



SCIENCE AND TECHNOLOGY ORGANIZATION
CENTRE FOR MARITIME RESEARCH AND EXPERIMENTATION



Reprint Series

CMRE-PR-2019-093

A communications and relative navigation architecture for underwater vehicle coordination

Thomas Furfaro, João Alves

June 2019

Originally published in:

OCEANS 2015 - MTS/IEEE Washington, 19-22 October 2015, Washington DC,
USA, doi: [10.23919/OCEANS.2015.7401936](https://doi.org/10.23919/OCEANS.2015.7401936)

About CMRE

The Centre for Maritime Research and Experimentation (CMRE) is a world-class NATO scientific research and experimentation facility located in La Spezia, Italy.

The CMRE was established by the North Atlantic Council on 1 July 2012 as part of the NATO Science & Technology Organization. The CMRE and its predecessors have served NATO for over 50 years as the SACLANT Anti-Submarine Warfare Centre, SACLANT Undersea Research Centre, NATO Undersea Research Centre (NURC) and now as part of the Science & Technology Organization.

CMRE conducts state-of-the-art scientific research and experimentation ranging from concept development to prototype demonstration in an operational environment and has produced leaders in ocean science, modelling and simulation, acoustics and other disciplines, as well as producing critical results and understanding that have been built into the operational concepts of NATO and the nations.

CMRE conducts hands-on scientific and engineering research for the direct benefit of its NATO Customers. It operates two research vessels that enable science and technology solutions to be explored and exploited at sea. The largest of these vessels, the NRV Alliance, is a global class vessel that is acoustically extremely quiet.

CMRE is a leading example of enabling nations to work more effectively and efficiently together by prioritizing national needs, focusing on research and technology challenges, both in and out of the maritime environment, through the collective Power of its world-class scientists, engineers, and specialized laboratories in collaboration with the many partners in and out of the scientific domain.



Copyright © MTS/IEEE, 2015. NATO member nations have unlimited rights to use, modify, reproduce, release, perform, display or disclose these materials, and to authorize others to do so for government purposes. Any reproductions marked with this legend must also reproduce these markings. All other rights and uses except those permitted by copyright law are reserved by the copyright owner.

NOTE: The CMRE Reprint series reprints papers and articles published by CMRE authors in the open literature as an effort to widely disseminate CMRE products. Users are encouraged to cite the original article where possible.

A Communications and Relative Navigation Architecture for Underwater Vehicle Coordination

Thomas C. Furfaro, João Alves
 NATO STO Centre for Maritime Research and Experimentation
 La Spezia, Italy
 {Thomas.Furfaro, Joao.Alves}@cmre.nato.int

Abstract—This paper presents an implementation of a communications stack for underwater communications and relative localisation. The application scenario, under the auspices of the EC MORPH project, is described. The high-level architecture is described, with the relevant hardware and software implementation details. A summary of the localisation methods, based on previous works, is given, followed by a discussion of the encoding mechanism.

Index Terms—underwater communications, underwater acoustics, underwater modem, networking, transport, underwater navigation, long baseline, communications applications

I. INTRODUCTION

Underwater communications systems for connecting marine endpoints suffer significant disadvantages when compared to analogous terrestrial systems. These systems are typically acoustic in nature, due to the high attenuation of light and radio-frequency sources in water. In particular, submerged systems suffer from low propagation speed, low bandwidth, and high channel temporal variability. A related problem is that of underwater navigation and localisation of mobile assets. The lack of GPS-like off-the-shelf solutions has created a hardware ecosystem with very high variability in navigation system performance and applicability, with no single technology providing high quality solutions. These communications and navigational limitations of submerged systems have a high impact on the functionality of cooperative autonomous systems.

A. Software Communications Architectures for the Underwater Domain

Largely as a result of the proliferation of non-standard proprietary modulation and coding schemes, the ecosystem of underwater communications devices generally lacks interoperability. Additionally, networking, transport and media access control layers in disparate devices from different manufacturers are equally incongruous, allowing for zero interoperation. Standardisation efforts, such as JANUS ([1], an open standard for modulation and coding of acoustic communications signals), have the potential to ameliorate the disparities, however, the standardisation process sometimes does not react quickly enough to the changing landscape of the state of the art. With this in mind, the emerging strategy is to employ software-defined open architecture modems (SDOAM) to provide a sandbox in which many complementary or even competing standards and components can co-exist. The current state of

several SDOAM efforts are presented in [2], [3]. Principally, the general strategy of an SDOAM is to provide common interfaces such that both open (such as JANUS) and proprietary (such as from a commercial modem provider) modules may interact, while providing sufficient protections and separation for proprietary or otherwise sensitive brain-ware. The communications architecture presented in this paper is designed (in particular regarding software) to provide modules that may be incorporated into a more formal SDOAM architecture.

