Investigation of Underwater Acoustic Networking Enabling the Cooperative Operation of Multiple Heterogeneous Vehicles

Nuno A. Cruz[*†], Bruno M. Ferreira[*†], Oleksiy Kebkal[‡], Anbal C. Matos[*†], Chiara Petrioli[§¶], Roberto Petroccia[§], Daniele Spaccini[§¶]

[*]Institute for Systems and Computer Engineering-Technology and Science (INESC TEC), Porto, Portugal.

[†]Faculdade de Engenharia da Universidade do Porto (FEUP), Porto, Portugal.

[‡]EvoLogics GmbH, Berlin, Germany.

[§]Dipartimento di Informatica, Università di Roma "La Sapienza," Via Salaria, 113, 00198, Roma, Italy. E-mail: {petrioli,petroccia,spaccini}@di.uniroma1.it.

[¶]Italian Interuniversity Consortium on Computer Science(CINI).

Abstract

In this paper we investigate the creation of an underwater acoustic network to support marine operations based on static and mobile nodes. Each underwater device combines communication, networking, and sensing capabilities, and cooperates with the other devices in coordinated missions. The proposed system is built upon the SUNSET framework, providing acoustic communications and networking capabilities to AUVs, ASVs and moored systems, using underwater acoustic modems. Specific solutions have been developed and tested to control the underwater nodes acoustically and to instruct the vehicles on keeping a given formation using acoustic links. One of the novelties of our approach was the development and utilization of a realistic simulation infrastructure to provide a very accurate representation of all the dynamic systems involved in the network, modeling the vehicle dynamics, the acoustic channel, and the communication messages. This infrastructure has been extensively used to investigate and validate the proposed solutions under different environmental conditions before the actual deployment of devices. Several experiments have then been conducted in lab and in field: the experimental results have confirmed the effectiveness of the proposed solutions and the reliability of the proposed simulation framework in estimating system performance.

Index Terms

Underwater acoustic networks, vehicle coordination, SUNSET, autonomous marine vehicles, distributed simulation framework.

I. INTRODUCTION

The main motivation for this paper arises from the need to establish an acoustic network for a fleet of heterogeneous vehicles, including Autonomous Underwater Vehicles (AUVs), Autonomous Surface Vehicles (ASVs) and moored systems. The utilization of robotic systems in the underwater environment has become increasingly common in the last years. In most cases, real-time communications with such platforms is needed for a combination of reasons such as safety and control, and the need to periodically get data from onboard sensors. This requirement is being extended to permanent real-time communications among vehicles and between vehicles and remote centers, due to emerging new ocean sampling paradigms which start to exploit the coordinated operation of complementary systems. With the ability to communicate and cooperate in a distributed way, such devices may constitute a more flexible, reliable and effective observation system for water borne operations.

In the underwater environment, both radio and optical signals are greatly attenuated and acoustics are still the main technology used for communications. Nonetheless, acoustic solutions suffer from long propagation delays, low data rates and several factors affecting the quality of the received signals (multipath, attenuation, etc), which complicate the implementation of dependable networks. In fact, some commercial solutions do provide the ability to exchange information between nodes but only some basic networking capabilities. In scenarios where multiple vehicles paper in cooperation, the communication network can take advantage of the multiple available nodes, either fixed or mobile, possibly from different vendors. Moreover, such a network has to be flexible enough to accommodate the dynamics of such nodes, both in terms of physical locations and in terms of performance.

In this paper we present and validate a novel paradigm where each underwater device combines communication, networking, and sensing capabilities, and cooperates with the other devices to accomplish a given task using acoustic links. The SUNSET framework has been used to provide networking capabilities and acoustic communication support to a set of heterogeneous underwater platforms, using Evologics acoustic modems for communication. New protocols and algorithms have been implemented for nodes communication and cooperation allowing to control the underwater nodes and to instruct the vehicles on keeping a given formation. Moreover, a simulation infrastructure has been implemented to investigate and validate the performance of the proposed solutions under different environmental conditions, before the actual deployment of the devices in field trials. The distributed nature of the infrastructure has been exploited to run multiple simulations with independent modules connected via Internet and spread across three different countries: the software emulating the vehicle dynamics was running in Porto, Portugal; SUNSET was running in Rome, Italy, providing the protocol solutions to reserve the channel and deliver data from source to destination; and, finally, a software emulating the Evologics acoustic modem was running in Berlin, Germany. The experimental results confirm the validity and reliability of the proposed emulation approach, showing a good agreement between the results from simulation and field trials (with separation between real and simulated position of vehicles within 1 meter). Moreover, the collect results demonstrate also the efficiency and reliability of the proposed solution where acoustic communication is used for vehicle control and cooperation, opening to several possibilities for future developments. The remainder of the paper is organized as follows. Section II introduces SUNSET, a generic tool to provide transparent communication and networking capabilities to underwater robotic platforms. Section III presents the technological assets used in this paper and summarizes their main characteristics in order to detail, in Section IV, the main efforts involved in the adaptations of the tools. Section V discusses the results obtained from simulation and in field experiments, and analyzes the performance of the coordinated system under different communication scenarios. Finally, Section VI concludes the work.

