The Pennsylvania State University
The Graduate School

ALLOCATION OF AUTHORITY AND REPONSIBILITY IN ADVANCED AIR MOBILITY OPERATIONS FOR PERFORMANCE-LIMITED AGENTS

A Thesis in

Aerospace Engineering

by

Ishan Ghimire

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 2024

The thesis of Ishan Ghimire was reviewed and approved by the following:

Amy Pritchett

Professor and Head of Aerospace Engineering

Jack Langelaan

Professor and Director of Graduate Studies

Eric Greenwood

Assistant Professor of Aerospace Engineering

Abstract

The operation of human-autonomy teams in Advanced Air Mobility (AAM) depends on the capabilities of each agent, what information they need to know and communicate, and the organizational structure that outlines what actions they perform and what outcomes they are responsible for. This thesis examines the Concept of Operations for Urban Air Mobility Maturity Level (UML) 4, hereafter referred to as CONOPS, to highlight the differences and commonalities in two principles for allocating authority and responsibility for all the actions needed to perform the CONOPS: one based by grouping the roles the different agents perform, and one based on the location of where the actions take place. Using Work Models that Compute (WMC), simulations of a case study involving multiple vehicles performing multiple missions examined the demands placed on the agents with each allocation.

Additionally, to represent the behavior of human agents more accurately within the teams, this thesis models agents as having limits. Some simulations examined the impact of limiting all agents in the number of actions they can simultaneously perform. In other simulations the human agents were represented as memory limited agents, with a maximum constant value on the number of discrete pieces of information that can be held in their working memory. Since all actions do not require the same amount of information to be held in short-term memory, this creates a dynamic limit to the number of actions simultaneously performed.

In considering the agents' taskload, there is a strong need to also model the monitoring actions done by the agents, where one agent is responsible for the outcomes of the actions executed by a different agent. This monitoring work significantly impacts the load on the agent, resulting in acute and chronic saturation of the agents' capabilities, and delays in safety critical monitoring.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I am very thankful for the collaborative efforts of Dr. Cody Fleming and Minghui Sun, and Natasha Neogi and Jon Holbrook, our technical monitors at NASA, as well as Dr. Martijn IJtsma for hosting the WMC repository. This work is sponsored by NASA Grant 80NSSC19K1702 from the Aeronautics Research Mission Directorate's System-Wide Safety project. The findings and conclusions in this thesis do not necessarily reflect the view of the funding agency.

1 Introduction

1.1 Allocation of Authority and Responsibility in Novel Concepts of Operation

Advanced Air Mobility (AAM) represents a vision for the future of air transportation, characterized by the integration of aerial vehicles such as drones, air taxis, and autonomous flying vehicles. However, this shift requires novel Concepts of Operation that support both the changes in performance associated with these new vehicles, and the new types of missions they enable.

An important aspect of these new concepts is how they allocate actions and responsibilities among a diverse array of agents. Within AAM operations, some actions and decision-making processes may be delegated to human operators, while others may be entrusted to autonomous agents. Effective allocation requires models and analytical tools to discern the demands placed on each agent by potential allocations. Moreover, these models must capture the properties of the actions undertaken in this dynamic environment where multiple vehicles are concurrently engaged in various missions.

This allocation must account for not only the allocation of authority to execute each action, but also responsibility for each action's outcome. This requires that the responsible agent monitor to ensure that the outcome aligns with safety and operational standards. These monitoring behaviors are instrumental in ensuring the safe and reliable operation of AAM systems. However, the impact of monitoring has not been measured.

Furthermore, the dynamics of teamwork and the distribution of responsibilities among agents can yield emergent behaviors that have not been fully analyzed. The interplay between these agents may result in novel patterns and interactions that are difficult to predict. Consequently, the ability to confidently anticipate the overall behavior of the AAM system becomes a substantial challenge that cannot be predicted analytically: Instead, simulation is needed to predict dynamic emergent effects.

The effectiveness of a Concepts of Operations is dependent on the capabilities of the agents involved. A fundamental aspect of this challenge involves the finite limits of human agents. The demands placed on human agents in AAM scenarios are diverse and dynamic, and comparing the allocation of actions to these limits is crucial.

1.2 Metrics of Allocation of Authority and Responsibility

Metrics of a Concepts of Operation's allocation of authority and responsibility can be categorized into three broad levels: first, "agent-level" metrics can be assessed from the timeline of actions each agent executes, the information needed to execute each action, and the information their actions create and update. These include:

- Agent Timeline: The experience of an agent in terms of the timeline of actions that they execute.
- Information Transfer per Agent: The quantity of information needing to be transferred to/from each agent.
- Agent Average and Max Taskload: The number of actions executed by an agent; this may be further broken down into "taskwork" actions and "monitoring" actions.

Second, mission-level metrics describe the performance and outcomes of a mission:

- Time to Completion: The duration required to complete all actions needed to finish a mission.
- Safety Impacts: The degree to which safety is maintained or compromised during a mission.
- Delays in Critical Actions: Instances where necessary steps are delayed, with the delays potentially extending the time to completion, or resulting in unacceptably unsafe outcomes.

Third, organization-level metrics describe the overall performance and behavior of the entire Concepts of Operations. Metrics here include:

- Action Trace: A comprehensive overview of all actions carried out or monitored by all the agents.
- Aggregate Information Transfer: The aggregate of information needing to be exchanged between all the agents, indicating the efficiency of communication, agent independence and agent interconnectedness.
- Total Time to Completion of a set of missions representing, for example, an entire day's operations as conducted by multiple vehicles.

1.3 Objectives

This thesis examines the allocation of authority and responsibility among agents within novel AAM Concepts of Operations, focusing on these three objectives:

1. Investigate different principles for assigning authority to agents to execute taskwork.
2. Investigate the impact of assigning responsibility for outcomes, thereby also allocating monitoring actions to the agents.
3. Assess the impact on these allocations of potential agent limits in the number of simultaneous actions each can complete, or a more detailed model of agents' fundamental limitations in working memory.

2 Background

This thesis examines the design of Concepts of Operation, using Advanced Air Mobility as a test case. Within Concepts of Operation, this thesis specifically examines the allocation of authority and responsibility, building on similar work in function allocation. Such allocations need to consider the limits and capabilities of the agents allocated the actions; this thesis particularly focuses on the limits imposed by working memory. Likewise, such allocations need to consider the monitoring they require agents to perform in addition to their taskwork. This chapter gives some background on this topic.

2.1 Advanced Air Mobility

Advanced Air Mobility (AAM) envisions a new air transportation system enabled by increasing automation in aircraft and flight management, electric VTOL aircraft, small unmanned aircraft systems, automated passenger aircraft, and other emerging and new technologies [1]. A recent NASA report [2] outlined a range of operating concepts framed around their maturity level (UML). UML- 1 is limited to testing, certification, and operation in limited environments, while UML-6 is highly integrated AAM operations that are commonplace with thousands of simultaneous flights even in complex urban environments [2].

This thesis examines AAM at UML-4, which requires medium complexity operations with automated systems operating in collaboration with human agents. Operation at this level of frequency cannot be done without automated aircraft and flight management systems. When considering a significant number of simultaneous operations in coordination with highly automated systems, the actions required of humans are significantly different from traditional airspace operations. Moreover, as more actions become automated, human oversight becomes an integral part of AAM operations, potentially changing the distribution and nature of the actions taken by humans. The operations need to satisfy requirements regarding efficiency, safety, reliability, and affordability to be implemented, and to transition from UML-4 to UML-5 and further towards full integration. Since UML-4 is a transitory phase towards full automation and integration, it will not be fully automated internally, and will need to interface with non-autonomous systems. UML-4 relies on dedicated operation centers, referred to as aerodromes, with Providers of Services to AAM (PSU) providing air traffic management (ATM) for regular operations under rules and regulations provided by airspace-governing bodies, namely the FAA. An in-depth, but not complete or prescriptive, exploration of the interactions between requirements and regulations, the parties involved, and the actions needed for the nominal AAM operations is given in a Concept of Operations document [2].

AAM remains in the developmental stage and requires extensive research and experimentation. The system's architecture and operations need to be defined comprehensively to ensure safety, reliability, and efficiency. Furthermore, many agents – human and automated – will be needed and their authority and responsibility to be concretely defined.

2.2 Concept of Operations

A Concepts of Operations defines the goals and objectives of the policies and constraints defining the limits of operation, the participants and stakeholders and their interactions, the responsibilities and authorities delegated, and the processes that enable the operation of the system detailed sufficiently to describe the actions expected of each agent and system component comprehensively [3]. However, a Concept of Operations does not detail the implementation of the system or examine the transition from current systems to the proposed system.

A Concepts of Operations document for AAM operations creates a frame of reference for the FAA, NASA, industry, and stakeholders to describe operation in the National Airspace System (NAS). This thesis specifically examines the CONOPS document created by NASA and numerous stakeholders to provides a possible scenario for UML-4 operation, without explaining how the increased automation and integration may progress through increasing maturity levels. The document explicitly highlights its limitations in providing explicit implementation methods, detailed operational procedures, except as illustrative examples [2]. Further research is required to detail all aspects of operations in the CONOPS, and enumerate all the participants and actions, at different levels of involvement from different perspectives, with the CONOPS document facilitating collaborative research between industry, regulatory bodies, and administrators. Of note to this thesis, this CONOPS document does not yet delineate duties of, and relationships between, agents. This allocation process, in essence, serves as a critical bridge between the conceptual stage (as presented in the CONOPS) and the practical implementation of the system. The next step in the development process, therefore, involves explicitly defining the agents involved in the operation and not only their general roles, but also which actions they are authorized to execute, and which outcomes they are responsible for.

2.3 Allocation of Authority and Responsibility Within a CONOPS

This thesis focuses on the allocation of authority and responsibility for every action required in a CONOPS. "Authority" is specifically defined here as identifying which agent is tasked with executing an action. "Responsibility" is defined here as identifying which agent is legally responsible for the outcome of an action, in a legal or policy sense, regardless of which agent executed it.

The allocation of authority and responsibility as defined here builds on discussion in the literature that are often framed as "function allocation." However, function allocation is often applied to determine which functions or tasks should be performed by humans and which should be automated or delegated to machines, computers, or other automated systems. Many of the principles developed for function allocation focus on one human interacting with one machine, and thus apply constructs as "levels of automation" that do not extend to the need of a Concepts of Operations involving many, distributed agents.

Likewise, many principles for function allocation focus on how to divvy up activity for one task, or a set of closely-related tasks; a notable example here is the creation of function allocations for aircraft guidance and control, which requires the examination of the specific capabilities of auto flight and autopilot, and flight direction systems, which allocate some actions to the pilot and others to the automation – without examining the other activities needed to operate an aircraft.

Finally, most studies of function allocation only examine the allocation of authority to execute the action, but not the allocation of responsibility for its outcome. As noted by Feigh and Pritchett [4], this obscures that the allocation of responsibility is a vital attribute of safety-critical operations, and requires significant actions on the part of the responsible agent, such as monitoring. With these caveats, this section reviews the relevant literature to this thesis.

2.3.1 Factors to be Considered in Allocation Decisions

Many aspects of the Concepts of Operations need to be considered when allocating actions. Older, T[5] outlined many criteria that affect how actions are allocated. Some of the criteria include:

1. Demands: Understanding the cognitive, physical, and sensory demands placed on humans as well as those placed on machine resources is crucial in determining which tasks are best suited for human operators.

2. Workload distribution: Balancing the workload between humans and machines to avoid overloading or underloading human operators and ensuring that they remain engaged and alert but not overwhelmed during critical tasks.

3. Redundancy: The assessment of potential for errors due to machine malfunction require making allocation decisions that create redundancy.

4. Environmental constraints: The environment, consisting of existing technology, finances or organizational structure can shape how actions are allocated.

5. Human well-being; From physical safety to psychological needs such as role importance, task variety or motivation, function allocations need to maintain the well-being of humans.

2.3.2 Requirements for Effective Allocations

In their extensive review of the function allocation literature, Feigh and Pritchett [6] summarized the essential requirements of an allocation of authority and responsibility:

1. Each Agent Must Be Allocated Functions That It Is Capable of Performing: This requirement emphasizes the importance of matching functions with the capabilities of each agent. It ensures that agents are assigned tasks that are within their skill sets, expertise, and operational capabilities. Allocating functions that an agent cannot perform effectively can lead to errors, inefficiency, and potential safety risks. A thorough understanding of each agent's abilities is essential to meet this requirement.

2. Each Agent Must Be Capable of Performing Its Collective Set of Functions: Beyond individual capabilities, this requirement considers the collective set of functions assigned to an agent or group of agents. It ensures that the combined workload and responsibilities do not exceed the capacity of the agents involved. Overloading an agent or team with an excessive number or complexity of functions can result in reduced performance, fatigue, and potential task failures.

3. The Function Allocation Must Be Realizable with Reasonable Teamwork: Effective teamwork is essential for successful function allocation. This requirement underscores the need for agents to work together seamlessly and efficiently to accomplish their allocated functions. It should not rely on extraordinary levels of coordination or communication that may be impractical or burdensome. Reasonable teamwork ensures that agents can collaborate effectively without excessive cognitive or organizational overhead.

4. The Function Allocation Must Support the Dynamics of the Work: In complex, multi-agent operations, many activities are coupled and inter-dependent. This makes the effect of the allocation hard to predict; other studies proposed the WMC framework to analyze the dynamic emergent effects, discussed in the next chapter.

5. The Function Allocation Should Be the Result of Deliberate Design Decisions Allocation should not be arbitrary or ad-hoc; rather, it should be a product of deliberate design decisions. Designers and decision-makers should carefully analyze the system's requirements, agent capabilities, and operational context to determine the most suitable allocation. A systematic and well-documented approach ensures that the allocation aligns with the system's goals and objectives.

2.3.3 Measures of Allocations

Kim, Pritchett and Feigh [7] noted that no one measure of an allocation is sufficient to assess the extent it has properly considered all the factors noted earlier and meet all its requirements. Further, they demonstrated how some measures may inherently conflict with each and need to be assessed sufficiently to identify key trade-offs.

From the literature, they noted several types of measures used in assessment of allocations. Some of these measures can be assessed earlier in design; others need to be assessed human-in-the-loop studies only possible once most of the technology and procedures are in place. These categories of measures are:

1. Workload/Taskload: Workload or taskload metrics assess the level of cognitive and physical demands placed on an agent when performing allocated functions. High workload can lead to stress, fatigue, and decreased performance, while low workload may result in underutilization of human or automated capabilities.

2. Mismatches Between Responsibility and Authority: This metric examines whether the agent responsible for a particular action also has the necessary authority and control to execute that action. Mismatches can result in additional monitoring, coordination or communication when done effectively, or frustration and confusion when implemented poorly.

3. Stability of the Human’s Work Environment: The stability of an agent's work environment assesses how disruptions, environmental factors, or sudden changes in the allocation impact an agent's ability to perform allocated tasks.

4. Coherency of a Function Allocation: Coherency metrics evaluate the logical consistency of the function allocation. It assesses whether tasks assigned to various agents or groups align with the overall goals and objectives of the system and reflect a common basis for the agents' actions.

5. Interruptions: Interruption metrics gauge the frequency and impact of interruptions on agents while performing their tasks. Frequent interruptions can disrupt workflow, reduce focus, and lead to errors. The extent of the interruptions can also vary, having a markedly different impact on the experience of an agent.

6. Automation Boundary Conditions: Automation boundary conditions examine the extent to which automated systems are placed near the boundary conditions in which they are designed to operate. These measures reflect the likely robustness of the automation's ability to execute its allocated actions.

7. System Cost and Performance: Metrics related to system cost and performance assess the economic and operational implications of the allocation. This includes evaluating the cost-effectiveness of automation, resource utilization, and system performance.

8. Human's Ability to Adapt to Context: An over specified allocation can prevent a human agent from making decisions outside of the prescribed roles, even when required, especially during off-nominal operations. Also, while the allocation may be created with an expected behavior in mind, human operators can behave in different ways, which can be beneficial to the individual agent's operation, as decided by the expert agent.

2.4 Monitoring

As noted in the previous section, an allocation needs to designate not only the authority to execute each action, but also the responsibility for each action's outcome. When different agents are assigned authority and responsibility for the same action – sometimes referred to as an authority-responsibility mismatch – then the responsible agent must provide some form of oversight [8]. While this oversight can, at a high-level, involve hierarchies within the organization, dedicated systems for alerting, and other broad constructs across the operation and organization, at a per-action level this oversight starts with the responsible agent needing to monitor.

How this monitoring should be conducted has not been widely studied in the literature and is difficult to state. The monitoring may be conducted simply, such as just checking for completion, or can be much more involved, such as checking the validity of the outcome, or even greater than the work of the action itself, when not only the outcome, but the process that went into obtaining the outcome also needs to be monitored. The level of monitoring required varies based on the relationship between the monitoring agent and the agent being monitored, the nature and criticality of the action itself, and regulatory requirements and procedures that dictate monitoring [9]. Thus, the extent of the workload generated by monitoring therefore is not generalizable due to the variability in function allocations, organizational structures, regulatory constraints etc. Likewise, early in the design process, the extent of monitoring that will result from the allocation is not apparent: while the function allocation determines which actions require monitoring, when these actions will occur, how many times they will occur, and how they will interact and interleave with the taskwork is not readily apparent in the design stage.

Monitoring can have several significant impacts that need to be assessed. First, the cognitive load imposed by monitoring may occupy a significant portion of an agent's information processing capacity. Beyond the instantaneous workload created by monitoring, the inclusion of monitoring actions can change the agent's timeline of activity. While some agents may be expected to only execute a small number of taskwork actions, how these agents may need to monitor many other actions is not well understood. Moreover, for the agents that are already heavily burdened, monitoring can overload their capabilities.

When an agent is overloaded, a common tendency is to prioritize immediate taskwork actions with a tangible deadline over monitoring. Additionally, low-intensity or frequency work, especially monitoring work, is difficult for human monitors to sustain. Either way, monitoring functions as an important safety-producing behavior which, when delayed, skipped, or glossed over, can fail to prevent safety critical outcomes, especially in complex systems with multiple agents and simultaneous actions such as in AAM operations.

2.5 Agent Limits and Working Memory

The previous sections have noted the need for an allocation of authority and responsibility to account for limits in the agents' capabilities. Every agent – human or machine – has limits. Some are situation-specific, such as whether the agent has been appropriately trained, has access to sufficient information, and has sufficient control authority. Others are more intrinsic to the agents, such as a limit on the number of items that can be held in working memory by humans.

While humans can perform multitasking with a small number of actions, the limits of this ability can be exceeded in many ways. In some instances, the working memory of humans can be easily be overwhelmed, while other in applications this capacity is orders of magnitude larger [10]. One difference between these two extremes is the type of information required to be held in working memory. For example, when considering verbal information, the length of words can influence the number of pieces of information that can be stored in working memory [11]. Looking at humans as agents within an operation, a compelling limit is well established on the number of items any human can hold in working memory at any time, commonly described as "7 plus-or-minus 2" [7]. Each cognitive activity the human needs to simultaneously perform may be framed as requiring at least one element to be held in working memory. At its simplest, an action may just require the human to remember some aspect of the environment to consider, which also serves as remembering that this action needs to be completed.

More involved actions may require more information to be accessed. Humans handle this information load in several ways. First, if an action requires all available memory, they need to focus exclusively on it. This results in the management of working memory such that other actions are prioritized lower, or delayed, until the current taxing action is completed.

Second, humans may learn to “chunk” information together such that they are storing in memory only a few concepts or patterns that describe an otherwise overwhelming number of raw information elements; for example, a human may only need to recall five names or phrases, rather than the large number of digits needed to spell them out. Humans can develop heuristics to reduce the load by noting the important parts of the information they need to remember, combining information together based on similarities or by recoding information into decision trees instead of remembering every piece of information [9]. The example mentioned earlier, comparing humans’ ability to remember names/phrases vs digits, relies on this method. The sequence of binary numbers does not lend itself easily to chunking, unless the numbers are combined into base 3 or 4 [10]. The rules for the decomposition of the words into their constitutive characters are learned memory that is not stored in working memory. Experts, or people familiar with the information being considered, can extend this ability to multiple levels of chunking, creating chunks of chunks, with increasing familiarity improving the number of fundamental parts stored in working memory, while maintaining the working memory limit imposed by human brain. This is valid for structured information, such as chess players being able to recall the positions of many pieces when shown a board mid-game, but not when the pieces were positioned at random. When recalling and reconstructing the board, they can utilize their knowledge of the potential prior moves that led up to the mid-game board, resulting in their recall being much better than that of novices, or if they were given unstructured information [12].

Third, information that is not stored in working memory can be made easy to access through lists, notebooks, databases, calendars, and other external signals. This is a function of both deliberate design decisions and through the emergent behaviors of experts or people familiar with their workflow and the information required of each action [13]. This design can incorporate the relationship between different actions that are close in location or temporally or related through shared information and resources. Information that is readily observable to the human agent can offload the amount of information that needs to be held in working memory. For example, a PSU in AAM operations may only need to remember the ID of an aircraft it is monitoring. With that in one element of working memory held “in the head” the agent can then query an interface to observe the full set of information about the aircraft’s status.

2.6 Agent Based Simulation

In AAM operations, with so many actors, stakeholders and events, emergent properties are hard to predict. Higher-order effects such as nuances in the inter-leaving of agents’ actions can result in similar operations having drastically different outcomes. To understand the behavior of the overall operation, and specifically as it emerges from the actions of all the agents, the entire set of actions in the Concepts of Operations needs to be simulated.

In the design phase of Concepts of Operation, it is much more economical to simulate the operations using agent-based simulation compared to human-in-the-loop testing. Compared to equation-based models that are more suited to physical systems such as fluid or structural mechanics, agent-based models are more suited to examine behaviors of agents and technologies whose actions define the performance of the Concepts of Operations [14]. With the bottom-up approach of agent-based modeling, the behavior of the agents can be modeled, with their interactions resulting in emergent behavior that describes the outcome of the overall scenario. This approach can identify some unforeseen interactions between agents and their environment that only appear as emergent properties and are not modeled within the agent or environment itself.

3 Computational Simulation (Case Study)

To measure the impact of different allocations of authority and responsibility in AAM Concepts of Operation, this thesis computationally simulates all the actions they require. This chapter details the simulation framework and the specific case study.

3.1 Work Models that Compute

Work Models that Compute (WMC) is an agent-based simulation framework that separates the dynamics of the action being executed from the internal dynamics of the agent executing the action. This makes it possible to rapidly reassign actions to different agents, create new agents, and trigger monitoring actions whenever there is a mismatch between authority and responsibility.

The actions are modeled independently from the agents, functioning as modifiers for the environment they act upon. The actions access or (“get”) resources and change (“set”) their values. Authority and responsibility for the actions are assigned to agent models during run-time, which can execute them but may also have their own internal dynamics. The agents keep track of the resources their actions require, and thus record their information requirements overall, as well as instances where they “get” information last “set” by an action performed by a different agent (which highlights where information would need to be transferred or communicated between agents in real operations).

The simulation core, during run-time, centers on a list of all the action models. Each action reports the next time it needs to be updated to accurately reflect its dynamics. The simulation core sorts the action list by this next update time. The simulation advances by identifying the next action and passing it to the agent model authorized to execute it. The agent model may choose to immediately call the action’s method representing its activity. This causes the action to “get” resource values reflecting the aspects of the operation environment that it responds to, and then calculate and "set” the resource values reflecting the output of its activity. Each action model also reports its duration, i.e., the time the agent model will be actively working on it.

This structure allows many different types of activity to be modeled. At one extreme, the aircraft dynamics and control actions get and set the resource values reflecting vehicle motion, updated frequently at the small timestep needed to approximate continuous motion. At the other extreme, actions may represent discrete events such as authorizing a flight by setting resource variables defining status according to a schedule or when triggered by other events.

Examining the agent models in WMC, unlike most forms of agent-based simulations, the models of activity are held in the simulation’s core, and agents are only passed actions at the time the agent is asked to execute them. Each action is retained within the agent's “active actions list” for its duration as a real-time representation of the tasks currently being executed by the agent. A “perfect” agent model can be appropriate to model all or some of the agents in situations where they are assumed to be capable of executing any number of actions simultaneously. This thesis also examines the impact of agents having limits on the number of actions they can execute simultaneously, as detailed further in Sections 6.1 and 6.2. When an agent is saturated and incapable of adding another action to its active actions list, any further action passed to the agent to execute is moved to a “delay list.” Actions within the delay list have their execution times adjusted to reflect when the agent will complete any of its active actions and reevaluate its capacity. If the agent's capacity at that moment allows for the addition of the action to its active actions list, the action next in the delay list is moved to the active actions list and executed. Conversely, if the agent's capacity remains insufficient, the action continues to be delayed until such a time when the agent can accommodate it.

3.2 Monitoring in WMC

The WMC framework can be commanded to examine, during run-time, every action to be executed for an authority-responsibility mismatch. This mismatch occurs when the agent authorized to execute an action is not its designated responsible agent. In such a case, the simulation code creates a parallel “monitoring action” and passes it to the responsible agent to execute in the same way as the regular “taskwork” action.

The monitoring actions can be either “basic” or “full” representing two different types of behaviors. Basic monitoring requires the responsible agent to access the resources set by the executing agent, representing the monitoring behavior where the responsible agent verifies the outcome of the taskwork. Full monitoring takes monitoring a step further: the responsible agent not only accesses the resources set by the executing agent but also the resources the executing agent accessed(“gets”). This enhanced level of monitoring represents the monitoring of not only the execution of the action, but also the process through which said action was executed. Thus, “full” monitoring typically adds much more to the agent’s information requirements and information transfer than “basic” monitoring.

