ARL-TR-8572 ● Nov 2018

US Army Research Laboratory

US Army Robotic Wingman Simulation: June 2018 Integration Workshop

by Kristin E Schaefer, Ralph W Brewer, E Ray Pursel, Michelle Desormeaux, Anthony Zimmermann, and Eduardo Cerame

Approved for public release; distribution is unlimited.

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

ARL-TR-8572 ● Nov 2018

US Army Research Laboratory

US Army Robotic Wingman Simulation: June 2018 Integration Workshop

by Kristin E Schaefer
Human Research and Engineering Directorate, ARL

Ralph W Brewer,
Vehicle Technology Directorate, ARL

E Ray Pursel and Michelle Desormeaux
Naval Surface Warfare Center, Dahlgren Division, Dahlgren, VA

Anthony Zimmermann
DCS Corporation, Alexandria, VA

Eduardo Cerame
US Army TARDEC, Warren, MI

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

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

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED (From - To)
November 2018	Technical Report	1 June–30 September 2018

4. TITLE AND SUBTITLE
US Army Robotic Wingman Simulation: June 2018 Integration Workshop

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Kristin E Schaefer, Ralph W Brewer, E Ray Pursel, Michelle Desormeaux, Anthony Zimmermann, and Eduardo Cerame

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
US Army Research Laboratory
ATTN: RDRL-HRF-D
Aberdeen Proving Ground, MD 21005-5425

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-8572

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
US Army Research Laboratory, ATTN: RDRL-HRF-D, Aberdeen Proving Ground, MD 21005-5425
US Army Tank Automotive Command, 6305 E Eleven Mile Rd, Warren, MI 48092

10. SPONSOR/MONITOR'S ACRONYM(S)
ARL/TARDEC

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This work is in support of the US Army Robotic Wingman Joint Capabilities Technology Demonstration. The purpose of this report is to document current status and updates made to the US Army Robotic Wingman software-in-the-loop (SIL) simulation testbed. Specific updates to the SIL include updates to the real-world vehicle software components (Robotic Technology Kernel, Autonomous Remote Engagement System, and Warfighter Machine Interface displays) so that they match the current versions onboard the real-world vehicle. In addition, the SIL subsystems were updated to improve the parallelization of simulation elements, including vehicle physics, modeling, and image generation to improve real-time run capabilities (Autonomous Navigation and Virtual Environment Laboratory [ANVEL]). A client–server architecture design was included through Unity3D to support the addition of a Long-Range Advanced Scout Surveillance System operator crewstation. Advancements were also made to improve the process for sharing terrain files between the ANVEL and Unity3D game engines; improve communication between the different subsystems, fix integration issues related to waypoint following, localization, and ARES Picatinny Lightweight Remote Weapon Station driver control; and add additional data collection-type features. Outcomes of this work will support the capability for rapid prototyping for autonomous mobility and lethality, system integration, human-in-the-loop evaluations, and training.

15. SUBJECT TERMS
human-robot interaction, simulation, Warfighter Machine Interface, Wingman, robot

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Kristin E Schaefer-Lay
Unclassified	Unclassified	Unclassified	UU	43	19b. TELEPHONE NUMBER (Include area code) 410-278-5972

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

Contents

List of Figures v

List of Tables vi

Summary vii

1. Introduction 1

1.1 Wingman Program 1

1.2 Wingman SIL Simulation Testbed 2

2. The Wingman SIL: Current State and Updates 4

2.1 ANVEL 4

2.2 UNITY3D 5

2.2.1 Unity Wingman SIL v2.0 5

2.2.2 Unity Wingman SIL v2.1 5

2.3 RTK 6

2.4 ARES 6

2.5 WMI Displays 6

2.6 LRAS3 Operator Crewstation 7

3. Integration Challenges 10

3.1 Sharing Terrain Files between ANVEL and Unity 10

3.2 Communication between SIL Subsystems 11

3.2.1 Missing Drive Cameras in WMI 11

3.2.2 PLWRWS Emulator/SPU Emulator 12

3.2.3 Loss of Localization over Time 12

3.2.4 ARES PLWRWS Driver Control Gains 12

3.2.5 ARES Video Mux 13

3.2.6 Communicating Information through the WMI Display 13

3.3 Robotic Vehicle Control Capabilities and Needs 16

3.3.1 Operation of Waypoint Following 16

Approved for public release; distribution is unlimited.

3.3.2 Using a Joystick Controller for the Robotic Vehicle Gunner Crewstation 17

4. Data Collection 17

4.1 Vehicle Data: ROS Bag Files 18

4.2 WMI Log Files: Additional Recording Capabilities 19

5. Testing 19

5.1 Localization and Slew-to-Cue 20

5.2 LRAS3 Operator Crewstation 20

5.3 Human Factors Assessment 21

5.3.1 General Issues 21

5.3.2 Vehicle Commander Crewstation 21

5.3.3 Robotic Vehicle Operator Crewstation 22

5.3.4 Robotic Vehicle Gunner Crewstation 22

6. Conclusions 23

6.1 Updates to the SIL Simulation Testbed 24

6.1.1 New HMMWV Vehicle Models 24

6.1.2 Develop Matching ANVEL and Unity3D Terrains 24

6.1.3 Update the LRAS3 Crewstation 25

6.1.4 New C2 Vehicle Driver Crewstation 25

6.1.5 Update the WMI Features 26

6.2 Data Collection Improvements 26

6.2.1 Table VI Performance Data 26

6.2.2 WMI Log files 27

6.2.3 ROS Bag Files 27

6.2.4 Synchronize Time 27

6.3 AAR Technology Development 27

7. References 29

List of Symbols, Abbreviations, and Acronyms 30

Distribution List 32

Approved for public release; distribution is unlimited.

List of Figures

Fig. 1 The C2 vehicle (left) and robotic combat vehicle (right) are shown in the top part of the figure. The robotic combat vehicle operating downrange at Camp Grayling is shown in the bottom part of the figure. 2

Fig. 2 Updated schematic depicting the multiple subsystems that are integrated within the Wingman SIL simulation testbed. Each color represents a different computer. 3

Fig. 3 LRAS3 handgrips (for control of the LRAS3) and gamepad (for control of operator's head movements) are used for integrating the LRAS3 operator crewstation into the Wingman SIL 8

Fig. 4 Simulated LRAS3 operator crewstation in Unity3D 8

Fig. 5 Simulated LRAS3 targeting options. a) Wide FOV, b) Wide FOV with e-zoom, c) Narrow FOV, and d) Narrow FOV with e-zoom and displaying target information. 9

Fig. 6 Example user map in the WMI from one of the Camp Grayling gunnery ranges. The placement of both the C2 vehicle (high-mobility multipurpose wheeled vehicle [HMMWV]) and the RCV (MRZR) is shown, as well as an identified hostile threat (red diamond), friendly target (blue square), and a potential target (yellow cloverleaf). 14

Fig. 7 Example menu options available to the vehicle commander for editing BSOs 15

Fig. 8 Additional tactical graphics that can be marked on the user map 16

Fig. 9 Example of the robotic vehicle gunner WMI display. The left portion is the threats screen widget where the gunner can easily identify potential targets. The center portion is the sensor display. In the sensor display, the camera sensor data come from ARES. It shows where the weapon system is aimed. Additional features include the red boxes denoting ARES-identified targets, and the newly added target detection widget (oval at the bottom of the screen). The right portion of the WMI is the map widget, as depicted in Figs. 6–8 23

Fig. 10 Example screenshot of AAR mission replay capabilities using Google Earth. The yellow symbols denote the BSOs representing potential target locations. All troop and vehicle targets with their associated target numbers can be depicted on the map. The text box shows specific information relevant to a selected object, in this case, the BSO. The C2 vehicle with the LRAS3 markers and RCV with turret location are also integrated into this AAR playback feature 28

Approved for public release; distribution is unlimited.

List of Tables

Table 1 LRAS3 button schematic 10

Approved for public release; distribution is unlimited.

Summary

Wingman is a US Army Tank Automotive Research, Development and Engineering Center (TARDEC)-led program that began in 2014 and became an Office of Secretary of Defense (OSD)-supported Joint Capabilities Technology Demonstration (JCTD) in 2017. The purpose of the Wingman program is to provide robotic technological advances and experimentation to increase the autonomous capabilities of mounted and unmanned combat support vehicles. Development of a software-in-the-loop (SIL) simulation testbed began in 2017 to help meet this objective.

The purpose of this report is to document current status and updates made to the SIL following the June 2018 Integration Workshop at Aberdeen Proving Ground. Updates to the simulation platforms and robotic vehicle software include the following:

1) The Autonomous Navigation and Virtual Environment Laboratory (ANVEL) game engine was updated to v3.0 to improve the backend, including parallelization of various simulation elements (e.g., vehicle physics, sensor modeling, and image generation) to help the simulation run in real-time.

2) The Unity3D Wingman executable was updated to v2.1 to support a client–server architecture for the integration of the Long-Range Advanced Scout Surveillance System (LRAS3) operator crewstation (previously a simulated human role).

3) The Robotic Technology Kernel (RTK) was updated with the code base from the Southwest Research Institute (SwRI) repository from the May 2018 Camp Grayling test event to match the current version on the real-world vehicle. The RTK handles the mobility autonomy for both the live and virtual robotic combat vehicle. Updates to the virtual machine (VM) and the RTK packages were made to support functionality with the ANVEL v3.0 plug-ins.

4) The Autonomous Remote Engagement System (ARES) software was updated to version 2018-06-18.3-dev to match the version being used on the real-world vehicle. This included an electro-optical/infrared (EO/IR) video switcher and updates to the fire controller, and the Picatinny Lightweight Remote Weapon Station (PLWRWS) interface.

