Project Deliverable

Project Agreement 9011103101

UNET 3 / IRIA – D4

Research & Development in Underwater Communications and Network Technology

Report on field testing of an ad-hoc easy-to-deploy underwater network and results of experiments

Prepared by

Mandar Chitre

Version 1.0

December 10, 2015

Contents

1. Introduction

This is the final deliverable of the UNET 3 (aka IRIA) project. The key objective of the project was to put together underwater communications and network technology components into an underwater network implementation, and demonstrate that network at sea. This objective was achieved with great success, as outlined in previous deliverables (D2, D3) and Chapter 2 of this deliverable. The networking technology developed as part of this project also provided the foundation for the MISSION experiments described in deliverables D2 and D3 of project IRAZU.

In addition to developing cutting-edge underwater networking technology (in the form of UnetStack – see deliverable D3), this project also explored a few alternate modulation techniques (see Chapter 3), lower power electronics for the modem (see deliverable D2) and a new wet-end (also in deliverable D2).

The primary focus of this report is the demonstration and testing of UnetStack at sea; this is presented in Chapter 2. During the experiments, we also tested a few alternate modulation schemes; the results are presented in Chapter 3. In Chapter 4, we summarize the key findings of the project and make some recommendations on future directions for the work.

2. Field tests with UnetStack

2.1 Experimental setup

We present test results for UnetStack from two experiments in Singapore waters.

2.1.1 MISSION 2013 experiment

The MISSION 2013 experiment was conducted in the central part of Selat Pauh from 15th to 29th November 2013. The experiment consisted of two very similar deployments (about 3-4 days each) with 7 static UNET nodes. The network geometry and topology for deployment #1 is shown in Figure 1. The geometry for deployment #2 was similar to that for deployment #1, with only node 27 moved to the west by several tens of meters. In both deployments, node 21 was a surface modem mounted from an anchored barge (Figure 2), while the other 6 nodes (22, 27, 28, 29, 31 and 34) were bottom-mounted UNET PANDA nodes (Figure 3). The network was operated from the barge with a laptop (Figure 4) connected via Ethernet cable to node 21.

2.1.2 UNET 2015 experiment

The UNET 2015 experiment was conducted in the eastern part of Selat Pauh from 14th to 23rd September 2015. The experiment employed 6 static UNET nodes as shown in Figure 5. Several additional nodes (up to 3 surface modems from Subnero) were also deployed from boats during parts of the experiment. Node 25 was a surface modem deployed at the barge (Figure 6) and connected to a laptop, while nodes 22, 28, 29, 31 and 34 were bottom-mounted UNET PANDA nodes. Nodes 51, 52 and 53 were the additional on-demand deployed surface modems.

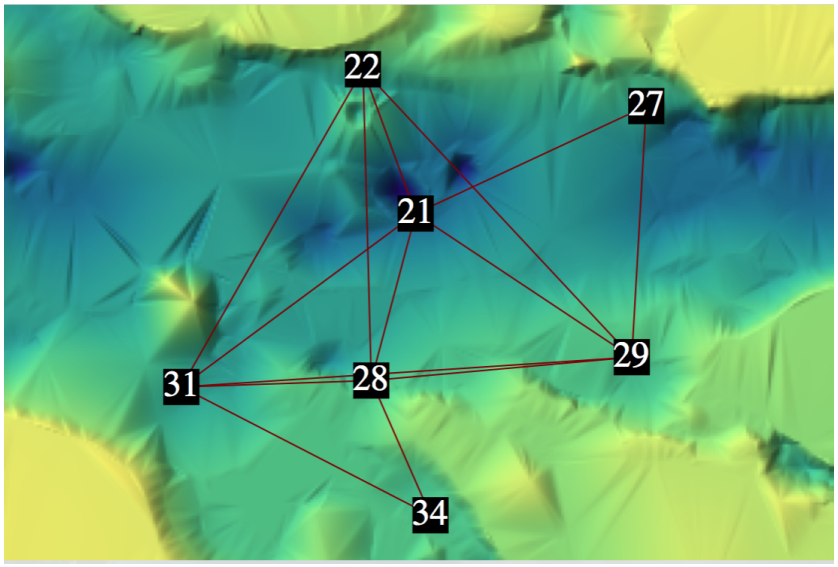


Figure 1: Network geometry and topology of the 7 static UNET nodes during MISSION 2013 deployment #1.

2.2 Test cases

The primary aim of the field experiment was to test the functionality of the network stack. Since low level functionality (e.g. sending and receiving of data frames) is exercised when higher level functionality (e.g. remote file transfer) is tested, we designed several high level test cases (outlined in detail in deliverable D3) to exercise most agents in the stack:

1. **Node information** (tests: Node information, Address resolution)
2. **Route discovery** (tests: Route discovery protocol, Physical)
3. **Route trace** (tests: Router, Reliable link, Aloha ACS, Physical)
4. **Remote parameter access** (tests: Remote control, Router, Reliable link, Aloha ACS, Physical)
5. **Range measurement** (tests: Ranging, Physical [timed transmission capability])
6. **Remote range measurement** (tests: Remote control, Router, Reliable link, Aloha ACS, Ranging, Physical)
7. **Single-hop file transfer** (tests: Remote control, Stop-and-wait transport, Reliable link, Aloha ACS, Physical)

Figure 2: A moored barge used as a gateway node and control station for the MISSION 2013 experiment.

8. **Multi-hop file transfer** (tests: Remote control, Stop-and-wait transport, Router, Reliable link, Aloha ACS, Physical)
9. **State persistence** (tests: State manager)

2.3 Test results

2.3.1 Node information

[Experiment: MISSION 2013]

We check information for node 22 (just prior to deployment) by asking for the node agent's parameters:

```
> node
```

Figure 3: ARL UNET PANDA nodes prepared for MISSION 2013 experiment at the barge.

```
node
----
address = 22
diveRate = 0.0
heading = 0.0
location = [-690.0, 777.0, -17.0]
mobility = false
nodeName = 22
origin = [1.217, 103.743]
speed = 0.0
time = 1385462517379
turnRate = 0.0
```

The node information is correctly listed. Similar tests were conducted for all nodes, and the results were correct.

We next test the address resolution for numeric node addresses (used in the experiment) as well as alphanumeric node addresses (not used in the experiment, but available for address resolution):

```
> host 21
21
> host "21"
21
> host "27"
27
> host "GREENSTAR"
116
```

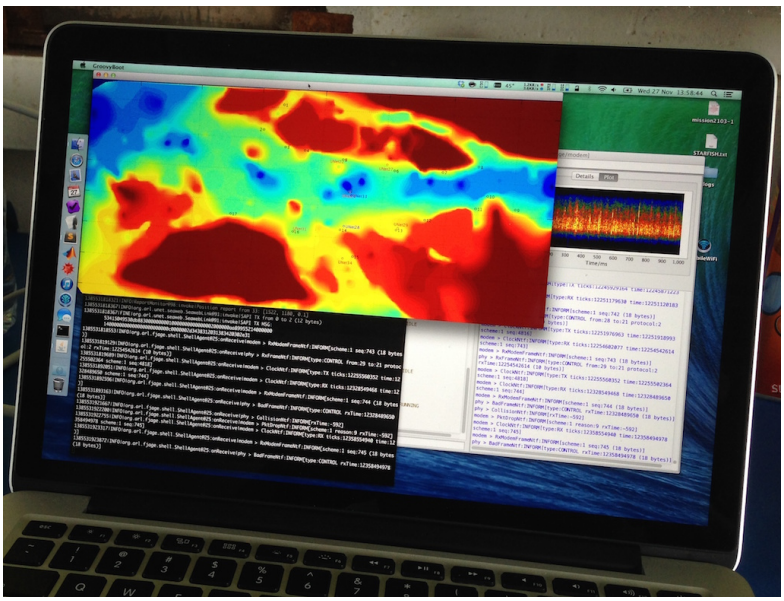


Figure 4: MISSION 2013 network being operated from a laptop connected to node 21.

```
> host "REDSTAR"
4
```

The functionality tested in this test case is local and does not involve communication between nodes. However, it tests the foundation for the rest of the test cases where inter-node communication is required.

[Experiment: UNET 2015]

More recently, we developed a new address assignment and resolution protocol for UnetStack (details available in Appendix A). Since this protocol was not developed when the test cases in D3 were developed, it was not tested in the basic tests in this section. However, we ran extensive tests using this protocol during the UNET 2015 experiment. The results from these tests are presented in Appendix A.

2.3.2 Route discovery & trace

[Experiment: MISSION 2013]

We start by checking direct connectivity to various nodes in the network:

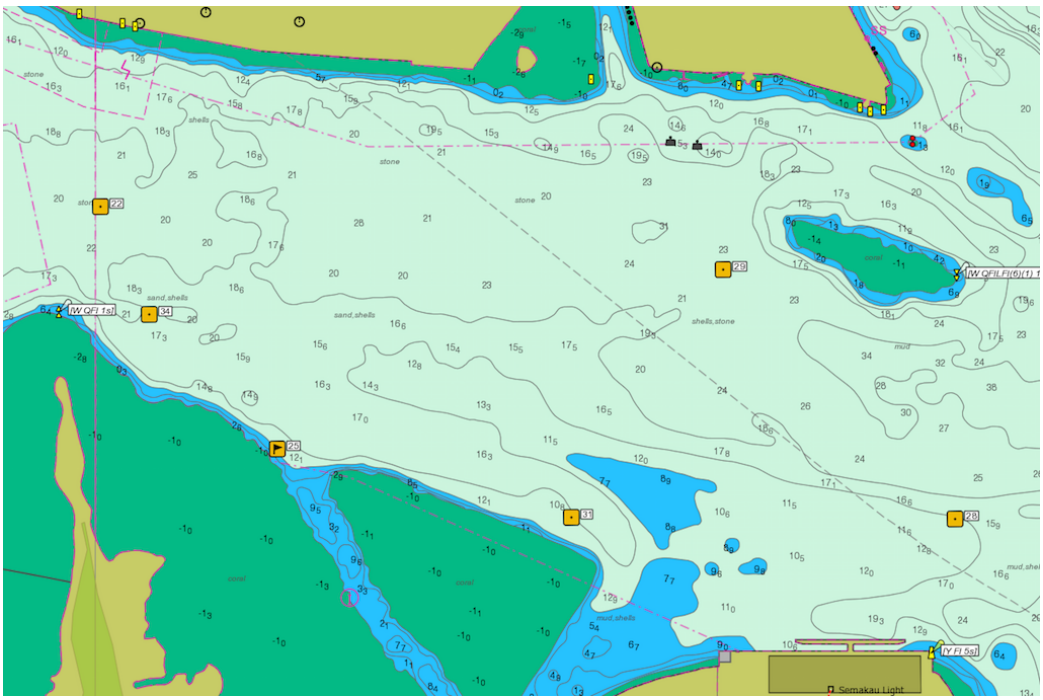


Figure 5: Surface modem (node 25) being deployed from the barge during UNET 2015 experiment.

```
> trace 27
TIMEOUT
> trace 29
[21, 29, 21]
> trace 28
[21, 28, 21]
> trace 31
[21, 31, 21]
> trace 27
TIMEOUT
> trace 22
[21, 22, 21]
> trace 34
TIMEOUT
```

The results show direct connectivity from 21 to 29, 28 and 31, but no connectivity to 27 or 34, at this point in time. Next we request discovery of routes to node 34, and check that the node is accessible via the newly discovered route:

Figure 6: Surface modem (node 25) being deployed from the barge during UNET 2015 experiment.

```
> rreq 34                              // find route to node 34
AGREE
RouteDiscoveryNtf:INFORM[to:29 nextHop:29 link:link reliability:true
    hops:1]
RouteDiscoveryNtf:INFORM[to:22 nextHop:22 link:link reliability:true
    hops:1]
RouteDiscoveryNtf:INFORM[to:28 nextHop:28 link:link reliability:true
    hops:1]
RouteDiscoveryNtf:INFORM[to:34 nextHop:28 link:link reliability:true
    hops:2 route:21-28-34]
> trace 34                     // trace the route to node 34
[21, 28, 34, 28, 21]
```

At a later point during the experiment, we display the routing table:

```
> routes
1: to 29 via link/27 [reliable, hops: 2, metric: 0.85]
2: to 22 via link/27 [reliable, hops: 3, metric: 1.6500001]
3: to 28 via link/28 [reliable, hops: 1, metric: 1.0]
4: to 27 via link/27 [reliable, hops: 1, metric: 7.0]
5: to 29 via link/29 [reliable, hops: 1, metric: 12.0]
6: to 22 via link/22 [reliable, hops: 1, metric: 8.0]
7: to 28 via link/22 [reliable, hops: 2, metric: 0.85]
8: to 31 via link/28 [reliable, hops: 0, metric: 1.0]
9: to 34 via link/28 [reliable, hops: 0, metric: 1.0]
```

Routes with the highest metric are used. For example, although two routes to node 29 exist, the direct route with metric 12.0 is used in preference to the indirect route via node 27 with metric 0.85. Although nodes 22, 28 and 29 are accessible directly, nodes 31 and 34 are only accessible via node 28.

2.3.3 Remote parameter access & command execution

Now that we have routes to various nodes, we can use higher layer functionality to access the nodes. We first test remote parameter access:

```
>  rnode(28, false).phy.dataScheme          // check data scheme
RemoteParamNtf:INFORM[dataScheme: 2]
> rnode(29,false).phy.controlScheme=5      // set control scheme
5
>  rnode(29).phy.controlScheme              // read back control scheme
RemoteParamNtf:INFORM[controlScheme: 5]
```

We also test remote command execution:

```
> rnode(31) >> 'trace 34'           // check route from node 31 to 34
OK
RemoteScriptNtf:INFORM[from: 31 text: '[31, 34, 31]']
> rnode(28).ocxo(1)                 // turn on OXCO on node 28
OK
RemoteTextNtf:INFORM[from: 28 text: 'OK']
> rnode(29,false).noise()           // check noise level at node 29
RemoteTextNtf:INFORM[from: 29 text: '75 dB']
```

In the first of the above interactions, we asked remote node 21 to execute the command "`trace 34`" and send the response back to us. We got an acknowledgement (OK) immediately. After a short while, we received the

response from node 31 that it was successfully able to trace the route to node 34, and that the command output was "`[31, 34, 31]`". The second interaction in the above listing asked remote node 28 to execute a script called "`ocxo`" with a parameter 1. This script turns on the oven-controlled clock oscillator on that node. The node executed the script and sent the script output ("`OK`") back to us. The last of the interactions asked node 29 to execute a script called "`noise`" and send us back the response, without bothering with an acknowledgement in between. The response (noise level of "`75 dB`") was sent to us after a short while.

2.3.4 Range & remote range measurement

[Experiment: MISSION 2013]

We next measure ranges to peer nodes from node 21:

```
> range 28
538.77
> range 29
941.33
> range 27
729.32
> range 22
519.1
> range 34
ERROR: no response from remote node
> range 34
1036.83
> range 31
ERROR: no response from remote node
> range 31
ERROR: no response from remote node
```

The range is returned in meters. As we see, the link to node 34 caused the range request to fail the first time round, but succeed on the second attempt. The link to node 31 was poor at this time, and returned no range. We next ask node 28 if it is able to measure a range to node 31:

```
> rnode(28,false)>>'range 31'       // ask node 28 to get range to node 31
RemoteTextNtf:INFORM[from: 28 text: '784.34']
```

We got back a response that the range between nodes 28 and 31 is 784.34 meters. We can also use the "R" script in section 3.3.6 of deliverable D3 to measure ranges between any pair of nodes:

```
> rnode(28,false).R(29)                    // range from node 28 to 29
RemoteTextNtf:INFORM[from: 28 text: '842.84']
> rnode(28,false).R(31)                    // range from node 28 to 31
RemoteTextNtf:INFORM[from: 28 text: '786.03']
> rnode(28,false).R(34)                    // range from node 28 to 34
RemoteTextNtf:INFORM[from: 28 text: 'FAIL']
> rnode(28,false).R(34)                    // retry!
RemoteTextNtf:INFORM[from: 28 text: '501.68']
> rnode(34,false).R(31)                    // range from node 34 to 31
RemoteTextNtf:INFORM[from: 34 text: '897.34']
```

The geometry of nodes shown in Figure 1 was estimated based on range measurement between every pair of nodes using an automated script. The method for inverting for geometry based on ranges is explained in detail in section 2.5 of the MISSION/IRAZU deliverable D2 .