3.3 Case Study: One Day of AAM Operations

The case study examined delivery of packages to 30 locations by 5 vehicles from a hub which acted as both the starting and ending point for all missions. The variables changed between different simulation runs were the number of agents involved, the allocation of actions to the different agents, the extent of monitoring involved, and the performance limits of the agents.

In this case study, the actions are shown in Table 1, spanning all the actions required to simulate an AAM Concepts of Operations. These actions are separated into 3 work models, each responsible for a different aspect of the operations[15]. Thirty instances of the Mission work model are created, and five instances of the Vehicle work model are created in this scenario.

Table 1. Actions present in AAM operations

Command Work Model	
Assign mission	Approve performance authorization
Authorize flight	Initiate takeoff planning
Allocate takeoff pad	Approve takeoff
Issue landing clearance	Initiate arrival sequence
Confirm clear for landing	Allocate landing pad
Mission Work Model	
Request performance authorization	Flight plan
File operations plan	Receive operations plan approval
Request flight authorization	Board/process payload
Unload aircraft	
Vehicle Work Model	
Assess battery	Charge battery
Systems check	Takeoff
Flight dynamics	Manage waypoint progress
Maintenance	

The sequence of the actions for one mission is shown in Figure 1. However, since this only represents one mission, this timeline does not include actions that only occur for multiple missions, such as charging the battery between missions.

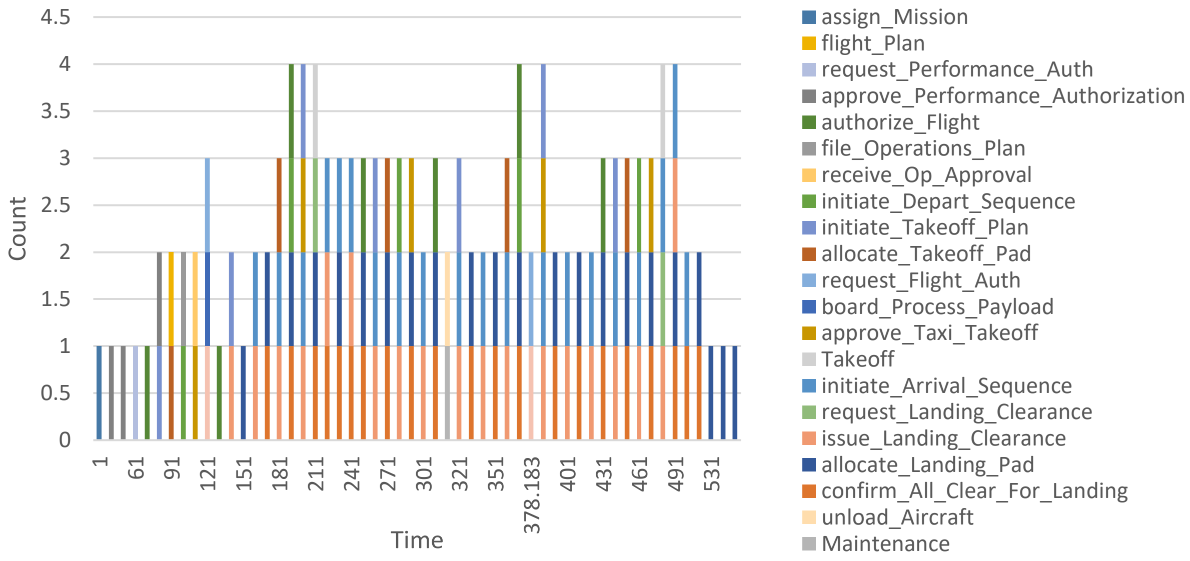


Figure 1. Actions and their time of execution for one mission

The combination of these actions results in the vehicle taking off from the central location, climbing in altitude, flying to the destination, landing and unloading the payload, taking off and climbing in altitude again before returning and finally landing. As can be seen in Figure 1, there are multiple recurring actions. These actions are modeled in the Command work model and periodically check whether current status needs their approval or decision for an upcoming state of flight. These recurring actions represent one method of coordination within the team, where actions signal availability for a subsequent action or a change in the mission status, requiring the recurring actions to examine for when they are required.

The resulting flight path of a vehicle on a sample mission is given in Figure 2 which plots the path of the vehicle in all 3 axes, and Figure 3 hat plots the altitude of the vehicle through time. It can be seen from this that one mission consists of two flights (the payload delivery flight, and then a return flight to base), and there is a period where the vehicle is on the ground at the destination. This can be due to maintenance, charging, systems check, or while the vehicle is waiting for a takeoff pad, or other coordination activities, and will not be the same for all missions.

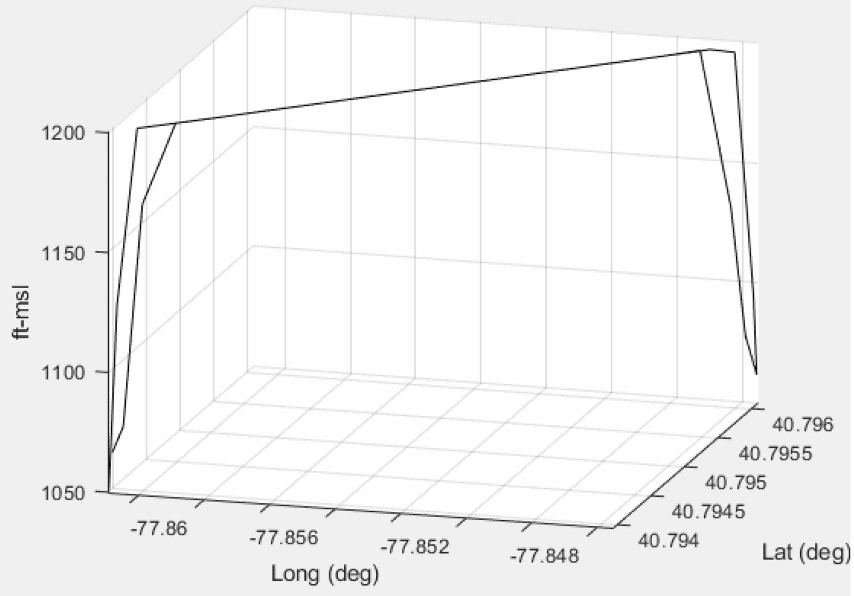


Figure 2. Flight path for UAS2 over one mission

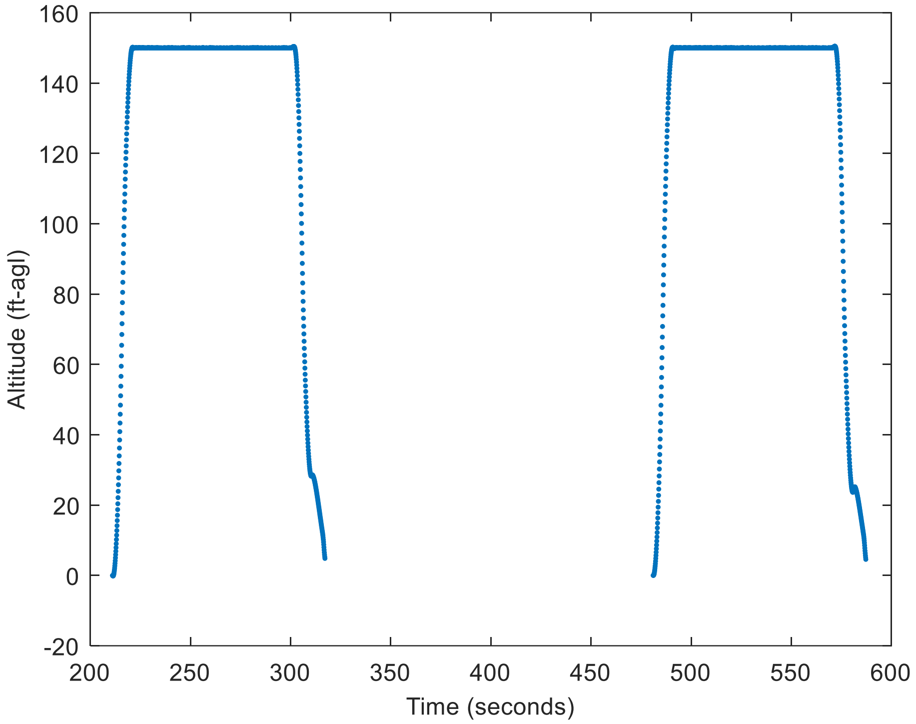
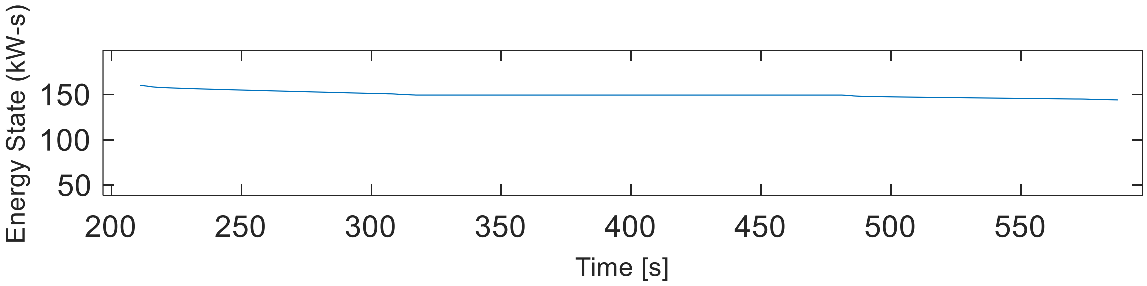
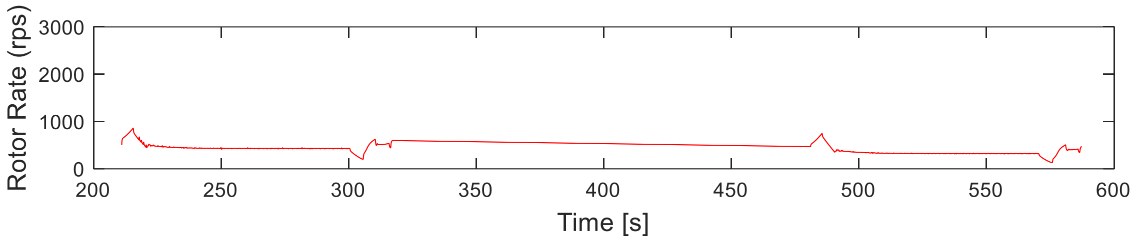


Figure 3. Altitude of UAS2 over time for one mission

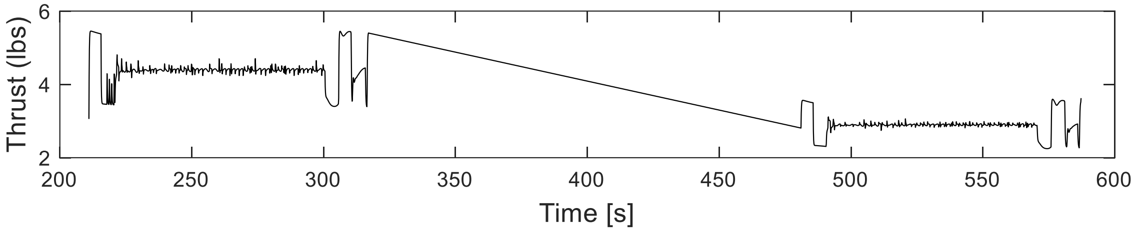
The vehicle is seen to end each flight at ~5ft above ground level. This is a simulation artifact, since altitude is only logged when the vehicle is "in-flight". Once the vehicle is horizontally +/- 5ft from the center of the landing pad and is 5ft above it, it switches to "land" action which in a single step models a reduction in power calculated to have the vehicle land smoothly. Since this does not need to be integrated, it is not logged. In order to achieve this flight profile, the vehicle controlled its rotor rate, power and thrust to execute the mission using a detailed dynamic model, as shown in Figure 4. The lower thrust during the latter portion of the operations is a result of the lighter weight of the vehicle due to the payload being unloaded. The continuous decrease in thrust is an artefact of the plotting method, which connected a high thrust value for the loaded vehicle at 317s to subsequent thrust value recorded, that for the unloaded vehicle resulting in a lower thrust at seen at 481s.



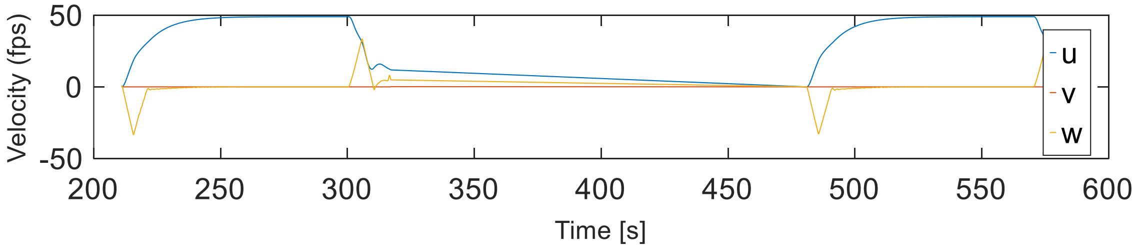
a) Energy state over time



b) Rotor rate over time



c) Thrust over time



d) Velocities in three axes over time

Figure 4. Rotor rate, thrust, velocity and power over time in one mission

4 Results: Allocation of Authority for Taskwork

The first objective of this thesis is to *investigate different principles for assigning authority to agents to execute taskwork*. This requires developing some principle by which the correct set of agents can be identified and then authority assigned to execute each action. This chapter develops two potential principles for allocation of authority within the AAM case study detailed in the previous chapter, and evaluates their impact at the agent, mission, and organization levels through WMC simulation.

4.1 Benchmark: Taskwork Required for AAM Operations

This thesis explores the relationship between the allocation of authority for taskwork and the performance of AAM operations using a case study involving 5 vehicles completing 30 total missions. The taskwork is largely independent of allocation of authority: i.e., the tasks to be performed are driven by the Concepts of Operations, regardless of which agents execute them. In this thesis's case study, for example, the flow of taskwork – if executed without delay by each agent – is shown in Figure 5. All the agents combined execute 5529 instances of taskwork actions which collectively take 13002s. These actions do not include actions for vehicle dynamics and control which occur every 0.02s. At each recorded time at least one action needs to be actively under execution, with a maximum of 17 task actions needed to be performed concurrently. Out of 5529 instances of task actions, 4978 occur in the Command work model, 326 occur in the Mission work model, and 225 occur in the Vehicle work model. Some of the Command work model's task actions recur frequently, checking status, but do not provide an outcome in 4407 instances and only provide an outcome such as an approval or clearance in 571 instances. Therefore, in total, 1132 task actions provide an outcome.

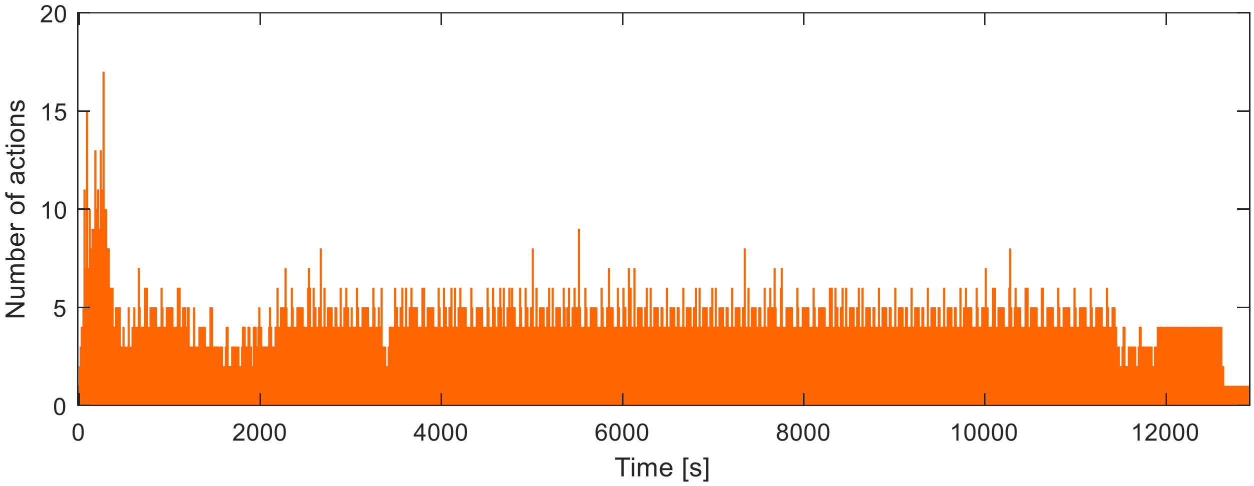


Figure 5. Count of task actions over time for cookie delivery mission scenario

4.2 Principles for Allocating Taskwork

While the NASA Concept of Operations document does not detail all the agents performing the taskwork, or the specific allocation of taskwork actions to them, it does imply some general organizing principles around **role** and **location** [2]**.** This section elaborates on these principles, detailing how each may be used to guide the detailed allocation of authority to agents.

Role-Based Allocation recognizes that different agents have unique objectives and roles within the operation. Assigning actions on these characteristics ensures that each agent's contributions align with their expertise and objectives. It assumes that actions can be executed regardless of the physical location of the agents or where the actions occur. It also takes into account the relationships they would have within and across the different organizations that make AAM operations possible. The agents and their roles were developed from the CONOPS Appendix B: Roles and Responsibilities [2].

Specifically, 6 roles are created with this principle for allocation of taskwork:

1. Provider of Services to UAM (PSU): Provides information to users to ensure shared operational status.
2. FAA: Outlines and ensures adherence to regulations in airspace.

3. GroundCrew: Performs duties on the ground with cargo (including loading/unloading), aircraft maintenance, servicing, etc.
4. AerodromeOperator: Operates aerodrome, coordinates landings and takeoffs, and communicates with other agents to coordinate AAM operations.
5. Aircraft: Captures functions performed by and inside the aircraft itself.
6. FleetOperator: Manages the coordination of all the aircraft, and their missions while meeting regulations and communicating with other agents in the AAM operations.

Location-Based Allocation, on the other hand, primarily revolves around where actions physically take place, prioritizing the spatial aspect of taskwork execution. Thus, actions that occur in the same physical location, regardless of other attributes such as agent interests, skills, or responsibilities, are assigned to the same agents. In doing so, this principle ignores the organizational structure, resulting in agents that do not have a coherent set of actions with regards to their job descriptions. This allocation is developed from CONOPS Appendix C: Gate-to-Gate operations.

Specifically, 5 important locations were recognized, resulting in 5 agents:

1. FleetHQ: Captures the actions occurring at the central, potentially remote, location where the fleet of vehicles is coordinated from.
2. InsideAircraft: Captures the actions that occur within the aircraft itself.
3. Gate: Captures all the actions that occur on the ground, at the gate.
4. FAAComputer: Captures the actions that allow coordination with regulatory bodies like the FAA who authorize the flights and missions and enable AAM operations.
5. VertiportComputer: Captures the actions that occur at the vertiports, related to the coordination of vehicle landing/takeoffs, coordinating cargo loading/unloading, and allocating space for these actions to occur.

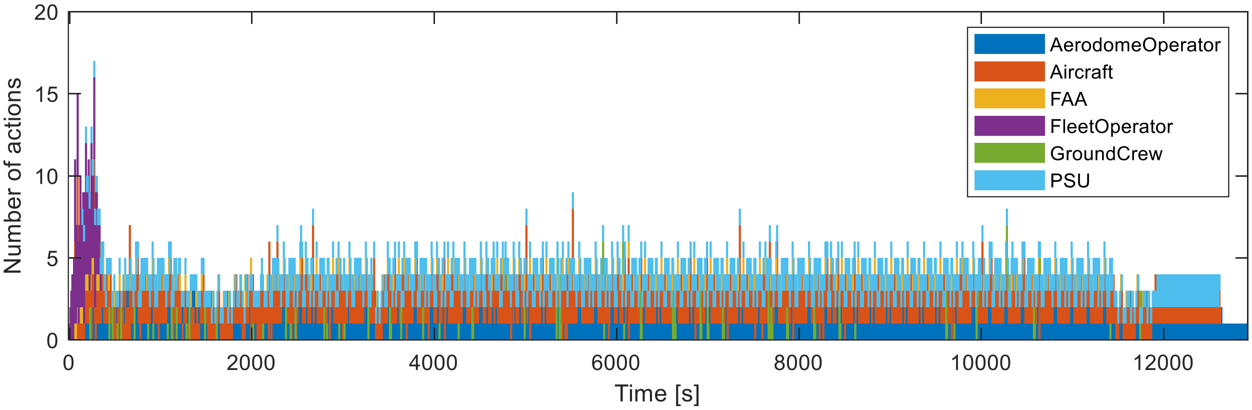
When applied to the taskwork actions required to execute the Concepts of Operations, these principles allocate authority as shown in Table 2. However, these allocations are based on an interpretation of the qualitative, underspecified description of the Concepts of Operations. For example, one section states: “The PSU assigns a takeoff slot and, in coordination with the AAM aerodrome operator, initiates departure sequencing.” Similarly, different agents are described as performing the same action at different points in the document. For example, the action to confirm all clear for landing is described as done independently by PSU, Aircraft and Crew, and the Aerodrome at different points in the document [2].

Table 2. Allocation of authority to execute taskwork

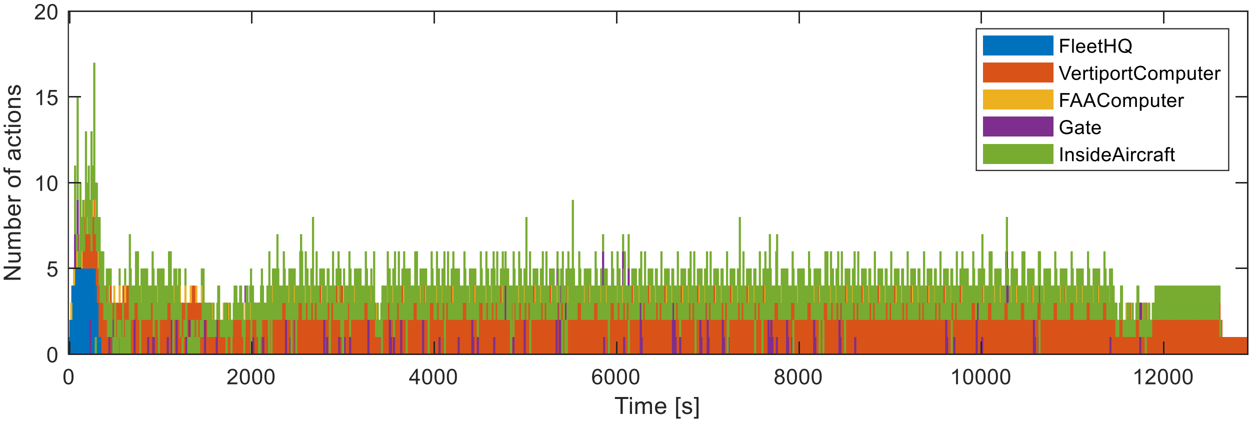
	Executing Agent	
Action	Location Based	Role Based
Assign mission	FleetHQ	FAAComputer
Flight plan	FleetHQ	FAAComputer
File operations plan	FleetHQ	FAAComputer
Request performance authorization	FleetHQ	FAAComputer
Approve performance authorization	FAAComputer	FAAComputer
Receive operations approval	FleetHQ	FleetHQ
Request flight authorization	InsideAircraft	VertiportComputer
Authorize flight	FAAComputer	FAAComputer
Initiate depart sequence	InsideAircraft	VertiportComputer
Initiate takeoff planning	InsideAircraft	VertiportComputer
Allocate takeoff pad	VertiportComputer	FleetHQ
Board/process payload	Gate	FleetHQ
Approve taxi/takeoff	VertiportComputer	VertiportComputer
Takeoff	InsideAircraft	InsideAircraft
Initiate arrival sequence	InsideAircraft	VertiportComputer
Request landing clearance	InsideAircraft	VertiportComputer
Allocate landing pad	VertiportComputer	FleetHQ
Issue landing clearance	VertiportComputer	InsideAircraft
Confirm all clear for landing	VertiportComputer	InsideAircraft
Unload aircraft	Gate	FleetHQ
Maintenance	Gate	FleetHQ
Systems check	InsideAircraft	FleetHQ
Assess battery	Gate	FleetHQ
Charge battery	Gate	FleetHQ

4.3 Measures of the Different Principles for Allocating Authority

To measure the allocations associated with the role and location principles, WMC simulations were conducted with each. At the start of each simulation, the appropriate agent models were created and allocated authority to execute the appropriate actions. Figure 6 shows the total action trace for the two principles. Note that the timeline of all actions combined is the same, but they are distributed differently between the agents.



