First Step Towards μNet: Open-Access Aquatic Testbeds and Robotic Ecosystem

Scott Mayberry
Junkai Wang
Qiuyang Tao
Fumin Zhang
mayberry@gatech.edu
wangjk@gatech.edu
qtao7@gatech.edu
fumin@gatech.edu
Georgia Institute of Technology
Atlanta, Georgia, USA

Shuai Dong
Connor Webb
Aijun Song
Xiaoyan Hong
sdong7@crimson.ua.edu
clwebb@crimson.ua.edu
song@eng.ua.edu
hxy@cs.ua.edu
The University of Alabama
Tuscaloosa, Alabama, USA

Dmitrii Dugaev
Zheng Peng
ddugaev@gradcenter.cuny.edu
zpeng@ccny.cuny.edu
The City College of New York
New York, New York, USA

ABSTRACT

Aquatic-based research requires a special set of facilities, expertise, and supporting personnel, all of which are costly and often not accessible to small research groups. A shared research infrastructure, particularly testbeds and open-access software, would lower the barrier to entry, decrease implementation time, and mitigate the risk of failure in harsh underwater environments. We have made the first step towards developing μNet, an open-access aquatic testbed and robotic ecosystem addressing the need for shared infrastructure. Our contributions include the initial steps towards an indoor testbed, an outdoor testbed, and an open-access software suite.

CCS CONCEPTS

• **Hardware** → *Networking hardware*; *Sound-based input / output*; *Physical verification*; **Sensor devices and platforms**; • **Networks** → **Network architectures**.

KEYWORDS

μNet, muNet, testbed, open-access, open-source, underwater, aquatic, network, ROS, MOOS-IvP, Aqua-Net, communication, surface vehicle, software, hardware, ecosystem

ACM Reference Format:
Scott Mayberry, Junkai Wang, Qiuyang Tao, Fumin Zhang, Shuai Dong, Connor Webb, Aijun Song, Xiaoyan Hong, Dmitrii Dugaev, and Zheng Peng. 2021. First Step Towards μNet: Open-Access Aquatic Testbeds and Robotic Ecosystem. In *WUWNet '21: The 15th International Conference on Underwater Networks & Systems, Nov 22–24, 2021, Shenzhen, China.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3491315.3491322

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
WUWNet'21, Nov 22–24, 2021, Shenzhen, China
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-9562-5/21/11…$15.00
https://doi.org/10.1145/3491315.3491322

1 INTRODUCTION

Ocean-related technologies, especially the infrastructure for observing and exploring the aquatic domain, are critical to our nation's economy and security. The recent 2018 November report [28] by the National Science & Technology Council identifies research infrastructure as "one of the highest priorities of the ocean science and technology community." In recent decades, autonomous underwater vehicles (AUVs) and autonomous surface vehicles (ASVs) have emerged as effective tools to collect samples in oceans, lakes, and estuaries [2, 5, 22, 30] for various underwater applications. The next research frontier is to use fleets of aquatic robots to perform distributed sampling in aquatic environments [19], including 1) tracking marine life to understand the life cycles of sharks, jellyfish, lobsters, etc. [20]; 2) monitoring and tracking fast-evolving plumes, algae, or other dynamic features [15]; and 3) search and rescue or other dense sampling missions [12, 26], enabled by the drastically decreasing costs of individual aquatic robots.

This exciting future has attracted increasing interest from computing and engineering research communities. However, high barriers of entry exist, preventing cross-disciplinary collaborations among the related research fields. Lack of shared infrastructure and, as a result, lack of research participation are the fundamental issues. Mobile underwater networking research requires specialized facilities, expertise, and supporting personnel, which are costly and often not accessible to small research groups. AUVs, ASVs, and underwater modems are still expensive to individual investigators even though the last decades have seen drastic price decreases for aquatic mobile platforms. Further, the underwater community does not have open-access mobile infrastructure to support broad research participation.

Responding to the infrastructure demand, we are developing an open-source, open-architecture, community-shared infrastructure for mobile underwater acoustic networks, namely μNet. The μNet infrastructure consists of three main components: an indoor testbed, a lake testbed, and an underlying open-architecture software suite. The overall goal is to support research and education in the joint communities of underwater signal processing, communications, networking, and robotics. We aim to expand research participation through the μNet infrastructure.

For applications requiring real-time communication, such as swarm control or an underwater Internet of Things (IoT), inter-vehicle communication utilizing acoustic communication (ACOMMS) is critical. However, ACOMMS have limited range and bandwidth, and carry significant overhead complexity in implementation. A network stack capable of balancing the limited bandwidth, range, and high complexity of an ACOMMS based system could provide a multitude of capabilities to the community such as: swarm communication between unmanned marine vehicles without the need for surfacing, surface data relayed throughout underwater network through anchor nodes, acoustic localization, and the ability for mesh networking for data relaying to address the limited communication range. However, the aquatic-based communities utilize separate systems, commonly MOOS-IvP and ROS, and neither is able to support an underwater communications network. Both MOOS-IvP and ROS need to be enhanced so as to support the emerging applications. In this paper, we introduce the initial efforts of developing adapters in MOOS-IvP and ROS so that the existing Aqua-Net, a system package that supports all the functions at the protocol stack for underwater acoustic communications and networking, can be used by the aquatic-based communities for their research and applications.

For mobile underwater platforms, existing networking efforts can be loosely sorted into two categories. In the first category, software development is centered on robot autonomy, building middle-ware over OSI network layers. Often, MOOS-IvP is used [2]. Nested autonomy, adaptation, and a modular acoustic networking framework for a single-hop network have been developed with MOOS-IvP [22–24]. The European UAN project further added a P2P communication structure to support emergent event reporting [3]. In the second category, software development is centered on networking distributed static sensors and mobile nodes. Many research and development efforts have enriched the network layer and MAC layer protocols as well as additional modules to support underwater communications. The platforms include Aqua-NET, SEANET, DESERT, SUNSET, UnetStack [4, 7, 10, 14, 16]. This division mirrors the two main underwater technology driving forces – AUVs and underwater sensor networks. The former focuses strongly on command-control and adaptable components while the latter focuses on networking protocols for both static and mobile nodes. **So far, there are no full-suite hardware and software solutions for mobile underwater networks that are accessible to the community.** The proposed open-source μ-Net framework merges the functionality of network-centric and autonomy-centric frameworks to support flexible networking capability in collaborative mission operations.

