



Integrating Real-Time Vehicle and Watercraft Modeling and Simulation Tools for Analysis of Amphibious Operations

John G. Monroe¹ (✉), Keith Martin², Mark Ewing³, Morgan Johnston², Mary Claire Allison², Zachary Aspin¹, Collin Davenport¹, Gary Lynch², David P. McInnis¹, and Tom McKenna²

¹ Geotechnical and Structures Laboratory, U.S. Army Engineer Research and Development Center (ERDC), 3909 Halls Ferry Road, Vicksburg, MS 39180-6199, USA

John.G.Monroe@erdc.dren.mil

² Coastal and Hydraulics Laboratory, U.S. Army ERDC, Vicksburg, USA

³ Department of Electrical and Computer Engineering, Mississippi State University, 406 Hardy Road, 216 Simrall Hall, Starkville, MS 39762, USA

Abstract. Amphibious operations are complex, multi-domain problems that occur in an unpredictable environment. As such, they require knowledge and understanding of the battlespace to be successful. This report describes the development and demonstration of a proof-of-concept tool that combines existing simulation capabilities for watercraft and ground vehicles. The new multi-domain co-simulation environment allows for the modeling and simulation of amphibious operations. During operation, the ship simulator half of the Ship-to-Shore proof-of-concept tool controls an amphibious vessel during its approach to the beach. Once at the shore, primary simulation control passes to the vehicle simulator. Then the ground vehicle is either manually or autonomously maneuvered from the vessel bay, across the beach, and further inland. This combined capability provides a novel environment for ship-to-shore mission assessment. This enables enhanced planning, rehearsal, and decision support prior to mission execution.

Keywords: Amphibious operations · Modeling and simulation · Ground vehicles · Watercraft

1 Introduction

Military history shows that understanding the battlefield and having the ability to rehearse in a similar environment often means the difference between operational success and failure. This paradigm is especially true of maritime and amphibious operations where the environment and surface conditions are complex and unpredictable. Modern modeling and simulation (M&S) tools provide the capability to model the interaction between the physical environment and military systems. M&S thus enables the visualization of possible issues before mission execution. Incorporating that understanding into virtual rehearsals supports identifying hazards and exploring possible mitigation plans during the assessment of proposed operational maneuvers.

1.1 Multi-domain Operations

Multi-Domain Operations (MDOs) began as a concept developed jointly by the U.S. Army and the U.S. Marine Corps (USMC). The goal of this joint warfighting concept is to maintain superiority with regard to peer and near-peer competitors. This is partially accomplished by synchronizing theater arrival to overcome a near-peer's layered, anti-access defense [1]. Using MDO methods, U.S. forces present multiple difficulties to a competitor in air, space, land, sea, and cyberspace. Future maritime and amphibious operations will demand rapid planning and execution to provide an asymmetric advantage to U.S. forces and allies.

1.2 M&S for MDO

Leveraging M&S to inform tactics will provide decision support to forward commanders and accelerate the operational planning timeline. The work discussed in the present study is relevant to two domains – sea and land. The proof-of concept simulation technology in this work could support planning and rehearsal of amphibious landings in a single area of operation (AO) or multiple locations along a coast.

An entirely new simulation tool can take years of effort at great expense to reach maturity. Instead, this work brought together two existing tools, the Ship/Tow Simulator (STS) and the Autonomous Navigation Virtual Environment Laboratory (ANVEL) over a period of a few months to provide a usable co-simulation solution to a complex, multi-domain problem. The Ship-to-Shore (S2S) tool fits within a broad set of ERDC MDO-related capabilities but provides valuable benefits in and of itself.

During operation, the ship simulator half of the S2S proof-of-concept tool controls an amphibious vessel during its approach to the shore. Once at the shore, the vessel ramp drops, and ANVEL then controls the vehicle – maneuvering from the vessel bay, across the shore, and then further inland. This combined capability provides a novel ship-to-shore mission assessment environment.

2 Existing Simulation Tools

2.1 Ship/Tow Simulator

Since the 1980s, the ERDC STS has served as a vital engineering tool to evaluate channel design for the U.S. Army Corps of Engineers (USACE). There are many ship simulators across the U.S., but all concentrate on training mariners. Conversely, the ERDC STS, located in the Coastal and Hydraulics Laboratory (CHL), is primarily a predictive engineering tool. A multi-disciplinary team of CHL engineers and scientists manage the STS.