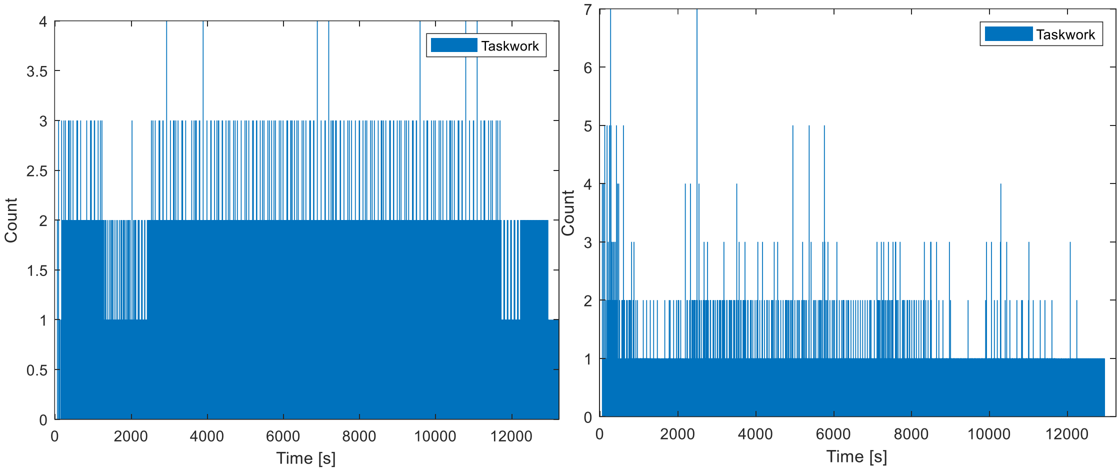
a) Role-Based



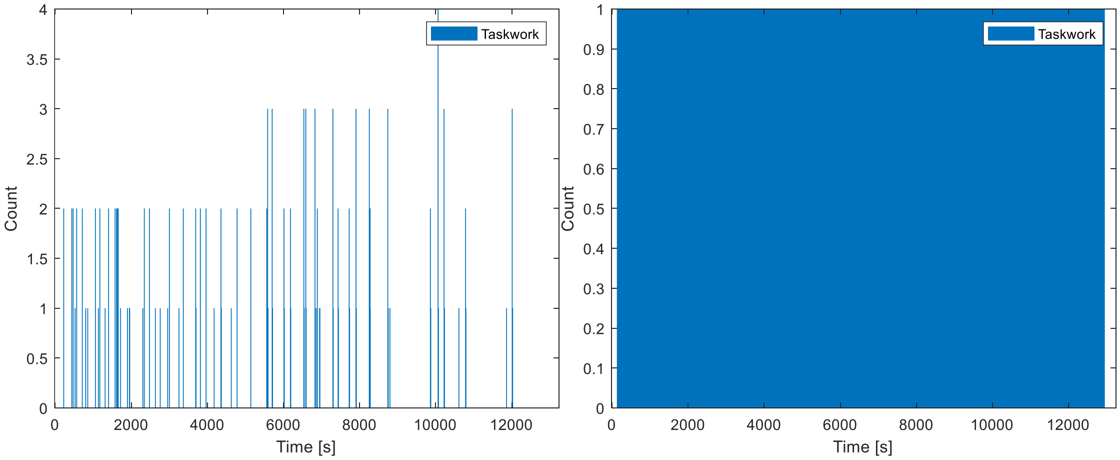
b) Location-Based

Figure 6. Action traces with only allocation of authority

Examining measures at the agent level, the time history of the agents' actions varies between the different allocations, and between the agents within each allocation. For example, Figure 7 shows the time histories of four agents in the Role-Based allocation. Some agents such as the PSU have a constant task load that does not fluctuate throughout the day since this agent is not simultaneously executing many tasks. The Aircraft agent executes tasks frequently throughout the day and the simultaneous load on the agent varies significantly. On the other hand, the Aerodrome Operator has a more predictable simultaneous task load. The Ground Crew executes a few tasks throughout the day with minor fluctuations in its simultaneous load. Thus, the maximum simultaneous taskload expected of each agent is different, as summarized in Figure 8.



a) AerodromeOperator *b) Aircraft*



a. GroundCrew *d) PSU*

Figure 7. Action traces for four agents in Role-Based allocation of authority

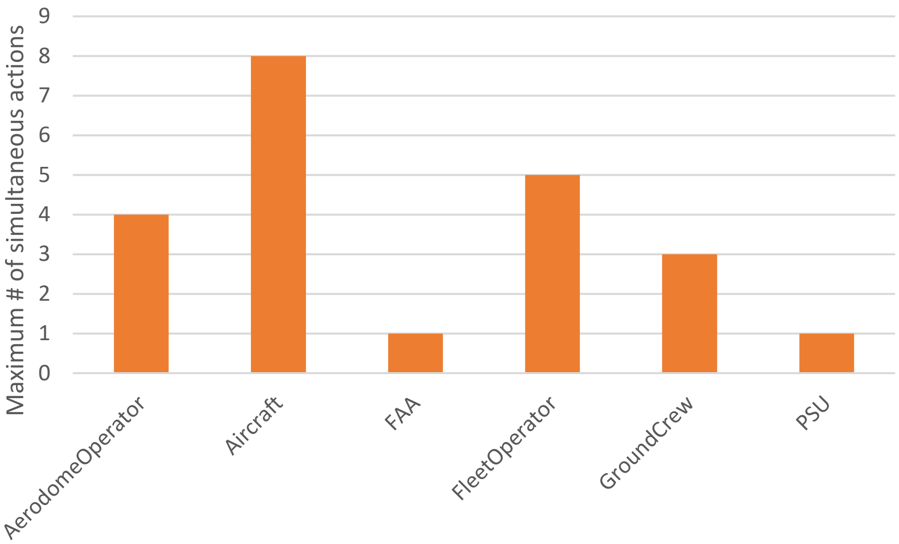
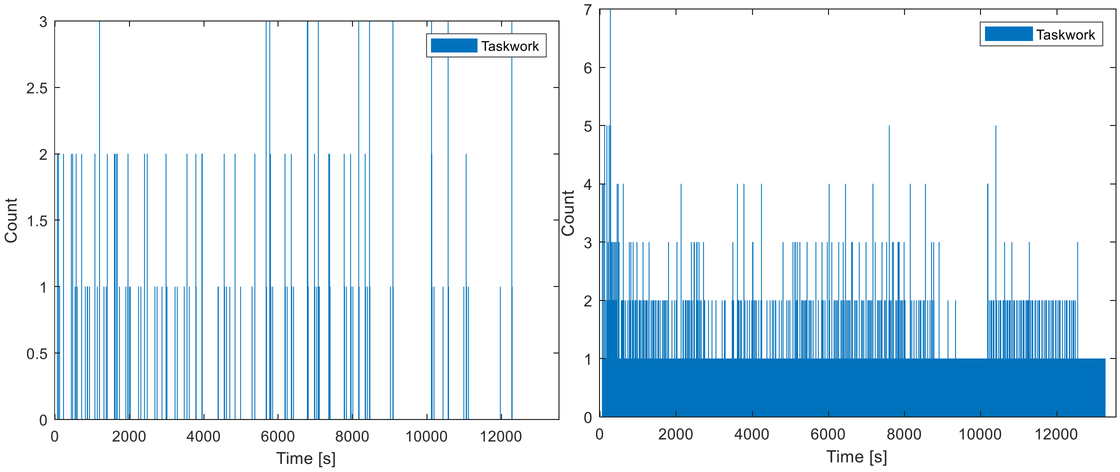
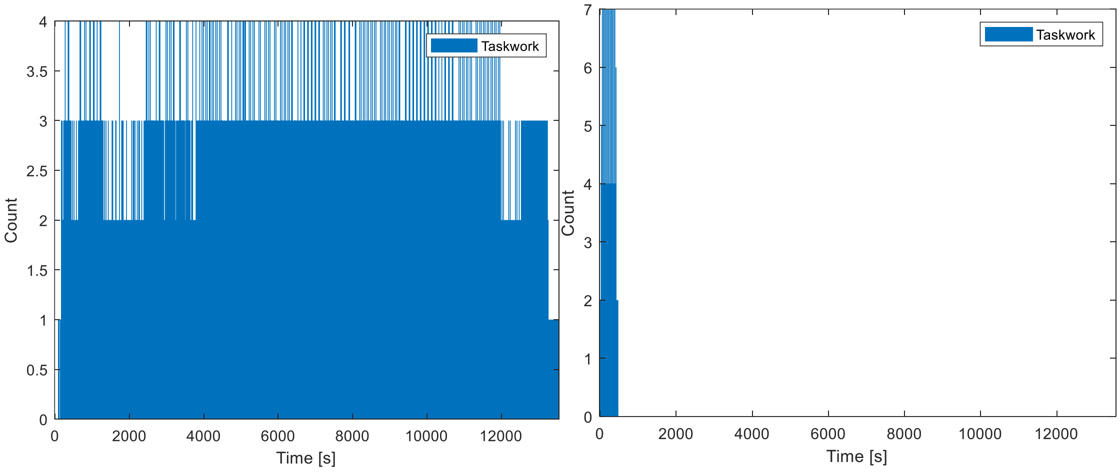


Figure 8. Maximum simultaneous taskwork only load for Role-Based agents

The timeline of actions executed by agents in the Location-Based allocation is shown in Figure 9. Action traces for four agents in Location-Based allocation of authority and the maximum simultaneous actions is shown in Figure 10. In this case, the InsideAircraft agent has high variability in the number of actions it needs to execute throughout time. The number of actions executed at the Vertiport agent is consistently high, although the maximum number of actions does not exceed that for the InsideAircraft agent. The Gate and FleetHQ agents do not have as many task actions as the InsideAircraft agent and the maximum number of actions they need to simultaneously execute is also not as high.



a) Gate *b) InsideAircraft*



b) VertiportComputer *d) FleetHQ*

Figure 9. Action traces for four agents in Location-Based allocation of authority

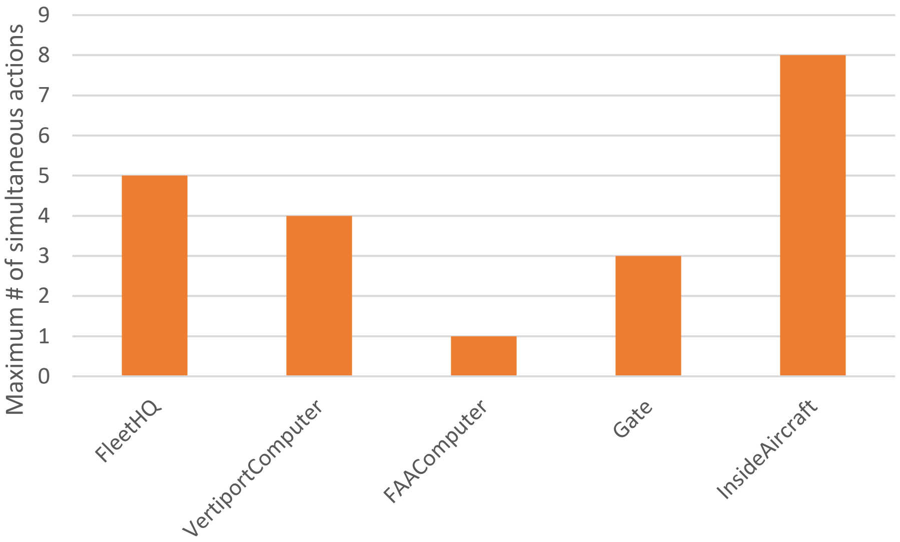
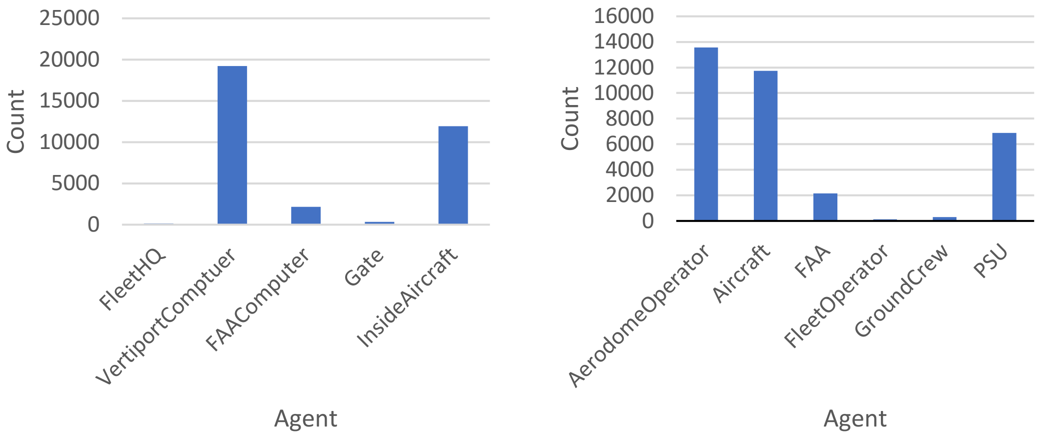


Figure 10. Maximum simultaneous taskwork load for Location-Based agents

The two allocations have similar aggregate scores. The Role-Based allocation requires 34780 instances of information transfer in aggregate, and the Location-Based allocation requires 33797. However, the different agents have very different information requirements, which require transferring different amounts of information, as shown in Figure 11.

At the mission and organization level, the time to complete any one mission and to complete the entire day's activities were identical since the agents can execute any number of tasks simultaneously, resulting in no delays. In these simulation runs, the agents perform all the taskwork as soon as it is required, with only variations in which agent is executing the action, not when the action is executed.



a) Location-Based *b) Role-Based*

Figure 11. Information transfer for the two allocations of authority

5 Results: Allocation of Responsibility

The second objective of this thesis is *to investigate the impact of also assigning responsibility for outcomes, thereby also allocating monitoring actions to the agents*. This requires developing some principles by which the correct set of agents can be assigned responsibility for each action. This chapter expands on the two principles previously developed by also allocating responsibility and evaluates the impact on metrics at the agent-, mission- and organization- levels throughout WMC simulations.

The NASA CONOPS document does not detail where monitoring should be performed or by whom, or detail who has responsibility for outcomes [2]. This information was inferred from the text of the CONOPS which includes descriptions of employment relations, who maintains conformance during certain phases of flight, who conducts strategic deconfliction and negotiations, and other descriptions that indicate, but do not explicitly outline, the responsible agents for all the actions. Some of these descriptions are given below:

1. Ground Services: “These services are provided by licensed and certified personnel employed by UAM aerodrome operators, fleet operators, or third parties contracted by either the UAM aerodrome operators or fleet operators”

2. Fleet Operator: “Monitors conformance” during descent and landing and is “responsible for operational control of aircraft and fleet operations.”

3. Pilot in Command (PiC): Holds “final authority and responsibility for the operation and safety of the flight.”

4. PSU: “PSUs also exchange data and record data as required by regulators (e.g., the FAA) for regulatory and fleet operator accountability purposes."

5. FAA: Collaborates with PSU to “provide regulatory and operational oversight.”

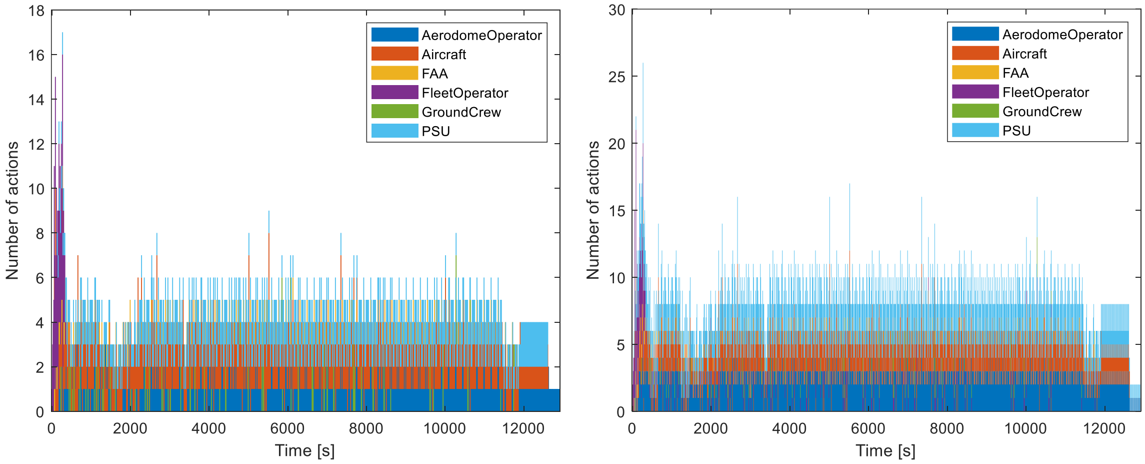
The NASA CONOPS also describes the same action as being done by multiple “stakeholders”. Since only one agent can be assigned as executing or responsible for an action in WMC, the agent not chosen to be the executing agent is taken to be the responsible agent. From these non-specific descriptions, a systematic assignment of responsible and executing agents was developed, as seen in Table 3.

Table 3. Allocation of responsibility and authority

	Location		**Role**	
Action	Agent Executing	Agent	Agent Executing	Agent Responsible
Assign mission	FleetHQ	FAAComputer	FleetOperator	FleetOperator
Flight plan	FleetHQ	FAAComputer	FleetOperator	FleetOperator
File operations plan	FleetHQ	FAAComputer	FleetOperator	PSU
Request performance authorization	FleetHQ	FAAComputer	FleetOperator	FleetOperator
Approve performance	FAAComputer	FAAComputer	FAA	FAA
Receive operations approval	FleetHQ	FleetHQ	FleetOperator	FleetOperator
Request flight authorization	InsideAircraft	VertiportCompute	Aircraft	Aircraft
Authorize flight	FAAComputer	FAAComputer	FAA	FAA
Initiate depart sequence	InsideAircraft	VertiportCompute	Aircraft	PSU
Initiate takeoff planning	InsideAircraft	VertiportCompute	Aircraft	AerodomeOperato
Allocate takeoff pad	VertiportCompute	FleetHQ	AerodromeOperato	PSU
Board/process payload	Gate	FleetHQ	AerodomeOperator	FleetOperator
Approve taxi/takeoff	VertiportCompute	VertiportCompute	AerodromeOperato	PSU
Takeoff	InsideAircraft	InsideAircraft	Aircraft	AerodomeOperato
Initiate arrival sequence	InsideAircraft	VertiportCompute	Aircraft	PSU
Request landing clearance	InsideAircraft	VertiportCompute	Aircraft	PSU
Allocate landing pad	VertiportCompute	FleetHQ	AerodromeOperato	PSU
Issue landing clearance	VertiportCompute	InsideAircraft	PSU	AerodomeOperato
Confirm all clear for landing	VertiportCompute	InsideAircraft	AerodomeOperator	Aircraft
Unload aircraft	Gate	FleetHQ	GroundCrew	AerodomeOperato
Maintenance	Gate	FleetHQ	GroundCrew	FleetOperator
Systems check	InsideAircraft	FleetHQ	Aircraft	FleetOperator
Assess battery	Gate	FleetHQ	GroundCrew	FleetOperator
Charge battery	Gate	FleetHQ	GroundCrew	FleetOperator

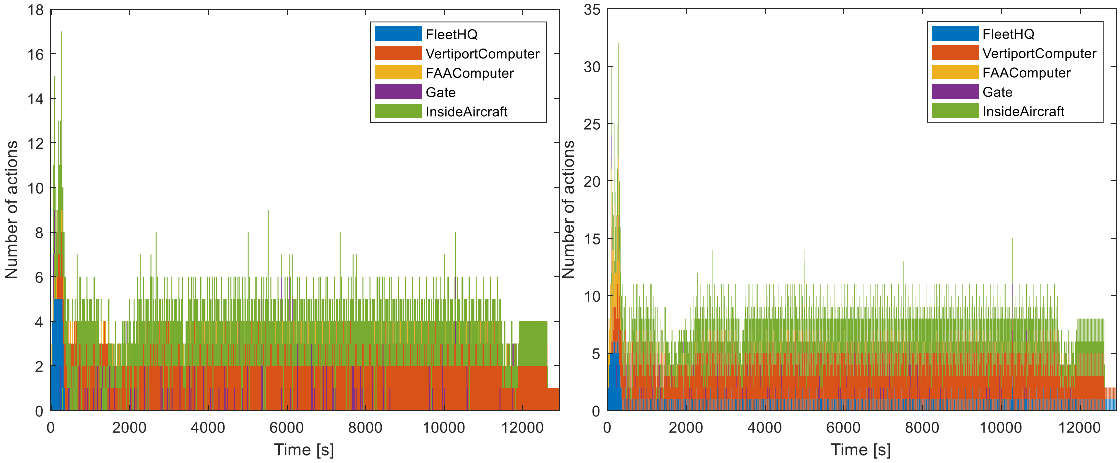
These allocations of authority and responsibility were then simulated in WMC in the same case study as before. Wherever responsibility is allocated for the outcomes to a different agent than that allocated authority to execute the action, the responsible agent is given a monitoring action to execute simultaneous with the authorized agent’s taskwork action. Figure 12 and Figure 13 compare the action traces of the

Location-Based and Role-Based allocation. The addition of allocating responsibility significantly increases the number of actions to be performed throughout the day's operations.



a) Authority Only *b) Authority and Responsibility*

Figure 12. Action traces for Role-Based allocation



a) Authority Only *b) Authority and Responsibility*

Figure 13. Action traces for Location-Based allocation

The agents also have different experiences with the allocation of responsibility. The action traces of four agents with the Role-Based allocation of responsibility are shown in Figure 15. The largest impact is on the PSU agent, whose maximum simultaneous taskwork actions never exceeded one. With the additional allocation of responsibility, the PSU agent has many monitoring actions and thus experiences a maximum of seven simultaneous actions. Conversely, GroundCrew agent is not responsible for any monitoring actions and thus is not affected. The Aircraft and AerodromeOperator agents experience points in time when the monitoring adds one additional action, and this is dispersed evenly throughout the day. Figure 14 summarizes the maximum simultaneous number of actions that each agent must execute with the addition of monitoring due to the allocation of responsibility.

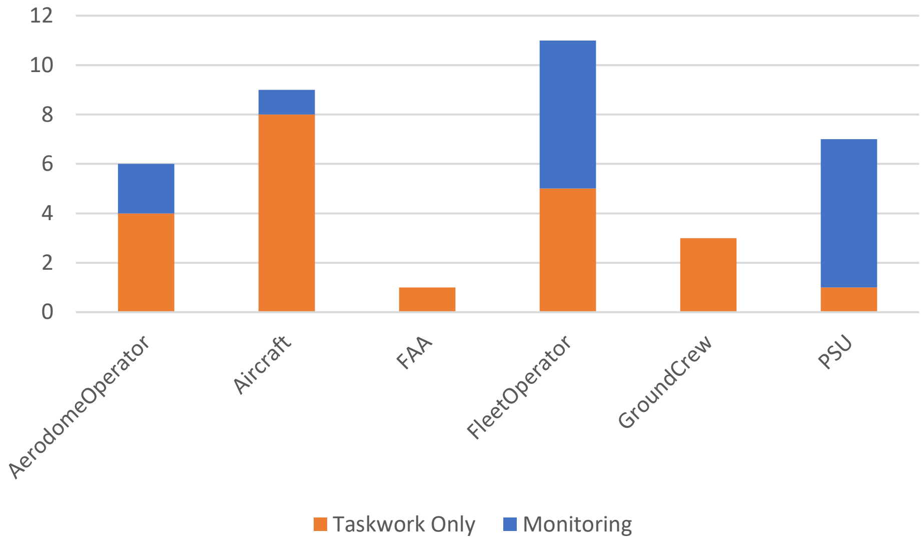
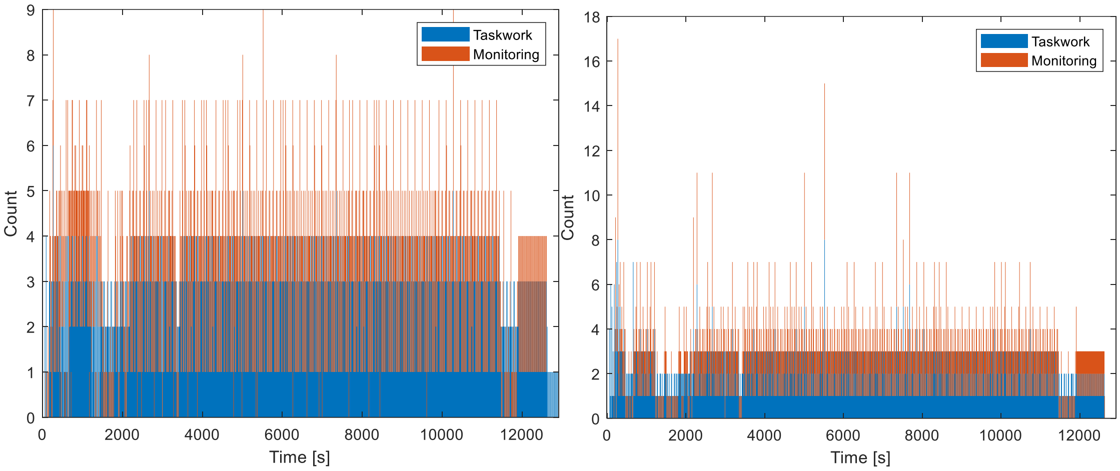
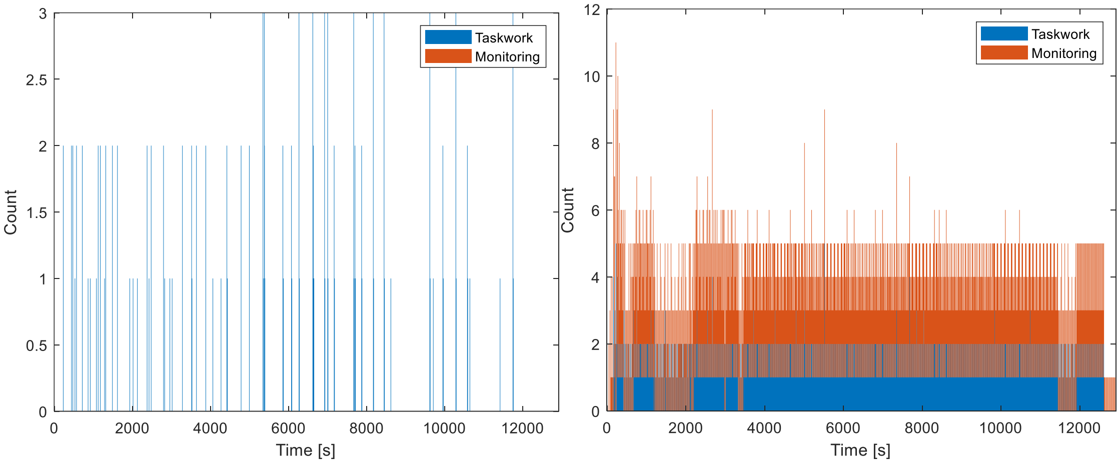


