

Methodology for the Selection of Pivotal Features in AFSIM  
Meta-Markov Models

Thesis

Presented in Partial Fulfillment of the Requirements for the Degree Master  
of Science in the Graduate School of The Ohio State University

By

James L. Enders, B.S.

Graduate Program in Department of Mechanical and Aerospace Engineering

The Ohio State University

May 8, 2023

Thesis Committee:  
Shawn Midlam-Mohler, Advisor  
Clifford Whitfield  
Nicholas Hanlon  
David Hillstrom

Distribution Statement A: Approved for public release. Distribution is unlimited. Case #:  
AFRL-2023-2188

© Copyright by  
James L. Enders  
2023

---

# ABSTRACT

Digital wargames have been used for over half-a-century to better understand the combat environment. From the testing of new strategies and tactics, to testing new technologies in low cost environments before acquisition, digital, and more importantly analytical wargames, are a key asset in any armed forces arsenal. There is a gap in this capability though. Current forms of wargames focus on quantifying the downstream effects of new ideas and technology. This leaves the explanation for how this happened to be described through intuition. Having the ability to statistically prove these intuitions would be invaluable. This is a difficult task to perform given the highly complex, chaotic, and layered system-of-systems that are required to accurately reflect the physical world in the digital world. Thus, a methodology is proposed to provide a means to down select the necessary features to be recorded and modelled. The Advanced Framework for Simulation, Integration, and Modelling (AFSIM) software tool was used to implement and simulate the scenario at hand. The kill chain framework is utilized to craft the recorded features to ensure the necessary information is encoded in the meta-model. Upon collection, features are aggregated into a meta-Markov model representing the probability of transitions between different sets of features over each run. Subsequent analysis can then be performed using centrality measures to identify points in the simulations that could be of interest to analyst.

Dedicated to those I love. Without you, none of this is possible.

---

## ACKNOWLEDGMENTS

I would like to acknowledge those around me that have helped support me up to and through this process. To my wife, thank you for being understanding of the long days and nights that invariably effected you. Your everlasting love was pivotal in being able to complete this work. To David, I may have been one of your first mentees, but you did one dang-up job I'd say. Thank you. To Shawn, and Cliff, your insights have been vital in pointing out the deficiencies and making me into a better engineer and scientist. Nicholas, without your guidance in the realm of AFSIM, none of this would be possible. Finally, my friends, you were amazing rubber duckies, a thousand thank yous.

---

## VITA

May 2018 .....	Associate of Arts, OSU
June 2018 .....	High School Diploma, PHS
May 2022 .....	Bachelors of Science, OSU

### **Fields of Study**

Major Field: Aerospace Engineering

# Contents

<b>Abstract</b> . . . . .	iii
<b>Dedication</b> . . . . .	iv
<b>Acknowledgments</b> . . . . .	v
<b>Vita</b> . . . . .	vi
<b>1 Introduction/Motivation</b> . . . . .	1
1.1 Digital Engineering . . . . .	1
1.2 Trade Space Analyses . . . . .	2
1.3 Levels of Scale/Resolution . . . . .	4
1.4 Wargaming . . . . .	7
<b>2 Background</b> . . . . .	9
2.1 Wargames: Why . . . . .	9
2.2 AFSIM . . . . .	12
2.3 Convergence Monitoring . . . . .	16
2.4 Markov Chains . . . . .	20
<b>3 Research Opportunities</b> . . . . .	24
3.1 Previous Work . . . . .	24
3.2 Current Inadequacies . . . . .	25
3.3 Problem to Solve . . . . .	27
<b>4 What to Model</b> . . . . .	28
4.1 The Past . . . . .	28
4.2 Kill Chain (F2T2EA) . . . . .	34
4.3 What Have We Learned . . . . .	37
4.4 Implementation . . . . .	38
4.5 Scalability . . . . .	40
<b>5 How to Identify</b> . . . . .	46
5.1 What Makes Pivotal . . . . .	46
5.2 History of Node Identification . . . . .	47
5.3 Example Usages . . . . .	50
5.4 Building the Metrics . . . . .	56

5.5	Outcome Score . . . . .	62
5.6	Combining the Metrics . . . . .	63
<b>6</b>	<b>Experiment . . . . .</b>	<b>65</b>
6.1	The Scenario . . . . .	65
6.2	Convergence Results . . . . .	74
6.3	Graphs of Interest . . . . .	78
6.4	Determining Viability . . . . .	83
6.5	Most Probable Paths . . . . .	86
6.6	Application of Metrics . . . . .	88
6.7	Score Analyzation . . . . .	91
6.8	Key Nodes . . . . .	98
<b>7</b>	<b>Conclusions . . . . .</b>	<b>102</b>
7.1	Results . . . . .	102
7.2	Future Work . . . . .	103

# List of Figures

1	Scale/Resolution Pyramid . . . . .	5
2	AFSIM Architectural Elements . . . . .	14
3	Tasker . . . . .	15
4	Example Behavior Tree . . . . .	16
5	Running Average of Two Die Summation Game . . . . .	18
6	Plotted Residual of Confidence Interval . . . . .	20
7	Variations in permutations . . . . .	42
8	Rook vs Hypercube Graph . . . . .	43
9	Curve Fit of Coefficient $B$ . . . . .	44
10	Progress of Scientific Techniques . . . . .	48
11	Example graph of the set $R$ . . . . .	60
12	Legend for the Disposition of Forces . . . . .	66
13	Dispositions for HVAA DCA Scenario . . . . .	66
14	Disposition of HVAA Escort Scenario . . . . .	68
15	One vs One and Two vs Two Dispositions . . . . .	68
16	Sample Rate Convergence Study . . . . .	73
17	PDF of Observed Transitions . . . . .	74
18	PDF & CDF of Time of Transitions . . . . .	75
19	PDF of Node Occurrence . . . . .	76
20	Platforms Killed by Type . . . . .	77
21	Platforms Killed by Name . . . . .	78
22	Visual representation of the state transition matrix . . . . .	80
23	Plain Layered Graph . . . . .	81
24	Visualization of Classes . . . . .	82
25	AFSIM vs Model Node Probabilities . . . . .	85
26	AFSIM vs Modelled Probability Distributions . . . . .	86
27	AFSIM vs Modelled PDF of Chain Lengths . . . . .	87
28	MPPT to Node 18 . . . . .	88
29	Directed Graph Using 'Layered' & OS . . . . .	92
30	MPP to Neutral Outcome . . . . .	93
31	MPP to Red Win . . . . .	94
32	MPP to Blue Win . . . . .	95
33	OIS vs OS . . . . .	96

34	IS vs AS . . . . .	97
35	Normalized Metrics Range Comparisons . . . . .	98
36	Location of Top Nodes . . . . .	100

# List of Tables

1	Variables of Radar Range Equation . . . . .	6
2	Results of Two Die Summation Game . . . . .	17
3	Types of Features . . . . .	40
4	Curve fit Coefficients . . . . .	43
5	Example centrality measures adapted from [1] . . . . .	51
6	Distance Measure Correlations . . . . .	52
7	Degree/Eigen Measure Correlations . . . . .	53
8	Neighborhood and Miscellaneous Measure Correlations . . . . .	53
9	Average Correlation of Measures . . . . .	55
10	Kill chain states used in scenario . . . . .	72
11	Node 1 Description . . . . .	79
12	AS Scoring Values . . . . .	88
13	OS Score Values . . . . .	89
14	Experimental Measures Correlations . . . . .	90
15	Top Ranked Nodes by Measure . . . . .	99

---

---

# CHAPTER 1

---

## INTRODUCTION/MOTIVATION

### 1.1 Digital Engineering

Digital engineering tools have brought a new age of data-driven design and analysis to the world. Idaho National Laboratory defines digital engineering as "a holistic approach to the design of a complex system: Design using models/data instead of documents..." [2] This has allowed for transformations in thinking across the globe into how engineering is performed and how it can integrate the various disciplines that the field covers. Where as before, experienced managers and personnel would have to come together and spend hours, days, or even years, creating plans that adequately combine and utilize the various different disciplines together. Now it can be done much faster and asynchronously. The continued progress and following of Moore's Law means that computing power continues to double. Subsequent simulation and modelling capabilities continue to grow just as fast. Networking capabilities cannot be understated as well. Without the vast network infrastructure across the globe, connecting disparate groups sharing information freely, the world would not be where it is today in any facet.

From humble beginnings in solving partial differential equations to answer unknown questions, to today's large language models, simulation and modelling (M&S) has taken great steps. Much in line with digital engineering and the computational revolution. Developing software has allowed for rapid prototyping as well as exploration of new techniques and processes that would be initially too expensive to explore in a physical experiment. Such tasks allow for optimization algorithms to be run that explore the design spaces. These facilitate the balancing of a large number of parameters set by the stakeholders. Much with how digital engineering has created a new paradigm in engineering that has spurred innovation, simulation and modelling has created its own industry.

Numerous software as a service (SaaS) companies have sprouted from the need for better M&S software. The software selection company *Capterra* provides an easy to use interface to explore the various softwares that exist.[3] Some such examples of M&S softwares are: SIMUL8, AnyLogic, Unreal Engine, Fusion360, SimScale, Solidworks, and Ansys Fluent to name a few. *Capterra* advertises over 200 products to satisfy M&S needs. Numerous open-source softwares also exist depending on the application to perform M&S. For computational fluid dynamics, OpenFOAM, code\\_signature, Gerris, or SU2 could be used. If it is necessary to perform 3D modelling, softwares such as FreeCAD, Blender, or LibreCAD could be utilized. Having competing softwares in the M&S space drives innovation to continue to make the industry better. The advent of the personal workstations and open-source softwares has allowed for a whole new generation of engineers to explore new ideas and bring them to fruition.

## 1.2 Trade Space Analyses

In the process of designing, a vast array of variables that are inextricably and sometimes unexplainably intertwined must be balanced to meet the overarching goals and objectives of the stakeholders. To do this, engineers determine the acceptable ranges for which certain

variables can fall that will satisfy the desired end state. Using simulation and modelling, a design of experiments (DOE) can be performed to determine the effect of variable changes on the overall design. This process can consume large amounts of time in the setup and run time of the models. The end goal generally being the ability to select some designs and their parameters to be created in the physical-world. Those are experimentally investigated to further ascertain the absolute truth with respect to the results of the designs. The amount of time required to perform these analyses is partially determined by the level of resolution in the types of equations that must be modelled to accurately determine the necessary outputs. When performing DOEs, when collated and aggregated effectively, Pareto fronts will appear between variables. These Pareto fronts are the result of the correlation between variables and their ranges associated with the change in the output variables. The selection of variables along this front present the optimal trade offs between that combination of variables to achieve the most efficient result in the output.

In Daniel P. Raymers' book *Aircraft Design: A Conceptual Approach*, Raymer describes trade space studies as trade studies and provides some examples that such an engineer may want to perform analyses on.[4] For example, in varying thrust to weight ratio and its linked variable, wing loading, a designer can explore the variations in output variables such as takeoff distance (Raymer, Equation 5.9) and sustained turns (Raymer, Equation 5.24).

$$\frac{W}{S} = (TOP)\sigma C_{L_{TO}} \frac{T}{W} \quad (1.1)$$

Where  $W/S$  is the wing loading,  $TOP$  is the takeoff parameter,  $C_{L_{TO}}$  is the coefficient of lift in takeoff configuration, and  $T/W$  is the thrust-to-weight ratio.

$$\frac{T}{W} = \frac{qC_{D_0}}{W/S} + \frac{W}{S} \left( \frac{n^2}{q\pi Ae} \right) \quad (1.2)$$

Where  $q$  is the dynamic pressure,  $C_{D_0}$  is the zero-lift drag coefficient,  $e$  is the Oswald efficiency factor,  $A$  is the area of the wing, and  $n$  is the load factor. Another example is

the variations in airfoil thickness to chord ratio and wing taper ratios. Variations in these variables causes changes in the potential top speed of the aircraft. Different methodologies exist in the design of aircraft to determine the necessary values for each variable in question based on stakeholder values and the laws of physics.

### 1.3 Levels of Scale/Resolution

Scale and resolution are some critical concepts to understand before continuing. In the context of this situation, scale refers to the number of systems modelled and resolution is in reference to the level of accuracy in the equations used. Continuing with the aircraft design example of a full aircraft model, scale would be in reference to the number of components modelled. The resolution would be the types of equations to use to calculate the aerodynamic properties at some point in the computational domain. Typically, as the level of scale increases, the corresponding level of resolution is lower. The ultimate goal is to be able to use high resolution equations at high levels of scale. This is currently possible though trade offs must be made when it comes to computational time competing with real-world timelines. For example, in modelling the airfoil of a wing, much care is taken in the selection of the model mesh, computation domain, and the simulation equations. Care is taken to ensure enough cells are located in the right places to provide computation domains that can accurately compute the solutions. Computationally intensive, but more accurate equations are modelled at this scale. As the number of components and interactions of these components increases, i.e. scale, the types of equations used become more generalized and less exact, i.e. resolution, as the size of the domain increases by either expanding the computational domain, or by increasing cell densities. In this example, modelling the airfoil as part of a wing, and the wing as part of the aircraft itself, a different level of resolution in the equations can be used. On the wing alone, the computation time of direct numerical simulations are non-conducive to gathering and acting on any of the data. Thus, the equations used are good at approximating

the values, such as lifting-line theory or vortex sheets, that will be achieved at the expense of accuracy. Accuracy would be reduced by not considering the interactions between various components such as the wing and body, or the wing flaps, or even the turbulence generated through skin friction across the fuselage. In the case of AFSIM the scale is similarly thought of as the number of systems in the simulation.

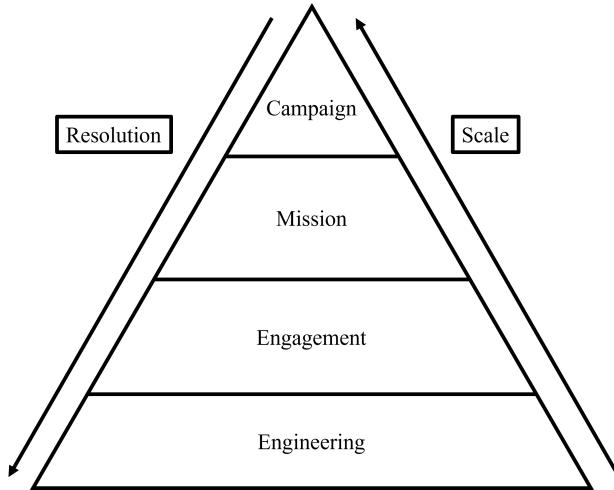


Figure 1: Scale/Resolution pyramid for combat simulation and modelling

The resolution is the types of equations that generate their outputs. Using 1, at the engineering level of scale, the equations used take into account as many variables as necessary to achieve the most accuracy possible. For example, take the modelling of a radar. Some of the engineering scale items that would be created would be the equations necessary to emulate the waveform and pulsing of the radar. It would then need to take into account the percolation of these waveforms through an atmosphere with a certain makeup. Equations such as these require a large number of variables to be tracked and other equations used to determine the exact values that can be expected. This is just one example of the hundreds of equations needed to accurately reflect the real world in the digital one.

Once leaving the engineering/sub-system level, one comes upon the first stage that requires direct implementation of human decision making. This is the one-on-one stage. Should a modeler desire, AFSIM can provide Operator-In-The-Loop (OIL) capabilities where hu-

mans control and make decisions like they would in the field but in the virtual environment.[5]

As the number of platforms continues to increase in the scenario, the engagement level gives way to the many-on-many stage. At this stage, as the name suggests, multiple aircraft exist on either side that then interact not only with the other side but within their own side, by communicating and coordinating tactics and strategies. This adds an additional layer to the required number of equations and models as tactics must be programmed in. Without intra-team tactics, the simulation is worthless as it is just a mix of one-vs-one engagements. The mission stage also allows for the achievement of larger mission objectives. Resolution wise, this means that the values used start to become more simplified. For example, Equation 1.3 is the equation needed to determine the amount of power returned to a radar.

$$P_r = P_{peak}(DC) \frac{G_x}{L_x} \frac{A_{xt}}{4\pi R_{xt}^2} \sigma \frac{A_{tr}}{4\pi R_{tr}^2} \frac{\lambda^2}{4\pi} \frac{G_r}{L_r} F_{40} F_{BW} F_{POL} \quad (1.3)$$

Table 1: Explanation of variables for radar range equation from [6]

<b>Variable</b>	<b>Description</b>
$P_r$	Reflected return power
$P_{peak}$	Peak transmitted power
DC	Duty-cycle of transmitter
$\lambda$	Wavelength of radiated signal
$G_x, G_r$	Gain of transmitter and receiver antennas
$L_x, L_r$	Internal losses within transmitter and receiver
$A_{xt}, A_{tr}$	One-way atmospheric attenuation factor
$R_{xt}, R_{tr}$	Range from transmitter-to-target and target-to-receiver
$\sigma$	Radar cross section of target
$F_{40}$	Pattern propagation factor
$F_{BW}$	Factor accounting for bandwidth mismatch of signal and transmitter
$F_{pol}$	Factor account for polarization mismatch of signal and receiver

For many of these values, it is a lookup table or empirically determined constants. The final level of this pyramid resides at the campaign level. With potentially hundreds if not thousands of modelled platforms, the resolution begins to be reduced to simple random draws based on empirically derived probability models. For example, instead of calculating the

trajectories and aerodynamics of a launched missile, a draw of the probability determines the damage done and/or the success of the strike. Performing these reductions have the benefits of decreasing computation time but does come with the problem of the results being a product of a probability distribution rather than a directly modelled value.

## 1.4 Wargaming

An introduction into the art and science of wargaming is necessary. In general, wargaming is able to take three forms, as defined by Appleget, Burks, and Cameron in their book *The Craft of Wargaming*.<sup>[7]</sup> Initially starting in the 1800s with the Prussian army as *Kriegsspiel*, there are three categories of wargames:

- Educational
- Experiential
- Analytical

In each category, there's a different focus which lends to differing ways that the games are built, interacted with, and ultimately used for. In educational games, such as *Kriegsspiel*, the focus of the game is on creating a means to explore the learning objectives desired for the individuals playing. In experiential games, the design is focused on representing the duties and tasks of the individual players as a way of exploring the decision making of those individuals in a controlled environment. For analytical games, the focus is on realistic data generation that lends itself to making technology and strategy choices. This research focuses on analytical wargaming and generating statistically significant results. It should be noted however that there is overlap amongst the three categorizations in how they can be performed and in what ways. As an example, AFSIM and its abilities to perform real-world modelling of a multitude of interconnected systems, has the ability to be a training platform for new

operators (educational), real-world simulator for larger scale exercises (experiential), and as a means of exploring the potential impacts of new technologies (analytical).

---

---

# CHAPTER 2

---

## BACKGROUND

### 2.1 Wargames: Why

First a definition of what a wargame is. From the NATO Term database, wargame is defined as "a simulation of a military operation in which participants seek to achieve a specified objective, given pre-established resources and constraints."<sup>[8]</sup> This definition defers slightly from the Joint Chiefs of Staff JP-05, *Joint Planning*, document defining wargaming as the "representations of conflict or competition in a synthetic environment, in which people make decisions and respond to the consequences of those decisions."<sup>[9]</sup> On a fundamental level, these definitions are the same. The goal is to simulate the environment required to perform tasks and evaluate the consequences of those actions.

As previously mentioned, wargames as is known today began in the 1800s with the Prussian military's *kriegsspiel*. Wargaming as an idea however is not a new one. It is understood that the concept of wargames has been around for some time.<sup>[7]</sup> Given the technology of the ages however, these games were more abstracted. For example, the game of *Go* is a form of abstract strategy game that is considered to be an early form of wargaming.

Chess is also considered to be an early form of wargaming with roots in India and multiplayer games. Focusing on the most relevant ancestor of modern wargaming, *kriegsspiel*, originates with the purpose of educating Prussian officers on different tactics in a tactile format. This original form is considered *rigid-kriegsspiel* as it contained multiple sets of specific rules to be followed that guided play. Since then, multiple forms of wargaming have sprouted, ranging from set-in-stone rules to define gameplay, to much more free and discussion oriented forms such as seminar games. Determining what type of form to take highly depends on the desired outcomes of the game. In analytical games, where the general purpose is data generation, the form to take is more closely aligned with *rigid-kriegsspiel*. Having clearly defined rule-sets and behaviors allows for the creation of true to form results.

As an example to the potential impacts of utilizing wargames, there is an example used by Applget et. al. of *Plan Orange*. *Plan Orange* was part of a series of wargames used by the Navy in the years between the World Wars. The Navy utilized wargames as the capstone event of Captain training. In this format, these games would be classified as an experiential wargame used to allow individuals to utilize the skills they've been taught in a controlled environment. As noted by Applget et. al. being able to run wargames repeatedly with varying parameters is a difficult task and typically doesn't happen due to time, repeatability, and other restraints. With *Plan Orange* however, the Navy was able to dedicate time, money, and manpower to run these events. Famously, at the outset of America's involvement in World War II in the Pacific, Admiral Nimitz sent two individuals to retrieve and comb through the game setups and results. Admiral Nimitz realized that there was a good chance that the current intelligence on Japanese force dispositions could possibly be found amongst the games. The results could be analyzed to find weak points. Later, in a speech to a graduating class from this same schoolhouse, Admiral Nimitz is quoted as saying,

"The war with Japan had been enacted in the game rooms at the War College by so many people and in so many different ways that nothing that happened during

the war was a surprise-absolutely nothing except the kamikaze tactics toward the end of the war. We had not visualized these.” [7]

Focusing on the forms of the games more closely aligned with the research goals of this thesis, a quick history of closed-form and closed-loop games are in order. In closed-form games, information that can be reasonably attained by the asymmetrical intelligence gathering resources of each player is thus given to each player.[7] This allows for information to be perceived by both sides rather than being directly fed the same ”ground truth” information. An example of this provided by Appleget is the hobby game of *Battleship*. Both sides slowly determine the location of the other through guessing and checking. This term should not be confused with the term closed-loop. This term, as used by Appleget et. al., defines a scenario where human decision making is completely removed from the wargaming process. In these cases, the necessary controllers for the platforms in the scenario are pre-programmed and the simulation is left to its own devices to run. AFSIM and this research utilizes closed-form and closed-loop wargames to record and analyze different data to answer the questions at hand. A software that can make use of closed-form and closed-loop principles allows for analytical wargames to occur. Though these have progressively become better over time, they are still not without their flaws. For example, when it comes to set up time for a simulation, there are a multitude of parameters and objectives that must be carefully defined to allow for an accurate wargame to be run. This typically requires many interfaces with the problem owners to properly scope the criterion that must be met and the measures to record for the subsequent analyses to be adequate.

Wargames that focus on education and experience building do not have external problem owners. That is the team designing the game who are typically subject matter experts. While these games can be set up more quickly, their ’simulation time’ is much longer. They are likely to be unrepeatable. Measurable data being a tertiary item since the point is the experience of playing within the game. Analytical wargames however, even though their setup may take longer, can provide much more control and a faster ability to iterate through

versions of a scenario where data can be recorded and analyzed. Assuming these processes are properly controlled, this allows for statistical analyses to be performed on whatever data is being recorded. Statistical significance comes into play in simulations that implement stochastic processes where measures are recorded and inferences from that data is made. This provides a means to understand and quantify otherwise unquantifiable processes. With these games, there do come assumptions on the way the data is generated and recorded that must be taken into account when making determinations from said data. As an example, analytical wargames in closed-loop formats, typically do not perform well when technological leaps are made and thus implemented. Appleget et. al. gives the example of the swapping out of the Sherman tank for the Abrams. If a simulation is run and new data is recorded, the underlying assumption made is that the strategies and tactics used for the Sherman tank are applicable entirely to the Abrams.[7] This however, is untrue. This is where care must be taken in performing technology swaps in analytical wargames. For direct one-to-one technology swaps it requires the capabilities of the new technology to not create changes so vast that the underlying logic-controllers of the platforms are ill-equipped to take advantage of the new abilities. As part of the design process of new controllers, the old logic-controllers can be used as a starting point. It is most likely in performing a swap that would require a new logic-controller can still borrow elements from old controllers. Similar to training operators on new technology who come from the old technology. In general, care must be taken in the setup of wargames to ensure that the results obtained meet the necessary criteria to be used.

## 2.2 AFSIM

AFSIM is the foundational software for this research. Starting with some history, AFSIM began as the Analytic Framework for Network Enabled Systems (AFNES) at Boeing.[10] This piece of software was utilized between 2003 and 2013 with a primary focus on simulating ad-

vanced air vehicle concepts in integrated air defense systems (IADS) scenarios.[10] However, starting in 2011, the Air Force Research Lab's Aerospace Vehicles Technolgoy Assessment & Simulation (AVTAS) Lab ran trade space analysis of the current modelling and simulation frameworks to select a simulation software to be further developed into a bigger program by AFRL.[10] In 2011, AFNES was selected to begin transformation into a more generalized modelling and simulation framework before being wholly transferred to AFRL.[10] At AFRL, it went through a rebranding to AFSIM and has continued to be developed. Put by West & Birkmire,

”[AFSIM] was designed to be a multi-domain platform, meaning it can model land-, sea-, air-, and space-based platforms, enabling modelers to include submarines, naval vessels, tanks, airplanes, helicopters, satellites, and even cyber agents in the same simulation, if needed.”[5]

As a general M&S software, AFSIM makes use of a number of architectural features that allow it to be both easy-to-use while allowing for a vast array of simulation resolutions. AFSIM is written in C++ [10] and consists of an integrated development environment called Wizard. [5] In this IDE, a user is able to write in a Java-like script-based format to implement platform definitions and controller logic. At a high-level, the script based framework allows for a plug-and-play model where files can be moved easily between scenarios.[5] Platforms are instantiated without much extra work needed to integrate them and quick changes can be made. This framework also allows for the aforementioned levels of resolution ranging from quick subsystem level interactions with high fidelity equations to broader, campaign level simulations modelling hundreds if not thousands of individual platforms making interwoven decisions. [5]

To do this, AFSIM has four key architectural elements [5]:

- attributes
- components

- links
- information

Attributes contain what a platforms name and type is as well as side affiliation. Components control the platforms behaviors. Links exist to control the information and general data exchange from the platform sub-system level to inter-platform communications. Finally, information consists of the data that is held by the platform and includes what is considered to be perceived by the virtual operator and the platform agent. By creating this layered architectural approach, emergent system-of-systems behavior is able to appear as a result of the interweaving of elements defining various platforms.

