Proceedings of the 2006 Winter Simulation Conference
L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, eds.

THE MIGRATION OF A COLLABORATIVE UAV TESTBED INTO THE FLAMES SIMULATION ENVIRONMENT

William M. Niland

Institute for Scientific Research, Inc.
119 Roush Circle, Industrial Park Road
Fairmont, WV 26505 U.S.A.

ABSTRACT

Future generations of Unmanned Air Vehicles (UAVs) will posses the ability to autonomously cooperate in teams to meet various military objectives. This is the focus of research at the U.S. Air Force Research Laboratory, which developed MultiUAV, a research tool used to simulate UAV teams collaborating autonomously in various mission scenarios. In a previous effort, Suppression of Enemy Air Defense (SEAD) mission capabilities were developed for MultiUAV and tested against Joint Integrated Mission Model (JIMM) scenarios. This architecture provided an accurate battlefield environment for small SEAD studies. To truly stress the collaborative algorithms in MultiUAV and build complex SEAD missions, a connection to a streamlined and user-friendly software tool was required. The FLexible Analysis Modeling and Exercise System (FLAMES) software has been chosen as the JIMM replacement. This paper describes the MultiUAV/FLAMES integration effort and provides results to illustrate MultiUAV conducting complex SEAD missions using battlefield information provided by FLAMES.

1 INTRODUCTION

Advances in technology have made it possible to field UAVs for dangerous military operations, thus allowing the removal of manned assets in certain threat environments. An area of increasing research for military planners is developing strategies that allow UAV teams to carry-out missions with little human intervention. Generally, many of the mission components can be planned in advance, but battlefield knowledge is limited since perceived enemy states and target positions change through the duration of the campaign. If the UAVs can autonomously collaborate during changes in battlefield knowledge, their combined resources can be utilized to achieve complex mission goals (Banda 2002). These sophisticated high-risk missions could be SEAD, Persistent Area Denial (PAD), or Combat Intelligence, Surveillance, and Reconnaissance (Combat-ISR) (Schulz et al. 2003).

Researchers at the U.S. Air Force Research Laboratory (AFRL) have written software to study various UAV teaming strategies. MultiUAV is a simulation environment developed to host a team of UAV models, each containing on-board cooperative control algorithms that dynamically decide each vehicle's tasking assignment as the battlefield changes (Rasmussen et al. 2002). To reduce the amount of effort required to add new capabilities, such as new cooperative control approaches, MultiUAV is based in the MATLAB/Simulink modeling environment.

Through a previous effort, MultiUAV was enhanced to support SEAD tasking assignments (Niland et al. 2005). This required a higher and faster flying Unmanned Combat Air Vehicle (UCAV) model to be added in MultiUAV. It also required MultiUAV to disable its internal target models and connect to Integrated Air Defense System (IADS) targets provided by JIMM. The IADS scenario supplies MultiUAV with entity information characteristic during a SEAD mission, such as enemy radar emissions and Surface-to-Air Missile (SAM) state data. The main drawback to JIMM was its cumbersome database configuration requirements. When JIMM scenario enhancements were required, such as moving the placement of a SAM site, a developer knowledgeable with JIMM database configuration files was consulted. This in turn impeded task assignment studies in MultiUAV.

Current research at AFRL has used FLAMES to contribute to IADS models built by the National Air and Space Intelligence Center (NASIC) (Panson 2004). FLAMES is a suite of software tools that allows the user to build, execute, and analyze varying levels of military scenarios. FLAMES' greatest asset is the reduction in time required for scenario development and modification. Since IADS models are developed in this environment by AFRL, the MultiUAV interface to JIMM is replaced in this effort with a connection to the FLAMES software. By utilizing these IADS scenarios, researchers can easily modify the threat layout to challenge the MultiUAV cooperative control algorithms.

A brief overview of the MultiUAV/JIMM environment is provided before a more detailed description of the AFRL-developed IADS models are discussed. The modifications required in both MultiUAV and FLAMES for interoperability follows. To illustrate more complex SEAD behaviors, a MultiUAV/FLAMES scenario is presented to show a successful integration effort. Lastly, conclusions and future work with the simulation environment are mentioned.

2 EXISTING MULTIUAV/JIMM SIMULATION

As previously stated, AFRL developed the MultiUAV research tool to implement and evaluate various cooperative control strategies for teams of UAVs. Due to its MATLAB/Simulink architecture, MultiUAV can be studied and contributed to by a broad range of researchers.