II. Framework for Acoustic Networking and Communication

To enable networking capabilities and to provide support for acoustic communication to the underwater nodes, the Sapienza University Networking framework for underwater Simulation Emulation and real-life Testing (SUNSET) (Petrioli and Petroccia, 2012) has been used. SUNSET has the potential to result in a radical paradigm shift in the way underwater experimental research is performed. Researchers and developers can easily implement novel protocols, test the protocol performance through simulations (investigating the impact of different deployments and of changing environmental and acoustic channel models parameters on performance) and then port the exact implemented code to different real underwater platforms for in field testing. SUNSET also seamlessly integrates heterogeneous underwater platforms: Different modems, communications technologies, and platforms (both mobile and static) are currently supported and thus can be interconnected together by means of SUNSET communication protocol stacks. In what follows we describe the SUNSET architecture and how networking and communication capabilities can be implemented and provided in a straightforward way to static and mobile underwater nodes.

A. *The SUNSET Architecture*

SUNSET is a new solution to seamlessly simulate, emulate and test in real-life novel communication protocols. It is based on the open source and well known network simulator ns-2 (The VINT Project) - and its extension ns2-Miracle [1] - and it has been made freely available to the research community (Petrioli and Petroccia, 2012). The reason for extending a well known open source architecture instead of designing an optimized ad hoc one is that it will be more easy to use by the networking community, thus speeding-up the investigation and development of new protocol solutions for underwater sensor networks, and fostering the interest of networking experts towards underwater protocol design. Moreover, the proposed approach is also extremely flexible and allows code reuse. SUNSET implements a layered structure and supports cross-layer messages for the communication among different layers. Moreover, SUNSET allows the coexistence of multiple solutions within each layer of the protocol stack (Figures 2.1 and 2.2). Different modules have been designed and implemented to make the execution of simulation and emulation easier and more transparent to the user (Timing, Utilities, Information Dispatcher), to provide procedures for debugging and protocol performance analysis (Debug and Statistics). Additional modules have also been provided for data packet compression (Packet Converter) and to allow the interaction with external devices (Drivers).

Using the proposed framework, anyone willing to implement its solution for simulation experiments using ns-2 can use the same code, either in emulation mode or in field, adopting real acoustic modems for data transmission and additional external devices for sensing and navigation operations.

Several routing, MAC and cross-layer solutions have already been implemented in SUNSET, including: TDMA, CSMA (Basagni et al, 2012b), Slotted CSMA, T-Lhoi (Syed et al, 2008), DACAP (Peleato et al, 2007), Flooding-base solutions, Depth Based Routing (Yan et al, 2008), Focused Beam Routing (Journet et al, 2008), an improved version of the routing solution presented in (Alves and Zappa, 2011), Channel-Aware Routing Protocol (Basagni et al, 2012a), etc.

When running simulations (Figure 2.1), SUNSET can use different underwater acoustic channel models, such as empirical formulas (Urick, 1983) and Bellhop ray tracing (Porter) via the WOSS (Guerra et al, 2009) interface.

When running in emulation mode (Figure 2.2), instead, real acoustic modems and additional devices are used. New modules and several drivers have been developed to allow a proper interaction with the external real hardware and to make transparent to the user the switch between a simulated underwater channel and the use of real acoustic modems.

Specifically, WHOI Micro-Modem (Freitag et al, 2005), Evologics modems (Evologics), Kongsberg modems (Kongsberg) and Teledyne Benthos modem (Teledyne) have been interconnected to SUNSET. SUNSET has also been successfully integrated and interfaced with different types of devices, including sensors for underwater measurements (Annunziatellis et al, 2012) and the AUV, ASV and NIB devices used for the experiments described later in this work.

The time needed to design and implement the specific driver is very short and proportional to the complexity of the operations required by the specific hardware. Radio, optical and other type of communication devices can be interfaced to SUNSET as well, thus providing the possibility to explore and take advantage of multiple communication interfaces available at a given node to improve network performance.

To make easier the deployment at sea, SUNSET code has successfully been ported on small portable devices (Gumstix, PC104 or other ARM-based systems), thus allowing us to embed it inside modem or AUV and ASV housings.

B. SUNSET Interaction with Mobile Vehicles

In the absence of real-time communication capabilities, preloaded missions are typically used when operating underwater vehicles resulting in no direct control on the vehicle once it is underwater. Instead, using the acoustic communications and networking capabilities provided by SUNSET, requests and commands can be delivered to a remote node (via single-hop or multi-hop transmissions), thus allowing the remote control of the devices using acoustic links in real-time. Mobile underwater vehicles can therefore be controlled while they are operating underwater, thus significantly improving the mission supervision and control flexibility when performing a given task.

To allow SUNSET to use and interface with different devices, such as AUV or ASV navigation control systems, a driver is needed at the application layer to properly handle the data exchanges and interactions with them (see Figure 2.1.b).

A generic interface has been developed for SUNSET to support basic commands of AUV/ASVs in a transparent and flexible way. This interface provides a generic API implementing the basic operations supported by mobile vehicles (navigating to a given location, changing depth or velocity, etc.), which are then translated into the specific vehicle instructions by the specific vehicle drivers. This generic interface can be easily extended to support additional operations according to the connected vehicle. In this way, new drivers for new vehicles can be easily developed and connected to the interface, without any modification to the interface itself.