2.3.5 Single and multi-hop file transfer

[Experiment: MISSION 2013]

To test file transfer over a single-hop network, we send a 240 byte file from node 21 to node 28:

```
> fput 'a.g',28       // send file a.g to node 28
AGREE
RemoteProgressNtf:INFORM[DatagramProgressNtf:INFORM[to:28 id:5 152/240
    bytes (63%)]]
RemoteProgressNtf:INFORM[DatagramProgressNtf:INFORM[to:28 id:5 240/240
    bytes (100%)]]
```

The file transfer of 240 bytes was completed in 38 seconds (51 bps), including acknowledgements and retransmits necessary for reliability. It is important to note that the aim of this test was the check the correctness and robustness of the file transfer protocol. The modulation scheme used for the transfer was the robust incoherent OFDM at about a raw link rate of 400 bps, and hence the file transfer times are quite long. Higher average data rates are feasible in good channel conditions, if a high rate coherent modulation scheme is

used.

Larger file transfers were also successfully completed:

```
> fput 'logs/sn2.txt', 28    // 2968 bytes in 428 seconds (56 bps)
       :
RemoteProgressNtf:INFORM[DatagramProgressNtf:INFORM[to:28 id:11 2983/2983
    bytes (100%)]]
> fput 'logs/sn3.txt', 29    // 2991 bytes in 727 seconds (33 bps)
       :
RemoteProgressNtf:INFORM[DatagramProgressNtf:INFORM[to:29 id:7 3006/3006
    bytes (100%)]]
```

To test file transfer over a multi-hop network:

```
> routes 34                  // check route to node 34
to 34 via link/28 [reliable, hops: 2, metric: 0.85]
> fget 34,'run.sh','abc.txt' // get file run.sh and save as abc.txt
RemoteFileNtf:INFORM[from: 34 filename: run.sh (746 bytes)]
FILE TRANSFERRED
```

The file transfer of 746 bytes across 2 hops took 216 seconds (55 bps per hop).

2.3.6 State persistence

The state persistence agent keeps track of changes to the stack settings and persists them by writing to a script that can be invoked to get back to the persisted state. The behavior of the agent does not require any data transfer across the network and hence it needed to be tested only on a single node, and not in a field experiment. It has been extensively used and tested, and the test results on a single node were reported in section 3.3.9 of deliverable D3. We therefore do not repeat the test results in this report.

3. Field tests with alternate modulation schemes

As part of this project, we also tested several modulation schemes other than the default OFDM-based schemes that the UNET-2 modem uses. One set of modulation schemes are based on frequency-shift keying (FSK) – either frequency-hopping binary FSK or m-FSK. The other set is based on single carrier phase shift keying (PSK) and robust channel-based equalization.

3.1 Frequency-shift keying

The WHOI micro-modem and the Teledyne Benthos modem employ FSK for communication. Although FSK offers low data rates, it is very robust. Hence the NATO standardization proposal (JANUS) also employs FSK. In order to provide compatibility with WHOI micro-modem and JANUS, and to test performance, we implemented FSK in the ARL UNET-2 modem. The implementation is very flexible; a few settings can be changed to switch between various flavors of FSK. During the UNET 2015 experiment, we tested the following flavors of FSK:

[Experiment: UNET 2015]

1. WHOI compatibility mode: binary FSK; Symbol rate: 160 baud, frequency hops: 13, bandwidth: 4,160 Hz, packet length: 20 bytes, signal duration: 1 second.
2. WHOI compatibility mode: binary FSK; Symbol rate: 80 baud, frequency hops: 7, bandwidth: 2,080 Hz, packet length: 10 bytes, signal duration: 1 second.
3. JANUS compatibility mode: binary FSK; Symbol rate: 160 baud, frequency hops: 13, bandwidth: 4,160 Hz, packet length: 20 bytes, signal

duration: 1 second.

4. Multi-carrier binary FSK (MC-BFSK); Symbol rate: 250 baud, number of carriers: 5, frequency hops: 5, bandwidth: 12,250 Hz (5 × 2,450 Hz), packet length: 140 bytes, signal duration: 1 second.
5. 4-FSK; Bandwidth: 14,000 Hz, signal duration: 1 second, variable symbol rate and tone separation (see table below).

The following table summarizes test results for the various test configurations:

Range	Modulation	Pkt-len. (bits)	Min. BER	Avg. BER	Max. BER	Link rate (bps)
1.9 km	WHOI 160 baud	160	0	0	0	160
1.9 km	WHOI 80 baud	80	0	0	0	80
1.9 km	JANUS	160	0	0	0	160
1.9 km	MC-BFSK	1,120	0	0.004	0.008	560
1.9 km	4-FSK: 300 Hz, 100 baud	1,920	0.005	0.006	0.008	960
1.9 km	4-FSK: 400 Hz, 80 baud	1,120	0.004	0.005	0.008	560
1.9 km	4-FSK: 500 Hz, 50 baud	640	0	0.001	0.002	320
1.9 km	4-FSK: 500 Hz, 200 baud	2,640	0.01	0.03	0.09	880
1.9 km	4-FSK: 500 Hz, 250 baud	3,200	0.002	0.025	0.06	1,067
754 m	4-FSK: 500 Hz, 50 baud	640	0.025	0.045	0.055	213
754 m	4-FSK: 400 Hz, 80 baud	1,120	0.03	0.05	0.16	186
2.3 km	4-FSK: 500 Hz, 50 baud	640	0.08	0.1	0.15	106
2.3 km	4-FSK: 500 Hz, 250 baud	3,200	0.09	0.11	0.16	1,067

The *link rate* in the table above is the effective error-free data rate estimated based on the forward error correction (FEC) code needed to correct all errors with high probability. Data rates of 160 bps may be achieved reliably without FEC, and up to 1 kbps may be achieved with FEC using FSK schemes. Higher data rates are achievable with OFDM/FEC or single carrier PSK/FEC, but with lower reliability.

3.2 Single carrier phase-shift keying

During the course of this project, we developed a new channel-estimate-based decision feedback equalizer (CEB-DFE) that deals with high platform mobility, exploits any sparse multi-path structure, and maintains robustness under impulsive noise. The key component of this DFE is a linear-complexity sparse channel estimator, which has the ability to detect and reject impulses based on two noise models. The receiver tolerates high transmitter/receiver mo-

bility, adapts to rapid channel fluctuations by exploiting channel sparseness, and achieves robustness under impulsive noise. By processing PSK signals from 3 mobile shallow-water acoustic links in Selat Pauh, the gain of the proposed receiver over existing equalizers was successfully demonstrated. Data rates of 9 kbps and 6 kbps were consistently achieved at 1.2 km and 3 km respectively, even in high Doppler situations. The receiver structure and the experimental results are detailed in Appendix B.

Although the results from the CEB-DFE receiver with PSK modulation are promising, the CEB-DFE could not be implemented on the current version of the UNET-2 modem due to memory limitations in the DSP. Although PSK signals were transmitted and received using the UNET-2 modems during several experiments, the CEB-DFE algorithm was only tested offline on a laptop. In the future, when the modem hardware is upgraded, we recommend that this algorithm be considered for implementation.

4. Project outcomes & recommendations

The key outcomes and achievements of the project are summarized below:

1. A new framework (UnetStack) for implementing underwater networking protocols was developed and tested. The framework is described in detail and is available online for community use at `www.unetstack.net`.
2. Several new network protocols were developed (e.g. AlohaACS, MACA-EA, juggling ARQ, route discovery, address assignment/resolution, file transfer, etc) using the UnetStack framework. These protocols were tested and demonstrated at sea.
3. The UnetStack framework was integrated with the ARL UNET-2 modem, and also tested with other modems (e.g. Evologics).
4. The UnetStack framework formed the foundation of several experiments at sea, including joint experiments with the Naval Postgraduate School (NPS, USA) during the MISSION/IRAZU project. The framework proved to be extremely valuable and flexible.
5. A study was undertaken to explore reduction in size and power consumption of the ARL UNET-2 modem (dry-end and wet-end). The study resulted in recommendations on how the modem could be redesigned in a later project. These recommendations will be incorporated by Subnero, a licensee of the UNET-2 modem technology, in the next version of the hardware.
6. Based on the study, the modem power supplies were redesigned and the power consumption was reduced to about 50% of the original.
7. A custom wet-end (electronics and transducer) was developed and tested at sea.

8. Several alternate modulation schemes were studied as part of the project. A Doppler compensation scheme for coherent OFDM was explored and found to provide modest performance improvement in practical scenarios, at the expense of significant computational power. A high-speed single-carrier CBE-DFE receiver was developed and tested at sea. This provides a better balance between performance and complexity for mobile nodes, and is able to use the sparsity of shallow water channels to improve communication performance.

9. Several flavors of FSK (including a WHOI compatibility mode, and a JANUS compatibility mode) were implemented on the UNET-2 modem and tested at sea.

Although we now have underwater communication and networking technology that allows us to easily deploy and operate underwater networks, many areas for improvement remain: Network protocol performance can be optimized based on desired applications. Adaptive modulation and error correction can be adopted based on channel conditions to optimize throughput and reliability. Newer computing technology can be used to drive down the cost, power requirements and size of the modem hardware. A new modem based on such technology will benefit from algorithms (such as CBE-DFE and stronger error correction codes) that could not be implemented on the UNET-2 modem due to hardware limitations. In all the UNET projects to date, we have not explored spatial diversity much; using spatial diversity to improve modem performance is a natural next step, in scenarios where multiple spatially separated hydrophones are feasible. Long-term measurements of modem performance are required to characterize the operating envelope and understand variability in channel conditions. In the next phase of the project, we hope to undertake such measurements. We anticipate that this will help us improve the reliability of, and operational confidence in, practical underwater networks of the future.

Appendix A

Design of an Address Assignment and Resolution Protocol for Underwater Networks

Rohit Agrawal
Department of EEE and E&I,
BITS Pilani-K. K. Birla Goa Campus.
Email: f2012267@goa.bits-pilani.ac.in

Mandar Chitre
Acoustic Research Laboratory,
Tropical Marine Science Institute,
National University of Singapore.
Email: mandar@arl.nus.edu.sg

Ahmed Mahmood
Acoustic Research Laboratory,
Tropical Marine Science Institute,
National University of Singapore.
Email: ahmed@arl.nus.edu.sg

***Abstract*—We propose a protocol that automates address assignment and resolution for nodes in an underwater network. The protocol resolves address conflicts and assigns a unique address to every node. It also informs each node about the addresses of the other nodes in the network. Using this protocol, any node can perform name resolution to find the address of a particular node. The protocol works in a distributed manner without dependence on any central node or database for address assignment and resolution. It is not only tested through simulations but also through deployment at sea.**

***Index Terms*—Underwater Networks, Address resolution, Address assignment, UnetStack.**

I. Introduction

Underwater communication has been used for many years, mainly for the purpose of oceanographic explorations. In the past few decades there have been major advances in this field and its applications now encompass diverse areas like (i) defense, (ii) oil explorations, (iii) pollution monitoring and (iv) disaster prevention [1] [2]. Some of the commonly cited challenges in underwater networks include low bandwidth, long propagation delay, half-duplex nature of the links, high packet loss, and time-variability [3] [4] [5]. To combat such challenges, it is important to develop and deploy optimized protocols that are specially designed for such environments.

Often nodes need to be deployed with minimum supervision. Such an application requires an address assignment and resolution protocol. The proposed protocol eases the burden on a user to assign addresses manually. Additionally, it also maintains a database of the neighboring nodes' addresses which may be used at any point in time to find the address of a particular node when sending data/control packets. It assigns non-conflicting addresses in the network, thereby ensuring the packets reach their intended destination. In addition to address assignment, the protocol also performs address resolution thereby enabling users to refer to each node by its name rather than a numerical address. Since centralized address resolution mechanisms are undesirable, (as a single point of failure could disable the entire network) we propose a fully distributed approach to address assignment and resolution. To avoid collisions between packets, we introduce a random back-off time and retransmit the packets to increase the probability of receiving at least one packet from each of the nodes. An appropriate mathematical model has also been proposed by which the optimum values of both number of retransmits and maximum back-off time can be ascertained for a desired probability. The protocol is not only tested through simulations on UnetStack [6] but also through deployment at sea.

The rest of this paper is organized as follows: In section II, we discuss the design and requirements of the protocol. In section III, we analyze the simulation results obtained with the UnetStack simulator [6]. In section IV, we discuss certain optimization techniques and suggest mathematical models to prevent packet collisions in the protocol. We present experimental results in section V. Finally, we discuss some concluding remarks in section VI.

II. Design of the Protocol

A. Requirements

In comparison to terrestrial wireless networks that use radio-frequency (RF) signals, underwater acoustic networks (UANs) have large propagation delays and lower data rates. Due to these characteristics, network protocols designed for terrestrial networks cannot be directly used in UANs [7]. The following requirements are taken into consideration while designing our proposed protocol:

- There should not be any conflicts in address assignment, i.e. at any point in time there should not be two nodes in the network with the same address.
- The protocol should perform name to address resolution in addition to address assignment.
- It should be distributed and not have any centralized or hierarchical dependence on any node or a set of nodes.
- It should take into account the large propagation delays in the network and the half duplex nature of communication links.
- The packets sizes used in the protocol should be small due to the lower data rates in underwater modems compared to terrestrial wireless networks.
- The protocol is designed for use in small networks (typically less than 100 nodes).

B. Address Assignment Protocol

Consider a network with n nodes where $n-1$ nodes have addresses assigned and the n^{th} node has just joined and

TABLE I
PACKET SUMMARY.