The public release version of MultiUAV can simulate a maximum of eight vehicles prosecuting a group of ten targets (MultiUAV 2006). Each vehicle contains an identical set of Embedded Flight Software (EFS) components and vehicle dynamics. Each vehicle's EFS contains a cooperation manager, which is the focus of most research. The cooperation manager is responsible for allocating mission tasks among the vehicles during discovery of new threats and changes to threat states. A typical mission unfolds by deliberately flying a team of UAVs into a known threat area. Once a team member detects a threat, notification is sent to all other vehicles. At that time, the cooperation manager on each vehicle decides who is in the best position to perform the desired tasks on the threat, such as classification, attack, or kill verification. The cooperation manager duplication on each vehicle limits the amount the inter-vehicle communication required while planning.

Since the internal target models in MultiUAV possessed no defense capability, they were insufficient for SEAD studies. Generally the threats found in a SEAD mission are SAM sites, which emit radar emissions while tracking enemies and fire upon enemies once they breach a certain area. In order to provide a realistic battlefield environment, the internal MultiUAV target models were disabled. MultiUAV was then interfaced to JIMM via the High Level Architecture (HLA) (Stolarik et al. 2004) to supply higher fidelity threats, allowing investigation into UAV task allocation approaches during SEAD missions.

Figure 1 and Figure 2 show a simple SEAD scenario in JIMM. Three vehicles, labeled 1, 2, and 3, are deliberately flown into the threat area of SAM site A. For this simple mission, an attack task and Battle Damage Assessment (BDA) task must be assigned to any detected threat. This example shows UCAV1 performing the attack in Figure 1. Notice the released weapon is circled. Later in the mission, illustrated in Figure 2, UCAV2 has oriented its Synthetic Aperture Radar (SAR) sensor over the attacked threat for BDA. This mission completes successfully and both aircraft rejoin UCAV3 along their predetermined search pattern.

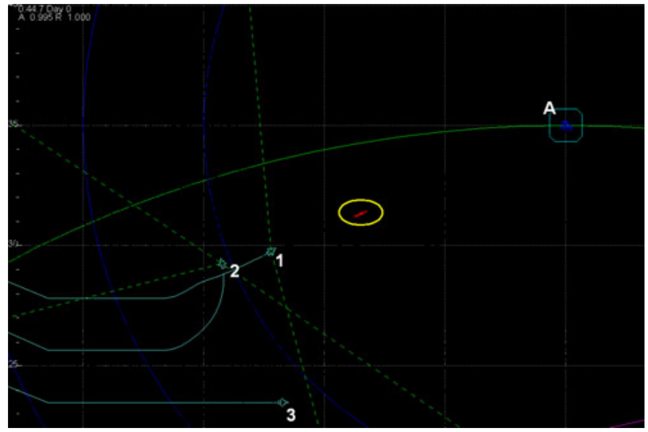

Figure 1: Attack by UCAV1 on SAM Site A

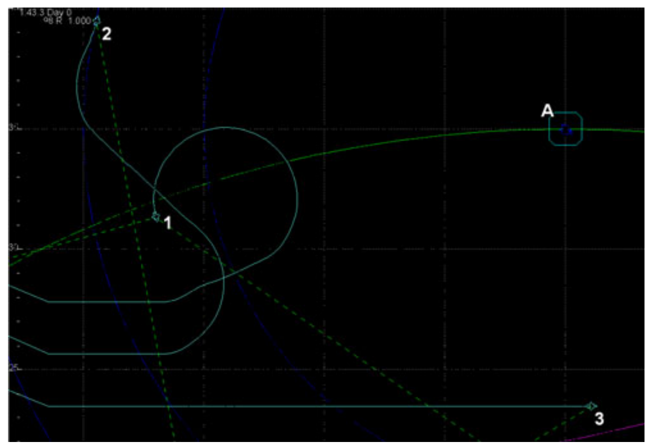

Figure 2: Kill Verification by UCAV2 on SAM Site A

In this configuration, MultiUAV is solely responsible for updating each vehicle's position and orientation. JIMM supplies each MultiUAV vehicle with two critical sensors. The first is a Radar Warning Receiver (RWR) capable of detecting energy emissions from enemy SAM sites. This sensor is used to trigger the task allocation algorithms among vehicles if the SAM begins to accurately track the vehicle positions. A SAR sensor is also supplied by JIMM. This allows any vehicle to query JIMM for Probability of Kill (PK) information. Along with these two sensors, JIMM supplies the entire IADS command and control simulation along with a two-dimensional viewer to watch scenario progression, as seen in the previous figures.