B. MORPH

The European Project MORPH (a Marine robotic system of self-ORGanizing, logically linked PHysical nodes), summarised in [4], is developing a cooperative fleet of heterogeneous autonomous marine vehicles to perform detailed 3D imaging and sonar surveys of topologically complex environments, including those whose mean planar orientation is far from horizontal. Such an operating paradigm is atypical of subsurface surveys, which are conventionally considered to be 2.5D – that is, data composed of a single vertical value for a single horizontal coordinate. The high resolution maps and images generated from MORPH are a scientific output to be used by marine biologists who are interested in the species that live in these high-relief areas. However, there are clearly alternative application scenarios in other commercial (pipeline inspection) and defence (hull inspection) industries.

This heterogeneous fleet, the MORPH Supra-Vehicle (MSV), is composed of 5 autonomous platforms, including 4 Autonomous Underwater Vehicles (AUVs), and 1 Unmanned Surface Vehicle, spaced on the order of 10s of meters apart, with some nodes within 5 meters of any bottom structure. The formation control and overall monitoring coordination of the MSV all rely heavily on the communication system presented.

In the MORPH fleet, the capability and responsibility of functions is distributed amongst the heterogeneous formation, instead of existing in a single monolithic vehicle. A Localisation and Sonar Vehicle (LSV) carries a tilting multi-beam sonar and progresses along the survey path in lead position that allows it to detect upcoming obstacles for the two Camera Vehicles (CV), who may be redirected along their path to negotiate obstacles and ensure survey consistency. A Global Communications Vehicle (GCV) simultaneously follows behind the LSV with a similar trajectory, forming a baseline anchor for the two CVs to perform Range-Only Formation

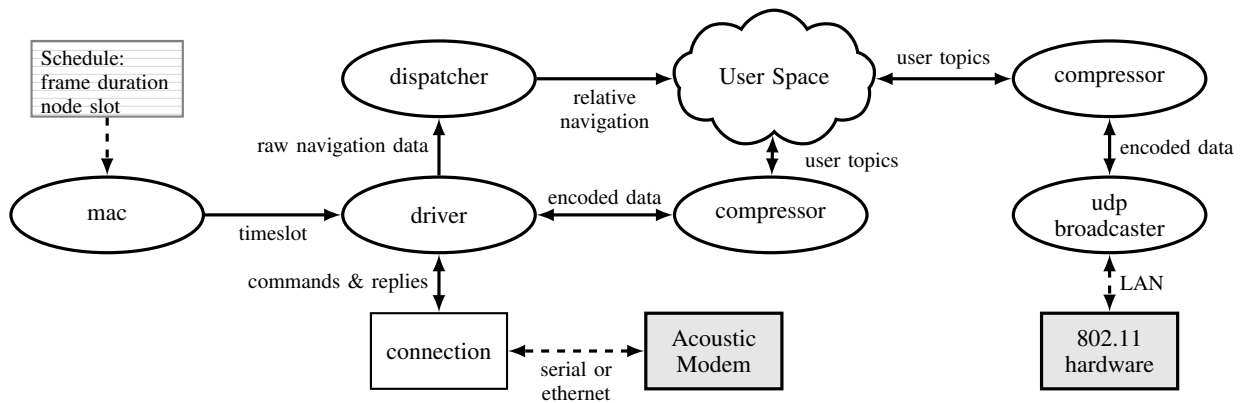


Fig. 1: Baseline acoustic communication system architecture, providing WiFi and acoustic interface.

control (ROF, [5]), and also connecting to the topside Surface Support Vehicle (SSV). The SSV provides a surface expression, providing GPS connectivity and an operator gateway for monitoring and communications. The CVs are conceptualised to be relatively naive and incapable with regards to global navigation – however, the acoustic communications system augments the localisation capabilities by providing a method by which CVs measure their position with respect to the other vehicles in the formation.