Packet Name	Abbreviation	Size (bytes)	Contents
Address Assignment Protocol			
Initial Hash Packet	IHP	2	Msg Id, Hash
Address Table Packet	ATP	$x+2$	Msg ID, Node address & x neighbour addresses
Conflict Notification Packet	CNP	2	Msg Id, Suggestion
New Hash Packet	NHP	2	Msg Id, New Hash
Final Address Packet	FAP	2	Msg Id, Final address
Address Resolution Protocol			
Initial Name Packet	INP	2	Msg Id, Node name
Resolution Failure Packet	RFP	2	Msg Id
Final Name Packet	FNP	variable	Msg Id, Node name
Resoluted Address Packet	RAP	2	Msg Id, Final address

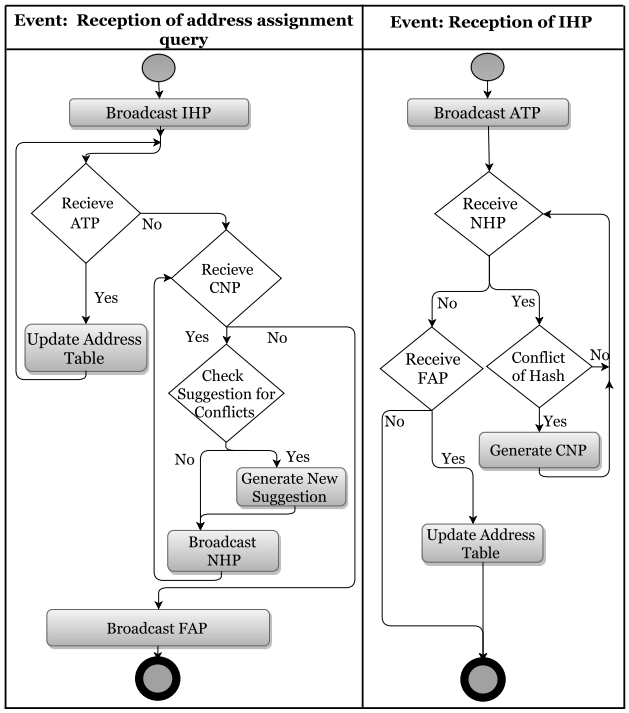


Fig. 1. Flow chart of the address assignment protocol.

needs address assignment. The address assignment begins by generating a hash based on the name of the node using the Fletcher check sum algorithm [8]. A hashing algorithm is used to convert the node name of variable length into a numeric address. This hash is broadcasted as the Initial Hash Packet (IHP) to all the nodes in the network to check for conflicts. The $n-1$ neighbor nodes check the received hash with their own addresses and within the entries of their address table for possible conflicts. If a conflict occurs, the corresponding node replies with a Conflict Notification Packet (CNP) indicating a conflict. The CNP also contains a suggestion which is the next numerically incremented hash that is not present in its own address table. For eg., if the conflicting hash is 74 and 75 is not present in the node's address table then 75 is broadcasted as a suggestion via the CNP. All $n-1$ nodes also send an Address Table Packet (ATP). This packet consists of their own address and x addresses from their own address table. Upon receiving the ATPs, each node updates its address table. This ensures that all nodes have the addresses of their immediate neighbors as well as some other nodes in the network. If the n^{th} node receives a CNP, it checks the received suggestion for conflicts within the entries of its own address table. If it faces a conflict, it generates a new suggestion that is not present in its own address table. The suggestion along with the message ID is then broadcasted as a New Hash Packet (NHP) to the neighboring nodes in the network. The nodes in the network check this new hash for conflicts and if it conflicts they reply back with a CNP again. This process continues until the conflict has been resolved. Upon resolution of the conflict, the node is assigned the nonconflicting address which is broadcasted as the Final Address Packet (FAP) to the neighboring nodes which then update their address table with the node's address.

Each packet that is sent out in this protocol has a unique message ID as its first byte. This helps in differentiating between packets and taking appropriate actions. Details of the packets are shown in Table I. To illustrate the message exchanges that happen as part of the address assignment protocol, a flow chart is presented in Fig. 1. The various cases of a nonconflicting hash and a conflicting hash are also explained with sequence diagrams presented in Fig. 2 and Fig. 3.

C. Address Resolution Protocol

When a node receives a node name for resolution via the address resolution query, it first checks its own node name and address table for the particular node name. In case the entry is missing, the protocol generates a hash corresponding to the node name and sends this name for the resolution to the node in the network having the corresponding hash as its address. There is a high chance of finding the entry in the latter's table because during its own address assignment it would have generated the same hash and received ATPs. In case it could not assign itself that hash, it may have information about the conflicting node having this hash as its address in its address table. If the entry is found, the node replies back with a Resoluted Address Packet (RAP) thereby completing address resolution. If the entry is not present, it replies back with a Resolution Failure Packet (RFP) to the node which received the address resolution query. On receiving a RFP, the node broadcasts a Final Name Packet (FNP). Any node which

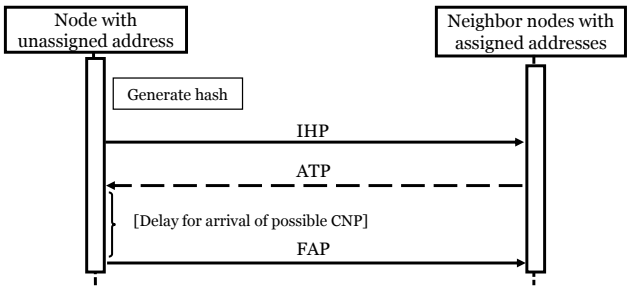


Fig. 2. Sequence diagram of the address assignment protocol for a nonconflicting hash.

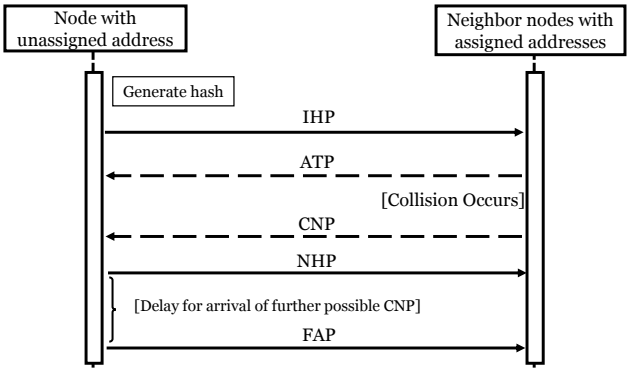


Fig. 3. Sequence diagram of the address assignment protocol for a conflicting hash.

has the particular entry in its table replies back with a RAP thereby completing address resolution.

Similar to address assignment, each packet in the address resolution protocol also has a unique message ID. Details of the packets are shown in Table I. To illustrate the message exchanges that take place as part of the address resolution protocol, a flow chart is presented in Fig. 4.

III. Simulation Results

The address assignment protocol and the address resolution protocol are simulated on the UnetStack Simulator [6]. In UnetStack, the stack consists of a collection of software agents that provide well-defined services. The address assignment protocol and the address resolution protocol are developed as agents in UnetStack. These agents are loaded on all the nodes in the network and the simulations are carried out.

A. Address Assignment Protocol

The address assignment protocol is simulated for an 8 and a 64 node network. Each ATP contains addresses of 4 neighbor nodes and hence each ATP is of 6 bytes. We discuss the simulation results in detail in the following subsections.

1) 8 Node Network: The protocol is simulated on a network with one unassigned address node and seven neighbor nodes which have addresses assigned. The network geometry is as shown in Fig. 5 with the node with an unassigned address at the origin. The node names are stated in quotes. For the

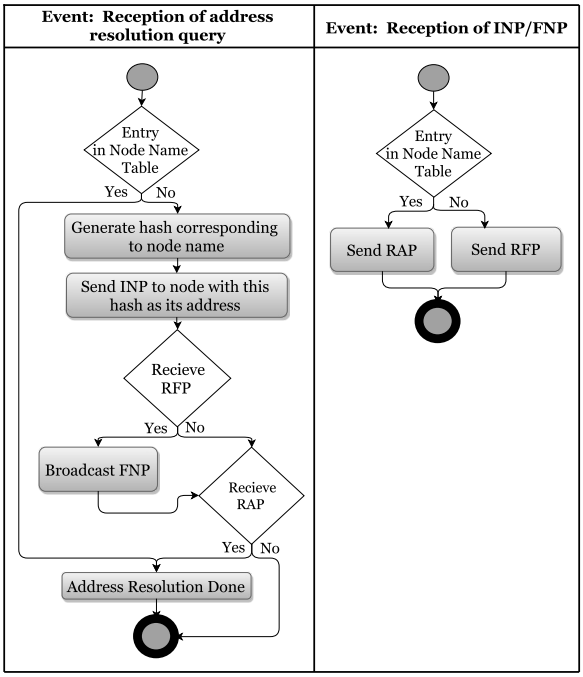


Fig. 4. Flow chart of the address resolution protocol.

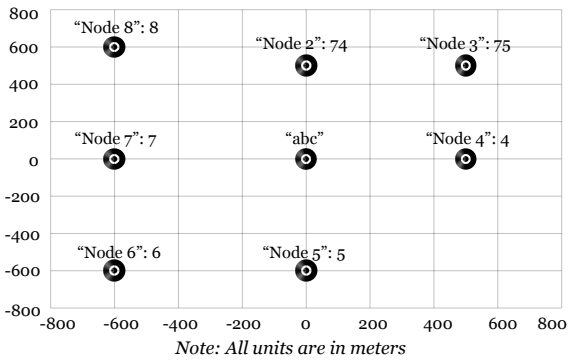


Fig. 5. Network geometry for address assignment protocol.

purpose of simulation and to depict a double conflict scenario i.e, a conflicting hash being resolved with the 2nd suggestion received, address tables of all nodes are kept blank except 'Node 3' which has an entry of 76. Node 'abc', which is at the origin does not store ATPs received from 'Node 3'. This prevents the conflict being resolved with the first suggestion received via the CNP. If ATPs from 'Node 3' are stored, the first NHP broadcasted will have nonconflicting address (77) thereby completing address assignment.

The node 'abc' begins address assignment by sending an address assignment query to the address assignment agent. The name 'abc' corresponds to a hash of 74 being generated. On receiving this hash the other 7 nodes send their addresses and 'Node 2' also sends a CNP containing the suggestion 75. Since 75 is not present in the address table of 'abc', this suggestion is now broadcasted as the NHP. Upon reception, 'Node 3'

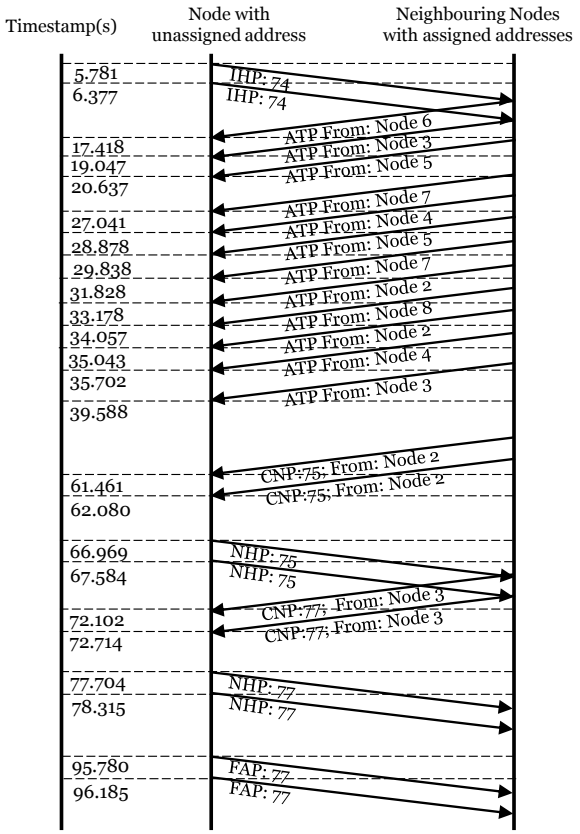


Fig. 6. Timeline of address assignment protocol.

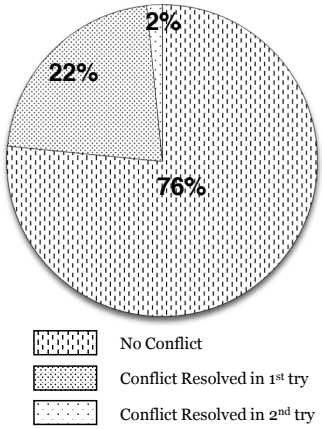


Fig. 7. Percentage of nodes facing no conflicts and conflicts.

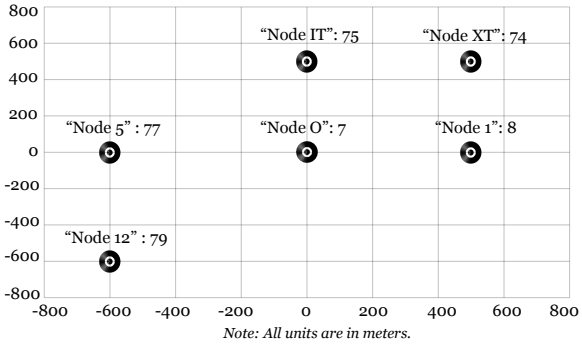


Fig. 8. Network geometry for address resolution protocol.

sends a CNP containing the suggestion 77. This suggestion is again broadcasted as a NHP by node 'abc' to check for further conflicts. Since this hash does not conflict with any address in the network, it is assigned to node 'abc'. A visualization of transmission and reception of packets in the form of a timeline is shown in Fig. 6.

2) 64 Node Network: To ensure scalability of the protocol, we also simulate a 64 node network. All 64 nodes are randomly placed in a circle of radius of 1 km. Each node is assigned a random node name. The network grows from a 1 node network gradually to a 64 node network where nodes keep joining one after the other and perform address assignment sequentially. Since initially there are lesser nodes in the network, there are lesser conflicts. However, as the network grows, (since most hashes are already taken) conflicts start occurring and they are resolved. Apart from testing scalability, the simulation also provides information of the number of tries needed for address assignment. With features like a hashing algorithm, address tables and suggestions provided on conflicts incorporated in the protocol, the objective is to reduce the number of conflicts that could occur each time a hash is broadcasted, thereby reducing the overall time required for address assignment.

Of the 64 nodes in the network, 49 nodes generate hashes which do not collide with existing nodes. 15 nodes face conflicts with the hash generated, 14 of which are resolved with the 1st suggestion received while only 1 node has to attempt a second retry for address assignment. A graphical illustration of the same in the form of a pie chart is shown in Fig. 7.

TABLE II
SIMULATION RESULTS FOR ADDRESS RESOLUTION.

Node Name Query	Number of Tries	Time taken (s)
Node 1	1	0.110
Node XT	2	3.870
Node IT	3	8.870

B. Address Resolution Protocol

The address resolution protocol is simulated on the network with node names in quotes and node addresses as shown in Fig. 8. The simulation provides information about the following three different possibilities of address resolution:

(a) The node name is present in the node's own address table.
(b) The node name is present in the address table of the node having the corresponding hash as its address.

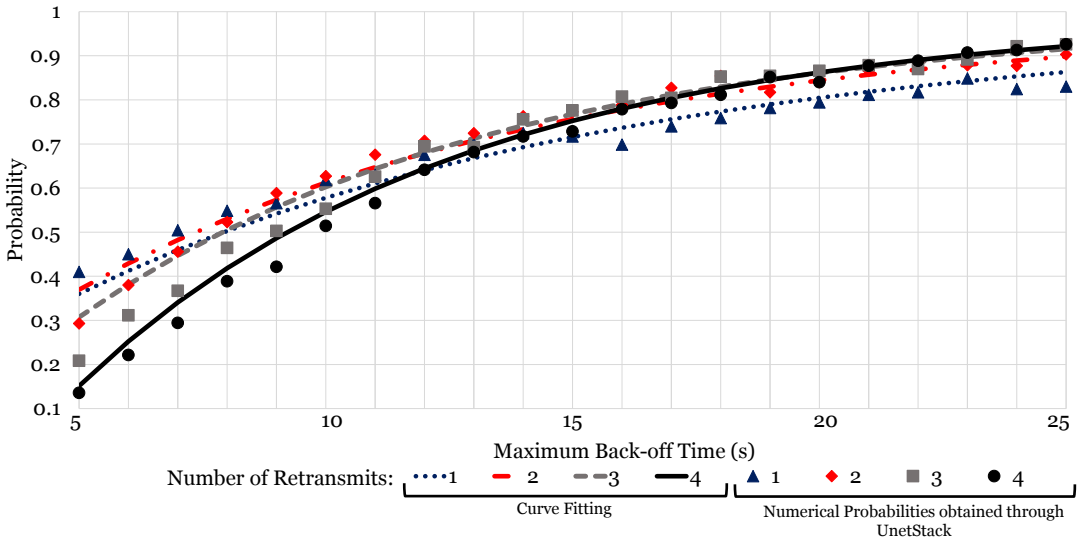


Fig. 9. Probability Curves obtained from UnetStack and through curve fitting.

(c) The third possibility is when both the previous attempts have failed and a FNP is broadcasted. Any node which has the particular entry in its table replies back with a RAP.

For the purpose of simulation and to depict all 3 scenarios of address resolution, mentioned above address tables of all nodes are kept blank except 'Node O' which has entry for 'Node 1'. Similar to address assignment, the address resolution protocol is simulated as an agent loaded on all the nodes in the UnetStack.

