Construction of a high-resolution under-ice AUV navigation framework using a multidisciplinary virtual environment

Supun Randeni
Massachusetts Institute of Technology
Cambridge, MA, USA
supun@mit.edu

Toby Schneider
GobySoft, LLC
Mashpee, MA, USA
toby@gobysoft.org

Henrik Schmidt
Massachusetts Institute of Technology
Cambridge, MA, USA
henrik@mit.edu

***Abstract*—We developed a new high resolution under-ice autonomous underwater vehicle (AUV) navigation framework with integrated acoustic communications, solely using a multidisciplinary physics-based virtual environment. This framework was then successfully demonstrated at the Ice Exercise 2020 (ICEX-20) conducted at 71 degree latitude in the Arctic Beaufort Sea.**

While this paper briefly describes the developed navigation framework, the primary focus is the virtual environment that was used to construct this framework. The virtual environment included sub-components to simulate the ocean environment, dynamics of the AUV, on-board sensors, arctic ice motion, ice buoy network, acoustic propagation as well as emulators to test the interface drivers between real subsystems. The navigation performance is evaluated using data from a 11 km untethered under-ice mission conducted at ICEX-20, and a similar mission conducted in the virtual environment.

***Index Terms*—under-ice AUV navigation, acoustic communication, virtual ocean, AUV simulators**

I. INTRODUCTION

Simulation environments play a key role in the development of AUV systems due to the high cost and low availability of at-sea testing time. For example, testing systems using virtual experiments can avoid excessive field troubleshooting time, which largely contributes to the success of field operations. Simulators are more crucial for developing new technologies for less accessible operational environments such as polar under-ice missions.

We have found that several key aspects are critical to successful simulations: sufficiently capturing the expected dominate sources of error in the real system within the simulated one, avoiding using the same simulated models to directly drive the real system's predictive/adaptive capabilities (thus avoiding the case where the simulator is "doomed to succeed" due to the commonality between simulated reality and predicted reality), and faithfully replicating the real system's software interfaces in the simulated one wherever possible.

The related concept of post-processing, or replaying actual sensor data from past missions, can be a powerful tool to develop new technologies and to improve existing systems, and should be employed where possible. However, this only allows testing a given mission configuration, limiting the ability to verify the compatibility of the new technology for other mission behaviors. Also, data replay tests are unlikely to utilize (and test) the entire pipeline of the embedded software system. In addition, suitable past mission data might not be available to develop new technological concepts for remote operational environments.

A. *Under-ice AUV navigation with integrated communications*

We developed a high resolution under-ice AUV navigation framework with integrated communications and tracking, which we successfully demonstrated at the Ice Exercise 2020 experiment (ICEX-20) conducted at 71 degree latitude in the Arctic Beaufort Sea using a *Bluefin 21* AUV (Fig. 1). The developed framework had a number of novel concepts and components (described in Section II); therefore, no suitable prior data were available for us in the unclassified domain that could be used for the construction of the system. Due to the high profile nature of the ICEX-20 deployment, our team only had 5 days of field time (which was reduced to 3 days due to weather) to demonstrate the system. Thus, all hardware and software system components were required to be confidently pre-tested and field ready prior to the deployment.

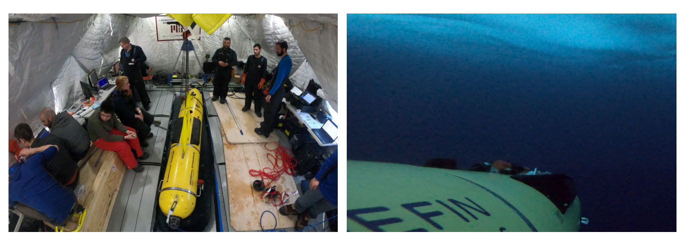

Fig. 1. The HydroMAN - ICNN navigation framework was demonstrated at the ICEX-20 conducted at 71 degree latitude in the Arctic Beaufort Sea using the MIT-owned Bluefin 21 AUV *Macrura*.

B. *A multidisciplinary virtual environment*

In this context, we developed a comprehensive virtual environment for the construction of the navigation framework.

Work was supported by the Office of Naval Research, Code 322OA and the Battelle-MIT postdoctoral fellowship program.

The virtual environment included an environmental simulator (with sub-components for the ocean environment, arctic ice motion and ocean acoustics environment), vehicle simulator (with vehicle dynamics and navigation sensor simulator sub-components), topside hardware simulator and an acoustic communications simulator with two levels of fidelity - a simpler software-only version called Netsim-UDP and a higher fidelity hardware-in-the-loop (HITL) version; i.e. Netsim [1]. The virtual environment also emulated the interfaces between the real subsystem components, which was critical to test the entire software pipeline in order to succeed by largely seamlessly switching from simulation to runtime in-field operations.

This virtual environment allowed us to construct the navigation framework with no prior data and with a minimum at-sea testing time, enabling our team to run hundreds of hours of mission simulations in various configurations to identify failure modes and to address them.

A brief outline on the developed navigation framework is given in Section II, followed by the details of the virtual environment in Section III. Section IV presents and discusses the results from the virtual environment in comparison to actual field performance.