The ERDC STS consists of three full mission ship bridges that feature hardware to replicate an actual ship bridge. The STS bridges include 11 floor-to-ceiling screens, an Electronic Chart Display and Information System (ECDIS), a binocular channel, and a radar. The bridge controls allow pilots to operate rudders, thrusters, and throttles and give tug commands via radio. Simulations occur in real-time, and the bridges can operate independently or be connected to capture ship-to-ship interaction. The ERDC STS can

simulate the navigational conditions of ports, harbors, inland waterways, and any other maritime environment. Figure 1 shows a craftmaster piloting a Landing Craft Utility (LCU) in the STS.



Fig. 1. Craftmaster piloting LCU toward the shore in one of the STS bridges.

For a typical civil works project, channel design or modifications are evaluated prior to construction with no risk by replicating the area of interest (AOI) in a virtual environment. The virtual harbor or waterway includes three main simulation components: environmental visuals (i.e., scene), vessel performance, and hydrodynamics. A database is created to represent the existing conditions in the AOI. Pilots familiar with the AOI will then use the STS to assess the existing-conditions database to determine whether any modifications are required to create a more realistic representation of the area. Particular attention is given to vessel response and water currents. Any identified areas of concern are addressed and then retested. Once the pilots have validated the existing conditions database, a copy is modified to reflect future conditions. Then, pilots begin testing the proposed modifications thoroughly to ensure there are no navigational concerns.

The concept of using ship simulation technology in support of amphibious operational planning was initially explored in 2015 [2]. Using Cook Inlet, AK, as an AO, a simulated amphibious landing was conducted at Anchorage, AK, using the LCU 1646. Anchorage was selected due to the similarity in conditions it shares with Inchon, Korea, which witnessed the last contested landing operation by the U.S. military. During this study, the operational parameters of axes of assault, timing, and lighting were examined. The data resulting from this study included the position, heading, and speed of the LCU and other parameters affecting the transit of the LCU. These data were used by the USMC subject matter experts (SMEs) to successfully develop an operational landing plan.

2.2 Autonomous Navigation Environment Laboratory

A major task in developing autonomous unmanned ground vehicle (UGV) systems is creating robust autonomy algorithms that exhibit reliable performance in austere communication or environmental conditions. Inclement weather, harsh or poor lighting,

sensor quality, and surface conditions can all adversely affect the decision-making of an autonomous UGV. Thus, the behavior of an autonomous system must be understood or anticipated for edge case conditions. M&S tools that augment physical testing are critical to the autonomous UGV testing and evaluation (T&E) process.

For over a decade, the Mobility Systems Branch (MSB) in ERDC's Geotechnical and Structures Laboratory (GSL) has been developing and using a suite of government-owned M&S tools to assist in autonomy development and risk reduction. These tools comprise the Virtual Autonomous Navigation Environment (VANE) [3] and can be configured to operate independently or in a co-simulation. VANE provides UGV vehicle-terrain and sensor-environment simulation capabilities – be it in a high-fidelity high-performance computing (HPC) environment or in real-time desktop-based simulations.

For real-time vehicle-terrain simulation, GSL-MSB has historically used ANVEL but has begun to incorporate Unreal Engine 4 (UE4) [4]. The core of VANE's sensor-environment simulations is the Environment Sensor Engine (ESE) [5], which was referred to as VANE until the definition was broadened to encompass all aspects of ground vehicle simulations. VANE also leverages parts of open source simulation projects such as Project Chrono [6] or Simulation of Urban Mobility (SUMO) [7]. Figure 2 shows VANE configured for software-in-the-loop simulation for autonomy development and evaluation.

