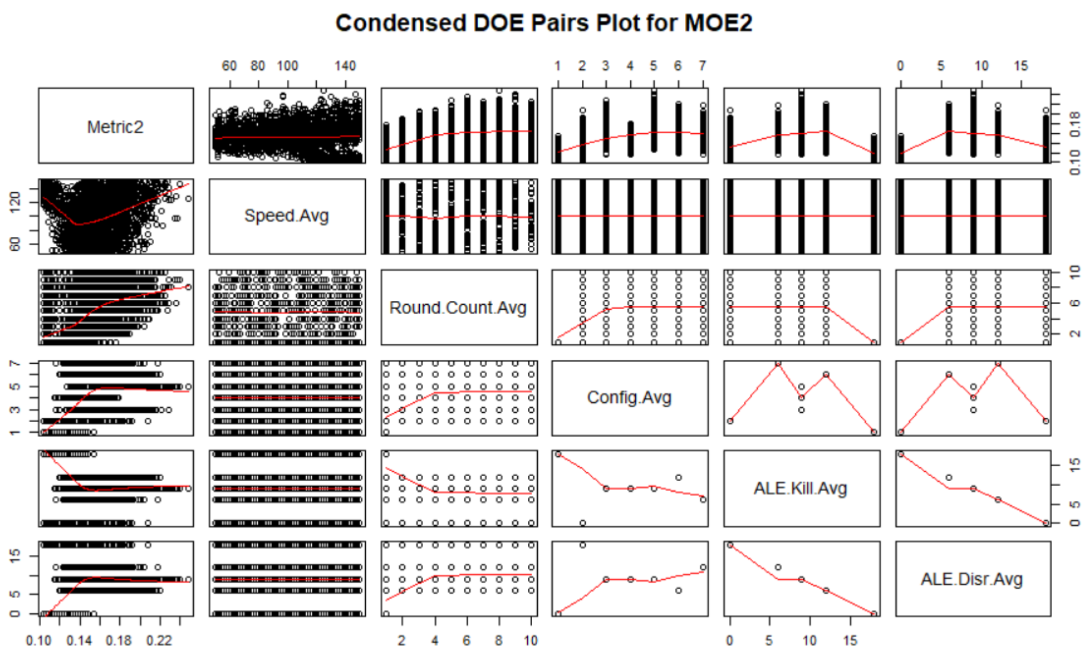


Figure 25. R Studio Screenshot of MOE2 Pairs Plot with Smoothers

To better enable the determination of critical range values for the most important characteristics, two and three-dimensional contour plots were created for each configuration and each MOE. On these plots, the number of rounds and speed were varied. The appendix shows these plots for all configurations and both MOEs, but Figure 26 shows a specific instance for the third configuration with MOE3. The contour plots were generated using the basic R Studio package, but the three-dimensional plots were created from the "plot3Drgl" library created by Karline Soetaert (2016) Additionally, the method of gradient coloring through a plot was derived from a Stackoverflow.com user "joran" (2012).

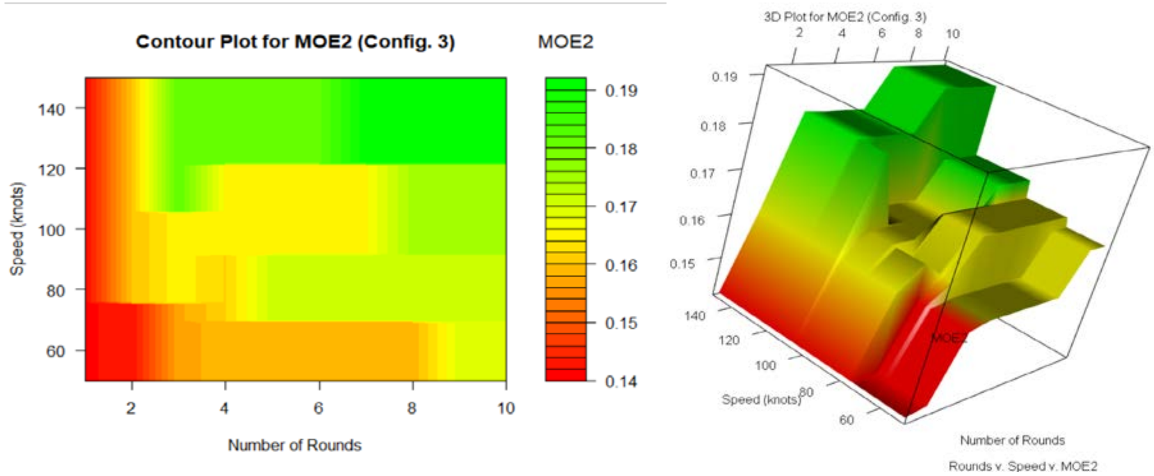


Figure 26. R Studio Screenshot of Contour and 3D Plot for Configuration 3 and MOE2

Just as with the linear regression analysis, certain instances provide for areas of focus outside of the extreme endpoints, suggesting that there could be a tradeoff with very little ramifications on performance. Certain configurations provide no benefit anywhere on the plot. This is specifically the case where there is a single type of ALE. This means that there is a critical importance to having a mixture of the types of ALE and a single type should not be sent into battle alone.

While each ALE configuration can provide case specific benefits, all MOE values for each MOE was summed across all configurations and plotted on a contour map. This provides a very quick and rough snapshot of areas of focus for the number of rounds and speed for each ALE and for each MOE. This is shown in Figure 27.

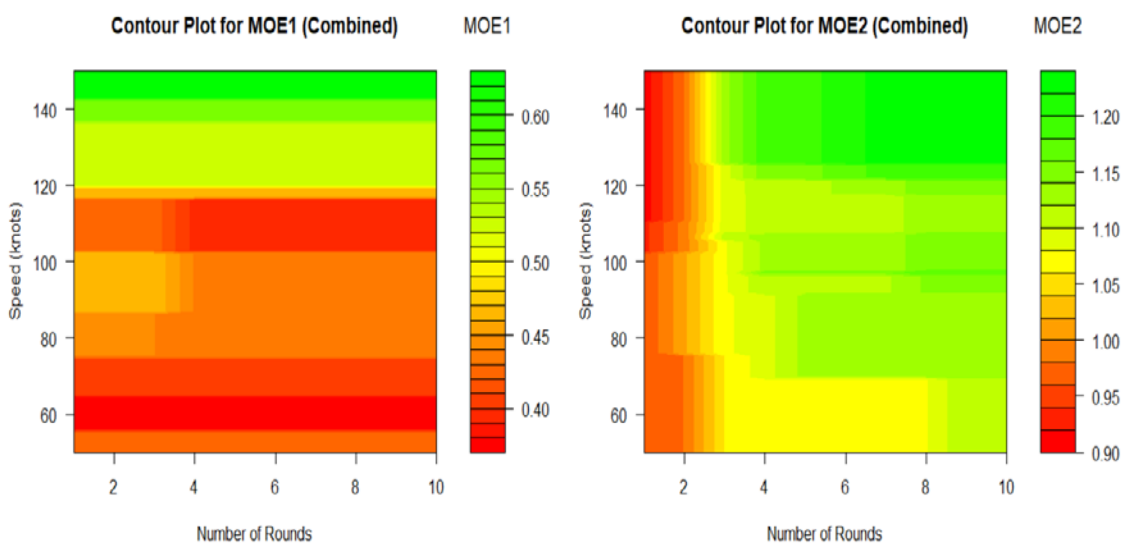


Figure 27. R Studio Screenshot of Contour Plot for Summarized MOE Metrics

The advantage of this method is that is provides quick recommendations for future study. For instance, if survivability is the most important aspect to designing the ALEs, speed is the most critical aspect and the vehicle should be designed to fly at a minimum of 135 knots, as evident from the MOE1 contour plot. On the other hand, if lethality is the primary objective, the vehicle is best designed to have at least three rounds and fly at a minimum of 130 knots, as shown from the MOE2 contour plot. However, reductions in speed provides only marginal losses to performance for MOE2.

There is a relationship between the MOEs. Despite the two metrics having different characteristic ranges that optimize their values, increases in one MOE has secondary benefits to the other MOE. In practical application, this makes sense. An ALE can usually only destroy the enemy if it is alive to shoot it, and shooting enemy assets also keeps the ALE alive. The pairs plot with smoothers in Figure 28 shows this seemingly linear relationship. The two MOEs have a correlation of .7759.