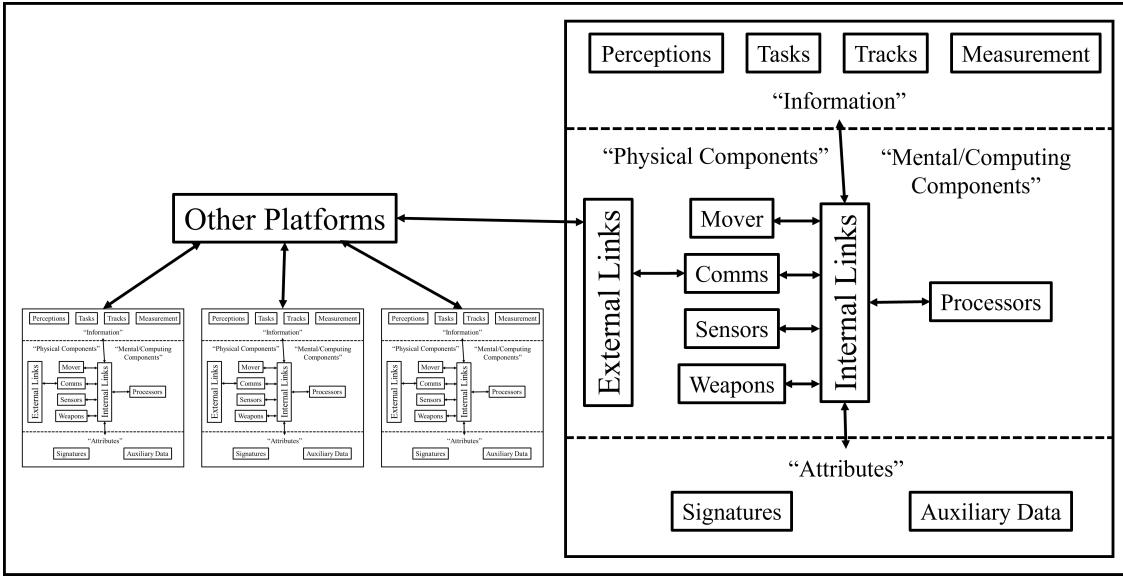


Figure 2: AFSIM architectural elements adapted from [11]

It is important to understand how AFSIM implements platform control logic given the requirement of operating closed-loop games. Without proper platform logic, results obtained by any simulation that is not a purely physical simulation of a technology is unusable. To do this, AFSIM makes use of two elements, the perception processor and task processor.[6] These two elements combined allow for the system to build up a knowledge base, and take action based on information gathered. Starting with the perception processor, this consists

of the information related to the perception of self, threats, assets, and peers while allowing for a tuning of parameters based on necessary levels of aggression and accepted tactics. For example, information gathered by the communication links and on-board radar contacts would be fed to the perception processor for classification. Next, the tasker. This element is designed with the purpose of coordinating information into actionable information and dispatching commands to perform necessary actions as a result of that information. To do this, the tasker consists of a task generator, task-asset pair evaluators, resource allocation algorithms and strategy settings that handles rejecting tasks based on broader objectives.

[6]

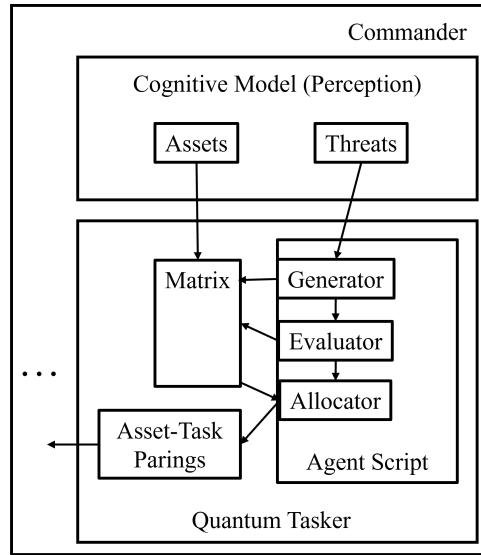


Figure 3: Representation of the internal workings of the tasker. Adopted from [6]

Once necessary steps are taken and decisions have been reached, to perform the tasks assigned, another system is used called the behavior tree. Bernal describes these behavior trees to be similar to the attack trees as defined by Buckshaw.[12] The overarching idea being a modular graphical format that allows for easy creation of logic to control behavior. This is done by using connector-node types. Selectors choose the first child behavior that passes the precondition check of the node. Sequences perform child behaviors in order until a child behavior fails the precondition check. Parallel nodes perform all the child behaviors

at the same time whose precondition check passes. Weighted-random nodes selects a child behavior based on a weighted random selection. Finally priority selectors choose the child behavior whose precondition check comes back with the largest precondition check value. With these five types a vast array of behaviors can be modelled and implemented while giving the capability to swap out restricted trees with unclassified versions without degrading the simulation as a whole.

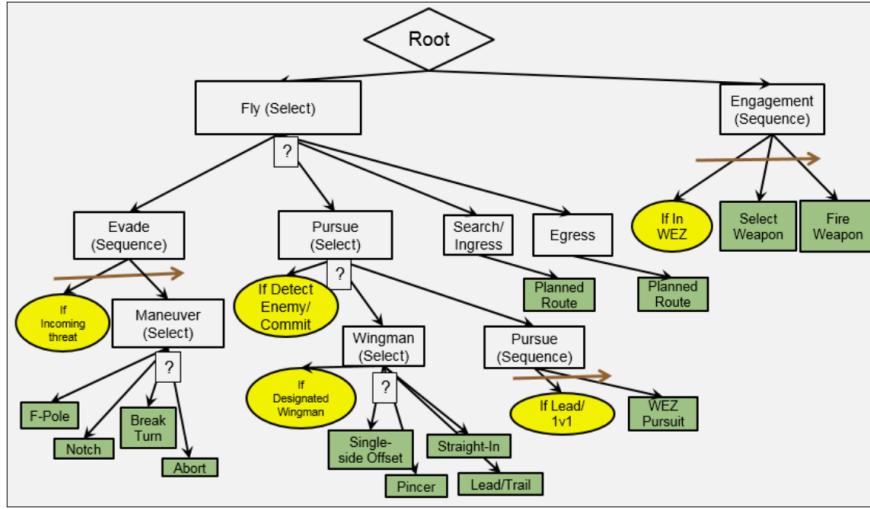


Figure 4: Example behavior tree for a sweep mission by [13]

## 2.3 Convergence Monitoring

When using simulations that contain stochastic elements, there is going to be variance in the recorded outputs of the system. Especially in the modelling of the real-world which can be considered a chaotic system that is heavily dependent on the initial conditions. Therefore, it is important to quantify this variance and present it as a means of understanding the outputs of the system and the ranges that they are capable of taking. By providing such measures, one can determine the acceptability of the information. An assumption that is implicit in these calculations is that the numbers being generated are the results of a system that will produce accurate numbers representative of the physical world.

As a way to illustrate the above points, consider a summation game using a pair of dice. It is assumed that the die sample from the same probability distribution for each possible number. Notice, it is not stated that the probabilities of selecting each number is uniform but rather are sampled from the same distribution. Each die interacts independently of the other. Thus their resulting numbers are independent. Given that this is a simple two dice example, determining all the possible end states from the combinations of the die is easy to compute and show in 2.

Table 2: Potential results of a summation two dice game

		Die 1						
		-	1	2	3	4	5	6
Die 2	1	2	3	4	5	6	7	
	2	3	4	5	6	7	8	
	3	4	5	6	7	8	9	
	4	5	6	7	8	9	10	
	5	6	7	8	9	10	11	
	6	7	8	9	10	11	12	

To determine the probability of getting each summed result, it is as simple as counting the number of times that the number is shown within the table. For the probability of rolling a 7 is 6/36. However, the probability of rolling a 2 is 1/36 and rolling a 10, 3/36. From the table, calculating the exact probability of each number is trivial. An accurately built scenario can be run and the probability of rolling each number can be estimated. This should not be construed as proving that simulations give the exact correct answer but rather demonstrate that the correct number can be contained within a confidence interval for which a certain level of significance can be given assuming that the underlying models of the simulation are true to their physical counterpart.

The above image illustrates a simulation where two uniformly distributed die are cast and the sum of the resulting numbers is made. Every time that a 7 is found, the sample probability of rolling a 7 is updated and shown above. Initially, the probability begins to

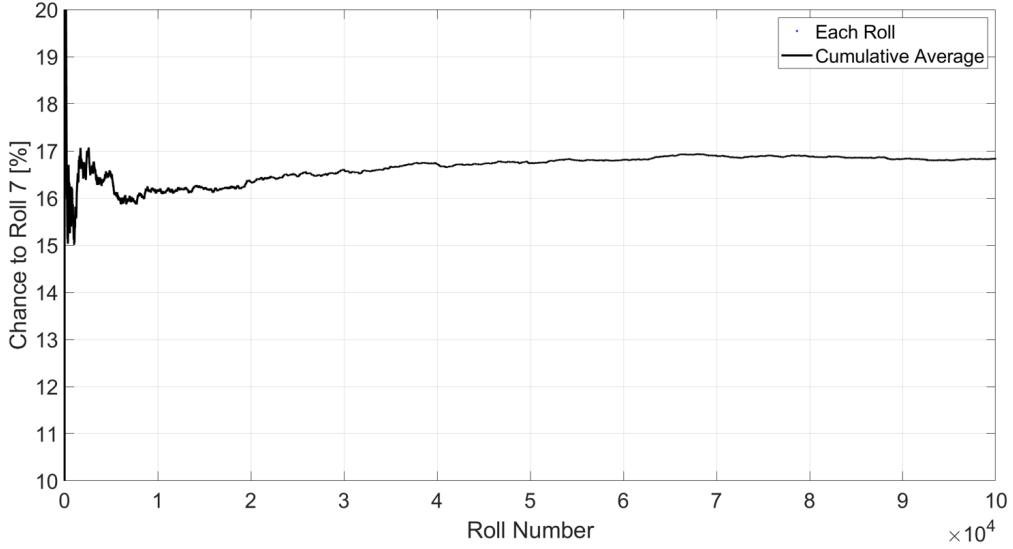


Figure 5: Example simulation demonstrating the estimation of the probability of rolling a 7 in a two dice summation game.

swing between different values quickly. If the simulation were to be stopped at any of these points, the probability of rolling a 7 would most certainly be incorrect. To quantify this swinging effect, a confidence interval is used to show that though the value is listed, the confidence interval indicates the potential range that the value could actually possibly be. As the simulation is continuously ran and more information is gathered, the effect is that the confidence interval will begin to shrink to demonstrate how the value that is shown is more confidently the actual value. Within a simulation, this value can be used to determine when a simulation should stop. There is a problem. This is, that one would have to know the value associated with the size of the confidence interval desired. In the dice example, since the real value is known, an interval could be given, such as  $\pm 0.5\%$ . In the case when the acceptable range of a value is not known and the simulation is being used to get the value of a confidence interval it cannot be used as the stopping criteria. This can be remedied through the use of a normalized confidence interval. This allows for a user to define two values that can be used in conjunction with each other to dynamically determine the stopping criteria. The equation for the confidence interval is shown below where  $\sigma$  is the standard deviation of the sample,  $n$  is the number of runs in the sample,  $\bar{x}$  is the sample average, and  $z^*$  is the

z-score of the confidence interval.

$$CI = \bar{x} \pm z^* \frac{\sigma}{\sqrt{n}} \quad (2.1)$$

In this equation, only one value  $z^*$  is given to size the confidence interval. A normal value for  $z^*$  is 2.56 associated with a 99% confidence interval. That is to say that with 99% confidence, the true value falls within the range of the estimated value. To convert this equation into a more useful value is to divide through by the mean. By doing this, the confidence interval becomes a function of the mean and is given as a percentage of the mean.

$$CI\% = 1 + \frac{z^*\sigma}{\bar{x}\sqrt{n}} \quad (2.2)$$

By focusing on the fractional value in Equation 2.2, a user can define stopping criterion based on the  $z^*$  and the percentage around the mean that is desired. For example, where as in the dice example, a value given is focused on the value of the interval around the mean, the new values allow for a more precise and dynamic range. In this example, the new stopping criterion would be to stop when the size of the 99% confidence interval is smaller than 1% of the given mean value. This new value can be plotted as a residual and is shown in 6 for the dice example.

This is how convergence intervals are implemented in AFSIM when recording events. Since the recorded values are dependent on sampling from a vast array of probabilities, providing an interval allows for a user to define the level of confidence necessary to answer the questions at hand.

It is also important to point out the Central Limit Theorem (CLT). CLT as defined by LaMorte states that, by adequately sampling from different distributions, then the resulting output can be approximated by a normal distribution. [14] This is key information to understand since the information being recorded from AFSIM is the result of numerous different distributions. The CLT allows for the underlying distributions to be disregarded so

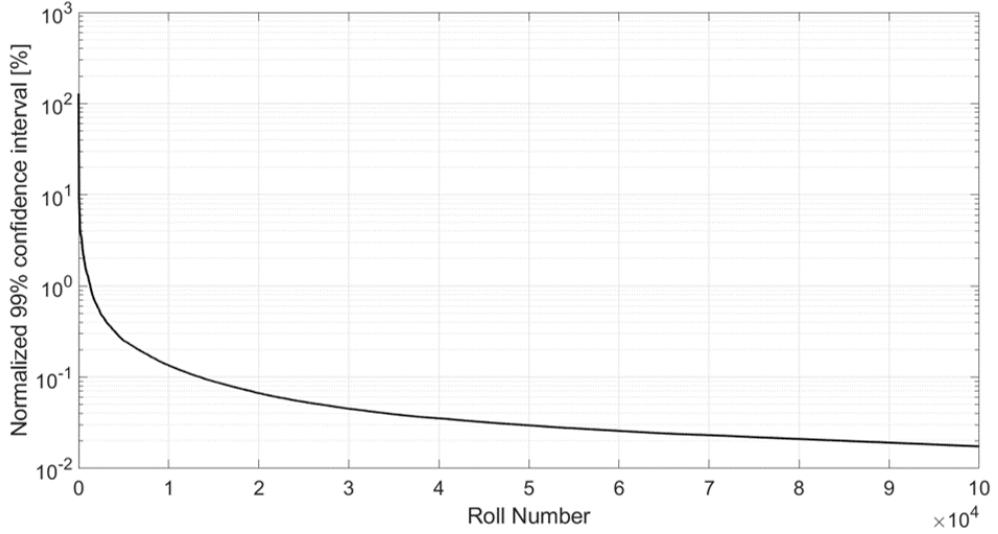


Figure 6: Plot of the confidence interval percentage for a two dice game summing to 7

long as the recorded variables are adequately sampled so that they are representative of the interwoven connections between them.

## 2.4 Markov Chains

Markov chains are an interesting mathematical concept that was discovered by Andrey Andreyevich Markov.[15] He focused on stochastic processes and modelling them in formal ways. The fundamental idea behind Markov chains is thus, the probabilities of transitions to the next state are independent of the route to the current state. Using the definition from Ross, Markov chains are more formally defined as follows.[16] Assume a set of finite states  $\{X_n, n = 0, 1, 2, \dots\}$ . If the process is at time  $n$ , shown as  $X_n = i$ , then the process is in state  $i$ . Additionally, when in state  $i$ , there is the fixed probability  $P_{ij}$  that denotes the probability of moving from state  $i$  to state  $j$  in the next step  $n$ . This can be shown below using equation 4.1 from Ross.[16]

$$P\{X_{n+1} = j | X_n = i, X_{n-1} = i_{n-1}, \dots, X_1 = i_1, X_0 = i_0\} = P_{ij} \quad (2.3)$$

Additionally, in this form, the summation of all transition probabilities,  $P_{ij}$ , across all

$j$ 's is equal to 1. That is to say that the summation of all the outgoing probabilities from a state are equal to 1. Systems in this form have interesting properties that can be useful when dealing with system dynamics. That is a system can be represented as a right stochastic matrix that can leverage matrix multiplication to determine the state of the system at some time. For example, to determine the probability of being in each state  $i$  at time  $n$  is as simple as performing matrix multiplication as follows.

$$\mathbf{v}\mathbf{P}^n = \mathbf{w} \quad (2.4)$$

Vector  $\mathbf{v}$ , is an initial distribution at the beginning of the process and vector  $\mathbf{w}$  is the probability of being in being in each state  $i$  after  $n$  steps.

This naturally flows into the question of, what happens to the distribution as the number of steps approaches infinity? Otherwise known as the stationary distribution. To understand the stationary distribution, one must understand the types of states that may exist in a chain. These are recurrent and transient. Simply put, a state is considered recurrent if in an infinitely running process, the probability of returning to the state is 1 and if less than 1 it is considered transient. As an example of these classifications, take the Gambler's Ruin problem. The Gambler's Ruin goes like so. You are in a casino at a game where you win 1\$ with some probability  $x$ . You lose 1\$ with probability  $1 - x$ . You start with a finite amount of money. You will stop when you either run out of money or reach some limit above your original amount,  $N$ . In this example, every node that isn't 0 or  $N$  are transient nodes. Given an infinite amount of time, i.e. assuming the casino can always continue playing, you will always end up at either 0 or  $N$  dollars and therefore quit playing. Should  $1 - x > x$ , then you will always end up at 0\$ given an infinite amount of playing time. Thus, the Gambler's Ruin.

States are able to communicate with one another through connections where the probability of reaching each other is non-zero. If a set of states are able to communicate with one another without going through an external state to the set, then the set is called a class.

Classes themselves can also have transience and recurrence that are inherited by the set of states that make it up. Should there exist only one class that is recurrent, then the Markov chain is considered ergodic. Otherwise known as being irreducible. That is to say that the chain is in its simplest form. In the case where there are multiple classes, a Markov chain can be condensed down into a chain where each class is represented as node. In doing this though, the Markov chain's nodes now take on a new meaning as they are now agglomerations of the underlying states.

Stationary distribution, sometimes labelled the limiting probabilities, can only be determined if the chain being analyzed is ergodic. This is a critical requirement. If a Markov chain is not ergodic then a limiting probability cannot be analytically determined and must be computationally approximated. Should the chain be ergodic, looking at the Equation 2.4 it can be rewritten as

$$\mathbf{w}\mathbf{P} = \mathbf{w} \quad (2.5)$$

This comes from the principle that a stationary distribution is unchanging after one step and thus does not change over infinite steps. To then solve for the stationary distribution is as simple as solving for the left eigenvector of the transition matrix. This can be a powerful tool when analyzing large matrices but does come at the cost of requiring ergodicity. If a transition matrix is not ergodic, then other tools can be used to understand what the long run form of the chain will look like. Lacking ergodicity typically indicates that there are classes that are transient and classes that are recurrent. In the recurrent classes, the stationary distributions can be calculated on a class level but not in the chain as a whole. To perform the calculation on the chain as a whole, numerical simulation must be determined where a prescribed amount of change between each step is reached.

In the case of implementation in this research, the entirety of a single run is contained within a Markov chain. Each node consists of a set of states based on the recorded features at some time in the simulation. The existence of a node implicitly means that the set of states

recorded existed at some point in the simulation and the existence of an edge connecting two different nodes indicates that the states transitioned between each other at some point. While each run contains its own Markov chain, and therefore chain of events, the power of modelling in this way comes from the agglomeration of every run's chain into a larger meta-Markov chain. This resulting chain indicates all the visited sets of states across all runs and the probabilities of transitioning to each other node based on the number of transitions realized within the simulation itself. If the interwoven and complex system-of-system dynamics can be contained within a Markov chain model through a subset of features, there exists an ability to determine key points in the simulations even though they are inherently chaotic.

---

---

# CHAPTER 3

---

## RESEARCH OPPORTUNITIES

### 3.1 Previous Work

Some work has been done in attempting to use results from analytical closed-form and loop wargames. A well done study on the effects of technology changes in analytical wargames is by Biltgen in his 2005 PhD thesis. In this, Biltgen attempts to model a large scale mission utilizing tens of platforms being controlled by a strategic evaluator and tactical decision makers.[17] He uses an off-the-shelf modelling engine called FLAMES. To do this, Biltgen lays out a methodology to define the problem, create a meta-general that uses a trained AI to make strategic decisions and smaller tactical level decisions for each platform, use high quality physics models for each platform, and perform multiple runs through multiple different iterations of platform design points to be able to ascertain the effects of platform changes on the mission success of the mission.[17]

More recent work is from primarily 2021 and students at the Air Force Institute of Technology receiving their Master's degrees. Tryhorn creates a means to model the fog of war in AFSIM. This provides a way for analytical wargames to have directly tunable

parameters that can be used as design points. As expected, as the fog of war is 'turned up' the ability to reach the same level of mission success is more difficult and less likely.[18] Crumpacker attempts to implement an AI policy that uses AFSIM as the training platform to create an AI model that can emulate human fighter pilot decision making and maneuvers.[19] Uniquely, Crumpacker allows for the AI to discover the ways to control the aircraft and what it means to win an engagement while following the laws of physics. The point of Crumpacker being the demonstration of AFSIM as a useful platform to create AIs that could be trained in a virtual environment and then plugged into a physical plane that would be useful in air-to-air combat.

Connors sets forth to create a methodology for performing analytical wargames on the implementation of new technologies. Generally, this can be considered as an extension of Biltgen's work. Connors clearly defines success criteria and utilizes ANOVA methods to quantify the changes in the success criteria when a new technology is implemented.[13] Finally, Bernal demonstrates a model that can be used by an end user to determine the usable design space of a set of variables based on the acceptable level of risk defined by the user.[12] Uniquely, Bernal enumerates the necessity of statistical significance in utilizing the results. All others make use of multiple runs but do not take into account the stochastic nature of the runs being used. Connors does recognize statistical significance but focuses rather on the differences seen between changing different parameters rather than the actual significance of individual results.

## 3.2 Current Inadequacies

Work that has been done so far focuses on quantifying the change in the recorded output variables versus the change in the input variables. As previously mentioned, Biltgen's work focused on the methodology, similar with Connors & Bernal, where the goal was to show how the methodology presented allowed the user to vary input parameters. The results could

then be recorded and the change caused by the variance in the input parameters could be quantified. This allowed for meta-models to be created where the selected input parameters could be varied beyond what was actually tested and the resulting output parameters could be interpolated or extrapolated to determine the new outputs. Where these models are useful is in lowering the number of design points that must be tested to be able understand the output variables. They fail to explain why the size of the change is the size it is or how it increases or decreases beyond general engineering intuition. Bernal performs similar work in input/output analysis with the difference in being a focus on statistical significance of each results design points and using genetic algorithms to interpolate the design space. The usage of the genetic algorithms here to explore the output space used in lieu of general mathematical models is due to the nature of the variance in the outputs not necessarily being connected to model in the first place. Biltgen attempts to address this by using what they called an AI to determine the rest of the design space. Similar to the rest, Tryhorn also performs input/output analysis but focuses on creating a tool that can be used to perform larger analyses given new tunable parameters. Some different work so far in this space would be the work by Crumpacker and the focus of using AFSIM as the test bed rather than measuring directly the input/outputs of the scenario. However, the AI that Crumpacker created is still a form of input/output machine where the actual explanation for why an action was performed is wrapped up inside a black box of many hyper specific parameters and activation functions.

This leaves rooms for methodologies and analyses that attempt to quantify the why. With the way analytical wargames are performed today, they are generally used as input/output devices. These results while useful, require the end user to infer how and why that change actually occurred in that way. For example, if practitioners desire a more effective air-to-air missile system, they may come to the assumption that having a faster missile will strike more targets than the current slower missile. They will create the necessary scenario and logic and then run multiple instances with faster and faster missiles representative of what could

actually be manufactured and employed. The resulting output of missile strikes may increase or decrease. This information in it of itself is useful to those individuals. What it is lacking however, is the reason why a faster missile did better or worse. For example, it could perform better as a faster missile reduces potential reaction times and currently the simulated enemy cannot react fast enough. On the other side of the spectrum, it may perform worse for the similar reasons related to reaction times. In this case, it could be performing worse due to the reaction time of the missile not being fast enough or capable of maneuvering fast enough to hit a reacting target. Where as with the demonstrated methodologies, the reason for the changes are left up to the users to determine through various means. If however, in combination with the inputs and outputs, was a way to analyze the agglomeration of these runs to identify key points in time that can be traced to the outputs, that would be a useful technology.

### 3.3 Problem to Solve

This research attempts to answer if there is a means by which the incredibly complex system-of-systems simulation of analytical wargames can be distilled into a meta-Markov chain that can be analyzed. Analysis would determine the points in time that can be considered pivotal to the understanding of the dynamics of the system. To answer this question, multiple other questions must be answered beforehand. First, understanding what to model as both the underlying scenario and the recorded features must be answered. Second the necessary heuristics must be created that will allow for programmatic identification of what is considered pivotal.

---

---

# CHAPTER 4

---

## WHAT TO MODEL

### 4.1 The Past

In a perfect world, where there are no limitations, the best simulation would be one where every variable and its transitions, be them continuous or discrete, would be recorded. One would then be able to sift through and collate this information into an easy to read composition that could be used to understand the progression of a scenario. However, the world is not perfect and there are limitations when it comes to human perception & cognition and computing power alike. Thus, it is necessary for reduction in recorded information to occur, so that analyses can be performed in a timely manner and displayed in a way that can be understood. The question is then posed, how does one determine what is acceptable to record to understand the scenario at hand. As a starting point, looking at what others have done before is useful.

Focusing on the input/output research that's come beforehand, a list can be compiled of all the different metrics that have been recorded. Connors focused on evaluating the effects of a new missile technology. Thus, their metrics were air-to-air focused. They were:

- Time to service target
- Percentage of targets destroyed
- Weapon effectiveness
- Average engagement distance
- Number of blue side hits
- Wildcard metric

This wildcard metric was reserved for Connors to fill in depending on what the composition of information indicated would be useful that was not already contained within the already recorded data. In the end, this became a metric to explain whether engagements occurred within or beyond visual range. Weapon effectiveness was defined as the percentage of accurate strikes to missiles fired. These metrics were recorded in-line with the simulation, however, only their final value at the end of the simulation was used to ascertain the effects of changing the weapon compositions.

Moving to Miller, the metrics recorded were similar in spirit to Connors with some changes. Miller uses a motivating framework called the kill chain as the driver for their innovation in autonomous munitions. Though this framework is used as motivation for their work, it does appear in other works in various capacities. For Miller, the focus of the research was on using a low-cost cruise missile (LCCM) in a suppression of enemy air defense (SEAD) mission. This means that the metrics measured by Miller focused on reflecting general case of air-to-ground attack. Miller attempts to model and use a wide area search (WAS) munition type. Work had been previously done by Jacques and Dunkel in this area. Borrowing from their works, Miller identifies groups of parameters that generally effect mission success. These being:

- Number and type of targets

- False target objects (FTOs)
- Size of search area
- Area coverage rate of munition
- Capability of sensor
- Effectiveness of ordinance
- Effectiveness of enemy threats
- Environmental factors

These parameters are recognized by Miller as having the ability to be broken down into smaller less abstract points that are more recordable. For example, size of search area is very clear and measurable, however, environmental factors is a broad category that would need further breakdown to understand the actual influential factors. Instead of further defining these values, Miller explains that the parameters explained feed into the more general probabilities. These are used to determine the definition of mission success. These being percentage of total targets detected ( $P_D$ ), percentage of detected targets correctly classified ( $P_{ID}$ ), percentage of classified targets attacked ( $P_A$ ), and percentage of attacked targets destroyed ( $P_K$ ). This is the equation presented by Jacques for identifying areas of weakness in a WAS munition.

$$P_{MS} = P_K * P_A * P_{ID} * P_D \quad (4.1)$$

Jacques equation is unique in that it does take into account the intermediate effects that would directly relate to mission success. Performing an analysis on a new technology or strategy that increases or decreases these values, at least provides an anchor in time for where an analyst should look for explanations as to why the total mission success increases or decreases.

