Underwater Acoustic Mavlink Communication for Swarming AUVS

Yukiko Muller[1†], Shiho Oshiro[2††], Takuma Motohara[2††], Atsushi Kinjo[3†††], Taisaku Suzuki[3†††], Tomohisa Wada[2††]
ymuller692@gmail.com, wada@ie.u-ryukyu.ac.jp
[†]Graduate School of Engineering and Science University of the Ryukyus, Senbaru 1, Nishihara, Okinawa, Japan
[††]Dept. of Engineering, University of the Ryukyus, Senbaru 1, Nishihara, Okinawa, Japan
[†††]National Institute of Technology, Okinawa College, Henoko 905, Nago-shi, Okinawa Japan

Summary
The objective of this project is to conduct an underwater survey. The primary goal is to develop a device that can achieve the desired output under test conditions. For this reason, certain practical considerations must be taken into account, and the implementation is then developed to be carried out to obtain stable performance with the available hardware based on that experiment. The experiment was performed via BlueROV2 (Remotely Operated Vehicle) using RaspberryPi and softwares such as QGC (QGroundControl) and ArduPilot. This paper explains the work, the results with the collected data and how we implemented the work is presented in the end. The intention of this experiment is to connect two PCs using RaspberryPi with MAVLink communication using a Commercial-Off-The-Shelf device.

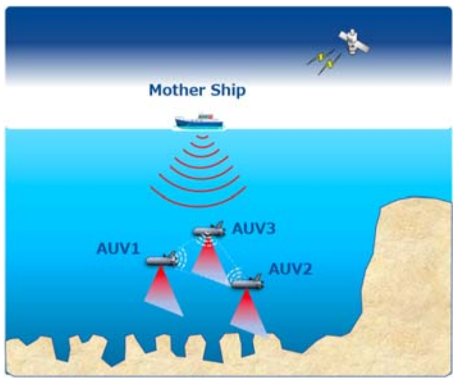
Keywords:
UWA, Swarming AUVs, MAVLink, RaspberryPi, OFDM

.1. Introduction

Developing Swarming AUVs for Deep Sea Exploring:

Humans are fascinated by the underwater realm, which has yet to be explored. Recently, there has been a surge in interest in underwater surveillance for scientific purposes, industrial exploitation, and defense against attack. Unmanned underwater vehicles have become increasingly relevant for military and research applications [1]. Underwater Acoustic (UWA) Communication has become a technology of utmost importance in recent decades. The high attenuation that occurs in an underwater environment, in addition to high bit error rates, substantial and variable propagation delays, and low bandwidth of acoustic channels, are some of the most specific challenges to underwater communications. Despite the aforementioned reasons, there is no doubt that underwater communications will encounter some of the most challenges. The peculiar features of the underwater acoustic channel pose major communication challenges. Underwater communications must rely on other physical means to transmit signals, such as sound, because electromagnetic waves do not propagate satisfactorily in underwater environments. Underwater acoustic channels typically have a wide delay spread, resulting in high frequency selectivity, due to the reverberation effect, in which the receiver observes multipath signals bounced from the surface and the bottom. UWA networks, on the contrary, have a lot of temporal and spatial variation. The UWA channel, which is frequency and time selective, presents significant challenges for high-performance and high-rate communications. Existing coherent underwater communication depends on linear or non-linear equalization techniques to eliminate inter-symbol interference (ISI) [2]. When a channel with the same delay spread is converted to the baseband discrete-time model, the symbol length decreases as the data rate increases. This presents major challenges for the channel equalizer, whose complexity would prevent rate change using the single-carrier solution currently in use. Researchers are now working on OFDM in underwater acoustic communications as a result of their performance.

This paper goes as follows: section 2 describes the Underwater OFDM Modem, section 3 introduces MAVLink in details, in section 4 the MAVLink communication experiment setup is performed, and finally, the results are presented.



Manuscript received April 5, 2021
Manuscript revised April 20, 2021
https://doi.org/10.22937/IJCSNS.2021.21.4.34

Fig 1: Swarming UAVs and mother ship

Figure 1 shows an illustrative diagram of an underwater communication system with Autonomous Underwater Vehicle (AUV) and mother ship that is controlled by UWA communication.

