NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

UNMANNED SURFACE LOGISTICS CONCEPT OF SUPPORT

by

Tai-shan Lin

March 2019

Thesis Advisor:	Paul J. Sanchez
Co-Advisor:	Douglas J. MacKinnon
Second Reader:	Susan M. Sanchez

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

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

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

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

4. TITLE AND SUBTITLE UNMANNED SURFACE LOGISTICS CONCEPT OF SUPPORT	**5. FUNDING NUMBERS**
6. AUTHOR(S) Tai-shan Lin	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

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

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

13. ABSTRACT (maximum 200 words)

The purpose of the Navy Supply Corps is to manage the logistical pipeline, so resources are delivered to the warfighter as required. Naval units are easily able to replenish their stores while they are in port, but difficulties arise when they get underway to conduct missions and training exercises.

The use of unmanned systems introduces a new naval unit class with many beneficial characteristics, including autonomous control for which minimal human supervision is required, reliability demonstrated by COLREGS, and spacious cargo transportability evidenced by enough topside space for a 20-ft. ISO container.

This thesis seeks to identify key influential factors and provide useful insights to logistically support naval readiness and the naval units' continued ability to complete their missions. Modeling and analytical innovations used in this research include implementation of a discrete event simulation program, use of design of experiments, and sophisticated statistical analysis. Results from the analysis indicate that the rate of new requests, USVs' top speed, and number of USVs have the most impact on turnaround times for both mission duration and request fulfillment. Properly utilized, USVs can be a strong contributor to the success of U.S. Navy missions.

14. SUBJECT TERMS USV, unmanned surface vehicle, drone, logistics, RAS	**15. NUMBER OF PAGES** 115
	16. PRICE CODE

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

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

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

UNMANNED SURFACE LOGISTICS CONCEPT OF SUPPORT

Tai-shan Lin
Lieutenant Commander, United States Navy
BS, University of Texas at Austin, 2001

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
March 2019

Approved by: Paul J. Sanchez
Advisor

Douglas J. MacKinnon
Co-Advisor

Susan M. Sanchez
Second Reader

W. Matthew Carlyle
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The purpose of the Navy Supply Corps is to manage the logistical pipeline, so resources are delivered to the warfighter as required. Naval units are easily able to replenish their stores while they are in port, but difficulties arise when they get underway to conduct missions and training exercises.

The use of unmanned systems introduces a new naval unit class with many beneficial characteristics, including autonomous control for which minimal human supervision is required, reliability demonstrated by COLREGS, and spacious cargo transportability evidenced by enough topside space for a 20-ft. ISO container.

This thesis seeks to identify key influential factors and provide useful insights to logistically support naval readiness and the naval units' continued ability to complete their missions. Modeling and analytical innovations used in this research include implementation of a discrete event simulation program, use of design of experiments, and sophisticated statistical analysis. Results from the analysis indicate that the rate of new requests, USVs' top speed, and number of USVs have the most impact on turnaround times for both mission duration and request fulfillment. Properly utilized, USVs can be a strong contributor to the success of U.S. Navy missions.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

ACTUV	Anti-Submarine Warfare Continuous Trail Unmanned Vessel
ARPA	Advanced Research Projects Agency
ASW	Anti-Submarine Warfare
CASREP	Casualty Report
CLF	Combat Logistics Force
CNO	Chief of Naval Operations
COLREGS	International Regulations for Preventing Collisions at Sea
COMNAVSUPSYSCOM	Commander, Naval Supply Systems Command
CONREP	Connected Replenishment-at-Sea
CONUS	Continental U.S.
DARPA	Defense Advanced Research Projects Agency
DES	Discrete Event Simulation
DOD	Department of Defense
DON	Department of Navy
FAD	Force Activity Designator
FIFO	First-In-First-Out
FLC	Fleet Logistics Center
FSC	Federal Supply Class
ISO	International Standardization for Organization

ISR	Intelligence, Surveillance, and Reconnaissance
MASINT	Measurement and Signature Intelligence
MDUSV	Medium Displacement Unmanned Surface Vehicle
MILSTRIP	Military Standard Requisitioning and Issue Procedures
MSC	Military Sealift Command
MSCLANT	MSC Atlantic
MSCPAC	MSC Pacific
MSCEURAF	MSC Europe and Africa
MSCCENT	MSC Central
MSCFE	MSC Far East
NAVSUP	Naval Supply Systems Command
NAVSUP BSC	NAVSUP Business Systems Center
NAVSUP GLS	NAVSUP Global Logistics Support
NAVSUP WSS	NAVSUP Weapons Systems Support
NEXCOM	Navy Exchange Service Command
NIIN	National Item Identification Number
NOLH	Nearly Orthogonal Latin Hypercube
NPS	Naval Postgraduate School
NWP	Naval Warfare Publication
NSN	National Stock Number
OCONUS	Outside the Continental U.S.
ONR	Office of Naval Research

PM	MSC Program
RDD	Required Delivery Date
SOF	Special Operations Forces
UAV	Unmanned Aerial Vehicle
UIC	Unit Identification Code
UND	Urgency of Need Designator
USN	U.S. Navy
USV	Unmanned Surface Vehicle
USTRANSCOM	U.S. Transportation Command
VERTREP	Vertical Replenishment-at-Sea

THIS PAGE INTENTIONALLY LEFT BLANK

Executive Summary

Chief of Naval Operations Admiral John Richardson's message of May 17, 2017, titled "The Future Navy," presents two consistent conclusions from studies and works conducted by many institutions. "First, the nation needs a more powerful Navy, on the order of 350 ships, that includes a combination of manned and unmanned systems. Second, more platforms are necessary but not sufficient. The Navy must also incorporate new technologies and new operational concepts" (Richardson 2017).

In order to support operations over a large area, naval logistics must be robust, versatile, and flexible. With the development of the unmanned surface vehicle (USV), not only is there an additional unit with new technologies that can now be added to our current fleet force, a variety of choices and options can be considered for different employment and logistical concepts.

The logistical concept of operations for the USV raises many interesting questions in its initial development. Some of these involve queries such as how fleet sustainability is affected by the number and speed of USVs, the number of supply requests tasked to the USV fleet, or the distribution of loading and offloading times. In order to attempt to answers these questions, a simulation model was constructed and combined with an experimental design that varied several input factors over reasonable ranges. Several measures of effectiveness (MOEs) were obtained from the simulation experiment, including the average utilization of the Fleet Logistics Centers (FLCs), the average utilization of the USVs, and the average mission time for USVs. The results were then analyzed statistically to determine the direct impact of the various factors as well as how they interacted to impact operational sustainability.

The analysis phase yielded the following insights.

- The baseline values led to some results that were in a steady-state condition while others were not. These results alone could provide valuable insights for the USVs concept of operation as well as additional considerations for the decision making process.
- The average rate at which new requests enter the system is a major factor, influencing

several MOEs. This seemed to be a dominant force in straining the model. If too many requests are generated within a short period of time, the system does not seem to be able to handle this workload.

- The "Average USV Mission Time" is mostly influenced by the speed of the USV. As expected, USVs deliver items faster, on average, when they travel quickly.
- At its highest speeds, the USV seems to be able to handle the majority of the workload generated by requesting entities. However, as mentioned before, if the requests flood the system at an extremely high rate, the USVs will not have the capacity to handle all of the requests.
- Building more USVs seems to help alleviate the workload requests. At lower numbers of USVs, the new request generation means will have a more substantial impact on the average USV mission times.

Two other MOEs are important to the requesting entities: the average request turnaround time, and the average number of unfilled requests in the queue. These MOEs were examined for a second set of data after a coding error found during the initial analysis process was identified and corrected, and the MOEs' values were included in the simulation output. These two MOEs are positively correlated with one another. The more requests there were remaining in queue, the longer the wait times for the unfilled requests.

Side-by-side comparisons of the average USV mission times and average request turnaround times show very minor differences after corrections in the code. This was confirmed by subsequent analysis.

With the concept of operations for the USV still being in its testing phases, the opportunities to expand on this area of research is boundless. The base objects such as the FLC, USV, and requests in the model have been established but can be further refined. Other improvements to the model can include improved geographical construct of the area of operations, calculations of opportunity costs gained or lost, improved prioritization of tasking, scheduled deviations and interruptions, development of a user-friendly interface to support operational planning, and optimizing the balance between the USV with other assets capable of logistical support.

With the USV physically capable of handling requests for various requesting entities, the model has shown USVs to be a viable option in incorporating this unit as part of the concept of operations for logistical support. Naval logistics' crucial portion of the warfighting capability is to support readiness for the warfighter. Incorporating this versatile asset to the current inventory of ships with its adaptive technology is a logistical force multiplier that should be leveraged and considered in future operations to modernize the force as the Chief of Naval Operations had intended.

References

Richardson JM (2017) The future navy. Speech, https://www.navy.mil/navydata/people/cno/Richardson/Resource/TheFutureNavy.pdf.

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgments

I whole-heartedly thank my advisors, Professor Paul Sanchez, Research Associate Professor Doug MacKinnon, and Professor Susan Sanchez, for the continued support through the entirety of this thesis project. This thesis could not have been written without their guidance, experience, and keen sense of the best way to proceed on this topic, especially as broad a topic as this. Seeing it through to the end, I have a much better and more thorough understanding of the SEED center, modeling, and analysis process. As I had once mentioned to them toward the end of the project, "if only I had known at the start of the research what I know now..."

Thanks to my wife, Emi, for her patience, love, and encouragement through the entire process. I am truly blessed and grateful because of her. Also, to my family in Texas (mother, brother, and sister) and my father in Taiwan: thank you for your support even from great distances. And, to my pet dog, Gohan, who managed to get enough sleep for everyone during my entire time here at Naval Postgraduate School, I am jealous...

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1: INTRODUCTION

The purpose of the Navy Supply Corps is to manage the logistical pipeline so that resources are delivered to the warfighter as they are required. Naval units are easily able to replenish their stores while they are in port. Difficulties arise when the naval units get underway to conduct missions and training exercises. With credible threats from countries such as China, Russia, and North Korea, the maritime environment continues to be a hazardous and difficult place to conduct operations.

An approach to tackling these difficulties directed by the Chief of Naval Operation's message requires the use of "a combination of manned and unmanned systems" in Richardson (2017). The use of unmanned systems introduces a new naval system class with many beneficial characteristics, including autonomous control, reliability, safety due, and spacious cargo transportability. The purpose of this thesis is to build a concept of application involving unmanned surface vehicles (USVs) to improve naval logistics operations for the customer.

1.1 SUPPLY CORPS

The Navy Supply Corps organization has a rich history since its inception on February 23, 1795, when Congress created the office of Purveyor of Public Supplies. Prior to this time, the newly formed government had little oversight and accountability to ensure proper management of all required procured supplies. Each purser acted independently to manage their various naval activities, and due to the lack of a unifying leadership, they had differing levels of standards. However, with the formation of this new office, the government had established a central point for governing with the ability to handle any potential inconsistencies and overcharges. President Washington nominated Tench Francis first to hold this office, and this action is symbolic for the creation of U.S. Navy Supply Corps.

Naval Supply Systems Command (2018) states that the mission of the Supply Corps is to deliver "sustained global logistics capabilities to the Navy and Joint Warfighter." In doing so, the Supply Corps can effectively support the broader mission of the Navy which is "to maintain, train and equip combat-ready naval forces capable of winning wars, deterring aggression and maintaining freedom of the seas."

Today, the Supply Corps community is trained in three main disciplines—supply chain management, acquisition management, and operational logistics—and are strategically placed around the world working alongside Navy Enlisted logistics professionals and civilians at afloat and shore-based activities. The ultimate goal is to provide and sustain mission readiness. The senior Supply Officer heads the Supply Department, which performs the following general functions.

Ashore functions:

- performs administrative functions;
- maintains enough space to store parts;
- manages stock control; and
- delivers parts to activities as needed.

Afloat functions:

- performs administrative functions;
- divides the department into logistics stock and service divisions;
- processes and manages maintenance requirements; and
- performs service functions for the well-being of crew.

1.1.1 Organizational Structure

The highest ranking, senior military officer at the top of the Department of the Navy is the Chief of Naval Operations (CNO). The CNO is directly in charge of many organizational commands (Figure 1.1). This thesis will focus on the Command, Naval Supply Systems Command (NAVSUP). The Commander in charge of NAVSUP is Commander, Naval Supply Systems Command (COMNAVSUPSYSCOM). In the naval culture, the terms for organizational names of commands (i.e., NAVSUP) and the commanders heading these commands (i.e., COMNAVSUPSYSCOM) may be used interchangeably.

1.1.2 Fleet Logistic Centers

Fleet Logistics Centers (FLCs) manage shore inventory for supply commands. They are organized into various tier levels.

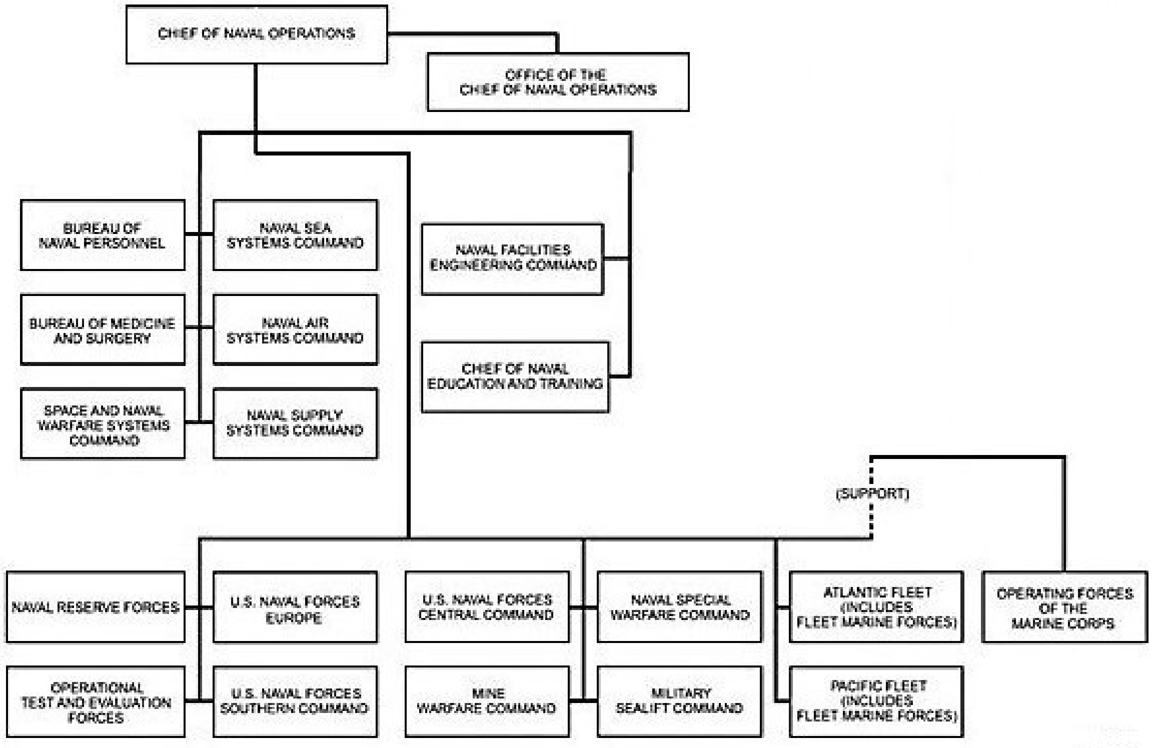


Figure 1.1. Organizational Chart: Department of Navy CNO to NAVSUP
After: United States Navy (2009).

Tier 2 commands

The commands working for NAVSUP, or tier 2 subordinate commands, include Navy Exchange Service Command (NEXCOM), NAVSUP Weapons Systems Support (NAVSUP WSS), NAVSUP Global Logistics Support (NAVSUP GLS), and NAVSUP Business Systems Center (NAVSUP BSC) (Figure 1.2).

Tier 3 commands

Multiple tier 3 FLCs work for NAVSUP GLS (Figure 1.2). FLCs play a vital first-line-of-support role as stock points of materiel for the fleet and have the ability to support the continental U.S. (CONUS), outside the continental U.S. (OCONUS), and afloat activities. To improve support for customers, these FLCs have been strategically placed around the world at locations such as Bahrain, Jacksonville, Norfolk, Pearl Harbor, Puget Sound, San Diego, Sigonella, and Yokosuka. "They are your point of contact for material and service requirements" (United States Navy 2009). Some overall managerial functions FLCs perform include monitoring fleet-wide levels of inventory, processing procured and incoming materiel, and approving final release of shipments to end-use customers. What uniquely identifies each of these material orders to FLCs is the requisition number which is composed of the service designator code, a Julian Day Number, and a document sequence number specific for that command.

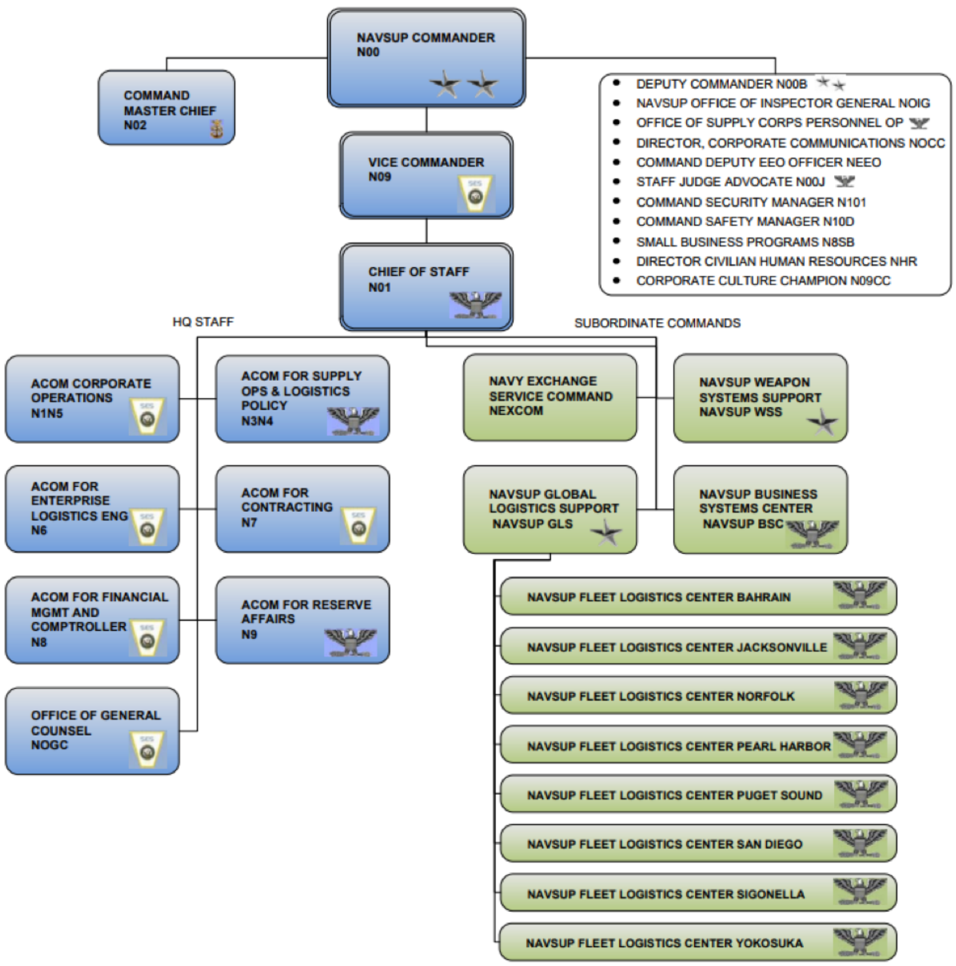


Figure 1.2. Organizational Chart: NAVSUP to NAVSUP GLS
Source: Stiner (2018).

Customer service functions provided by FLC departments

Services provided by FLCs are divided into the following categories, and organized as departments.

Inventory Control:

- provides points of contact for fleet and shore customers;
- maintains procedures for follow-up on requisitions, cancellations, and repairable turn-in items;
- ensures adequate levels of stocks and relevant records;
- determines stock materiel requirements for FLC stock and looks for ways to fulfill those requirements; and
- troubleshoots requisitions that have not been processed through the automated data processing system.

Purchase or Contracting:

- processes FLC purchase requests.

Materiel:

- maintains storage facilities for inventory control;
- maintains storage for parts and processes requisitions; and
- prepares and packs materials for proper shipment.

Fuel:

- maintains storage facilities for fuel inventory control; and
- makes fuel deliveries to nearby facilities.

1.1.3 Military Sealift Command

The Military Sealift Command (MSC) is another subordinate command under the CNO via the U.S. Transportation Command (USTRANSCOM). Their mission is to "support the joint warfighter across the full spectrum of military operations" and also to "provide on-time logistics, strategic sealift, as well as specialized missions anywhere in the world, under any condition, 24/7, 365 days a year" (Military Sealift Command 2018a, p. 1). The MSC operates from five maritime areas divided across the globe.

MSC Major Area Commands (see Figure 1.3):

MSC Atlantic (MSCLANT) – Headquartered in Norfolk, VA
MSC Pacific (MSCPAC) – Headquartered in Point Loma, CA
MSC Europe and Africa (MSCEURAF) – Headquartered in Naples, Italy
MSC Central (MSCCENT) – Headquartered in Manama, Bahrain
MSC Far East (MSCFE) – Headquartered in Sembawang Wharves, Singapore

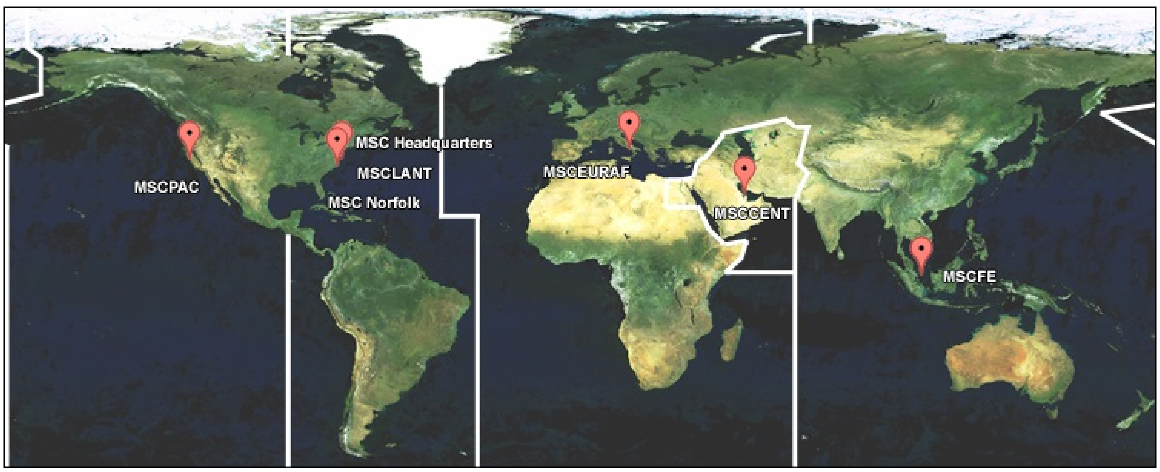


Figure 1.3. MSC Area Commands
Source: Military Sealift Command (2018b).