5) The Warfighter Machine Interface (WMI) displays were updated to v5.2.11.

6) Specific improvements were made to the process for sharing terrain files between the ANVEL Ogre 3-D game engine and the Unity3D game engine, thereby enhancing communication between the different subsystems.

7) Other advancements to the SIL subsystems included fixing the integration issues related to waypoint following, localization, and the ARES PLWRWS driver control, as well as adding additional data collection-type features.

Specific testing through the entire SIL was conducted to evaluate the localization and slew-to-cue features, integration of the LRAS3 operator crewstation, and human factors needs and issues associated with WMI use. Outcomes of this work will directly support the capability for rapid prototyping for autonomous mobility and lethality, system integration, human-in-the-loop evaluations, and training to use the real-world vehicle.

Approved for public release; distribution is unlimited.

1. Introduction

This section provides an overview of the Wingman program and the associated development of the software-in-the-loop (SIL) simulation testbed to support rapid prototyping, training, and human-in-the-loop experimentation. Additional information about the Wingman program, as well as SIL development and capabilities, is provided in Schaefer et al. (2017a, 2017b, 2019) and Brewer et al. (2019).

1.1 Wingman Program

The Wingman program began in fiscal year 2014 to provide robotic technological advances and experimentation to increase the autonomous capabilities of mounted and unmanned combat support vehicles. It became an Office of Secretary of Defense (OSD)-supported Joint Capabilities Technology Demonstration (JCTD) program in FY17, with a major goal of advancing manned–unmanned teaming initiatives while iteratively defining and decreasing the gap between autonomous vehicle control and required level of human interaction.

At present, the Wingman team includes a single manned command and control (C2) vehicle working together with a single unmanned weaponized robotic combat vehicle (RCV) operating in a joint gunnery task (Fig. 1). Onboard the C2 vehicle is a five-man crew: C2 vehicle driver, C2 vehicle gunner and Long-Range Advanced Scout Surveillance System (LRAS3) operator, commander, robotic vehicle operator, and robotic vehicle gunner. The main outcome of this program is to demonstrate a successful gunnery qualification on a live-fire range using the weaponized platforms against stationary and moving targets placed in a tactical array, during day and limited visibility conditions, from both offensive and defensive postures (TRADOC 2015).

Approved for public release; distribution is unlimited.

Fig. 1 The C2 vehicle (left) and robotic combat vehicle (right) are shown in the top part of the figure. The robotic combat vehicle operating downrange at Camp Grayling is shown in the bottom part of the figure.

1.2 Wingman SIL Simulation Testbed

The Wingman SIL simulation testbed was developed so that Wingman team members could have regular access to the real-world vehicle software within a simulated environment. The goal was to advance the software, test the integration, and assess the interactions between the five-man crew, simulated hardware, and software in a safe and cost-effective setting. More specifically, it provides a joint capability for advancing and testing the integration of the individual subsystems of the real-world vehicle. These include the Robotic Technology Kernel (RTK) for autonomous mobility (developed by the US Army Tank Automotive Research

Approved for public release; distribution is unlimited.

Development and Engineering Center [TARDEC]); the Autonomous Remote Engagement System (ARES), which supports the autonomous targeting and weapons systems control (developed by the Naval Surface Warfare Center Dahlgren Division [NSWCDD]); and the Warfighter Machine Interface (WMI), which provides interactive displays for the Wingman commander, robotic vehicle operator, and robotic vehicle gunner team members (developed by DCS Corp).

The development of the SIL allows the team to test the subsystem capabilities in a simulated Table VI evaluation. The design supports integration and assessment of robotic functionality and human–robot teaming capabilities* in a repeatable and systematic platform. For Wingman, simulation platforms are used to substitute the real-world environment and real-world vehicles with a virtual environment and virtual vehicles. The SIL as a whole is then able to integrate all of the real-world vehicle software (listed previously) within a virtual space. Two different simulation platforms were integrated into the current SIL configuration: Autonomous Navigation and Virtual Environment Laboratory (ANVEL), which supports the RTK software for autonomous mobility, and Unity3D, which supports the integration of the ARES software for weapons systems targeting and firing with dynamic results, such as hitting a target (Fig. 2).

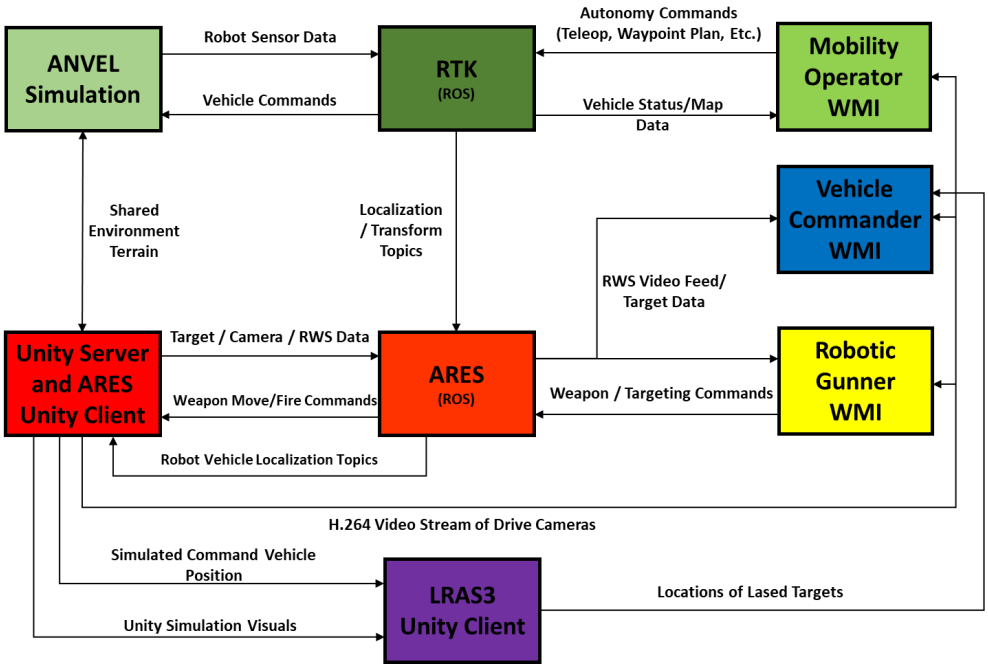

Fig. 2 Updated schematic depicting the multiple subsystems that are integrated within the Wingman SIL simulation testbed. Each color represents a different computer.

* These human–robot teaming efforts and assessments are being led by the US Army Research Laboratory (ARL).

Approved for public release; distribution is unlimited.

2. The Wingman SIL: Current State and Updates

This section describes the current state of and updates made to each of the SIL subsystems prior to the integration workshop. It also identifies areas of testing and potential integration issues that were addressed.

2.1 ANVEL

ANVEL* is a virtual simulation tool that specializes in unmanned ground vehicle technologies and applications. It was designed to support autonomous mobility and directly supports integration of the RTK. In preparation for the integration workshop, ANVEL was updated in the Wingman SIL (located at TARDEC) from v2.5 to v3.0 to test new features, capabilities, and identify potential integration issues. This new version of ANVEL provides improvements to the backend, including improved parallelization of various simulation elements (e.g., vehicle physics, sensor modeling, and image generation). This helps keep the simulation running in real-time even when one of the SIL subsystems is computationally demanding.

As part of the ANVEL upgrade, new builds of the RTK plug-ins were generated. This was critical to maintain compatibility with the newest ANVEL API. However, during integration, it was found that turning on the optimization in Visual Studio 2015 prevented the plug-in from functioning correctly. To fix this issue, the optimization of the plug-in had to be disabled. Preliminary testing with ANVEL v3.0 and the plug-ins was conducted using an updated virtual machine (VM) and an older build of the WMI from CVMS 2017. With the configuration of this WMI crewstation, the simulated vehicle generated camera feeds and could be driven with teleoperation, waypoints, and goal points.

At the start of the integration workshop, the SIL located at ARL was updated with ANVEL v3.0 (from v1.5) and the updated VM was installed. One integration issue detected was that the update required a new ANVEL start-up process. Rather than placing a vehicle in the default environment with the AnvelSimInit.xml file, new python scripts were loaded into the C:\Rivet\AppData\Local\Anvel\Scripts \AnvelStartupScripts folder. These scripts run at launch and spawn a vehicle at a desired location. This new method changed the vehicle name during the simulation (from PolarisMRZR4_withVelodyneHDL32E to robot_1), which directly affected generation of the camera feeds. To fix this issue, a change to the VideoStreams.xml file was required.

* ANVEL was developed by Quantum Signal, LLC. More information can be found at https://www.quantumsignal.com/case_studies/anvel.php.

Approved for public release; distribution is unlimited.

2.2 UNITY3D

Unity3D* (also referred to as Unity) is a game engine that provides a customizable and dynamic virtual environment. It is currently integrated in the Wingman SIL because its functionality directly supports ARES. It also provides the flexibility to customize mission events (e.g., dynamic targeting features), which is essential to support and test the weapons targeting and firing protocols for the Wingman program. The client–server architecture also supports the integration of multiple crewstations. Two updated versions of the Wingman executable (Unity) were developed to support additional crewstations and robotic asset support.