The indoor testbed is for lab tests with a moderate number of nodes, and uses multiple miniaturized robots developed at Georgia Tech (GT), GT-OSVs and GT-MURs (the latter not discussed in this paper). The two types of robots are equipped with low-cost, open-source acoustic modems. The testbed will support two means of communication among the robots. WiFi links are used to connect all the robots above the water surface. Acoustic modems support communications underwater. Our objective for the indoor testbed is to provide remote access to the community. With its low-cost, open-source design, the indoor testbed can be replicated by community users.

The lake testbed has fewer nodes than the indoor testbed. It uses two commercial autonomous underwater vehicles (AUVs) and one boat that can be deployed in a lake. The commercial software-defined acoustic modems are installed on the boat and AUVs to perform digital communications or waveform transmissions. Due to the nature of field experiments, it is not possible to provide remote access. We will conduct scheduled field tests in a local lake on behalf of the community users.

Discussed in this paper are overviews and current statuses of the open-access software, the lake testbed, and the indoor testbed.

2 OPEN-ACCESS SOFTWARE

μ-Net is an open-access infrastructure for underwater networks and robotics. Different from existing platforms such as SEANET, DESERT, SUNSET, UnetStack [4, 7, 10, 16], μNet attaches great importance to bridging the gap between the robotics and networking communities. Therefore, μNet emphasizes the support of ROS [9] and MOOS-IvP[1], two popular robotic software frameworks. In the meantime, by leveraging Aqua-Net [14], an underwater networking suite, we can enable the inter-vehicle networking capability for both ROS and MOOS-IvP.

2.1 ROS and MOOS-IvP

ROS is a widely-used open-access software framework for robot software development. Running in Unix-based platforms, ROS provides services such as hardware abstraction, low-level device control, utility functions, message exchanges and package management. In order for ROS processes, or *nodes* in ROS terminology, to communicate with each other, the ROS system adopts a publisher/subscriber model. A ROS node can create a *topic* and any nodes subscribing to the *topic* will be able to receive the messages in the *topic*.

MOOS, short for "Mission Oriented Operating Suite", is a suite of software modules that organizes different software processes running on an autonomous vehicle. It provides a core middleware (MOOS Core) consisting of a database (MOOSDB) and a publish and subscribe system. The applications in MOOS exchange information through topics of interests maintained by MOOSDB through the pub/sub system. IvP, standing for "interval programming", for multi-task optimization is developed and added as a specific MOOS application, namely, IvP Helm. IvP Helm receives the information sent by different sensor applications and makes control decisions for the unmanned marine systems. The information exchanges are all through MOOSDB and the pub/sub system.

2.2 The Aqua-Net integration

Both ROS and MOOS-IvP lack the capability of supporting underwater inter-vehicle communication among multiple unmanned marine vehicles. Ideally, since the transmission range of acoustic signals is limited, the ability of relaying packets among unmanned marine vehicles can be adopted.

Aqua-Net [14] presents a suite of protocols for underwater wireless networks. It can host protocols from transport, network, and medium access control layers. Each protocol or module operates as an independent unit in the OS. The protocol stack is flexible and re-configurable with a number of implemented networking protocols

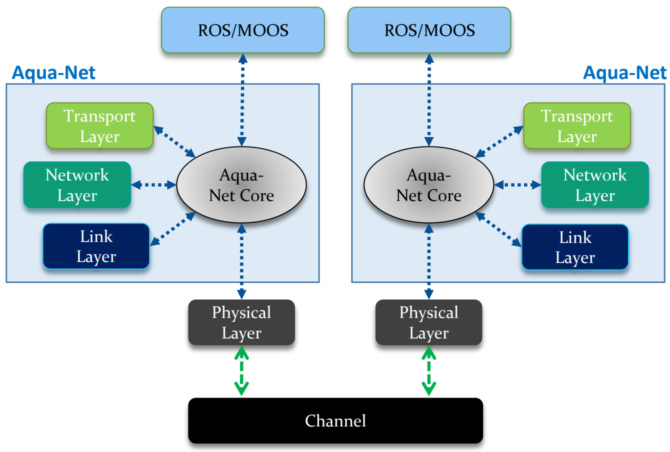

Figure 1: ROS and MOOS applications can work as Aqua-Net users

and modem drivers [8, 13, 17, 18, 29], some of which [13, 17, 18, 29] have been empirically tested in field experiments.

In this project, we integrate Aqua-Net into ROS and MOOS-IvP ecosystems to enable the networking capability of Aqua-Net. As shown in Fig. 1, both ROS and MOOS applications can work as Aqua-Net users. Through Aqua-Net APIs, they will be able to communicate with the corresponding applications on remote systems without worrying about networking problems such as the medium access control, multi-hop routing, and/or reliable data transfer.

2.3 Case Study: Aqua-Net with ROS

As a case study, we will discuss how ROS can operate with Aqua-Net. Fig. 2 demonstrates a four-vehicle scenario in which ROS and Aqua-Net work together for inter-vehicle communications.

In a ROS system, there is an essential component called the ROS core. The ROS application software needs to connect to a ROS core in order to interact with another ROS application software. This connection can be local, as shown in Fig. 2 between *App1* and *App2* on *Vehicle1*. This connection can also go through the Internet when the vehicle is on the surface and WiFi is available. For example, *App4* on *Vehicle4* is able to reach *Vehicle2*'s ROS core via the Internet and thus talk to *App4* on *Vehicle2*.

