3-11-2011

Architecture Based Workload Analysis of UAS Multi-Aircraft Control: Implications of Implementation on MQ-1B Predator

Jason D. McGrogan

Michael F. Schneider

ARCHITECTURE BASED WORKLOAD ANALYSIS OF UAS MULTI-AIRCRAFT CONTROL: IMPLICATIONS OF IMPLEMENTATION ON MQ-1B PREDATOR

THESIS

Michael F. Schneider
Civilian, USAF

Jason D. McGrogan
Major, USAF

AFIT/GSE/ENV/11-M02

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GSE/ENV/11-M02

ARCHITECTURE BASED WORKLOAD ANALYSIS OF UAS MULTI-AIRCRAFT CONTROL: IMPLICATIONS OF IMPLEMENTATION ON MQ-1B PREDATOR

THESIS

Presented to the Faculty

Department of Systems and Engineering Management

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Systems Engineering

Jason D. McGrogan, MS
Major, USAF

Michael F. Schneider, BMEE
Civilian, USAF

March 2011

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

AFIT/GSE/ENV/11-M02

ARCHITECTURE BASED WORKLOAD ANALYSIS OF UAS MULTI-AIRCRAFT CONTROL: IMPLICATIONS OF IMPLEMENTATION ON MQ-1B PREDATOR

Jason D. McGrogan, MS

Major, USAF

Michael F. Schneider, BMEE

Civilian, USAF

Approved:

/signed/ ____________________ ________
John M. Colombi, PhD (Chairman) Date

/signed/ ____________________ ________
Michael E. Miller, PhD (Member) Date

/signed/ ____________________ ________
David S. Long, Col, USAF (Member) Date

AFIT/GSE/ENV/11-M02

Abstract

An increased demand for use of Unmanned Aircraft Systems (UASs) without commensurate increases in pilot manpower has prompted proposals for simultaneous control of multiple aircraft by a single pilot or Multi-Aircraft Control (MAC). To understand the potential effects of MAC, an IMPRINT Pro, Multi-Resource Theory, pilot workload model was developed from pedigreed system architecture. Feedback from active UAS pilots was used to validate the model and establish a workload saturation threshold value of 60, above which pilots may experience performance degradation over extended periods of time. The model predicts that pilots experience low workload when operating one or two UASs during benign operations, and operate 91% of the time below a workload of 25 without saturation. However, conflict from multi-task overlap builds rapidly when the pilot is required to operate three or more aircraft. The percentage of time over the saturation threshold increases to 21% with four aircraft under benign operating conditions. When dynamic events are introduced the workload becomes unmanageable, with estimates regularly over 100 due to multi-task overlap and communication activities. The analysis indicates the need for techniques and technology to reduce task and communications demands on UAS pilots to effectively implement MAC.

Acknowledgments

The authors would like to sincerely express our gratitude to our thesis advisors, Dr. John Colombi, Dr. Michael Miller, and Col. David Long. They constantly challenged and supported us throughout our thesis research, enabling the depth and quality of our work. We would also like to thank the North Dakota Air National Guard 178th Reconnaissance Squadron and 119th Operational Support Squadron for allowing a valuable and informative look inside the job of an MQ-1B pilot. Without their knowledge and experience we would not have been able achieve the level of fidelity that we did in our analysis. Finally we would like to thank our sponsors, the UAS Medium Altitude UAS Division along with the 711th Human Performance Wing, for allowing us to be a part of this critical research. Their input and funding provided the basis for our entire thesis work.

Jason McGrogan, Major, USAF

Michael Schneider, Civilian, USAF

Table of Contents

Page

Page

List of Figures

Page

Page

List of Tables

Page

ARCHITECTURE BASED WORKLOAD ANALYSIS OF UAS MULTI-AIRCRAFT CONTROL: IMPLICATIONS OF IMPLEMENTATION ON MQ-1B PREDATOR

I. Introduction

1.1 Unmanned Aerial Vehicles on the Battlefield

The U.S. Department of Defense continues to increase tasking for Unmanned Aircraft Systems (UASs) to support ongoing operations in Iraq and Afghanistan. The primary role of UASs is to provide Intelligence, Surveillance, and Reconnaissance (ISR) for the Joint Force. They can provide persistence, endurance, and efficiency beyond what is possible with manned ISR platforms (USAF, 2009), and without putting a human in harm's way. The increasing numbers of UASs on the battlefield provide combatant commanders with unprecedented levels of information, but have put a strain on aspects of pilot induction, training, and retention (USAF, 2009).

One of the most well known, and prevalent, UASs on the battlefield today is the MQ-1B Predator. The MQ-1B Predator is a medium sized UAS with a documented effective radius of 500NM and an endurance of over 24 hours. It can be configured with an Electro-Optical/Infrared (EO/IR) sensor or a Synthetic Aperture Radar (SAR). Due to its utility, the MQ-1B Predator rapidly became ubiquitous as part of joint operations. Initially, the MQ-1B Predator was a dedicated ISR platform, providing streaming video to warfighters in theater and joint organizations in the United States.

The increased demand and proven capability of UASs, and the MQ-1B in particular, has spurred an increase in procurement and development of UAS technologies. In addition to the ISR role, UASs have increasingly become multi-role, with the ability to strike. For example, the MQ-1B Predator was modified with a laser designator and the ability to carry AGM-114 Hellfire missiles. With this added strike capability, the MQ-1B Predator is able to execute the entire Find-Fix-Track-Target-Engage-Assess (F2T2EA) process (USAF & USA, 2009). The F2T2EA process represents the entire kill chain from finding a target through assessing the effects of a strike. A single platform that can perform the F2T2EA process so effectively is invaluable to combatant commanders. This unique mix of ISR and strike capability rapidly made the MQ-1B the weapon of choice for high value targets in the Iraq and Afghanistan AORs as evidenced by the media coverage it has elicited.

1.2 UAS Manpower Limitations

Medium sized, multi-role UASs are in high demand in Iraq and Afghanistan. From 2004 to 2009 there has been a 660% growth in MQ-1B Predator and MQ-9 Reaper Combat Air Patrols (CAPs) (USAF, 2009). Every prediction indicates a continued increase in this demand.

Although the ability of the medium sized UAS to remain on station for significantly longer periods of time than manned aircraft is a primary benefit, the long duration flight also presents challenges. While the air vehicles are unmanned, the systems are remotely piloted and require an experienced, highly-trained pilot, sensor operator, and mission intelligence coordinator for operation throughout each 24 hour combat air patrol (CAP). As a result, multiple pilots are required to support a single CAP, even though only a single pilot is required at any point in time. In fact, the ability to exchange fatigued pilots for rested pilots is one of the features that enable the UAS to remain on station for extended periods of time. This fact, coupled with demand for ever increasing numbers of CAPs has resulted in a situation where manpower is rapidly becoming the limiting factor to operations.

One proposed solution requires an individual pilot to simultaneously control multiple aircraft during their shift. This solution, termed Multi-Aircraft Control (MAC), could reduce the number of pilots required to perform the desired number of CAPs and provide a solution to the manpower problem. The number of aircraft a pilot is controlling is termed the MAC ratio. A MAC ratio of 1 is actually no MAC since the pilot is not controlling multiple aircraft. The theory of MAC assumes that a single pilot can effectively control multiple UASs.

Currently, no rigorous analysis of all of the critical factors effecting the implementation of MAC has been performed. The critical factors effecting the implementation of MAC are those that have a significant impact on the pilot's ability to effectively operate multiple UASs simultaneously. These critical factors are hypothesized to be major drivers of system interface design and operations concept formulation. A sound analytic basis is required to assess the full implementation of MAC to ensure that all critical factors and their interactions are considered to avoid degradation of mission performance. Only with a solid understanding of all the factors that affect the implementation of MAC can an effective and operationally suitable system be designed and implemented.

1.3 Scope of MAC Research

The future MAC concept for MQ-1B is evaluated using the Architecture Based Analysis Process (ABEP) method to assess pilot effectiveness through the use of workload modeling in order to identify and assess the critical factors relating to the implementation of MAC. New architectural products are developed as necessary to facilitate model creation. Human Performance Modeling (HPM) is used to assess the pilot's performance in the context of missions, to include benign and dynamic ISR operations, strike missions, emergency operations, and aircraft handover/changeover. Critical factors relating to pilot performance are subsequently analyzed to assess the effectiveness of the system architecture. The MQ-1B pilot is the focus of this research and only the interactions and tasks thereof are addressed. Pilot control interfaces are abstracted to the level necessary for HPM and specific Human-Computer Interaction issues are not addressed. The sensor operator and mission intelligence coordinator are excluded from this analysis along with aircraft, satellite, GCS, and communication considerations. They are all taken to be external to the system under analysis and are assumed to perform optimally except under the emergency condition. This analysis does not investigate the effectiveness of workload mitigation strategies, instead it address the workload imposed on the pilot by the system, assuming the pilot will perform the operations that are primarily allocated to them by the system.

1.4 Purpose of MAC Research

The purpose of the thesis is to identify the critical factors and their effects on pilot workload involved in implementing MAC with the current MQ-1B system architecture.

MAC is a shift in the paradigm of a pilot controlling a single aircraft. During MAC, a pilot will be forced to spread their attention across multiple aircraft performing different mission, ideally without any impact to the mission effectiveness. This new paradigm demands substantially more of the pilots and the entire system to support simultaneous, geographically separated operations. To make informed decisions on the operations concepts and the technology required, a thorough and in-depth study of the critical factors and their interactions in MAC is required. The system architecture and simulation tools developed as part of this analysis provide a method to assess the effect of the selected factors on pilot workload during MAC and how they can be manipulated to achieve a desired outcome. This analysis provides data which can impact the development of operations concepts, current and future acquisition of MAC technologies for UASs, as well as provide a set of tools to analyze future system modifications. This analysis is the first step of many to characterize the challenges of MAC and better implement the systems and practices to best take advantage of this new paradigm of UAS operations.

1.5 Methodology for MAC Analysis

This analysis follows the Architecture Based Evaluation Process (ABEP) for the analysis of MAC implementation. ABEP is a process for using system architecture views to generate a model of the system. The model represents the system architecture as currently designed so it can be used to evaluate the effectiveness of the architecture to meet the requirements of the system.

This analysis uses the existing system architectures for UAS operations to develop human view architecture focused around the UAS pilot. Existing system architecture is very broad so it had to be scaled down and scoped to fit the needs of this analysis. The human view architecture captures all of the pilot's system interfaces and tasks related to piloting the UAS.

This analysis used the Improved Performance Integration Tool (IMPRINT) Pro human performance modeling software to characterize the workload experienced by the pilot as part of the system. The Army Research Laboratory, Human Research and Engineering Directorate, developed the Improved Performance Research Integration Tool (IMPRINT) as a human performance modeling tool for military applications. The human view architecture was used as the basis for the IMPRINT model. The model was set up to represent all of the tasks that a pilot would have to accomplish during different mission modes throughout a normal shift, with the flexibility to alter the number of aircraft a single pilot controlled and the mission profile that each of these aircraft flew. This model arrangement provided the flexibility to explore the workload implications of numerous scenarios and factors of MAC.

Extensive discussions with MQ-1B pilots were used to validate the system architecture and model development. The data from the pilot discussion allowed model assumptions and information to be refined. The discussions with the MQ-1B pilots also provided a firsthand assessment of the difficulties of performing Predator operations. This allowed the model output to be validated and provided the foundation for establishing a saturation threshold for the maximum amount of workload a pilot can manage without workload mitigation strategies or mission degradation.

The data analysis was broken up into two major phases. Phase I of the data analysis addressed every possible combination of mission phase and number of aircraft with a few select restrictions. First, order did not matter with the different combinations of mission phases. Second, no more than two dynamic events could occur simultaneously, because the workload generated was so high as to be impractical. Phase II was set up to provide direct comparison of the most relevant mission scenarios to illustrate the impact and interactions of different mission phases on workload. Phase II was set up to represent a nominal pilot's shift of 2.5-3 hours with the pilot changing over with another pilot at the beginning and the end of their shift. Phase II also addressed the workload drivers by analyzing tasks, workload channels, and conflict generated during different mission phases.

II. Background

2.1 MQ-1B Predator UAS

The MQ-1B Predator, depicted in Figure 1, is a medium-altitude, long-endurance UAS used for close air support, air interdiction, and ISR. The MQ-1B Predator refers to the entire system including four aircraft, Ground Control System (GCS), satellite link, and the operations and maintenance crew. The Predator operations crew consists of a rated pilot, an enlisted sensor operator, and an enlisted mission intelligence coordinator. The Predator air vehicle is equipped with a Multi-spectral Targeting System, which has an infrared sensor, TV cameras, and a laser designator. The Predator can be equipped with two laser guided AGM-114 Hellfire missiles. (USAF, 2010)

Figure 1. MQ-1B Predator UAS (Airforce-Technology.com, 2011)

General Characteristics
Primary Function: Armed reconnaissance, airborne surveillance and target acquisition
Contractor: General Atomics Aeronautical Systems Inc.
Power Plant: Rotax 914F four cylinder engine
Thrust: 115 horsepower
Wingspan: 55 feet (16.8 meters)
Length: 27 feet (8.22 meters)
Height: 6.9 feet (2.1 meters)
Weight: 1,130 pounds (512 kilograms) empty
Maximum takeoff weight: 2,250 pounds (1,020 kilograms)
Fuel Capacity: 665 pounds (100 gallons)
Payload: 450 pounds (204 kilograms)
Speed: Cruise speed around 84 mph (70 knots), up to 135 mph
Range: Up to 770 miles (675 nautical miles)
Ceiling: Up to 25,000 feet (7,620 meters)
Armament: Two laser-guided AGM-114 Hellfire missiles
Crew (remote): Two (pilot and sensor operator)
Initial operational capability: March 2005
Unit Cost: $20 million (fiscal 2009 dollars) (includes four aircraft, a ground control station and a Predator Primary Satellite Link)
Inventory: Active force, 130; ANG, 8; Reserve, 0 (USAF, 2010)

2.1.1 MQ-1B GCS

The Predator GCS has two workstations as shown at the right side of Figure 2. The pilot workstation is on the left side of the center equipment rack and the sensor operator workstation is on the right side of the center equipment rack. The current GCS configuration is built around the pilot/sensor operator pair. The pilot and sensor operator work side by side on a single mission as seen in Figure 3.

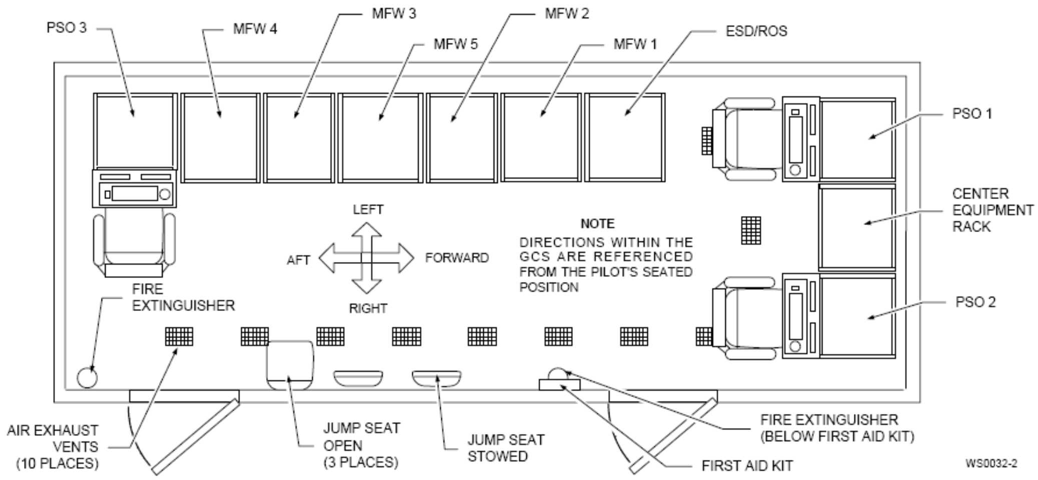


Figure 2. Top-View of the Current Ground Control Station (Bagnall et al., 2010)

Figure 3. Picture of MQ-1B Predator GCS (Eaton et al., 2006)

A prototype configuration for a MAC GCS is seen in Figure 4. In this GCS there are two pilot workstations so that an on-call pilot can assume control of one or more of the MQ-1B Predators in the event of an emergency or dynamic situation.

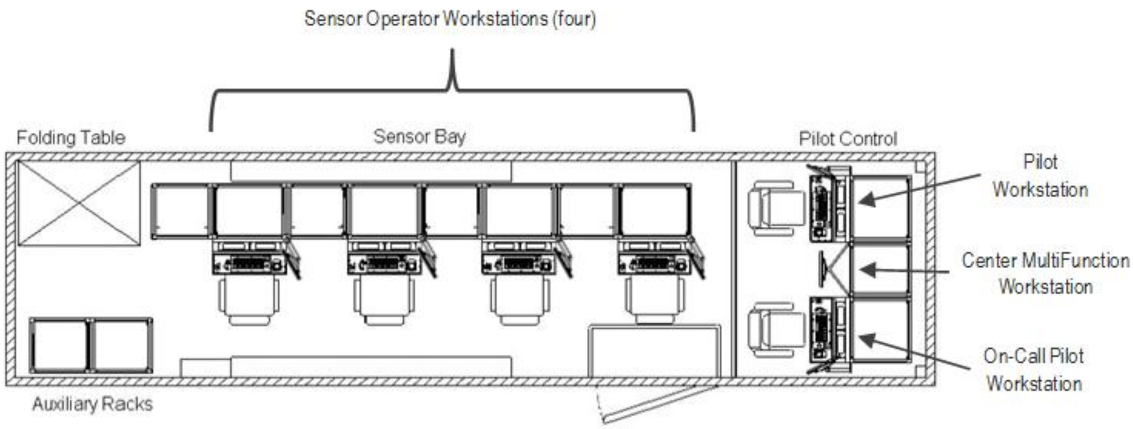


Figure 4. Top-View of Prototype MAC Ground Control Station (Bagnall et al., 2010)

2.2 MAC UAS Manpower Study

Since the ultimate objective of MAC is to reduce the number of pilots required for operations, it is necessary to analyze at the manpower savings generated from implementing MAC. An initial manpower study was performed in parallel to this research to characterize the savings of MAC and the influence of mission parameters. A discrete event simulation was used to track the usage rates of pilot resources as aircraft entities moved through a stochastic model. The model decomposed the mission into launch, transit, benign, dynamic, emergency, and recovery sequences. The number of aircraft, MAC ratio, operational profile, and reliability varied to provide an exploration of their effects. The operational profile is the percentage of aircraft that perform benign missions in which MAC could be used versus the percentage of aircraft performing dynamic missions in which in which a pilot controlled a single aircraft. Reliability is represented as the percentage of aircraft which experience an emergency. These parameters were varied along realistic values to predict the number of pilots necessary at each

MAC ratio and then tested with extreme values. Figure 5 is the percent pilot savings of a representative run and reveals a diminishing trend in the percent reduction in pilots as the MAC ratio increased. (McGrogan & Schneider, 2011)

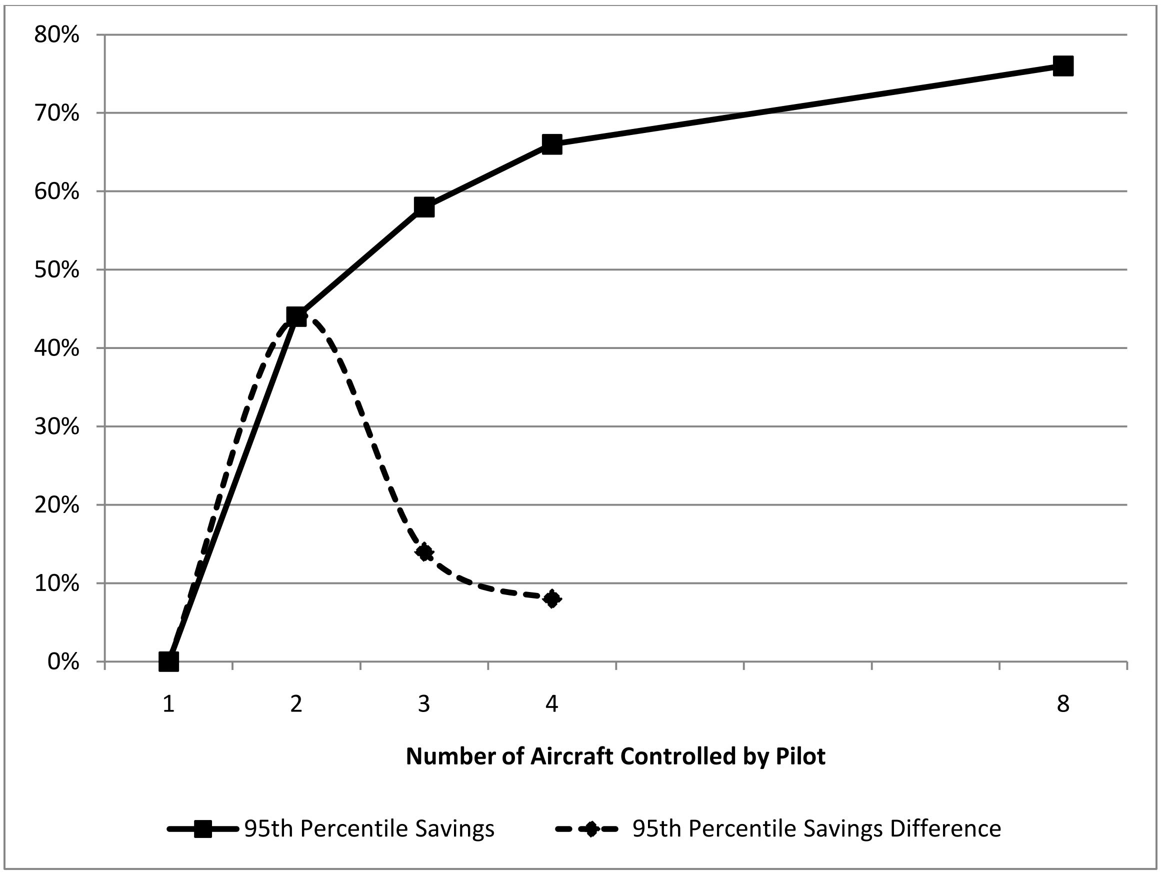


Figure 5. Percent Pilot Savings for Different MAC Ratios (McGrogan & Schneider, 2011)

The number of pilots required to maintain operations at MAC ratio 2 was reduced by 45% over no MAC. However the effect of increasing to MAC ratio 3 and 4 is lessened each time. The manpower savings for MAC ratio 3 increased 14% over MAC ratio 2 to 60% and the manpower savings for MAC ratio 4 increased only 7% over MAC ratio 3 to 67%. It is important to note that this model assumed each MAC ratio to be operationally feasible with no workload limits. (McGrogan & Schneider, 2011)

2.3 Executable Architecture

System architecture is used to provide different system representations to aid in system design and modification. However, system architecture provides a static model and does not provide an effective model of the dynamic nature of a system (Wang & Dagli, 2008). Triggers and resource flows can be represented graphically in system architecture, but a designer cannot observe the system reaction to inputs or resource transfers between nodes. Executable architectures bridge the gap between static architecture representation and a simulation that can represent the system dynamically. The construction of an executable architecture typically involves a manual process consisting of a set of regimented steps to capture all of the relevant information in the static system architecture and transfer it to a simulation environment. Research continues to examine more automated methods, such as extensions to Object Constraint Language (Booch, Rumbaugh, & Jacobson, 2005).

One executable architecture process, Architecture Based Evaluation Process (ABEP), has been applied to standard Department of Defense Architecture Framework (DoDAF) products to generate a dynamic simulation (Dietrichs, Griffin, Schuettke, & Slocum, 2006). With ABEP, the simulations are tied directly to the accepted DoDAF architecture views to ensure that the assumptions and design decisions in the architecture can be modeled directly. The ABEP process has been applied in multiple domains; Dietrichs, Griffin, Schuettke, and Slocum (2006), Bornejko, Glasscock, and Spenkle (2008), and Seibert, Stryker, Ward, and Wellbaum (2009); and was chosen for this research, because it provided a sound foundation for turning "To Be" system architecture into a model for evaluating future system performance. The 8 step ABEP process is enumerated below.

Architecture Based Evaluation Process (Dietrichs et al., 2006)

1. Design Operations Concept of system to be evaluated.

Ops concept provides the system description which the architecture will model, and the models will simulate/evaluate.

2. Identify Measures of Effectiveness (MOE) relevant to the decision/evaluation

Identify the metrics that represent the effectiveness of the system.

3. Identify required level of abstraction for architecture to show traceability to MOE's

Analyze the Ops Concept to determine if MOE's are measured at the output of the system, within the system (requiring 'drilling' into the system activities), or at the output of activities external to the system (requiring external systems diagram)

4. Identify architecture views necessary to capture structure/relationships

a. Structure (OV-1, OV-2, and OV-5) In order to first develop the structure of the analysis, nearly all evaluations will require the OV-1 (High Level Operations Concept), OV-2 (Operational Node Connectivity Description), and OV-5 Operational Activity Model views. The level of abstraction (A-1, A-0, AO etc.) of the OV-5 is initially identified in the previous step.

b. Decision Logic (OV-6a) to capture the logic of the system, nearly all evaluations will require the OV-6a Rules Model, developed to match the level of abstraction used for the OV-5's.

c. As Required: SV-2, SV-4, SV-7,OV-6b, OV-6c Depending on the complexity, consideration for time and dependency on internal performance inputs, some or all of the listed views may be required.

5. Develop architecture views

Develop architecture views in accordance with DODAF to include all relevant activities and entities. If an integrated architecture already exists, then acquire the required architecture views.

6. Develop Modeling Simulation to replicate architecture

a. Select Modeling tool best suited to meet evaluation requirements (i.e. Excel spreadsheet vs. discrete model simulation program)

b. Model structure to match architecture (OV-2, OV-5)

c. Model decision logic to match OV-6a.

d. Calculate MOE's at output of activities as functions of design parameters

7. Evaluate Model Completeness

Does model consider all relevant aspects (processes, assumptions, input variables and outputs, MOE's) of the system/concept?

a. IF so, continue to step 8.

b. IF model not complete, return to step 3 with the following considerations.

i. Determine additional architecture view and/or level of abstraction required to achieve traceability between system and the missing aspect.

ii. Develop required additional architecture

iii. Modify model to include additional architecture view.

iv. Re-evaluate Step 7 until model captures all relevant aspects of the concept.

8. Evaluate model for MOE results, requirements and key parameters

a. Once the model is complete, evaluate the system's ability to meet target metrics.

b. Vary design parameters and perform sensitivity analysis to identify key parameters.

c. Compare sensitivity analysis to target MOE's to establish requirements and KPPs.

d. Identify critical performance parameters in the SV-7 Systems Performance Parameters Matrix.

e. Vary system design and design parameters to evaluate the system's robustness and its rate of degradation.

2.4 Multi-Aircraft Systems Operations Concept

The current operations crew structure will be modified to accommodate MAC. The current operations crew consists of the pilot, the sensor operator, and the mission intelligence coordinator. The pilot is an Air Force officer and a rated pilot. The pilot controls the aircraft and commands the mission. The sensor operator is enlisted and controls the sensors on the aircraft. The sensor operator works directly with the pilot to accomplish the mission objectives. The mission intelligence coordinator is enlisted and interfaces directly with the intelligence community to coordinate on the essential elements of information. The mission intelligence coordinator reduces the amount of communication between the intelligence community and the pilot and sensor operator. Under MAC a single pilot will control multiple aircraft, but there is still a sensor operator and mission intelligence coordinator for each aircraft.

The operations concept for MAC is not formally defined, but current DoD doctrine addresses the need for a growth in UAS operations and the current and future requirements for UAS support. The *Air Force Flight Plan* lays out the challenges and drivers that are spurring a movement towards multi-aircraft control. The demand is increasing for highly capable airborne platforms able to conduct the entire F2T2EA chain. UASs are an effective and economical means to satisfy this user need, particularly once air superiority is well established. Multi-aircraft control has the potential to significantly reduce pilot manpower requirements in fielding UASs on the battlefield. (USAF, 2009)

The *Army-Air Force Enabling Concept* does not cover the topic of multi-aircraft control directly. While this document identifies the challenges that are spurring multi-aircraft control, it does not identify a solution to these challenges. This document is focused on the effect brought to the battlefield by multi-role UASs. By focusing on the effect that needs to be delivered by UASs on the battlefield this document effectively sets the goal for the capability that UAS need to be able to accomplish under multi-aircraft control. MAC has the potential to increase the number of UASs on the battlefield without increasing pilot manpower and perhaps even decreasing manpower, but the joint warfighter still requires a highly capable multi-role UAS for performing the entire F2T2EA process (USAF & USA, 2009). The implementation of MAC will be measured against the ability of the UAS to meet the demands of the joint warfighter to provide the complete F2T2EA process.

Seibert et al (2010) focused on a multi-aircraft control using a modified RQ-11A Raven UAS. This AFIT thesis addressed the employment of multiple small UAVs for the performance of ISR. The authors explored the use of relay UAVs to extend the line-of-sight control range. The authors used discrete event simulation to optimize the performance of a single operator performing launch, recovery, aircraft control, and sensor operation. This thesis also addressed the Human Computer Interface (HCI) of multi-aircraft control for small, short range UAVs (Seibert, Stryker, Ward, & Wellbaum, 2010).

2.5 Workload as a Measure of Performance

It is difficult to predict a pilot's ability to manage the F2T2EA process without in-depth testing, but workload predictions can be used to highlight critical factors that may cause the pilot to become over-tasked and to inform more focused evaluations(D. K. Mitchell & McDowell, 2008). Workload, specifically mental workload, remains a challenge to fully define. Operationally, favorable workload conditions have been characterized as "a situation in which the operator feels comfortable and can manage task demands intelligently, and maintain good performance" (Hart, 1991). While this definition qualitatively describes the desired condition it lacks any indication of an evaluation procedure. More quantitatively, Young and Stanton propose that:

> "The mental workload of a task represents the level of attentional resources required to meet both objective and subjective performance criteria, which may be mediated by task demands, external support, and past experience." (Young & Stanton, 2001)

This definition contains definitions of workload from Stanton (2005) and Miller (2003) and has the four key pieces common to definitions of mental workload(Stanton, Salmon, Walker, Baber, & Jenkins, 2005)(Miller, Crowson, & Narkevicius, 2003):

> "(1) imposed task demands – if the difficulty, number, rate, or complexity of the demands imposed on an operator are increased, workload is assumed to increase; (2) the level of performance an operator is able to

> achieve – if errors increase or control precision degrades, workload is assumed to increase; (3) the mental and physical effort an operator exerts – workload reflects an operator's response to a task, rather than task demands directly; and (4) an operator's perceptions – if an operator feels effortful and loaded, then workload has, in fact, increased even though task demands or performance have not changed." (Huey & Wickens, 1993)

For the purpose of this analysis workload will correspond to the imposed task demands and effort (in the form of conflict workload); the operator's level of performance and perceptions are not addressed directly, but will be discussed during data analysis as their effects on mission effectiveness will be examined. Based on these definitions it is clear that the increase in workload, through the addition of multiple vehicles, can degrade performance as the pilot reaches cognitive saturation. The Yerkes-Dodson Law correlates psychological arousal with performance of complex tasks as an inverted "U" curve. At low arousal levels performance is poor and increases with increases in arousal to the optimal point after which the subject is over stimulated and performance is reduced as arousal increases (Yerkes, 1908). Mental workload has the same effect as psychological arousal so as the workload increases past the optimal point, performance is degraded (C. Wickens, 2003).

In an effort to study workload, a front end analysis was performed by the Survivability/Vulnerability Information Analysis Center (SURVIAC) on current traditional operations (Eaton et al., 2006). This “Optimizing Human Performance™ Front End Analysis (FEA) methodology” used interviews with pilots and sensor operators along with operational and training observations to study MQ-1B tasks.

The result is a detailed, quantitative, and qualitative, set of task lists, sequences, times, and observations. These data were collected with the aim of forming the basis of a workload study.

Step six of ABEP develops a simulation to replicate the architecture; since workload is the dependent variable, a simulation environment which incorporates methods of workload calculation is desirable. The Army Research Laboratory, Human Research and Engineering Directorate, has developed IMPRINT, which is a computer based, discrete event simulation platform, with integrated mental workload calculation based on Multiple Resource Theory (MRT)(D. K. Mitchell, 2000). As a predictive theory, MRT proposes that four mental dimensions or channels are available to process information and perform tasks. These channels are allocated to concurrent tasks with the difficulty of the tasks and the demand conflict between channels driving the overall mental workload value (C. D. Wickens, 2008). The channel values for a given task are based on the McCracken and Aldrich Workload Demand Values, an accepted and validated scale ranging from 0.0 to 7.0 (McCracken & Aldrich, 1984).

Due to the concurrent nature of tasks imposed on an MQ-1B pilot, navigating while communicating, and monitoring, MRT is an appropriate theory for this application. Other theories predict mental workload: Single Channel Theory (SCT), Single Resource Theory (SRT), and Visual, Auditory, Cognitive, and Perceptual (VACP). However, a study comparing MRT to SCT and SRT mental workload predictions in the domain of UAV control both conventional and MAC was performed for the Army to evaluate the effects of auditory response and task automation on the performance of single operator UASs. (Dixon, Wickens, & Chang, 2005). MRT correctly predicted a performance increase observed in human testing which was not predicted by either SCT or SRT. MRT has many similarities to VACP, but further differentiates between listening and speaking. MRT also has a conflict workload concept lacking in VACP which improves the fidelity of the model.

Two recently developed workload prediction theories potentially increase the fidelity of workload estimations. The Malleable Attentional Resource Theory (MART) was proposed by Young and Stanton and differs in assumption regarding the workload capacity of the operator (Young & Stanton, 2002). In contrast to MRT which assumes resource channel capacity is fixed, MART asserts that the resource capacities vary with respect to demand such that at low workload demand performance is degraded and at high workload capacity may expand beyond nominal capacity before performance is degraded. The effects explained by MART are similar to those of the observed vigilance decrement (Parasuraman & Rovira, 2005). While MRT addresses three of the components of the workload definition, operator perception is unaccounted for. A dynamic workload model which incorporated the operator's perception stipulated that workload is a vector of three dimensions: time to act, perceived distance till goal completion, and the effort required to accomplish the goal (Hancock & Caird, 1993). This view increases mental workload as the time to act is constrained and the time till goal completion increases. MRT can be used to calculate the effort required to accomplish the goal. While time to act is contextually dependent, a task analysis can provide the necessary data. However, the pervieced distance to completion remains difficult to determine and in the context of complex MQ-1B piloting tasks is a level of fidelity beyond this analysis. Validation of these two theories is ongoing and they lack the wide spread acceptance and validation of MRT. The increased fidelity and pedigree offered by MRT as a predictor of mental workload for complex tasks and interfaces makes it appropriate for this analysis.

IMPRINT has been used successfully for many years by the DoD to model future systems and to explore function allocation and manpower levels through workload and human performance modeling (D. K. Mitchell & Samms, 2009)(D. K. Mitchell, 2003). Extensive IMPRINT modeling was performed on the Army's, now cancelled, Future Combat System to integrate unmanned air and ground vehicles into the operational force. One report, similar to the analysis performed here, details the modeling and testing efforts to integrate multiple small UAVs into a unit using VACP and appropriate overload conditions. The findings indicate that overload increases with increased number of aircraft and while the visual and cognitive channels were overloaded substantially more at two aircraft, overall overload did not spike until the operator controlled three aircraft. (Pomranky & Wojciechowski, 2007)

IMPRINT Pro is the current software platform and models workload as a calculation during a discrete event, task-based simulation. Since the SURVIAC FEA provides a task network to model with workload values drawn from MRT, a dynamic, stochastic, simulation platform, like IMPRINT Pro, can be used to analyze the increased workload as a function of the number of aircraft that are simultaneously controlled. Assumptions regarding the current location on the Yerkes-Dodson curve provide the ability to predict suitability and to highlight conditions which result in high workload, and are likely to reduce pilot performance in a MAC condition.

2.6 Architectural Views

Addressing the role of the human in the system is a critical part of system design. Humans have a complex and crucial role in the system that needs to be captured in the system architecture, but DoDAF does not sufficiently capture all of the implications of human factors. With some improvements, DoDAF can effectively capture the complex interconnected nature of human factors considerations in systems architecting (Hardman, Colombi, Jacques, & Miller, 2008), as other architectural frameworks have accomplished.

For example, Human Views (HVs) were developed to add human factors considerations to the Ministry of Defense Architecture Framework (MODAF), which is based on DoDAF 1.0. The Human View Handbook for MODAF (2009) introduces the topic of HVs and provides a structure for the various human views and their relationship with existing MODAF architecture views. Seven HVs have been proposed: HV-A: Personnel Availability, HV-B: Quality Objectives and Metrics, HV-C: Human Interaction Structure, HV-D: Organization, HV-E Human Functions and Tasks, HV-F: Roles and Competencies, and HV-G: Dynamic Drivers of Human Behavior (Systems Engineering & Assessment Ltd, 2009). The HVs capture the requirements for human operators and traces how the human influences the design of the system (Handly & Houston, 2010). The information from “To Be” DoDAF architectures for Predator operations can be merged with HV architectures to identify the interfaces of the pilot with the system and other human roles (MITRE, 2009).

A methodology was developed to use the HVs to develop a simulation in the IMPRINT (Handly & Smillie,). This process provides a direct tie between the human factors architecture and a predictive simulation tool enabling systems engineers to verify architecture and analyze the effects of changes to system design. The process for using HVs to create a model in IMPRINT is given in Table 1.

Table 1. Process for Creating an IMPRINT Model from Human View Architectures (Handly & Smillie,)

STEP	IMPRINT MODEL	HUMAN VIEW DATA
1	Operators	HV-D Roles
2	Mission Network Diagram	HV-C Tasks
3	Warfighter Assignment	HV-D Task-Role Matrix
4	Resource-Interface (RI) Pairs	HV-C System Interfaces

5	Task Time and Accuracy and Task Effects	HV-G Performance Standards/ Measures
6	Performance Moderators	HV-B Constraints
OUTPUTS	Mission Results Task Performance Operator Workload	HV-B Constraints HV-G HV-G / HV-B

III. Methodology

3.1 ABEP Application in MAC Analysis

The ABEP process was used as the framework for this analysis to identify and characterize the critical factors impacting the implementation of MAC. Each step of the ABEP process is addressed below with its application and variation for this research. This process provides a strong foundation on which to base the development of the workload model.

3.1.1 Design operations concept of system to be evaluated

As described in Section 2.3, the concept operations for UAS operations is well established. The addition of MAC to UAS operations should be completely transparent to the allied units that interface with the MQ-1B so the existing concept of operations should be utilized for this analysis. This research intentionally avoided developing concepts of operations for applying workload mitigation strategies to address excessive workload or handing off aircraft to on-call pilots during times when a single pilot cannot manage the workload. This research is meant to identify the critical factors associated with MAC and not verify a particular workload mitigation strategy. Preliminary experimentation with workload mitigation strategies indicated that these techniques effectively obscured the workload imposed by the system and did not facilitate the analysis of critical factors. Further the development and optimization of workload mitigation strategies was beyond the scope of the present thesis.

3.1.2 Identify Measures of Effectiveness (MOE) relevant to the decision/evaluation

Section 2.4 presents background on this step of ABEP. To identify and characterize the critical factors of MAC, the ability of the pilot to maintain current system effectiveness while controlling multiple aircraft is estimated, not the effectiveness of the UAS. Instead of an MOE that relates to mission accomplishment, this analysis will use pilot workload to infer the ability of the pilot to maintain system effectiveness under MAC scenarios. Some saturation threshold value that indicates excessive workload, and thus a point at which the mission effectiveness is impacted, must be established in order to effectively use workload as an MOE. Workload is a subjective measure with no units associated with it. A saturation threshold value beyond which pilot performance will be assumed to be degraded will be established as part of model validation of single aircraft operation.

3.1.3 Identify required level of abstraction for architecture to show traceability to MOEs

The MOE must be evaluated from a perspective that is within the system since it evaluates the workload imposed on the pilot by the rest of the UAS. The interfaces and interactions of the pilot with the rest of the system will need to be modeled as well as communication events occurring between the pilot and other actors external to the system. The workload generated from within the system will need to be combined with workload generated from outside of the system to capture the conflict that it generates. The MOE only addresses with the pilot workload so interfaces and tasks that do not directly affect the pilot can be disregarded for this analysis.

3.1.4 Identify architecture views necessary to capture structure/relationships

The "to be" OV-1 High Level Operations Concept, along with the OV-2 Node Connectivity Diagram for UAS operations, forms the basic structure for the analysis of pilot workload (MITRE, 2009). To accurately capture the pilot's interfaces with the rest of the system, the information from these architecture views will need to be placed in an HV-C Human Interaction Structure. The HV-C captures the critical elements from the existing DoDAF architecture and presents them in an anthropocentric fashion. The architecture was created to view the communication paths used by a MQ-1B Predator pilot and to represent the interface with the Predator UAS. The objective was not to represent a specific control layout, but to capture potential factors influencing pilot workload. An HV-E is necessary to turn the pilot's job performance into a series of executable tasks. These tasks are needed to generate model tasks and functions in IMPRINT along with the proper sequencing. Finally, an HV-G Dynamic Drivers of Human Behaviors is used to capture quantitative and qualitative aspects of each of the individual tasks so they can be effectively represented in the model. This view will provide task length and difficulty as required in IMPRINT.

3.1.5 Develop Architecture Views

The UAS pilot architecture is developed in detail in Section 3.2. This architecture is the basis for the IMPRINT Pro workload model that is the core of this research.

3.1.6 Develop modeling simulation to replicate architecture

The IMPRINT model development is described in detail in Section 3.4. The model was developed by tying the human view architecture directly to IMPRINT model elements (Handly & Smillie,).

3.1.7 Evaluate model completeness

The IMPRINT model was evaluated in Section 3.5 for its ability to meet pilot task assessments in Section 3.3 and accurately reflect the architectural views in Section 3.2.

3.1.8 Evaluate model for MOE results, requirements, and key parameters

Chapter 4, the Analysis and Results, examines the model output data and evaluates it based on mission parameters and the redline saturation threshold established for evaluating the MOE. Critical factors that potentially affect the MQ-1B pilot's performance and their ability to adequately perform the mission under MAC can be found from this data.

3.2 UAS Operations Architectural Views

The starting point of the ABEP analysis is the system architecture for UAS operations. Multiple system level views exist for UAS operations, but they do not effectively and concisely represent the interactions and functions of the pilot as part of the system. The existing architecture will be the basis for the development of human views that are constructed around the pilot's interfaces and roles in the system. The first architectural view to be addressed is the constrained "to be" OV-1 High Level Operational Concept found in Figure 6.

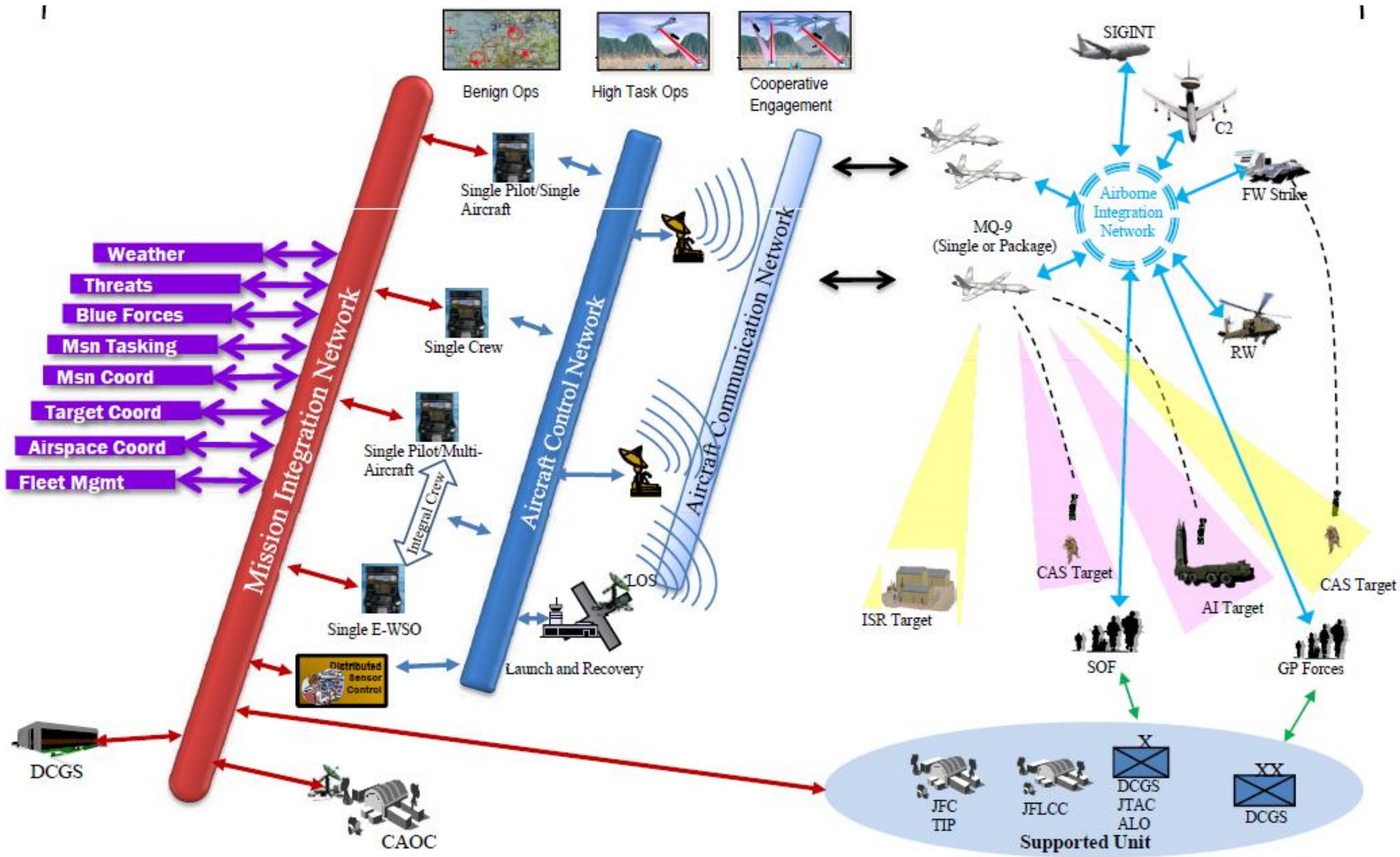


Figure 6. OV-1 UAS High Level Operations Concept

The OV-1 depicts the high level inputs and interactions that an MQ-1B crew has for a mission. The Mission Integration Network delivers information to the crew from supported units in the combat area, the Combined Air Operations Center (CAOC), and the Distributed Common Ground Station (DCGS).

Through this network the crew must assimilate information on weather, threats, blue forces, mission tasking, mission coordination, target coordination, airspace coordination, and fleet management. In addition to all of those interactions and inputs, interactions and inputs also occur through the Aircraft Control Network. With the interactions necessary to control the aircraft, the aircrew also interacts with all of the allied aircraft sharing the airspace and any allied units on the ground that may be in direct communication with MQ-1B. As can be seen in this OV-1, the control of the aircraft comprises only a small portion of the interactions to which the MQ-1**B** crew must attend. This architecture involves multiple levels of control and communication that must be managed and synchronized to facilitate mission execution.

The OV-2, Operational Node Connectivity Diagram, and OV-3, Information Exchange Matrix, are not reproduced here due to the large size of these architecture views. However, both of these views will be discussed here because they provide inputs into the HV-C Human Interaction structure. The OV-2 for UAS operations contains major nodes for the Combined Air Operations Center CAOC, Weather Operations Center (WOC), Launch and Recovery Element (LRE) Base, Squadron #1, Supported Unit, Intel Exploitation, Area of Responsibility (AOR) Air Traffic Control (ATC), and Joint

Airspace Player. The MQ-1B mission pilot is part of the Mission Crew within the Squadron #1 Primary node. The pilot has at least one connection to each of the other major nodes in the OV-2 and in some cases multiple connections to different elements within the primary node. The OV-2 does not provide the level of information required to begin to break down the complexity of these interactions, but it does provide the framework necessary to begin to characterize the human interactions within the system. To determine the specific information that is passed between the pilot and these other nodes, the analysis needs to include the OV-3. In the OV-3 the MQ-1B pilot is the originator node of 45 information events and the mission crew is the originator node of 16 communication events such as establish clearance and route of flight, target confirmation, and provide damage assessment. The pilot is also the receiving node of 39 information events and the mission crew is the receiving node of 20 information events such as receive target prioritization, intelligence data on target and essential elements of information, and receive mission area weather forecast. This demonstrates the complexity and the volume of interactions that the MQ-1B pilot has within the UAS operations system. Clearly information exchange is a very significant part of the UAS operations concept and must be adequately represented. The MQ-1B pilot is not only responsible for the control of the aircraft; they are also critical members in a multi-path communications infrastructure (MITRE, 2009).

The HV-C Human Interaction Structure in Figure 7 synthesizes the information from the OV-1 and OV-2 into a human-focused view that centers on the MQ-1B pilot and pilot interactions. This permits the pertinent information for this analysis to be collected and presented in a single comprehensive view. The link between the pilot and the MQ-1B is not a direct path; instead it must pass through the controls and displays of the GCS. Technically the pilot does not interact with the MQ-1B and only directly interacts with the GCS, but the MQ-1B is represented in this view since representing the aircraft is necessary to maintain the focus of the analysis.

The interactions become more complicated on the communications side of the HV-C. The pilot has multiple means of communication with multiple actors in multiple nodes. The pilot interacts primarily with the other two members of the crew, the mission coordinator and the sensor operator, over the GCS intercom. The intercom can also be used to interact with the operations supervisor and the mission intelligence coordinator. A large amount of the pilot's interactions are over the intercom with the sensor operator and the mission coordinator. These two team members can potentially reduce the communications workload on the pilot by handling much of the communication load. The rest of the pilot's communications are through one of multiple chat windows and radio systems. The pilot must communicate with the Launch and Recovery Element for handoff of the aircraft, the WOC for AOR weather, the supported units' operations, intelligence, and maneuver units, intelligence exploitation, air traffic control within the AOR, joint airspace players, and the CAOC. The HV-C brings together the interactions and systems relevant to the MQ-1B pilot in a straightforward way that aides in system design decisions.