Bernal, takes a different route over the previous two authors in understanding the input/output paradigm. The focus lies in creating a methodology that allows a user to understand the allowable input ranges depending on the desired output. A citation of Buckshaw notes that a common practice is to provide a multi-weighting variable schema that can allow for decision makers to change values as necessary to fit their needs. To do this, it is pointed out again, that a larger framework is required that can provide guidance on what to use. Bernal points to multiple usages of the F2T2EA, or kill chain, framework that helps define the general process for completing missions. Armed with this information, Bernal proposes a general equation for a single 'harm' metric. This equation takes into the account the effective usage of resources to perform the task at hand ( $E$ ) while also considering the cost of using said resources ( $C$ ). In general, the larger the value of the metric, the less risky a set of resources and costs are considered.

$$J_H = \sum_{j=1}^M (E_j) - \sum_{i=1}^N (C_i) \quad (4.2)$$

In applying this harm metric, Bernal poses an example of an air-to-ground mission. An integrated air defense (IADs) scenario consisting of airborne attackers for whom the harm metric will be applied versus the ground based defenders. Objectives are clearly set that the blue side attacker's goals are to destroy as many surface-to-air missile (SAM) sites as possible. The red side objective is to protect the SAMs. Transforming those goals into measurable values, Bernal uses three ratios that can be defined as:

- Percentage of blue side losses
- Percentage of red SAM's destroyed
- Blue weapon effectiveness

Additionally, weights are placed on to each ratio that allows for finer tuning of the metric to better represent the desires of the analyst.

$$J_H = w_1 \frac{A}{A_{Tot}} - w_2 \frac{S}{S_{Tot}} - w_3 \frac{A_M - A_{M,des}}{A_{M,Tot}} \quad (4.3)$$

Fundamentally, much like with Connors, Bernal does not contain a way to explain why the variations in the inputs would necessarily change the outputs. However, Connors, Miller, and Bernal, could create plots of their metrics where each value is plotted with respect to simulation time. This would allow for an analyst who is doing multiple runs or even singular runs to identify points in time where metrics jump to new values. Assuming the analyst has the ability to replicate the simulations exactly and can navigate to these times, a visual analysis of the scenario could provide insights as to why something occurred. Doing this still requires multiple steps and is not necessarily easy automatable to provide broad understanding to the analyst.

The way Biltgen goes about defining the metrics they used for their analysis revolved around a multi-stage approach. Initially, they started out with a list of candidate metrics that would be generally useful to know with regards to a mission. These were:

- The duration of the conflict
- Percentage of targets destroyed by category
- Number of missions flown
- Amount of munition expended
- Monetary cost of those munitions
- Number of theater ballistic missiles expended
- Average time to target
- Percentage of blue side losses
- Percentage of time critical targets killed

Most of these exist in the similar veins to Connors and Bernal. To further specify the metrics to use, Biltgen focuses on an air-to-ground mission built around the precision engagement and global attack paradigm of the Air Force. Using information from the Air Force, this meant using metrics related to:

- The percentage of effects achieved
- Percent of successful engagements
- Time for effect to be achieved
- Percent of accessible Earth
- Cost to perform the attack

It is recognized that there are no clearly established common metrics that are used across all mission types and domains. Interestingly though, a common framework appears to exist to describe missions in general. This is the previously mentioned kill chain.

In exploring previous work in the field of analytical wargaming, it appears as though the metrics used to define mission success are highly dependent on the desires of the analyst performing the simulations with the goal of answering the questions posed. There are some common threads that do exist across these works. These being:

- The percentages associated with number of attacking units lost/returned
- Number of munitions successfully used
- Number of targets destroyed

These metrics, though useful in understanding the changes a new technology can have on the level of success of a mission, still do not provide a means to explain how the technology caused the outputs it did. Those metrics could potentially be used but that would require a multi-step manual process that revolves around going to a point in time of singular

simulations to understand how these effects were achieved. This is not conducive to making statistically significant statements as to why something occurred or changed. There does appear to be a framework indicated in multiple sources revolving around the kill chain, or F2T2EA, that explains the steps required associated with warfare.

## 4.2 Kill Chain (F2T2EA)

The kill chain is the general process that explains the intermediate steps in combat. The kill chain takes on many forms but most generally comprises of 6 elements: find, fix, track, target, engage, assess. Using these elements, one can describe from beginning to end, the necessary steps that must be taken to enact change on the battlefield. There are variations on this. Most commonly with the addition of elements to the beginning or end of the set. Biltgen cites a change to the process by AFRL where the addition of 'anticipate' was placed in front and 'anyone, anytime, anywhere' were placed at the end.[17] Making it AF2T2EA4. These additions were made to reflect the research areas at the time that were of interest to fill what had been identified as capability gaps.[17] Functionally, F2T2EA is still the core components related to the process of enacting change on the battlefield. Notably, even though the focus of F2T2EA is typically kinetic engagements as will be shown, it is a framework that can apply to electronic warfare and even search & rescue operations. With this in mind, demonstrating some past usages of the kill chain is necessary to understand how it could be used to further the goal of providing a means to understand why something has occurred.

Work on F2T2EA as a modelling regime has primarily focused on providing the end user with the capability to determine if a mission can be done successfully in a certain amount of time. In Bloye, an air-to-ground mission is used where time critical targets are to be engaged. These targets were considered hidden or otherwise unreachable for some length of time. Then, they would become vulnerable, and given the disposition of assets at that time, a model could be sampled to provide an understanding of what targets could be

successfully attacked within the estimated amount of time available. Given that information, the assets could be appropriately dispatched to achieve the greatest effect. In this context, Bloye slightly refactors the kill chain into seven elements: detect, track, identify, approve, launch, control, assess. This refactor allowed for more precise definitions to be used that more closely aligned with the steps necessary for the attacks to take place. Of particular note, the addition of an approval step. In this step, the information up to that point is sent to some higher authority, typically at the command and control (C2) level, who would then parse the information and determine if the kill chain could proceed. A particularly useful description of the kill chain that explains this refactor is as follows:

"kill chains focus on the target, not a particular attacking platform. The phases of the kill chains are not things that a platform must do to attack a target with success, but instead things that some combinations of platforms for each target must achieve." [20]

Using an imaginative air-to-ground strike in the mythical country of Ithaca, Bloye was able to provide a tool, that once a target became accessible, a set of probabilities could be calculated based on the distance, speed, and munitions profile of the attacking asset. Each asset would have its probabilities assessed for each target, and the appropriate vectoring of each strike could be made that maximized the number of strikes that could reasonably be made.

Similar to Bloye, Jacques performed a similar analysis. As already pointed out, Jacques formed an equation made up of a set of probabilities that describes the ability of a WAS munition to complete a mission. Without stating it, Jacques quantifies the find stage into a single probability described as the chance of encountering a potential target. The next step is the fixing of the potential target as something that could be hit and that is contained within the probability that it could be detected. The track and target stages are wrapped up into one probability labelled as the correct identification as an adversary to be attacked. From there, the engage stage is placed into the probability that the attack is successful. The

product of all these probabilities, would be the value associated with the assess stage. Using this explanation, and previously explained information, an analyst could potentially utilize 4.1 to begin to explain why a new technology causes a change in the final value. All without directly mentioning the kill chain but very closely following it.

Similar to Bloye and Jacques, Farrell focuses on modelling the kill chain using probabilities of each step. Farrell attempts to take a broader approach that would allow for greater applicability than Bloye. Beginning from a math and statistics approach first Farrell puts forth a framework that would allow for a user to define each step in the kill chain as its own probability function. Be it a Gaussian, Poisson, Erlang, or even Dirac Delta, and using empirical data to determine the necessary coefficient values that would be used in each equation. These functions could be sampled to determine the amount of time needed for that step. Farrell warns against simply using the means and standard deviations across all the steps to determine whether the kill chain can be completed in time. Rather, by allowing each step to be an independent variable with its own unique distribution, a convolution of each step can be performed. By doing this, a final graph can be produced that represents the probability of completing the kill chain within a certain amount of time. This would mean an analyst simply need to input the estimated amount of time a target is vulnerable for and given the asset laydowns to determine the probability of success. This is a much quicker process. The described process that Farrell's process beats is one where the means and standard deviations for each step are added up. This assumes a Gaussian distribution for each step which doesn't always accurately represent the underlying process. The resulting value would be a mean with some standard deviation that is not easily interpretable in high stress situations to quantify mission success.

## 4.3 What Have We Learned

A review of the literature on analytical wargames, it is evident, determining how a scenario works is not well known nor has it been pursued. What these processes have shown however, is that there does appear to exist a framework that can be considered the general steps to achieve an outcome in an engagement. This being F2T2EA. Subsequent analysis of the available literature on F2T2EA shows that it has predominantly been used to describe whether a mission can be completed in a certain amount of time. The combining of F2T2EA along with analytical wargaming has the potential to be the guiding framework necessary to perform the variable reduction so that a reasonable and programmatic analysis can be performed. With this combination being the guiding framework, the question arises of what variables to record that represent the state of the system at some time. Thus, understanding what a state means is necessary.

The work from Wasson is useful in understanding this. Wasson recognizes that there is a controversy in systems engineering when it comes to the definitions of phases, modes, and states.[21] It is pointed out that these problems arise from:

- Lack of explicit standard definitions from the relevant authorities
- Inadequate training on the meaning of these terms at the academic and industrial level
- Misuse and interchanging of terms by organizations and managers

After analyzing various sources, Wasson settles on the following definitions. Phases are designations applied to segments of the system or life cycle. Modes are abstract labels applied to a user selectable option. States have a more nuanced definition depending on the context of the usage. There are four contexts: system, operational, dynamic, and physical. In a system state, a system is employed as an asset. For operational states, these are used to describe the operational condition of the state. In dynamic states, these describe the time-based rate of change and environment-dependent dynamics of the system. Finally, in

physical states, these describe the actual physical orientations of system components that are achieving some directly measurable outcome.

To apply this information to what is being accomplished here, the kill chain categories would be best labelled as phases of the system. As they describe points in the life cycle of the mission being analyzed. The recorded information that will be used to more granularly explain the system and used in the building of the Markov chains would be best labelled as operational states. Now that there are clear definitions, clearly communicating what is being done can happen.

## 4.4 Implementation

Multiple different scenarios and missions have been discussed. To demonstrate using F2T2EA as a guide for down-selecting variables, consider two different scenarios.

1. Simple air-to-air one versus one
2. Air-to-ground one versus one

Using the F2T2EA framework, the number of variables can be generally reduced without losing too much information. Given that the kill chain phases can be made into action verbs of the attacking agent, one would assume that when recording variables in the simulation, it's best to assign whether or not a phase is active to the attacking agent. However, these phases should rather be made into past tense verbs. In this form, instead of asking whether an attacking agent has found a target, the question would shift to whether the possessive platform has been found by someone. In this form, it allows for easier understanding of events later in analysis. Uniquely, by posing the framework in a possessive form, it requires the usage of a white cell in the simulation who is capable of looking into the perceptions of each sides platforms and determine whether certain conditions are met. This is making use of the ground truth.

Determining what should be considered for the air-to-air engagement with the framework of the kill chain is fairly straightforward. In the scenario, it can generally be considered that both aircraft are in the find phase as a function of the nature of the scenario. For the fix phase, the white cell determines if the perceived radar contact by the other team is in fact the unit in question. Similarly, the target phase is achieved when the white cell determines that the unit has correctly classified the perceived radar contact. The engagement phase can be simply understood as whether or not a missile has been shot at a unit or is currently being shot at. If necessary, further allowances can be made to increase the number of states that describe each phase.

In this air-to-air engagement, it would be conducive to create two different states that individually define whether or not a unit is being fired upon, i.e. target, or is the unit firing, i.e. shooter. The final phase, assess, can be a state that defines whether or not the unit is alive. By extension, should it be desired and due to the way that each phase's question is asked, each missile fired has the ability to hold its own set of states. This can provide a tunable level of granularity. Subsequently, this allows for Markov chains to be created for sets of states on a platform as well as sets of platforms. A greater explanation of this will be discussed in later chapters in the more in-depth example. In the air-to-ground, or ground-to-air depending on the framing, scenario follows a similar set of questions. Fundamentally, the questions asked do not care about the type of domain that the unit itself exists in. The differences are more likely to come with respect to when within the simulation values are flipped. In an air-to-air engagement of similarly equipped platforms, the phases of fix and target, will most likely transition at very similar times. The stochastic nature of the simulations will typically be more evidently borne out in the engagement components of the chain with regards to when and whose state flip first. In the air-to-ground scenario, a ground station will be able to hold a much more powerful radar that will have the ability to detect and also classify an airborne target much earlier than a platform equipped radar could. In this case, it is much more likely that the states for fixed and targeted will be flipped earlier

than the ground stations. These are simple examples for which a user could define states to be recorded using the kill chain framework to inform them of the questions to ask in formulating the definitions for each state of interest.

## 4.5 Scalability

The process of down-selecting the information being recorded into a set of features, is still not without the problem of scale. For every selected feature and platform created, there is an exact number of possible permutations of that platform and features. This occurs for every platform. These unique permutations are then combined themselves into new sets of permutations and connected together into a Markov chain.

When defining each feature to be recorded, the inter-dependency of these features must be understood. For example, if the alive status and detection status of a platform is being recorded, using the definitions from before, it follows that the unit must be alive to be detected. This inter-dependency must be kept in mind when creating different features so as not to prescribe types of connections artificially. In the case of the alive status of a platform, this should normally flow from on to off within a simulation. The detection status however can flow between being on and off depending on the time in the simulation. These features can be defined into different types based on the number of substates they contain and the direction of flow between those. Table 3 defines five different types of features.

Table 3: Description of types of features based on their number of substates and direction of transitions between those substates with examples

Type	Number of Substates	Direction	Example Progression	Example State
I	Binary	Unidirectional	$1 \rightarrow 0$	isAlive
II	Binary	Bidirectional	$1 \leftrightarrow 0$	isDetected
III	Non-Binary	Unidirectional	$2 \rightarrow 1 \rightarrow 0$	howDamaged
IV	Non-Binary	Bidirectional	$2 \leftrightarrow 1 \leftrightarrow 0$	howDetected
V	Non-Binary	Multidirectional	Many Examples	–

The type of feature is determined by the number of substates, binary or non-binary, and the allowed transitions between those substates, unidirectional, bidirectional,

or multidirectional. Type V features are those that cannot be defined exactly in the other types. As a means to focus this writing, Type V features won't be further explored or modelled. In the subsequent models and equations, it is assumed that one type of feature is being modelled at any one time.

The first step in understanding the effects of different types, is to determine the permutations that can arise from their implementations. Take a scenario that contains seven units where each unit is recording a binary-type feature. There are 128 different permutations of the resulting states. In the flip case, two units with non-binary features that can take on seven different substates. The number of permutations is only 49.

$$C = n^m \quad (4.4)$$

The number of permutations ( $C$ ) that comes from the number of units ( $m$ ), containing singular features, defined by the number of substates ( $n$ ) that feature can take, is defined by Equation 4.4. The number of permutations can also be thought of as the number of unique nodes that could be contained within the resulting graph of those units and features. Equation 4.4 is more sensitive to the number of units that contain the same feature than it is to the number of substates a feature can take. For example, Figure 7 shows that a graph from 2 units that can take on 50 states each results in only 2500 nodes. Inversely, 50 units with only 2 substates each creates a little over  $10^{15}$  nodes.

This information is useful to know when moving up the scale and resolution pyramid to campaign level simulations where the number of units is high. In simulations such as these, it would be advisable to keep the number of potential substates for each unit lower. Now that the number of nodes can be calculated, the number of edges is needed. This is a difficult task to perform as the type of graph that can be outputted, for example a rook graph of the form  $L_{n,n}$  and hypercube graph of the form  $Q_m$  can be created by flipping the number of substates with the number of units with that feature. Figure 8 demonstrates these different structures. The rook graph on left contains 24 edges while the hypercube graph on right

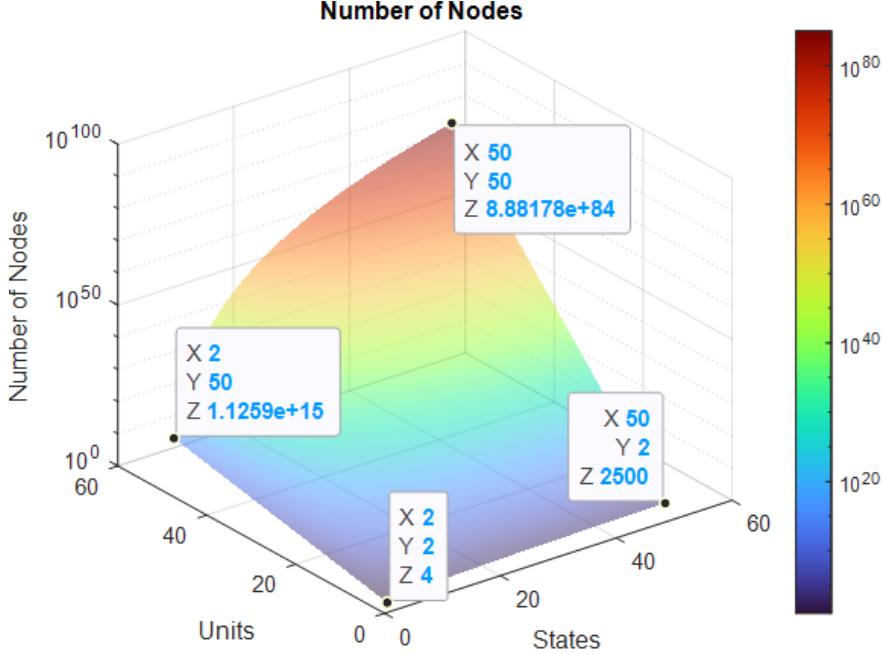


Figure 7: Number of nodes depending on the number of substates a feature can take and the number of units that contain the same feature

contains 32.

Figure 8 also demonstrates the naming process for determining edges that can and cannot exist. The goal is to represent a single subststate value change for each edge. To determine if an edge should exist between two nodes, the nodes are defined by a vector. Each position in the vector represents the number of units. The value in that position is the subststate value for that unit. All unique permutations can then be generated. It is then a simple process of starting at the vector consisting of all units having the highest possible value for the subststate. From this point, a units value can be declinated, the subsequent node representing that new vector found, and the position in the adjacency matrix representing that edge set to 1. This process is general to be useful for all permutations of units and subststates.

Graphs of permutations were made holding either number of units or subststates constant. After creating each graph, the number of edges could be counted and stored. To reduce the number of variables to plot, a ratio,  $R$ , was made between number of units,  $m$ , and number of edges counted,  $E$ . Plots could then be made for each number of subststates,  $n$ ,

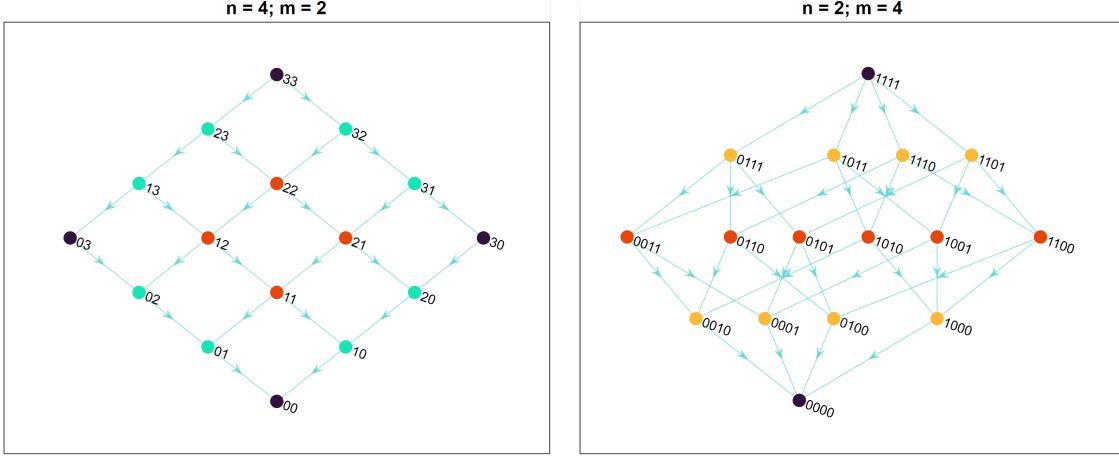


Figure 8: Demonstration of the different structures between flipping the units and substates with a rook graph, left, and hypercube graph, right

with the variations in the ratio based on the number of units. These plots indicated perfect exponential curve fits.

$$R = Ae^{-Bm} \quad (4.5)$$

The general form of the curve fit is Equation 4.5. The coefficients for five different curve fits corresponding to the substate values of  $2 \rightarrow 6$  were then recorded, Table 4.

Table 4: Coefficients of the exponential curve fitted plots of the ratio  $R$

State (n)	A	B
2	1	-0.693
3	1.5	-1.099
4	1.33	-1.386
5	1.25	-1.609
6	1.20	-1.792

The leading coefficient,  $A$ , is a ratio of the number of substates over the number of substates minus one. The raised coefficient,  $B$ , follows a near perfect natural logarithm. This fit suggests, given an analysis in Excel, that the additional term beyond the algorithm is an artifact of the floating point multiplication Excel performs when calculating the curve fits for the previous plots. Thus, the additional term can be reliably disregarded, and defining the relationship of  $B$  to be natural logarithm of the number of substates.

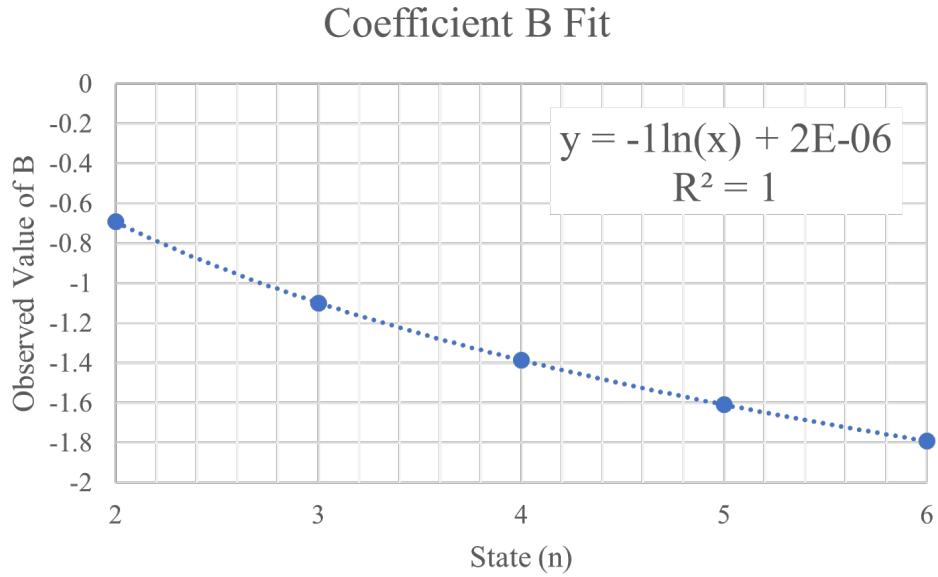


Figure 9: Curve fit of the calculated values of  $B$

Given a general curve fit for the ratio of  $R$ , values can be substituted and algebraic manipulations made to back out the equation for the number of edges. This function being a combination of the number of units with the feature defined by its number of substates. This results in Equation 4.6.

$$E = \frac{m(n-1)}{n^{1-m}} \quad (4.6)$$

Armed with this equation, it is found that it closely resembles the equation for a Hamming Graph. A Hamming Graph, as defined by Wolfram, as the graph Cartesian product of a complete graph with itself some number of times. [22]

$$H(m, n) = \frac{m(n-1)}{2n^{-m}} \quad (4.7)$$

Comparing Equations 4.6 and 4.7, it can be seen that the difference is in the denominator. The Hamming equation can be recovered from Equation 4.6 when the number of substates is set to 2. That is to say, when the graph takes on the shape of a rook graph.

When the number of substates is greater than 2, Hamming's equation no longer works.

Much like with Equation 4.4, Equation 4.6 is much more sensitive to the number of units present than it is to the number of substates each unit can have.

---

---

# CHAPTER 5

---

## HOW TO IDENTIFY

### 5.1 What Makes Pivotal

A key objective of this research is identifying key points in time that can be considered pivotal in the stream of events. Up to this point, the assumptions made are that a scenario has been selected and has been accurately programmed into AFSIM. For the scenario in question, the kill chain framework's set of questions have been asked and answered in Table 10, and were recorded during the simulation. The simulation has been run to a desired level of convergence and the results have been aggregated into a meta-Markov chain. In this Markov chain, every node consists of the set of platforms contained in the simulation. For each platform, there is a set of states that describe the position of that platform in the kill chain. Every edge leading out of the node exists to demonstrate that at least one state for one of the platforms transitioned to a different value. Thus, having the ability to identify what node is important in the entire set of states is useful.

To identify what states could be considered pivotal, understanding what information is accessible to the analyst is necessary. In the Markov chain format, the simulation can also

be labelled as a directed graph. There are four types of information that can be utilized. The first is the values of the individual states held by each platform. For example, if a state describes whether or not a platform is alive, knowing the nodes that indicate a platform has died could be considered pivotal nodes. Next type takes into account some larger portion of the set of states for each platform that make up the node. In this case, knowing if a shooting platform is not detected by the targeted platform could be considered a pivotal point in time. Moving into a larger scope, local graph properties could indicate pivotal information. Local graph properties are the properties associated with the information derived from the edges going into and coming out of each node and the differences of that node and it's neighbors. A pivotal node could be indicated by having large amounts of input edges. This would mean that many other nodes feed into this node and could be indicative of a choke point in the simulation. Finally, there are global graph properties. These properties explain the placement of the node with respect to all other nodes and their connections to every other node. This property is what most likely is thought of when considering normal node identification.