What Aqua-Net can bring to the ROS system is the ability to form a network under the water. We developed an Aqua-Net adapter for ROS, illustrated as light blue boxes in Fig. 2. The ROS application software can attach to the ROS core as before, but their data packets can be handed over to Aqua-Net by the adapters. As shown in Fig. 2, *App3* on *Vehicle 1, 2 and 3* can communicate with each other. If these vehicles are within the transmission range with each other, then the medium access control (MAC) protocols in Aqua-Net will address the packet collisions. If these three vehicles form a multi-hop network, then the routing protocols in Aqua-Net will take care of the multi-hop routing problem. A ROS developer can also choose to use the Aqua-Net APIs directly by modifying the ROS application software, as in the case of *App3* on *Vehicle4*. Note the different color of *App3* on *Vehicle4* indicates it is a modified version of *App3*.

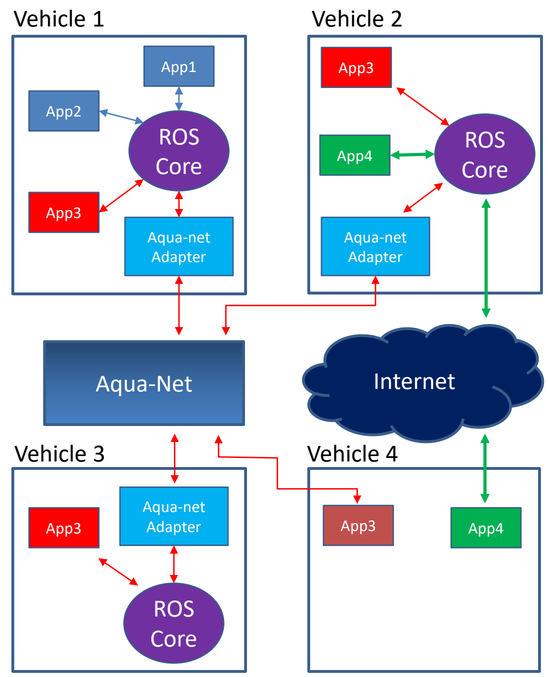

Figure 2: The integration of Aqua-Net with the ROS ecosystem

2.3.1 *The design of Aqua-Net ROS adapter.*

The Aqua-Net ROS adapter software is implemented as an independent module/package for the ROS framework. Thus, it can be plugged into any existing ROS installation on a particular robot/hardware to automatically start the Aqua-Net stack of protocols and to enable a two-way communication among the underwater vehicles. The architecture of the Aqua-Net ROS adapter is presented in Fig. 3.

The adapter offers a convenient way to send/receive data over an underwater network by providing a ROS developer a unified set of ROS topics for both outbound and inbound communications, i.e. *aquanet_outbound* and *aquanet_inbound*. If a user wants to send data from a particular ROS application to a remote application on the other side of a network, a user publishes the corresponding data to the *aquanet_outbound* topic, stating the destination node and the data to be sent. To receive incoming data, a user simply subscribes to the *aquanet_inbound* topic.

Fig. 3 presents a simple one-way communication scenario, where the ROS application on *Node1* sends data over Aqua-Net to another ROS application on *Node2*. The corresponding Aqua-Net ROS adapter instances on both sides are responsible for: retrieving original messages from the local ROS applications; converting the messages into a serialized transport unit/frame for the Aqua-Net stack; sending the frame; receiving the frame and conducting the backward operation with data deserialization and passing the original messages to the ROS application on the destination side (*Node2*).

2.3.2 *DCCL/GPB Message Support.*

The Aqua-Net ROS adapter package provides optional support for the DCCLv3 [21] and GPB (Google Protocol Buffer) message marshalling library. The library presents an extension to the GPB serialization/deserialization tool