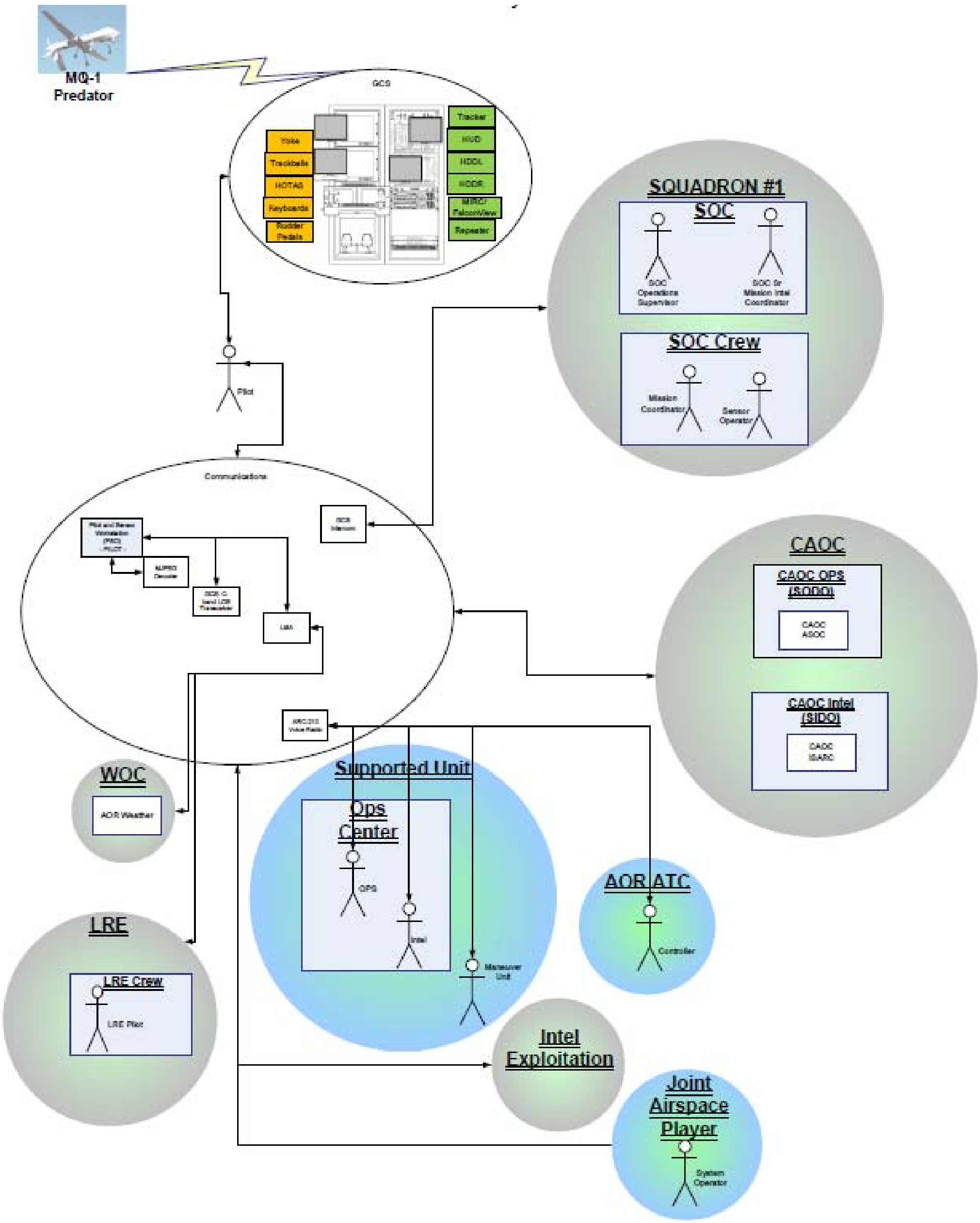


Figure 7. HV-C Human Interaction Structure for UAS Operations

The SURVIAC Front End Analysis (FEA) heavily informed the HV-E Human Functions and Tasks and the HV-G Dynamic Drivers of Human Behavior. The FEA breaks down the pilot's workload into a discrete hierarchical task list that covers the entire range of operations. The HV-E, Figure 9, is similar to the flow chart from the FEA in Figure 8, but with some necessary modifications. The FEA flow chart, Figure 8, represents both the launch and recovery element and the primary MQ-1B pilot actions; consequently the portions that were outside of our scope were removed. The HV-E only represents primary tasks and does not break them into subtasks. Also the FEA flowchart had multiple logical inconsistencies that had to be corrected for the HV-E. (Eaton et al., 2006)

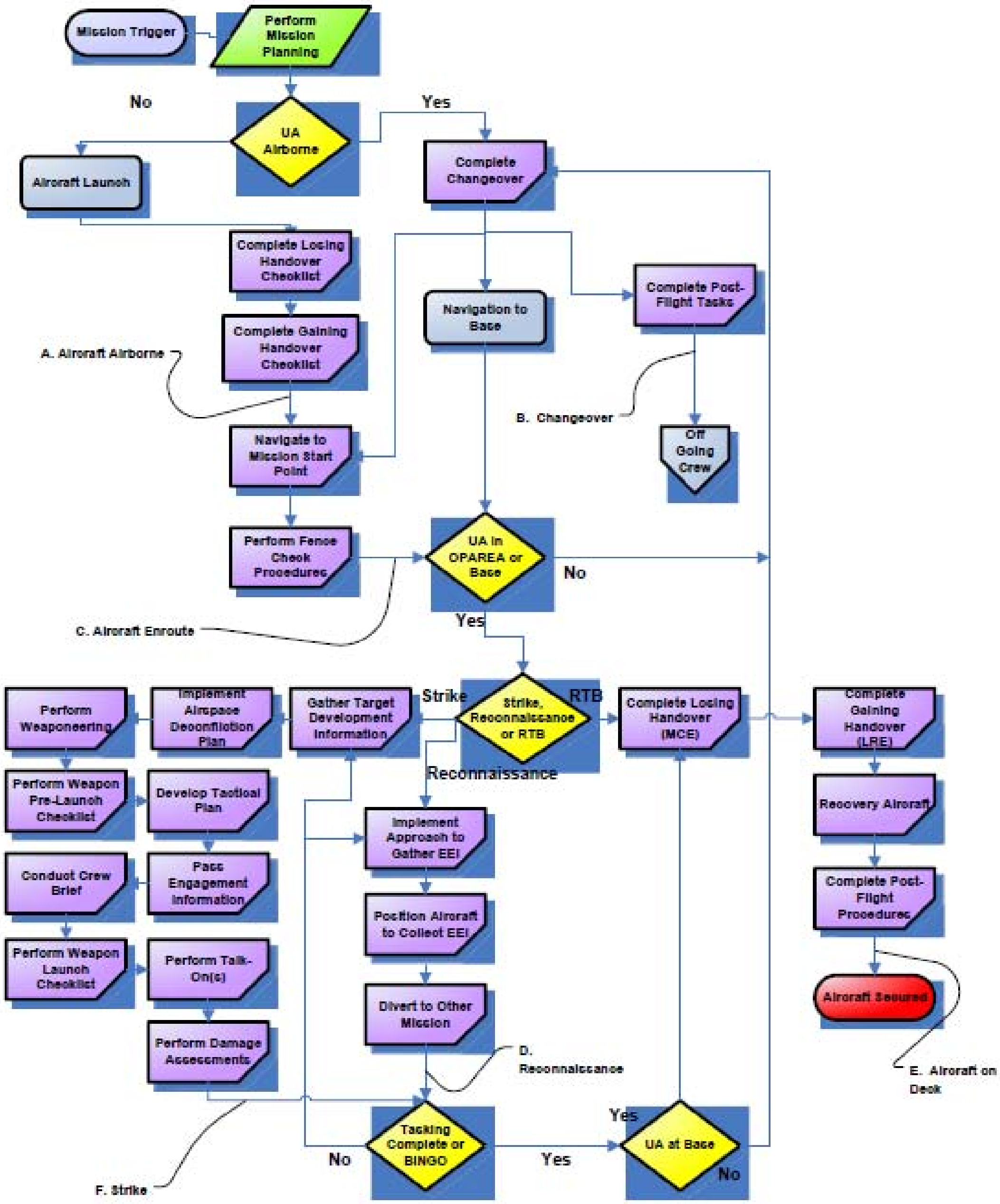


Figure 8. SURVIAC FEA Flow Chart of MQ-1B Pilot Tasks (Eaton et al., 2006)

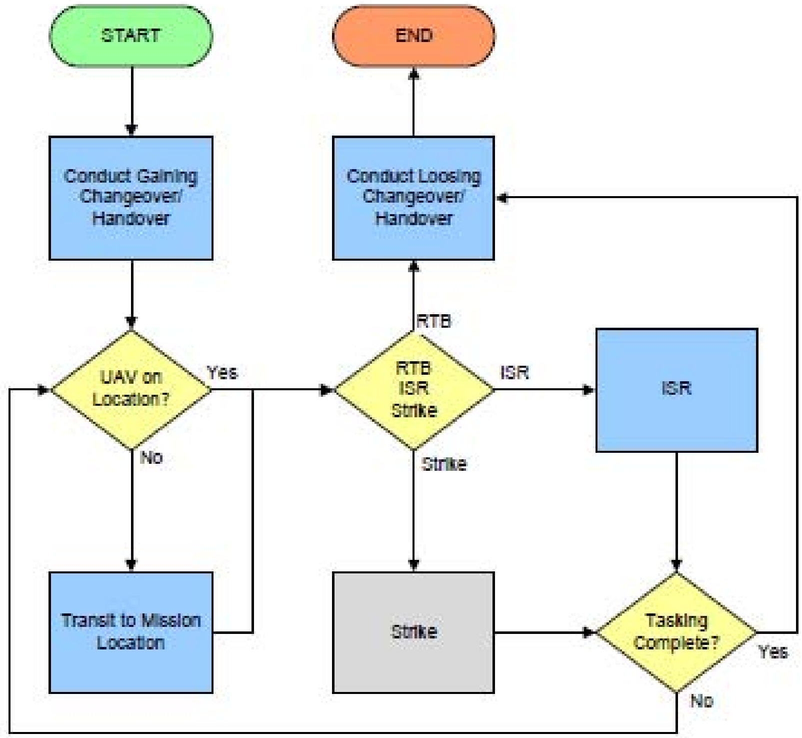


Figure 9. HV-E Human Functions and Tasks of MQ-1B Pilot

The FEA Flowchart does not clearly depict who performs the tasks in the flow chart. The changeover and handover tasks change aircraft control between two pilots, but there is not any indication of this change in responsibility in the flowchart.

The handover is the transfer of control between the launch and recovery element and the mission element. These two crews are in separate GCSs. The changeover occurs when a pilot replaces another pilot in the same GCS when their shift is complete. The changeover and handover are the first events that are relevant to the analysis and are the first tasks in the HV-E. The FEA Flowchart begins with a mission trigger leading into mission planning and a check to determine if the aircraft is airborne. If the aircraft is not airborne, it is launched by the launch and recovery element and then the aircraft is handed over to the mission crew. If the aircraft is airborne it triggers a changeover, which is one of the logical inconsistencies. A changeover only occurs when a new pilot replaces another pilot when their shift is over. Depending on whether a changeover or handover is completed, the aircraft is navigated to base or the mission area. Both of these are transit tasks and the tasks performed by the pilot are identical during each task so they are both included in the transit task in the HV-E. The HV-E routes to the transit task anytime the aircraft is not at the desired location, which simplifies the architecture and removes some redundancy in the FEA flowchart. The FEA flowchart also had a redundant decision block after navigation to base or the mission start point. After that decision point the FEA flowchart routes into a decision to do strike, reconnaissance, or return to base. The HV-E has a very similar decision point to do strike, ISR, or Return To Base (RTB). The HV-E does not explicitly breakout all of the subtasks associated with the major tasks. The FEA flowchart has another logical inconsistency in the strike and reconnaissance subtasks. These tasks are not sequential as indicated in the flow chart. Some of them are performed concurrently and others are subtasks of other tasks in the sequence. The reconnaissance branch always ends with the task Divert to Other Mission, but a “divert” is a task that should interrupt the normal flow of the mission and not be a sequential part of the mission. The HV-E borrows much of the task information from the FEA flowchart, but also simplifies the information from the flowchart and corrects the logical errors.

The HV-G is a matrix of all specific task related data. The task names and descriptions come from the FEA while task times are derived from discussions that were held with experienced pilots. The HV-G is a repository of the data that was collected on each of these tasks and serves as a primary source of task data for the model and analysis. The HV-G does not create any new or unique data, rather it is a view that concisely collects all of the necessary data in one place in a format that in conducive to model creation.

3.3 MAC Model Development

The model is developed from the perspective of determining the workload the system imposes on the pilot during a 2-3 hour shift. The model does not consider workload mitigation strategies that the pilot may employ such as task delaying or task offloading. Further the model does not consider effects of task success or failure. Instead, the model strives to predict the workload imposed by operational tasks, assuming that the system requires all tasks to be performed as they are imposed on the operator. A sample of the raw data output of the model is in Appendix E.

The model is composed of three essential elements: functions, tasks, and artifacts. Functions, depicted as gray boxes, are a method of grouping tasks in IMPRINT Pro to permit cleaner layout and aid model comprehension. This model uses functions to group communication tasks, specific aircraft tasks, and mission module tasks. A task is the most basic element of the model and has an associated time and workload. These tasks drive the model and the model produces output workload value in response to the presence of a task. Artifacts are tasks which have no workload associated with them, are used to run the model, and are performed by an automated agent. Some artifacts have associated times that represent actual times within the domain, delay times, mission times, etc. All "START" and "END" tasks are artifacts necessary to run the model. Much of the model logic is contained in the artifacts.

The high level model layout is depicted in Figure 10. The pilot's tasks are replicated within Function 1 "AC1", 10 "AC2", 11 "AC3", and 12 "AC4" with the exception of the communication tasks which are all in a centralized location in Function 8 "Communicate".

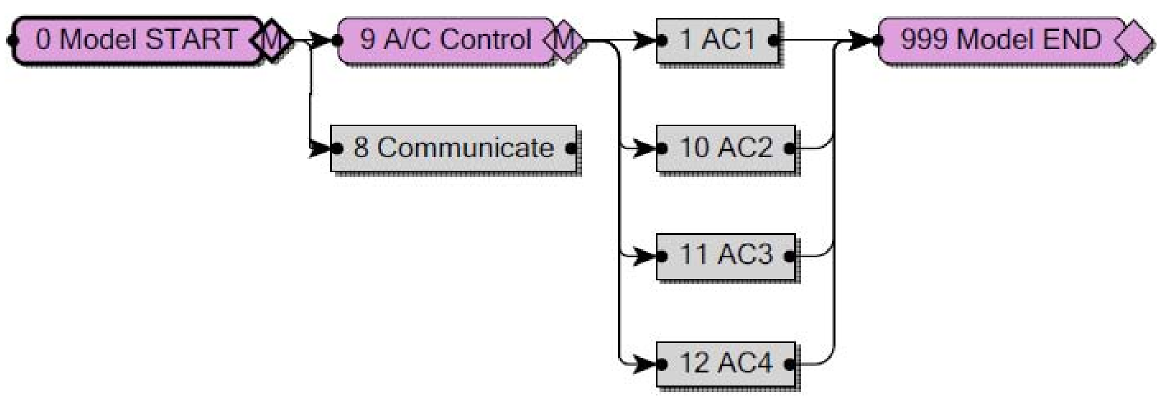


Figure 10. Top-Level MAC IMPRINT Model Layout

Task 9 "A/C Control" is a modeling artifact which controls how many aircraft are under the pilot's control and when the pilot takes control of each aircraft. Figure 12 depicts the layout of each aircraft function. Each aircraft function is identical to every other aircraft except for the tail number, which uniquely represents each aircraft under the pilot's control during MAC.

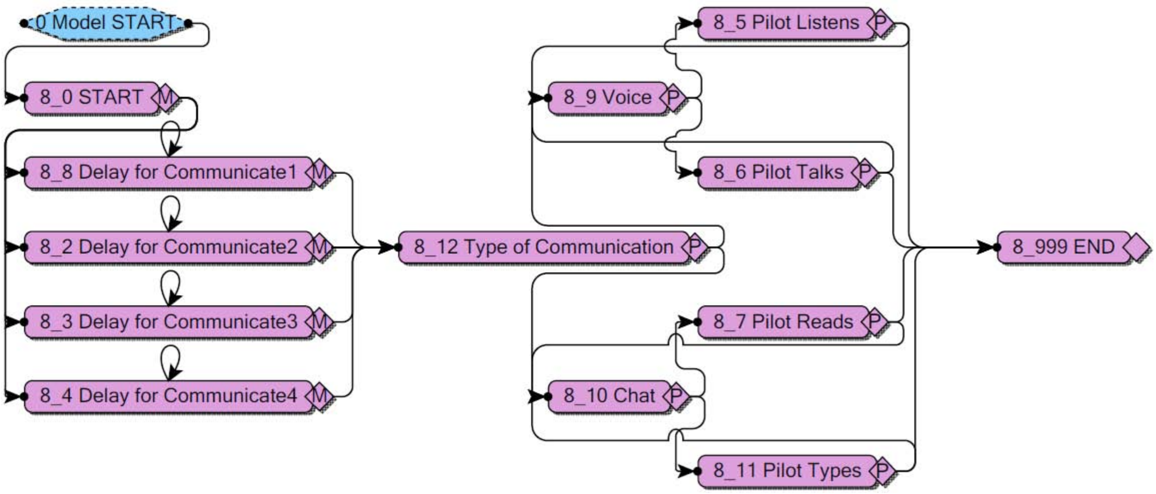


Figure 11. Communicate Function from MAC IMPRINT Model

Function 8 "Communicate", (Figure 11) operates from an event generator for each aircraft; which is triggered when the pilot assumes control of an aircraft. The event generator artifacts, Tasks 8_2 through 8_4 and 8_8, operate continuously with delay times based on exponential distributions specified by the mission module of each aircraft. These events flow into the generic communication tasks on the right side of Figure 11. This arrangement replicates the stochastic nature of communication and the increase in frequency during different phases of the mission. Based on discussions with experienced MQ-1B Predator pilots, chat is the most frequent type of communication during most mission phases. Therefore, the communications module is set up probabilistically 25/75 voice/chat. The direction of communication, incoming or outgoing, is split evenly between listening and talking on voice and 90/10 read/type on chat. After the pilot has listened, talked, read or typed, there is an increased probability that this communication event will result in a complementary communication event through the same medium rather than simply exiting the communication module. For example if a pilot listens, there is an increased probability that this will trigger another listen or speaking event. The construction of this module replicates the conversational nature of communication in which a pilot listening to an allied unit may respond verbally, and a pilot reading text based chat communication may respond by returning a text message.

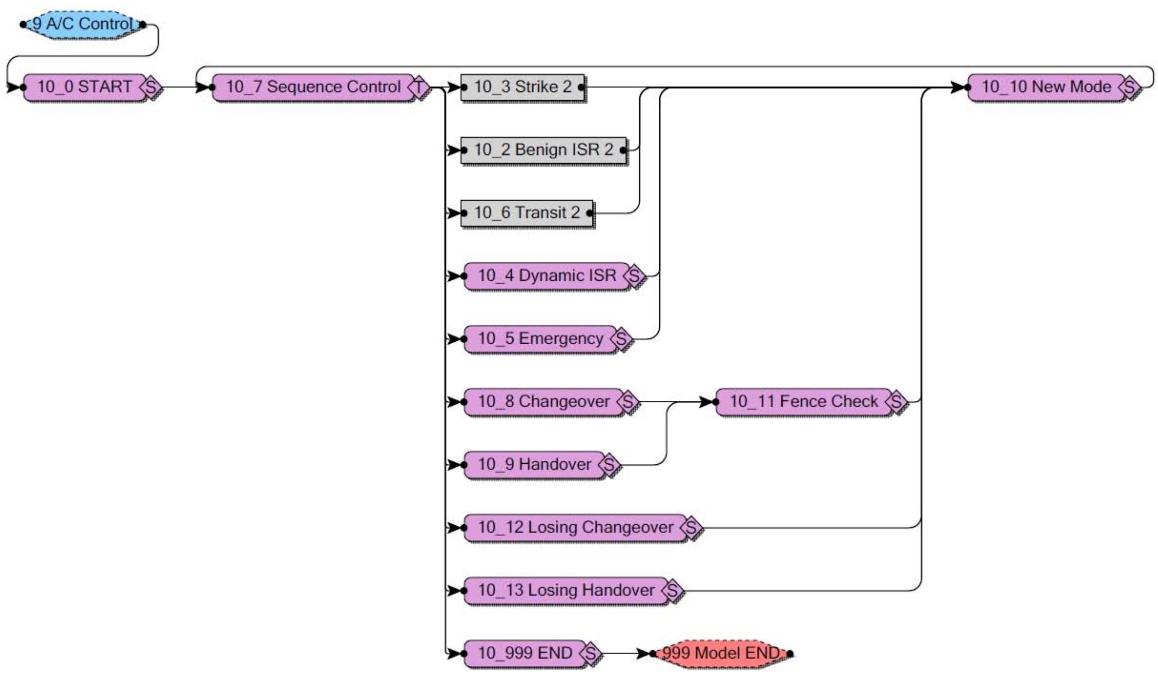


Figure 12. Aircraft Mission Layout from MAC IMPRINT Model

Each aircraft has an identical set of mission segments which can be performed as depicted in Figure 12. Tasks 10_7 "Sequence Control" and 10_10 "New Mode" are modeling artifacts which determine the mission module the aircraft will enter next. Blocks 10_2 through 10_6, 10_8, 10_9, and 10_11 through 10_13, are mission modules, which produce workload and control the length of time the aircraft is in a given mission module. Each module is composed of one or more tasks which model the workload on the pilot for a specific length of time. Performed in sequence as an operational profile they form the basis of the workload output. These are the only tasks outside of communication which produce workload. The SURVIAC Front End Analysis serves as a basis for each module (Eaton et al., 2006).

"Changeover" and "Handover" are continuous, single task, events during which the pilot assumes or relinquishes control of an aircraft. Changeover is when a pilot switches control with another pilot in the same Ground Control Station (GCS). Handover is when a pilot relinquishes or assumes control of an aircraft with another pilot in a different Ground Control Station (GCS). A Handover is when a Mission Control Element (MCE) pilot transfers control with a Launch and Recovery Element (LRE) pilot. Due to the substantial endurance of the aircraft, changeovers are far more frequent than handovers. Fence Check is a task initially relegated to transit in the Front End Analysis, but after consulting with experienced MQ-1B Predator pilots it seemed more appropriate to place it after gaining Changeover and Handover where it is more frequently performed. TheMQ-1B Predator pilots also differentiated between gaining and losing activities in task length. Thus Losing Changeover and Handover are separate tasks with different workload and task times than gaining operations.

"Dynamic ISR" and "Emergency" are continuous single tasks which represent periods of increased activity. Following a vehicle leaving a compound or providing overwatch to a firefight are examples of "Dynamic" ISR. MQ-1B Predator Pilots agree that these mission modes require total continuous attention and are more demanding than other segments. They also experience the most frequent communication events during these mission modes.

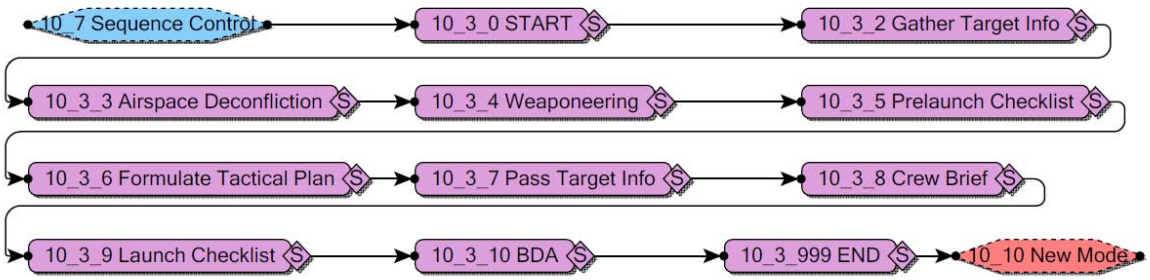


Figure 13. Strike Mission Module from MAC IMPRINT Model

"Strike", Figure 13, is based directly on the Front End Analysis and is a sequential processing of tasks. However, MQ-1B Predator pilots noted that there is substantial overlap, parallel processing, and long lead preparation that complicate discrete event simulation. The method of performing those tasks is variable among individuals and circumstances and was not studied in depth.

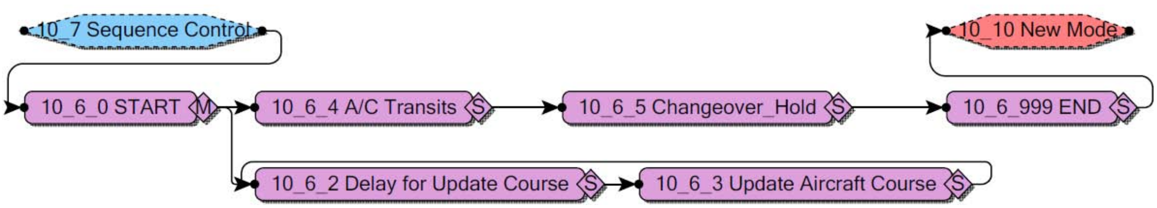


Figure 14. Transit Mission Module from MAC IMPRINT Model

The Transit mission module (Figure 14) contains a single task with associated workload, 10_6_3 "Update Aircraft Course." Update Aircraft Course is iterated through a Delay artifact which simulates the variable nature of transit navigation. When the pilot inputs a navigational course, the UAV performs the necessary aviating tasks to fly the course. Otherwise, the system imposes no other tasks on the pilot, thus the iterated task loop. Task 10_6_4 “A/C Transits” is a modeling artifact which represents the total transit length. The “Changeover_Hold” artifact will be described later.

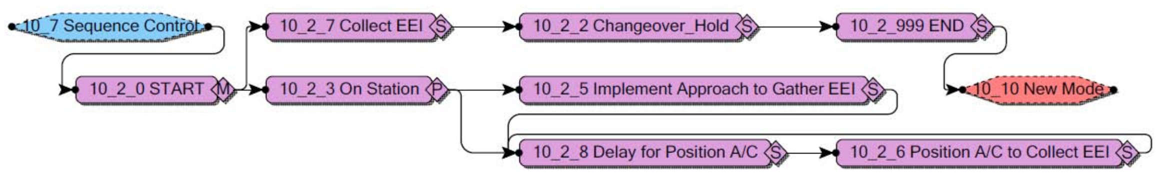


Figure 15. Benign ISR Mission Module from MAC IMPRINT Model

Benign ISR (Figure 15) is composed of two primary tasks 10_2_5 “Implement Approach to Gather EEI” and 10_2_6 “Position A/C to Collect EEI”. EEI in this context stands for Essential Elements of Information which could be video, pictures, or signal intelligence depending on the mission. When the aircraft arrives at the location (on station) to collect the information, the pilot performs the “Implement Approach” task. However, the endurance allows for the possibility that a pilot is taking over an aircraft that is already on station and does not need to implement an approach, this is represented by the probabilistic routing of 10_2_3 “On Station” artifact. In either case, these activities start both the positioning loop and the general “Collect EEI” artifact. Similar to the transit module, the pilot interacts with the aircraft as necessary, through the “Position A/C to Collect EEI,” to maintain orientation for the sensor operator. The same task loop architecture as in the transit module is used. The “Collect EEI” artifact performs the same role as “A/C Transits” and represents the amount of time the aircraft is on station.

To accurately represent multiple changeover events, as would happen when a pilot took control of several aircraft at the beginning of a shift, the model logic structures these events to occur sequentially before any other mission tasks are performed. Similarly at the end of a shift these changeover tasks would be performed sequentially as the outgoing pilot briefs the incoming pilot. So in the benign ISR and transit modules the "Changeover_Hold" artifact only releases the entity when all aircraft are prepared for changeover. If none of the aircraft changeover, this is not used.

Finally, the analysis architecture of this model is housed in several macros. These allow the analyst to control when each aircraft arrives, the sequence and times of mission modules for each aircraft, and module communication frequency distributions. This information is executed in artifacts like "A/C Control" and "Sequence Control" as well as the time keeper artifacts, "Collect EEI" and "A/C Transits". "A/C Control" stages when the aircraft are released to the pilot. The aircraft can be released to the pilot simultaneously at the beginning of a shift, or staggered over the course of a shift. Alternately, the effects of a handover in the middle of an operation could be studied by releasing one aircraft later in the shift. "Sequence Control" in each aircraft function reads the script for each aircraft and routes it to the appropriate mission module. Time keeper artifacts have a duration based on the desired stochastic distribution for each module. Thus to run a particular scenario, an analyst modifies the script in the macro to set the model parameters and then runs the model. The code for the model macros is in Appendix G.

3.4 MAC Model & Concept Validation

Models approximate reality, and the closer the approximation is to reality the more useful the model becomes. Validation for the MAC workload model was informal in approach due to the size and scope of the project. Informal validation is appropriate to preliminary studies and has been used for many similar HPM efforts (Wong, 2010). The DoD Modeling and Simulation Coordination Office (M&SCO) lists recommended practices for informal validation; these include Desk Checking, Face Validation, Reviews, and Walkthroughs (Modeling and Simulation Coordination Office (M&S CO), 2006).

The replication of the pilot tasks in the model required an in-depth desk checking process which examined each task in the model to ensure that the parameters (times, model logic, variable references, etc.) were correct. This culminated in a series of test runs to ensure the model output reflected the inputs and model logic flow. To verify that the model ran as expected, these runs were scrutinized at the task execution level to observe start and end times of each task, and task overlap and failure to execute. This desk check process was repeated for each model iteration throughout development. These iterations were also subject to walkthroughs with the committee to ensure modeling techniques and logic was appropriate to the model.

Early in development the architecture of the model was codified as a framework for the modeling effort. The scope and perspective of the project were also agreed upon early in the project limiting the model to the tasks and workload of the pilot. Periodic reviews with Subject Matter Experts (SMEs) and advisors ensured that the model architecture and scope were appropriate. These reviews led to both model and modeling changes to more accurately approximate the operational reality of multi aircraft control.

In early November 2010, towards the end of model development, the input and execution parameters of the model were scrutinized by ten experienced MQ-1B Predator pilots of the 119th Air National Guard Wing in Fargo, ND. This included model flow, task times, frequencies of iterated tasks, and difficulties of tasks and mission modules.(McGrogan & Schneider, 2010) These discussions resulted in model modifications of changeover, handover, and fence check. The times and frequencies were compiled and used as model parameters, which validate the inputs to the model.

The overall feasibility of MAC for MQ-1B was also discussed with pilots, some of whom were proficient with the prototype MAC GCS. These discussions indicated that dynamic type operations with a single aircraft; such as, strike, emergency, and dynamic ISR were very difficult and consumed the entire attention of the pilot for the duration of the operation when performed with the current system and piloting paradigm. Periods of benign operation, such as transit and benign ISR, can include significant down time which could permit more than one aircraft to be controlled, especially if the sensor operator is given significant localized control as is the case in the prototype. It was acknowledged, however, that although a majority of the operational time is committed to benign operations, the dynamic nature of missions in an active area of operations results in the unpredicted and urgent occurrence of dynamic mission segments and the high workload associated with these dynamic events provides the opportunity for unpredictable, unsustainable increases in workload when flying MAC. Their input is consistent with the model output in Figure 16.

The nearly instantaneous spikes are communication events, with the longer periods of increased workload indicating a pilot task. This data will be discussed in depth in chapter four. In comparison to the dynamic ISR segment the benign ISR and transit segments appear uninteresting with long periods of no workload. This is consistent with the pilots' assessment of benign operations requiring little input and minimal communication. Dynamic ISR is substantially more difficult, not because the task of giving the aircraft commands is more complex, but the occurrence of new tasks and communication events increases very significantly with some communication events happening simultaneously, hence the higher spikes. At least qualitatively, the output of the model under no MAC is validated by pilots who are actively engaged in operations.

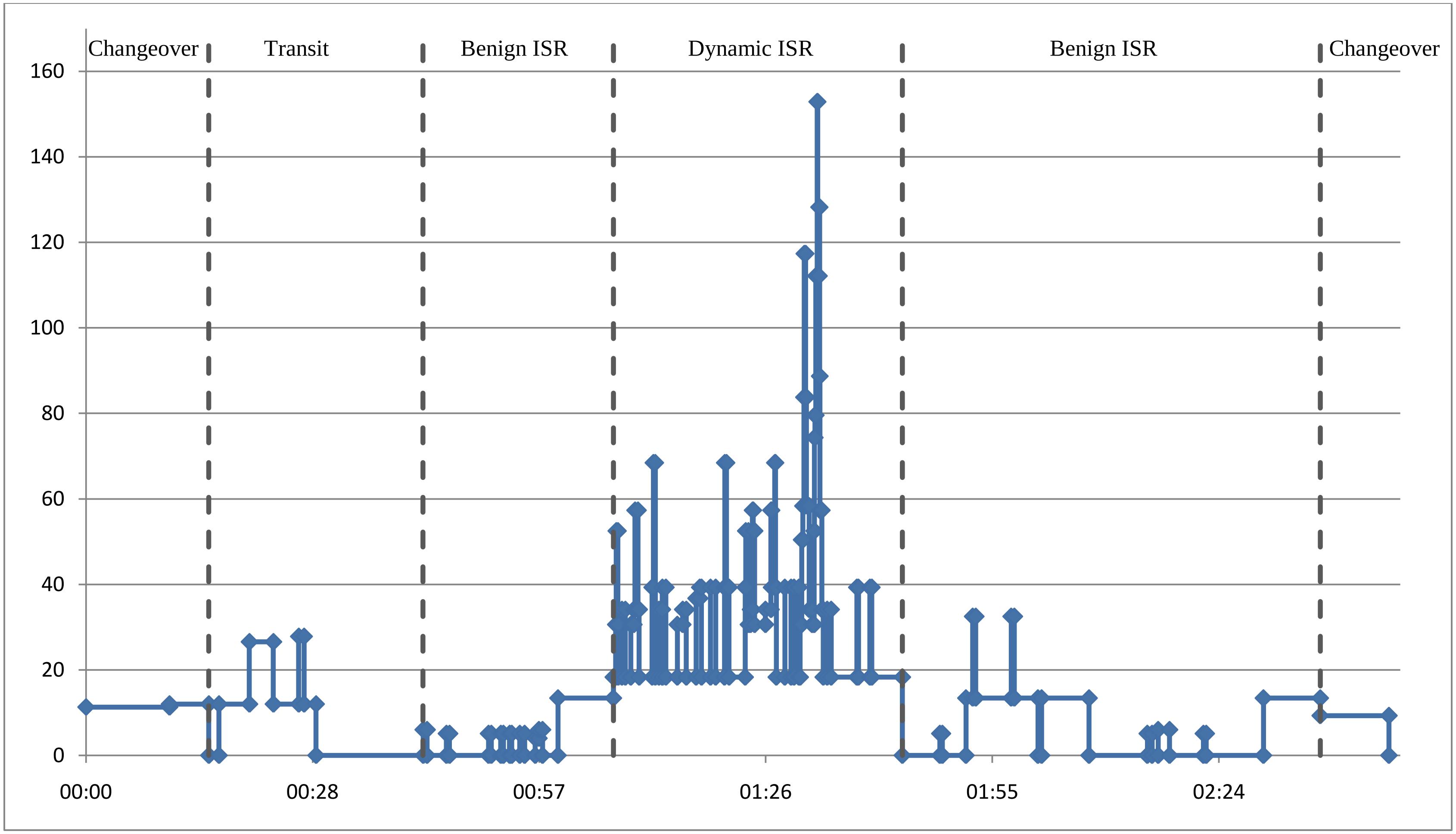


Figure 16. IMPRINT Model Single Aircraft Workload Trace

MQ-1B pilots considered dynamic ISR missions as workload intensive, requiring workload mitigation strategies. This led to the development of a relative workload limit for this analysis. A long model run was performed with a 12 hour dynamic mission to develop a robust Probability Density Function (pdf) of dynamic ISR found below in Figure 17.

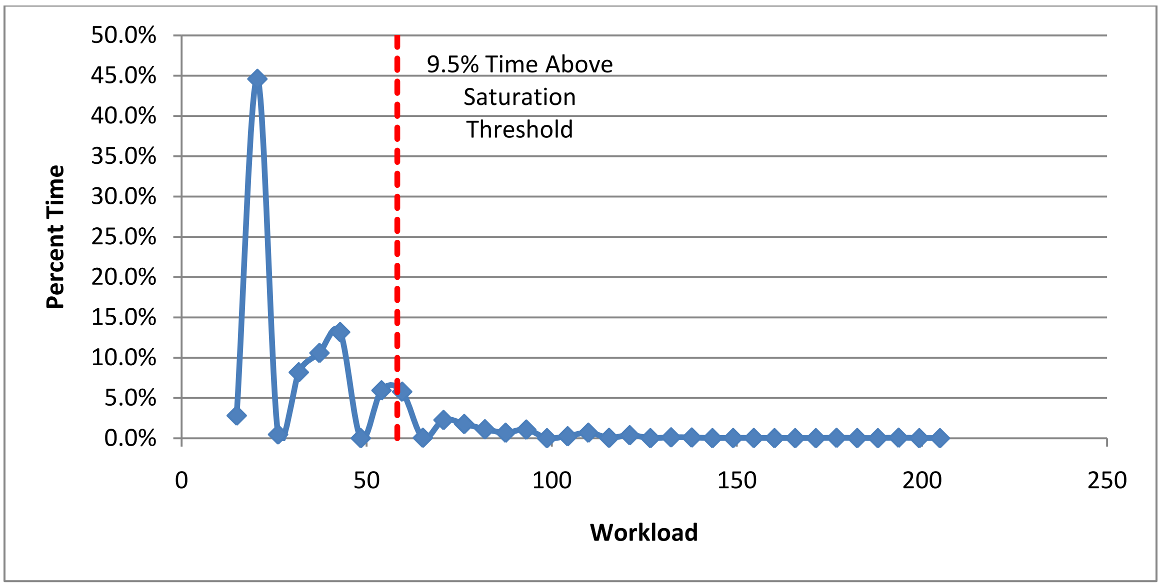


Figure 17. Probability Density Function of 12 Hr Dynamic Mission to Establish Saturation Threshold

The 90th percentile of workload was 58.3 which was approximated as 60 and is used throughout the analysis to assess the pilot's level of task saturation. The threshold is included with all data as a dotted red line. Events above 60 are considered to be near or above the saturation threshold where the system is imposing more work than the pilot can effectively perform. This finding is consistent with other IMPRINT models of workstation operations which set the saturation threshold at 60 (D. K. Mitchell, 2003). Workload above this saturation threshold level is predicted to require workload management strategies or else result in potential mission degradation. This topic is discussed further in Chapter 5. It should be reiterated that the model produces the workload imposed on the pilot by the system and assumes perfect mission effectiveness with no failures.

Model validity was established through informal methods which ensured the underling scope and assumptions were appropriate to the analysis. The standards regarding architecture, input, and output were codified and followed, assuring consistent model execution. SMEs validated the times, difficulties, and frequencies of model tasks, as well as face validity, model flow, and qualitative output. These efforts increased the realism of the model and lend credibility to its usefulness.

3.5 MAC Analysis Methodology

The analysis is divided into two phases to properly assess the critical factors affecting workload. Phase I covers every possible combination of mission scenarios; while Phase II represents shifts for a single pilot. Due to the limitations of IMPRINT Pro, each run was performed manually so the analysis was designed to minimize the number of runs while providing the data necessary to perform the desired analyses. To accomplish this goal, Phase I was designed to help limit Phase II to a shorter list of critical mission scenarios.

Phase I focused on the possible missions a pilot could be called on to fly, the mission-space. This consisted of all the combinations, including repetition, of mission conditions possible for MAC ratios 1 to 4. For example, under MAC ratio 2 a pilot may be in a condition in which one aircraft is in benign ISR and another in transit, or both in transit, or both in an emergency. Order is irrelevant since it does not matter to the pilot which specific aircraft is in each state, simply that the condition exists. Two restrictions were imposed on the combinations. First, strike would not be performed in MAC ratios 2 to 4. Second, no more than two dynamic type events, dynamic ISR or emergency, would occur simultaneously. The first restriction is from the operations concepts for MAC; the second restriction is based on models with three dynamic type events resulting in basal workload values more than twice that of any other condition and three times the assumed nominal human limit (e.g., the red-line value). Workload mitigation strategies are assumed to manage communication spikes and short term overload conditions, however longer overload conditions are assumed to be detrimental to mission effectiveness. These restrictions reduced the mission-space to 53 conditions which were investigated through a series of 10 runs. Each condition was two hours long, nominal pilot shift, to ensure that the stochastic workload behavior was fully described. Appendix A contains the Phase I run matrix.

Phase II replicated a series of shift scenarios to study areas in which workload represented realistic values for a single pilot's shift. These runs were between 2.5 and 4 hours long, consistent with normal shift lengths. Sixteen runs were performed to examine the baseline conditions for all ratios of MAC and a mission profile with a single dynamic task to assess the feasibility of each MAC ratio and the implications of an unexpected dynamic event. Ten additional runs were performed to explore the more borderline conditions of MAC. Appendix C contains the Phase II run matrix for these ten runs. The runs are useful to analyze the data for workload drivers.

IV. Analysis and Results

4.1 MAC Analysis – Phase I

The run matrix for Phase I of the analysis is in Appendix A. This run matrix has 10 runs with 75 different combinations of MAC ratio and mission profile. This represents every possible combination of mission modules available ignoring order and situations with more than two dynamic events. These runs are not operationally representative, because they are up to ten hours long and have an unrealistic number and sequence of mission phases. These combinations were designed to explore the interactions and conflicts of these different mission phases in order to more effectively identify and characterize the critical factors in operationally realistic runs in Phase II. The first run, depicted in Figure 18, has all of the mission phases under no MAC and lasted 9 hours and 20 minutes. This run is the baseline for comparing the remaining runs with varying ratios of MAC.

The sharp spikes in workload throughout the graph indicate communication events that are generated at different rates based on the mission module the aircraft is in. For instance, while Aircraft 1 is in transit there are only occasional communications spikes, but while Aircraft 1 is in dynamic ISR the communication spikes are so frequent that they blend together and overlap, producing higher spikes. The communication spikes are taller for dynamic ISR than transit because there is more conflict generated between the communication events and another ongoing task, not because the communication event is any more complicated. This suggests that communication spikes may be a critical factor for MAC. When Aircraft 1 is in transit or benign ISR, there is very little workload generated and there are even stretches when there is no workload. It is reasonable to assess that a pilot could control multiple aircraft in these mission phases without much difficulty.

The dynamic ISR and emergency phases are more complex than benign ISR and transit. The dynamic ISR phase has numerous communications spikes well above the saturation threshold. Even with a single aircraft, the model indicates that this mission phase drives workload up to critical levels and necessitates workload management strategies with some work offloading. This level of workload was corroborated by the MQ-1B pilots who stated that they are task saturated during dynamic ISR events and have to offload some of their communication tasks to the SO or MC(McGrogan & Schneider, 2010). The emergency phase is not as workload intensive as dynamic ISR, but the pilot is constantly engaged with resolving the aircraft emergency. Multitasking may be possible from a workload perspective, but pilots may not be able to switch their attention away from this aircraft long enough to address any tasks associated with another aircraft. This model is not sophisticated enough to model this situation so emergency will only be addressed from a workload perspective in this analysis.

The strike phase is included here to present a complete baseline of all MQ-1B mission phases. The preliminary operations concepts for MAC implicitly state that there will be no MAC for strike missions. Even though the workload for a strike mission is not as intensive as a dynamic ISR event, the potential for blue force fratricide and collateral civilian casualties with a live weapon release make any level of multitasking an unacceptable risk.

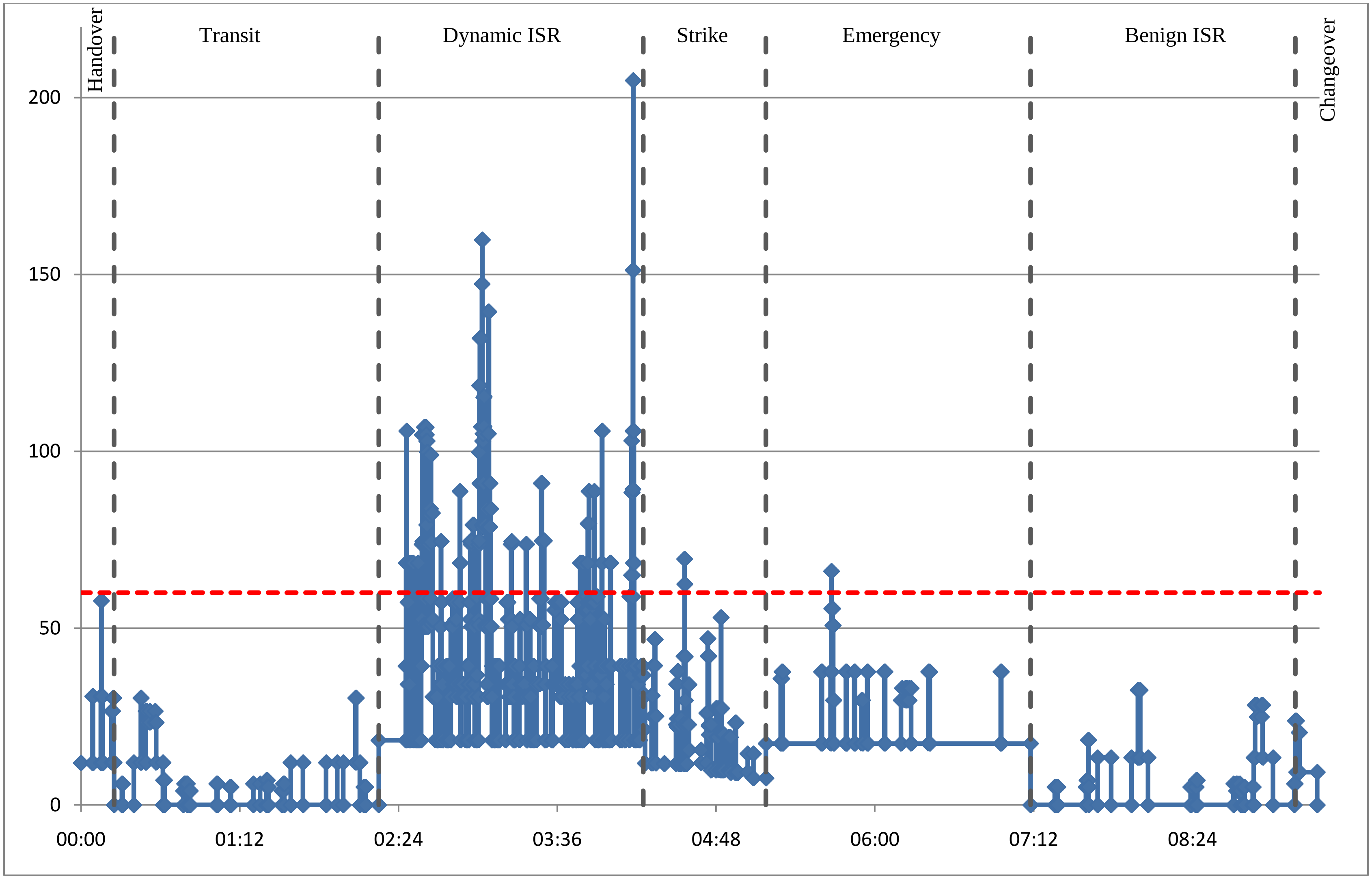


Figure 18. IMPRINT Model Workload Trace of Phase I Run 1 MAC Ratio 1 with Saturation Threshold

The workload graphs for run 2 with MAC ratio 2 and run 8 with MAC ratio 4 are represented in Figures 19 and 20 respectively. The rest of the workload graphs are in Appendix B. These two workload graphs provide insight into some of the interactions and implications of MAC during different mission scenarios and provide the basis for the Phase II setup. MAC ratio 3 is not presented at this point, because it does not provide any unique insights for this discussion and will be addressed in detail in Phase II.

Figure 19 represents one of the Phase I runs performed at MAC ratio 2. The workload level is low until the second marker where both aircraft enter the transit mission sequence. The workload is similar in the last sequence where both aircraft are in benign ISR. Both of these sections represent ideal circumstances with the lowest ratio of MAC possible however there are some workload spikes above the. These spikes above the saturation threshold are infrequent and most of the workload is well below the saturation threshold suggesting that this workload is manageable with some task sequencing and communications offloading, when necessary.

At the third marker the second aircraft experiences an emergency while the first aircraft remains in transit. The mean workload immediately increases and more of the workload approaches the saturation threshold. This mission scenario may be manageable as long as the aircraft in transit does not require immediate attention for anything critical.

At the next marker one of the aircraft is performing benign ISR and the other aircraft is performing dynamic ISR. The workload is frequently above the saturation threshold with a high sustained workload between the spikes. Communication is a driving factor in the workload spikes, but the primary tasks are providing the conflict which amplifies the workload values in the presence of communication. Simultaneously piloting one aircraft in dynamic ISR and a second aircraft in dynamic ISR would be a difficult with the possibility of mission degradation.

After the next marker, one aircraft enters dynamic ISR while the other has an emergency. This is a potential scenario that may arise with the use of MAC. Even the lowest points on the workload graph are above the saturation threshold with spikes over four times the saturation threshold value. Pilots in this situation would be unable to effectively split their attention between two aircraft in a dynamic situation and would have to choose between mission failure and the potential for aircraft loss.

At the first marker in Figure 20 three of the aircraft are in benign ISR and one of them is in dynamic ISR. With a single dynamic situation using a MAC ratio of 4, the workload immediately becomes completely unmanageable. Most of the workload is above the saturation threshold with spikes up to five times the saturation threshold value. Simple workload management strategies cannot reduce this level of workload to a manageable level. With the increased number of aircraft there is an increased chance of mission degradation. As the MAC ratio increases the pilot has less attention to split between the aircraft that are in a benign mission sequence.