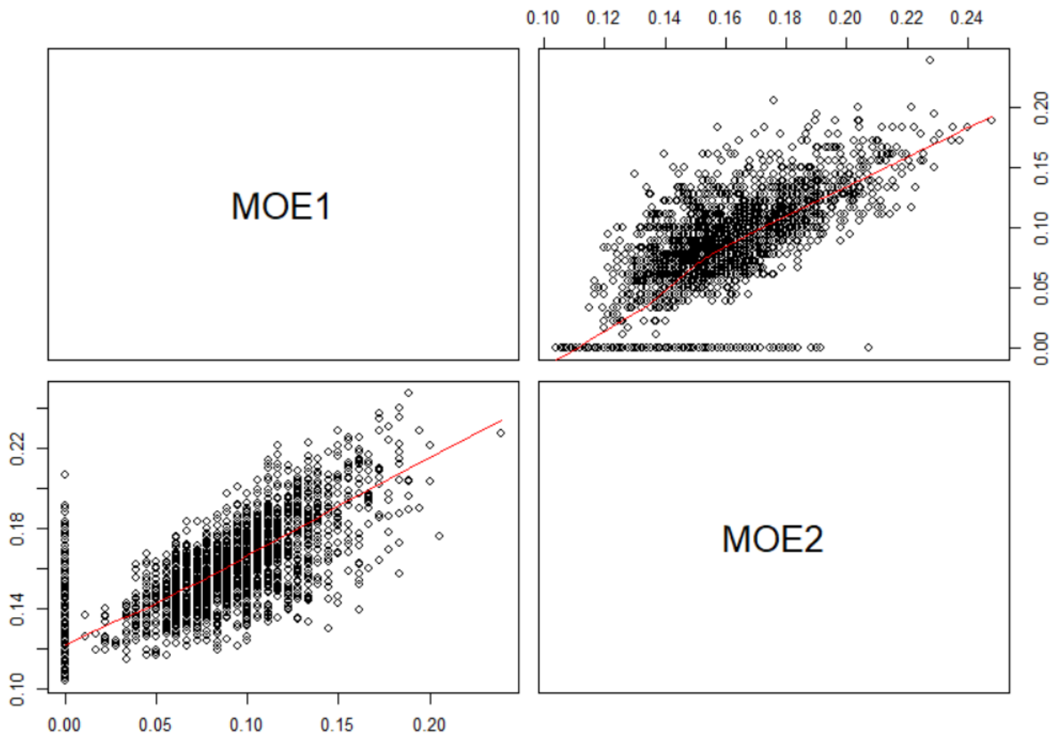


Figure 28. R Studio Screenshot of Pairs Plot Between MOEs with Smoothers

E. SUGGESTION FOR BEST CONFIGURATION

Visualization can certainly provide ranges for feasible and reasonably performing ALE, but an optimal configuration is the most difficult aspect to determine. The biggest reason for this is because, at such an early stage in development of the vehicle, it is currently unknown how important survival and lethality are relative to each other. Using an extremely basic method of giving MOE1 and MOE2 equal weights, a single configuration comes out at as maximizing this value. This configuration is provided in Table 15.

Table 15. Optimal Characteristics

Characteristic	Optimal Value
# Kill ALE	9
# Disrupt ALE	9
Speed	150 knots
Pk(near)	.9
Pk(far)	.16
Radar Distance	7.8 miles
Kill Distance	14.9 miles
Burst Radius	64.136 meters
# Rounds	5
Mean Time to Detection	1.281 seconds
Configuration	5

Many of the characteristics, such as speed, trend towards their maxim, as is reasonable to assume. But some of the values, such as the radar distance, kill distance, burst radius, and number of rounds are not at their maximum values. This is optimistic that a vehicle would work best outside of being the objectively best design; however, it also warrants additional study using higher resolution modeling tools. Additionally, having greater fidelity in the coefficients and importance of each MOE would provide an optimal solution with a more realistic framework. Depending on how heavily one MOE is weighed compared to another, this optimal set of characteristics can vary substantially, and even create a range of optimal characteristics.

The fifth configuration being the optimal configuration is likely for two reasons. First, it provides an even-split mixture of the two types of ALEs. Throughout analysis, having a mixture provides the best results to friendly forces. Second, because of the disrupt ALEs have more rounds then the kill ALEs, the kill ALEs may detect targets and the disrupt ALEs can quickly respond because of the type of formation they are in. The ALEs can provide cover to each other if they can detect but must rely on another platform for lethality.

In looking at the worst performers, there were a total of six configurations that provided no measurement in MOE1 or MOE2. While there is no clear pattern to predict or provide recommendations to how the ALE should not be designed, there is one item of note. Three of these six worst designs involved having all of one type of ALE, either all kill variants or all disrupt variants. Figures 19 through 21 show this to be the case. Configurations on the end, with completely imbalanced mixes of ALE variants perform poorly. This bolsters the argument to ensure that there is some mix of ALEs in any real-world combat scenario.

With regards to whether the vehicle should be recoverable or disposable, the ALE should be disposable, if at all possible. Based on the scenario and the ranges for the DOE, an average of 1.1799 ALEs survive each replication. This provides a survival rate of 6.55%. With such a low rate, it would be extremely cost ineffective if these vehicles were designed for multiple use cases. However, in other scenarios against a less formidable foe, a recoverable vehicle may be more realistic.

V. SUMMARY

In order to determine success, a quick glance back at the research questions reveal whether this study of the ALE provides actionable insight for the design of this platform. Success in these areas provide the basis for the creation of this future critical capability.

A. CRITICAL CHARACTERISTICS AND THEIR VALUES

1. Critical Variables, Interactions, and Optimal Values

The most important characteristics, regardless of whether the performance was modeled by a linear regression or a random forest, are the configuration type, speed, number of rounds, and the number of each type of ALE. The biggest interactions in the modeling of both MOEs are the interaction between the configuration type and the speed, and the interaction between configuration and number of rounds.

The optimal characteristics for the ALE is provided in Table 14, but, in general, the configuration can be summarized as a being in a evenly split formation between kill and disrupt ALEs, being really fast and extremely deadly up close, being able to detect and kill at moderate ranges, with a fairly modest number of rounds. However, this is predicated on equally weighted MOEs and can vary with MOE coefficient refinement.

2. Long-Term Survivability

With only 1.8 ALEs surviving a battle on average, the ALE should be designed to be disposable. If this is not the case, the vehicle should be used only in the most strategic of circumstances or have a significant number of decoys deployed in advance of its use. Other options include optimized routes to targets or deployment behind initial friendly force engagements, but these options may reduce the ALE's surveillance benefits.

3. Future Research Characteristic Ranges

Future research should focus first on the ALE configuration on the battlefield. It is important to refine distances between the types of ALEs and where they are arrayed in formations. The first configuration should focus on an even distribution of kill and disrupt

ALEs, and progress to other configurations from there. The next most important characteristic for research is speed. Maximum possible speeds should be explored first, and if this is unattainable, a speed of anything greater than 130 knots should be a goal. However, a range of speeds between 90 to 105 knots are a good starting point in terms of cost savings while still achieving reasonable performance. Finally, the minimum number of rounds on an ALE is ideally three. Therefore, future studies should look to incorporate two to five rounds in their research.

B. LOOKING BACK ON THE STUDY

Looking back on the study, there were some complications and issues that should be addressed. The biggest issue is that, not having personally created JDAFS, some of the vagueness of the results are on account of the system being much of a "black box." Certain characteristics and parameters can be modified, but much of the model is hard to interpret without an intimate knowledge of the underlying relationships between methods. Certain modeling aspects were also difficult to recreate or implement, such as the lack of the effects of terrain, the inability to precisely model disruption, or limitations on entities without dramatically affecting computer performance. The inability of setting the seed for the simulations also creates an issue with reproducibility. When combined with the limited number of runs capable in the model, these two issues are a cause for concern and should be addressed in future uses of JDAFS.

However, given the nature of the study and the fact that these initial insights will feed further analysis, these issues are not necessary shortfalls. Rather, they are initial limitations that can be overcome with a more time and resource intensive effort. This quick method of analysis did provide some critical insights that will likely save the government resources through the design and implementation of the ALE vehicles and the FARA ecosystem.

C. ADDITIONAL RESEARCH ASSISTANCE PROVIDED

In addition to providing the analysis in this study, I conducted a class on my methodology used in JDAFS that was recorded for TRAC – Monterey. Because JDAFS has not been used to this level with this many replications, the method was rather unique in combining so many different types of programming languages to solve an issue like this. The class had two major benefits. The first benefit was a way of instructing other analysts on how to carry on further research as necessary, particularly on classified systems. The second and more important benefit is a reinvigoration in using easily implemented models to support more complicated simulations. Not only will JDAFS likely be used in other studies, but success in this study can hopefully garner additional resources to validate these efforts and refine the product for the sponsor's needs.

D. LOOKING FORWARD TO FUTURE RESEARCH

Future research on this topic should start with implementing JDAFS and this methodology on a classified network against real-world threats. Doing this would either validate the design characteristics or refine assumptions required to conduct analysis in an unclassified environment. Due to the inability to access a classified network, this was not possible during this study.

The next step in additional research would be to narrow the characteristics' ranges and increase the number of levels in a future DOE. Doing this could provide insight into sensitivity analysis for the characteristics and determine if other parameters had more subtle effects not picked up in this research.

Based on watching more than a few of the simulations in the JDAFS graphical interface, it becomes apparent that one of the biggest reasons for success was the introduction of decoys into the system. The IADS assets spent many of their allocated rounds shooting at the drones. Decreasing the number of drones on the battlefield would have been even more detrimental to the ALE survival rates. Varying decoy distances and quantity would also provide critical information to what should be done in terms of tactics on the battlefield.