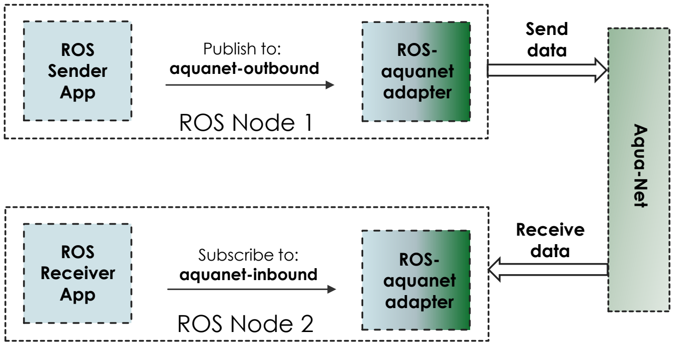

Figure 3: The architecture of Aqua-Net ROS adapter

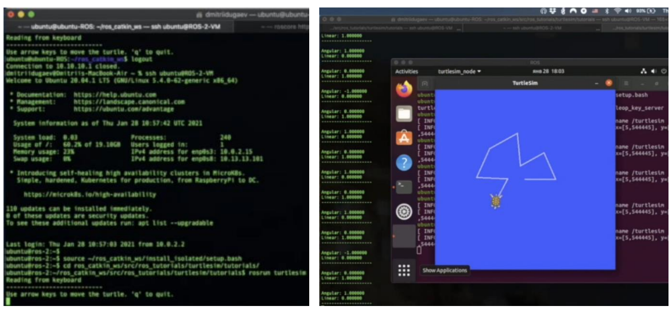

Figure 4: The ROS turtlesim demo using Aqua-Net

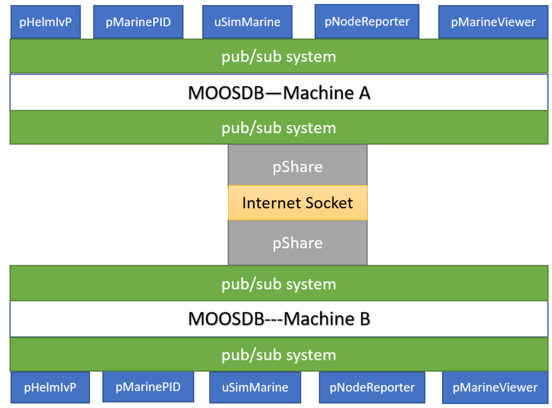

Figure 5: MOOS-IvP inter-vehicle communication

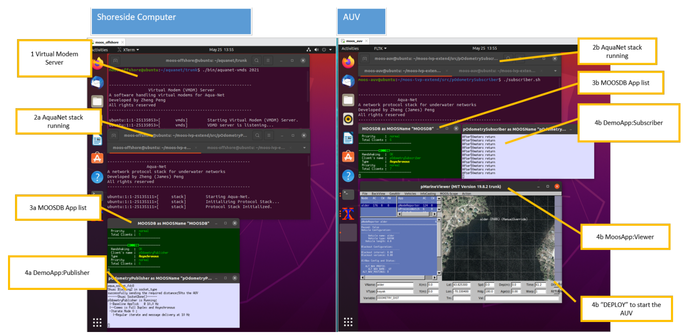

Figure 6: MOOS-IvP demo with Aqua-Net

and aims to preserve the amount of bits required to transmit a particular piece of information, described in the DCCL message fields. An information field is a combination of a data structure (i.e., an integer, double, enumeration, etc.) and values that data structure can take, such as minimum/maximum value and precision.

2.3.3 Operation Example. As a baseline example showcasing the operational behavior of the Aqua-Net ROS adapter, the `turtlesim` software demo from the standard ROS repository has been selected. The software consists of two main components: the *turtle_teleopkey* program for capturing the keystrokes from a keyboard; and the *turtlesim* program which receives the *ros-twist* messages from the *turtle_teleopkey* program and visualizes the movement of a turtle on a screen. Thus, a simple communication scenario is presented: one side sends messages over a topic to the other side which receives the messages and processes them further.

With the help of Aqua-Net, we managed to control the turtle from a remote computer. As demonstrated in Fig. 4, the left terminal and right terminal are on separate computers, with the left computer sending velocity updates to the right computer. This shows that we successfully enabled the communication between a sender (*turtle_teleopkey*) and a receiver (*turtlesim*) using Aqua-Net. With this setup, we successfully repeated the demo in a multi-hop and a single-hop multi-vehicle scenario.

2.4 Case Study: Aqua-Net with MOOS-IvP

2.4.1 Inter-vehicle Communication in MOOS-IvP. MOOS-IvP supports communication between vehicles through surface wireless links. This functionality is offered through the application *pShare*. When sending a message, the sender side application publishes the message to the local MOOSDB, which *pShare* receives locally. *pShare* then sends the message through a UDP socket to the remote vehicle. *pShare* at the remote vehicle, upon receiving the message from the UDP socket, publishes the message to its local MOOSDB. The related application will receive this message from MOOSDB. This two-vehicle communication architecture is shown in Figure 5.

To allow the vehicles to communicate underwater, the underwater acoustic communication capability needs to be added to MOOS-IvP. Considering the MOOS Core architecture, this enhancement is developed in two stages: compatibility of MOOS-IvP with Aqua-Net validated through integration with a simple MOOS application, and an Aqua-Net adapter that acts as a plugin for MOOS-IvP.

2.4.2 Operation Example. Here we introduce the result from the first stage through a case study. In this case, a shoreside computer communicates with a remote AUV to obtain location updates through Aqua-Net. All the components (the protocol stack, from physical layer to the transport layer) of Aqua-Net are running on both machines to build the complete underwater network. To validate the software development in a controlled environment, we developed a virtual modem system to emulate the physical layer.

The study case is an application of *pOdometry*. The *pOdometry Publisher* runs at the shoreside, and the *pOdometry Subscriber* runs at the AUV. The *Publisher* sends a distance request to the remote

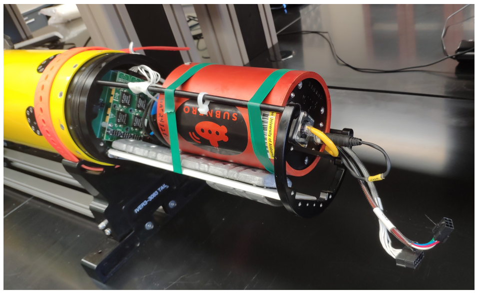

Figure 7: Subnero Embedded Modem inside Payload Section Cavity.

Subscriber through Aqua-Net. Once the AUV receives the request, it sails up to the required distance and returns. Fig. 6 is a screenshot of a real demo, showing the running components at both machines.

The offshore computer sends a 50m distance request (Fig. 6.4a) to the AUV through Aqua-Net. Fig. 6.4b shows the AUV moved 50 meters and successfully returned. This result verifies that Aqua-Net can indeed be used in MOOS-IvP for inter-vehicle communications. In addition, Fig. 6 shows the steps in running MOOS-IvP over Aqua-Net.

3 LAKE TESTBED

The lake testbed consists of two AUVs and one boat deployed in a local lake. Currently, research is devoted to the development of one AUV for the testbed, while The boat and other AUV will be added later using similar work outlined in this section.

The lake testbed uses the i3XO EcoMapper AUV by L3Harris, which is comprised of three main components: the frontseat computer, the backseat computer, and the payload section. The payload section contains various sensors that provide information to the computers. For our payload, a Subnero Embedded Modem (WNC-M25MSE3) was placed in the cavity of the payload section (Fig. 7), and connected to an acoustic transceiver placed on the top of the AUV.

The frontseat computer is the main computer of the AUV, receiving inputs from various sensors (GPS, depth, compass, etc.) and controlling movement using a program called the Underwater Vehicle Console (UVC). The UVC either operates with preprogrammed missions or in a manual mode allowing the user to control the vehicle remotely. The frontseat computer programming is mostly inaccessible to the user, and any changes performed to the frontseat computer voids the warranty of the AUV. L3Harris has provided a secondary computer, called the backseat computer, that the user can change and experiment with freely. All three sections of the AUV are connected via Ethernet ports, which allows communication between each section.