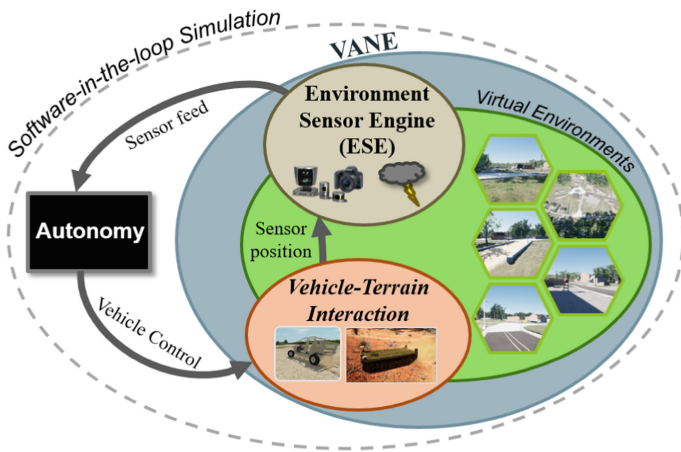


Fig. 2. VANE configured for software-in-the-loop simulations.

The VANE::ANVEL tool was developed specifically for research, development, testing, and evaluation of UGVs. Users of ANVEL build complete end-to-end, systemic models of their intelligent vehicle system, place the models in a virtual environment, and perform interactive testing while tracking key variables and collecting data from virtual sensors. Several Army robotics programs have made extensive use of ANVEL to enable early autonomous algorithm development before physical systems have been available for testing.

3 Motivation for Amphibious M&S

In 21st-century warfare, it is imperative that the U.S. military maintain rapid and asymmetric movement throughout the Pacific. The challenges presented by modern Anti-access/Area-denial (A2/AD) capabilities go far beyond those encountered during Operation Chromite at Inchon, Korea or even Normandy, France. In a testimony to the House Subcommittee on Seapower and Projection Forces, Bradley Martin of the RAND Corporation noted:

Amphibious operations have always assumed the need to overcome an opposing force and to establish a degree of battlespace dominance before attempting operations, but the reach and lethality of modern weapons systems make aspects of amphibious operations particularly challenging today [8].

Regardless of the preliminary effort required to make amphibious operations even feasible by gaining air and sea superiority in an AO, landing forces will still face significant hazards posed by the natural environment. As the examples of Normandy and Inchon show, knowledge of the battlespace is critical to success.

In the approach to the shore, water can obscure hazards to the landing craft while also being a hazard itself. As was seen at Inchon, knowledge of sea conditions is critical to know when and where a landing can be made. Once ground vehicles disembark, the slopes and soft soil that often face amphibious troops are among the most challenging terrains in terms of vehicle mobility – even without any enhancements from opposing forces. These factors make the ability to simulate multi-domain environments critical in the planning and preparation stages of amphibious operations. Integration of M&S at both the strategic and tactical levels can enable the rapid assessment of operation feasibility and allow the visualization of an operation before mission execution. Incorporating the understanding gained by virtual trials allows for the assessment of proposed operational maneuvers to identify hazards and explore possible mitigation plans.

The STS enables a craftmaster to test proposed operational maneuvers for getting a force to the shore, and co-simulation with ANVEL for vehicle-terrain interactions can enable rehearsal or analysis of short- to medium-range land maneuvers with regard to timing, space constraints, route planning, etc. ANVEL does not natively address more computationally intensive mobility issues such as trafficability, soil deformation, or long-range maneuver; but such tools exist at ERDC if a comprehensive study is desired.

Going beyond the expected utility of amphibious simulations, the context of Inchon exposes another crucial benefit. LTG Jack C. Fuson saw combat in three wars. He served with an engineer amphibious brigade in WWII conducting numerous amphibious beach assaults in the Pacific. He was a port commander during the Korean and Vietnam Wars, and at the end of his career, he served as the Army Deputy Chief of Staff for Logistics. With that breadth and depth of experience, LTG Fuson said in 1994,

If war were to occur, in all probability the [Army] Transportation Corps would be called upon to duplicate the actions of the engineer amphibious units in World War II. ... Looking back on my firsthand experience with the difficulties in learning how to accomplish this mission in World War II, I know that it would be a slow, costly, and difficult job to reinvent such capabilities in the future (qtd. in [9]).

The success of Operation Chromite in September 1950 largely depended on the fact that most of the planners and operators had amphibious landing experience from the island-hopping campaigns in the Pacific during WWII. That type of institutional knowledge has since passed out of operational memory and into textbooks. The U.S. military conducts training that includes amphibious operations, but these are expensive and dangerous, as shown by the tragic accident in August 2020 when seven Marines and a sailor drowned when their Assault Amphibious Vehicle (AAV) sank off the California coast during a training exercise [10]. M&S can augment physical training and rehearsals, which supports safe and smooth ship and craft operations.