Figure 14. Maximum simultaneous taskwork and monitoring load for Role-Based agents



a) AerodromeOperator

b) Aircraft

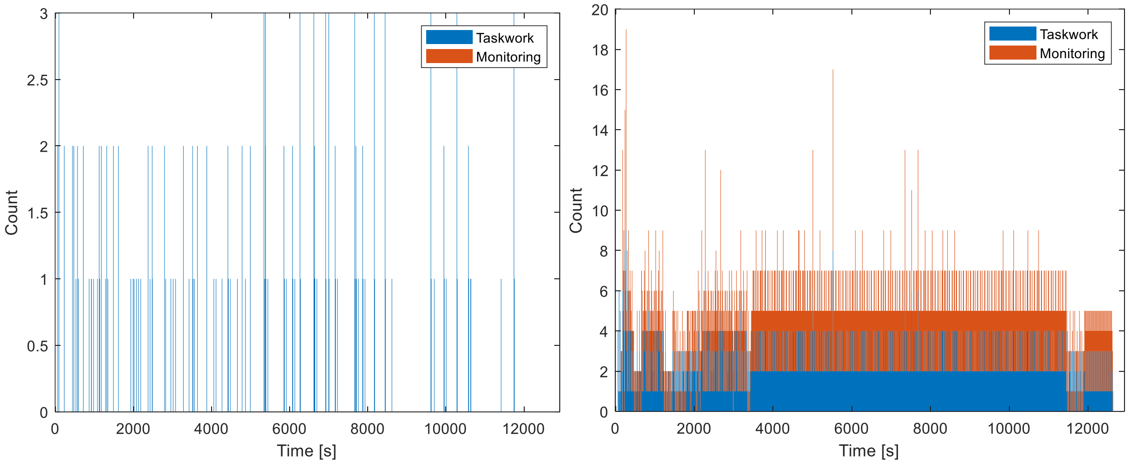


c)GroundCrew

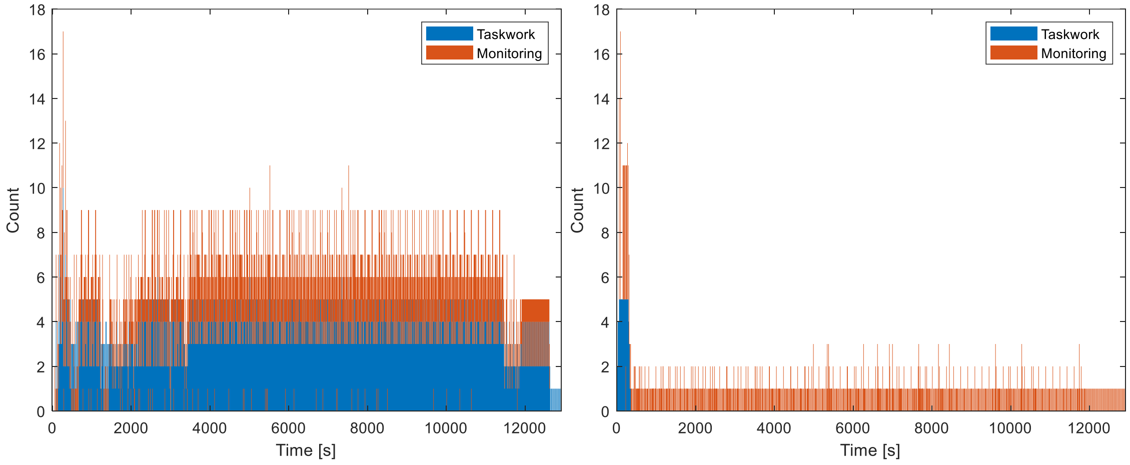
d)PSU

Figure 15.Action traces for four agents with Role-Based allocation of authority and responsibility

Likewise, Figure 16 shows the monitoring and task actions executed by four agents with the Location-Based allocation. This figure shows that monitoring impacts some agents significantly while not changing the experience of other agents at all. As seen in Figure 16, the Gate agent executes no monitoring actions, while the FleetHQ agent executes and monitors many more actions simultaneously. Also, while the FleetHQ agent only executes taskwork actions for a short time early in the day, it continues monitoring throughout the day. Thus, only considering taskwork actions would paint an incomplete picture of this agent's day.



a) Gate *b) InsideAircraft*



a) VertiportComputer *d) FleetHQ*

Figure 16. Action traces for four agents with Location-Based allocation of authority and responsibility

As seen in Figure 17, the maximum number of simultaneous actions increase for all agents except for the Gate agent, which only executes taskwork actions. The other agents are the responsible agents for some actions, with the FleetHQ agent having to execute the most additional monitoring actions simultaneously.

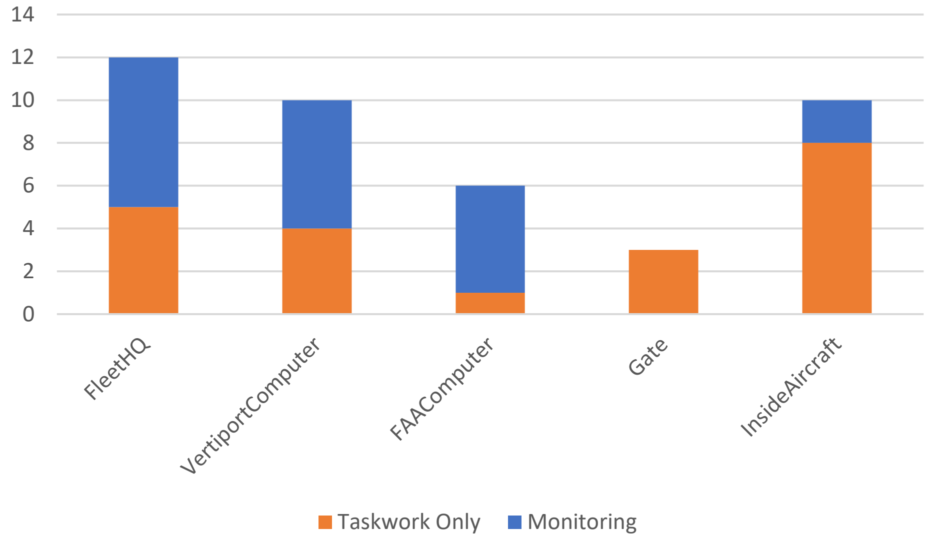


Figure 17. Maximum simultaneous taskwork and monitoring load for Location-Based agents

Monitoring also significantly increases the information each agent requires and the information transfer this demands between agents. Unlike taskwork, whose information requirements are defined by the algorithm representing their output, no established algorithm exists for monitoring. Instead, this thesis applied two different algorithms to describe potential monitoring behavior.

1. Basic Monitoring assumes the monitoring agent “gets” the output of the taskwork action to evaluate it for reasonableness. Thus, if the taskwork’s outcome is represented by M variables, it requires M elements of information, which must be transferred to the agent authorized to execute the taskwork action.
2. Full Monitoring assumes that the monitoring action “gets” not only the outcomes(s) of the taskwork, but also all the inputs to the taskwork. Thus, if the taskwork’s outcome is represented by M variables and its inputs by N variables, the monitoring action requires M+N elements of

information, which again also demands additional information transfer to the monitoring agent, especially if the taskwork requires inputs that the monitoring agent does not have.

These two behaviors represent two different approaches to monitoring, with different information requirements for each agent and amounts of information that must be transferred between agents for the purpose of monitoring. In this simulation, a further distinction was made for those taskwork actions in the Command work model that recur frequently, checking status, but only provide an outcome – such as an approval or clearance – when needed. In these cases, monitoring was assumed to only need information when the taskwork provided an output.

The extent of monitoring impact the total information transfer, is seen in Figure 18. As seen in Figure 19 and Figure 20, the information transfer induced by monitoring can disproportionately affect some agents, while not impacting other agents at all. This mirrors the allocation of responsibility, since some agents have many monitoring actions, while other agents have none. For both allocations, the agent representing the operations at the gate are not affected significantly as they are only responsible for task actions. However, other agents, exemplified by the PSU agent as seen in Figure 20, are dramatically impacted by monitoring. Only analyzing the task actions might have resulted in the conclusion that the PSU does not need infrastructure or systems to facilitate communication, but if it is responsible for many of the monitoring activities, it becomes the agent that requires the most information transfer from all the other agents. This behavior is still present, although not as pronounced, in the Location-Based allocation, with some agents more than doubling the information transfer they need.

Figure 18. Aggregate information transfer for different types of monitoring extents and allocation of authority and responsibility

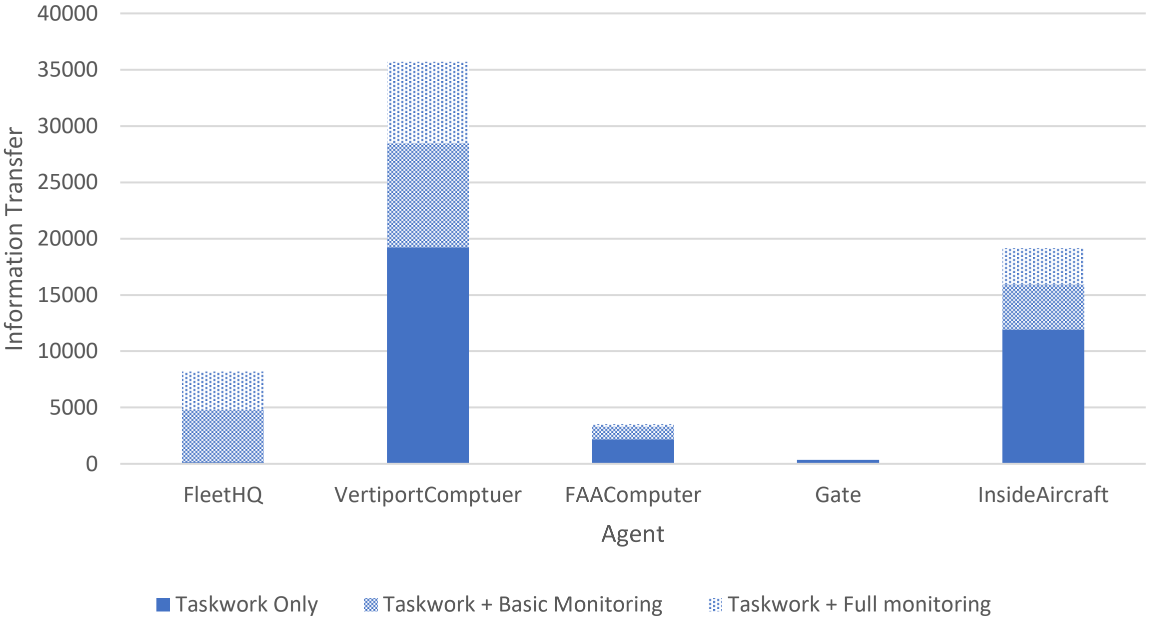


Figure 19 Information transfer per agent for Location-Based allocation of authority and responsibility

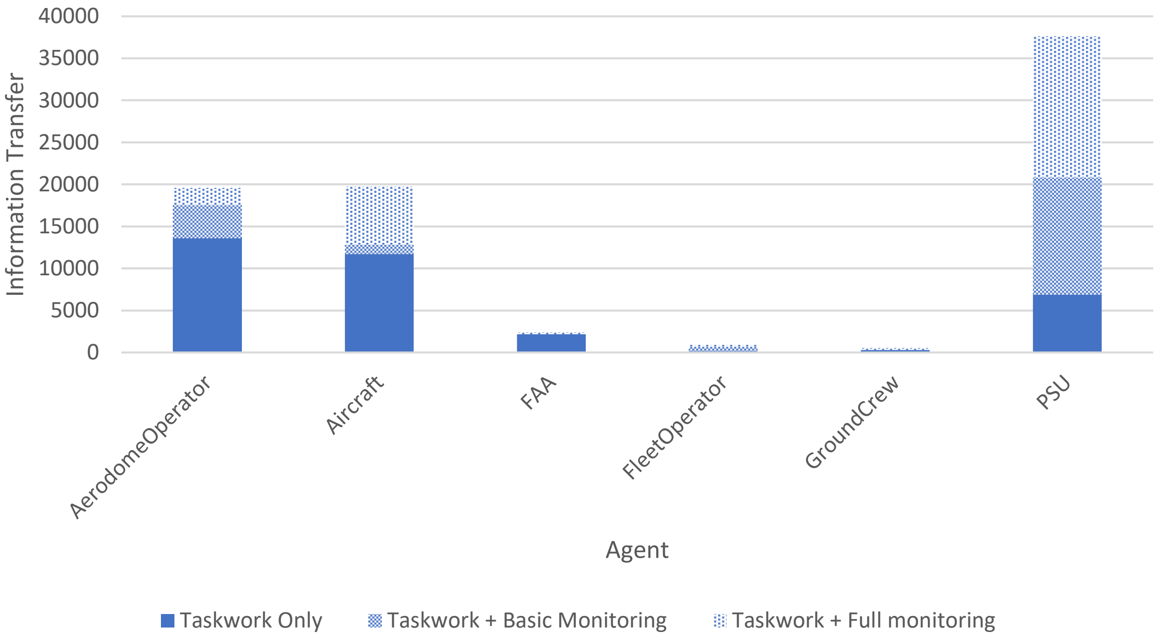


Figure 20. Information transfer per agent for Role-Based allocation of authority and responsibility

6 Results: Impact of Agent Limits

The third objective of this thesis is to *assess the impact on these allocations of potential agent limits in the number of simultaneous actions each can complete, or a more detailed model of agents' fundamental limitations in working memory*. Thus, this chapter implements agent models that limit the number of actions that each performs. Additionally, since each action is not identical, a more detailed weighting of the actions is applied to better relate them to agent limitations, particularly with regard to the short-term memory requirements of executing taskwork and monitoring actions.

6.1 Next-Available-Agent Allocation to Task Limited Agents

As a starting point in the analysis of the impact of agent task limits, a next-available-agent allocation scheme was applied to the same AAM case study as before. Simulations examined 25 different combinations of number of agents (from 1 to 5) and number of simultaneous actions each can execute (again, from 1 to 5). In this case, every instance that an action needs to be executed, it is passed to an agent that is available. If all agents are saturated, the agent that will next complete an active action is given this incoming action, which is delayed until the agent can attend to it.

Table 4. Time [s] required to complete the day's operations with agents with different tasklimits

Tasklimit	Number of Agents	1	2	3	4	5
1		46411	24982	19572	17632	21522
2		24751	17942	16602	14602	16412
3		16471	14232	13442	13082	14052
4		14531	13342	13190	13074	13002
5		14231	13222	13092	13002	13002

From Table 4, it can be observed that the total time to completion for tasklimited agents is equal to the minimum possible time (13002s) if the sum of the tasklimits of all the agents is greater than or equal to the maximum simultaneous task load, which for this case study is 17. On the other hand, if more actions are required at any instant than the set of agents can simultaneously perform, there is no way to assign authority to agents that can prevent delays. In the worst case, one agent required to perform all actions sequentially delays the time to completion out to 46411s.

Table 4 also illustrates that the impact of adding more agent is not symmetric with the impact of the same number of agents being able to perform more actions. Five agents each with a tasklimit of "1" delays the time to completion to 21522s, while 1 agent with a tasklimit of 5 can complete in 14231s. This is due to the different inter-leaving of the two queuing mechanisms underlying each case. The impact of number of agents and their task limit is thus expected to vary between different scenarios depending on how often the agents are saturated and how inter-leaved their actions are.

6.2 Modeling Agents Limited by Working Memory

Previous attempts at modeling the behavior of human agents have either considered each action to contribute the same load on the agent or have included agents without limits [15]. A more accurate representation would allow for different actions to contribute differently to agents' limits. This thesis models agents as being fundamentally limited by working memory. Each action requires different amounts of information to be held in working memory. The agent will only execute a new action when it has available capacity in its short-term memory for that specific action; otherwise, it will delay the action and wait until the current load on its short-term memory is reduced such that it has the capacity for the upcoming action. This maximum capacity is taken from the previous description of a human's memory limit in Chapter 2, resulting in a memory limit of nine items as a basis for analysis for this thesis.

6.2.1 Working Memory Requirements of Taskwork Actions

Implementing this agent model requires each action to report the number of information "chunks" that must be held in working memory to complete it. In the case of taskwork actions, the information needed to perform the action is already known as the resource variables needed to execute the action.

Relating this to memory requirements requires analyzing each action within the assumed context of identifying (1) which information element can be confirmed via a display or interface without needing to be held in working memory, and (2) which information can be "chunked" together as one element in memory, and which needs to be stored in working memory as multiple elements of information.

For this case study, the information needed to be held in working memory is shown in Table 5. Some actions, such as Assign Mission, may need to hold four elements in working memory even to identify appropriate information from the interface; this will effectively weight this action by a factor of four in its demands on the agent. Other actions need only hold one element in working memory.

Table 5. Information required in memory derived to execute each action

Action	Information Required in Memory				Number of information elements from interface
Allocate landing pad	number of missions	vertiport pad	vehicle mission ID	-	5
Allocate takeoff pad	number of missions	vertiport pad	vehicle mission ID	-	5
Approve performance authorization	mission assigned vehicle	performance authorization status	-	-	8
Approve takeoff	assigned mission ID		-	-	5
Assign mission	number of missions	number of vehicles	request time	assigned mission ID	18
Authorize flight	number of vehicles	assigned mission ID		-	9
Confirm clear for landing	number of vehicles	assigned mission ID	arrival time	-	5
Initiate arrival sequence	landing pad ID	next available vertiport	assigned Mission ID	-	11
Initiate departure sequence	takeoff pad ID	next available vertiport	assigned mission ID	-	7
Initiate takeoff planning	number of vehicles	assigned mission ID	-	-	13
Issue landing clearance	number of vehicles	assigned mission ID	-	-	15
Board/process payload	mission ID	-	-	-	2
File operations plan	mission ID	-	-	-	2
Flight plan	assigned vehicle ID	assigned mission ID	-	-	7
Receive operations plan approval	mission ID	assigned vehicle ID	-	-	2
Request flight authorization	mission ID	assigned vehicle	-	-	3
Request landing clearance	mission ID	assigned vehicle ID	-	-	3
Request performance authorization	mission ID	-	-	-	2
Unload aircraft	mission ID	-	-	-	4
Assess Battery	vehicle ID	-	-	-	2
Charge Battery	vehicle ID	-	-	-	2
Maintenance	vehicle ID	assigned mission ID	-	-	6
Systems Check	vehicle ID	-	-	-	2
Takeoff	vehicle ID	-	-	-	7

6.2.2 Working Memory Requirements of Monitoring Actions

Similarly, anytime a monitoring action is triggered by an authority-responsibility mismatch, its requirements on working memory must also be calculated. This depends strongly on the type of monitoring that is being assumed. The working memory requirements of a "basic" monitoring action is one since the responsible agent only needs to remember one piece of information: the outcome. However, the working memory requirement of a "full" monitoring action must both span the internal calculations and compare its internal calculations to the output. Thus, if the underlying taskwork requires "n" elements, "full" monitoring requires "n+1" elements to be held in the working memory of the agent responsible for monitoring the outcome.

6.3 Impact of Memory-Limited Agents

Continuing this chapter's analysis of the impact of agent limits; this thesis then applied the agent model detailed in the previous section (i.e. with a limit on working memory) in simulation evaluations using the "Role-" and "Location-" Based allocations developed and analyzed with perfect agents in previous chapters. This section assesses the impact of applying the memory-limited agent model in this case study on metrics at the mission level, organizational level, and agent level.

Examining the impact of the agent's memory limits on the mission, many agents need to delay their assigned actions. Collectively, this delays overall time to completion compared to the 13002s found with unlimited agents, as shown in Figure 21. This time to completion varies significantly depending on the types of monitoring and the principle behind the allocation of authority and responsibility. When utilizing a Location-Based allocation, the missions tend to take longer to complete. Conversely, employing a Role-Based allocation consistently results in faster time to completion. Note, the Role-Based allocation has six agents, one more than the five agents in the Location-Based allocation, which may distribute the work more.

Figure 21.Time to completion for Role and Location-Based allocations of authority and responsibility to memory-limited agents

When the allocation requires monitoring, the time to completion also substantially increases for both allocations. Given the higher memory requirements for full monitoring, the time to completion is delayed further when full monitoring is assumed compared to basic monitoring.

Examining the metrics at the agent's level, different agents exhibit varying patterns of saturation and task execution. As seen in Figure 22, some agents are operating at their memory limits and delaying any further actions during a higher percentage of their actions. While some agents perform a large percentage of their actions at their limit, they may not perform many actions overall, while others may perform many actions and only be saturated for a smaller proportion of them. Figure 23 shows the number of actions an agent executes while it is at saturation.

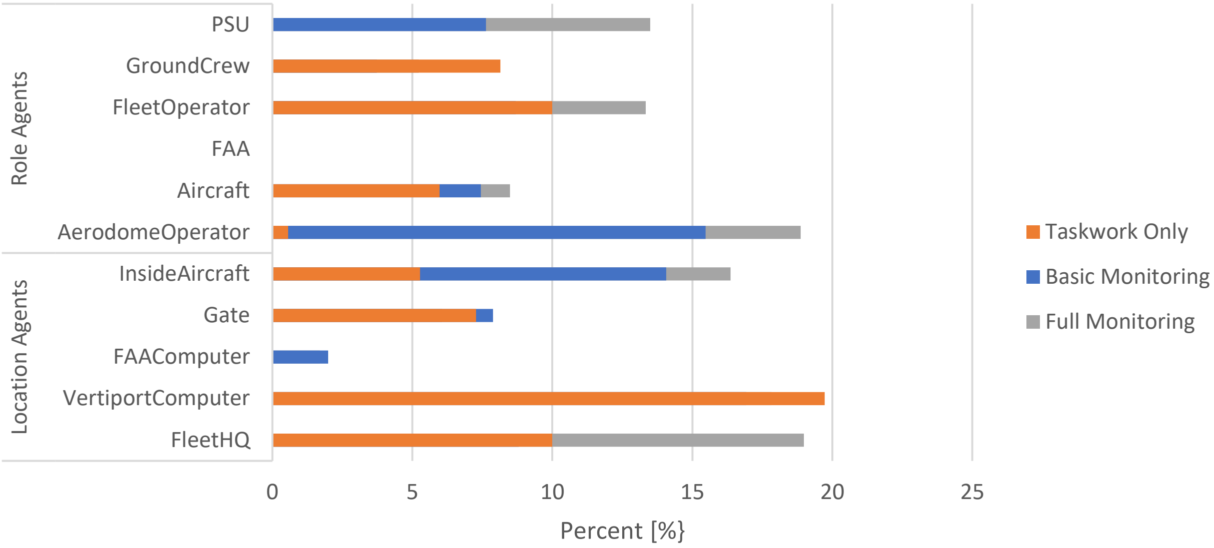


Figure 22. Percent of actions for each agent when it was at saturation when executing/monitoring the action, for Role-Based and Location-Based allocation of authority and responsibility, for full monitoring, basic monitoring, or taskwork only

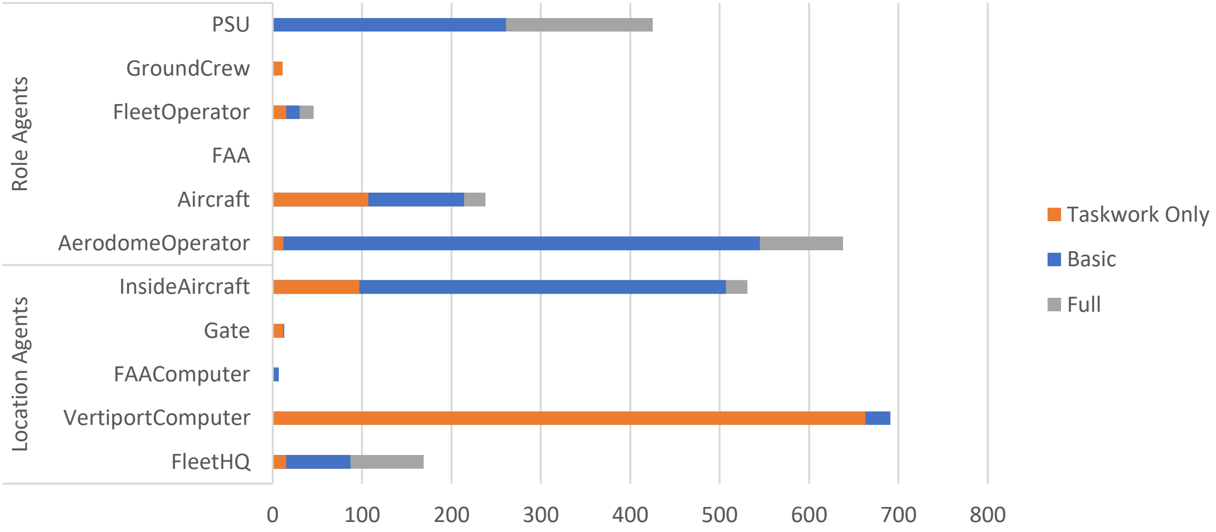


Figure 23. Number of actions for each agent when it was at saturation when executing/monitoring the action, for Role-Based and Location-Based allocation of authority and responsibility, for full monitoring, basic monitoring, or taskwork only

The intervals over which agents experience higher and lower demands on their working memory varies between agents. For example, Figure 24 shows the timeline of cumulative count of information elements that the VertiportComputer agent has needed to hold in working memory at some point in time. This agent is already frequently at its limit with its allocation of taskwork actions to execute.