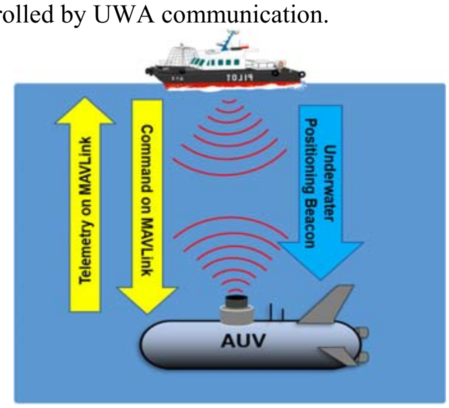


Fig 2: MAVLink Between Ship and AUV

Figure 2 summarizes the MAVLink communication between the ship and AUV. It is designed to send continuous telemetry streams (includes a drone's position, velocity, attitude, etc.). The sound from the ship is received by the receiving transducer, which is converted by the signal board. Next, the packet is processed by the RaspberryPi, and transmitted to DL (downlink). The transmitter then communicates to UL (uplink) vice-versa for transmission.

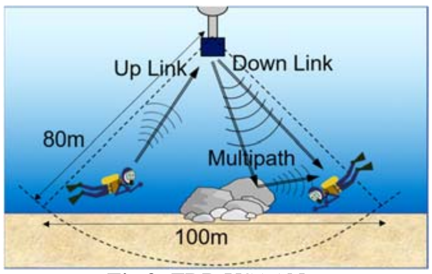


Fig 3: TDD USAAN

Figure 3 displays the overview system of TDD. UL, is the BS (base station), and DL is the MS (mobile station). An underwater communication device was implemented using UAVTool for wired LAN connection between the PCs. UL and DL packets were sent and received each second.

2. Underwater OFDM Modem

Over the last few decades, OFDM has sparked a lot of interest. It's been used in a variety of broadband networking systems [3]. OFDM's success on radio channels encourages its use on UWA channels. Researchers have been working on developing distributed and scalable underwater wireless sensor networks that will support underwater applications such as ocean observation for scientific exploration, commercial exploitation, coastline defense, and target detection in military events [4][5].

OFDM splits the available bandwidth into a large number of overlapping subbands, resulting in a long symbol length in comparison to the channel's multipath spread. As a result, ISI (Inter Symbol Interference) can be ignored in each subband, significantly reducing receiver complexity during channel equalization. On channels with frequency-selective distortion, multi-carrier modulation is a compelling alternative to single carrier broadband modulation. It works by dividing the total available bandwidth into several narrow subbands, with the channel transfer function appearing constant (ideal) within each subband. The OFDM modulation technique is the foundation of the project. OFDM achieves acceptable transmitter/receiver complexity, sufficient bandwidth, and multiple channel compensation options. OFDM is a digital multi-carrier modulation scheme based on frequency division multiplexing. Data is carried by a large number of closely spaced orthogonal subcarriers. Every subcarrier's data is divided into several parallel data streams or channels. A conventional modulation scheme is used to modulate each subcarrier.

An acoustic modem transmits data underwater. It converts digital data into special underwater sound signals, then receives the signals and converts back to the digital data. It is used for underwater telemetry, ROV and AUV command and control, diver communications underwater tracking and data recording, etc., which requires underwater wireless communications. In this experiment, it was assumed that the prepared RaspberryPi was inserted between the RaspberryPi in the ready-made BlueROV2 and the acoustic modem to function as an interface. Ethernet was used to send UDP (User Datagram Protocol) communication to the RaspberryPi, and USB is converted to Tether from the RaspberryPi to the RaspberryPi in the ready-made ROV.

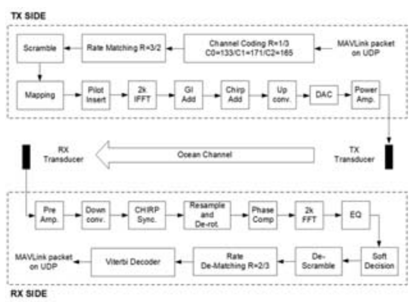


Fig 4: OFDM Modem Block Diagram

As shown in figure 4, the top side is the TX (transmitter) whereas on the bottom side is the RX (receiver) side. In between are the transducers for both the TX and RX. The block diagram begins with sending MAVLink packets on UDP from the TX side.