4 Co-simulation Implementation

In the S2S simulation environment, each constituent M&S tool retains supremacy over its own traditional domain. Thus, the primary new development required to enable the proof-of-concept co-simulation tool was the creation of a communication bridge between STS and ANVEL to synchronize craft position in the two simulators. This chapter will discuss this development and the creation of a correlated virtual environment in which the co-simulation occurred during a demonstration of the S2S proof-of-concept.

Figure 3 shows a high-level overview of the co-simulation architecture. The STS software K-Sim [11], running in CHL Bridge-A, passed the position and orientation of the LCU to an ANVEL application programming interface (API) script that was controlling four instances of ANVEL. The four instances consisted of one master instance that was responsible for simulating the vehicle performance and vehicle-terrain-interaction and three viewer instances, each connected to a screen in Bridge-B. Each instance ran on a separate laptop for the sake of efficiency. No other hardware or software in Bridge-B was required. For simplicity, these four instances of ANVEL will be referred to as ANVEL_M, ANVEL_{V1}, ANVEL_{V2}, and ANVEL_{V3}. When referred to collectively, the viewer instances will be termed ANVEL_{V1-3} (or ANVEL_{V1-N}, since exactly three viewers is not necessary).

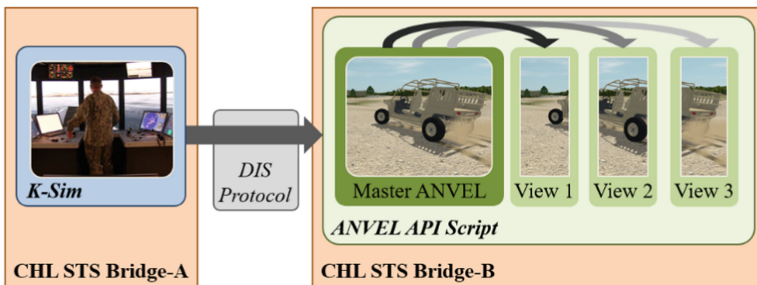


Fig. 3. Amphibious co-simulation architecture.

The co-simulation took place in two stages. In the first stage, the ANVEL simulation was a passive mimic of K-Sim until the LCU reached the shore. At that point, the second stage started, and ANVEL_M took control of the vehicle model as it was manually driven

onto the beach and then inland. During the initial demonstration of the S2S tool, the vehicle was controlled manually. However, ANVEL retains the capability to integrate autonomy for software-in-the-loop simulations. Thus, an S2S co-simulation of UGVs disembarking from a landing craft would be possible should it be required.

4.1 Ship/Tow Simulator Modifications

K-Sim does not have a native API like ANVEL. Thus, K-Sim developer Kongsberg added new communication hooks to the software in the form of a Distributed Interactive Simulation (DIS) interface [12]. This enabled synchronization of the two sides of the amphibious simulation by allowing ANVEL to receive and replicate the position of the LCU being piloted in K-Sim. To achieve a faster turn-around of the integration of K-Sim with ANVEL, Kongsberg leveraged the previous development efforts of the integration of its Polaris Simulator with the Mariners Skills Suite.

Kongsberg also provided a DIS proxy application developed in this previous effort to enable the GSL teams to test the communication protocols in the absence of an active K-Sim connection. The DIS proxy replicated the effect of passing Entity State Protocol Data Units (PDUs) to and from K-Sim. This tool allowed the ANVEL API script to be developed and tested using a recording of the LCU motion instead of requiring the full simulation configuration.

4.2 ANVEL API Code

Unlike K-Sim, no direct modification of or addition to the ANVEL software was required. Instead, all new development was external to ANVEL and utilized the native ANVEL API to communicate with and control the ANVEL simulation. Three Python files contained the code that controlled all aspects of the ANVEL simulation. These were *ANVEL_Startup.py*, *classAnvelAPI.py*, and *Ship2Shore.py*. *ANVEL_Startup* was used only as a quick reboot option following the crash of one of the ANVEL instances, which happened occasionally during development and testing. *Ship2Shore* contained all the code for communicating with K-Sim and coordinating vehicle and view motion in $ANVEL_M$ and $ANVEL_{V1-3}$. It also called *classAnvelAPI*, which contained the communication protocols to establish a connection between the core logic of *Ship2Shore* and the APIs of the ANVEL instances.