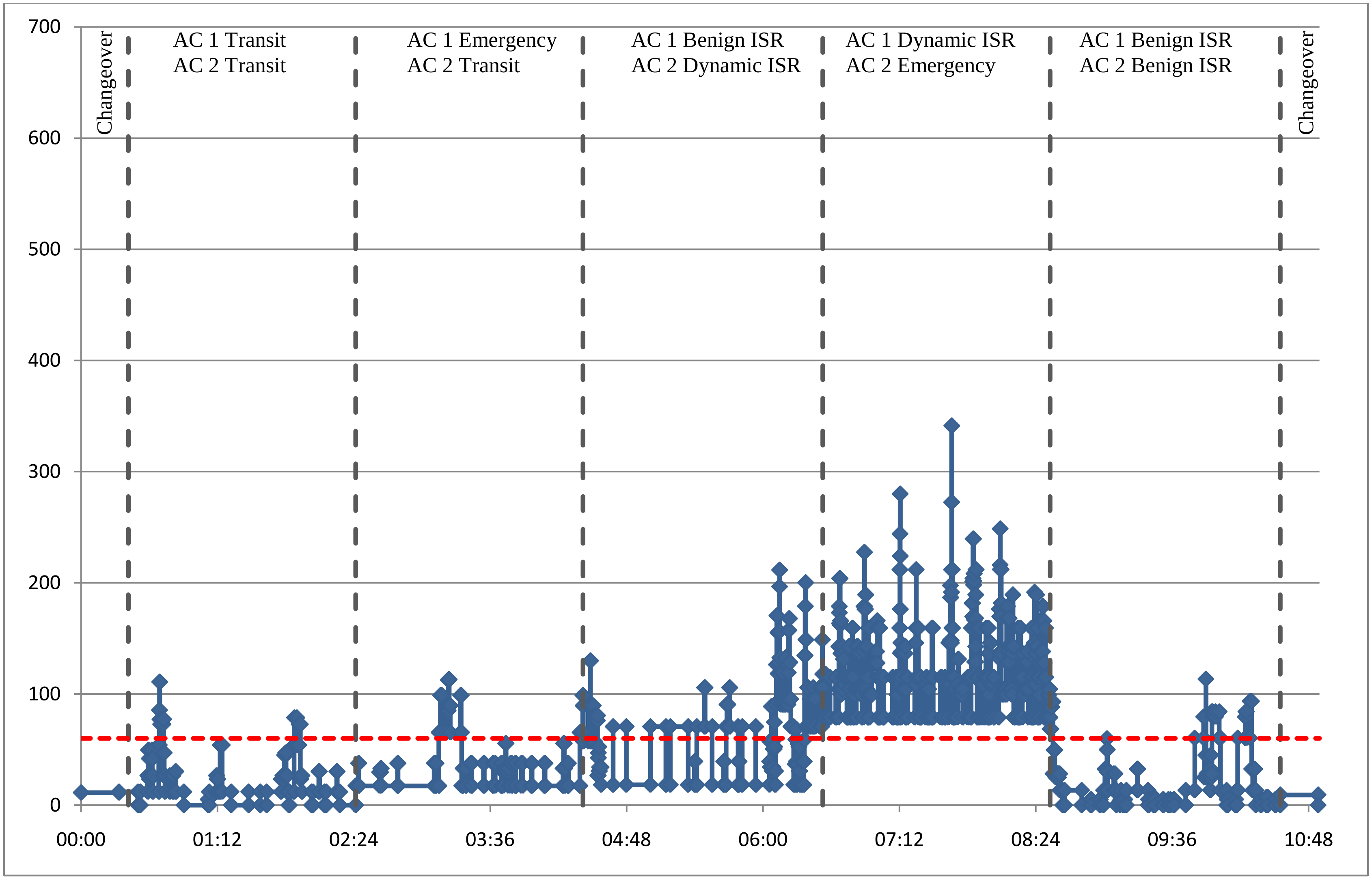


Figure 19. IMPRINT Model Workload Trace of Phase I Run 2 MAC Ratio 2

The next two mission segments have two dynamic situations. The first segment has a dynamic ISR and an emergency and the second segment has two dynamic ISRs. A situation with more than one dynamic drives the workload so high above the saturation threshold that some spikes extend to values ten times the saturation threshold value. Under these conditions, it is likely that the pilot will have to decide which of the aircraft in a dynamic mode to focus on with a complete exclusion to the other aircraft in a dynamic mode. These results indicate that it is simply not possible to manage more than one aircraft in a dynamic mode, because dynamic tasks cannot be delayed until the pilot has the ability to address them. Even a few minutes of this situation would be unacceptable.

The next marker has one aircraft in an emergency and the rest in a benign mission segment. The workload in this segment is mostly above the saturation threshold with numerous communication spikes well above the saturation threshold. Even this situation would not be manageable for more than a few minutes. Tasks from the aircraft in benign mission segments would have to be delayed while the pilot focused on the aircraft in the emergency situation. With more aircraft there is a larger chance that one of the aircraft in a benign mission mode will have a mission critical task arise during this time.

The last mission segment indicates the ideal situation for MAC with all of the aircraft in a benign mission mode. Even in this ideal situation the workload spikes above the saturation threshold repeatedly. There is no longer any time when the system is not imposing some level of workload on the pilot. This mission segment appears to be possible, but it would increase the constant level of workload experienced by the pilot and may cause pilot burn out in the long term.

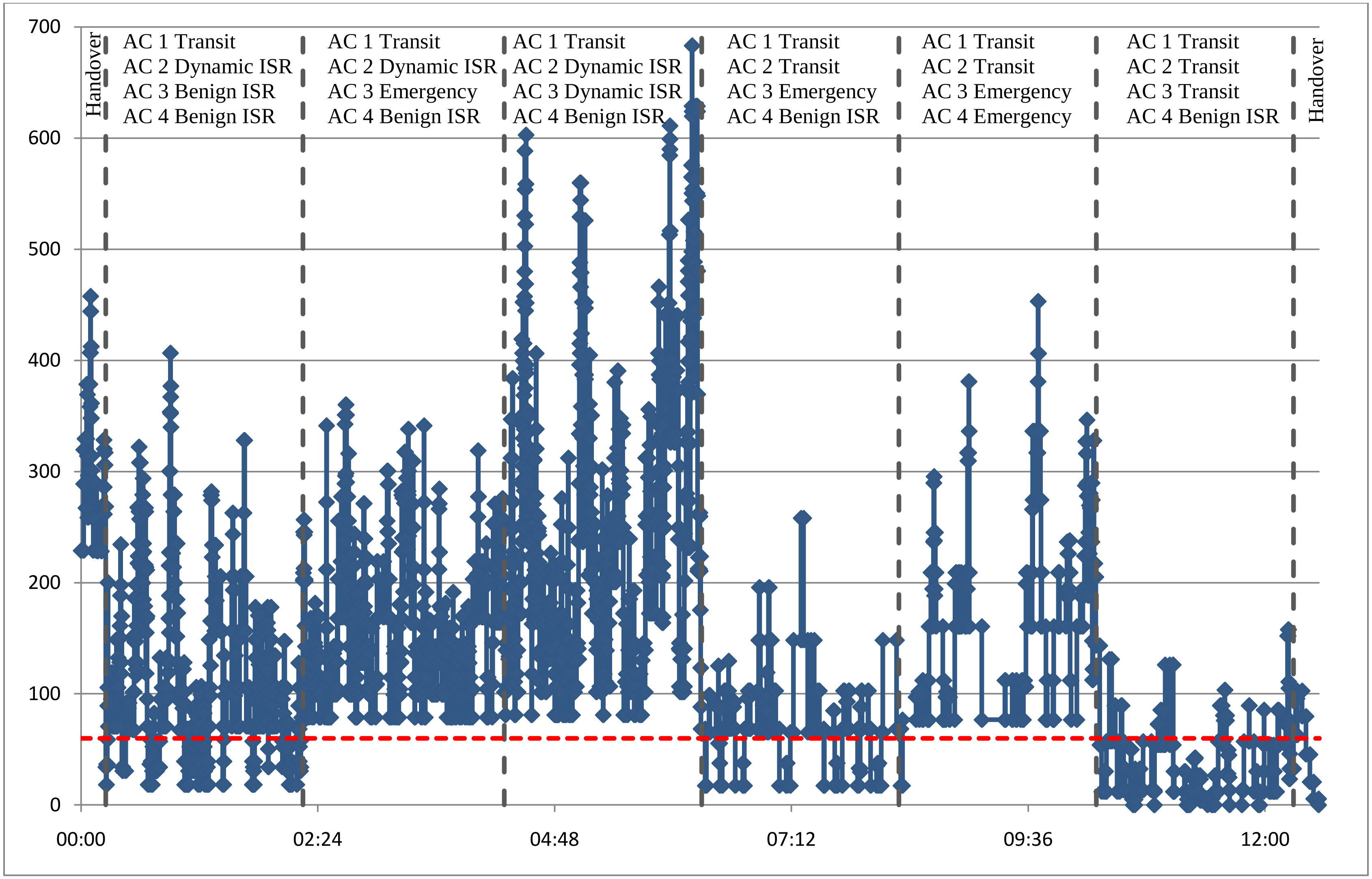


Figure 20. IMPRINT Model Workload Trace of Phase I Run 8 MAC Ratio 4

This analysis provides the data necessary to see the relevant mission combinations needed to run Phase II while avoiding impossible and redundant scenarios. This data will not be analyzed any further due to the artificial nature of ten hour pilot shifts with unlikely mission sequences. Phase I provided a complete overview of all the possible combinations of mission scenarios to allow Phase II to be more focused on the combinations that will provide the most useful information to this analysis.

4.2 MAC Analysis – Phase II

First a purely benign mission will be compared directly to a benign mission with a single dynamic event to investigate the impact of an unanticipated dynamic event during a normal mission sequence at every MAC ratio. Only a single dynamic event at a time is modeled in Phase II of this analysis, because Phase I clearly indicated that more than one dynamic event imposes an unrealistic level of workload on the pilot at all ratios of MAC. These mission sequences represent operationally realistic mission profiles for a single pilot doing one shift in the GCS. The data from Phase I indicates that the transit and benign ISR mission modes generate similar workload traces with benign ISR producing slightly more workload. Likewise the dynamic ISR and emergency mission modes also produce similar workload traces with dynamic ISR producing a higher workload value. Only benign and dynamic ISR mission modes are used for the initial comparison, because they generate the most workload. The rest of the workload graphs from Phase II are in Appendix D.

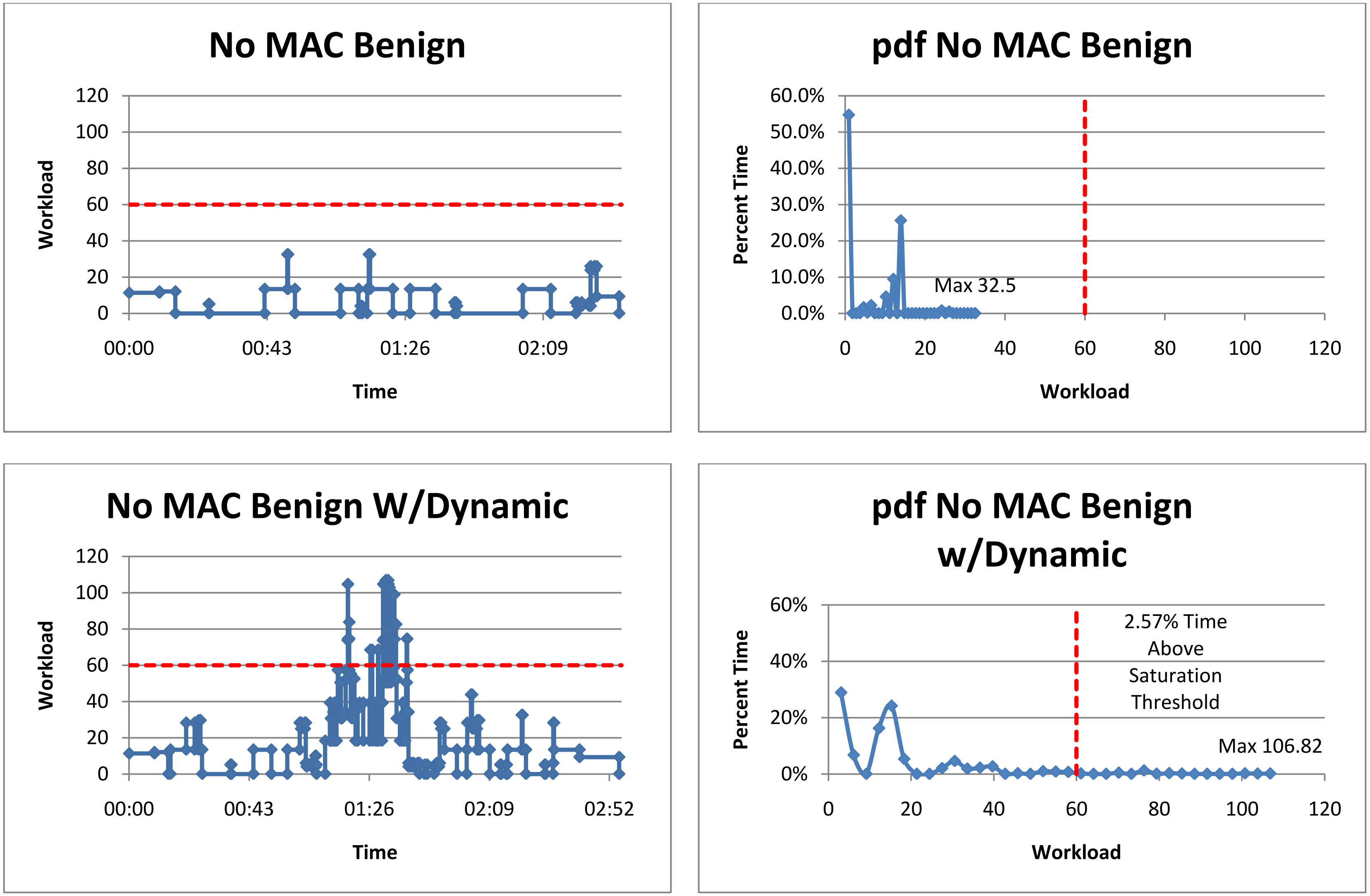


Figure 21. IMPRINT Model Workload Trace and Workload pdf of No MAC Data Comparison

4.2.1 MAC Model Results for No MAC

4.2.1.1 No MAC Workload Comparison

The comparison of no MAC data, given in the quad-graph in Figure 21, provides the baseline for the subsequent comparisons in this section. The top of the quad-graph depicts the mission with only benign ISR. As seen previously, a mission with a single aircraft performing only benign ISR can be uneventful. The mission starts and ends with a changeover event and has numerous lulls in workload in between. The pdf illustrates that the most common level of workload imposed by the system is zero. This means that the pilot would spend more time monitoring the system rather than actively interacting with the system. This is consistent with MQ-1B pilot discussions. Even with communication events occurring at the same time as other tasks, the workload is never higher than 33, which is barely half of the saturation threshold value of 60.

The mission sequence found in the bottom of the quad-graph in Figure 21 is a typical pilot's shift with an unplanned dynamic ISR event occurring in the middle of a benign ISR. This mission starts and ends with a changeover and immediately goes into a benign ISR mission. The dynamic ISR event occurs in the middle of the pilot's shift and lasts approximately 30 minutes. A dynamic ISR event may last much longer than this, but the length was chosen to provide an illustration of the effects of a short dynamic situation during a pilot's shift with a longer event having a proportionally larger impact.

The portion of the mission where the aircraft enters the dynamic ISR mission mode can be clearly seen on the workload trace. The first part of the workload trace is well below the saturation threshold and appears very similar to the workload trace of the all benign mission in the top of the quad-graph. The workload level rises dramatically and the more frequent communication events spike the workload to values above the saturation threshold. Even though some of the spikes exceed 100 on the workload graph, it is important to note that the workload is above the saturation threshold value for only 2.57% of the total shift. This represents a difficult, but manageable level of workload based on discussions with the MQ-1 pilots. The spikes above the saturation threshold will require some workload mitigation strategies to ensure that there is no mission degradation. The pdf in Figure 21 for the benign ISR with the dynamic event proves that the majority of the workload is well below the saturation threshold.

4.2.1.2 No MAC Workload Drivers

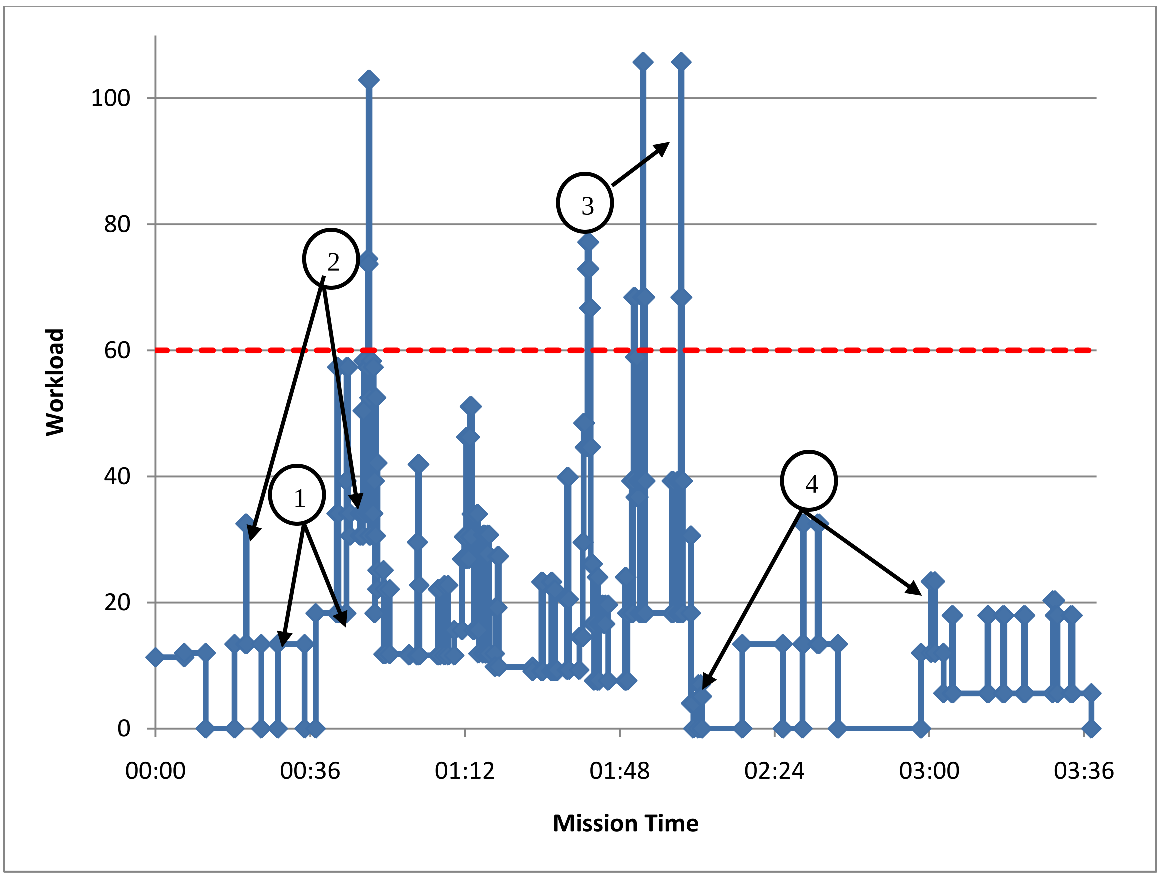


Figure 22. IMPRINT Model Workload Trace of Complex No MAC Mission (Run 3)

Figure 22 is the workload trace of a mission which escalates from benign ISR into a strike mission before the aircraft is returned to base. As previously discussed, benign and dynamic ISR are substantively different in terms of pilot workload. Marker 1 in the figure points out these two conditions, benign on the left and dynamic on the right. The task difficulty of these differs due to an interface shift and a visual resource shift. Benign ISR is performed using waypoints manipulated by a trackball and keyboard in much the same way a figure is manipulated in a document. Dynamic ISR uses the traditional flight controls, throttle and flight stick, because it requires more precise and rapid adjustment. Due to remote operation, there is delay of a couple seconds between when the pilot issues commands and observes the aircraft reacting. In benign ISR this is inconsequential, however, during dynamic ISR the pilot exerts direct control over the aircraft and this delay increases the difficulty. Benign ISR is most frequently performed on stationary targets, or in an area of interest observing specific targets. Whereas, dynamic ISR requires a higher situational awareness to anticipate target movements and maintain orientation. The effect of the interface and visual shift is that reorientation of the aircraft in benign ISR has a workload of 13.4, and in dynamic ISR is 18.3. This difference gains significance when communication is overlaid. Marker 2 indicates two nearly identical tasks, the left is benign ISR with a chat communication the pilot must read, and the right is a dynamic with the same chat. The conflict on the visual channel drives the workload to 32.5 and 39.3 respectively, nonlinearly increasing the workload due to conflict.

In higher task situations workload induced by intra-channel and cross-channel conflict dominates rapidly. Marker 3 is a case where the pilot is performing a dynamic

ISR and has three chat messages come in simultaneously. This case is realistic when considering that pilots have at a half dozen chat windows open throughout the mission. The task demand for this case is 33.6, with a conflict of more than double: 72.2. The pilot is simultaneously trying to assimilate a large quantity of visual information which results in a workload of 105.8, two thirds of which is driven by the visual intra-channel conflict. This type of conflict is exacerbated through the addition of more aircraft which will be investigated in following sections.

Marker 4 designates an example of cross-channel conflict. The left arrow is a grouping of chat and verbal communication that occur during no other tasks and are very low workload, less than 10. The right arrow is a verbal communication which occurs during a transit course update. In this case, a workload event of 4 (verbal communication) nearly doubles the workload from 12 (no comm.) to 23.3 (with comm.), with 7.3 of that as cross-channel conflict.

It should be noted that only one percent of this mission was over the saturation threshold which places it squarely within the realm of the practical. The observations regarding conflict dominance in a conventional control condition indicate that interface adjustments would reduce workload from intra-channel conflict. Automation and other interface adjustment could lower task demand which fundamentally drives workload.

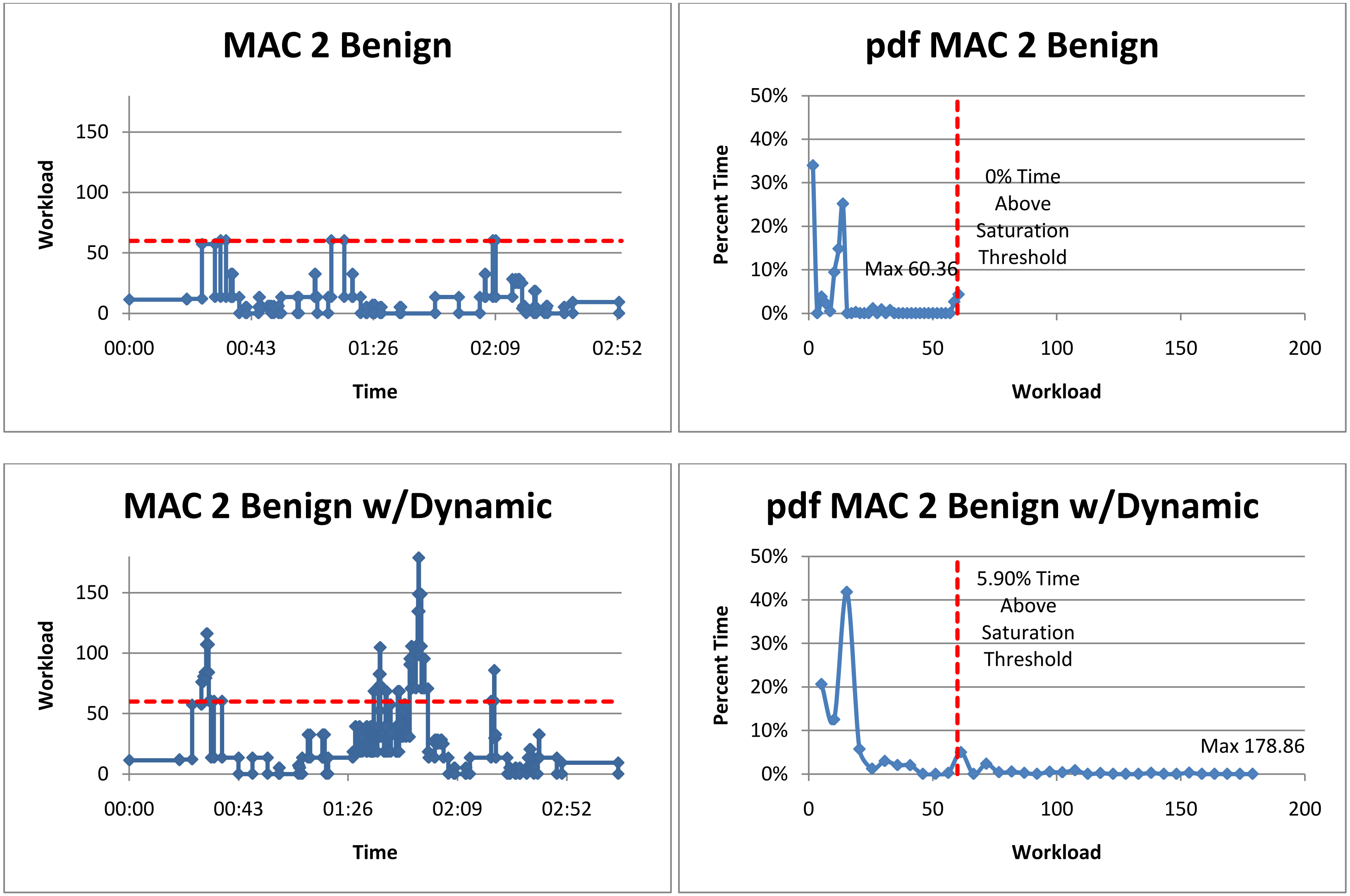


Figure 23. IMPRINT Model Workload Trace and Workload pdf of MAC Ratio 2 Data Comparison

4.2.2 MAC Model Results for MAC Ratio 2

4.2.2.1 MAC Ratio 2 Workload Comparison

Figure 23 is the quad-graph for MAC ratio of two. The top left graph depicts the workload trace for two aircraft in benign ISR. The mission begins and ends with changeovers the same as with the no MAC mission; however there are now two changeover events in sequence to account for the additional aircraft and crew briefs required for the additional aircraft. The workload for two aircraft is now much busier than it was with a single aircraft and there are now some communication spikes up to the saturation threshold. There are still periods of little or no workload. The pdf illustrates that no workload is imposed by the system during nearly 35% of the pilot's shift. Even with multiple spikes to the saturation threshold the overwhelming majority of the workload is at relatively low workload levels. This situation would be easily manageable by a pilot with little risk of and mission degradation.

The lower graphs on the quad-graph in Figure 23 represent two aircraft being flown in benign ISR with a single aircraft experiencing a dynamic event for approximately half an hour before returning to benign ISR. The benign portions of the graph have moderate workload with manageable spikes above the saturation threshold, but when one of the aircraft begins a dynamic ISR event the workload increases significantly. The pdf points out that the majority of the workload is still well below the saturation threshold, but now 5.9% of the workload is above the saturation threshold. This workload level appears to still be manageable, but it will require the pilot to employ workload mitigation strategies and is not a situation that should be maintained for long periods of time.

4.2.2.2 Mac Ratio 2 Workload Drivers

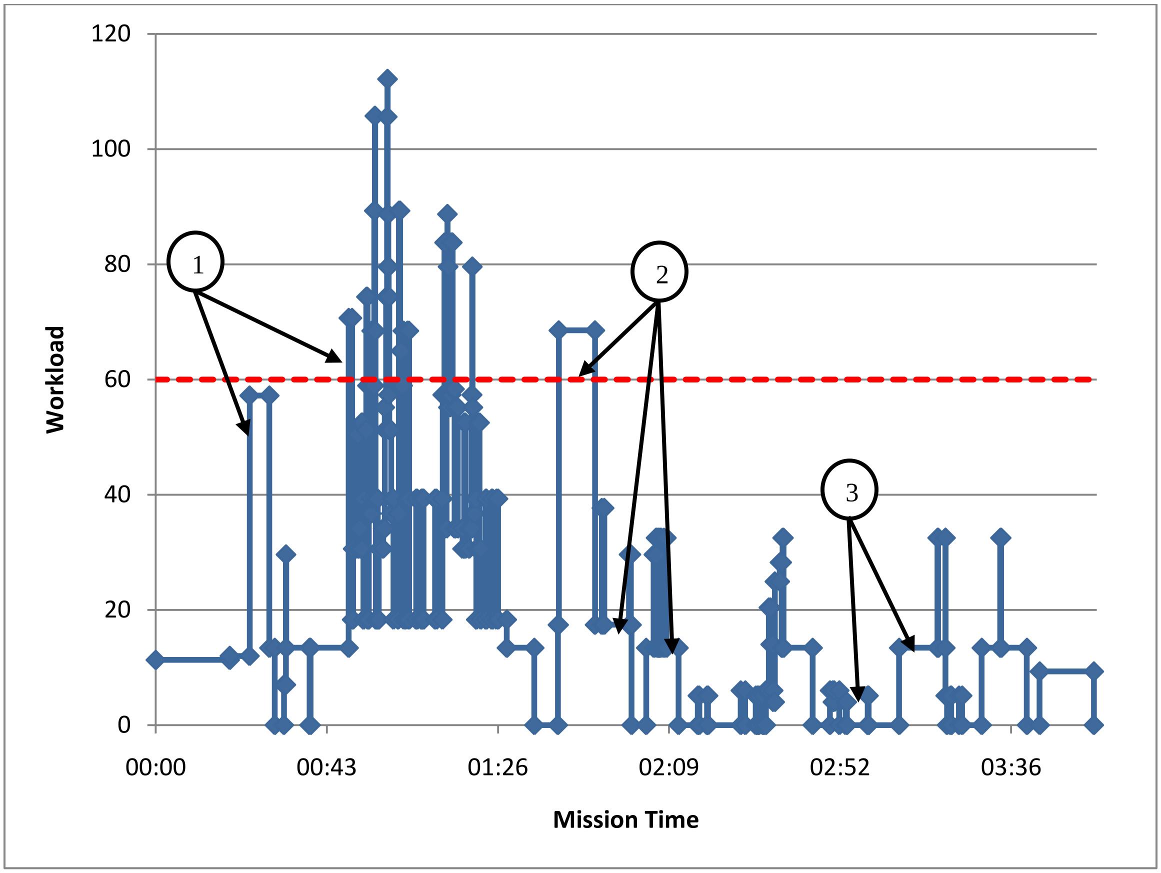


Figure 24. Model Workload Trace of Complex MAC Ratio 2 Mission (Run 5)

A primary concern when increasing the MAC ratio is task overlap. It is accepted that communication will be constantly overlapping primary piloting tasks, but when MAC is not used, piloting tasks do not overlap one another. Instances of prolonged overlap, those called out in Marker 1, are of greater concern than communication spikes. The first is an overlap between fence check and the benign ISR initialization task, the second is a short overlap between benign and dynamic ISR. In the first case, the system requires the pilot to review the system status of one aircraft while giving orbit commands to the second, looking at two screens at once while thinking about two different things. During the second case, one aircraft needs commands through the keyboard and trackball, while the other needs to be flown with throttle and stick. Primary task overlap like this generates intra-channel conflict along three channels, cognitive, visual, fine motor, and the increased task difficulty in those channels increases cross channel conflict as well. Performed individually these tasks have workload below 20, when they are conflicted the total workload jumps to 57 and 70 with conflict being 50% to 60% of the total. The result is during overlap the workload increase up to or over the saturation threshold. Marker 2 in Figure 24 is a third case of task overlap, in this instance between benign ISR and emergency. Even at MAC ratio 2 multi-task overlap is a clear critical factor in the implementation of MAC.

The pilot can employ workload mitigation strategies in these types of situations. For example, after the emergency is evaluated, the criticality may be low enough that the pilot can switch from the emergency to the relocation of the benign ISR aircraft and back before the emergency worsens. The pilot may be able to delay relocating the benign aircraft or authorize the sensor operator to relocate the aircraft. An important observation of MAC ratio 2 is that the workload remains significantly below the saturation threshold for much of the mission. Marker 3 draws attention to the low workload tasks which comprise 81% of the mission time. The pilot may be able to manage short duration task overlap by employing workload mitigation strategies. The additional aircraft only raised the time above saturation threshold to 5.5%, considering that the saturation threshold represent the 90th quartile from what is considered a difficult mission; this may present acceptable increased workload with corresponding acceptable risk of mission degradation.

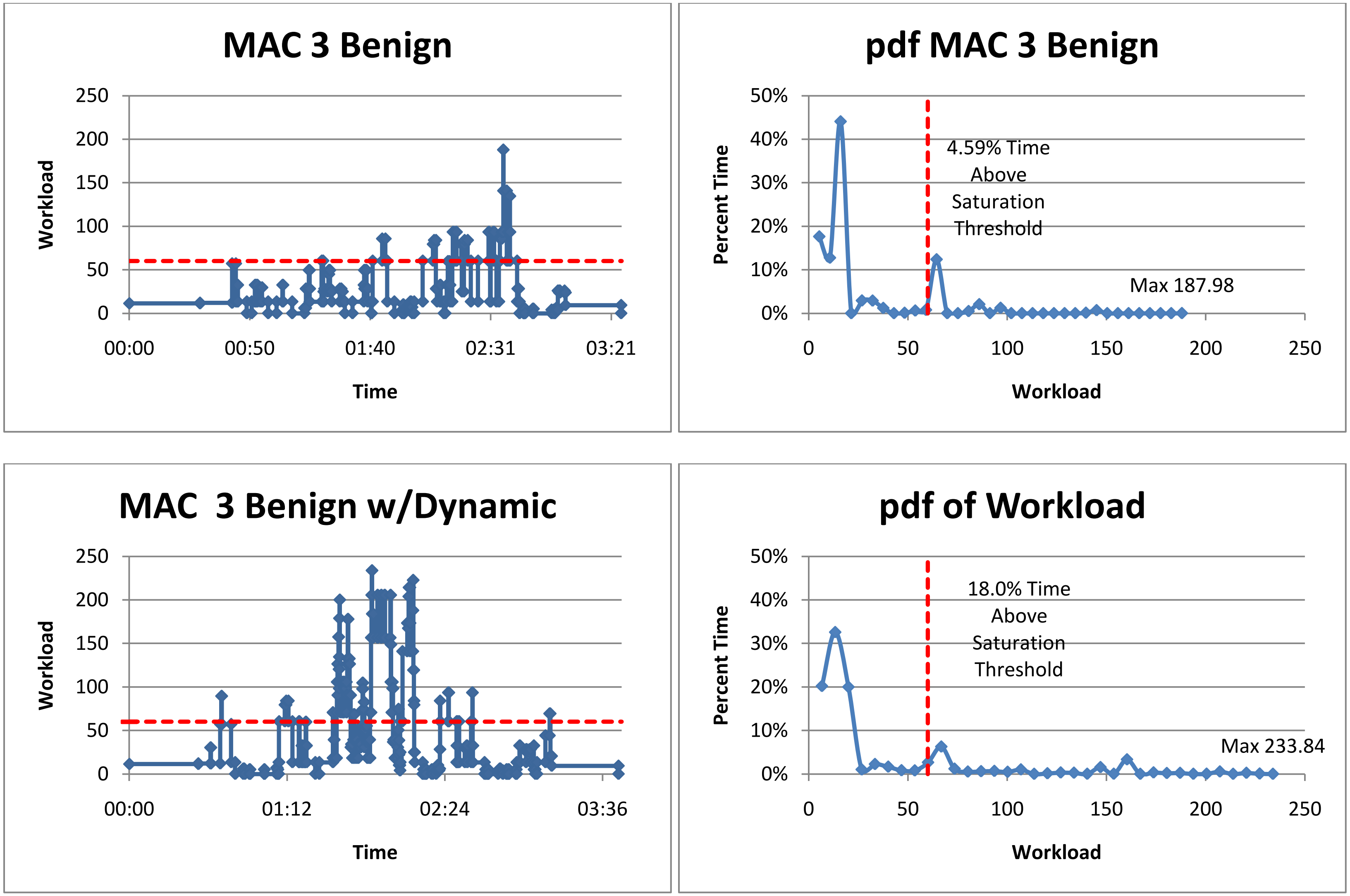


Figure 25. Model Workload Trace and Workload pdf of MAC Ratio 3 Data Comparison

4.2.3 Model Results for MAC Ratio 3

4.2.3.1 MAC Ratio 3 Workload Comparison

Figure 25 is the quad-graph for MAC ratio of three. The top graphs represent a mission controlling three aircraft, all in benign ISR. The workload trace demonstrates that under ideal circumstances this ratio of MAC is difficult. Without any dynamic events 4.59% of the workload is above the saturation threshold and the workload peaks at 188. Portions of the workload appear to be easily manageable, but the large spike on the right hand side of the workload trace confirms that even with infrequent, low difficulty tasks, the workload can become unmanageable at times. The difficulty of using workload management techniques to manage this spike would depend on the time critical nature of some of these tasks. If these tasks can be delayed without impacting any of the missions then this workload might be easily manageable. Theoretically, since all of these tasks are for benign ISR they are not as time critical as tasks for a dynamic event, but this is not something that can be easily quantified.

The bottom two graphs in Figure 25 depict MAC ratio of three with a single dynamic event. The workload levels during the dynamic event max out at 234. This is a workload level that may be unmanageable even with workload mitigation strategies. It is clear from the pdf that the majority of the workload is still manageable, but the tail on the pdf is getting longer and now 18% of the time the workload imposed by the system is over the saturation threshold. As the MAC ratio increases it becomes apparent that unplanned dynamic events have a major impact on the ability of the pilot to manage MAC. An unplanned dynamic event is unmanageable even for short periods of time.

These unplanned dynamic events are another critical factor in the implementation of MAC.

Figure 25 also demonstrates a new phenomenon with higher ratios of MAC. The mission begins and ends with sequential changeovers for each of the aircraft the same as previous missions. Changeovers are events that cannot be performed simultaneously because they involve giving or receiving a verbal briefing about the mission and status of each aircraft. With the increasing length of these sequential activities the amount of useful piloting time is reduced. Theses changeover tasks also impose a relatively low workload, which artificially lowers the pdf of the workload for the entire shift.

4.2.3.2 MAC Ratio 3 Workload Drivers

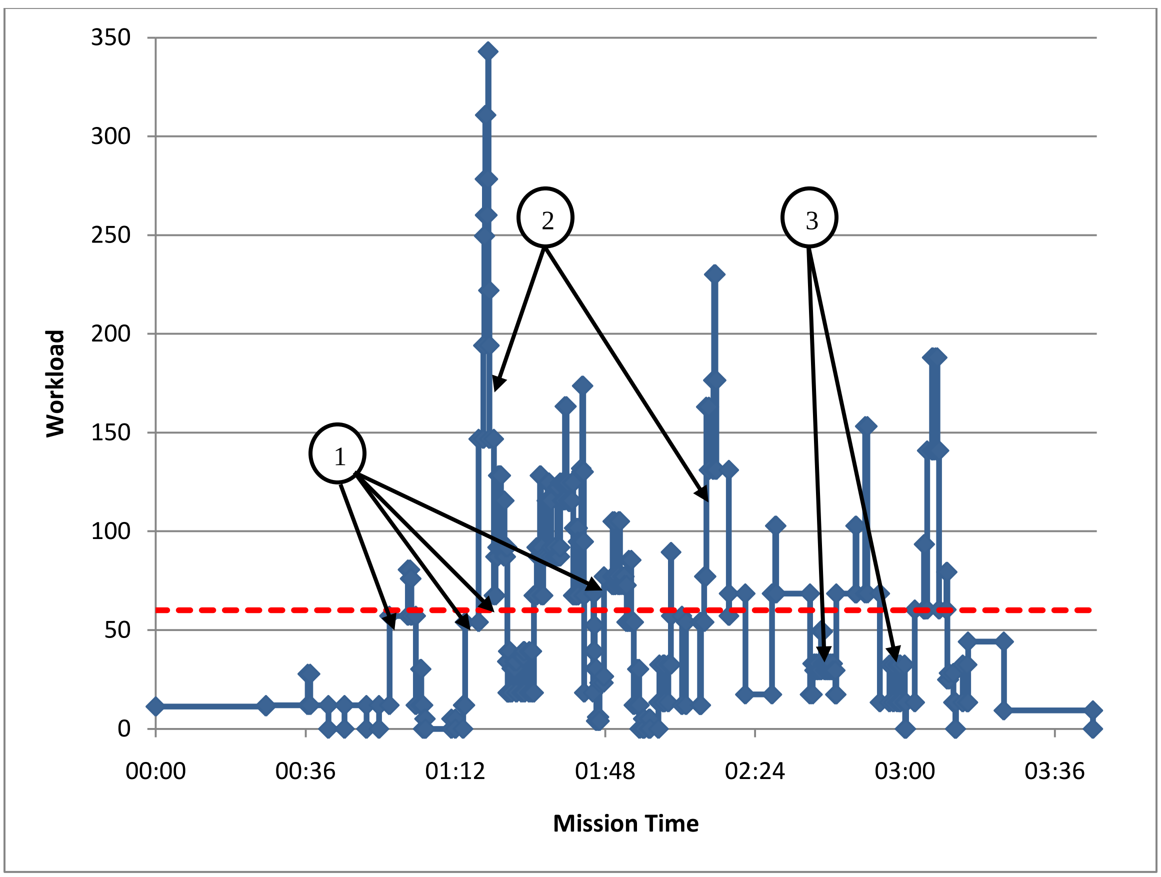


Figure 26. Model Workload Trace of Complex MAC Ratio 3 Mission (Run 6)

Increasing the number of aircraft provides three times more opportunities for primary task overlap which is a major driver in pilot overload. Marker 1 designates multiple instances of double primary task overlap approaching overload which include dynamic/benign and benign/benign operational tasks. Workload mitigation strategies can be employed, but where they were the exception in MAC ratio 2, they have now become the rule. If many of the piloting tasks are offloaded to the SO, as is done in the prototype MAC, this becomes manageable. However, this effectively places the pilot in a role of supervisory control over enlisted UAS "operators" who can perform a subset of piloting functions without the formal training of pilots.

Triple task overlap, Marker 2 in Figure 26, doubles the workload of dual task overlap. These are conditions which are impossible to perform as the workload jumps from 54 to 131, and triples the conflict from 30 to 93. This is well beyond all but the highest communication spikes (99^{th} percentile) of a conventional dynamic ISR mission, and it is necessary for several minutes to maintain perfect mission effectiveness. These instances are dangerous, albeit infrequent, events which have a high potential of mission degradation. The double and triple task overlaps are the driving force behind 28% of the mission time above the saturation threshold, an increase of 23% over MAC ratio 2, further reinforcing multi-task overlap as one of the critical factors in MAC.

The communication model provides an elegant demonstration of its operation in this run. Marker 3 points to two instances, the leftmost is a verbal conversation composed of voice calls and responses, similar to a phone or radio conversation, and neatly illustrates the recursive functionality of the communication model. The rightmost arrow designates a series of reading tasks in which the pilot “catches up” on the chat messages.

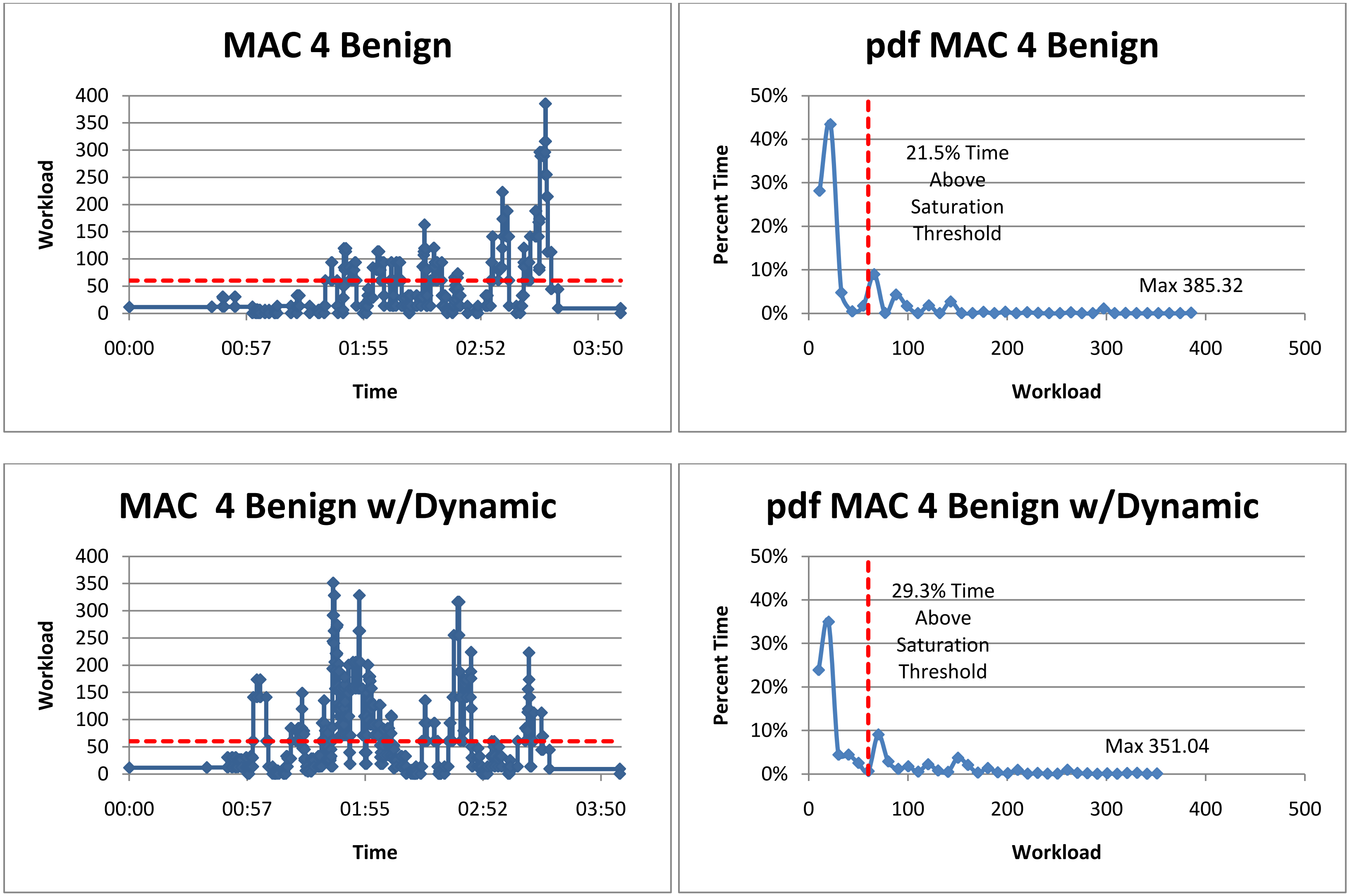


Figure 27. Model Workload Trace and Workload pdf of MAC Ratio 4 Data Comparison

4.2.4 Model Results for MAC Ratio 4

4.2.4.1 MAC Ratio 4 Workload Comparison

Figure 27 is the quad-graph for MAC ratio of four. The workload trace for four aircraft in benign ISR is higher than that of a single aircraft in dynamic ISR. A single aircraft in dynamic ISR was the baseline for a difficult but manageable mission with some workload mitigation strategies necessary. Under ideal circumstances with all aircraft in benign ISR, the workload for MAC ratio of four exceeds the baseline for a difficult mission. The workload spikes to 385 and is above the saturation threshold 21.5% of the time. Even with robust workload mitigation strategies this is a very difficult mission for the pilot and has a high chance of mission degradation. Without a single dynamic event, this mission pushes the limits of a realistic level of workload.

The bottom of Figure 27, which depicts MAC ratio of four with a single dynamic event, reinforces the observations about the difficulty of MAC ratio 4. Even the portions that do not involve any dynamic tasks spike well above the saturation threshold. The small portion of the workload trace that does have a dynamic event becomes completely unmanageable. The workload is consistently above the saturation threshold with only brief dips below the saturation threshold. The workload peaks at 351 and is now above the saturation threshold 29.3% of the time. During the time when one of the aircraft is in dynamic ISR, the workload would be unmanageable.

The peak for this workload trace is less than the peak for the workload trace with four aircraft in benign ISR due to the stochastic nature of the model. The benign ISR mission modes generate tasks intermittently, but if multiple aircraft happen to generate tasks at approximately the same time, workload spikes briefly due to the conflict between the tasks which occurs during the benign ISR mission mode in this figure.

4.2.4.2 MAC 4 Workload Drivers

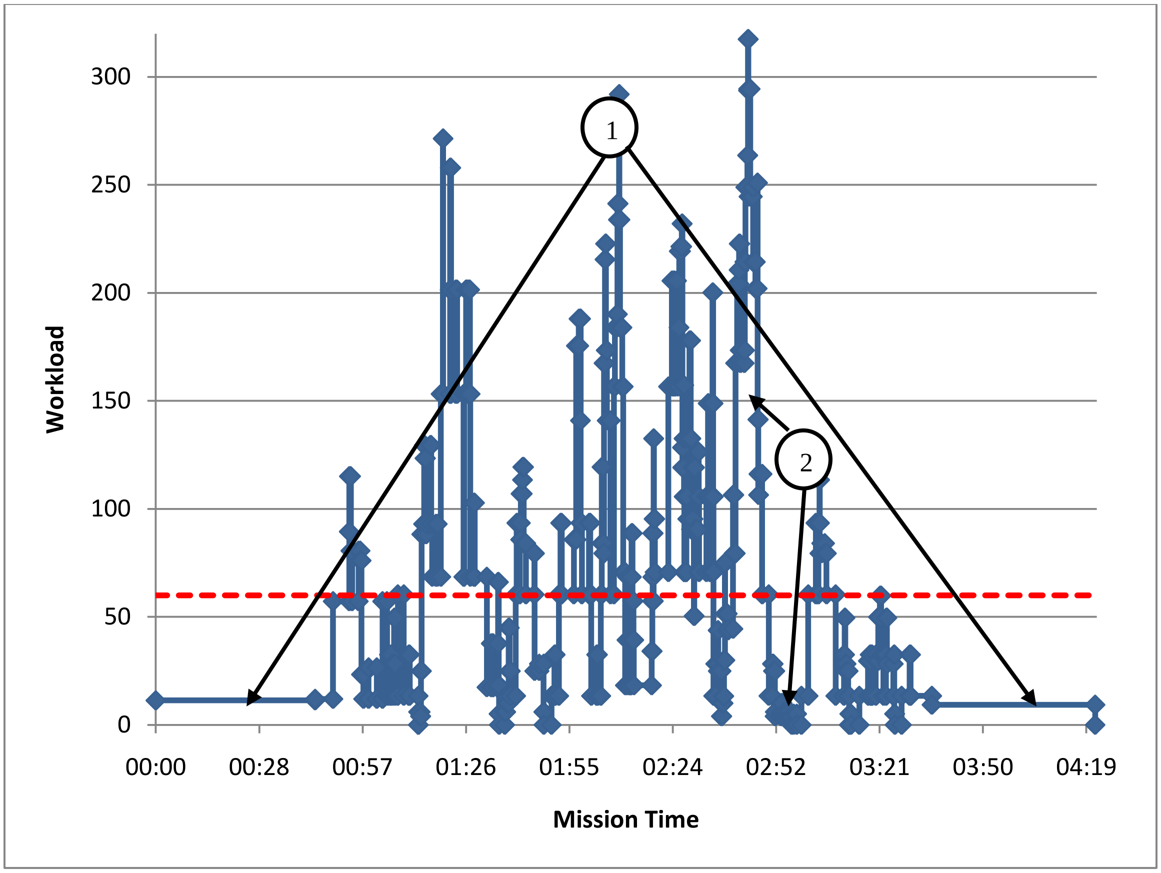


Figure 28. Model Workload Trace of Complex MAC Ratio 4 Mission (Run 10)

A significant observation of MAC ratio 4 is the extensive changeover time involved. Operationally, changeovers would be performed serially and so they are modeled as such. A side effect is an increased GCS time with a diminishing mission time. Marker 1 designates these changeovers which are 40% of the total mission time at MAC ratio 4 vs. 20% at no MAC. A result is the pilots perform fewer mission related tasks during a shift, and then transfer the situational awareness (SA) to another pilot to enable them to perform mission related tasks. At higher MAC ratios changeover constriction stands out as a critical factor of MAC.