2.2.1 Unity Wingman SIL v2.0

The Unity simulation environment was modified from a single standalone application to a client–server architecture to enable SIL support for additional crew roles (e.g., C2 vehicle driver and LRAS3 operator, and commander view through the windshield). In the previous version, the virtual environment, ARES interface (between the environment and ARES), and experimenters' graphical user interface (GUI) were an integrated single "player". In v2.0, the virtual environment and experimenters' GUI are a Unity server and the ARES interface is a separate client. The server must be running to operate the SIL; however, the ARES client is now optional. With this separation into server and clients, the SIL can easily incorporate multiple clients, thereby including all five of the Wingman crewstations. This modular client–server architecture will also allow the SIL to include additional robotic assets for systems integration testing, demonstration, and experimentation.

2.2.2 Unity Wingman SIL v2.1

The architecture introduced in v2.0 enabled the addition of an LRAS3 operator client to the SIL. The LRAS3 client interfaces with the vehicle commander's WMI in the same way (in terms of protocol and messaging) that the WMI normally interfaces with the LRAS3 hardware. Like the ARES client, the LRAS3 operator client is an optional component, and multiples of this client could be easily implemented. During the integration workshop, the new Unity client–server architecture was tested and verified with the full Wingman SIL simulation testbed (including the latest ARES and RTK applications and the LRAS3 operator client's interface with the commander's WMI).

* Unity was developed by Unity Technologies. More information can be found at http://unity3d.com.

Approved for public release; distribution is unlimited.

2.3 RTK

RTK is a government-owned, -designed, and -maintained modular autonomy kit for science and technology development that provides a set of common robotic capabilities (including autonomous mobility) across a variety of platforms and efforts. It enables capabilities to build on each other rather than replace existing ones. An updated VM was generated for the integration workshop, pulling in code from the SwRI repository using the workspace generated at the start of the May 2018 Camp Grayling test event. This brought the SIL in line with the code base currently installed on the real-world vehicle. Some additional updates to the RTK packages for anvel_sim and anvel_to_ros were made to ensure functionality with the ANVEL 3.0 plug-ins. This RTK build is capable of controlling the simulated vehicle through teleoperation or waypoint following using either an exploratory planner or a narrow route-following planner. The route-following planner is the primary planner used with the Wingman JCTD.

2.4 ARES

The main functions of ARES support automated engagement for slew-to-cue, video-based automatic target detection, and user-specified target selection. The ARES subsystem of the SIL was updated to match the latest development branch of the ARES code (2018-06-18.3-dev). The major changes included an EO/IR video switcher and updates to the fire controller and the Picatinny Lightweight Remote Weapon Station (PLWRWS) interface to match the firmware. The 18.3-dev codebase did not include changes from the week prior at the Wingman system test and integration event held at NSWCDD, during which localization and subsequent fire control were debugged.

Additional changes made during the integration workshop included adding the virtual PLWRWS control gains (that were developed during the Camp Grayling integration and test event) and updating the PLWRWS Emulator messaging to match the ARES PLWRWS driver (which was changed to match the new firmware). This was critical, as the control gains compensate for the unrealistically immediate starting and stopping of the virtual PLWRWS. Since these gains have to be adjusted for each real mount (even for different serial numbers of the same type of mount), adjusting them for the virtual mount is not atypical.

2.5 WMI Displays

Three versions of the WMI displays support the commander, robotic vehicle gunner, and robotic vehicle operator. These displays are customizable to support the unique needs of each role, as well as to enable shared situation awareness

Approved for public release; distribution is unlimited.

amongst the team members. During the integration workshop, WMI components underwent a migration of source code repositories to match those from the May 2018 Camp Grayling event. WMI software was delivered to match the real-world vehicle software (wingman5.2.11), matching the build in the Defense Intelligence Information Enterprise (DI2E) Bitbucket* repository. Authorized Wingman program users now have access to a single cohesive repository of source code, issue tracking, and documentation tools hosted in the Department of Defense domain. A subset of required WMI software components—the user maps and software development kit required to compile the source code—has yet to be migrated to the DI2E.

2.6 LRAS3 Operator Crewstation

The LRAS3 operator crewstation was introduced at the integration workshop. For Wingman, the LRAS3 operator's primary task is to find targets, determine the target's locations, and relay that location information to the Wingman team (targeting information is sent from the LRAS3 to the WMI displays). The LRAS3 operator crewstation is set up as a client in the Unity simulation subsystem and runs on a networked computer, separate from the Unity server. It interfaces with the commander's WMI—establishes a connection, performs messaging handshaking, sends heartbeat messages containing position and status information, and sends targeting information—all in accordance with the LRAS3 Interface Control Document (ICD).

Prior to the integration workshop, the LRAS3 operator crewstation was able to move in the virtual environment. Two modes for control were developed: 1) using the gamepad controller and 2) using the LRAS3 handgrips and gamepad (Fig. 3). The handgrips are a separate TCP/IP networked device; the gamepad is a common USB device.

*Bitbucket was created by Atlassian. It is a code repository that provides teams a central location for project planning and collaboration, as well as code testing and deployment. More information can be found at https://www.atlassian.com/software/bitbucket.

Approved for public release; distribution is unlimited.

Fig. 3 LRAS3 handgrips (for control of the LRAS3) and gamepad (for control of operator's head movements) are used for integrating the LRAS3 operator crewstation into the Wingman SIL

The virtual LRAS3 is positioned on the virtual C2 vehicle in the Unity environment (Fig. 4). The operator can change their point of view (via head movement), adjust the movement of the sight in all directions, swing the sight around the vehicle's ring mount, and zoom in and out on targets in the display. A subset of the LRAS3 buttons (eZoom, Zoom, and laser rangefinder [LRF]) was implemented for three zoom toggle options.

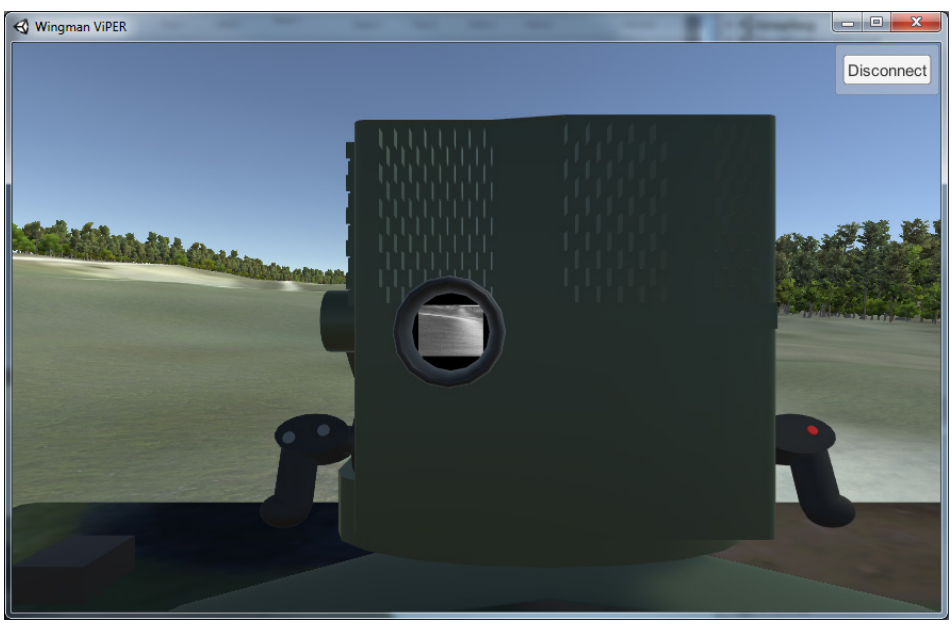

Fig. 4 Simulated LRAS3 operator crewstation in Unity3D

Following the integration workshop, the LRAS3 operator crewstation was able to transmit messages to the WMI in the same way as the actual LRAS3 transmits messages to the WMI. Following integration testing, minor adjustments were made to improve data and timing accuracy. Four zoom toggles were developed to support wide, wide + e-zoom, narrow, and narrow + e-zoom field of view (FOV) targeting options (Fig. 5). A button schematic for the controllers was also developed and is shown in Table 1.

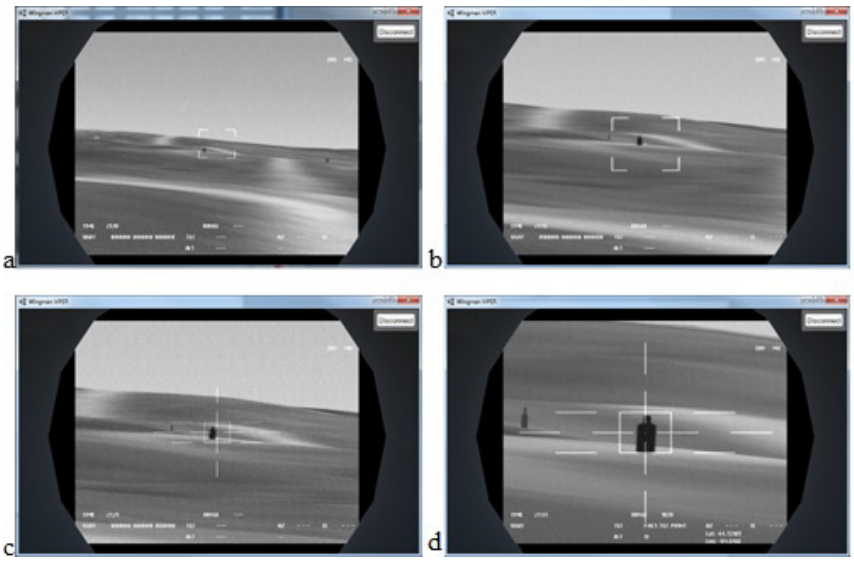

Fig. 5 Simulated LRAS3 targeting options. a) Wide FOV, b) Wide FOV with e-zoom, c) Narrow FOV, and d) Narrow FOV with e-zoom and displaying target information.