The final recommendation for future research is to take this unclassified scenario and DOE and run it through a different model. The first suggested model would be to use Map Aware Non-Uniform Automata (Lauren et al. 2002). This model allows for modifications to terrain, unit attitudes, and armor thickness. Having a second model validate the results from JDAFS, or provide alternate recommendations, would provide additional information to decision makers and analysts.

E. CONCLUSION

With the proper characteristics, the ALE can provide immense benefits to friendly forces by disrupting the enemy’s ability to control the airspace over an engagement area. However, with the current design from this study, this effect is limited to the airspace and provides negligible direct benefits to the Soldier on the ground. While the ALE has an important role, its inability to survive the engagement means it should be produced and procured as inexpensively as possible or be used in limited scenarios. This analysis will shortly be used to inform an AWARS study—with careful planning and a good design, friendly forces will see the benefits sooner rather than later.

APPENDIX. ADDITIONAL VISUALIZATIONS

Figure 29 and Figure 30 are full partition trees for the predictive models for MOE1 and MOE2. Abbreviated forms of these forests are displayed in Chapter IV with the most important characteristics, but these trees drill down to within one standard deviation of the best possible number of splits.

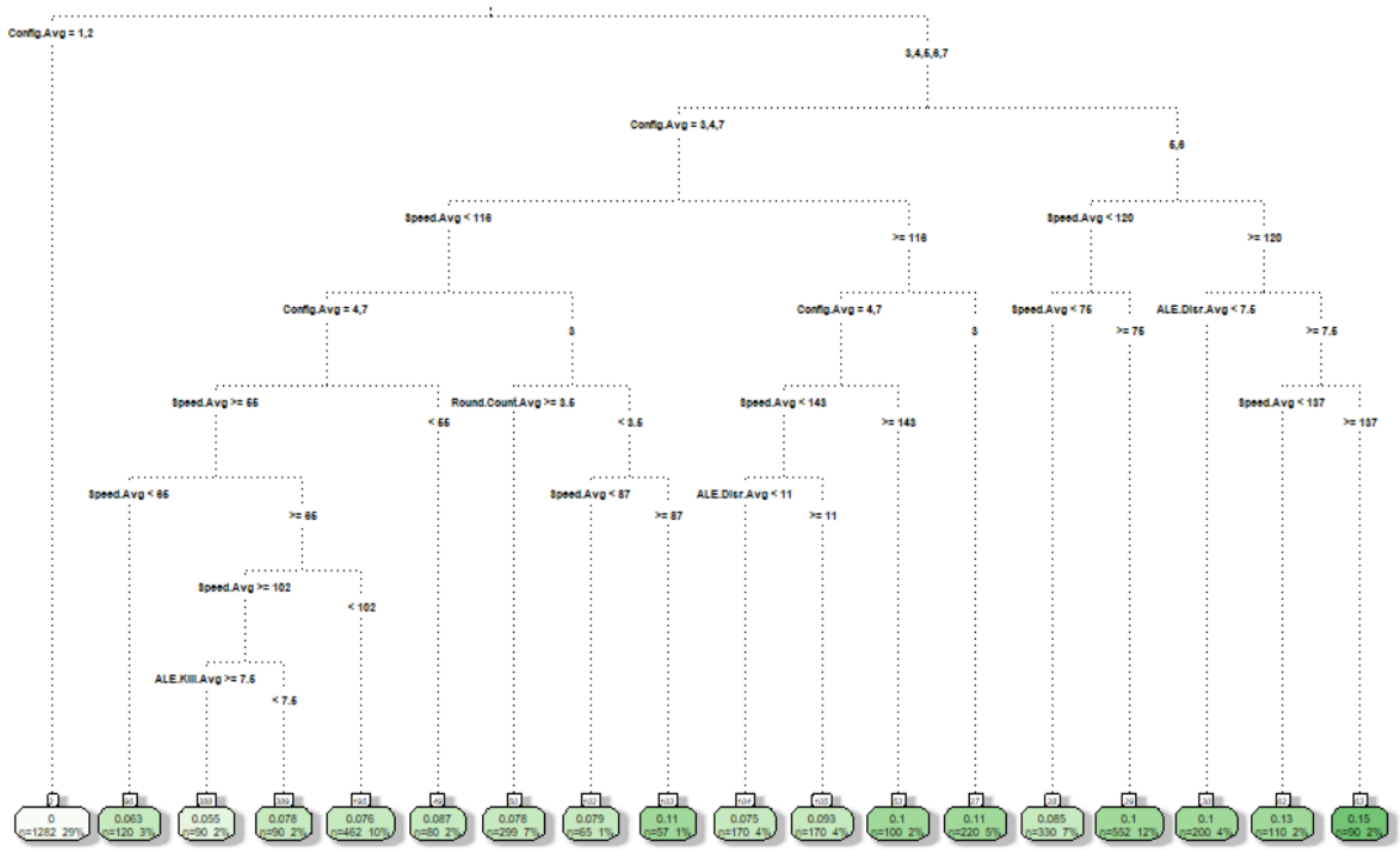
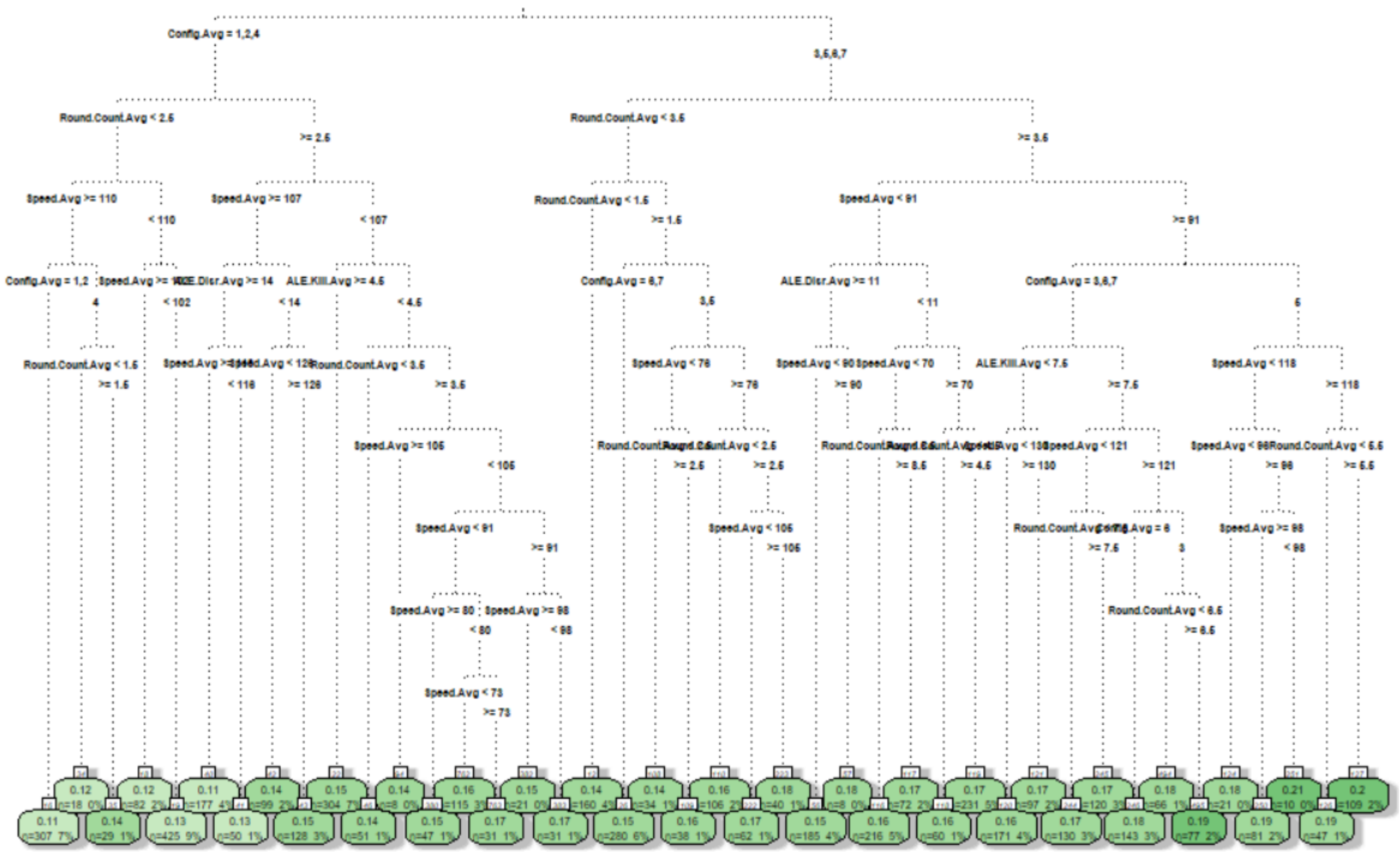


Figure 29. Full Random Forest for MOE1



Rattle 2020-May-15 22:12:27 Fab

Figure 30. Full Random Forest for MOE2

Figure 31 and Figure 32 show the pairs plot for the DOE characteristics. These are expanded versions from the pairs plots in Chapter IV with all the variables used for the ALEs. These plots have smoothers that show how the interactions between two variables contribute to predicting the MOEs. Most additional interactions from the figures in Chapter IV do not increase the predictive nature of the model, as per the plots in Figures 31 and 32.