Each maritime area is represented by five distinct commands (Figure 1.4).

MSC Inventory and Ship Management Programs

At any given time, MSC has an inventory of approximately 110 to 120 ships at its disposal to support the fleet in various ways. With the recent re-organization of its force, they have established three major mission sets to classify their ships, which are further segregated into eight different programs (PMs). The modification in its structure was performed to make the organization more efficient and more effective in supporting its mission.

Mission Set 1—Combat Logistics Force:

PM1 Fleet Oiler
PM6 Fleet Ordnance and Dry Cargo

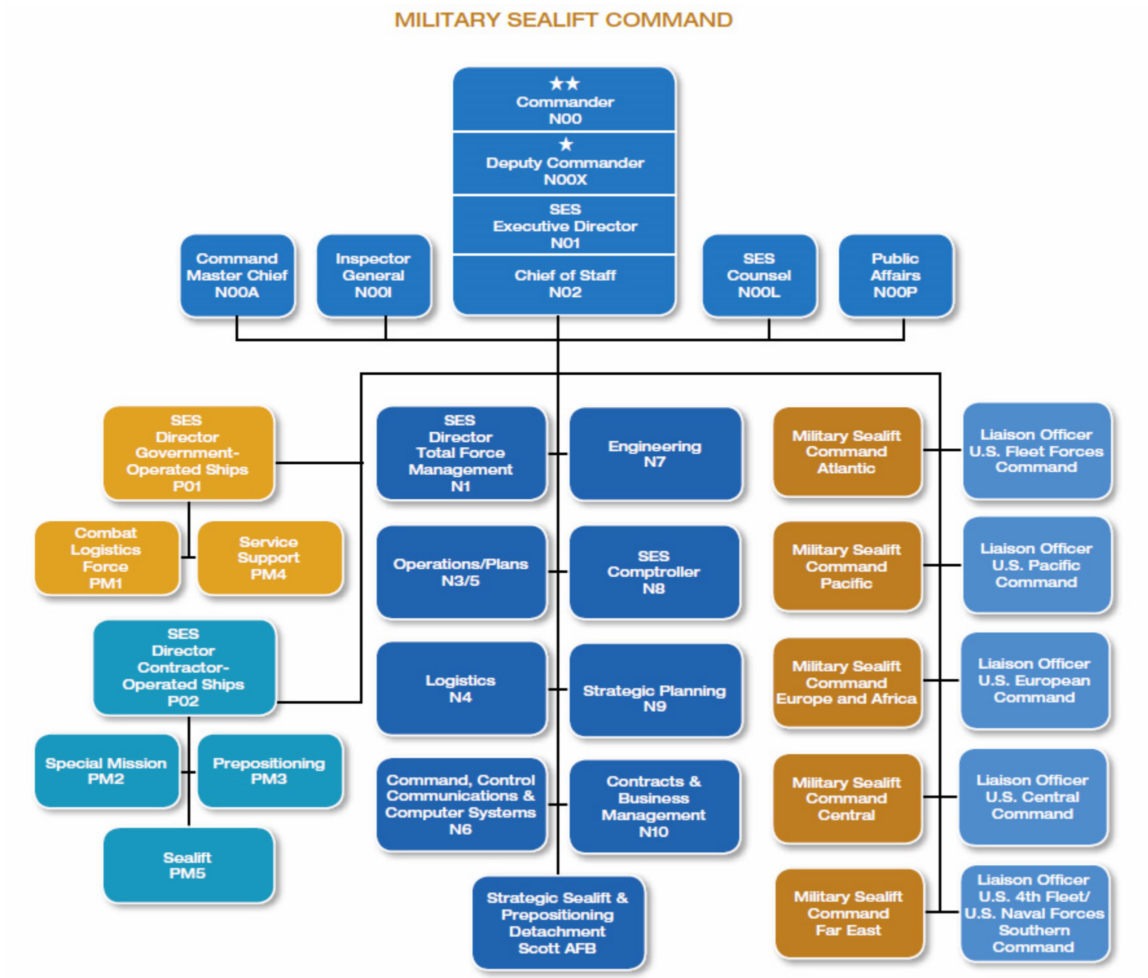


Figure 1.4. Organizational Chart: Military Sealift Command (MSC)
Source: Military Sealift Command (2012, p. 6).

Mission Set 2—Fleet Support and Special Mission:

PM2 Special Mission
PM3 Prepositioning
PM4 Service Support
PM7 Afloat Staging Command Support
PM8 Expeditionary Fast Transport

Mission Set 3—Combatant Command Support:

PM5 Sealift

(Source: Military Sealift Command 2018b)

The Combat Logistics Force

Out of the three categories, Mission Set 1—Combat Logistics Force—is the one most directly supporting the Navy. The Combat Logistics Force is composed of two programs, the Fleet Oiler program and the Fleet Ordnance and Dry Cargo program. As the name suggests, the Fleet Oiler program supports the Navy by providing different types of fuel to ship and aircraft combatants and consists of about 15 vessels in its inventory capable of performing those missions. The method of delivery is connected replenishment at sea, and all the ships are designated as T-AOs (Fleet Replenishment Oilers). The multi-product vessels in the Fleet Ordnance and Dry Cargo program re-supply ships with dry, chill, and frozen foods, spare parts, potable water, and some ammunition if necessary. 12 T-AKE (Dry Cargo/Ammunition) and 2 T-AOE (Fast Combat Supply) ships are assigned to this program. Methods of delivery for this type of transfer can consist of connected (CONREP), vertical (VERTREP), or a combination of both types of replenishment-at-seas.

Underway replenishment (UNREP) is the broad term used to describe the method of all types of delivery between two ships while the ship is underway. The two main methods used by ships are either CONREPs or VERTREPs. With CONREPs, or connected replenishments-at-sea, one servicing vessel is sailing alongside either one or two vessels receiving the service. A highline tensioned wire is rigged between these ships to facilitate transfer of fuel and stores. With VERTREPs, the ships are usually still sailing close to each other but may be further apart compared to CONREPs. One or several helicopters are the main vehicles used to lift stores, parts, or personnel and transfer them to the requesting ship(s).

The support of the MSC force augments the mission parameters and capabilities of the U.S. Navy by allowing units to "remain at sea for prolonged periods of time, possibly in areas of the world where friendly re-supply ports are not available, and remain fully ready to carry out any assigned tasks" (Pike 1999). World events are inherently unpredictable, and by having a military force capable of operating for extended periods, the Navy can respond where it is needed more quickly.

1.2 DEVELOPMENT OF THE USV CONCEPT

In 1957, when the USSR launched Sputnik 1 and demonstrated its technological superiority to the world, the United States in response created the Advanced Research Projects Agency (ARPA) to remain competitive against foreign adversaries. Later, ARPA would be renamed to the Defense Advanced Research Projects Agency (DARPA). DARPA's mission has endured through time and is "to make pivotal investments in breakthrough technologies for national security" (DARPA 2018). By working together in research and development programs with people from academic, government, and corporate institutions, they have been able to transform revolutionary concepts and ideas into more practical innovations for military capabilities and have greatly aided in the advances of military technology.

1.2.1 An Unmanned Maritime Innovation: The Sea Hunter

One such concept DARPA started developing in 2014 was an unmanned vessel capable of tracking diesel electric submarines. DARPA's new program was called the Anti-Submarine Warfare Continuous Trail Unmanned Vessel (ACTUV) program.

DARPA set the following three goals for the ACTUV Program.

Unmanned Research the possibilities of creating a potential platform where once the vessel reaches operational phase, personnel are no longer required to be physically onboard. Due to not having to sustain manning when out to sea, space and accessibility does not need to be allocated for personnel, crew support systems are not required, nor is reserve buoyancy required. By not having to support a crew, the idea was to be able to construct smaller and cheaper unit vessels.

Complex Systems When the unmanned surface vehicle deploys, the systems onboard should be sophisticated and complex enough to be able to operate autonomously, complying with the International Regulations for Preventing Collisions at Sea (COLREGS) procedures, avoiding collisions, and interacting with a smart opponent. In addition, the unmanned vessel capabilities must be operational over long distances with limited remote, supervisory, human intervention.

ASW Capable The unmanned vessel must be able to perform its primary mission, which is to track quiet diesel electric submarines over the operational area.

The ship named Sea Hunter was the product of the ACTUV program research.

An interesting addendum to these goals is that DARPA recognized that even if the Sea Hunter's main purpose was to track quiet submarines, "the core platform and autonomy technologies are broadly extendable to underpin a wide range of missions and configurations for future unmanned naval vessels" (Walan 2016). Some of these other missions include Intelligence, Surveillance, and Reconnaissance (ISR), mine countermeasures, and logistics resupply of manned Navy vessels. This thesis focuses on the logistics capabilities of the unmanned surface vehicle and evaluates the robustness of such applications in various scenarios.

1.2.2 Physical Characteristics of the Sea Hunter

Figure1.5 visually shows some of the characteristics for the USV. Additionally, from *Jane's Fighting Ships*, the characteristics for the anti-submarine warfare unmanned vessel are as follows (Saunders 2018):

Capabilities: Sea Hunter / MDUSV:

- length overall of 132 feet;
- range of 18,359 NM @ 8 knots;
- range of 10,000 NM @ 12 knots;
- on-station times of 60 to 90 days; and
- cost of $20 million.

1.2.3 Transfer to ONR

With the completion of the basic design of the Sea Hunter, DARPA transferred one of its best innovations to the Office of Naval Research (ONR). "Established on August 1, 1946, ..., the Office of Naval Research has been a pioneer in the public support of science and technology research that benefits both the naval services and the nation" (US Navy 2018). Upon transfer, ONR has continued working on improvements and tailoring the design envisioning what the future fleet could look like. The fleet is transitioning from a small

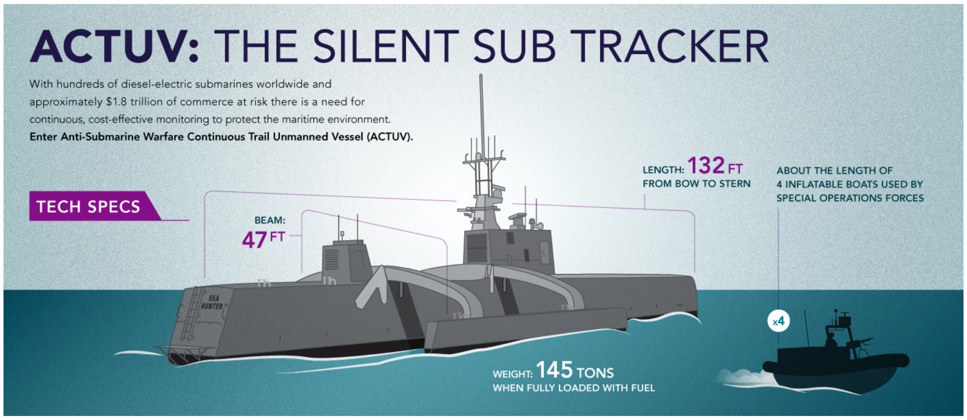


Figure 1.5. MDUSV Dimensions
Source: Rosamond (2017).

number of individual, high-value targets to larger numbers of assets that have broader and more extensive capabilities.

In April 2016, the first stage of operational testing included making sure the vessel could move in open waters. The second stage of testing in October 2016, involved making sure the sensing and autonomy suites were operational. Once the basic sensors were verified, from February to September 2017, the latest third stage testing made sure the USV would comply with at-sea rules of the road and the International Regulations for Preventing Collisions at Sea (COLREGS) to ensure the USV would be capable of avoiding dangers and other vessels during navigation.

One of the most important features of the USV is its ability to be versatile and switch between varying missions. Towards the end of the third stage of testing, DARPA and ONR jointly tested "the flexibility to handle diverse missions by switching among modular payloads" (Outreach 2018). With the USV's ability to handle various payloads, the intent of this research is to leverage the USV's capability to take on various amounts of supplies to support logistics going out to afloat maritime units. Figure 1.6 shows the four different classes of USVs —small, 7M, 7M SS, and 11M— each with different mission capabilities.

USV MP Priority	Joint Capability Area (JCA)	Seapower Pillar	USV Mission	X-Class (small)	Harbor Class (7M)	Snorkeler Class (7M SS)	Fleet Class (11M)
1	Battle Space Awareness (BSA) / Access/ Littoral Control	Sea Shield	Mine Countermeasures (MCM)		MCM Delivery, Search / Neutralization	MCM Search, Towed, Delivery, Neutralization	MCM Sweep, Delivery, Neutralization
2	BSA / Access/ Littoral Control	Sea Shield	Anti-Submarine Warfare (ASW)			Maritime Shield	Protected Passage and Maritime Shield
3	BSA, HLD, Non-Trad Ops, 7 Others	FORCEnet	Maritime Security		ISR/ Gun Payloads		7M Payloads
4	BSA / Access/ Littoral Control	Sea Shield	Surface Warfare (SUW)		SUW, Gun	SUW (Torpedo), Option	SUW, Gun & Torpedo
5	BSA / Access/ Littoral Control/ Non-Trad Ops	Sea Strike	Special Operation Forces (SOF) Support	SOF Support	SOF Support		Other Delivery Missions (SOF)
6	BSA, C&C, Net Ops, IO, Non-Trad Ops, Access, Littoral Control	Sea Strike	Electronic Warfare		Other IO	High Power EW	High Power EW
7	BSA, Stability, Non-Trad Ops, Littoral Control	Sea Shield	Maritime Interdiction Operations (MIO) Support	MIO USV for 11M L&R	ISR/ Gun Payloads		

Figure 1.6. Four Classes of USVs: Small, 7M, 7M SS, and 11M
Source: U.S. Navy (2007).

1.3 NATURE AND SCOPE OF THE PROBLEM

MSC has recently begun to show signs of trouble with readiness within the past five years. "Mission-limiting casualties among those ships have increased 250%..., mostly due to aging engine and diesel generator failures" (Ziezulewicz 2017). Most of the readiness issues MSC has encountered are related to unscheduled maintenances caused by the aging fleet. In the event of a large-scale war, these trends could potentially affect the Navy's preparedness in a negative way.

With the recent developments in technologies and changing times, MSC has recognized the importance unmanned vessels by stating "the world is increasing its use of unmanned systems, so it is vital that we leverage opportunities such as this one to maintain the Navy's superiority in that realm" (Leshak 2018). In concept, unmanned vessels are not only faster to respond since manning requirements are minimal to none, but in the event there is a casualty with a USV, a replacement USV would be much cheaper and faster to respond to any logistics requests. In doing so, MSC would be able to maintain its pace with changing times and continue its support of their Combat Logistics Force mission requirements.

Safety is also a concern. LT Knutson from the USS San Antonio stated "a RAS in general, is one of the most dangerous evolutions a ship can do, just by the nature of the event" (Hilley 2009). Due to the massive sizes of both the delivering and receiving ships, the replenishment-at-sea evolutions can be extremely dangerous, especially when compounded with the speed at which they are required to transit. Comparatively, with the USV being approximately 1–2% the size of a Destroyer, the transfer of materiel from the USV to the receiving vessel should be much safer.

Based on current trends and the continuously aging MSC fleet, "maintaining old equipment still in use after 16 years ... is becoming more and more expensive, and the kicker is that the equipment that's coming online is even more expensive to maintain and operate than the old equipment it is replacing" (Larter 2017). Given this, it would seem to make more sense to build cheaper vessels that are not huge investment units but are instead cost-efficient, easily replaceable smaller units. Cost-wise, USVs are approximately 1% the cost of an average Navy vessel and if employable, would seem to be a preferred, viable alternative.

1.4 LITERATURE REVIEW

Leidos initially began constructing the first unmanned surface vehicle back in 2014, then subsequently launched its first prototype in 2016 (Leidos 2016). With the initial testing phases focusing on COLREGs, maneuverability, and its primary missions, scant time has elapsed and there has not been a whole lot of research on the logistics mission aspects for the USV. The following represents some of the research work conducted to place the USV in various scenarios and see how effectively the USV would respond.

The Navy is currently researching various techniques to improve success against enemy encounters. One of the techniques looked at is the concept of distributed lethality. With this, the naval offensive units have enough fire power to warrant attention from the adversary and forcefully disperse their group. LT Casola began his research work on this and determined "in order to accomplish this goal while remaining fiscally responsible, it is vital to consider what types provide the most benefit while remaining affordable" (Casola 2017). The use of the unmanned surface vehicle with a mission of acting as a mobile anti-ship cruise missile-launching platform could represent a less risky alternative while also keeping costs down.

The efficacy of using USVs in routine operations including interdiction of drug trafficking, piracy, etc. has been researched as well in See (2017). As is the nature of this type of dangerous work, risk to personnel is inherent as are errors in human judgment. With effective guidance command intervals and protocols, See proved in his thesis "that using USVs with the appropriate intercept guidance for the maritime interdiction missions is a viable alternative or complement to the current operations involving only manned vessels" (See 2017).

Toh (2017) has focused his research efforts on the USV in terms of its navigational aspects. With the USV becoming operationally constrained when a human is directly involved in controlling it, he discovered the USV would need to develop its own situational awareness. Applying the MATLAB program, he was able to "demonstrate the feasibility of using a computer vision-based technique to provide a situational awareness capability to a USV" (Toh 2017). With continued improvements such as this, the USV could prove to be a more effective unit autonomously operating on its own.

Faculty research at the Naval Postgraduate School has contributed to the USV area of research as well. CAPT Kline worked with LCDR Solem to examine a scenario to see whether a P-8 Poseidon aircraft could perform better when operating with a USV in an ASW-type situation. The measures of effectiveness they used included calculations of probabilities of kill and time to kill. With the application of the USV unit, the probability of kill of the submarine unit increased by 52% while the time to kill decreased by 15% (Solem 2017). Based on the research, it was apparent the USV was an effective force multiplier and useful in those types of scenarios.

Further research identified important characteristics of the USV, such as the various ranges of the USV, speed, and payload capacities. The model developed for this research was built in such a way that it can easily be extended to include these attributes.

1.5 THESIS OVERVIEW

This thesis is limited to the maritime environment from sea to shore side. In order to compare current concepts of operations of CLF ships, the potential avenues of approach with USVs and UAVs may be used. Current replenishment-at-sea processes is a one-size-fits-all approach in which regardless of the amount of size of material, a CLF ship must embark on a journey to meet the U.S. Navy war vessel to make the logistics transfer. If the capacity of the CLF ship is maximized each time the trip is made, the resources and investments going in to the scheduled trip may make the trip operationally effective. However, if only a small quantity of parts and material are actually needed, the trip may not be as effective and potentially, this CLF ship could have been allocated to service another U.S. Navy vessel more in need of replenishment instead. With the usage of the unmanned vehicles, which are much more versatile, flexible, and can carry smaller payloads, they may be more effective and faster for delivering comparatively smaller payloads. Typically, the usual turnaround time for the preparation alone takes "coordination involving a 10-day lead time" (Hilley 2009). By simplifying the process, the purpose of this model would be to find optimal ways in which this time could be shortened.

Chapter 2 describes the methodology used beginning with problem framing and introducing the approach used to tackle the scenario. Chapter 3 provides the assumptions for the model, the resulting event graph, and finally the description of parameters, state variables, and individual events for the USV model. Chapter 4 presents the factor descriptions and ranges of interest, experimental design used, and the results and findings. Chapter 5 summarizes the key findings and opportunities for further research in the area of USV alternative mission work.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2: BACKGROUND AND METHODOLOGY

In this chapter, we provide a short description of the discrete event simulation (DES) methodology. The Navy's requisition ordering process is described as well to assist in understanding the logic of various components of the simulation model. Measures of effectiveness which capture the behaviors of interest for this research are proposed. Input requirements to the model are determined by identifying model elements that are necessary but cannot be computed within the model.

2.1 DISCRETE EVENT SIMULATION

Discrete event simulation is a well-known modeling technique (Law 2015; Banks et al. 2010; Nelson 2013; Kelton et al. 2014; Smith et al. 2017) in which a computer program models a system by tracking the evolution of its state over time. A DES model has the following components:

state variables provide an instantaneous picture of the system state, and are subject to change over time;

simulation clock keeps track of the simulated time;

event methods provide the logic of state variable transitions and subsequent event scheduling associated with a given event;

event list is a priority queue of pending events, ordered by time of occurrence;

model initialization sets initial values for all state variables and the simulation clock, and ensures that at least one event method is placed in the event list;

model executive controls the order of execution by pulling the next event notice from the event list, updating the simulation clock to the corresponding time, and executing the corresponding event method.

These fundamental components are usually supplemented with additional libraries, such as those for random variate generation.

Schruben (1983) created a concise notation for DES modeling, known as "Event Graphs."

Events are represented as vertices in the graph, and event scheduling relationships are described using directed edges. Scheduling edges can be annotated with the time delay until the scheduled event will occur (if it is non-zero), and any conditions required for the subsequent event to be scheduled. A properly constructed event graph, in conjunction with a suitable library implementation of the simulation clock, event list, and model executive, maps quite directly into a programming implementation of the model. One such library implementation is called SimpleKit (Sanchez 2009, 2018), and was used to implement the thesis model in the Ruby programming language (Matsumoto 2018).