These extended periods of low workload also reduce the workload pdf and artificially skew the probability density function for the mission to lower workload levels. For the purpose of comparison, the changeovers were removed from the data set which resulted in a pdf of the mission where 36% of the mission time was spent above the saturation threshold. To put that in context, 30% of the mission time was the pilot performing a single task (workload less than 20). In Figure 28 there are nearly as many spikes above saturation threshold as dips below. Marker 2 indicates one such case where there is a ten minute spike of workload over the saturation threshold. This workload spike is caused by double and triple benign task overlap. The subsequent workload valley is from a communication exchange that lasts for 6 minutes. The increased frequency of double and triple task overlap, and the associated conflict workload, drives MAC ratio 4 missions beyond the workload limit of pilots.

4.3 Summary of MAC Analysis and Results

If workload over the saturation threshold corresponds to points in the mission where the pilot's effectiveness could be degraded then the data indicates a rapidly increasing loss in pilot effectiveness as the MAC ratio increases. The ability of a pilot to manage multiple aircraft is based on the assumption that the large amounts of untasked time in a typical pilot's shift can be better utilized. This may not be a valid assumption. While excessive untasked time creates boredom and reduces situational awareness, a pilot needs some time in between tasks to monitor the system and plan for future actions.

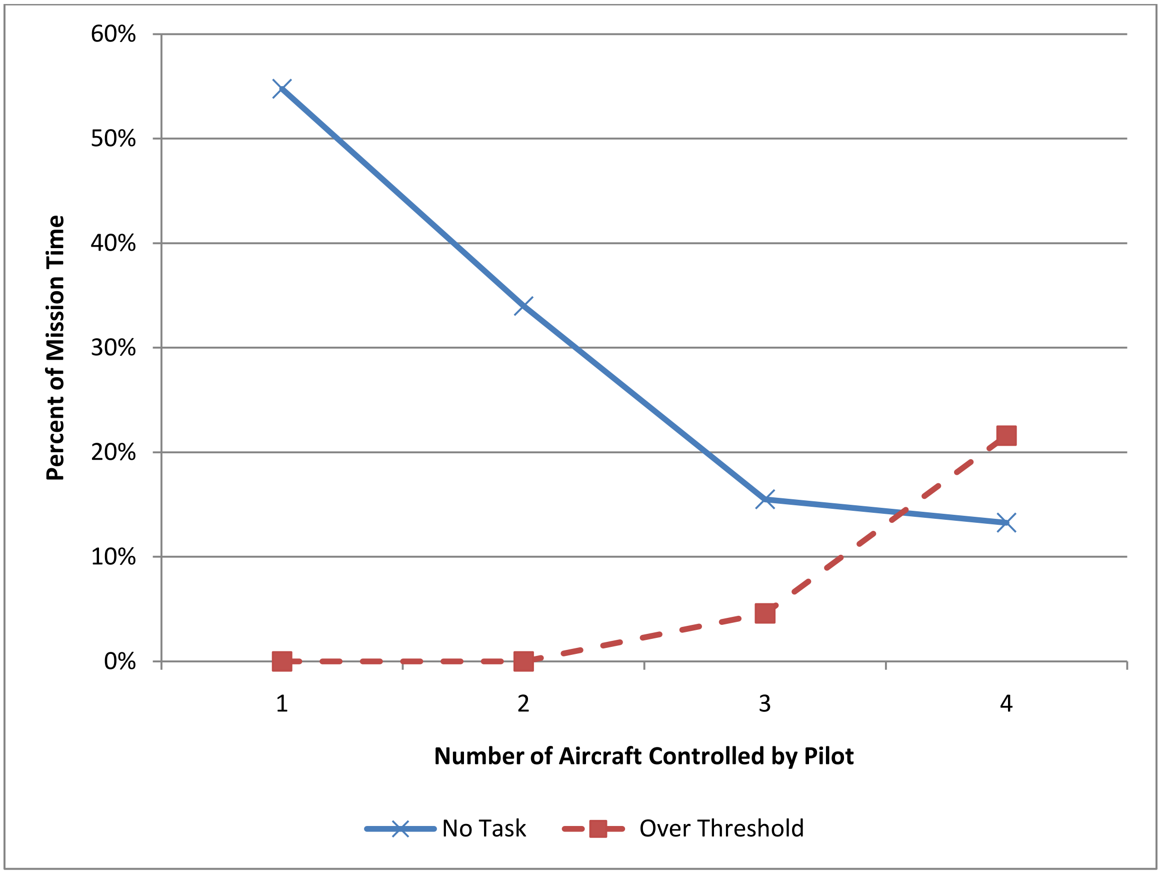


Figure 29. Benign Untasked Time vs. Time Over Saturation Threshold of Workload Data from Model Output for Various MAC Ratios

Figure 29 illustrates the growth of untasked time vs. time over saturation threshold by MAC ratio for aircraft in benign ISR. While there is some potential for the degradation of pilot effectiveness using MAC ratio of 3, there clearly is a significant increase in the potential for the degradation of pilot effectiveness using MAC ratio of 4 even under ideal circumstances. There are large amounts of untasked time at low MAC ratios, when there are no dynamic events. There is a consistent drop in untasked time from no MAC to MAC ratio 2 and MAC ratio 2 to MAC ratio 3. This suggests that these increases in the ratio of MAC make effective use of the untasked time of the pilot. However when the MAC ratio increases from 3 to 4 there is a much smaller drop in pilot untasked time with a corresponding jump in time above saturation threshold. Going from MAC ratio 3 to MAC ratio 4 is more likely to cause task conflict rather than effective use of the pilot's untasked time. This occurs because MAC is not able to make the most effective use of this ideal time since there is no inherent sequencing of tasks.

The probability of whether a task occurs during the untasked time or overlaps with another task is a function of the amount of untasked time and the number of tasks performed. There will be limitations on how much a pilot is able to effectively sequence multiple tasks that occur simultaneously since many of these tasks have some degree of time criticality. Delaying benign tasks can cause mission degradation in the form of the aircraft arriving to the mission area late, or potential essential elements of information being missed because the aircraft was not in the proper position.

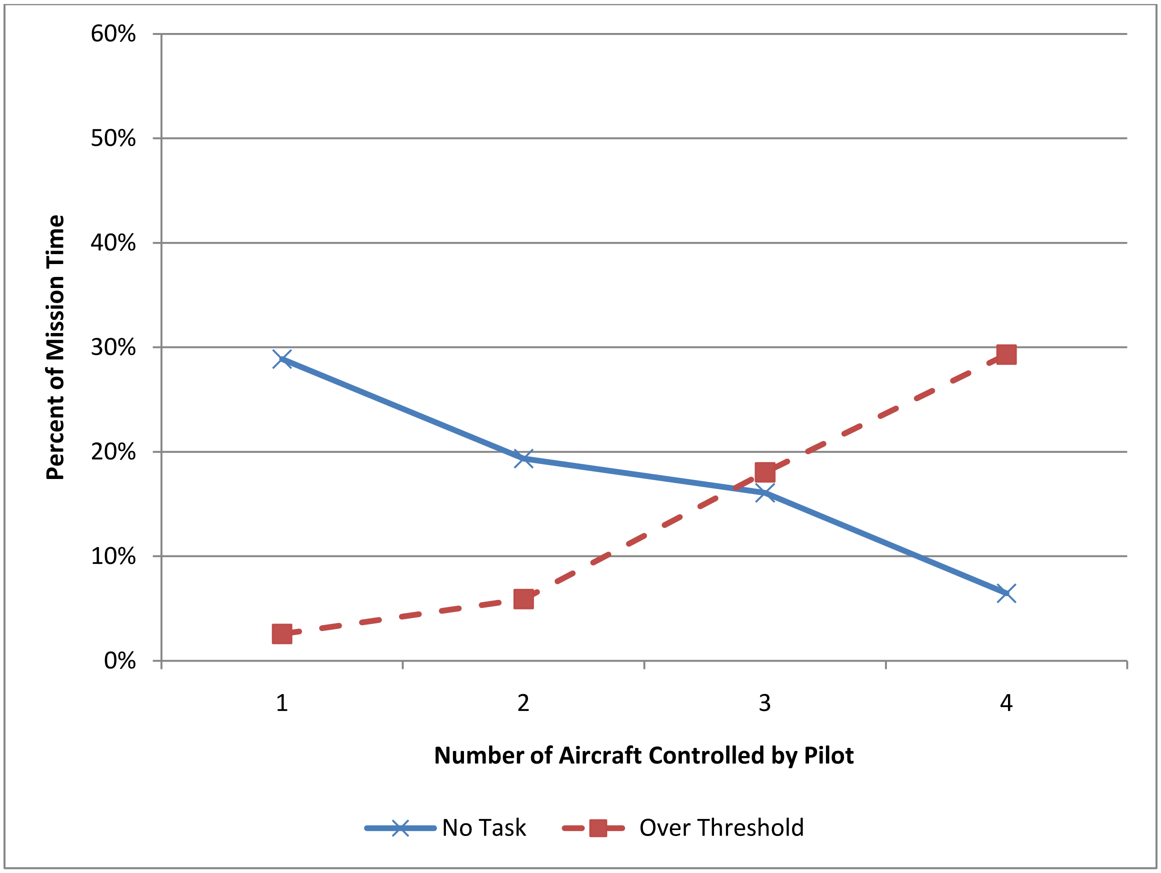


Figure 30. Benign w/Dynamic Untasked Time vs. Time Over Saturation Threshold of Workload Data from Model Output for Various MAC Ratios

Figure 30 characterizes the pilot untasked time vs. the time over saturation threshold for all aircraft in benign ISR with a single dynamic event occuring. Predictably, more time over saturation threshold occurs at lower MAC ratios. Under these conditions there isa potential for the degradation of pilot effectivness at a MAC ratio of 2. There is a steady increase in time over saturation threshold as the MAC ratio is increased to 3 and 4. A MAC ratio of 3 now represents the point where there is a significant potential for the degradation in pilot effectivness, as opposed to MAC ratio 4 for the all benign scenario in

Figure 29. Similarily the percent of pilot untasked time is much lower than is was for the benign scenario. Transitioning from no MAC to MAC ratio 2 there is no longer an efficient use of the pilot untasked time. It appears that with a single dynamic event the workload is now high enough that there is no longer an effective means of utilizing pilot untasked time.

At higher ratios of MAC the untasked time is reduced greatly fromcomparison to no MAC. This becomes a concern as this time is used to update the pilot's situational awareness. Although it is out of the scope of this analysis to predict that amount of untasked time necessary for a pilot to maintain situational awareness, it follows that as the MAC ratio increases so does the amount of time necessary to maintain situational awareness. Since higher MAC ratios should require more untasked time for system monitoring, but in fact have less this indicates another critical factor.

The effects of increasing the MAC ratio are complex with higher order interactions having more dominant roles. The increased probability of double and triple piloting task overlap drives conflict workload outside the saturation threshold of this analysis. Dynamic tasks further inflate the potnetial mission degredation and their overlap with other piloting tasks is unacceptable. Effective mission time is also decreased with increasing MAC ratio as the gaining and losing changeovers and handovers take more time during a shift. Untasked time and overload time concisely illustrate the trends of this data. Increasing the number of tasks the pilot performs, through addition of aircraft, results in more overload time due to task overlap induced conflict and less unoccupied time in which to mangage the workload.

V. Conclusions and Recommendations

5.1 Overview of Conclusions and Recommendations from MAC Research

The results from Chapter 4 indicate the presence of five main critical factors that have significant implications for the implementation of MAC: multi-task overlap, communication spikes, unplanned dynamic events, changeover constriction, and system monitoring. Each of these critical factors is addressed in detail in Section 5.2. These factors are not a comprehensive list of all of the factors which must be addressed to implement MAC; rather they are the factors which had the largest effect on the model output. Section 5.2 suggests some potential methods for addressing these critical factors. However, it should be noted that these suggestions are concepts that were either derived by the authors or suggestions made by members of the MQ-1B community and were not tested to determine their effectiveness.

5.2 Critical Factors and Implications of MAC

5.2.1 Multi-Task Overlap

Given the current GCS interface, direct multi-task overlap is impossible for most MQ-1B control tasks. The requirement for pilots to have their hands occupied in four separate places, or simultaneously look at two screens is impractical. However it should be restated that this analysis models mental workload and assumes the pilot can physically sequence the elements of the task so they are humanly possible. Multi-task overlap, in this context, is two or more coincident mental tasks. The mental workload conflict of overlap is addressed in Chapter 4 and is a major driver of workload values above the workload saturation threshold. Workload due to conflict dominates the total workload value. With double task overlap 60% of the workload is generated by conflict while with triple task overlap 75% of the workload is generated by conflict. The implications are clear, if MAC is to be realized multi-task overlap must be addressed.

While task automation may be an effective method of avoiding multi-task overlap, a common non-technology solution to multitasking involves user initiated workload mitigation strategies. These include task delegation, task rejection, task delay, and task switching. Underlying these strategies is a concept of priority. Each task must be weighed with respect to the other tasks, the mission context, criticality, and time sensitivity, to determine how to appropriately address it.

Task delegation involves assigning another crew member, typically the SO, to complete one of the overlapping tasks. The current MAC prototype relies heavily on delegation. During benign ISR operations the SO is given an altitude and airspace in which to direct the aircraft's course. This frees the pilot to engage other tasks and monitor the aircraft. However, in the event of an emergency or a dynamic ISR, several problems can arise. Since the pilot may not have full situational awareness of the aircraft or the mission and the SO is not fully trained as a pilot to be able to manage these events mission effectiveness may be degraded. This concern increases at higher MAC ratios as the pilot is required to interact effectively with the individual SOs while tracking more aircraft.

Task rejection is refusal to accept the new task or abandoning the current task for a higher priority task. In task rejection, the rejected task is not performed later; it is abandoned for a higher priority task. This assumes that the rejected task may eventually become moot if it is not performed. For example, if a benign ISR aircraft requires reorientation whilst another is in an emergency, and the pilot rejected the task imposed by the benign ISR aircraft, the pilot would ignore the reorientation request. While this strategy reduces the potential of multi-task overload, it also endangers the mission by increasing the potential of missing opportunities. In a time-critical mission like ISR, task rejection is seldom operationally realistic without decreasing mission effectiveness.

Alternatively, delaying tasks based on their priority could avoid multi-task overlap. Pushing the task off until a current higher priority tasks are complete may be acceptable in some benign situations when time sensitivity is less important. However, in time critical environments, task delay is only an option if the task could be executed later resulting in the same effects, otherwise it is task rejection.

Ideally overlapping tasks would be worked concurrently, task switching offers an approximation of concurrent task execution. If all overlapping tasks can sustain short duration delays with minimal mission degradation; then the pilot can switch from changing an aircraft course during transit, to altering an ISR orbit, or executing emergency procedures, and back to the original task. This process of task switching requires higher cognitive demand and is likely to increase short term memory requirements more than task delay, but it is a better solution, when available, because there is less time delay between completion of individual activities within each task.

Workload strategies are typically applied ad hoc. However to limit the frequency and effects of misapplication, the Air Force codifies them into Tactics, Techniques, and Procedures (TTPs). The presence of multi-task overlap in MAC necessitates a reevaluation of MQ-1B TTPs to ensure they allow for the benefits of all the workload mitigation strategies while codifying their proper application. These strategies can be modeled in IMPRINT in future research to inform the TTPs of MAC. Ultimately higher levels of automation are crucial to eliminating multi-task overlap, but in the immediate application, workload mitigation strategies are an effective solution to reducing the workload effect of multi task overlap.

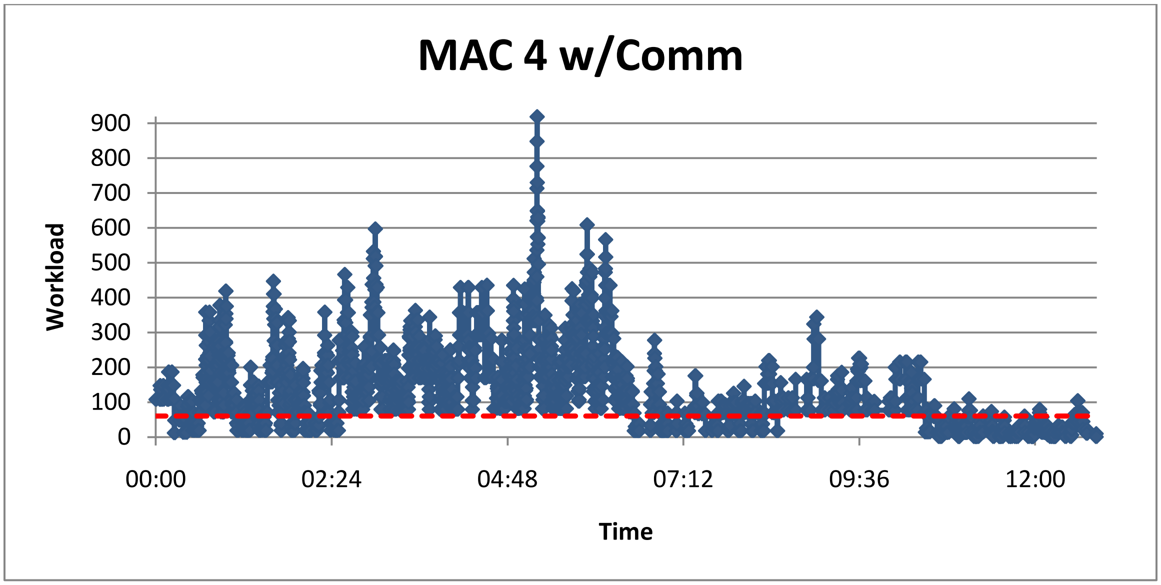
5.2.2 Communication Spikes

Communication is one of the biggest drivers of the extreme spikes in the workload traces. This finding is consistent with input from MQ-1B pilots who describe the communication load during dynamic operations as overwhelming (McGrogan & Schneider, 2010). It is important to understand the significance of the model output with respect to workload. The extreme spikes in workload caused by overlapping communication events are not necessarily the workload experienced by the pilot, rather it is the workload imposed by the system.

In a realistic scenario a person would carry on a single conversation at a time and would delay a second or third conversation or interrupt the first based on the criticality of each conversation or perhaps the immediacy demanded by the mode of communication. This is especially true of the real time text-based chat that is available to the pilot. It is easily possible to delay reading chat communication or delaying a response while the pilot is working on some other task. Some communication events are more critical than others and this model does not differentiate between time critical communication and routine communication. It would be a simple matter to delay a routine call from air traffic control during a conversation with a supported unit. However it would not be a simple matter to delay communication from one of multiple stakeholders that may be actively engaged in an ongoing dynamic mission. In a benign environment a pilot would be able to manage a routine communication while simultaneous performing a relatively simple task. However during a dynamic mission segment failing to respond quickly to a communication event may adversely impact an ongoing task. If the communication is critical the pilot will have to weigh the impact of responding to the communication or delaying such a response until there is a break in the workload.

It quickly becomes clear that communication is an area that requires active workload mitigation strategies at even low ratios of MAC. Some communication may be able to be offloaded to the sensor operator or the mission intelligence coordinator, but during multiple active missions, a pilot will now have to address with multiple sensor operators and mission intelligence coordinators who now compete for the pilot's attention. As the MAC ratio increases, simple communication offloading may not be sufficient or appropriate for resolving the additional workload. Unfortunately, there is no readily apparent technology solution to the communication challenge.

The GCS's suite of communications equipment makes the MQ-1B pilot accessible to a very wide range of stakeholders. The MQ-1B plays a pivotal role on the modern battlefield and therefore, numerous people are interested in the intelligence the MQ-1B provides. Reducing the frequency and number of sources of these communications events could be a first step to addressing the workload spikes.



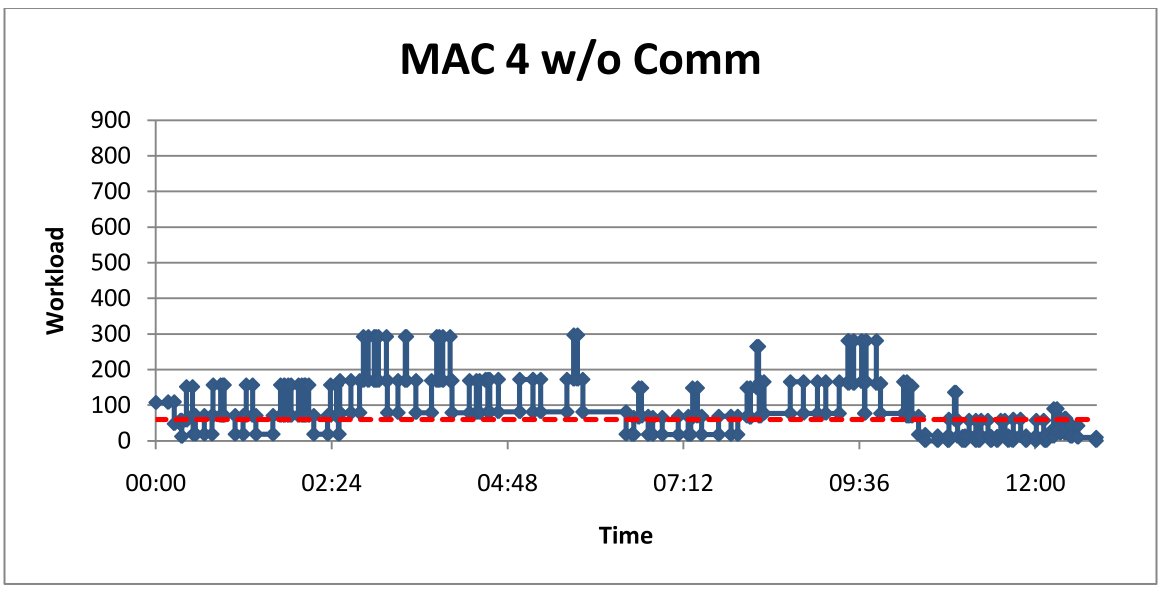


Figure 31. Comparison of Workload Trace with and without Communication of MAC Ratio 4 with a Dynamic mission

When communication is removed from the model entirely, the mean workload drops 23 points and the maximum drops by 622 points. Figure 31 compares the workload from runs with the same mission profile, one with normal communication and one without any communication. The most obvious difference is that workload spikes are eliminated during the mission and workload has only a few different discrete values. Communication adds variability to the workload as seen by the 803 point spread between the mean and maximum workload values with communication compared to the 204 point spread between the mean and the maximum workload values without communication. This spread is caused by the additional workload conflict as the pilot's attention is drawn away from the task of controlling the MQ-1B. It is unrealistic to think that the workload imposed by the system could ever be reduced to the workload trace without any communication, but it is important to strive to reduce the variability and conflict caused by all of these communication events. Internal to the GCS, it may be possible to consolidate and simplify the methods of communication to reduce the burden on the pilot. Also, if the underlying tasks of controlling the MQ-1B are simplified, that could in turn reduce the conflict generated from a simultaneous communications event.

There are no simple solutions to resolving the workload spikes generated from communication, but this is an area that warrants additional research. Future system development should have reducing the communication burden as one of the primary requirements.

5.2.3 Unplanned Dynamic Events

Unplanned dynamic events have a profound impact on the implementation of MAC. Based on discussions with MQ-1B pilots, the majority of dynamic ISR events are unplanned as they arise unpredictably during benign ISR mission segments (McGrogan & Schneider, 2010). It is not operationally feasible to avoid dynamic events when using

MAC, as these events are natural extensions of the ISR mission and potentially represent high priority missions. An operations concept for MAC should include robust procedures for resolving the eventuality of a dynamic event occurring during a benign ISR mission.

As demonstrated in Chapter 4, MAC is feasible for a MAC ratio of 3 providing that the missions are completely benign in nature. When a single 30 minute dynamic event is added to the mission profile a MAC ratio of 3 is no longer sustainable without significant potential for mission degradation. A MAC ratio of 2 is the highest achievable with a single 30 minute dynamic event inserted into the mission profile. Even then there are multiple spikes in workload above the saturation threshold which may endanger the mission. If the dynamic event were to last longer than 30 minutes, it is likely that even a MAC ratio of 2 would be hard to sustain. Since the dynamic events are part of the nature of the MQ-1B's mission, it is not possible to reduce their length or frequency. Instead the focus must be on how best to address a dynamic event when it arises during MAC.

A potential method for dealing with unexpected dynamic events is to use an on call pilot who can establish control of some of the aircraft that the MAC pilot is controlling. If this technique is used, then the on-call pilot will have to be onsite and able to take control of an MQ-1B on very short notice since the original pilot will have to maintain all of the aircraft until they can pass some of them off to another pilot. The problem with this technique is the lack of time for the transfer of situational awareness for the aircraft that the on-call pilot is taking control of. During a typical changeover the outgoing pilot gives the incoming pilot a detailed mission brief to avoid loss of situational awareness. During a dynamic event there is no time to perform the detailed mission brief so the on-call pilot would have to assume control of one or more MQ-1Bs with little to no background about the current mission or aircraft status.

An on call pilot will be able to perform the basic tasks related to aviating the aircraft so there is no chance of an aircraft loss; however, they will not know the details of the mission, airspace, or other allied units that may be involved with the mission. This technique for handling unexpected dynamic events carries the potential for mission degradation due to the lack of situational awareness of the on-call pilot.

5.2.4 Changeover Constriction

As noted in Section 4.2.4, increasing the MAC ratio decreases effective mission time. This is due to the increased time necessary to acquire and relinquish aircraft control. Unlike mission tasks which are executed concurrently, changeovers and handovers must be performed sequentially as the pilot is briefed on, assumes control of, and performs a systems check on each aircraft. Discussions with MQ-1B pilots revealed that a typical gaining changeover takes approximately 9 minutes, fence check approximately 8 minutes, and losing changeover around 7 minutes (McGrogan & Schneider, 2010), resulting in the effective loss of 24 minutes of mission time to effect a pilot change. When a pilot, controls only one aircraft, this 24 minutes of effective loss has minimal impact in a typical 150 to 180 minute shift. However, when this time is multiplied by three or four aircraft to permit the pilot to assume control of these aircraft during MAC, it takes an hour to assume full control of all of the aircraft potentially reducing the effective mission time for a single pilot to 90 to 120 minutes.

Such a reduction in effective mission time runs counter to the objectives of MAC which is designed to reduce the number of pilots necessary. If the shift length is constant, more pilots will be needed to conduct the same effective mission time. This assumes that there is time in the mission for the outgoing pilot to brief the incoming pilot which, based on the mission in Figure 28, there may not be. The higher rate of pilot turnover might result in the loss of situational awareness from one pilot to the next during changeovers.

A solution to this constriction is to overlap the changeover briefings as much as possible. Aircraft with common operational and tactical situations can be briefed more quickly as a whole. The result is to limit the use of MAC to situations in which the aircraft are operating jointly, which allows for a common tactical picture. Of course this may not frequently be possible in an unpredictable battle space with unanticipated dynamic events and new mission taskings.

Another possibility is to reduce the quantity of information necessary to check and brief. Currently the altitude of the aircraft must be checked in five different places to ensure the aircraft will execute commands as anticipated. Automation and interface improvements could solve this and other problems and reduce the changeover and fence check time.

5.2.5 System Monitoring

The underlying theory for the successful use of MAC is that pilot's untasked time can be used to control additional aircraft. Chapter 4 discusses how the additional tasks from controlling another aircraft decreases pilot untasked time while increasing the amount of time spent over the saturation threshold. However this assumes that pilot untasked time is wasted and can be put to better use doing something else. The flaw with this assumption is that it only takes into account the discrete tasks performed by the pilot and does not consider the continuous activities related to maintaining situational awareness of the aircraft and the mission.

If the workload traces are to be taken literally then when the pilot experiences no workload from the system they do not see, hear, touch, or think about anything. This is inaccurate, but the implications are not as obvious. While a pilot is experiencing zero workload from the system they will still be monitoring it. This may involve interacting with the mouse or keyboard and accessing different display screens for system status information. The pilot will also be performing various cognitive tasks, including planning future actions with respect to various mission scenarios and aircraft constraints.

The anticipating and planning tasks necessary for effective mission execution, are not captured in this model. The workload trace represents the workload imposed on the pilot by the system, but that does not mean that each of these tasks is initiated by the system. The pilot does not passively wait for an external trigger before performing necessary tasks. A pilot will need to be proactive, resolving tasks before they become critical and predicting external events and planning for different eventualities.

It is a misnomer to characterize the time spent with no workload as “idle” since the pilot will often be performing preparatory tasks that would be very difficult to characterize in an IMPRINT workload model, hence the term untasked is used in this analysis instead of idle. It is certainly true that excessive time without performing any system tasks may include some actual idle time, but that is not true of all of the down time between tasks. Reduction in the down time between tasks reduces the pilot's ability to monitor the system and perform cognitive tasks to plan future portions of the mission.

This analysis is not sufficient to determine the down time required to allow the pilot to maintain situational awareness of the aircraft and the system to perform the necessary planning tasks. However as the MAC ratio increases so do the requirements for maintaining situational awareness and planning. It would be logical to assume that there may be conflict generated by maintaining situational awareness on multiple aircraft. There is certainly a danger of getting details of different aircraft and missions confused. Pilots may have to reduce their level of situational awareness on each aircraft to simultaneously maintain situational awareness of all of the aircraft. Otherwise they may risk making a mistake because they confused the status of two different aircraft. This obviously has implications for maintaining pilot effectiveness when using MAC.

5.3 Recommendations for Future MAC Research

There are numerous expansions and extensions to this research on MAC for the MQ-1B. The model is currently designed to represent the workload imposed upon on the pilot by the system rather than the workload the pilot actually experiences. To expand the model further and examine how the pilot actually manages the workload it will be necessary to change the model to allow for realistic task accomplishment instead of unlimited simultaneous task execution as is represented in the existing model.

5.3.1 Model Operations Crew

The scope for the simulation and the analysis can be expanded to cover additional areas of this subject matter. This model was limited to the MQ-1B pilot and treated other members of the operations crew as external to the system. The model can be expanded to include both the sensor operator and the mission intelligence coordinator. This will allow the entire operations crew to work as a team to perform the mission. By modeling the operations crew rather than the pilot, it would be possible to investigate the ripple effect caused by a delay by any member of the crew impacting the task completion of another member of the crew. Alternate crew sizes and responsibilities could also be explored to optimize workload in the MAC paradigm.

5.3.2 Mitigation Strategies

There should be further analysis of the impact of different operations concepts and tactics, techniques and procedures on MAC operations. This analysis uses existing operations concepts and postulates the use of mitigation strategies for reducing the workload experienced by the pilot. Further research on the processes of task prioritization, delegation, rejection, delay, and switching can determine their effectiveness for dealing with excessive workload and how best to implement these strategies in future UAS operations concepts.

5.3.3 Manpower Studies

When the data from the preliminary manpower analysis in Chapter 2 is compared with the data from Chapter 4, where increasing levels of MAC carry increasing potential for mission degradation, there is a stark cost/benefit tradeoff at higher (e.g., greater than 2) MAC levels. Higher levels of MAC produce diminishing manpower savings while the potential for mission degradation increases substantially. Implementing higher levels of MAC are in effect getting less manpower savings for a greater cost. Further research is necessary to understand the operational and human-systems integration implications of MAC.

5.3.4 Human Validation of Multi-Aircraft Data

Due to the limited data available for MAC of the MQ-1B, this model was only validated for a pilot controlling a single aircraft. All of the MAC data is an extrapolation on the single aircraft model, from which it derives validity. To provide greater confidence in the MAC data for this model, further validation should be accomplished to compare the MAC data to actual human performance while controlling multiple MQ-1Bs.

5.3.5 Automation in MAC

One of the limitations of using the MQ-1B for MAC is the limited amount of automation in the system. The current level of automation was developed under the paradigm of aiding a pilot in controlling a single aircraft. There should be future research regarding the implementation of additional automation to allow for limited levels of decision making within the MQ-1B control system. Future automation should facilitate a shift in the paradigm to that of a single pilot having supervisory control of multiple MQ-1Bs. Under supervisory control a pilot would have broad knowledge over the high level status of multiple aircraft rather than detailed knowledge of a single aircraft. A pilot would be able to monitor and direct the MQ-1Bs as they performed the mission semi-autonomously.

5.3.6 Impact of Workload on Task Completion

This model currently presents perfect task completion for all tasks. The implications of task failure must them be inferred from the data. It would be valuable to expand the model to account for the chances of task failure and the effect on mission effectiveness. As workload increases so would the chance of task failure.

5.3.7 Modeling Situational Awareness

This analysis lacks a model of how an MQ-1B pilot maintained situational awareness of the aircraft they were controlling. The amount of information a pilot would need to keep track of would increase with each aircraft they controlled while the amount of time they had to monitor that information would decrease.

5.3.8 Workload Reduction Modalities

During development of this thesis, it was realized that Multiple Resource Theory (MRT) or related methods can be used to estimate three different metrics of workload. Further each of these metrics, and perhaps the relationships among these metrics, might be relevant to system design. The first metric is used to quantify a value referred to as the channel task demand. Each task requires a specific amount of resources in the available channels (visual, auditory, speech, cognitive, fine motor, gross motor and tactile). Channel task demand workload represents the demand imposed on the human information processing resources by the system interface, regardless of human limitations. Channel task demand workload values are based on task difficulty and are a sum of the model inputs for a specific task.

At the opposite extreme is a metric referred to as the system imposed workload. This metric accounts for the intricacies of human information processing and is calculated directly by MRT. It also assumes that the operator must perform all tasks as they become available, and does not account for the limitations of human performance.

The system imposed workload includes not only the channel task demand workload, but the conflict workload generated in, and between, each human information processing channel, assuming that the work must be performed by the user as the task is presented to the user. System imposed workload values are used in this analysis to assess the potential for mission degradation during various modalities.

The third metric is referred to as the execution workload, which represents the workload under which the operator functions. The execution workload value represents the workload of the tasks as they are conducted by a human within the limitations of human performance. This quantity recognizes that the user has a finite ability to respond to workload demands. As such, the execution workload value cannot exceed the saturation threshold. Under conditions in which the system imposed workload exceeds the execution workload, the user will be forced to implement one or more of the workload mitigation strategies, which might result in suboptimal task performance. Execution workload correlates to empirical data, but is difficult to model due to the abundance of mitigation strategies and their specific application during a mission.

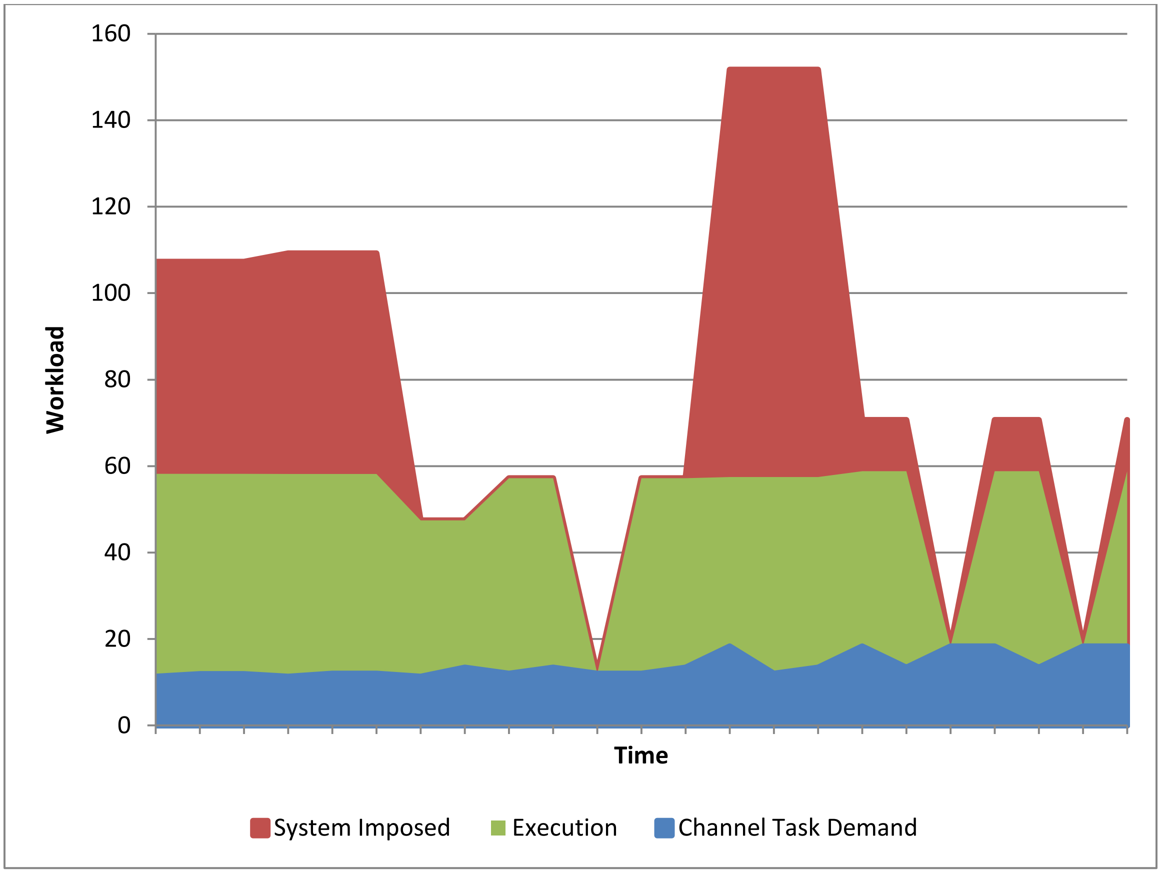


Figure 32. Comparison of Simulated Workload Trace from Channel Task Demand, System Imposed, and Actual Execution Metrics

An example of the three workload metrics is depicted in Figure 32. Note that in certain portions of the workload traces the system induced workload and execution workload are identical when the workload is below the saturation threshold. All three metrics are coincident if no conflict is present. The recognition of the presence of these three metrics implies a novel approach to workload management, because workload reduction can take a form consistent with each of these models. A common method for managing workload is channel task demand, which reduces task complexity or eliminates tasks to be performed by the operator. This type of reduction can be accomplished through automation or by changing the format of information input and output to reduce the complexity of the processing necessary to effect the transformation. This workload reduction method is a valid form of decreasing workload, but there are other options. The conflict between tasks can also be reduced through optimal task allocation and interface integration specifically designed to reduce conflict. Through this method, rather than simplifying the tasks, the tasks are altered to minimize the conflict experienced by performing multiple tasks simultaneously. Information can be shifted from one channel to another channel to avoid an overloaded channel or reduce the conflict between channels. Lastly, execution workload can be reduced through the development of Tactics, Techniques, and Procedures (TTPs) for use as workload strategies to keep the workload below the saturation threshold through task prioritization, shedding, and delay.

The result of this analysis suggests that the appropriate method to reduce workload for MAC is not to focus solely on the task demand difficulty; rather the conflict generated workload between the different channels of concurrent tasks must also be addressed. A thorough analysis is required on how this task conflict can be addressed and how these modifications will impact the effectiveness of MQ-1B MAC operations.

5.4 Summary of MAC Research Conclusions and Recommendations

It can be challenging to predict the complexities of a paradigm shift like MAC. On the surface the concept seems straightforward, but upon detailed analysis numerous critical factors are revealed. MAC does more than increase the number of aircraft a pilot can control; MAC also changes the way that the aviation community has thought about piloting for over 100 years. This analysis identified five critical factors that significantly impact the pilot’s ability to maintain mission effectiveness under MAC: multi-task overlap, communication spikes, unplanned dynamic events, changeover constriction, and systems monitoring. These critical factors are consistent with concerns expressed by pilots discussing MAC and should be addressed in future architecture and systems development. Further study is necessary to fully characterize the impact of MAC on mission effectiveness and the implications of optimizing system induced workload through adaptive modality selection.

Appendix A: Phase I Run Matrix

Table 2. Phase I Run Matrix

Run Number	MAC Ratio	Gaining	2	3	4	5	6	Terminal	Losing
1	1	Handover	Transit	Dynamic ISR	Strike	Emergency		Benign ISR	Changeover
2	2	Changeover	Transit	Emergency	Benign ISR	Dynamic ISR		Benign ISR	Changeover
		Changeover	Transit	Transit	Dynamic ISR	Emergency		Benign ISR	Changeover
3	2	Handover	Emergency	Dynamic ISR	Emergency	Dynamic ISR		Transit	Handover
		Handover	Benign ISR	Dynamic ISR	Emergency	Transit		Benign ISR	Handover
4	3	Changeover	Benign ISR	Transit	Dynamic ISR	Emergency	Transit	Benign ISR	Changeover
		Changeover	Benign ISR	Dynamic ISR	Dynamic ISR	Emergency	Benign ISR	Benign ISR	Changeover
		Changeover	Emergency	Emergency	Emergency	Benign ISR	Dynamic ISR	Benign ISR	Changeover
5	3	Handover	Benign ISR	Transit	Dynamic ISR	Emergency	Transit	Benign ISR	Handover
		Handover	Benign ISR	Transit	Dynamic ISR	Emergency	Benign ISR	Benign ISR	Handover
		Handover	Dynamic ISR	Transit	Benign ISR	Dynamic ISR	Emergency	Transit	Handover
6	3	Changeover	Transit	Transit	Dynamic ISR	Emergency	Benign ISR	Transit	Changeover
		Handover	Transit	Transit	Dynamic ISR	Emergency	Dynamic ISR	Transit	Changeover
		Handover	Dynamic ISR	Emergency	Transit	Transit	Emergency	Benign ISR	Handover

7	4	Changeover Changeover Changeover Changeover	Transit Transit Dynamic ISR Benign ISR	Transit Transit Dynamic ISR Emergency	Transit Transit Dynamic ISR Dynamic ISR	Transit Transit Transit Emergency	Transit Transit Transit Dynamic ISR	Transit Transit Transit Transit	Changeover Changeover Changeover Changeover
8	4	Handover Handover Handover Handover	Transit Dynamic ISR Benign ISR Benign ISR	Transit Dynamic ISR Emergency Benign ISR	Transit Dynamic ISR Dynamic ISR Benign ISR	Transit Transit Emergency Benign ISR	Transit Transit Emergency Emergency	Transit Transit Transit Benign ISR	Handover Handover Handover Handover
9	4	Changeover Changeover Handover Handover	Dynamic ISR Benign ISR Benign ISR Benign ISR	Dynamic ISR Emergency Benign ISR Benign ISR	Dynamic ISR Dynamic ISR Benign ISR Benign ISR	Transit Emergency Benign ISR Benign ISR	Transit Emergency Emergency Benign ISR	Transit Transit Benign ISR Benign ISR	Changeover Changeover Handover Handover
10	4	Changeover Changeover Changeover Changeover	Emergency Emergency Benign ISR Benign ISR	Emergency Benign ISR Benign ISR Benign ISR	Benign ISR Benign ISR Benign ISR Benign ISR			Transit Benign ISR Benign ISR Benign ISR	Handover Handover Handover Handover