## 5.2 History of Node Identification

The format that the simulation information has been put into can be considered a directed graph. Thus, there is potential that traditional graph analysis methods could be useful in identifying nodes of interest. The field of sociology heavily uses graph analysis in the form that is useful to this research. As Freeman explains in the paper *Going the Wrong Way on a One-Way Street: Centrality in Physics and Biology*, the older sciences, physics/chemistry/biology, have borrowed centrality analysis from sociology when tradition indicates that sociology should be borrowing from them and generally does![23] Centrality analysis is the defining of nodes in graph based on a set of necessary criterion contextually necessary to the graph. [24]

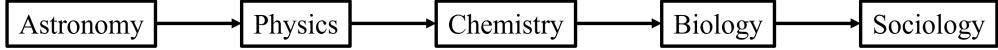


Figure 10: Progression of scientific techniques through the sciences as proposed by August Comte and described by Freeman [23]

Centrality analysis as it is known today originates from work in the MIT Group Networks Laboratory in the late 1940s.[23] Work has continued in this field predominantly by sociologists in the analysis of importance of individuals in webs of friendships and connections. After nearly sixty-years as a nearly exclusive sociological field, a landmark work published in *Physical Review E* by a set of physicist make a direct and pointed use of centrality analysis to quantify the degradation of network performance when certain nodes are removed.[23] As per Freeman, some work had been done in internet packet switching, power grid analysis, and general circuit analysis but nothing had the effect like the work published in *Physical Review E*.[23] From here, a wave of adoption occurred with utilizing centrality analysis. Most notably, the field of biology has adopted centrality analysis in the identification of important points in amino acid residues (AARs), protein-protein interaction networks (PPINs) and general metabolic networks.[23] Biologists had previously identified certain nodes contained in these graphs that had a 'pivotal' part to play in the broader scope. It was found that centrality analysis had the ability to identify these nodes just using the information contained in the graph.

Arguably, centrality analysis is a young field. Even though it is young, three fundamental categories of centrality analysis have been determined. These are:

- Degree
- Closeness
- Betweenness

Centrality analysis allows for the creation of new metrics to score nodes within a network depending on the context of the graph. However, those metrics can be functionally decomposed into some combination of the three previous metrics. These metrics can be generally

defined as follows. Degree is the measure of the number of edges corresponding to a node. In a directed graph, this measure is further split into inc degree and out degree representing the number of incoming/outgoing edges. Closeness is the average number of edges that must be traversed to reach the edges of the graph from the node in question. This measure generally does not take into account the weight/length of the edge but rather that the edge itself exists in the first place. Betweenness can be simply defined but is the most complex of the three. Defined, it is the measure of the amount of control a node has on the flow through the graph.

All of these measures can be normalized to a theoretical graph so that comparisons across graphs can occur. This is done by refactoring the graph to care about only certain portions. To do this means to disregard the weight of the edges between nodes and turn the graph into an undirected star graph, in the cases of degree and betweenness, and a line graph for closeness. By doing this, the centrality measures must only be applied to the center of the point. For degree, the maximum possible number in graph is simply  $n - 1$ . Where  $n$  is the number of nodes in the graph. Since a node in this format cannot connect to itself, subtract one from the value to achieve the value necessary. Thus the equation for normalized degree is below with the numerator coming from Nieminen and denominator from Freeman.[24]

$$C_D(p_k) = \frac{\sum_{i=1}^n a(p_i, p_k)}{n - 1} \quad (5.1)$$

Where the function  $a(p_i, p_k)$  takes on the value of 1 when the edges between  $i$  and  $k$  exist and 0 otherwise. The closeness measure uses a slightly different form of the theoretical graph as previously explained. Though, elegantly, it appears very similar to degree. Beauchamp provides an equation to calculate the normalized value of closeness in a graph. [24]

$$C_C(p_k) = \frac{n - 1}{\sum_{i=1}^n d(p_i, p_k)} \quad (5.2)$$

In this equation  $d(p_i, p_k)$  is defined as the geodesic distance, or smallest number of edges, needed to traverse the graph between nodes  $i$  and  $k$ . For closeness, the maximum value that

can be achieved by a node in a graph of size  $n$  is to be at the end of a single chain of nodes singularly connected to each other where the number of edges between the beginning and end of the chain is  $n - 1$  edges. Moving to the more complex betweenness metric. This requires a slightly more in depth mathematical set up. Consider a graph where the number of shortest paths between any two nodes is represented by  $g_{ij}$ . The function  $g_{ij}(k)$  is the number of the shortest paths in  $g_{ij}$  for which the node  $k$  exists as an intermediate node that is not the nodes  $i$  or  $j$ . From here, the function  $b_{ij}(p_k)$  can be considered the ratio of the number of shortest paths for which  $k$  exists as an intermediary in the set of paths  $g_{ij}$ .

$$b_{ij}(p_k) = \frac{g_{ij}(p_k)}{g_{ij}} \quad (5.3)$$

Using this function, it is simple to then determine the total betweenness of a node in a network by summing the values of  $b_{ij}$  across all sets of  $i$  and  $j$ . Freeman proved that the equation for the maximum value for which a nodes betweenness can take occurs in the star graph of the same size  $n$ . [24]

$$\frac{n^2 - 3n + 2}{2} \quad (5.4)$$

Thus, the normalized value for betweenness is:

$$C_B(p_k) = \frac{2 \sum_{i=1}^n \sum_{j \neq i} b_{ij}(p_k)}{n^2 - 3n + 2} \quad (5.5)$$

These sets of equations and descriptions of the functional components of centrality measures provides a baseline for which further centrality measures can be created.

### 5.3 Example Usages

Some example usage of centrality measures and their pros and cons are necessary before being able to determine valuable measures for the context of this research. In sociology, most

networks that are analyzed consist of people, nodes, that know each other through various means, edges. In sociology however, determining through other means, the importance of nodes is difficult and highly subjective. In biology however, there is the ability to determine importance of a node based on the ability to potentially remove a node or prevent the traversal of edge and see the measurable cascading effects of this on the system. A sizable amount of literature exists on the exploration of centrality measures on PPINs. The work by Ashtiani et al. is useful in exploring the pros and cons of different measures in real world systems. An analysis was performed to gather as many different applicable algorithms that existed for identifying important proteins. Ashtiani et al. were able to categorize the different algorithms into five groups. These are distance, degree, eigenvalue, neighborhood, and miscellaneous. These can be seen in Table 5

Table 5: Example centrality measures adapted from [1]

<b>Distance Based</b>	Average Distance	<b>Degree Based</b>	Authority Score
Barycenter			Degree Centrality
Closeness Centrality (Freeman)			Diffusion Degree
Closeness Centrality (Latora)			Kleinburg's Hub Centrality Scores
Decay Centrality			Leverage Centrality
Eccentricity of the Vertices			Lobby Index (Centrality)
Lin Centrality		<b>Eigen Based</b>	Eigenvector Centralities
Radiality Centrality			Katz Centrality (Katz Status Index)
Residual Closeness Centrality			Laplacian Centrality
<b>Neighborhood Based</b>	Cluster Rank	<b>Miscellaneous</b>	Information Centrality
Density of Maximum Neighborhood Component			Markov Centrality
Maximum Neighborhood Component			Shortest-Paths Betweenness Centrality
Subgraph Centrality Scores			Geodesic K-Path Centrality
			Harary Graph Centrality

Of note, degree, closeness, and betweenness centralities are all present in this graph within their own categories. Quickly explaining the two categories that do not contain the three previously defined metrics. For eigen-based metrics, these are generated by calculating various values based on the different eigenvectors associated with the adjacency matrix associated with the graph. These matrices are of the same format as the matrix that explains a Markov chain but they are not held to the right stochastic requirement. In undirected formats, adjacency matrices consist of only ones and zeros. Ones represent the existence of an edge between two nodes and are symmetric about the diagonal. This makes calculating

real eigenvalues and vectors easy and is where the application of these metrics exists. In the directed case, it is not required to be symmetric. This makes calculating eigen-based centrality measures very difficult without using the complex domain. For those reasons, eigen-based measures are not used in this research.

The second category is neighborhood-based measures. These measures attempt to quantify a node within its position within a local graph and then the position of that local graph in the global graph. For example, in the cluster rank metric, the definition provided by the centrality metric aggregation website CentiServer, attempts to quantify a nodes influence on its neighborhood by the number of followers it has, and a clustering coefficient.[25] In the cases of maximum neighborhood component (MNC) and density MNC, the neighborhood of a node is based on the largest component of a graph that the node belongs too.[26][27] In the Markov chain example, component is the same as class. With these measures, they typically place the similar if not the same values on sets of nodes. Since it is the goal of this research to highlight select nodes with specific information, these types of measures are not directly pursued.

The measures listed were programmed and ran so that each node in the PPIN could have a score for each. Then, correlation analysis was performed to explain any inherent connections between the metrics in the case of the network analyzed. These tables are reproduced in Table 6, Table 7, and Table 8.

Table 6: Comparison of distance based centrality measures adapted from [1]

#	Name	1	2	3	4	5	6	7	8	9
<b>1</b>	<b>Average Distance</b>	-0.96	-0.93	0.59	-0.96	-0.94	-0.96	-1	-0.93	
<b>2</b>	<b>Barycenter</b>		0.99	-0.6	1	0.99	1	0.96	0.99	
<b>3</b>	<b>Decay Centrality</b>			-0.6	0.99	1	0.99	0.93	1	
<b>4</b>	<b>Eccentricity of the Vertices</b>				-0.6	-0.61	-0.6	-0.59	-0.6	
<b>5</b>	<b>Closeness Centrality (Freeman)</b>					0.99	1	0.96	0.99	
<b>6</b>	<b>Closeness Centrality (Latora)</b>						0.99	0.94	1	
<b>7</b>	<b>Lin Centrality</b>							0.96	0.99	
<b>8</b>	<b>Radiality Centrality</b>								0.93	
<b>9</b>	<b>Residual Closeness Centrality</b>									

In retrieving the data, there was no way to create a table to display the correlations

Table 7: Comparison of degree and eigen based centrality measures adapted from [1]

#	Name	1	2	3	4	5	6	7	8	9
<b>1</b>	<b>Authority Score</b>		0.88	0.96	1	0.39	0.9	1	0.95	1
<b>2</b>	<b>Degree Centrality</b>			0.97	0.88	0.62	0.98	0.88	0.97	0.88
<b>3</b>	<b>Diffusion Degree</b>				0.96	0.49	0.97	0.96	0.99	0.96
<b>4</b>	<b>Kleinburg's Hub Centrality Scores</b>					0.39	0.9	1	0.95	1
<b>5</b>	<b>Leverage Centrality</b>						0.59	0.39	0.5	0.39
<b>6</b>	<b>Lobby Index (Centrality)</b>							0.9	0.95	0.9
<b>7</b>	<b>Eigenvector Centralities</b>								0.95	1
<b>8</b>	<b>Laplacian Centrality</b>									0.95
<b>9</b>	<b>Katz Centrality (Katz Status Index)</b>									

Table 8: Comparison of neighborhood and miscellaneous based centrality measures adapted from [1]

#	Name	1	2	3	4	5	6	7	8	9
<b>1</b>	<b>Cluster Rank</b>		-0.05	0.91	0.89	0.42	-0.02	0	0.92	0.11
<b>2</b>	<b>Density of Maximum Neighborhood Component</b>			0.02	0	0.12	0	0.01	0.02	0.04
<b>3</b>	<b>Maximum Neighborhood Component</b>				0.84	0.55	-0.02	-0.01	1	0.35
<b>4</b>	<b>Subgraph Centrality Scores</b>					0.31	-0.02	-0.01	0.85	0.13
<b>5</b>	<b>Geodesic K-Path Centrality</b>						-0.01	-0.02	0.56	0.32
<b>6</b>	<b>Harary Graph Centrality</b>							0.58	-0.02	-0.01
<b>7</b>	<b>Information Centrality</b>								-0.01	-0.03
<b>8</b>	<b>Markov Centrality</b>									0.35
<b>9</b>	<b>Shortest-Paths Betweenness Centrality</b>									

amongst all five categories simultaneously. Using the provided data, the three tables could be created. These are sectioned so that Table 6 is entirely distance based measures. Table 7 is degree and eigen based. Table 8 is neighborhood and miscellaneous. It would be accurate to presume that measures that are contained within the same category would be positively correlated. As made evident, in the distance based metrics, there appear to be a few exceptions to this rule. However, upon closer inspections of these values, it is realized that these measures use inverses that place the increasing metric in the denominator. Therefore, the values determined by these metrics will approach zero rather than infinity like their counterparts. Thus, their negative correlations.

Understanding the information in Table 7 for degree and eigen based measures, all measures are correlated. Positively correlated at that. This is unique as it would be assumed that across different categories, there would be at least some discrepancy to indicate the necessity for the unique categories. It appears as though, that most of the metrics between

the two categories will increase together and that the differences between them are more nuanced and focused on the values themselves. Unlike the degree and eigen based metrics, the neighborhood and miscellaneous measures illustrate the expected non-correlation between categories and stronger inter-categorical correlations. Notably, harary centrality and information centrality measures show near zero correlation with nearly all the other metrics compared against them. This would indicate that the usage of these two metrics would indicate different nodes of importance within the same graph. Thus, if a user wanted to select these metrics, understanding the underlying concepts as to what will be brought out is necessary to be able to make an informed decision. For example, the harary measure is very similar to eccentricity in Table 7 and is a measure of the inverse of the distance between a node and the furthest reachable node.[28] In information centrality, the goal is to create a measure that will highlight nodes that exist as the terminal point on many short paths taking into account the number of connections and the inverse of their length.[29] With this in mind, in a PPIN where being more centrally located, in the physical sense of the word, the harary measure would be a better option between the two metrics. If it is desired to nodes in the PPIN where the protein representing node is at the end of multiple short processes, information centrality would be used. This demonstrates the diversity in the number of measures that already exist. Different nodes will be scored based on the way each measure is set up, and thus, measure selection is contextually dependent on the graph at hand and the desires of the analyst.

The above information is for one sample PPIN. A question to ask is, are measures correlated across different networks? That question is asked directly by Valente et al. which focused on fifty-eight different social networks that had been previously catalogued. These networks varied in size and other parameters given that the questionnaires answered to create the networks focused on different phenomenon.[30] Using these networks, twelve different metrics were measured for each node in a graph. A correlation table was then made for each graph across the different metrics. Across all the networks, the correlation table values were

averaged. A mean and standard deviation for these networks were also calculated. The metrics measured included the standard degree, closeness, and betweenness metrics. Furthermore, much like with Ashtiani et al. various forms of radiality and eigenvector were calculated as the graphs were undirected.

Table 9: Average correlation values across n=58 social networks. Adapted from [30]

#	Measure	1	2	3	4	5	6	7	8	9	10	11	12
1	Indegree	-											
2	Outdegree	0.3											
3	Degree	0.78	0.71										
4	Between	0.62	0.54	0.7									
5	S-Between	0.69	0.5	0.85	0.67								
6	Closeness-In	0.55	0.16	0.45	0.37	0.31							
7	Closeness-Out	0.18	0.81	0.56	0.39	0.38	0.02						
8	S-Closeness	0.4	0.64	0.66	0.37	0.44	0.42	0.65					
9	Integration	0.7	0.26	0.58	0.5	0.41	0.9	0.15	0.51				
10	Radiality	0.21	0.86	0.61	0.44	0.41	0.06	0.98	0.67	0.19			
11	S-Int/Rad	0.45	0.7	0.73	0.43	0.5	0.44	0.69	0.99	0.54	0.72		
12	Eigenvector	0.71	0.69	0.92	0.64	0.72	0.44	0.55	0.63	0.57	0.59	0.71	
-	Average	0.51	0.56	0.69	0.52	0.53	0.37	0.49	0.58	0.48	0.52	0.63	0.65
-	Standard Deviation	0.21	0.23	0.14	0.16	0.14	0.27	0.22	0.25	0.28	0.17	0.12	0.12

Looking at the Table 9, it is interesting to see that across all networks, the average individual correlations are all positive. Some average correlation values across the networks that are near 0 are the correlations between what would be considered different categories of metrics. Some examples being between closeness & degree and radiality to both degree & closeness. Betweenness in these networks are somewhat more strongly correlated to the other categories of metrics when compared to degree and closeness. This can potentially be explained in that the flow of the network that betweenness tries to capture is not nearly as strong a phenomenon across the set of networks analyzed. Valente et al. provides the average correlation and standard deviations associated with the metrics across all networks and then across all metrics. With this information, it is more evident that the betweenness metric is unique across the other metrics. Whereas closeness and degree centrality scores have some very evidently low correlations, when taking into account all correlations, they have a roughly average level of correlation.

As an example to demonstrate the potential usage of metrics as a means of determining actions to be had. Christakis et al. performed a long term study to see if they could identify quantitatively the spread of obesity. By creating a network where nodes represented individuals and then the edges indicating the social connections that two individuals, nodes, had with each other and, after controlling for various necessary parameters such as genetics and socioeconomic groupings, they found that using a form of the closeness metric could demonstrate the potential affect a person would have on those around them.[31] It was estimated that at only one degree of separation from an obese individual, the chance that the individual in question would become obese themselves was  $\approx 50\%$ . At just three degrees of separation, this had decreased to  $\approx 10\%$  and at six degrees of separation the probability was  $\approx 0\%$ . From this information, it was concluded that stopping the spread of obesity could be realistically borne out by intervening at the individual level in social groups. As another interesting note, a similar network was created representing physical locations of individuals with respect to the other individuals in the study. It was determined that the geographical separation of individuals played a negligible role in the spread of obesity.

## 5.4 Building the Metrics

Armed with the knowledge of what is being recorded in a scenario using the kill chain framework, the way that information is encoded in the subsequent Markov chain, the components of centrality analysis and, how they have been used together, a list of objectives can be made to inform what the metrics should do. Each of the metrics should attempt to achieve as many if not all of the following objectives:

1. Contain physical meaning
2. Make use of the fundamental components of centrality analysis
3. Contextually relevant

#### 4. Generally uncorrelated

Starting with Objective 1. Without some underlying physical meaning derived from the information encoded, the results of the metric become unusable. Following this objective generally outlaws the usage of eigen-based metrics. As stated previously, the adjacency matrices associated with directed graphs are highly asymmetric which leads to eigenvectors and eigenvalues that contain complex components. These complex components cannot be easily connected to a physical meaning in the graph. With this in mind, they are not continued to be explored as usable in this research. Objective 2 focuses on using the principles of centrality analysis as the building blocks for the metrics. There is no need to reinvent the wheel if there is not a need to. Objective 3 may seem redundant when in conjunction with Objective 1 but, they are different. In the wargaming scenarios discussed here, the broad goal is to identify key points in time that could change the outcome. Objective 3 provides a bounding box that moves designs towards metrics that will do just that and not create metrics that will identify other, potentially irrelevant nodes. Finally, Objective 4 is to ensure that the number of calculations necessary to tell a clear picture of what is considered important and what is not is kept low so that in larger graphs, the computational time does not take unnecessarily too long.

Beginning with the first fundamental element of metric design, degree. In degree and out degree metrics are proposed. These measures utilize the local graph properties of a node. Nodes of high in degree, considering this is a Markov chain, would indicate a node that is very likely to be reached by its immediate neighbors. Nodes of high out degree indicate a node with a multitude of potential actions that could be taken. On the low end for either metric, the lowest value in degree can take is zero. This would indicate a node that is only ever a starting node that is never reached again. For out degree, the lowest possible value is one. A node with an out degree of one is self-looping and, in the case of this research, indicates that a node is the final set of states achieved in a simulation. These metrics make use of the first three objectives and should fulfill Objective 4.

Using the second fundamental form of centrality, closeness, a metric called tree rank is proposed. Closeness as been previously defined makes use of the shortest paths between sets of two nodes. In the case of the Markov chain representation of a scenario, when analyzed for the shortest paths between two nodes, this would result in a path that has no physical meaning. The shortest path would be the minimization of the sum of probabilities. The sum of probabilities would assume that the events in question are mutually exclusive. However, this is not the case since this a model of a Markov chain where the foundational principle is that all events are independent. This would mean the shortest path, or minimization function, should be focused on the product of every probability. To allow for the application of currently accepted shortest path algorithms, such as Dijkstra's, a transformation is required. The transformation is defined below where  $T(e)$  is the value of the edge after the application of the transformation and  $e$  is the weight of the edge originally.

$$T(e) = -\log_{10}(e) \quad (5.6)$$

This transformation allows for the product of edges, defined as probabilities, to be turned into a summation of edge values where high probability edges take on smaller values in the transformed graph. There is however a case that must be taken into account. In the case of an edge that has the original value of 1, the resulting output would be 0. This would mean the removal of the edge in question. This is unacceptable. As a fix to this, since the

$$\lim_{x \rightarrow 1} [-\log_{10}(x)] = 0 \quad (5.7)$$

the replaced value should be as near 0 as possible without being 0. Given that this analysis is taking place on a computer, the smallest value that can be held by the computer is the smallest floating point value available to it. In the case here, MATLAB is used on a 64-bit operating system resulting in the smallest possible floating point number being  $2^{-52}$  or  $2.22 * 10^{-16}$ . Thus, whenever an edge value of 1 is to be transformed, is defaulted into this

value to preserve the existence of the edge while maintaining the approximate size it should for the algorithms to run on.

Now that the edges have been transformed into a form that preserves their information the shortest path algorithms may run. The resulting paths now hold the physical meaning of being the most probable path possible between two nodes. To determine the probability of said path occurring, the inverse of the transformation is applied to the sum. With this information, the metric design can continue. It is posited, that tree rank use the ideas of closeness but instead of being the average number of steps necessary to reach the edges of the graph, it becomes the average number of steps for the node to reach a terminal state. By doing this, Objective 3 will be satisfied. The tree rank of a node would be composed of the following components.

Consider a graph for which the nodes that make up the graph are the set  $N$ . Within  $N$ , there exists some subset of terminal nodes,  $Z$ , for which  $Z \in N$ . For every node,  $z$  in  $Z$ , there exists another subset of  $N$  which contains the nodes that can reach the terminal node in question. Denote this,  $R$ . This subset of nodes can be refactored so that every node in  $R$  appears only once, and is only on the most probable path between itself and the terminal node in question. This would mean that every node would have at most only one out edge, while the number of in edges is determined by the number of immediate predecessor nodes that must go through that node to be on their own most probable path. Thus,  $R$  will appear as tree where the terminal node is the root and every other node feeds into it. An example of this appears in Figure 11. Node 1 would be a part of set  $Z$  and nodes 2 through 6 would be a part of set  $R$ .

Continuing, consider the function

$$b(p_z) = \max d(p_i, p_z), \{i \in R\} \quad (5.8)$$

where the function  $b(p_z)$  returns the maximum number of steps required to take from the most probable path tree of all nodes that can reach node  $z$ . In Figure 11, this value is 2. A

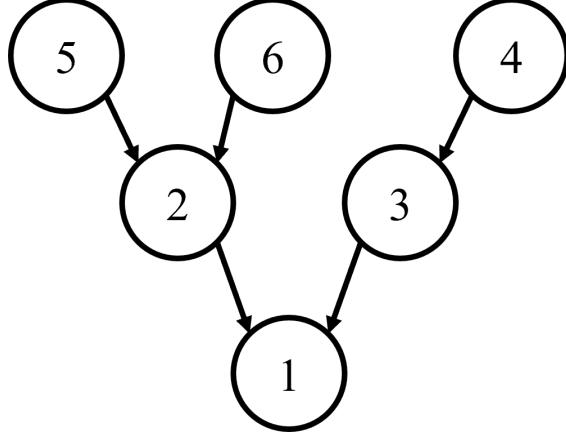


Figure 11: Example graph of the set  $R$

node contained within this shortest path tree would take on a score as follows:

$$tr(p_k, p_z) = \frac{d(p_k, p_z)}{b(p_z)} \quad (5.9)$$

The function  $d(p_k, p_z)$  returns the number of steps from  $z$  that node  $k$  would need to take to reach  $z$ . A fractional value is then returned where nodes that are closer, by the number of steps to  $z$ , will be near zero whereas the nodes that are the furthest from  $z$  will be one. In Figure 11, nodes 6 would have a score of 1 and node 3 would have a score of  $1/2$ . For each terminal node in the set  $Z$ , every node  $N \notin Z$ , is scored based on their positions on the most probable path tree to that terminal node. The final score for a node is then the average of these values across all most probable path trees.

$$TR(p_k) = \frac{\sum_{i=1}^Z tr(p_k, p_i)}{Z} \quad (5.10)$$

Tree rank takes on values  $(0, 1]$ . If a node were to score a 0, this would indicate that the node never appears on any most probable path trees and given the nature of the graph, this is not possible. Even though a value of 0 is not possible, values very near 0 are. These values would indicate that the node in question appears on only one or a few most probable path trees and is most indicative of existing within only one step distance from that node. Nodes

that score near 1 are indicative of existing far away from a large number of terminal nodes if not all of them. As nodes decrease in value closer to 0, this would mean that the likely outcomes that could sprout from this node are becoming more limited and more prescribed.

With metrics utilizing the first two fundamental centrality measures, betweenness is next. The metric as it is currently defined is useful already. However, with a slight modification, this metric can encode even more useful information. Normalized betweenness is the percentage of realized control potential that a node can have in a graph of size  $N$ . In performing simulations and encoding the information into a Markov chain, individual nodes have the frequency of their occurrence recorded. A simple fractional value can be created that relates the number of times a node is observed ( $f_o$ ) in the simulations to the number of total simulations ran ( $f_t$ ). This value provides a an understanding of the number of times the node did control the flow of the scenario. By combining this value and normalized betweenness, a metric is created that quantizes the potential control a node can have when taking into account the set of most probable paths and the number of times this node actually did control the flow. This metric can be called realized potential.

$$RP(p_k) = \frac{C_B(p_k)}{f_o(p_k)/f_t} \quad (5.11)$$

With the metrics, of degree, tree rank, and realized potential, the local and global graph properties are quantified. There is still information that is unutilized. These being the values of the substates and their aggregations into a singular nodal value. The importance of substate values is highly dependent on the scenario being recorded and the desired information that an analyst wants. In the example air-to-air states example, an analyst may desire to rank nodes based on their ability to have been a shooter without being a target themselves. They may also care more specifically if the shooter themselves were not even detected. This type of setup could be used to determine if a stealth fighter concept is working appropriately. Therefore, given the highly variant nature of what could be considered important a metric called analyst score ( $AS(p_k)$ ) can be made that allows for an analyst to input values that

will score nodes based on individual substate values.

It would be useful however, to understand the change to these scores will have across the graph. Given that the information is encoded in a Markov chain, knowing the relative next-step change in analyst score is useful to know. Since every nodes edge weights are the probability of moving to each of the successor nodes, a simple weighted sum of the absolute value of the differences between the analyst scores can be calculated.

$$IP(p_k) = \sum_{i=1}^m e_i |AS(p_k) - AS(p_i)| \quad (5.12)$$

Where, for each node  $k$ , there are  $m$  successor nodes and the edge  $e$  that connects the node  $k$  with node  $i$  is used as the weight in the summation of the differences. In using this metric, an analyst can determine the impact potential ( $IP$ ) a node will have on the analyst score metric. Values of  $IP$  will be highly dependent on the scoring system used in  $AS$ .