2.2 MILSTRIP REQUISITION ORDERING PROCESS

The details of the Navy's method for ordering parts can be understood succinctly through introspection of the MILSTRIP, or Military Standard Requisitioning and Issue Procedures. The MILSTRIP is the basis for what the Navy Supply system uses to communicate between entities when ordering most materiel. It is broken down into a string of approximately 80 numerals, or codes, to communicate detailed information quickly and efficiently. The string of 80 numerals can be categorized into 17 different sections as described below. Further details of MILSTRIP processes and systems can be found in MILSTRIP MILSTRAP Desk Guide (NAVSUP 2003).

document identifier designates the purpose of the requisition (i.e., requisition, cancellation, and referrals);

routing identifier indicates the recipient's address for the requisition;

media and status code represents the status of the recipient and how the requisition will be transmitted;

stock number field the 13-digit National Stock Number (NSN) composed of the 4-digit Federal Supply Class (FSC) number and the 9-digit National Item Identification Number (NIIN);

unit of issue code the unit standard used to order materiel (i.e., foot, gallon, and pound);

quantity number of units needed based on the unit of issue;

document number identifies which service sent the requisition (i.e., Navy, Army, etc.), the unit identification code (UIC), date requisitioned using the Julian Day Number, and a unique serial number particular to that day;

demand code entries the type of demand for the requisitioned item (i.e., recurring, nonrecurring, and inactive);

supplementary address represents another entity that will receive the materiel, billing, or status in case the request-originating entity will not be the one receiving the part;

signal code represents the entity that will receive the materiel, billing, or status;

fund code indicates where the funding will be charged from;

distribution code the two parts of this code comprise of the monitoring activity (i.e., FLC and MSC) and the cognizance symbol indicating the inventory manager for that specific part;

project code represents the specific project or program of the requisitioned materiel;

priority designator is derived from the Force Activity Designator (FAD) and the Urgency of Need Designator (UND) and represents how quickly the part is needed within the FAD and UND limitations;

required delivery date (RDD) indicates a corrected date in case the default, standard delivery date is not considered soon enough;

advice code is an optional code that can provide further information on repairable materiel; and

remarks field is another optional code which may be used to clarify any special situations.

2.3 CONCEPTUAL MODEL

We now describe the primary components of the conceptual model. A detailed description of the model development appears in Chapter 3.

2.3.1 Simplified Requisition Process

For simplification purposes of this particular study, not all the sections of the MILSTRIP code will be used. The codes of interest are the quantity, stock number, and priority designator of the requisitioned materiel.

Figure 2.1 summarizes the simplified requisitioning process. The requesting entity, usually an underway vessel, sends out a MILSTRIP to a nearby FLC. The FLC will verify whether they have the materiel available for issue. If they have the part available, potential methods of delivery are researched, whether it be air or surface, and set up with the ship. If the FLC does not have the part available, it will send the MILSTRIP off as a referral to another FLC site which would have verified having the part in stock and available for issue.

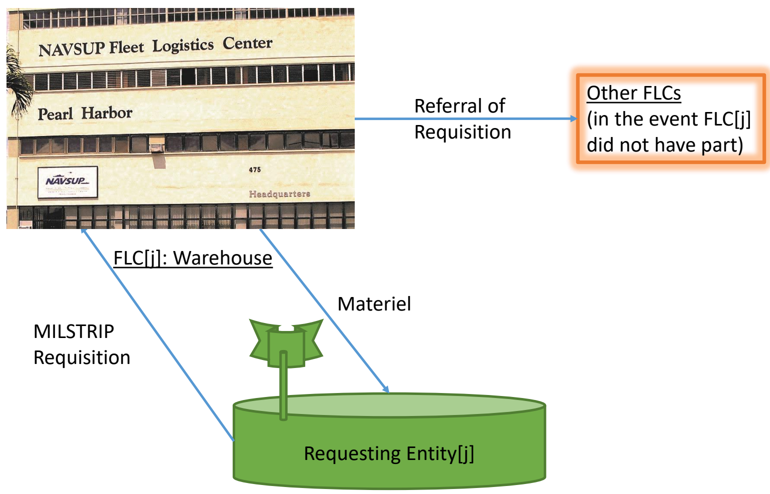


Figure 2.1. MILSTRIP Requisitions—The Navy Ordering System Source: Henggeler (2014)

.

2.3.2 Objects and Their Attributes

The operational logistics system of interest can be described based on a collection of several distinct groups of objects to accomplish its mission. Each of these objects has an associated set of attributes which maintain their respective portions of the state of the system.

FLCs (Fleet Logistics Centers), or the warehouses which stock the materiel, are where the USVs draw their parts from. FLCs are given the following attributes:

name unique nomenclatures for each FLC; and

location fixed within the area of interest.

USVs (unmanned surface vehicles) are the items of interest and are being evaluated for their value in providing efficient logistics alternatives. USVs are given the following attributes:

name uniquely identifies each of the USVs;

top speed limits the speed the USV can travel at;

cargo capacity limits the amount of cargo the USV can carry;

fuel capacity limits the amount of fuel the USV can keep in its own fuel tanks;

destination keeps track of where it will be traveling to;

moving keeps track of whether it is currently moving or not moving;

bearing identifies the ship's heading when moving; and

rendezvous location keeps track of the USV destination to deliver materiel necessary to the requesting entity.

Requests are orders for materiel and have the following attributes:

requesting name is the requesting entity's identifier;

stock number is the requested part identifier;

quantity is the required amount of the part;

priority identifies the importance level of the requested part, as described in CASREPs (Casualty Reports). Ships routinely categorize the importance of their orders for parts based on the following categories taken from the Naval Warfare Publication 1-03.1, Operational Reports (Department of Navy 2012). Note that CASREP category 1 is not used.

> **"CASREP Category 2** – A deficiency exists in mission essential equipment which causes a minor degradation in any primary mission, or major degradation or total loss of secondary mission;
>
> **CASREP Category 3** – A deficiency exists in mission essential equipment which causes a major degradation, but not the loss of a primary mission; and
>
> **CASREP Category 4** – A deficiency exists in mission essential equipment that is worse than casualty CAT 3, and causes a loss of at least one primary mission" (Department of Navy 2012).

Although the USV model does not specifically address and categorize requests in accordance with the above CASREP categories, randomized priority levels of 1 through 13 are used to mimic relative importance from high to low, respectively. Priority 1 items are dispatched before items with larger priority numbering.

rendezvous location Keeps track of where the USV will need to be headed towards in order to deliver materiel necessary to the requesting entity; and

time Keeps track of the time of request in order to prioritize requests.

A USV's availability for tasking is identified by its presence or absence in the

`USV_available` set. All USVs are initially placed in the set. USVs are removed from the set upon assignment to a mission, and added back into the set upon completion of their current mission.

2.4 MEASURES OF EFFECTIVENESS

We are interested in quantifying the system behavior as a function of the model parameterization. This is accomplished by defining suitable measures of effectiveness (MOEs) which reflect the system performance. MOEs chosen for this model are:

Average Number of Requests in Queue Over time, the number of requests that reside in the queue give insight into whether we are or are not meeting demand for materiel. Smaller average numbers of requests in queue are sustainable over time, while large and growing averages indicate that the system is not meeting demand and will not be sustainable over time.

Average FLC Utilization This represents the average proportion of time the FLCs are busy over the course of the simulation runs. If the FLCs' service rate is faster than the rate at which USVs arrive to receive service, this will result in a sustainable situation. However, if the FLCs' service rate is slower than the rate at which USVs arrive to receive service, this will result in an unsustainable situation.

Average USV Utilization This represents the average proportion of time the USVs are busy over the course of the simulation runs. If the USVs' service rate is faster than the rate at which requests arrive in the system, this will result in a sustainable situation. However, if the USVs' service rate is slower than the rate at which requests arrive in the system, this will result in an unsustainable situation.

Average USV Mission Time Tallying the start time as when the USV is tasked with a mission and the end time as when the USV delivers the materiel to the requesting unit, this statistic is the average time across all completed USV missions.

Average Completed Request Turnaround Tallying the start time as when the requesting entity places an order and the end time as when the USV delivers the materiel to the requesting unit, this statistic is the average turnaround time for all completed requests. An important distinction from "Average USV Mission Time" is this start time is earlier than when a USV is tasked with the request. Therefore, this statistic represents the customer's perspective.

Average Unfilled Request Time This statistic also represents the customer's perspective. For unfilled requests, the start time is when a request enters the system and the end time is when the halt time for the model occurs. In cases where only small numbers of requests enter the system, most requests should be handled by the system effectively with a resulting lower average time for unfilled requests. However, if too many requests enter the system and are consequently not processed, the result is a higher average time for unfilled requests indicative of a system unable to handle the workload.

It is important to consider both of the last two MOEs, rather than just one. From the customer's perspective, the "Average Completed Request Turnaround" and the "Average Unfilled Request Time" should both be low.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3:
BUILDING THE SIMULATION MODEL

This chapter provides a description of the USV simulation model, including the assumptions that were made in running the model, and detailed descriptions of what the model parameters and state variables are, and the model events that form the basic building blocks of the model. Model verification took place throughout the model coding process.

3.1 MODEL ASSUMPTIONS

Models are always an approximation of reality, and thus involve assumptions and simplifications. The assumptions made for this model are delineated below.

USVs: Always fully maintained / available Most ships and equipment require some type of maintenance to ensure proper working conditions. Application of timing and scheduling would entail designating scheduled and unscheduled maintenances for the USV. USVs entering a maintenance period would result in the USV being not available for tasking. At the time of the writing of this thesis, the USV is still in its testing phases without established, designated maintenance periods. For this reason, a simplification was made in which the USVs would be always fully functioning without entering maintenance periods.

USV -> FLC: Pier is always available for USVs to pull in When ships enter port, there are times when other ships have already moored at all the piers. If such a case were to happen, normally a vessel would have to stand by out in the ocean until a pier becomes available. With USVs being much smaller than conventional vessels and the thought that the Navy would place a great importance on the roles USVs would play, the model assumes that USVs are able to pull in to bases where FLCs are located any time it is necessary.

FLCs: Requested parts are always stocked without fail at one of the FLC locations In reality, when a part is requested by a ship, there are situations in which sometimes the part is either out of stock at the time or simply not listed to be carried by an FLC. If none of the FLCs have this item in stock at the time, the part would have to be backordered and not be available until one is shipped in from external depot sources.

The delay time for this variable can vary widely from several days to several years. With this element being close to unpredictable, the model has simplified this to the part being always available at one of the FLCs.

FLCs: Requested materiel ready-for-issue (RFI) Some materiel may be subject to a repair cycle and unavailable for issue. Repair cycles can also widely vary for several days to several years. To avoid this unpredictable behavior, the model was simplified to assume that the part would always be available for issue at one of the FLCs.

FLCs: Materiel within payload limits of the 20-feet ISO container It is possible that some parts requested by the requesting entities are physically so large they would not be able to fit in the USV's 20-feet ISO (International Organization for Standardization) container. For parts that are so large, the USVs would not be able to physically handle transport of such materiel due to weight restrictions, the fact it would affect safety during sea state maneuverings, etc. In those situations, the assumption is that the MSC fleet would instead handle the abnormally bulky items and deliver them to the requesting units.

FLCs: USV will fully replenish fuel upon reaching FLC When USVs pull in to port, it would make sense they would replenish their fuel enough to make the trip out to the requesting entity's rendezvous location and back. What makes the USV unique and desirable is that on a full tank, it is able to traverse large distances for several months. This capability can only be leveraged if the system is designed to fully replenish its fuel once it reaches the base with the FLC.

FLCs: Continuous operations Normal operations for FLC facilities are similar to civilian institutions with the appropriate business hours conducted throughout daytime. However, the assumption is made in this model the FLC is open 24/7 to completely support the needs of the USV upon entering port. Upon completion of services, the USV is unconstrained as well to depart for its next destination at will.

Rendezvous locations: Constant designated area In this model, the rendezvous locations have been arbitrarily specified to remain within a designated area in the Pacific Ocean. This square area has its four corners located at Hokkaido, Japan, Washington, U.S., northeastern Indonesia, and Guatemala. These coordinates can easily be adjusted by changing the coding where it is initialized.

Uniform standardized ship-to-ship connectors for transfer of materiel Many ships have different types of capabilities in order to take on supplies from external entities. Some can use their onboard cranes, some have helicopters, or some can transfer directly via the well deck. The USV was initially designed to conduct ASW, minesweeping, and other missions besides the logistical transfer materiel from USV to other ships. At the time of the writing of this research, the design for the USV had not included a standardized method for efficient transfers. For this reason, the transfer is assumed to be standardized and the same regardless of what platform the USV is transferring materiel to.

Oceans distance calculated as 2D rather than 3D The earth is accepted to be a 3D surface, so Great Circle distance formulas should be used to calculate distances between two locations. However, we represented the operating environment as a 2D sandbox to simplify initial model development.

3.2 MODEL FORMULATION

Event graphs are a pictorial representation of the discrete event model and an effective tool to conceptually convey the contents of the model. Events are shown as vertices, or blue circles. As each event transpires, state transitions are performed first and then event scheduling occurs. Solid arrows represent the scheduling relationships between events. Each scheduling delay is indicated by a subscripted "t" along the base of the scheduling edge. If no delay is given, it is implicitly zero. A dotted arrow represents a cancelling edge between those two events. If the condition for the cancelling edge becomes true, the event associated with the head of the cancelling edge is cancelled and instead schedules the event associated with the tail event. The USV model has no cancelling edge. Figure 3.1 depicts the USV model's event graph in its entirety.

3.2.1 Model Parameters and State Variables

Model parameters are values that do not vary over a run of the simulation. The set of all parameters defines a particular scenario for each run of the model.

Parameters

Model parameters affect the behavior of a scenario. Some parameters are easily altered between runs, while others have been hard-wired into the model. Key parameter descriptions are given below.

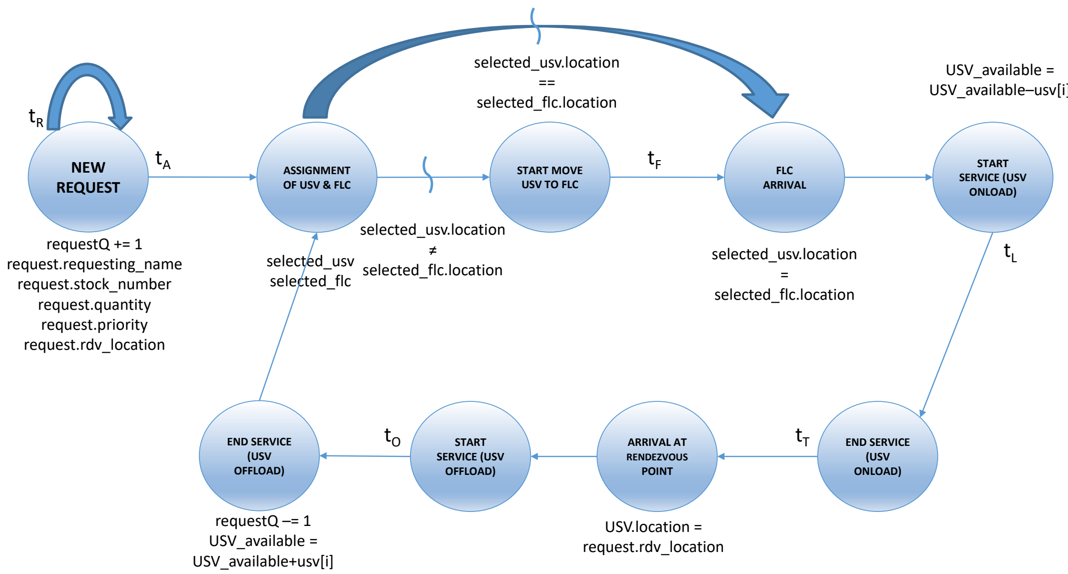


Figure 3.1. Event Graph Model

New Request Mean (Type: Decimal) This mean represents the time between requests. The larger the value, the longer the time is in between requests, resulting in fewer requests per time interval.

Shape Parameter for the New Request Distribution (Type: Decimal) The gamma distribution was chosen for this model component to provide flexible modeling of positive, continuous variables with a skewed distribution.

FLC Onload Mean (Type: Decimal) This mean represents the service time to load the USV with the appropriate parts for its mission. The service time aggregates:

- time to pull in to the harbor;
- time to moor to the pier;
- time for crane services to set up (if necessary);
- time to physically load the materiel or containers onto the USV;
- time to re-fuel;
- time to conduct any maintenance; and
- time for the USV to pull away from the pier and away from the harbor

rather than separating these out individually and summing them. The larger the value, the longer the time is for services to take place, resulting in fewer loadings accomplished per time interval.

Shape Parameter for the FLC Onload Distribution (Type: Decimal) The gamma distribution was chosen for this model component to provide flexible modeling of positive, continuous variables with a skewed distribution. The original gamma distribution curve starts at zero resulting in some onload times being zero, or occurring instantaneously. To prevent this, the curve is shifted by one hour to simulate reality instead of using straight gamma distributions.

Offload Mean (Type: Decimal) This mean represents the time to offload materiel onto the requesting vessel. The service time includes

- time to pull alongside the USV;
- time to tie up the USV to the requesting vessel;
- time to transfer the materiel from the USV to the requesting entity (cranes, if necessary); and
- time to untie and release the USV for other pending missions

rather than separating these out individually and summing them. Similar to the onload service time, the larger the value, the longer the time is for services to take place, resulting in fewer offloads accomplished per time interval.

Shape Parameter for the Offload Distribution (Type: Decimal) The gamma distribution was chosen for this model component to provide flexible modeling of positive, continuous variables with a skewed distribution.

Top USV Speed (Type: Decimal) The value represents the highest speed the USV can travel. The larger the value, the more missions it can accomplish per time interval.

Number of USVs (Type: Integer) This value represents the number of USVs allotted for the scenario. If there are enough requests in the queue and the USVs have not already reached 100% utilization, then more USVs can carry out more missions per time interval.

Number of Days to Run (Type: Integer) This value represents how many days of operations will be simulated.

State Variables

The state of the system is maintained in state variables, which can change over the course of the simulation run. The USV model incorporates the following state variables.

request When a requesting unit places an order, a request object is created with the following attributes;

request.requesting_name name of the requesting unit;

request.stock_number part requested by the requesting unit;

request.quantity amount of materiel of the specified part requested by the requesting unit;

request.priority urgency of the materiel requested by the requesting unit; and

request.rdv_location rendezvous location requested by the requesting unit.

requestQ A priority queue that stores pending requests that have not yet been assigned;

USV_available pool of USVs that are available for taking on missions;

assignedFLCCounter Tally of the number of busy FLCs;

numDeliveries Tally of the number of deliveries made;

numAssigned Tally of the number of USVs assigned;

avgTimeMission Statistics object to record the mission time data;

assignedUSVCounter Tally of the number of assigned USVs;

avgUSVUtil Statistics object to record utilization of the USV fleet; and

avgRequestTurnaround Statistics object to record turnaround time for requests.

3.2.2 Model Events

Events are points in time in which the system changes state by having state variables change or by scheduling further events. Brief descriptions of the events follow. Amplifying information on the various events used in the model can be found in the code in Appendix A.1.

Initialization In the initialization event, various sets are populated with initial values. FLCs are created and placed in the FLC set. Parts are generated with unique stock numbers, and initial stock levels are assigned to the various FLCs. Therefore, a unique inventory for each of the different FLCs is possible. USVs are created with initial starting locations and top speed attributes. Finally, the first request and the model halt are scheduled.

new_request A new request object is created with the following attributes.

time the time of the request

ship the ship name of the entity ordering the part;

stock number the part the requesting entity is ordering;

quantity the number of parts being ordered;

priority the urgency of the request; and

rendezvous location the rendezvous location for delivery of the request.

The next request event is scheduled at a time determined by the request arrival distribution. A usv_assignment event is scheduled immediately.

usv_assignment The set of all FLCs carrying the required part is paired with the set of all USVs which are currently available to evaluate which USV can provide the shortest delivery time to the rendezvous location. If no USV is available or no FLC has inventory of the requested part, then no assignment will occur. If both requirements are met, then if the USV is already on location, then an flc_arrival event is scheduled with no delay. Otherwise, a start_move event is scheduled with no delay.

start_move The assigned USV is tasked to move and told its destination. The time to destination is calculated and the end_move event is scheduled with that delay.

end_move USV is notified it has arrived at its destination.

flc_arrival A start_onload event is scheduled with no delay.

start_onload (start service) The assigned FLC counter is incremented, and its state change is recorded for statistical purposes. An end_onload event is scheduled with a delay determined by the onload service distribution.

end_onload (end service) The assigned FLC counter is decremented, and its state change is recorded for statistical purposes. A start_move is scheduled with no delay, and an arrival_rdv event is scheduled with a delay calculated based on distance to the rendezvous location and the USV's speed.

arrival_rdv A start_offload event is scheduled with no delay.

start_offload (start service) An end_offload event is scheduled with a delay determined by the offload service distribution.

end_offload (end service) Statistics are tallied for assigned USVs, average USV utilization, number of deliveries completed, average mission time, and average request turnaround time. The number of USVs assigned is decremented, and the USV is placed back in the USV available set. The priority queue is checked to see whether if it's empty, and if it is, then the closest FLC is determined and a start_move event is scheduled with no delay. Otherwise, a usv_available event is scheduled with no delay.

Translation to Model Coding into Ruby programming language

A properly constructed event graph translates quite directly into code. The descriptions given above were used to generate a Ruby implementation of the USV model, which can be found in Appendix A.1.

CHAPTER 4: IMPLEMENTATION, RESULTS, AND ANALYSIS

With the model development complete, the attention turns to experimentation. A designed experiment can be applied to examine the relationships between the various factors and the responses of interest, and "allow us to determine the driving factors, detect interactions between input variables, identify points of diminishing or increasing rates of return, and find thresholds or change points in localized areas" (MacCalman et al. 2016, p. 1).

4.1 MEASURES OF EFFECTIVENESS

Six measures of effectiveness (MOEs) that capture different aspects of the logistic system's performance over time are explored in this study. Four are investigated in detail in Sections 4.5, and we discuss these first.

The first MOE is the average number of requests in queue. Recall from Section 2.4 that if the system is in steady state, then the average number of requests in queue will stabilize. Although the actual number of requests in queue varies over time, the logistics operations are sustainable over time. Conversely, if the average number of requests in queue continues to grow over time, this indicates that the logistics system is in unstable conditions. 10,000 days, or a little over 27 years, is chosen as a sufficiently long duration to simulate the logistics system and obtain meaningful results. Because the model is executed for a fixed length of simulated time, low or high values of this MOE are indicative of steady-state or non-stationary conditions, respectively.

The second MOE is the average FLC utilization and represents the average percentage of time the FLCs are busy handling a service call for USVs in the model simulation. If the average rate at which FLCs are handling USV service calls is consistently greater than the rate at which USVs arrive at an FLC to be processed, then the average FLC utilization will tend to be lower. If the average rate at which FLCs are handling USV service calls is consistently less than the rate at which USVs arrive at an FLC to be processed, then the average FLC utilization will tend to be higher. From analysis of the FLC utilization results, the potential benefits of faster FLC service times can be determined.

The third MOE is the average USV utilization and represents the average percentage of time the USVs are busy processing a request from a requesting entity in the model simulation. If the average rate at which USVs are processing a request is consistently greater than the rate at which new requests are being generated, then the average USV utilization will tend to be lower. If the average rate at which USVs are processing a request is consistently less than the rate at which new requests are being generated, then the average USV utilization will tend to be higher. From analysis of the USV utilization results, the potential benefits of building more USVs to satisfy demand can be determined.