Thus, increasing the monitoring extent may not lead to further saturation of that agent. This is because agents who are already highly saturated with only their task actions are less able to accommodate additional monitoring actions. This observation underscores the importance of considering an agent's taskwork (and working memory requirements induced by taskwork) when also making the agent responsible for monitoring actions.

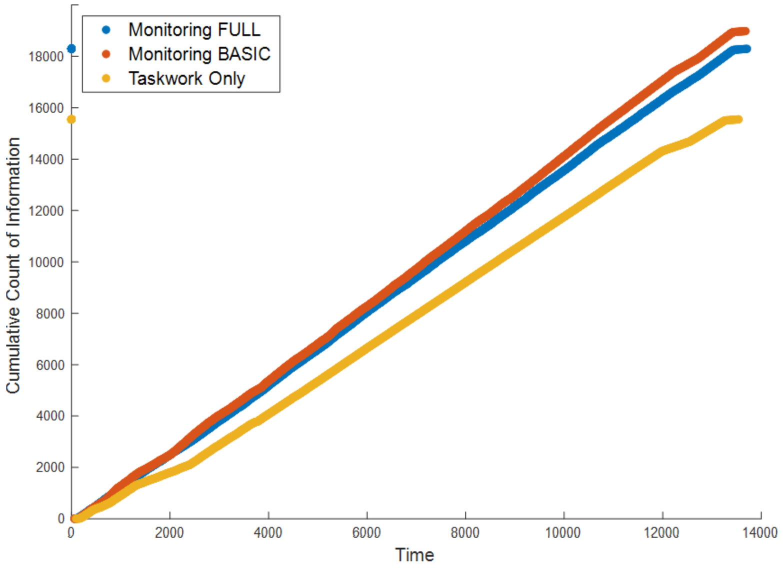


Figure 24. Cumulative mental load timeline, for full monitoring, basic monitoring, and taskwork only scenarios, for VertiportComputer agent

In contrast, as seen in Figure 25, agents that have fewer taskwork actions, and are disproportionately responsible for monitoring such as the PSU agent in the Role-Based allocation, are significantly impacted by the working memory demands of monitoring.

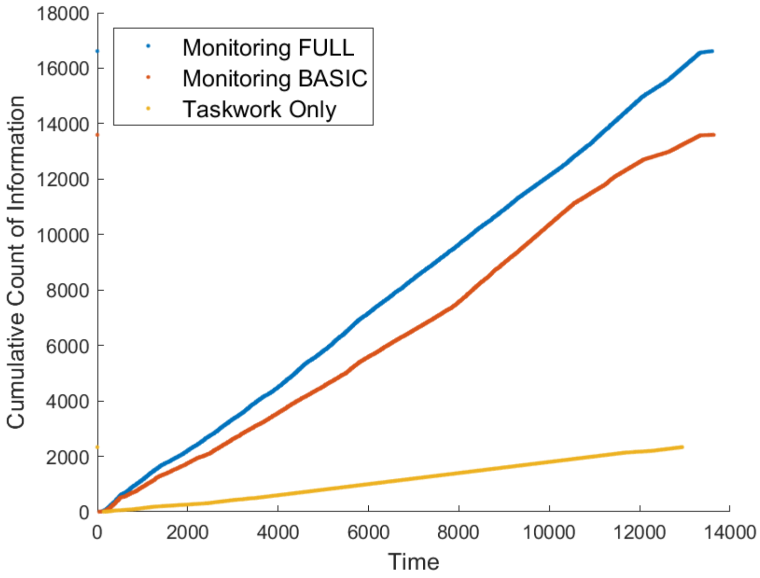


Figure 25. Cumulative mental load timeline, for full, basic, and taskwork only scenarios, for PSU agent

When the agents get saturated, they delay actions until they have the capacity to include an additional action in their active actions list. Figure 26 shows the delays in monitoring and task actions for each allocation and monitoring type, with the number of task action delays plotted in the positive y-axis and the number of monitoring action delays shown in the negative y-axis. First, it is apparent that monitoring actions tend to get delayed when "full" monitoring is required. Second, the Location-Based allocation of authority and responsibility has more delays compared to Role-Based allocation.

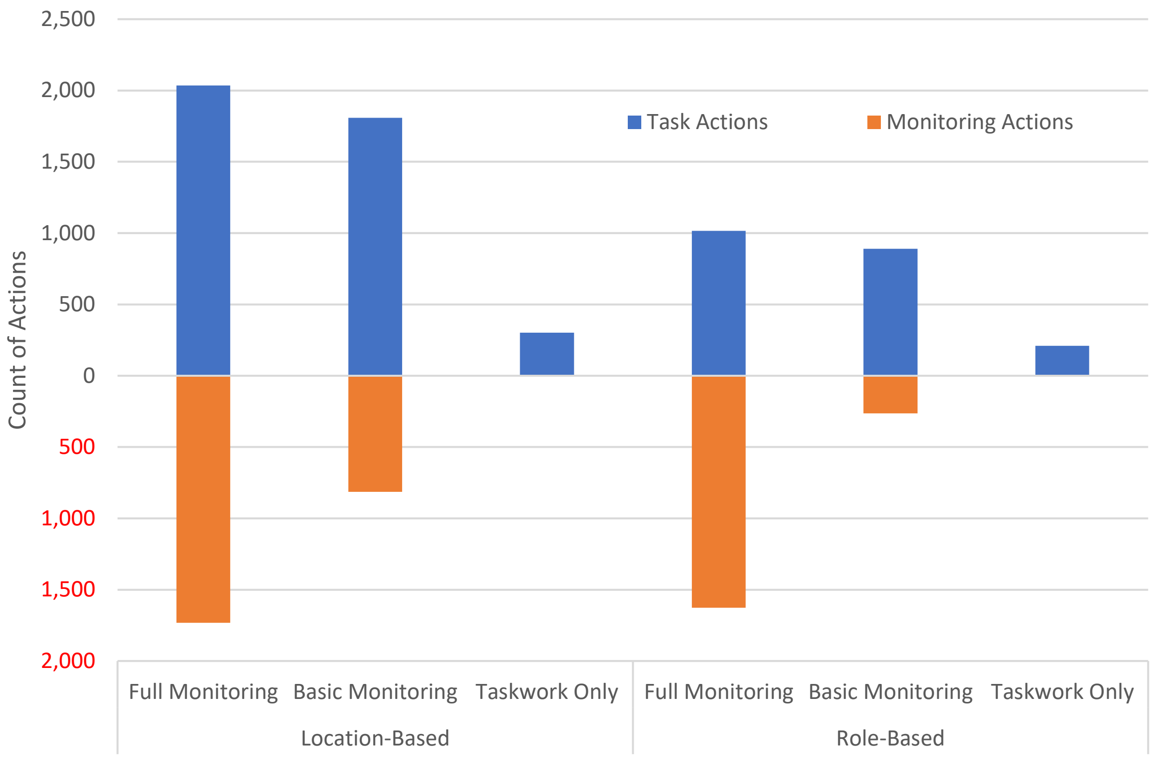


Figure 26. Count of task and monitoring actions delayed with full and basic monitoring, for Role and Location-Based allocations

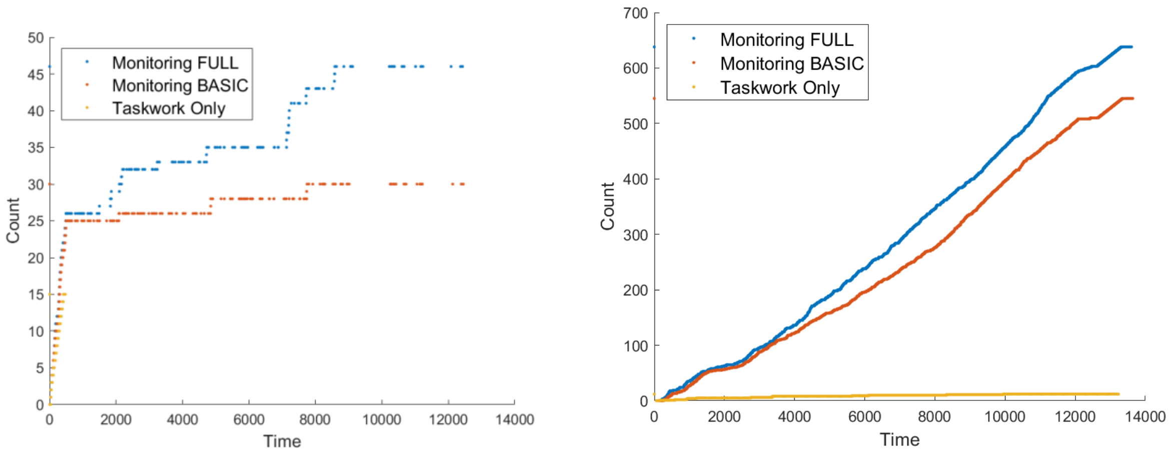
Similarly, the same is true for taskwork actions: the Role-Based allocation of authority and responsibility results in fewer delayed taskwork actions. The relationship between the number of taskwork actions delayed and the monitoring extent shows that full monitoring results in more taskwork action delays for both Role-Based and Location-Based allocations of authority and responsibility.

As seen in Figure 22 and Figure 25, the agents have different degrees of saturation, both as a percentage of their total actions and as an absolute number. This indicates that their experiences differ from one another, and more insights can be obtained from looking at the timeline of an agent's experience. Looking at the agent-specific timelines of the cumulative count of actions executed when the agent's working memory was saturated, some commonalities and differences can be found between the agents from both allocations of authority and responsibility. Figure 27 charts how many times over the course of the simulation an agent was executing an action at saturation. Each time an agent executes an action where its working memory is at capacity, and there is no space for an additional element of information, an increment is added to the chart. Looking at this metric, Figure 27 illustrates three distinct saturation behaviors:

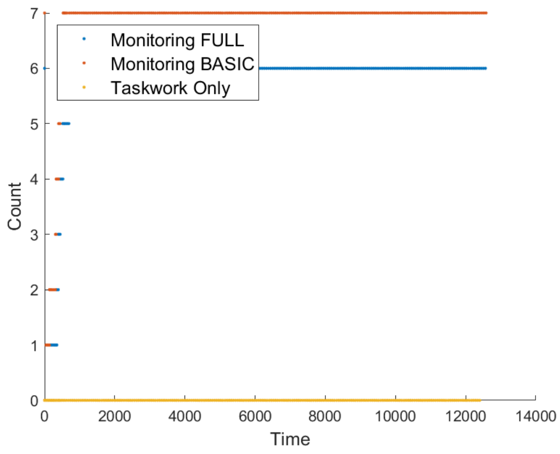
1. Intermittently Saturated: The FleetOperator agent experiences periods of high working memory requirements, often associated with acute spikes in workload or specific action demands. These intermittent episodes of saturation are typically brief and punctuate the agent's work timeline.

2. Constantly Saturated: The Aerodrome agent’s timeline depicts a state of constant and chronic saturation. This agent consistently operates at or near their working memory limits. Such chronic saturation may indicate a consistently high workload or the need for more efficient allocation of authority and responsibility to prevent working memory overload.

3. Not Saturated: The FAAComputer agent’s timeline illustrates an agent who rarely experiences saturation after an initial busy period. Once through this period, this agent's timeline shows that they generally operate comfortably below their working memory limits.



a)FleetOperator *b)AerodromeOperator*



c)FAAComputer

Figure 27. Timeline of when three different agents were saturated

Similarly, monitoring can impact when and how the count of information that has been held in an agent’s working memory accumulates. For example, the four agents shown in Figure 28 represent four distinct patterns:

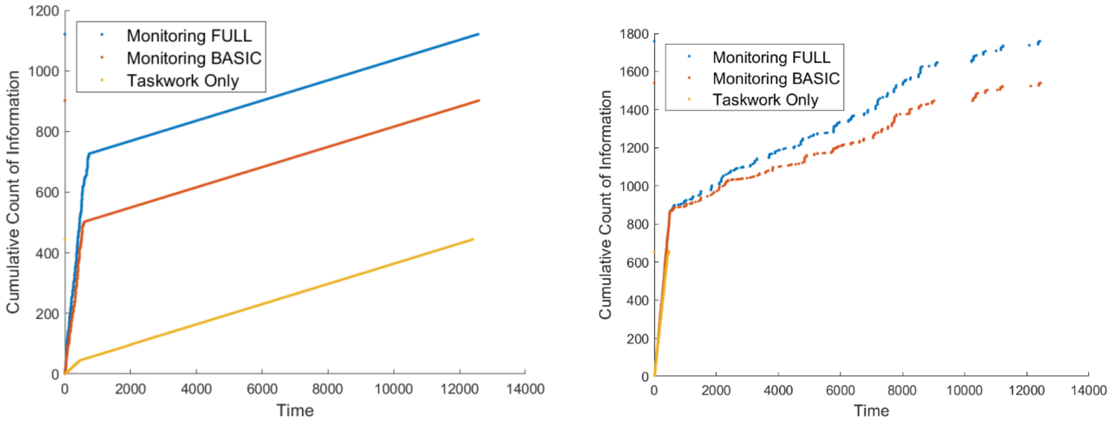
1. Initial Difference: The FAAComputer agent shows that some agents may experience a higher rate of increase in the cumulative number of information elements held in working memory during an initial phase of operations, which then lowers to a lower rate of accumulation after that. Responsibility for monitoring increases these rates.

2. Late Difference: Conversely, the FleetOperator agent shows that an agent's timeline of the record of how many information elements have been held in working memory may not be significantly impacted by monitoring actions initially but may become more pronounced or divergent as time goes on.

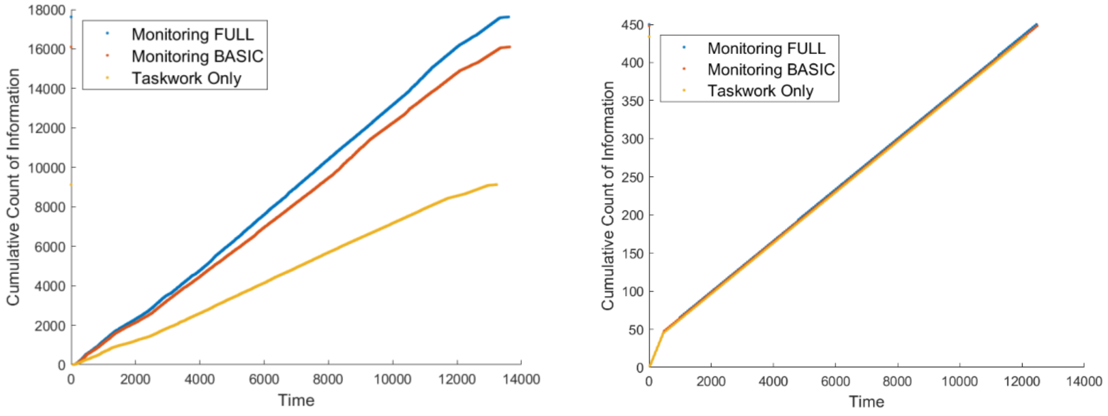
3. Constant Difference: The AerodromeOperator agent has a constant increase in the number of information elements that had been held in working memory once it is also assigned responsibility for monitoring. This corresponds to monitoring consistently throughout the entire work period.

4. No Difference: In the FAA agent's timeline, the introduction of monitoring may not significantly alter the agent's work timeline. This suggests that the agent's primary function is to complete taskwork actions.

These distinct patterns highlight the varied nature of agents' experiences when they are allocated responsibility to monitor. Understanding these variations is crucial for optimizing allocation of authority and responsibility.



a) FAAComputer *b) FleetOperator*



c)AerodromeOperator *d)FAA*

Figure 28. Effect of monitoring on cumulative mental load

6.4 Safety Implications

Monitoring actions serve as an important contributor to safety in aviation operations. Thus, their delay can be described as a reduction in protection should unsafe conditions arise, and the consequences can be severe.

As exemplar of the evolution of an unsafe condition that monitoring may serve to identify, Figure 30 shows the timeline of the battery energy state of UAS2, contrasting the expected values at the end of each flight for a fully functional battery, compared to the actual timeline of energy state with a degraded battery. By about 5000s into the day's operations, a degraded battery can decrease below the reference value "0" considered safe for flight.

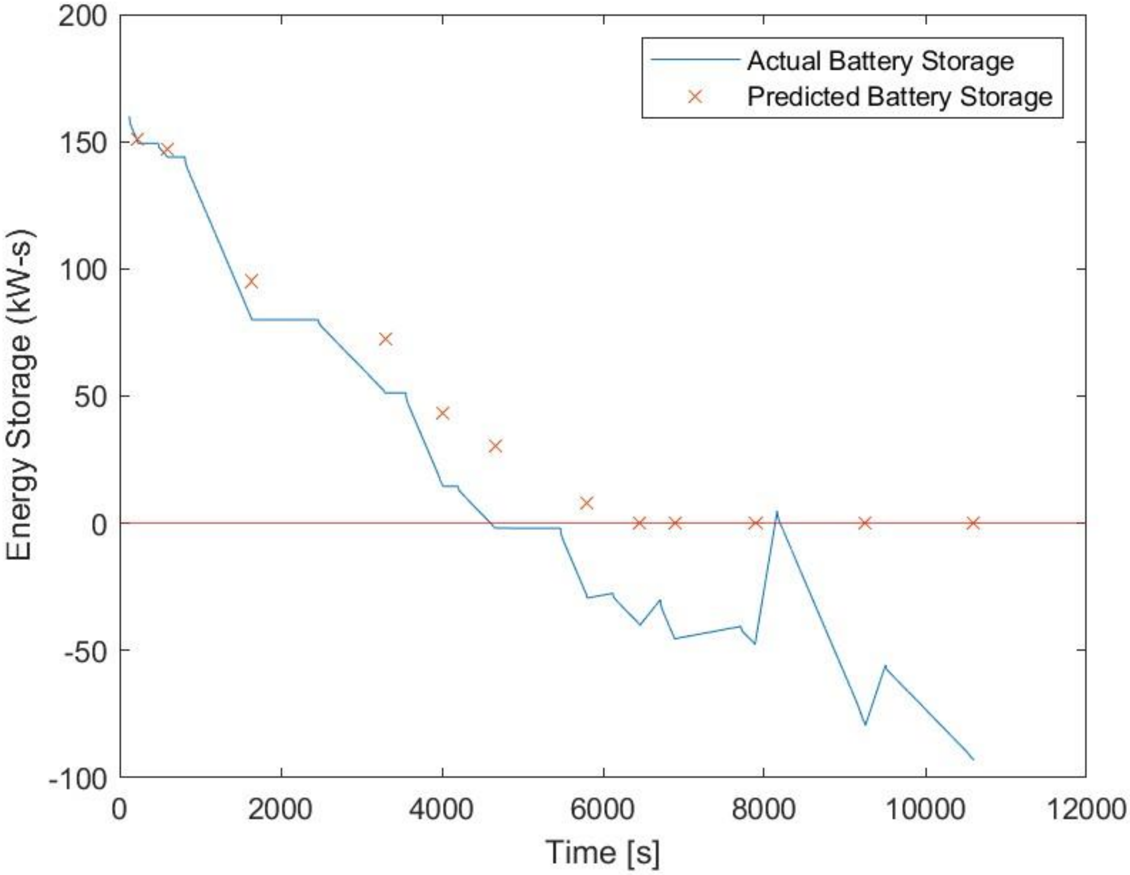


Figure 29. Predicted versus actual energy storage for UAS2 with a degraded battery

In this case study critical monitoring actions were delayed to the point that they could not be expected to catch the battery state before takeoff. To demonstrate this, Figure 30 portrays the sequence of Takeoff actions relative to the Maintenance and monitoring of that Maintenance associated with each flight for a Location-Based allocation of authority and responsibility with full monitoring when the agents are not limited and there are no delays. Each flight is followed by the simultaneous execution of a Maintenance action and the monitoring of that action.

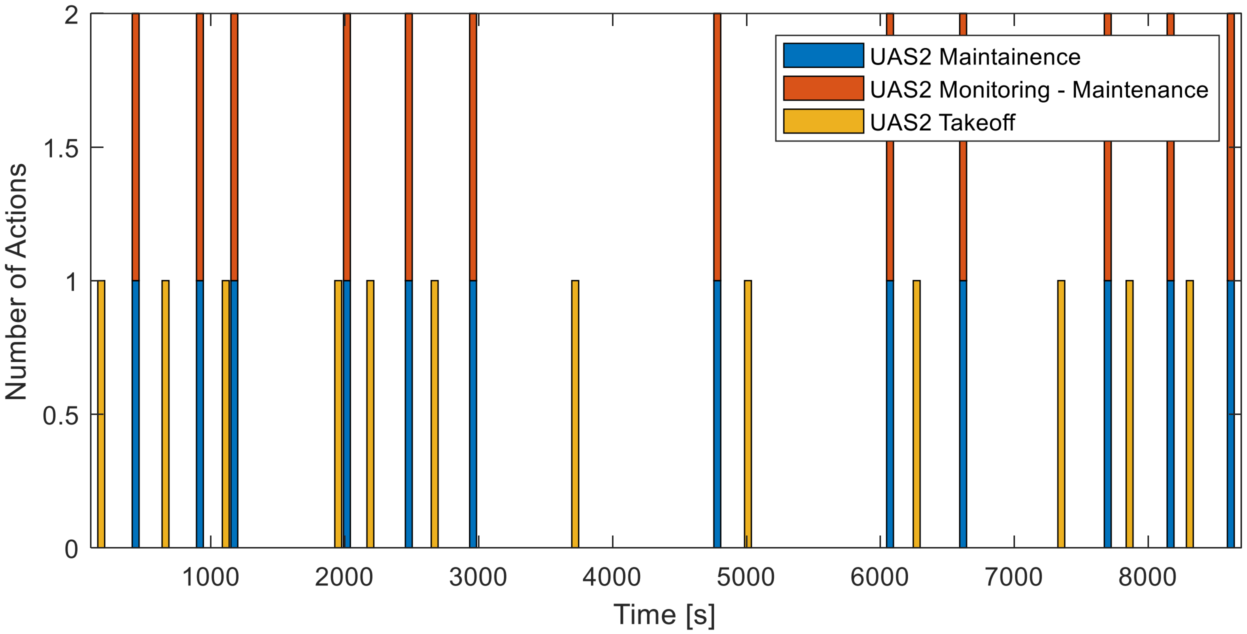


Figure 30. Sequence of Takeoff, Maintenance, and Monitoring-Maintenance actions when there are no delays

For the same allocation of authority and responsibility and for the same vehicle performing the same missions, Figure 31 demonstrates the sequence of actions when delays caused by the limitation in working memory are considered. The first observation is that the Maintenance task and monitoring actions no longer reliably occur simultaneously. The execution and monitoring of Maintenance can occur at different times and not identify critical safety issues, such as lower-than-predicted battery state, until the later time that the monitoring occurs. For example, Figure 31 demonstrates where the vehicle took off between the execution and monitoring of Maintenance.

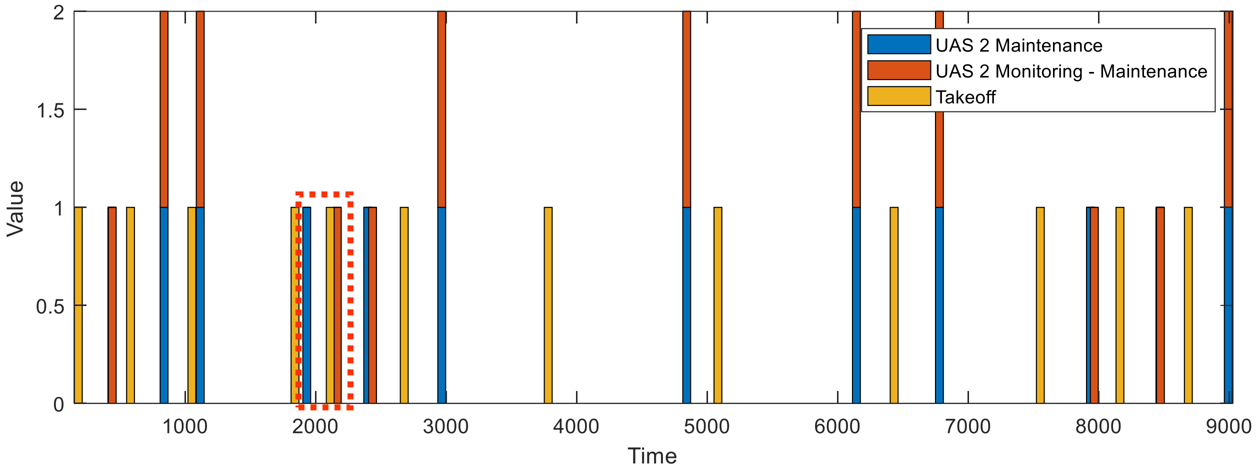


Figure 31. Sequence of Takeoff, Maintenance and its monitoring actions when working memory limits are considered

This underlines the importance of timely monitoring in the context of AAM and other complex systems to ensure their safety. Moreover, considering agent limits, such as working memory limits, is essential when designing complex operations like AAM to avoid delays in monitoring actions and other similar safety-producing behaviors.