Three address resolution queries are sent to to 'Node O'. The purpose of each query is to test the three different possibilities of address resolution. For the first possibility of entry being in the node's own address table, a query for 'Node 1' is sent. For the second possibility of the entry being known by node with corresponding hash as its address, a query for node name 'Node XT' is sent. Since 'Node XT' corresponds to the hash 74, the INP is sent directly to this node to which it replies with a RAP. The final possibility is when both the previous attempts have failed. In this case the FNP is broadcasted to all nodes. To validate this, we send a query for node name 'Node IT' which corresponds to a hash of 74. Since the node with address 74 ('Node XT') has a blank address table it replies back with a RFP. Following this, a FNP is broadcasted, to which 'Node IT' replies with its own address as the RAP. A summarized table containing the details of the time taken for the three different queries is shown in Table II.

IV. Reducing Packet Collisions

As stated previously, our protocol updates the address table of a new node in the network by receiving ATPs from all neighboring nodes. If there are n such nodes, then packet collisions are bound to happen especially as n increases. To reduce these collisions, a random back-off time is introduced. Each packet is also retransmitted to ensure *at least one* ATP is received from each neighbor node.

It is prudent to find suitable values of the maximum back-off time and number of retransmits to minimize packet collision. We note that if both parameters are increased, packet collisions may be reduced. However, doing so would be accompanied by long delays to achieve address assignment. We simulate a network of 9 nodes on UnetStack where 8 nodes send ATPs to 1 node. In Fig. 9, we show the probability (data points) of receiving at least one ATP from each of the 8 neighbor nodes against the maximum back-off time for retransmissions within $\{1, 2, 3, 4\}$. The maximum back-off time is measured in seconds and constrained to the set $\{5, \ldots, 25\}$. For each back-off time and retransmit pair, 100 simulations are run on the UnetStack and the probability of receiving at least one ATP from each node is computed.

Though one can undertake more rigorous simulations to evaluate probabilities for intermediate back-off times and higher number of retransmits, a more useful method would predict these probabilities via an appropriate mathematical model. This is done next.

A. Curve Fitting

To fit the data points in Fig. 9 with a suitable mathematical model, we first note that the latter should output values in the interval $[0, 1]$ and take as its argument a function of the back-off time (x_1) and number of retransmits (x_2). Let us assume N data points. Then the i^{th} coordinate pair may be represented as $(x_1^{(i)}, x_2^{(i)}, y^{(i)}) \; \forall \; i \in \{1, 2, \ldots, N\}$, where $x_1^{(i)}$ is the back-off time, $x_2^{(i)}$ is the number of retransmits and $y^{(i)}$ is corresponding probability.

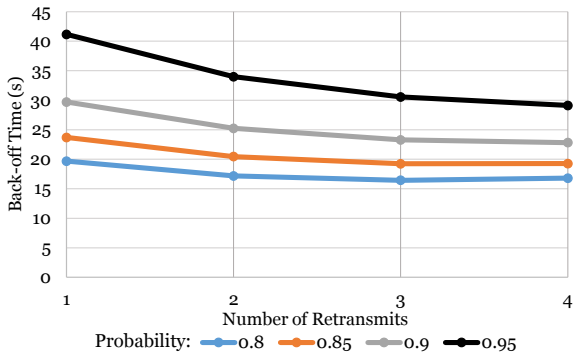


Fig. 10. Back-off Time vs Number of Retransmits for various desired probabilities.

For our fitting problem, we assume the polynomial feature vector $\mathbf{x} = [x_1, x_2, x_1^2, x_2^2, x_1x_2]^\mathsf{T}$. Similarly, the i^{th} data vector is $\mathbf{x}^{(i)} = [x_1^{(i)}, x_2^{(i)}, x_1^{2(i)}, x_2^{2(i)}, x_1^{(i)}x_2^{(i)}]^\mathsf{T}$. Defining

$$\boldsymbol{\mu}_\mathbf{x} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}^{(i)} \text{ and}$$

$$\boldsymbol{\sigma}_\mathbf{x}^2 = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}^{(i)} - \boldsymbol{\mu}_\mathbf{x})$$

as the data mean and variance vectors respectively, we have the normalized feature vector

$$\bar{\mathbf{x}} = \begin{bmatrix} 1 \\ (\mathbf{x} - \boldsymbol{\mu}_\mathbf{x})/\boldsymbol{\sigma}_\mathbf{x} \end{bmatrix}. \tag{1}$$

We employ the mathematical model:

$$h_{\boldsymbol{\theta}}(\bar{\mathbf{x}}) = 1 - e^{-\boldsymbol{\theta}^\mathsf{T}\bar{\mathbf{x}}}, \tag{2}$$

where $\boldsymbol{\theta} = [\theta_0, \theta_1, \ldots, \theta_5]^\mathsf{T}$ are the parameters that are to be estimated.

By adopting the aforementioned framework, the problem may be reduced to that of linear regression. On taking the logarithm of (2) and simplifying, we obtain

$$\boldsymbol{\theta}^\mathsf{T}\bar{\mathbf{x}} = \underbrace{\log(1 - h_{\boldsymbol{\theta}}(\mathbf{x}))}_{\bar{h}_{\boldsymbol{\theta}}(\bar{\mathbf{x}})}.$$

Let $\bar{y}^{(i)} = \log(1 - y^{(i)})$, then the mean squared error (MSE) between the *transformed* hypothesis $\bar{h}_{\boldsymbol{\theta}}(\bar{\mathbf{x}})$ and data probabilities is

$$J(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N}(\bar{y}^{(i)} - \bar{h}_{\boldsymbol{\theta}}(\bar{\mathbf{x}}^{(i)}))^2$$

$$= \frac{1}{N}\sum_{i=1}^{N}(\bar{y}^{(i)} - \boldsymbol{\theta}^\mathsf{T}\bar{\mathbf{x}}^{(i)})^2, \tag{3}$$

where $\bar{\mathbf{x}}^{(i)}$ is the i^{th} normalized data vector corresponding to (1), i.e.,

$$\bar{\mathbf{x}}^{(i)} = \begin{bmatrix} 1 \\ (\mathbf{x}^{(i)} - \boldsymbol{\mu}_\mathbf{x})/\boldsymbol{\sigma}_\mathbf{x} \end{bmatrix}. \tag{4}$$

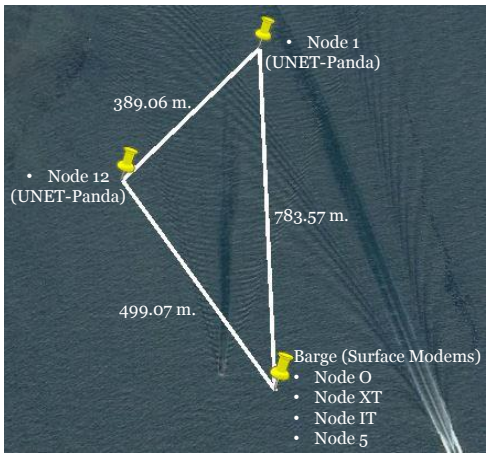


Fig. 11. Node deployment as a part of UNET-2015.

Minimizing (3) is straightforward and may be done via the normal equations [9]

$$\hat{\boldsymbol{\theta}} = (\bar{\mathbf{X}}^\mathsf{T}\bar{\mathbf{X}})^{-1}\bar{\mathbf{X}}^\mathsf{T}\bar{\boldsymbol{y}}, \tag{5}$$

where

$$\bar{\mathbf{X}}^\mathsf{T} = [\bar{\mathbf{x}}^{(1)}, \bar{\mathbf{x}}^{(2)}, \ldots, \bar{\mathbf{x}}^{(N)}] \text{ and} \tag{6}$$

$$\bar{\mathbf{y}} = [\bar{y}^{(1)}, \bar{y}^{(2)}, \ldots, \bar{y}^{(N)}]. \tag{7}$$

For the data points in Fig. 9, $N = 100$. The fitting resulted in

$$\hat{\boldsymbol{\theta}} = \begin{bmatrix} 1.16 \\ 0.54 \\ 0.12 \\ -0.07 \\ -0.31 \\ 0.34 \end{bmatrix}, \ \boldsymbol{\mu}_\mathbf{x} = \begin{bmatrix} 13.0 \\ 2.5 \\ 221 \\ 7.5 \\ 32.5 \end{bmatrix} \text{ and } \boldsymbol{\sigma}_\mathbf{x} = \begin{bmatrix} 7.24 \\ 1.12 \\ 194.11 \\ 5.70 \\ 24.64 \end{bmatrix}.$$

In Fig. 9, we have also plotted the fitted curves and have color-coded them to their corresponding data points. Clearly, the mathematical model in (2) tracks the data points well. The probability of receiving ATPs as a function of the maximum back-off time and number of retransmits is given by

$$P(x_1, x_2) = 1 - e^{-\boldsymbol{\theta}^\mathsf{T}\bar{\mathbf{x}}}. \tag{8}$$

Eq. (8) now may be used to find (x_1, x_2) pairs for a desired probability. We evaluate the two parameters for four sample probabilities in Fig. 10. The corresponding back-off time and number of retransmits can then be ascertained. For our use, we set the back-off time to 19s and the number of retransmits to 2 to achieve $P(19, 2) = 0.85$ for our protocol.

V. Experimental Results

The protocol was tested in the waters of Singapore near Selat Puah as a part of the UNET-2015 experiments. The deployed network consisted of 6 nodes: 2 bottom mounted UNET-Panda [10] nodes, and 4 surface modems deployed from a barge. The locations of the nodes are shown in Fig. 11. The exact depth of the modem varied depending on the

prevailing currents and each of them were about 2 m off the seabed. The water depth in the area is between 7 and 20 m, typically shallow close to the islands and deeper in the middle of the channel. The acoustic modem installed in the UNET-Panda is the ARL UNET-2 modem [11]. It operates in the 18-36 kHz frequency band and has a maximum range of about 2.5 km.

A. Address Assignment Protocol

Each node in the network performed address assignment sequentially starting from 'Node O' to 'Node 12' as illustrated in Table III. 'Node O' and 'Node XT' generated hashes which did not collide and hence were able to assign addresses as the corresponding hash. 'Node 1' and 'Node IT' faced conflicts which were resolved with the 1st suggestion received. For the purpose of creating a double conflict scenario for 'Node 5' and 'Node 12', 'Node O' contained only 2 entries in its address table which were 76 and 78. ATPs sent by 'Node O' were neither stored by 'Node 5' nor 'Node 12'. This forced both the nodes to attempt a second retry for address assignment. The duration of the experiment starting from the address assignment query sent by 'Node O' to FAP being sent by 'Node 12' was 960 seconds.

TABLE III
EXPERIMENTAL RESULTS FOR ADDRESS ASSIGNMENT PROTOCOL.

Node Name	Hash Generated	Address Assigned
Node O	7	7
Node 1	7	8
Node XT	74	74
Node IT	74	75
Node 5	75	77
Node 12	77	79

B. Address Resolution Protocol

The address resolution protocol was tested on this established network, with all nodes having addresses assigned. Testing of the address resolution protocol was performed in a similar manner as the simulations in UnetStack. We tested the three different possibilities of address resolution by sending three possible queries. All queries were sent to 'Node O'. For the purpose of testing, all address tables were kept blank except for 'Node O' which had an entry for 'Node 1'. Hence the query for 'Node 1' took one attempt. The second query for 'Node XT' which corresponds to a hash of 74 took 2 attempts as it was not present in Node O's address table. The third query for 'Node IT' which also corresponds to a hash of 74 took three attempts since it was neither present in Node O's address table nor in the node which had an address of 74 which was 'Node XT'. The three different queries and the respective timings are shown in Table IV.

TABLE IV
EXPERIMENTAL RESULTS FOR ADDRESS RESOLUTION PROTOCOL.

Node name query	Number of attempts	Time taken (s)
Node 1	1	0.028
Node XT	2	3.673
Node IT	3	12.739

VI. CONCLUSION

We proposed and analyzed the address assignment and resolution protocol in detail. Not only does it perform well in simulation but also when deployed at the sea. The scalability of the protocol was also been demonstrated. A mathematical model reducing packet collisions has also been suggested. Future work may incorporate a routing protocol in the agent.

REFERENCES

[1] S. Rajasegarar, J. Gubbi, O. Bondarenko, S. Kininmonth, S. Marusic, S. Bainbridge, I. Atkinson, and M. Palaniswami, "Sensor network implementation challenges in the great barrier reef marine environment," in *Proceedings of the ICT-MobileSummit 2008 Conference*, 2008.

[2] K. Liu, Z. Yang, M. Li, Z. Guo, Y. Guo, F. Hong, X. Yang, Y. He, Y. Feng, and Y. Liu, "Oceansense: Monitoring the sea with wireless sensor networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 14, no. 2, pp. 7–9, 2010.

[3] I. F. Akyildiz, D. Pompili, and T. Melodia, "State-of-the-art in protocol research for underwater acoustic sensor networks," in *Proceedings of the 1st ACM international workshop on Underwater networks*. ACM, 2006, pp. 7–16.

[4] J. Partan, J. Kurose, and B. N. Levine, "A survey of practical issues in underwater networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 11, no. 4, pp. 23–33, 2007.

[5] M. Chitre, S. Shahabudeen, and M. Stojanovic, "Underwater acoustic communications and networking: Recent advances and future challenges," *Marine technology society journal*, vol. 42, no. 1, pp. 103–116, 2008.

[6] M. Chitre, R. Bhatnagar, and W.-S. Soh, "Unetstack: an agent-based software stack and simulator for underwater networks," in *Oceans-St. John's, 2014*. IEEE, 2014, pp. 1–10.

[7] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: research challenges," *Ad hoc networks*, vol. 3, no. 3, pp. 257–279, 2005.

[8] J. Fletcher, "An arithmetic checksum for serial transmissions," *IEEE Transactions on Communications*, pp. 247–252, 1982.

[9] S. Boyd and L. Vandenberghe, *Convex Optimization*, ser. Berichte über verteilte messysteme. Cambridge University Press, 2004. [Online]. Available: https://books.google.com.sg/books?id=mYm0bLd3fcoC

[10] M. Chitre, I. Topor, R. Bhatnagar, and V. Pallayil, "Variability in link performance of an underwater acoustic network," in *OCEANS-Bergen, 2013 MTS/IEEE*. IEEE, 2013, pp. 1–7.

[11] M. Chitre, I. Topor, and T.-B. Koay, "The unet-2 modem — an extensible tool for underwater networking research," in *OCEANS, 2012-Yeosu*. IEEE, 2012, pp. 1–7.

Appendix B

Robust Equalization of Mobile Underwater Acoustic Channels

Konstantinos Pelekanakis, *Senior Member, IEEE*, and Mandar Chitre, *Senior Member, IEEE*

***Abstract*—Several underwater acoustic channels exhibit impulsive ambient noise. As a consequence, communication receivers implemented on the basis of the Gaussian noise assumption may yield poor performance even at moderate signal-to-noise ratios (SNRs). This paper presents a new channel-estimate-based decision feedback equalizer (CEB–DFE) that deals with high platform mobility, exploits any sparse multipath structure, and maintains robustness under impulsive noise. The key component of this DFE is a linear-complexity sparse channel estimator, which has the ability to detect and reject impulses based on two noise models: contaminated Gaussian and symmetric alpha stable ($\mathrm{S}\alpha\mathrm{S}$). By processing phase-shift keying (PSK) signals from three mobile shallow-water acoustic links, the gain of the proposed receiver over existing equalizers is demonstrated.**

***Index Terms*—Affine projection sign algorithm (APSA), Doppler compensation, improved-proportionate normalized least mean squares (IPNLMS), interpolation, motion synchronization, normalized least mean squares (NLMS), outliers, recursive least squares (RLS), resampling, sparse equalization.**

I. Introduction

UNDERWATER acoustic channels are severely bandlimited due to low-frequency ship noise and absorption of high-frequency energy. In addition, any transmitted sound signal undergoes both time and frequency spreading [1]. For instance, medium range (1–10 km) acoustic links are typically confined to less than 40 kHz of bandwidth and experience multipath delay spreads that can easily exceed 60 ms.