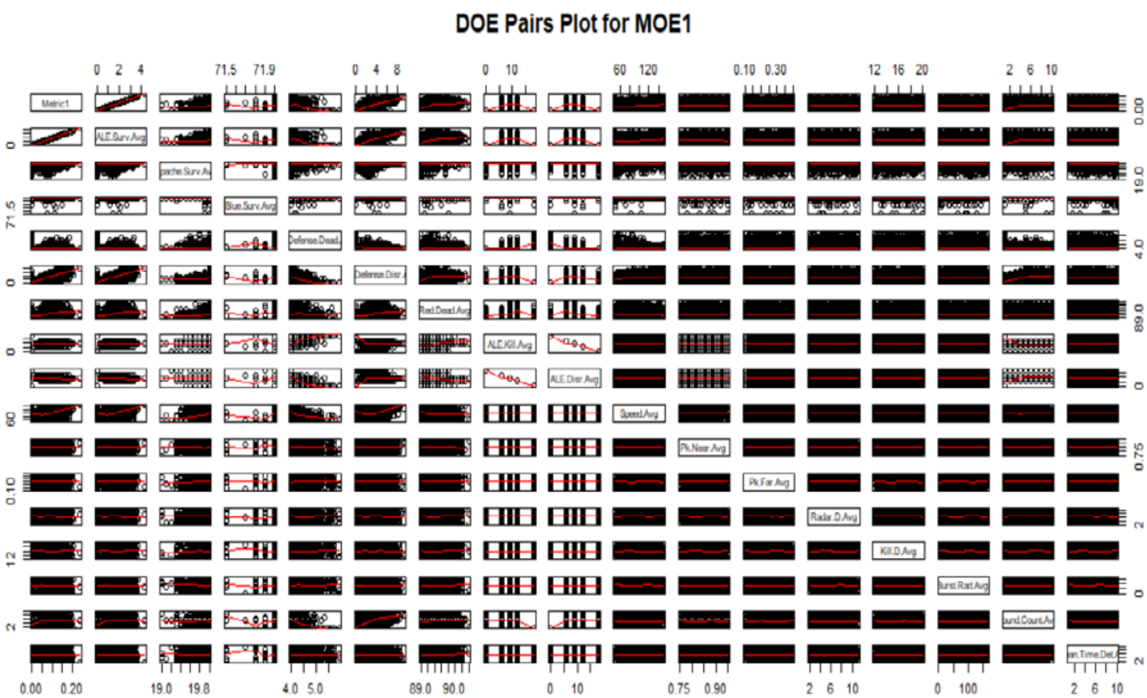


Figure 31. R Studio Screenshot of MOE1 Pairs Plot with Smoothers

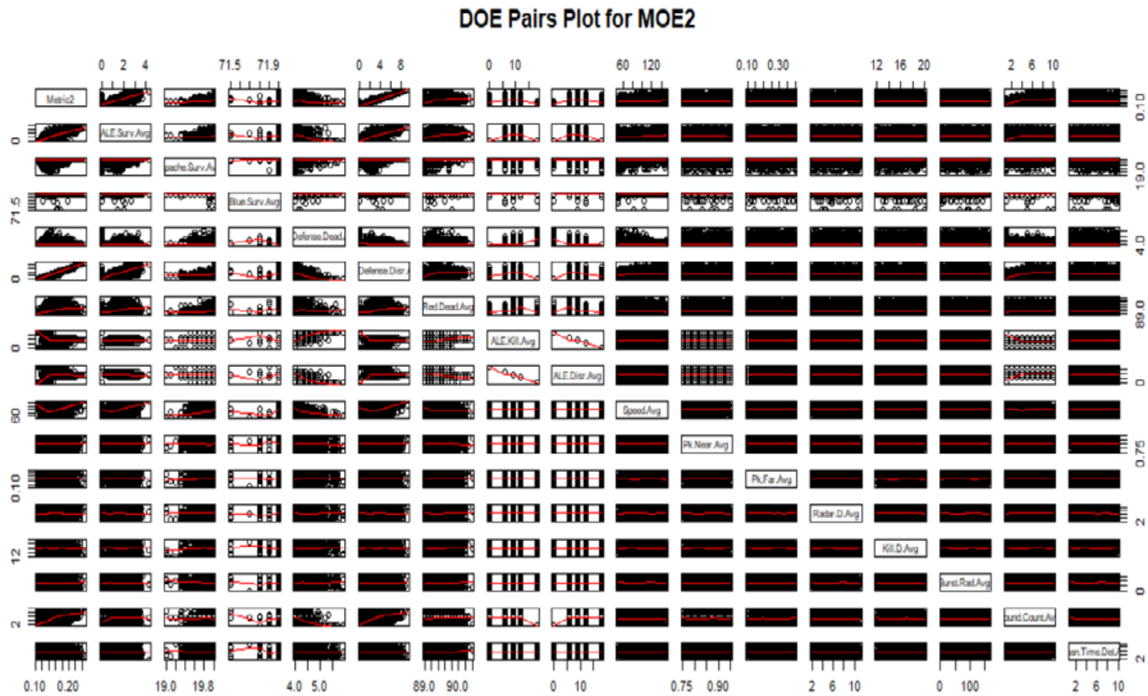


Figure 32. R Studio Screenshot of MOE2 Pairs Plot with Smoothers

Figure 33 through Figure 46 show side by side two- and three-dimensional plots for each configuration for each MOE. With an assumption that configuration can adequately represent the number of kill and disrupt ALE, the top five most important characteristics to the random forests are represented. This assumption of correlation between the configuration and ALEs by type is fairly lose, but in order to provide visual representation for tradeoff analysis, this assumption was necessary. Figure 32 and Figure 33 show only two-dimensional measures. What this means is, based on MOE1, having a single type of ALE does not do much of anything to improve or hinder ALE performance. The real benefit is mixing ALE types within a scenario. It is important to show each of these plots because they provide a picture of how the configurations affect performance and how combining them can provide a much better picture for recommendations on optimal speed and number of rounds for the ALE.

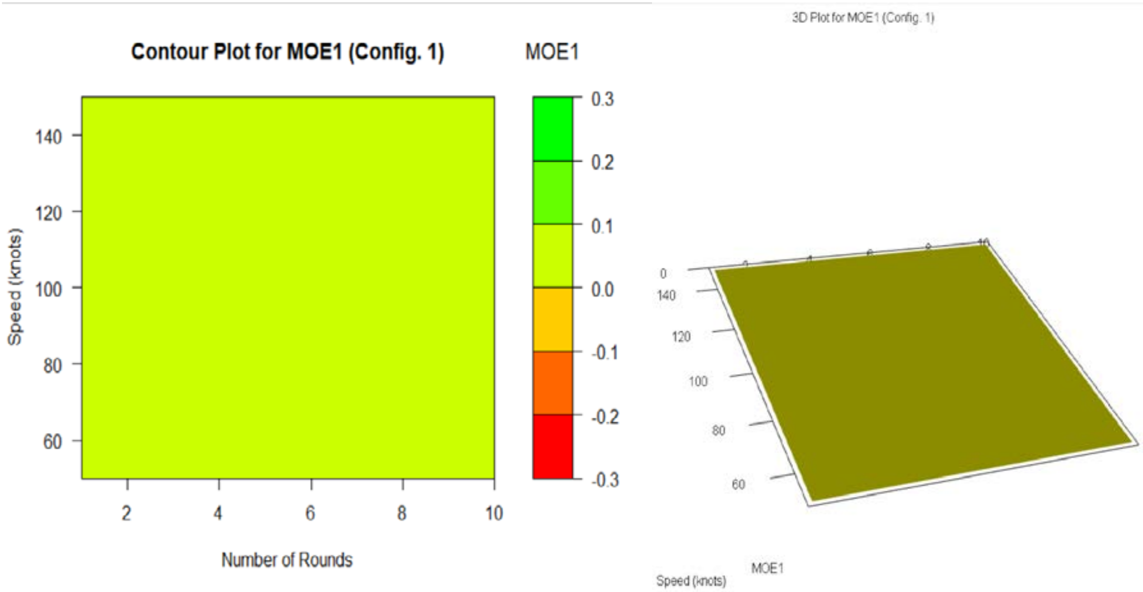


Figure 33. R Studio Screenshot of Contour and 3D Plot for Configuration 1 and MOE1