The IADS scenario developed in JIMM worked well for simple missions. Typically there were places in the scenario where the UCAV team could encounter two SAM sites concurrently. If the MultiUAV researcher wanted to stress the cooperative control algorithms with denser SAM concentrations, modifications to the JIMM configuration databases were needed. This required JIMM database experts to be consulted, which delayed progress with MultiUAV development.

If other scenario changes were required, such as decreasing the detection radius of the SAM site, changing the weapon payload or munition type in the UCAV, increasing the number of vehicles present in JIMM, or changing the UCAV sensor behavior supplied by JIMM, database experts were again required. Another limitation with the MultiUAV/JIMM environment is the real-time synchronization technique required by the IADS scenario. In order to utilize HLA time-management services to enable a faster and constructive simulation environment, an individual familiar with JIMM would need to reconfigure the scenario.

Though JIMM is a powerful tool built to model advanced IADS simulations, it requires a great deal of inner-working knowledge to build and modify scenario behaviors. For the MultiUAV researcher, an equally powerful tool with a simpler user interface is needed to conduct SEAD cooperative control studies.

3 CONNECTING MULTIUAV TO FLAMES

FLAMES is chosen as the appropriate replacement to JIMM since it provides an easier user interface to develop and modify scenarios. Another contributing factor is due to AFRL supporting a number of IADS models in the FLAMES architecture. This section provides a brief overview of FLAMES, the IADS models developed by AFRL, and the modifications necessary for a MultiUAV/FLAMES simulation environment.

3.1 FLAMES Overview

FLAMES is an object-oriented framework for developing constructive simulations and provides interfacing capabilities to connect other constructive, virtual, or live simulations together. At its core, it is comprised of a suite of software tools that build, execute, view, and analyze a given scenario. FLAMES organizes all system-specific computations in components called models. Models can represent physical entities, such as vehicles, communication devices, or weapons. Models can also represent human reasoning and decision making or natural and made-made environmental features. A large set of models are included with the standard FLAMES release, and developers can create their own models according to FLAMES standards.

3.2 FLAMES IADS Model

Several existing IADS models developed by NASIC are used by AFRL for enemy defense system studies. Unclassified portions of those models are used in this research to build a fictitious layout with the intent to challenge the MultiUAV cooperative control decision making. The flexibility of FLAMES allows the researcher to place entities in any desired layout, while using the underlying command and control logic developed by NASIC. A high-level snapshot of the scenario used in this research can be seen in Figure 3.