## 5.5 Outcome Score

There is still a graph property that has not been utilized yet. This comes from the usage of Markov chains to model the scenario in a discrete form. Remember, Markov chains do not take into account the past when deciding the next step to take. Encoding a scenario in this way could mean that the momentum that is inherent in a scenario is not being taken into account. However, if the state selection is appropriate, the Markov chain model can replicate the chains as they appear in the simulation. If this is the case, one could start at any random point in the chain, begin a random walk, and create an accurate representation of what could happen in the scenario based on the values that have been observed. Markov chains also contain the property that given the existence of a terminal node and enough steps, a random walk will reach a terminal node. With this information in mind, one can score each terminal node based on the level of success at that point. For example, terminal nodes can be scored positively for each friendly platform that remains and negatively for

remaining enemy platforms.

Using this type of scoring system, one could start at a node, and perform a random walk through the chain until reaching a terminal node. Once reaching that terminal node, that score can be recorded and a new random walk can begin from the starting node. Every time a terminal node is reached, the random walk stops and the score of that node is recorded. As the number of walks increase, an expected score can be found with a corresponding confidence interval. Thus, with enough random walks, a score can be given to a node based on what likely result is going to be achieved. This is called the outcome score ( $OS$ ). Similar to  $AS$ , and how a relative change can be calculated in the form of  $IP$ ,  $OS$  can have a score calculated using the same idea and is called the outcome impact ( $OI$ ).  $OS$  makes use of a unique global graph property from Markov chains & substate values  $OI$  makes use of local graph properties.

There are now eight metrics that currently satisfy the first three objectives. From the understanding of the fundamental elements of centrality measures, these measures should generally be uncorrelated and satisfy Objective 4 but will need to be tested to verify this.

## 5.6 Combining the Metrics

Each of the eight metrics pull out different information with regards to each node. Subsequently, this would mean that identifying a pivotal node in the graph is matter of selecting one, some, or all of the metrics. There are a couple of ways that one could combine these metrics together to form a singular, all encompassing, metric. One such way is to do a weighted sum of all metrics across the board. The weights for each metric would need to be determined by the analyst since each metric pulls out different information. It is important to note that when performing summations across metrics that contain different ranges and distributions, it is important to standardize these metrics. That are various ways to do that ranging from simple maximum value normalization to more intricate range scaling and lay-

ered statistical calculations. Some such methods that can be used are asymmetric weighted sum and uniform weighted average using the  $z$ -scores. Using  $z$ -scores, a final metric that uses equal contributions of the absolute value of each metric would create a tiered list. This list would range from nodes scoring near 0, indicating an 'average' node, to some unbounded larger value indicating highly unique nodes that could be of interest for further exploration.

---

---

# CHAPTER 6

---

## EXPERIMENT

### 6.1 The Scenario

The framework and measures previously described were applied and tested in an AFSIM scenario. Multiple difference scenarios were evaluated for feasibility. These were:

- One vs One Air-to-Air
- Two vs Two Air-to-Air
- High Value Airborne Asset (HVAA) Escort
- HVAA Defensive Counter Air (DCA)

The desire of this research was to demonstrate the methodology while balancing run time and complexity in the forms of implementation & analysis time as well as inherent knowledge of the user to confirm randomness viabilities. Each scenario was measured against these to determine the best candidate. To demonstrate the scenarios, a relative deposition of forces has been made for each scenario. They follow the general format of being a top-down view

of the combat area. Inside the combat area are symbols representing the platforms in play. A legend has been provided for reference.

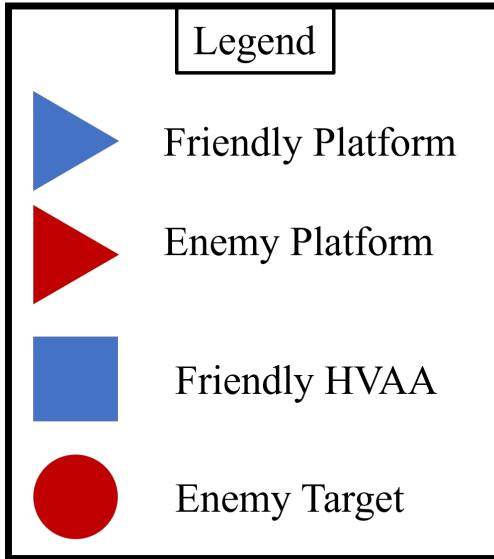


Figure 12: Legend for the Disposition of Forces

Each scenario consists of the same main elements. There are some number of friendly platforms and assets and some number of enemy platforms and targets. The units are color coded to represent their respective sides where blue platforms are friendly and red platforms are the opposition. The shapes utilized demonstrate the general capabilities of the platforms. Similar shapes represent similar capabilities.

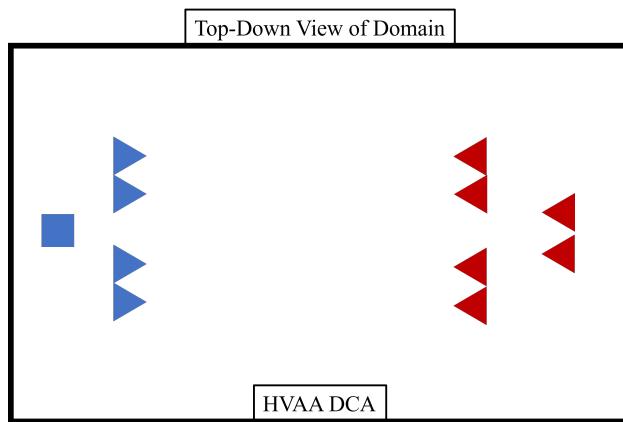


Figure 13: Dispositions for HVAA DCA Scenario

Focusing on the HVAA scenarios initially. The HVAA DCA mission is focused on the

demonstration of the C2 element that can be programmed in. Here, the HVAA is a long-range radar with C2 capabilities over its friendly platforms that are general fighters. The enemy platforms are of similar capabilities to the friendly platforms. The blue forces are outnumbered four to six but do have the long-range radar and C2 capability to be vectored in for better initial starting positions. The red team is on the approach directly towards blue team. The general goals for either side are:

- Blue: Successful defense of the HVAA
- Red: Successful destruction of the HVAA

Unlike the HVAA DCA scenario, the HVAA escort scenario focuses on an offensive mission. Here, similarly, blue team is outnumbered four to six in terms of air-to-air fighting platforms. The blue team does not have the long-range radar and C2 capabilities that the HVAA DCA mission had. The HVAA here is a general bomber with no air-to-air offensive capabilities whose sole goal is the destruction of the ground target protected by red forces. The blue air-to-air platforms mission is the escorting of the HVAA so it can complete it's mission. Should the HVAA be destroyed before a successful attack can be made, the blue team loses. Put simply the goals are:

- Blue: Successful destruction of the ground target
- Red: Successful destruction of the HVAA

Moving to the simpler scenarios. These scenarios are more focused on the replication of air-to-air combat between either single platforms, or platforms and their wingman. The two-vs-two and one-vs-one dispositions are below and are very similar. The main difference between the two are the implementation of logic where platforms can work together and generate strategies to attempt to overcome the red team. Unlike the HVAA scenarios, these scenarios are not inherently asymmetric as each platform on either side is similarly capable.

In the case of these two scenarios, the goals for both sides are similar.

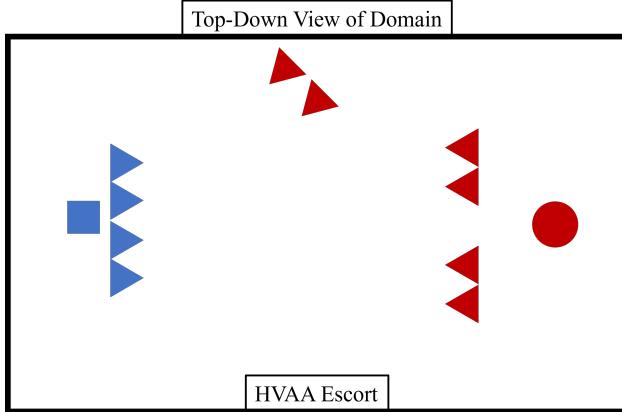


Figure 14: Disposition of HVAA Escort Scenario

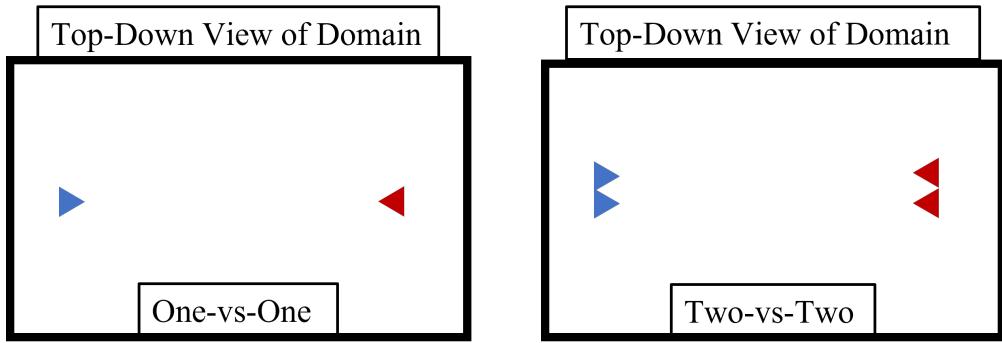


Figure 15: One vs One and Two vs Two Dispositions

- Blue: Achieve air dominance
- Red: Achieve air dominance

Comparing across all the scenarios, the HVAA scenarios incur increased run time from their representation of more platforms. There is also more variety in the types of platforms modelled and domains crossed. As a means of demonstrating a methodology, these scenarios contain so much complexity that the reader would require an increased level of knowledge into these types of mission than is capable of being explained in this format and would take away from the purpose of this writing. Thus, the HVAA scenarios are disregarded. This leaves the air-to-air head on scenarios. The one-vs-one mission, though useful, doesn't make use of inter-platform tactics that AFSIM is capable of modelling. Without degrading the results and taking away from the purpose of running the simulations in the first place,

the two-vs-two scenarios are the perfect balance of all criterion. Inside this scenario, the platforms in question make use of a general radar locking missile. Each platform comes equipped with four of these.

Now that a scenario has been selected, some stochastic element changes were made to help demonstrate the power of the methodology. AFSIM, as previously explained, has the ability to instantiate platforms from a general model of said platform. In the case of the two-vs-two scenario, both blue and red teams use the same base models. Even though AFSIM implements levels of randomness across interactions and models, within instantiations, platforms can be randomized within certain bounds. As distributed, the scenario came with some hard-coded values for variables that can take on a ranges of values. Some examples of these are below:

- Risk Aversion: Desire to press an attack given being attacked themselves
- Engagement Distance: Distance that should an enemy exist within, an engagement will be made
- Confidence in Hit: Desired level of confidence in a launched weapon to successfully hit target

AFSIM provides the dynamic means for which the variables above and many others can be randomly set based on different distributions relevant to the scenario and variable at hand. When AFSIM generates the units within each instance of the scenario, each call to the model file will sample the distributions and set the value accordingly. These stochastic elements can be abstracted as a representation of the unique human elements of each platform and how they make decisions.

Now, with an adequate scenario selected, the states to be recorded can be determined. Following the methodology, this invokes the kill chain. Since the sides are equally matched, only one set of states need to be determined for one platform that can be replicated across all platforms. Starting with the first 'F', Find. In this scenario, it is already assumed that

both teams are 'finding' each other. Thus, there isn't any 'not finding' actions taking place. Since there will be no transitions, the find step does not need to be modelled and assumed to be inherent. The second 'F', Fix. Remember, to have something be fixed means that it exists on radar but is not yet identified friend or foe. In this scenario, the fix state can be modelled for a unit by looking into the radar scopes of the opposing platforms and having the white cell determine if the contact on the radar is in fact the unit in question. If this is the case, the state can be changed from off to on. A similar methodology is applied to the first 'T', Track. The difference being the determination by the white cell to switch the value if the contact has been identified.

Upon implementation of these states, and subsequent recordings, it was found that in the example scenario, these states can be combined. The finding was made that near instantaneous identification of the radar contacts were made by corresponding platforms. With this information in mind, a combined state was made that focused on the fix portion of the kill chain. This state would turn on if a unit was determined by the white cell to exist on the radar scope of an opposing platform regardless of identification. This combining can be seen as a simplification step. Should testing have revealed larger and more variant gaps in the identification step time frame, it would have been pertinent to separate the two states.

The second 'T', Target is next. This is the step where depending on the scenario, the ISR information would be fed to the required C2 element. In this scenario, it is assumed that all airborne non-friendly contacts are to be deemed hostile and engaged. This also occurs near instantaneously at the identification of a radar contact as a foe. Thus, a state that directly follows from the definition of the target state would be nearly useless. With this in mind, a different state is proposed that encapsulates the essences of the step. This is a state defined by the engagement distance of the unit. This state is turned on by the existence of an opposing platform within the platforms engagement distance determined previously. The adjudication of the condition to be true or false is put on to the white cell. By doing this, two pieces of information are encoded rather than one. Should this value be set by the platform

itself, the state will only change based on the perception of the opposing platform on its radar scope. This can be useful except in scenarios where an enemy is able to get behind or at least outside of the radars scope but within engagement distance. Thus, assumptions have to be made by the perceiving unit as to whether or not the platform is still within engagement distance. By having this value set by a white cell that has access to the ground truth, this information can be set regardless of perception by a unit. Thus, the state encodes two pieces of information. For one, by taking the state in a vacuum, the on state indicates whether an engagement should happen. Used in conjunction with the detection states of opposing platforms, it can be understood whether the scenario is in a meta-state where the detection of the opposing force will immediately initiate an engagement at a distance smaller than that of a 'normal' engagement.

Engage is the next step. Engaging a target is a vague term that allows for interesting interpretations on the ways to model it. In this, taking into account the scenario at hand, engaging a target would mean to shoot at it. A state can be easily modelled off of being a shooter. It is also useful to know who is being engaged. Rather than creating a state that's value is set to the target being engaged, as this would break later analysis, a state can be defined where if the platform the state is attached to is the target of someone else, it turns on. This allows for the engage step to be described by two states per platform. The shooter state being on when the platform is engaging an enemy. The target state is on when the platform in question is the target of a shooter.

Finally, the assess stage. Like has been stated, the assess stage is to determine the effects of the actions taken. In this scenario, the assess stage would be the determination if a platform has died or not. Thus, a state can be determined for each platform describing whether or not it is alive in the scenario at hand. Inherent in the definitions of these states is a hierarchy. This sprouts from the usage of the ground truth information to set the values of these states. It flows that, for the other states to be on, it requires the the alive state to be on. Therefore, if a platform has been destroyed, the values of the subsequent states can

be set to zero. Given this hierarchy, the argument can be made to remove the the alive state and describe the alive status of a platform based on the inference of the current state based on the route taken. This is to say, if a platform's states are all zero, without knowledge of the what the previous state was, it is impossible to determine whether it means that platform is alive and nothing is happening or if it has died. Due to this, as a means of being able to describe a node fully without needing to know previous steps, the alive state is included in the recording.

Table 10: Kill chain states used in scenario

Kill Chain	State	Definition
Find	N/A	N/A
Fix	isDetected	1: Platform exists in radar scope of enemies
Track		0: Platform is not on any enemy radars
Target	inRange	1: Enemy platform is within platforms engagement distance
		0: No enemies inside platform engagement distance
Engage	isShooter	1: Platform is attempting to shoot a target
		0: Platform is not attempting to shoot anything
	isTarget	1: Enemy platform is attempting to shoot at it
		0: No enemies are attempting to shoot at it
Assess	isAlive	1: Platform exists in simulation 0: Platform does not exist in the simulation

With the definitions of the states to use, it is important to define the sampling frequency of those variables in the simulation. AFSIM provides a unique interface that could be leveraged for this task. AFSIM has what it calls observers. These are conditions checked in-situ with the simulation to trigger certain actions. Unfortunately, given time constraints, these were not successfully implemented though progress is promising. This provides an interesting problem that is all too common in simulations such as this. Since the change of a state can not directly trigger the recording of that change, a sampling of the current status of each platform must be made and checked against the previously recorded values. Unlike signals analysis where the frequency of the desired signals to be analyzed can determine the minimum sampling frequency through the Nyquist equation, there is no directly attainable value that can be determined. An analysis of different sampling frequencies must be made.

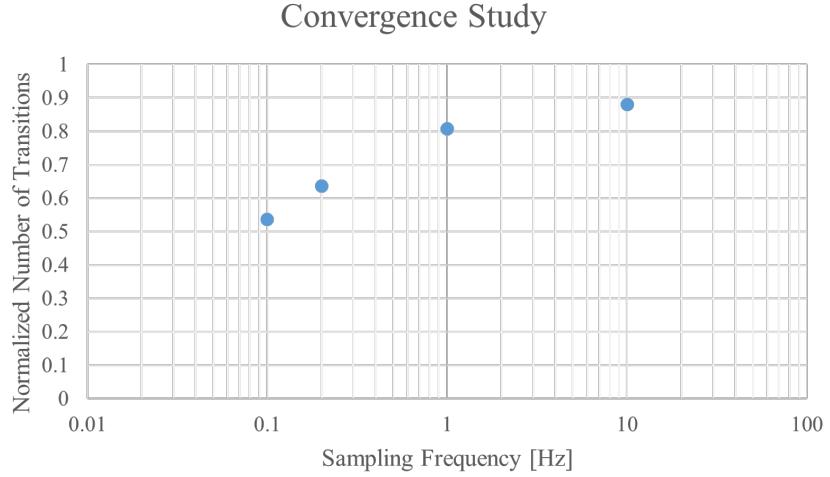


Figure 16: Normalized number of transitions recorded depending on sampling frequency

Analysis showed that a 10 Hz sampling frequency delivered the best results out of the tested frequencies. However, this came at a dramatically high cost in terms of computation time. Evident from 16, there appears to be a somewhat logarithmic behavior with respect to the number of transitions seen when compared to sampling frequency. It was found with the 0.1 Hz frequency that the edge connections between nodes was at unacceptable levels. Not only does the number of transitions change as a function of the sampling frequency, but also the connections between nodes. The 0.1 Hz frequency created connections between nodes that represented large numbers of state changes making the information nearly unusable. The desired quality of an edge is the representation of a single state value change. Whereas the 10 Hz sampling rate is the best in terms of number of observed transitions as well as the quality of the connections the computational power available makes using this sampling rate an unusable value. A nice compromise can be made in using the 1 Hz rate. The 1 Hz rate achieves an average number of transitions within 10% of the 10 Hz value and upon inspection of the resulting graphs, the quality of the edge connections is allowable. Considering the scenario in question as well as the expected separation between transitions, the 1 Hz rate should contain mostly high quality edge connections. Should the desired implementation using observers have been achieved, no compromise in number of transitions seen or edge

quality would have been made.

## 6.2 Convergence Results

The two-vs-two scenario was run using a 1 Hz sampling rate in the OSU created app for batch running and analysis of AFSIM. The stopping criterion was set at  $< 1\%$  variance in the 99% confidence interval of the average number of transitions. A secondary metric was recorded along with the stopping criterion that could prove useful in describing the scenario. This was the number of platforms killed at the end of the simulations. As a stopping criterion was not set for this metric, the subsequent information was still recorded. The simulation was stopped when the convergence value had reached 0.761%.

It was found that the average number of transitions recorded was  $26.2 \pm 0.761\%$ . Given that there cannot be partial transitions, this number can be reasonably rounded down to 26 transitions per run. This number, though useful, is not descriptive of the whole picture. There is a range of observations seen and a probability density function of the observable transitions can be created.

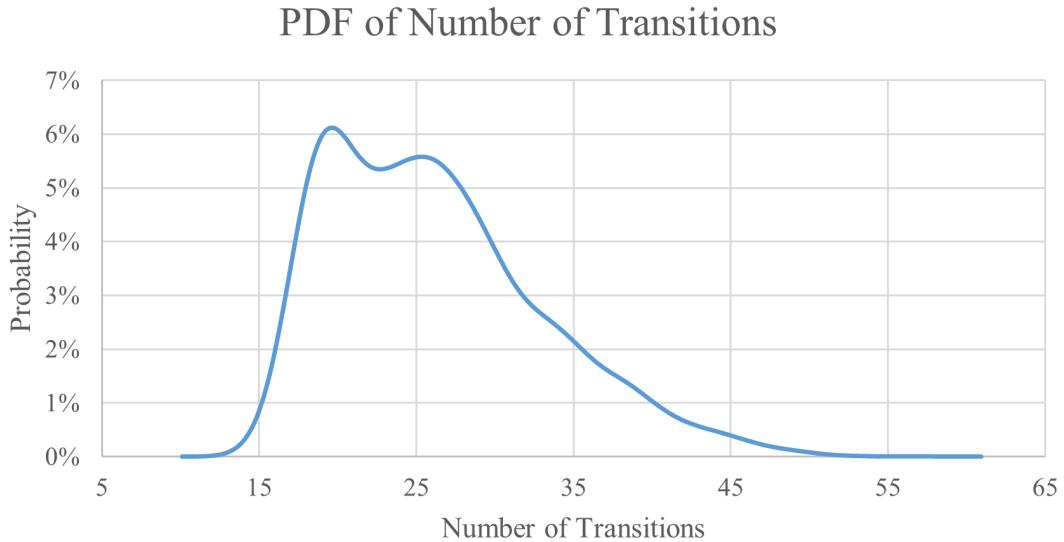


Figure 17: PDF of the number of observed transitions

It can be seen that the distribution of the number of observed transitions takes on a right skewed shape. There is a quick ramp up of chain lengths beginning around 13 transitions and peaks around 19 transitions. The probabilities of chain lengths are fairly equal between 20 and 27 before beginning to drop until reaching nearly 0% around 50. It should be noted that this recognizes transitions between nodes that are made up of every platform, to include the missiles and their states. The chain lengths would indicate that there is a nearly five times difference between the shortest chains and the longest ones. This demonstrates the highly variant nature of the simulation and the bias towards shorter paths but longer paths can and do exist.

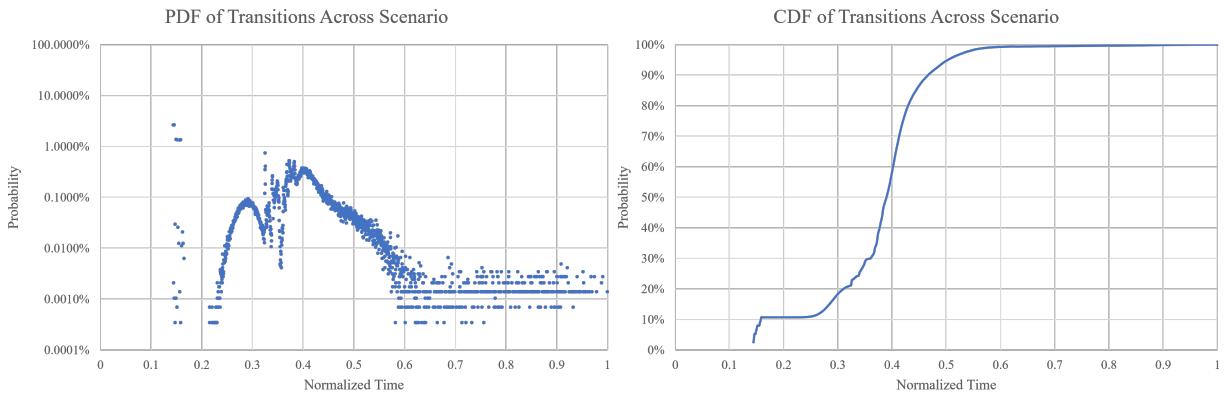


Figure 18: PDF and CDF of the times of transitions to occur across the scenario with normalized time

It can also be relevant to understand when these transitions occurred in the course of the simulation. The above figure uses bins of uniform size across the time domain to assess the number of transitions at a granular level. The probability transitions occurring at that point are shown above. It can be seen from the PDF that there are a large number of transitions at the very beginning. Continuing through to the  $\approx 0.2$  and  $\approx 0.35$  time point, there is a range of times of increasing chances of occurrence. This number increases through the until reaching a peak around 0.4. From here, there is a pretty consistent trail off of chances until 0.55 where the probabilities begin to break up. This would indicate that transitions occurring through these described phases are pretty consistent but the other areas are much

more variant in terms of the number of transitions that will occur. This is most evident past 0.6. Beyond this point, the number of transitions at points are highly variant. Observing the CDF, the plateau regions indicate these areas where the probabilities of transitions are generally unlikely. The CDF indicates that by the time the simulation is 0.6 done, the transitions that will occur have occurred while nothing happens before the 0.15 time.

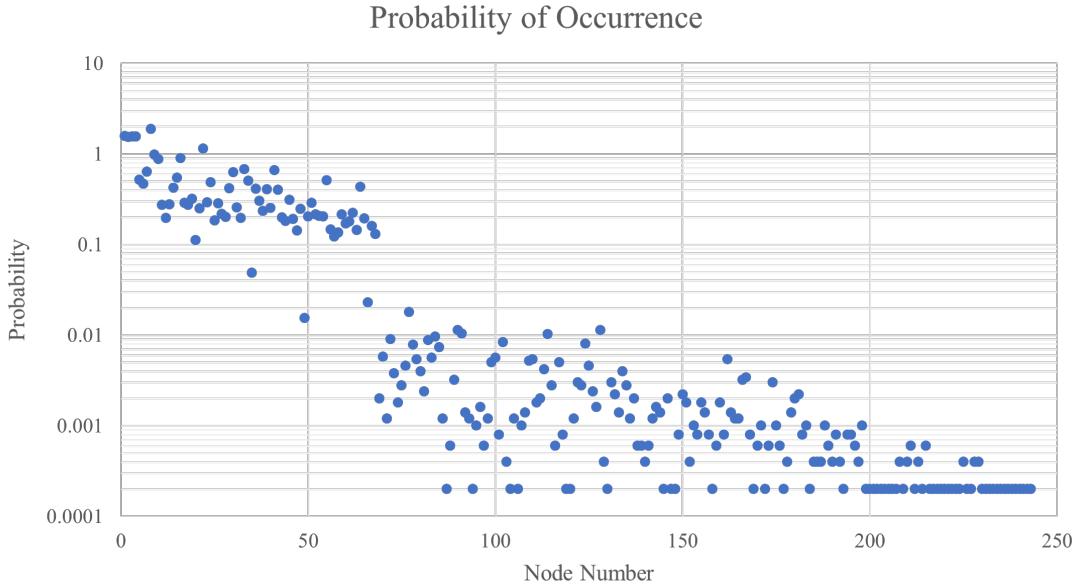


Figure 19: Probability of node occurring on a chain at random

The number of times a node is observed across all the simulations is also relevant. A probability of randomly being selected can be calculated from this. By doing this, the replication of a chain of nodes can be created by sampling these probabilities. The representation of this for the total platform space is provided below. In combination with the number of nodes in a chain, replications of similar length chains can be created. There is a problem. This originates in that the entire nodal space is being sampled. Since the entire space is being sampled, it is probable that the nodes that make up the space do not tell a coherent story of the simulation. Each node has a set of mostly unique connections to different nodes that can actually come after the node in question. This is the power of the Markov chain. Without the meta-model, a recreation of events in a coherent manner is impossible to occur through

normal statistical sampling. Embedded in the Markov chain through the edge connections are the frequencies of occurrence for each node.