C. Prior work

Simulation environments have been utilized for embedded software development since as early as the Apollo mission [2]. In the context of AUVs, Devie and Lemaire [3] proposed a hardware simulator for a long range AUV, which served the primary purpose of providing the software team with a tool to integrate navigation algorithms without the real platform in the loop. This proposed HITL system simulated the sensors and actuators, and their interaction with the environment. An interface layer was to provide the communication path for data between the simulator and the platform's computer.

Existing simulation environments include modular, general purpose simulator toolboxes that can be utilized by users for systems development. For example, *UWSim* [4] is a general purpose underwater vehicle simulator that provides high fidelity rigid body dynamics modeling and simulated sensors (including multibeam, contact sensors, pressure sensors, GPS, inertial measurement units (IMU), and sonars) that can be used to develop embedded software systems through robot operating system (ROS). *MORSE* [5] is a similar, but a more general purpose robot simulator that supports embedded software development in middlewares such as Mission Oriented Operating System (MOOS) and ROS.

II. HydroMAN - ICNN navigation framework

A. Autonomy and navigation system architecture

At ICEX-20, the vehicle operated under the payload autonomy architecture [6], and the manufacturer-provided software (running in the Bluefin's frontseat computer) was limited to the low-level vehicle control while the LAMSS Autonomy System running in the MIT's backseat computer was responsible for the navigation, acoustic communication and high-level autonomy (see Fig. 2). The Bluefin computer provided frontseat information and sensor measurements to the MIT's backseat computer via Ethernet, using the Bluefin Standard Payload Interface [7]. The navigation sensor suite included a 300 kHz Teledyne RDI DVL [8], Sea-Bird Electronics, Inc. model SBE 37-SI conductivity-temperature-depth (CTD) sensor [9] and a GPS. The vehicle was also outfitted with a navigation grade IxBlue PHINS C7 INS [10] and a 10 kHz WHOI Micro-Modem [11], which had direct connections to the backseat computer.

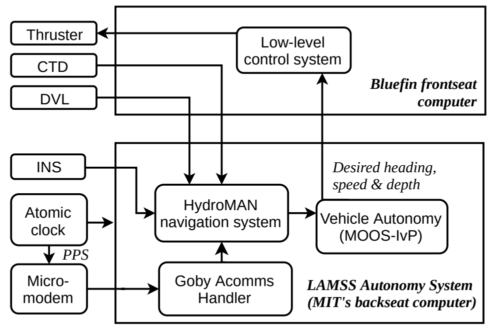

Fig. 2. Autonomy and navigation system architecture of the LAMSS Autonomy System.

The deployed navigation framework had two key components, an on-board AUV navigation engine named HydroMAN (which stands for the hydrodynamic model aided navigation), and an integrated communication and navigation network (ICNN), outfitted with a network of ice buoy modems that provided acoustic navigation aiding to the HydroMAN while simultaneously transmitting data. The HydroMAN navigation engine utilized the ICNN position updates together with the measurements from the INS, upward looking DVL, pressure sensor, a self-adapting embedded vehicle dynamic model and surface ice drift information to compute the final vehicle navigation solution.

The navigation solution computed by the HydroMAN was then provided to the MOOS-IvP vehicle autonomy software, which decided the desired heading, depth and speed. These desired commands were then sent to the Bluefin's low-level control system, which drived the thruster and gimbal mechanism in order to execute these commands.

B. Integrated communication and navigation network (ICNN)

The ICNN [12] was designed to provide communication as well as navigation aiding to underwater vehicles. The ICNN (Fig. 3) included four ice buoys, each outfitted with a WHOI Micro-Modem with a four-element receiver array and a single transmitter transducer. The receive and transmit elements were split between shallow (30m) and deep (90m) depths to provide better coverage due to the acoustic shadow zone in the new Arctic environment caused by the Beaufort Lens phenomenon (i.e. a layer of warm water has created a strong acoustic duct between approximately 100 - 200 m depth that traps sound within the duct, helping the transmission with much higher coherence and signal preservation for up to 80 - 100 km; however, causing negative impacts for communications between inside and outside of the duct even at short ranges [13]). Each ice buoy had a radio link to the ice-camp, and a GPS for time synchronization and buoy position ($x^{gps}_{\mathrm{buoy|E}}$).

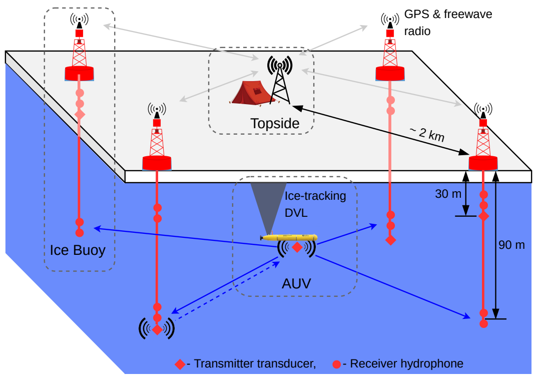

Fig. 3. The overview of the integrated communication and navigation network (ICNN) that provided navigation aiding to the HydroMAN, together with the topside tracking and command control capability.