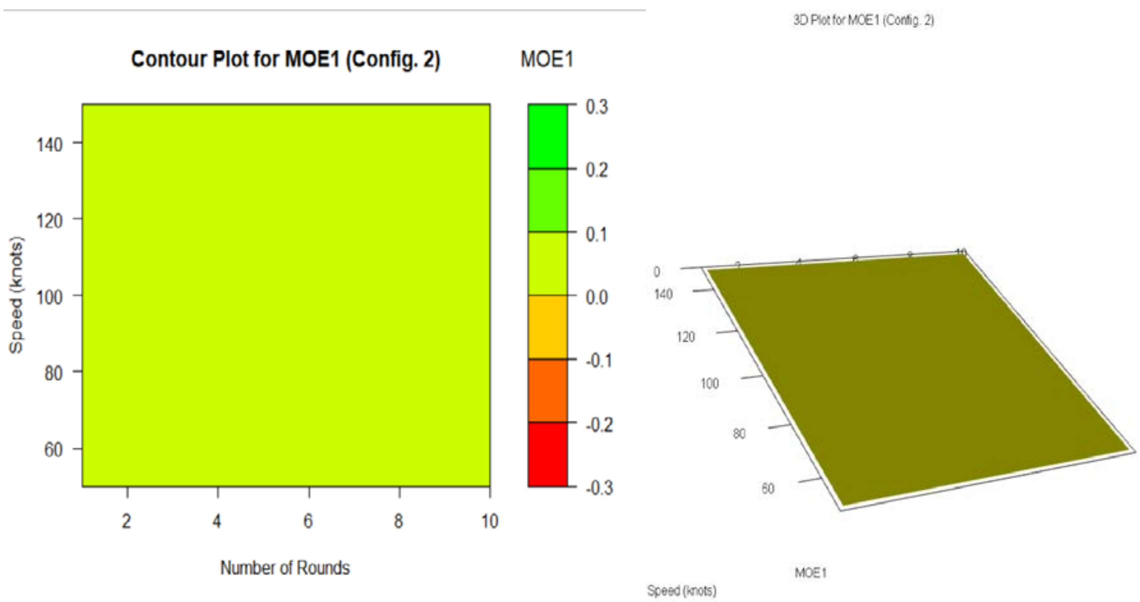


Figure 34. R Studio Screenshot of Contour and 3D Plot for Configuration 2 and MOE1

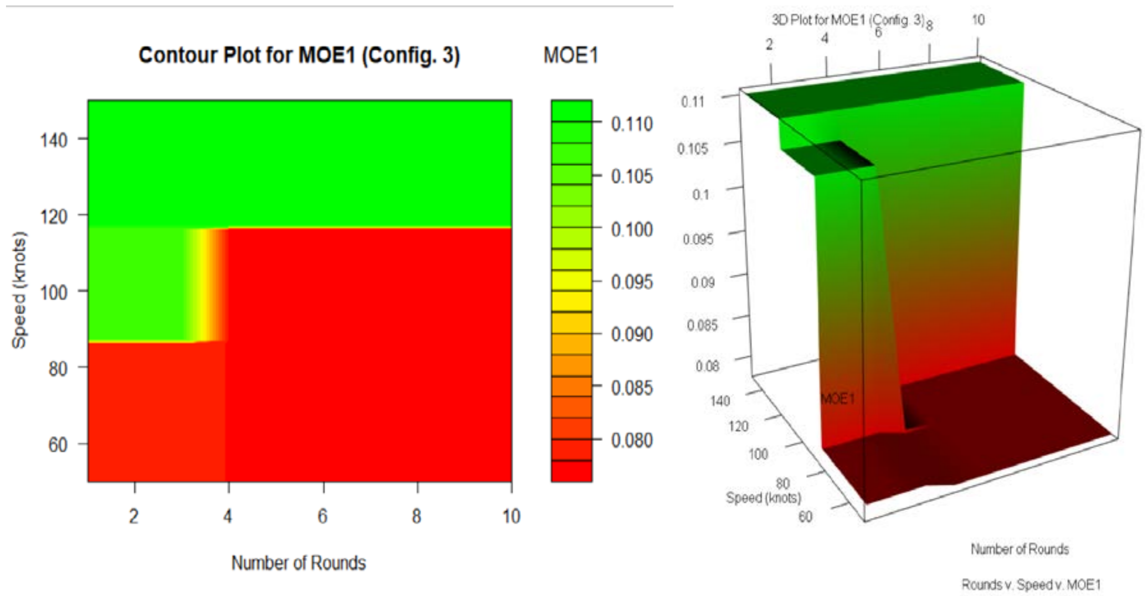


Figure 35. R Studio Screenshot of Contour and 3D Plot for Configuration 3 and MOE1

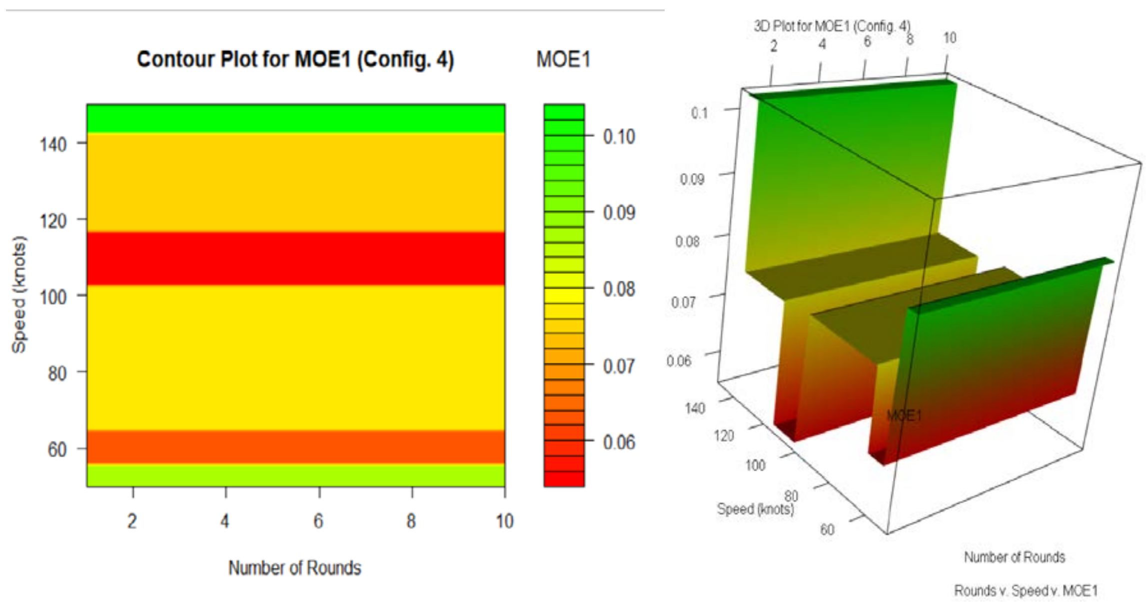


Figure 36. R Studio Screenshot of Contour and 3D Plot for Configuration 4 and MOE1

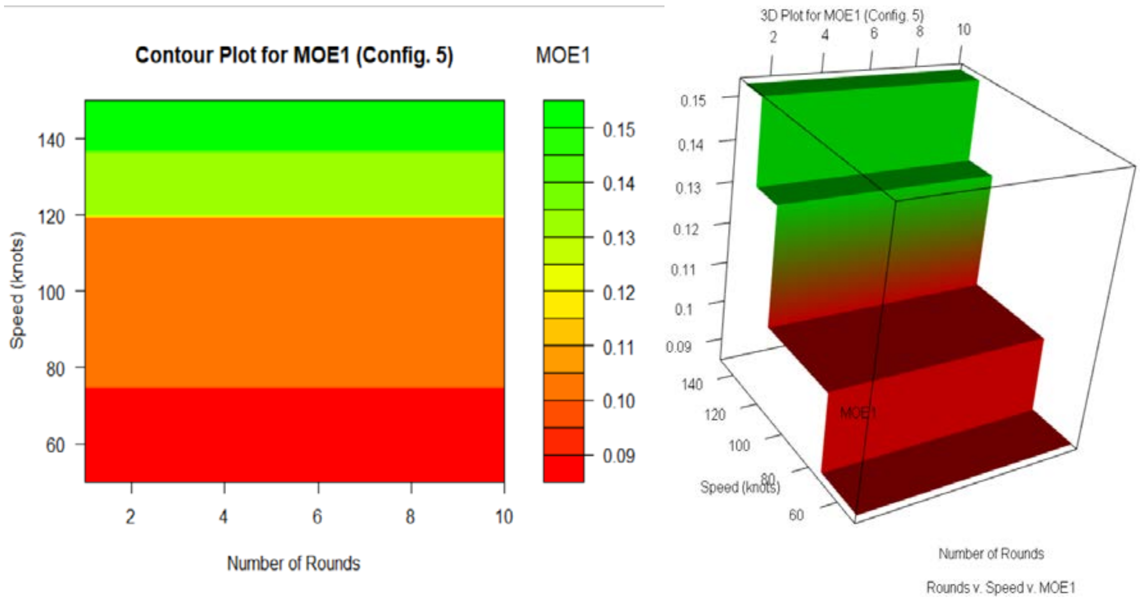


Figure 37. R Studio Screenshot of Contour and 3D Plot for Configuration 5 and MOE1