Approved for public release; distribution is unlimited.

Table 1 LRAS3 button schematic

Functionality	Grip/handlebar control	Joystick control (Logitech dual action)	Joystick and grips
1. Move LRAS unit	Turn/tilt grips	Left joystick	Use grips
2. Move user (head movement)	NA	Right joystick	. . .
3. Swivel LRAS On HUMVEE	NA	D-pad left/right on joystick	. . .
4. Toggle user view (look around/through LRAS unit)	NA	Click right joystick or Btn 2	. . .
5. Toggle LRAS E-zoom	E-ZOOM	Btn 7 - left back trigger	Use grips
6. Toggle LRAS mechanical zoom level	TRIGGER	Btn 8 – right back trigger	Use grips
7. Lase target with LRAS	LRF	Btn 3	. . .

Messaging with the commander's WMI was tested and verified at the integration workshop. Two main items were discovered that will be changed in the future. First, there are currently only two zoom levels (each with an additional electronic zoom level).Second, the movement of the LRAS3 while zoomed in fully must have finer resolution. Future work for the LRAS3 operator crewstation includes rectifying the above issues, as well as incorporating a video server to transmit the LRAS3 video to the commander's WMI. Additionally, controller maps are needed to illustrate the mappings of the gamepad and handgrip buttons.

3. Integration Challenges

Although a number of integration issues had previously been identified and rectified (Schaefer et al. 2017a, 2017b), others were identified during the integration workshop. These include the ongoing issue of sharing terrain files between ANVEL and Unity, communication between the SIL subsystems, and updating vehicle control capabilities.

3.1 Sharing Terrain Files between ANVEL and Unity

Generating higher quality terrains for ANVEL and Unity3D that can be properly aligned and synchronized between the two platforms has been an ongoing challenge. The original method for developing shared terrain files is described in

ARL-TR-8254 (Schaefer et al. 2017b). Since that time, Quantum Signal created a tutorial on generating ANVEL height map terrains using data from the USGS EarthExplorer*. This method provides higher quality ANVEL terrains than the previous method, which used mesh files from Unity3D. It also allows for multiple surface materials and vehicle dynamics across the terrain.

To test this new method, the height map was copied from the ANVEL directory. The file extension was changed from .raw16 to .raw, and the file was imported into Unity as a terrain height map. The terrain generated in Unity did not show any of the smoothing that had been done in ANVEL using the terrain editing tools. This meant that there were sharp steps in elevation, which prevents this terrain from being used in Unity. Additional exploration is required to find a terrain-smoothing solution for this new method to be a viable alternative to the current practice.

Until a finalized process is developed, adjustments were made to the existing terrain files to better align the ANVEL and Unity terrain elevations. The simulated vehicle was placed at specified points on the terrain and the latitude, longitude, and elevation from RTK was used to specify the position of the corresponding point in Unity.

3.2 Communication between SIL Subsystems

Wingman requires multiple team members to work together to accomplish joint tasks while interacting with a robotic Wingman asset. Although each team member has an independent role, the outcomes of their actions directly impact the effectiveness of the entire team. Effective collaboration between team members is dependent on the capabilities of the various subsystems. Because multiple subsystems are trying to access much of the same data, communication between them is critical for effective team operations.

3.2.1 Missing Drive Cameras in WMI

When testing with the updated WMI and RTK VM, we found that the drive camera feeds would not show up in the WMI even though the video streams could be pulled up with VLC on the WMI computer. This issue was resolved by removing the spaces in the video stream names in the /dsat_adonis/launch/config-anvel.xml file (e.g., “Drive Front” was changed to “DriveFront”). This fix had been made during the gunnery evaluation event that took place at Camp Grayling and is related to newer WMI versions. It is important to note that after this change was implemented,

* USGS EarthExplorer can be found at https://earthexplorer.usgs.gov/. It is a product developed by the US Department of the Interior US Geographical Survey.

the WMI will continue to receive a video stream from ANVEL even if the RTK is shut down.

3.2.2 PLWRWS Emulator/SPU Emulator

An SPU Emulator was developed to accommodate multiple types of controllers for use in the SIL for the robotic vehicle gunner crewstation. For the robotic gunner to accurately interact with the remote weapon system, the SPU Emulator software that allows the Xbox 360 controller to be operational was modified to match changes made in the way the ARES interprets SPU messaging. This is critical because the method that ARES uses to gain motion control of the PLWRWS changed slightly during vehicle software integration. More specifically, the message field from the PLWRWS-to-ARES and the fields of configuration (OCU) messages from ARES-to-PLWRWS changed. Additionally, the SPU button mappings for ARES changed to include a Charge button. Therefore, the mapping of the buttons on the Xbox 360 controller had to be updated to accommodate the changes in the SPU Emulator. Updating and maintaining an SPU Emulator will be unnecessary once an SPU is acquired for the SIL.

3.2.3 Loss of Localization over Time

The current organization of the subsystems in the SIL requires communication of vehicle localization data to be transmitted from ANVEL through the RTK software for vehicle mobility to ARES and then to Unity3D. This process is needed to localize the weapon system onboard the simulated vehicle. During testing, localization would not maintain an accurate position in the virtual environment if the vehicle remained in one position for too long. This was manifest in Mapviz, when the far_field_odom (orange) path rotated to a new position while the near_field_odom (green) path remained stationary. This is a critical issue because the far_field_odom shift affects the message being sent to ARES and causes the Unity vehicle to rotate suddenly to the incorrect position. However, when the RTK vehicle resumed movement, the two paths realigned and the Unity orientation was corrected. The exact cause for this is unknown and requires further investigation.

3.2.4 ARES PLWRWS Driver Control Gains

In the real-world system, the ARES PLWRWS driver control gains compensate for the nonlinear accelerations of the gimbal motors. This prevents ARES from commanding the mount to move too quickly when zoomed in. ARES can give the mount large speed commands for small movements (when zoomed in, for example) because it takes time for the mount to accelerate up to speed.

Approved for public release; distribution is unlimited.

ARES must give the virtual PLWRWS lower speeds for the same small movement because the virtual PLWRWS does not require time for acceleration. As such, the accelerations of the PLWRWS Emulator do not match those of the real-world vehicle's PLWRWS, which makes it necessary to determine the control gains for the Emulator. Since these gains are normally adjusted for each real-world vehicle mount (including different serial numbers of the same mount type), adjusting them for the Emulator is not uncommon. However, these gains are different from the ones maintained in the ARES PLWRWS driver "develop" software branch. Maintaining these control gains for the SIL while development of the real-world vehicle continues requires the SIL team to maintain a separate software branch of the ARES PLWRWS driver that includes the specific virtual PLWRWS control gains. This means that parallel updates to the ARES PLWRWS driver "simulation" branch will be required when they are made to the "develop" branch.

3.2.5 ARES Video Mux

ARES communicates to the robotic gunner primarily through a camera-based sensor on the gunner's WMI display. The integration workshop revealed that the new ARES software package (synched remap) changed the way in which ARES processes camera inputs. This caused a critical issue in the SIL because the process that Unity uses to pass video to ARES as a virtual camera was disrupted. This disruption was mitigated by installing and launching a synched remap as a regular part of ARES, along with changing the topic name where the Unity video was being published.

3.2.6 Communicating Information through the WMI Display

The WMI display is one of the key ways used by the commander, robotic vehicle gunner, and robotic vehicle operator to communicate with the weaponized Robotic Combat Vehicle (RCV). The user map is critical for all three crewstations. For Wingman, it is satellite imagery for a given area rather than a simulation world environment. This imagery is typically downloaded from a commercial online service (e.g., Google and Bing), where it is provided at varying zoom levels for optimal situation awareness and to ensure consistent WMI performance (Fig. 6). Since user maps are installed separately from the Wingman software release, certain map locations may not be enabled by default. Through editing the "C:\RVCA\config\ssmi\wsms-database.xml" file, it is possible to enable the preferred maps to match the installed maps.

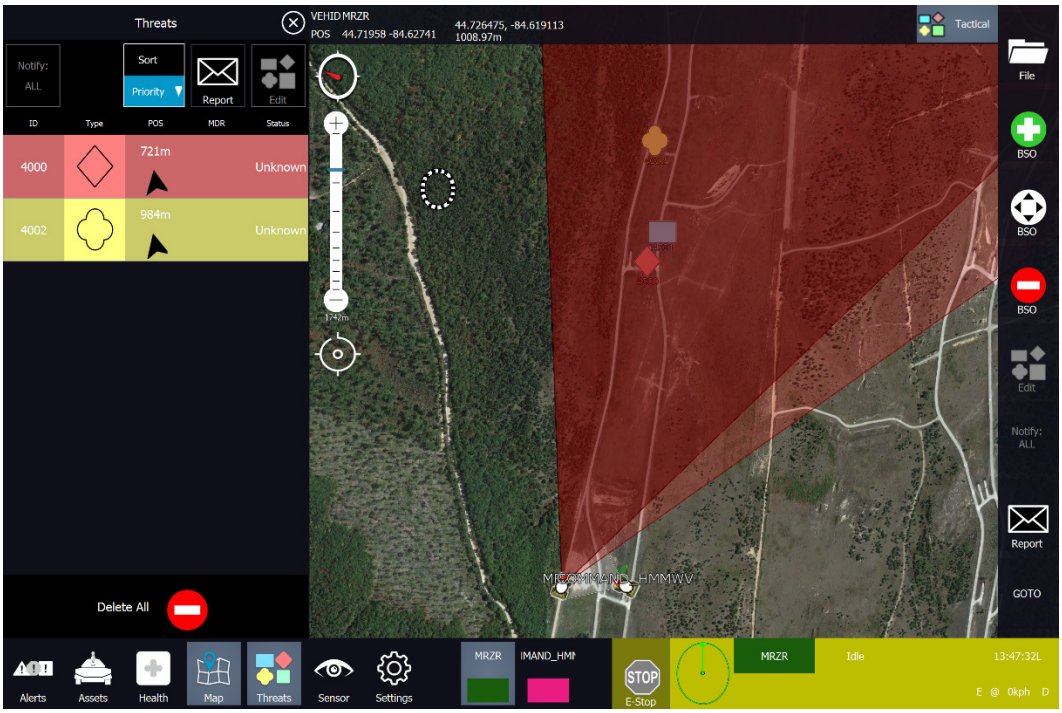

Fig. 6 Example user map in the WMI from one of the Camp Grayling gunnery ranges. The placement of both the C2 vehicle (high-mobility multipurpose wheeled vehicle [HMMWV]) and the RCV (MRZR) is shown, as well as an identified hostile threat (red diamond), friendly target (blue square), and a potential target (yellow cloverleaf).