On the backseat computer, a set of C++ programs called MOOS-IvP are running. Using the MOOSDB database, the MOOS-IvP software is able to make autonomous decisions and control the AUV. Another MOOS application, called iOceanServerComms, a third-party module created for Ecomapper AUVs, facilitates communication between the frontseat and backseat computer through a serial connection [6]. Serial commands are sent to the UVC on the frontseat computer controlling the movement of the AUV. After downloading and installing these software packages from MIT, a test was performed at Lake Palmer, a lake on the campus of the University of Alabama in Tuscaloosa, AL, to show that the movement of the AUV was controlled by MOOS-IvP autonomy software running on the backseat computer. The AUV moved successfully along the path determined by the backseat mission.

Because the acoustic transmitter is on top of the AUV, it may be above the water for some parts of the mission. In order to prevent dry transmission and subsequent damage to the transceiver equipment, the acoustic modem needs to maintain an account of the location, including the depth information, of the AUV. To do this, a MOOS application called iModemComms, was created to allow communication between the MOOS programs running on the backseat computer and the Subnero modem's API. This application takes GPS and depth information from the backseat computer and sends the information to the modem. After completing this application, a test was performed that showed that the modem had an accurate account of the current GPS and depth of the AUV.

In the future, programs will be created to facilitate shore-to-vehicle communication, enabling real-time manual control. Also, as stated previously, another AUV and a boat will be added to complete the lake testbed.

4 INDOOR TESTBED AND OPEN-ACCESS HARDWARE

In this section, we describe the open-access hardware and controls architecture that enables the indoor testbed. When combined with the open-access software, this section provides the components required for a fully open-access robotic platform. The open-access hardware includes the Omnidirectional Surface Vehicle (OSV), an iteration upon a previous design [27], low-cost hardware packages, and a basic controls package.

The indoor testbed accessibility is still in development, but with plans to provide the community open-access to all hardware and software.

4.1 OSV System Design

The OSV Research Platform is a low-cost, modular vehicle that can communicate with a multitude of underwater vehicles/nodes. The base configuration includes the hull with four thrusters in an "X" configuration, a custom ACOMMS package, GPS, IMU, a low-level controller, a power management system, and an on-board computer.

4.1.1 Mechanical Structure and Manufacturability. The OSV mechanical system consists of two sections, a bottom-hat and top-hat. The bottom-hat (Fig. 8) contains the thrusters, foam thruster housings, and a marine plywood bottom plate to act as an interface between the bottom-hat and top-hat. The top-hat, not pictured, consists of a mounting plate to the bottom-hat, and holds all electronic hardware in a water-tight enclosure.

The bottom-hat foam half-spheres (Fig. 8) are machined on a 4 axis mill, and are directly bolted to the blue marine plywood plate. The 3D printed thruster housings are press fit into the foam half-spheres, and the thruster is mounted in the housings. The foam half-spheres were painted with a thin layer of epoxy to prevent

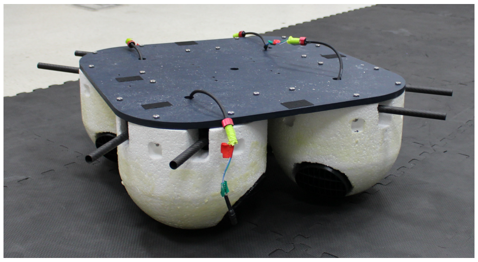

Figure 8: OSV Mechanical Structure Bottom-Hat. Pictured is the foam thruster housings mounted to a bottom plate.

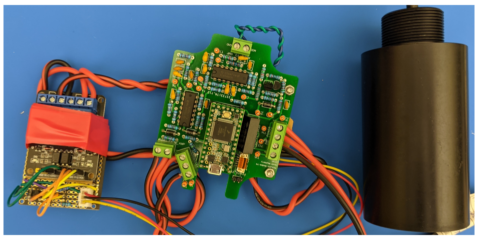

Figure 9: The acoustic modem and the transducer.

foam leakage into the environment and to protect from damage during transportation.

The blue plate, made of marine plywood, provides structural rigidity. It can be manufactured on either a CNC wood bed, a laser cutter, or a water jet. Marine paint was used to seal the plywood to prevent warping. All fasteners exposed to the aquatic environment are 316 stainless steel, meeting marine environment standards.

4.2 Low-Cost Hardware

Many underwater subsystems have no low-cost alternatives and/or are black box. Providing low-cost, high-quality, open-access alternatives was deemed necessary to decrease the financial barrier to entry into the aquatic research space. Our low-cost hardware includes: an acoustic modem, an infrared modem, a low-level controller, and a power management system. The hardware are functioning version 1 models, and are actively being improved with a target to meet the Pixhawk open-source standards.

4.2.1 Acoustic Modem. Acoustic modems are used to communicate between underwater nodes. Modem systems, including a transducer, can cost more than $8000 each on the low-end, increasing the barrier to entry for aquatic research requiring real-time underwater communication. Our contribution is a low-cost open-source modem (Fig. 9), combined with the ability to utilize a low-cost transducer, all at a price point below $250.

Our acoustic modem is capable of a 2000 baud rate utilizing frequency-shift keying (FSK), and can communicate on any carrier frequency between 20-40kHz. The receiver portion of the modem has channel state information, and can demodulate carrier frequencies up to 100kHz. The modem acts as a transparent layer for serial communication between robotic systems, and interfacing with the modem only requires an available serial port (RS232) , with future plans to add a RS-485 interface.

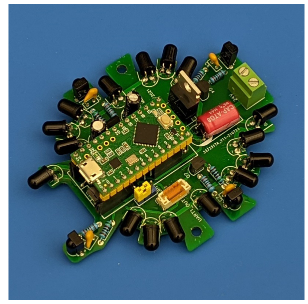

Figure 10: The IR modem with radial IR emitters

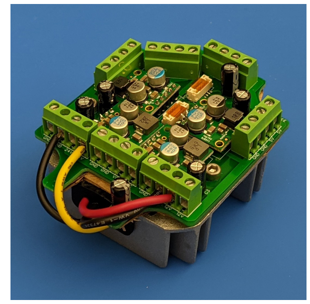

Figure 11: The power management system

The acoustic modem accepts ASCII messages as well as JSON packets through the serial port. When sending ASCII messages, the modem transmits with the previous configuration. When sending a JSON, the modem can be configured on-the-fly. JSON commands can update the baud rate, change the carrier and modulation frequencies, activate/deactivate hamming error correction, and can add dropped message pass-through, a functionality that allows partially received messages to be passed to the receiver to extract any information possible. Future functionality will add the NMEA 0183 communication standard.