7 Conclusions

7.1 Summary of Results

This thesis examined three aspects of the allocation of authority and responsibility between agents in the AAM Concepts of Operation. The first aspect examined principles for allocating authority to execute the taskwork actions needed to perform the Concepts of Operations. This Concepts of Operations, in the 5 vehicle, 30 mission case study examined here requires 5529 instances of taskwork actions, including 4407 checks of status and 571 approvals or clearances. If each action is performed when it is needed, the total time to completion is 13002s, including flight times, and up to 17 actions needed to be executed simultaneously.

This thesis examined the NASA Concept of Operations. This document does not specify the allocation of authority to agents. It does suggest several roles for agents; this discussion was used to create a Role-Based principle for allocating authority. Likewise, it specifies locations where agents will be positioned, and work needs to be performed; this discussion was used to create a Location-Based principle for allocating authority.

These allocations establish very different experiences for their agents: Some are executing many actions simultaneously throughout the day, while others have episodic spikes in the number of actions they need to execute. Some never need to execute more than 1 action at a time, while some need to perform 8 or 9. The information required by each also varies significantly, as does the information transfer needed to provide it.

The second aspect examined the allocation of responsibility for the outcomes of the taskwork. Here, having different agents allocated authority for an action versus responsibility for its outcome triggered a monitoring action assigned to the responsible agent. In the aggregate, the addition of monitoring significantly increased the total number of actions that needed to be executed, and the overall information transfer needed between the agents. This additional load was not always distributed evenly: Some agents needed to perform many simultaneous actions (taskwork and monitoring) and require much more information transfer.

The third aspect examined agent-models which only executed a limited number of actions at any instant; actions beyond their ability to execute were delayed until the earlier actions were completed. As a baseline, this thesis first explored fundamental impacts of agent limits on the number of actions they could perform in the context of a next available allocation that never left an agent idle while another was saturated, but instead allocated agents dynamically to the next available agent. In this case, the ideal timeline requires up to 17 actions to be executed simultaneously; thus, any combination of number of agents and per-agent task limits that cannot handle 17 simultaneous actions delays the total time to complete the day's operations from the ideal 13002s, in the worst case of a single agent limited to a single action at a time, a time to completion of 46411s.

Next, this thesis modelled agents as being limited by their working memory, a well-documented limitation. Each action was examined for the number of information elements it requires holding in working memory, based on assumptions of what information may instead be retrieved from an interface. When these memory limited agents are allocated actions by the Role-Based or Location-Based allocations, several significant effects are found. First, the time to complete the day’s operations increases with the Role-Based allocation and increases more with the Location-Based allocation. This highlights that a particular allocation may be particularly sensitive to agent limits, particularly if it assigns more cognitively demanding actions to one agent, forcing the other agents to wait for them.

Second, monitoring – especially full monitoring – corresponds to substantially more delays. Thus, a CONOPS whose taskwork appears reasonable must also be considered for the extra demands created by monitoring relative to the agents’ fundamental limits.

Finally, the impact of these limits was demonstrated to delay monitoring past the time where it would be most-able to detect safety-concerns. In this case study, for example, monitoring of a maintenance action was assumed to represent a critical opportunity to catch lower-than-predicted battery levels before take-off. However, this monitoring was found to be frequently delayed, in some instances after the vehicle had taken off with too little energy to complete its flight.

7.2 Discussion

This thesis demonstrates that the allocation of authority and responsibility amongst a team of agents is a fundamental decision in the design of a Concepts of Operations. The case study demonstrated how significant an impact the allocation can have, highlighting the value of considering it early in the design. Likewise, the case study demonstrated how computational simulation can be used to assess the performance of an allocation, even as it emerges from the interleaved actions of many agents.

Several factors were examined. First, diminishing returns were observed when increasing the number of agents or their individual capabilities. This finding highlights the need for a balanced approach in deploying agents, as excessively scaling up the number or capability of agents may not necessarily yield proportionate improvements in performance. In scenarios where the requirements exceed the collective capabilities of agents, it was shown that a smaller number of highly capable agents outperformed a larger number of less capable agents.

Second, different principles for allocating authority and responsibility resulted in different experiences of the agents. Some may be lightly loaded, and some so heavily loaded that, when modeling as being limited, the agents had to delay actions to the point that the day's operation needed longer to complete and to the point that safety-critical monitoring did not occur in time.

Third, the impact of monitoring needs to be explicitly considered. In this case study- as is common- the CONOPS implies monitoring should occur, without explicitly detailing ow it will be conducted and in many cases specifying who is monitoring for what. This thesis modeled the underlying phenomenon of allocating responsibility for the outcome of each action as a legal construct that establishes who should monitor which activity. The case study also explored two types of hypothesized monitoring behavior: basic monitoring of the outcome and full monitoring of the process of the activity itself.

The addition of monitoring to the agent's taskload, and the significant increase in information that they must consider and must transfer between each other highlights that monitoring needs to be carefully considered in the design of a Concepts of Operations. Analysis should be conducted not just at a high-level but also with sufficient description of how it should be conducted to assess its likelihood of cognitively overloading the agents, with commensurate aggregate effects on the broader operating ability to complete in a timely and safe manner.

7.3 Future Work

This thesis provides an initial examination of several broader phenomenon that merit further study. First, this thesis applies only a single case study examining 30 missions and 5 vehicles, with the ideal sequence of taskwork always the same in the different test cases. Given how sensitive the emergent effects appear to be to how these actions are inter-leaved, it would also be interesting to re-run the simulations in cases where even the order of the missions is varied within the same overall day's operations, as well as examining how findings scale to larger operations.

Likewise, this case study fixed the description of taskwork of the CONOPS. There may be other sequences of taskwork that can satisfy the CONOPS goals without causing issues such as spikes in the agents' tasks. Thus, future work can examine the co-design of the taskwork and its allocation to agents to execute.

Similarly, this thesis only examined two potential allocations of authority and responsibility, each to a fixed number of agents. Significant differences were found between the two. This suggests the number of agents and the details of allocation can be further explored to identify which engender better performance. For example, instances where simulations find agent models are consistently saturated can be examined in detail to see if, for example, particular actions might be allocated to others r additional agents brought into the same role or location to divvy up the task load.

In this thesis, the only form of teamwork that is modeled is monitoring. Many other forms of teamwork can also be included, especially where they may further add to agents' taskload or the dynamic sequencing and inter-leaving of activities. For example, information transfer was assumed to occur instantly and without requiring any extra action by the transmitting agent. However, actual operations may involve some delay in transmission. Further, agents may need to actively seek, or ask-for-and-transmit information. This would represent further activity on their part, adding to their load; any delay in sending or receiving information would then be another contributor to the agent-level delays that can roll-up to impact the mission and organization.

Finally, the models of monitoring applied here illustrate ow monitoring can take many forms. Monitoring has not been studied sufficiently to understand cognitive mechanisms by which it is conducted; the approach taken here can be expanded to explore more detailed models of how individuals monitor each other.

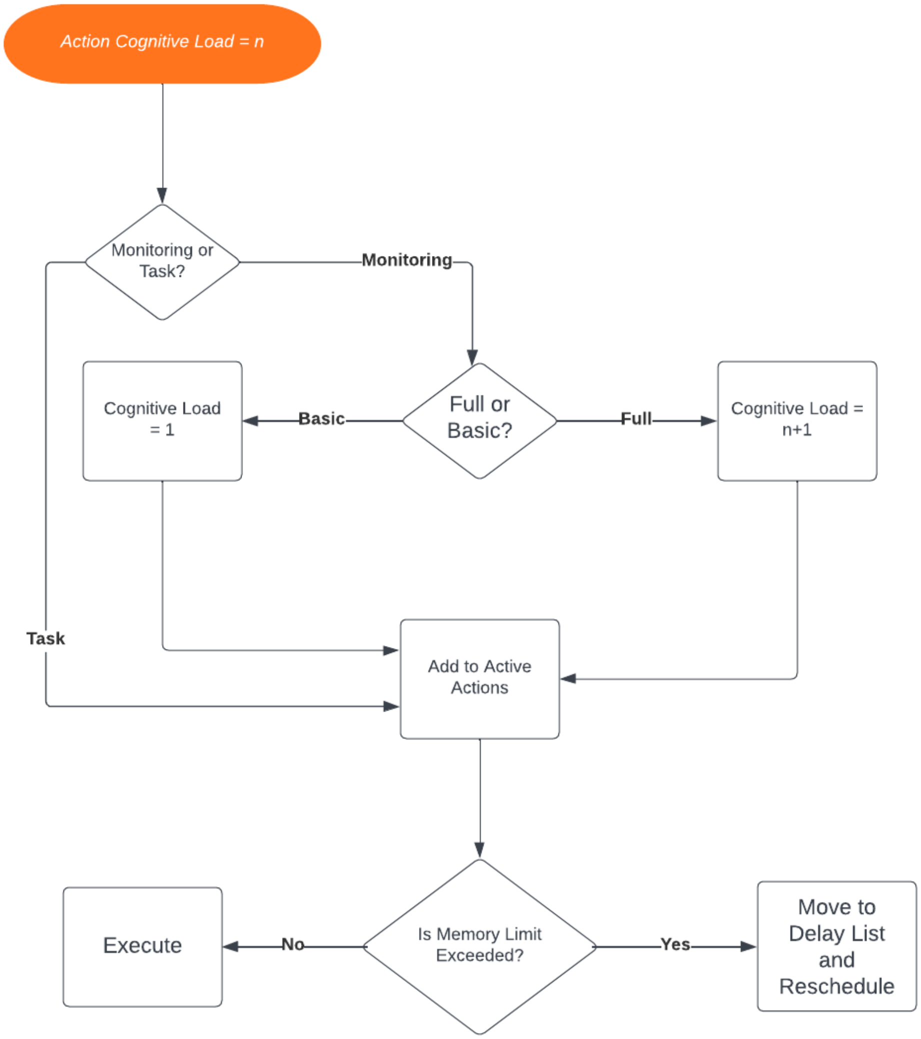
Likewise, monitoring may be better viewed as part of a team activity, in which agents collectively review their shared actions. Again, the computational framework employed here ca easily be extended to layer models of team dynamics into the simulations to see how they may jointly review their collective efforts.

Appendix

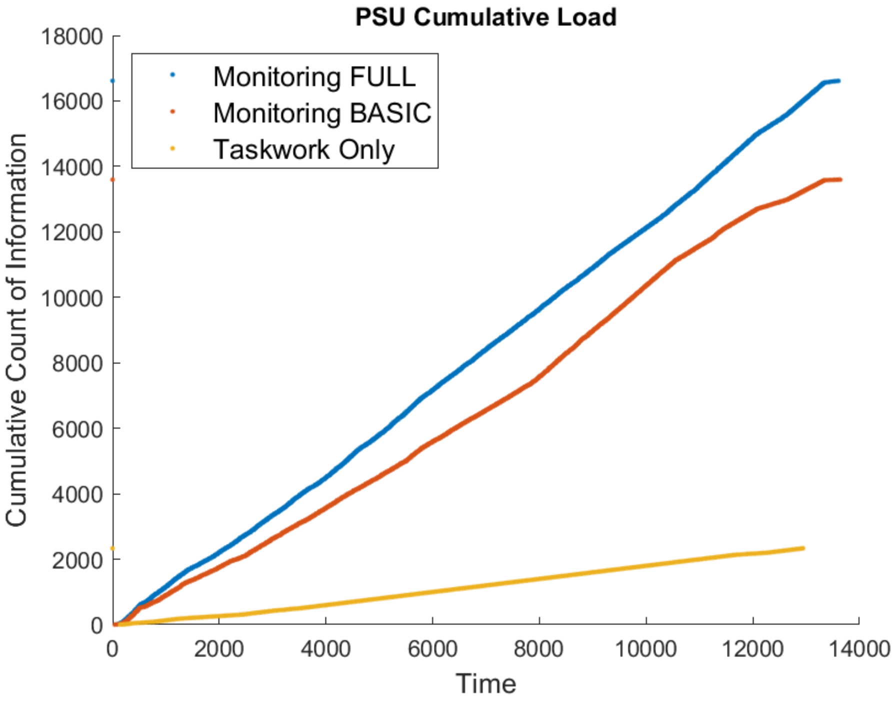
This appendix contains information about the logic used by the memory-limited agent, how it adds information to its working memory and how this agent decides to take on another task or delay the task due to saturation. This flow chart also includes how the monitoring actions' load on the working memory of the agent depends on the extent to which the agent is monitoring.

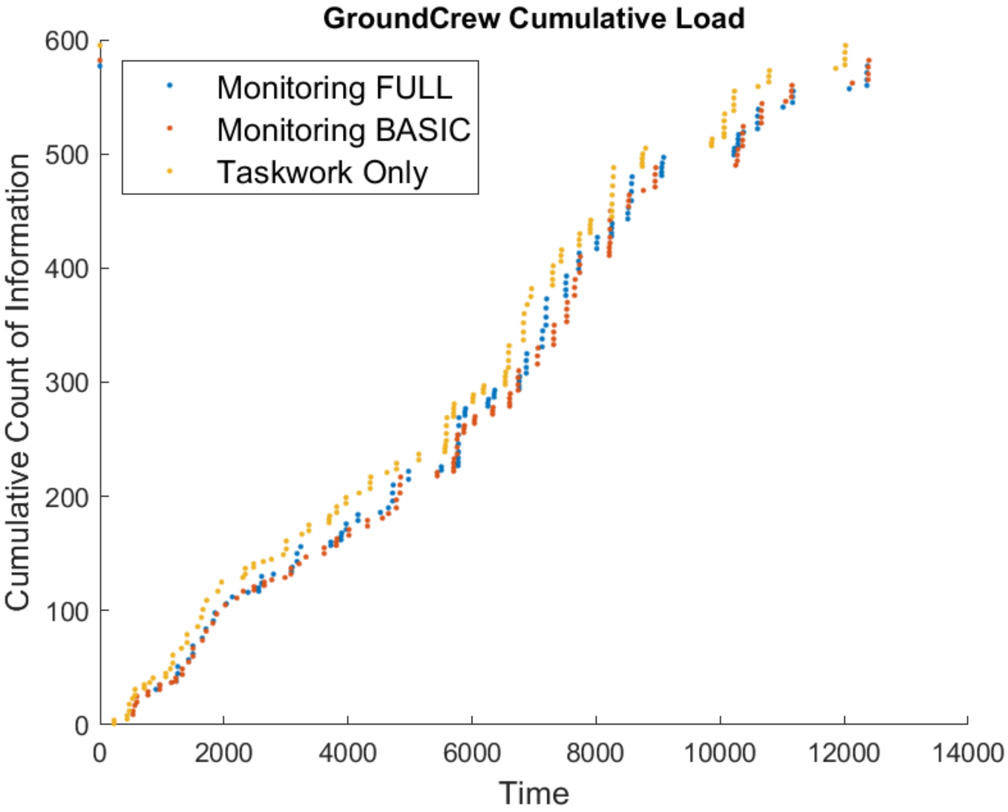
Sections B and C contain the cumulative load and count timeline of saturation of all the agents, since some of this data was not presented for all the agents present and only for representative samples to illustrate the different experiences agents have.

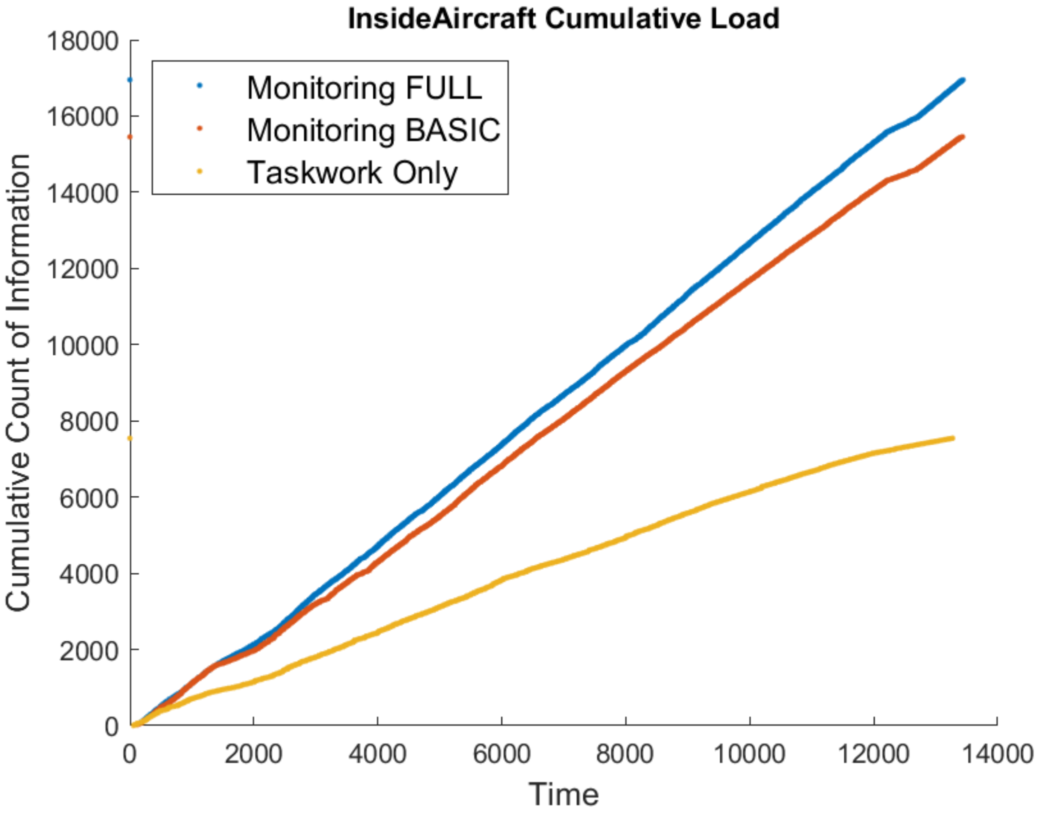
A. Working-Memory-Limited Agent Decision Flow Chart

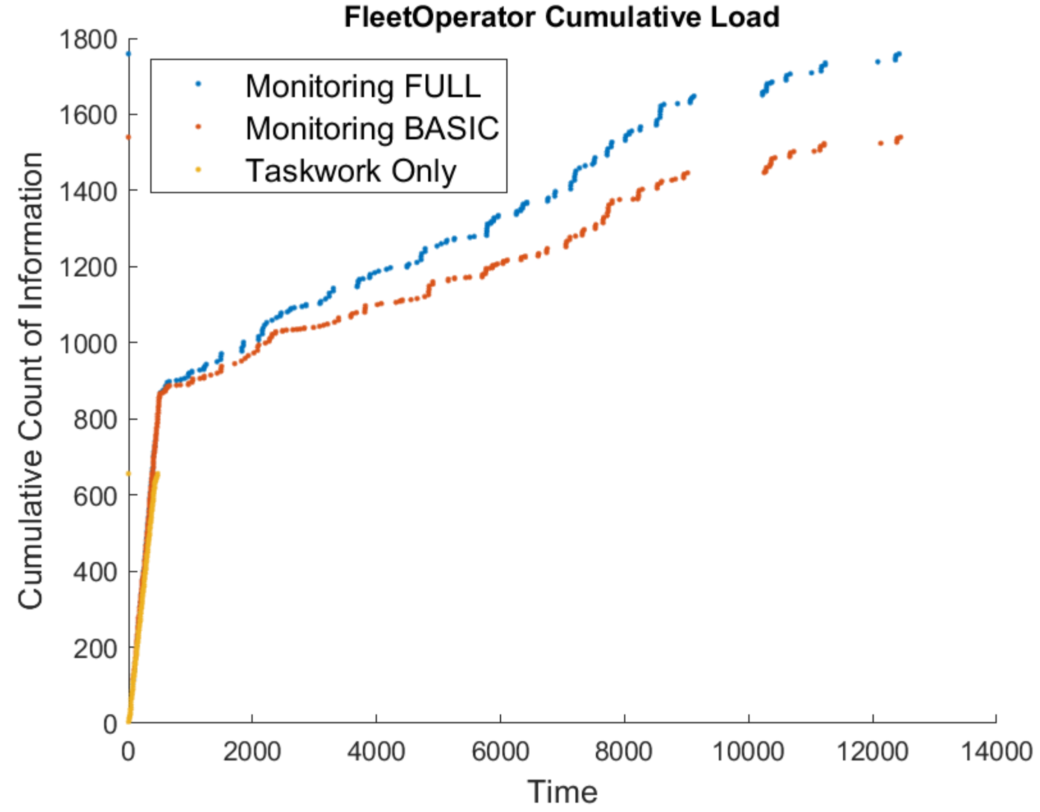


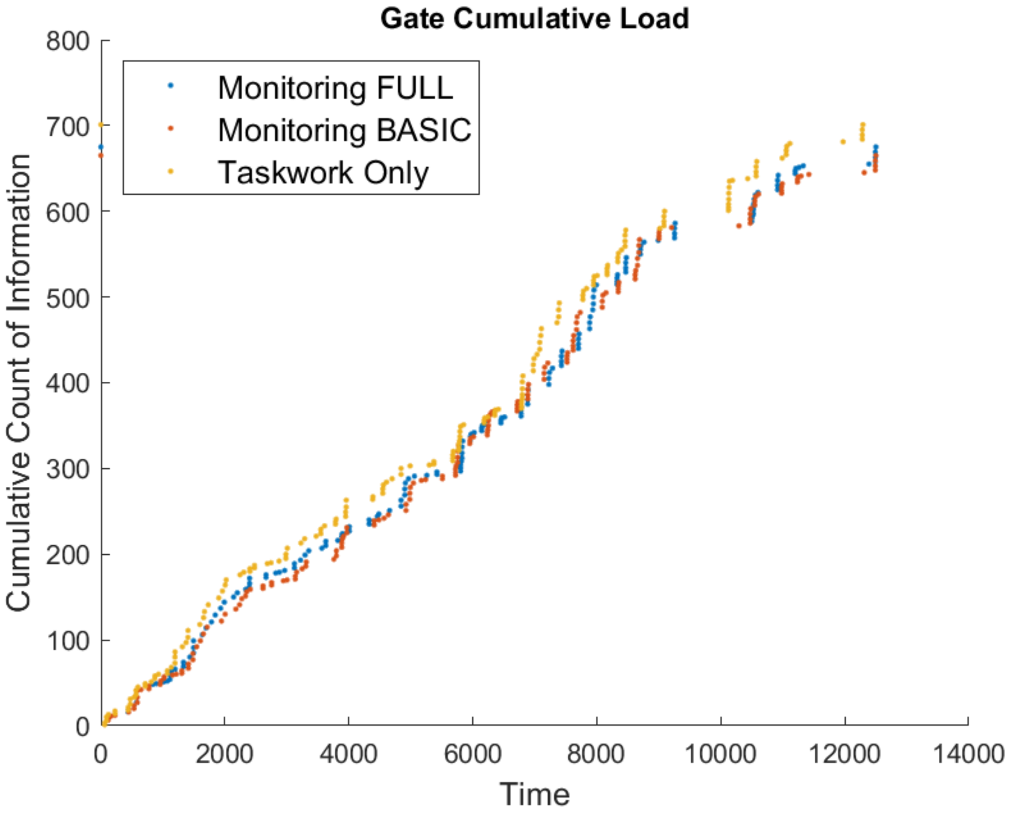
B. Count of Cumulative Load For All Agents