The fourth MOE is the average USV mission time, and represents the average length of time USVs take to deliver materiel to the customer once they accept missions from the queue. This value includes transit times, onload service times, and offload service times. Onload and offload service times require a much shorter length of time when compared to the USVs' transit times over the expanse of the ocean. As a result, the variability for the transit times will have a more significant impact than onload and offload service times. The model is designed to greedily select the USV that can complete the mission in the shortest time possible. However, if stresses such as a high number of requests in queue are placed on the system, the model will be forced to select alternate USVs that are not optimal in terms of mission times. Sensitivity analysis will show to what degree the system is able to handle this type of stress.

Choosing how to set up a USV logistics system may involve tradeoffs among these MOEs. For example, increasing the number of USVs could potentially lower the average time to complete a mission. As the number of USVs increase, the likelihood also increases for a well-positioned USV to be chosen for a minimum mission completion time as opposed to being constrained in choosing a less optimally located USV. However, as the number of USVs increase, the trade-off is that the average USV utilization will also decrease. From an economic standpoint, as more USVs are added to the system, there are diminishing returns for the amount of time each USV is kept busy, and some USVs could sit idle for days before being assigned to handle a request.

4.2 DESIGNED EXPERIMENT

A designed experiment allows us to vary model inputs or other model parameters in a structured manner to assess their effects on the MOEs of interest. To set up an experiment, the analyst needs to specify the factors to vary, suitable ranges of possible values for these factors, and an appropriate design.

4.3 MODEL INPUTS

The current USV model has nine different inputs than can be easily specified to characterize the operational logistics environment. Model inputs are described in Section 3.2.1.

4.4 FACTORS AND FACTOR RANGES

Factors are model inputs that are purposefully varied in a designed experiment to see how they affect the MOEs. Brief descriptions of the factors follow. The low and high factor settings shown in Table 4.1 were selected by the author as reasonable values for initial analysis purposes. If these inputs are reality-based and relatable with the real-world, then once the model processes these inputs and achieves results, the results will make sense in the context of the world we live in.

For example, the low and high values for `newRequestGAMean` translate to real-world requests entering the model ranging from twice per day to once every ten days. The low and high values translate to `onloadGAMean` real-world range values of mean onload service times ranging from four to twelve hours. The low and high values for `offloadGAMean` translate to real-world range values of mean offload service times ranging from one to three hours. These `onloadGAMean` and `offloadGAMean` means are based on shifted gamma distributions to avoid minimum times extremely close to zero. The decimal range of 2.0 through 10.0 for the three gamma distribution shape parameters (`newRequestGAShape`, `onloadGAShape`, and `offloadGAShape` change the shape of the respective distribution from highly skewed to only slightly skewed. The minimum `top_speed` of 8 knots is consistent with current Seahunter capabilities reported as 8–12 knots (Section 1.2.2). The top speed of 25 knots extends the current capability. `numberUSV` is perhaps the factor of primary interest for those interested in developing a concept of operations for USV logistics. The low setting of 6 USVs represents a system where there are equal numbers of USVs and

FLCs. The high value of 38 was selected because it represents a substantial increase in the number of USVs available, and so that there were exactly 33 values for this integer-valued factor.

Table 4.1. Factors and Factor Ranges

Factor	Description	Minimum	Maximum
`newRequestGAMean`	Mean Time Between New Requests (days)	0.5	10.0
`newRequestGAShape`	Shape Parameter for the New Request Distribution	2.0	10.0
`onloadGAMean`	Mean FLC Onload Time (days)	0.167	0.500
`onloadGAShape`	Shape Parameter for the FLC Onload Distribution	2.0	10.0
`offloadGAMean`	Mean Offload Time (days)	0.042	0.125
`offloadGAShape`	Shape Parameter for the Offload Distribution	2.0	10.0
`top_speed`	Top USV Speed (knots)	8	25
`numberUSV`	Number of USVs	6	38

Six of the factors are determined by the use of gamma distributions in the model. The gamma was chosen because its parameters provide flexibility to yield a variety of shapes while preserving a specified mean. Exploring these factors is of interest for practical reasons. If the MOEs are insensitive to them, then future researchers will not have to expend energy in trying to discern what specific distributions to apply to their work, and future logistics planners will have confidence that guidance from simulation studies is not overly sensitive to real-world variations from specific modeling assumptions.

4.4.1 Nearly orthogonal Latin hypercube design

The nearly orthogonal Latin hypercube (NOLH) worksheet was used as the foundation to design the experiments. Designs such as NOLHs can be applied to examine the relationships between the various factors and "allow us to determine the driving factors, detect interactions between input variables, identify points of diminishing or increasing rates of return, and find thresholds or change points in localized areas" (MacCalman et al. 2016).

Figure 4.2 shows the NOLH design for the factors and ranges specified above. A column for `haltTime` is also included. Although it is kept constant at 10,000 days for this experiment, it is a model input that could be changed in future studies. The design has 33 design points, and tests each of the 8 input factors at 33 levels. For comparison purposes, if a full factorial (gridded) design for 8 factors at 33 levels each was used, there would be approximately 1.4E+12 design points. Given that the model takes about one second to simulate a given design point, this would translate to over 44,000 years to perform a single replication of the study. Using the efficiency of the NOLH design allowed completion of 10 replications of 11 stacks of the design in a few hours on a current generation laptop computer.

One of the advantages of applying the nearly orthogonal Latin hypercube to the experiments is the space-filling property of the design. As more of the space is tested in the experimental design space, more potential insights can be gained to understand the solution space better. Based on the design of experiments from Figure 4.2, the space-filling characteristic can be seen in Figure 4.1 for each pairing of the tested input factors. Being able to test the entire spectrum of combinations of the factors is critically important to ensuring certain scenarios are not missed causing misinterpretation of the analyses.

4.4.2 IMPLEMENTATION

10 replications were performed on 353 design points resulting in 3,530 experiments completed in approximately four hours. Simulated runs were conducted on a MacBook Pro with a 3.1GHz processor Intel Core i7 and a memory of 16GB 2133MHz LPDDR3. The operating system used was the macOS High Sierra Version 10.13.6.

Table 4.2. Nearly Orthogonal Latin Hypercube Matrix

low level	**0.5**	**2**	**0.166667**	**2**	**0.041667**	**2**	**8**	**6**	**10000**
high level	**5**	**10**	**0.5**	**10**	**0.125**	**10**	**25**	**38**	**10000**
decimals	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**0**	**0**
factor name	**newRequestGAMean**	**newRequestGAShape**	**onloadGAMean**	**onloadGAShape**	**offloadGAMean**	**offloadGAShape**	**top_speed**	**numberUSV**	**haltTime**
	5	2.75	0.3125	3.5	0.11458	7	19.6875	21	10000
	4.57813	10	0.20833	5	0.08073	3.5	20.75	16	10000
	4.4375	5.5	0.46875	3.25	0.04427	6.75	20.21875	7	10000
	3.03125	9	0.5	5.25	0.11979	3.25	21.8125	8	10000
	4.71875	2.25	0.32292	3.75	0.09896	7.75	14.90625	24	10000
	4.85938	9.5	0.27083	4.25	0.07813	3.75	10.65625	34	10000
	3.59375	5.75	0.48958	4	0.04167	7.25	14.375	35	10000
	2.89063	7.5	0.47917	4.75	0.11719	4	11.71875	38	10000
	3.45313	4	0.23958	6.25	0.10156	4.5	8	12	10000
	3.875	7.25	0.26042	7.5	0.0599	6.25	9.59375	18	10000
	3.73438	3.75	0.41667	9.75	0.07031	2.5	10.125	11	10000
	4.01563	7.75	0.38542	9.5	0.10417	9.75	15.96875	19	10000
	3.17188	3.25	0.22917	6.5	0.09115	3	24.46875	31	10000
	4.29688	6.75	0.29167	9	0.05469	6.5	23.9375	29	10000
	3.3125	3.5	0.44792	9.25	0.07292	2	19.15625	30	10000
	4.15625	7	0.36458	10	0.10938	9.25	17.5625	27	10000
	2.75	6	0.33333	6	0.08333	6	16.5	22	10000
	0.5	9.25	0.35417	8.5	0.05208	5	13.3125	23	10000
	0.92188	2	0.45833	7	0.08594	8.5	12.25	28	10000
	1.0625	6.5	0.19792	8.75	0.1224	5.25	12.78125	37	10000
	2.46875	3	0.16667	6.75	0.04688	8.75	11.1875	36	10000
	0.78125	9.75	0.34375	8.25	0.06771	4.25	18.09375	20	10000
	0.64063	2.5	0.39583	7.75	0.08854	8.25	22.34375	10	10000
	1.90625	6.25	0.17708	8	0.125	4.75	18.625	9	10000
	2.60938	4.5	0.1875	7.25	0.04948	8	21.28125	6	10000
	2.04688	8	0.42708	5.75	0.0651	7.5	25	32	10000
	1.625	4.75	0.40625	4.5	0.10677	5.75	23.40625	26	10000
	1.76563	8.25	0.25	2.25	0.09635	9.5	22.875	33	10000
	1.48438	4.25	0.28125	2.5	0.0625	2.25	17.03125	25	10000
	2.32813	8.75	0.4375	5.5	0.07552	9	8.53125	13	10000
	1.20313	5.25	0.375	3	0.11198	5.5	9.0625	15	10000
	2.1875	8.5	0.21875	2.75	0.09375	10	13.84375	14	10000
	1.34375	5	0.30208	2	0.05729	2.75	15.4375	17	10000

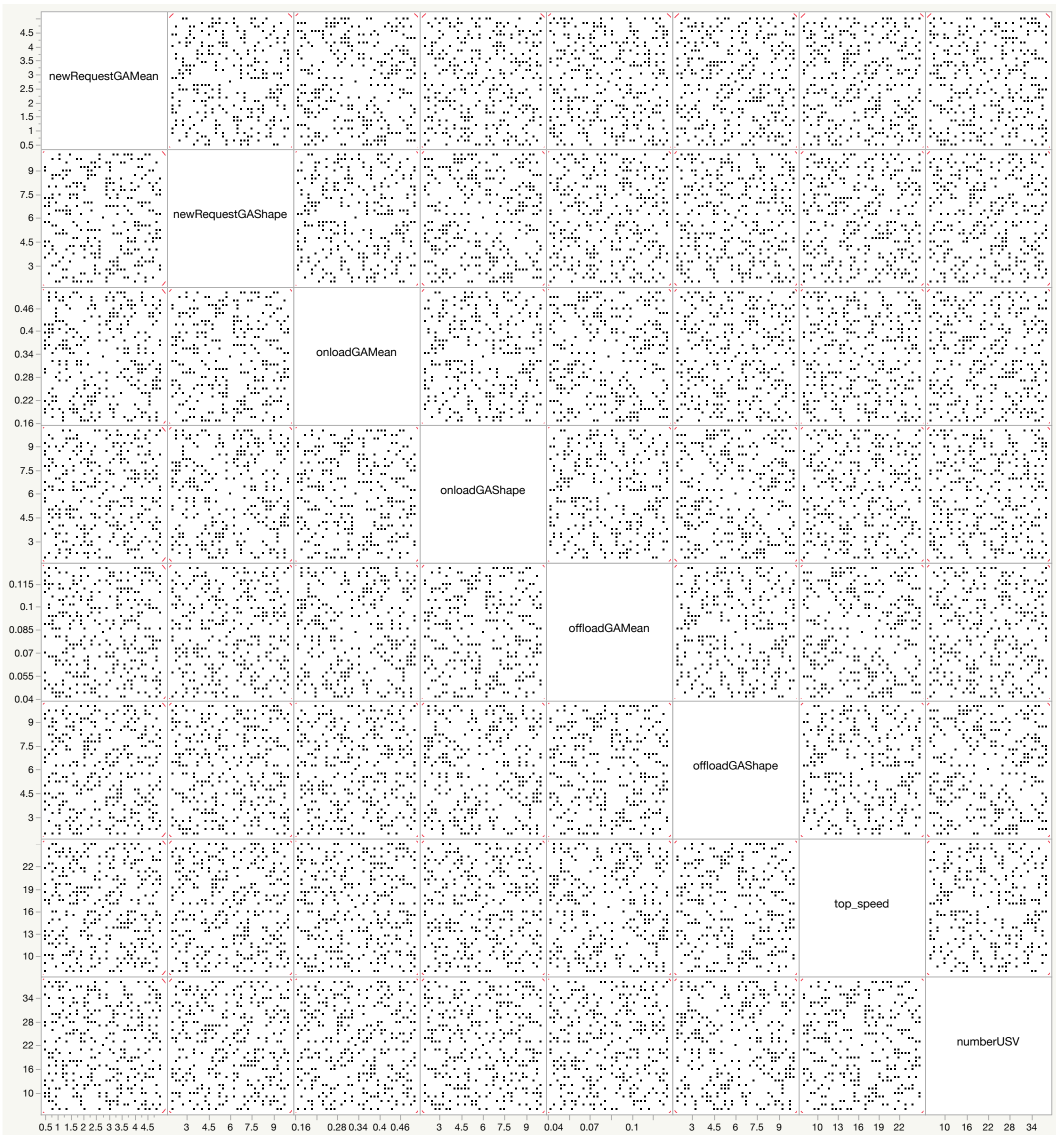


Figure 4.1. Space-Filling Scatterplot Matrix

4.5 ANALYSIS OF RESULTS

After completion of the simulation runs, each of the experiments had model outputs for all four primary MOEs. Several different types of analytical tools can be used to gain a better understanding of these different responses of interests. First, to get an initial general overview of the output data, a histogram and accompanying summary statistics were used to observe the amount of variation over the data set, and from it, decide whether fitting increasingly complex metamodels would yield any interesting insights. In particular, the data sets should be interesting enough where further analysis could determine relationships between the input factors and the MOEs. With the approach of this thesis being a "proof of concept", an analysis template for the two different metamodels was created and can be easily applied again for future studies. From these metamodels, the factors having the most impact on the responses could be determined.

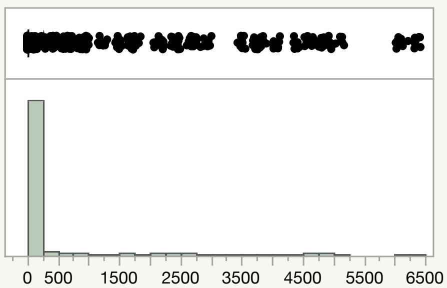
4.5.1 Output Summary Statistics

The following charts depict histograms for each of the model outputs. Graphical representations such as histograms are important in that they help provide extra insight into how the model is processing the data, and verify that the minimum and maximum MOE values make sense. Each of the histograms were produced by using the JMP statistical analysis program. Each histogram include results from all of the design points, so it does not represent an empirical distribution of independent, identically distributed output values.

Figure 4.2 for "Average Number of Requests in Queue" shows a histogram highly skewed to the right. The figure shows all three quartiles to be equal to zero meaning for over 75% of the experiments of the 27-year runs, a USV was always on standby ready to process a request as soon as one entered the system. This means there were probably too many USVs being idle waiting for something to do. Although it may not seem as if having an overabundance of USVs may be an issue, there are opportunity costs which also need to be considered weighing against having too many idle USVs. In essence, these idle USVs could be doing something else instead of waiting around for tasking. For this experiment, the maximum number of requests in queue was 6,422. This number is considerably larger than zero and increases over time, depicting a broken system not able to keep the average number of requests in queue in check.

Figure 4.2 shows the results for individual runs, but the variability across design points is also informative. Summaries over the random replications (not shown) reveal that the design point averages for this MOE range from 0.00 to 6,422, and the design point standard deviations range from 0.00 to 1,187.

AvgNumRequestsInQueue



Quantiles

100.0%	maximum	6422.18183
99.5%		6036.882538
97.5%		4025.2258805
90.0%		292.103977
75.0%	quartile	0
50.0%	median	0
25.0%	quartile	0
10.0%		0
2.5%		0
0.5%		0
0.0%	minimum	0

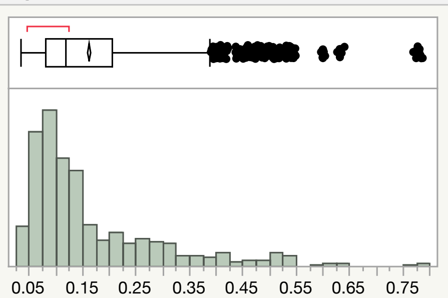
Summary Statistics

Mean	253.0656
Std Dev	934.14528
Std Err Mean	15.722698
Upper 95% Mean	283.8921
Lower 95% Mean	222.23911
N	3530

Figure 4.2. Histogram of Output: Average Number of Requests in Queue

Figure 4.3 for "Average FLC Utilization" shows a histogram skewed to the right. With utilizations, the value is bounded above by 1.0, and ideally should not be too close to that limit. The figure shows the utilization ranges from 0.034 to 0.788 across the individual runs, so none of the runs had utilizations approaching 1.0. Averaging across replications (not shown) reveals that the average utilization ranges from 0.04 to 0.28. This wide range across design points indicates that at least some of the factors influence this MOE.

AvgFLCUtil



Quantiles

100.0%	maximum	0.78778
99.5%		0.63200425
97.5%		0.51618375
90.0%		0.325111
75.0%	quartile	0.2049
50.0%	median	0.11878
25.0%	quartile	0.081715
10.0%		0.058834
2.5%		0.042953
0.5%		0.03617
0.0%	minimum	0.03355

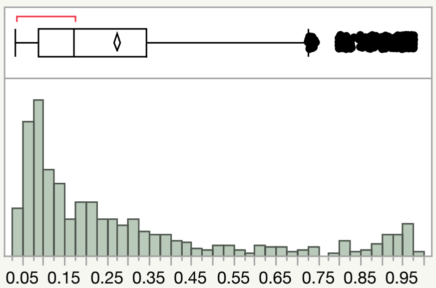
Summary Statistics

Mean	0.1616768
Std Dev	0.1223441
Std Err Mean	0.0020592
Upper 95% Mean	0.1657141
Lower 95% Mean	0.1576395
N	3530

Figure 4.3. Histogram of Output: Average FLC Utilization

Figure 4.4 for "Average USV Utilization" shows a histogram skewed to the right. With utilizations, the value is bounded above by 1.0, and ideally should not be too close to that limit. The figure shows the utilization ranges from 0.034 to 0.977 across the individual runs, so a few of the runs had utilizations approaching 1.0. Averaging across replications (not shown) reveals that the average utilization ranges from 0.01 to 0.53. This wide range across design points indicates that at least some of the factors influence this MOE.

AvgUSVUtil



Quantiles

100.0%	maximum	0.97729
99.5%		0.9725869
97.5%		0.95614625
90.0%		0.803043
75.0%	quartile	0.3427725
50.0%	median	0.171375
25.0%	quartile	0.0872875
10.0%		0.06112
2.5%		0.04496275
0.5%		0.03752925
0.0%	minimum	0.03474

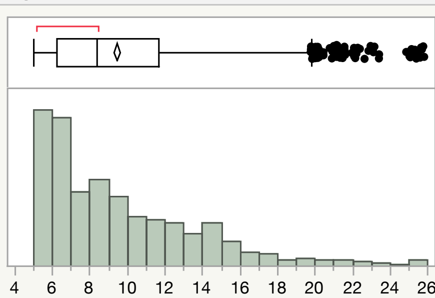
Summary Statistics

Mean	0.273326
Std Dev	0.2624008
Std Err Mean	0.0044165
Upper 95% Mean	0.2819851
Lower 95% Mean	0.2646668
N	3530

Figure 4.4. Histogram of Output: Average USV Utilization

Figure 4.5 for "Average USV Mission Time" shows a histogram skewed to the right. The figure shows the time ranges from 5.03 to 25.84 across the individual runs. Averaging across replications (not shown) reveals that the average USV mission time ranges from 5.51 to 13.39. This wide range across design points indicates that at least some of the factors influence this MOE.

AvgTimeMission



Quantiles

100.0%	maximum	25.84385
99.5%		25.0778626
97.5%		19.77212275
90.0%		14.881075
75.0%	quartile	11.67772
50.0%	median	8.40126
25.0%	quartile	6.276815
10.0%		5.602488
2.5%		5.30630475
0.5%		5.1237292
0.0%	minimum	5.03536

Summary Statistics

Mean	9.4517843
Std Dev	3.9421182
Std Err Mean	0.0663502
Upper 95% Mean	9.581873
Lower 95% Mean	9.3216957
N	3530

Figure 4.5. Histogram of Output: Average USV Mission Times

4.5.2 Partition Tree Metamodeling

Partition trees are useful nonparametric metamodels. The partition tree algorithm recursively splits the data into smaller subsets using simple conditional statements about the factors, such as “number of USVs < 7”. This type of analysis is applied to each of the four outputs studied.

Average Number of Requests in Queue

For the column contributions shown in Figure 4.6, after running the partition tree model analysis, six splits, and pruning, the RSquare value was 0.60. The two main contributions were the `newRequestGAMean` factor which explained approximately 95% of the RSquare while the `numberUSV` factor was responsible for approximately 5% of the RSquare.

RSquare	RMSE	N	Number of Splits	AICc
0.603	588.87062	3530	6	55063.9

Column Contributions

Term	Number of Splits	SS	Portion
newRequestGAMean	4	1754444953	0.9456
numberUSV	1	100745519	0.0543
offloadGAMean	1	218425.449	0.0001
newRequestGAShape	0	0	0.0000
onloadGAMean	0	0	0.0000
onloadGAShape	0	0	0.0000
offloadGAShape	0	0	0.0000
top_speed	0	0	0.0000

Figure 4.6. Column Contributions to the Average Number of Requests in Queue Partition Tree

For the partition tree shown in Figure 4.7 and Figure 4.8, colored circles were used to highlight some key aspects of the partition model. Red circles indicates the means of interest. Blue circles indicate the conditional statements leading up to the first relevant mean of interest. Green circles indicate the conditional statements leading up to the second relevant mean of interest. Purple circles indicate common conditional statements leading up to both means of interests.