Appendix B: Phase I Workload Graphs

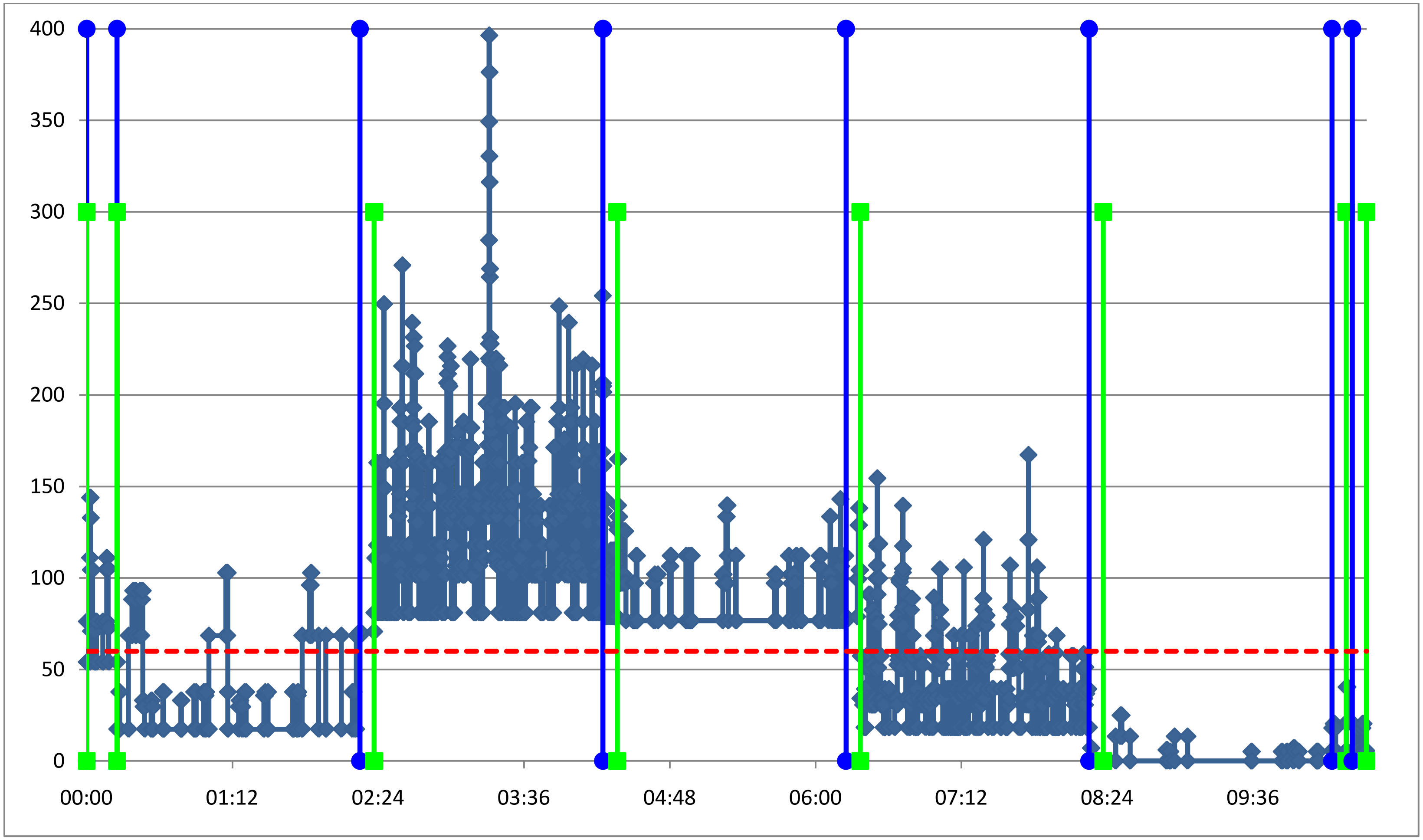


Figure 33. Phase I Run 3 MAC 2 Workload Graph

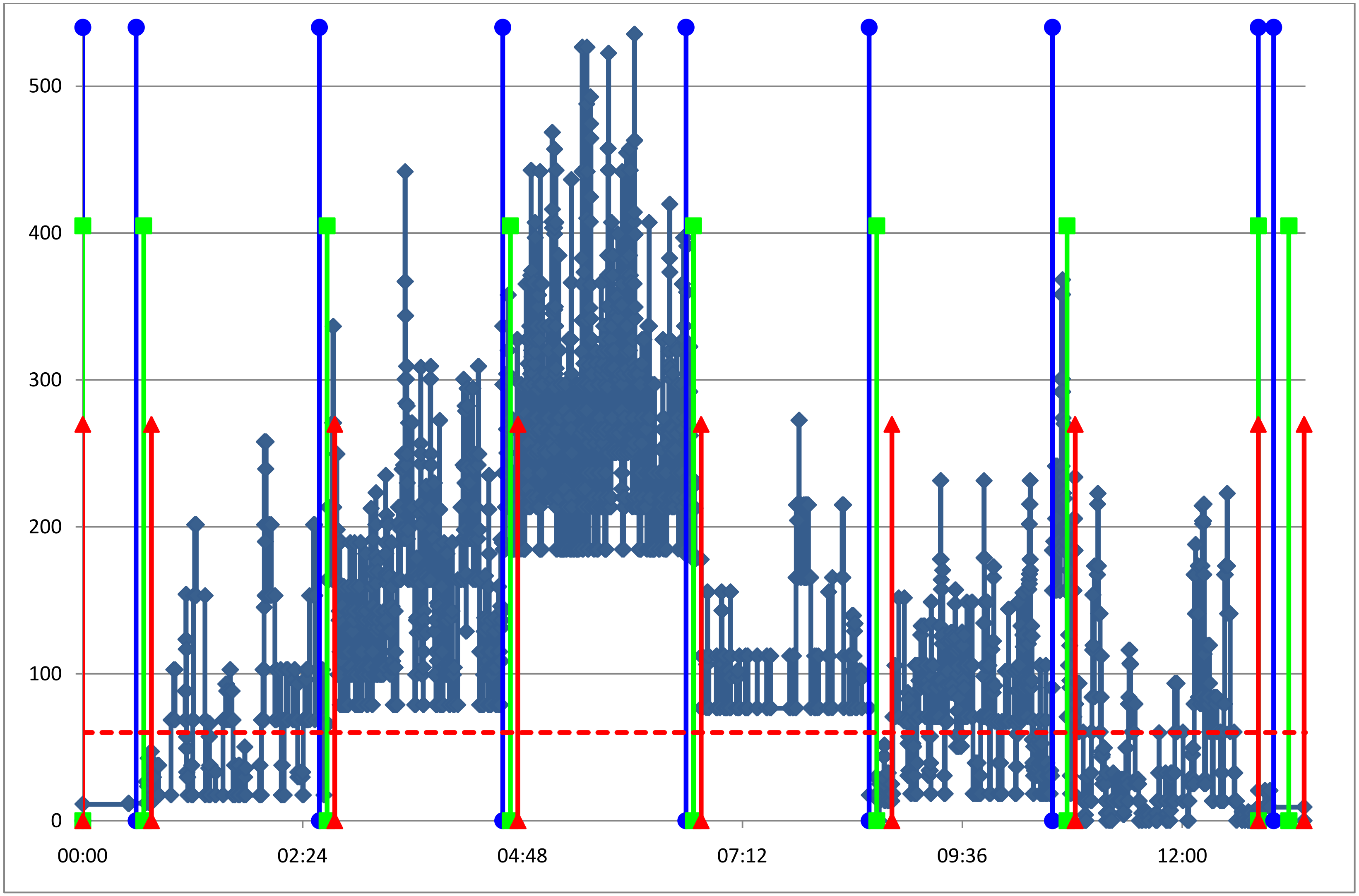


Figure 34. Phase I Run 4 MAC 3 Workload Graph

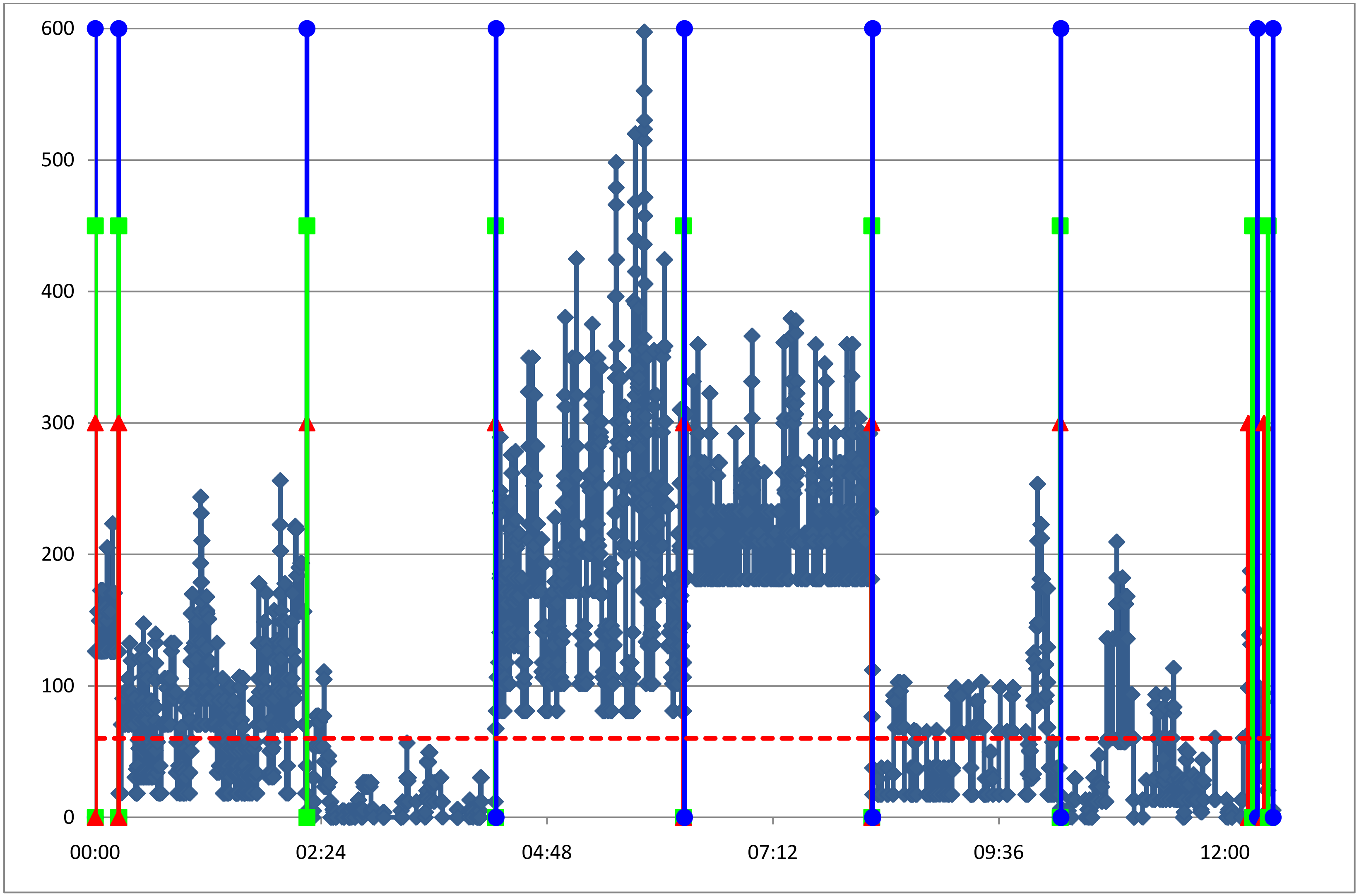


Figure 35. Phase I Run 5 MAC 3 Workload Graph

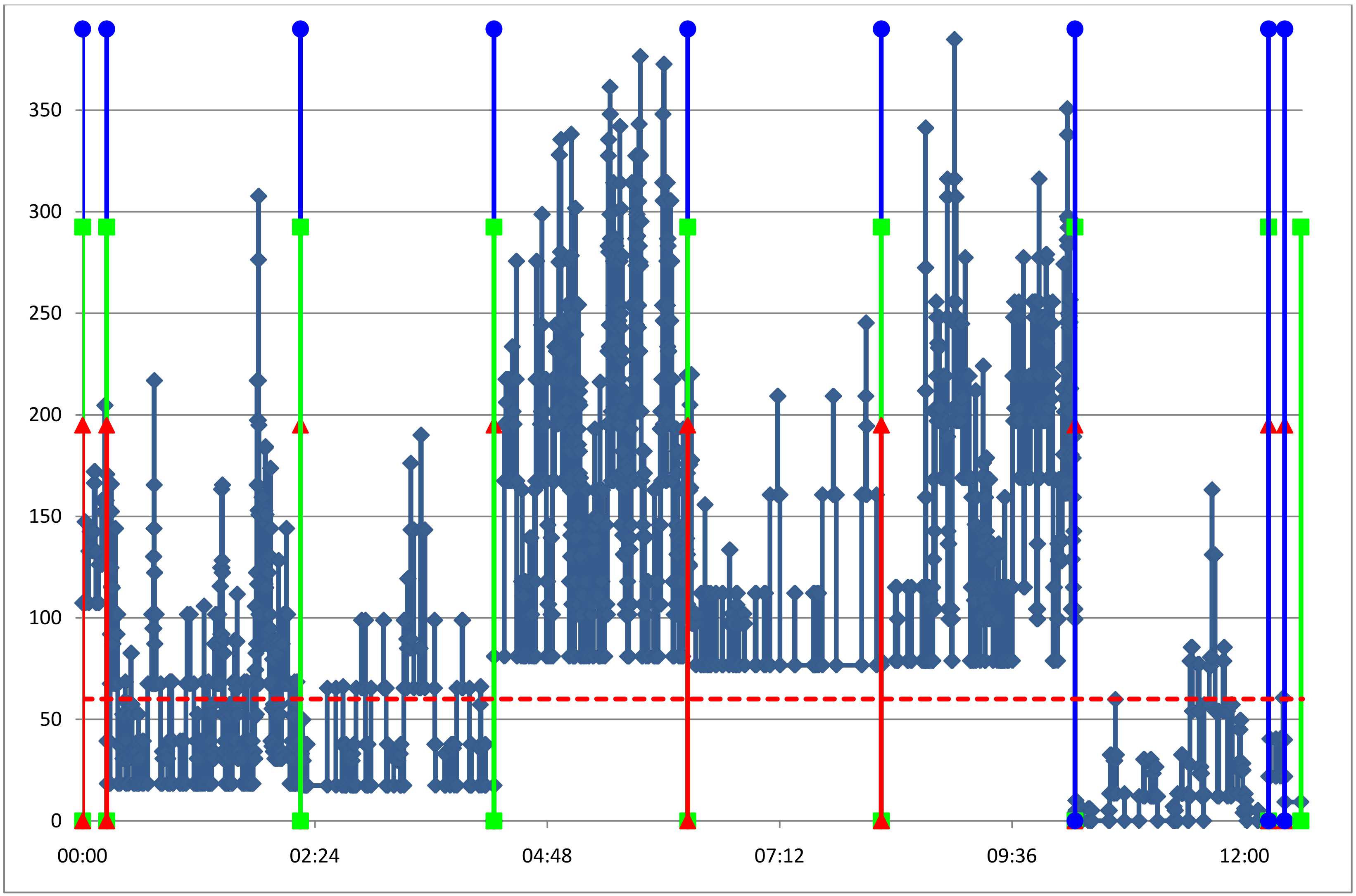


Figure 36. Phase I Run 6 MAC 3 Workload Graph

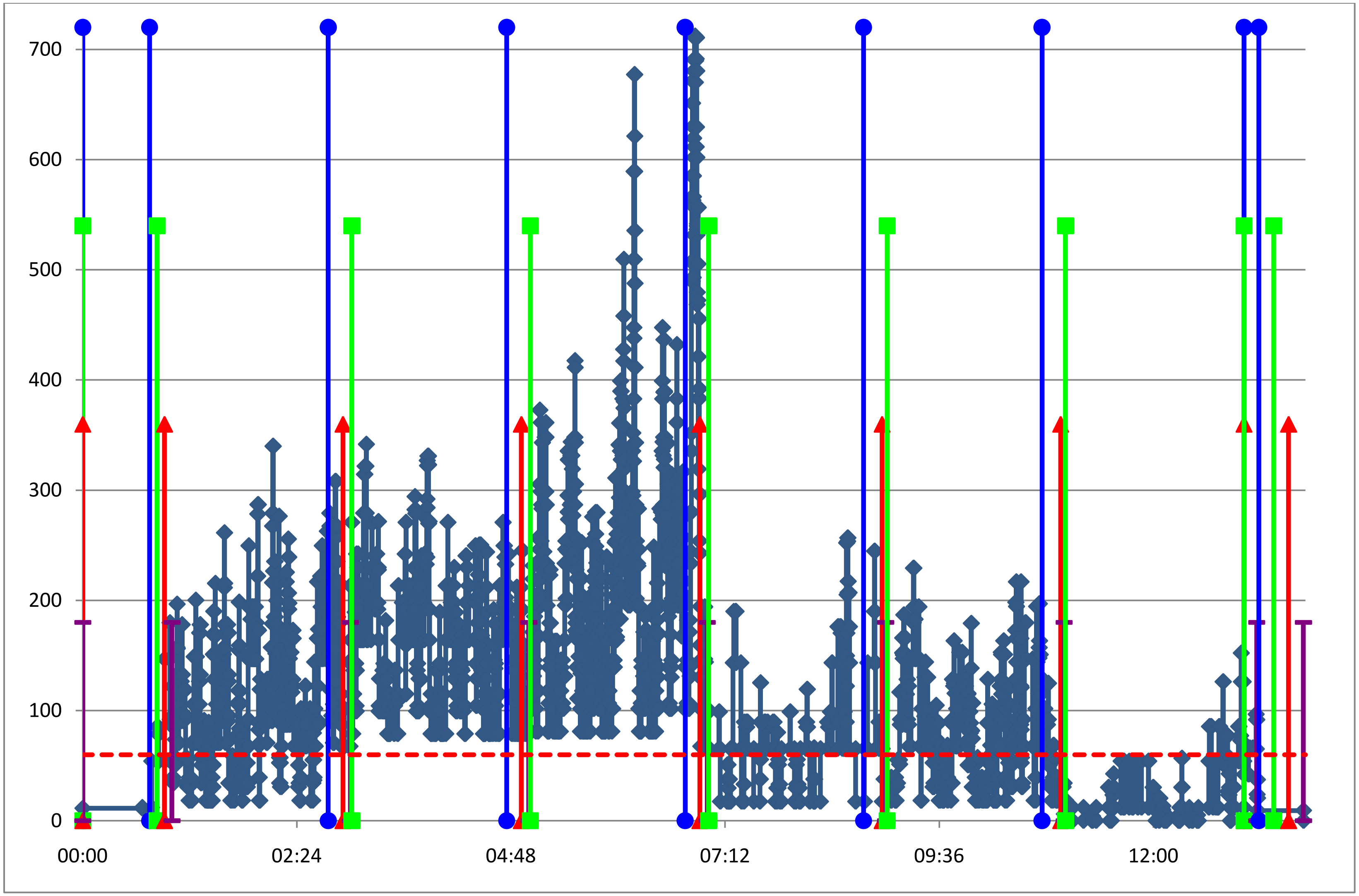


Figure 37. Phase I Run 7 MAC 4 Workload Graph

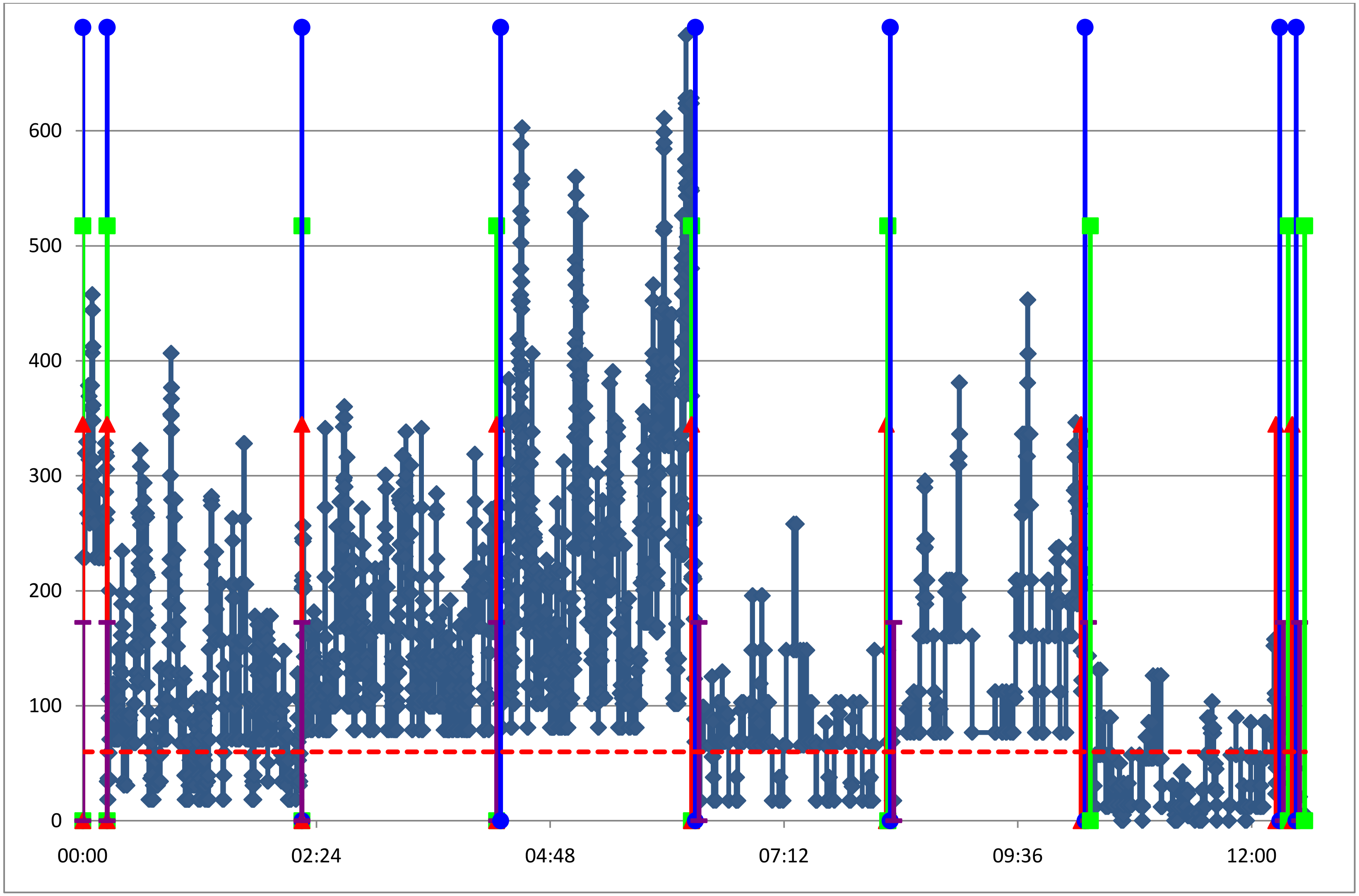


Figure 38. Phase I Run 8 MAC 4 Workload Graph

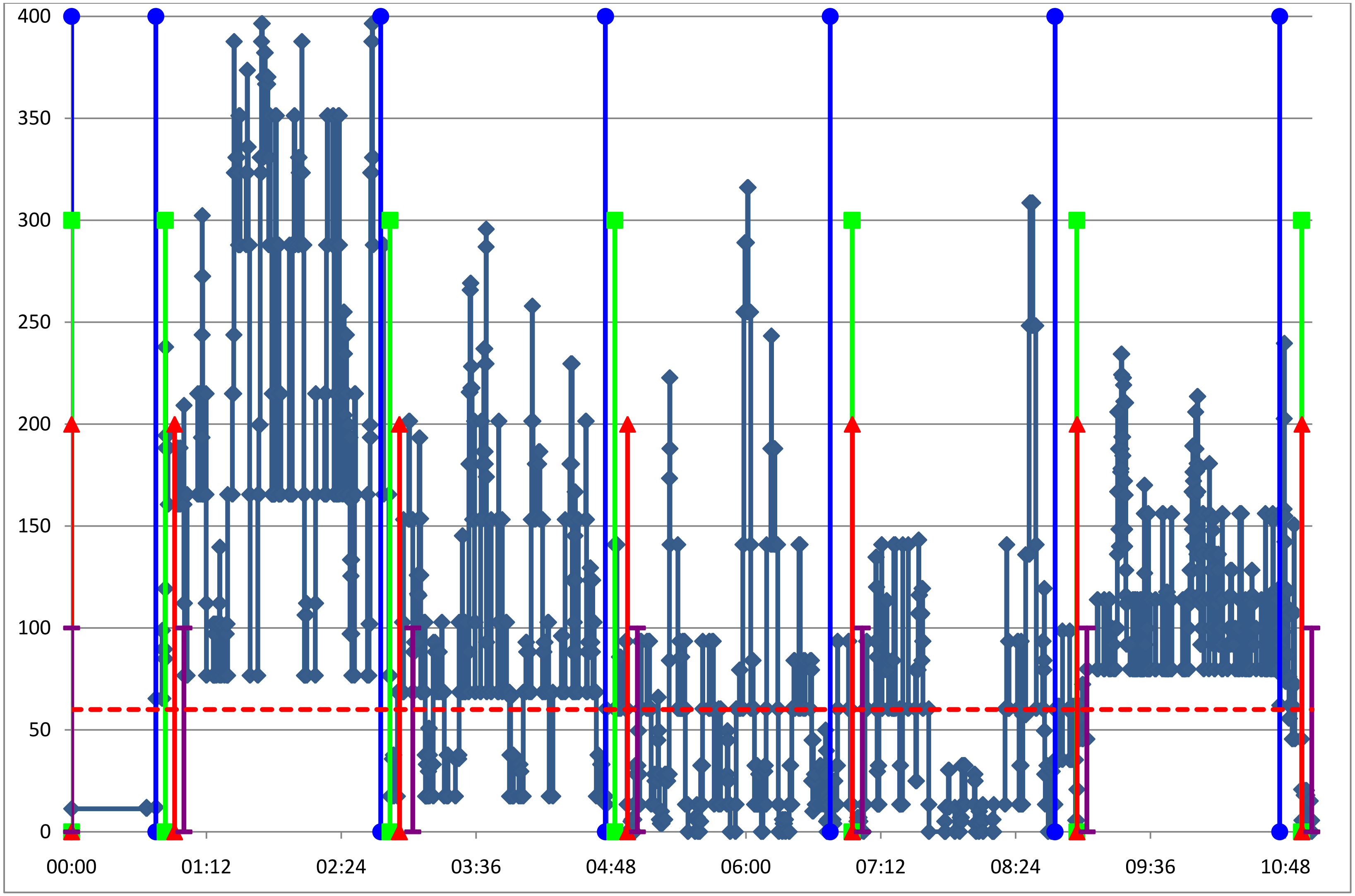


Figure 39. Phase I Run 10 MAC 4 Workload Graph

Appendix C: Phase II Run Matrix

Table 3. Phase II Run Matrix

Run Number	MAC Ratio	Gaining	2	3	4	5	6	Terminal	Losing
1	1	Handover	Transit	Dynamic ISR	Strike	Emergency		Benign ISR	Changeover
2	2	Changeover	Transit	Emergency	Benign ISR	Dynamic ISR		Benign ISR	Changeover
		Changeover	Transit	Transit	Dynamic ISR	Emergency		Benign ISR	Changeover
3	2	Handover	Emergency	Dynamic ISR	Emergency	Dynamic ISR		Transit	Handover
		Handover	Benign ISR	Dynamic ISR	Emergency	Transit		Benign ISR	Handover
4	3	Changeover	Benign ISR	Transit	Dynamic ISR	Emergency	Transit	Benign ISR	Changeover
		Changeover	Benign ISR	Dynamic ISR	Dynamic ISR	Emergency	Benign ISR	Benign ISR	Changeover
		Changeover	Emergency	Emergency	Emergency	Benign ISR	Dynamic ISR	Benign ISR	Changeover
5	3	Handover	Benign ISR	Transit	Dynamic ISR	Emergency	Transit	Benign ISR	Handover
		Handover	Benign ISR	Transit	Dynamic ISR	Emergency	Benign ISR	Benign ISR	Handover
		Handover	Dynamic ISR	Transit	Benign ISR	Dynamic ISR	Emergency	Transit	Handover
6	3	Changeover	Transit	Transit	Dynamic ISR	Emergency	Benign ISR	Transit	Changeover
		Handover	Transit	Transit	Dynamic ISR	Emergency	Dynamic ISR	Transit	Changeover
		Handover	Dynamic ISR	Emergency	Transit	Transit	Emergency	Benign ISR	Handover

7	4	Changeover	Transit	Transit	Transit	Transit	Transit	Transit	Changeover
		Changeover	Transit	Transit	Transit	Transit	Transit	Transit	Changeover
		Changeover	Dynamic ISR	Dynamic ISR	Dynamic ISR	Transit	Transit	Transit	Changeover
		Changeover	Benign ISR	Emergency	Dynamic ISR	Emergency	Dynamic ISR	Transit	Changeover
8	4	Handover	Transit	Transit	Transit	Transit	Transit	Transit	Handover
		Handover	Dynamic ISR	Dynamic ISR	Dynamic ISR	Transit	Transit	Transit	Handover
		Handover	Benign ISR	Emergency	Dynamic ISR	Emergency	Emergency	Transit	Handover
		Handover	Benign ISR	Benign ISR	Benign ISR	Benign ISR	Emergency	Benign ISR	Handover
9	4	Changeover	Dynamic ISR	Dynamic ISR	Dynamic ISR	Transit	Transit	Transit	Changeover
		Changeover	Benign ISR	Emergency	Dynamic ISR	Emergency	Emergency	Transit	Changeover
		Handover	Benign ISR	Benign ISR	Benign ISR	Benign ISR	Emergency	Benign ISR	Handover
		Handover	Benign ISR	Benign ISR	Benign ISR	Benign ISR	Benign ISR	Benign ISR	Handover
10	4	Changeover	Emergency	Emergency	Benign ISR			Transit	Handover
		Changeover	Emergency	Benign ISR	Benign ISR			Benign ISR	Handover
		Changeover	Benign ISR	Benign ISR	Benign ISR			Benign ISR	Handover
		Changeover	Benign ISR	Benign ISR	Benign ISR			Benign ISR	Handover