During operations, the vehicle transmitted a datagram precisely on the second (time synchronized by an on-board atomic clock) at a pre-defined time schedule. The transmitted datagram, encoded by the dynamic compact control language (DCCL) [14], included the compressed vehicle navigation solution computed by the HydroMAN ($x^{hm}_{\mathrm{auv|E}}$). Upon successful receipt of this acoustic transmission by a buoy modem, the one-way travel time (OWTT) from the vehicle to the buoy was computed, and was sent to the topside together with the received datagram and $x^{gps}_{\mathrm{buoy|E}}$ (see the *LAMSS Autonomy System - Topside* block of Fig. 4).

When the topside received this datagram from two buoys, the *Tracker* (Fig. 4) immediately obtains the horizontal group velocities for each vehicle-to-buoy acoustic transmission path (i.e. using $x^{hm}_{\mathrm{auv|E}}$ to $x^{gps}_{\mathrm{buoy|E}}$) from the *Group Velocity Computer* in order to reliably convert the OWTTs to horizontal range. The *Group Velocity Computer* continuously tracks the horizontal group velocity using the current impulse response estimates obtained from the environmental model aided Virtual Acoustic infrastructure embedded in the topside autonomy system, with the assumption that the group velocity is a smooth function of the range.

The two horizontal ranges along with $x^{gps}_{\mathrm{buoy|E}}$ were used to compute the vehicle position ($x^{icnn}_{\mathrm{auv|E}}$) using least squares trilateration, with the aid of $x^{hm}_{\mathrm{auv|E}}$ to break the ambiguity. When datagrams arrived from a third and fourth buoy, the solution was recalculated using the same algorithm providing a new best solution. The position uncertainty covariance was computed using the trilateration error.

After a pre-defined time period, the best available $x^{icnn}_{\mathrm{auv|E}}$ was transmitted to the vehicle as a correction to the vehicle reported position; i.e. as δx, where $\delta x = x^{icnn}_{\mathrm{auv|E}} - x^{hm}_{\mathrm{auv|E}}$ (this is in order to further compress the datagram). The datagram also included the position uncertainty covariance and surface ice drift velocity ($\nu^{gps}_{\mathrm{I|E}}$) measured by a topside GPS for HydroMAN's DVL ice-track correction.

C. HydroMAN navigation engine

The HydroMAN system (see *LAMSS Autonomy System - AUV* block of Fig. 4) utilized the measurement information from a navigation grade INS, upward looking DVL, pressure sensor, ICNN navigation aiding, surface ice drift velocity, together with a self-adapting embedded vehicle dynamic model to compute the AUV's navigation solution.

The upward looking DVL measured the velocity of the AUV relative to the surface ice ($\nu^{dvl}_{\mathrm{auv|I}}$); however, the Arctic ice drifts relative to the earth (up to around 2 knots depending on the weather). To avoid navigation drifts and filter divergence, the ice drift was corrected to obtain the DVL velocity relative to earth ($\nu^{dvl}_{\mathrm{(auv|E)}}$) as given in Equation 1 by using the latest available ice drift velocity update ($\nu^{gps}_{\mathrm{(I|E)}}$) sent from the topside.

$$\nu^{dvl}_{\mathrm{(auv|E)}} = \nu^{dvl}_{\mathrm{(auv|I)}} + \nu^{gps}_{\mathrm{(I|E)}} \tag{1}$$

The ICNN navigation updates received by the vehicle as δx were converted to position ($x^{icnn}_{\mathrm{(t_{TX})}}$) by an ICNN pre-processor within the HydroMAN. The ICNN navigation updates were outdated by approximately 20-seconds when they were received by the AUV; i.e. $t_{TX} - t_{RX}$, where t_{TX} is the vehicle transmission time that was used to compute the ICNN position, and t_{RX} is the time that the vehicle received the update. The ICNN pre-processor extrapolates the navigation update to the current time-stamp ($x^{icnn}_{\mathrm{(t_{RX})}}$) by using the vehicle navigation prediction from the self-adapting dynamic model as given in Equation 2:

$$x^{icnn}_{\mathrm{(t_{RX})}} = x^{icnn}_{\mathrm{(t_{TX})}} + \left[x^{\overline{model}}_{\mathrm{(t_{RX})}} - x^{\overline{model}}_{\mathrm{(t_{TX})}} \right] \tag{2}$$

where, $x^{\overline{model}}_{\mathrm{(t)}}$ is the navigation solution from the self-adapting dynamic model at the time-stamp t.