This generic infrastructure allows SUNSET to provide a powerful transparent abstraction layer to robots which do not need to know the details of the networking and communication layers (which protocols have been used, what acoustic modem or communication device) but are able to communicate and cooperate with each other or with other nodes in the network by transmitting and receiving their generated messages through SUNSET.

Moreover, SUNSET allows to encode and decode the remote commands in order to reduce as much as possible their size. This permits to transmit complex commands with several parameters and instructions in a single short transmission.

C. SUNSET ranging capabilities

Data transmission and ranging estimation are two operations which usually interfere with each other: When the vehicle is ranging it cannot transmit acoustic data and when it transmits data it cannot estimate ranging to the other nodes. Moreover, they usually require separate hardware. Being able to exchange data and information while ranging using the same hardware would effectively improve the vehicle and network capabilities. For this reason, a new SUNSET module which combines together communication and ranging operations has been designed and investigated. This new module uses Evologics modem synchronous instant messages [2] to estimate ranging information while exchanging data packets with the other nodes in the network. Distances are estimated based on the round trip delay between ranging request and response messages. Moreover, in each range request and response message additional information, such as the estimated position of the node, can be transmitted, allowing the use of mobile nodes as transponders for navigation support.

III. Technological Assets

In order to effectively deploy an acoustic network for the operation of heterogeneous vehicles, it is fundamental to adapt the framework to the specific assets available for the implementation (mainly vehicles and communication devices). To prepare such implementation, in this section we describe the devices considered in our heterogeneous network and the benefits envisaged from the implementation of an underwater communications network to support cooperation of marine vehicles.

A. Robotic Platforms

1) The MARES AUV: The MARES AUV [3] is a small sized vehicle developed by the Oceans Systems Group, at FEUP/INESC TEC, in Portugal (Figure 1). It has a torpedo shaped body, with about 1.5 m of length and 20 cm of diameter, weighting slightly over 30 kg in its basic configuration. The vehicle can dive up to 100m of depth, gathering ocean data with a set of on-board sensors, and the body modularity can be exploited to easily include additional systems. Unlike similar sized vehicles, MARES does not have any moving fins to adjust heading and depth. Instead, 4-DOF motion is provided solely by independent thrusters, which allow the vehicle to travel at velocities up to 2m/s or hover in the water column and/or approach any structure for detailed inspection. The MARES navigation system is based on a Long Base Line acoustic positioning system: the vehicle software continuously fuses ranges to a set of acoustic beacons, together with compass heading, inertial data, depth and velocity, to compute the estimated position in real-time. Depending on the environmental conditions, the acoustic beacons can be located at a maximum distance of 1-2 Km from the AUV. At the end of the mission, a smoothing algorithm improves the accuracy of the position estimates.

With the new acoustic network, MARES is able not only to communicate with the other nodes, but also to obtain ranges to these nodes, in real time, in order to improve vehicle navigation.

2) Navigation and Instrumentation Buoys: Navigation and Instrumentation Buoys (NIBs) are moored surface platforms with rechargeable batteries, an ARM-based single board computer with Wi-Fi, a GPS receiver and a radio modem (Figure 2). NIBs can be connected to a great variety of sensors and transmit

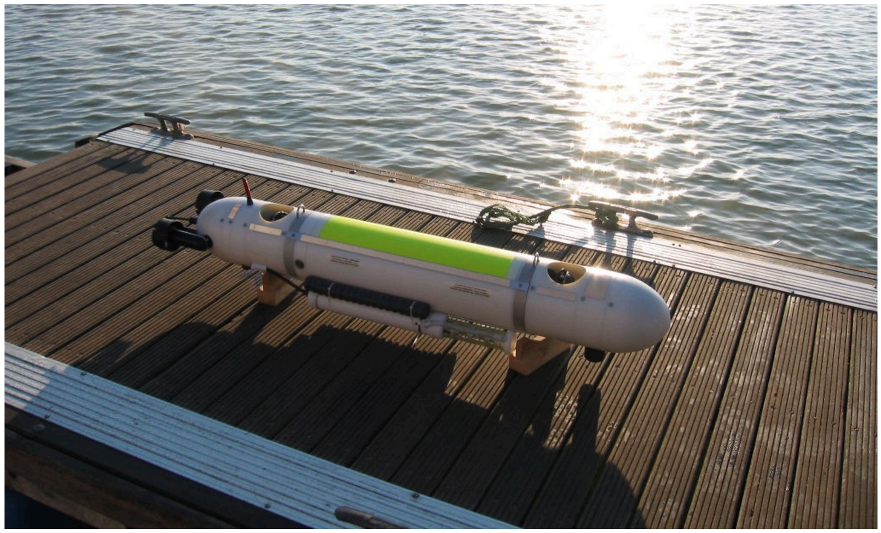

Figure 1: The MARES AUV with an external CTD sensor.

Figure 2: A Navigation and Instrumentation Buoy.

data in real-time. During AUV missions, they get relevant information about the local environment (such as current profiles or data from reference sensors, for example), to allow for post-mission data processing and interpretation of AUV data.