The bandwidth limitation renders coherent modulation, i.e., systems that allocate several bits of information per hertz of occupied bandwidth, more attractive than incoherent modulation. To cope with the multipath-induced intersymbol interference (ISI), numerous coherent systems rely on adaptive decision feedback equalizers (DFEs) showing excellent performance in various seas [2]–[4].

Manuscript received January 30, 2015; revised June 04, 2015; accepted August 06, 2015. This work was presented in part at the 2014 Conference on Underwater Communications and Networking (UComms), Sestri Levante, Italy, Sep. 2014.

Associate Editor: M. Stojanovic.

K. Pelekanakis was with the Acoustic Research Laboratory, National University of Singapore, 119223 Singapore. He is now with the Centre for Maritime Research and Experimentation, La Spezia 19126, Italy (e-mail: konstantinos.pelekanakis@cmre.nato.int).

M. Chitre is with the Acoustic Research Laboratory, Tropical Marine Science Institute, National University of Singapore, 119223 Singapore and also with the Department of Electrical and Computer Engineering, National University of Singapore, 117576 Singapore (e-mail: mandar@arl.nus.edu.sg).

Digital Object Identifier 10.1109/JOE.2015.2469895

DFEs can be divided into two classes. The first class adjusts its filter coefficients based on the channel impulse response, which in turn is estimated from the received signal. The second class adjusts its filter coefficients directly from the received signal. In fast varying channels such as shallow waters where multiple reflections from the moving sea surface are significant, channel-estimate-based DFEs (CEB–DFEs) optimize their coefficients faster than direct adaptation DFEs (DA–DFEs) and consequently show better performance [5]. Additional gains are possible if any available knowledge about the channel structure can be incorporated into the channel estimator. For instance, exploiting channel sparseness (i.e., a big fraction of the energy of the channel impulse response is concentrated in a small fraction of its duration) leads not only to better channel estimation accuracy but also to reduction of receiver computational complexity since only the significant channel coefficients can be retained in the equalization process [7].

Although CEB–DFEs have been thoroughly tested in underwater acoustic channels, by and large, if not all, the results are related to the assumption that the noise probability density function (pdf) is Gaussian. However, a number of underwater acoustic environments have impulsive noise sources, such as ice cracking [8] and snapping shrimp noise [9], [10]. For such environments, minimum mean squared error (MMSE) adaptive equalizers may suffer severe performance degradation and so robust equalizers become an attractive solution. Robust equalization is a mature subject in wireless radio channels and may be useful as a starting point. For example, robust equalizers for the Middleton class-A noise model [11]–[14] and the ($\mathrm{S}\alpha\mathrm{S}$) noise model [15]–[18] have been introduced.

In this paper, a novel CEB–DFE receiver that can cope with impulsive noise is presented. Robust performance is achieved via the channel estimator, which is able to identify and suppress noise impulses. Based on our recent work in [19], we employ two channel estimation algorithms: the improved-proportionate M-estimate affine projection algorithm (IPMAPA) and the improved-proportionate p-norm affine projection algorithm (IPpNAPA). Both algorithms are robust under impulsive noise, exploit sparse multipath, and require linear computational complexity with respect to the channel parameters. In addition, the receiver can cope with nonconstant platform motion without the need of periodically inserting training data for synchronization. The performance of the proposed receiver is tested in three mobile shallow-water links by processing phase-shift keying (PSK) signals. Our results firmly conclude that the performance of the CEB–DFE is improved by using IPMAPA/IPpNAPA rather than traditional channel estimators.

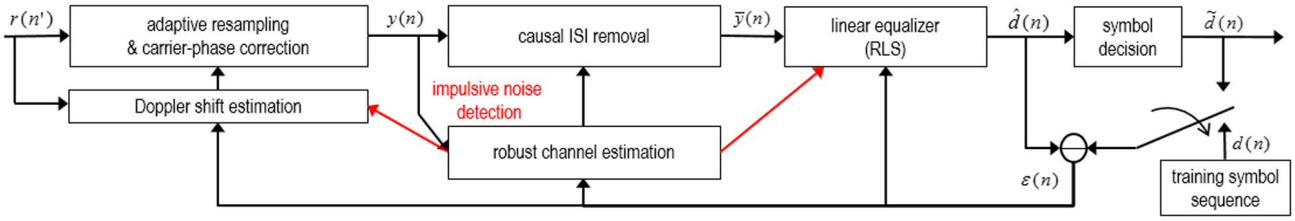


Fig. 1. Block diagram of the proposed receiver structure. The red dashed arrows indicate that both Doppler and equalization adaptation stop at time $n+1$ when an impulse is detected at time n.

The remainder of the paper is organized as follows. Section II presents the transmitter, and Section III details the design of the new CEB–DFE, including the algorithms IPMAPA and IPpNAPA. Section IV discusses the experimental layout and presents the demodulation results. Section V concludes the paper.

Notation: Superscripts $\top$, $\dagger$, and $*$ stand for transpose, Hermitian transpose, and conjugate, respectively. Column vectors (matrices) are denoted by boldface lowercase (uppercase) letters. Let $z \in \mathbb{C}$ and $p \geq 1$. The L_p-norm of z is defined as $|z|_p \triangleq (|\mathrm{Re}\{z\}|^p + |\mathrm{Im}\{z\}|^p)^{1/p}$. Let $\mathbf{z} \in \mathbb{C}^N$. The L_p-norm of $\mathbf{z}$ is defined as $\|\mathbf{z}\|_p \triangleq \left(\sum_{i=0}^{N-1} |z_i|_p^p\right)^{1/p}$. The complex gradient of a scalar function $f(\mathbf{z})$ with respect to $\mathbf{z}$ is denoted as $\nabla_{\mathbf{z}} f(\mathbf{z})$ and is defined in [20]

II. Transmitter

Our goal is to achieve high rate communications relying on coherent modulation. To this end, the transmitter uses linear and memoryless modulation methods based on M-ary constellations [21]. The information-bearing symbol stream is pulse shaped via a raised cosine (RC) filter with symbol interval T and rolloff factor γ. The baseband signal is given by

$$u(t) = \sum_n d(n) g(t - nT) \tag{1}$$

where $\{d(n)\}$ represents the information-bearing sequence of M-ary symbols, and $g(t)$ is the RC response. The signal $u(t)$ is modulated onto a carrier f_c and transmitted through the ocean. The occupied frequency range is $f_c \pm (1+\gamma)/(2T)$.

Remark 1: Note that error-correction coding could improve system performance, however, it would impede understanding of how efficiently the receiver mitigates the ISI. For this reason, channel coding is omitted in this work.

III. Receiver

From the communications perspective, time-varying multipath propagation dominates the characterization of any underwater acoustic link. Thus, the UWA channel is typically modeled as a linear time-varying system, which is described by the (lowpass equivalent) input delay-spread function $h(\tau, t)$. The variable t corresponds to the time variations of the impulse response due to physical processes (e.g., moving surface waves, tides, currents, and internal waves) while the variable τ represents the channel multipath delay for a fixed value of t. The lowpass equivalent (baseband) output $r(t)$ is related to the input $u(t)$ via the formula [6]

$$r(t) = \int_{-\infty}^{+\infty} h^*(\tau, t) u(t - \tau) d\tau + w(t). \tag{2}$$

Equation (2) may also be interpreted as a system with impulse response $h(\tau, t)$ at time t when an impulse is applied at time $t - \tau$. Here $w(t)$ models the additive impulsive ambient and thermal noise, which is independent from $u(t)$.

Transmitter–receiver motion induces a different time scale at the received signal. Since the signal bandwidth is usually comparable to the center frequency, time scaling is not well represented by just a Doppler shift. Hence, the baseband received signal (with respect to the center frequency f_c) is expressed as

$$r'(t) = r(t + \Delta t - \tau_0) e^{j2\pi f_c(\Delta t - \tau_0)} + w(t) \tag{3}$$

where Δ stands for the (time-varying) dilation/compression factor and τ_0 is the arrival time of the beginning edge of the signal. From noise perspective, platform motion is immaterial because the noise bandwidth is much larger than the signal bandwidth for all practical purposes.

The proposed CEB–DFE receiver can be seen in Fig. 1. The processing line of the received signal includes three stages: motion compensation, adaptive channel estimation, and decision feedback equalization.

A. Adaptive Resampling

The standard method for motion synchronization is to compute the time difference between two known pulse (e.g., chirp) transmissions [22]. If there is a deviation from the expected time difference, it is directly translated into a simple scaling factor. This method may be well suited for constant velocity platforms, however, it is not efficient for rapid platform acceleration (e.g., autonomous underwater vehicles). Furthermore, it suffers an overhead, i.e., a significant amount of time is not devoted for communications. Motion compensation via adaptive resampling [23], [24] offers the possibility to transmit very long communication signals with no extra overhead for synchronization. Here, the novelty is that adaptive resampling is performed in conjunction with adaptive channel estimation at the symbol rate. As a result, fast platform motion is decoupled from slow environmental fluctuations leading to improved channel estimates.

Let us denote $r(n') \triangleq r(n'T/4)$ the baseband signal sampled at four samples/symbol. Motion-induced time scaling is compensated by resampling the signal via linear interpolation. The output of the linear interpolator, denoted as $y(n)$, is downsampled to two samples per symbol and is given by

$$y(n) = \left(I(n)r(n') + (I(n)-1)\, r(n'+1)\right) e^{-j\phi(n)} \quad (4)$$

$$\phi(n) = \phi(n-1) + 2\pi\,(I(n)-1)\, f_c \frac{T}{2} \quad (5)$$

$$I(n) = I(n-1) + K_1\theta(n-1) \quad (6)$$

$$\theta(n-1) = \mathrm{Im}\left\{\hat{d}(n-1)\tilde{d}(n-1)^*\right\} \quad (7)$$

where $n' = \{1,3,\ldots\}, n = \{1,2,\ldots\}, \tilde{d}(n)$ is the decided symbol when the DFE operates in decision-directed mode [or $\tilde{d}(n) = d(n)$ when the DFE operates in training mode], $\hat{d}(n)$ is the soft output of the equalizer, $\theta(n)$ is the phase error measurement of the transmitted symbol $d(n)$, $I(n)$ is the one-tap linear interpolator $(I(0) = 1)$, K_1 is a first-order phase-locked loop (PLL) tracking parameter, and $\phi(n)$ is the carrier-phase estimate $(\phi(0) = 1)$. Typical values for K_1 range between $10^{-5} - 10^{-4}$ depending on the platform speed. We have noticed that once a value is chosen, it can remain fixed for the entire duration of the signal regardless the motion fluctuation.

B. Robust Adaptive Channel Estimation

After motion compensation, the received signal (at time nT) can be expressed in a vector form as

$$y(n) = \mathbf{h}(n)^\dagger \mathbf{u}(n) + w(n) \quad (8)$$

where

$$\mathbf{u}(n) = \begin{bmatrix} u\left(nT - (N_c+1)\frac{T}{2}\right) \\ \vdots \\ u(nT) \\ \vdots \\ u\left(nT + N_a\frac{T}{2}\right) \end{bmatrix} \quad (9)$$

and

$$\mathbf{h}(n) = \begin{bmatrix} h\left((N_c-1)\frac{T}{2}, nT\right) \\ \vdots \\ h(0, nT) \\ \vdots \\ h\left(-N_a\frac{T}{2}, nT\right) \end{bmatrix} \quad (10)$$

are the samples of the transmitted signal and the channel impulse response (including transmit and receive filters), respectively. $z(n)$ denotes the noise. Parameters N_c and N_a denote, respectively, the causal and acausal taps with respect to the channel tap $h(0, nT)$. Note that $\mathbf{u}(n)$ assumes values based on either known (past and future) symbols (training mode) or decided (past) symbols (decision-directed mode). The goal here is to reliably estimate $\mathbf{h}(n)$ in the presence of impulsive noise. Let us call this estimate as $\hat{\mathbf{h}}(n)$.

In impulsive noise environments, it is well known that L_2-norm-based channel estimation algorithms are not appropriate even in moderate signal-to-noise ratios (SNRs). Recently, the authors have introduced a framework that systematically generates sparse robust adaptive algorithms [19]. For the purposes of this work, we use two algorithms from that framework: the IPMAPA and the IPpNAPA. Both algorithms are generated by minimizing the following cost function:

$$J(n) = \sum_{i=n-L+1}^{n} f\left(\bar{e}(i)\right) + \delta\, \mathbf{r}(n)^\dagger \mathbf{P}(n-1)\mathbf{r}(n) \quad (11)$$

$$\bar{e}(i) = y(i) - \hat{\mathbf{h}}(n)^\dagger \mathbf{u}(i), \qquad i = (n-L+1)T, \ldots, nT \quad (12)$$

$$\mathbf{r}(n) = \hat{\mathbf{h}}(n) - \hat{\mathbf{h}}(n-1) \quad (13)$$

where $f()$ is a scalar *loss function* whose purpose is to downweight noise impulses [IPMAPA and IPpNAPA depend on the choice of $f()$], $\bar{e}(i)$ is the posterior error considered over a window of length L symbol intervals (typically $L < 10$ depending on channel coherence time), $\delta \geq 0$ is a regularization parameter, and $\mathbf{P}(n)$ is Hermitian positive–definite matrix whose entries depend on $\hat{\mathbf{h}}(n)$. The term $\mathbf{r}(n)^\dagger \mathbf{P}(n-1)\mathbf{r}(n)$ denotes the Riemannian distance between $\hat{\mathbf{h}}(n)$ and $\hat{\mathbf{h}}(n-1)$ and its purpose is to exploit channel sparseness [19]. By setting $\nabla_{\mathbf{r}(n)^*} J(n) = 0$ (algebra and definitions are described in Appendix A), IPMAPA/IPpNAPA can be summarized by the following equations:

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu \mathbf{A}(n)\mathbf{B}(n)\mathbf{e}(n)^* \quad (14)$$

$$\mathbf{A}(n) = \mathbf{G}(n-1)\mathbf{U}(n) \quad (15)$$

$$\mathbf{B}(n) = \left(\mathbf{U}(n)^\dagger \mathbf{A}(n) + \delta\, \mathbf{Q}(n)^{-1}\right)^{-1} \quad (16)$$

where $\mathbf{U}$ is the $(N_c + N_a) \times L$ matrix of input samples, $\mathbf{G}$ is a diagonal matrix of size $N_c + N_a$, and $\mathbf{Q}$ is a diagonal matrix of size L. Initialization starts with $\hat{\mathbf{h}}(0) = \mathbf{0}$. As already mentioned, the choice of the loss function and subsequently the matrix $\mathbf{Q}$ will generate either IPpNAPA or IPMAPA.

1) The IPpNAPA: The key assumption is that the passband noise is modeled by the SαS pdf. The SαS distribution, denoted as $S(\alpha, \delta)$, is defined by means of its characteristic function $\varphi(\omega) = e^{-(\delta|\omega|)^\alpha}$. The characteristic exponent $\alpha \in (0, 2]$ controls the heaviness of the pdf tails and the scale parameter $\delta > 0$ controls the spread of the pdf around zero. When $\alpha = 2$, the SαS pdf boils down to the Gaussian pdf $N(0, 2\delta^2)$. Parameters α and δ can be estimated from the ambient noise using fractile-based estimators [26].