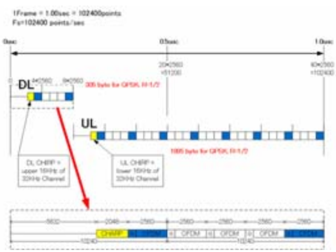


Fig 5: TDD Frame

The frame configuration of the system is as shown in Figure 5. The TDD frame consists of one DL and one UL signal. A DL signal with a synchronous subframe and three data subframes. One packet of data is sent every second. DL and UL packets are sent and received in 1 second. QSPK and R = 1/2, so the DL communicates with 305 bytes and the UL communicates with 1895 bytes.

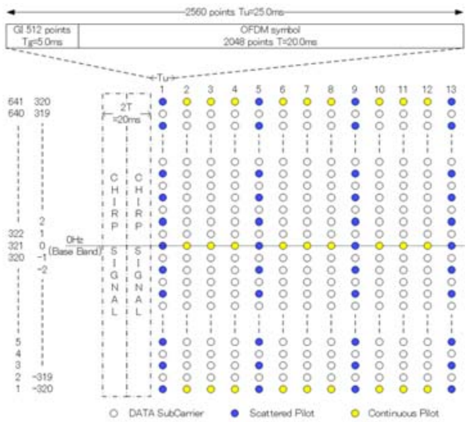


Fig 6: Time Frequency Diagram

In Figure 6, the time-frequency diagram is presented. Blue dots correspond to SP (scattered pilot) while the yellow ones correlate with CP (continuous pilot). The CTF (channel transfer function) is measured using the SP assigned to each subcarrier. To obtain the entire CTF value on the time-frequency map, the CTF value of the blue subcarrier is interpolated in two dimensions. Just an even number of sub-carriers on both edges is assigned to 13 CP. The chirp sign is a sweeping signal with a linear frequency.

Table 1: System Parameters

Parameters	Value
TX-RX Elements	1 TX and 1 RX Transducer
Sampling Frequency	102400 Hz
TX Frequency	16k – 48k Hz
Band Width	32k Hz
FFT Size	2048
OFDM symbol length T	20.0 ms (2048 points)
GI length Tg	5.0ms (512 points)
Effective Symbol length Tu=T+Tg	25.0ms
Chirp Signal Length for Frame Sync	20.0ms
Guard Time between DL/UL packet	55.0ms
SubCarrier Spacing	50.0 Hz
Number of Subcarrier	641
Number of Pilot in OFDM symbol	Zadoff –Chu, N_{zc}= 352 and 41

Table 1 shows the detail system features. This system has 1 TX transducer and 1 RX transducer. The size of the FFT is 2048 points. The OFDM symbol length is 20.0 ms and the number of subcarriers are 641. Then bandwidth of the signal is 32 kHz, and the range of transmit frequency is 16 kHz - 48 kHz.

3. MAVLINK

MAVLink has a lightweight messaging protocol that allows drones to communicate with each other [6]. MAVLink is designed in a modern hybrid publish-subscribe and point-to-point (P2P) pattern in which data streams are sent or published as topics while configuration sub-protocols, such as the mission protocol or parameter protocol, are P2P with retransmission. XML files are used to describe messages. Each XML file, also known as a “dialect”, describes the message collection supported by a specific MAVLink framework. In common.xml, the reference message set used by most ground control stations and autopilots is specified. As a key feature, there are a number of programming languages that can be used on a variety of microcontrollers and operating systems, but Windows will mostly be used in this experiment. MAVLink has two versions. V2 (version 2) and V1 (version). MAVLink 2 is a backwardcompatible upgrade to the MAVLink protocol that aims to increase MAVLink communication's versatility and protection.

MAVLink plays an important role in our project. For the experiment in the network environment, all port numbers have been unified to 50000; the MS (mobile station) and BS (base station) both have assigned IPs (internet protocol) for each PCs. The BS IP address is different from the MS IP address. The BS IP address 192.168.x.xxx while the MS IP 192.168.x.xxx. Between BS and MS, one packet is sent and received every second; DL (down link) and UL (up link) packets are sent and received per 1 second, consequently, one packet is sent and received every second. In the BS block, MatLab UAVTool sends MAVLink packets over Ethernet to the other PC. For communication between the BS and the TS (terminal station), TLV (type length value) data packets are used. Packets with a CRC (cyclic redundancy check) are sent to the communication device using the UDP protocol in DL transmission.