The vehicle dynamic model embedded in the HydroMAN system was specifically designed to provide navigation aiding for AUVs by predicting the linear velocities of the vehicle u, v and w (i.e. $\nu^{model}_{\mathrm{(auv|W)}}$), based on the conservation of energy, using Equations 3 - 5 [15]:

$$\begin{aligned} u_{(t)} =& \alpha_1 N_{prop} + \alpha_2 N^2_{prop} + \alpha_3 q_{(t)} p^2 \\ &+ \alpha_4 r_{(t)} v^2_{(t-1)} + \alpha_5 q_{(t)} w^2_{(t-1)} + \alpha_6 p^2_{(t)} \\ &+ \alpha_7 q^2_{(t)} + \alpha_8 r^2_{(t)} + \alpha_9 p_{(t)} r^2_{(t)} + \alpha_{10} z_{(t)} \end{aligned} \tag{3}$$

$$\begin{aligned} v_{(t)} =& \beta_1 q_{(t)} p^2_{(t)} + \beta_2 p^2_{(t)} + \beta_3 r_{(t)} u^2_{(t-1)} + \beta_5 q^2_{(t)} \\ &+ \beta_4 q_{(t)} u^2_{(t-1)} + \beta_6 r^2_{(t)} + \beta_7 p_{(t)} r^2_{(t)} + \beta_8 z_{(t)} \end{aligned} \tag{4}$$

$$\begin{aligned} w_{(t)} =& \gamma_1 r_{(t)} u^2_{(t-1)} + \gamma_2 q_{(t)} u^2_{(t-1)} + \gamma_3 q_{(t)} p^2_{(t)} + \gamma_4 p^2_{(t)} \\ &+ \gamma_5 q^2_{(t)} + \gamma_6 r^2_{(t)} + \gamma_7 p_{(t)} r^2_{(t)} + \gamma_8 z_{(t)} \end{aligned} \tag{5}$$

where α_n, β_n and γ_n are AUV dependent dynamic model parameters that were estimated using a real-time recursive least squares system identification algorithm. $p_{(t)}$, $q_{(t)}$ and $r_{(t)}$ are measured roll, pitch and yaw angular velocities, and N_{prop} is the propeller resolutions per minute. $u_{(t-1)}, v_{(t-1)}$ and $w_{(t-1)}$ are the linear velocities estimated at the previous time-stamp using Equations 3-5

This vehicle dynamic model excludes effects due to external energy transfers; e.g. currents. Therefore, the vehicle velocity estimated by the model was relative to the water column (i.e. $\nu^{model}_{(\mathrm{auv|W})}$). To avoid navigation drifts and filter divergence, the water current velocity and dynamic model uncertainties were estimated on the fly using $\nu^{dvl}_{\mathrm{auv|E}}$ and $x^{icnn}_{(\mathrm{t_{RX}})}$, and corrected the model predicted velocity as given in Equation 6. The self-adapting dynamic model allows for a robust navigation solution during DVL ice-lock dropouts.

$$\overline{\nu^{model}_{(\mathrm{auv|E})}} = \nu^{model}_{(\mathrm{auv|W})} + \nu_{(\mathrm{W|E})} + \sigma_{\nu^{model}_{(\mathrm{auv|W})}} \tag{6}$$

where, $\nu_{(\mathrm{W|E})}$ and $\sigma_{\nu^{model}_{(\mathrm{auv|W})}}$ are the water current velocity and the uncertainty of the dynamic model velocity predictions determined by the model adapting algorithm.

The HydroMAN navigation engine included a two-layer sensor fusion algorithm; i.e. an error-state extended Kalman filters (EKF) to compute a running estimate of the bias errors of $\nu^{dvl}_{(\mathrm{auv|E})}$, $\overline{\nu^{model}_{(\mathrm{auv|E})}}$ and $\nu^{ins}_{(\mathrm{auv|E})}$. Bias corrected velocity measurements were then fused with $x^{icnn}_{(\mathrm{t_{RX}})}$ and depth measurements using a second EKF to obtain the final navigation solution.

III. Multidisciplinary simulation environment

The developed simulation environment allowed us to run virtual experiments with the LAMSS Autonomy System (topside and AUV) software in the loop, including the navigation, communication, autonomy subsystems and inter-subsystem interface drivers. The architecture is designed to make the LAMSS Autonomy System agnostic to whether it is deployed in a real or virtual environment.

Fig. 4 shows the navigation and communication focused software and hardware components, with the vehicle and topside hardware subsystems shown in green, and simulators that were developed to emulate them and their interactions with the Arctic ocean environment shown in blue.

A. Environmental simulator

The environmental simulator included a number of interconnected sub-components to replicate the physical ocean and arctic environment surrounding the AUV and topside hardware (ice buoys and ice-camp) using high-fidelity, physics-based environmental modeling infrastructures.