First, the partition tree was split until the RSquare value leveled off. Next, any branches yielding high mean values of average number of requests in queue were pruned since

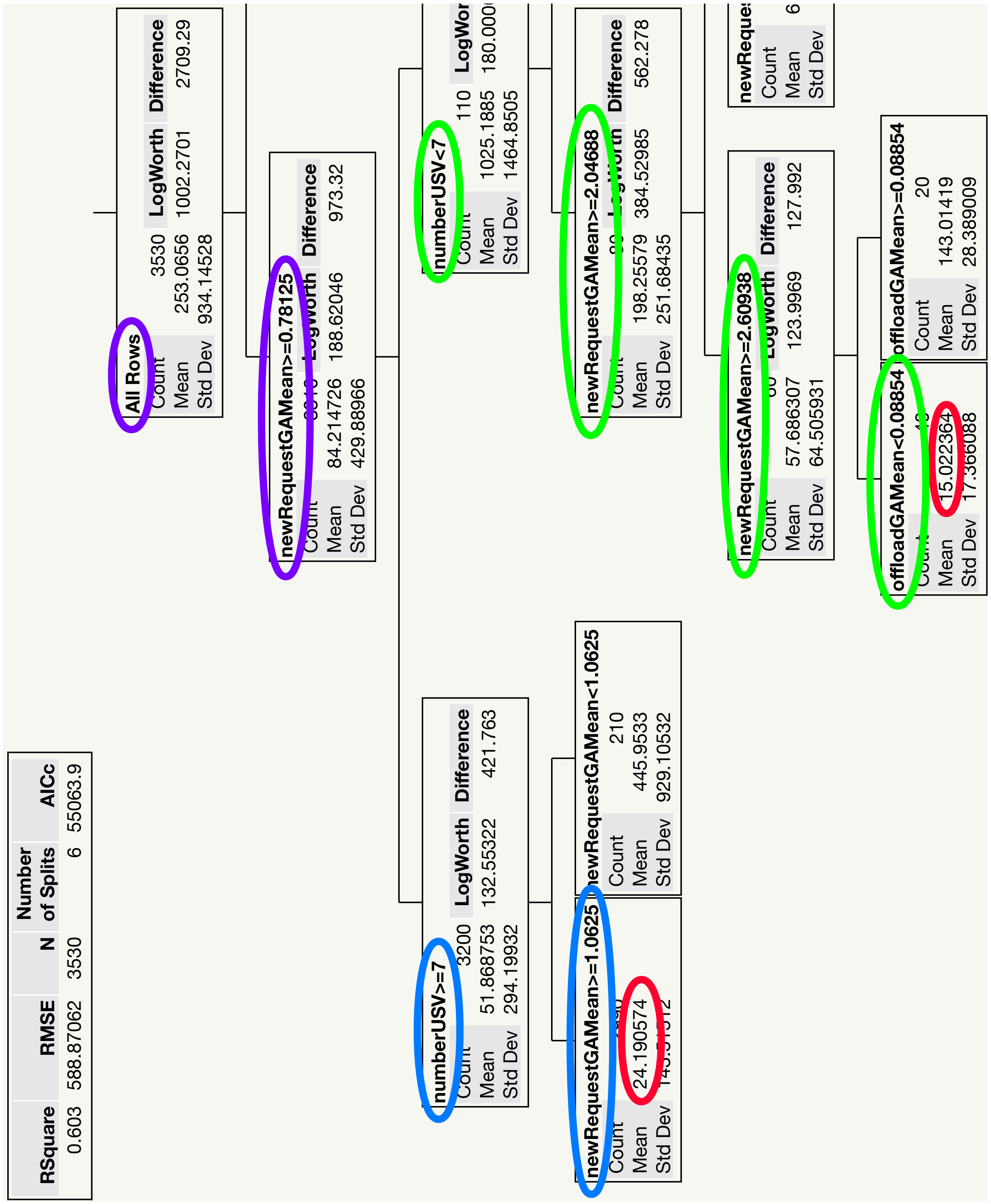


Figure 4.7. Partition Tree: Average Number of Requests in Queue (Page 1 of 2)

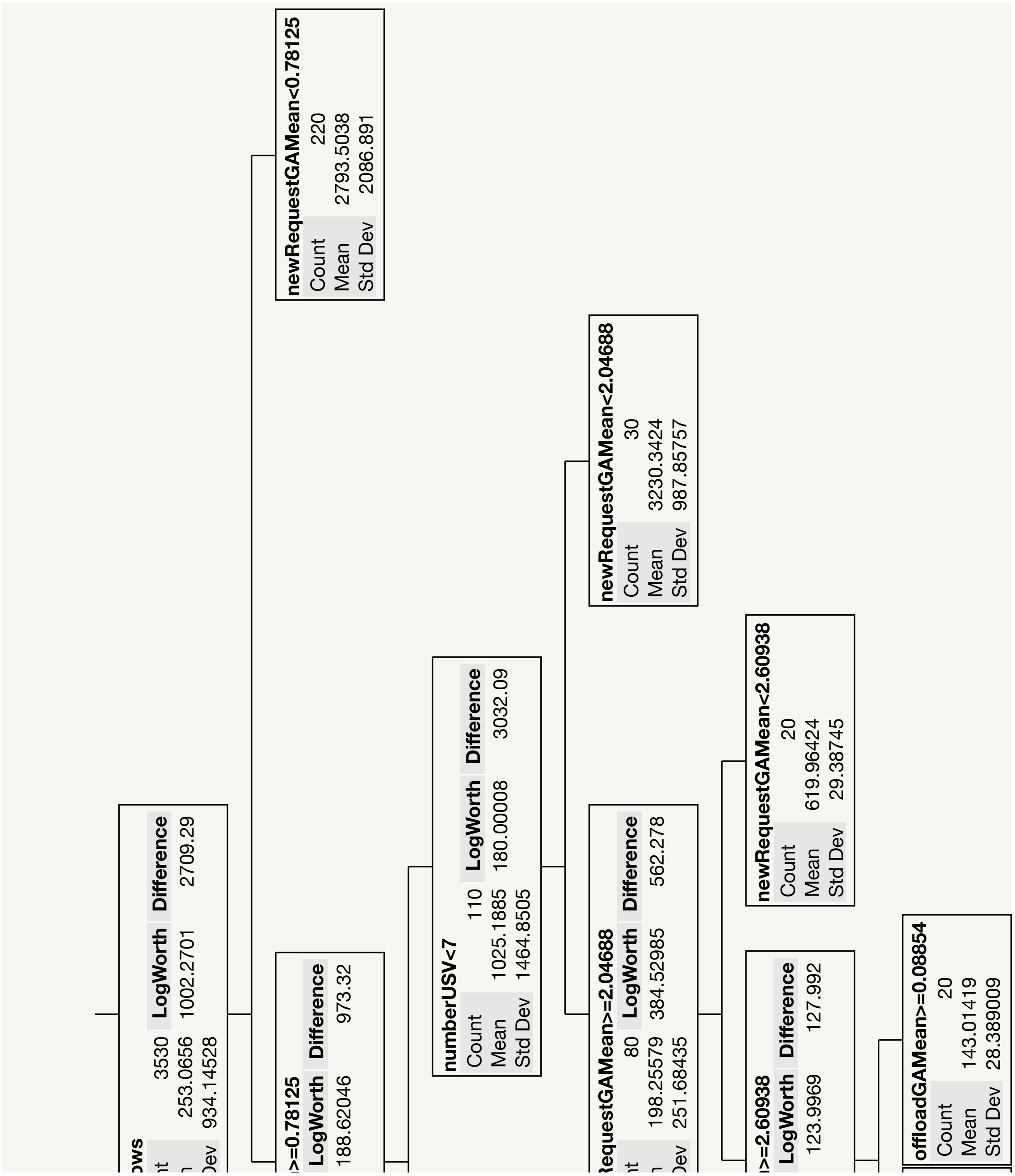


Figure 4.8. Partition Tree: Average Number of Requests in Queue (Page 2 of 2)

those results were undesirable. In this case, the threshold chosen to prune the tree was 50 average requests in queue. Although the RSquare would decrease in value because of the pruning, a simpler tree to explain the various key relationships was more desirable to having simply a high RSquare value with little insight. The two most meaningful and relevant "Average Number of Requests in Queue" leaves were when the mean = 15.02 and the mean = 24.19. When the characteristics were `newRequestGAMean` >= 2.61, `numberUSV` < 7, and `offloadGAMean` < 0.089 the mean was 15.02. When the characteristics were `newRequestGAMean` >= 1.06 and `numberUSV` >= 7, the mean was 24.19.

Of particular interest in this first partition model is if the generation of new requests is too fast (i.e., `newRequestGAMean` < 0.78), then there are no other conditional statements further down that side of the tree that could provide an option sustainable for the system. For this side of the partition tree, the average for the right-most leaf is a backlog of 2,794 requests in queue.

Average FLC Utilization

For the column contributions shown in Figure 4.9, after running the partition tree model analysis, nine splits, and pruning, the RSquare value was around 0.798. The two main contributions were the `newRequestGAMean` factor which explained approximately 81% of the result while the `onloadGAMean` factor was responsible for approximately 19% of the result.

RSquare	RMSE	N	Number of Splits	AICc
0.798	0.0550244	3530	9	-10434

Column Contributions

Term	Number of Splits	SS	Portion
newRequestGAMean	5	34.2653066	0.8132
onloadGAMean	4	7.86929347	0.1868
newRequestGAShape	0	0	0.0000
onloadGAShape	0	0	0.0000
offloadGAMean	0	0	0.0000
offloadGAShape	0	0	0.0000
top_speed	0	0	0.0000
numberUSV	0	0	0.0000

Figure 4.9. Column Contributions to the Average FLC Utilization Partition Tree

For the partition tree shown in Figure 4.10 and Figure 4.11, colored circles were used to highlight some key aspects of the partition model. Red circles indicate the means of interest. Blue circles indicate the conditional statements leading up to the lowest mean of interest. Green circles indicate the conditional statements leading up to the highest mean of interest. Purple circles indicate common conditional statements leading up to both means of interests.

The partition tree was split until the RSquare value leveled off. For the characteristics of `newRequestGAMean` >= 3.17, `numberUSV` >= 20, and `top_speed` >= 14.37, the lowest mean was 0.0596, . For the characteristics of `newRequestGAMean` < 2.19, `numberUSV` < 14, and `top_speed` < 15.44, the highest mean was 0.878.

Average USV Utilization

For the column contributions shown in Figure 4.12, after running the partition tree model analysis, nine splits, and pruning, the RSquare value was approximately 0.791. The three main contributions were the `newRequestGAMean` factor which explained approximately 61% of the result, the `numberUSV` factor which was responsible for approximately 28% of the result, and the `top_speed` factor which accounted for 11% of the result.

For the partition tree shown in Figure 4.13 and Figure 4.14), colored circles were used to highlight some key aspects of the partition model. Red circle indicates the mean of interest. Blue circles indicate the conditional statements leading up to the lowest mean of interest. Green circles indicate the conditional statements leading up to the highest mean of interest. Purple circles indicate common conditional statements leading up to both means of interests.

The partition tree was split until the RSquare value leveled off. The characteristics of `newRequestGAMean` >= 3.03 and `onloadGAMean` < 0.20 had lowest "Average USV Utilization" means at 0.043. The characteristics of `newRequestGAMean` < 1.62 and `onloadGAMean` >= 0.38 had the highest "Average USV Utilization" means at 24.19.

Average USV Mission Time

For the column contributions shown in Figure 4.15, after running the partition tree model analysis, nine splits, and pruning, the RSquare value was around 0.815. The three main

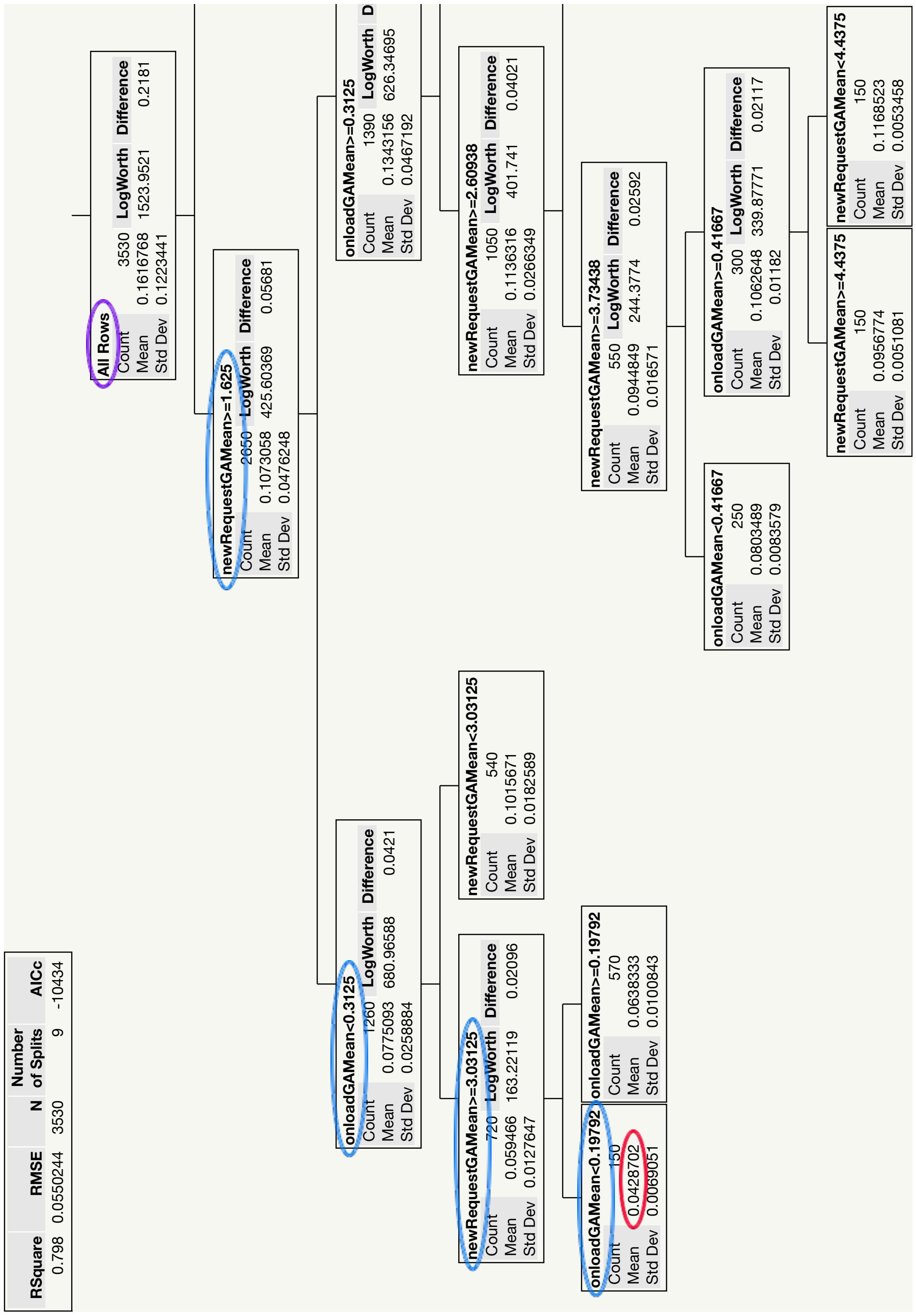


Figure 4.10. Partition Tree: Average FLC Utilization (Page 1 of 2)

contributions were the `top_speed` factor which explained approximately 90% of the result, the `numberUSV` factor which was responsible for approximately 6% of the result, and the `newRequestGAMean` factor which accounted for approximately 4% of the result.

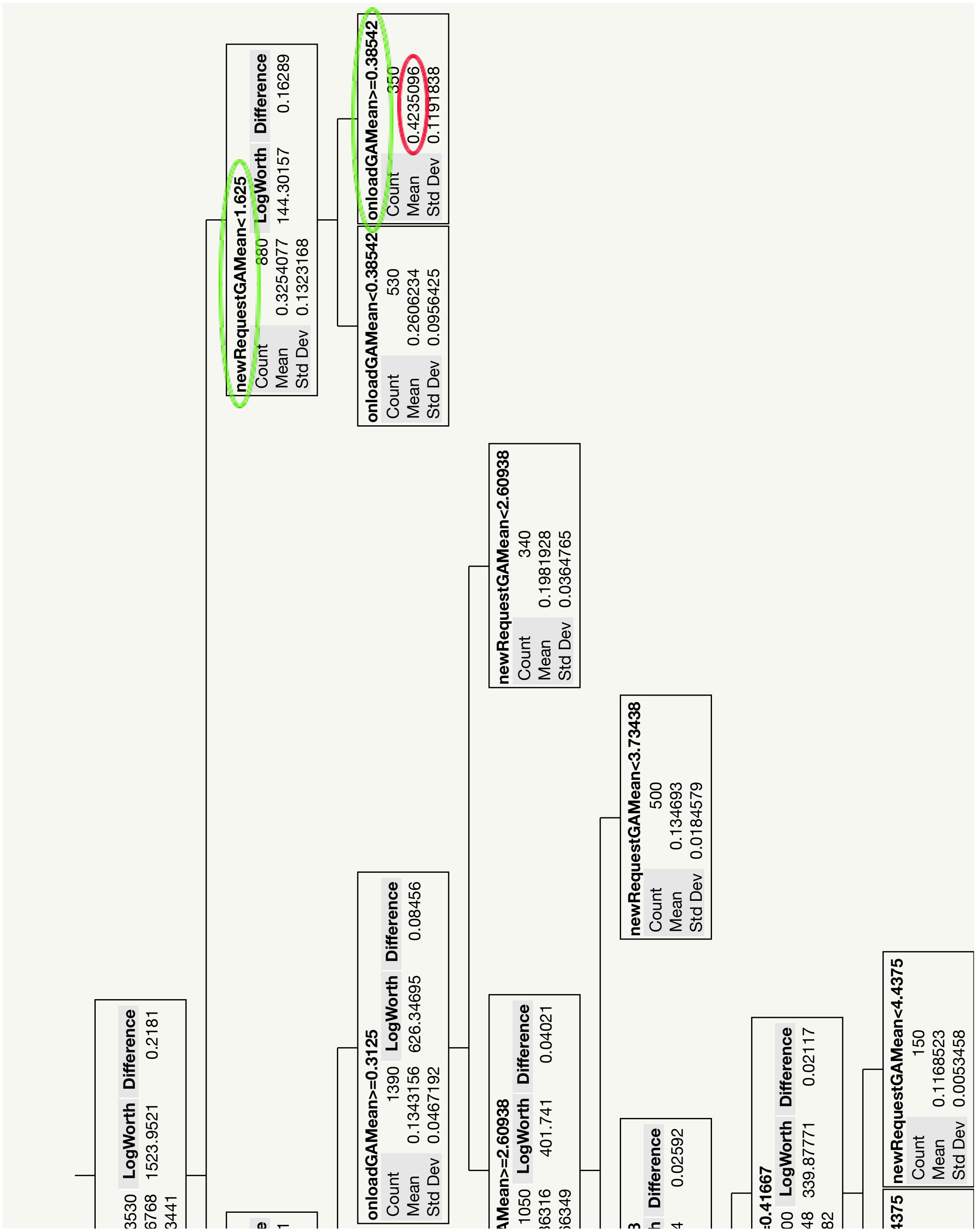


Figure 4.11. Partition Tree: Average FLC Utilization (Page 2 of 2)

RSquare	RMSE	N	Number of Splits	AICc
0.791	0.1198521	3530	12	-4931.9

Column Contributions

Term	Number of Splits	SS	Portion
newRequestGAMean	5	117.096249	0.6090
numberUSV	4	54.6561034	0.2843
top_speed	3	20.5273432	0.1068
newRequestGAShape	0	0	0.0000
onloadGAMean	0	0	0.0000
onloadGAShape	0	0	0.0000
offloadGAMean	0	0	0.0000
offloadGAShape	0	0	0.0000

Figure 4.12. Column Contributions to the Average USV Utilization Partition Tree

For the partition tree shown in Figure 4.16 and Figure 4.17, colored circles were used to highlight some key aspects of the partition model. Red circles indicate the means on leaves of interest. Blue circles indicate the conditional statements leading up to the shortest duration mean of interest. Green circles indicate the conditional statements leading up to the longest mean of interest. Purple circles indicate common conditional statements leading up to both means of interests.

The partition tree was split until the RSquare value leveled off. For characteristics of `newRequestGAMean` >= 0.78 and `top_speed` >= 24.47, the shortest mean was 5.37, the most restrictive . For the characteristic of `top_speed` < 9.59, the longest mean was at 16.43.

4.5.3 Regression Metamodel

Analyses other than partition trees are also possible. We illustrate this for the "Average USV Mission Time" MOE. We found `newRequestGAMean`, `top_speed`, and `numberUSV` to be the most significant factors after applying reciprocal transformations to all three as can be seen in Figure 4.18.

The results from this particular model are significant as can be seen from Figure 4.18. The regression seems to be well-fit with an R-Square value of 0.93 and a p-value < 0.0001.