Below is a categorized list of all *Ship2Shore* classes with brief descriptions. As seen from the list, *Ship2Shore* handles setting up all ANVEL instances, getting and converting position data from K-Sim, passing that information to $ANVEL_M$, and calculating and sending the appropriate view locations and orientations for $ANVEL_{V1-N}$.

- *Supporting Code*

- *CoordinateConversions*

- Used to convert between the World Geodetic System (WSG) 84 Cartesian (K-Sim) the local (x, y) ANVEL coordinate system.

- Utilities

A collection of static methods used for various tasks including finding, starting, or killing operating system (OS) processes; joining threads; and displaying log or debug messages

- *Communication with K-Sim*

- KSimOffset

Data structure that defines calibration for coordinate transformation between K-Sim and ANVEL

- KSimComms: inherits KSimOffset

Used by ANVEL API code to send and receive live data with K-Sim

- KSimReplay: inherits KSimOffset

Used by ANVEL API code to send and receive data saved from K-Sim

- *ANVEL Multi-instancing*

- ViewVeh: inherits AnvelObject (see below)

Adds an ANVEL vehicle in an ANVEL viewer instance

- ViewANVELs: inherits Utilities

Instances of ANVEL that receive vehicle and view commands from ANVEL_M for showing ANVEL simulation on multiple bridge screens

- AnvelInstance: inherits Utilities

Contains methods for controlling vehicles and world view for ANVEL_M and ANVEL_{V1-3}

- *Conducting Amphibious Simulation*

- AnvelObject: inherits Utilities

Methods for adding, removing, and positioning ANVEL objects

- MasterVeh: inherits AnvelObject

Adds an ANVEL vehicle in ANVEL_M

- `MasterANVEL`: inherits `Utilities`

Instance of `ANVEL` that runs vehicle simulation and communicates with `K-Sim`

- `AmphibiousLanding`: inherits `Utilities`

Controls `ANVEL` simulation during Stage 1 (LCU moving), Stage 2 (vehicle moving), and the transition between them

Supports both multi-`ANVEL` simulation (i.e., `ANVELM` and `ANVELV1-N`) and single `ANVEL` simulation (i.e., `ANVELM` only)

External scripts communicate with the `ANVEL` API over a TCP/IP (Transmission Control Protocol/Internet Protocol) connection. *Ship2Shore* runs on the primary `ANVEL` computer (i.e., the one with `ANVELM`) and communicates with `ANVELM` and `ANVELV1-N` over multiple socket connections—one connection per instance of `ANVEL`. The *Ship2Shore* code could run on a computer not running `ANVEL` and still communicate with `ANVELM` and `ANVELV1-N` over IP. However, having it on the same machine as `ANVELM` reduces the overall communication time, which helps the simulation to stay real-time.

Many of the *Ship2Shore* classes and almost all of *classAnvelAPI* stemmed from existing `ANVEL` API scripts originally developed for UGV simulations. The two entirely new functionalities created to enable the S2S co-simulation were the network communication with `K-Sim` and a multi-instancing approach to `ANVEL` simulations to utilize multiple screens in one of the CHL bridges. The latter of these was not strictly necessary for the co-simulation, as only `ANVELM` actually performs vehicle simulations, and `ANVELV1-3` only duplicate the motion for viewing purposes. However, expanding the vehicle field of view by using multiple screens increases driver immersion.

4.3 Python DIS

The `ANVEL` API already provides a clean interface into the data values of simulated entities, but `K-SIM` does not provide such an interface. In order to integrate the two simulation platforms, the DIS standard was selected to facilitate data transport. DIS utilizes a User Datagram Protocol (UDP) in the transport layer and several PDUs to encode the data passed between simulations. Since there is no acknowledge/not-acknowledge phase within the UDP, the protocol is ideal for real-time simulations, since it prevents situations in which a process is iteratively trying to receive data that could be several simulation cycles old.

For simplicity, the only PDU used for MDO was the Entity State PDU (Fig. 4). The Entity State PDU contains the information that describes the vessels of interest. The bytes in the PDU are identified starting with the first byte, referred to as “Byte 0.” The bytes of interest are highlighted blue in Fig. 4 but are also given in the list below. The Entity State PDUs were exchanged between `K-SIM` and `ANVEL` at 60 Hz, which led to a smooth presentation of the LCU moving in `ANVEL`.