At the center of the environmental simulator is the virtual ocean; a representation of the physical properties of the ocean in the operational area by a 4-dimensional (time, latitude, longitude and depth) database of current, temperature, salinity, and bathymetry, produced by the HYCOM ocean modeling framework [16] (due to modular architecture of the simulator,

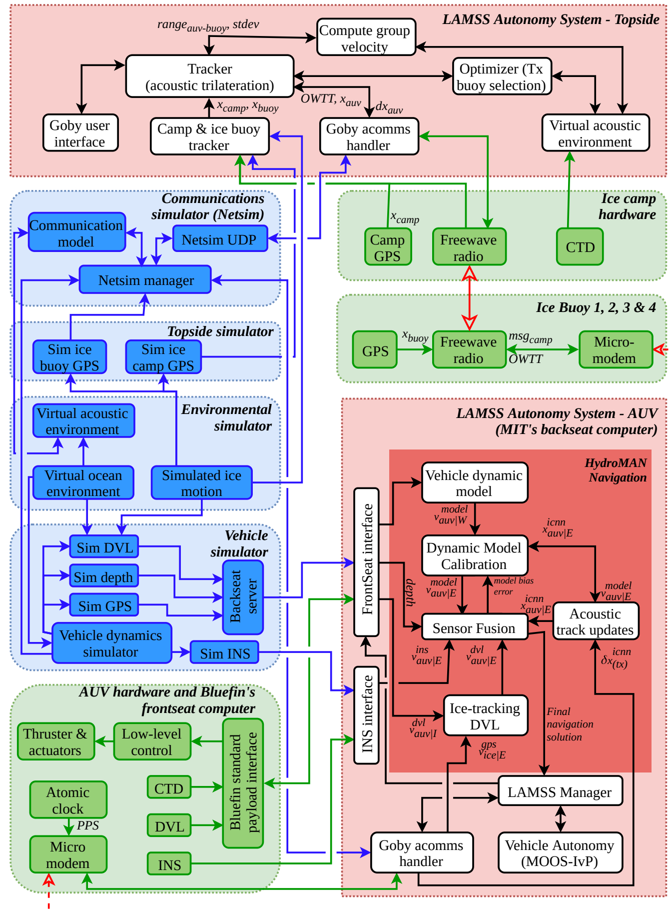

Fig. 4. The software and hardware components of the navigation framework with the *LAMSS Autonomy System - Topside* showing the software related to the ICNN, and *LAMSS Autonomy System - AUV* showing the software related to the HydroMAN. The green blocks show the AUV and the hardware components in ice buoys and ice-camp, while the blue blocks describe the simulation models that were used to replicate them and their interactions with the arctic ocean environment during the virtual experiments. These simulation models were used for the development, troubleshooting and testing of the navigation framework in the actual runtime condition.

alternative modeling and assimilation frameworks such as MSEAS [17] or historically sampled data could also be used).

The virtual acoustic modeling infrastructure of the environmental simulator utilizes the high-fidelity BELLHOP acoustic ray-tracing model [18] to produce impulse responses for the operating ocean environment using the current environmental features such as sound speed, bathymetry, etc. published by the virtual ocean sub-module. The computationally intensive, underlying ocean representation models (e.g. BELLHOP) are only required to be updated every few minutes, consistent with the slowly varying ocean.

The environmental simulator included a virtual arctic environment that simulated the surface ice thickness, and the linear and rotational motion of the surface ice. This also allowed simulation of ice fractures where ice buoys could drift independently to each other if they became separated onto different ice floes.

B. Vehicle simulator

The vehicle simulator consisted of two principal subsystems; a vehicle dynamics simulator that replicated the motion response of the AUV in the virtual ocean environment, and a sensor simulator that replicated the navigation as well as payload sensors (e.g. towed hydrophone array) on-board the vehicle to produce raw sensor measurements.

A low-level platform control sub-module in the vehicle dynamics simulator (replicating the frontseat low-level control system) consumed the backseat commands provided by the LAMSS Autonomy System (i.e. desired speed, heading and depth commands from MOOS-IvP helm) to produce the anticipated thruster and actuator commands. In response to these low-level control commands and environmental effects (i.e. water current and bathymetry information from the environmental simulator), a high-fidelity 6-degree-of-freedom (6-DOF) physics-based vehicle dynamic simulator computed the ground truth attitude, velocity and position of the vehicle.

This high-fidelity vehicle dynamics model used in the simulator was fundamentally and model parameter-wise different to the conservation of energy based dynamic model employed in the HydroMAN navigation system. The latter was specifically designed to accurately predict the linear velocities of the vehicle for navigation aiding, while the simulator was a traditional Newton's equations of motion based dynamic model that was not designed for a high degree of accuracy, but to realistically replicate the 6-DOF vehicle motion. Thus, the HydroMAN's model replicated the vehicle velocity response more accurately as compared to the simulator's dynamic model, avoiding the navigation system construction being doomed to succeed.

The simulated DVL (Sim DVL) is one of the sub-modules in the sensor simulator community, which replicated the on-board DVL. This high-fidelity simulator consumed the ground truth vehicle pose and velocity produced by the vehicle dynamics simulator and the environmental factors such as the water currents, surface ice drift, altitude, vehicle to surface ice distance in order to compute the velocity and altitude measurements in bottom-tracking, water-tracking or ice-tracking configurations. A pre-configurable error model was incorporated to include the sensor's inherent random, scale and bias errors (obtained from manufacturer's specifications) to the measurements. Simulated DVL also replicated the sensor's blanking distance and maximum range (i.e. if the vehicle to ice/bottom distance is within the blanking distance or beyond the maximum range, Sim DVL will publish blank measurements).