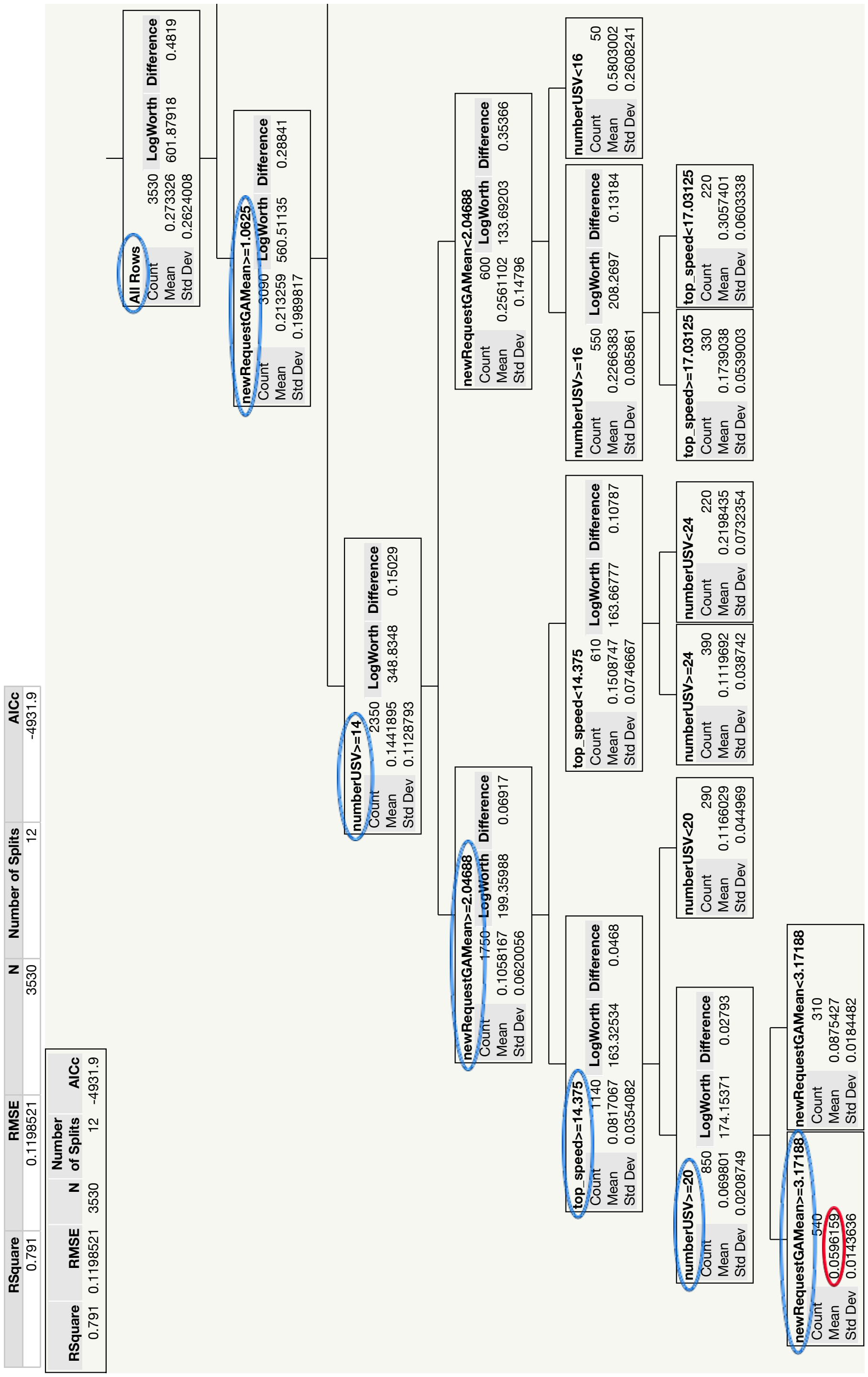


Figure 4.13. Partition Tree: Average USV Utilization (Part 1 of 2)

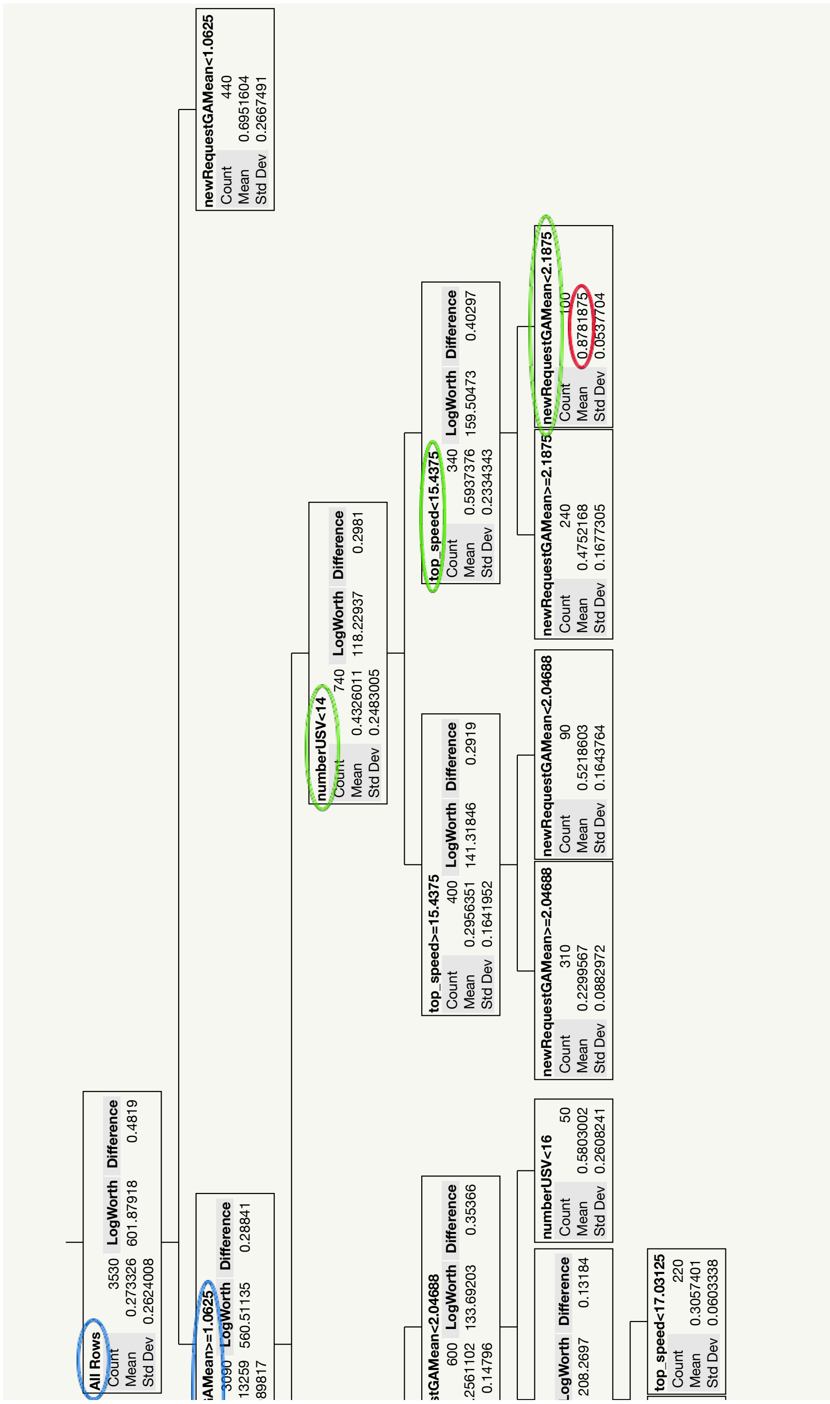


Figure 4.14. Partition Tree: Average USV Utilization (Part 2 of 2)

RSquare	RMSE	N	Number of Splits	AICc
0.815	1.6942854	3530	14	13772.3

Column Contributions

Term	Number of Splits	SS	Portion
top_speed	6	40211.3918	0.8994
numberUSV	3	2769.49592	0.0619
newRequestGAMean	4	1702.2748	0.0381
newRequestGAShape	1	25.3137825	0.0006
offloadGAShape	0	0	0.0000
onloadGAMean	0	0	0.0000
onloadGAShape	0	0	0.0000
offloadGAMean	0	0	0.0000

Figure 4.15. Column Contributions to the Average USV Mission Time Partition Tree

Figure 4.19 graphically summarizes the model terms with strong impacts, yielding some interesting findings.

1. At both low and high new request generation means, the faster the USVs travelled, the shorter the average USV mission times.
2. At low new request generation means, increasing the number of USVs contributed to a decrease in the average USV mission times while at high new request generation means, increasing the number of USVs had little impact on average USV mission times.
3. At high top speeds, the new request generation means and the number of USVs had little impact on the average USV mission times. The missions were always completed in approximately five days.
4. However, at low speeds, both higher new request generation means and lower numbers of USVs can contribute to longer average USV mission times.
5. At both low and high quantities of USVs, the USV speed is inversely proportional to the average USV mission times.
6. Finally, high new request generation means can elongate average USV mission times when the number of USVs is low. However, when the number of USVs is high, new request generation means seem to have little impact on the average USV mission times.

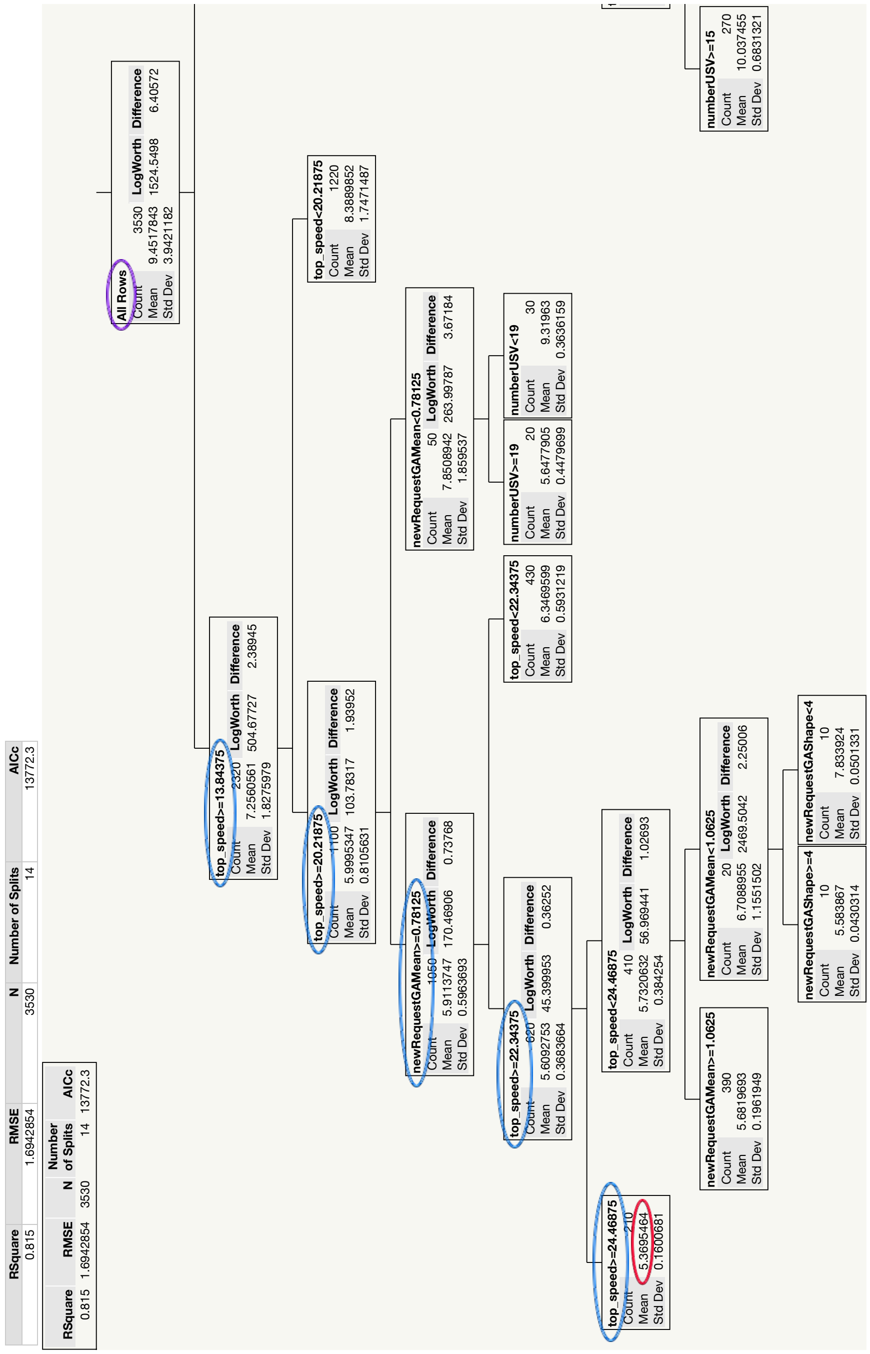


Figure 4.16. Partition Tree: Average USV Mission Time (Part 1 of 2)

These findings are consistent with the trends and patterns seen in the "Average USV Mission Time" partition tree model.

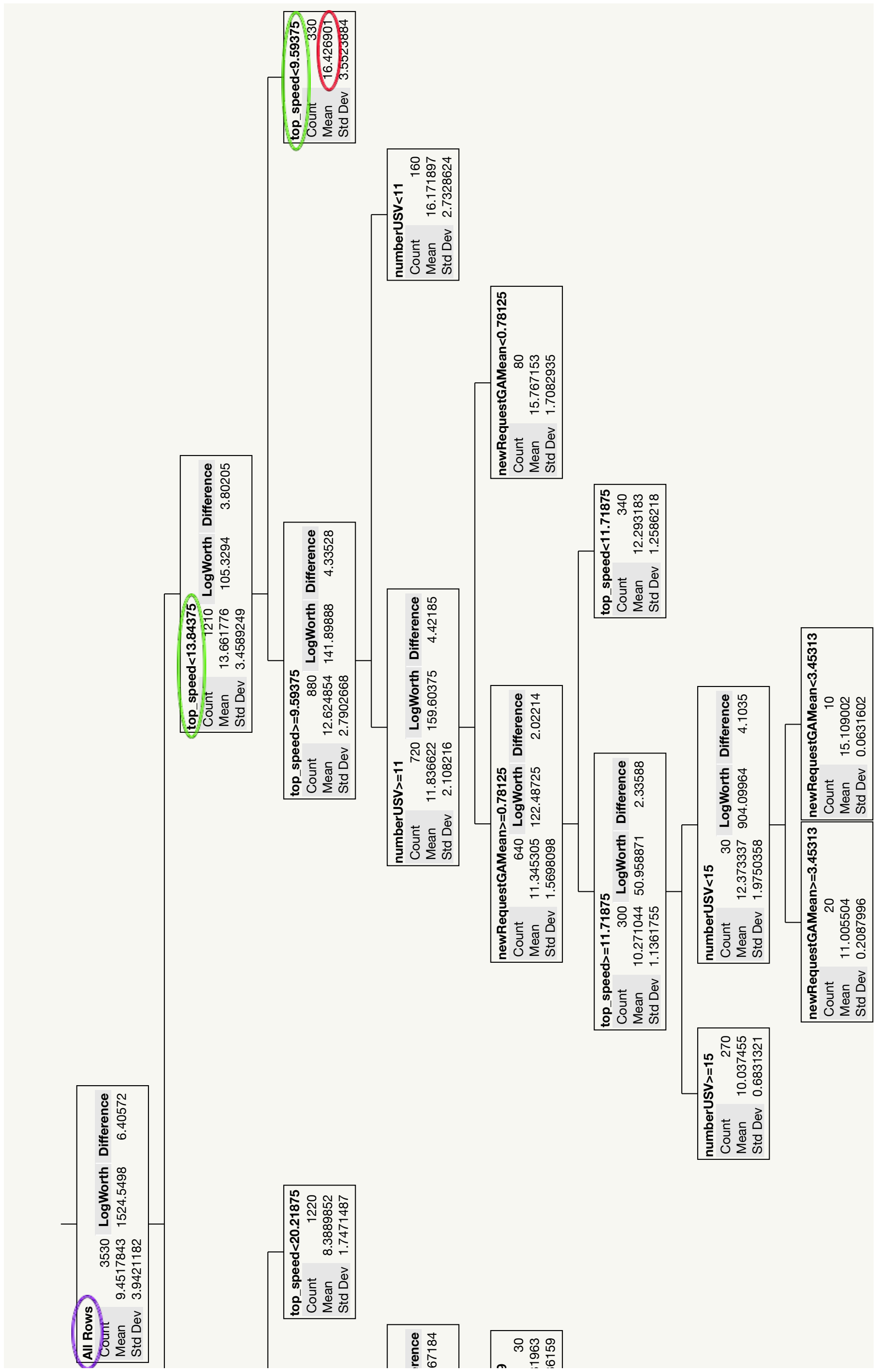
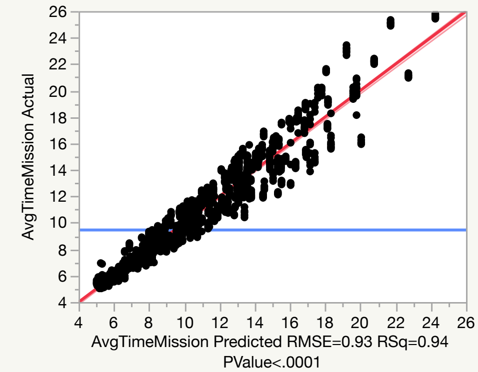


Figure 4.17. Partition Tree: Average USV Mission Time (Part 2 of 2)

Actual by Predicted Plot



Effect Summary

Source	LogWorth	PValue
Reciprocal(top_speed)*Reciprocal(numberUSV)	362.977	0.00000
Reciprocal(newRequestGAMean)*Reciprocal(top_speed)	334.193	0.00000
Reciprocal(numberUSV)	214.591	0.00000 ^
Reciprocal(newRequestGAMean)*Reciprocal(numberUSV)	210.704	0.00000
Reciprocal(newRequestGAMean)	193.891	0.00000 ^
Reciprocal(top_speed)	173.847	0.00000 ^

Lack Of Fit

Source	DF	Sum of Squares	Mean Square	F Ratio
Lack Of Fit	338	2688.0254	7.95274	70.5003
Pure Error	3185	359.2816	0.11280	Prob > F
Total Error	3523	3047.3071		<.0001*
				Max RSq
				0.9934

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	4.1786405	0.129415	32.29	<.0001*
Reciprocal(newRequestGAMean)	-5.195487	0.163716	-31.73	<.0001*
Reciprocal(top_speed)	50.346912	1.687577	29.83	<.0001*
Reciprocal(numberUSV)	-55.96612	1.663991	-33.63	<.0001*
Reciprocal(newRequestGAMean)*Reciprocal(top_speed)	74.879455	1.709657	43.80	<.0001*
Reciprocal(newRequestGAMean)*Reciprocal(numberUSV)	39.415407	1.184305	33.28	<.0001*
Reciprocal(top_speed)*Reciprocal(numberUSV)	994.90435	21.57347	46.12	<.0001*

Prediction Profiler

Figure 4.18. Regression Summary for Average USV Mission Time.

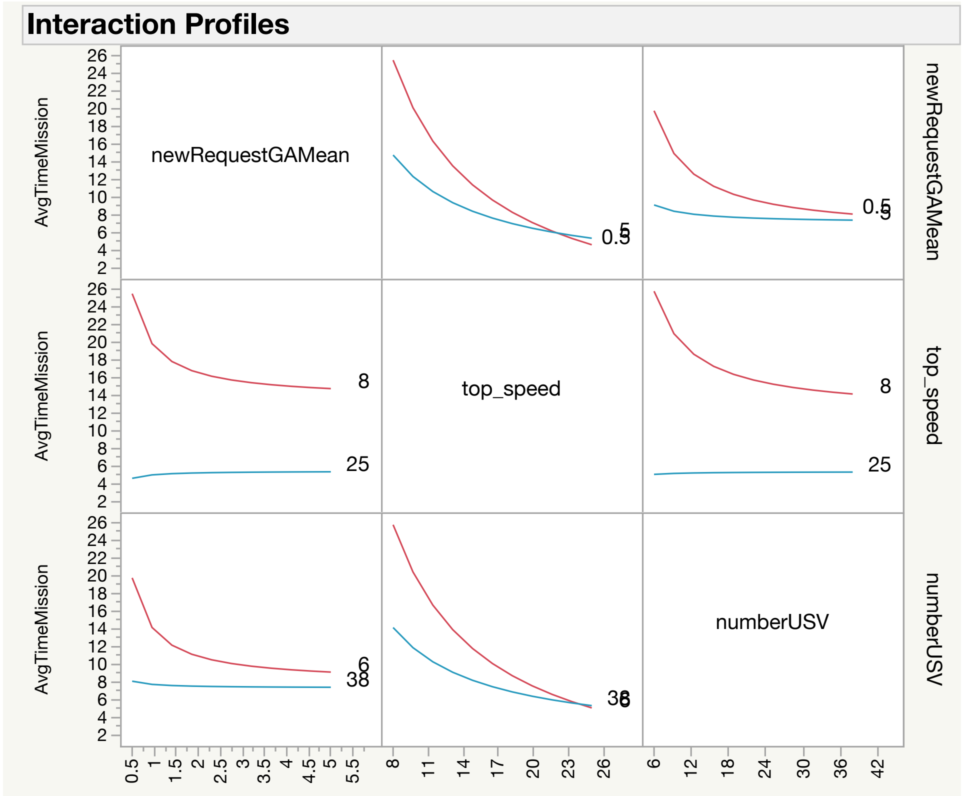


Figure 4.19. Interaction Profile

4.6 SUMMARY OF FINDINGS

Table 4.3 summarizes the key column contributions for each of the four primary MOEs and also shows how these factors directionally affect the model outputs. The "+" sign denotes a positive contribution to improving the measure of effectiveness while the "–" sign indicates the opposite. A higher value for a utilization MOE can represent either an improved or degraded condition depending on how close the rates are to 100%. For our experiment, given that many design points led to unstable queues, we decided that increasing average USV utilization is bad but increasing FLC utilization is good. The four key factors influencing the MOEs are mean new requests, mean onload, USV top speed, and number of USVs.

For the "Average Number of Requests in Queue," both the mean new requests and number of USVs decreased the MOE as their values increased. For the "Average FLC Utilization," a decrease in mean new requests and an increase in mean onload time increased the MOE. For the "Average USV Utilization," a decrease in mean new requests, USV top speed, and number of USVs increased the MOE. And finally, for the "Average USV Mission Time," increases in mean new requests, USV top speed, and number of USVs all contributed to a lower MOE.

With the exception of mean onload, each of these factors have effects on two or more MOEs, as shown by a mix of "+"s and "–"s for each factor, respectively. Mean onload time is different from the others in that an increase in its value is important for a single MOE, the "Average FLC Utilization." Tables such as this one can help decision makers evaluate tradeoffs among MOEs if, for example, increasing one factor improves one MOE but worsens another.

Table 4.3. MOEs with Directional Factors

	FACTOR NAMES	AVG NUMBER OF REQUESTS IN QUEUE	AVG FLC UTILIZATION	AVG USV UTILIZATION	AVG MISSION TIME
MEAN NEW REQUESTS	newRequestGAMean	+	–	–	+
SHAPE OF NEW REQUESTS DISTRIBUTION	newRequestGAShape				
MEAN ONLOAD	onloadGAMean		+		
SHAPE OF ONLOAD DISTRIBUTION	onloadGAShape				
MEAN OFFLOAD	offloadGAMean				
SHAPE OF OFFLOAD DISTRIBUTION	offloadGAShape				
USV TOP SPEED	top_speed			–	+
NUMBER OF USVs	numberUSV	+		–	+

Some other interesting results can be observed from these analyses. The variations in the shapes of the gamma distributions, as well as the offload service time mean, did not affect any of the four MOEs substantially. With mean onload service times ranging only from four to twelve hours and mean offload services times ranging from only one to three hours, their variations were not significant to the model. Average USV mission times were 9.45 ± 3.94 days, so the few hours made little difference in the grand scheme. Of course, all findings from the analyses are limited to the range of factor values investigated.

4.6.1 ADDITIONAL MEASURES OF EFFECTIVENESS

While running the analysis for the first four MOEs, a few bugs were found in the program. The "Average USV Mission Time" analyzed earlier in this chapter was initially called "Average Total Time" even though it initially took into consideration only the time after a USV began to onload at an FLC. Consequently, the bugs were corrected so that the "Average USV Mission Time" began when a USV was tasked to process a request from queue, and two other MOEs were added that are of interest to the requesting entities.