NIBs are also used as acoustic transponders for AUV navigation. They receive and decode acoustic signals sent by the vehicle and respond with other coded pings. Since they are deployed in known positions, the AUV can determine its own position by trilateration. Alternatively, the buoys may send synchronized signals, so that the AUV may estimate ranges based only on one way travel time [4]. During AUV missions, the buoys also relay navigation information back to the mission control station, allowing for vehicle trajectory tracking [5]. The GPS location of the NIBs is logged, to allow for post-

Figure 3: Zarco and Gama ASVs during in field trials at the Douro River.

mission correction of sensor data location, compensating for dislocation due to wind and currents. With the availability of the acoustic network, these variations can be forwarded to the AUVs in real-time, together with other information regarding the environmental characteristics (currents, salinity, etc), thus significantly improving the AUVs position estimation.

3) Zarco and Gama ASVs: Zarco and Gama are two small Autonomous Surface Vehicles (ASVs) (Figure 3), developed by the Ocean Systems Group to conduct experiments in navigation, control, and vehicle cooperation [6], [7]. The vehicles have been used to carry several types of sensors for bathymetry and water quality monitoring, and they have also been used to transport acoustic beacons for AUV navigation, acting as virtual moorings. The availability of differential GPS (DGPS) receivers ensures an absolute position with errors less than one meter. Both ASVs have multiple communication capabilities based on radio (from Wi-Fi to long range RS-232 radio modems), taking advantage of the high bit rates and low propagation delays to establish a permanent connection. The implementation of the acoustic network provides a backup link to these vehicles and, more importantly, results in advanced new features: the ASVs can be used as moving gateways to interact with AUVs or operate as mobile transponders for AUV navigation.

4) Vehicle emulator: The evaluation of guidance, control and localization algorithms through simulation is fundamental before final deployment of marine systems in real scenarios, and the Ocean Systems Group has developed simulators to mimic the dynamic behavior of the vehicles used in this work. They are based on 6-DOF models of the vehicles (or vehicle emulators) simulated on Matlab/Simulink, and their effectiveness have already been demonstrated in previous works [8], [9], [10]. The emulators imitate the outputs of the sensors and allow different environmental scenarios by changing current drifts, for example. To each vehicle corresponds a single emulator, thus making the approach extensible to as many vehicles as required.

The same exact version of the control layer used onboard is used to govern the motion, by feeding the dynamics emulator with commands, and to interact with SUNSET in order to transmit the node position to other cooperative platforms and obtain references. The current emulation architecture makes the software implementation transparent when moving to field experiments. Furthermore, the control layer provides a common interaction between external processes and the several heterogeneous vehicles, thus interfacing with SUNSET independently of their characteristics or types.

B. Control Architectures of the Heterogeneous Vehicles

1) Control architectures for individual autonomous vehicles: One of the most important aspects in the control of an individual autonomous vehicle is the specification of trajectories. Versatile trajectories and, more generally, behaviors, require the control architecture to include elemental maneuvers. A control software has been designed so that each robotic platform (ASV or AUV) is able to perform the same set of maneuvers:

- **Line-following**: The vehicle tracks a line, while keeping a given velocity;
- **Circle-following**: The vehicle tracks an arc defined by the center and radius;
- **Target tracking** (station-keeping): The vehicle tracks a point and remains stationary;
- **Waypoint**: This maneuver is concluded once the vehicle reaches a preset position;
- **Free maneuver**: Each degree of freedom can be commanded directly by setting either velocity or pose references.

Using this set of maneuvers, any type of complex trajectory can be followed by setting a coherent sequence of instructions. This can be set either by using a static mission script or by instructing the vehicles on-the-fly via a communication link. From the robotics point of view, the maneuvers can be seen as a set of feasible tasks. This small, yet versatile, set of tasks enables a clear interaction with and among the vehicles, thus fitting well in a distributed scenario using a higher-level cooperation algorithm.

This architecture has already been successfully implemented in several robotic heterogeneous marine platforms such as Zarco and Gama ASVs [6], the MARES AUV [3] and, more recently, the TriMARES Hybrid AUV/ROV [11], [8].

2) Coordination algorithm: The control scheme used in this paper to achieve coherent motion of the team is based on the method presented in [12]. This centralized approach assumes the existence of a virtual leader, which can be coincident with one of the robots, and an unlimited number of followers. The leader collects the positions of the different followers and instructs them on the position references to keep the required formation. One of the main advantages of this method is that under, individual bounded errors, the formation is guaranteed to be stable. This is especially important since it makes it possible to abstract from the lower level control layers, which are beyond the scope of this work, and therefore disregard the individual errors caused by control parameter mismatches or drifts, for example, as long as they remain bounded. Furthermore, the mathematical construction of the method makes the formation tolerant to communication losses and delays. As long as bounded delays and periodical exchange of feedback data and commands are guaranteed, the formation can be maintained and smooth evolution can be obtained. Here, we summarize the main mathematical derivations and adaptations. For the original complete derivation, the reader is referred to [12]. We consider the case with N vehicles moving in formation. We denote the position of the vehicles as $\boldsymbol{\eta_i}(t) \in \mathbb{R}^n, i = 1, \ldots, N$ and the one corresponding to the virtual leader as $\boldsymbol{\eta_0}(t) \in \mathbb{R}^n$. The later is assumed to be continuous and differentiable. Assume that a smooth path $\boldsymbol{\rho_0}(s_0(t))$ defines the trajectory of the virtual leader, where the scalar $s_0(t)$ parametrizes the path, i.e. $\boldsymbol{\eta_0}(t) = \boldsymbol{\rho_0}(s_0(t))$.