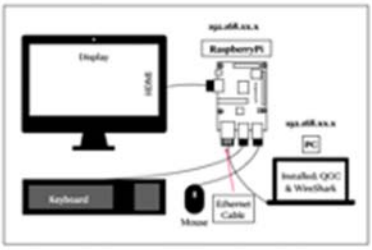


Fig 7: Experiment 1 Scheme diagram

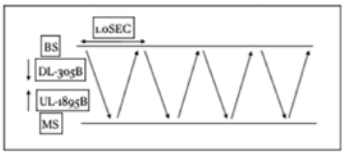


Fig 8: Experiment 2 Scheme diagram

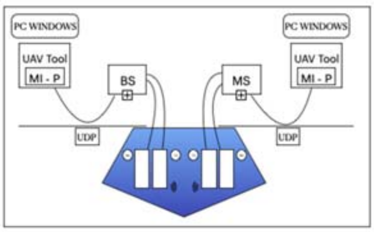


Fig 9: Experiment 3 & 4 Scheme diagram

Table 2: Hardware Look Control Output

Field Name	Type	Units	Description
time_boot_ms	uint32_t	ms	Timestamp (time since system boot)
roll	float	rad	Roll angle (-pi..+pi)
pitch	float	rad	Pitch angle (-pi..+pi)
yaw	float	rad	Yaw angle (-pi..+pi)
rollspeed	float	rad/s	Roll angular speed
Pitch speed	float	rad/s	Pitch angular speed
Yaw speed	float	rad/s	Yaw angular speed

"https://mavlink.io/en/"
Message ID = 30 (ATTITUDE) 28 bytes

Table 3: Hardware Look Control Output

Field Name	Type	Units	Values	Description
time_usec	uint64_t	us		Timestamp (UNIX Epoch time or time since boot). The receiving end can infer timestamp format by checking for the magnitude of the number.
controls				Control outputs -1 .. 1. Channel assignment depends on the simulated hardware.
mode			MAV_MODE_FLAG	System mode. Includes arming state.
flags				Flags as bitfield, 1: indicated simulation using lockstep.

"https://mavlink.io/en/"
Message ID = 93 (HIL_ACTUATOR_CONTROLS) 93 bytes

4. MAVLINK COMMUNICATION EXPERIMENT SETUP

Experiment setting

In order to verify the operation of our modem, we first tested the analog components and digital components separately and then tested the integrated system. There were four experiments that were conducted;

1. Wired LAN connection communication between ArduPilot and QGC
2. Wired LAN connection communication between PCs using MatLab UAVTool
3. One-way communication experiment by inserting an underwater wireless communication device between PC to PC

4. Two-way communication experiment by inserting an underwater wireless communication device between PC to PC

The essentials entails that made sure the project runs smoothly:

- Laptop (2 PCs): The transmitting side laptop is used to generate data with MatLab. A python script is used to transmit the data to the RaspberryPi and from it to the transmitting hardware. On the receiving side, the laptop receives the data via another PC and the demodulation is performed
- RaspberryPi: is a single-board computer
- MatLab: is a program for number analysis
- MAVLink: is a lightweight messaging protocol for drone communication
- Python: is a programming language
- WireShark: is a packet acquisition/protocol analysis software that supports a wide range of protocols
- QGC (QGroundControl): provides flight control and missions to MAVLink-enabled drones
- SITL (Simulation In The Loop): is an ArduPilot simulation tool use loop to check a series of communications
- ArduPilot: is an open-source autopilot software that can control a variety of vehicles

In experiment 1, we performed an experiment without underwater contact between the base station (QGC) and the ROV (ArduPilot). It is then used in experiment 2 to interact between QGC and ArduPilot. UAVTool was used to carry out the transmission and receiving of the MAVLink packets. In experiment 3 and 4, an underwater wireless communication system was placed between the PCs in the two experiments to explore the MAVLink protocol and discover underwater wireless communication at the same time. QGC and WireShark were installed on a Windows PC prior to the experiment. The MAVLink protocol was interpreted using WireShark. In addition to that, Ubuntu OS was installed on a RaspberryPi and ArduPilot was installed via Ubuntu. The static IP address for the network was also set for Windows PC. Lastly, between the RaspberryPi and the Windows PC, UDP communication is used.