The fifth MOE is the "Average Request Turnaround Time." The revised code defines the start time to be when the customer actually places the order. By doing so, this new metric will calculate any delays in scheduling due to requests in queue waiting to be processed because of lack of USV availability.

The sixth MOE is the "Average Time for Unfilled Requests." The collected data can be viewed as arising from two groups. The first is where most requests have been processed in a timely manner so the queue length is relatively stable. The second is where requests have been stuck in the queue and never been processed. An example of this latter condition occurs when there are simply not enough USVs to process the amount of requests entering the system. Without enough USVs to process all requests, the priority queue would continue to build and the number of unfilled requests would grow in an unbounded fashion. This might not affect all types of requests in the same way. For example, numerous high priority requests might enter the system and continuously supersede those requests with lower priorities. Were this to occur repeatedly, the lower priority requests would never get processed.

The simulation model does not make use of specified random number seeds, so we were unable to leverage common random numbers to compare the two models. Instead, we made 10 replications of 33 design points for the revised model. Figure 4.20 provides side-by-side box and whisker plots for the average amount of time USVs spend on-mission after being irrevocably committed to fulfilling a request, or “Average USV Mission Time.” The results from the revised model and the original model are very similar, with the former having a very slightly lower median delivery time but outliers which are higher than those of the latter. The clusters of outliers correspond to design points with very large queues, and might merit more investigation in a future study. However, a t-test shows no significant difference in the means between the original and revised model (p-value > 0.05) and the partition tree on the revised model (not shown) included the same set of important factors. Most interesting is the difference between the MOEs from the USV and customer perspectives. "Average USV Mission Time" ranges from 5.2–24.6 days with an average of 9.3 days, while "Average Completed Request Turnaround Time" ranges from 5.2–249.2 days with an average of 24.6 days.

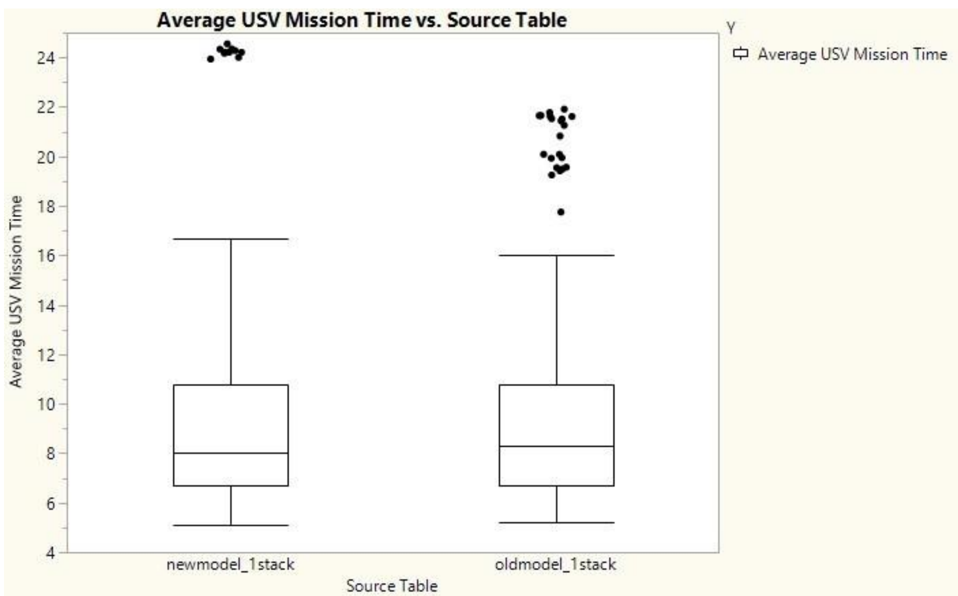


Figure 4.20. Box Plot for "Average USV Mission Time" from Original and Revised Simulation Model.

We do not do a complete analysis of "Average Unfilled Request Time " but provide some initial insights by comparing it to "Average Number Requests in Queue" Figure 4.21 plots the average time long unfilled requests have resided in the queue at the time when the simulation is halted versus how many such requests were in the queue. Clusters are clearly visible. Note that the halt time is 10,000 days of operations, so having an average near 5,000 is a clear indication that the system is operating in an unsustainable configuration. In queueing terms, the requests are arriving faster than they can be processed, and the request queue is growing in an unbounded fashion. Because of the clear relationship between these two MOEs, the partition trees are very similar.

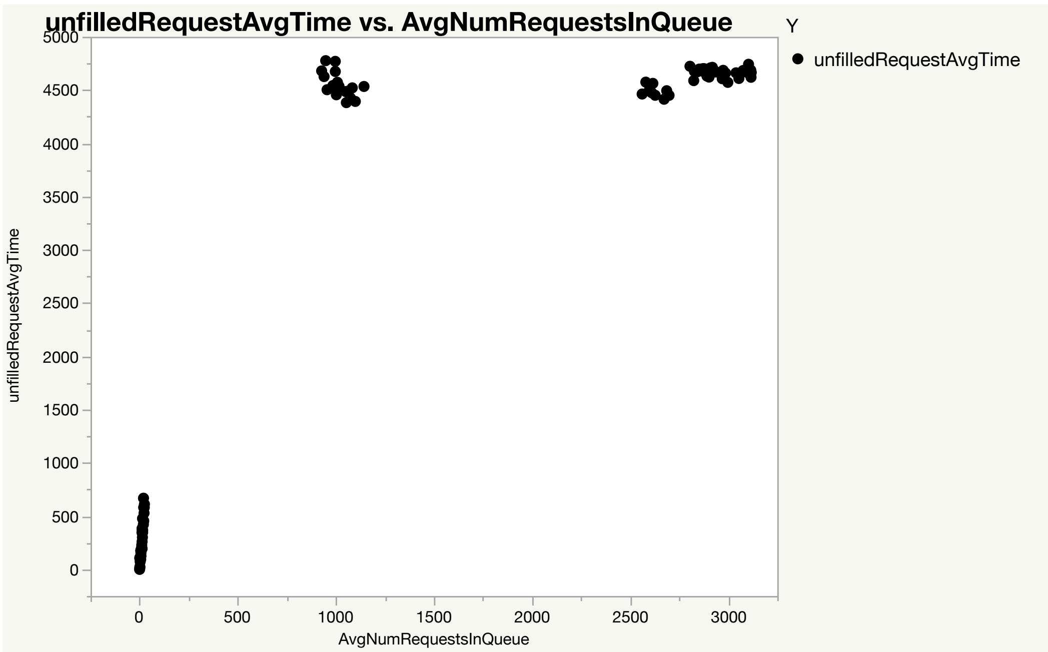


Figure 4.21. "Average Time for Unfilled Requests" versus "Average Number Requests in Queue" for Revised Simulation Model.

We subsequently reran the full experiment using the revised model. Although some of the partition tree splits occurred at slightly different places, the important factors and other qualitative insights did not change.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5: CONCLUSIONS AND FUTURE WORK

5.1 CONCLUSIONS

The purpose of the Navy Supply Corps is to manage the logistical pipeline so that resources are delivered to the warfighter as they are required. Naval units are easily able to replenish their stores while they are in port. Difficulties arise when the naval units get underway to conduct missions and training exercises. With credible threats from countries such as China, Russia, and North Korea, the maritime environment continues to be a hazardous and difficult place to conduct operations. Maintaining high readiness levels is a growing concern.

The use of unmanned systems introduces a new naval system class with many beneficial characteristics, including: autonomous control in which minimal human supervision is required, reliability demonstrated by long range and duration of ability to travel, safety due to programmed compliance with regulations for avoiding collisions at sea, and spacious cargo transportability.

This thesis uses a model-based approach to address these concerns, and a prototype discrete-event simulation model is built to represent naval logistics operations. USVs are used to fulfill requests from requesting entities by transporting materiel from one of several Fleet Logistics Centers to ship-designated rendezvous locations.

Six primary measures of effectiveness (MOEs) are considered: the "Average Number Requests in Queue," "Average FLC Utilization," "Average USV Utilization," "Average USV Mission Time," "Average Request Turnaround," and "Average Time of Unfilled Request." Eight factors are varied in a designed experiment to determine how strongly they influence the MOEs. The primary findings follow. For the "Average Number Requests in Queue," the new request generation mean contributes the most to its behavior. As the new request generation mean increases, the "Average Number Requests in Queue" increased as well. For the "Average FLC Utilization," the new request generation mean and onload service mean is able to explain the most about its behavior. For the "Average USV Utilization," the new request generation mean, the number of USVs, and the USV top speed contribute the most to its behavior. For the "Average Mission Time," the USV's top speed, the number of USVs, and the new request generation means explain the most about its behavior. For the "Average Request Turnaround," results from both revised and original models indicate that the same factors that affect "Average USV Mission Time" explain the most about its behavior. The "Average Time of Unfilled Request" is, unsurprisingly, strongly correlated with "Average Number Requests in Queue" showing unstable queuing conditions when exceeding an average of 1,000 in queue. There is also a strong direct proportion between these two MOEs when the "Average Number Requests in Queue" is close to zero.

Overall, the new request generation mean is a dominant force in straining the model. If too many requests are generated within a short period of time, the system does not handle this workload well. At their highest speeds, the USVs handle the majority of the workload generated by requesting entities. However, if the requests flood the system at an extremely high rate, the USVs will not have the capacity to handle all of the requests. Building more USVs alleviate the workload requests. At lower numbers of USVs, the new request generation means will have a more significant impact on the average USV mission times.

With this concept of operations for USVs, there are trade-offs that the decision maker must consider between maintaining flexibility in deliveries and the necessity of leveraging larger vessels. For example, building more USVs results in more units being able to handle better the generated requests from requesting entities. But, at some point, there is a level of saturation where the addition of another USV does not improve delivery times to the customer. Also, the small USVs are restricted based on their capacity and must leave those materiel it cannot handle for the larger, traditional MSC vessels.

Implications

Some important considerations can be derived from these primary findings. The generation of new requests could have significant leverage in how the model operates. With low levels of request, the user can mitigate increased queue lengths by increasing USV speeds, increasing the number of USVs, or decreasing service times. However, as the rate of generation of new requests increases, the system begins to lose its robustness and will eventually not be able to handle the fast influx of requests. Future work must take this into consideration to decide what methods would be the best to handle this type of situation.

Decision-makers sometimes wonder about the implications of how selecting different shapes of distributions could affect the outputs for the models. Sensitivity analysis using partition tree modeling shows the shape of the gamma distribution to have little impact. From this, future researchers will not have to expend energies in trying to discern what specific shape distribution to apply to their work.

After studying the directional impacts of the factors on the MOEs, the decision-maker must understand the trade-offs involved in either increasing or decreasing any of the input factors. Based on the analysis performed, there was no single factor that could be adjusted unidirectionally to beneficially affect all MOEs. The decision-maker could make a more informed decision by understanding the level of trade-offs the changes could affect the outputs.

Prioritization of the queue brings to light another interesting consequence. The system was designed to prioritize high priority requests over lower priority requests. For example, even if the queue had built up a queue length of 30 low priority requests, if a single high priority request were to enter the system, the USV would be compelled to process that request over others. Although initially this will satisfy the requesting entity with the higher priority request, this prioritization causes other lower priority requests to sit in the queue for potentially a very long time. As a result, the requesting entity may have to upgrade what used to be a low priority to a higher priority request, duplicate orders may be placed into the system, or illegitimately, some entities may begin to order everything as high priority since they have lost faith in the system. As mentioned in the previous implication, this is not an ideal situation since flooding the system will serve to only stress the already unstable system leading to an eventual breakdown of the entire USV support system.

The traditional thinking for supplying afloat units consists of two options. First, ships can pull in to port and onload materiel from Fleet Logistics Centers or other onshore fixed structures. Alternatively, afloat units can receive materiel from rigid, infrequent, scheduled deliveries via the Combat Logistics Force. During peace times, this type of thinking may be enough to sustain the force for normal operations. However, if a more versatile, robust, and faster replenishment of supplies is needed during more unstable and unpredictable times, the traditional method may not be enough to maintain a functioning force in adversarial situations. Instead, a system capable of prioritizing and modularizing requirements could be the warfighting tool used towards fulfilling the requirements for higher readiness levels.

The basic concept of support using unmanned surface vehicles has been developed with this research. Based on the frequency with which requests are placed into the system by requesting entities, suitable levels for the number of USVs, speed of the USVs, the rate of onload of materiel, and the rate of offload to requesting unit values can be chosen to achieve desired performance goals. These can assist the Navy in determining issues such as what rate of demand the USV system can sustain and how the logistics support operations should be structured. With the significantly lower costs for building, operating, and maintaining these USVs, the Navy should consider augmenting the Military Sealift Command fleet with these units.

5.2 RECOMMENDATIONS FOR FUTURE WORK

The USV logistics model can be expanded in several ways to provide further insights. Several aspects of the model could be enhanced by adding more detail. For example, Fleet Logistics Centers were treated as mostly a static object in this research. As the algorithm calculated which combination of FLC and USV to use, part stock number and quantities were verified for availability. However, once the materiel was loaded onto the USV, the amount available at the FLC was not decremented. The FLC object could further be refined by applying standard inventory reorder policies, modeling the policy-driven supply levels of materiel, as well as appropriately decrementing the FLC inventory in response to delivery to USVs. These enhancements would add a more realistic dimension to the modeling of the FLC inventory.

The USV object could also be further refined for the scenarios considered in this research. Materiel loaded onto the unmanned surface vehicle is containerized into a single 20-feet ISO container strapped above decks. However, the USV has capacity within itself to house more materiel when necessary. Further research could be conducted to modularize the cargo loaded in more detail inside the USV since the dimensions of the cargo will play a larger role due to restrictions in space and dimension.

Request priorities were completely randomized with a uniform, even distribution. For example, requests of priority two have the same chance of appearing as requests of priority ten. In reality, however, there are comparatively higher numbers of lower priority requests instead of high priority requests. Additional future research in this area could study the approximate historical distribution of CASREP occurrences and seek to emulate that distribution with the model to better refine the new request generation event.

The geographical construct may also be improved. For the model to be able to select a location to deliver any requested materiel to the requesting entity, a 2-D sandbox to represent the Pacific region. The four corners of the region are located at Hokkaido, Japan, Washington State, Northeast Indonesia, and Guatemala. An obvious extension given the size of the area is to represent the sandbox as the surface of a sphere. For more modeling flexibility, other coordinates of the region could be modified, but by using USVs we are no longer constrained to using historical rendezvous points. For example, delivery locations could be determined or adjusted dynamically.

In addition to delivering materiel to vessels out in the ocean, some particular coastal on land operations may require assistance with supplies and materiel. For example, a Special Operations Force (SOF) could possibly require a supply boost if operations are prolonged longer than anticipated. In such cases, USVs should have the option to approach land within safe limits and provide the equipment and support required. For this reason, there must be some calculations where the scenario simulates rendezvous locations within close range of land. Expanding the model to include UAVs may better reflect new concepts of operations.

Opportunity costs associated with the decision to select a specific USV to go to a specific FLC could potentially have consequences. By making a greedy selection, the potential benefits of alternative choices may be missed. For example, even if a USV can reach a particular FLC most quickly, using that particular USV may negatively impact future requests. The delivery delays due to the USV transiting to the FLC could lead to other detrimental consequences.

The current design of the model has requests prioritized by their importance, with ties broken on a FIFO basis. However, once a USV takes on a tasking, it becomes rigid in completing its mission. Another enhancement would allow USVs to be re-tasked, and the lower priority request to be returned to the request queue, if a higher priority request is received before loading commences. This would improve the prioritization of tasking.

Plans may not always go smoothly, and many different types of events could unpredictably affect the course of a mission for a USV. Weather could jeopardize sea states and make it harder for USVs to navigate. Required emergency maintenance could unpredictably appear causing unexpected delays. Although designed to avoid collisions, the USV could encounter a situation that is unavoidable, resulting in a ship-to-ship mishap. These unexpected situations could be represented with randomness and the resulting consequences assessed. Piracy and combat are other potential sources of loss or delay, and should be considered.

USVs must be balanced with currently available assets in order to leverage their capabilities, and cannot be thought of as a total replacement of current systems. This research is primarily focused on the concept of using USVs as delivery agents for requesting units at sea. There are situations where USVs may not able to take on specific cargo due to weight or capacity restrictions, cargo type restrictions, or classification of cargo. As such, an oiler or container vessel may be more suited to completing high-capacity tasks. In other situations involving time-critical parts, the use of UAVs may be more appropriate. With the object-oriented architecture of this model, incorporation of unmanned aerial vehicle objects can be easily implemented. Based on the type of operation and anticipated modes of use, the types of vehicles used can be further refined, so appropriate mixes of different types of logistical assets can be chosen to robustly support fleet operations.

Adding a user-friendly interface for the managers would be an important step toward shifting paradigms from traditional thinking to more innovative approaches. By making the model accessible, more interest would be generated for the use of USVs as well as the desire to learn how to improve the process. Depending on the number of demands gauged by the influx of requests, the user could adjust the input parameters such as higher or lower speeds, the number of USVs in service, and how to staff FLCs for faster or slower service times.

As with any designs and concepts of operations, future designers of this model must be careful in being vigilant on whether requesting entities begin to abuse the system. Placing high priority orders for non-critical parts would be less than ideal and eventually cause abnormal responses from consumers such as ordering all of their parts at highest priority levels possible. By staying ahead of such an eventuality, a well-designed model could greatly benefit military readiness levels and provide outstanding customer service logistics.

5.3 FINAL REMARKS

Simulation modeling analysis was used to gain insight about the concept of operations for unmanned surface vehicles. The logistical application of the concept of USVs is a novel idea and has not yet been put in to practice for the U.S. naval fleet. Having built a solid model with influential attributes identified, the decision maker now has the tool to decide how to keep the Navy's logistical readiness high and flexible in many different situations. The model's versatile architecture is also robust enough to keep pace with continuing technological developments, such as UAVs, and analyze how these different types of assets could potentially affect the system.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX: USV_MODEL