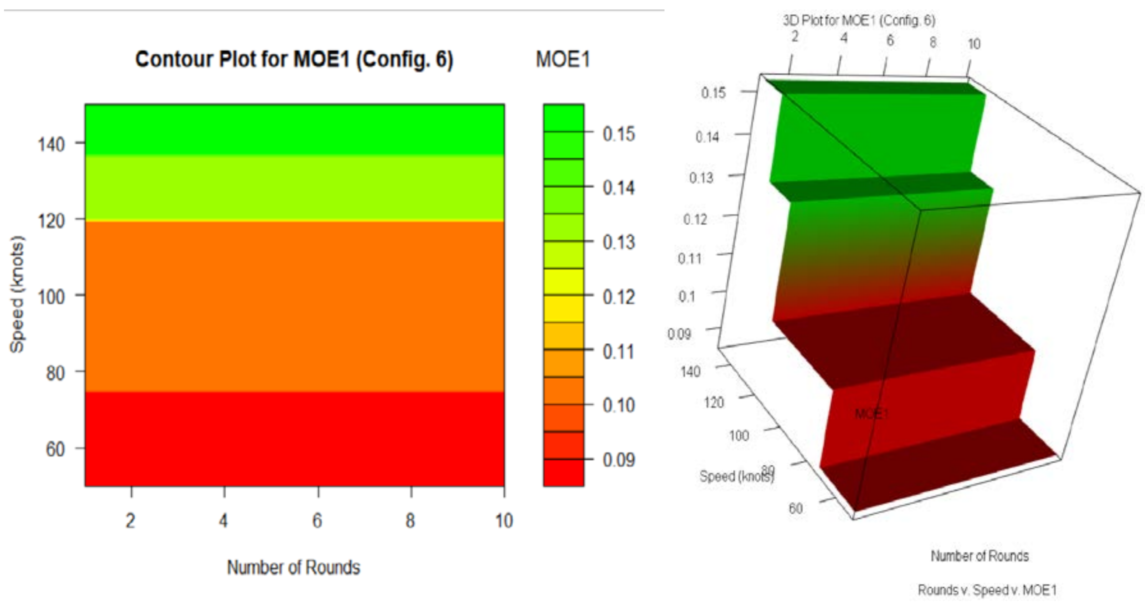


Figure 38. R Studio Screenshot of Contour and 3D Plot for Configuration 6 and MOE1

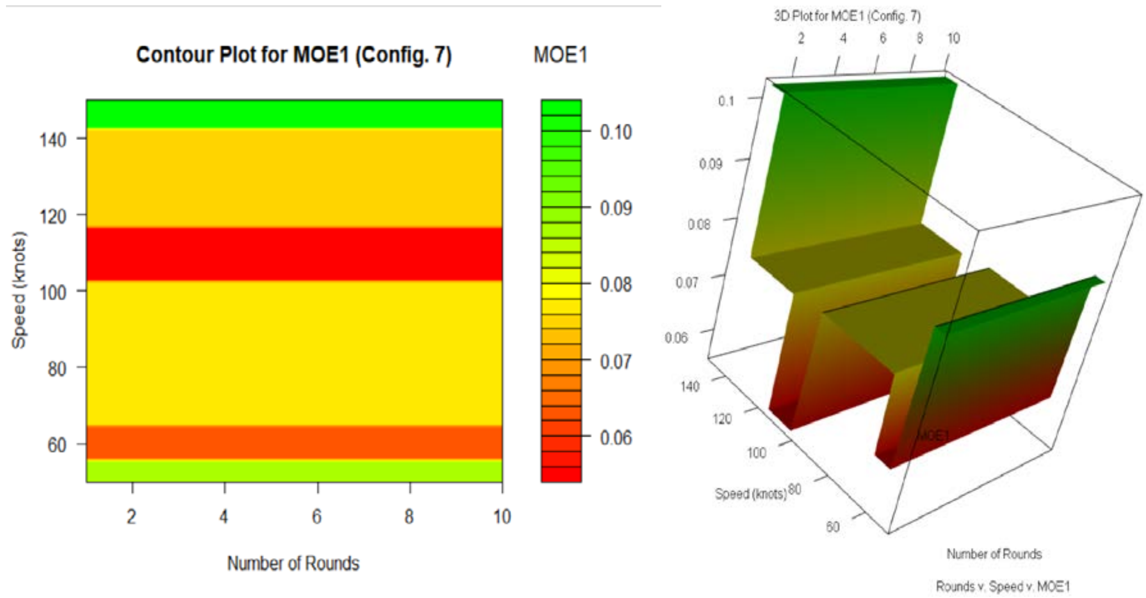


Figure 39. R Studio Screenshot of Contour and 3D Plot for Configuration 7 and MOE1

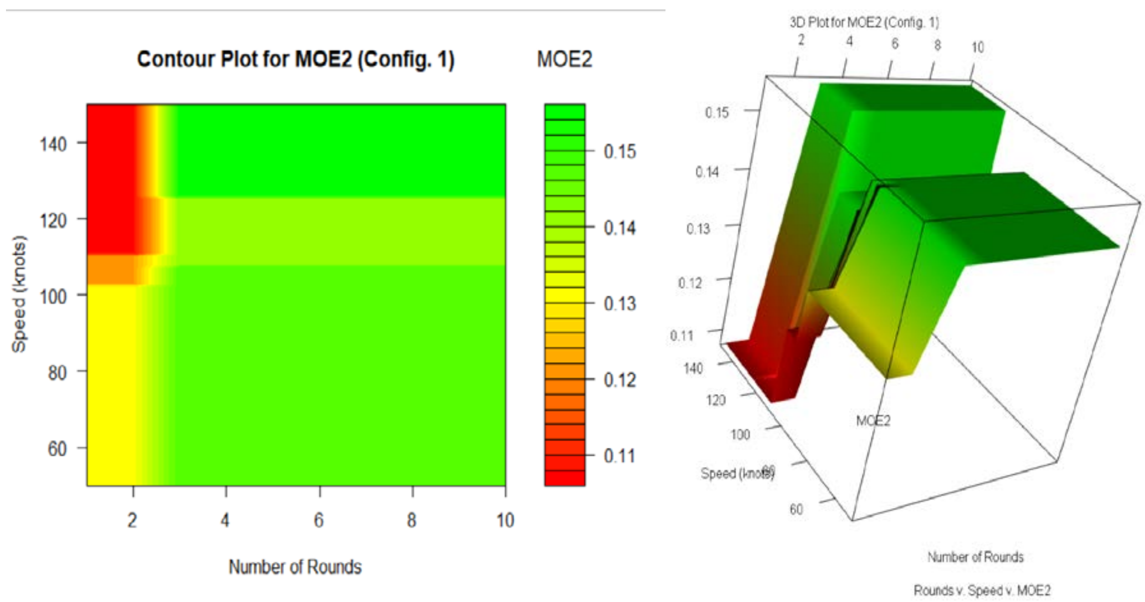


Figure 40. R Studio Screenshot of Contour and 3D Plot for Configuration 1 and MOE2

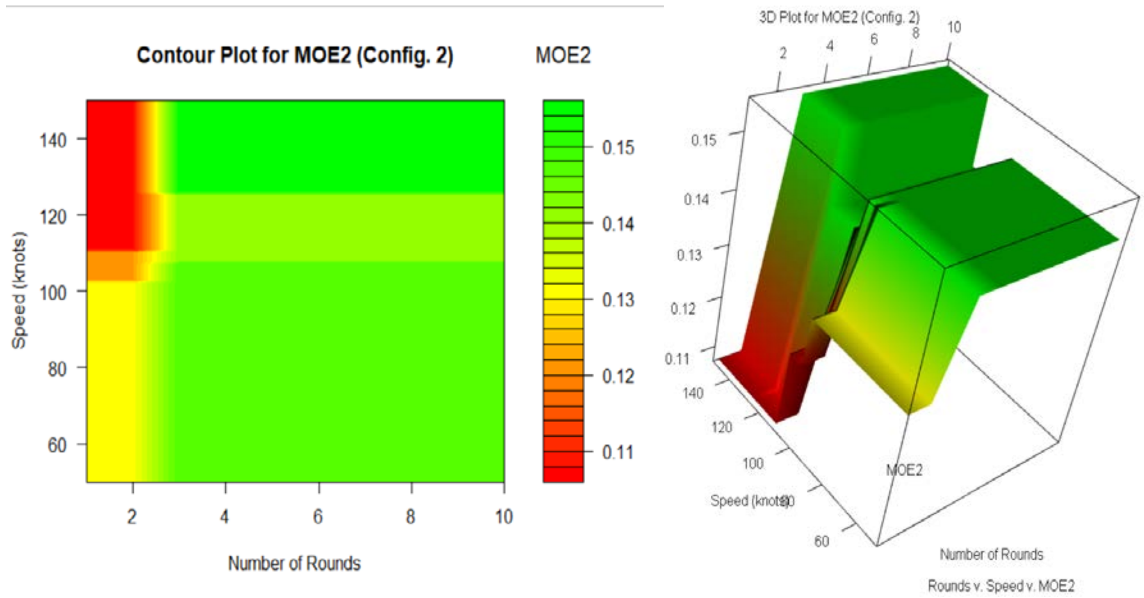


Figure 41. R Studio Screenshot of Contour and 3D Plot for Configuration 2 and MOE2