II. COMMUNICATIONS ARCHITECTURE

The underwater communications architecture for MORPH has two primary roles: to move data from point to point (i.e., the usual role of any communications system); and to provide measurements for relative localisation of submerged assets. Both of these functions are integral to the autonomous cooperation of the different vehicles. The relative localisation content is composed of acoustic range and direction (USBL) measurements. The end-to-end data delivery function provides a channel over which coordination information, as well as command and control cues, may be sent. Both of these aspects are critical to a formation that is both dense, in terms of node spacing, and within range of structures that are potential hazards to vehicle integrity.

Fig. 1 shows the high-level outline of the MORPH underwater communications architecture. A modem-specific driver interfaces with a connection node that interfaces directly with the acoustic hardware, separating the over-the-wire interface implementation details from the core state machine of the driver. The driver is informed periodically of the permissible time slots available to it, via the MAC process. The MAC process uses a simple Time-Division Multiple-Access protocol based on a static, runtime-configured schedule. The driver emits raw timing and USBL information to a dispatcher node, which calculates ranging measurements and emits parsed relative localisation fixes to other modules in the MORPH system. Conversely, these “user” nodes publish data to be encoded and serialised to a compressor node, which collects and composes encoded binary messages for transmissions from the modem, all sent back to the driver.

A. Networking

The acoustic network has a simple mesh topology, with each node transmitting to every other node. That is, all nodes operate in a promiscuous mode, capable of receiving all transmissions from all other nodes. Due to the short distances between each modem, connectivity, in practice, is very high, and so routing on the acoustic segment is not considered. Moreover, the vast majority of transmitted data focuses on distributed control algorithms, so the latency that would be induced by any multi-hop routing would therefore diminish the utility of the data itself.

B. Hardware

The architecture is built primarily on top of off-the-shelf hardware, with the underwater segment made by set of EvoLogics S2C 18/32kHz acoustic modems, and the above-air segment formed by traditional 2.4 GHz wireless network equipment.

Each of the five vehicles has both the aerial and underwater hardware, but during normal mission runtime, only the surface vehicle has an expression in both domains, providing a bridge. Moreover, the modem of the surface vehicle also contains USBL hardware, allowing direct measurement of the bearing and azimuth of all other submerged modems upon message reception. This is an invaluable asset for mission monitoring during trials, providing frequent (0.2 Hz per node) fixes of the MSV formation.

C. Software Implementation

The MORPH project uses the ROS middleware, [6], as inter-process communication (IPC) method. ROS provisions IPC via an asynchronous, message-based publish-subscribe paradigm, using TCP/IP sockets as the underlying communication medium. ROS messages are described in a simple human-readable interface description language (IDL) encapsulated in `msg` files. ROS has language bindings for both C++ and Python.

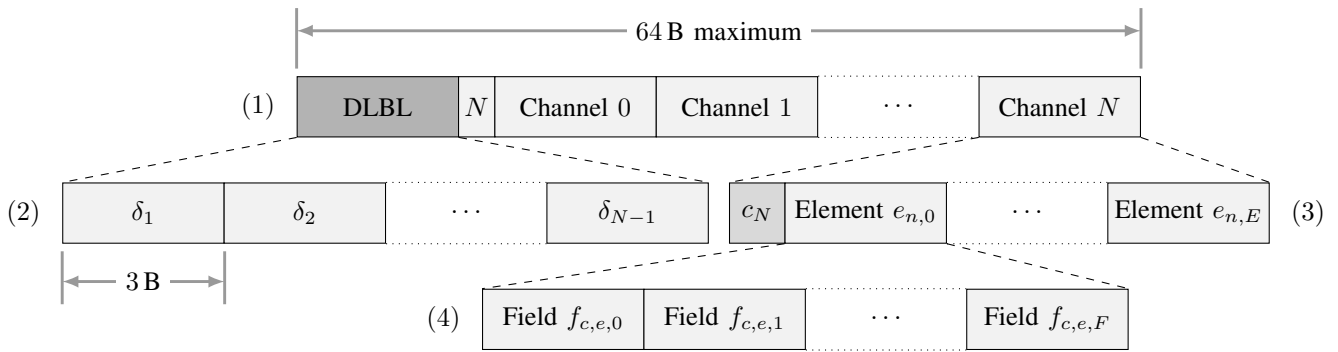


Fig. 2: MORPH acoustic communications packetisation scheme.