Combined with the open-access Aqua-net build, a network of acoustic modems can be configured on-the-fly. JSON commands can be broadcast to all listening nodes. The JSON message is passed from the physical layer to the upper layers, where the JSON is detected and passed back down to the modem for configuration. In this way, the modem acts as a transparent communication layer, and it cannot be updated without consent of the upper layer.

4.2.2 IR Modem. Our IR modem (Fig. 10) is capable of a 5600 baud rate. The IR modem provides a secondary communication method for underwater and nearby surface vehicles, and can be utilized when acoustic communication is non-optimal, such as in shallow water or when the acoustic channel bandwidth is at capacity.

The IR modem has functionality trade-offs compared to the acoustic modem, and the decision to use either communication method is determined by the channel environment. IR light has spatial limitations and is attenuated in water. Uncommonly bright days can reduce the sensitivity of the IR sensors, requiring receivers to be closer for reliable communication. The IR modem, however, is easier to implement, has less multipath effect considerations, and can transmit faster compared to the acoustic modem.

The IR modem has less on-the-fly configurable options, as the IR hardware sensors only detect single, preset carrier frequencies. The TSOP382xx IR sensors used have preset IR detection frequencies of 38kHz for the TSOP38238 and 56kHz for the TSOP38256. They cannot be modulated faster than 1/10 of their carrier frequency, limiting the maximum baud rate to 3800 or 5600, respectively.

Future work will explore different IR sensors for on-the-fly configuration of detection frequencies, increasing transmission speeds.

4.2.3 Low-level Controller. The low-level controller interacts with onboard sensors and performs low-level control. The controller contains a BNO055 IMU, and has multiple serial, I2C, and SPI ports.

The controller can interface with up to nine motors/servos, and also provides machine learning capabilities for lightweight models such as TensorFlow Lite.

4.2.4 Power Management System. The power management system (Fig. 11) accepts a 4S LiPo battery as input and provides a range of voltages with sufficient wattage for most systems: two 5V 4A sources, a 3.3V 4A source, a 12V 3A source, and a 24V 3A source.

4.3 Controls

4.3.1 Localization System. The localization system of the OSV consists of a GPS ground station, a GPS on-board module, and an IMU. The GPS ground station, together with the GPS on-board module, enables centimeter accurate real-time kinematic positioning for the OSV. We utilize the local North-East-Down coordinates as the inertial frame. GPS is denied in indoor environments, requiring other SLAM strategies when implementing the OSV for the indoor testbed. However, for our experimentation, we will utilize GPS data.

Although the GPS module has centimeter-level accuracy, distortion caused by packet loss induces noise into the localization system and limits the accuracy of the control system. To mitigate these effects, we apply a multi-rate Kalman filter fusing the IMU and GPS data, increasing the localization accuracy[11][25]. Only planar motion is considered as the OSV is constrained to the surface of the water. The position of the center of the OSV in the inertial frame is denoted as ${}^I\boldsymbol{p} = [{}^Ip_x,\ {}^Ip_y]^\top \in \mathbb{R}^2$. The instantaneous acceleration of the center of the OSV is described by ${}^I\boldsymbol{a} = [{}^Ia_x,\ {}^Ia_y]^\top \in \mathbb{R}^2$. We can formulate the measurement process and dynamics in X or Y direction in discrete-time state equation form as:

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_d\boldsymbol{x}_k + \boldsymbol{B}_d u_k + \boldsymbol{w}_k \tag{1}$$

$$z_k = \boldsymbol{H}\boldsymbol{x}_k + v_k \tag{2}$$

$$\boldsymbol{A}_d = \begin{bmatrix} 1 & T_a \\ 0 & 1 \end{bmatrix} \quad \boldsymbol{B}_d = \begin{bmatrix} T_a^2/2 \\ T_a \end{bmatrix} \quad \boldsymbol{H} = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{3}$$

where $\boldsymbol{x}_k = [{}^Ip_k,\ {}^I\dot{p}_k]^\top$ is the position and time derivative of position at time k in X or Y direction, $u_k = {}^Ia_{m,k}$ is measured acceleration at time k in X or Y direction, $z_k = {}^Ip_{m,k}$ is measured position at time k in X or Y direction, T_a is the sampling time of the IMU, and $\boldsymbol{w}_k$ and v_k are random variables of process noise and measurement noise, respectively. With the discrete-time state equations, we can formulate the Kalman filter updating rules as:

Time update:

$$\hat{\boldsymbol{x}}_k^- = \boldsymbol{A}_d\hat{\boldsymbol{x}}_{k-1} + \boldsymbol{B}_d u_{k-1} \tag{4}$$

$$\boldsymbol{P}_k^- = \boldsymbol{A}_d\boldsymbol{P}_{k-1}\boldsymbol{A}_d^\top + \boldsymbol{Q}_d \tag{5}$$

Measurement update:

$$\boldsymbol{K}_k = \boldsymbol{P}_k^-\boldsymbol{H}^\top[\boldsymbol{H}\boldsymbol{P}_k^-\boldsymbol{H}^\top + \boldsymbol{R}_d]^{-1} \tag{6}$$

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k[z_k - \boldsymbol{H}\hat{\boldsymbol{x}}_k^-] \tag{7}$$

$$\boldsymbol{P}_k = [\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}]\boldsymbol{P}_k^- \tag{8}$$

where Q_d is the covariance matrix of the process noise. R_d is the covariance matrix of the measurement noise. K_k is the Kalman gain for the Kalman filter. We apply time and measurement updates when both IMU and GPS data arrive, but we only apply time updates when the IMU data arrives without GPS data due to the faster update rate of the IMU compared to the GPS module.