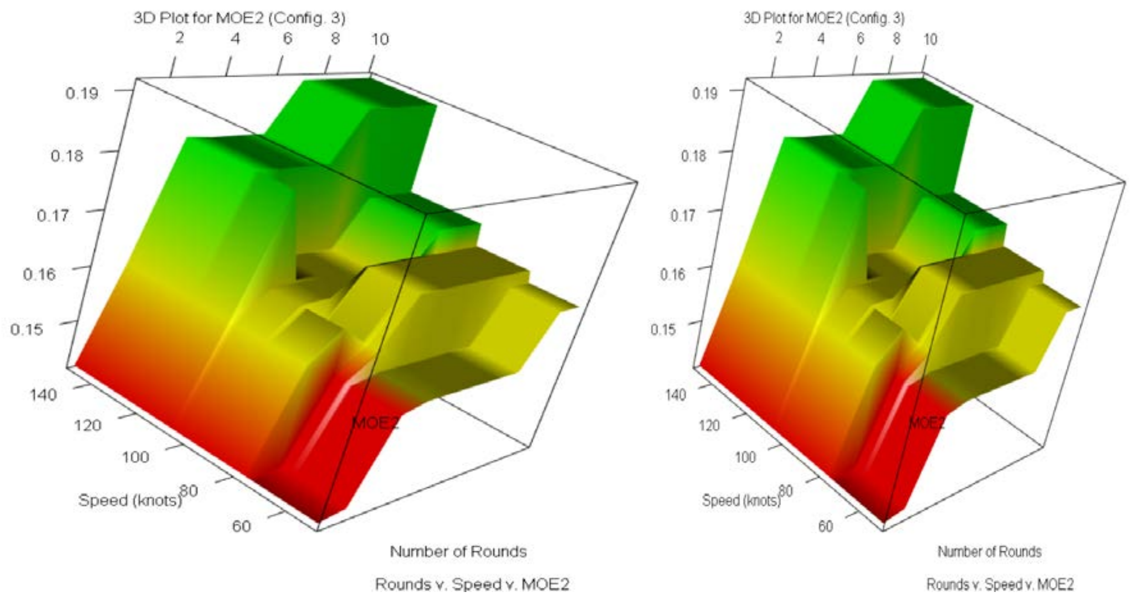


Figure 42. R Studio Screenshot of Contour and 3D Plot for Configuration 3 and MOE2

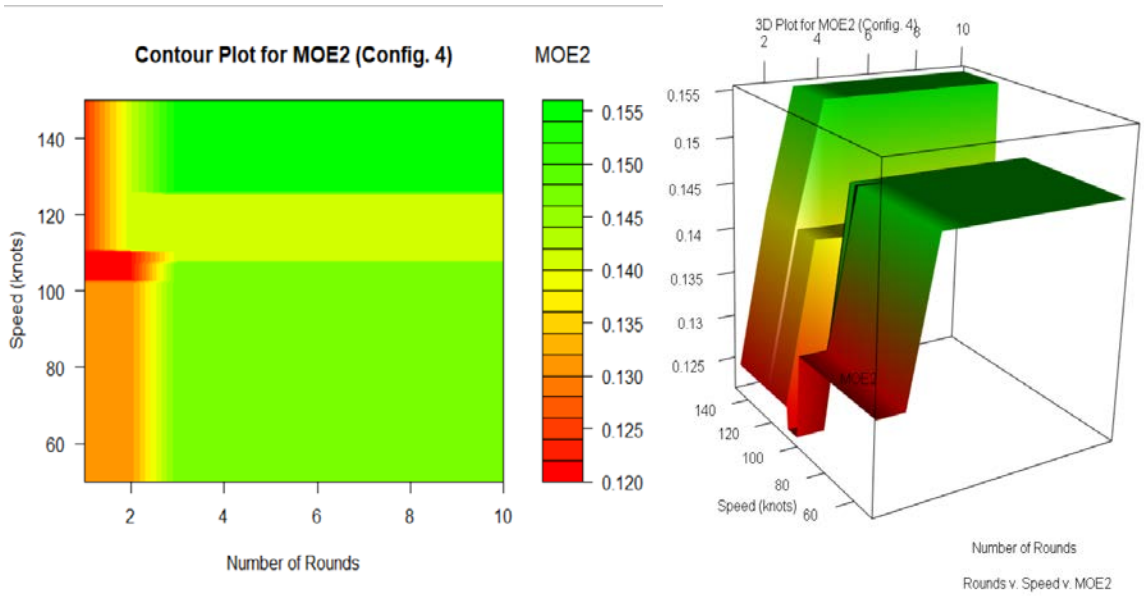


Figure 43. R Studio Screenshot of Contour and 3D Plot for Configuration 4 and MOE2

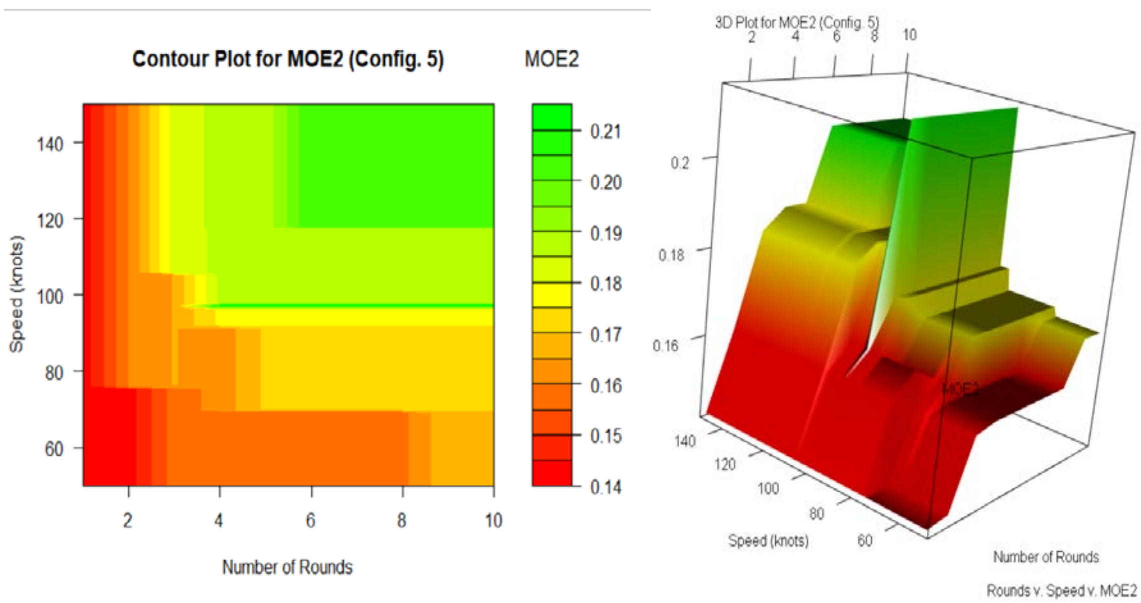


Figure 44. R Studio Screenshot of Contour and 3D Plot for Configuration 5 and MOE2

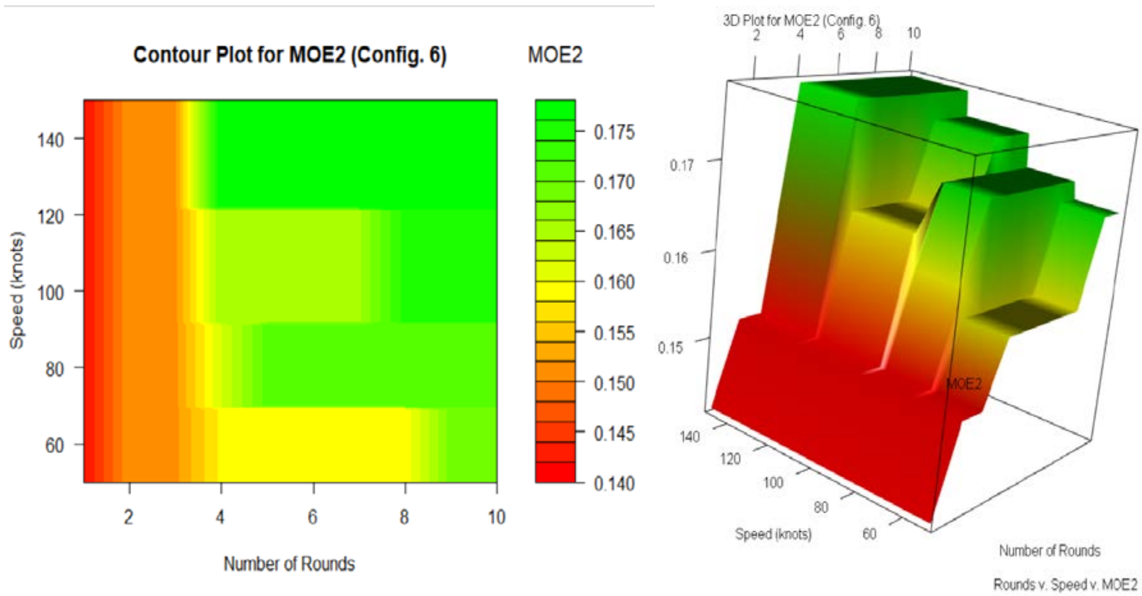


Figure 45. R Studio Screenshot of Contour and 3D Plot for Configuration 6 and MOE2