D. Relative Navigation

Relative localisation is provided by the system in two modes: relative range measurements, and relative azimuth and bearing measurements. The angular values are measured directly by the USBL modem hardware – an array of five hydrophones mounted in a tetrahedral shape is used to estimate the incoming signal direction. In the MORPH architecture, the GCV and SSV both have USBL units. The range measurements, however, are a relatively novel component of the system, utilising Distributed Long BaseLine, presented in [7]. DLBL is a LBL formulation for relative underwater ranging that removes the traditional distinction between receivers and beacons, instead providing a method for all nodes in the network to have equivalent localisation roles and capabilities. In short, DLBL provides a way of using unsynchronised clocks to perform two-way travel time (TWTT) measurements by precisely controlling time of transmission start and integrating the in-hardware turn-around-time in normal acoustic packets. In this way, the range measurements can be performed without synchronised clocks, and without the explicit acknowledgement usually incurred by two-way ranging, while, under favourable PER conditions, enjoying the best attributes of each approach, as described in [7]. The DLBL measurements may then be used by higher-level localisation algorithms to determine relative and absolute positions, such as that presented in [8], or even directly in formation control algorithms, such as that used in MORPH [9].

Furthermore, DLBL has been conceived to integrate simply into existing acoustic communications devices, with varying networking and MAC algorithms – DLBL does not inherently induce any timing requirements (in the MAC sense) in the host communications environment and adds very small overhead to acoustic message payloads.

Summarising DLBL operation briefly, each node in the network precisely time tags every reception from other nodes, and in the subsequent transmission replies with a payload containing the time difference between those previously-recorded receptions and the current transmission, as a sort of implicit acknowledgement. In this way, the turn-around time of each “reply” can be chosen to comply to the requirement of the MAC.

E. Encoding

Included in the improvements of the communications sub-system is the introduction of a new, generalised, encoding and decoding program, named the “compressor”. The compressor, written in Python, allows users to configure the compression, transmission, and decompression of arbitrary ROS topics. This includes topics with arbitrary structures with arbitrary depth. This new software replaces the legacy encoding mechanism, which required additional topics and structures to be explicitly provisioned in the source code. Additionally, the compressor provides new latching options, the ability to send topic updates on a specified interval, as well as a plug-in system that lets users introduce their own kind of encoding and decoding.

Fig. 2 shows the packetisation scheme encapsulated with the new compressor. Using the current EvoLogics modem firmware, each packet can contain up to 64 B of data, Fig. 2(1), with the ensonified signal duration ranging from 160 ms for a single user byte transmitted to a 850 ms long signal for a full-length, 64 B message. Because the packet error rate is sensitive to the packet length, we prefer to keep the packet length as small as possible, and in practice the total length is less than 30 B long. The payload generated by the communications stack begins with timing information required by the DLBL algorithm, composed of a 3 B, μ s-resolution delay, δ_n , for each foreign node in the network, as shown in Fig. 2(2). This vector of delays, Δ , is encoded in every outgoing packet, is a fixed length for a given mission configuration, and is required to perform relative ranging between nodes. This operation is performed at the modem driver level, as hardware-specific timing information is included.

Following the localisation information, the role of the compressor begins. After the modem driver constructs the delay vector, it requests a payload from the compressor. The compressor provides an already-encoded set of channel payloads, with a leading length, C , indicating the number of channel payloads included in the particular message.

Each channel, as shown in Fig. 2(3), is composed of a channel index value, c , as well as a series of elements, E_n . The channel index, c , is any of the globally configured set of possible channels, C , $c \in C$. Additionally, channel indices may not repeat within a single message, $\forall c \in C, \forall i, j \in$

```

- latching:      false
  from:         "lsv"
  to:           "gcv"
  rate:         10.0
  elements:
  - topic:      {
      in : "target_reference",
      out : "lsv/target_reference",
      msg : "geometry_msgs/Vector3Stamped"
    }
  fields:
  - field:     "vector.x"
    bits:     8
    min:      0
    max:     6.2831
  - field:     "vector.y"
    bits:     6
    min:    -0.052359878
    max:     0.052359878

```

Listing 1: Example of a channel configuration in the compressor process. The message type, `geometry_msgs/Vector3Stamped`, contains a timestamp and id field, `header`, and a vector, `vector`, with the x, y, z components.

$N, c_i = c_j \iff i = j$. This global channel set is defined in the configuration file that each node has at mission start.