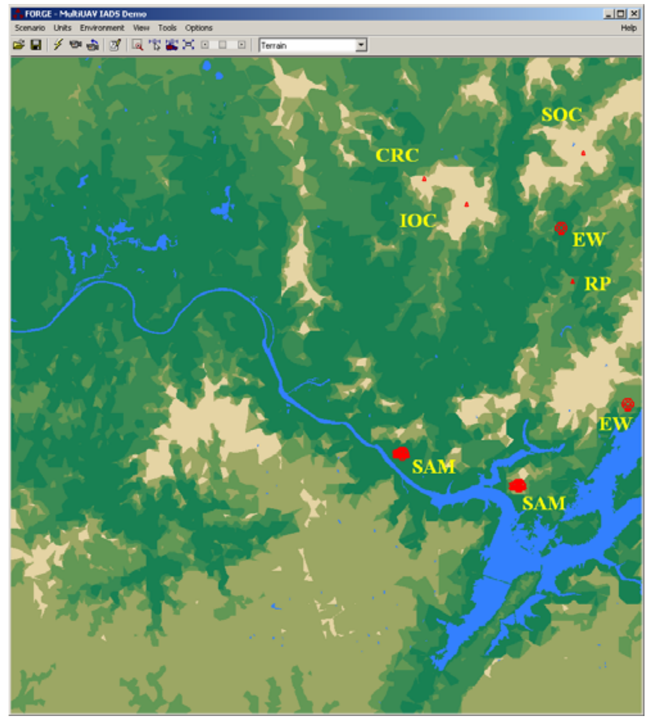

Figure 3: Fictitious IADS Scenario

As seen in Figure 3, multiple components make up the example IADS scenario. Two Early Warning (EW) radar sites exist to make early detections of approaching enemy units. As radar plots are established for approaching enemies, this information is passed through the Radar Post (RP), Confined Resource Center (CRC), and finally arrives at the Sector Operations Center (SOC). The SOC receives these radar plots and establishes tracks for the approaching enemies. The SOC is the highest level in the command chain, and makes high-level decisions on when to attack approaching enemies and what assets to use for engagement. Since MultiUAV can only prosecute immobile ground threats, the IADS has been limited to the use of SAM Battalions for defense. The ability for the IADS to deploy other assets for enemy engagement has been removed from the scenario.

If the SOC decides to engage the approaching enemy, appropriate commands are passed through the Intercept Operations Center (IOC) and routed accordingly to the SAM Battalion. Once the approaching enemy is in range, the tracking radar in the battalion will begin to track the enemy position with greater precision and fire with the appropriate SAM Battery unit. This command and control hierarchy is summarized in Figure 4.

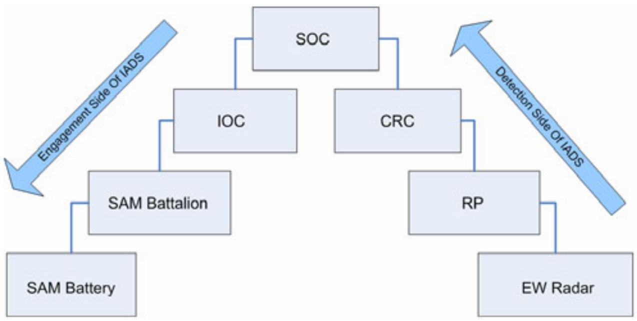

Figure 4: IADS Command and Control Hierarchy

MultiUAV will challenge the IADS by deliberately flying the UCAV fleet into the SAM Battalion group. The complexity of the scenario can be altered by adding, subtracting, and changing the location of SAM Battalions through the FLAMES scenario editor. Similar to JIMM, the IADS scenario in FLAMES will provide each vehicle with a RWR for threat detection and a SAR sensor for Battle Damage Assessment (BDA).

The sensors provided to each UCAV, along with the sensors, weapon systems, and communication devices supplied to each IADS component, is considered a FLAMES equipment model. Equipment models are attached to a base platform in FLAMES. If the user wishes to shrink the detection radius of an EW radar unit, or increase the drag coefficient of a stored munition in a UCAV, the only modification needed is to the relevant instance of that equipment model. If it is desired to change the behavior of all instances of the equipment model, the developer can change the base class through the FLAMES user-interface or in the underlying C-code. This object-oriented approach eases the complexity of scenario manipulation.

3.3 MultiUAV Remote Client Interface

FLAMES provides an Interactive Client and Interactive Server Option if a developer wishes not to migrate their existing model entirely into FLAMES. This option includes a set of services that allow external systems to interact with the FLAMES kernel over a network connection. In this effort, the HLA connection in MultiUAV was disabled and a new layer of software was added. The MultiUAV Remote Client Interface (RCI) utilizes the services provided in the FLAMES Interactive Client Option and allows each vehicle residing in MultiUAV to act as a physical entity in the IADS scenario.

The MultiUAV RCI attaches each vehicle in MultiUAV with an instance of a UCAV platform found in the FLAMES scenario. This attachment allows the MultiUAV vehicles to remotely interact with any equipment models residing on the UCAV. In the IADS scenario, each UCAV contains a RWR and SAR sensor model along with an Air-to-Surface Missile (ASM) weapon system model. The RWR sensor actively scans an area in front of the vehicle while processing, sorting, and tracking energy emissions found in the environment. When commanded, the SAR sensor takes a snap-shot of the terrain in its field-of-view and performs BDA on physical units through basic image processing techniques. The ASM contains a payload of two Joint Direct Attack Munition (JDAM) precision guided bombs. It controls the release and tracking of the weapon during flight.