In experiment 2, prior to preparation, we would like to be able to modify and increase the data size and speed of the MAVLink packets being sent. However, because it is a bit complicated in ArduPilot, we tried to create a PC to PC (Windows) communication environment using UAVToolbox which is a MatLab tool that can easily handle MAVLink packets.

In experiment 3, we performed a bidirectional communication experiment by having an underwater communication system installed between the PCs in which WireShark was used to test and validate the MAVLink protocol.

In experiment 4, we aimed to increase the amount of data in the packet inclusion to assure that two-way communication was yet to be achieved.

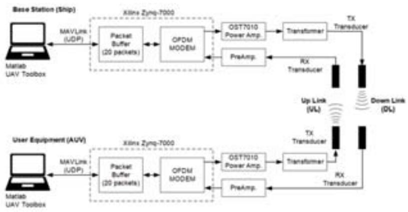


Fig 10 Experimental Setup

As shown in Figure 10, Experimental Setup, where BS (Base Station) is the Mother ship and UE (User Equipment) is the AUV, the provided diagram demonstrates. The transmitter (TX) is on the top and the outer side is DL (Down Link). The receiver (RX) is on the bottom and inner side is UL (Up Link). From the BS to the AUV is connected to MAVLink UDP Packet Buffer is 20 packets.

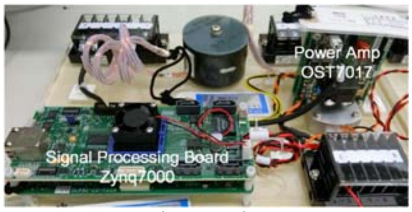


Fig 11: Modem Hardware System

Figure 11 displays the developed small hardware system of BS and UE. This device can communicate over a vertical distance from the bottom of the sea to its surface.

5. EXPERIMENT RESULT

Results on experimental data (Simulation results)

Following successful completion of simulations, we tested our system in the underwater channel. Experiment setting along with the hardware are selected in a way to have the optimal performance while data rate and useful bandwidth at minimum. In the provided Figures 12 and 13, is how the experiment was set up. Figure 14, displays any signs of movements that occurred.

In experiment 1, QGC and ArduPilot were able to communicate via wired LAN connection. In experiment 2, One-way communication was conducted with MatLab UAVTool on one side and MatLab UAVTool on the other side after communication in a local loop was recognized. As a result, there was progress in communication. UDP is the protocol. This packet is an MAVLink v2 packet, according to the analysis. In experiment 3 & 4, on the receiving side, the one actually transmitted can be specified by comparing the sequence number and payload of the acquired packets. In UL correspondence, on the other hand, all of the contents of the transmitted packets are set to fixed values, just as they were before. Even if the contents are identical, the byte string transmitted on the BS side can be obtained. As a result, it was possible to say that the UL can be transmitted and obtained. It was possible that the bidirectional communication was detected.

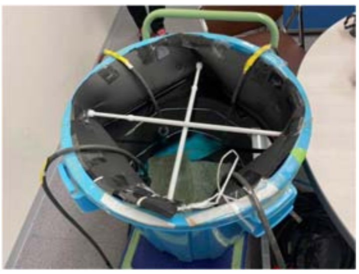


Fig 12: Commercial-Off-The-Shelf

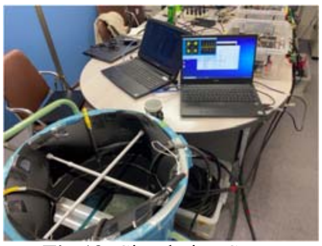


Fig 13: Simulation Setup

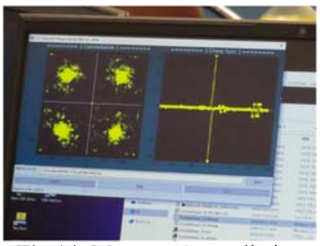


Fig 14: Measure Constellation

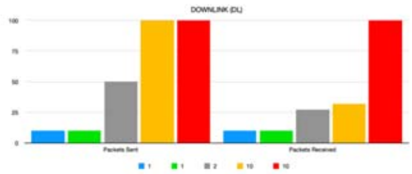


Fig 15: DL for packets sent and received

In Figure 15, the downlink for both the packets that were sent and received is shown. The left-side of the provided figure indicates the packets sent while the right-side is the receiving end. The blue color demonstrates 1 packet. Accordingly, for every 1 packet sent per second, 10 packets were sent and received. The green color demonstrates, for every 1 packet sent per second, 100 packets were sent and received. The gray color demonstrates, for every 2 packets sent per second, 10 packets were sent and received. The yellow color demonstrates, for every 10 packets sent per second, 50 packets were sent and 27 packets were received. The red color demonstrates, for every 10 packets sent per second, 100 packets were sent and 32 packets were received.