Every channel has several configurable attributes, as seen in Listing 1, including a source and destination pattern (`from` and `to`, respectively), a latching specification (true or false, default is false), a transmit rate, and a series of *elements*. The source and destination attribute for each channel can include lists of nodes, or regular expressions that match node names. A latching attribute with value `true` indicates that the given channel should continuously transmit information even if no new data is received from the host environment. In the default case, where `latching = false`, channels are only transmitted if new data is received from the host. The last configurable attribute for channels is the `rate`. The rate, a duration specified in seconds, allows one to selectively throttle the transmission of channels. A channel with a rate configured to a non-zero value will transmit, approximately, data at that rate. The default value, with `rate = 0`, will cause transmission to happen at the driver transmit rate (every cycle).

Fig. 2(3) shows how each packed channel is composed of a series of elements. The elements that correspond to a channel are encoded in every channel transmission – if any elements of that channel are sent, all elements are sent. In this way, a specific channel will have a fixed bit length for the entire configuration lifetime. With respect to ROS, an element corresponds to a topic and message type. The topic from which each element’s data is read is contained in the element’s `topic['in']` attribute, and the `topic['out']` attribute is the topic on which the element will be published in the receiving nodes. ROS does not provide a way of dynamically looking up the message type of a topic, thus the message type must be specified in the configuration, contained in the `topic['msg']` field.

An element is composed of a set of fields, Fig. 2(4). The fields themselves are a subset of all of the message elements

TABLE I: Reception Statistics for 3-Node Experiment

RX/TX (PSR,RSR)	TRANSMITTER		
	sparus	mblack	girona
[USR] sparus	—	211/225 (0.938, 0.898)	211/225 (0.938, 0.867)
RECEIVER	mblack	214/228 (0.939, 0.890) [0.939]	225/228 (0.987, 0.912) [0.987]
	girona	201/227 (0.885, 0.859)	210/227 (0.925, 0.912)

contained in the specified message type. In this way, the message type, which is a structure or arbitrary depth, can be represented, at least partially, by an element and its fields. The specific fields to be encoded are in the `elements[i].fields` section of the configuration. At start time, the process loads the specified message type and inspects the fields configured to determine their basic type (any of `char`, `byte`, `string`, `bool`, `float`, `int`) and automatically loads the correct *codec* for the basic type, unless a custom codec is specified with the `elements[i].fields[j].codec['module']` and `elements[i].fields[j].codec['class']`. The fields are typically of numeric type (`float` or `int`), as the example in Listing 1, so the minimum, maximum, and encoded bits must be specified. Custom codecs simply derive from the `Field` base type, and implement the `_encode()` and `_decode()` member methods.

The compressor is necessarily implemented in Python, as `rospy`, the Python module for ROS, provides *introspection*, while the C++ `roscpp` does not provide the same functionality. The compressor design and flexible configuration interface utilises this attribute heavily, and would otherwise be difficult to implement without making custom modification to the `roscpp` and `gen_msg` packages provided by ROS.

The compressor explicitly does not support any advanced queuing mechanism. Simply, data received by the compressor is considered to be the most relevant, so old data is overridden.

III. EXPERIMENTAL RESULTS

The MORPH communications architecture presented here has been subject to constant development, beginning in 2012. An integral part of the development cycle is use in real sea-trials, providing the communications and relative navigation backbone. The most recent trial (as of submission) was the GIRONA15.2, held in the harbor of Sant Feliu de Guíxols, off the Catalan coast. During this trial, three partner institutions (from IST in Lisbon, Portugal; ATLAS Elektronik in Bremen, Germany; UdG in Girona, Spain) brought AUVs to participate in collaborative formation control experiments, with assistance from CMRE and IUT (from Ilmenau, Germany).

During these experiments, the MORPH communications stack provided an inter-vehicle message relay system, as well