- Bytes 20–27 contain the description of the vessel.
- Bytes 48–71 contain the location of the vessel as double-precision Cartesian representations of the x, y, and z coordinates in meters with reference to WGS 84.
- Bytes 72–83 contain the floating point representation of the yaw, pitch, and roll of the vessel in radians.
- Bytes 84–87 are used to identify whether the LCU ramp was in the up or down position.

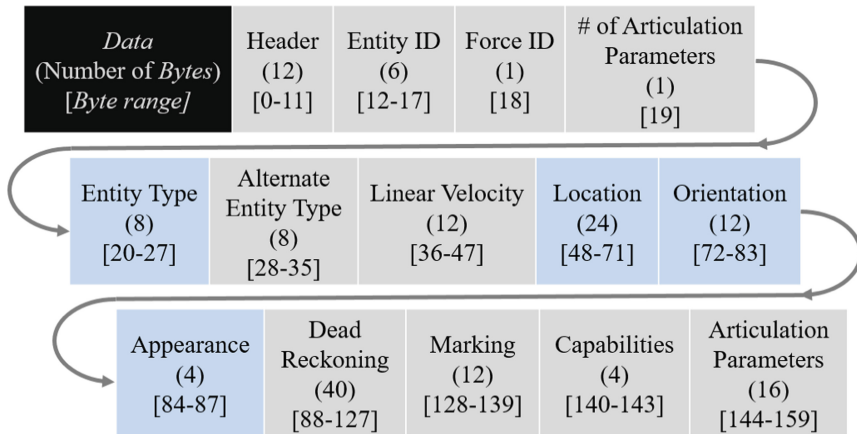


Fig. 4. Entity state PDU byte structure.

4.4 ANVEL Multi-instancing

In the STS bridges, each of the eleven floor-to-ceiling screens connects to a separate computer. During normal STS operation, an instance of K-Sim runs on each computer while a twelfth instance coordinates the overall simulation and communicates with other bridges if a multi-craft simulation is being conducted. The result is that the eleven screens display a unified simulation in real-time that is visually seamless down to the water ripples from one screen to the next. The ANVEL portion of the S2S proof-of-concept was constructed in a similar fashion.

The *Ship2Shore* Python script includes classes that allow ANVEL_M to communicate with K-Sim to get craft position, perform simulations of a single vehicle, and communicate the craft and vehicle pose to ANVEL_{V1-3}. ANVEL_M also communicates individual view orientations to ANVEL_{V1-3} to display the ANVEL simulation on multiple (i.e., three) bridge screens, with each screen connected to a workstation laptop running one of the dependent ANVEL instances. The *Ship2Shore* code supports an arbitrary number of dedicated ANVEL viewing instances—including none, in which case the ANVEL_M is responsible for both simulation and display of the ground vehicle.

To send view information from $ANVEL_M$ to $ANVEL_{V1-N}$ with as little delay as possible, the `ViewANVELs` class assigns the various $ANVEL_V$ connections to different threads on the master computer. This allows `ViewANVELs` to send the view update commands in parallel. If there are more view instances than available threads, some or all of the threads are assigned more than one $ANVEL_V$ connection. Although all threads would still execute in parallel, a thread with more than one $ANVEL_V$ instance will send commands to its members in series. However, since there were only three instances of $ANVEL_V$ during the S2S demo, each connection held a dedicated communication thread.

Originally, ANVEL was installed on the eleven bridge computers, creating $ANVEL_{V1-11}$, and $ANVEL_M$ communicated with all of them from a separate laptop. However, this approach resulted in three problems. First, the bridge computers were less powerful than the workstation laptops and failed to run ANVEL in real-time. Second, there was noticeable frame desynchronization between some of the eleven screens due to the way $ANVEL_M$ passes view orientations to $ANVEL_{V1-N}$. Finally, because the ANVEL view(s) is locked to the ground vehicle, which is shown in the 3rd person, small bumps and tilts of the vehicle caused large and violent motion on the screens farthest from the center. Additional development could alleviate this last issue by setting the view to mimic only the vehicle's rotation about the z-axis (in addition to position), instead of mimicking roll, pitch, and yaw. However, since the physical configuration of the bridge screens limits the vehicle view to the 3rd person, as opposed to a 1st person view from inside the vehicle cab, allowing the use of all eleven screens would not have added any useful information to the simulation visuals.