For a given a mission, each vehicle should assume a specific position in the formation, with respect to the virtual leader. This position is usually pre-determined when designing the mission and is given by $\boldsymbol{\tilde{\eta}_i^*} = \boldsymbol{\eta_i^*} - \boldsymbol{\eta_0}$. For any given instant, the dynamic desired positions of the vehicles are given by $\boldsymbol{\tilde{\eta}_i^d}$, also referred to the virtual leader position. Note that these are individual references that must converge to $\boldsymbol{\tilde{\eta}_i^*}$, but they may not be coincident for all t. The actual real relative positions of the vehicles, with respect to the virtual leader, are computed in real time throughout the mission and are defined as $\boldsymbol{\tilde{\eta}_i} = \boldsymbol{\eta_i} - \boldsymbol{\eta_0}$.

In order to drive the individual references to their respective positions in the formation, a potential-field-like function is used in this scheme. Consider a *formation constraint function* $F : \mathbb{R}^n \times \mathbb{R}^n \times \cdots \times \mathbb{R}^n \to \mathbb{R}^+$, differentiable and strictly convex, for which the arguments of $F(\tilde{\boldsymbol{\eta}}_1^{\boldsymbol{d}}, \tilde{\boldsymbol{\eta}}_2^{\boldsymbol{d}}, \ldots, \tilde{\boldsymbol{\eta}}_N^{\boldsymbol{d}}) = 0$ are uniquely determined, i.e. $F^{-1}(0) = (\tilde{\boldsymbol{\eta}}_1^{*}, \tilde{\boldsymbol{\eta}}_2^{*}, \ldots, \tilde{\boldsymbol{\eta}}_N^{*})$. A possible solution for the present problem can be $F(\tilde{\boldsymbol{\eta}}_1^{\boldsymbol{d}}, \tilde{\boldsymbol{\eta}}_2^{\boldsymbol{d}}, \ldots, \tilde{\boldsymbol{\eta}}_N^{\boldsymbol{d}}) = \sum_{i=1}^{N} \|\tilde{\boldsymbol{\eta}}_i^{\boldsymbol{d}} - \tilde{\boldsymbol{\eta}}_i^{*}\|^2$, which means that we aim at minimizing the overall tracking error. We note that the behavior of the desired positions $\tilde{\boldsymbol{\eta}}^{\boldsymbol{d}}{}_i$ has not been defined yet. We can design it to meet the convergence properties for the formation. We define it to be parametrized by $s_i(t)$, that is, $\tilde{\boldsymbol{\eta}}_i^{\boldsymbol{d}} = \tilde{\boldsymbol{\eta}}_i^{\boldsymbol{d}}(s_i(t))$. Its time derivative can be written as $\dot{\tilde{\boldsymbol{\eta}}}_i^{\boldsymbol{d}} = \frac{\partial \tilde{\boldsymbol{\eta}}_i^{\boldsymbol{d}}}{\partial s_i} \dot{s}_i$. We take

$$\frac{\partial \tilde{\boldsymbol{\eta}}_i^{\boldsymbol{d}}}{\partial s_i} = \nabla_{\tilde{\boldsymbol{\eta}}_i^{d}} F(\tilde{\boldsymbol{\eta}}_1^{\boldsymbol{d}}, \tilde{\boldsymbol{\eta}}_2^{\boldsymbol{d}}, \ldots, \tilde{\boldsymbol{\eta}}_N^{\boldsymbol{d}})$$

and

$$\dot{s}_i = g_i(\|\tilde{\boldsymbol{\eta}}_i - \tilde{\boldsymbol{\eta}}_i^{\boldsymbol{d}}\|)$$

where $g : \mathbb{R}^+ \to \mathbb{R}^+$ is a positive decreasing function of its arguments, that verifies $g(\alpha) > 0$,$\forall 0 < \alpha < M$, with $M > 0$.

In practice, this means that each vehicles reference follows a path that minimizes the gradient of the formation constraint function, $F(\tilde{\boldsymbol{\eta}}_1^{\boldsymbol{d}}, \tilde{\boldsymbol{\eta}}_2^{\boldsymbol{d}}, \ldots, \tilde{\boldsymbol{\eta}}_N^{\boldsymbol{d}})$, while its evolution rate, $s_i(t)$, is given by a decreasing function of the distance to the reference. In other words, the reference evolution will slow down when the distance to the reference $\tilde{\boldsymbol{\eta}}_i^{\boldsymbol{d}}$ increases.