In many practical situations, the real and imaginary parts of the baseband (complex) noise $z(n)$ follow the same $S(\alpha, \bar{\delta})$ but are generally dependent. The scale parameter $\bar{\delta}$ is equal to $\epsilon \cdot \delta$, ϵ being a constant that depends on the passband-to-baseband filtering [27]. In addition, it is known that SαS distributions lack

moments of order $p \geq \alpha$, but all moments of order $p < \alpha$ do exist [28]. This motivates the usage of the L_p-norm, $p \in [1, \alpha)$ as a loss function. As a result, the diagonal elements $\{q(e(n-k))\}_{k=0}^{L-1}$ of the $\mathbf{Q}$ matrix are given by [19]

$$q(e) = \begin{cases} 1, & 0 \leq |e|_2 < \xi \\ \dfrac{p}{2e^*}\Big[|\mathrm{Re}\{e\}|^{p-1}\,\mathrm{sgn}\left(\mathrm{Re}\{e\}\right) \\ \quad -j\ |\mathrm{Im}\{e\}|^{p-1}\,\mathrm{sgn}\left(\mathrm{Im}\{e\}\right)\Big], & \xi \leq |e|_2 < \Delta \\ 0, & \Delta \leq |e|_2. \end{cases} \tag{17}$$

The threshold parameters ξ and Δ are responsible for detecting and downweighting impulses in the intervals $[\xi, \Delta]$ and rejecting any impulse stronger than Δ. These thresholds are proportional to $\bar{\delta}$ but their exact value depends on the received SNR. In contrast, the choice $p = a - 0.15$ is fairly robust [19].

2) The IPMAPA: The key assumption is that the noise

$$z(n) = y(n) - \mathbf{h}(n)^{\dagger}\mathbf{u}(n) \tag{18}$$

is modeled as complex Gaussian noise, but "contaminated" with impulses (or outliers). Using an M-estimator function $f()$ is the typical approach to achieve robustness against impulses, however, redescending M-estimators is a more effective solution because they can differentiate between gross and moderate impulses. A typical redescending M-estimator is Hampel's loss function [25]. Based on this function, the diagonal elements $\{q(e(n-k))\}_{k=0}^{L-1}$ of matrix $\mathbf{Q}$ are given by [19]

$$q(e) = \begin{cases} 1, & 0 \leq |e|_2 < \xi \\ \dfrac{\xi}{|e|_2}, & \xi \leq |e|_2 < \Delta \\ \xi\dfrac{|e|_2 - \mathrm{T}}{\Delta - \mathrm{T}}\dfrac{1}{|e|_2}, & \Delta < |e|_2 < \mathrm{T} \\ 0, & \mathrm{T} < |e|_2. \end{cases} \tag{19}$$

The threshold parameters ξ, Δ, and T are responsible for detecting and downweighting impulses in the intervals $[\xi, \Delta]$ and $[\Delta, T]$ and rejecting any impulse with amplitude greater than T. These thresholds are computed based on the assumption that the impulse-free signal $|z(n)|_2$ is Rayleigh distributed with scale parameter (or mode) σ. We choose the thresholds as: $\xi = 2.45\sigma$, $\Delta = 2.72\sigma$, and $\mathrm{T} = 3.03\sigma$ [19]. A robust recursive estimate of σ (denoted as $\hat{\sigma}$) can be obtained through the prior error signal $e(n) = y(n) - \hat{\mathbf{h}}(n-1)^{\dagger}\mathbf{u}(n)$ and the median operator [25] as follows:

$$\hat{\sigma}(n) = \sqrt{\frac{(\sigma_r^2(n) + \sigma_i^2(n))}{2}} \tag{20}$$

$$\hat{\sigma}_r^2(n) = \lambda_\sigma \hat{\sigma}_r^2(n-1) + c(1-\lambda_\sigma)\mathrm{med}\left(\bar{\mathbf{e}}_r(n)\right) \tag{21}$$

$$\hat{\sigma}_i^2(n) = \lambda_\sigma \hat{\sigma}_i^2(n-1) + c(1-\lambda_\sigma)\mathrm{med}\left(\bar{\mathbf{e}}_i(n)\right) \tag{22}$$

$$\bar{\mathbf{e}}_r(n) = \left[e_r^2(n) \ldots e_r^2(n-N_w+1)\right]^{\top} \tag{23}$$

$$\bar{\mathbf{e}}_i(n) = \left[e_i^2(n) \ldots e_i^2(n-N_w+1)\right]^{\top} \tag{24}$$

$$c = 1.483\left(1 + \frac{5}{N_w - 1}\right) \tag{25}$$

where N_w is the observation window of the error signal, $\hat{\sigma}_r^2(n)$ and $\hat{\sigma}_i^2(n)$ denote, respectively, the estimated variance of the real $(e_r(n))$ and imaginary part $(e_i(n))$ of $e(n)$, c is a finite sample correction factor, and λ_σ is a forgetting factor. Note that $O(N_w \log_2(N_w))$ operations are required for the computation of $\hat{\sigma}^2(n)$. Furthermore, $\bar{\mathbf{e}}(n)$ always contains signal-dependent noise from channel estimation errors and thus, parameter N_w should be chosen to be smaller than the channel coherence time.

C. Decision Feedback Equalization

Let L_c and L_a denote, respectively, the causal and acausal taps of the linear equalizer and let

$$\mathbf{y}(n) = \begin{bmatrix} y\left(nT - (L_c+1)\frac{T}{2}\right) \\ \vdots \\ y(nT) \\ \vdots \\ y\left(nT + L_a\frac{T}{2}\right) \end{bmatrix} \tag{26}$$

be the vector of received samples that corresponds to the nth symbol interval. The received signal can be written as

$$\mathbf{y}(n) = \mathbf{H}(n)^{\dagger}\bar{\mathbf{u}}(n) + \mathbf{z}(n) \tag{27}$$

where

$$\bar{\mathbf{u}}(n) = \begin{bmatrix} u\left(nT - (N_c + L_c - 2)\frac{T}{2}\right) \\ \vdots \\ u(nT) \\ \vdots \\ u\left(nT + (N_a + L_a)\frac{T}{2}\right) \end{bmatrix} \tag{28}$$

and

$$\mathbf{H}(n) = \begin{bmatrix} \mathbf{h}(n-N_c+1) & \cdots & 0 & \cdots & 0 \\ 0 & \ddots & \vdots & & \vdots \\ \vdots & & \mathbf{h}(n) & & \vdots \\ \vdots & & \vdots & \ddots & 0 \\ 0 & \cdots & 0 & \cdots & \mathbf{h}\left(n+N_a+L_a\right) \end{bmatrix} \tag{29}$$

is the $(L_c + L_a + N_c + N_a - 1) \times (L_c + L_a)$ convolution matrix. Let us now partition $\mathbf{H}(n)$ into causal and acausal parts as follows:

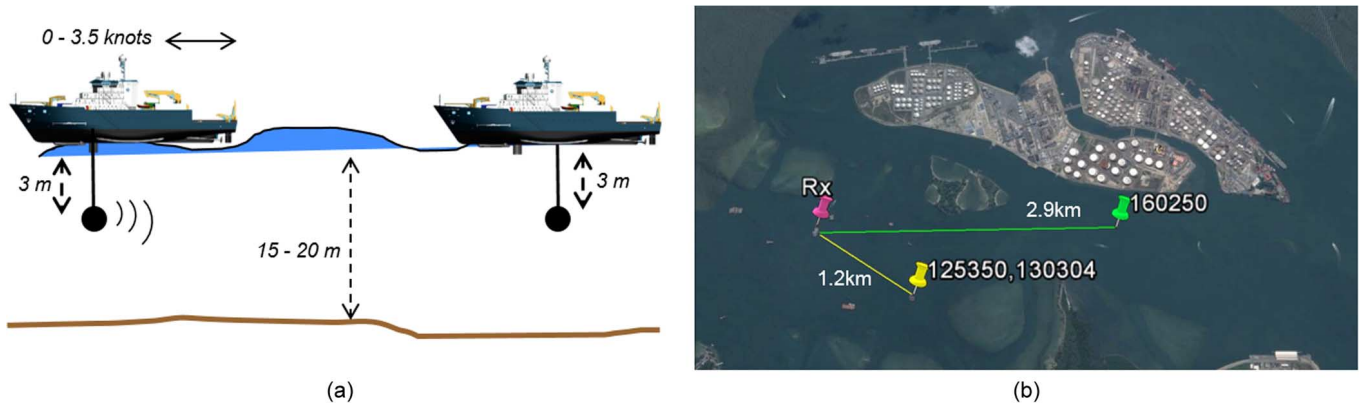
$$\mathbf{H}(n) = \begin{bmatrix} \mathbf{H_c}(n) \\ \mathbf{H_{ac}}(n) \end{bmatrix} \tag{30}$$

where $\mathbf{H_c}(n)$ includes the rows of $\mathbf{H}(n)$ that correspond to the causal input signal

$$\bar{\mathbf{u}}_{\mathbf{c}}(n) = \begin{bmatrix} u\left(nT - (N_c + L_c - 2)\frac{T}{2}\right) \\ \vdots \\ u(nT - T) \end{bmatrix} \tag{31}$$

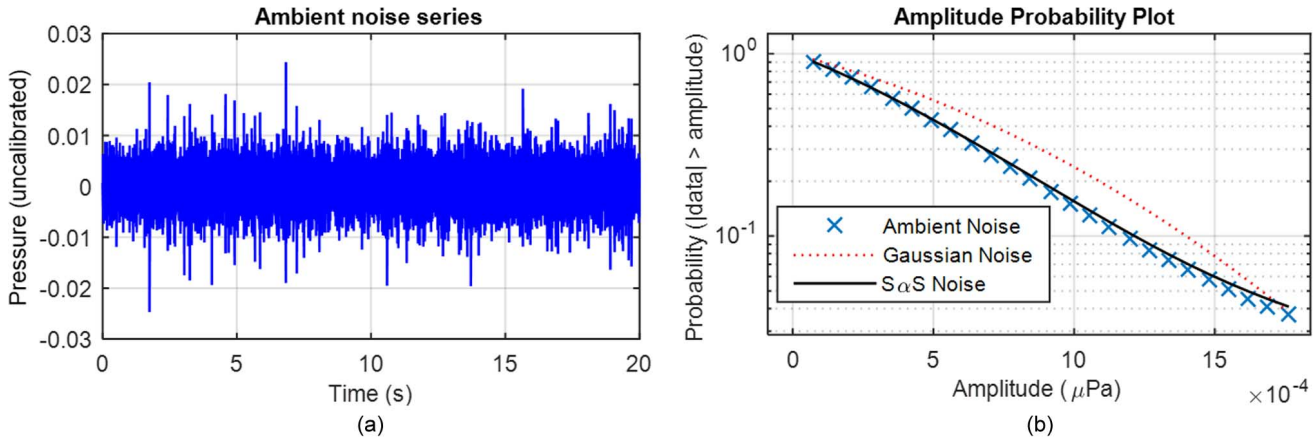
and $\mathbf{H_{ac}}(n)$ includes the rows of $\mathbf{H}(n)$ that correspond to the acausal input signal

$$\bar{\mathbf{u}}_{\mathbf{ac}}(n) = \begin{bmatrix} u\left(nT - \frac{T}{2}\right) \\ \vdots \\ u\left(nT + (N_a + L_a)\frac{T}{2}\right) \end{bmatrix}. \tag{32}$$



(a) (b)

Fig. 2. (a) Experimental setup. (b) Map of sea of Selat Pauh with the transmitter/receiver GPS coordinates.



(a) (b)

Fig. 3. (a) Recorded ambient noise series. The hydrophone output is voltage (proportional to acoustic pressure). (b) Amplitude probability plots showing that the SαS distribution is a better fit than the Gaussian distribution. SαS fit parameters: $\alpha = 1.67$ and $\delta = 0.00044$.

The ISI-free signal at the input of the linear equalizer can be constructed as follows:

$$\bar{\mathbf{y}}(n) = \mathbf{y}(n) - \hat{\mathbf{H}}_c(n)^\dagger \bar{\mathbf{u}}_\mathbf{c}(n). \tag{33}$$

The soft estimate $\hat{d}(n)$ of the transmitted symbol is obtained by

$$\hat{d}(n) = \mathbf{p}(n)^\dagger \bar{\mathbf{y}}(n) \tag{34}$$

where $\mathbf{p}(n)$ denotes the equalizer filter. Adaptation of $\mathbf{p}(n)$ at every data symbol arrival is achieved via the exponentially weighted recursive least squares (RLS) algorithm [29]

$$\mathbf{p}(n) = \mathbf{p}(n-1) + \text{RLS}\left(\varepsilon(n), \bar{\mathbf{y}}(n), \delta', \lambda\right) \tag{35}$$

$$\varepsilon(n) = \hat{d}(n) - \tilde{d}(n) \tag{36}$$

$$\lambda = 1 - \frac{1}{4(L_c + L_a)} \tag{37}$$

where δ' is a regularization parameter and λ is the exponential weighting factor. The RLS computational complexity is $O((L_c + L_a)^2)$, however, as indicated in [7] and verified in our experimental results, $\mathbf{p}(n)$ can be designed shorter than the total delay spread of the channel and still deliver good performance.

Remark 2: This DFE implementation is robustified via the channel estimator. If a strong impulse occurs at time n, it means that the prior error signal $e(n)$ with high probability is larger than Δ (for IPpNAPA) or T (for IPMAPA). In such and only event, the receiver modifies (6) to $I(n+1) = I(n)$ and (35) to $\mathbf{p}(n+1) = \mathbf{p}(n)$. This robustification strategy is sufficient to deal with a small fraction of impulses in the data as the experimental results confirm.

IV. Sea Experiment and Results

The proposed receiver was tested in experimental data obtained in a shallow-water environment. To assess the receiver performance based on the IPMAPA and the IPpNAPA, we use linear complexity algorithms from the adaptive filter literature. These are: improved-proportionate normalized least mean squares (IPNLMS) [30], normalized least mean squares (NLMS) [29], and affine projection sign algorithm (APSA) [31]. We stress that IPNLMS is a sparse-aware algorithm (all sparse adaptive filters used in this paper employ the same $\mathbf{G}$ matrix; see Appendix A), but not robust under impulsive noise. The NLMS algorithm is neither sparse aware nor robust in the presence of impulses. The APSA uses the L_1-norm of the error signal and therefore is robust under impulsive noise, but cannot exploit channel sparseness.

A. Experimental Layout

The experiment was conducted in the sea of Selat Pauh, Singapore, on October 23, 2013. The projector (transmitter) was deployed off a vessel and submerged about 3 m below the sea surface. The received signals were recorded at a different vessel 3 m below the sea surface. The sea depth was about 15–20 m and the sound-speed profile was isovelocity (1540 m/s). The sea surface was calm but often the links encountered ship wakes.