4.5 Virtual Environment Development

For a demonstration of the S2S proof-of-concept, a virtual representation of Mile Hammock Bay (MHB) in North Carolina, USA was developed. The geometry of both the K-Sim and ANVEL scenes was based on off-the-shelf light detection and ranging (LIDAR) from the National Oceanic and Atmospheric Administration (NOAA), but the K-Sim scene required additional data to define typical wind and current conditions that would affect the LCU. The assets in the two scenes were precisely correlated, and the coordinate transformation class in the ANVEL API script ensured that where the LCU landed in K-Sim and where the vehicle disembarked in ANVEL matched. Figure 5 shows a satellite view of the MHB site and the LCU and vehicle paths for the S2S demonstration.

The NOAA aircraft-based LIDAR data allowed the MHB site to be modeled without an in situ data collection. However, because of the relatively low resolution of the point cloud, all routes intended for ground vehicle travel had to be smoothed using ANVEL's terrain editing tools. Additionally, to simplify the scene for better performance, impassible forests at a distance from the landing area and the planned vehicle route were represented by two-dimensional walls with tree textures. Supplementary three-dimensional tree models placed in front of the walls and near the vehicle route produced the illusion of an actual forest. An AO satellite image provided the approximate locations for tree and vegetation placement as well as terrain surface types and locations.

The landing/dock area was imported into ANVEL directly from K-Sim to ensure that the geometry matched exactly during the landing procedure. However, further inland, the

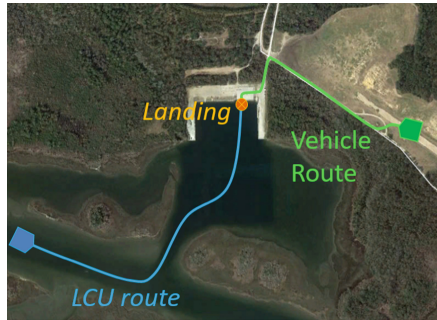


Fig. 5. Satellite image of Mile Hammock Bay with approximate demonstration path.

ANVEL scene includes buildings, static vehicles, and traffic barriers not present in the K-Sim version, as these objects were not visible from the LCU. Although the physical MHB site is devoid of such objects, during the S2S demonstration they acted to define the travel path for the ground vehicle and show the versatility of the M&S environment. M&S allows mission rehearsal to be conducted in a virtual environment both as it exists in the real world and as it may be configured in the future.

4.6 Demonstration

This proof-of-concept S2S M&S tool was exhibited alongside many other ERDC sensing and analysis tools at the Multi-domain Operations Demonstration held at the ERDC Waterways Experiment Station (WES) in August 2019. The event was attended by leaders and stakeholders from across the Army as well as from a NATO ally. When a group of visitors would arrive at the S2S part of the event, the demonstration would begin in one of the STS bridges. There a replay of an LCU landing was shown while an ERDC SME described STS and its role in the overall simulation. Although a replay was utilized for repeatability, the LCU position data was still being streamed to the ANVEL simulation in an adjacent bridge. The group would then walk into the ANVEL bridge as the LCU approached the shore (still being controlled on the ANVEL side by the data coming from the STS replay) and listen to another ERDC SME describe ANVEL and its part in the demonstration (as shown in Fig. 6). Meanwhile, the LCU would hit the beach, and another ERDC SME would take control of the vehicle and drive it inland, as shown in Fig. 5. Each demonstration went smoothly over the course of the event, and there were no communication issues between STS and ANVEL_M or between ANVEL_M and ANVEL_{V1-3}.



Fig. 6. ANVEL portion of S2S demonstration with annotations.

5 Conclusion

The S2S technology developed in this study could potentially play a significant role in the preparation and rehearsal stages of MDO. S2S M&S incorporates hydrodynamics (including waves), ship motion, vehicle performance, and terrain parameters. This enables the analysis of a landing operation from its origination in the well-deck(s) of a ship(s) stationed below the horizon to the beach to the final upland objective. By combining both sea and land domains into a single co-simulation tool, planners can assess risk and determine the timing and logistics of littoral and upland operations in real-time. This analysis allows planners to develop operational plans based on potential weather, sea, and land conditions and gives operational commanders significantly more data than previously available to make go/no-go decisions. Although simulations of various plans would need to be run in real-time, at least for the STS given the human-in-the-loop requirement, playback of recorded simulations could be played back at faster-than-real-time or condensed down to GIS-based reports for planners' analysis.