The virtual leader acts in a similar fashion but considers takes into account the errorss of each all vehicles. The time derivative of its position is given by $\dot{\boldsymbol{\eta}}_0(t) = \frac{\partial \rho_0(s_0)}{\partial s_0} \dot{s}_0$, where we choose

$$\dot{s}_0 = \prod_{i=1}^{N} g_i(\|\tilde{\boldsymbol{\eta}}_i - \tilde{\boldsymbol{\eta}}_i^{\boldsymbol{d}}\|).$$

This choice for the evolution rate of the virtual leader was inspired by the work of (Egerstedt and Hu, 2001), but instead of an exponential relationship with the error, we generalize to a broader range of functions $g_i(\|\tilde{\boldsymbol{\eta}}_i - \tilde{\boldsymbol{\eta}}_i^{\boldsymbol{d}}\|)$. As a result, similarly to the references of the real vehicles, the evolution of the virtual leader position will tend to slow down as the individual errors grow. The demonstration of the convergence of this control is very similar to the one found in (Egerstedt and Hu, 2001), whereby we will not demonstrate it here. Nonetheless, the adaptations made in the presented approach ensure convergence of the formation only if the individual tracking errors are upper bounded by a scalar $M > 0 : \|\tilde{\boldsymbol{\eta}}_i - \tilde{\boldsymbol{\eta}}_i^{\boldsymbol{d}}\| < M$.

The current implementation considers the following functions:

$$g_i(\|\tilde{\boldsymbol{\eta}}_i - \tilde{\boldsymbol{\eta}}_i^{\boldsymbol{d}}\|) = \begin{cases} \bar{g}_i - \alpha_i \|\tilde{\boldsymbol{\eta}}_i - \tilde{\boldsymbol{\eta}}_i^{\boldsymbol{d}}\|, & \text{if } 0 \leq \alpha_i \|\tilde{\boldsymbol{\eta}}_i - \tilde{\boldsymbol{\eta}}_i^{\boldsymbol{d}}\| \leq \bar{g}_i \\ 0, & \text{if } \alpha_i \|\tilde{\boldsymbol{\eta}}_i - \tilde{\boldsymbol{\eta}}_i^{\boldsymbol{d}}\| \leq \bar{g}_i \end{cases}$$

with the positive constants $\bar{g}_i$ and α_i. These constants laters are tuning parameters that define the rigidity of the formation and its maximumal speed. Note that if $\alpha_i \|\tilde{\boldsymbol{\eta}}_i - \tilde{\boldsymbol{\eta}}_i^{\boldsymbol{d}}\| \geq \bar{g}_i$ for any $i = 1, \ldots, N$, the leader stops and the formation will eventually stop after stabilizationreaching steady state. In (Egerstedt and Hu, 2001) the evolution rate is always positive, even if the errors are arbitrarily large. This is reasonable under normal operation with consistent communications links, but may not be adequate when the communication infrastructure suffers from a great variety of perturbations, as it is the case of underwater acoustic communications.

If the rigidity of the formation can be relaxed, a proper tuning of the these parameters $\bar{g}_i$ and α_i allows faster evolution of the overall formation. For example, decreasing α_i would make the error corresponding to the vehicle i less penalizing in the evolution of the formation thus providing a mechanism for increasing the overall velocity of the formation. Naturally, this can also be achieved with more accurate tracking controllers, which is beyond the scope of this work. Note that α_i also defines the maximum tracking error allowed to the vehicle i before stopping the formation. For safety reasons, both $\bar{g}_i$ and α_i have to be carefully chosen to avoid potential collisions and the range of admissible values depends on the formation geometry ($\tilde{\boldsymbol{\eta}}_i^{\boldsymbol{*}}$, defined above), i.e., smaller distances require more rigid formations.

Finally, an important feature is that the coordination algorithm can be run in any of the formation mates (provided they are not communication-impaired) thus extending the area of operation of autonomous vehicles.

C. Acoustic Modems

1) The Evologics S2C acoustic modems: For the tests Evologics S2CR 18/34 acoustic modems (Evologics) were used. This model is designed for medium-range operations in shallow water. The transducer has horizontally omnidirectional beam pattern, used frequency range is 18 to 34 KHz and maximum data rate up to 13.9 Kbit/s.

Physical layer of the Evologics acoustic modems is based on the Sweep Spread Carrier (S2C) technique (Kebkal and Bannasch, 2002). Modems continuously spread the signal energy over a wide range of frequencies and adapt the signal structure so that the multipath components do not interfere with each other. At the receiver end, advanced signal processing collects the energy and converts the received signals into narrow band signals. This results in achieving significant depression of multipath disturbances and substantial system gain, enabling successful decoding of signals also in crucial environments even when they are heavily masked by noise.

The S2C physical layer allows to detect accurately the time of signal arrival and an extended set of data-link layer commands allows to implement upper layer protocols, efficiently combining acoustic communication with ranging task ([2], [13]).

2) The S2C modem emulator: The major purpose of the Evologics modem emulator is to minimize development costs of upper layer protocols and to simplify the integration of acoustic modems into underwater infrastructures. The main consideration for modem emulator design was that an application, developed with the emulator, must work with real acoustic modems without any code modifications.

This defines the following design requirements:

- Real-time emulation of a large number of network nodes.
- Same source code for both the emulator and the real modem's hardware.
- Equal command set for both the emulator and the modem.
- Remote emulator access via Internet.

A multitude of instances of the emulated acoustic modem representing an underwater acoustic network made of many nodes can be configured and launched on the dedicated server, remotely accessible via Internet. The user is granted remote access to this acoustic network. Specifically, each modem can be accessed via TCP/IP socket. This means that it can be used to connect together different devices and software modules placed in different locations over Internet. The components of the modem emulator are described in details in [14]. There are no differences in the source code between the emulation environment and real hardware.