3.4 Interactions Between MultiUAV and the IADS

The example IADS model discussed earlier will not be stimulated unless enemy units approach the detection area of the EW radar nodes. As the vehicles fly their predetermined route by MultiUAV, the RWR employed on each UCAV in the FLAMES scenario is monitored. EW radar emissions are detected by the vehicles, but ignored since they are non-lethal threats. On the other hand, once the tracking radar of a SAM Battalion is detected, the respective UCAV notifies all other team members of a threat detection. In MultiUAV, each vehicle's cooperative control algorithms decide what team member is in the best position to attack the SAM Battalion and verify the destruction, or perform BDA, on the SAM Battalion. The cooperative control algorithms are called at each new SAM detection and tasking for vehicles is updated accordingly.

Once a vehicle is in position to attack a battalion, MultiUAV commands the ASM equipment model to release a weapon. The FLAMES scenario is then responsible for modeling the release, fly-out, and termination point of the JDAM. When the vehicle responsible for BDA of the battalion is in position, MultiUAV commands the SAR equipment model in FLAMES to detect the presence of any killed units in the field-of-view of the SAR. This information is returned to MultiUAV and the threat can be considered killed or re-attacked if necessary. The MultiUAV controlled UCAVs must also be cautious of the SAM battalion threat area. If a vehicle comes within firing range of a SAM battery unit, it risks being fired upon. The flow of information between MultiUAV and the IADS scenario in FLAMES is summarized in Figure 5.

4 SIMULATION RESULTS

To demonstrate MultiUAV interacting with the IADS scenario, an example mission is presented. This involves a group of three UCAVs engaging two SAM Battalions. For ease of reference, the UCAVs are identified with numbers while the SAM Battalions are lettered.

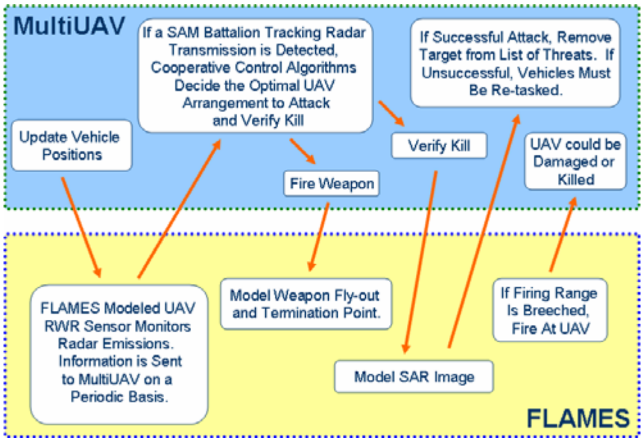

Figure 5: MultiUAV/IADS Exchange of Information

This is a fairly simple task assignment problem for MultiUAV, since four tasks (two attacks, two BDAs) are solved in total. It is presented here to detail the significant interactions between MultiUAV and the IADS scenario in FLAMES. It is important to note that this environment has been used successfully to stress the cooperative control algorithms in scenarios with a larger density of SAM battalions (Darrah et al. 2006).

4.1 SAM Battalion Discoveries

The default MultiUAV search pattern involves each vehicle flying from the left and towards the right side of the scenario. Therefore, the group of UCAVs in this study are placed well to the left of the SAM Battalions. As the vehicles fly into the detection area of the EW radar sites, the RWR on each vehicle notes the presence of enemy radar emissions. The data processors in MultiUAV ignore this information and wait for tracking radar emissions originating from the appropriate SAM Battalion. Figure 6 illustrates a typically SAM Battalion. Notice the tracking radar unit in the center of the figure and a battery unit sitting directly to the right.

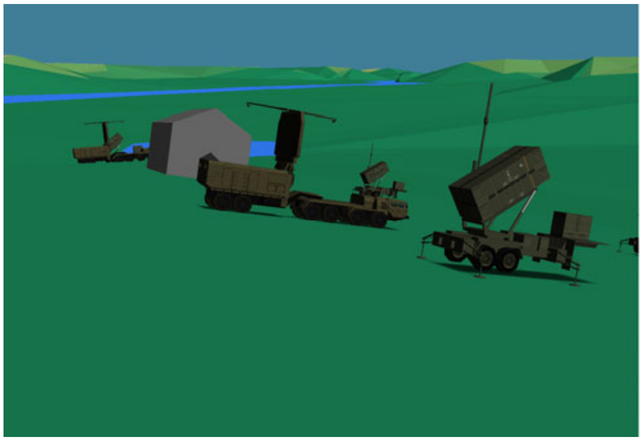

Figure 6: SAM Battalion

Once the mission commences, UCAV1 notes the first enemy emission from SAM Battalion A at 30 seconds. At this time, UCAV2 and UCAV3 are notified of the threat. The MultiUAV cooperation manager on each vehicle then decides the most efficient strategy:

- UCAV2 attacks SAM Battalion A at 50 seconds
- UCAV1 performs BDA on SAM Battalion A at 116 seconds

As UCAV1 and UCAV2 begin to fly their updated route plan, UCAV1 detects a second enemy emission originating from SAM Battalion B at 40 seconds. This can be seen in Figure 7. The lines connecting the vehicles and battalions indicate energy emissions between the objects. Notice vehicles 1 and 3 have already started to implement the previous task assignment plan. At this point, UCAV1 notifies all team members of a new threat. The MultiUAV cooperation manager disregards the prior assignment and calculates the most efficient strategy to prosecute both threats:

- UCAV3 attacks SAM Battalion B at 58 seconds and SAM Battalion A at 74 seconds
- UCAV1 performs BDA on SAM Battalion A at 117 seconds
- UCAV2 performs BDA on SAM Battalion B at 126 seconds

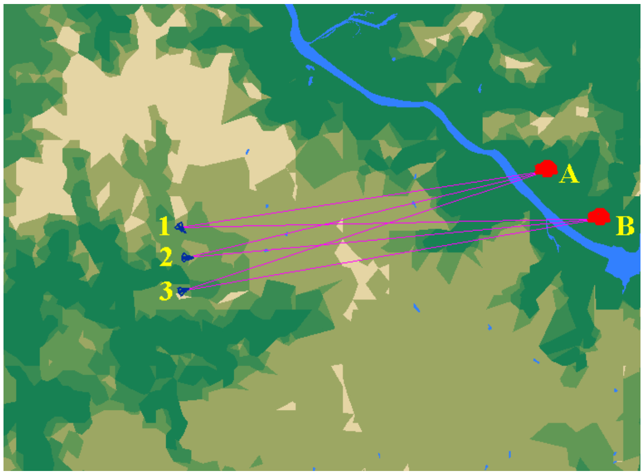

Figure 7: Discovery of SAM Site B at 40 seconds

4.2 Attack Tasks

As each vehicle implements its plan, MultiUAV positions UCAV3 appropriately for both attack tasks. In order to guide the JDAM on the appropriate path, the attacking vehicle must be oriented at the SAM during weapon release. UCAV3 orients itself first at SAM Battalion B and releases the first JDAM at 58 seconds. Figure 8 shows the second JDAM release towards SAM Battalion A at 75 seconds. Notice the aircraft is oriented appropriately for weapon release. Also note that the timing of the weapon releases occur almost identical to the original MultiUAV plan. For clarity, each JDAM is circled. In the figure it can be seen that the first JDAM is very close to its desired termination point while the second JDAM has just left the bomb bay of UCAV3. Meanwhile UCAV1 and UCAV2 are orienting themselves for BDA.

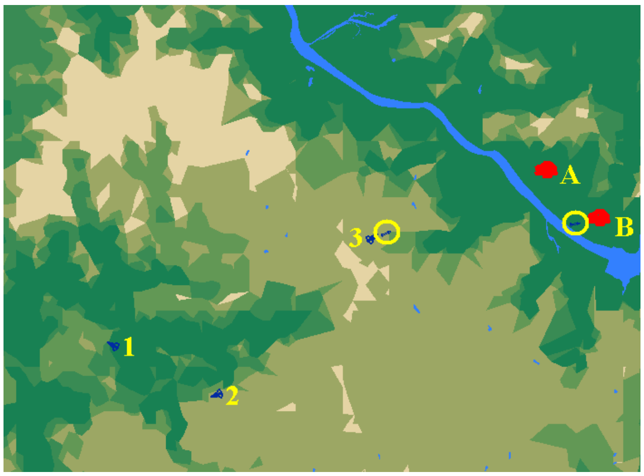

Figure 8: Second Attack by UCAV3 at 75 seconds

Slightly later in the simulation, it can be seen that the released JDAM arrives at SAM Battalion B in Figure 9. It can easily be seen that the JDAM successfully prosecuted the SAM Battalion. The MultiUAV vehicles will not know this until BDA is performed.