ANVEL and the STS are both valuable tools in their respective domains. However, ERDC SMEs created a new Ship-to-Shore M&S capability by combining these existing tools into a co-simulation architecture. The S2S proof-of-concept represents one of the ways the ERDC M&S tools are evolving to engineer and win the future fight. These technologies are outside the scope of this report, but [1] introduces how various ERDC capabilities can work together to support MDO. Many possible improvements are being explored to transform the proof-of-concept into a robust simulation tool for MDO. These improvements include expanding the types of operations that can be simulated, increasing the size of operations simulated, and improving the technology itself through research and development (R&D) initiatives.

In addition to the air and sea superiority that modern A2/AD capabilities require an invasion force to obtain, future maritime and amphibious operations will demand rapid planning and execution to maintain an asymmetric advantage. ERDC researchers have a broad range of expertise that has resulted in many cutting-edge tools that can support the planning and rehearsal of amphibious operations. These capabilities would allow

commanders to assess hazards throughout the ship-to-shore cycle and promote mission success by providing knowledge and virtual experience of the battlespace in a tactically relevant time frame.

References

1. Boone, N.: Engineering the Theater. *Army AL&T Magazine* 2020 (Winter), 47–53 (2020)
2. Cialone, M.A., et al.: Analysis of the effect of environmental conditions in conducting amphibious assaults using a ship simulator/vessel-response model proof-of-concept study. ERDC/CHL TR-17-4, Vicksburg, MS: Engineer Research and Development Center. <https://apps.dtic.mil/dtic/tr/fulltext/u2/1037449.pdf>
3. Kieffer, C., Holland, M., Monroe, J.G.: Modeling and simulation for unmanned ground vehicles. *Power of ERDC Podcast*, U.S. Army Engineer Research and Development Center, August 2021. <https://poweroferdcpodcast.org/modeling-and-simulation-for-unmanned-ground-vehicles/>
4. Epic Games: Unreal Engine (2019). www.unrealengine.com
5. Goodin, C., et al.: Unmanned ground vehicle simulation with the virtual autonomous navigation environment. In: 2017 International Conference on Military Technologies (ICMT), Brno, Czech Republic. IEEE (2017). <https://doi.org/10.1109/MILTECHS.2017.7988748>
6. Tasora, A., et al.: Chrono: an open source multi-physics dynamics engine. In: Kozubek, T., Blaheta, R., Šístek, J., Rozložník, M., Čermák, M. (eds.) HPCSE 2015. LNCS, vol. 9611, pp. 19–49. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-40361-8_2
7. Lopez, P., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P.: Microscopic traffic simulation using SUMO. In: 21st IEEE Intelligent Transportation Systems Conference (ITSC) (2018)
8. Martin, B.: Amphibious operations in contested environments: insights from analytic work. RAND Corporation, Santa Monica (2017). <https://www.rand.org/pubs/testimonies/CT476.html>. Accessed 13 Nov 2020
9. Boose, D.W., Jr.: *Over the Beach: US Army Amphibious Operations in the Korean War*. Combat Studies Institute Press, Fort Levenworth (2008)
10. Harkins, G.: Marine AAV hit rough seas, rapidly took on water before sinking. *Military.com* 2020. <https://www.military.com/daily-news/2020/08/03/marine-aav-hit-rough-seas-rapidly-took-water-sinking.html>. Accessed 13 Nov 2020
11. Kongsberg: K-Sim navigation: simulator system maximizing performance. Kongsberg (2020). <https://www.kongsberg.com/globalassets/digital/maritime-simulation/k-sim-navigation/docs/k-sim-navigation-brochure.pdf>. Accessed 21 Jan 2020
12. IEEE. 2012: “IEEE 1278.1-2012 - IEEE Standard for Distributed Interactive Simulation--Application Protocols”. IEEE Std 1278.1-2012 (Revision of IEEE Std 1278.1-1995)