The simulated INS (Sim INS) utilized the ground truth vehicle information and a sensor error model (according to manufacturer's specifications) to produce the raw INS acceleration measurements. The HydroMAN navigation engine utilized the velocity output provided by the INS manufacturer's sensor fusion algorithm rather than the raw acceleration measurements. Therefore, Sim INS mimicked this process with a relatively low-fidelity algorithm to produce the velocity output. The accelerations and velocity measurements were then packed in the same message format as the actual INS.

In addition, the sensor simulator included a vehicle GPS simulator that produced position information when the simulated vehicle is on the surface, and a depth sensor simulator that provided vehicle depth information, utilizing the ground truth position information from the vehicle dynamics simulator.

C. Topside simulator

The topside simulator replicated the GPS sensors located at the ice-camp and ice buoys, utilizing the simulated Arctic surface ice motion produced by the environmental simulator.

D. Acoustic communications simulator

The acoustic communications simulator, which simulated the propagation of acoustic data packets between the simulated AUV and the ice-buoys, provided two levels of fidelity. The simpler, software-only version (Netsim-UDP) and a higher fidelity, HITL version that uses a benchtop version of the actual acoustic modems used in the field test (Netsim) [1].

The Netsim-UDP obtained the impulse responses (from BELLHOP within the environmental simulator) between the ground truth positions of the AUV and ice buoys (from the vehicle and topside simulators) to determine the transmission path. Then, the data packets were delayed according to their transmission path. Netsim-UDP also supported a configurable range to packet-success rate in order to simulate unsuccessful transmissions.

The higher fidelity Netsim version used the actual Micro-Modem signals and convolved them with the impulse response (from the environmental simulator) to provide more accurate simulated communications performance. More information regarding Netsim is given in Schneider and Schmidt [1].

E. Hardware interface emulators

The simulation environment was designed to make the complete LAMSS Autonomy System (both AUV and topside) pipeline to be deployed agnostic to whether it is launched in a virtual environment. This is accomplished by building the simulators as separate MOOS communities, and interfacing them with the LAMSS Autonomy System by using the real driver software. For instance, the *Backseat server* in the vehicle simulator converted the simulated vehicle sensor data to the Bluefin Standard Payload Interface [7]; therefore, the *Frontseat interface* of the LAMSS Autonomy System - AUV was tested in the simulation environment. This procedure was employed for all hardware interfaces (e.g. modems and topside hardware). This modularity also allowed for the simulators to be incrementally developed and implemented as the missions demand.

IV. Results and discussion

A. Actual and virtual deployments

The developed navigation framework was deployed at the ICEX-20, where we conducted various missions including an untethered 11 km run at various ranges and depths on the March 11, 2020. In this discussion, we compare the navigation

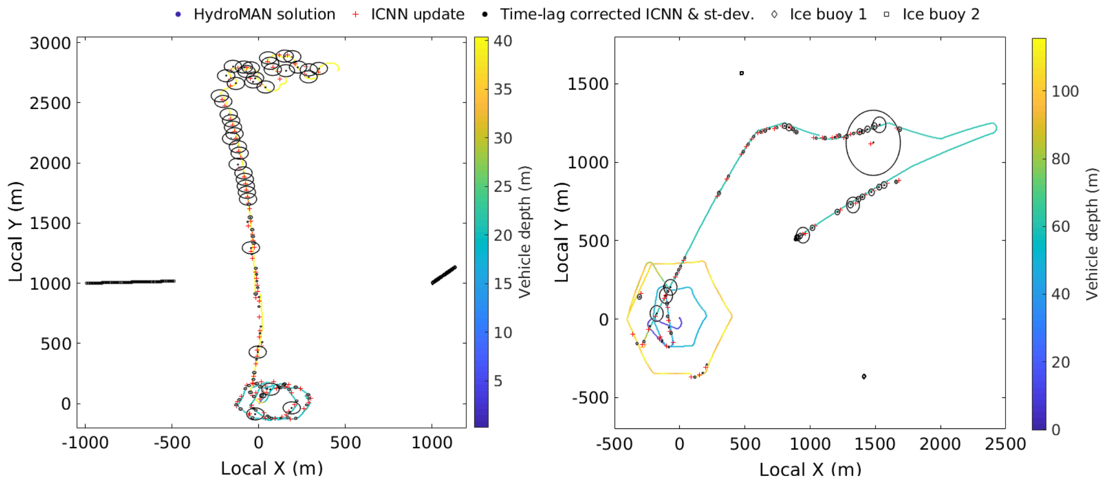

Fig. 5. Right-panel: the navigation track of the AUV from a 11 km untethered mission conducted during ICEX-20 at Beaufort Sea, Arctic on March 11, 2020. Left-panel: the navigation track of a similar mission conducted in the virtual environment.

performance of this run against a similar mission conducted in the virtual environment (using Netsim as the communications simulator).

Fig. 5 left-panel shows the navigation track of the simulated mission while the right-panel showing the actual mission with the line color corresponding to the vehicle depth below the surface ice. The raw ICNN navigation-aiding updates as received by the vehicle are plotted in red cross symbols while the time-lag corrected ICNN position updates are shown in black dots with the black circles around them illustrating the position error covariance.