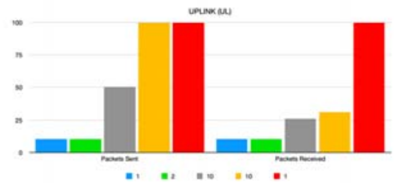


Fig 16: UL for packets sent and received

In Figure 16, the uplink is shown. Similarly to Figure 15, the left-side of the provided figure indicates the packets sent while the right-side is the receiving end. The difference is that for the yellow color 10 packets sent per second, 50 packets were sent and 26 packets were received. And for the red color 10 packets sent per second, 100 packets were sent and 31 packets were received.

Conclusion

This paper describes the design, simulation and testing of an underwater experiment using BlueROV2, but due to some experimental constraint, the method tested was based on a commercial-off-the-shelf that can be implemented in real life. The receiver was tested by simulation using UWA models in order to investigate the system. Firstly, we were able to communicate via wired LAN. Secondly, we were able to successfully use the needed applications for the project to implement the necessity. Lastly, we combined all working progress and ran the tests.

All of the 4 experiments came to fruition as a result of their execution.

Acknowledgment

This study has been carried out as part of the Strategic Information and Communications R&D Promotion Program (SCOPE) Project of the Ministry of Internal Affairs and Communications Japan.

References

[1] L. Freitag, M. Grund, J. Preisig, and M. Stojanovic, "Acoustic communications and Autonomous Underwater Vehicles," 2004.
[2] D. B. Kilfoyle and A. B. Baggeroer, "The state of the art in underwater acoustic telemetry," 2000.
[3] G. Andrews, A. Ghosh, and R. Muhamed, Fundamentalsof WiMAX understanding broadband wireless networking, 2007.
[4] I. F. Akyildiz, D. Pompili, and T. Melodia, "Challenges for efficient communication in underwater acoustic sensor networks," 2004.
[5] J. Cui, J. Kong, M. Gerla, and S. Zhou, "Challenges: Building scalable mobile underwater wireless sensor networks for aquatic applications," 2006.
[6] A. Doufexi, S. Armour, M. Butler, A. Nix, D. Bull, J. McGeehan, and P. Karlsson, "A comparison of the HIPERLAN/2 and IEEE 802.11a wireless LAN standards," 2002.
[7] "IEEE standard for local and metropolitan area networks part 16: air interface for fixed broadband wireless access systems," 2002.
[8] S. Ahmadi, "An overview of next-generation mobile WiMAX technology," IEEE Communications Magazine, vol. 47, pp. 84–98, June 2009.
[9] MAVlink, "ATTITUDE (#30)", https://mavlink.io/en/messages/common.html#ATTITUDE
[10] MAVlink, "HIL ACTUATOR CONTROLS (#93)", https://mavlink.io/en/messages/common.html#HIL_ACTUATOR_CONTROLS
[11] Shiho Oshiro, Yuta Sakuma, Ryuki Chibana, Atsushi Kinjo, Yusuke Onna, Suguru Kuniyoshi, Rie Saotome, Hajime Toma, Fumiaki Takemura, Taisaku Suzuki, Tomohisa Wada, "A Prototype Wireless Underwater Robot Control system using a 32 kHz Bandwidth Underwater Small Area Acoustic Network (USAAN)", IJCSNS International Journal of Computer Science and Network Security, VOL.20 No.6, pp.91-98, June 2020
[12] Mohammad Arif Hoq, Shiho Oshiro, Tomohisa Wada, "Two steps Doppler Compensation Algorithm from moving AUV to AUV/Mother Ship for OFDM-based UWA communication system", IJCSNS International Journal of Computer Science and Network Security, VOL.20 No.12, December 2020