Map imagery is important for corroborating real-world landmarks to vehicle location for purposes of vehicle teleoperation, route planning, and execution. Specific features associated with this type of map imagery are available in the Settings option. These include enabling satellite imagery (if available) through the Background checkbox, enabling and configuring grid lines in latitude and longitude, Universal Transverse Mercator (UTM) maps, and a Military Grid Reference System that includes both grid lines and grid spacing for widgets. Map features also need to communicate information about the area of operations (Fig. 7). These include adaptable battle space objects (BSOs) that provide additional context about potential threats including affiliation (friendly, hostile, neutral, unknown), class (recon, combat support, infantry, artillery, buildings, armor, aviation, weapon systems, anti-armor, combat service support, and unknown), size (section, team, squad, platoon, not selected), activity (attacking, destroyed, securing, stationary, bypassing, engaging, withdrawing, moving, delaying, fortifying, no activity, assembling, defending, reconnoitering, and unknown), and operation condition (present, damaged, fully capable, destroyed).

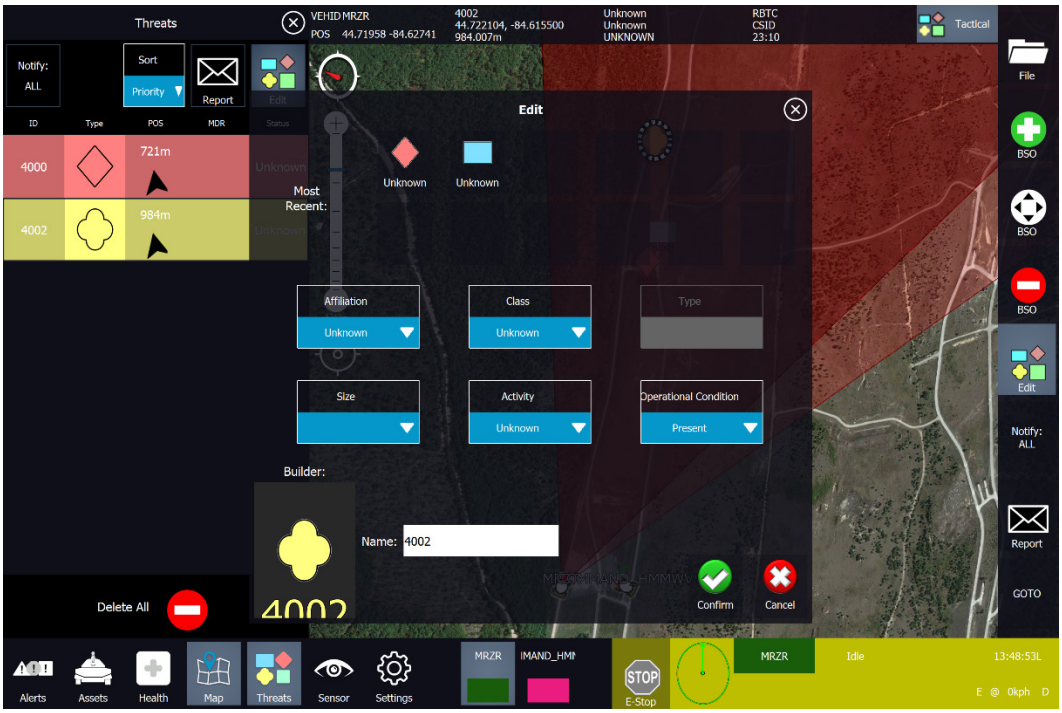

Fig. 7 Example menu options available to the vehicle commander for editing BSOs

These map features also support a large number of tactical mission graphics (TMGs) as defined in DISA (2014) (Fig. 8). TMGs may be used to denote geographical features (e.g., phase lines), custom graphics for chemical lights, other maneuvers, military operations, and so on. TMGs may be preloaded from a configuration file or added at run time on the Tactical Map using the "+ BSO" and the "Mission Graphic" buttons. The ability to draw TMGs from the WMI is a new capability for the Wingman project. Multiple issues with phase line visibility (e.g., transparency, visibility, and customizability) were reported. This issue is being tracked as TARDEC-1730, and a fix has been identified and committed to the DI2E repository.

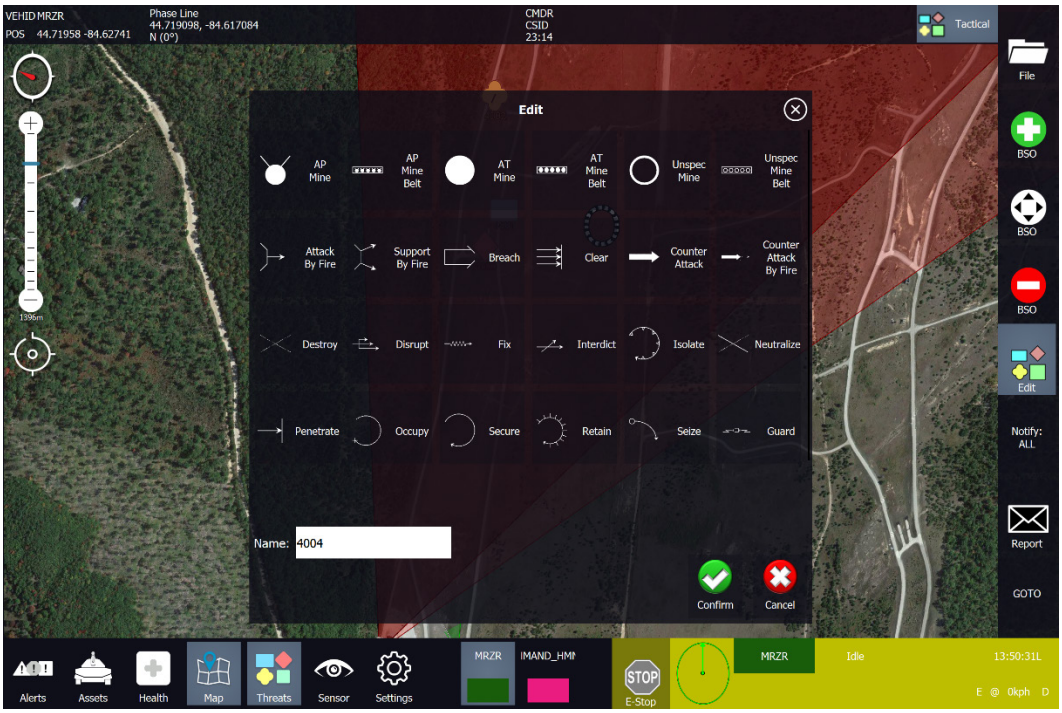

Fig. 8 Additional tactical graphics that can be marked on the user map

3.3 Robotic Vehicle Control Capabilities and Needs

Two types of autonomy-enabled control allocation are required for Wingman operation for mobility and lethality. For current operations, this includes waypoint following for the robotic vehicle operator and controller type and usage for the robotic vehicle gunner crewstations.

3.3.1 Operation of Waypoint Following

Waypoint following is a critical human-in-the-loop functional capability for mobility in Wingman operations. When testing with the new VM and updated WMI, the vehicle would not execute a loaded waypoint plan, with either waypoints or goal points. A code change for the inhabited take over (ITO) state in the controller.cpp file corrected the issue. This file resides in the sumet low-level controller ROS package at SUMET/sumet_vehicle_interface/ sumet_low_level_controller/src/controller.cpp. The change replaced the line "MSG->ITO_enabled =!roboticMode()" with "MSG->ITO_enabled=FALSE". It is anticipated that this change fixed an issue where the simulated vehicle autonomy would think it was in an ITO state. The vehicle state from the by-wire kit was not being simulated correctly. This stopgap solution was implemented during the Camp Grayling testing, but a better long-term solution, coordinated with the behaviors group, is necessary.

A second waypoint following issue was also detected during integration. The Wingman WMI uses two primary types of points in a plan—waypoints and goal points. Waypoints use an exploratory planner that does not require a large number (high density) of points to execute a plan. Goal points use a route-following planner that expects points to be close together (i.e., a high point density is required). Each works with a different mobility planner, and the RTK does not currently have the ability to switch between planners during route execution. The operator needs to choose either waypoints or goal points and not mix the two into a single mission plan. This issue is being tracked as two related issues—TARDEC-1736 and TARDEC-1772—and a fix has been identified and committed to the DI2E repository.