To provide further understandings of the results of the scenarios, the platform killed metrics can be observed. The average number of blue platforms killed is  $1.47 \pm 1.38\%$  and the number of red platforms killed is  $1.40 \pm 1.5\%$ . On average,  $8.04 \pm 0.98\%$  missiles were launched.

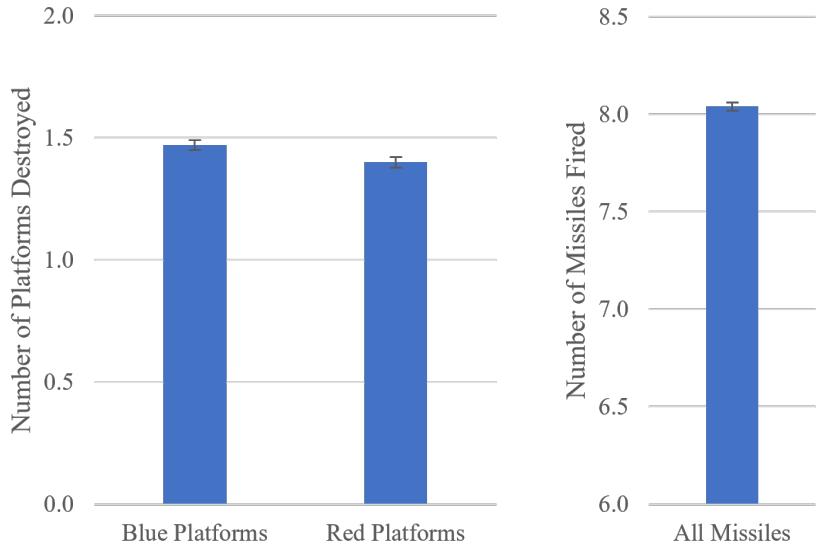


Figure 20: Bar charts of the number of platforms killed aggregated by platform type

These values would indicate that the blue and red teams are evenly matched as the long run averages are nearly identical. There cannot however be fractional values of a platform. With this mind, it would seem that it is more likely than not that the average simulation will end in a draw with both sides having one platform each. The number of missiles fired is nearly double the number of potential targets in the scenario. This would indicate that the missiles are sometimes missing their respective targets. The reason for those misses cannot be determined exactly from the data here as to what is the cause of these misses. Further analysis can however be done to determine what units are typically lost in the scenarios.

Knowing that at least one unit from either side is destroyed in each run. It can be useful to know which one that would be. Figure 21 indicates that Blue 1 and Red 1 happen to

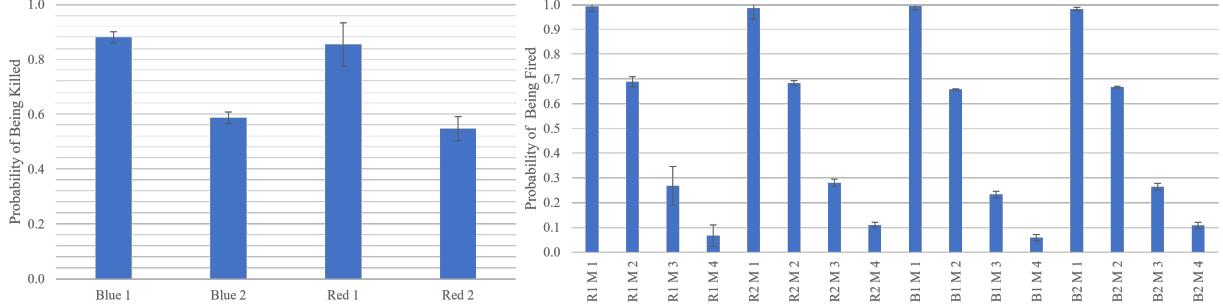


Figure 21: Bar charts representing the number of platforms killed aggregated by platform name

be destroyed more often than their respective wingmen at a nearly 90% rate. Interestingly though, the probabilities for the wingmen are nearly 50% each. This would indicate, in combination with Figure 20, that the extra four-tenths of a unit is the result of either a blue or red unit being destroyed and causing the final value of the scenario to be lopsided towards one team over the other. When observing the number of missiles fired for each unit, even though four missiles were equipped to each platform, only two were ever used consistently. As an analyst, this is useful information in determining future steps as well as for interpreting the later graphs. For an example, if a new technology was to be tested, it would need to be placed in the first position of the firing order to ensure it was modelled appropriately. In the case of interpreting the graphs, knowing that there are extraneous missiles in flight that won't connect helps quantify the scenario in a general format. As was mentioned with the previous input/output researchers, if it was needed to understand the times for when these missiles were fired, simple graphs could be generated to display that.

### 6.3 Graphs of Interest

Up to this point, the information that has been parsed is all general information that can be stated about the scenario as a whole. That type of analysis, most importantly the platform killed information, is what would be typically outputted and used by analytical wargamers and developers trying new technologies. Should a new tech or new missile be introduced like

Connors, comparisons between a baseline scenario such as this and one with the new missile would make great use of those metrics to quantify the general changes that result in that new technology.

That information is useful in this research as guide of what general trends should be seen. The crux of this research comes in the analysis of the meta-Markov chains. To do this, a model was created that made use of the red and blue platforms. The missiles were neglected in this model as the information they provide is simply that they exist. Since the focus is on describing the transitions of the platforms, the model does not need to include the missiles. They can however be implemented if it is necessary to understand what missiles are fired, where in the chains, and if they miss or not. That is unnecessary here. Thus each node takes on a set of values as represented by the Table 11 below of the initial node, Node 1.

Table 11: Description of the states for each platform in Node 1

<b>Node 1</b>	<i>isAlive</i>	<i>isDetected</i>	<i>isInRange</i>	<i>isShooter</i>	<i>isTarget</i>
<i>Blue 1</i>	1	0	0	0	0
<i>Blue 2</i>	1	0	0	0	0
<i>Red 1</i>	1	0	0	0	0
<i>Red 2</i>	1	0	0	0	0

This node is the start of every single Markov chain. This node describes the state of the scenario where every platform is alive but haven't been detected. No one is within engagement distance of the other. No one is attempting to shoot or is being shot at. Every node follows the format as in the above table. As it was described for each state being recorded, a value being on takes on the value of 1 and if it is off it is a 0. Each platform can also be assigned a number that allows for easier storing of the nodes set of states in a vector. By placing these values in a vector and then concatenating vectors one on top of the other, it allows for easy scoring of nodes for analyst score. As each set of states is discovered while aggregating the runs together, the vector is formed and if it is unique it is assigned a unique number. This number is the node number and is what is displayed in the state transition matrix and is what is used to retrieve the vector describing the scenario at that node. The

state transition matrix (STM) for this aggregation is displayed in Figure 22.

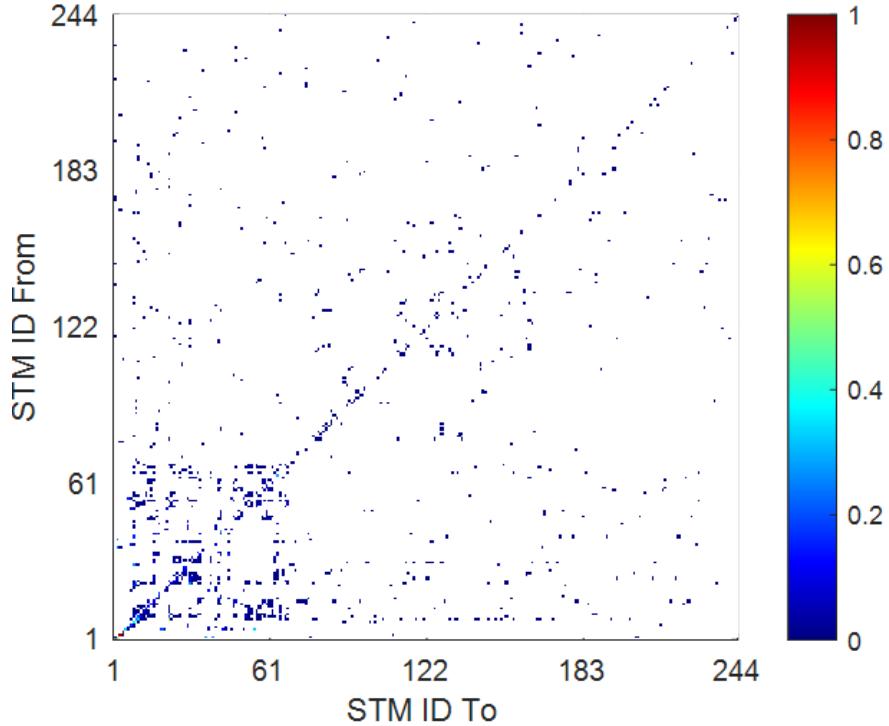


Figure 22: Visual representation of the state transition matrix

In this format, to read the STM, you would read the left axis first. This is the 'From' axis. You would select a node you want to move from. After selecting the node, i.e. row, move along the row until reaching the column of the desired node. The bottom axis is the 'To' axis. If the element in that position is non-zero, then the edge between the two exists and the value is the probability of that edge being traversed from that node. In the Figure 22, only non-zero elements are colored. Notably, most of the edges have low probabilities as there are large number of edges for most nodes. An artifact of the aggregation process that exists within the STM is the left to right diagonal. This is due to the process numerically ordering newly discovered nodes in chronological order. The structure in the lower left-hand of the image is not an artifact. This is the result of those nodes existing within the same class. Since they are discovered so closely to one another, it is easy to see there class connections in the STM. However, visual inspection of the STM should not be used to determine the classes inside an STM except for on very small matrices. The STM of a Markov chain

is not conducive to understanding the information encoded within. This is where other visualization methods come into play.

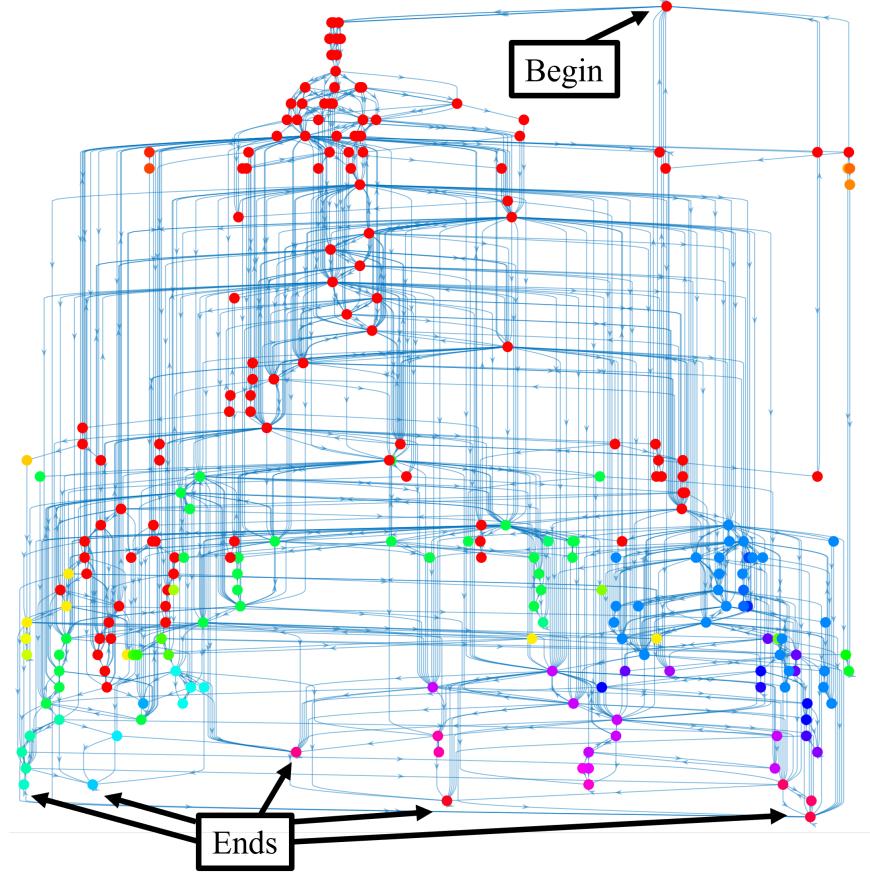


Figure 23: Visual representation of the Markov chain utilizing the 'layered' layout setting

This format already begins to tell a story. This format makes use of the MATLAB digraph objects. These objects have the ability to be manipulated through various prebuilt algorithms. The most useful layout for this research is the 'layered' layout. This is built off of the work from Gansner et al., Barth et al., and Brandes et al.[32][33][34] In the succeeding graphs, unless otherwise noted, node 1 is set to the highest most position. All subsequent nodes are placed on their respective Y-level based on the earliest discovery of the node with the constraint that all predecessor nodes exist higher than the node itself. In this form, due to the hierarchy of states, it is easy to see a natural progression through time of the Markov chain from top-to-bottom.

As is described in Figure 24, nodes are colored based on their respective classes. Unlike some Markov chains, it is easy to determine the number of classes that should exist within a graph of this form. Since the alive state can only move from being on to off, this means that classes are dependent on whether or not a unit is alive or not. The Figure 23, though nice for other means, is not the most useful figure in describing classes.

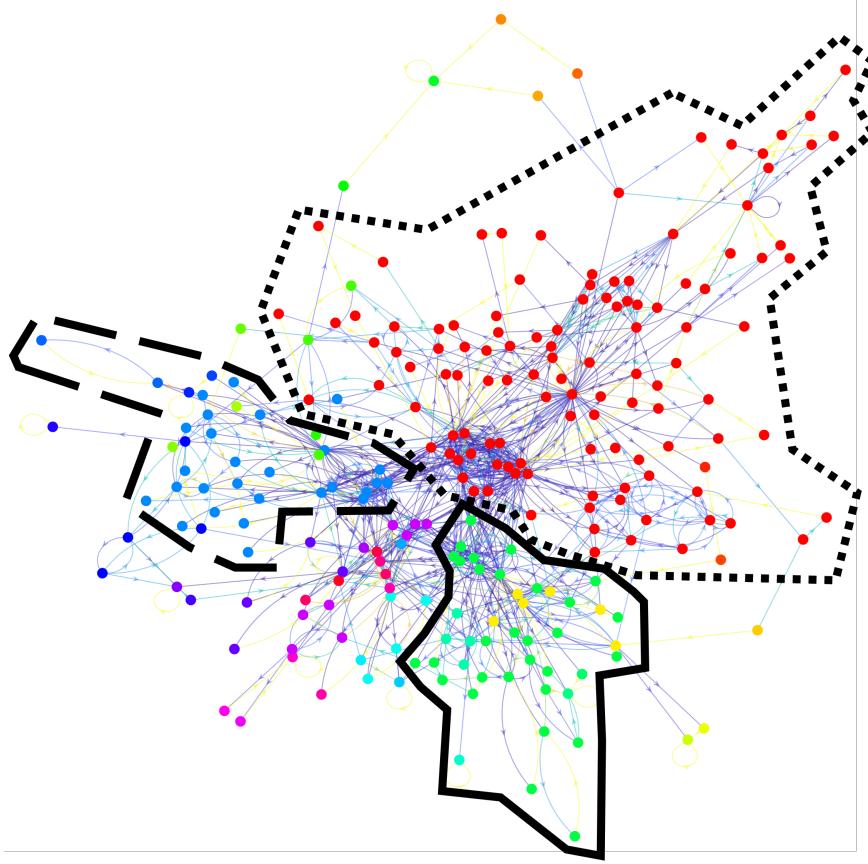


Figure 24: Visual representation of the Markov chain using the 'force' layout setting

A more useful format to understand the classes and their interactions is to make use of another MATLAB pre-built algorithm. This is the 'force' algorithm in two-dimensions. This makes usage of the works by Fruchterman et al.[35] In this algorithm, the general means of node placement is based on the weight of edge connections. Since classes are defined by their inter-class communications, and communication comes through the form of edges, classes naturally aggregate together. Three clusters are specifically highlighted in the above figure. The red class is the class where all units are alive. Since this is the state every Markov

chain starts in, it flows that this class would be the largest as progress is made through the kill chain states for particular platforms until one is destroyed. The matte blue cluster is indicative of Blue 2. The green nodes are indicative of Red 1. Given the maximum number of permutations of four platforms that can take on one of two sub-states each, there should be sixteen different classes. Due to human perception of colors, it is difficult to determine in the figure but there are in fact sixteen classes.

Since some classes are not as nearly as large as the other classes, this is indicative that there are some nodes in those classes that are never traversed. Due to the nature of the state definitions, they are not explicitly required to come one after the other. In this way, a full class would be represented by a  $K_n$ -complete graph. The lack of this  $K_n$ -complete structure is just as important as if it did exist. It can be easily reasoned that given an infinite number of runs, every single combination of states that can be generated, will be generated. At the same time as each state combination is realized, it would extend that every possible transition would also appear. Since an infinite number of runs is not possible, a finite sample of those permutations can be achieved. It would follow, since this is a probabilistic representation of the larger potential graph, that the information encoded in this graph is the more important and generally more likely information. The lack of the potential permutations contained within each individual class, as well as the absence of some transitions between realized states, should not be inferred as meaning that those permutations and transitions will not occur. In this case, it simply indicates that those states and their transitions are more unlikely to occur to the point that modelling them becomes frivolous in the larger picture.

## 6.4 Determining Viability

It is of the utmost importance that the meta-Markov model accurately reflect the simulated information from AFSIM. If the meta-Markov model is unable to recreate the information, then the model is unusable. To verify that the model creates similar information, a simple

set of random walks is performed. The simulation was setup so that Node 1, Table 11, was always the starting state. Every node that exists beyond Node 1 is based on the results of the simulation. The power of modelling the simulating in the form of a Markov chain is the ability to perform random walks.

To perform a random walk, start at Node 1. At each node, create a table that contains an ordered list of edges and their weights. Next, query a number from a uniform distribution between 0 and 1. Iterate through the table, adding each new edge weight together, until the sum is greater than the queried value. At this point, the last edge whose value did not cause the sum to go over the value, will be traversed. Move to the new node. Repeat the process. Unlike the simulations in AFSIM, this algorithm is significantly faster.

To determine if the model was sufficient, the described random walk process was implemented. The frequency of observations for each node was recorded and the length of the individual Markov chains was also recorded. The model was randomly walked the same number of times that the AFSIM simulation was run. The result, near perfect replication. Using the same number runs as AFSIM, the average length of Markov chains was exact to one decimal place with 99% confidence intervals of 0.795% for AFSIM, and 0.785% for the model. These values alone indicate that the model is capturing all relevant information from the AFSIM chains. These are not the only values that need to be used however.

Figure 25 shows the comparisons in the relative probabilities of each node. The first 100 nodes appear to have nearly identical probabilities. Beyond 100, the probabilities begin to fluctuate but the general trends are present between the two datasets. To better understand the distribution of the probabilities, a range comparison can be made. These values are plotted on a log scale so the comparison of the range needs to be of the  $\log_{10}$  values of the probabilities.

Figure 26 demonstrates again, near perfect replication in the distribution of probabilities. There are some slight variations. For example, the average for either are  $10^{-4.37}$  for AFSIM and  $10^{-4.19}$  for the modelled data. In general, the interquartile ranges of both are nearly

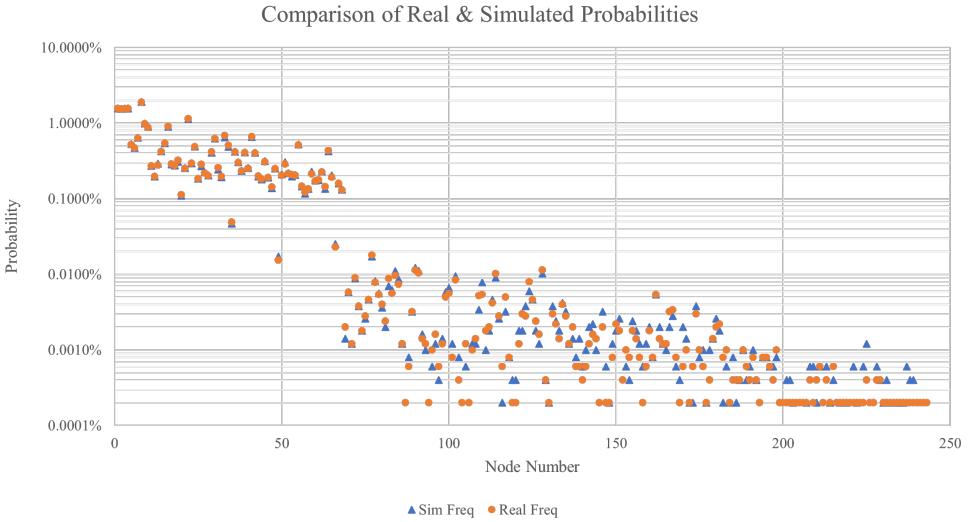


Figure 25: Comparison of the probabilities of reaching each node between AFSIM and modelled data

identical and with the simulation data being slightly biased towards more probable values. Interestingly, there are 24 different nodes that are present in the AFSIM chains that were never observed in the simulation data. That is to be expected in a model like this and the way it was simulated. The model is stochastic. The likelihood of perfect replication of all states across both data sets would be a highly unlikely event. Thus, the necessity to compare distributions and not individual values for verification.

Figure 27 demonstrates the most striking differences between the two datasets. Though the average chain lengths are equal, the overall distribution of chain lengths are slightly different. The simulated chain lengths have a more normal distribution with a peak around 15 transitions in length. The distribution of real data has a slightly higher peak around 13 transitions. Even though their peaks are different, the tails of each distribution nearly precisely follow each other. The simulated distribution does not precisely model the distribution between 13 and 20 but, beyond these points, the simulation data is very accurate.

Given the totality of information, it can be reasonably assumed that the meta-Markov model, can be used to accurately reproduce the AFSIM information. There is one notable issue that must be discussed. This is that, unlike AFSIM, the model, in its current form,

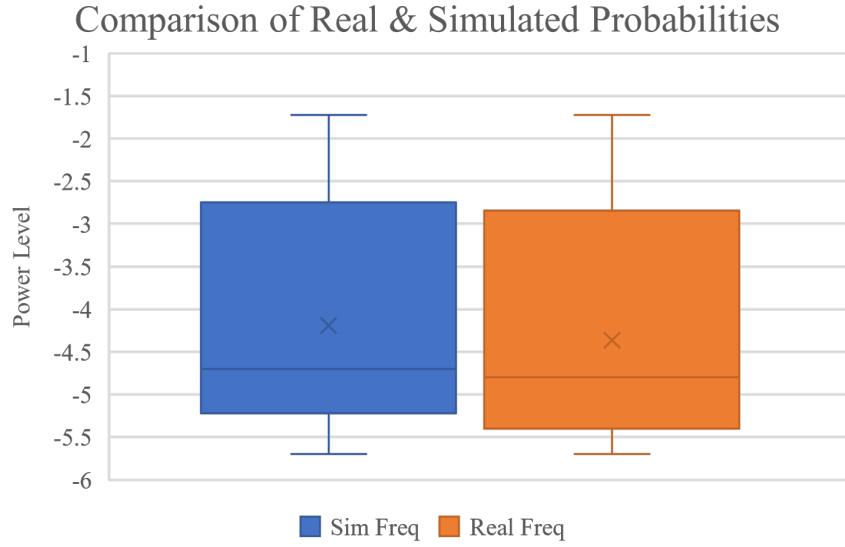


Figure 26: Comparison between the distributions of the probabilities of observing each node in the AFSIM and modelled data

is incapable of modelling events that did not occur in the AFSIM data first. Should it be necessary to find a node consisting of a certain set of substate values, there is a non-zero probability that this node does not exist in the model. To remedy this, as has been previously discussed, more runs of ASFIM will draw out more nodes. This does come at the cost of increased computational time. Though, once a model has been generated, the results can be replicated at magnitudes faster rate. Showing that the model does replicate AFSIM information means that the key attribute of OS can be used. This is the ability to start at any node and accurately recreate chains.

## 6.5 Most Probable Paths

Moving back into layered formats of the Markov chain. In the definition of tree rank, the most probable path tree was referenced. Even though an example was given, Figure 28 is the practical example of the most probable path tree (MPPT) for Node 18. Node 18 is the representation of total loss by both teams. No red or blue team members survive. In this view, Node 18 is placed at the bottom and the predecessor nodes are placed in layers

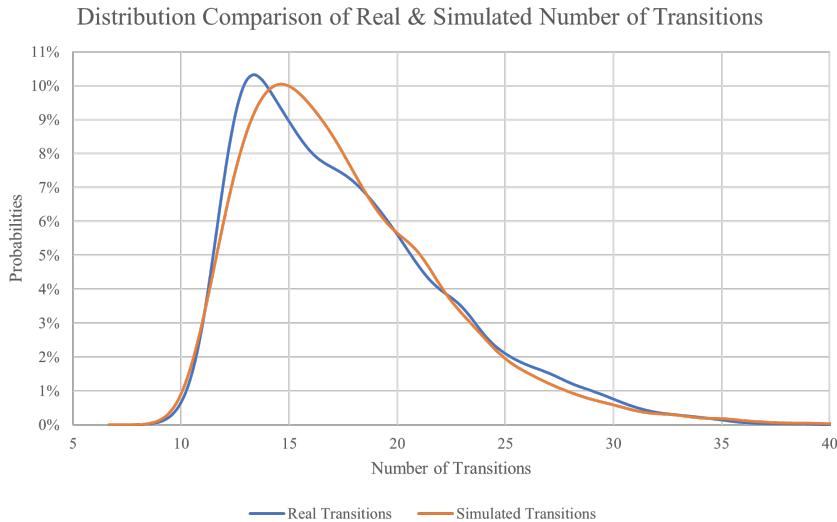


Figure 27: Comparison PDF between AFSIM and modelled chain lengths

depending on the number of steps away from the node they are. Their respective edges are colored by the probabilities of the step. Node 1 is annotated as it is the universal start. Since the nodes are colored based on their classes, and there are four units to be destroyed to reach Node 18, one can trace the path from 1 to 18 and record the number of color changes. In this tree, it can be seen that the colors go from red at first, to light green, to purple, and then to a different shade of red. Each color representing a different platform destroyed.