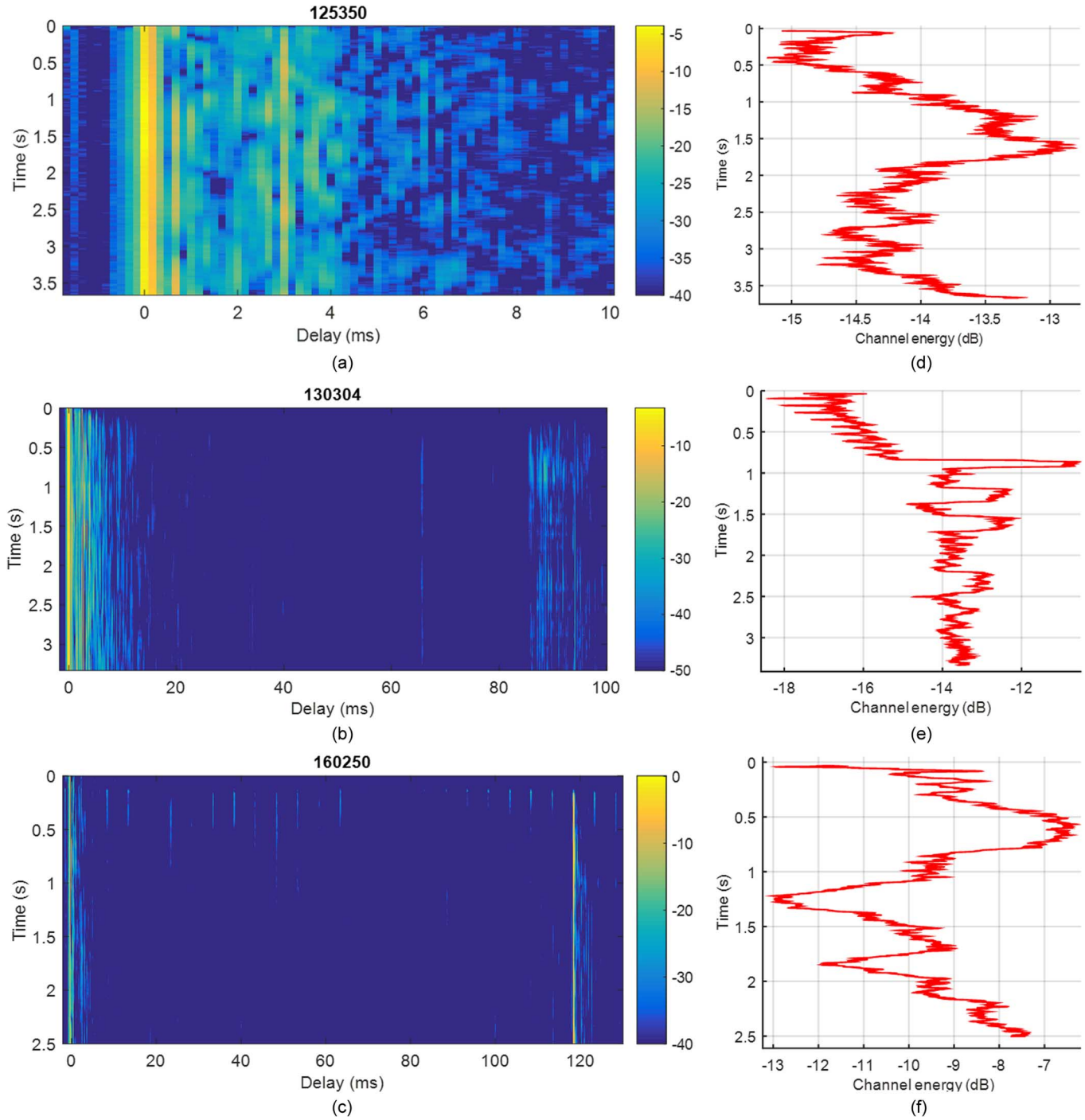


Fig. 4. (a)–(c) Channel impulse responses for links 125350, 130304, and 160250. For all impulse responses, the x-axis shows multipath delay, the y-axis shows absolute time, and the z-axis shows the channel amplitude in decibel scale. (d)–(f) Channel energy versus time for links 125350, 130304, and 160250.

Fig. 2(a) illustrates the experimental layout. Two signal formats are considered, i.e., 4- and 8-PSK signals modulated by a pseudonoise sequence. The baud rate was 3000 symbols/s, the carrier frequency was 17 kHz, and the rolloff factor of the RC pulse was 0.7 resulting in a bandwidth of 14 450–19 550 Hz. Several transmissions at different ranges and vessel velocities were repeated. In this paper, we report results based on two ranges: 1.2 km (files 125350 and 130304) and 2.9 km (file 160250). The links 125350 and 130304 have almost the same range (note that the vessels were slightly drifting due to sea currents), yet link 125350 was utilized 10 min before link 130304. Fig. 2(b) shows the transmit/receive positions. The sampling rate at the receiver was 250 kHz. Before we present the demodulation results, it is instructive to gain insight into the ambient noise and channel characteristics.

B. *Ambient Noise*

Here, we present an ambient noise data set (14 450–19 550 Hz) recorded during the experiment. Fig. 3(a) clearly shows that the noise series includes instantaneous (impulse-like) sharp sounds. The source of these impulses is due to snapping shrimp [9]. Studies have shown that the SαS distribution efficiently

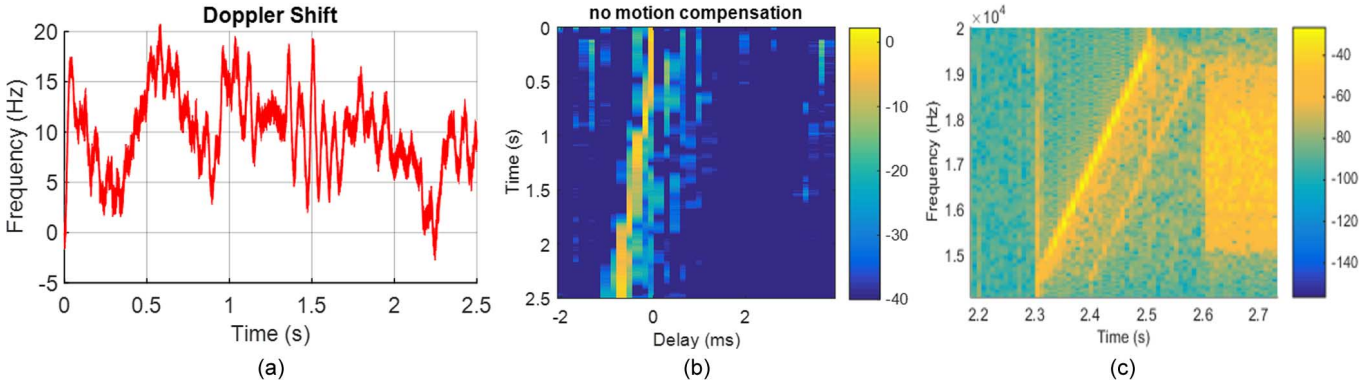


Fig. 5. (a) Motion-induced Doppler shift as function of time for link 160250. (b) Channel response of link 160250 without motion compensation. (c) Spectrogram of the beginning edge of the received signal (link 130304). The amplitude is in decibels.

TABLE I
FIXED PARAMETERS IN ALL ACOUSTIC LINKS

Link	125350	130304	160250
range	1.2 km	1.2 km	2.9 km
modulation	8-PSK	8-PSK	4-PSK
bit rate	9 kbps	9 kbps	6 kbps
# of TX symbols	11,000	10,000	7,500
$N_a + N_c$	71	611	793
$L_c + L_a$	21	21	25
K_1	$3{\cdot}10^{-5}$	$3{\cdot}10^{-5}$	$6{\cdot}10^{-5}$
β	-0.5	0.5	0.5
L	1	1	1
p	1.52	1.52	1.52
δ_{NLMS}	10	10	10
λ_σ	0.99	0.99	0.99

models snapping shrimp dominated ambient noise [9], [10]. Fig. 3(b) verifies this result by plotting the SαS fit along with the Gaussian and the empirical fit of the noise samples of Fig. 3(a).

C. Channel Characterization

Fig. 4(a)–(c) shows the time evolution of the amplitude of the baseband impulse response of all links. Wideband Doppler distortion due to motion is compensated. The channel estimates are generated by using the IPNLMS receiver in training mode. Due to high received SNR, these channel estimates are very reliable since the receiver can achieve error-free communications. Also note that a reduced filter size equalizer that does not span the total delay spread of the channel is used. Table I summarizes the receiver parameters for each link. To gain further insight into how fast each link fluctuates, Fig. 4(d)–(f) shows the time-varying energy of each link by computing $\|\hat{\mathbf{h}}(n)\|_2^2$.

Several important features can be observed from these responses. The first is that channels 130304 and 160250 exhibit very sparse and long multipath spread as compared to channel 125350. For further validation, Fig. 5(c) shows the spectrogram of the beginning edge of the received signal for the 130304 link. A chirp pulse (0.2 s) and its delayed replica due to multipath can be clearly seen. A similar spectrogram can be obtained for channel 160250 (omitted for brevity). It is yet unclear the origin of the distant reflector for channels 130304 and 160250. We believe this reflector is either an island or a reef close to the area of operations [see Fig. 2(b)].

The second feature is the channel short-time variability due to constructive/destructive multipath interference. Fig. 4(d) indicates that the energy of channel 125350 fluctuates about 2 dB over a period of 3.5 s. The mean Doppler shift (not shown for brevity) due to drifting oscillates between $\pm$ 8 Hz. Fig. 4(e) shows that the energy of channel 130304 fluctuates about 7 dB over a period of 3.2 s. The mean Doppler shift oscillates in a similar fashion as in channel 125350. Note the rapid 5-dB energy increase at 0.9 s. This large fluctuation is challenging to be tracked by adaptive receivers. For instance, it makes a standard DA–DFE running on RLS to fail regardless the high SNR. Fig. 4(f) illustrates that the energy of channel 160250 fluctuates about 6 dB over a period of 2.5 s. The energy lows observed at 1.2 and 1.8 s are due to a prominent Lloyd's Mirror effect [32].

Fig. 5(a) shows the time-varying Doppler shift of the received signal for the channel 160250. The positive Doppler validates that the transmitter was propelling toward the receiver at a speed of about 2 kn. Fig. 5(b) illustrates the effect of motion on the channel response when it is left uncompensated. The plot focuses on the multipath arrivals close to 0 ms for visualization purposes. Clearly, the channel response varies very rapidly and eventually makes the DFE to fail.

D. Demodulation Results

The performance metric is the symbol error probability (SER) as a function of the received SNR. Since the data were originally acquired in very high SNR, the following plots are computed by scaling and adding extra ambient noise to the original data. At every SNR, the SER is computed after averaging 15 independent ambient noise recordings. Hence, the reported plots illustrate the average performance of each receiver given the link realization. We emphasize that only the first 900 transmitted symbols were used as a training set for the channel estimator and the RLS equalizer. The fixed receiver parameters for all links are listed in Table I. However, the choice for some channel estimation parameters is not straightforward as it depends on the channel coherence time and received SNR. To ensure a fair comparison between all algorithms, their respective parameters are optimized among some representative values such that the

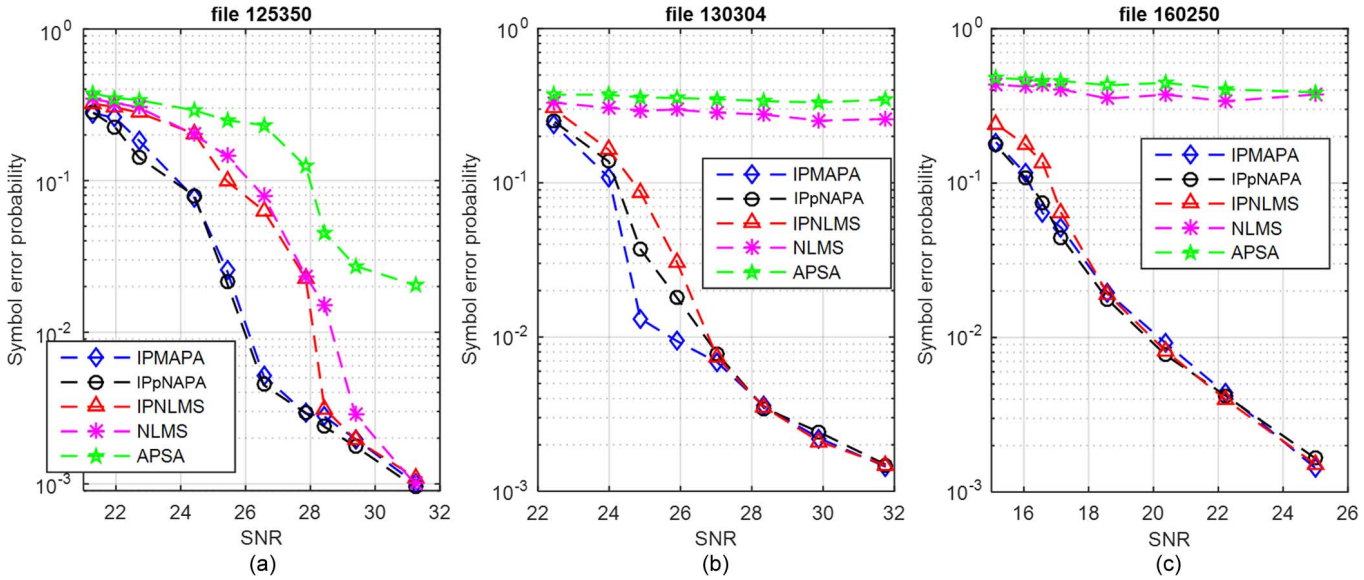


Fig. 6. SER versus SNR for all algorithms: (a) channel 125350, (b) channel 130304, and (c) channel 160250.

lowest SER is obtained for a given received SNR. We consider the following parameters and their representative values:

- $\mu \in \{0.1, 0.2, \ldots, 0.5\}$ for NLMS, IPNLMS, IPMAPA, IPpNAPA;
- $N_w \in \{4, 8, 12\}$ for IPMAPA;
- $\xi \in \{2\bar{\delta}, 4\bar{\delta}, 7\bar{\delta}\}$, $\Delta \in \{8\bar{\delta}, 13\bar{\delta}, 18\bar{\delta}\}$ for IPpNAPA;
- $\mu \in \{0.25\bar{\delta}, 0.5\bar{\delta}, \bar{\delta}, 2\bar{\delta}, 3\bar{\delta}\}$ for APSA [31].

Note that the IPMAPA and the IPpNAPA are optimized, respectively, over 15 and 45 different combinations. Given that both algorithms obtain linear computational complexity, optimization based on an exhaustive search of all combinations during the training period is possible. After the training period, the optimized parameters remain fixed for the rest of the signal duration.

Fig. 6(a)–(c) shows the SER performance of all algorithms for the links 125350, 130304, and 160250, respectively. We observe the following.

- At relative low SNRs, the proposed algorithms outperform all other algorithms in all links. At higher SNRs, noise impulsiveness is not the main challenge and so the sparse algorithms deliver similar performances.
- For the 8-PSK links (125350 and 130304), the proposed algorithms require more than 20-dB SNR to deliver good performance. For the 4-PSK link (160250), however, the proposed algorithms require 5 dB less power to deliver a similar performance. This 5-dB difference is explained by noting that 4-PSK constellations are less sensitive to motion-induced Doppler and impulsive noise than 8-PSK constellations.
- With respect to IPNLMS, the proposed algorithms demonstrate 2-dB power savings for a SER of 10^{-2} for the link 125350. Moreover, IPMAPA shows an advantage of 1 dB at SER $= 10^{-2}$ for the link 130304. On the other hand, the proposed algorithms show no power gain at SER $= 10^{-2}$ for channel 160250. This can be explained by noting that link 125350 does not suffer from deep fades, link 130304 suffers from minor fades, while link 160250 suffers two deep fades at 1.2 and 1.8 s [see Fig. 4(e)]. Hence, it seems that in the absence of strong fading, more gain is achieved with the proposed methods.
- The commonly used NLMS and APSA can deliver good performance only for link 125350. Unfortunately, both algorithms fail in the other two (extended-multipath) links due to their inability to exploit channel sparseness.
- Since the ambient noise is well modeled as SαS, one would expect IPpNAPA to deliver better performance than IPMAPA. However, our plots show that the algorithms have similar performance. These results are consistent with our findings in [19] because when α is larger than about 1.5, then the performance of IPMAPA is close to IPpNAPA.

V. Conclusion

A new channel-estimate-based decision feedback equalizer receiver for underwater acoustic communications was presented. By using data from three shallow-water links, the proposed receiver managed to: 1) tolerate high transmitter/receiver mobility; 2) adapt to rapid channel fluctuations by exploiting channel sparseness; and 3) achieve robustness under impulsive noise. This result was demonstrated by testing different channel estimation algorithms and comparing the respective SER performances. The algorithms tested were IPMAPA [19], IPpNAPA [19], IPNLMS [30], NLMS [29], and APSA [31]. The best performance was achieved by IPMAPA/IPpNAPA as these algorithms are designed to suppress impulsive noise while exploiting channel sparseness.