Appendix D: Phase II Workload Graphs

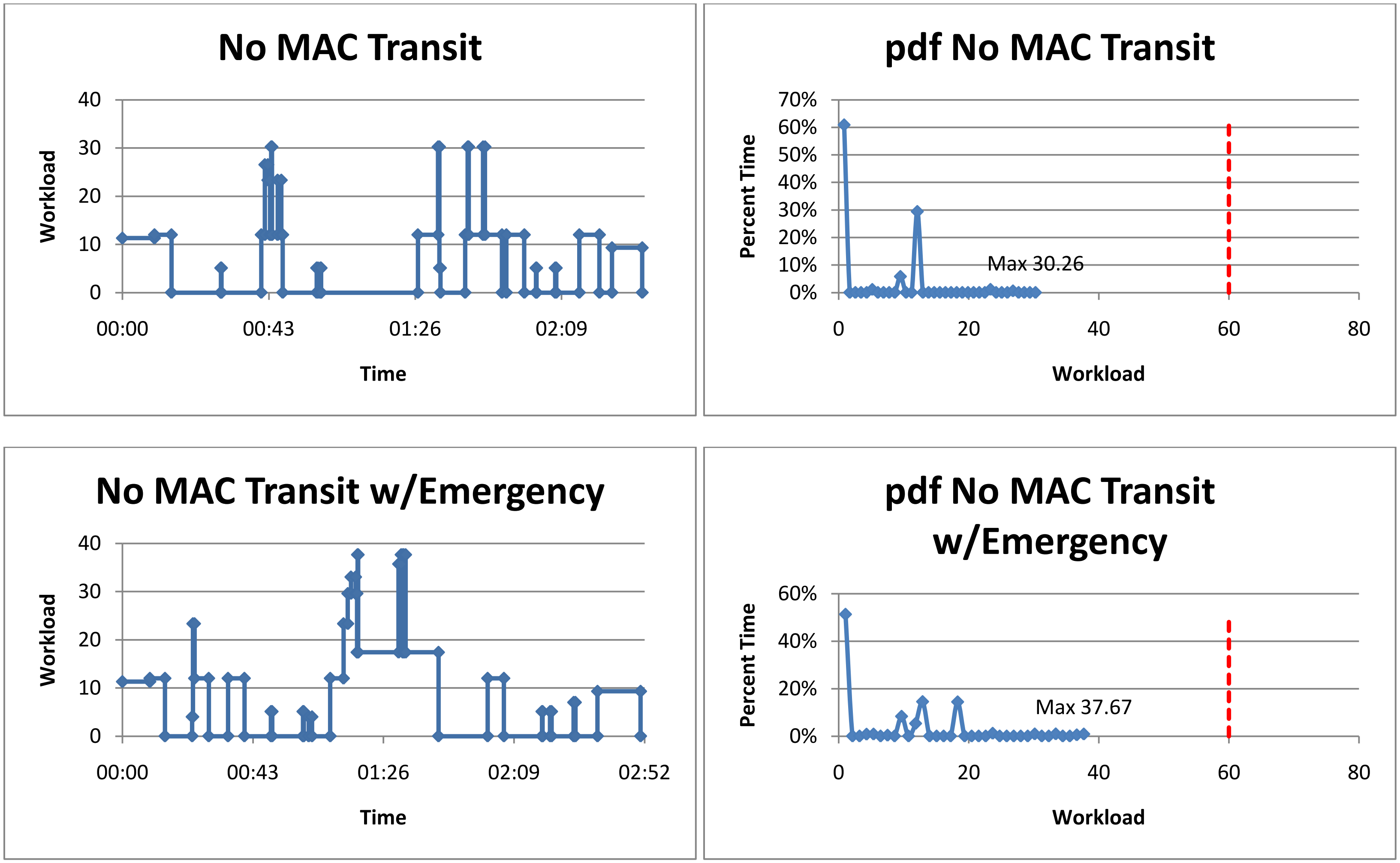


Figure 40. No MAC Data Comparison

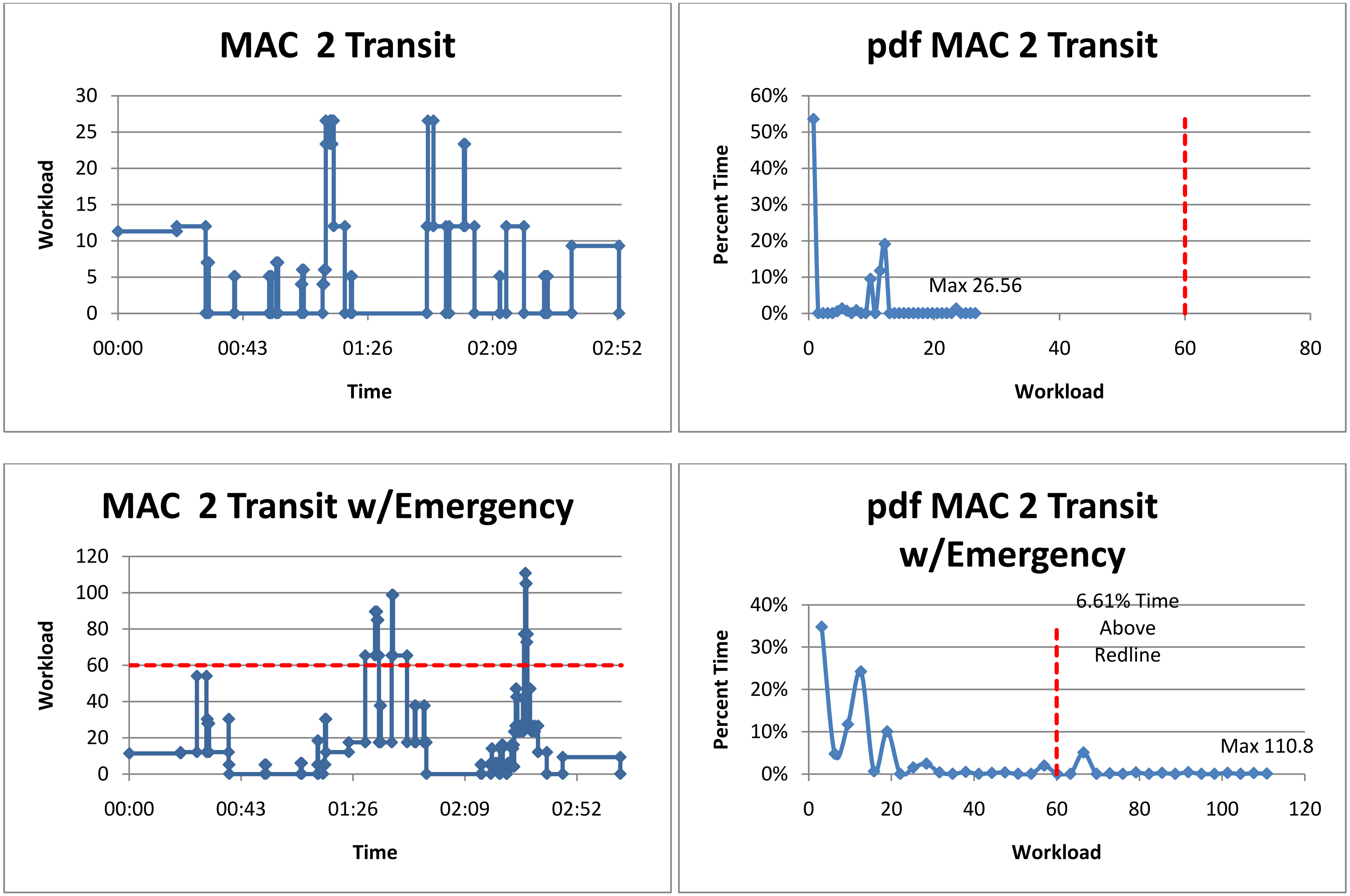


Figure 41. MAC 2 Data Comparison

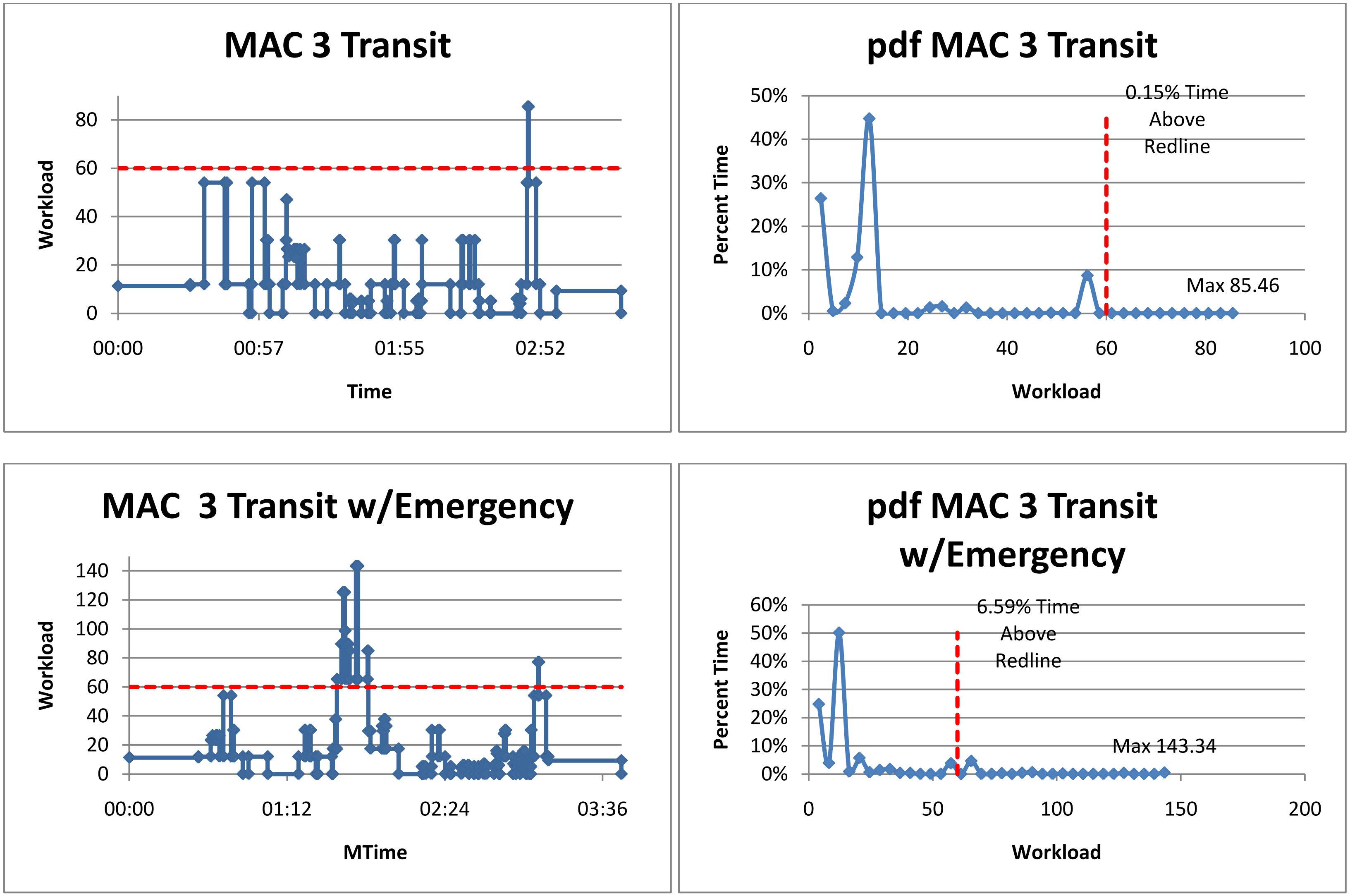


Figure 42. MAC 3 Data Comparison

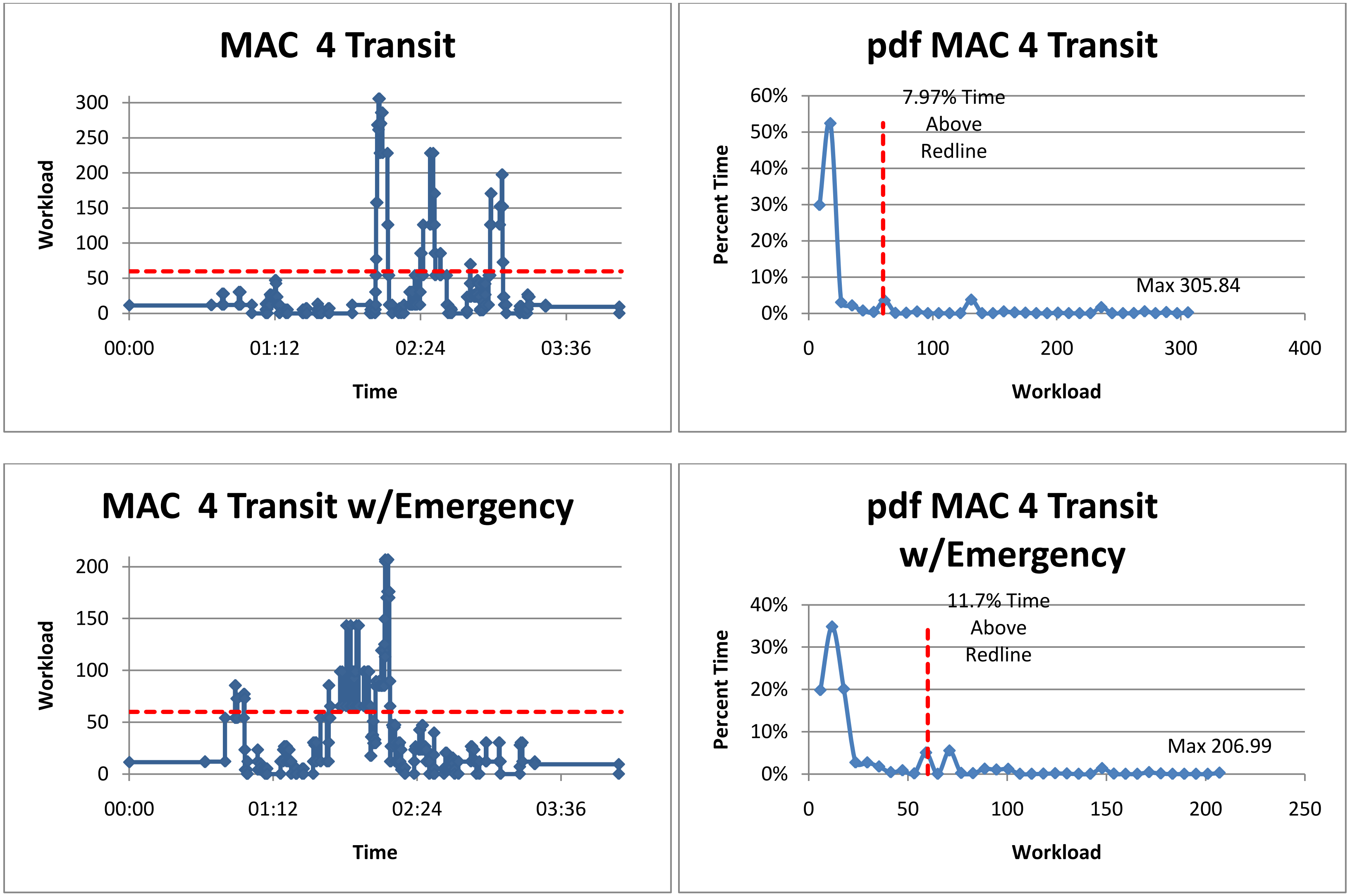


Figure 43. MAC 4 Data Comparison

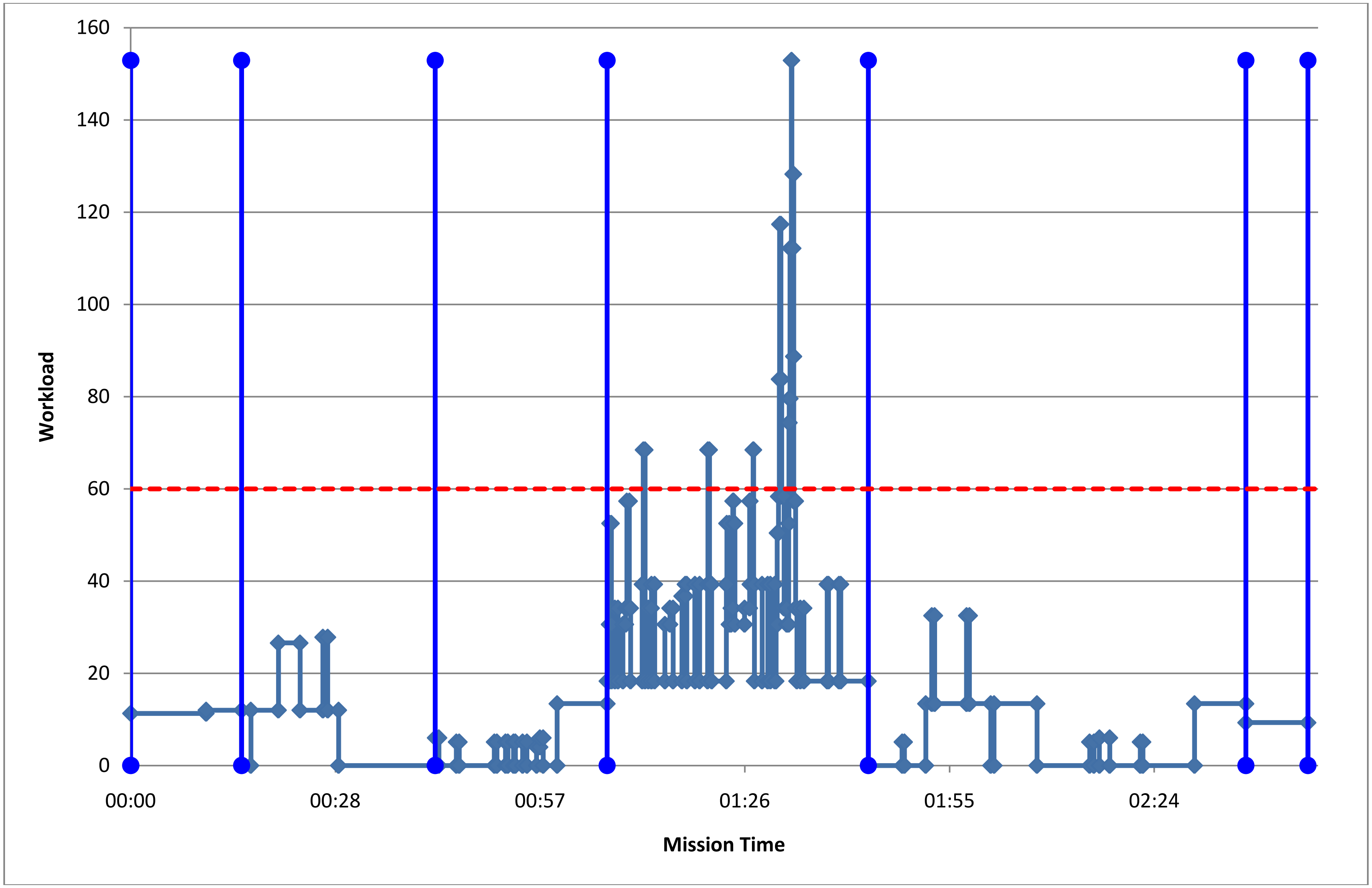


Figure 44. Phase II Run 1 No MAC Workload Graph

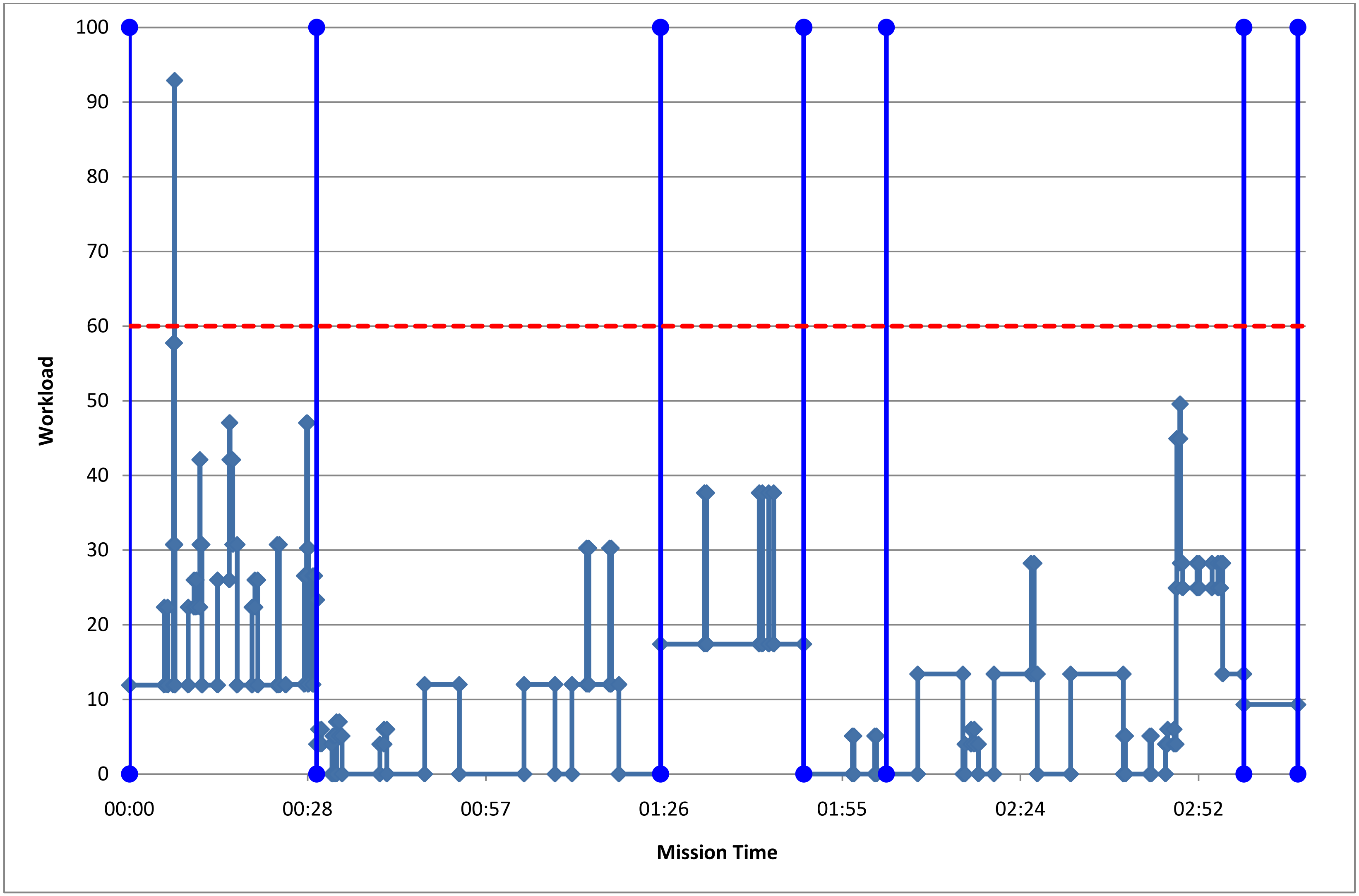


Figure 45. Phase II Run 2 No MAC Workload Graph

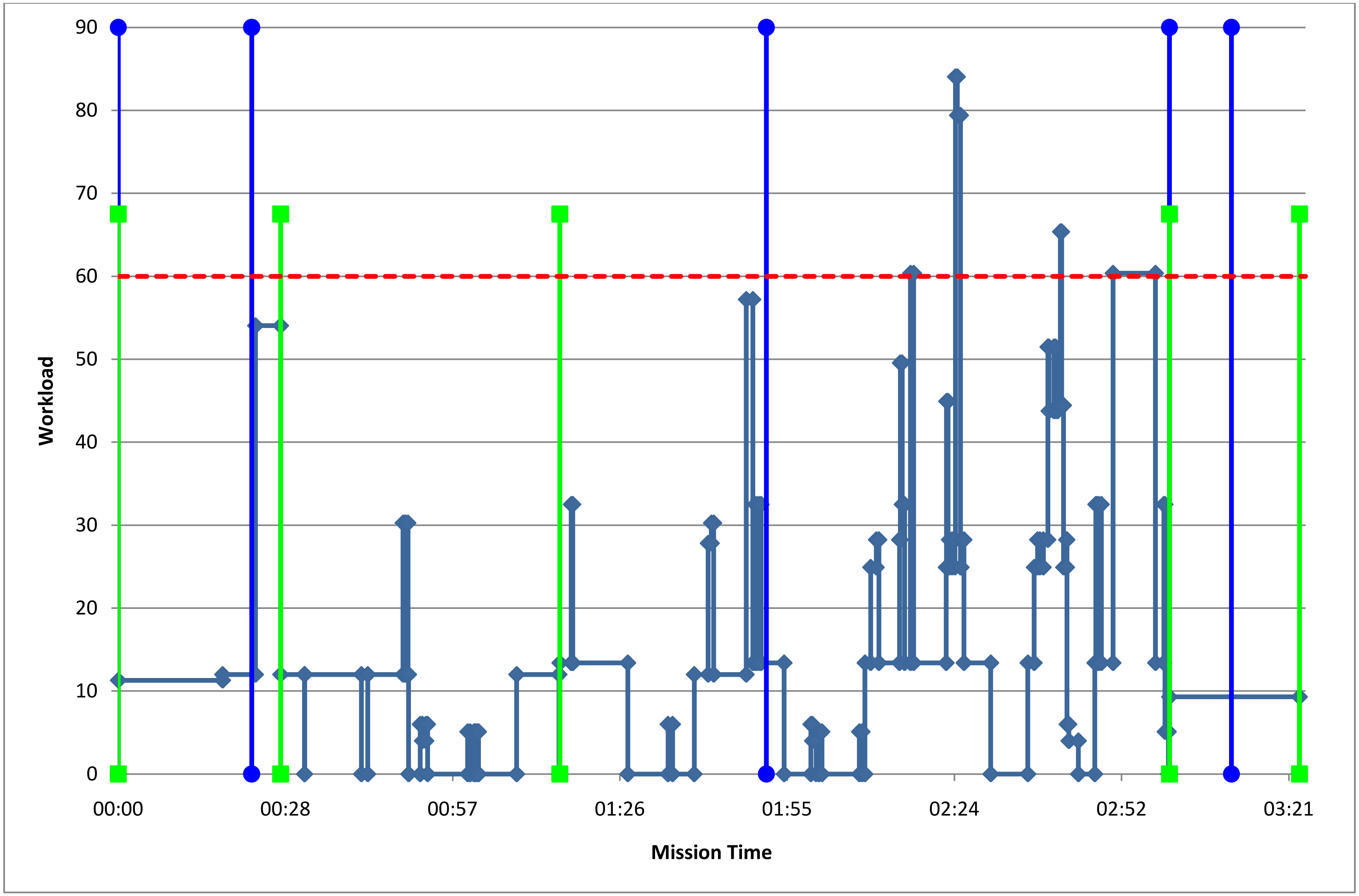


Figure 46. Phase II Run 4 MAC 2 Workload Graph

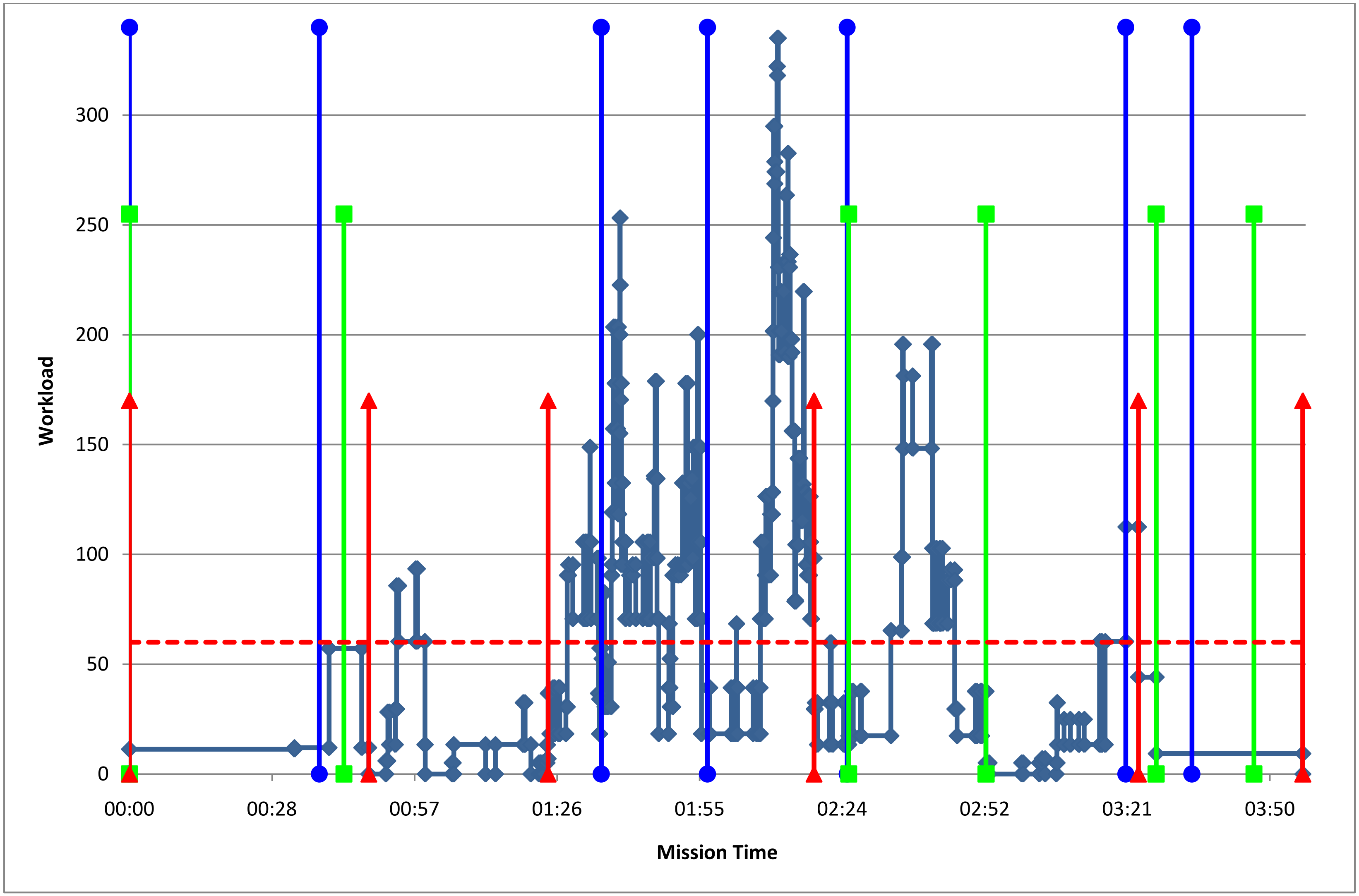


Figure 47. Phase II Run 7 MAC 3 Workload Graph

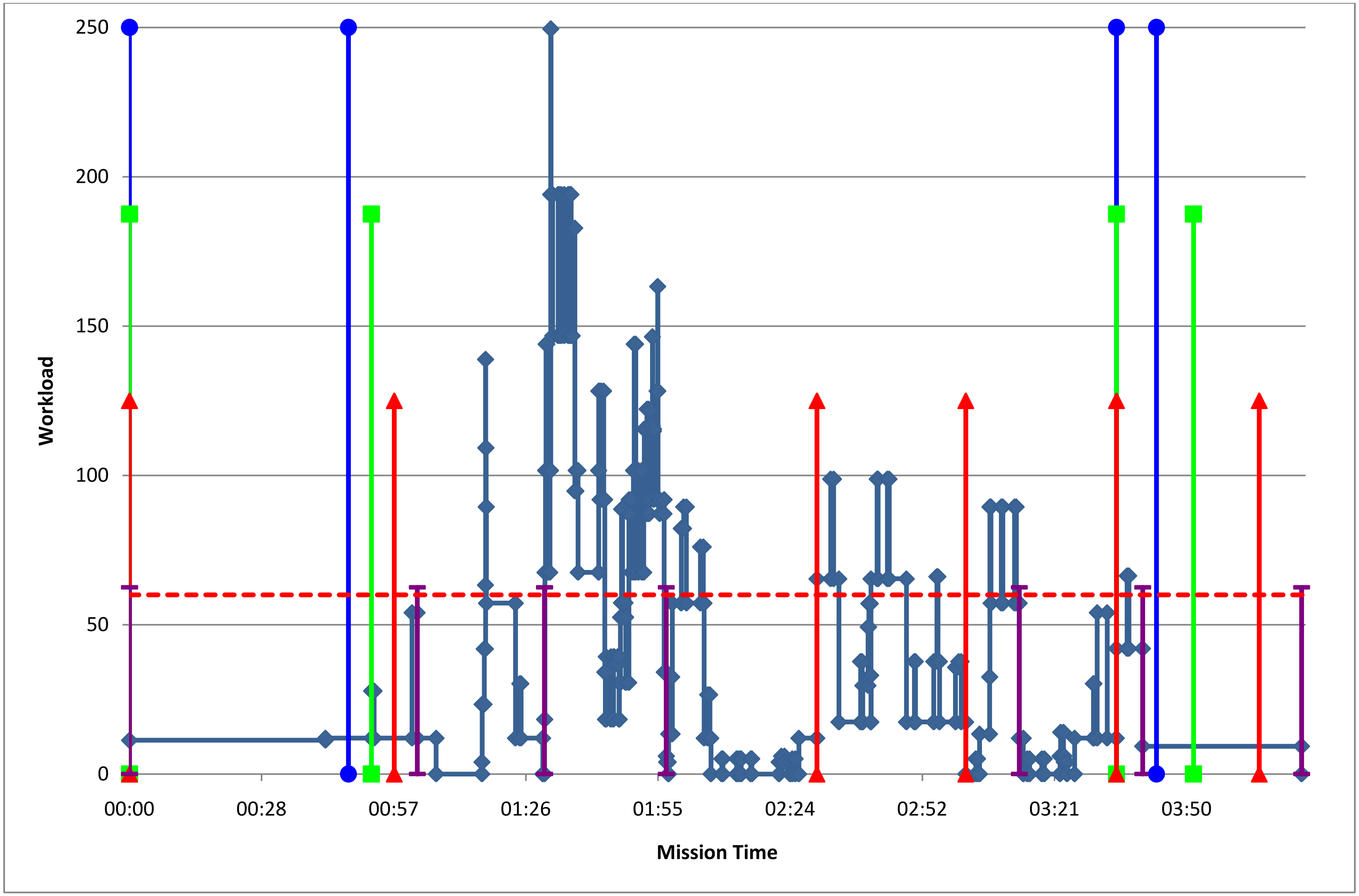


Figure 48. Phase II Run 8 MAC 4 Workload Graph

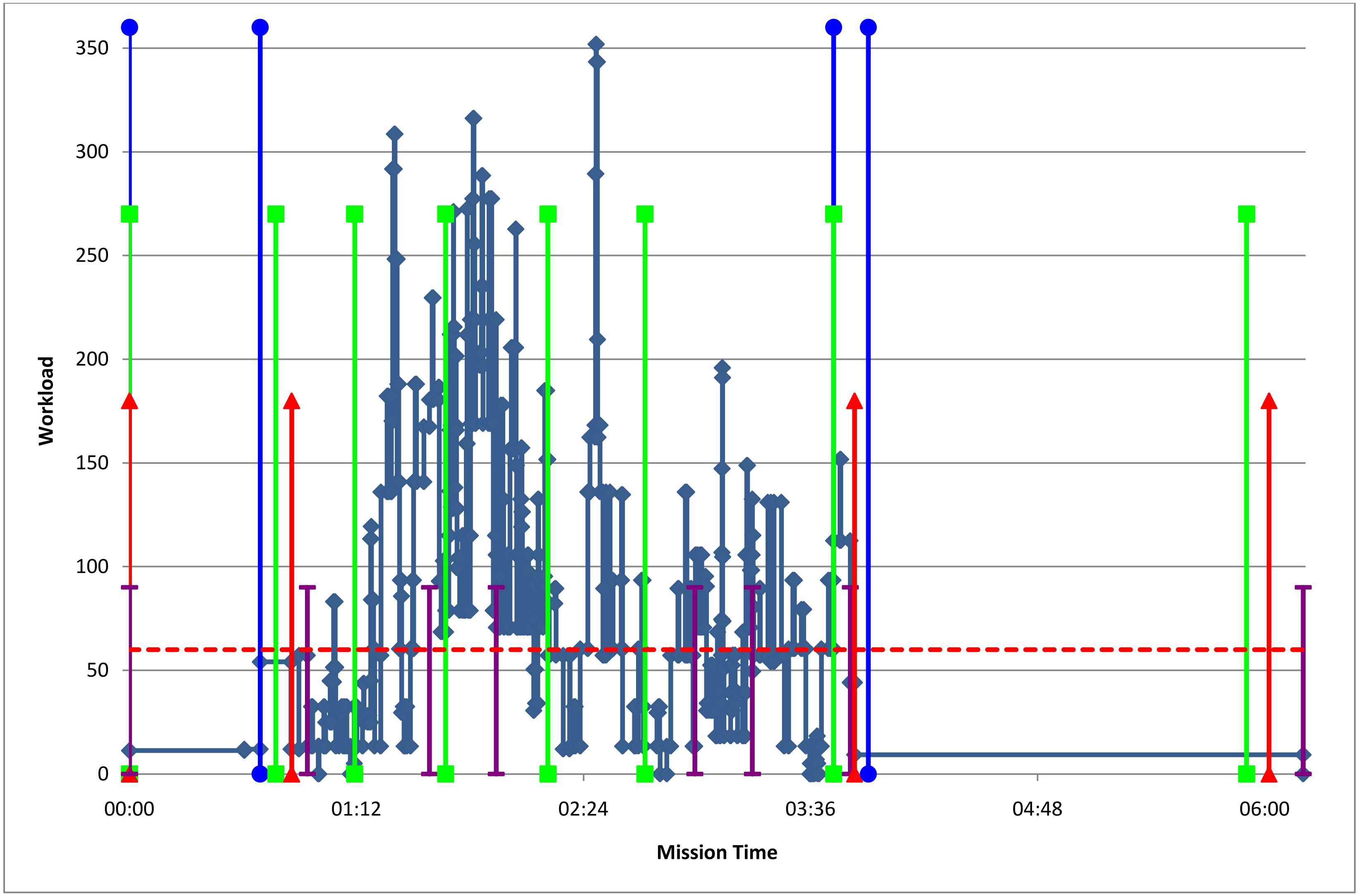


Figure 49. Phase II Run 9 MAC 4 Workload Graph

Appendix D: Sample IMPRINT Operator Workload Detail Output

Table 4. Sample IMPRINT Output: Phase II MAC 2 Transit w/Emergency

IMPRINT Operations Model Report
Operator Workload Detail

Analysis Name: MQ-1 MAC Workload
Analysis Version: 6
RNS: 1
Mission: AC Module
Mission ID: 4
Date: 14-Dec-2010

Clock	Function Name	Task Name	Overall Workload	Single Task Demand	Total Conflict Value	Auditory	Cognitive	Fine Motor	Speech	Visual
00:00:00.00	AC1	Changeover	11.30	11.30	0.00	6.00	5.30	0.00	0.00	0.00
00:00:00.00	Communicate	START	11.30	0.00	0.00	0.00	0.00	0.00	0.00	0.00
00:00:00.10	AC1	Changeover	11.30	11.30	0.00	6.00	5.30	0.00	0.00	0.00
00:09:05.69	AC2	Changeover	11.30	11.30	0.00	6.00	5.30	0.00	0.00	0.00
00:19:46.82	AC1	Fence Check	12.00	12.00	0.00	0.00	6.80	2.20	0.00	3.00
00:24:46.82	AC2	Fence Check	12.00	12.00	0.00	0.00	6.80	2.20	0.00	3.00
00:26:04.02	AC2	Fence Check	54.04	12.00	30.04	0.00	6.80	2.20	0.00	3.00
00:26:04.02	Transit 1	Update Aircraft Course	54.04	12.00	30.04	0.00	6.80	2.20	0.00	3.00
00:29:46.82	Transit 1	Update Aircraft Course	12.00	12.00	0.00	0.00	6.80	2.20	0.00	3.00
00:29:56.45	Communicate	Pilot Reads	30.26	5.10	13.16	0.00	0.00	0.00	0.00	5.10
00:29:56.45	Transit 1	Update Aircraft Course	30.26	12.00	13.16	0.00	6.80	2.20	0.00	3.00
00:30:10.69	Communicate	Pilot Types	27.82	7.00	8.82	0.00	0.00	7.00	0.00	0.00

Clock	Function Name	Task Name	Overall Workload	Single Task Demand	Total Conflict Value	Auditory	Cognitive	Fine Motor	Speech	Visual
00:30:10.69	Transit 1	Update Aircraft Course	27.82	12.00	8.82	0.00	6.80	2.20	0.00	3.00
00:30:46.00	Transit 1	Update Aircraft Course	12.00	12.00	0.00	0.00	6.80	2.20	0.00	3.00
00:38:20.57	Communicate	Pilot Reads	30.26	5.10	13.16	0.00	0.00	0.00	0.00	5.10
00:38:20.57	Transit 1	Update Aircraft Course	30.26	12.00	13.16	0.00	6.80	2.20	0.00	3.00
00:38:24.87	Communicate	Pilot Reads	5.10	5.10	0.00	0.00	0.00	0.00	0.00	5.10
00:52:15.41	Communicate	Pilot Reads	5.10	5.10	0.00	0.00	0.00	0.00	0.00	5.10
00:52:30.54	Communicate	Pilot Reads	5.10	5.10	0.00	0.00	0.00	0.00	0.00	5.10
01:06:08.58	Communicate	Pilot Listens	6.00	6.00	0.00	6.00	0.00	0.00	0.00	0.00
01:12:37.80	Communicate	Pilot Reads	5.10	5.10	0.00	0.00	0.00	0.00	0.00	5.10
01:12:41.27	Communicate	Pilot Reads	18.36	5.10	8.16	0.00	0.00	0.00	0.00	5.10
01:12:41.27	Communicate	Pilot Reads	18.36	5.10	8.16	0.00	0.00	0.00	0.00	5.10
01:14:44.41	Communicate	Pilot Reads	5.10	5.10	0.00	0.00	0.00	0.00	0.00	5.10
01:15:00.05	Communicate	Pilot Reads	5.10	5.10	0.00	0.00	0.00	0.00	0.00	5.10
01:15:16.42	Communicate	Pilot Reads	5.10	5.10	0.00	0.00	0.00	0.00	0.00	5.10
01:15:35.90	Communicate	Pilot Reads	30.26	5.10	13.16	0.00	0.00	0.00	0.00	5.10
01:15:35.90	Transit 1	Update Aircraft Course	30.26	12.00	13.16	0.00	6.80	2.20	0.00	3.00
01:15:40.77	Communicate	Pilot Reads	30.26	5.10	13.16	0.00	0.00	0.00	0.00	5.10
01:15:40.77	Transit 1	Update Aircraft Course	30.26	12.00	13.16	0.00	6.80	2.20	0.00	3.00
01:15:57.79	Transit 1	Update Aircraft Course	12.00	12.00	0.00	0.00	6.80	2.20	0.00	3.00
01:24:42.12	AC1	Emergency	17.40	17.40	0.00	0.00	6.80	5.50	0.00	5.10
01:31:04.26	AC1	Emergency	65.35	17.40	35.95	0.00	6.80	5.50	0.00	5.10
01:31:04.26	Transit 2	Update Aircraft Course	65.35	12.00	35.95	0.00	6.80	2.20	0.00	3.00
01:34:50.53	AC1	Emergency	89.55	17.40	54.15	0.00	6.80	5.50	0.00	5.10
01:34:50.53	Communicate	Pilot Listens	89.55	6.00	54.15	6.00	0.00	0.00	0.00	0.00
01:34:50.53	Transit 2	Update Aircraft Course	89.55	12.00	54.15	0.00	6.80	2.20	0.00	3.00
01:35:07.34	AC1	Emergency	89.55	17.40	54.15	0.00	6.80	5.50	0.00	5.10

Clock	Function Name	Task Name	Overall Workload	Single Task Demand	Total Conflict Value	Auditory	Cognitive	Fine Motor	Speech	Visual
01:35:07.34	Communicate	Pilot Listens	89.55	6.00	54.15	6.00	0.00	0.00	0.00	0.00
01:35:07.34	Transit 2	Update Aircraft Course	89.55	12.00	54.15	0.00	6.80	2.20	0.00	3.00
01:35:35.41	AC1	Emergency	84.90	17.40	51.50	0.00	6.80	5.50	0.00	5.10
01:35:35.41	Communicate	Pilot Talks	84.90	4.00	51.50	0.00	0.00	0.00	4.00	0.00
01:35:35.41	Transit 2	Update Aircraft Course	84.90	12.00	51.50	0.00	6.80	2.20	0.00	3.00
01:36:03.17	AC1	Emergency	65.35	17.40	35.95	0.00	6.80	5.50	0.00	5.10
01:36:03.17	Transit 2	Update Aircraft Course	65.35	12.00	35.95	0.00	6.80	2.20	0.00	3.00
01:36:21.76	AC1	Emergency	17.40	17.40	0.00	0.00	6.80	5.50	0.00	5.10
01:36:44.93	AC1	Emergency	37.67	17.40	15.17	0.00	6.80	5.50	0.00	5.10
01:36:44.93	Communicate	Pilot Reads	37.67	5.10	15.17	0.00	0.00	0.00	0.00	5.10
01:37:04.72	AC1	Emergency	17.40	17.40	0.00	0.00	6.80	5.50	0.00	5.10
01:41:17.40	AC1	Emergency	65.35	17.40	35.95	0.00	6.80	5.50	0.00	5.10
01:41:17.40	Transit 2	Update Aircraft Course	65.35	12.00	35.95	0.00	6.80	2.20	0.00	3.00
01:41:23.97	AC1	Emergency	98.78	17.40	64.28	0.00	6.80	5.50	0.00	5.10
01:41:23.97	Communicate	Pilot Reads	98.78	5.10	64.28	0.00	0.00	0.00	0.00	5.10
01:41:23.97	Transit 2	Update Aircraft Course	98.78	12.00	64.28	0.00	6.80	2.20	0.00	3.00
01:41:49.64	AC1	Emergency	65.35	17.40	35.95	0.00	6.80	5.50	0.00	5.10
01:41:49.64	Transit 2	Update Aircraft Course	65.35	12.00	35.95	0.00	6.80	2.20	0.00	3.00
01:47:13.81	AC1	Emergency	17.40	17.40	0.00	0.00	6.80	5.50	0.00	5.10
01:50:13.75	AC1	Emergency	37.67	17.40	15.17	0.00	6.80	5.50	0.00	5.10
01:50:13.75	Communicate	Pilot Reads	37.67	5.10	15.17	0.00	0.00	0.00	0.00	5.10
01:50:31.77	AC1	Emergency	17.40	17.40	0.00	0.00	6.80	5.50	0.00	5.10
01:53:43.45	AC1	Emergency	37.67	17.40	15.17	0.00	6.80	5.50	0.00	5.10
01:53:43.45	Communicate	Pilot Reads	37.67	5.10	15.17	0.00	0.00	0.00	0.00	5.10
01:53:58.29	AC1	Emergency	17.40	17.40	0.00	0.00	6.80	5.50	0.00	5.10
02:15:40.20	Communicate	Pilot Reads	5.10	5.10	0.00	0.00	0.00	0.00	0.00	5.10

Clock	Function Name	Task Name	Overall Workload	Single Task Demand	Total Conflict Value	Auditory	Cognitive	Fine Motor	Speech	Visual
02:19:33.30	Communicate	Pilot Listens	6.00	6.00	0.00	6.00	0.00	0.00	0.00	0.00
02:19:55.95	Communicate	Pilot Listens	14.00	6.00	4.00	6.00	0.00	0.00	0.00	0.00
02:19:55.95	Communicate	Pilot Talks	14.00	4.00	4.00	0.00	0.00	0.00	4.00	0.00
02:20:02.40	Communicate	Pilot Talks	4.00	4.00	0.00	0.00	0.00	0.00	4.00	0.00
02:20:33.96	Communicate	Pilot Talks	4.00	4.00	0.00	0.00	0.00	0.00	4.00	0.00
02:21:17.00	Communicate	Pilot Listens	6.00	6.00	0.00	6.00	0.00	0.00	0.00	0.00
02:21:40.41	Communicate	Pilot Talks	4.00	4.00	0.00	0.00	0.00	0.00	4.00	0.00
02:21:56.79	Communicate	Pilot Talks	4.00	4.00	0.00	0.00	0.00	0.00	4.00	0.00
02:22:26.19	Communicate	Pilot Listens	6.00	6.00	0.00	6.00	0.00	0.00	0.00	0.00
02:23:07.77	Communicate	Pilot Talks	4.00	4.00	0.00	0.00	0.00	0.00	4.00	0.00
02:23:11.01	Communicate	Pilot Listens	14.00	6.00	4.00	6.00	0.00	0.00	0.00	0.00
02:23:11.01	Communicate	Pilot Talks	14.00	4.00	4.00	0.00	0.00	0.00	4.00	0.00
02:23:32.42	Communicate	Pilot Listens	14.00	6.00	4.00	6.00	0.00	0.00	0.00	0.00
02:23:32.42	Communicate	Pilot Talks	14.00	4.00	4.00	0.00	0.00	0.00	4.00	0.00
02:23:48.83	Communicate	Pilot Talks	16.00	4.00	8.00	0.00	0.00	0.00	4.00	0.00
02:23:48.83	Communicate	Pilot Talks	16.00	4.00	8.00	0.00	0.00	0.00	4.00	0.00
02:24:13.95	Communicate	Pilot Talks	4.00	4.00	0.00	0.00	0.00	0.00	4.00	0.00
02:24:32.78	Communicate	Pilot Talks	4.00	4.00	0.00	0.00	0.00	0.00	4.00	0.00
02:25:07.90	Communicate	Pilot Talks	4.00	4.00	0.00	0.00	0.00	0.00	4.00	0.00
02:25:42.20	Communicate	Pilot Talks	4.00	4.00	0.00	0.00	0.00	0.00	4.00	0.00
02:26:09.08	Communicate	Pilot Listens	6.00	6.00	0.00	6.00	0.00	0.00	0.00	0.00
02:26:39.74	Communicate	Pilot Talks	4.00	4.00	0.00	0.00	0.00	0.00	4.00	0.00
02:27:07.71	Communicate	Pilot Talks	4.00	4.00	0.00	0.00	0.00	0.00	4.00	0.00
02:27:28.05	Communicate	Pilot Talks	16.00	4.00	8.00	0.00	0.00	0.00	4.00	0.00
02:27:28.05	Communicate	Pilot Talks	16.00	4.00	8.00	0.00	0.00	0.00	4.00	0.00
02:27:43.43	Communicate	Pilot Listens	14.00	6.00	8.00	6.00	0.00	0.00	0.00	0.00

Clock	Function Name	Task Name	Overall Workload	Single Task Demand	Total Conflict Value	Auditory	Cognitive	Fine Motor	Speech	Visual
02:27:56.25	Communicate	Pilot Listens	14.00	6.00	4.00	6.00	0.00	0.00	0.00	0.00
02:27:56.25	Communicate	Pilot Talks	14.00	4.00	4.00	0.00	0.00	0.00	4.00	0.00
02:28:06.77	Communicate	Pilot Talks	16.00	4.00	8.00	0.00	0.00	0.00	4.00	0.00
02:28:06.77	Communicate	Pilot Talks	16.00	4.00	8.00	0.00	0.00	0.00	4.00	0.00
02:28:34.60	Transit 1	Update Aircraft Course	23.34	12.00	11.34	0.00	6.80	2.20	0.00	3.00
02:28:35.56	Communicate	Pilot Talks	23.34	4.00	7.34	0.00	0.00	0.00	4.00	0.00
02:28:35.56	Transit 1	Update Aircraft Course	23.34	12.00	7.34	0.00	6.80	2.20	0.00	3.00
02:29:06.79	Communicate	Pilot Listens	26.56	6.00	8.56	6.00	0.00	0.00	0.00	0.00
02:29:06.79	Transit 1	Update Aircraft Course	26.56	12.00	8.56	0.00	6.80	2.20	0.00	3.00
02:29:13.74	Communicate	Pilot Listens	47.04	6.00	23.94	6.00	0.00	0.00	0.00	0.00
02:29:13.74	Communicate	Pilot Reads	47.04	5.10	23.94	0.00	0.00	0.00	0.00	5.10
02:29:13.74	Transit 1	Update Aircraft Course	47.04	12.00	23.94	0.00	6.80	2.20	0.00	3.00
02:29:27.14	Communicate	Pilot Reads	42.51	5.10	21.41	0.00	0.00	0.00	0.00	5.10
02:29:27.14	Communicate	Pilot Talks	42.51	4.00	21.41	0.00	0.00	0.00	4.00	0.00
02:29:27.14	Transit 1	Update Aircraft Course	42.51	12.00	21.41	0.00	6.80	2.20	0.00	3.00
02:29:29.45	Communicate	Pilot Reads	42.51	5.10	21.41	0.00	0.00	0.00	0.00	5.10
02:29:29.45	Communicate	Pilot Talks	42.51	4.00	21.41	0.00	0.00	0.00	4.00	0.00
02:29:29.45	Transit 1	Update Aircraft Course	42.51	12.00	21.41	0.00	6.80	2.20	0.00	3.00
02:29:49.49	Communicate	Pilot Talks	23.34	4.00	7.34	0.00	0.00	0.00	4.00	0.00
02:29:49.49	Transit 1	Update Aircraft Course	23.34	12.00	7.34	0.00	6.80	2.20	0.00	3.00
02:29:56.01	Communicate	Pilot Listens	26.56	6.00	8.56	6.00	0.00	0.00	0.00	0.00
02:29:56.01	Transit 1	Update Aircraft Course	26.56	12.00	8.56	0.00	6.80	2.20	0.00	3.00
02:30:28.31	Communicate	Pilot Talks	23.34	4.00	7.34	0.00	0.00	0.00	4.00	0.00
02:30:28.31	Transit 1	Update Aircraft Course	23.34	12.00	7.34	0.00	6.80	2.20	0.00	3.00
02:31:02.24	Communicate	Pilot Listens	26.56	6.00	8.56	6.00	0.00	0.00	0.00	0.00
02:31:02.24	Transit 1	Update Aircraft Course	26.56	12.00	8.56	0.00	6.80	2.20	0.00	3.00

Clock	Function Name	Task Name	Overall Workload	Single Task Demand	Total Conflict Value	Auditory	Cognitive	Fine Motor	Speech	Visual
02:31:32.02	Communicate	Pilot Talks	23.34	4.00	7.34	0.00	0.00	0.00	4.00	0.00
02:31:32.02	Transit 1	Update Aircraft Course	23.34	12.00	7.34	0.00	6.80	2.20	0.00	3.00
02:32:02.80	Communicate	Pilot Listens	41.90	6.00	19.90	6.00	0.00	0.00	0.00	0.00
02:32:02.80	Communicate	Pilot Talks	41.90	4.00	19.90	0.00	0.00	0.00	4.00	0.00
02:32:02.80	Transit 1	Update Aircraft Course	41.90	12.00	19.90	0.00	6.80	2.20	0.00	3.00
02:32:05.47	Communicate	Pilot Listens	26.56	6.00	8.56	6.00	0.00	0.00	0.00	0.00
02:32:05.47	Transit 1	Update Aircraft Course	26.56	12.00	8.56	0.00	6.80	2.20	0.00	3.00
02:32:25.96	Communicate	Pilot Listens	77.16	6.00	47.16	6.00	0.00	0.00	0.00	0.00
02:32:25.96	Transit 1	Update Aircraft Course	77.16	12.00	47.16	0.00	6.80	2.20	0.00	3.00
02:32:25.96	Transit 2	Update Aircraft Course	77.16	12.00	47.16	0.00	6.80	2.20	0.00	3.00
02:32:31.69	Communicate	Pilot Listens	77.16	6.00	47.16	6.00	0.00	0.00	0.00	0.00
02:32:31.69	Transit 1	Update Aircraft Course	77.16	12.00	47.16	0.00	6.80	2.20	0.00	3.00
02:32:31.69	Transit 2	Update Aircraft Course	77.16	12.00	47.16	0.00	6.80	2.20	0.00	3.00
02:32:51.60	Communicate	Pilot Listens	110.80	6.00	75.70	6.00	0.00	0.00	0.00	0.00
02:32:51.60	Communicate	Pilot Reads	110.80	5.10	75.70	0.00	0.00	0.00	0.00	5.10
02:32:51.60	Transit 1	Update Aircraft Course	110.80	12.00	75.70	0.00	6.80	2.20	0.00	3.00
02:32:51.60	Transit 2	Update Aircraft Course	110.80	12.00	75.70	0.00	6.80	2.20	0.00	3.00
02:32:58.92	Communicate	Pilot Reads	105.05	5.10	71.95	0.00	0.00	0.00	0.00	5.10
02:32:58.92	Communicate	Pilot Talks	105.05	4.00	71.95	0.00	0.00	0.00	4.00	0.00
02:32:58.92	Transit 1	Update Aircraft Course	105.05	12.00	71.95	0.00	6.80	2.20	0.00	3.00
02:32:58.92	Transit 2	Update Aircraft Course	105.05	12.00	71.95	0.00	6.80	2.20	0.00	3.00
02:33:19.06	Communicate	Pilot Talks	72.72	4.00	44.72	0.00	0.00	0.00	4.00	0.00
02:33:19.06	Transit 1	Update Aircraft Course	72.72	12.00	44.72	0.00	6.80	2.20	0.00	3.00
02:33:19.06	Transit 2	Update Aircraft Course	72.72	12.00	44.72	0.00	6.80	2.20	0.00	3.00
02:33:32.26	Communicate	Pilot Listens	77.16	6.00	47.16	6.00	0.00	0.00	0.00	0.00
02:33:32.26	Transit 1	Update Aircraft Course	77.16	12.00	47.16	0.00	6.80	2.20	0.00	3.00

Clock	Function Name	Task Name	Overall Workload	Single Task Demand	Total Conflict Value	Auditory	Cognitive	Fine Motor	Speech	Visual
02:33:32.26	Transit 2	Update Aircraft Course	77.16	12.00	47.16	0.00	6.80	2.20	0.00	3.00
02:33:44.98	Communicate	Pilot Listens	26.56	6.00	8.56	6.00	0.00	0.00	0.00	0.00
02:33:44.98	Transit 2	Update Aircraft Course	26.56	12.00	8.56	0.00	6.80	2.20	0.00	3.00
02:33:53.22	Communicate	Pilot Listens	26.56	6.00	8.56	6.00	0.00	0.00	0.00	0.00
02:33:53.22	Transit 2	Update Aircraft Course	26.56	12.00	8.56	0.00	6.80	2.20	0.00	3.00
02:34:22.55	Communicate	Pilot Listens	47.04	6.00	23.94	6.00	0.00	0.00	0.00	0.00
02:34:22.55	Communicate	Pilot Reads	47.04	5.10	23.94	0.00	0.00	0.00	0.00	5.10
02:34:22.55	Transit 2	Update Aircraft Course	47.04	12.00	23.94	0.00	6.80	2.20	0.00	3.00
02:34:22.98	Communicate	Pilot Listens	47.04	6.00	23.94	6.00	0.00	0.00	0.00	0.00
02:34:22.98	Communicate	Pilot Reads	47.04	5.10	23.94	0.00	0.00	0.00	0.00	5.10
02:34:22.98	Transit 2	Update Aircraft Course	47.04	12.00	23.94	0.00	6.80	2.20	0.00	3.00
02:34:47.76	Communicate	Pilot Listens	26.56	6.00	8.56	6.00	0.00	0.00	0.00	0.00
02:34:47.76	Transit 2	Update Aircraft Course	26.56	12.00	8.56	0.00	6.80	2.20	0.00	3.00
02:34:48.56	Communicate	Pilot Listens	26.56	6.00	8.56	6.00	0.00	0.00	0.00	0.00
02:34:48.56	Transit 2	Update Aircraft Course	26.56	12.00	8.56	0.00	6.80	2.20	0.00	3.00
02:35:22.40	Communicate	Pilot Talks	23.34	4.00	7.34	0.00	0.00	0.00	4.00	0.00
02:35:22.40	Transit 2	Update Aircraft Course	23.34	12.00	7.34	0.00	6.80	2.20	0.00	3.00
02:35:50.34	Communicate	Pilot Talks	23.34	4.00	7.34	0.00	0.00	0.00	4.00	0.00
02:35:50.34	Transit 2	Update Aircraft Course	23.34	12.00	7.34	0.00	6.80	2.20	0.00	3.00
02:36:30.37	Communicate	Pilot Listens	26.56	6.00	8.56	6.00	0.00	0.00	0.00	0.00
02:36:30.37	Transit 2	Update Aircraft Course	26.56	12.00	8.56	0.00	6.80	2.20	0.00	3.00
02:36:54.72	Communicate	Pilot Listens	26.56	6.00	8.56	6.00	0.00	0.00	0.00	0.00
02:36:54.72	Transit 2	Update Aircraft Course	26.56	12.00	8.56	0.00	6.80	2.20	0.00	3.00
02:37:30.74	Communicate	Pilot Listens	26.56	6.00	8.56	6.00	0.00	0.00	0.00	0.00
02:37:30.74	Transit 2	Update Aircraft Course	26.56	12.00	8.56	0.00	6.80	2.20	0.00	3.00

Appendix F: Model Notes

Changeover

Changeover Serial Processing. Changeover is modeled differently from other modules. When a pilot changes over from another pilot in the GCS they spend several minutes talking over the mission and the aircraft. In a MAC configuration this would be similar but each aircraft would be talked through individually and sequentially. To ensure this is modeled properly, the number of gaining changeovers is counted in Initialize Variables along with the losing changeovers which are double counted. These are stored in three variables, CountGC, CountLC, and Changeover_hold. A fourth variable Changeover is a counter used in logic statements to release entities. The release condition for the Changeover task in each AC function is the variable Changeover equal to the tail number of the aircraft, 1-4. The task then increments the value of Changeover. This continues until the value of Changeover equals the value of CountGC whereupon Changeover is reset to 1 so Fence Check can be performed in the same manner. Likewise, Losing Changeover operates serially in ascending tail numbers.

Changeover Hold. Since losing changeovers are performed sequentially and all at once the Changeover_Hold structure was implemented in the two benign modules transit and benign ISR. This task prevents the entity from leaving the module until all other entities are ready to leave. They are then release simultaneously to sequence control and losing changeover where they are performed sequentially. The variable Changeover_hold is set to the total number of losing changeovers in Initialize Variables.

It is then decremented to keep the entity in the module. This replicates the pilot performing mission related tasks until the point when all aircraft changeover. They are not simply forgotten when the mission time expires.

Changeover Restrictions. As a result of the modeling architecture described above there is a restriction on how changeovers can be performed. Since they are performed serially by tail number, if any aircraft is gained through a changeover it should be aircraft 1, then 2, etc. Similarly, if any aircraft is lost through a changeover it should start with tail number 1 and proceed from there. If this is not followed the model will not behave as expected.

Model Run Script

Time_Sequence_Control is a floating point array variable which contains the script for each aircraft necessary to run the model. The three dimensional array starts with the tail number of the aircraft, 1-4. The second two columns of the array designate the sequence of modules to be processed and the time length for them to be processed. Thus Time_Sequence_Control [x,y,z], x is the tail number, y is the module, and z is how long it should last. The y=0 value is the current y index of the script the aircraft is in. For example, the second aircraft in initial handover for 10 minutes would be Time_Sequence_Control [2,0,0]=1 (first row of the script), Time_Sequence_Control [2,1,0]=2 (2 is the designation for gaining handover), and Time_Sequence_Control [2,1,1]=10 (module lasts for 10 min). The third aircraft in a half hour benign ISR after a changeover and transit is Time_Sequence_Control [3,0,0]=3 (third row of the script), Time_Sequence_Control [3,3,0]=3 (3 is the designation for benign ISR), and Time_Sequence_Control [3,3,1]=30 (module lasts for 30 min). This single variable is therefore in control of most of the model and all the information is in one place.

Translators are built into Sequence Control and Initialize Variables macros. This translates the strings of the script into the appropriate numerical values for storage in Time_Sequence_Control and then translates them back to strings for use by the Status variable. These translators are necessary since MicroSaint Sharp run on the C# programming language which does not allow for mixed type arrays. Thus two variables with translators, Status and Time_Sequence_Control, take the place of four.

Run scripts are restricted to beginning with either changeover or handover, having less than eight mission modules which include either a transit or benign ISR before a losing changeover or handover. Time_Sequence_Control[x,y,z], y is only 10 bins in size, 0-9. 0 is reserved for the status, 1 is gaining, 2-6 are any mission module, 7 must be either transit or benign ISR, 8 is losing changeover or handover, and 9 is END. The translator in Initialize Variables takes care of the END scripting for the user.

Appendix G: Model Macro Code

InitializeVariables Macro

```
//Declarations remain unchanged: don't touch
string[,] Aircraft_Module       = new string [5,10];
double[,] Module_Time                   = new double [5,10];
double[] Changeover_Time   = new double [5];
double[] Handover_Time                  = new double [5];
double[] Transit_Time                   = new double [5];
double[] Benign_Time                    = new double [5];
double[] Dynamic_Time                   = new double [5];
double[] Strike_Time            = new double [5];
double[] Emergency_Time                 = new double [5];
double[] Losing_CO_Time                 = new double [5];
double[] Losing_HO_Time                 = new double [5];


/*Input aircraft squences below.
Possible values are commented to the right. Select and copy into quotes.
*/
Aircraft_Used=2;                //Number of Aircraft to Release
Model.PrintOutput("Number of Aircraft Released: " + Aircraft_Used);
//////////////////////////////////Aircraft 1/////////////////////////////
Start_Time[1]=0;                                        //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[1,1] ="Changeover";                     // Changeover or Handover
Only
Module_Time[1,1]     =0;                                        // If 0 then default
will be used
Aircraft_Module[1,2] ="Benign";                         // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,2]     =Distributions.Triangular(20, 15,      35);
                // If 0 then default will be used
Aircraft_Module[1,3] ="Dynamic";                        // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,3]     =Distributions.Triangular(50, 30,      65);
                // If 0 then default will be used
Aircraft_Module[1,4] ="Benign";                         // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,4]     =Distributions.Triangular(60, 30,      65);
                // If 0 then default will be used
```

```
Aircraft_Module[1,5] ="Losing Changeover";                    // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[1,5]     =0;                                  // If 0 then default
will be used
Aircraft_Module[1,6] ="Benign";                      // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,6]     =120;                                // If 0 then default
will be used
Aircraft_Module[1,7] ="Losing Handover";                  // Transit or Benign
Only
Module_Time[1,7]     =10;                                 // If 0 then default
will be used
Aircraft_Module[1,8] ="Losing Changeover";     // Losing Changeover or Losing
Handover Only
Module_Time[1,8]     =0;                                  // If 0 then default
will be used
/////////////////////////////////Aircraft 2/////////////////////////////
Start_Time[2]=0;                                     //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[2,1] ="Changeover";                  // Changeover or Handover
Only
Module_Time[2,1]     =0;                                  // If 0 then default
will be used
Aircraft_Module[2,2] ="Benign";                      // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,2]     =Distributions.Triangular(50, 30,      65);
            // If 0 then default will be used
Aircraft_Module[2,3] ="Transit";                     // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,3]     =Distributions.Triangular(30, 15,      45);
            // If 0 then default will be used
Aircraft_Module[2,4] ="Emergency";                        // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[2,4]     =Distributions.Triangular(20, 15,      25);
            // If 0 then default will be used
Aircraft_Module[2,5] ="Transit";                     // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,5]     =Distributions.Triangular(90, 75,      125);
            // If 0 then default will be used
Aircraft_Module[2,6] ="Losing Changeover";                    // Transit,
Benign, Dynamic, Strike, Emergency
```

```
Module_Time[2,6]     =0;                                          // If 0 then default
will be used
Aircraft_Module[2,7] ="Losing Changeover";                                  // Transit or
Benign Only
Module_Time[2,7]     =10;                                         // If 0 then default
will be used
Aircraft_Module[2,8] ="Losing Changeover";       // Losing Changeover or Losing
Handover Only
Module_Time[2,8]     =0;                                          // If 0 then default
will be used
////////////////////////////////Aircraft 3///////////////////////////
Start_Time[3]=0;                                           //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[3,1] ="Changeover";                        // Changeover or Handover
Only
Module_Time[3,1]     =0;                                          // If 0 then default
will be used
Aircraft_Module[3,2] ="Transit";                           // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,2]     =Distributions.Triangular(130,       120,     160);
                     // If 0 then default will be used
Aircraft_Module[3,3] ="Losing Changeover";                                  // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[3,3]     =0;                                          // If 0 then default
will be used
Aircraft_Module[3,4] ="Benign";                            // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,4]     =120;                                        // If 0 then default
will be used
Aircraft_Module[3,5] ="Benign";                            // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,5]     =120;                                        // If 0 then default
will be used
Aircraft_Module[3,6] ="Emergency";                                // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[3,6]     =120;                                        // If 0 then default
will be used
Aircraft_Module[3,7] ="Benign";                            // Transit or Benign Only
Module_Time[3,7]     =120;                                        // If 0 then default
will be used
Aircraft_Module[3,8] ="Losing Changeover";       // Losing Changeover or Losing
Handover Only
```

```
Module_Time[3,8]     =10;                                        // If 0 then default
will be used
//////////////////////////////////Aircraft 4/////////////////////////////
Start_Time[4]=0;                                        //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[4,1] ="Changeover";                     // Changeover or Handover
Only
Module_Time[4,1]     =0;                                         // If 0 then default
will be used
Aircraft_Module[4,2] ="Transit";                        // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[4,2]     =Distributions.Triangular(130,          120,     160);
                     // If 0 then default will be used
Aircraft_Module[4,3] ="Losing Changeover";                              // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[4,3]     =0;                                         // If 0 then default
will be used
Aircraft_Module[4,4] ="Benign";                         // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[4,4]     =Distributions.Triangular(30, 15,       40);
               // If 0 then default will be used
Aircraft_Module[4,5] ="Losing Changeover";                              // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[4,5]     =0;                                         // If 0 then default
will be used
Aircraft_Module[4,6] ="Benign";                         // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[4,6]     =120;                                       // If 0 then default
will be used
Aircraft_Module[4,7] ="Benign";                         // Transit or Benign Only
Module_Time[4,7]     =120;                                       // If 0 then default
will be used
Aircraft_Module[4,8] ="Losing Changeover";         // Losing Changeover or Losing
Handover Only
Module_Time[4,8]     =10;                                        // If 0 then default
will be used


Model.PrintOutput("Sequences Read");
/*Input Distributions for module times (in minutes) after the "Distributions." below.
These are calculated individually for each aircraft according to the distributions below.
These distributions will only be used if the task lengths above are set to 0.
```

```
The For loop is necessary for stochastic integrity, do not mess with it.*/
for ( int i = 1; i <= 4; i++ ){
Changeover_Time[i]=Distributions.Triangular(X,X,X);
Handover_Time[i]=   Distributions.Triangular(X,X,X);
Transit_Time[i]=    Distributions.Triangular(X,X,X);
Benign_Time[i]=     Distributions.Triangular(X,X,X);
Dynamic_Time[i]=    Distributions.Triangular(X,X,X);
Strike_Time[i]=     Distributions.Triangular(X,X,X);
Emergency_Time[i]=  Distributions.Triangular(X,X,X);
Losing_CO_Time[i]=  Distributions.Triangular(X,X,X);
Losing_HO_Time[i]=  Distributions.Triangular(X,X,X);
}

Model.PrintOutput("Module Times Calculated");
/*Input null, Mean, and Null for communication Exponential distributions in minutes*/
double[] Changeover_Comm={0,0,0};          //Changeover
double[] Handover_Comm   ={0,4,0};         //Handover
double[] Transit_Comm    ={0,14,0};        //Transit
double[] Benign_Comm     ={0,10,0};        //Benign ISR
double[] Dynamic_Comm    ={0,1,0};         //Dynamic ISR
double[] Strike_Comm     ={0,3,0};         //Strike
double[] Emergency_Comm  ={0,9,0};         //Emergency
double[] Losing_CO_Comm={0,0,0};           //Losing Changeover
double[] Losing_HO_Comm  ={0,4,0};         //Losing Handover
Model.PrintOutput("Comm Times Set");
//////////////////////////////////Translation Code (don't touch)/////////////////////
//convert minutes to seconds for clock operators
for ( int i = 1; i <= 4; i++ ){
        Start_Time[i] = 60*Start_Time[i];
}
//Dump Comm data into Global Variable
for ( int i = 0; i <= 2; i++ ){
        Comm_Time[0,i]=Changeover_Comm[i];        //Changeover
        Comm_Time[1,i]=Handover_Comm[i];          //Handover
        Comm_Time[2,i]=Transit_Comm[i];           //Transit
        Comm_Time[3,i]=Benign_Comm[i];            //Benign
        Comm_Time[4,i]=Dynamic_Comm[i];                   //Dynamic
        Comm_Time[5,i]=Strike_Comm[i];            //Strike
        Comm_Time[6,i]=Emergency_Comm[i];         //Emergency
        Comm_Time[7,i]=Losing_CO_Comm[i];         //Losing Changeover
        Comm_Time[8,i]=Losing_HO_Comm[i];         //Losing Handover
}
```

```
/* In Time_Sequence_Control[x,y,z](5,10,2) x is the aircraft designator (1-4),
y is the row, only use rows 1-9. Row 0 holds step information.
z=0 is the module to be executed, z=1 is the time to execute it in minutes.

Time_Sequence_Control[x,y,0]=1;             Changeover
Time_Sequence_Control[x,y,0]=2;             Handover
Time_Sequence_Control[x,y,0]=3;             Transit
Time_Sequence_Control[x,y,0]=4;             Benign ISR
Time_Sequence_Control[x,y,0]=5;             Dynamic ISR
Time_Sequence_Control[x,y,0]=6;             Strike
Time_Sequence_Control[x,y,0]=7;             Emergency
Time_Sequence_Control[x,y,0]=8;             Losing Changeover
Time_Sequence_Control[x,y,0]=9;             Losing Handover
Time_Sequence_Control[x,y,0]=999;           End

*/
bool TranslateComplete;
int count;
//////////////Aircraft code
//Loop through each aircraft to assign sequences 1 to end
for ( int TailNumber = 1; TailNumber <= Aircraft_Used; TailNumber++ ){
	Time_Sequence_Control[TailNumber,0,0]=0;

	count=1;
	TranslateComplete=false;
	while (!TranslateComplete){
		if(TailNumber<=Aircraft_Used){
			Snapshot_Status=Aircraft_Module[TailNumber,count];
			Snapshot_TailNumber=TailNumber;
			Model.TriggerSnapshot("RunInfo");
		}
		switch (Aircraft_Module[TailNumber,count])
		{
			case "Changeover":
			{
				Time_Sequence_Control[TailNumber,count,0]=1;
				if(Module_Time[TailNumber,count]==0.0){

	Time_Sequence_Control[TailNumber,count,1]=Changeover_Time[TailNumber];
				}
```

```
                    else{

     Time_Sequence_Control[TailNumber,count,1]=Module_Time[TailNumber,count
];
                    }
//Count number of gaining changeovers
                    CountGC++;
                    break;
               }
               case "Handover":
               {
                    Time_Sequence_Control[TailNumber,count,0]=2;
                    if(Module_Time[TailNumber,count]==0.0){

     Time_Sequence_Control[TailNumber,count,1]=Handover_Time[TailNumber];
                    }
                    else{

     Time_Sequence_Control[TailNumber,count,1]=Module_Time[TailNumber,count
];
                    }
                    break;
               }
               case "Transit":
               {
                    Time_Sequence_Control[TailNumber,count,0]=3;
                    if(Module_Time[TailNumber,count]==0.0){

     Time_Sequence_Control[TailNumber,count,1]=Transit_Time[TailNumber];
                    }
                    else{

     Time_Sequence_Control[TailNumber,count,1]=Module_Time[TailNumber,count
];
                    }
                    break;
               }
               case "Benign":
               {
                    Time_Sequence_Control[TailNumber,count,0]=4;
                    if(Module_Time[TailNumber,count]==0.0){
```

```
        Time_Sequence_Control[TailNumber,count,1]=Benign_Time[TailNumber];
                            }
                            else{

        Time_Sequence_Control[TailNumber,count,1]=Module_Time[TailNumber,count
];
                            }
                            break;
                        }
                        case "Dynamic":
                        {
                            Time_Sequence_Control[TailNumber,count,0]=5;
                            if(Module_Time[TailNumber,count]==0.0){

        Time_Sequence_Control[TailNumber,count,1]=Dynamic_Time[TailNumber];
                            }
                            else{

        Time_Sequence_Control[TailNumber,count,1]=Module_Time[TailNumber,count
];
                            }
                            break;
                        }
                        case "Strike":
                        {
                            Time_Sequence_Control[TailNumber,count,0]=6;
                            if(Module_Time[TailNumber,count]==0.0){

        Time_Sequence_Control[TailNumber,count,1]=Strike_Time[TailNumber];
                            }
                            else{

        Time_Sequence_Control[TailNumber,count,1]=Module_Time[TailNumber,count
];
                            }
                            break;
                        }
                        case "Emergency":
                        {
                            Time_Sequence_Control[TailNumber,count,0]=7;
                            if(Module_Time[TailNumber,count]==0.0){
```

```
		Time_Sequence_Control[TailNumber,count,1]=Emergency_Time[TailNumber];
				}
				else{

		Time_Sequence_Control[TailNumber,count,1]=Module_Time[TailNumber,count
];
				}
				break;
			}
			case "Losing Changeover":
			{
				Time_Sequence_Control[TailNumber,count,0]=8;
				if(Module_Time[TailNumber,count]==0.0){

		Time_Sequence_Control[TailNumber,count,1]=Losing_CO_Time[TailNumber];
				}
				else{

		Time_Sequence_Control[TailNumber,count,1]=Module_Time[TailNumber,count
];
				}
//Count number of losing changeovers for both changeover_hold and serial processing
				Changeover_Count++;
				CountLC++;
				break;
			}
			case "Losing Handover":
			{
				Time_Sequence_Control[TailNumber,count,0]=9;
				if(Module_Time[TailNumber,count]==0.0){

		Time_Sequence_Control[TailNumber,count,1]=Losing_HO_Time[TailNumber];
				}
				else{

		Time_Sequence_Control[TailNumber,count,1]=Module_Time[TailNumber,count
];
				}
				break;
			}
		}
```

```
			//Check for mission end and increment count
			if(Aircraft_Module[TailNumber,count]=="Losing
Changeover"||Aircraft_Module[TailNumber,count]=="Losing Handover"){
				TranslateComplete=true;
			}
			count++;
		}
//Set last sequence to exit the model
		Time_Sequence_Control[TailNumber,count,0]=999;
		Time_Sequence_Control[TailNumber,count,1]=0;
}


Model.PrintOutput("Changeover_Count=" + Changeover_Count);
Model.PrintOutput("Translation Complete");
Model.PrintOutput("Variables Initialized");
```

Sequence Control Macro

```
/*
Reads Time_Sequence_Control and translates it into the Status for the tactical path
decision of Sequence Control Task. It then aborts and restarts the comm spinners to
ensure the
proper parameters for comm frequency distributions.
*/
int Seq;
string CommSpinner="";
Time_Sequence_Control[TailNumber,0,0]++;
//Read sequence value, convert to int
Seq=System.Convert.ToInt32(Time_Sequence_Control[TailNumber,System.Convert.ToI
nt32(Time_Sequence_Control[TailNumber,0,0]),0]);
//Set Status to proper String value
switch (Seq)
{
	case 1:
	{
		Status[TailNumber]="Changeover";
		break;
	}
	case 2:
	{
		Status[TailNumber]="Handover";
		break;
	}
	case 3:
	{
		Status[TailNumber]="Transit";
		break;
	}
	case 4:
	{
		Status[TailNumber]="Benign";
		break;
	}
	case 5:
	{
		Status[TailNumber]="Dynamic";
		break;
```

```
		}
		case 6:
		{
			Status[TailNumber]="Strike";
			break;
		}
		case 7:
		{
			Status[TailNumber]="Emergency";
			break;
		}
		case 8:
		{
			Status[TailNumber]="Losing Changeover";
			break;
		}
		case 9:
		{
			Status[TailNumber]="Losing Handover";
			break;
		}
		case 999:
		{
			Status[TailNumber]="END";
			break;
		}
}
//Snapshot records time when change occured for charting in post processing
Snapshot_Status= "Aircraft " + TailNumber + " is in " + Status[TailNumber];
Snapshot_TailNumber=0;
Snapshot_Time=Clock/(3600*24);
Model.TriggerSnapshot("RunInfo");
Model.PrintOutput(Snapshot_Status + " at " + Clock/3600);
Snapshot_Status= "";
// It triggers twice so the spreadsheet data can be easily charted in Excel
Model.TriggerSnapshot("RunInfo");
//Abort/Start Comm spinners
switch (TailNumber)
{
		case 1:
		{
			CommSpinner="8_8";
```

```
			break;
		}
		case 2:
		{
			CommSpinner="8_2";
			break;
		}
		case 3:
		{
			CommSpinner="8_3";
			break;
		}
		case 4:
		{
			CommSpinner="8_4";
			break;
		}
	}
	Model.Abort("ID",CommSpinner);
	Model.Start(CommSpinner);
```