A.1 MODEL SOURCE CODE: MAIN.RB

```
# !/usr/bin/env ruby -w

# USV_model
# @Author:: Tai-shan Lin
# @Copyright:: Copyright (c) Tai-shan Lin
# @License:: MIT

# Make sure you install simplekit before trying to run this!
# Run 'gem install simplekit' (Win) or 'sudo gem install simplekit' (Mac)
require 'simplekit'
require 'random_variates'
require 'quickstats'

require_relative 'location'
require_relative 'flc'
require_relative 'usv'
require_relative 'request'
require_relative 'part'
require_relative 'timevaryingstats'

class USV_model
  include SimpleKit
  # attr_reader :FLCs, :USV_available, :USV_loading, :USV_onMission,
  #                :parts, :requests
  def initialize(newRequestGAMean:, newRequestGAShape:,
                onloadGAMean:, onloadGAShape:,
                  offloadGAMean:, offloadGAShape:,
                haltTime:, numberUSV:, top_speed:, inputset:,
    debugmode: false)
    @inputset = inputset
    @debugmode = debugmode
    @FLCs = {}
    @USV_available = {}
    # @USV_loading = {}
    # @USV_onMission = {}
    @parts = {}
    @requests = PriorityQueue.new
    @numberUSV = numberUSV
    @top_speed = top_speed

    @avgNumRequestsInQueue = TimeVaryingStats.new
    @avgFLCUtil = TimeVaryingStats.new
```

```
    @avgUSVUtil = TimeVaryingStats.new
    @avgTimeMission = QuickStats.new
    @avgRequestTurnaround = QuickStats.new

    # Mean = alpha * beta
    @request_arrival_distribution = Gamma.new(alpha: newRequestGAShape,
                beta: newRequestGAMean / newRequestGAShape)
    @onload_service_distribution = Gamma.new(alpha: onloadGAShape,
                beta: (onloadGAMean - 1.0/24) / onloadGAShape)
    @offload_service_distribution = Gamma.new(alpha: offloadGAShape,
                beta: (offloadGAMean - 1.0/24) / offloadGAShape)
    @haltTime = haltTime
  end

  # to kickstart the event graph
  def init
    @ships = %w[USSRed USSBlue USSGreen USSOrange USSCyan USSYellow]

    @FLCs.merge!(
      'San Diego' => FLC.new('San Diego', Location.new(32, -117)),
      'Pearl Harbor' => FLC.new('Pearl Harbor', Location.new(21, -157)),
      'Yokosuka' => FLC.new('Yokosuka', Location.new(35, 139)),
      'Busan' => FLC.new('Busan', Location.new(35, 129)),
      'Hong Kong' => FLC.new('Hong Kong', Location.new(22, 114))
    )
    flc_names = @FLCs.keys

    @numberUSV.times do |i|
      name = "USV#{'%03d' % i}"
      flc_loc = @FLCs[flc_names[rand(0...flc_names.length)]].location
      @USV_available[name] = USV.new(name, flc_loc,
                top_speed: @top_speed, model: self)
    end

   @parts.merge!(
      '12345' => Part.new(stock_number: '12345', weight: 2.0),
      '23456' => Part.new(stock_number: '23456', weight: 4.0),
      '34567' => Part.new(stock_number: '34567', weight: 6.0),
      '45678' => Part.new(stock_number: '45678', weight: 8.0),
      '56789' => Part.new(stock_number: '56789', weight: 10.0)
    )

    @FLCs['San Diego'].add_inventory('12345', 10)
    @FLCs['San Diego'].add_inventory('23456', 5)
    @FLCs['San Diego'].add_inventory('34567', 0)
    @FLCs['San Diego'].add_inventory('45678', 0)
    @FLCs['San Diego'].add_inventory('56789', 0)

    @FLCs['Pearl Harbor'].add_inventory('12345', 0)
    @FLCs['Pearl Harbor'].add_inventory('23456', 10)
    @FLCs['Pearl Harbor'].add_inventory('34567', 5)
```

```
  @FLCs['Pearl Harbor'].add_inventory('45678', 0)
  @FLCs['Pearl Harbor'].add_inventory('56789', 0)

  @FLCs['Yokosuka'].add_inventory('12345', 0)
  @FLCs['Yokosuka'].add_inventory('23456', 0)
  @FLCs['Yokosuka'].add_inventory('34567', 10)
  @FLCs['Yokosuka'].add_inventory('45678', 5)
  @FLCs['Yokosuka'].add_inventory('56789', 0)

  @FLCs['Busan'].add_inventory('12345', 0)
  @FLCs['Busan'].add_inventory('23456', 0)
  @FLCs['Busan'].add_inventory('34567', 0)
  @FLCs['Busan'].add_inventory('45678', 10)
  @FLCs['Busan'].add_inventory('56789', 5)

  @FLCs['Hong Kong'].add_inventory('12345', 5)
  @FLCs['Hong Kong'].add_inventory('23456', 0)
  @FLCs['Hong Kong'].add_inventory('34567', 0)
  @FLCs['Hong Kong'].add_inventory('45678', 0)
  @FLCs['Hong Kong'].add_inventory('56789', 10)

  @requestCount = 0
  @numAssigned = 0
  @numDeliveries = 0
  @totalinQueue = 0
  @FLCCounter = 0
  @USVCounter = 0
  @assignedUSVCounter = 0
  @assignedFLCCounter = 0

  schedule(:new_request, @request_arrival_distribution.next)
  schedule(:final_report, @haltTime)
end

def start_move(usv:, **args)
  usv.start_move(args)
  if args[:destination]
    time_to_dest = usv.location.distance_to(args[:destination]) / args[:speed]
    schedule(:end_move, time_to_dest, usv: usv)
  end
end

def end_move(usv:)
  usv.end_move
end

def new_request
  # parameterize ships and parts for ease of randomization
  ship = @ships[rand(@ships.size)]
  k = @parts.keys
  part = k[rand(k.size)]
```

```
  req = Request.new(time: model_time, requesting_name: ship,
              stock_number: part, quantity: rand(1..10),
              priority: rand(1..9),
              rdv_location: Location.new(rand(0..45), rand(-124..145)))
  @requests.push req
  @requestCount += 1
  @avgNumRequestsInQueue.update(time: model_time,
              value: @requestCount)

  if @debugmode
    STDERR.puts ’ ’
    STDERR.puts "==========#{model_time.round(2)}: " +
      "Request #{@requestCount} added to the system. " +
      "RDV Location: (#{req.rdv_location.x}, #{req.rdv_location.y}) " +
      "=========="
  end

  schedule(:new_request, @request_arrival_distribution.next)
  schedule(:usv_assignment, 0)
end

def usv_assignment
  request = @requests.peek

  min_total_time = Float::INFINITY
  selected_flc = nil
  selected_usv = nil

  for flc in @FLCs.values do
    next unless flc.how_many(request.stock_number) > 0
    for usv in @USV_available.values do
      time_to_flc_rdv = (usv.location.distance_to(flc.location) +
              flc.location.distance_to(request.rdv_location)) / usv.top_speed
      next unless time_to_flc_rdv < min_total_time
      min_total_time = time_to_flc_rdv
      selected_flc = flc
      selected_usv = usv
    end
  end

  unless selected_flc.nil? || selected_usv.nil?
    selected_usv.rdv_location = request.rdv_location

    request = @requests.pop
    @requestCount -= 1
    @avgNumRequestsInQueue.update(time: model_time,
              value: @requestCount)

    @USV_available.delete(selected_usv.name) if
              @USV_available.include?(selected_usv.name)
```

```
    if selected_usv.location == selected_flc.location
      schedule(:flc_arrival, 0, usv: selected_usv, request: request)
    else
      schedule(:start_move, 0, usv: selected_usv,
        speed: selected_usv.top_speed, destination: selected_flc.location)
      time_to_flc = selected_usv.location.distance_to
              (selected_flc.location) / selected_usv.top_speed
      schedule(:flc_arrival, time_to_flc, usv: selected_usv,
              request: request)
    end
    selected_usv.mission_starttime = model_time
    selected_usv.latest_request = request

    @assignedUSVCounter += 1
    @avgUSVUtil.update(time: model_time,
              value: @assignedUSVCounter)
    STDERR.puts "#{selected_usv.mission_starttime.round(2)}: " +
              "#{selected_usv.name} takes on this request." if @debugmode
  @numAssigned += 1
  end
end

def flc_arrival(usv:, request:, **_args)
  schedule(:start_onload, 0, usv: usv, request: request)
end

def start_onload(usv:, request:, **_args)
  @assignedFLCCounter += 1
  @avgFLCUtil.update(time: model_time, value: @assignedFLCCounter)
  schedule(:end_onload, @onload_service_distribution.next + 1.0/24,
              usv: usv, request: request)
end

def end_onload(usv:, request:, **_args)
  @assignedFLCCounter -= 1
  @avgFLCUtil.update(time: model_time, value: @assignedFLCCounter)
  schedule(:start_move, 0, usv: usv, speed: usv.top_speed,
              destination: request.rdv_location)
  time_to_rdv = usv.location.distance_to(request.rdv_location)/usv.top_speed
  schedule(:arrival_rdv, time_to_rdv, usv: usv)
end

def arrival_rdv(usv:, **_args)
  schedule(:start_offload, 0, usv: usv)
end

def start_offload(usv:, **_args)
  schedule(:end_offload,
              @offload_service_distribution.next + 1.0/24, usv: usv)
end
```

```
def end_offload(usv:, **_args)
  @assignedUSVCounter -= 1
  @avgUSVUtil.update(time: model_time, value: @assignedUSVCounter)
  @numDeliveries += 1
  @numAssigned = @numAssigned - 1
  @avgTimeMission.new_obs(model_time - usv.mission_starttime)
  @avgRequestTurnaround.new_obs(model_time - usv.latest_request.time)
  if @debugmode
    STDERR.puts ' '
    STDERR.puts "$$$$$$$$$$ #{model_time.round(2)}: #{usv.name} " +
              "delivered to location (#{usv.location.x.round(0)}, " +
              "#{usv.location.y.round(0)}) after " +
              "#{(model_time - usv.mission_starttime).round(2)} days." +
              "$$$$$$$$$$"
    STDERR.puts "$$$$$$$$$$ Total number of deliveries made by all " +
              "USVs: #{@numDeliveries}"
    STDERR.puts ' '
  end
  @USV_available[usv.name] = usv
  STDERR.puts "#{model_time.round(2)}: #{@totalinQueue} are in " +
              "queue." if @debugmode

  min_total_time = Float::INFINITY
  closest_flc = nil

  if @requests.empty?
    for flc in @FLCs.values do
      time_to_flc = usv.location.distance_to
              (flc.location) / usv.top_speed
      next unless time_to_flc < min_total_time
      min_total_time = time_to_flc
      closest_flc = flc
      end
    schedule(:start_move, 0, usv: usv, speed: 0.5 * usv.top_speed,
              destination: closest_flc.location)
  else
    schedule(:usv_assignment, 0)
  end
end

def final_report
  print "AvgNumRequestsInQueue,"
  print "AvgFLCUtil,"
  print "AvgUSVUtil,"
  print "AvgTimeMission,"
  print "avgRequestTurnaround,"
  print "unfilledRequestAvgTime\n"
  print @inputset

  print "#{(@avgNumRequestsInQueue.average).round(5)}"
  print ",#{(@avgFLCUtil.average).round(5)}"
```

```
    print ",#{(@avgUSVUtil.average / @numberUSV).round(5)}"
    print ",#{(@avgTimeMission.average).round(5)}"
    print ",#{(@avgRequestTurnaround.average).round(5)}"
    unfilledRequestStats = QuickStats.new
    while req = @requests.pop
      unfilledRequestStats.new_obs(model_time - req.time)
    end
    result = unfilledRequestStats.average
    result = 0.0 if result.nan?
    print ",#{(result).round(5)}"
    puts
    schedule(:halt, 0)
  end
end

if ARGV.length != 9
  STDERR.puts "\nMust supply nine command-line arguments:\n"
  STDERR.puts "\tNew Request Mean (double)"
  STDERR.puts "\tShape for Gamma Distribution of " +
              "New Request Mean (double)"

  STDERR.puts "\tFLC Onload Mean (double)"
  STDERR.puts "\tShape for Gamma Distribution of " +
              "FLC Onload Mean (double)"

  STDERR.puts "\tOffload Mean (double)"
  STDERR.puts "\tShape for Gamma Distribution of " +
              "#Offload Mean (double)"

  STDERR.puts "\tTop Speed (double)"
  STDERR.puts "\tNumber of USVs (int)"
  STDERR.puts "\tNumber of Days to Run (int)"
  STDERR.puts "\nExample: ruby #{File.basename($PROGRAM_NAME)} " +
              "1.0 3.0 10.2 4.0 14.3 7.0 20 6 10000\n"
else
  newRequestGAMean = ARGV[0].to_f
  newRequestGAShape = ARGV[1].to_f
  onloadGAMean = ARGV[2].to_f
  onloadGAShape = ARGV[3].to_f
  offloadGAMean = ARGV[4].to_f
  offloadGAShape = ARGV[5].to_f
  top_speed = ARGV[6].to_f
  numberUSV = ARGV[7].to_i
  haltTime = ARGV[8].to_i
  print "newRequestGAMean,newRequestGAShape,onloadGAMean," +
              "onloadGAShape,offloadGAMean,offloadGAShape,top_speed," +
              "numberUSV,haltTime,"
  inputset = ARGV.join(",") + ","
  # To run in debug mode, add 'debugmode: true' to the USV_model.new cmd
  USV_model.new(newRequestGAMean: newRequestGAMean,
    newRequestGAShape: newRequestGAShape,
```

```
    onloadGAMean: onloadGAMean,onloadGAShape: onloadGAShape,
    offloadGAMean: offloadGAMean, offloadGAShape: offloadGAShape,
    haltTime: haltTime, top_speed: top_speed,
    numberUSV: numberUSV, inputset: inputset).run
end
```

A.2 OBJECT: CLASS USV

```
require_relative 'location'

class USV
  # attributes of USV
  attr_reader :name, :top_speed, :cargo_capacity, :fuel_capacity,
              :destination, :moving, :bearing
  attr_accessor :rdv_location, :mission_starttime, :latest_request

  def initialize(name, location, top_speed: 25.0, cargo_capacity: 2.0,
              fuel_capacity: 1000.0, model: nil)
    @name = name
    @top_speed = top_speed
    @cargo_capacity = cargo_capacity
    @fuel_capacity = fuel_capacity
    @my_model = model

    # Current state
    @current_location = location
    @current_cargo = 0.0
    @current_fuel = 0.0
    @current_speed = 0.0
    @bearing = nil
    @moving = false
    @latest_request = nil
    @mission_starttime = mission_starttime
    @start_time = nil
    @destination = nil
    @trip_distance = 0.0
    @rdv_location = nil
  end

  def start_move(time: nil, speed: nil, destination: nil,
              bearing: nil)
    time ||= @my_model.model_time
    @current_location = location(time: time)
    if speed && speed > 0
      @current_speed = [speed, @top_speed].min
      @start_time = time
      @start_location = @current_location
      @moving = true
    end

    # Determine bearing.
    if destination.nil? && bearing.nil?
      print('No destination or bearing given to start_move.')
      exit
    end
    bearing = ((90 - bearing) % 360.0) * Math::PI / 180.0 if bearing
    @bearing = bearing || @current_location.bearing_to(destination)
```

```ruby
    @destination = destination
    @start_location
  end

  def end_move(time: nil)
    time ||= @my_model.model_time
    pos = location(time: time)
    # @current_location = @destination.distance_to(pos) < 1E-10 ?
                @destination : pos
    @current_location = pos
    @current_speed = 0.0
    @start_time = nil
    @start_location = nil
    @bearing = nil
    @destination = nil
    @moving = false
  end

  def location(time: nil)
    time ||= @my_model.model_time
    elapsed_time = time - (@start_time || time)
    if @moving && (elapsed_time > 0)
      current_x = @start_location.x + @current_speed * elapsed_time *
                Math.cos(@bearing)
      current_y = @start_location.y + @current_speed * elapsed_time *
                Math.sin(@bearing)
      @current_location = Location.new(current_x, current_y)
    end
    @current_location
  end
end

# Test functionality
if $PROGRAM_NAME == __FILE__
  usv = [USV.new('#1', Location.new(0, 0)), USV.new('#2',
                Location.new(0, 0), top_speed: Math.sqrt(8.0))]
  p usv
  usv[1].start_move(time: 0.0, speed: 20.0, bearing: 45.0)
  (1..15).each do |t|
    usv[0].start_move(time: t, speed: 1.0,
                destination: Location.new(3.0, 4.0)) if t == 5
    usv[0].end_move(time: t) if t == 10
    usv.each { |u| puts "#{t}, #{u.location(time: t).inspect}" }
  end
end
```

A.3 OBJECT: CLASS FLC

```ruby
require_relative 'location'

class FLC
  # attributes of FLC
  attr_reader :name, :location

  def initialize(name, location)
    @name = name
    @location = location
    @inventory = Hash.new(0)
  end

  def how_many(part_id)
    @inventory[part_id]
  end

  def add_inventory(part_id, qty)
    @inventory[part_id] += qty if qty >= 0
  end

  def draw_inventory(part_id, qty)
    allocated = [@inventory[part_id], qty].min
    @inventory[part_id] -= allocated
    allocated
  end
end

# Test functionality
if $PROGRAM_NAME == __FILE__
  my_flc = FLC.new("Monterey", Location.new(1.0, 2.0))
  p my_flc
  my_flc.add_inventory("florp", 5)
  my_flc.add_inventory("blorp", 10)
  p my_flc
  puts "Location of #{my_flc.name} is #{my_flc.location.inspect}"
  qty = [10, 5, 1_000_000]
  %w{florp blorp bloop}.each.with_index do |part, i|
    puts "#{part}: #{my_flc.how_many(part)}"
    puts "requested #{qty[i]}, got " +
               "#{my_flc.draw_inventory(part, qty[i])}"
    puts "#{part}: #{my_flc.how_many(part)}"
  end
end
```

A.4 OBJECT: CLASS REQUEST

```
require_relative 'location'

class Request
  # attributes of Request
  # requesting_name: name of requesting entity
  # stock_number: part number
  # quantity: required amount
  # priority: 1 (highest) through 9 (lowest)
  # rdv_location: rendezvous location, specified by requesting entity
  attr_reader :requesting_name, :stock_number, :quantity,
              :priority, :rdv_location, :time

  def initialize(requesting_name: nil, stock_number: nil, quantity: 0,
                 priority: 0, rdv_location: nil, time: nil)
    @requesting_name = requesting_name
    @stock_number = stock_number
    @quantity = quantity
    @priority = priority
    @rdv_location = rdv_location
    @time = time
  end

  include Comparable
  # comparison operator
  def <=> (other)
    if self.priority == other.priority
      self.time <= other.time ? -1 : 1
    elsif self.priority < other.priority
      -1
    else
      1
    end
  end
end

# internally runs the program for testing purposes
if $PROGRAM_NAME == __FILE__
  r = Request.new(requesting_name: 'USSBlue', stock_number: '12345',
                 quantity: 5, priority: 3,
                 rdv_location: Location.new(rand(270)-124, rand(46)))
  p r
  puts r.rdv_location
end
```

A.5 OBJECT: CLASS PART

```
class Part
  # attributes of Part
  attr_reader :stock_number, :weight

  def initialize(stock_number: nil, weight: 0)
    @stock_number = stock_number
    @weight = weight
  end
end

if $PROGRAM_NAME == __FILE__
  # r = Part.new(stock_number: '12345', weight: 15)
  # r = Part.new(weight: 15, stock_number: '12345')
  r = Part.new(stock_number: '12345')
  p r
  puts r.stock_number
  puts r.weight
end
```

A.6 OBJECT: CLASS LOCATION

```
class Location
  # Define an immutable location object
  attr_reader :x, :y

  def initialize(x, y)
    @x = x
    @y = y
  end

  # distance to "other" location starting from current location
  def distance_to(other)
    Math.sqrt((other.x - @x)**2 + (other.y - @y)**2)
  end

  # find bearing from receiver location to other location
  def bearing_to(other)
    Math.atan2((other.y - @y), (other.x - @x))
  end
end
```

A.7 OBJECT: CLASS TIMEVARYINGSTATS

```
class TimeVaryingStats
  attr_reader :average

  def initialize
    @last_update_time = 0.0
    @last_update_value = 0.0
    @running_tally = 0.0
  end

  def update(time:, value:)
    elapsed_time = time - @last_update_time
    @running_tally += elapsed_time * @last_update_value
    @last_update_value = value
    @last_update_time = time
    @average = @running_tally / time if time > 0
  end
end

if __FILE__ == $PROGRAM_NAME
  values = [1, 2, 3, 0]
  times = [10, 20, 100, 200]
  stats = TimeVaryingStats.new
  values.each_with_index { |v,i| p stats.update(value: v, time: times[i]) }
  p stats.average
end
```

THIS PAGE INTENTIONALLY LEFT BLANK

List of References

Banks J, II JSC, Nelson BL, Nicol DM (2010) *Discrete-Event System Simulation* (Upper Saddle River, NJ: Prentice-Hall, Inc.), 5th edition.

Casola KJ (2017) *System architecture and operational analysis of Medium Displacement Unmanned Surface Vehicle Sea Hunter as a surface warfare component of distributed lethality*. Master's thesis, Naval Postgraduate School, Monterey, CA, URL https://calhoun.nps.edu/handle/10945/55579.

DARPA (2018) About DARPA. Accessed December 4, 2018, URL https://www.darpa.mil/about-us/about-darpa.

Department of Navy (2012) *Naval Warfare Publication 1-03.1, Operational Reports*.

Henggeler T (2014) A cascade of "waterfall support" for RIMPAC 2014. *Navy Supply Corps Newsletter* (October 22), URL http://scnewsltr.dodlive.mil/2014/10/22/a-cascade-of-waterfall-support-for-rimpac-2014/.

Hilley MK (2009) Logistics lauded as lynchpin for units conducting counterpiracy ops. *Navy News Service* (January 21), URL http://www.navy.mil/submit/display.asp?story_id=41967.

Kelton WD, Sadowski RP, Zupick NB (2014) *Simulation with Arena* (New York: McGraw-Hill Education), 6th edition.

Larter DB (2017) Analyst: With ballooning costs for a smaller Navy, can it really afford 355 ships? *Defense News* (December 7), URL https://www.defensenews.com/naval/2017/12/07/analyst-with-ballooning-costs-for-a-smaller-navy-can-it-really-afford-355-ships/.

Law AM (2015) *Simulation Modeling and Analysis* (New York: McGraw-Hill Education), 5th edition.

Leidos (2016) Leidos participates in christening of DARPA's innovative, highly autonomous prototype unmanned vessel. *CISION©PR Newswire* (April 7), URL https://www.prnewswire.com/news-releases/leidos-participates-in-christening-of-darpas-innovative-highly-autonomous-prototype-unmanned-vessel-300247996.html.

Leshak B (2018) Navy's newest oceanographic survey ship completes first operational mission. *Sealift* (January), URL http://www.msc.navy.mil/sealift/2018/January/maury.htm.

MacCalman A, Vieira H, Lucas T (2016) Second-order nearly orthogonal Latin hypercubes for exploring stochastic simulations. *Journal of Simulation* 11(2):137–150.

Matsumoto Y (2018) Ruby. Accessed December 4, URL https://www.ruby-lang.org/en/.

Military Sealift Command (2012) Annual Report: MSC 2012 in review. Accessed December 4, 2018, URL https://www.msc.navy.mil/annualreport/2012/organization.htm.

Military Sealift Command (2018a) About MSC. Accessed December 4, URL http://www.msc.navy.mil/mission/.

Military Sealift Command (2018b) MSC organization. Accessed December 4, URL https://www.msc.navy.mil/organization/.

Naval Supply Systems Command (2018) Naval Supply Systems Command website. Accessed December 4, URL https://www.navsup.navy.mil/public/navsup/supplycorps/.

NAVSUP (2003) *MILSTRIP, MILSTRAP Desk Guide*. Naval Supply Systems Command.

Nelson BL (2013) *Foundations and Methods of Stochastic Simulation* (Springer).

Outreach (2018) DARPA: actuv "sea hunter" prototype transitions to office of naval research for further development. URL https://www.darpa.mil/news-events/2018-01-30a.

Pike J (1999) Underway replenishment (unrep). URL https://fas.org/man/dod-101/sys/ship/unrep.htm.

Richardson JM (2017) The future navy. Speech, URL https://www.navy.mil/navydata/people/cno/Richardson/Resource/TheFutureNavy.pdf.

Rosamond J (2017) Dsei: Navy poised to order second vessel for actuv sea hunter test program. URL https://www.tapatalk.com/groups/warships1discussionboards/lpd-s-modified-to-serve-as-asw-mothership-t32265-s120.html.

Sanchez PJ (2009) Fundamentals of simulation modeling. *Engineering Management Review, IEEE* 37(2):23–33.

Sanchez PJ (2018) Simplekit gem. URL https://rubygems.org/gems/simplekit.

Saunders S (2018) Anti-submarine warfare unmanned vessel. *Jane's Fighting Ships* URL http://janes.ihs.com/FightingShips/Display/1767400.

Schruben LW (1983) Simulation modeling with event graphs. *Communications of the ACM* 26(11):957–963.

See HA (2017) *Coordinated Guidance Strategy for Multiple USVs During Maritime Interdiction Operations*. Master's thesis, Naval Postgraduate School.

Smith JS, Sturrock DT, Kelton WD (2017) *Simio and Simulation: Modeling, Analysis, Applications, 4th Edition* (CreateSpace Independent Publishing Platform).

Solem K (2017) *Quantifying the Potential Benefits of Anti-submarine Warfare (ASW) Continuous Trail Unmanned Vessels (ACTUV) in a Tactical ASW Scenario*. Master's thesis, Naval Postgraduate School, URL https://my.nps.edu/web/seed/theses.

Stiner K (2018) Navsup naval supply systems command. URL https://www.dawnbreaker.com/portals/phase3/office-of-the-assistant-secretary-of-the-navy-research-development-and-acquisition/syscoms/navsup/.

Toh YJB (2017) *Development of a Vision-Based Situational Awareness Capability for Unmanned Surface Vessels*. Master's thesis, Naval Postgraduate School.

United States Navy (2009) *Logistics Specialist: Supply Fundamentals*. NAVEDTRA 15004, S/N 0504-LP-109-2423. Accessed December 4, URL http://navybmr.com/study%20material/NAVEDTRA%2015004.pdf.

US Navy (2007) *The Navy Unmanned Surface Vehicle (USV) Master Plan*. URL http://www.navy.mil/navydata/technology/usvmppr.pdf.

US Navy (2018) A legacy of innovation. URL https://www.onr.navy.mil/en/About-ONR/History.

Walan AMG (2016) DARPA: Anti-submarine warfare (asw) continuous trail unmanned vessel (actuv). URL https://www.darpa.mil/program/anti-submarine-warfare-continuous-trail-unmanned-vessel.

Ziezulewicz G (2017) Navy's resupply and surge sealift ships facing readiness decline. URL https://www.navytimes.com/news/your-navy/2017/08/28/report-navys-resupply-and-surge-sealift-ships-facing-readiness-decline/.

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California