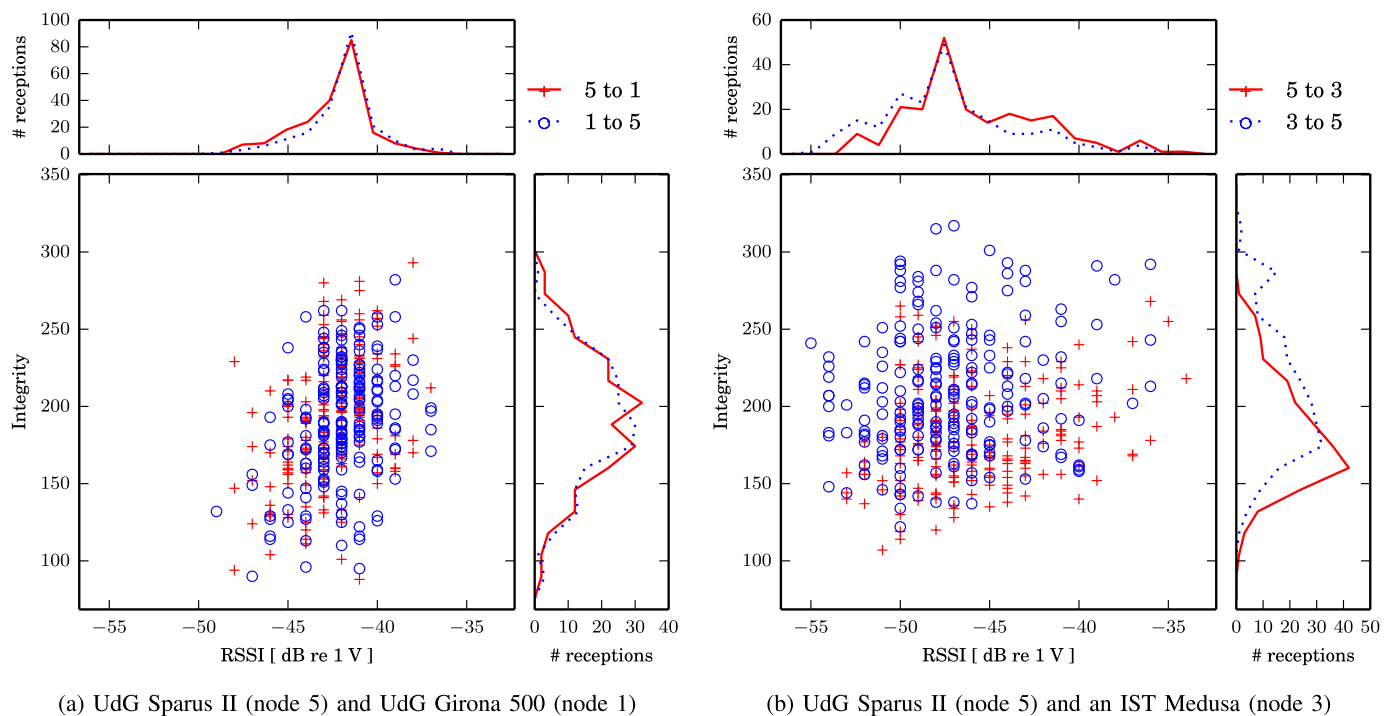


Fig. 3: Received Signal Strength Indicator (RSSI) and signal integrity distributions (inversely proportional to the amount of distortion) for two links of a 3-node mission.

as critical inter-node range and position information. Presented here is a subset of the experiment data, focusing on a trial from July 3, 2015, including the AUVs IST Medusa, UdG Girona 500, and UdG Sparus II. The experiment took place in a harbour environment, over 20 min, with approximately 8 m–12 m of total water depth, with the vehicles operating close (on the order of 3 m) to solid concrete walls. The inter-node distance is nominally on the order of 10 m, though it changes dynamically as a result of the formation control algorithms. MAC is performed by a simple TDMA algorithm, with a total cycle duration of 5 s and single vehicle slot time of 1 s. Table I shows high level communications system performance for node pairs, including the total number of packets transmitted and received, as well as the packet success rate (PSR), range success rate (RSR). Additionally, the USBL-fix success rate (USR), is shown for the node (`mblack`) that has a USBL receiver. A USBL measurement is considered successful if a azimuth and bearing to the transmitter can be estimated. The RSR metric, for a receiver is calculated as the ratio of the number of successful range measurements divided by the number of transmissions from the foreign node. The DLBL is based on two-way packet exchanges, so the RSR upper bound is dictated by the PSR for both the outgoing and incoming acoustic links. As is listed, PSR values experienced are very high (88.5%–98.7%), with a slightly lower RSR range (86%–92%).

In this scenario, the IST Medusa and UdG Girona 500 are at comparable water depths (~ 4 m) for the duration of the mission, fulfilling the roles of GCV and LSV, respectively,

while the UdG Sparus II, as a CV, navigates with respect to the ranges measured between itself and the GCV and LSV, while maintaining an position relative to the local wall. Figure 3 shows the comparative link quality between two node pairs. The received signal strength indicator is a measure of the power received on the receiver node of the link, while the integrity is a qualitative measure of the distorting inter-symbol interference (ISI) and multipath effects. These metrics are provided natively by the modem manufacturer’s firmware.