The number nodes in each color is also representative of the scenario simulation that would generate such a result. In the red class, there are large number of nodes, indicating a large number of state changes. Most likely these are detections, coming into range, and platforms beginning to target each other. There is then the transition to light green which only has one of that color, this indicates that a missile was already in the air targeting that unit in particular and was destroyed fairly quickly after the first unit was destroyed. Then the transition to purple. Containing two nodes. Again, it follows that this is likely caused by missiles already in the air that hit their target shortly after the previous unit was destroyed. Leaving no time for other states to change predominantly through maneuvering of the platforms. Finally, the transition to the other shade of red. This shade consists of the final alive platform being killed by an inflight missile.

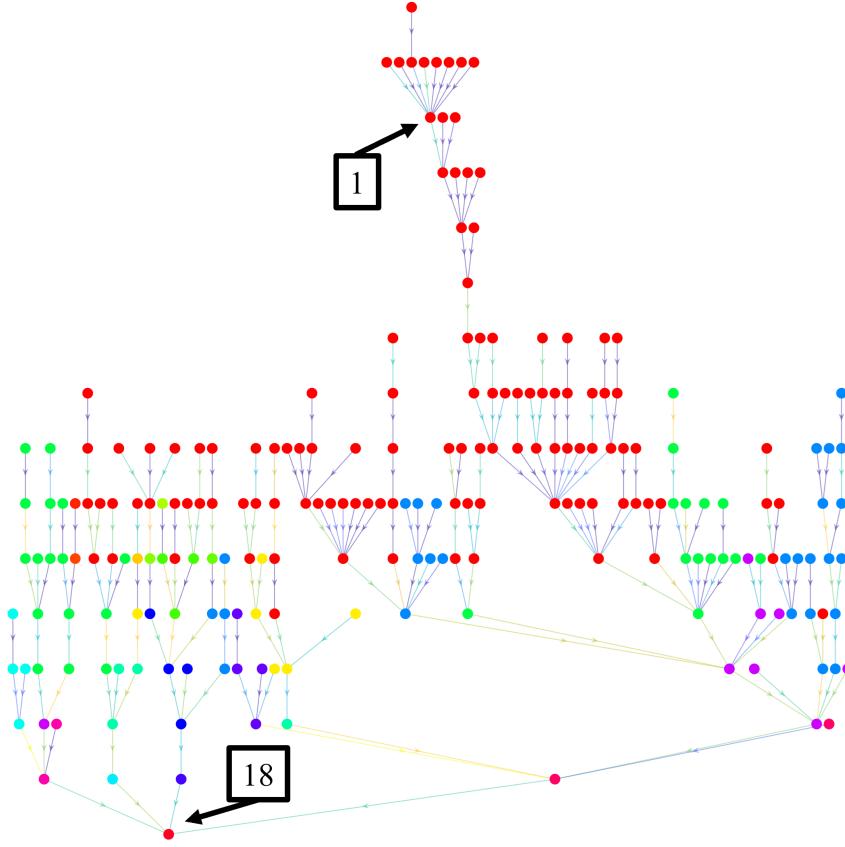


Figure 28: Most probable path tree of all nodes that can reach Node 18

## 6.6 Application of Metrics

Now that the aggregated meta-Markov chain has been created, the application of the designed metrics can be made. There are two metrics that require user input to calculate. Those being AS and OS. For analyst scores, the goal was to score individual platforms based on there ability to engage another target. The scores utilized are in Table 12.

Table 12: Analyst score value table

<b>State</b>	<i>isAlive</i>	<i>isDetected</i>	<i>isInRange</i>	<i>isShooter</i>	<i>isTarget</i>
<i>Platform</i>	10	-3	5	6	-7

Since a unit must be alive in order for them to affect change in the system, a score of 10 points was given to the alive state. If a unit was also a shooter, a score of 6 was added on. Should the platform also be shooting at a platform within it's engagement distance,

that was an additional 5 points. If the platform itself was detected at all, that is a negative attribute to have in this process, and therefore the value is dropped by 3 points. Using the same idea, when a unit was also a target of another unit, this is potentially devastating state and drops the score by 7 points. Using this scoring metric, high scoring nodes will be those where they are able to shoot a target within their engagement distance without having been spotted or being targeted themselves. Low scoring nodes will be those that do not have a unit within their engagement distance, are not shooting anyone, and are detected themselves while being shot at.

The outcome score metric also requires a scoring of final nodes based on analyst determined criterion. In this scenario, the scoring is based off of Table 13. In this table, a final node is given 3 points for each alive blue platform. On the inverse, for every red platform alive, 3 points are removed. The score of a node is averaged over the number of alive units in that node. Using this scoring system, the maximum score a node can achieve can be  $\pm 3$  points. At 3 points, this would be considered an absolute win by the blue team as all blue team platforms remain while no red team survived. A score of  $-3$  points represents an absolute loss on the part of the blue team as all red team units survived and no blue team units did. A neutral score of 0 is achieved when there is a balanced combination of red and blue units.

Table 13: Outcome score values

<b>State</b>	<i>isAlive</i>	<i>isDetected</i>	<i>isInRange</i>	<i>isShooter</i>	<i>isTarget</i>
<i>Blue Platform</i>	3	0	0	0	0
<i>Red Platform</i>	-3	0	0	0	0

With the user-input required scoring systems implemented, the metrics for every node can be determined. The total score value of a node was to be made up of a uniform average of the scores. Due to the varying ranges of each metric, a standardized format was necessary. In this case, a z-score was calculated for each metric and the score of the node was the corresponding z-score value. By utilizing the z-score metric, outliers can be easily identified that represent unusual characterizations in the graph while maintaining the correlations

between metrics. For the total value of a node, the absolute value of the z-score for each metric was used. With this in mind, highest scores indicate particular nodes of interest. A correlation table, Table 14, is shown that demonstrates the metrics against various values.

Table 14: Correlations of states, metrics, and total scores for the aggregate Markov chain

	<i>Node</i>	<i>RP</i>	<i>AS</i>	<i>TR</i>	<i>IS</i>	<i>OS</i>	<i>OIS</i>	<i>In Degree</i>	<i>Out Degree</i>	<i>Total Score</i>
<i>Node</i>	1									
<i>RP</i>	0.0286	1								
<i>AS</i>	0.3019	0.1769	1							
<i>TR</i>	0.0273	-0.12	-0.189	1						
<i>IS</i>	0.035	-0.212	0.1607	-0.092	1					
<i>OS</i>	-0.026	0.0046	-0.026	-0.045	-0.007	1				
<i>OIS</i>	-0.267	0.1912	0.1087	-0.243	0.1699	0.0457	1			
<i>In Degree</i>	-0.646	0.0819	0.0212	-0.199	-0.118	-0.005	0.418	1		
<i>Out Degree</i>	-0.562	0.0849	0.0507	-0.026	-0.135	-0.016	0.409	0.827154	1	
<i>Total Normed Score</i>	-0.152	0.3501	0.1	0.2232	0.3866	0.032	0.274	0.362441	0.26688105	1
<i>Total Raw Score</i>	-0.28	0.4759	0.2259	0.086	0.0675	0.5828	0.556	0.465443	0.51318199	1

It is desired for the metrics utilized to pull out unique nodes and be uncorrelated with each other so as to not incur overlap and unnecessary calculations. In Table 14, the state number is included to quantify the connection of discovery order of nodes to any metrics. There is also two forms of total score. In the row labelled 'Total Normed Score', this is the calculation of the score as described using z-scores. The row labelled 'Total Raw Score' makes use of the non-normalized, non-standardized metrics. The maximum level of positive correlation amongst the metrics is the connections between in and out degree. This level of positive correlation is expected. The maximum level of negative correlation amongst the metrics again makes usage of degree and the node number. This is most likely a result of nodes that are discovered later, have a correspondingly lower number of observations in the data. The maximum number of in and out edges a node can have is set by the number of times it is observed in the data set.

The node number has a slight positive correlation to the AS and slight negative correlation to OIS. For AS, this is most likely an artifact of the numbering of the early nodes as a function of time. Node number should not be used as an indicator of time progression in the simulation. It can be used as a loose indicator of time progression when looking at the first couple of nodes. This comes from the first set of nodes being numbered by their order

of discovery which happens to be through time. In general, the metrics are uncorrelated to weakly correlated. This would indicate that the metrics are scoring nodes based on unique criterion. Thus, satisfying Objective 4.

Some correlations are of particular interest. OIS and IS are weakly correlated when there corresponding correlations to degree the are inverse of each other. OIS is positively correlated while IS is negatively correlated. This demonstrates how, though they utilize the same principles, do not correlate with each other. Also of note, the consistent negative correlations, though weak, of TR to all other metrics and near zero correlations of OS to all other metrics. Where as all metrics contain some mix of positive and negative correlations, these metrics contain correlations of single types.

Observing the total score correlations for the metrics, the normed score does a better job of preventing strong correlations across the metrics. The raw total score has two metrics that are nearly uncorrelated but the existence of the strong correlations amongst the other metrics are not useful. By using the normed scores, strong correlations are reduced. Since the normed total score is based off the absolute value of the z-score, the direction of correlation does not matter. With this in mind, using a uniform weight across all metrics indicates a fairly even set of contributions besides AS and OS. It could potentially be useful to reduce the weighting of the other metrics and boost the weights of AS and OS to ensure that the magnitudes of the correlation are even. In this scenario, this will not be performed and analysis continued.

## 6.7 Score Analyzation

The scores have been calculated, and can be analyzed. Graphs can be particularly useful when it comes to understanding the meaning behind values. The directed graph representation of the meta-Markov model helps with understanding the scenario at hand. Using pre-built algorithms in MATLAB us useful up to a point. The 'layered' algorithm provides

a means for ordering nodes in the same layer based on their relative location in time. The X-axis however, is not being used for a directly quantifiable use. This axis can therefore be used to plot nodes based on their metric scores. Most notably, the OS metric would provide a way to show the possible end state that will come from being in that state.

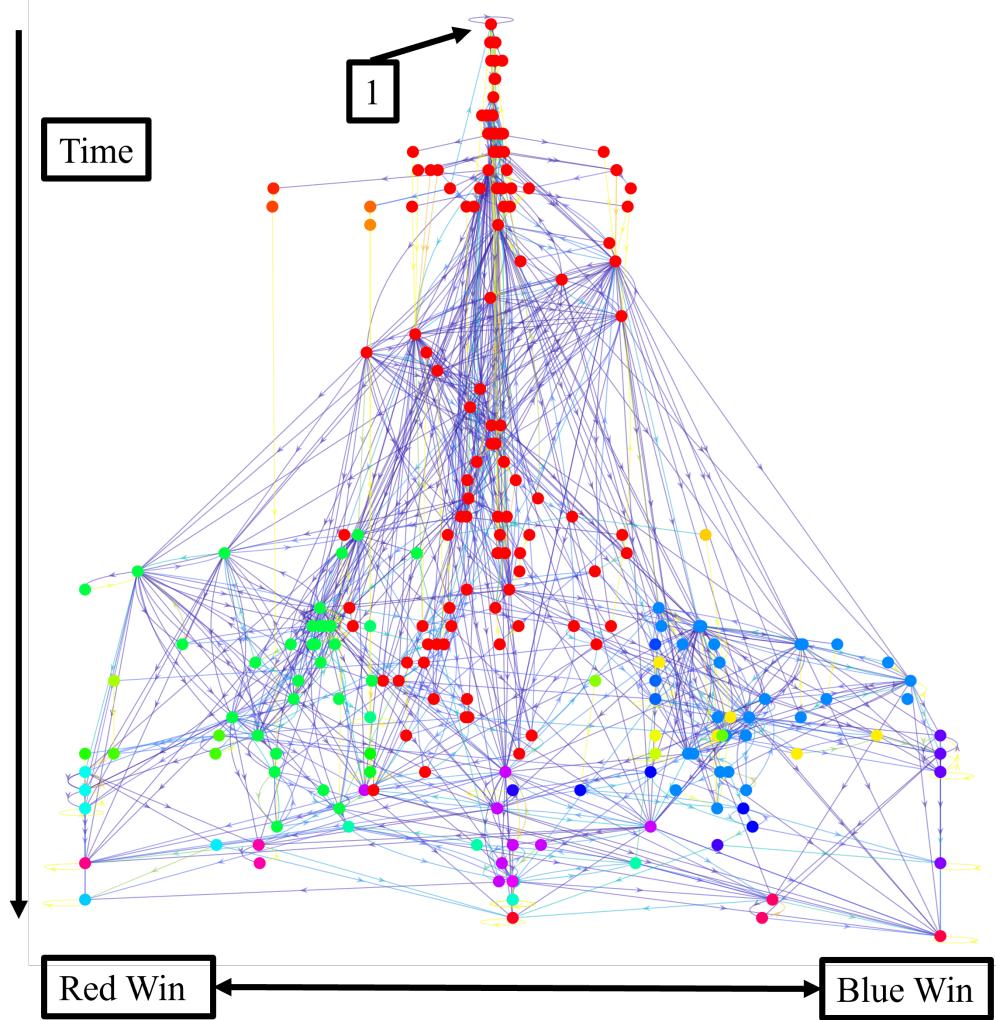


Figure 29: Directed graph representation of the meta-Markov model using the 'layered' layout for Y, and OS metric for the X location of nodes.

Figure 29 demonstrates a potential usage of OS. Unlike with Figure 23, a more direct story can be told by interpreting the locations of each node. As can be seen with Node 1, this node is at the beginning of time while also being centrally located. This would indicate that being in Node 1 does not bias an outcome towards any particular side. The same

algorithms that are used to generate a MPPT, like Figure 28, can be used to determine the most probable path between two nodes on this graph.

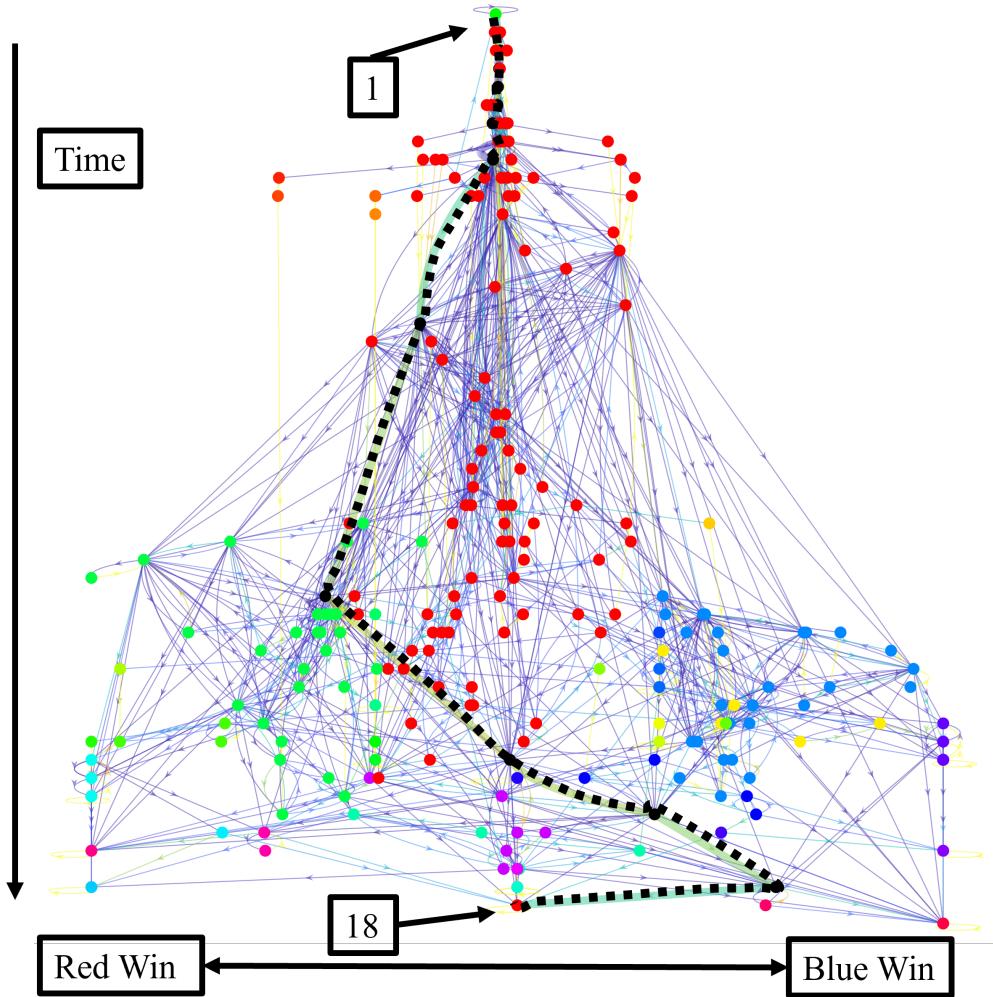


Figure 30: Highlighted most probable path between the start node, Node 1, and the total loss node, Node 18

Figure 30 represents the path between the starting node, Node 1 and Node 18. Corresponding to total loss on both sides and therefore a neutral outcome based on the scoring system used in Table 13. It can be traced out that, initially, there's a bias towards a potential red win before swinging over to the blue win side. This is indicative of the red team scoring a quick hit on a blue team member. Eventually, the remaining blue team is able to score a kill, evening the odds. The blue team member was also able to get the second red team member, however, the red team member was able to get a missile off that connected with

the blue team member after it was destroyed. Thus, resulting in a neutral outcome.

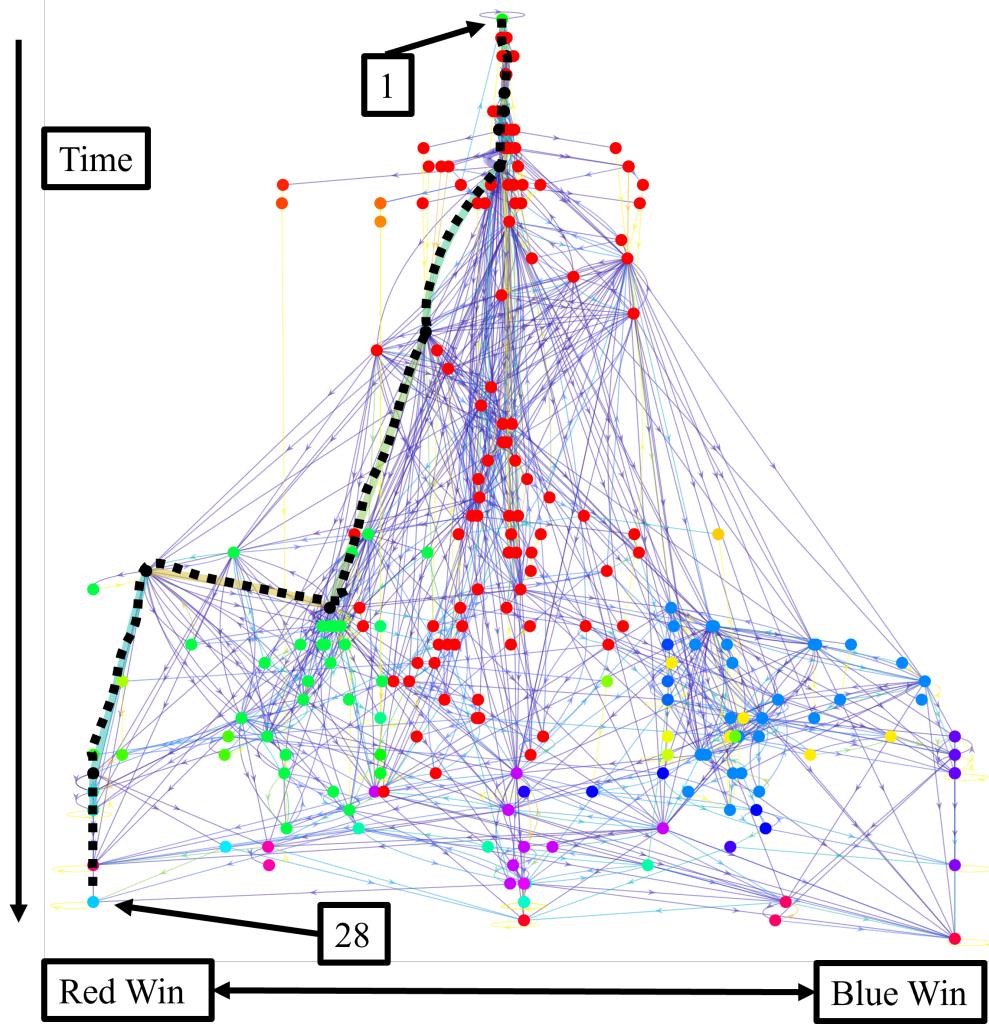


Figure 31: Highlighted path representing most likely case for a total red win

A similar analysis can be performed where the most probable path between the start node and the desired end node, in this case Node 28, can be highlighted. Figure 31 shows this most likely case. This can be easily interpreted as red scoring a quick kill at the beginning of the simulation. Followed by some maneuvering before red team is able to secure the second kill and secure a victory. Compare this to Figure 32 of a blue win. There paths are very similar. This would make sense as the other information has borne out that both sides were evenly matched. It would follow then that whatever happens that would cause an outright red win, if flipped in favor of the blue team, would cause a similar chain of events.

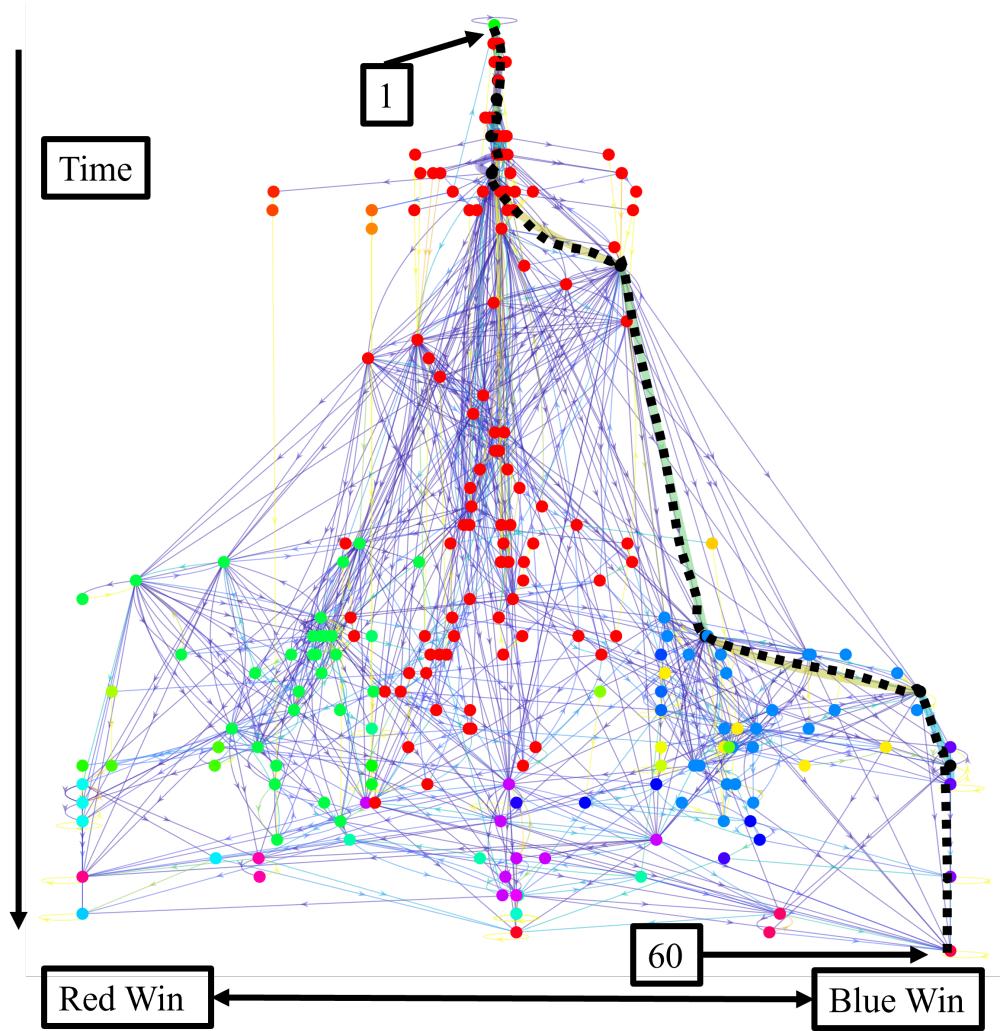


Figure 32: Highlighted path representing the most likely chain leading to a blue win

Plotting the nodes based on their 'layered' position and OS is not the only format that the graph can take. A useful format of the model is one where OS and OIS are plotted together. Figure 33 demonstrates this capability. In this format, it is easy to see that within each class, the same general OS score is applied where as there is a striation of increasing OIS. This would indicate that progression through each class makes consistent steps towards changing the OS towards a more exact result.

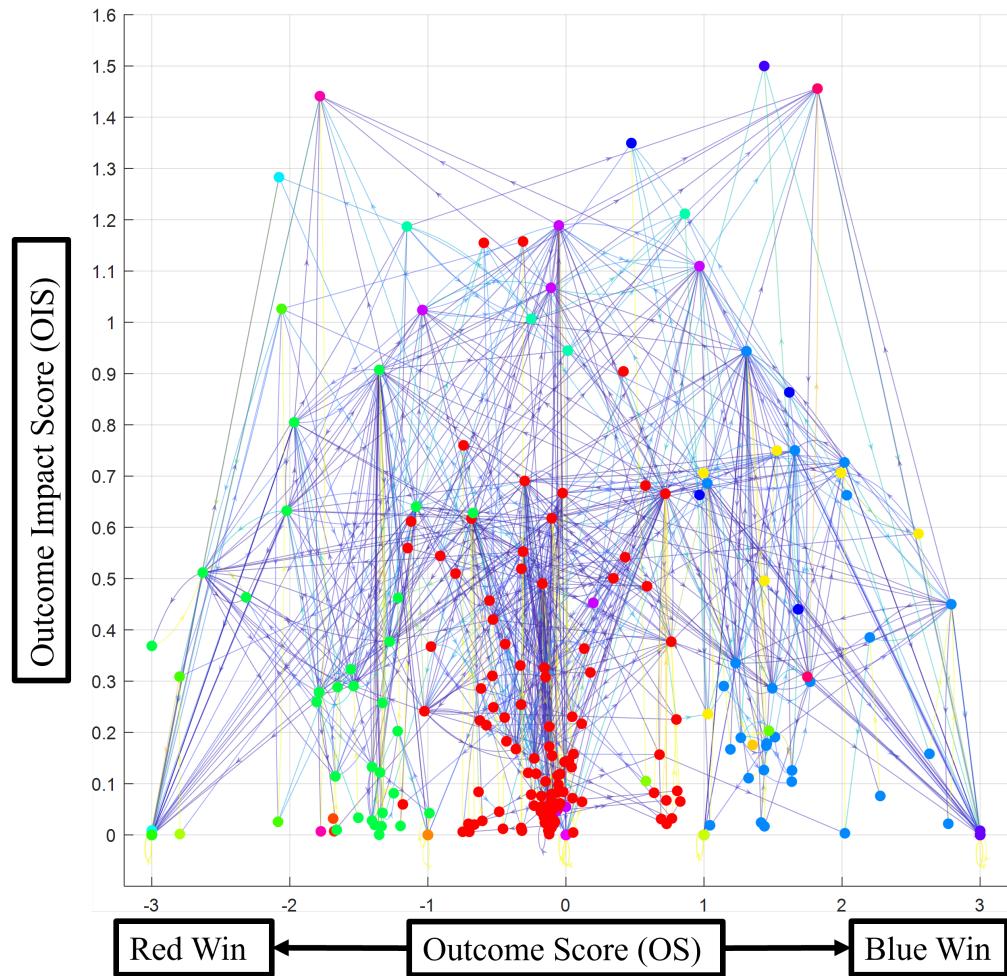


Figure 33: Directed graph of model with the OIS metric of a node being its Y position and OS metric used for the X

Similar to the way Figure 33 can be created for OIS and OS. One can be made for AS vs IS. Figure 34 demonstrates this result. It is more difficult to ascertain inter-class striations of values like in Figure 33. It can however be concluded, which nodes are the outliers that

are shown in Figure 35. It is also useful to see that the outliers on the IS metric all have edges that lead to Node 18, in the lower left-hand corner at a score of (0, 0). These edges are not the only ones however, so it cannot be inferred that a high IS metric would result in edges leading to Node 18. This is the power of these plots. By using the calculated metrics, plotting nodes based on their scores, coloring nodes based on class, and preserving the edge connections between nodes, new information that would otherwise be inaccessible can now be utilized.