Phase I Run Code

```
------------------Run 1---------------------------------------------------------------------------------
----------
Aircraft_Used=1;                    //Number of Aircraft to Release
Model.PrintOutput("Number of Aircraft Released: " + Aircraft_Used);
//////////////////////////////////Aircraft 1//////////////////////////////
Start_Time[1]=0;                                              //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[1,1] ="Handover";                   // Changeover or Handover Only
Module_Time[1,1]     =10;                                             // If 0 then default
will be used
Aircraft_Module[1,2] ="Transit";                            // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,2]     =120;                                            // If 0 then default
will be used
Aircraft_Module[1,3] ="Dynamic";                          // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,3]     =120;                                            // If 0 then default
will be used
Aircraft_Module[1,4] ="Strike";                              // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,4]     =0;                                               // If 0 then default
will be used
Aircraft_Module[1,5] ="Emergency";                                 // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[1,5]     =120;                                            // If 0 then default
will be used
Aircraft_Module[1,6] ="Benign";                              // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,6]     =120;                                            // If 0 then default
will be used
Aircraft_Module[1,7] ="Losing Changeover";                              // Transit or
Benign Only
Module_Time[1,7]     =10;                                             // If 0 then default
will be used
Aircraft_Module[1,8] ="Losing Changeover";         // Losing Changeover or Losing
Handover Only
Module_Time[1,8]     =0;                                               // If 0 then default
will be used
```

```
------------------Run 2---------------------------------------------------------------------------------
----------
Aircraft_Used=2;                    //Number of Aircraft to Release
Model.PrintOutput("Number of Aircraft Released: " + Aircraft_Used);
//////////////////////////////////Aircraft 1/////////////////////////////
Start_Time[1]=0;                                            //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[1,1] ="Changeover";                         // Changeover or Handover
Only
Module_Time[1,1]     =10;                                          // If 0 then default
will be used
Aircraft_Module[1,2] ="Transit";                            // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,2]     =120;                                         // If 0 then default
will be used
Aircraft_Module[1,3] ="Emergency";                                 // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[1,3]     =120;                                         // If 0 then default
will be used
Aircraft_Module[1,4] ="Benign";                             // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,4]     =120;                                         // If 0 then default
will be used
Aircraft_Module[1,5] ="Dynamic";                            // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,5]     =120;                                         // If 0 then default
will be used
Aircraft_Module[1,6] ="Benign";                             // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,6]     =120;                                         // If 0 then default
will be used
Aircraft_Module[1,7] ="Losing Changeover";                                // Transit or
Benign Only
Module_Time[1,7]     =10;                                          // If 0 then default
will be used
Aircraft_Module[1,8] ="Losing Changeover";         // Losing Changeover or Losing
Handover Only
Module_Time[1,8]     =0;                                           // If 0 then default
will be used
//////////////////////////////////Aircraft 2/////////////////////////////
Start_Time[2]=0;                                            //Use in Aircraft Control to
release entities into Aircraft functions
```

```
Aircraft_Module[2,1] ="Changeover";                    // Changeover or Handover
Only
Module_Time[2,1]     =10;                                     // If 0 then default
will be used
Aircraft_Module[2,2] ="Transit";                       // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,2]     =120;                                    // If 0 then default
will be used
Aircraft_Module[2,3] ="Transit";                       // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,3]     =120;                                    // If 0 then default
will be used
Aircraft_Module[2,4] ="Dynamic";                       // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,4]     =120;                                    // If 0 then default
will be used
Aircraft_Module[2,5] ="Emergency";                            // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[2,5]     =120;                                    // If 0 then default
will be used
Aircraft_Module[2,6] ="Benign";                        // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,6]     =120;                                    // If 0 then default
will be used
Aircraft_Module[2,7] ="Losing Changeover";                           // Transit or
Benign Only
Module_Time[2,7]     =10;                                     // If 0 then default
will be used
Aircraft_Module[2,8] ="Losing Changeover";     // Losing Changeover or Losing
Handover Only
Module_Time[2,8]     =0;                                      // If 0 then default
will be used
------------------Run 3---------------------------------------------------------------------------------
----------
Aircraft_Used=2;            //Number of Aircraft to Release
Model.PrintOutput("Number of Aircraft Released: " + Aircraft_Used);
//////////////////////////////////Aircraft 1//////////////////////////////
Start_Time[1]=0;                                       //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[1,1] ="Handover";              // Changeover or Handover Only
Module_Time[1,1]     =10;                                     // If 0 then default
will be used
```

```
Aircraft_Module[1,2] ="Emergency";                         // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[1,2]     =120;                                 // If 0 then default
will be used
Aircraft_Module[1,3] ="Dynamic";                      // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,3]     =120;                                 // If 0 then default
will be used
Aircraft_Module[1,4] ="Emergency";                         // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[1,4]     =120;                                 // If 0 then default
will be used
Aircraft_Module[1,5] ="Dynamic";                      // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,5]     =120;                                 // If 0 then default
will be used
Aircraft_Module[1,6] ="Transit";                      // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,6]     =120;                                 // If 0 then default
will be used
Aircraft_Module[1,7] ="Losing Handover";                   // Transit or Benign
Only
Module_Time[1,7]     =10;                                  // If 0 then default
will be used
Aircraft_Module[1,8] ="Losing Changeover";      // Losing Changeover or Losing
Handover Only
Module_Time[1,8]     =0;                                   // If 0 then default
will be used
//////////////////////////////////Aircraft 2/////////////////////////////
Start_Time[2]=0;                                      //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[2,1] ="Handover";               // Changeover or Handover Only
Module_Time[2,1]     =10;                                  // If 0 then default
will be used
Aircraft_Module[2,2] ="Benign";                       // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,2]     =120;                                 // If 0 then default
will be used
Aircraft_Module[2,3] ="Dynamic";                      // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,3]     =120;                                 // If 0 then default
will be used
```

```
Aircraft_Module[2,4] ="Emergency";                    // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[2,4]    =120;                              // If 0 then default
will be used
Aircraft_Module[2,5] ="Transit";                  // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,5]    =120;                              // If 0 then default
will be used
Aircraft_Module[2,6] ="Benign";                   // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,6]    =120;                              // If 0 then default
will be used
Aircraft_Module[2,7] ="Losing Handover";               // Transit or Benign
Only
Module_Time[2,7]    =10;                               // If 0 then default
will be used
Aircraft_Module[2,8] ="Losing Changeover";    // Losing Changeover or Losing
Handover Only
Module_Time[2,8]    =0;                                // If 0 then default
will be used
------------------Run 4--------------------------------------------------------------------------------
----------
Aircraft_Used=3;           //Number of Aircraft to Release
Model.PrintOutput("Number of Aircraft Released: " + Aircraft_Used);
//////////////////////////////////Aircraft 1/////////////////////////////
Start_Time[1]=0;                                  //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[1,1] ="Changeover";               // Changeover or Handover
Only
Module_Time[1,1]    =10;                               // If 0 then default
will be used
Aircraft_Module[1,2] ="Benign";                   // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,2]    =120;                              // If 0 then default
will be used
Aircraft_Module[1,3] ="Transit";                  // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,3]    =120;                              // If 0 then default
will be used
Aircraft_Module[1,4] ="Dynamic";                  // Transit, Benign, Dynamic,
Strike, Emergency
```

```
Module_Time[1,4]     =120;                                          // If 0 then default
will be used
Aircraft_Module[1,5] ="Emergency";                                  // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[1,5]     =120;                                          // If 0 then default
will be used
Aircraft_Module[1,6] ="Transit";                          // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,6]     =120;                                          // If 0 then default
will be used
Aircraft_Module[1,7] ="Benign";                           // Transit or Benign Only
Module_Time[1,7]     =120;                                          // If 0 then default
will be used
Aircraft_Module[1,8] ="Losing Changeover";          // Losing Changeover or Losing
Handover Only
Module_Time[1,8]     =10;                                           // If 0 then default
will be used
//////////////////////////////////Aircraft 2/////////////////////////////
Start_Time[2]=0;                                          //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[2,1] ="Changeover";                       // Changeover or Handover
Only
Module_Time[2,1]     =10;                                           // If 0 then default
will be used
Aircraft_Module[2,2] ="Benign";                           // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,2]     =120;                                          // If 0 then default
will be used
Aircraft_Module[2,3] ="Dynamic";                          // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,3]     =120;                                          // If 0 then default
will be used
Aircraft_Module[2,4] ="Dynamic";                          // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,4]     =120;                                          // If 0 then default
will be used
Aircraft_Module[2,5] ="Emergency";                                  // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[2,5]     =120;                                          // If 0 then default
will be used
Aircraft_Module[2,6] ="Benign";                           // Transit, Benign, Dynamic,
Strike, Emergency
```

Module_Time[2,6] =120; // If 0 then default will be used
Aircraft_Module[2,7] ="Benign"; // Transit or Benign Only
Module_Time[2,7] =120; // If 0 then default will be used
Aircraft_Module[2,8] ="Losing Changeover"; // Losing Changeover or Losing Handover Only
Module_Time[2,8] =10; // If 0 then default will be used
///////////////////////////////////Aircraft 3/////////////////////////////
Start_Time[3]=0; //Use in Aircraft Control to release entities into Aircraft functions
Aircraft_Module[3,1] ="Changeover"; // Changeover or Handover Only
Module_Time[3,1] =10; // If 0 then default will be used
Aircraft_Module[3,2] ="Emergency"; // Transit, Benign, Dynamic, Strike, Emergency
Module_Time[3,2] =120; // If 0 then default will be used
Aircraft_Module[3,3] ="Emergency"; // Transit, Benign, Dynamic, Strike, Emergency
Module_Time[3,3] =120; // If 0 then default will be used
Aircraft_Module[3,4] ="Emergency"; // Transit, Benign, Dynamic, Strike, Emergency
Module_Time[3,4] =120; // If 0 then default will be used
Aircraft_Module[3,5] ="Benign"; // Transit, Benign, Dynamic, Strike, Emergency
Module_Time[3,5] =120; // If 0 then default will be used
Aircraft_Module[3,6] ="Dynamic"; // Transit, Benign, Dynamic, Strike, Emergency
Module_Time[3,6] =120; // If 0 then default will be used
Aircraft_Module[3,7] ="Benign"; // Transit or Benign Only
Module_Time[3,7] =120; // If 0 then default will be used
Aircraft_Module[3,8] ="Losing Changeover"; // Losing Changeover or Losing Handover Only

```
Module_Time[3,8]     =10;                                                  // If 0 then default
will be used
------------------Run 5---------------------------------------------------------------------------------
----------
Aircraft_Used=3;                  //Number of Aircraft to Release
Model.PrintOutput("Number of Aircraft Released: " + Aircraft_Used);
//////////////////////////////////Aircraft 1//////////////////////////////
Start_Time[1]=0;                                                  //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[1,1] ="Handover";                       // Changeover or Handover Only
Module_Time[1,1]     =10;                                                  // If 0 then default
will be used
Aircraft_Module[1,2] ="Benign";                                   // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,2]     =120;                                                 // If 0 then default
will be used
Aircraft_Module[1,3] ="Transit";                                  // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,3]     =120;                                                 // If 0 then default
will be used
Aircraft_Module[1,4] ="Dynamic";                                  // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,4]     =120;                                                 // If 0 then default
will be used
Aircraft_Module[1,5] ="Emergency";                                         // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[1,5]     =120;                                                 // If 0 then default
will be used
Aircraft_Module[1,6] ="Transit";                                  // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,6]     =120;                                                 // If 0 then default
will be used
Aircraft_Module[1,7] ="Benign";                                   // Transit or Benign Only
Module_Time[1,7]     =120;                                                 // If 0 then default
will be used
Aircraft_Module[1,8] ="Losing Handover";  // Losing Changeover or Losing Handover
Only
Module_Time[1,8]     =10;                                                  // If 0 then default
will be used
//////////////////////////////////Aircraft 2//////////////////////////////
Start_Time[2]=0;                                                  //Use in Aircraft Control to
release entities into Aircraft functions
```

```
Aircraft_Module[2,1] ="Handover";                    // Changeover or Handover Only
Module_Time[2,1]     =10;                                                   // If 0 then default
will be used
Aircraft_Module[2,2] ="Benign";                               // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,2]     =120;                                                  // If 0 then default
will be used
Aircraft_Module[2,3] ="Transit";                              // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,3]     =120;                                                  // If 0 then default
will be used
Aircraft_Module[2,4] ="Dynamic";                              // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,4]     =120;                                                  // If 0 then default
will be used
Aircraft_Module[2,5] ="Emergency";                                    // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[2,5]     =120;                                                  // If 0 then default
will be used
Aircraft_Module[2,6] ="Benign";                               // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,6]     =120;                                                  // If 0 then default
will be used
Aircraft_Module[2,7] ="Benign";                               // Transit or Benign Only
Module_Time[2,7]     =120;                                                  // If 0 then default
will be used
Aircraft_Module[2,8] ="Losing Handover"; // Losing Changeover or Losing Handover
Only
Module_Time[2,8]     =10;                                                   // If 0 then default
will be used
/////////////////////////////////Aircraft 3/////////////////////////////
Start_Time[3]=0;                                              //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[3,1] ="Handover";                    // Changeover or Handover Only
Module_Time[3,1]     =10;                                                   // If 0 then default
will be used
Aircraft_Module[3,2] ="Dynamic";                              // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,2]     =120;                                                  // If 0 then default
will be used
Aircraft_Module[3,3] ="Transit";                              // Transit, Benign, Dynamic,
Strike, Emergency
```

```
Module_Time[3,3]     =120;                                          // If 0 then default
will be used
Aircraft_Module[3,4] ="Benign";                          // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,4]     =120;                                          // If 0 then default
will be used
Aircraft_Module[3,5] ="Dynamic";                         // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,5]     =120;                                          // If 0 then default
will be used
Aircraft_Module[3,6] ="Emergency";                                  // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[3,6]     =120;                                          // If 0 then default
will be used
Aircraft_Module[3,7] ="Transit";                         // Transit or Benign Only
Module_Time[3,7]     =120;                                          // If 0 then default
will be used
Aircraft_Module[3,8] ="Losing Handover"; // Losing Changeover or Losing Handover
Only
Module_Time[3,8]     =10;                                           // If 0 then default
will be used
------------------Run 6--------------------------------------------------------------------------------
----------
Aircraft_Used=3;                //Number of Aircraft to Release
Model.PrintOutput("Number of Aircraft Released: " + Aircraft_Used);
//////////////////////////////////Aircraft 1//////////////////////////////
Start_Time[1]=0;                                         //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[1,1] ="Changeover";                      // Changeover or Handover
Only
Module_Time[1,1]     =10;                                           // If 0 then default
will be used
Aircraft_Module[1,2] ="Transit";                         // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,2]     =120;                                          // If 0 then default
will be used
Aircraft_Module[1,3] ="Transit";                         // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,3]     =120;                                          // If 0 then default
will be used
Aircraft_Module[1,4] ="Dynamic";                         // Transit, Benign, Dynamic,
Strike, Emergency
```

Module_Time[1,4]   =120;   // If 0 then default
will be used
Aircraft_Module[1,5] ="Emergency";   // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[1,5]   =120;   // If 0 then default
will be used
Aircraft_Module[1,6] ="Benign";   // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,6]   =120;   // If 0 then default
will be used
Aircraft_Module[1,7] ="Transit";   // Transit or Benign Only
Module_Time[1,7]   =120;   // If 0 then default
will be used
Aircraft_Module[1,8] ="Losing Changeover";   // Losing Changeover or Losing
Handover Only
Module_Time[1,8]   =10;   // If 0 then default
will be used
//////////////////////////////////Aircraft 2/////////////////////////////
Start_Time[2]=0;   //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[2,1] ="Handover";   // Changeover or Handover Only
Module_Time[2,1]   =10;   // If 0 then default
will be used
Aircraft_Module[2,2] ="Transit";   // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,2]   =120;   // If 0 then default
will be used
Aircraft_Module[2,3] ="Transit";   // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,3]   =120;   // If 0 then default
will be used
Aircraft_Module[2,4] ="Dynamic";   // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,4]   =120;   // If 0 then default
will be used
Aircraft_Module[2,5] ="Emergency";   // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[2,5]   =120;   // If 0 then default
will be used
Aircraft_Module[2,6] ="Dynamic";   // Transit, Benign, Dynamic,
Strike, Emergency

Module_Time[2,6]   =120;   // If 0 then default will be used
Aircraft_Module[2,7] ="Transit";   // Transit or Benign Only
Module_Time[2,7]   =120;   // If 0 then default will be used
Aircraft_Module[2,8] ="Losing Changeover";   // Losing Changeover or Losing Handover Only
Module_Time[2,8]   =10;   // If 0 then default will be used
//////////////////////////////////Aircraft 3/////////////////////////////
Start_Time[3]=0;   //Use in Aircraft Control to release entities into Aircraft functions
Aircraft_Module[3,1] ="Handover";   // Changeover or Handover Only
Module_Time[3,1]   =10;   // If 0 then default will be used
Aircraft_Module[3,2] ="Dynamic";   // Transit, Benign, Dynamic, Strike, Emergency
Module_Time[3,2]   =120;   // If 0 then default will be used
Aircraft_Module[3,3] ="Emergency";   // Transit, Benign, Dynamic, Strike, Emergency
Module_Time[3,3]   =120;   // If 0 then default will be used
Aircraft_Module[3,4] ="Transit";   // Transit, Benign, Dynamic, Strike, Emergency
Module_Time[3,4]   =120;   // If 0 then default will be used
Aircraft_Module[3,5] ="Transit";   // Transit, Benign, Dynamic, Strike, Emergency
Module_Time[3,5]   =120;   // If 0 then default will be used
Aircraft_Module[3,6] ="Emergency";   // Transit, Benign, Dynamic, Strike, Emergency
Module_Time[3,6]   =120;   // If 0 then default will be used
Aircraft_Module[3,7] ="Benign";   // Transit or Benign Only
Module_Time[3,7]   =120;   // If 0 then default will be used
Aircraft_Module[3,8] ="Losing Handover"; // Losing Changeover or Losing Handover Only
Module_Time[3,8]   =10;   // If 0 then default will be used

```
-----------------Run 7---------------------------------------------------------------------------------
----------
Aircraft_Used=4;                    //Number of Aircraft to Release
Model.PrintOutput("Number of Aircraft Released: " + Aircraft_Used);
/////////////////////////////////Aircraft 1/////////////////////////////
Start_Time[1]=0;                                            //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[1,1] ="Changeover";                         // Changeover or Handover
Only
Module_Time[1,1]     =10;                                           // If 0 then default
will be used
Aircraft_Module[1,2] ="Transit";                            // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,2]     =120;                                          // If 0 then default
will be used
Aircraft_Module[1,3] ="Transit";                            // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,3]     =120;                                          // If 0 then default
will be used
Aircraft_Module[1,4] ="Transit";                            // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,4]     =120;                                          // If 0 then default
will be used
Aircraft_Module[1,5] ="Transit";                            // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,5]     =120;                                          // If 0 then default
will be used
Aircraft_Module[1,6] ="Transit";                            // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,6]     =120;                                          // If 0 then default
will be used
Aircraft_Module[1,7] ="Transit";                            // Transit or Benign Only
Module_Time[1,7]     =120;                                          // If 0 then default
will be used
Aircraft_Module[1,8] ="Losing Changeover";          // Losing Changeover or Losing
Handover Only
Module_Time[1,8]     =10;                                           // If 0 then default
will be used
/////////////////////////////////Aircraft 2/////////////////////////////
Start_Time[2]=0;                                            //Use in Aircraft Control to
release entities into Aircraft functions
```

```
Aircraft_Module[2,1] ="Changeover";                          // Changeover or Handover
Only
Module_Time[2,1]     =10;                                             // If 0 then default
will be used
Aircraft_Module[2,2] ="Transit";                             // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,2]     =120;                                            // If 0 then default
will be used
Aircraft_Module[2,3] ="Transit";                             // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,3]     =120;                                            // If 0 then default
will be used
Aircraft_Module[2,4] ="Transit";                             // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,4]     =120;                                            // If 0 then default
will be used
Aircraft_Module[2,5] ="Transit";                             // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,5]     =120;                                            // If 0 then default
will be used
Aircraft_Module[2,6] ="Transit";                             // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,6]     =120;                                            // If 0 then default
will be used
Aircraft_Module[2,7] ="Transit";                             // Transit or Benign Only
Module_Time[2,7]     =120;                                            // If 0 then default
will be used
Aircraft_Module[2,8] ="Losing Changeover";          // Losing Changeover or Losing
Handover Only
Module_Time[2,8]     =10;                                             // If 0 then default
will be used
//////////////////////////////////Aircraft 3/////////////////////////////
Start_Time[3]=0;                                             //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[3,1] ="Changeover";                          // Changeover or Handover
Only
Module_Time[3,1]     =10;                                             // If 0 then default
will be used
Aircraft_Module[3,2] ="Dynamic";                             // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,2]     =120;                                            // If 0 then default
will be used
```

```
Aircraft_Module[3,3] ="Dynamic";                    // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,3]    =120;                                  // If 0 then default
will be used
Aircraft_Module[3,4] ="Dynamic";                    // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,4]    =120;                                  // If 0 then default
will be used
Aircraft_Module[3,5] ="Transit";                    // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,5]    =120;                                  // If 0 then default
will be used
Aircraft_Module[3,6] ="Transit";                    // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,6]    =120;                                  // If 0 then default
will be used
Aircraft_Module[3,7] ="Transit";                    // Transit or Benign Only
Module_Time[3,7]    =120;                                  // If 0 then default
will be used
Aircraft_Module[3,8] ="Losing Changeover";      // Losing Changeover or Losing
Handover Only
Module_Time[3,8]    =10;                                   // If 0 then default
will be used
//////////////////////////////////Aircraft 4/////////////////////////////
Start_Time[4]=0;                                    //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[4,1] ="Changeover";                 // Changeover or Handover
Only
Module_Time[4,1]    =10;                                   // If 0 then default
will be used
Aircraft_Module[4,2] ="Benign";                     // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[4,2]    =120;                                  // If 0 then default
will be used
Aircraft_Module[4,3] ="Emergency";                         // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[4,3]    =120;                                  // If 0 then default
will be used
Aircraft_Module[4,4] ="Dynamic";                    // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[4,4]    =120;                                  // If 0 then default
will be used
```

Aircraft_Module[4,5] ="Emergency"; // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[4,5] =120; // If 0 then default
will be used
Aircraft_Module[4,6] ="Dynamic"; // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[4,6] =120; // If 0 then default
will be used
Aircraft_Module[4,7] ="Transit"; // Transit or Benign Only
Module_Time[4,7] =120; // If 0 then default
will be used
Aircraft_Module[4,8] ="Losing Changeover"; // Losing Changeover or Losing
Handover Only
Module_Time[4,8] =10; // If 0 then default
will be used
------------------Run 8--

Aircraft_Used=4; //Number of Aircraft to Release
Model.PrintOutput("Number of Aircraft Released: " + Aircraft_Used);
//////////////////////////////////Aircraft 1//////////////////////////////
Start_Time[1]=0; //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[1,1] ="Handover"; // Changeover or Handover Only
Module_Time[1,1] =10; // If 0 then default
will be used
Aircraft_Module[1,2] ="Transit"; // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,2] =120; // If 0 then default
will be used
Aircraft_Module[1,3] ="Transit"; // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,3] =120; // If 0 then default
will be used
Aircraft_Module[1,4] ="Transit"; // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,4] =120; // If 0 then default
will be used
Aircraft_Module[1,5] ="Transit"; // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,5] =120; // If 0 then default
will be used

```
Aircraft_Module[1,6] ="Transit";                    // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,6]     =120;                                 // If 0 then default
will be used
Aircraft_Module[1,7] ="Transit";                    // Transit or Benign Only
Module_Time[1,7]     =120;                                 // If 0 then default
will be used
Aircraft_Module[1,8] ="Losing Handover"; // Losing Changeover or Losing Handover
Only
Module_Time[1,8]     =10;                                  // If 0 then default
will be used
//////////////////////////////////Aircraft 2/////////////////////////////
Start_Time[2]=0;                                    //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[2,1] ="Handover";               // Changeover or Handover Only
Module_Time[2,1]     =10;                                  // If 0 then default
will be used
Aircraft_Module[2,2] ="Dynamic";                    // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,2]     =120;                                 // If 0 then default
will be used
Aircraft_Module[2,3] ="Dynamic";                    // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,3]     =120;                                 // If 0 then default
will be used
Aircraft_Module[2,4] ="Dynamic";                    // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,4]     =120;                                 // If 0 then default
will be used
Aircraft_Module[2,5] ="Transit";                    // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,5]     =120;                                 // If 0 then default
will be used
Aircraft_Module[2,6] ="Transit";                    // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,6]     =120;                                 // If 0 then default
will be used
Aircraft_Module[2,7] ="Transit";                    // Transit or Benign Only
Module_Time[2,7]     =120;                                 // If 0 then default
will be used
Aircraft_Module[2,8] ="Losing Handover"; // Losing Changeover or Losing Handover
Only
```

```
Module_Time[2,8]     =10;                                        // If 0 then default
will be used
/////////////////////////////////Aircraft 3/////////////////////////////
Start_Time[3]=0;                                        //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[3,1] ="Handover";                   // Changeover or Handover Only
Module_Time[3,1]     =10;                                        // If 0 then default
will be used
Aircraft_Module[3,2] ="Benign";                         // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,2]     =120;                                       // If 0 then default
will be used
Aircraft_Module[3,3] ="Emergency";                              // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[3,3]     =120;                                       // If 0 then default
will be used
Aircraft_Module[3,4] ="Dynamic";                        // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,4]     =120;                                       // If 0 then default
will be used
Aircraft_Module[3,5] ="Emergency";                              // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[3,5]     =120;                                       // If 0 then default
will be used
Aircraft_Module[3,6] ="Emergency";                              // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[3,6]     =120;                                       // If 0 then default
will be used
Aircraft_Module[3,7] ="Transit";                        // Transit or Benign Only
Module_Time[3,7]     =120;                                       // If 0 then default
will be used
Aircraft_Module[3,8] ="Losing Handover"; // Losing Changeover or Losing Handover
Only
Module_Time[3,8]     =10;                                        // If 0 then default
will be used
/////////////////////////////////Aircraft 4/////////////////////////////
Start_Time[4]=0;                                        //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[4,1] ="Handover";                   // Changeover or Handover Only
Module_Time[4,1]     =10;                                        // If 0 then default
will be used
```

Aircraft_Module[4,2] ="Benign"; // Transit, Benign, Dynamic, Strike, Emergency
Module_Time[4,2] =120; // If 0 then default will be used
Aircraft_Module[4,3] ="Benign"; // Transit, Benign, Dynamic, Strike, Emergency
Module_Time[4,3] =120; // If 0 then default will be used
Aircraft_Module[4,4] ="Benign"; // Transit, Benign, Dynamic, Strike, Emergency
Module_Time[4,4] =120; // If 0 then default will be used
Aircraft_Module[4,5] ="Benign"; // Transit, Benign, Dynamic, Strike, Emergency
Module_Time[4,5] =120; // If 0 then default will be used
Aircraft_Module[4,6] ="Emergency"; // Transit, Benign, Dynamic, Strike, Emergency
Module_Time[4,6] =120; // If 0 then default will be used
Aircraft_Module[4,7] ="Benign"; // Transit or Benign Only
Module_Time[4,7] =120; // If 0 then default will be used
Aircraft_Module[4,8] ="Losing Handover"; // Losing Changeover or Losing Handover Only
Module_Time[4,8] =10; // If 0 then default will be used
------------------Run 9--
Aircraft_Used=4; //Number of Aircraft to Release
Model.PrintOutput("Number of Aircraft Released: " + Aircraft_Used);
/////////////////////////////////Aircraft 1/////////////////////////////
Start_Time[1]=0; //Use in Aircraft Control to release entities into Aircraft functions
Aircraft_Module[1,1] ="Changeover"; // Changeover or Handover Only
Module_Time[1,1] =10; // If 0 then default will be used
Aircraft_Module[1,2] ="Dynamic"; // Transit, Benign, Dynamic, Strike, Emergency
Module_Time[1,2] =120; // If 0 then default will be used

```
Aircraft_Module[1,3] ="Dynamic";                    // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,3]     =120;                               // If 0 then default
will be used
Aircraft_Module[1,4] ="Dynamic";                    // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,4]     =120;                               // If 0 then default
will be used
Aircraft_Module[1,5] ="Transit";                    // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,5]     =120;                               // If 0 then default
will be used
Aircraft_Module[1,6] ="Transit";                    // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,6]     =120;                               // If 0 then default
will be used
Aircraft_Module[1,7] ="Transit";                    // Transit or Benign Only
Module_Time[1,7]     =120;                               // If 0 then default
will be used
Aircraft_Module[1,8] ="Losing Changeover";     // Losing Changeover or Losing
Handover Only
Module_Time[1,8]     =10;                                // If 0 then default
will be used
//////////////////////////////////Aircraft 2//////////////////////////////
Start_Time[2]=0;                                    //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[2,1] ="Changeover";                 // Changeover or Handover
Only
Module_Time[2,1]     =10;                                // If 0 then default
will be used
Aircraft_Module[2,2] ="Benign";                     // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,2]     =120;                               // If 0 then default
will be used
Aircraft_Module[2,3] ="Emergency";                       // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[2,3]     =120;                               // If 0 then default
will be used
Aircraft_Module[2,4] ="Dynamic";                    // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,4]     =120;                               // If 0 then default
will be used
```

```
Aircraft_Module[2,5] ="Emergency";                              // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[2,5]     =120;                                      // If 0 then default
will be used
Aircraft_Module[2,6] ="Emergency";                              // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[2,6]     =120;                                      // If 0 then default
will be used
Aircraft_Module[2,7] ="Transit";                          // Transit or Benign Only
Module_Time[2,7]     =120;                                      // If 0 then default
will be used
Aircraft_Module[2,8] ="Losing Changeover";          // Losing Changeover or Losing
Handover Only
Module_Time[2,8]     =10;                                       // If 0 then default
will be used
//////////////////////////////////Aircraft 3//////////////////////////////
Start_Time[3]=0;                                          //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[3,1] ="Handover";                   // Changeover or Handover Only
Module_Time[3,1]     =10;                                       // If 0 then default
will be used
Aircraft_Module[3,2] ="Benign";                           // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,2]     =120;                                      // If 0 then default
will be used
Aircraft_Module[3,3] ="Benign";                           // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,3]     =120;                                      // If 0 then default
will be used
Aircraft_Module[3,4] ="Benign";                           // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,4]     =120;                                      // If 0 then default
will be used
Aircraft_Module[3,5] ="Benign";                           // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,5]     =120;                                      // If 0 then default
will be used
Aircraft_Module[3,6] ="Emergency";                              // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[3,6]     =120;                                      // If 0 then default
will be used
Aircraft_Module[3,7] ="Benign";                           // Transit or Benign Only
```

Module_Time[3,7]   =120;   // If 0 then default will be used
Aircraft_Module[3,8] ="Losing Handover"; // Losing Changeover or Losing Handover Only
Module_Time[3,8]   =10;   // If 0 then default will be used
/////////////////////////////////Aircraft 4/////////////////////////////
Start_Time[4]=0;   //Use in Aircraft Control to release entities into Aircraft functions
Aircraft_Module[4,1] ="Handover";   // Changeover or Handover Only
Module_Time[4,1]   =10;   // If 0 then default will be used
Aircraft_Module[4,2] ="Benign";   // Transit, Benign, Dynamic, Strike, Emergency
Module_Time[4,2]   =120;   // If 0 then default will be used
Aircraft_Module[4,3] ="Benign";   // Transit, Benign, Dynamic, Strike, Emergency
Module_Time[4,3]   =120;   // If 0 then default will be used
Aircraft_Module[4,4] ="Benign";   // Transit, Benign, Dynamic, Strike, Emergency
Module_Time[4,4]   =120;   // If 0 then default will be used
Aircraft_Module[4,5] ="Benign";   // Transit, Benign, Dynamic, Strike, Emergency
Module_Time[4,5]   =120;   // If 0 then default will be used
Aircraft_Module[4,6] ="Benign";   // Transit, Benign, Dynamic, Strike, Emergency
Module_Time[4,6]   =120;   // If 0 then default will be used
Aircraft_Module[4,7] ="Benign";   // Transit or Benign Only
Module_Time[4,7]   =120;   // If 0 then default will be used
Aircraft_Module[4,8] ="Losing Handover"; // Losing Changeover or Losing Handover Only
Module_Time[4,8]   =10;   // If 0 then default will be used
------------------Run 10---
Aircraft_Used=4;   //Number of Aircraft to Release

```
Model.PrintOutput("Number of Aircraft Released: " + Aircraft_Used);
//////////////////////////////////Aircraft 1/////////////////////////////
Start_Time[1]=0;                                    //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[1,1] ="Changeover";                 // Changeover or Handover
Only
Module_Time[1,1]     =10;                                  // If 0 then default
will be used
Aircraft_Module[1,2] ="Emergency";                         // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[1,2]     =120;                                 // If 0 then default
will be used
Aircraft_Module[1,3] ="Emergency";                         // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[1,3]     =120;                                 // If 0 then default
will be used
Aircraft_Module[1,4] ="Benign";                     // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,4]     =120;                                 // If 0 then default
will be used
Aircraft_Module[1,5] ="Transit";                    // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,5]     =120;                                 // If 0 then default
will be used
Aircraft_Module[1,6] ="Losing Handover";                   // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[1,6]     =120;                                 // If 0 then default
will be used
Aircraft_Module[1,7] ="Transit";                    // Transit or Benign Only
Module_Time[1,7]     =120;                                 // If 0 then default
will be used
Aircraft_Module[1,8] ="Losing Handover"; // Losing Changeover or Losing Handover
Only
Module_Time[1,8]     =10;                                  // If 0 then default
will be used
//////////////////////////////////Aircraft 2/////////////////////////////
Start_Time[2]=0;                                    //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[2,1] ="Changeover";                 // Changeover or Handover
Only
Module_Time[2,1]     =10;                                  // If 0 then default
will be used
```

Aircraft_Module[2,2] ="Emergency"; // Transit, Benign, Dynamic, Strike, Emergency
Module_Time[2,2] =120; // If 0 then default will be used
Aircraft_Module[2,3] ="Benign"; // Transit, Benign, Dynamic, Strike, Emergency
Module_Time[2,3] =120; // If 0 then default will be used
Aircraft_Module[2,4] ="Benign"; // Transit, Benign, Dynamic, Strike, Emergency
Module_Time[2,4] =120; // If 0 then default will be used
Aircraft_Module[2,5] ="Benign"; // Transit, Benign, Dynamic, Strike, Emergency
Module_Time[2,5] =120; // If 0 then default will be used
Aircraft_Module[2,6] ="Losing Handover"; // Transit, Benign, Dynamic, Strike, Emergency
Module_Time[2,6] =120; // If 0 then default will be used
Aircraft_Module[2,7] ="Transit"; // Transit or Benign Only
Module_Time[2,7] =120; // If 0 then default will be used
Aircraft_Module[2,8] ="Losing Handover"; // Losing Changeover or Losing Handover Only
Module_Time[2,8] =10; // If 0 then default will be used
//////////////////////////////////Aircraft 3/////////////////////////////
Start_Time[3]=0; //Use in Aircraft Control to release entities into Aircraft functions
Aircraft_Module[3,1] ="Changeover"; // Changeover or Handover Only
Module_Time[3,1] =10; // If 0 then default will be used
Aircraft_Module[3,2] ="Benign"; // Transit, Benign, Dynamic, Strike, Emergency
Module_Time[3,2] =120; // If 0 then default will be used
Aircraft_Module[3,3] ="Benign"; // Transit, Benign, Dynamic, Strike, Emergency
Module_Time[3,3] =120; // If 0 then default will be used

```
Aircraft_Module[3,4] ="Benign";                    // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,4]     =120;                               // If 0 then default
will be used
Aircraft_Module[3,5] ="Benign";                    // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,5]     =120;                               // If 0 then default
will be used
Aircraft_Module[3,6] ="Losing Handover";                 // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[3,6]     =120;                               // If 0 then default
will be used
Aircraft_Module[3,7] ="Transit";                   // Transit or Benign Only
Module_Time[3,7]     =120;                               // If 0 then default
will be used
Aircraft_Module[3,8] ="Losing Handover"; // Losing Changeover or Losing Handover
Only
Module_Time[3,8]     =10;                                // If 0 then default
will be used
//////////////////////////////////Aircraft 4/////////////////////////////
Start_Time[4]=0;                                   //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[4,1] ="Changeover";                // Changeover or Handover
Only
Module_Time[4,1]     =10;                                // If 0 then default
will be used
Aircraft_Module[4,2] ="Benign";                    // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[4,2]     =120;                               // If 0 then default
will be used
Aircraft_Module[4,3] ="Benign";                    // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[4,3]     =120;                               // If 0 then default
will be used
Aircraft_Module[4,4] ="Benign";                    // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[4,4]     =120;                               // If 0 then default
will be used
Aircraft_Module[4,5] ="Benign";                    // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[4,5]     =120;                               // If 0 then default
will be used
```

Aircraft_Module[4,6] ="Losing Handover"; // Transit, Benign, Dynamic, Strike, Emergency
Module_Time[4,6] =120; // If 0 then default will be used
Aircraft_Module[4,7] ="Benign"; // Transit or Benign Only
Module_Time[4,7] =120; // If 0 then default will be used
Aircraft_Module[4,8] ="Losing Handover"; // Losing Changeover or Losing Handover Only
Module_Time[4,8] =10; // If 0 then default will be used

Phase II Comparison Run Code

```
------------------BISR Run---------------------------------------------------------------------------
---------------
Aircraft_Used=1;                    //Number of Aircraft to Release
Model.PrintOutput("Number of Aircraft Released: " + Aircraft_Used);
///////////////////////////////////Aircraft 1//////////////////////////////
Start_Time[1]=0;                                            //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[1,1] ="Changeover";                         // Changeover or Handover
Only
Module_Time[1,1]     =0;                                          // If 0 then default
will be used
Aircraft_Module[1,2] ="Benign";                             // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,2]     =Distributions.Triangular(130,         120,     160);
                     // If 0 then default will be used
Aircraft_Module[1,3] ="Losing Changeover";                              // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[1,3]     =0;                                          // If 0 then default
will be used
Aircraft_Module[1,4] ="Benign";                             // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,4]     =120;                                        // If 0 then default
will be used
Aircraft_Module[1,5] ="Losing Changeover";                              // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[1,5]     =0;                                          // If 0 then default
will be used
Aircraft_Module[1,6] ="Transit";                            // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,6]     =120;                                        // If 0 then default
will be used
Aircraft_Module[1,7] ="Transit";                            // Transit or Benign Only
Module_Time[1,7]     =120;                                        // If 0 then default
will be used
Aircraft_Module[1,8] ="Losing Changeover";          // Losing Changeover or Losing
Handover Only
Module_Time[1,8]     =10;                                         // If 0 then default
will be used
///////////////////////////////////Aircraft 2//////////////////////////////
```

```
Start_Time[2]=0;                                        //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[2,1] ="Changeover";                     // Changeover or Handover
Only
Module_Time[2,1]     =0;                                     // If 0 then default
will be used
Aircraft_Module[2,2] ="Benign";                         // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,2]     =Distributions.Triangular(130,     120,    160);
                     // If 0 then default will be used
Aircraft_Module[2,3] ="Losing Changeover";                        // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[2,3]     =0;                                     // If 0 then default
will be used
Aircraft_Module[2,4] ="Losing Changeover";                        // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[2,4]     =0;                                     // If 0 then default
will be used
Aircraft_Module[2,5] ="Emergency";                           // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[2,5]     =120;                                   // If 0 then default
will be used
Aircraft_Module[2,6] ="Emergency";                           // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[2,6]     =120;                                   // If 0 then default
will be used
Aircraft_Module[2,7] ="Transit";                        // Transit or Benign Only
Module_Time[2,7]     =120;                                   // If 0 then default
will be used
Aircraft_Module[2,8] ="Losing Changeover";         // Losing Changeover or Losing
Handover Only
Module_Time[2,8]     =10;                                    // If 0 then default
will be used
/////////////////////////////////Aircraft 3/////////////////////////////
Start_Time[3]=0;                                        //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[3,1] ="Changeover";                     // Changeover or Handover
Only
Module_Time[3,1]     =0;                                     // If 0 then default
will be used
Aircraft_Module[3,2] ="Benign";                         // Transit, Benign, Dynamic,
Strike, Emergency
```

```
Module_Time[3,2]     =Distributions.Triangular(130,          120,     160);
                         // If 0 then default will be used
Aircraft_Module[3,3] ="Losing Changeover";                               // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[3,3]     =0;                                        // If 0 then default
will be used
Aircraft_Module[3,4] ="Benign";                            // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,4]     =120;                                      // If 0 then default
will be used
Aircraft_Module[3,5] ="Benign";                            // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,5]     =120;                                      // If 0 then default
will be used
Aircraft_Module[3,6] ="Emergency";                              // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[3,6]     =120;                                      // If 0 then default
will be used
Aircraft_Module[3,7] ="Benign";                            // Transit or Benign Only
Module_Time[3,7]     =120;                                      // If 0 then default
will be used
Aircraft_Module[3,8] ="Losing Changeover";         // Losing Changeover or Losing
Handover Only
Module_Time[3,8]     =10;                                       // If 0 then default
will be used
//////////////////////////////////Aircraft 4/////////////////////////////
Start_Time[4]=0;                                           //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[4,1] ="Changeover";                        // Changeover or Handover
Only
Module_Time[4,1]     =0;                                        // If 0 then default
will be used
Aircraft_Module[4,2] ="Benign";                            // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[4,2]     =Distributions.Triangular(130,          120,     160);
                         // If 0 then default will be used
Aircraft_Module[4,3] ="Losing Changeover";                               // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[4,3]     =0;                                        // If 0 then default
will be used
Aircraft_Module[4,4] ="Benign";                            // Transit, Benign, Dynamic,
Strike, Emergency
```

```
Module_Time[4,4]     =Distributions.Triangular(30, 15,       40);
                // If 0 then default will be used
Aircraft_Module[4,5] ="Losing Changeover";                              // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[4,5]     =0;                                         // If 0 then default
will be used
Aircraft_Module[4,6] ="Benign";                          // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[4,6]     =120;                                       // If 0 then default
will be used
Aircraft_Module[4,7] ="Benign";                          // Transit or Benign Only
Module_Time[4,7]     =120;                                       // If 0 then default
will be used
Aircraft_Module[4,8] ="Losing Changeover";         // Losing Changeover or Losing
Handover Only
Module_Time[4,8]     =10;                                        // If 0 then default
will be used
------------------DISR Run----------------------------------------------------------------------------
---------------
Aircraft_Used=1;                  //Number of Aircraft to Release
Model.PrintOutput("Number of Aircraft Released: " + Aircraft_Used);
//////////////////////////////////Aircraft 1/////////////////////////////
Start_Time[1]=0;                                         //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[1,1] ="Changeover";                      // Changeover or Handover
Only
Module_Time[1,1]     =0;                                         // If 0 then default
will be used
Aircraft_Module[1,2] ="Benign";                          // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,2]     =Distributions.Triangular(60, 45,       80);
                // If 0 then default will be used
Aircraft_Module[1,3] ="Dynamic";                         // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,3]     =30;                                        // If 0 then default
will be used
Aircraft_Module[1,4] ="Benign";                          // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,4]     =Distributions.Triangular(60, 45,       80);
                // If 0 then default will be used
Aircraft_Module[1,5] ="Losing Changeover";                              // Transit,
Benign, Dynamic, Strike, Emergency
```

```
Module_Time[1,5]     =0;                                        // If 0 then default
will be used
Aircraft_Module[1,6] ="Transit";                        // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,6]     =120;                                      // If 0 then default
will be used
Aircraft_Module[1,7] ="Transit";                        // Transit or Benign Only
Module_Time[1,7]     =120;                                      // If 0 then default
will be used
Aircraft_Module[1,8] ="Losing Changeover";      // Losing Changeover or Losing
Handover Only
Module_Time[1,8]     =10;                                       // If 0 then default
will be used
//////////////////////////////////Aircraft 2/////////////////////////////
Start_Time[2]=0;                                        //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[2,1] ="Changeover";                     // Changeover or Handover
Only
Module_Time[2,1]     =0;                                        // If 0 then default
will be used
Aircraft_Module[2,2] ="Benign";                         // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,2]     =Distributions.Triangular(130,     120,    160);
                     // If 0 then default will be used
Aircraft_Module[2,3] ="Losing Changeover";                              // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[2,3]     =0;                                        // If 0 then default
will be used
Aircraft_Module[2,4] ="Losing Changeover";                              // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[2,4]     =0;                                        // If 0 then default
will be used
Aircraft_Module[2,5] ="Emergency";                              // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[2,5]     =120;                                      // If 0 then default
will be used
Aircraft_Module[2,6] ="Emergency";                              // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[2,6]     =120;                                      // If 0 then default
will be used
Aircraft_Module[2,7] ="Transit";                        // Transit or Benign Only
```

```
Module_Time[2,7]     =120;                                        // If 0 then default
will be used
Aircraft_Module[2,8] ="Losing Changeover";         // Losing Changeover or Losing
Handover Only
Module_Time[2,8]     =10;                                         // If 0 then default
will be used
////////////////////////////////////Aircraft 3//////////////////////////////
Start_Time[3]=0;                                        //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[3,1] ="Changeover";                     // Changeover or Handover
Only
Module_Time[3,1]     =0;                                          // If 0 then default
will be used
Aircraft_Module[3,2] ="Benign";                         // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,2]     =Distributions.Triangular(130,          120,     160);
                     // If 0 then default will be used
Aircraft_Module[3,3] ="Losing Changeover";                                  // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[3,3]     =0;                                          // If 0 then default
will be used
Aircraft_Module[3,4] ="Benign";                         // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,4]     =120;                                        // If 0 then default
will be used
Aircraft_Module[3,5] ="Benign";                         // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,5]     =120;                                        // If 0 then default
will be used
Aircraft_Module[3,6] ="Emergency";                                // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[3,6]     =120;                                        // If 0 then default
will be used
Aircraft_Module[3,7] ="Benign";                         // Transit or Benign Only
Module_Time[3,7]     =120;                                        // If 0 then default
will be used
Aircraft_Module[3,8] ="Losing Changeover";         // Losing Changeover or Losing
Handover Only
Module_Time[3,8]     =10;                                         // If 0 then default
will be used
////////////////////////////////////Aircraft 4//////////////////////////////
```

```
Start_Time[4]=0;                                //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[4,1] ="Changeover";             // Changeover or Handover
Only
Module_Time[4,1]    =0;                              // If 0 then default
will be used
Aircraft_Module[4,2] ="Benign";                 // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[4,2]    =Distributions.Triangular(130,    120,    160);
                    // If 0 then default will be used
Aircraft_Module[4,3] ="Losing Changeover";                  // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[4,3]    =0;                              // If 0 then default
will be used
Aircraft_Module[4,4] ="Benign";                 // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[4,4]    =Distributions.Triangular(30, 15,    40);
               // If 0 then default will be used
Aircraft_Module[4,5] ="Losing Changeover";                  // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[4,5]    =0;                              // If 0 then default
will be used
Aircraft_Module[4,6] ="Benign";                 // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[4,6]    =120;                            // If 0 then default
will be used
Aircraft_Module[4,7] ="Benign";                 // Transit or Benign Only
Module_Time[4,7]    =120;                            // If 0 then default
will be used
Aircraft_Module[4,8] ="Losing Changeover";      // Losing Changeover or Losing
Handover Only
Module_Time[4,8]    =10;                             // If 0 then default
will be used
------------------Transit Run--------------------------------------------------------------------------
-----------------
Aircraft_Used=1;          //Number of Aircraft to Release
Model.PrintOutput("Number of Aircraft Released: " + Aircraft_Used);
//////////////////////////////////Aircraft 1//////////////////////////////
Start_Time[1]=0;                                //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[1,1] ="Changeover";             // Changeover or Handover
Only
```

```
Module_Time[1,1]     =0;                                          // If 0 then default
will be used
Aircraft_Module[1,2] ="Transit";                            // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,2]     =Distributions.Triangular(130,      120,    160);
                     // If 0 then default will be used
Aircraft_Module[1,3] ="Losing Changeover";                            // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[1,3]     =0;                                          // If 0 then default
will be used
Aircraft_Module[1,4] ="Benign";                             // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,4]     =120;                                        // If 0 then default
will be used
Aircraft_Module[1,5] ="Losing Changeover";                            // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[1,5]     =0;                                          // If 0 then default
will be used
Aircraft_Module[1,6] ="Transit";                            // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,6]     =120;                                        // If 0 then default
will be used
Aircraft_Module[1,7] ="Transit";                            // Transit or Benign Only
Module_Time[1,7]     =120;                                        // If 0 then default
will be used
Aircraft_Module[1,8] ="Losing Changeover";       // Losing Changeover or Losing
Handover Only
Module_Time[1,8]     =10;                                         // If 0 then default
will be used
/////////////////////////////////Aircraft 2/////////////////////////////
Start_Time[2]=0;                                            //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[2,1] ="Changeover";                         // Changeover or Handover
Only
Module_Time[2,1]     =0;                                          // If 0 then default
will be used
Aircraft_Module[2,2] ="Transit";                            // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,2]     =Distributions.Triangular(130,      120,    160);
                     // If 0 then default will be used
Aircraft_Module[2,3] ="Losing Changeover";                            // Transit,
Benign, Dynamic, Strike, Emergency
```

```
Module_Time[2,3]     =0;                                          // If 0 then default
will be used
Aircraft_Module[2,4] ="Losing Changeover";                                  // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[2,4]     =0;                                          // If 0 then default
will be used
Aircraft_Module[2,5] ="Emergency";                                // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[2,5]     =120;                                        // If 0 then default
will be used
Aircraft_Module[2,6] ="Emergency";                                // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[2,6]     =120;                                        // If 0 then default
will be used
Aircraft_Module[2,7] ="Transit";                          // Transit or Benign Only
Module_Time[2,7]     =120;                                        // If 0 then default
will be used
Aircraft_Module[2,8] ="Losing Changeover";        // Losing Changeover or Losing
Handover Only
Module_Time[2,8]     =10;                                         // If 0 then default
will be used
///////////////////////////////////Aircraft 3/////////////////////////////
Start_Time[3]=0;                                          //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[3,1] ="Changeover";                       // Changeover or Handover
Only
Module_Time[3,1]     =0;                                          // If 0 then default
will be used
Aircraft_Module[3,2] ="Transit";                          // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,2]     =Distributions.Triangular(130,       120,     160);
                     // If 0 then default will be used
Aircraft_Module[3,3] ="Losing Changeover";                                  // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[3,3]     =0;                                          // If 0 then default
will be used
Aircraft_Module[3,4] ="Transit";                          // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,4]     =120;                                        // If 0 then default
will be used
Aircraft_Module[3,5] ="Benign";                           // Transit, Benign, Dynamic,
Strike, Emergency
```

```
Module_Time[3,5]     =120;                                          // If 0 then default
will be used
Aircraft_Module[3,6] ="Emergency";                                  // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[3,6]     =120;                                          // If 0 then default
will be used
Aircraft_Module[3,7] ="Benign";                             // Transit or Benign Only
Module_Time[3,7]     =120;                                          // If 0 then default
will be used
Aircraft_Module[3,8] ="Losing Changeover";          // Losing Changeover or Losing
Handover Only
Module_Time[3,8]     =10;                                           // If 0 then default
will be used
//////////////////////////////////Aircraft 4/////////////////////////////
Start_Time[4]=0;                                            //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[4,1] ="Changeover";                         // Changeover or Handover
Only
Module_Time[4,1]     =0;                                            // If 0 then default
will be used
Aircraft_Module[4,2] ="Transit";                            // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[4,2]     =Distributions.Triangular(130,         120,     160);
                     // If 0 then default will be used
Aircraft_Module[4,3] ="Losing Changeover";                                  // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[4,3]     =0;                                            // If 0 then default
will be used
Aircraft_Module[4,4] ="Benign";                             // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[4,4]     =Distributions.Triangular(30, 15,      40);
                // If 0 then default will be used
Aircraft_Module[4,5] ="Losing Changeover";                                  // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[4,5]     =0;                                            // If 0 then default
will be used
Aircraft_Module[4,6] ="Benign";                             // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[4,6]     =120;                                          // If 0 then default
will be used
Aircraft_Module[4,7] ="Benign";                             // Transit or Benign Only
```

```
Module_Time[4,7]     =120;                                        // If 0 then default
will be used
Aircraft_Module[4,8] ="Losing Changeover";         // Losing Changeover or Losing
Handover Only
Module_Time[4,8]     =10;                                         // If 0 then default
will be used
------------------Emergency Run-------------------------------------------------------------------
----------------------
Aircraft_Used=1;                //Number of Aircraft to Release
Model.PrintOutput("Number of Aircraft Released: " + Aircraft_Used);
//////////////////////////////////Aircraft 1/////////////////////////////
Start_Time[1]=0;                                       //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[1,1] ="Changeover";                    // Changeover or Handover
Only
Module_Time[1,1]     =0;                                          // If 0 then default
will be used
Aircraft_Module[1,2] ="Transit";                       // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,2]     =Distributions.Triangular(60, 45,      80);
                    // If 0 then default will be used
Aircraft_Module[1,3] ="Emergency";                                // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[1,3]     =30;                                         // If 0 then default
will be used
Aircraft_Module[1,4] ="Transit";                       // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,4]     =Distributions.Triangular(60, 45,      80);
                    // If 0 then default will be used
Aircraft_Module[1,5] ="Losing Changeover";                                  // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[1,5]     =0;                                          // If 0 then default
will be used
Aircraft_Module[1,6] ="Transit";                       // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,6]     =120;                                        // If 0 then default
will be used
Aircraft_Module[1,7] ="Transit";                       // Transit or Benign Only
Module_Time[1,7]     =120;                                        // If 0 then default
will be used
Aircraft_Module[1,8] ="Losing Changeover";         // Losing Changeover or Losing
Handover Only
```

```
Module_Time[1,8]     =10;                                           // If 0 then default
will be used
/////////////////////////////////Aircraft 2/////////////////////////////
Start_Time[2]=0;                                         //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[2,1] ="Changeover";                      // Changeover or Handover
Only
Module_Time[2,1]     =0;                                            // If 0 then default
will be used
Aircraft_Module[2,2] ="Transit";                         // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,2]     =Distributions.Triangular(130,      120,    160);
                     // If 0 then default will be used
Aircraft_Module[2,3] ="Losing Changeover";                          // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[2,3]     =0;                                            // If 0 then default
will be used
Aircraft_Module[2,4] ="Losing Changeover";                          // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[2,4]     =0;                                            // If 0 then default
will be used
Aircraft_Module[2,5] ="Emergency";                                  // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[2,5]     =120;                                          // If 0 then default
will be used
Aircraft_Module[2,6] ="Emergency";                                  // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[2,6]     =120;                                          // If 0 then default
will be used
Aircraft_Module[2,7] ="Transit";                         // Transit or Benign Only
Module_Time[2,7]     =120;                                          // If 0 then default
will be used
Aircraft_Module[2,8] ="Losing Changeover";          // Losing Changeover or Losing
Handover Only
Module_Time[2,8]     =10;                                           // If 0 then default
will be used
/////////////////////////////////Aircraft 3/////////////////////////////
Start_Time[3]=0;                                         //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[3,1] ="Changeover";                      // Changeover or Handover
Only
```

```
Module_Time[3,1]     =0;                                          // If 0 then default
will be used
Aircraft_Module[3,2] ="Transit";                          // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,2]     =Distributions.Triangular(130,       120,     160);
                     // If 0 then default will be used
Aircraft_Module[3,3] ="Losing Changeover";                              // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[3,3]     =0;                                          // If 0 then default
will be used
Aircraft_Module[3,4] ="Benign";                           // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,4]     =120;                                        // If 0 then default
will be used
Aircraft_Module[3,5] ="Benign";                           // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,5]     =120;                                        // If 0 then default
will be used
Aircraft_Module[3,6] ="Emergency";                                // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[3,6]     =120;                                        // If 0 then default
will be used
Aircraft_Module[3,7] ="Benign";                           // Transit or Benign Only
Module_Time[3,7]     =120;                                        // If 0 then default
will be used
Aircraft_Module[3,8] ="Losing Changeover";         // Losing Changeover or Losing
Handover Only
Module_Time[3,8]     =10;                                         // If 0 then default
will be used
//////////////////////////////////Aircraft 4/////////////////////////////
Start_Time[4]=0;                                          //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[4,1] ="Changeover";                       // Changeover or Handover
Only
Module_Time[4,1]     =0;                                          // If 0 then default
will be used
Aircraft_Module[4,2] ="Transit";                          // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[4,2]     =Distributions.Triangular(130,       120,     160);
                     // If 0 then default will be used
Aircraft_Module[4,3] ="Losing Changeover";                              // Transit,
Benign, Dynamic, Strike, Emergency
```

```
Module_Time[4,3]     =0;                                          // If 0 then default
will be used
Aircraft_Module[4,4] ="Benign";                             // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[4,4]     =Distributions.Triangular(30, 15,      40);
                     // If 0 then default will be used
Aircraft_Module[4,5] ="Losing Changeover";                              // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[4,5]     =0;                                          // If 0 then default
will be used
Aircraft_Module[4,6] ="Benign";                             // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[4,6]     =120;                                        // If 0 then default
will be used
Aircraft_Module[4,7] ="Benign";                             // Transit or Benign Only
Module_Time[4,7]     =120;                                        // If 0 then default
will be used
Aircraft_Module[4,8] ="Losing Changeover";        // Losing Changeover or Losing
Handover Only
Module_Time[4,8]     =10;                                         // If 0 then default
will be used
```