IV. CONCLUSION

This paper presents a software architecture for underwater localisation and communications with relevant performance statistics. The work is driven by the requirements of the MORPH EC project, which conceptually uses a heterogeneous fleet of marine vehicles to perform high-fidelity 3D surveys. This network topology is characterised by inter-node distances on the order of 10 m. The architecture, as a virtue of the modem hardware (USBL) and DLBL algorithm, produces relative angle and range fixes between nodes. The inter-node ranges are based on a TWTT measurement, while the USBL angle measurements are generated by the modem firmware asynchronously with packet reception. The communications function of the architecture is enabled largely by the compressor process, an encoding tool that enables run-time configuration of data channels to be transmitted. Additionally, high level communications performance is presented, in terms of packet success rate and range success rate metrics, as well as link quality measurements.

ACKNOWLEDGEMENT

This research has been partially funded by the European Union Seventh Framework Programme (FP7-ICT-2011-7) under grant agreement no. 288704.

REFERENCES

- [1] J. R. Potter, J. Alves, D. Green, G. Zappa, K. McCoy, and I. Nissen, "The janus underwater communications standard," in *UComms 2014 Conference Proceedings*, 2014.
- [2] J. Potter, J. Alves, T. Furfaro, A. Vermeij, N. Jourden, G. Zappa, A. Berni, and D. Merani, "Software-defined open-architecture modems: State of the art and the way forward," in *UComms 2014 Conference Proceedings*, 2014.
- [3] H. Dol, P. Casari, and T. van der Zwan, "Software-defined open-architecture modems: State of the art and the way forward," in *UComms 2014 Conference Proceedings*, 2014.
- [4] J. Kalwa, A. Pascoal, P. Ridaou, A. Birk, M. Eichhorn, L. Brignone, M. Caccia, J. Alves, and R. Santos, "The European R&D-Project MORPH: Marine robotic systems of self-organizing, logically linked physical nodes," in *9th IFAC Conference on Manoeuvring and Control of Marine Craft*, Sep. 2012.
- [5] J. M. Soares, A. P. Aguiar, A. M. Pascoal, and A. Martinoli, "Joint asv/auv range-based formation control: Theory and experimental results," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 5579–5585.
- [6] Open Source Robotics Foundation, "Robot operating system (ROS)," <http://www.osrfoundation.org/> and <http://www.ros.org/>.
- [7] T. Furfaro and J. Alves, "An application of distributed long baseline – node ranging in an underwater network," in *UComms 2014 Conference Proceedings*, 2014.
- [8] V. Ludovico, J. Gomes, J. Alves, and T. Furfaro, "Joint localization of underwater vehicle formations based on range difference measurements," in *Underwater Communications and Networking (UComms), 2014*, Sept 2014, pp. 1–5.
- [9] P. Abreu, M. Bayat, and A. M. Pascoal, "Formation control in the scope of the MORPH project. part II: Implementation and results," in *IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles (NGCUV)*, Girona, Spain, April 2015.

Document Data Sheet

<i>Security Classification</i>		<i>Project No.</i>
<i>Document Serial No.</i> CMRE-PR-2019-093	<i>Date of Issue</i> June 2019	<i>Total Pages</i> 6 pp.
<i>Author(s)</i> Thomas Furfaro, João Alves		
<i>Title</i> A communications and relative navigation architecture for underwater vehicle coordination		
<i>Abstract</i> <p>This paper presents an implementation of a communications stack for underwater communications and relative localisation. The application scenario, under the auspices of the EC MORPH project, is described. The high-level architecture is described, with the relevant hardware and software implementation details. A summary of the localisation methods, based on previous works, is given, followed by a discussion of the encoding mechanism.</p>		
<i>Keywords</i> Underwater communications, underwater acoustics, underwater modem, networking, transport, underwater navigation, long baseline, communications applications		
<i>Issuing Organization</i> NATO Science and Technology Organization Centre for Maritime Research and Experimentation Viale San Bartolomeo 400, 19126 La Spezia, Italy [From N. America: STO CMRE Unit 31318, Box 19, APO AE 09613-1318]		Tel: +39 0187 527 361 Fax: +39 0187 527 700 E-mail: library@cmre.nato.int