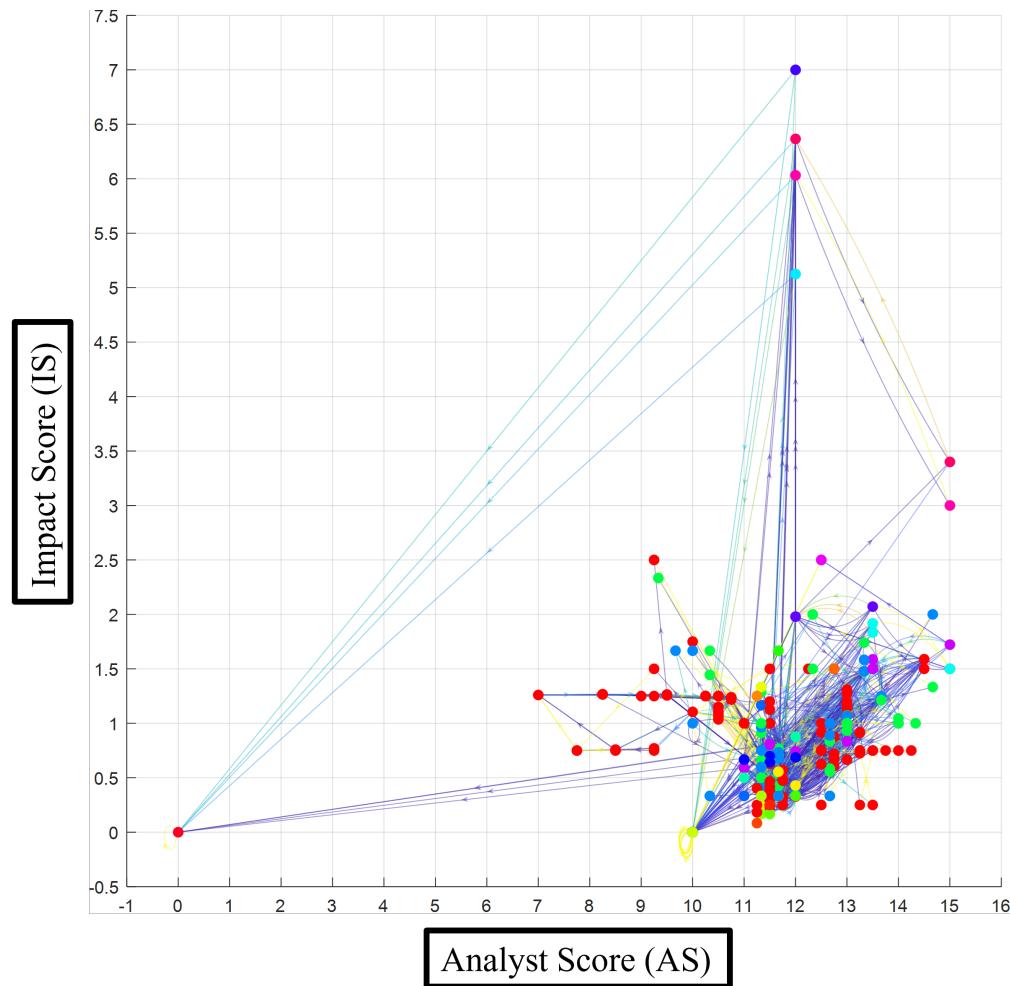


Figure 34: Directed graph of the model with the IS metric on the Y and the AS metric on the X for each node

## 6.8 Key Nodes

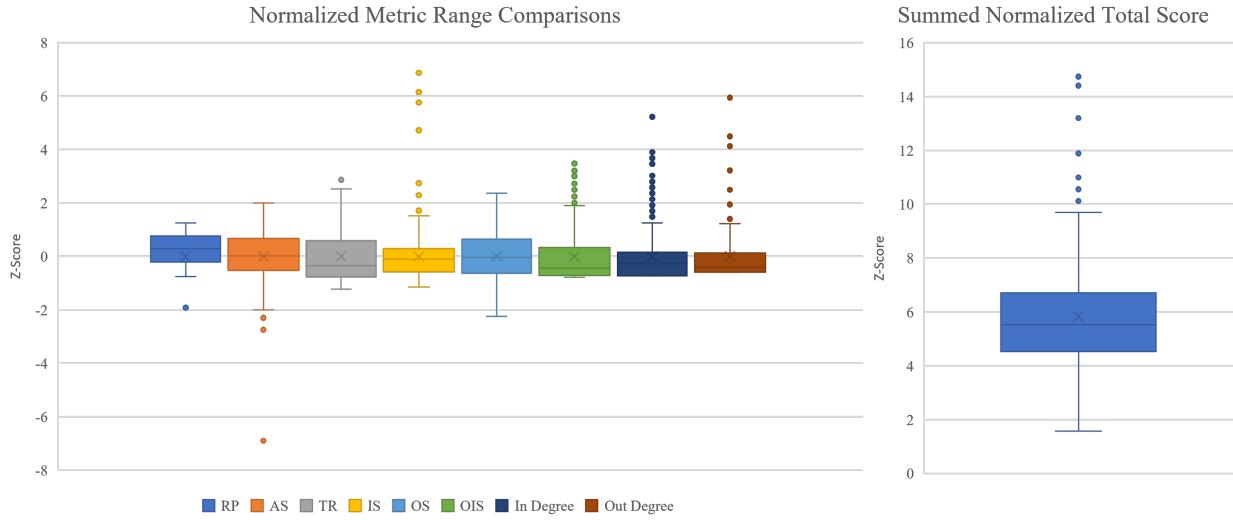


Figure 35: Normalized metric range comparisons

It is useful to understand the ranges of values that can be observed for each metric using the z-score values. Table 35 demonstrates these ranges using box-and-whisker plots with outlier points kept. Of particular interest is the symmetry and lack of outliers for OS. Both RP and AS share this aspect of symmetry but do contain notable outliers. TR, IS, OIS, in degree, and out degree all have strong asymmetries in their distributions with outliers present. TR is different from the rest in containing only one identifiable outlier while the others contain many outliers. The existence of these outliers and asymmetries track with the magnitude of the correlations in Table 14. In generating the total normed score, it is also useful to understand its distribution.

The distribution of total scores is fairly symmetrical around a value of 5.8. There are however a large number of outliers that could be of particular interest to an analyst. A table of nodes for the top five scores for each metric is provided in Table 15 to include the top five total scores. Uniquely, it can be seen that for the first four nodes of the top five nodes for total score, they each appear twice in the top five nodes of all other metrics. This is solid indication that the total score metric is pulling out the truly unique nodes based on

the different metrics. These top five nodes thus require further investigation to determine their meanings.

Table 15: Top five nodes for each metrics score

<b>Rank</b>	<i>RP</i>	<i>TR</i>	<i>AS</i>	<i>IS</i>	<i>OS</i>	<i>OIS</i>	<i>In</i>	<i>Out</i>	<i>Total</i>
<b>1</b>	184	184	155	182	49	182	8	8	182
<b>2</b>	243	146	168	17	50	17	16	9	17
<b>3</b>	242	118	170	45	168	45	30	22	8
<b>4</b>	146	194	175	89	101	95	10	10	45
<b>5</b>	241	138	198	175	98	89	22	30	18

The node locations are enumerated in Figure 36. Visual inspection indicates that Node 8 is a node that would be thought of as important given a visual inspection. Node 8 consists of every platform being alive, detected, and having an enemy platform in range while having an aggregate total score rank of 3. Nodes 182 and 17 are ranks 1 and 2 respectively and are actually quite close in positions in the graph. However, closer inspection reveals that they are of different classes and have edges that come from different sources. Notably however, they do share similar successors. Node 182 is described by Blue 1 being alive, being detected, and an enemy in range. Node 17 is similar but is Blue 2. These nodes are unique in that the detected and in range states are on without enemy platforms existing. These are due to an artifact with the sampling frequency. These nodes lie on the boundary between a neutral outcome and a blue win. Even though missiles are not directly modelled here, it is easy to understand that a blue win will come with the survival of either blue platforms. It is also likely that it could end in a draw if the missile connects. Node 45 on the opposite side of the graph is the same type of state as nodes 17 and 182 but with respect to Red 2. Node 18 is placed at rank five. Interestingly, this is a terminal state with every unit destroyed and thus a neutral outcome. The existence of node 18 high in the aggregate total is indicative of the absolute value of the z-scoring methodology. In most of the metrics, Node 18 is poorly scored.

The identification of these nodes through quantitative means provides justification for

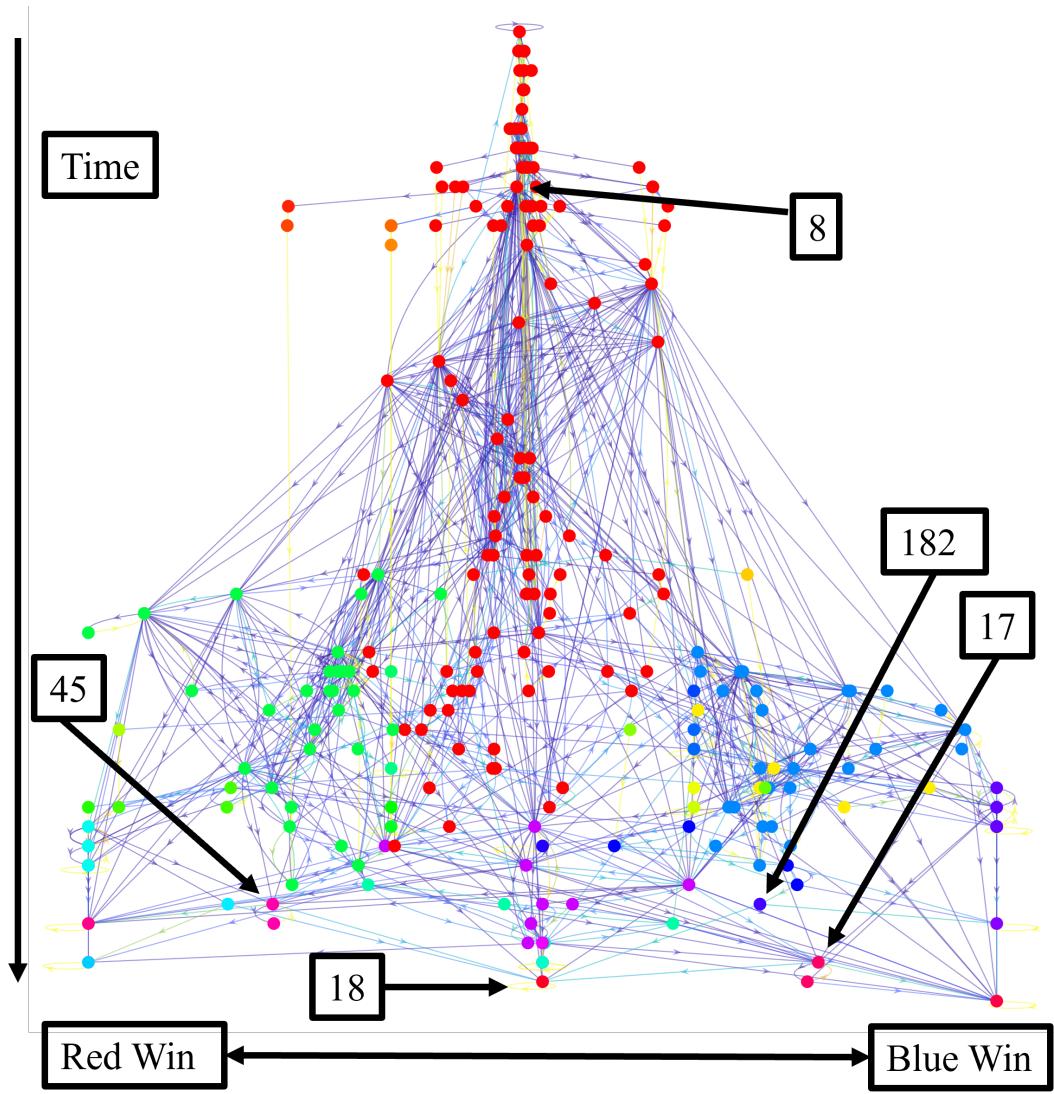


Figure 36: Locations of the top five ranked total score nodes

an analyst to further study the various sections of the scenario. Using a tool generated by OSU, an analyst can check through all of the Markov chains recorded, and pull out specific runs of the simulations that contain the nodes in question. From here, an analyst could create playback files and perform a deep dive into further causes in the simulation that could be mitigated by various strategies. Should it be desired, an analyst could work down the total score rankings and perform explorations as needed to further explain the scenario. Without these rankings, an analyst would have to use engineering intuition and random luck to search for the needle in the haystack. Utilizing this methodology, statistically significant and quantifiable justifications can be made in selecting certain phases in a scenario for further exploration.

---

---

# CHAPTER 7

---

## CONCLUSIONS

### 7.1 Results

Analytical wargamining has evolved from humble beginnings as simple programs on room size computers to complex layered interactions that can be reasonably ran on workstation desktops. With this evolution, has come the need for new techniques that could be used to make the information usable. One of the challenges faced by analysts is the sheer volume of potential information that can be recorded. As a way to potentially solve this problem, a method was described to create features that use the kill chain, F2T2EA, as a starting point. These features were created with the intention of explaining how the scenario progressed.

Another problem faced by analysts is the ability to determine where in a scenario they should look. Thus, it was proposed to create a meta-Markov model. This model being made up of smaller, individual run, Markov chains describing how the features changed at each time. In doing this, it would allow for an analyst to aggregate runs together to determine the numerous different ways a scenario could go. It was then shown that this meta-Markov model had the ability to replicate the underlying information. With the information embedded in

this format, metrics could be applied that would score nodes based on various criterion that could be important to an analyst. Using these various metrics, and depending on the desires of the analyst, nodes in the chain could be automatically detected. With these identified nodes, an analyst would have anchor points in the scenario for where to continue looking to answer the questions being posed, should the identified nodes not provide the answers directly.

## 7.2 Future Work

The work performed used the residual based around the number of transitions in a run. This metric, though useful to determine an average and distribution of runs, is not the best metric to use. Considering that the end goal is the creation of a meta-Markov chain to model the results, and if the format of each node is already known, the better metric to use to determine when to stop would be the convergence interval of each transition probability in the graph. This would allow for the confidence of each transition probability to be known within a certain level of confidence to further ensure that the subsequent analyses are using the best possible data possible. As it currently stands, the convergence interval for each transition probability is not calculated.

When working with scalability, the calculations performed made use of only Type I through Type IV state types. It is useful to understand the potential resulting graphs that will arise when implementing Type V states. It can be reasonably assumed that the resulting graph and number of edges that will come from Type V will be larger than the other types. Similarly, the work performed exclusively focused on single type graphs. This information is useful, but it is necessary to understand the interactions of Type I and II states when combined together and how they will effect each other in the resulting graph. Once the number of edges can be determined, it is useful to know what the graph adjacency matrices look like. If this is performed, comparisons between simulated results and analytical results

can be made. This comparison could reveal edges that should not exist that do in the results, potentially indicating faults in the recording process. This comparison also allows for the creation of a metric that can score nodes based on their level of similarity or dissimilarity to the analytical graph.

For the methodology demonstrated. It would be useful to test its usage in multiple different combat domains, to further determine the viability of the feature selection framework. The metrics used proved to be particularly useful in the demonstration of determining unique nodes of interest beyond simple visual inspections. It could however prove to be useful to generate an algorithm or metric that could pull out sets of interconnected states. These states in that set could then be compared against other sets in unique ways to reveal new information.

---

## BIBLIOGRAPHY

- [1] M. Ashtiani, A. Salehzadeh-Yazdi, Z. Razaghi-Moghadam, H. Hennig, O. Wolkenhauer, M. Mirzaie, and M. Jafari, “A systematic survey of centrality measures for protein-protein interaction networks,” *BMC Systems Biology*, vol. 12, no. 1, p. 80, Dec. 2018. [Online]. Available: <https://bmcsystbiol.biomedcentral.com/articles/10.1186/s12918-018-0598-2>
- [2] N. Striker, “Digital Engineering.” [Online]. Available: <https://inl.gov/digital-engineering/>
- [3] “Best Simulation Software 2023 | Capterra.” [Online]. Available: <https://www.capterra.com/simulation-software/>
- [4] D. P. Raymer, *Aircraft design: a conceptual approach*, 5th ed., ser. AIAA education series. Reston, VA: American Institute of Aeronautics and Astronautics, 2012.
- [5] T. D. West and B. M. Birkmire, “AFSIM: The Air Force Research Laboratory’s Approach to Making M&S Ubiquitous in the Weapon System Concept Development Process,” *Cyber Security & Information Systems Information Analysis Center*, vol. 7, no. 3, pp. 50–55, Dec. 2019. [Online]. Available:

able: <https://csiac.org/articles/afsim-the-air-force-research-laboratorys-approach-to-making-ms-ubiquitous-in-the-weapon-system-concept-development-process/>

- [6] J. Zeh, B. M. Birkmire, N. A. Chinnici, P. D. Clive, J. A. Johnson, A. W. Krisby, J. E. Marjamaa, L. B. Miklos, M. J. Moss, and S. P. Yallaly, “AFSIM Version 2.0 Overview and Technical Reference,” Aug. 2016, case #: 88ABW-2016-6198.
- [7] J. Appleget, R. Burks, and F. Cameron, *The Craft of Wargaming*. Naval Institute Press, 2020.
- [8] “War Game,” Dec. 2021. [Online]. Available: <https://nso.nato.int/natoterm/Web.mvc>
- [9] “JP 5-0 Joint Planning,” Dec. 2020. [Online]. Available: [https://irp.fas.org/doddir/dod/jp5\\_0.pdf](https://irp.fas.org/doddir/dod/jp5_0.pdf)
- [10] P. D. Clive, J. A. Johnson, M. J. Moss, J. M. Zeh, B. M. Birkmire, D. D. Hodson, and S. Louis, “Advanced Framework for Simulation, Integration and Modeling (AFSIM).”
- [11] N. Hanlon, E. Garcia, D. Casbeer, and M. Pachter, “AFSIM Implementation and Simulation of the Active Target Defense Differential Game,” in *2018 AIAA Guidance, Navigation, and Control Conference*. Kissimmee, Florida: American Institute of Aeronautics and Astronautics, Jan. 2018. [Online]. Available: <https://arc.aiaa.org/doi/10.2514/6.2018-1582>
- [12] I. J. Bernal, “Optimization of Processes with High Levels of Outcome Variability Factoring in Risk,” Master’s thesis, The Ohio State University, 2021. [Online]. Available: [https://etd.ohiolink.edu/apexprod/rws\\_etd/send\\_file/send?accession=osu1618826624327751&disposition=inline](https://etd.ohiolink.edu/apexprod/rws_etd/send_file/send?accession=osu1618826624327751&disposition=inline)
- [13] C. D. Connors, “Agent Based Modeling Methodology for Analyzing Weapon Systems,” Master’s thesis, Air Force Institute of Technology, Mar. 2015. [Online]. Available: <https://scholar.afit.edu/etd/103>

- [14] W. W. LaMorte, “Central Limit Theorem,” Jul. 2016. [Online]. Available: [https://sphweb.bumc.bu.edu/otlt/mph-modules/bs/bs704\\_probability/BS704\\_Probability12.html](https://sphweb.bumc.bu.edu/otlt/mph-modules/bs/bs704_probability/BS704_Probability12.html)
- [15] “Andrey Andreyevich Markov | Russian mathematician | Britannica,” Jan. 2023. [Online]. Available: <https://www.britannica.com/biography/Andrey-Andreyevich-Markov>
- [16] S. M. Ross, *Introduction to probability models*, 10th ed. Amsterdam ; Boston: Academic Press, 2010, oCLC: ocn444116127.
- [17] P. T. Biltgen, “A Methodology for Capability-Based Technology Evaluation for Systems-of-Systems.pdf,” Ph.D. Dissertation, Georgia Institute of Technology, 2007. [Online]. Available: <https://smartech.gatech.edu/handle/1853/14520>
- [18] D. N. Tryhorn, “Exploring Fog of War Concepts in Wargame Scenarios,” Master’s thesis, Air Force Institute of Technology, Mar. 2021. [Online]. Available: <https://scholar.afit.edu/etd/4909>
- [19] J. B. Crumpacker, “Air Combat Maneuvering via Operations Research and Artificial Intelligence Methods,” Master’s thesis, Air Force Institute of Technology, Mar. 2021. [Online]. Available: <https://scholar.afit.edu/etd/4922>
- [20] B. A. Bloye, “Optimizing the Air-to-Ground Kill Chain for Time-Sensitive Targets,” *Naval Postgraduate School*, p. 74, Sep. 2009. [Online]. Available: <https://apps.dtic.mil/sti/pdfs/ADA509155.pdf>
- [21] C. S. Wasson, “3.3.1 System Phases, Modes, and States: Solutions to Controversial Issues,” *INCOSE International Symposium*, vol. 21, no. 1, pp. 279–294, Jun. 2011. [Online]. Available: <https://onlinelibrary.wiley.com/doi/10.1002/j.2334-5837.2011.tb01205.x>

- [22] E. W. Weisstein, “Hamming Graph,” publisher: Wolfram Research, Inc. [Online]. Available: <https://mathworld.wolfram.com/>
- [23] L. Freeman, “Going the wrong way on a one-way street: Centrality in physics and biology,” *Journal of Social Structure - JoSS*, Jan. 2008. [Online]. Available: [https://www.researchgate.net/profile/Linton-Freeman-2/publication/228928101\\_Going\\_the\\_wrong\\_way\\_on\\_a\\_one-way\\_street\\_Centrality\\_in\\_physics\\_and\\_biology/links/543fe1540cf2be1758cfee0b/Going-the-wrong-way-on-a-one-way-street-Centrality-in-physics-and-biology.pdf](https://www.researchgate.net/profile/Linton-Freeman-2/publication/228928101_Going_the_wrong_way_on_a_one-way_street_Centrality_in_physics_and_biology/links/543fe1540cf2be1758cfee0b/Going-the-wrong-way-on-a-one-way-street-Centrality-in-physics-and-biology.pdf)
- [24] L. C. Freeman, “A Set of Measures of Centrality Based on Betweenness,” *Sociometry*, vol. 40, no. 1, p. 35, Mar. 1977. [Online]. Available: <https://www.jstor.org/stable/3033543?origin=crossref>
- [25] “CentiServer - Centrality - ClusterRank.” [Online]. Available: <https://www.centiserver.org/centrality/ClusterRank/>
- [26] “CentiServer - Centrality - MNC Maximum Neighborhood Component.” [Online]. Available: [https://www.centiserver.org/centrality/MNC\\_Maximum\\_Neighborhood\\_Component/](https://www.centiserver.org/centrality/MNC_Maximum_Neighborhood_Component/)
- [27] “CentiServer - Centrality - DMNC-Density of Maximum Neighborhood Component.” [Online]. Available: [https://www.centiserver.org/centrality/DMNC-Density\\_of\\_Maximum\\_Neighborhood\\_Component/](https://www.centiserver.org/centrality/DMNC-Density_of_Maximum_Neighborhood_Component/)
- [28] “CentiServer - Centrality - Harary.” [Online]. Available: [https://www.centiserver.org/centrality/Harary\\_Graph\\_Centrality/](https://www.centiserver.org/centrality/Harary_Graph_Centrality/)
- [29] “CentiServer - Centrality - Information Centrality.” [Online]. Available: [https://www.centiserver.org/centrality/Information\\_Centrality/](https://www.centiserver.org/centrality/Information_Centrality/)

- [30] T. W. Valente, K. Coronges, C. Lakon, and E. Costenbader, “How Correlated Are Network Centrality Measures?” *Connections (Toronto, Ont.)*, vol. 28, no. 1, pp. 16–26, Jan. 2008. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2875682/>
- [31] N. A. Christakis, “The Spread of Obesity in a Large Social Network over 32 Years,” *N Engl J Med*, vol. 357, pp. 370–379, 2007.
- [32] E. Gansner, E. Koutsofios, S. North, and K.-P. Vo, “A technique for drawing directed graphs,” *IEEE Transactions on Software Engineering*, vol. 19, no. 3, pp. 214–230, Mar. 1993. [Online]. Available: <http://ieeexplore.ieee.org/document/221135/>
- [33] W. Barth, P. Mutzel, and M. Junger, “Simple and Efficient Bilayer Cross Counting,” 2004.
- [34] U. Brandes and B. Köpf, “Fast and Simple Horizontal Coordinate Assignment,” G. Goos, J. Hartmanis, J. van Leeuwen, P. Mutzel, M. Jünger, and S. Leipert, Eds., vol. 2265. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 31–44, book Title: Graph Drawing Series Title: Lecture Notes in Computer Science. [Online]. Available: [http://link.springer.com/10.1007/3-540-45848-4\\_3](http://link.springer.com/10.1007/3-540-45848-4_3)
- [35] T. M. J. Fruchterman and E. M. Reingold, “Graph drawing by force-directed placement,” *Software: Practice and Experience*, vol. 21, no. 11, pp. 1129–1164, Nov. 1991. [Online]. Available: <https://onlinelibrary.wiley.com/doi/10.1002/spe.4380211102>