3.3.2 Using a Joystick Controller for the Robotic Vehicle Gunner Crewstation

The robotic vehicle gunner uses a custom handle for controlling the real-world vehicle. This handle was not available for use in the SIL at the time of the workshop. Therefore, different gamepad and joystick controllers were implemented. The Wingman WMI natively supports a variety of hand controllers (e.g., mouse, game controller, and joystick). At this workshop, a candidate joystick (CHCombatStick) was incorporated for ARES control. The default start script "start-handle-auto" was used to interpret the USB device, and the "start-Handle2Plwrws" script was used to convert the user control to map to the Armaments Research & Development Engineering Center (ARDEC) PLWRWS handgrip commands. WMI "handle" commands are natively logged in the WMI log files.

During integration, a critical issue was discovered with integration of the CHCombatStick. Specific button presses had an unanticipated outcome of introducing control messages to the robotic vehicle operator crewstation, which caused the simulated RCV to drive forward instead of independently controlling the weapon system. This issue is being tracked as TARDEC-1738, and a fix has been identified and committed to the DI2E repository. The new script guarantees that the axes are remapped and that the ARES operator will never affect the teleoperation of the vehicle. Joystick axis and button mapping are defined in the file "C:\RVCA\config\services\handle.xml".

4. Data Collection

Data collection is an essential part of research and development, as well as experimentation and training. The goal of data collection through the SIL is to set up processes and procedures that can also support real-world data collection and

after action review (AAR) capabilities. Three different types of data are collected: performance data, vehicle and system data, and user interface interactions.

The performance data are collected from Unity3D to assist with scoring on the Table VI qualification evaluation. Specific collected performance data include timing related to target initiation, target firing, and target destruction. The process for this data collection is specific to the SIL. In the field, performance and qualification data is hand-calculated by a Master Gunner or vehicle crew evaluator.

Vehicle and system data include placement and movement of the C2 vehicle, the RCV, and the weapon system. These data are collected through a ROS bag file on the RTK software module (see Section 5.1). These are the same files used to store data collected from the real-world Wingman vehicle.

It is also essential (for data analysis purposes) to record all interactions with the three WMI displays and controllers. A comma-separated values (CSV) log file was created to record every possible interaction with the WMIs and the controllers (see Section 5.2 for specific information about this log file).

Outcomes from this work will be used to develop a training program and will support research investigating the usability of both the SIL and real-world systems. Specific data will be used to support the development of AAR features.

4.1 Vehicle Data: ROS Bag Files

ROS bag files are used to store data such as vehicle state, localization, path planning, sensor feeds, by-wire commands, and diagnostic messages. The updated VM contains the new ROS bag recording scripts used during the May 2018 test event at Camp Grayling so that the same data could be collected for both the real-world vehicle and for the simulation. This new script collects data from all various topics in the RTK that could be useful in further analysis and begins standardizing it for RTK-related projects. The previous script used in the Wingman SIL only collected data related to vehicle localization and weapon state.

The scripts for running a bag file were placed in the home directory of the VM and the bag files were saved in the ~/bags directory. To start the recording of a bag file, go to the home folder (type “cd” and hit the “Enter” key) and then type “./record_bag.py” and hit the “Enter” key. Once launched, press “Ctrl + C” to return to the command line. To end the bag file recording, run “./check_bags.py” to bring up the list of currently running bag files. Press the number of the corresponding bag file to stop (this will most likely be 0) and press the “Enter” key.

One problem with running RTK in a VM is the limited hard drive space (the bag files can quickly fill the storage space if they run for a moderate amount of time). The ROS bag will automatically stop recording if the remaining hard drive space drops below a certain threshold.

4.2 WMI Log Files: Additional Recording Capabilities

The WMI Logger executable is capable of logging most of the situation awareness information that is displayed to the user. It also records the user-generated press–release gestures with pixel locations and time stamps. The WMI Logger captures the following data in CSV format:

- Sensor commands (e.g., zoom, focus, and polarity)
- Sensor data (e.g., azimuth, elevation, and FOV)
- Mobility commands (e.g., teleoperation/planning, gear shift, throttle, brake, and steering)
- Mobility status (e.g., UTM coordinate system location, elevation, yaw, pitch, roll, and velocity)
- BSO updates (e.g., UTM location, elevation, relative azimuth, and elevation to WMI's default asset)
- user gestures (e.g., press, release, swipe, pan, and bounding boxes)

Logger output may be filtered in the files "C:\RVCA\config\services \logger-[user].xml", where "[user]" is "ares", "cmdr", "rbgn", or "rbtc". The ARES and PLWRWS lethality data is not currently logged on the WMI; rather, it is collected in RTK and stored in the ROS bag file.

5. Testing

The integration workshop enabled the team to perform a full-system test to determine if certain issues were simulation- or subsystem-specific. Basic feasibility testing of proposed SIL upgrades was also performed. Currently, this type of testing can only be done by the SIL team at a fully-attended event. The subject-matter experts for each SIL subsystem (ANVEL, RTK, ARES, Unity, and WMIs) must be present to run in anything other than the "standard" configuration. Duplicate SIL simulation testbeds at TARDEC, ARL, and NSWCDD are being developed. This, along with a configuration management system, will change future "all-hands" events.

5.1 Localization and Slew-to-Cue

An ongoing challenge has been the capability of ARES to slew to an external cue. Known real-system localization issues, as well as the previously mentioned PLWRWS driver control gains, are part of the problem. The contributing issues were more thoroughly investigated to determine what may be software and what may be SIL-specific.

Specific to the SIL, some of these issues were due to a discrepancy between the ANVEL and Unity terrain height references. There is a difference between the two simulations; at any given point on the terrain, the elevations appear to differ by about 30m. The vehicle on the ground in ANVEL is displayed in Unity as about 30m above the ground. This was mitigated early in the SIL development by "clamping" the vehicle to the ground in Unity, thus providing a consistent visual representation. By creating external BSOs from the LRAS3 and commander-created on the WMI, it became apparent that the "clamping" of the RCV was masking the elevation discrepancy. Consensus was reached to resolve the issue long-term by sharing terrains enabled by the new ANVEL tools. In the meantime, elevation errors between LRAS3 client, the Unity server, and ANVEL will be adjusted manually.

An additional issue contributing to the slew-to-cue challenge was an error in the elevation sent by the LRAS3 client to the vehicle commander's WMI. The LRAS3 was incorrectly calculating the difference between the target's and the LRAS3's elevations. This error was discovered and fixed during testing.

One ongoing issue that continues to be addressed is that the weapon consistently slews to a position offset from the target position. During the hardware system integration event held at NSWCDD (which took place the week prior to this integration workshop), the technical team debugged the Wingman's localization system and incorporated a manual offset into the WMI to mitigate slew-to-cue issues with the hardware system. The SIL team should determine whether or not these mitigations are appropriate for the SIL simulation testbed.

5.2 LRAS3 Operator Crewstation

The new LRAS3 operator crewstation was tested against the vehicle commander's WMI. The handshaking and message passing as described in the LRAS3 ICD was tested and confirmed. For the primary test, the LRAS3 client was able to pass a BOM5000 (Far Target) message to the vehicle commander's WMI. The specific content of the heartbeat message was modified to accurately describe the LRAS3's current pointing direction and its current field of view. Message passing was

confirmed when a corresponding BSO appeared on the WMI. Remaining issues to resolve include better control resolution while zoomed in, implementing the correct fields of view, and deconflicting the two user controllers (handgrip and gamepad) inputs. Future work will include implementing electronic zoom and video and refining the user control interface.

5.3 Human Factors Assessment

To prepare the system for Soldier and Marine use, an early human factors test was conducted to identify potential design issues that could impact the usability of the SIL, and in turn the actual vehicle. Of particular interest were the WMI displays for the vehicle commander, robotic vehicle gunner, and robotic vehicle operator crewstations.

5.3.1 General Issues

Two general issues were uncovered: the widgets on the WMI displays needed to be adjusted to map to the SIL WMI screen size; and an issue was detected on the gimbal status on the LRAS feed. Because the current WMI SIL computers are a different size than the real-world vehicle WMI machines, some information was not visible for certain widgets. This is critical for communication of key information, including the transmittal of RCV error messages to the human team members through the WMI display. To fix this issue, a new device type is being created in the .xml file so that all widgets have adequate real estate on the SIL displays. The second issue was with the display of gimbal status on the LRAS feed. This is extra information was unnecessary and could potentially cause confusion by the user. Therefore, it was suppressed.

5.3.2 Vehicle Commander Crewstation

The primary purpose of the commander crewstation is to enable the commander's ability to provide critical information to the robotic vehicle operator and robotic vehicle gunner to support team operations. Specific features include the capability to add phase lines and BSOs on the map. Phase lines provide added information to the team regarding where to travel or areas of critical need. BSOs provide critical information regarding objects in the environment (i.e., friendly vs. hostile targets). During the assessment, some issues included the use of these features during a simulated Table VI gunnery exercise. First, phase lines were difficult to see on the map. Additional dynamic transparency features such as color, line thickness, and multipoint functions (TARDECWMI-1730) should be available for phase lines to increase the visibility and usability of this feature. Second, a critical safety issue was discovered when the vehicle commander attempted to edit the features of a

BSO while the robotic vehicle gunner was targeting a different BSO. The system interpreted the "edit icon" as a "slew-to-cue" command, so the vehicle commander's action caused the gun to slew during firing. Editing a BSO should not cause the gun slew during firing operations (TARDECWMI-1734).