The time-series of the ice buoys 1 and 2 (Fig. 5) shows that the surface ice drift was lower during the actual mission presented here due to the calm weather. However, this drift can become significant (i.e. >0.5 m s $^{-1}$) depending on weather, and in such events correcting the DVL velocity becomes mission critical. In the simulated mission presented here, the ice-camp and ice buoys 2 and 3 were coupled together and they had a drift speed of 0.13 m s $^{-1}$ (heading east) and a rotation rate of 1 degree per hour while the ice buoy 1 was decoupled from the ice-camp with a drift speed of 0.05 m s $^{-1}$ (heading north-east). Thus, we were able to develop and test the HydroMAN system for potentially worse scenarios.

The virtual environment allowed us to simulate various potential failure modes. For example, in the simulated mission presented here, the ICNN network was limited to three ice buoys. Therefore, when the vehicle was beyond 1.5-km north of the ice-camp (ice-camp was at 0,0), most vehicle transmissions were not successfully received by the ice buoy 3, limiting the ICNN updates to 2-modem solutions. As a result, the ICNN navigation-aiding updates had larger error covariances. However, the ICNN-HydroMAN framework was able to provide a robust solution throughout the mission.

One of the indexes we used to evaluate the navigation accuracy is the navigation correction made by the ICNN. Fig. 6 upper and lower panels shows the statistical distributions of the ICNN navigation corrections for the simulated and actual

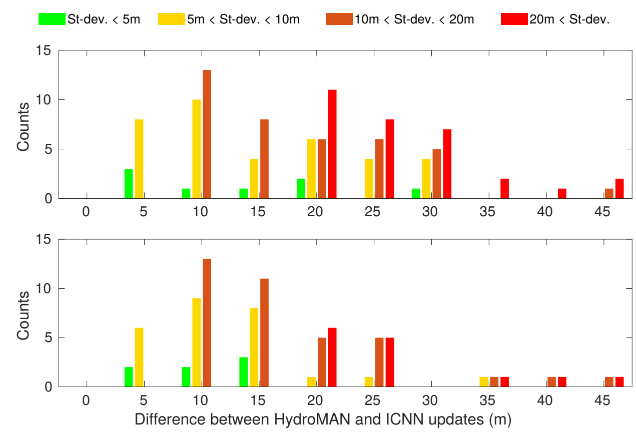

Fig. 6. The statistical distributions of the ICNN navigation corrections for the simulated (upper-panel) and actual (lower-panel) missions. The observations are categorized by the standard deviations of the ICNN navigation-aiding updates.

missions (respectively). The observations are categorized by the standard deviations of the ICNN navigation solution.

In the actual mission, the ICNN corrections were below 15 m for over 88% of ICNN updates with standard deviations less than 10 m, and most of the updates that provided larger corrections had higher uncertainties. For example, all ICNN solutions with a standard deviation above 20 m had a correction above 20 m.

A similar trend was seen from the virtual experiments. Detailed data analysis and observations showed that the actual system provided better navigation performances as compared to the virtual environment. This is since we have used larger error models for the simulated sensors than their actual hardware systems. This is critical to succeed when the systems are entirely constructed in a virtual environment.

During operations, the buoy optimizer algorithm in the topside community (Fig. 4) decided the optimal receiver hydrophone layer (i.e. deep or shallow), and the transmission buoy, in order to maximize the communication and navigation performance. When the shallow hydrophone layer was used in actual missions, the standard deviations of the ICNN solutions were below 10 m, and all updates provided a navigation correction less than 15 m. That is, the majority of the inaccurate navigation updates were from the deep layer (90 m deep), likely caused by drifts in ice buoy mooring lines.

The ICNN-HydroMAN navigation framework was able to provide navigation accuracies in the low tens of meters, on the same order of magnitude as GPS in the high latitudes. Hundreds of hours of simulations were conducted by our team during the development, troubleshooting and testing phases of the navigation framework, and vehicle autonomy software. This high-fidelity simulation environment was the primary reason for the successful field results during an operation conducted within an extremely tight time window.

B. Potential improvements

The fidelity of the virtual environment can be further improved in a several directions. For example, the vehicle simulator did not include the Bluefin frontseat computer's safety features. Embedding them into the vehicle simulator would help foreseeing potential mission failures due to issues in the manufacturer's software system.

The simulated ice buoys in the topside simulator assumed that the mooring lines are vertical; i.e. excluding the potential drifts in the deep hydrophones, and results provided evidence for such drifting. Thus, modeling the mooring dynamics due to currents in the virtual ocean would be beneficial. Improving the fidelity of the ice buoy simulator will allow the developers to improve the ICNN algorithm; for example, inter-buoy transmissions can be conducted during AUV operations to also localize the ICNN modem positions (both shallow and deep), and hence identify uncertainties such as mooring line drifts by comparing with the buoy position from the GPS. Identified drifts can be then used to counteract the vehicle navigation uncertainties (i.e. similar to the concept of real-time kinematic (RTK) GPS positioning technology).