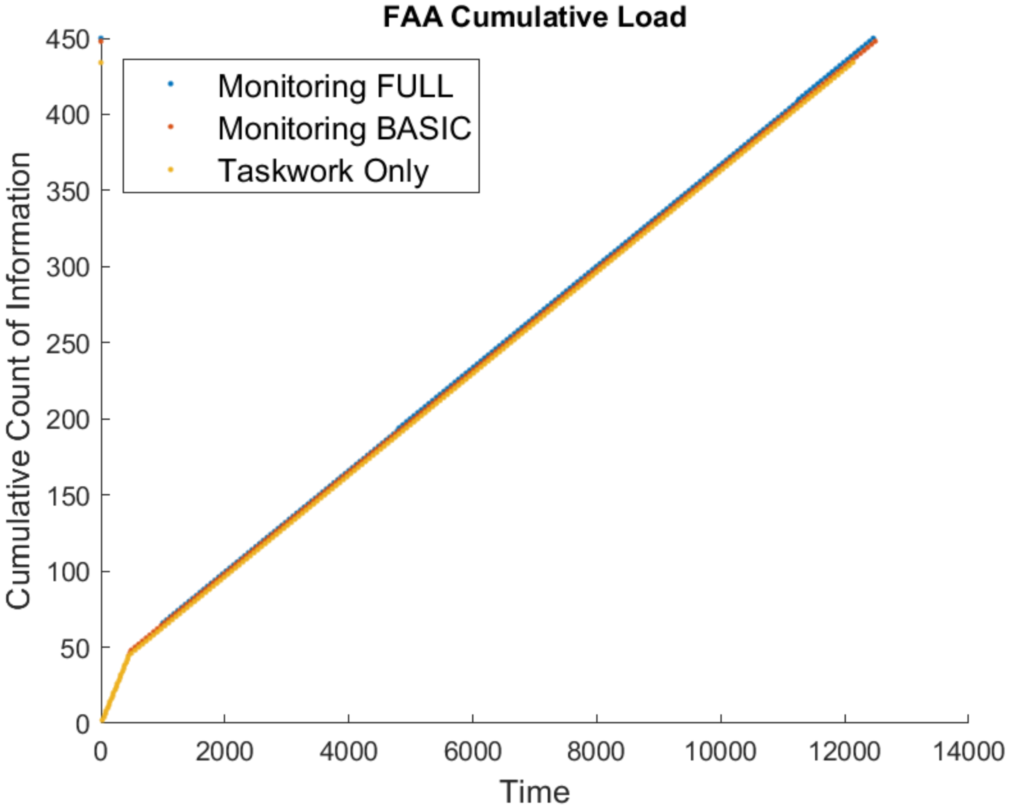


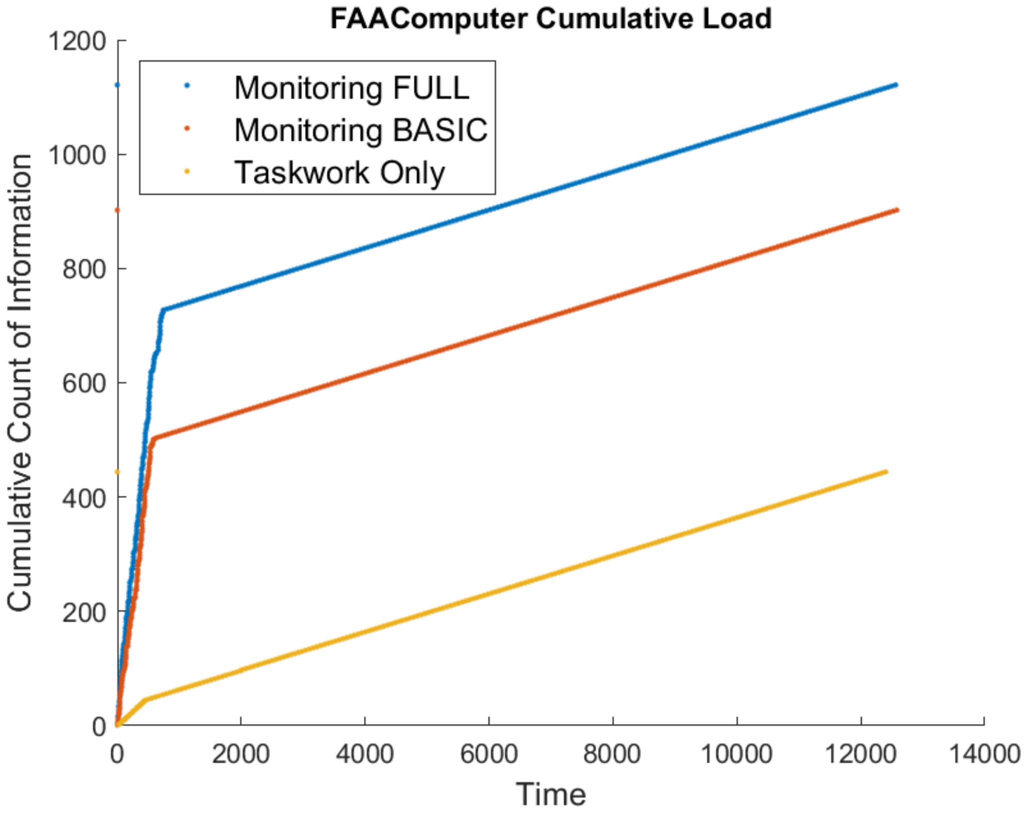


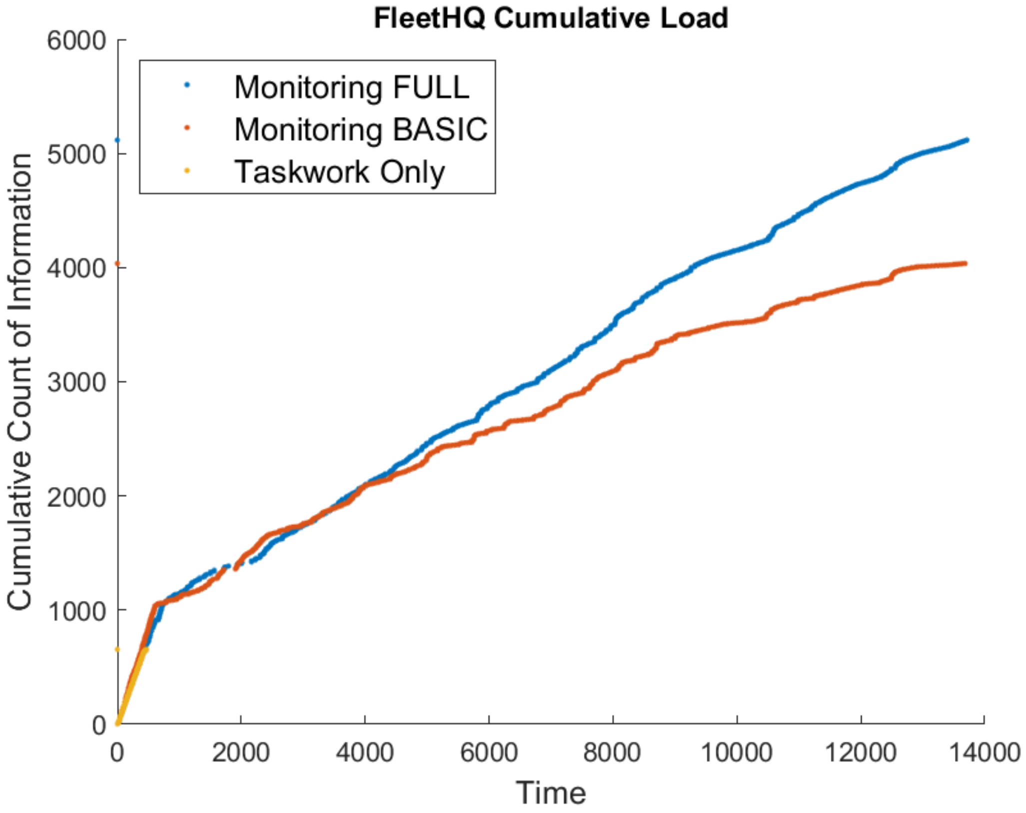

InsideAircraft Cumulative Load
Monitoring FULL
Monitoring BASIC
Taskwork Only
Cumulative Count of Information
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
0
2000
4000
6000
8000
10000
12000
14000
Time

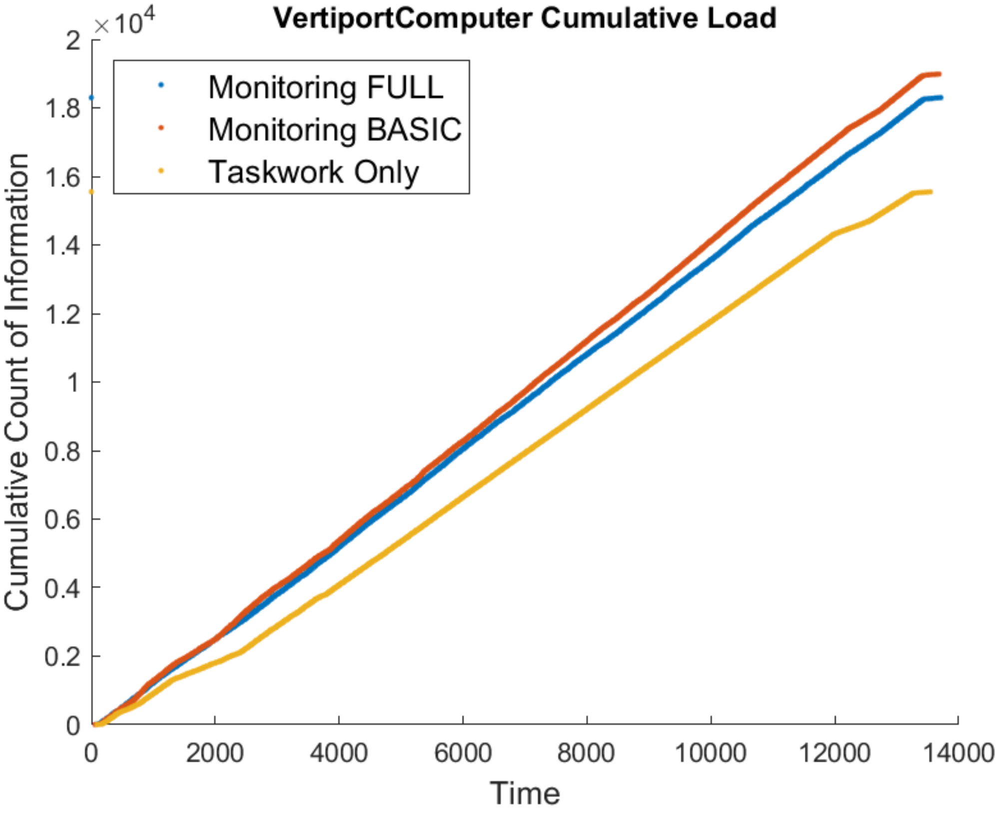

FleetOperator Cumulative Load
Monitoring FULL
Monitoring BASIC
Taskwork Only
Cumulative Count of Information
1800
1600
1400
1200
1000
800
600
400
200
0
0
2000
4000
6000
8000
10000
12000
14000
Time

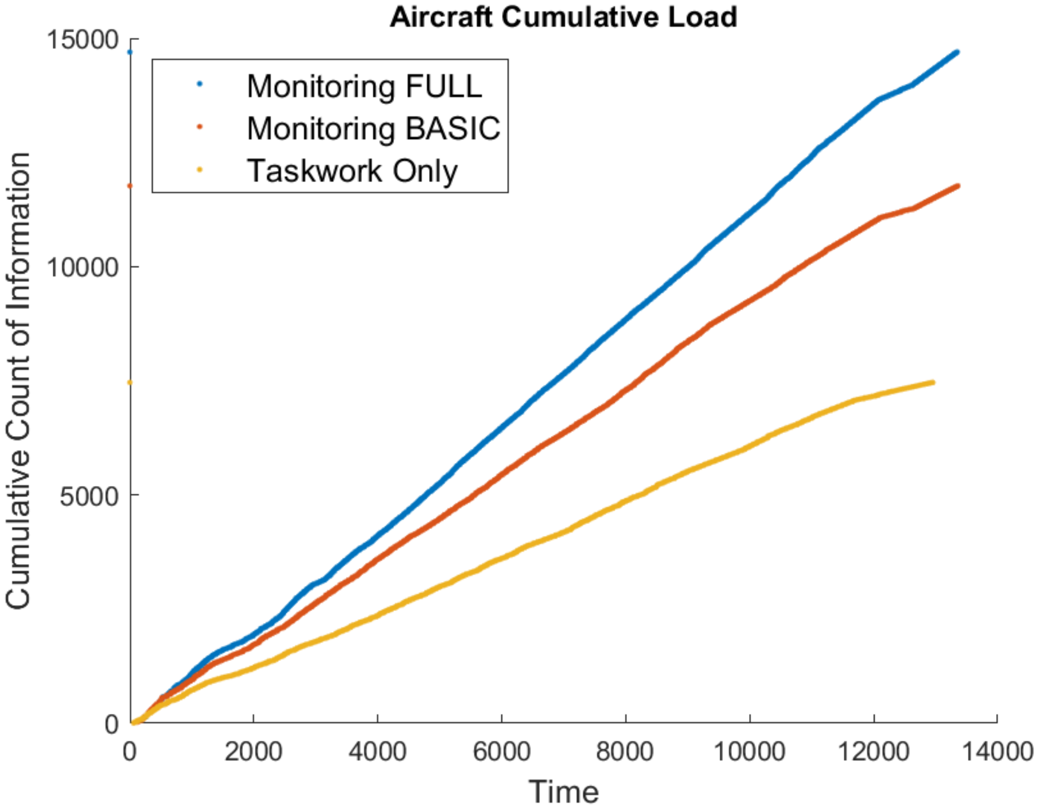

Gate Cumulative Load
Monitoring FULL
Monitoring BASIC
Taskwork Only
Cumulative Count of Information
800
700
600
500
400
300
200
100
0
0
2000
4000
6000
8000
10000
12000
14000
Time

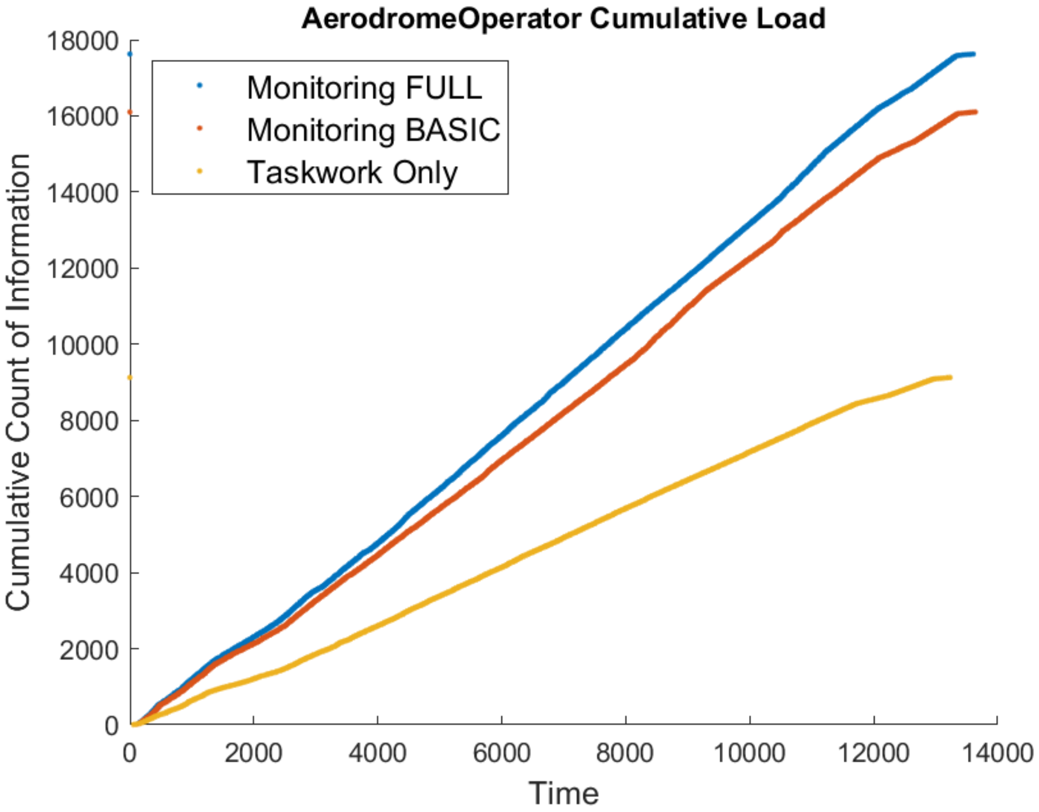

FAA Cumulative Load
Monitoring FULL
Monitoring BASIC
Taskwork Only
Cumulative Count of Information
450
400
350
300
250
200
150
100
50
0
0
2000
4000
6000
8000
10000
12000
14000
Time


FAAComputer Cumulative Load
Monitoring FULL
Monitoring BASIC
Taskwork Only
Cumulative Count of Information
0
200
400
600
800
1000
1200
0
2000
4000
6000
8000
10000
12000
14000
Time

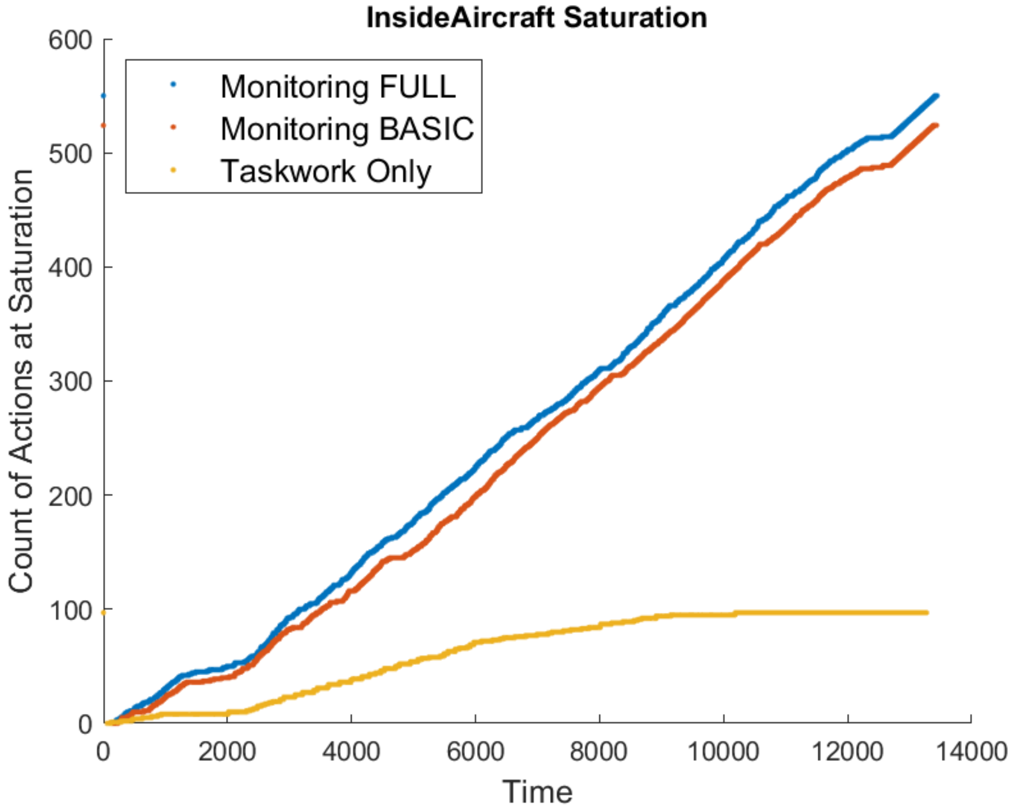

FleetHQ Cumulative Load
Monitoring FULL
Monitoring BASIC
Taskwork Only
Cumulative Count of Information
0
1000
2000
3000
4000
5000
6000
0
2000
4000
6000
8000
10000
12000
14000
Time

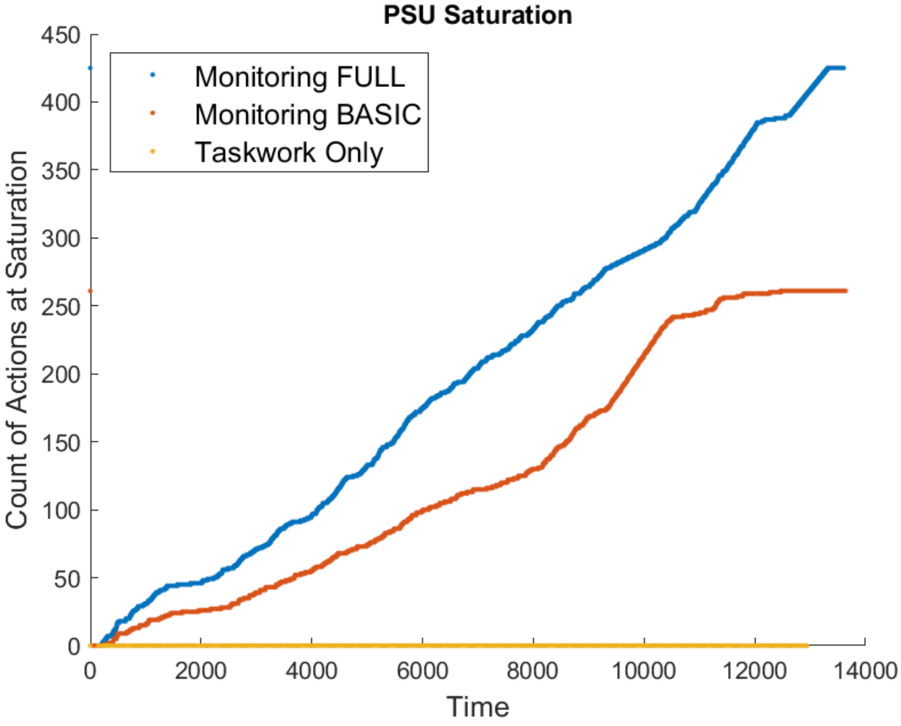

VertiportComputer Cumulative Load
×10^4
Monitoring FULL
Monitoring BASIC
Taskwork Only
Cumulative Count of Information
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
2000
4000
6000
8000
10000
12000
14000
Time

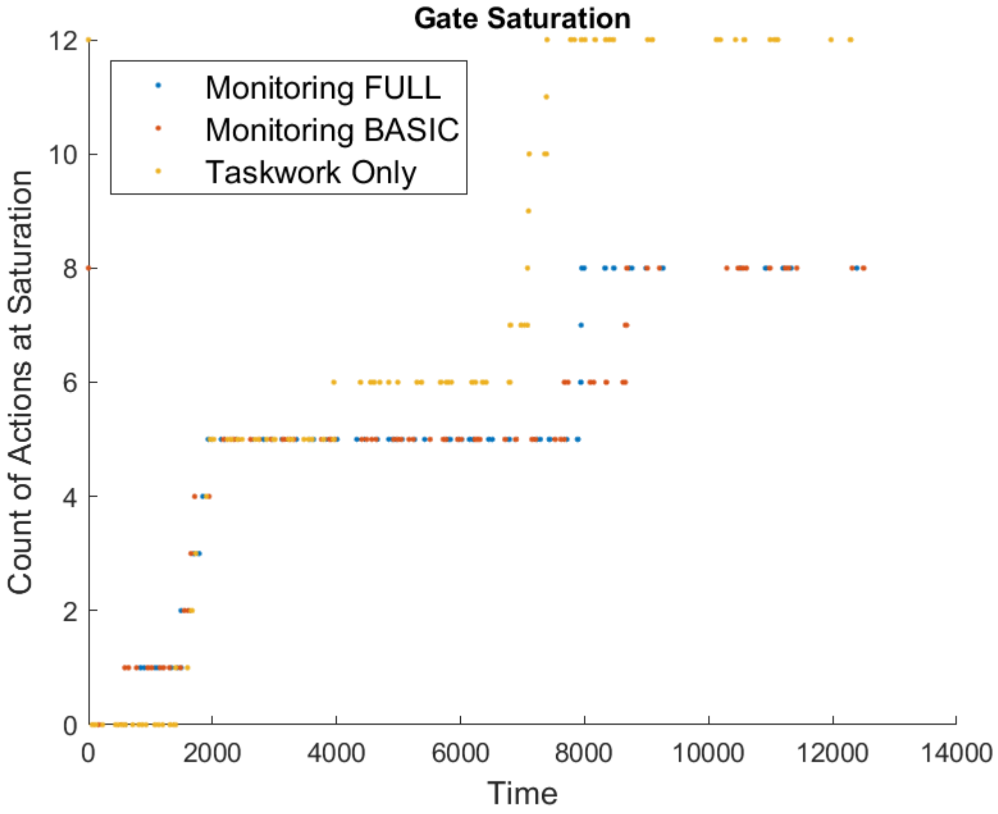

Aircraft Cumulative Load
Monitoring FULL
Monitoring BASIC
Taskwork Only
Cumulative Count of Information
15000
10000
5000
0
0
2000
4000
6000
8000
10000
12000
14000
Time

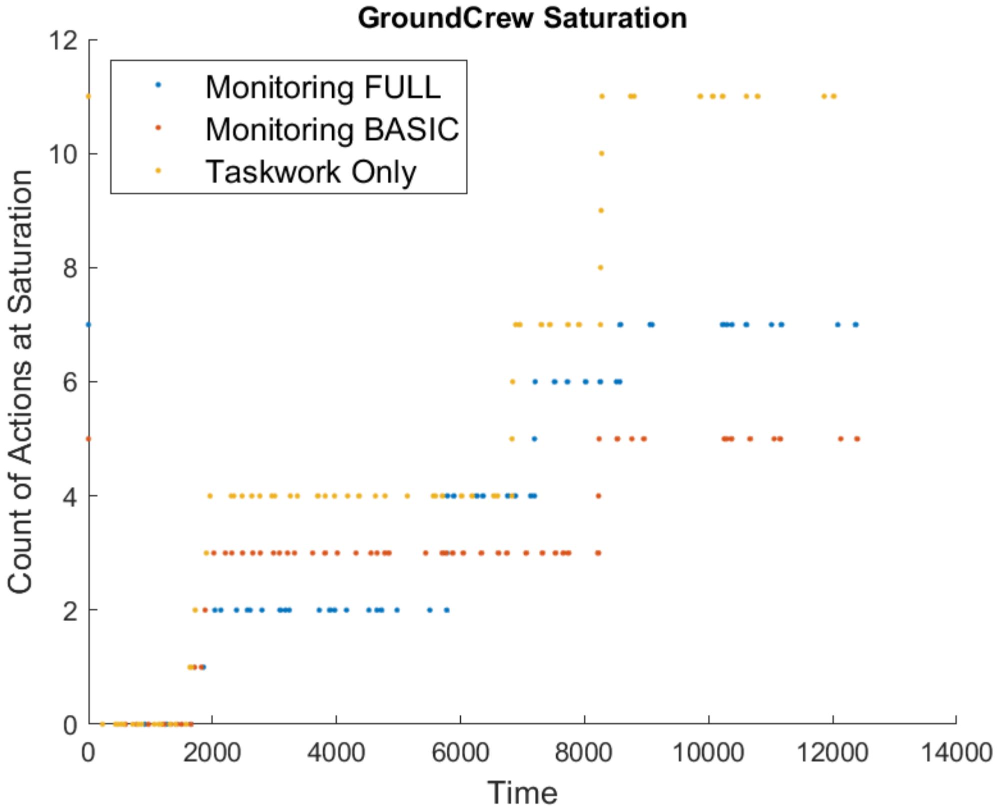

AerodromeOperator Cumulative Load
Monitoring FULL
Monitoring BASIC
Taskwork Only
Cumulative Count of Information
0
2000
4000
6000
8000
10000
12000
14000
16000
18000
Time
0
2000
4000
6000
8000
10000
12000
14000