Phase II Complex Run Code

```
------------------Run 1---------------------------------------------------------------------------------
----------
Aircraft_Used=1;                  //Number of Aircraft to Release
Model.PrintOutput("Number of Aircraft Released: " + Aircraft_Used);
//////////////////////////////////Aircraft 1/////////////////////////////
Start_Time[1]=0;                                        //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[1,1] ="Changeover";                     // Changeover or Handover
Only
Module_Time[1,1]     =0;                                         // If 0 then default
will be used
Aircraft_Module[1,2] ="Transit";                        // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,2]     =Distributions.Triangular(35, 15,      45);
                  // If 0 then default will be used
Aircraft_Module[1,3] ="Benign";                         // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,3]     =Distributions.Triangular(30, 15,      45);
                  // If 0 then default will be used
Aircraft_Module[1,4] ="Dynamic";                        // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,4]     =Distributions.Triangular(40, 20,      50);
                  // If 0 then default will be used
Aircraft_Module[1,5] ="Benign";                         // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,5]     =Distributions.Triangular(45, 30,      60);
                  // If 0 then default will be used
Aircraft_Module[1,6] ="Losing Changeover";                              // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[1,6]     =0;                                         // If 0 then default
will be used
Aircraft_Module[1,7] ="Losing Changeover";                              // Transit or
Benign Only
Module_Time[1,7]     =10;                                        // If 0 then default
will be used
Aircraft_Module[1,8] ="Losing Changeover";         // Losing Changeover or Losing
Handover Only
Module_Time[1,8]     =0;                                         // If 0 then default
will be used
```

```
------------------Run 2--------------------------------------------------------------------------------
----------
Aircraft_Used=1;                  //Number of Aircraft to Release
Model.PrintOutput("Number of Aircraft Released: " + Aircraft_Used);
/////////////////////////////////Aircraft 1/////////////////////////////
Start_Time[1]=0;                                              //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[1,1] ="Handover";                  // Changeover or Handover Only
Module_Time[1,1]     =0;                                              // If 0 then default
will be used
Aircraft_Module[1,2] ="Transit";                              // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,2]     =Distributions.Triangular(60, 45,      75);
                  // If 0 then default will be used
Aircraft_Module[1,3] ="Emergency";                                // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[1,3]     =Distributions.Triangular(35, 15,      45);
                  // If 0 then default will be used
Aircraft_Module[1,4] ="Transit";                              // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,4]     =Distributions.Triangular(10, 8,       25);
                  // If 0 then default will be used
Aircraft_Module[1,5] ="Benign";                               // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,5]     =Distributions.Triangular(45, 27,      60);
                  // If 0 then default will be used
Aircraft_Module[1,6] ="Losing Changeover";                              // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[1,6]     =0;                                              // If 0 then default
will be used
Aircraft_Module[1,7] ="Losing Changeover";                              // Transit or
Benign Only
Module_Time[1,7]     =10;                                             // If 0 then default
will be used
Aircraft_Module[1,8] ="Losing Changeover";        // Losing Changeover or Losing
Handover Only
Module_Time[1,8]     =0;                                              // If 0 then default
will be used
------------------Run 3--------------------------------------------------------------------------------
----------
Aircraft_Used=1;                  //Number of Aircraft to Release
Model.PrintOutput("Number of Aircraft Released: " + Aircraft_Used);
```

```
//////////////////////////////////Aircraft 1/////////////////////////////
Start_Time[1]=0;                                         //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[1,1] ="Changeover";                      // Changeover or Handover
Only
Module_Time[1,1]     =0;                                       // If 0 then default
will be used
Aircraft_Module[1,2] ="Benign";                          // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,2]     =Distributions.Triangular(30, 15,   45);
            // If 0 then default will be used
Aircraft_Module[1,3] ="Dynamic";                         // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,3]     =Distributions.Triangular(20, 10,   25);
            // If 0 then default will be used
Aircraft_Module[1,4] ="Strike";                          // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,4]     =0;                                       // If 0 then default
will be used
Aircraft_Module[1,5] ="Dynamic";                         // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,5]     =Distributions.Triangular(15, 10,   20);
            // If 0 then default will be used
Aircraft_Module[1,6] ="Benign";                          // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,6]     =Distributions.Triangular(30, 10,   50);
            // If 0 then default will be used
Aircraft_Module[1,7] ="Transit";                         // Transit or Benign Only
Module_Time[1,7]     =Distributions.Triangular(20, 10,   25);
            // If 0 then default will be used
Aircraft_Module[1,8] ="Losing Handover"; // Losing Changeover or Losing Handover
Only
Module_Time[1,8]     =0;                                       // If 0 then default
will be used
------------------Run 4---------------------------------------------------------------------------------
----------
Aircraft_Used=2;           //Number of Aircraft to Release
Model.PrintOutput("Number of Aircraft Released: " + Aircraft_Used);
//////////////////////////////////Aircraft 1/////////////////////////////
Start_Time[1]=0;                                         //Use in Aircraft Control to
release entities into Aircraft functions
```

```
Aircraft_Module[1,1] ="Changeover";                     // Changeover or Handover
Only
Module_Time[1,1]     =0;                                      // If 0 then default
will be used
Aircraft_Module[1,2] ="Transit";                        // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,2]     =Distributions.Triangular(90, 75,     125);
                     // If 0 then default will be used
Aircraft_Module[1,3] ="Benign";                         // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,3]     =Distributions.Triangular(50, 30,     65);
                     // If 0 then default will be used
Aircraft_Module[1,4] ="Losing Changeover";                          // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[1,4]     =0;                                      // If 0 then default
will be used
Aircraft_Module[1,5] ="Dynamic";                        // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,5]     =120;                                    // If 0 then default
will be used
Aircraft_Module[1,6] ="Benign";                         // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,6]     =120;                                    // If 0 then default
will be used
Aircraft_Module[1,7] ="Losing Handover";                      // Transit or Benign
Only
Module_Time[1,7]     =10;                                     // If 0 then default
will be used
Aircraft_Module[1,8] ="Losing Changeover";        // Losing Changeover or Losing
Handover Only
Module_Time[1,8]     =0;                                      // If 0 then default
will be used
//////////////////////////////////Aircraft 2/////////////////////////////
Start_Time[2]=0;                                        //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[2,1] ="Changeover";                     // Changeover or Handover
Only
Module_Time[2,1]     =0;                                      // If 0 then default
will be used
Aircraft_Module[2,2] ="Transit";                        // Transit, Benign, Dynamic,
Strike, Emergency
```

```
Module_Time[2,2]     =Distributions.Triangular(50, 30,       65);
                    // If 0 then default will be used
Aircraft_Module[2,3] ="Benign";                          // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,3]     =Distributions.Triangular(90, 75,       125);
                    // If 0 then default will be used
Aircraft_Module[2,4] ="Losing Changeover";                                  // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[2,4]     =0;                                              // If 0 then default
will be used
Aircraft_Module[2,5] ="Emergency";                                    // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[2,5]     =120;                                            // If 0 then default
will be used
Aircraft_Module[2,6] ="Benign";                          // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,6]     =120;                                            // If 0 then default
will be used
Aircraft_Module[2,7] ="Losing Changeover";                                  // Transit or
Benign Only
Module_Time[2,7]     =10;                                             // If 0 then default
will be used
Aircraft_Module[2,8] ="Losing Changeover";         // Losing Changeover or Losing
Handover Only
Module_Time[2,8]     =0;                                              // If 0 then default
will be used
------------------Run 5--------------------------------------------------------------------------------
----------
Aircraft_Used=2;                  //Number of Aircraft to Release
Model.PrintOutput("Number of Aircraft Released: " + Aircraft_Used);
/////////////////////////////////Aircraft 1/////////////////////////////
Start_Time[1]=0;                                         //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[1,1] ="Changeover";                      // Changeover or Handover
Only
Module_Time[1,1]     =0;                                              // If 0 then default
will be used
Aircraft_Module[1,2] ="Benign";                          // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,2]     =Distributions.Triangular(20, 15,       35);
                    // If 0 then default will be used
```

```
Aircraft_Module[1,3] ="Dynamic";                              // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,3]     =Distributions.Triangular(50, 30,       65);
                  // If 0 then default will be used
Aircraft_Module[1,4] ="Benign";                               // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,4]     =Distributions.Triangular(60, 30,       65);
                  // If 0 then default will be used
Aircraft_Module[1,5] ="Losing Changeover";                                    // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[1,5]     =0;                                              // If 0 then default
will be used
Aircraft_Module[1,6] ="Benign";                               // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,6]     =120;                                            // If 0 then default
will be used
Aircraft_Module[1,7] ="Losing Handover";                              // Transit or Benign
Only
Module_Time[1,7]     =10;                                             // If 0 then default
will be used
Aircraft_Module[1,8] ="Losing Changeover";         // Losing Changeover or Losing
Handover Only
Module_Time[1,8]     =0;                                              // If 0 then default
will be used
//////////////////////////////////Aircraft 2/////////////////////////////
Start_Time[2]=0;                                              //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[2,1] ="Changeover";                           // Changeover or Handover
Only
Module_Time[2,1]     =0;                                              // If 0 then default
will be used
Aircraft_Module[2,2] ="Benign";                               // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,2]     =Distributions.Triangular(50, 30,       65);
                  // If 0 then default will be used
Aircraft_Module[2,3] ="Transit";                              // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,3]     =Distributions.Triangular(30, 15,       45);
                  // If 0 then default will be used
Aircraft_Module[2,4] ="Emergency";                                    // Transit, Benign,
Dynamic, Strike, Emergency
```

```
Module_Time[2,4]	=Distributions.Triangular(20, 15,	25);
		// If 0 then default will be used
Aircraft_Module[2,5] ="Transit";				// Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,5]	=Distributions.Triangular(90, 75,	125);
		// If 0 then default will be used
Aircraft_Module[2,6] ="Losing Changeover";				// Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[2,6]	=0;					// If 0 then default
will be used
Aircraft_Module[2,7] ="Losing Changeover";				// Transit or
Benign Only
Module_Time[2,7]	=10;					// If 0 then default
will be used
Aircraft_Module[2,8] ="Losing Changeover";	// Losing Changeover or Losing
Handover Only
Module_Time[2,8]	=0;					// If 0 then default
will be used
------------------Run 6--------------------------------------------------------------------------------
----------
Aircraft_Used=3;		//Number of Aircraft to Release
Model.PrintOutput("Number of Aircraft Released: " + Aircraft_Used);
//////////////////////////////////Aircraft 1//////////////////////////////
Start_Time[1]=0;					//Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[1,1] ="Changeover";			// Changeover or Handover
Only
Module_Time[1,1]	=0;					// If 0 then default
will be used
Aircraft_Module[1,2] ="Transit";				// Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,2]	=Distributions.Triangular(30, 15,	40);
		// If 0 then default will be used
Aircraft_Module[1,3] ="Benign";				// Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,3]	=Distributions.Triangular(30, 15,	40);
		// If 0 then default will be used
Aircraft_Module[1,4] ="Dynamic";				// Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,4]	=Distributions.Triangular(30, 15,	40);
		// If 0 then default will be used
```

```
Aircraft_Module[1,5] ="Benign";                          // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,5]    =Distributions.Triangular(60, 48,     75);
                    // If 0 then default will be used
Aircraft_Module[1,6] ="Losing Changeover";                              // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[1,6]    =0;                                          // If 0 then default
will be used
Aircraft_Module[1,7] ="Benign";                          // Transit or Benign Only
Module_Time[1,7]    =120;                                        // If 0 then default
will be used
Aircraft_Module[1,8] ="Losing Changeover";        // Losing Changeover or Losing
Handover Only
Module_Time[1,8]    =10;                                         // If 0 then default
will be used
///////////////////////////////////Aircraft 2//////////////////////////////
Start_Time[2]=0;                                         //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[2,1] ="Changeover";                      // Changeover or Handover
Only
Module_Time[2,1]    =0;                                          // If 0 then default
will be used
Aircraft_Module[2,2] ="Transit";                         // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,2]    =Distributions.Triangular(120,        100,    130);
                              // If 0 then default will be used
Aircraft_Module[2,3] ="Benign";                          // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,3]    =Distributions.Triangular(30, 15,     40);
                    // If 0 then default will be used
Aircraft_Module[2,4] ="Losing Changeover";                              // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[2,4]    =0;                                          // If 0 then default
will be used
Aircraft_Module[2,5] ="Emergency";                               // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[2,5]    =120;                                        // If 0 then default
will be used
Aircraft_Module[2,6] ="Benign";                          // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,6]    =120;                                        // If 0 then default
will be used
```

```
Aircraft_Module[2,7] ="Benign";                    // Transit or Benign Only
Module_Time[2,7]     =120;                               // If 0 then default
will be used
Aircraft_Module[2,8] ="Losing Changeover";     // Losing Changeover or Losing
Handover Only
Module_Time[2,8]     =10;                                // If 0 then default
will be used
/////////////////////////////////Aircraft 3////////////////////////////
Start_Time[3]=0;                                   //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[3,1] ="Changeover";                // Changeover or Handover
Only
Module_Time[3,1]     =0;                                 // If 0 then default
will be used
Aircraft_Module[3,2] ="Transit";                   // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,2]     =Distributions.Triangular(90, 80,      110);
                  // If 0 then default will be used
Aircraft_Module[3,3] ="Emergency";                       // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[3,3]     =Distributions.Triangular(30, 15,      40);
                  // If 0 then default will be used
Aircraft_Module[3,4] ="Benign";                    // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,4]     =Distributions.Triangular(30, 15,      40);
                  // If 0 then default will be used
Aircraft_Module[3,5] ="Losing Changeover";                     // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[3,5]     =0;                                 // If 0 then default
will be used
Aircraft_Module[3,6] ="Dynamic";                   // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,6]     =120;                               // If 0 then default
will be used
Aircraft_Module[3,7] ="Benign";                    // Transit or Benign Only
Module_Time[3,7]     =120;                               // If 0 then default
will be used
Aircraft_Module[3,8] ="Losing Changeover";     // Losing Changeover or Losing
Handover Only
Module_Time[3,8]     =10;                                // If 0 then default
will be used
```

```
------------------Run 7---------------------------------------------------------------------------------
----------
Aircraft_Used=3;                //Number of Aircraft to Release
Model.PrintOutput("Number of Aircraft Released: " + Aircraft_Used);
//////////////////////////////////Aircraft 1/////////////////////////////
Start_Time[1]=0;                                        //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[1,1] ="Changeover";                     // Changeover or Handover
Only
Module_Time[1,1]     =0;                                        // If 0 then default
will be used
Aircraft_Module[1,2] ="Benign";                         // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,2]     =Distributions.Triangular(60, 48,      75);
                // If 0 then default will be used
Aircraft_Module[1,3] ="Transit";                        // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,3]     =Distributions.Triangular(30, 15,      40);
                // If 0 then default will be used
Aircraft_Module[1,4] ="Benign";                         // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,4]     =Distributions.Triangular(30, 15,      40);
                // If 0 then default will be used
Aircraft_Module[1,5] ="Transit";                        // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,5]     =Distributions.Triangular(30, 15,      40);
                // If 0 then default will be used
Aircraft_Module[1,6] ="Losing Changeover";                              // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[1,6]     =0;                                        // If 0 then default
will be used
Aircraft_Module[1,7] ="Benign";                         // Transit or Benign Only
Module_Time[1,7]     =120;                                      // If 0 then default
will be used
Aircraft_Module[1,8] ="Losing Changeover";          // Losing Changeover or Losing
Handover Only
Module_Time[1,8]     =10;                                       // If 0 then default
will be used
//////////////////////////////////Aircraft 2/////////////////////////////
Start_Time[2]=0;                                        //Use in Aircraft Control to
release entities into Aircraft functions
```

```
Aircraft_Module[2,1] ="Changeover";                    // Changeover or Handover
Only
Module_Time[2,1]     =0;                                    // If 0 then default
will be used
Aircraft_Module[2,2] ="Benign";                        // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,2]     =Distributions.Triangular(90, 80,      110);
                // If 0 then default will be used
Aircraft_Module[2,3] ="Emergency";                          // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[2,3]     =Distributions.Triangular(30, 15,      40);
                // If 0 then default will be used
Aircraft_Module[2,4] ="Benign";                        // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,4]     =Distributions.Triangular(30, 15,      40);
                // If 0 then default will be used
Aircraft_Module[2,5] ="Losing Changeover";                        // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[2,5]     =0;                                    // If 0 then default
will be used
Aircraft_Module[2,6] ="Benign";                        // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,6]     =120;                                  // If 0 then default
will be used
Aircraft_Module[2,7] ="Benign";                        // Transit or Benign Only
Module_Time[2,7]     =120;                                  // If 0 then default
will be used
Aircraft_Module[2,8] ="Losing Changeover";        // Losing Changeover or Losing
Handover Only
Module_Time[2,8]     =10;                                   // If 0 then default
will be used
//////////////////////////////////Aircraft 3/////////////////////////////
Start_Time[3]=0;                                       //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[3,1] ="Changeover";                    // Changeover or Handover
Only
Module_Time[3,1]     =0;                                    // If 0 then default
will be used
Aircraft_Module[3,2] ="Benign";                        // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,2]     =Distributions.Triangular(30, 15,      40);
                // If 0 then default will be used
```

```
Aircraft_Module[3,3] ="Dynamic";                          // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,3]     =Distributions.Triangular(60, 48,       75);
                     // If 0 then default will be used
Aircraft_Module[3,4] ="Benign";                           // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,4]     =Distributions.Triangular(60, 48,       75);
                     // If 0 then default will be used
Aircraft_Module[3,5] ="Losing Changeover";                               // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[3,5]     =0;                                         // If 0 then default
will be used
Aircraft_Module[3,6] ="Emergency";                               // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[3,6]     =120;                                       // If 0 then default
will be used
Aircraft_Module[3,7] ="Transit";                          // Transit or Benign Only
Module_Time[3,7]     =120;                                       // If 0 then default
will be used
Aircraft_Module[3,8] ="Losing Changeover";         // Losing Changeover or Losing
Handover Only
Module_Time[3,8]     =10;                                        // If 0 then default
will be used

------------------Run 8---------------------------------------------------------------------------------
----------
Aircraft_Used=4;                  //Number of Aircraft to Release
Model.PrintOutput("Number of Aircraft Released: " + Aircraft_Used);
//////////////////////////////////Aircraft 1/////////////////////////////
Start_Time[1]=0;                                          //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[1,1] ="Changeover";                       // Changeover or Handover
Only
Module_Time[1,1]     =0;                                         // If 0 then default
will be used
Aircraft_Module[1,2] ="Transit";                          // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,2]     =Distributions.Triangular(145,          130,     165); // If 0 then
default will be used
Aircraft_Module[1,3] ="Losing Changeover";                               // Transit,
Benign, Dynamic, Strike, Emergency
```

```
Module_Time[1,3]     =0;                                          // If 0 then default
will be used
Aircraft_Module[1,4] ="Transit";                          // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,4]     =120;                                        // If 0 then default
will be used
Aircraft_Module[1,5] ="Transit";                          // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,5]     =120;                                        // If 0 then default
will be used
Aircraft_Module[1,6] ="Transit";                          // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,6]     =120;                                        // If 0 then default
will be used
Aircraft_Module[1,7] ="Transit";                          // Transit or Benign Only
Module_Time[1,7]     =120;                                        // If 0 then default
will be used
Aircraft_Module[1,8] ="Losing Changeover";         // Losing Changeover or Losing
Handover Only
Module_Time[1,8]     =10;                                         // If 0 then default
will be used
//////////////////////////////////Aircraft 2/////////////////////////////
Start_Time[2]=0;                                          //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[2,1] ="Changeover";                       // Changeover or Handover
Only
Module_Time[2,1]     =0;                                          // If 0 then default
will be used
Aircraft_Module[2,2] ="Transit";                          // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,2]     =Distributions.Triangular(145,       130,     165);
                     // If 0 then default will be used
Aircraft_Module[2,3] ="Losing Changeover";                               // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[2,3]     =0;                                          // If 0 then default
will be used
Aircraft_Module[2,4] ="Transit";                          // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,4]     =120;                                        // If 0 then default
will be used
Aircraft_Module[2,5] ="Transit";                          // Transit, Benign, Dynamic,
Strike, Emergency
```

```
Module_Time[2,5]     =120;                                          // If 0 then default
will be used
Aircraft_Module[2,6] ="Transit";                              // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,6]     =120;                                          // If 0 then default
will be used
Aircraft_Module[2,7] ="Transit";                              // Transit or Benign Only
Module_Time[2,7]     =120;                                          // If 0 then default
will be used
Aircraft_Module[2,8] ="Losing Changeover";         // Losing Changeover or Losing
Handover Only
Module_Time[2,8]     =10;                                           // If 0 then default
will be used
//////////////////////////////////Aircraft 3/////////////////////////////
Start_Time[3]=0;                                                    //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[3,1] ="Changeover";                       // Changeover or Handover
Only
Module_Time[3,1]     =0;                                            // If 0 then default
will be used
Aircraft_Module[3,2] ="Transit";                              // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,2]     =Distributions.Triangular(90, 80,      110);
                     // If 0 then default will be used
Aircraft_Module[3,3] ="Emergency";                                  // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[3,3]     =Distributions.Triangular(30, 15,      40);
                // If 0 then default will be used
Aircraft_Module[3,4] ="Transit";                              // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,4]     =Distributions.Triangular(30, 15,      40);
                // If 0 then default will be used
Aircraft_Module[3,5] ="Losing Changeover";                              // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[3,5]     =0;                                            // If 0 then default
will be used
Aircraft_Module[3,6] ="Transit";                              // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,6]     =120;                                          // If 0 then default
will be used
Aircraft_Module[3,7] ="Transit";                              // Transit or Benign Only
```

```
Module_Time[3,7]     =120;                                          // If 0 then default
will be used
Aircraft_Module[3,8] ="Losing Changeover";         // Losing Changeover or Losing
Handover Only
Module_Time[3,8]     =10;                                           // If 0 then default
will be used
/////////////////////////////////Aircraft 4/////////////////////////////
Start_Time[4]=0;                                              //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[4,1] ="Changeover";                           // Changeover or Handover
Only
Module_Time[4,1]     =0;                                            // If 0 then default
will be used
Aircraft_Module[4,2] ="Benign";                               // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[4,2]     =Distributions.Triangular(30, 15,      40);
                     // If 0 then default will be used
Aircraft_Module[4,3] ="Dynamic";                              // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[4,3]     =Distributions.Triangular(30, 15,      40);
                     // If 0 then default will be used
Aircraft_Module[4,4] ="Benign";                               // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[4,4]     =Distributions.Triangular(60, 48,      75);
                     // If 0 then default will be used
Aircraft_Module[4,5] ="Transit";                              // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[4,5]     =Distributions.Triangular(30, 15,      40);
                     // If 0 then default will be used
Aircraft_Module[4,6] ="Losing Changeover";                                // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[4,6]     =0;                                            // If 0 then default
will be used
Aircraft_Module[4,7] ="Transit";                              // Transit or Benign Only
Module_Time[4,7]     =120;                                          // If 0 then default
will be used
Aircraft_Module[4,8] ="Losing Changeover";         // Losing Changeover or Losing
Handover Only
Module_Time[4,8]     =10;                                           // If 0 then default
will be used
------------------Run 9--------------------------------------------------------------------------------
----------
```

```
Aircraft_Used=4;                    //Number of Aircraft to Release
Model.PrintOutput("Number of Aircraft Released: " + Aircraft_Used);
//////////////////////////////////Aircraft 1/////////////////////////////
Start_Time[1]=0;                                            //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[1,1] ="Changeover";                         // Changeover or Handover
Only
Module_Time[1,1]     =0;                                         // If 0 then default
will be used
Aircraft_Module[1,2] ="Transit";                            // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,2]     =Distributions.Triangular(145,         130,     165); // If 0 then
default will be used
Aircraft_Module[1,3] ="Losing Changeover";                            // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[1,3]     =0;                                         // If 0 then default
will be used
Aircraft_Module[1,4] ="Transit";                            // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,4]     =120;                                       // If 0 then default
will be used
Aircraft_Module[1,5] ="Transit";                            // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,5]     =120;                                       // If 0 then default
will be used
Aircraft_Module[1,6] ="Transit";                            // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,6]     =120;                                       // If 0 then default
will be used
Aircraft_Module[1,7] ="Transit";                            // Transit or Benign Only
Module_Time[1,7]     =120;                                       // If 0 then default
will be used
Aircraft_Module[1,8] ="Losing Handover";  // Losing Changeover or Losing Handover
Only
Module_Time[1,8]     =10;                                        // If 0 then default
will be used
//////////////////////////////////Aircraft 2/////////////////////////////
Start_Time[2]=0;                                            //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[2,1] ="Changeover";                         // Changeover or Handover
Only
```

```
Module_Time[2,1]     =0;                                        // If 0 then default
will be used
Aircraft_Module[2,2] ="Benign";                        // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,2]     =Distributions.Triangular(30, 15,      75);
                    // If 0 then default will be used
Aircraft_Module[2,3] ="Benign";                        // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,3]     =Distributions.Triangular(30, 15,      40);
                    // If 0 then default will be used
Aircraft_Module[2,4] ="Dynamic";                       // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,4]     =Distributions.Triangular(30, 15,      40);
                    // If 0 then default will be used
Aircraft_Module[2,5] ="Benign";                        // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,5]     =Distributions.Triangular(30, 15,      40);
                    // If 0 then default will be used
Aircraft_Module[2,6] ="Transit";                       // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,6]     =Distributions.Triangular(60, 48,      75);
                    // If 0 then default will be used
Aircraft_Module[2,7] ="Losing Changeover";                          // Transit or
Benign Only
Module_Time[2,7]     =120;                                      // If 0 then default
will be used
Aircraft_Module[2,8] ="Losing Changeover";       // Losing Changeover or Losing
Handover Only
Module_Time[2,8]     =10;                                       // If 0 then default
will be used
//////////////////////////////////Aircraft 3/////////////////////////////
Start_Time[3]=0;                                       //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[3,1] ="Changeover";                    // Changeover or Handover
Only
Module_Time[3,1]     =0;                                        // If 0 then default
will be used
Aircraft_Module[3,2] ="Benign";                        // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,2]     =Distributions.Triangular(145,        130,     165);
                            // If 0 then default will be used
```

```
Aircraft_Module[3,3] ="Losing Changeover";                                    // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[3,3]     =0;                                                      // If 0 then default
will be used
Aircraft_Module[3,4] ="Dynamic";                                   // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,4]     =120;                                                    // If 0 then default
will be used
Aircraft_Module[3,5] ="Emergency";                                            // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[3,5]     =120;                                                    // If 0 then default
will be used
Aircraft_Module[3,6] ="Emergency";                                            // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[3,6]     =120;                                                    // If 0 then default
will be used
Aircraft_Module[3,7] ="Transit";                                   // Transit or Benign Only
Module_Time[3,7]     =120;                                                    // If 0 then default
will be used
Aircraft_Module[3,8] ="Losing Changeover";          // Losing Changeover or Losing
Handover Only
Module_Time[3,8]     =10;                                                     // If 0 then default
will be used
//////////////////////////////////Aircraft 4/////////////////////////////
Start_Time[4]=0;                                                   //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[4,1] ="Changeover";                                // Changeover or Handover
Only
Module_Time[4,1]     =0;                                                      // If 0 then default
will be used
Aircraft_Module[4,2] ="Benign";                                    // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[4,2]     =Distributions.Triangular(30, 15,      40);
                    // If 0 then default will be used
Aircraft_Module[4,3] ="Emergency";                                            // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[4,3]     =Distributions.Triangular(30, 15,      40);
                    // If 0 then default will be used
Aircraft_Module[4,4] ="Benign";                                    // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[4,4]     =Distributions.Triangular(60, 48,      75);
                    // If 0 then default will be used
```

```
Aircraft_Module[4,5] ="Dynamic";                              // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[4,5]     =Distributions.Triangular(30, 15,       40);
                // If 0 then default will be used
Aircraft_Module[4,6] ="Benign";                               // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[4,6]     =Distributions.Triangular(30, 15,       40);
                // If 0 then default will be used
Aircraft_Module[4,7] ="Losing Changeover";                                   // Transit or
Benign Only
Module_Time[4,7]     =0;                                              // If 0 then default
will be used
Aircraft_Module[4,8] ="Losing Changeover";        // Losing Changeover or Losing
Handover Only
Module_Time[4,8]     =10;                                             // If 0 then default
will be used
------------------Run 10---------------------------------------------------------------------------------
-----------
Aircraft_Used=4;                //Number of Aircraft to Release
Model.PrintOutput("Number of Aircraft Released: " + Aircraft_Used);
//////////////////////////////////Aircraft 1/////////////////////////////
Start_Time[1]=0;                                              //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[1,1] ="Changeover";                           // Changeover or Handover
Only
Module_Time[1,1]     =0;                                              // If 0 then default
will be used
Aircraft_Module[1,2] ="Benign";                               // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,2]     =Distributions.Triangular(30, 15,       40);
                // If 0 then default will be used
Aircraft_Module[1,3] ="Emergency";                                    // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[1,3]     =Distributions.Triangular(30, 15,       40);
                // If 0 then default will be used
Aircraft_Module[1,4] ="Benign";                               // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,4]     =120;                                            // If 0 then default
will be used
Aircraft_Module[1,5] ="Losing Changeover";                                   // Transit,
Benign, Dynamic, Strike, Emergency
```

```
Module_Time[1,5]     =0;                                        // If 0 then default
will be used
Aircraft_Module[1,6] ="Transit";                         // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[1,6]     =120;                                      // If 0 then default
will be used
Aircraft_Module[1,7] ="Transit";                         // Transit or Benign Only
Module_Time[1,7]     =120;                                      // If 0 then default
will be used
Aircraft_Module[1,8] ="Losing Changeover";        // Losing Changeover or Losing
Handover Only
Module_Time[1,8]     =10;                                       // If 0 then default
will be used
/////////////////////////////////Aircraft 2/////////////////////////////
Start_Time[2]=0;                                         //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[2,1] ="Changeover";                      // Changeover or Handover
Only
Module_Time[2,1]     =0;                                        // If 0 then default
will be used
Aircraft_Module[2,2] ="Benign";                          // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[2,2]     =Distributions.Triangular(145,      130,    165);
                     // If 0 then default will be used
Aircraft_Module[2,3] ="Losing Changeover";                              // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[2,3]     =0;                                        // If 0 then default
will be used
Aircraft_Module[2,4] ="Losing Changeover";                              // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[2,4]     =0;                                        // If 0 then default
will be used
Aircraft_Module[2,5] ="Emergency";                              // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[2,5]     =120;                                      // If 0 then default
will be used
Aircraft_Module[2,6] ="Emergency";                              // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[2,6]     =120;                                      // If 0 then default
will be used
Aircraft_Module[2,7] ="Transit";                         // Transit or Benign Only
```

```
Module_Time[2,7]     =120;                                          // If 0 then default
will be used
Aircraft_Module[2,8] ="Losing Changeover";          // Losing Changeover or Losing
Handover Only
Module_Time[2,8]     =10;                                           // If 0 then default
will be used
///////////////////////////////////Aircraft 3//////////////////////////////
Start_Time[3]=0;                                         //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[3,1] ="Changeover";                      // Changeover or Handover
Only
Module_Time[3,1]     =0;                                            // If 0 then default
will be used
Aircraft_Module[3,2] ="Benign";                          // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,2]     =Distributions.Triangular(145,      130,     165);
                     // If 0 then default will be used
Aircraft_Module[3,3] ="Losing Changeover";                                // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[3,3]     =0;                                            // If 0 then default
will be used
Aircraft_Module[3,4] ="Benign";                          // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,4]     =120;                                          // If 0 then default
will be used
Aircraft_Module[3,5] ="Benign";                          // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[3,5]     =120;                                          // If 0 then default
will be used
Aircraft_Module[3,6] ="Emergency";                                  // Transit, Benign,
Dynamic, Strike, Emergency
Module_Time[3,6]     =120;                                          // If 0 then default
will be used
Aircraft_Module[3,7] ="Benign";                          // Transit or Benign Only
Module_Time[3,7]     =120;                                          // If 0 then default
will be used
Aircraft_Module[3,8] ="Losing Changeover";          // Losing Changeover or Losing
Handover Only
Module_Time[3,8]     =10;                                           // If 0 then default
will be used
///////////////////////////////////Aircraft 4//////////////////////////////
```

```
Start_Time[4]=0;                                    //Use in Aircraft Control to
release entities into Aircraft functions
Aircraft_Module[4,1] ="Changeover";                 // Changeover or Handover
Only
Module_Time[4,1]     =0;                                    // If 0 then default
will be used
Aircraft_Module[4,2] ="Benign";                     // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[4,2]     =Distributions.Triangular(60, 48,      75);
                    // If 0 then default will be used
Aircraft_Module[4,3] ="Dynamic";                    // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[4,3]     =Distributions.Triangular(30, 15,      40);
                    // If 0 then default will be used
Aircraft_Module[4,4] ="Benign";                     // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[4,4]     =Distributions.Triangular(30, 15,      40);
                    // If 0 then default will be used
Aircraft_Module[4,5] ="Losing Changeover";                          // Transit,
Benign, Dynamic, Strike, Emergency
Module_Time[4,5]     =0;                                    // If 0 then default
will be used
Aircraft_Module[4,6] ="Benign";                     // Transit, Benign, Dynamic,
Strike, Emergency
Module_Time[4,6]     =120;                                  // If 0 then default
will be used
Aircraft_Module[4,7] ="Benign";                     // Transit or Benign Only
Module_Time[4,7]     =120;                                  // If 0 then default
will be used
Aircraft_Module[4,8] ="Losing Changeover";      // Losing Changeover or Losing
Handover Only
Module_Time[4,8]     =10;                                   // If 0 then default
will be used
```

Bibliography

Airforce-Technology.com. (2011). *Predator RQ-1/MQ-1/MQ-9 reaper - unmanned aerial vehicle (UAV), USA.* Retrieved February 9, 2011, from http://www.airforce-technology.com/projects/predator/

Bagnall, T., Gosakan, M., Hart, K., Lorenzen, C., Mui, R., Plott, B., et al. (2010). *MQ-1 predator crew workload analysis* No. DTIC MSIAC N61339-03-D-0300 DO 0219). Alion Science and Technology, MA&D Operation 4949 Pearl East Circle, Suite 200 Boulder, CO 80301: USAF 711th Human Performance Wing Human Performance Integration Directorate (711 HPW/HP).

Booch, G., Rumbaugh, J., & Jacobson, I. (2005). *The unified modeling language user guide* (2nd ed.). Reading, Massachusetts: Addison-Wesley.

Dietrichs, T., Griffin, R., Schuettke, A., & Slocum, M. *Integrated architecture study for weapon borne battle damage assessment system evaluation.* Capstone Graduate Research Project, AFIT/IGSE-06J. Graduate School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson Air Force Base, Ohio, June 2006

Dixon, S. R., Wickens, C. D., & Chang, D. (2005). "Mission control of multiple unmanned aerial vehicles: A workload analysis". *Human Factors, 47*(3), 479-487.

Eaton, G., Geier, P., Kalita, S., Nagy, J., Palmer, B., Phillips, A., et al. (2006). *U.S. air force unmanned aircraft systems performance analyses: Predator pilot front end analysis (FEA) report* No. SURVIAC-TR-2006-XXX). Wright-Patterson AFB, OH: Survivability/Vulnerability Information Analysis Center (SURVIAC).

Hancock, P. A., & Caird, J. K. (1993). Experimental evaluation of a model of mental workload. *Human Factors, 35*(3), 413-429.

Handley, H., & Houston, N. (2010). "NATO human view architecture and human networks". Paper presented at the *MODSIM World 2009 Conference and Expo,* 181-186.

Handly, H., & Smillie, R. *Human view dynamics using IMPRINT,* Space and Naval Warfare (SPAWAR) Systems Command under contract N66001-04-D-0005. Pacific Science & Engineering Group, San Diego, CA. (2009)

Hardman, N., Colombi, J., Jacques, D., & Miller, J. (2008). Human systems integration within the DOD architecture framework. Paper presented at the *IIE Annual Conference and Expo 2008, May 17, 2008 - May 21,* 840-845.

Hart, S. G. (1991). Pilots' workload coping strategies. *AIAA/NASA/FAA/HFS Conference on Challenges in Aviation Human Factors: The National Plan,* Tysons Corner, VA.

Huey, B., & Wickens, C. (1993). *Workload transition: Implications for individual and team performance* No. AD-A274538). Washington, D.C.: National Academy of Sciences - National Research Council, Committee on Human Factors.

McCracken, J., & Aldrich, T. (1984). *Analyses of selected LHX mission functions: Implications for operator workload and system automation goals* (No. ADA232330). Fort Rucker AL: Anacapa Sciences Inc.

McGrogan, J., & Schneider, M. (2010). *Pilot Discussions, Fargo ND, Nov 2010.* Unpublished manuscript.

McGrogan, J., & Schneider, M. (2011). *Predicting the Impact of Multi-Aircraft Control on UAS operations.* Unpublished manuscript.

Miller, N., Crowson, J., & Narkevicius, J. (2003). Human characteristics and measures in system design. In H. Booher (Ed.), *Handbook of Human Systems Integration* (pp. 713). Hoboken, NJ: Wiley-Interscience.

Mitchell, D. K. (2003). *Advanced improved performance research integration tool (IMPRINT) vetronics technology test bed model development* (No. ARL-TN-0208). Aberdeen Proving Ground, MD, USA: U.S. Army Research Laboratory.

Mitchell, D. K. (2000). *Mental workload and ARL workload modeling tools* (No. ARL-TN-161). Aberdeen Proving Ground, MD, USA: U.S. Army Research Laboratory.

Mitchell, D. K., & McDowell, K. (2008). Using modeling as a lens to focus testing. Paper presented at the *2008 International Symposium on Collaborative Technologies and Systems, CTS'08, may 19, 2008 - may 23,* 477-482. Retrieved from http://dx.doi.org/10.1109/CTS.2008.4543967

Mitchell, D. K., & Samms, C. (2009). Workload warriors: Lessons learned from a decade of mental workload prediction using human performance modeling. Paper presented at the *53rd Human Factors and Ergonomics Society Annual Meeting 2009, HFES 2009, October 19, 2009 - October 23, , 2* 819-823.

MITRE. (2009). *Air force unmanned aircraft systems unconstrained architectures,* USAF.

Modeling and Simulation Coordination Office (M&S CO). (2006). *Verification, validation, and accreditation recommended practices guide.* Retrieved January, 19, 2011, from http://vva.msco.mil/

Parasuraman, R., & Rovira, E. (2005). *Workload modeling and workload management: Recent theoretical developments* No. ARL-CR-0562). ABERDEEN PROVING GROUND MD: U.S. Army Research Laboratory's Human Research and Engineering Directorate.

Pomranky, R., & Wojciechowski, J. (2007). *Determination of mental workload during operation of multiple unmanned systems* (Technical Report No. ARL-TR-4309). Aberdeen Proving Ground, MD, USA: U.S. Army Research Laboratory.

Seibert, M., Stryker, A., Ward, J., & Wellbaum, C. (2010). *System Analysis and Prototyping for Single Operator Management of Multiple Unmanned Aerial Vehicles Operating Beyond Line of Sight.* M.S. Thesis, AFIT/IGSE-06J. Graduate School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson Air Force Base, Ohio, March 2010.

Stanton, N., Salmon, P., Walker, G., Baber, C., & Jenkins, D. (2005). *Human factors methods: A practical guide for engineering and design*. Burlington, VT, USA: Ashgate.

Systems Engineering & Assessment Ltd. (2009). *The human view handbook for MODAF*. MoD HFI DTC, Bristol, UK

USAF. (2009). *United states air force unmanned aircraft systems flight plan 2009-2047.* Washington DC: Headquarters, United States Air Force.

USAF. (2010). *MQ-1B predator.* Retrieved 8/27, 2010, from http://www.af.mil/information/factsheets/factsheet.asp?fsID=122

USAF, & USA. (2009). Army-Air Force Theater UAS enabling concept. HQ ACC/A3YU Langley AFB, VA and TRADOC TCM UAS Fort Rucker, AL

Wang, R., & Dagli, C. H. (2008). Approach to discrete events system modeling using SysML in conjunction with colored petri net.

Wickens, C. (2003). *Introduction to human factors engineering* (2nd ed.) Prentice Hall.

Wickens, C. D. (2008). "Multiple resources and mental workload. (cover story)". *Human Factors, 50*(3), 449-455. doi:10.1518/001872008X288394

Wong, D. T. (2010). "Validating human performance models of the future orion crew exploration vehicle". *54th Annual Meeting of the Human Factors and Ergonomics Society 2010,* San Francisco, CA, United States.

Yerkes, R. (1908). "The relation of strength of stimulus to rapidity of habit-formation". *The Journal of Comparative Neurology and Psychology, 18*(5), 459-482. Retrieved from WOS database.

Young, M. S., & Stanton, N. A. (2001). "Mental workload: Theory, measurement, and application". In W. Karwowski (Ed.), *International encyclopediaof ergonomics and human factors: Volume 1* (pp. 507-509). London: Taylor & Francis.

Young, M. S., & Stanton, N. A. (2002). "Malleable attentional resources theory: A new explanation for theeffects of mental underload on performance". *Human Factors, 44*(3), 365-375.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, 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 the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to an penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
24-03-2011	Master's Thesis	March 2010 – March 2011

TITLE AND SUBTITLE	
Architecture Based Workload Analysis of UAS Multi-Aircraft Control: Implications of Implementation on MQ-1B Predator	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	**5d. PROJECT NUMBER** JON 305
McGrogan, Jason, D., Major, USAF Schneider, Michael, F., GS11, USAF	**5e. TASK NUMBER**
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)	**8. PERFORMING ORGANIZATION REPORT NUMBER**
Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765	AFIT/GSE/ENV/11-M02
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSOR/MONITOR'S ACRONYM(S)** ASC/WIIM
Maj Michael Alexander Medium Altitude UAS Division 2640 Loop Road West Wright-Patterson AFB, OH 45433-7106 (937)255-4640 michael.alexander@wpafb.af.mil	**11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
An increased demand for use of Unmanned Aircraft Systems (UASs) without commensurate increases in pilot manpower has prompted proposals for simultaneous control of multiple aircraft by a single pilot or Multi-Aircraft Control (MAC). To understand the potential effects of MAC, an IMPRINT Pro, Multi-Resource Theory, pilot workload model was developed from pedigreed system architecture. Feedback from active UAS pilots was used to validate the model and establish a workload saturation threshold value of 60, above which pilots may experience performance degradation over extended periods of time. The model predicts that pilots experience low workload when operating one or two UASs during benign operations, and operate 91% of the time below a workload of 25 without saturation. However, conflict from multi-task overlap builds rapidly when the pilot is required to operate three or more aircraft. The percentage of time over the saturation threshold increases to 21% with four aircraft under benign operating conditions. When dynamic events are introduced the workload becomes unmanageable, with estimates regularly over 100 due to multi-task overlap and communication activities. The analysis indicates the need for techniques and technology to reduce task and communications demands on UAS pilots to effectively implement MAC.

15. SUBJECT TERMS
UAS, Workload, Multi-Aircraft Control, Modeling

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			John Colombi, AFIT/ENV, ADVISOR
U	U	U	UU	230	**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-6565, ext 3347 (john.colombi@afit.edu)

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18