5.3.3 Robotic Vehicle Operator Crewstation

The primary purpose of the robotic vehicle operator crewstation is to support the operator's capability to monitor and change the mobility features of the robotic vehicle. One such feature is waypoint following. A human factors test of this crewstation during a Table VI engagement revealed limited visibility of the individual waypoints. This made it difficult to insert new points, move current points, and delete unnecessary points. The recommendation to address this issue was to create a visibility toggle for the waypoint markers on the digital map (TARDECWMI-1732). A second issue was that only one type of marker (waypoint or goal point) can be used per path. Waypoint and goal point markers use different planners for mobility. When a combination of markers is used, the system does not know which planner to access. The short-term solution is to lock out the options so that the selected marker is the only type available for the entire path (TARDECWMI-1736). Longer-term solutions are being discussed.

5.3.4 Robotic Vehicle Gunner Crewstation

Improved transparency features on the robotic vehicle gunner WMI are needed to support target detection, target acquisition, tracking, and firing. Three changes to the sensory display (a widget in the WMI) were made prior to the integration workshop. The first change removed the "accept and reject" features from the process for communicating target assignment from the commander to the robotic vehicle gunner. In this process, the commander selected a target BSO from the target list, a message was sent to the gunner to either accept or reject the target, and the autonomous slew-to-cue function was initiated after the gunner accepted the target. As described in Section 5.3.2, additional modifications are needed to ensure that only the robotic vehicle gunner has control of the slew-to-cue features. Also, the commander needs additional ways to identify the target order so that effective communication and teaming may occur.

The second change made to the sensor display was the addition of diamonds across the top of the display that signaled the targets. The target order was updated so that the Number 1 diamond always displayed the most recent targeting information to come through the system. The problem with this feature was that a gunner could easily lose track of the current target to engage, leading to increased performance

times and added frustration with the system. Following a discussion with the larger Wingman team, the decision was made to remove this feature from the display.

The third change that was made to the sensor display was the addition of a target detection widget (Fig. 9). This is an oval-shaped icon at the bottom of the camera sensor screen that provides additional situation awareness to the robotic gunner by marking where targets are located in reference to the current orientation of the robotic vehicle. It also provides the estimated distance that the targets are away from the robotic vehicle. This type of information can improve an individual's understanding of the task and system, team communication, and performance for target acquisition, tracking, and firing.

Fig. 9 Example of the robotic vehicle gunner WMI display. The left portion is the threats screen widget where the gunner can easily identify potential targets. The center portion is the sensor display. In the sensor display, the camera sensor data come from ARES. It shows where the weapon system is aimed. Additional features include the red boxes denoting ARES-identified targets, and the newly added target detection widget (oval at the bottom of the screen). The right portion of the WMI is the map widget, as depicted in Figs. 6–8.

6. Conclusions

This section outlines future work being done on the SIL simulation testbed. The three major areas include 1) needed updates to the simulation environment, 2) planned improvements for data collection, and 3) future development of AAR technology. All outcomes will support a more integrated SIL that can be used for technical development of the software subsystems, human-in-the-loop assessment, and future training of Soldiers and Marines prior to use of the real-world vehicles.

Approved for public release; distribution is unlimited.

6.1 Updates to the SIL Simulation Testbed

This section describes five main taskers that need to be initiated in the near term. First, new vehicle models of the HMMWV are needed to better match the mobility and sensor data of the real-world RCV. Second, additional development is needed to better match terrain files in ANVEL and Unity3D. Third, since the LRAS3 operator crewstation is new to the SIL, further advancements are needed to support training and integration. Outcomes from testing during this integration workshop also identified the need for the integration of the C2 vehicle driver crewstation. Without which, it is not possible for the LRAS3 operator to be able to see all the targets on the range. Finally, specific features of the WMIs need to be updated to improve operations and teaming.

6.1.1 New HMMWV Vehicle Models

To improve research and development associated with accurate modeling of the RCV mobility and sensor feeds, the simulated RCV needs to be updated to use the HMMWV. The SIL currently uses the MRZR vehicle model, which is much smaller than the HMMWV. To reach this end, Quantum Signal has provided their updated HMMWV models for both the RCV and C2 vehicles for integration into the SIL simulation testbed. The first integration step is the addition of two drive cameras and two Velodyne 32 sensors to the vehicle definition to match the RTK launch file. These locations will be compared to the actual robotic vehicle when it returns to TARDEC. As part of this process, the RTK Light Detection and Ranging (LIDAR) filter for the simulated HMMWV will be updated to properly match the final geometry of the model and to prevent the system from viewing the vehicle itself as an obstacle. The second integration step is to either generate a new platform controller or modify the existing one so the simulated vehicle is displayed as a HMMWV on the WMI rather than an MRZR.

6.1.2 Develop Matching ANVEL and Unity3D Terrains

A new ANVEL terrain with the proper terrain orientation, terrain features and textures, and range foliage has been generated using height maps. Work is ongoing to determine how to best integrate these height maps into Unity3D—whether to import data from Unity3D into ANVEL or to finalize the new ANVEL height map method for import into Unity3D. This task is being led by TARDEC and will require support from Quantum Signal. Development of the terrain files for the Fort Benning gunnery ranges will be generated once the final method has been determined.

6.1.3 Update the LRAS3 Crewstation

At the conclusion of the integration workshop, the LRAS3 Operator Unity3D client provided the LRAS3 functions to complete the current Wingman workflow:

1) Find virtual targets in the scene (unaided, using the gamepad).
2) Find virtual targets in the scene using the LRAS3 sensors (pointing and selecting appropriate zoom levels using the E-Zoom button and Trigger).
3) Obtain localization information of a target by aiming the sensor and pressing the LRF button.
4) Automatically send that localization information to the vehicle commander's WMI.

For this crewstation to be used for training, a rudimentary task analysis should be completed to ensure the functions necessary to the role are satisfied. Anecdotally, based on discussions with people who have performed as Wingman LRAS3 operators, the current configuration should fulfill a minimal operator requirement. Future technical and functional development includes focus, sight select, and a refinement of the narrow view reticle.

6.1.4 New C2 Vehicle Driver Crewstation

With the integration of the LRAS3 crewstation into the SIL, it becomes paramount for the C2 vehicle to move through the gunnery range so that the LRAS3 operator can access all possible targets. Two options are available for implementing this feature—integrate the C2 driver crewstation or add some simulation features to virtually move the C2 vehicle. In the immediate future, the C2 vehicle driver crewstation will be developed as another Unity3D client. The station will include a steering wheel and pedals, and the driver's point of view will be from the C2 vehicle's driver seat. The driver simulation will be relatively low fidelity.

There is currently a mechanism to script the movement of the C2 vehicle within the Unity simulation. It was originally set up to move from its current battle position to a forward battle position when the robotic vehicle cleared the fourth engagement or when it passed a certain waypoint. Using this method, we can re-enable the C2 vehicle movement based on some criteria (e.g., end of an engagement, time, and robotic vehicle triggers) and automatically disable that mechanism if there is a driver client connected. Mission goals will advance the development of this crewstation design.

Approved for public release; distribution is unlimited.

6.1.5 Update the WMI Features

Updating the WMI displays will improve interaction functionality for both the simulated and real-world robotic vehicles. These updates include the following:

1) Customizability of the display
    a. Phase lines should have dynamic features such as thickness, color, multi-point features (TARDECWMI-1730).
    b. Real-time functionality between the vehicle commander and robotic vehicle gunner crewstations should be such that the commander can edit a BSO status during an active engagement (TARDECWMI-1734).
    c. Create a new device type in the .xml file for the SIL tablet interfaces to allow all widgets the real estate necessary to display all messages (TARDECWMI-1734).
2) Functionality of waypoints
    a. Once either the waypoint or goal point option is selected, the other option should be locked out (TARDECWMI-1772). This will prevent issues with the path planner.
    b. A visibility toggle is needed for waypoints once they are placed in the map. This will allow the operator added ease in moving or deleting waypoints.

6.2 Data Collection Improvements

There are four main taskers for data collection that need to be initiated in the near term. Since simulation provides a unique means to automatically collect and display Table VI performance data (rather than using traditional stopwatch and paper-pencil approaches), adapting the technical approach for scoring will support more robust training and AAR capabilities. Updates are also needed for the WMI log files (e.g., user interactions) and ROS bag files (e.g., location, elevation, LRAS3, turret location and movement) to support both SIL and real-world data collection efforts for manned-unmanned teaming interactions. But data collection is occurring on multiple Wingman subsystems. Therefore, it is critical to integrate a way to synchronize time across these subsystems.

6.2.1 Table VI Performance Data

This tasker only applies to the simulation environment. In the field, performance data are recorded and calculated using an observational approach. In Unity3D, we

are currently able to collect all significant events and associated timing data. The next steps include the following:

1) Determine if any processes are in place for auto calculation of scores, or, at a minimum, provide appropriate feedback to the Master Gunner reviewing the engagements such that scores may be calculated real-time to provide a successful AAR.
2) Save the data log files with an independent naming scheme to avoid overwriting data.
3) Communicate this performance data from Unity to the WMIs.

6.2.2 WMI Log files

The WMI log files were designed initially as an engineering tool to test the functionality of the WMI. These have been updated to collect sensor commands, sensor data, mobility commands, mobility status, BSO updates, and user press and release gestures to support data analysis related to manned-unmanned teaming. Therefore, additional descriptions of the file features, as well as all newly developed WMI widgets, are needed to improve the current logging capabilities for the development of python scripting for close to real-time data sorting and analysis.

6.2.3 ROS Bag Files

The ROS bag files (bag2csv script) need to be updated for all of the topics contained in the full RTK bag file. As an ongoing task, the RTK development group needs to review the collection scripts prior to making any additional changes to determine if anything should be added or removed for future events.

6.2.4 Synchronize Time

Synchronizing time across all the subsystems is important for accurate data collection and analysis. Consistent timestamps across the system will ensure that user effectiveness metrics are accurate. An Network Time Protocol server will need to be integrated into the SIL.