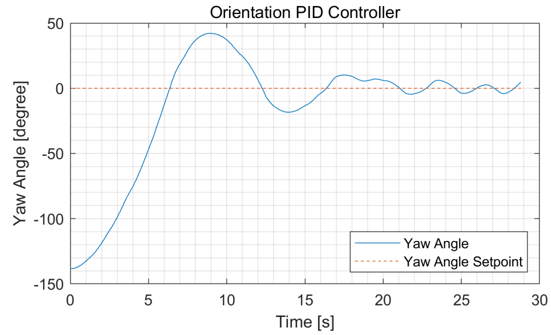

Figure 12: Performance of the orientation PID Controller

4.3.2 Control System. The OSV is designed to achieve position tracking without changing its heading direction. The omnidirectional design allows for decoupled motion control: the orientation control and the position control. Therefore, three PID controllers, one for the orientation control and two for the position control in X and Y directions, are applied to the OSV:

1) Orientation PID Controller: The OSV uses the error between zero and current heading direction to calculate the torque to keep the orientation at zero;

2) Position PID Controllers: The OSV uses the error between the desired position and current position to calculate the force to track the desired position.

4.4 Experiments

A series of experiments were designed to validate the performance of the current control system.

4.4.1 Yaw Angle Stabilizing. Fig. 12 demonstrates the performance of the orientation PID controller with a zero yaw angle setpoint. The position PID controllers were disabled, and the OSV started with an initial yaw angle of -140 degrees and rotated to the setpoint. An oscillation with an amplitude of 5 degrees was observed.

4.4.2 Position Tracking. Fig. 13 demonstrates the performance of the position PID controllers in X and Y. The setpoint was 1 meter north of the origin, $(0, 1)$ in the defined inertial frame. The OSV started from the origin and ended within 15 centimeters of the goal.

5 CONCLUSION

The initial step towards an open-access shared infrastructure has been taken. The open-access network software, the outdoor testbed, and the indoor testbed are in development, but significant progress has been made.

The current open-access software can be extended to include more sophisticated networking functionalities, including but not limited to: enhancing interoperability with a traditional TCP/IP stack, introducing a suite of detailed documentation and testing solutions for ROS/MOOS-Ivp developers, and integrating the existing ROS/MOOS-Ivp open-source projects as a separate plugin.

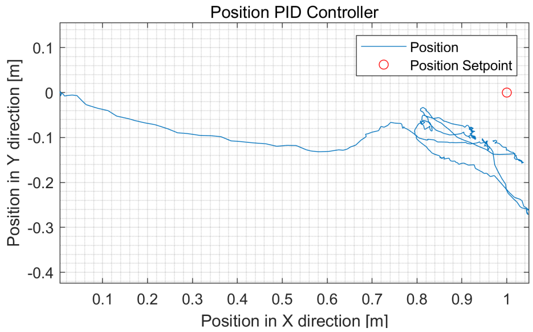

Figure 13: Performance of the Position PID Controllers

For the lake testbed, a Subnero acoustic modem was integrated on the EcoMapper AUV. Experimental results demonstrated that the backseat computer controls the movement of the AUV as well as the location and depth information were correctly updated in the embedded modem. Future developments will add modem-to-shore communication and another AUV and boat to the testbed.

The hardware developed for the indoor testbed has met the low-cost, open-source targets set out by the researchers, while the OSV has been successfully deployed with the prototype controls package. Future work for the indoor testbed includes: unifying the open-access network software with the OSV, developing a low-cost indoor localization system, finalizing the OSV design and implementation, adding AUVs, improving upon the developed hardware, and implementing community access for the indoor testbed.

ACKNOWLEDGMENTS

This research is supported by NSF grants CNS-1828678 and CNS-2016582.

REFERENCES

[1] Michael R. Benjamin. 2013. MOOS-IvP Autonomy Tools Users Manual Release 13.2. MIT Computer Science and Artificial Intelligence Lab, www.moos-ivp.org/docs/. www.moos-ivp.org/docs/

[2] Michael R Benjamin, Henrik Schmidt, Paul M Newman, and John J Leonard. 2010. Nested autonomy for unmanned marine vehicles with MOOS-IvP. *Journal of Field Robotics* 27, 6 (2010), 834–875.

[3] Andrea Caiti, Vincenzo Calabrò, Andrea Munafò, Gianluca Dini, and Angelica Lo Duca. 2013. Mobile Underwater Sensor Networks for Protection and Security: Field Experience at the UAN11 Experiment. *Journal of Field Robotics* 30, 2 (2013), 237–253. https://doi.org/10.1002/rob.21447 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.21447

[4] Mandar Chitre, Rohit Bhatnagar, and Wee-Seng Soh. 2014. UnetStack: An Agent-based Software Stack and Simulator for Underwater Networks. In *Proceedings of 2014 Oceans - St. John's*. St. John's, NL, Canada. https://ieeexplore.ieee.org/document/7003044/

[5] Thomas B Curtin, James G Bellingham, Josko Catipovic, and Doug Webb. 1993. Autonomous oceanographic sampling networks. *Oceanography* 6, 3 (1993), 86–94.

[6] Donald P. Eickstedt and Scott R. Sideleau. 2009. The backseat control architecture for autonomous robotic vehicles: A case study with the Iver2 AUV. *Oceans 2009* (2009). https://doi.org/10.23919/oceans.2009.5422212

[7] Emrecan Demirors et al. 2018. The SEANet Project: Toward a Programmable Internet of Underwater Things. In *Proceedings of The Fourth Underwater Communications and Networking Conference (UComms)*. Lerici, Italy. https://ieeexplore.ieee.org/document/8493207

[8] Zheng Peng et al. 2014. Ocean-TUNE UCONN Testbed: A Technology Incubator for Underwater Communication and Networking. In *IEEE Underwater Communications Networking (UCOMMS)*. Sestri Levante, Italy.

[9] Open Source Robotics Foundation. [n.d.]. Robot Operating System (ROS). [online] Available: http://www.ros.org/. http://www.ros.org/

[10] Roberto Francescon, Filippo Campagnaro, Emanuele Coccolo, Alberto Signori, Federico Guerra, Federico Favaro, and Michele Zorzi. 2021. An Event-Based Stack for Data Transmission Through Underwater Multimodal Networks. In *IEEE UCOMMS*.