Rather than attempting to reproduce the detailed physical effect of acoustic wave propagation, the physical layer and channel models are oriented to reproduce the phenomena of signal propagation, essential for upper layer protocols. This phenomena includes in particular long propagation delays, packet collisions, bit error rate as a function of channel impulse response [13], rays bending according to sound velocity profile. The physical layer simulator introduces a propagation delay by holding the packet for a timeout that corresponds to the signal propagation time between the signal source and the receiver, detects collisions and drops collided packets or modifies the packet according to some bit error rate.

Another phenomenon, important for the upper layer protocols' development, in particular, combining communication and localization algorithms, is support of node's mobility. The physical layer simulator can obtain position update in real time via a TCP/IP socket and update on the fly the propagation delays between the nodes accordingly. The motion model itself is considered as external component that can be connected to the emulator.

Thus the emulator, implementing the described above functions, essentially reduces the need for real hardware during the R&D and testing, as well as enables to extend the number of possible test scenarios, hardly manageable even with real modems at the development stage. In particular:

- The emulator allows arbitrary propagation delays between network nodes.
- The emulator supports real-time testing of multiple underwater acoustic network nodes at once, while several dozens of modems, batteries, buoys, anchor chains and other accessories are an unaffordable luxury for a network protocol developer.
- The emulator fully supports the S2C modems' cross-layer synchronization mechanisms, essential for implementing accurate positioning protocols.
- The physical layer simulator supports collision detection, user-defined demodulation and synchronization error rates and mobility of the nodes, allowing to test applications and upper layer protocols in different operating conditions and debug the code to improve system stability without involving expensive underwater infrastructures.

IV. Implementation of a Distributed Testing Infrastructure

We have used SUNSET to set up a distributed simulator/emulator framework to investigate the performance of the coordination algorithm described above before performing in field tests. SUNSET implements a really flexible and general architecture where different devices or software can be interconnected together in several ways, including also TCP sockets and Internet connection. This means that, using SUNSET, different components, locally or remotely connected together, can work and cooperate to provide a more general and capable simulation and emulation framework to the end user. As presented in

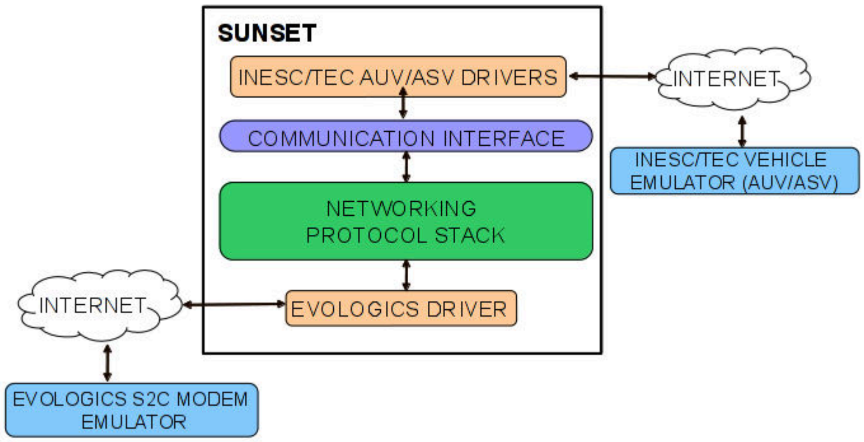

Figure 4: Distributed testing infrastructure.

Figure 4, SUNSET has been used to interconnect, via Internet, the vehicle emulator software developed at INESC TEC/FEUP and the S2C modem emulator developed by Evologics GmbH. Several instances (one per vehicle) of the software emulating the vehicle operations were running at INESC TEC/FEUP in Porto, Portugal. SUNSET, running in Rome, Italy, (one SUNSET instance per vehicle) was providing the communication support and the networking capabilities to reserve the channel and to deliver data from source to destination. A software emulating the acoustic modems operations was then running at Evologics GmbH in Berlin, Germany. One emulated modem was connected to each SUNSET instance, mimicking the exact operations of the real modem. The S2C modem emulator was also introducing the acoustic underwater propagation delays according to the simulated node positions. To better represent realistic scenarios the estimated probability of error on the correct packet reception was also introduced using SUNSET. Using this configuration and multi-countries distributed framework, we have been able to test in a controlled environment the same software controlling the vehicle and the modem, the same protocol stack solutions, and the same driver implementations then used during the in field experiments. Several simulations have been performed with the twofold objective to a) validate the proposed communications solutions and the performance of the coordination algorithm and b) investigate the performance of the proposed system under a wide set of possible scenarios beyond the ones enabled by our in field deployment.

In what follows we briefly describe how SUNSET was interconnected to the software emulating the behavior of an underwater vehicle and the S2C modem emulator to provide our distributed framework.

A. SUNSET Integration with Vehicles

Specific drivers have been designed to interface SUNSET with the different vehicles (AUV and ASV) presented above. Since the same command and status interface runs on MARES AUV and Zarco and Gama ASVs, the same SUNSET interface and connections are used for both types of vehicles. From the interface point of view, the only differences stands in the vehicle operations implemented by the specific drivers, as the operations supported by the ASVs are a subset of the ones implemented by the AUV. Using these drivers and interfaces SUNSET has been able to interact with the different vehicles in order to collect the position information required by the coordination algorithm, and to instruct the nodes on the navigation operations to perform according to the leader requests. Requests and commands can been transmitted by all the vehicles in a transparent way ignoring lower levels information, making use of the SUNSET networking capabilities. The same interfaces are used to interconnect to the vehicle emulator.