Appendix A
Derivation of (14)

Let us define

$$\bar{e}(n) = y(n) - \hat{\mathbf{h}}(n)^{\dagger}\mathbf{u}(n) \tag{38}$$

$$e(n) = y(n) - \hat{\mathbf{h}}(n-1)^{\dagger}\mathbf{u}(n) \tag{39}$$

$$\mathbf{r}(n) = \hat{\mathbf{h}}(n) - \hat{\mathbf{h}}(n-1) \tag{40}$$

where $\bar{e}(n)$ and $e(n)$ are the *a posteriori* and *a priori* error at symbol time nT, respectively, and $\mathbf{r}(n)$ denotes the channel update vector. Note that we can express $\bar{e}(n)$ in terms of $e(n)$ and $\mathbf{r}(n)$ as

$$\bar{e}(n)^* = e(n)^* - \mathbf{u}(n)^\dagger \mathbf{r}(n). \tag{41}$$

Via the loss function $f(\mathbf{z})$, $\mathbf{z} \in \mathbb{C}$, we also define

$$\psi(z) = \frac{\partial f(z)}{\partial z} \tag{42}$$

$$q(z) = \frac{\psi(z)}{z^*}. \tag{43}$$

We now compute $\nabla_{\mathbf{r}(n)^*} J(n)$, where $J(n)$ is given by (11). Computing each term individually, we write

$$\nabla_{\mathbf{r}(n)^*} \left(\sum_{i=n-L+1}^{n} f\left(\bar{e}(i)\right) \right)$$

$$= - \sum_{i=n-L+1}^{n} \psi\left(\bar{e}(i)\right) \mathbf{u}(i) \tag{44}$$

$$= - \sum_{i=n-L+1}^{n} q\left(\bar{e}(i)\right) \bar{e}(i)^* \mathbf{u}(i) \tag{45}$$

$$\overset{(41)}{=} - \sum_{i=n-L+1}^{n} q\left(\bar{e}(i)\right) \left(e(i)^* - \mathbf{u}(i)^\dagger \mathbf{r}[n]\right) \mathbf{u}(i) \tag{46}$$

$$= - \sum_{i=n-L+1}^{n} q\left(\bar{e}(i)\right) e(i)^* \mathbf{u}(i)$$

$$+ \sum_{i=n-L+1}^{n} q\left(\bar{e}(i)\right) \left(\mathbf{u}(i)\mathbf{u}(i)^\dagger\right) \mathbf{r}(n) \tag{47}$$

$$= -\mathbf{U}(n)\mathbf{Q}(n)\mathbf{e}(n)^* + \mathbf{U}(n)\mathbf{Q}(n)\mathbf{U}(n)^\dagger \mathbf{r}(n) \tag{48}$$

where

$$\mathbf{U}(n) = [\mathbf{u}(n)\ \mathbf{u}(n-1) \dots \mathbf{u}(n-L+1)] \tag{49}$$

is the $(N_c + N_a) \times L$ matrix of input samples

$$\mathbf{Q}(n) = \begin{bmatrix} q\left(\bar{e}(n)\right) & & 0 \\ & \ddots & \\ 0 & & q\left(\bar{e}(n-L+1)\right) \end{bmatrix} \tag{50}$$

is an $L \times L$ diagonal matrix and

$$\mathbf{e}(n) = \left[eq(nT) \dots z(n-L+1)T)\right]^\top \tag{51}$$

is the prior error vector. We also have

$$\nabla_{\mathbf{r}[n]^*} \left(\delta \mathbf{r}(n)^\dagger \mathbf{P}(n-1)\mathbf{r}(n)\right) = \delta \mathbf{P}(n-1)\mathbf{r}(n). \tag{52}$$

Setting $\nabla_{\mathbf{r}(n)^*} J(n) = 0$ and combining terms, we write

$$\left(\delta \mathbf{P}(n-1) + \mathbf{U}(n)\mathbf{Q}(n)\mathbf{U}(n)^\dagger\right) \mathbf{r}(n) = \mathbf{U}(n)\mathbf{Q}(n)\mathbf{e}(n)^*.$$

From the above equation, we note that $\mathbf{Q}(n)$ depends on $\bar{e}(n)$, which is unknown at time nT. At steady state, however, we can assume that $\bar{e}(n) \simeq e(n)$ and so we evaluate $\mathbf{Q}(n)$ accordingly. Solving for $\mathbf{r}(n)$ via the matrix inversion lemma [29], we have

$$\mathbf{r}(n) = \mathbf{A}(n)\mathbf{B}(n)\mathbf{e}(n)^* \tag{53}$$

where

$$\mathbf{A}(n) = \mathbf{P}(n-1)^{-1}\mathbf{U}(n) = \mathbf{G}(n-1)\mathbf{U}(n) \tag{54}$$

$$\mathbf{B}(n) = \left(\mathbf{U}(n)^\dagger \mathbf{A}(n) + \delta\, \mathbf{Q}(n)^{-1}\right)^{-1}. \tag{55}$$

Using a step size parameter $\mu \in (0, 1]$, we manage to manipulate the change of the tap values from one iteration to the next. Finally, the channel update equation is deduced as follows:

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n) + \mu \mathbf{A}(n)\mathbf{B}(n)\mathbf{e}(n)^*. \tag{56}$$

A. The $\mathbf{G}$ Matrix

As already mentioned above, the purpose of the $\mathbf{P} = \mathbf{G}^{-1}$ matrix is to exploit channel sparseness. A good choice for $\mathbf{G}$ is the diagonal matrix of [30]. Effectively, $\mathbf{G}$ assigns a variable step size parameter to each channel filter tap. This parameter is a function of the tap's previously estimated magnitude. As a result, active filter taps (i.e., taps with significant values) converge fast, which makes the overall algorithm to have fast convergence in sparse channels. We stress that no prior knowledge of the significant tap position is required. The diagonal elements $\{g(\hat{h}_k(n))\}_{k=0}^{N_c+N_a-1}$ of $\mathbf{G}(n)$ are given by [19], [30]

$$g\left(\hat{h}_k(n)\right) = \frac{1-\beta}{2(N_c+N_a)} + \frac{(1+\beta)\left|\hat{h}_k(n)\right|_1}{2\left\|\hat{\mathbf{h}}(n)\right\|_1 + \varepsilon} \tag{57}$$

where ε denotes a small positive constant to avoid division by zero during initialization of the algorithm. Parameter β controls the sparseness of the solution. Typically, one needs to consider four values: $\beta = -1$ (diffuse channel), $\beta = -0.5$ (low sparseness), $\beta = 0$ (sparse), and $\beta = 0.5$ (very sparse). Once β is chosen, parameter δ is given by [30]

$$\delta = \frac{1-\beta}{2(N_c+N_a)} \delta_{\text{NLMS}}$$

where δ_{NLMS} is chosen as in the NLMS algorithm [29]. Note that the channel update equation (56) requires $O(N_c+N_a)$ computational complexity because $\mathbf{G}(n)$ is diagonal and $N_c+N_a > L$ for typical underwater acoustic channels.

ACKNOWLEDGMENT

The authors would like to thank Z. Nan and G. Chua of DSO National Laboratories for their leadership during the sea trials in Singapore.

REFERENCES

[1] A. Baggeroer, "Acoustic telemetry—An overview," *IEEE J. Ocean. Eng.*, vol. OE-9, no. 4, pp. 229–235, Oct. 1984.

[2] M. Stojanovic, "Recent advances in high-speed underwater acoustic communications," *IEEE J. Ocean. Eng.*, vol. 21, no. 2, pp. 125–136, Apr. 1996.

[3] D. B. Kilfoyle and A. B. Baggeroer, "The state of the art in underwater acoustic telemetry," *IEEE J. Ocean. Eng.*, vol. 25, no. 1, pp. 4–27, Jan. 2000.

[4] M. Chitre, S. Shahabudeen, and M. Stojanovic, "Underwater acoustic communications and networking: Recent advances and future challenges," *Mar. Technol. Soc. J.*, vol. 42, pp. 103–116, 2008.

[5] B. Blair, "Analysis of and techniques for adaptive equalization for underwater acoustic communication," Ph.D. dissertation, Massachusetts Inst. Technol./Woods Hole Oceanogr. Inst., Cambridge/Woods Hole, MA, USA, 2011.

[6] J. C. Preisig, "Performance analysis of adaptive equalization for coherent acoustic communications in the time-varying ocean environment," *J. Acoust. Soc. Amer.*, vol. 118, pp. 263–278, 2005.
[7] M. Stojanovic, "Efficient processing of acoustic signals for high rate information transmission over sparse underwater channels," *J. Phys. Commun.*, pp. 146–161, 2008.
[8] M. Bouvet and S. C. Schwartz, "Comparison of adaptive and robust receivers for signal detection in ambient underwater noise," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 37, no. 5, pp. 621–626, May 1989.
[9] M. A. Chitre, "Underwater acoustic communications in warm shallow water channels," Ph.D. dissertation, Dept. Electr. Comput. Eng., Nat. Univ. Singapore, Singapore, 2006.
[10] M. W. Legg, "Non-Gaussian and non-homogeneous Poisson models of snapping shrimp noise," Ph.D. dissertation, Dept. Imaging Appl. Phys., Curtin Univ. Technol., Bentley, W.A., Australia, 2010.
[11] X. Wang and R. Chen, "Blind turbo equalization in Gaussian and impulsive noise," *IEEE Trans. Veh. Technol.*, vol. 50, no. 4, pp. 1092–1105, Jul. 2001.
[12] R. Schober and L. Lampe, "Sequence detection and adaptive channel estimation for ISI channels under class-A impulsive noise," *IEEE Trans. Commun.*, vol. 52, no. 9, pp. 1523–1531, Sep. 2004.
[13] B. Chen, C. Tsai, and C. Hsu, "Robust adaptive MMSE/DFE multiuser detection in multipath fading channel with impulse noise," *IEEE Trans. Signal Process.*, vol. 53, no. 1, pp. 306–317, Jan. 2005.
[14] A. E. El-Mahdy and N. M. Namazi, "Turbo equalisation of time varying multipath channel under class-A impulsive noise," *Inst. Electr. Eng. Proc.—Commun.*, vol. 153, no. 3, pp. 341–348, 2006.
[15] A. T. Georgiadis and B. Mulgrew, "Adaptive Bayesian decision feedback equaliser for alpha-stable noise environments," *Signal Process.*, vol. 81, no. 8, pp. 1603–1623, 2001.
[16] J. Choi, M. Bouchard, and T. H. Yeap, "Recurrent neural equalization for communication channels in impulsive noise environments," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2005, vol. 5, pp. 3232–3237.
[17] L. Sen, Q. Tianshuang, and Z. Daifeng, "Adaptive blind equalization for MIMO systems under α-stable noise environment," *IEEE Commun. Lett.*, vol. 13, no. 8, pp. 609–611, Aug. 2009.
[18] S. Niranjayan and N. C. Beaulieu, "The BER optimal linear rake receiver for signal detection in symmetric alpha-stable noise," *IEEE Trans. Commun.*, vol. 57, no. 12, pp. 3585–3588, Dec. 2009.
[19] K. Pelekanakis and M. Chitre, "Adaptive sparse channel estimation under symmetric alpha-stable noise," *IEEE Trans. Wireless Commun.*, vol. 13, no. 6, pp. 3183–3195, Jun. 2014.
[20] D. H. Brandwood, "A complex gradient operator and its application in adaptive array theory," *Inst. Electr. Eng. Proc. H—Microw. Opt. Antennas*, vol. 130, no. 1, pp. 11–16, 1983.
[21] J. G. Proakis, *Digital Communications*, 4th ed. New York, NY, USA: McGraw Hill, 2000, ch. 4.
[22] B. Li *et al.*, "Multicarrier communication over underwater acoustic channels with nonuniform Doppler shifts," *IEEE J. Ocean. Eng.*, vol. 33, no. 2, pp. 198–209, Apr. 2008.
[23] C. P. Shah *et al.*, "Low-complexity iterative receiver structure for time-varying frequency-selective shallow underwater acoustic channels using BICM-ID: Design and experimental results," *IEEE J. Ocean. Eng.*, vol. 36, no. 3, pp. 406–421, Jul. 2011.
[24] T. Riedl and A. Singer, "MUST-READ: Multichannel sample-by-sample turbo resampling equalization and decoding," in *Proc. MTS/IEEE OCEANS Conf.*, Norway, 2013, DOI: 10.1109/OCEANS-Bergen.2013.6608187.
[25] S. C. Chan and Y. Zou, "A recursive least M-estimate algorithm for robust adaptive filtering in impulsive noise: Fast algorithm and convergence performance analysis," *IEEE Trans. Signal Process.*, vol. 52, no. 4, pp. 975–991, Apr. 2004.
[26] M. A. Chitre, J. R. Potter, and S. H. Ong, "Optimal and near-optimal signal detection in snapping shrimp dominated ambient noise," *IEEE J. Ocean. Eng.*, vol. 31, no. 2, pp. 497–503, Apr. 2006.
[27] A. Mahmood, M. Chitre, and M. Armand, "PSK communication with passband additive symmetric α-stable noise," *IEEE Trans. Commun.*, vol. 60, no. 10, pp. 2990–3000, Oct. 2012.
[28] M. Shao and C. L. Nikias, "Signal processing with fractional lower order moments: Stable processes and their applications," *Proc. IEEE*, vol. 81, no. 7, pp. 986–1009, Jul. 1993.
[29] S. Haykin, *Adaptive Filter Theory*, 4th ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2002, ch. 5.
[30] J. Benesty and S. L. Gay, "An improved PNLMS algorithm," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2002, vol. 2, pp. 1881–1884.
[31] T. Shao, Y. R. Zheng, and J. Benesty, "An affine projection sign algorithm robust against impulsive interferences," *IEEE Signal Process. Lett.*, vol. 17, no. 4, pp. 327–330, Apr. 2010.
[32] F. B. Jensen, W. A. Kuperman, M. B. Porter, and H. Schmidt, *Computational Ocean Acoustics*. New York, NY, USA: AIP Press/Springer-Verlag, 1994, ch. 1.

Konstantinos Pelekanakis (S'06–M'09) received a Diploma from the Department of Electronic and Computer Engineering, Technical University of Crete, Chania, Greece, in 2001 and the M.Sc. and Ph.D. degrees in mechanical and ocean engineering from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 2004 and 2009, respectively.

From 2009 to 2015, he worked with the Acoustic Research Laboratory (ARL), National University of Singapore (NUS), Singapore, as a Research Fellow. Currently, he is a Scientist at NATO Science and Technology Organization, Centre for Maritime Research and Experimentation, La Spezia, Italy. His current research lies in the areas of underwater acoustic communications and robust underwater signal processing.

Dr. Pelekanakis has served as the Vice Chairman and Secretary for the IEEE Oceanic Engineering Society (Singapore chapter) and also as a reviewer of many international journals and conferences.

Mandar Chitre (M'03–SM'11) received the B.Eng. (honors) and M.Eng. degrees in electrical engineering from the National University of Singapore (NUS), Singapore, in 1997 and 2000, respectively, and the M.Sc. degree in bioinformatics and the Ph.D. degree from the Nanyang Technological University (NTU), Singapore, in 2004, and 2006, respectively.

From 1997 to 1998, he worked with the Acoustic Research Laboratory (ARL), NUS, as a Research Engineer. From 1998 to 2002, he headed the technology division of a regional telecommunications solutions company. In 2003, he rejoined ARL, initially as the Deputy Head (Research) and is now the Head of the laboratory. He also holds a joint appointment with the Department of Electrical & Computer Engineering at NUS as an Assistant Professor. His current research interests are underwater communications, autonomous underwater vehicles, and underwater signal processing.

Dr. Chitre has served on the technical program committees of the IEEE OCEANS, WUWNet, Defense Technology Asia (DTA), and Water Side Security (WSS) conferences and has served as reviewer for many international journals. He was the chairman of the student poster committee for the 2006 IEEE OCEANS Conference in Singapore. In the past years, he has served as the Vice Chairman, Secretary, and Treasurer for the IEEE Oceanic Engineering Society (Singapore chapter) and is currently the IEEE Technology Committee Co-Chair of Underwater Communication, Navigation & Positioning. He also serves as a Technical Co-Chair for the IEEE 2012 International Conference on Communication Systems (ICCS).