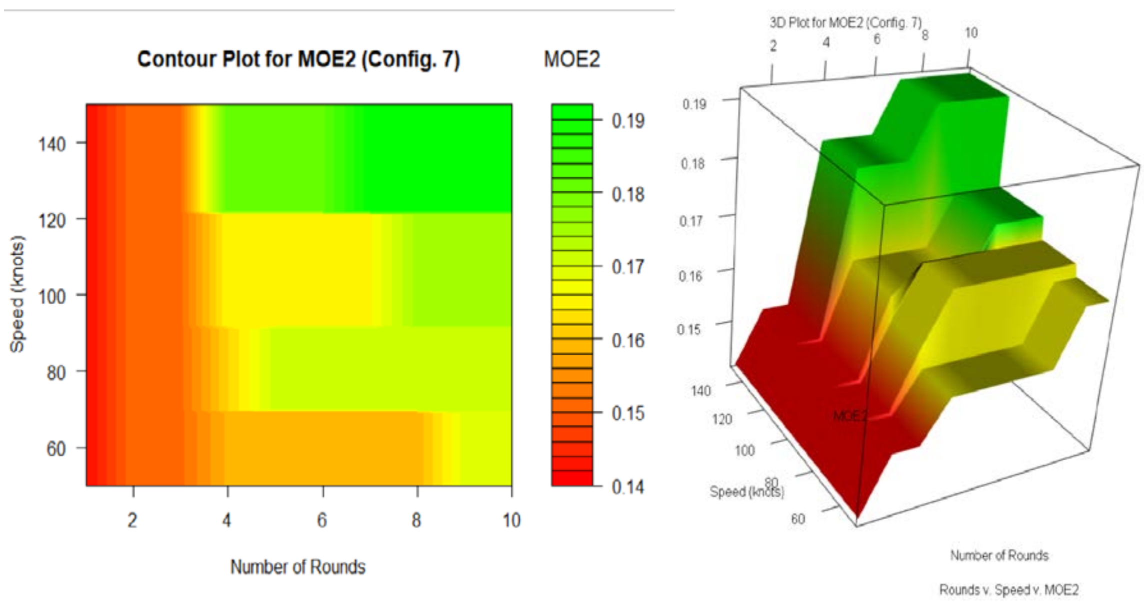


Figure 46. R Studio Screenshot of Contour and 3D Plot for Configuration 7 and MOE2

LIST OF REFERENCES

Anderson BL (2011) Changing the paradigm simulation, a method of first resort. Master's thesis, Naval Postgraduate School, Monterey, CA. http://hdl.handle.net/10945/5481.

BananaRepublic (2015) Win 10 ODBC MS Access ACCDB not available. Microsoft Answers. Accessed January 15, 2020, https://answers.microsoft.com/en-us/msoffice/forum/all/win-10-odbc-ms-access-accdb-not-available/c76ed684-aefa-4c68-8c62-4cb79cd7dc30.

Cioppa TJ, Lucas TW (2007) Efficient nearly orthogonal and space-filling Latin hypercubes. *Technometrics* (1), 45–55.

Faraway JJ (2015) *Linear Models with R* (CRC Press, Boca Raton, FL).

Freye JT (2007) Design of experiment analysis for the Joint Dynamic Allocation of Fires and Sensors (JDAFS) simulation. Master's thesis, Naval Postgraduate School, Monterey, CA. http://hdl.handle.net/10945/3365.

Grau LW, Bartles CK (2016) *The Russian Way of War: Force Structure, Tactics, and Modernization of the Russian Ground Forces*. (Foreign Military Studies Office, KS).

Havens ME(2002) Dynamic allocation of fires and sensors. Master's thesis, Naval Postgraduate School, Monterey, CA. http://hdl.handle.net/10945/5109.

Hester J, Wickham H (2020) odbc: Connect to ODB Compatible Databases (using the DBI Interface). R package version 1.2.2. https://CRAN.R-project.org/package=odbc.

JMP Pro, Version 15.0.0 (SAS Institute Inc., Cary, NC) 1989–2019.

Johnson E, Ahner D, Buss A, Stock K (2008) Team 10: Joint Dynamic Allocation of Fires and Sensors (JDAFS) Joint Starting Conditions Analysis, *Proc of the International Data Farming Workshop 16* (Naval Postgraduate School, Monterey, CA), http://hdl.handle.net/10945/35644.

joran (2012) Colour points in a plot differently depending on a vector of values. Accessed May 10, 2020, https://stackoverflow.com/questioins/9946630/colour-points-in-a-plot-differently-depending-on-a-vector-of-values.

Koyak R (2019) Unit 1: Classification and Regression Trees. Class notes, OA4106: Advanced Data Analysis, Naval Postgraduate School, August 20, 2019.

Lauren MK, Stephen RT (2002) Map-aware non-uniform automata (MANA) – a New Zealand approach to scenario modelling. *Journal of Battlefield Technology* 5(1), https://pdfs.semanticscholar.org/1e16/ 8c5e9c3d2748d5648166ba7fa40228f1d01c.pdf.

Naval Postgraduate School (2018) A Gentle Introduction to Hamming. Haming.uc.nps.edu. Accessed April 1, 2020, https://hamming.uc.nps.edu/Gentle-Intro-to-Hamming.html.

Programming on nielsenmark.us (2018) Setting up an ODBC connection with MS SQL Server on Windows. R-bloggers. Accessed January 15, 2020, https://www.r-bloggers.com/setting-up-an-odbc-connection-with-ms-sql-server-on-windows/.

R CORE Team (2020) R: A language and environment for statistical computing. (R Foundation for Statistical Computing, Vienna, Austria), https://www.R-project.org/.

Ripley B, Lapsley M (2019). RODBC: ODBC Database Access. R package version 1.3-16. https://CRAN/R-project.org/package=RODBC.

Ruby, Version 2.6.3p62 (2019) https://ruby-lang.org/en/.

Sanchez PJ (2018) Datafarming Ruby Gem, https://rubygems.org/gems/datafarming.

Soetaert K (2016). Plot3Drgl: Plotting Multi-Dimensional Data – Using ‘rgl’. R package version 1.0.1. https://CRAN.R-project.org/package=plot3Drgl.

Tate S (2020) Future Vertical Lift: Army selects Future Attack Reconnaissance Aircraft Prototype performers. army.mil. Accessed on 28 April 2020, https://www.army.mil/article/234002/ future_vertical_lift_army_selects_future_attack_reconnaissance_aircraft_prototype_performers.

Therneau T, Atkinson, B (2019). Rpart: Recursive Partitioning and Regression Trees. R package version 4.1-15. https://CRAN.R-project.org/package=rpart.

TRAC (2020a) Advanced Warfighting Simulation (AWARS). Accessed April 30, 2020, https://www.trac.army.mil/20150910%20AWARS.pdf.

TRAC (2020b) COMBATXXI Accessed April 30, 2020, https://www.trac.army.mil/ COMBATXXI.pdf.

U.S. Army Acquisition Support Center (2018) Weapon System Handbook 2008. asc.army.mil. Accessed November 3, 2019, https://asc.army.mil/docs/wsh2/2018-wsh.pdf.

U.S. Army, Army Training Network (2019) Operational Environment / OPFOR Publications. atn.army.mil. Accessed November 3, 2019, https://atn.army.mil/tradoc-g2/operational-environment-opfor-publications/opfor-threat-force-structure.

U.S. Army, Army Training Network (2019) Worldwide Equipment Guide (WEG) Publications. atn.army.mil. Accessed November 3, 2019, https://atn.army.mil/tradoc-g2/operational-environment-opfor-publications/worldwide-equipment-guide-(weg).

U.S. Army Directorate of Force Management (2019) Force Management System. Accessed November 3, 2019, https://fmsweb.fms.army.mil/.

Wickham H (2019). Stringr: Simple, Consistent Wrappers for Common String Operations. R package version 1.4.0. https://CRAN.R-project.org/package=stringr.

Wickham H, Hester J, Ooms J (2019) xml2: Parse XML. R package version 1.2.2. https://CRAN.R-project.org/package=xml2.

Williams GJ (2011). *Data Mining with Rattle and R: The Art of Excavating Data for Knowledge Discovery, Use R!* (Springer, New York*).*

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California