[11] Hyung-Jik Lee and Seul Jung. 2009. Gyro sensor drift compensation by Kalman filter to control a mobile inverted pendulum robot system. In *2009 IEEE International Conference on Industrial Technology*. 1–6. https://doi.org/10.1109/ICIT.2009.4939502

[12] Francesco Maurelli, Pedro Patrón, Joel Cartwright, Jamil Sawas, Yvan Petillot, and David Lane. 2012. Integrated MCM missions using heterogeneous fleets of AUVs. In *OCEANS, 2012-Yeosu*. IEEE, 1–7.

[13] Zheng Peng, Son Le, Michael Zuba, Haining Mo, Hao Zhou, Jun-Hong Cui, Shengli Zhou, Zaihan Jiang, and Jeffrey A. Schindall. 2013. Field Test Experience of an Underwater Wireless Network in the Atlantic Ocean. In *Proceedings of MTS/IEEE OCEANS*. Bergen, Norway. https://ieeexplore.ieee.org/document/6608083

[14] Zheng Peng, Zhong Zhou, Jun-Hong Cui, and Zhijie Shi. 2009. Aqua-Net: An Underwater Sensor Network Architecture: Design, Implementation, and Initial Testing. In *Proceedings of IEEE/MTS OCEANS*. Biloxi, MS, USA. http://ubinet.engr.uconn.edu/zhengpeng/publications/PID975859.pdf

[15] Stephanie Petillo, Arjuna Balasuriya, and Henrik Schmidt. 2010. Autonomous adaptive environmental assessment and feature tracking via autonomous underwater vehicles. In *OCEANS 2010 IEEE-Sydney*. IEEE, 1–9.

[16] Chiara Petrioli, Roberto Petroccia, John Potter, and Daniele Spaccini. 2014. The SUNSET Framework for Simulation, Emulation and At-sea Testing of Underwater Wireless Sensor Networks. *Elsevier Ad Hoc Networks* (2014). http://www.sciencedirect.com/science/article/pii/S1570870514001838

[17] Lina Pu, Yu Luo, Haining Mo, Son Le, Zheng Peng, Jun-Hong Cui, and Zaihan Jiang. 2014. Comparing Underwater MAC Protocols in Real Sea Experiments. *Computer Communications* 56, 1 (2014), 47–59. http://dx.doi.org/10.1016/j.comcom.2014.09.006

[18] Lina Pu, Yu Luo, Yibo Zhu, Zheng Peng, Shruti Khare, Jun-Hong Cui, Benyuan Liu, and Lei Wang. 2012. Impact of Real Modem Characteristics on Practical Underwater MAC Design. In *Proceedings IEEE OCEANS*. Yeosu, Korea, 1–6.

[19] Signe Redfield. 2013. Cooperation between underwater vehicles. In *Marine Robot Autonomy*. Springer, 257–286.

[20] Brooks Reed, Josh Leighton, Milica Stojanovic, and Franz Hover. 2016. Multi-vehicle dynamic pursuit using underwater acoustics. In *Robotics Research*. Springer, 79–94.

[21] T. Schneider, S. Petillo, H. Schmidt, , and C. Murphy. 2015. The Dynamic Compact Control Language Version 3. In *IEEE OCEANS 2015-Genova*. https://ieeexplore.ieee.org/abstract/document/7271608

[22] Toby Schneider and Henrik Schmidt. 2010. Unified command and control for heterogeneous marine sensing networks. *Journal of Field Robotics* 27, 6 (2010), 876–889.

[23] Toby Schneider and Henrik Schmidt. 2013. Model-based adaptive behavior framework for optimal acoustic communication and sensing by marine robots. *IEEE Journal of Oceanic Engineering* 38, 3 (2013), 522–533.

[24] Toby E. Schneider and Henrik Schmidt. 2012. Goby-Acomms Version 2: Extensible Marshalling, Queuing, and Link Layer Interfacing for Acoustic Telemetry. In *Proceedings of the 9th IFAC Conference on Manoeuvring and Control of Marine Craft*. Arenzano, Italy. http://gobysoft.org/dl/mcmc2012_goby2.pdf

[25] Andrew Smyth and Meiliang Wu. 2007. Multi-rate Kalman filtering for the data fusion of displacement and acceleration response measurements in dynamic system monitoring. *Mechanical Systems and Signal Processing* 21, 2 (2007), 706–723. https://doi.org/10.1016/j.ymssp.2006.03.005

[26] Christopher C Sotzing and David M Lane. 2010. Improving the coordination efficiency of limited-communication multi–autonomus underwater vehicle operations using a multiagent architecture. *Journal of Field Robotics* 27, 4 (2010), 412–429.

[27] Qiuyang Tao, Kuo Huang, Chang Qin, Bo Guo, Robin Lam, and Fumin Zhang. 2018. Omnidirectional Surface Vehicle for Fish Cage Inspection. In *OCEANS 2018 MTS/IEEE Charleston*. 1–6. https://doi.org/10.1109/OCEANS.2018.8604674

[28] The National Science & Technology Council. [n.d.]. Science and Technology for America's Oceans: A Decadal Vision. https://www.whitehouse.gov/wp-content/uploads/2018/11/Science-and-Technology-for-Americas-Oceans-A-Decadal-Vision.pdf. Accessed: Feb 19, 2019.

[29] Li Wei, Zheng Peng, Hao Zhou, Jun-Hong Cui, Shengli Zhou, Zhijie Shi, and James O'Donnell. 2013. Long Island Sound Testbed and Experiments. In *MTS/IEEE OCEANS'13*. San Diego, USA, 1–6.

[30] Fumin Zhang, Giacomo Marani, Ryan N Smith, and Hyun Taek Choi. 2015. Future trends in marine robotics. *IEEE Robotics & Automation Magazine* 22, 1 (2015), 14–122.