6.3 AAR Technology Development

AARs are a traditional and critical part of the qualification process on a Table VI gunnery exercise. They are typically conducted by a Master Gunner or evaluator, and are used to identify appropriate team coordination, scoring, and potential errors or failures. As we move toward a manned-unmanned team qualification, these AARs will be even more critical. The SIL offers a unique capability to provide

AAR technical support for the Master Gunner. These features will expand the SIL as a potential training tool.

The primary goal is to be able to provide the Master Gunner with adequate tools to support an effective AAR for the gunnery exercise, as well as technical support for interacting with the crewstations. Some specific subtasks include the following:

1) Support replay of engagements from both a display and overhead view (Fig. 10).
2) Incorporate set markers for key events (e.g., start of a new engagement, time when targets were active, or time when targets were destroyed).
3) Communicate performance data from Unity to the WMI.
4) Determine what type of data from the vehicle bag files is relevant for the AAR.
5) Build scripts to parse the WMI log files for quick and readable feedback on how the system was used.
6) Determine what type of information Master Gunners require for a successful AAR.

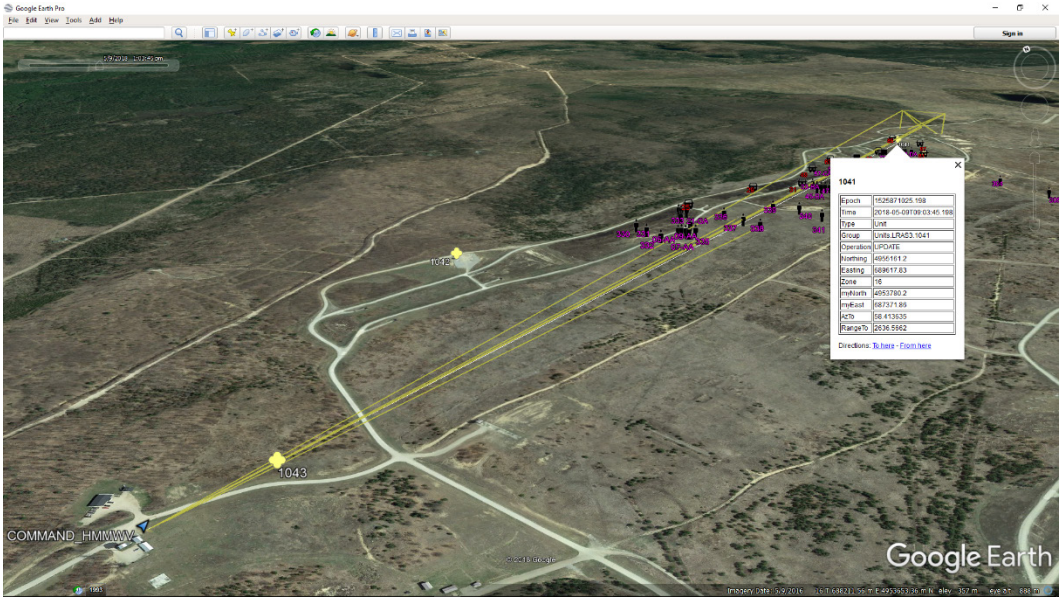

Fig. 10 Example screenshot of AAR mission replay capabilities using Google Earth. The yellow symbols denote the BSOs representing potential target locations. All troop and vehicle targets with their associated target numbers can be depicted on the map. The text box shows specific information relevant to a selected object, in this case, the BSO. The C2 vehicle with the LRAS3 markers and RCV with turret location are also integrated into this AAR playback feature.

7. References

Brewer RW, Cerame E, Pursel ER, Zimmermann A, Schaefer KE. Manned-unmanned teaming: US Army robotic wingman vehicles. In: Cassenti D, editor. AHFE 2018: Advances in Human Factors in Simulation and Modeling. Springer, Cham. 2019. p. 89–100. (Advances in Intelligent Systems and Computing; vol 780.)

[DISA] MIL-STD-2525C. Joint military symbology. Ft Meade (MD): Defense Information Systems Agency (DISA), Standards Management Branch; 2014 June 10.

Schaefer KE, Brewer RW, Perelman BS, Pursel ER, Cerame E, Drnec K, Paul V, Haynes B, Donavanik D, Gremillion G, et al. Challenges with developing driving simulation systems for robotic vehicles. In: Cassenti D, editor. AHFE 2018: Advances in Human Factors in Simulation and Modeling. Springer, Cham. 2019. p. 139–150. Advances in Intelligent Systems and Computing; vol 780.

Schaefer KE, Brewer RW, Pursel ER, Zimmermann A, Cerame E, Briggs K. Outcomes from the first Wingman software-in-the-loop integration event. Aberdeen Proving Ground (MD): Army Research Laboratory (US); 2017a June. Report No.: ARL-TN-0830.

Schaefer KE, Brewer RW, Pursel ER, Zimmermann A, Cerame E. Advancements made to the Wingman software-in-the-loop (SIL) simulation: how to operate the SIL. Aberdeen Proving Ground (MD): Army Research Laboratory (US); 2017b Dec. Report No.: ARL-TR-8254.

US Army Training and Doctrine Command (TRADOC). Training and qualification, crew. Washington (DC): Headquarters, Department of the Army; 2015 Mar 17. Training Circular No.: TC 3-20.31.

List of Symbols, Abbreviations, and Acronyms

AAR	after action review
ANVEL	Autonomous Navigation and Virtual Environment Laboratory
ARDEC	Armaments Research & Development Engineering Center
ARES	Autonomous Remote Engagement System
ARL	US Army Research Laboratory
BSO	battle space object
C2	command and control
CSV	comma separated values
CVMS	Combat Vehicle Modernization Strategy
DI2E	Defense Intelligence Information Enterprise
EO/IR	electro–optical/infrared
FOV	field of view
GUI	graphical user interface
HMMWV	high-mobility multipurpose wheeled vehicle
ICD	interface control document
ITO	inhabited takeover
JCTD	Joint Capabilities Technology Demonstration
LIDAR	Light Detection and Ranging
LRAS3	Long-Range Advanced Scout Surveillance System
LRF	laser rangefinder
NSWCDD	Naval Surface Warfare Center Dahlgren Division
OSD	Office of Secretary of Defense
PLWRWS	Picatinny Lightweight Remote Weapon Station
RCV	Robotic Combat Vehicle
ROS	Robotic Operating System

Approved for public release; distribution is unlimited.

RTK	Robotic Technology Kernel
SIL	software-in-the-loop
SwRI	Southwest Research Institute
TARDEC	Tank, Automotive Research Development Engineering Center
UTM	Universal Transverse Mercator
VM	virtual machine
WMI	Warfighter Machine Interface

1 (PDF) DEFENSE TECHNICAL
INFORMATION CTR
DTIC OCA

2 (PDF) DIR ARL
IMAL HRA
RECORDS MGMT
RDRL DCL
TECH LIB

1 (PDF) GOVT PRINTG OFC
A MALHOTRA

1 (PDF) ARL
RDRL HRB B
T DAVIS
BLDG 5400 RM C242
REDSTONE ARSENAL AL
35898-7290

1 (PDF) ARL
RDRL HRB A
J EVERETTE
BLDG E2929
DESERT STORM DR
FORT BRAGG NC
28310-0001

3 (PDF) DIR TARDEC
RDTA RTI GVR
K BRIGGS
T UDVARE
E CERAME

6 (PDF) DIR NSWCDD
CO NSWCDD H33
R POFF
T HARRIS
L BEALE
B WHEELER
E R PURSEL
M DESORMEAUX

4 (PDF) DCS CORP
B WOOD
X NHAM
J SONG
A ZIMMERMANN

7 (PDF) ARL
SFC PAUL RAY SMITH CTR
RDRL HRA M CLARKE
RDRL HRA I MARTINEZ
RDRL HRR R SOTTILARE
RDRL HRA C A RODRIGUEZ
RDRL HRA B J HART
RDRL HRA A C METEVIER
RDRL HRA D B PETTIT
12423 RESEARCH PARKWAY
ORLANDO FL 32826

1 (PDF) USA ARMY G1
DAPE HSI M SAMS
300 ARMY PENTAGON
RM 2C489
WASHINGTON DC 20310-0300

1 (PDF) USAF 711 HPW
711 HPW/RH K GEISS
2698 G ST BLDG 190
WRIGHT PATTERSON AFB OH
45433-7604

1 (PDF) USN ONR
ONR CODE 341 J TANGNEY
875 N RANDOLPH STREET
BLDG 87
ARLINGTON VA 22203-1986

1 (PDF) USA NSRDEC
RDNS D D TAMILIO
10 GENERAL GREENE AVE
NATICK MA 01760-2642

1 (PDF) OSD OUSD ATL
HPT&B B PETRO
4800 MARK CENTER DRIVE
SUITE 17E08
ALEXANDRIA VA 22350

<u>ABERDEEN PROVING GROUND</u>

19 (PDF) ARL
RDRL HR
J LANE
J CHEN
P FRANASZCZUK
K MCDOWELL
K OIE
RDRL HRB
J LOCKETT
RDRL HRB C
J GRYNOVICKI
RDRL HRB D
D HEADLEY

B LANCE
RDRL HRF A
A DECOSTANZA
RDRL HRF B
A EVANS
RDRL HRF C
J GASTON
RDRL HRF D
A MARATHE
S HILL
K SCHAEFER-LAY
RDRL VTA
G SIPHER
H EDGE
M FIELDS
R BREWER

Approved for public release; distribution is unlimited.