V. Conclusions

We developed the ICNN - HydroMAN navigation framework solely using a multidisciplinary virtual environment. This framework was then successfully demonstrated at ICEX-20 conducted at 71 degree latitude in the Arctic Beaufort Sea. This effort shows the value of virtual environments to develop new AUV technologies with a minimum at-sea testing time.

It was critical to simulate the real environmental conditions and their interactions with the physical systems; however, it was also critical to emulate the interface drivers between the real subsystem components in order to succeed. The virtual environment must not share the same models (e.g. vehicle models and ray tracing models) with the developed system in order to avoid the virtual environment being doomed to succeed. In addition, the modular construction of the virtual environment allowed for the simulators to be incrementally developed and implemented as the missions demand.

Acknowledgment

We thank the developers of the open source software components used in this work: Paul Newman (MOOS), Mike Benjamin (MOOS-IvP), and Mike Porter (BELLHOP), and present and past members of LAMSS who contributed to the incremental development of the LAMSS Autonomy System.

We greatly appreciate the help of Dennis Giaya, Lee Freitag, and the team in the Micro-Modem group at the Woods Hole Oceanographic Institution for their excellent support for the ICEX-20. Also, many thanks to the Arctic Submarine Lab, UIC Science, General Dynamics - Mission Systems and the rest of the ICEX-20 team for making this experiment possible.

References

[1] T. Schneider and H. Schmidt, "Netsim: A realtime virtual ocean hardware-in-the-loop acoustic modem network simulator," in *2018 Underwater Communications and Networking Conference (UComms)*. IEEE, 2018.

[2] J. A. Lawrence and H. E. Smith, "The role of jsc engineering simulation in the apollo program," *Simulation*, vol. 57, no. 1, pp. 9–16, 1991.

[3] F. Devie and J. Lemaire, "A flexible hardware in the loop simulator for a long range autonomous underwater vehicle," in *IEEE Oceanic Engineering Society. OCEANS'98. Conference Proceedings (Cat. No. 98CH36259)*, vol. 3. IEEE, 1998, pp. 1359–1363.

[4] M. Prats, J. Perez, J. J. Fernandez, and P. J. Sanz, "An open source tool for simulation and supervision of underwater intervention missions," in *2012 IEEE/RSJ international conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 2577–2582.

[5] G. Echeverria, N. Lassabe, A. Degroote, and S. Lemaignan, "Modular open robots simulation engine: Morse," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 46–51.

[6] M. R. Benjamin, H. Schmidt, P. M. Newman, and J. J. Leonard, "Nested autonomy for unmanned marine vehicles with moos-ivp," *Journal of Field Robotics*, vol. 27, no. 6, pp. 834–875, 2010.

[7] D. Goldberg, "Huxley: A flexible robot control architecture for autonomous underwater vehicles," in *OCEANS 2011 IEEE-Spain*. IEEE, 2011, pp. 1–10.

[8] Teledyne, *Navigator Doppler Velocity Log (DVL) Technical Manual*, Teledyne RD Instruments, 2017.

[9] Seabird, *SBE 37-SI MicroCAT CT(D)*, Sea-Bird Scientific, 2018.

[10] iXblue, *Phins Compact C7 - Technical Description*, iXblue, 2019.

[11] L. Freitag, M. Grund, S. Singh, J. Partan, P. Koski, and K. Ball, "The whoi micro-modem: An acoustic communications and navigation system for multiple platforms," in *Proceedings of OCEANS 2005 MTS/IEEE*. IEEE, 2005, pp. 1086–1092.

[12] T. Schneider, H. Schmidt, and S. Randeni, "Self-adapting under-ice integrated communications and navigation network," in *2020 Underwater Communications and Networking Conference (UComms)*. IEEE, 2020.

[13] H. Schmidt and T. Schneider, "Acoustic communication and navigation in the new arctica model case for environmental adaptation," in *2016 IEEE Third Underwater Communications and Networking Conference (UComms)*. IEEE, 2016, pp. 1–4.

[14] T. Schneider, S. Petillo, H. Schmidt, and C. Murphy, "The dynamic compact control language version 3," in *OCEANS 2015-Genova*. IEEE, 2015, pp. 1–7.

[15] S. Randeni, E. M. Fischell, and H. Schmidt, "An AUV dynamic model based on the conservation of energy for underwater navigation aiding," *IEEE Journal of Oceanic Engineering (Under Review)*, 2020.

[16] A. Wallcraft, E. Metzger, and S. Carroll, "Software design description for the hybrid coordinate ocean model (hycom) version 2.2. naval research laboratory tech. rep," NRL/MR/7320-09-9166, 155 pp.[Available online at https://hycom. org , Tech. Rep., 2009.

[17] K. D. Heaney, P. F. Lermusiaux, T. F. Duda, and P. J. Haley, "Validation of genetic algorithm-based optimal sampling for ocean data assimilation," *Ocean Dynamics*, vol. 66, no. 10, pp. 1209–1229, 2016.

[18] M. B. Porter, "The bellhop manual and users guide: Preliminary draft," *Heat, Light, and Sound Research, Inc., La Jolla, CA, USA, Tech. Rep*, 2011.