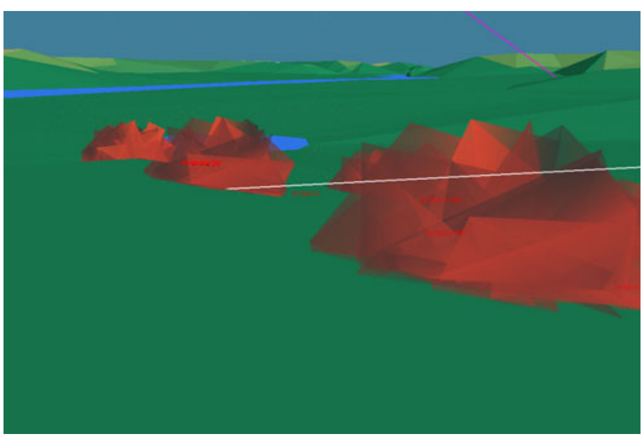

Figure 9: JDAM Impact at SAM Battalion B

4.3 BDA Tasks

At this point, both SAM Battalions need BDA in order to verify the prior attack. MultiUAV directs vehicles 1 and 2 to orient themselves approximately 20 kilometers from the original battalion location. Since the field-of-view for the SAR is 90^0 to the right of the vehicle heading, both aircraft fly a path tangent to the threat location. Once the UCAV achieves steady and level flight, its SAR footprint is placed over the attacked battalion and a snapshot is taken of the area. Figure 10 is included to show UCAV2 flying away from the SAM Battalions for proper SAR orientation.

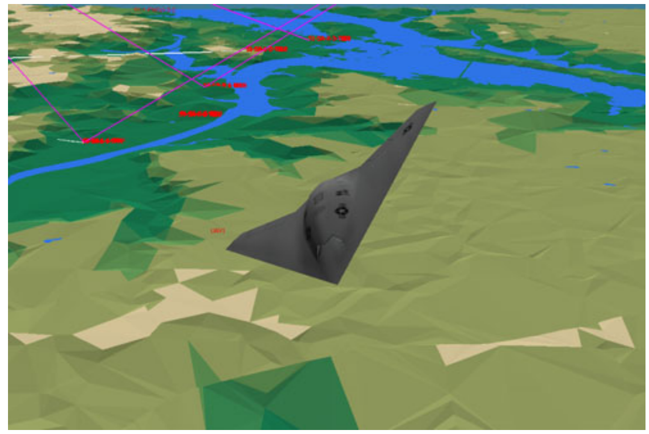

Figure 10: UCAV2 Flying Into BDA Position

UCAV1 performs BDA on SAM Battalion A at 110 seconds. After processing the SAR image, the battalion is deemed killed since the critical tracking radar components are no longer functional. If the battalion loses its tracking radar component, the enemy UCAVs can no longer be attacked. In Figure 11, the final BDA task is completed when UCAV2 performs kill verification on SAM Battalion B at 120 seconds. Note that a SAR footprint is imposed off the right side of the vehicle for clarity. After SAM Battalion B is verified, all three aircraft resume their original search patterns. If more battalions are discovered, the sequence of events covered in this section are repeated.

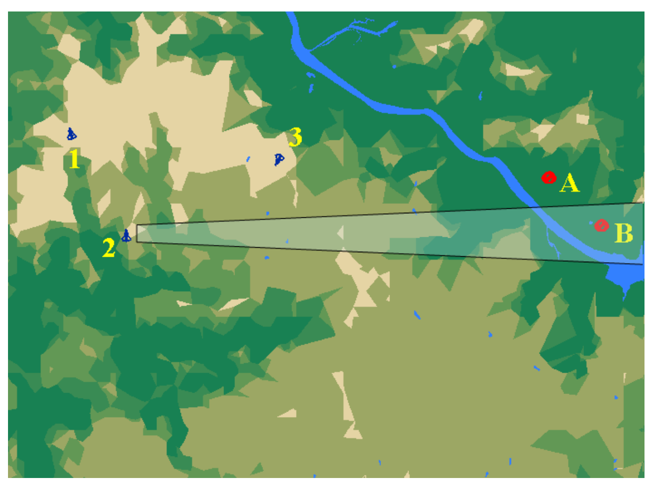

Figure 11: Final BDA Task by UCAV2 at 120 Seconds

5 ANALYSIS AND CONCLUSION

The simulation results show MultiUAV successfully integrated with the FLAMES simulation environment. The overall goal of this effort is to reduce the burden on MultiUAV researchers modeling high fidelity battlefields. Thus, MultiUAV can be heavily focused on cooperative control research.