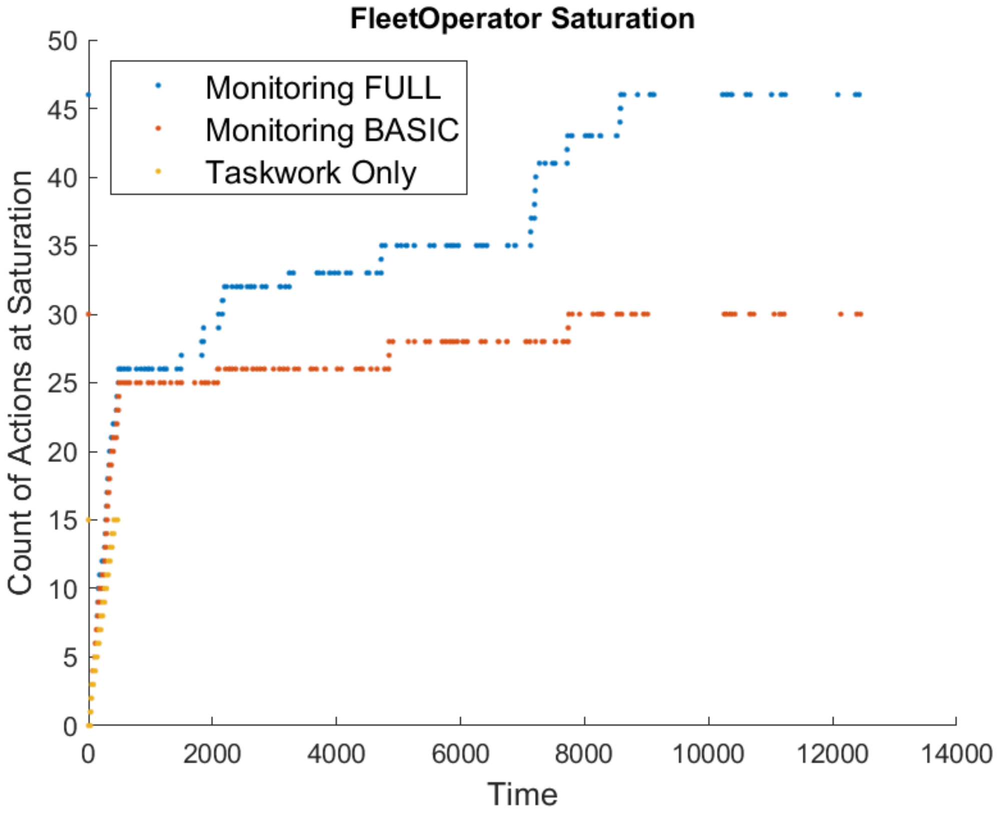
C. Count of Saturation for All Agents

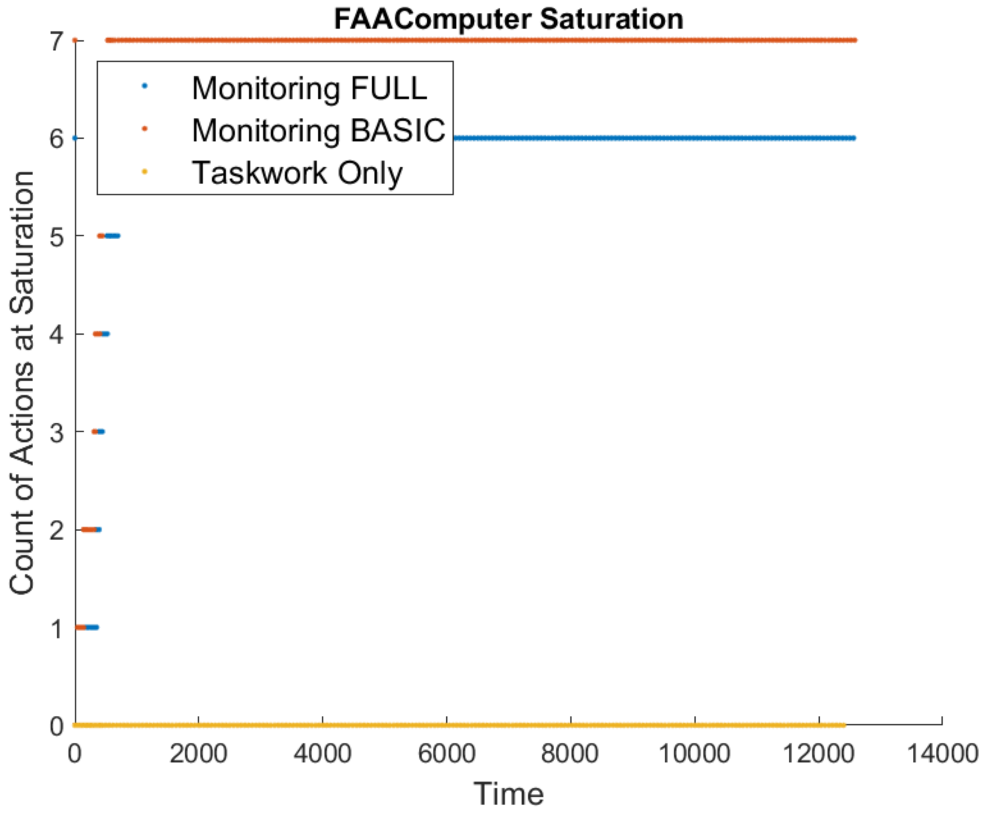


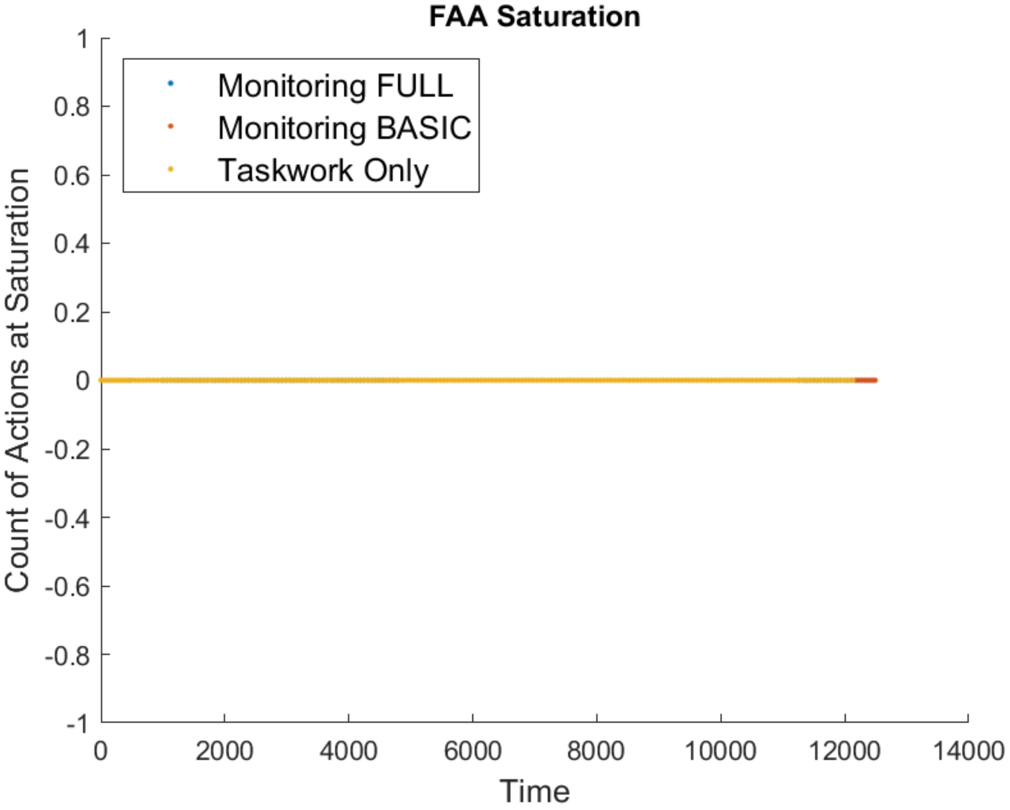


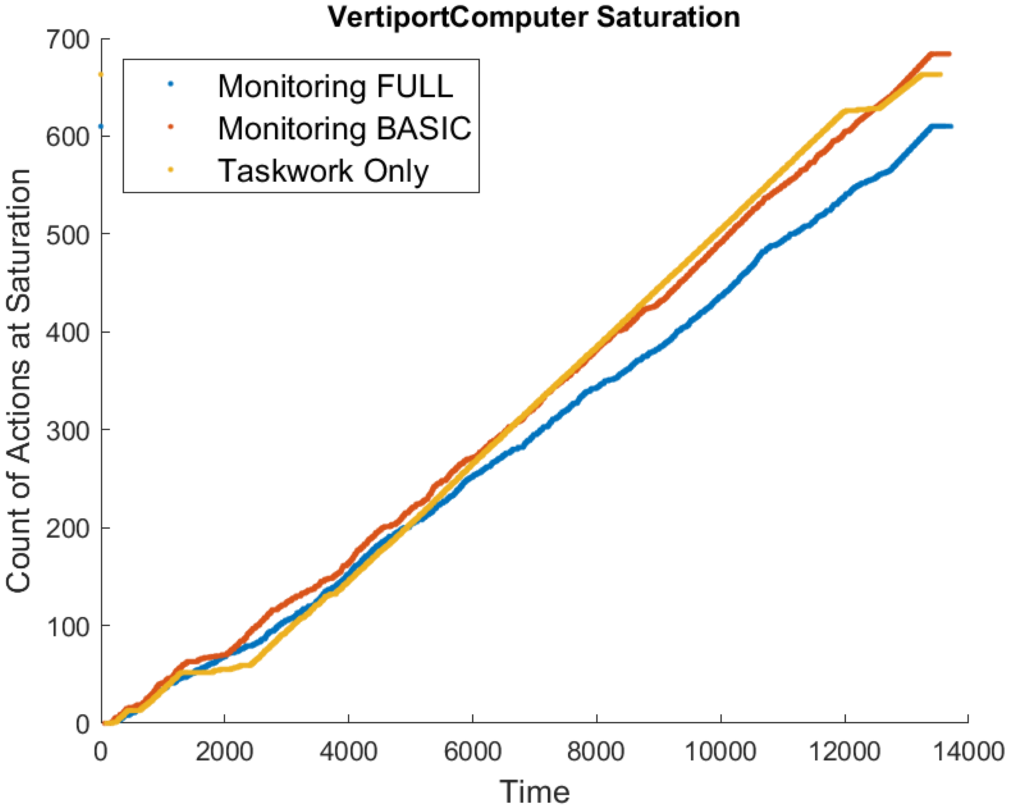

Gate Saturation
Monitoring FULL
Monitoring BASIC
Taskwork Only
Count of Actions at Saturation
Time
0
2
4
6
8
10
12
0
2000
4000
6000
8000
10000
12000
14000

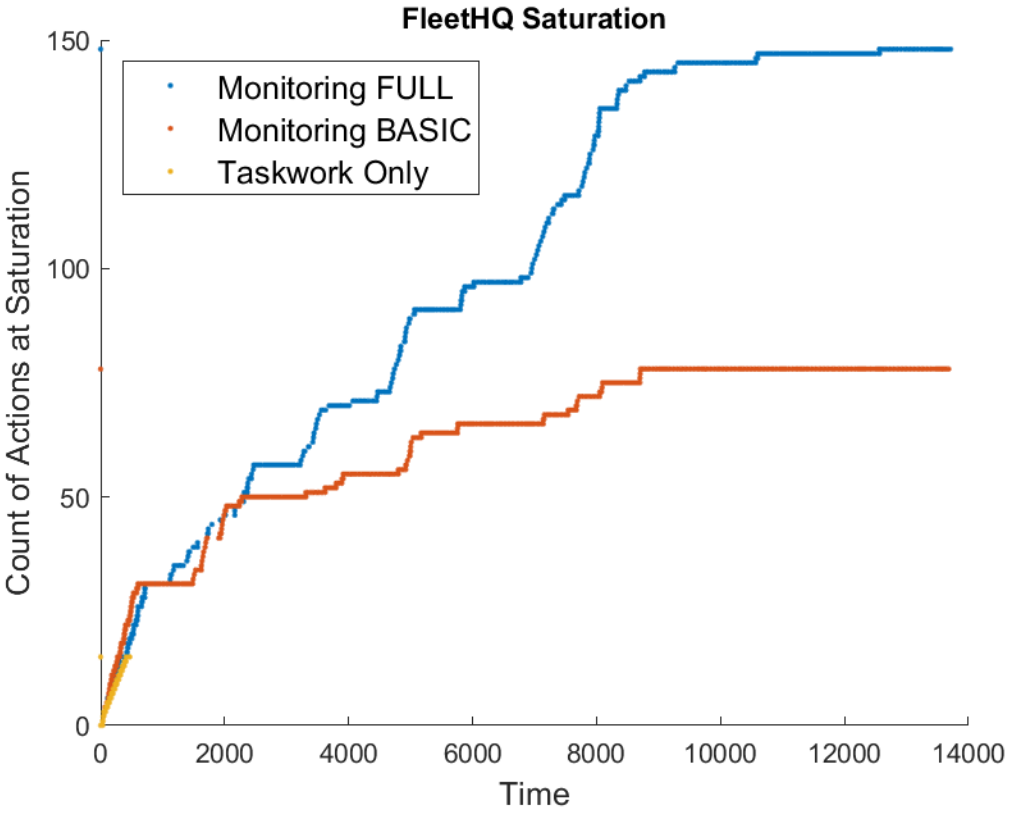

GroundCrew Saturation
Monitoring FULL
Monitoring BASIC
Taskwork Only
Count of Actions at Saturation
Time
0
2
4
6
8
10
12
0
2000
4000
6000
8000
10000
12000
14000

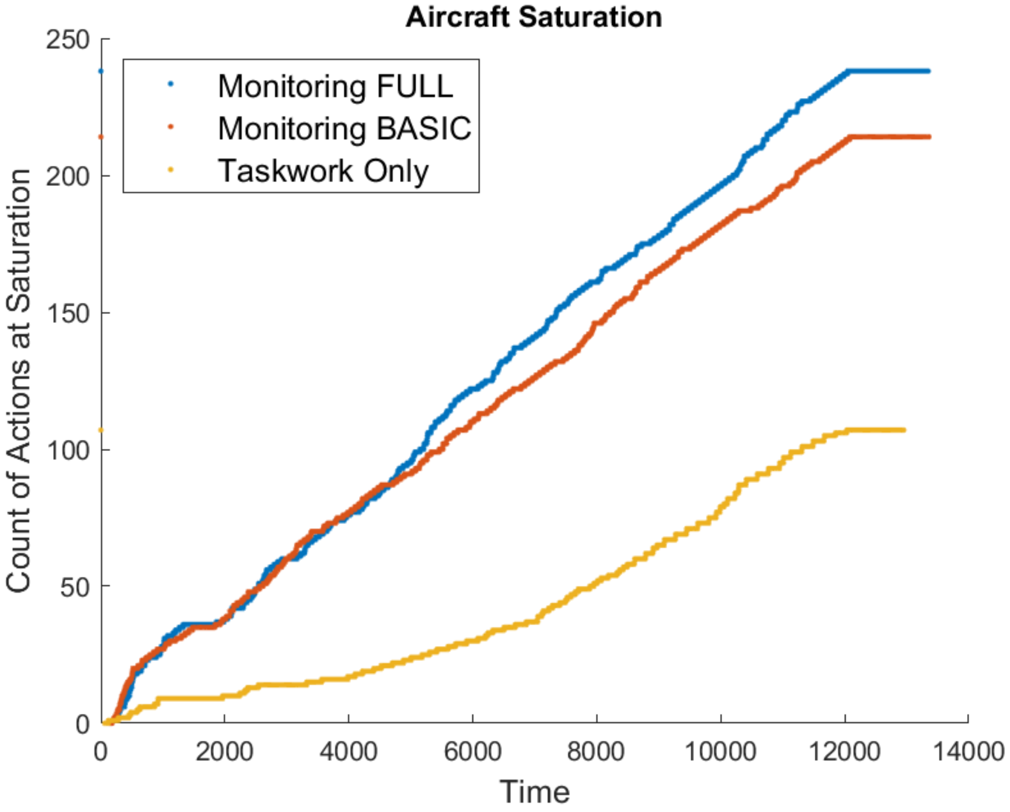

FleetOperator Saturation
Monitoring FULL
Monitoring BASIC
Taskwork Only
Count of Actions at Saturation
0
5
10
15
20
25
30
35
40
45
50
Time
0
2000
4000
6000
8000
10000
12000
14000

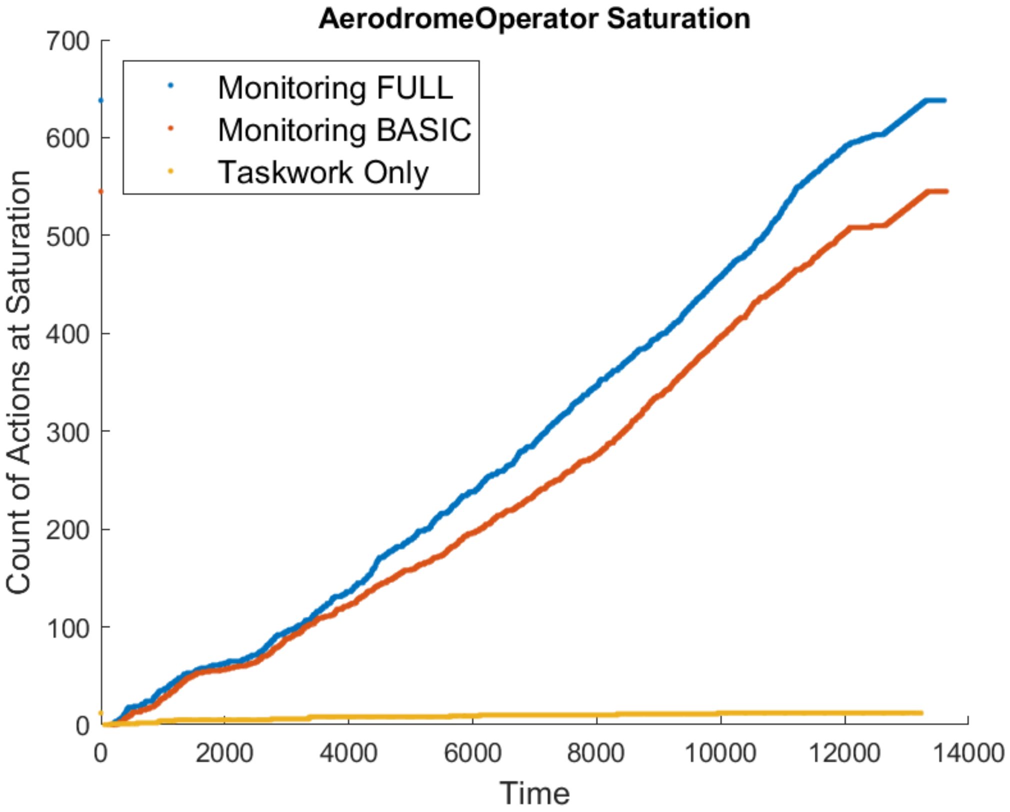

FAAComputer Saturation
Monitoring FULL
Monitoring BASIC
Taskwork Only
Count of Actions at Saturation
0
1
2
3
4
5
6
7
Time
0
2000
4000
6000
8000
10000
12000
14000


FAA Saturation
Monitoring FULL
Monitoring BASIC
Taskwork Only
Count of Actions at Saturation
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
2000
4000
6000
8000
10000
12000
14000
Time


VertiportComputer Saturation
Monitoring FULL
Monitoring BASIC
Taskwork Only
Count of Actions at Saturation
700
600
500
400
300
200
100
0
0
2000
4000
6000
8000
10000
12000
14000
Time


FleetHQ Saturation
Monitoring FULL
Monitoring BASIC
Taskwork Only
Count of Actions at Saturation
150
100
50
0
0
2000
4000
6000
8000
10000
12000
14000
Time


Aircraft Saturation
Monitoring FULL
Monitoring BASIC
Taskwork Only
Count of Actions at Saturation
250
200
150
100
50
0
0
2000
4000
6000
8000
10000
12000
14000
Time


AerodromeOperator Saturation
Monitoring FULL
Monitoring BASIC
Taskwork Only
Count of Actions at Saturation
Time
0
100
200
300
400
500
600
700
0
2000
4000
6000
8000
10000
12000
14000

References

[1] R. Goyal, C. Reiche, C. Fernando, and A. Cohen, "Advanced Air Mobility: Demand Analysis and Market Potential of the Airport Shuttle and Air Taxi Markets," *Sustainability,* vol. 13, no. 13, p. 7421, 2021. [Online]. Available: https://www.mdpi.com/2071-1050/13/13/7421.

[2] Brian P Hill *et al.*, "UAM Vision Concept of Operations (ConOps) UAM Maturity Level (UML) 4NASA Technical Reports Server : NTRS," [Washington, D.C.] : [U.S. National Aeronautics and Space Administration], NTRS, 20205011091, December 2, 2020 2020. [Online]. Available: https://ntrs.nasa.gov/citations/20205011091

[3] "IEEE Guide for Information Technology - System Definition - Concept of Operations (ConOps) Document," *IEEE Std 1362-1998,* pp. 1- 24, 12-22-1998 1998, doi: doi: 10.1109/IEEESTD.1998.89424.

[4] K. Feigh and A. Pritchett, "Function Allocation between Human and Automation and Between Air and Ground," in *Encyclopedia of Aerospace Engineering*, pp. 1-12.

[5] M. T. Older, P. E. Waterson, and C. W. Clegg, "A critical assessment of task allocation methods and their applicability," *Ergonomics,* vol. 40, no. 2, pp. 151-171, 1997/02/01 1997, doi: 10.1080/001401397188279.

[6] K. M. Feigh and A. R. Pritchett, "Requirements for Effective Function Allocation:A Critical Review," *Journal of Cognitive Engineering and Decision Making,* vol. 8, no. 1, pp. 23-32, 2014, doi: 10.1177/1555343413490945.

[7] A. R. Pritchett, S. Y. Kim, and K. M. Feigh, "Modeling Human–Automation Function Allocation," *Journal of Cognitive Engineering and Decision Making,* vol. 8, no. 1, pp. 33-51, 2014, doi: 10.1177/1555343413490944.

[8] A. Villa and A. Pritchett, "Computational Modeling of Monitoring within Human-Autonomy Teams with Distributed Information," in *AIAA SCITECH 2022 Forum*.

[9] A. R. Pritchett, S. Y. Kim, and K. M. Feigh, "Measuring Human-Automation Function Allocation," *Journal of Cognitive Engineering and Decision Making,* vol. 8, no. 1, pp. 52-77, 2014, doi: 10.1177/1555343413490166.

[10] G. A. Miller, "The magical number seven, plus or minus two: Some limits on our capacity for processing information," *Psychological Review,* vol. 63, no. 2, pp. 81-97, 1956, doi: 10.1037/h0043158.

[11] S. R. Schmidt, "2.09 - Distinctiveness and Memory: A Theoretical and Empirical Review," in *Learning and Memory: A Comprehensive Reference*, J. H. Byrne Ed. Oxford: Academic Press, 2008, pp. 125-144.

[12] W. G. Chase and H. A. Simon, "Perception in chess," *Cognitive Psychology,* vol. 4, no. 1, pp. 55-81, 1973/01/01/ 1973, doi: https://doi.org/10.1016/0010-0285(73)90004-2.

[13] K. Beach, "Becoming a bartender: The role of external memory cues in a work-directed educational activity," *Applied Cognitive Psychology,* vol. 7, no. 3, pp. 191-204, 1993, doi: https://doi.org/10.1002/acp.2350070304.

[14] D. Helbing and S. Balietti, "How to Do Agent-Based Simulations in the Future: From Modeling Social Mechanisms to Emergent Phenomena and Interactive Systems Design," *Urban Economics & Regional Studies eJournal,* 2013.

[15] A. Villa, "Computational Modeling of Monitoring within Human-Autonomy Teams with Novel Delegations of Authority and Responsibility," MS Department of Aerospace Engineering, Pennsylvania State University, 2022.