FLAMES provides significant enhancements over the JIMM software from the researcher's perspective. For example, the example IADS layout built for this research (Figure 3) is performed by simply placing units in a desired configuration through a graphical user interface. If a similar requirement is needed in JIMM, consultation from the original IADS author is required. On average, using FLAMES also generates a 50% decrease in computational time over JIMM while running distributed simulations with MultiUAV. One last benefit is the tightly integrated two-dimensional and three-dimensional viewers in FLAMES. They are useful for detailed post-analysis of the UCAV task allocation assignments provided by MultiUAV.

Further MultiUAV research will involve a mix of heterogeneous teams with electronic jamming requirements imposed for full SAM prosecution. This task will be eased through jamming equipment models supplied by FLAMES. UAV urban operation studies will also be conducted with MultiUAV. FLAMES will aid in this by supplying accurate, three-dimensional rendering of urban areas while supplying line-of-sight calculations for surveillance missions.

There are a few drawbacks when using FLAMES. By employing the Interactive Client Option, MultiUAV's HLA interface is disabled. This results in a loss of flexibility, since the Interactive Client option utilizes a series of proprietary system calls for FLAMES interaction. FLAMES also utilizes a complicated and costly licensing system, thus reducing flexibility. JIMM is a free product for U.S. government employees and contractors. JIMM also does not require runtime licensing constraints.

ACKNOWLEDGMENTS

This effort was sponsored by AFRL under contract number FA8650-04-C-3402.

REFERENCES

Banda, S. 2002. Future Directions in Control for Unmanned Air Vehicles. In *Proceedings of AFOSR Workshop on Future Directions in Control*. Arlington, VA.

MultiUAV. Institute for Scientific Research, Inc. 2006. Available at <http://www.isr.us/research_sim_muav.asp>.

Darrah, M., W. Niland, B. Stolarik, and L. Walp. 2006. UAV Cooperative Task Assignments for a SEAD Mission Using Genetic Algorithms. Accepted for Publication at the 2006 Guidance, Navigation, and Control Conference. Keystone, CO: American Institute of Aeronautics and Astronautics.

Niland, W., B. Stolarik, J. Harman, J. Petersavage, and S. Rasmussen. 2005. The Inclusion of a Supplementary Mission Scenario into the MultiUAV Research Tool. In *Proceedings of the 2005 Guidance, Navigation, and Control Conference*. San Francisco, CA: American Institute of Aeronautics and Astronautics.

Panson, D. 2004. Integrated Air Defense Systems: Modeling and Simulation Based Analysis. Presented at the 2004 PC Working Group. Las Vegas, NV.

Rasmussen, S. and P. Chandler. 2002. MultiUAV: A Multiple UAV Simulation for Investigation of Cooperative Control. In *Proceedings of the 2002 Winter Simulation Conference*, ed. E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, 869-877. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers. Available at <http://www.informs-sim.org/wsc02papers/113.pdf> [accessed March 22, 2006].

Schulz, C., D. Jacques, and M. Pachter. 2003. Cooperative Control Simulation Validation Using Applied Probability Theory. In *Proceedings of the 2003 Guidance, Navigation, and Control Conference.* Austin, TX: American Institute of Aeronautics and Astronautics.

Stolarik, B., W. Niland, and B. Givens. 2004. Multiple UAV Communication Modeling Using the HLA. In *Proceedings of the 2004 Simulation Interoperability Workshop*. Alexandria, VA: Simulation Interoperability Standards Organization. Available at <http://www.isr.us/pdfs/publishedpapers/04S-SIW-104.pdf> [accessed March 21, 2006].

AUTHOR BIOGRAPHY

WILLIAM M. NILAND is a Member of the Research Staff working under the Defense Program at the Institute for Scientific Research, Inc. He received a Bachelor of Science in Electrical and Computer Engineering from West Virginia University (WVU) and is currently pursuing a Masters of Science in Electrical Engineering from WVU. His current work supports AFRL on various UAV programs, including decentralized cooperative control research, automated aerial refueling flight test support, and distributed simulation integration. Previous work experience includes hardware design and fabrication for various UAV sensor test-beds and reverse engineering of MATLAB/Simulink intelligent flight control models. His e-mail is <wniland@isr.us>.