B. SUNSET Integration with S2C Modem Emulator

SUNSET has been connected via Internet to the S2C modem emulator. Two different TCP connections have been used, one for data and one for control information. For the normal acoustic communication interaction with the modem (data flow), the Evologics driver developed for SUNSET has been used. The control flow was instead used to instruct the emulator on the position of the vehicles to properly compute and introduce the corresponding propagation delay in the acoustic communication. To handle the control message interaction with the emulator, a specific module has been implemented in SUNSET.

V. Experimental Results

In this section we first present the set up of a heterogeneous underwater network with different underwater nodes communicating and cooperating together via acoustic links. We then investigate the performance of the presented coordination algorithm through simulation and in field tests. We also demonstrate the viability of the proposed distributed framework to assess the performance of the investigated solutions under different environmental and acoustic channel conditions before the actual devices deployment. Several trials with increasing complexity have been conducted at the Ocean Systems Group laboratory and at the Douro river, close to Porto, Portugal, and simulation and in field results have been compared. SUNSET running on an Gumstix embedded device has been integrated inside each vehicle (AUV and ASV) and instrumentation buoy and connected to Evologics modems S2C R 18/34 for acoustic communication. In all the investigated scenarios a single-hop network has been considered and different MAC protocols have been used.

A. Heterogeneous Network Set Up

In February 2012, first tests have been conducted in the test tank at the Ocean Systems Group laboratory to investigate the MARES AUV driver and its integration with SUNSET. CSMA [15] protocol has been used at the MAC layer. One Evologics modem has been connected to a control station running SUNSET on a PC and another one to the Gumstix inside the vehicle. Acoustic commands have been sent from the control station to instruct the vehicle on the requested operations to perform. Preliminary results have been positive as the AUV was able to receive, process and correctly execute commands in real-time, including the variations of pitch and yaw angles (and rates) and of vehicle depths. In July 2012, further tests have been conducted with two ASVs and two NIBs, both at the laboratory and at the Douro River, close to Porto. During these experiments, the two ASVs (Zarco and Gama) were cooperating using acoustic messages, instructing each other on the tasks to perform. Again, the CSMA protocol has been used at the MAC layer. Even though a few data packets were lost at the MAC layer, the networking capabilities of SUNSET allowed to correctly deliver all request commands to the destination node and the mission ended successfully, with both ASVs following the expected trajectory at the river. During the July tests, acoustic communications have been used by the four nodes to estimate inter-node distances and to then broadcast these information to the other nodes in the network. Underwater and surface vehicles can use the collected information to support control and localization algorithms. Evologics modem features [2] have been used for ranging estimation. Additional details on the conducted tests can be found in [16].

B. Coordinated Operation of ASVs

In what follows we investigate the performance of the presented coordination algorithm and we define the metric used to evaluate the accuracy of the vehicles formation. Since the idea is to run both simulation and real tests and to compare the obtained results to validate the coordination algorithm itself and validate the presented distributed framework, in all the investigated tests only surface vehicles have been considered. Real life tests with surface vehicles are much easier to handle and provide a higher control on the system, allowing to run test in short time with a higher degree of freedom. Moreover, GPS information collected on the surface contribute for a more accurate comparison between simulated and real data, as it is more precise than typical trilateration or triangulation-based algorithms used underwater. We point out however that both the considered AUV and ASV use the same control software, which means that performing in field tests including also AUV requires zero effort in terms of software integration. Next step will be investigating a scenario where AUVs, ASVs, and NIBs are deployed communicating and cooperating together.

We start evaluating the performance of the coordination algorithm through simulation, using the distributed framework described above. We then perform in field tests at the Douro River and compare simulation and in field results assuming the same number of vehicles and approximately the same environmental and channel (packet error rate) conditions to validate the efficiency and accuracy of the proposed distributed framework. Finally, we show the coordination algorithm performance through simulation when considering more vehicles in the network.

For all these tests, a TDMA protocol has been used at the MAC layer, repeating a period of x slots, one for the leader and $(x-1)$ slots for the followers (one per follower). According to the coordination algorithm, the leader instructs the followers to keep a given formation. After the leader sends the references to all the vehicles, each follower replies with its position in its own slot.

1) Performance metrics: It is expectable that, under different environmental conditions and different parameters, the performance of the formation algorithm with respect to the geometry varies. It is important to define a generic measure to assess the performances of the overall system. As in typical multi-vehicle missions the relative positions of the cooperative robots are important, we give emphasis to the distance among pairs of vehicles to characterize the performances of the coordinated team. To quantify the error in the formation geometry, we have used the following metric, which expresses the absolute error average resulting from the sum of the absolute difference between the actual distance between each pair of vehicles and the desired one:

$$M = \frac{1}{t_f - t_i} \binom{N}{2}^{-1} \int_{t_i}^{t_f} \left(\sum_{i=1}^{N} \sum_{j=i+1}^{N} |\rho_{ij}^* - \rho_{ij}(t)| \right) dt$$

where t_i and t_f are the starting and ending time, respectively; $\rho_{ij}^* = ((\boldsymbol{\eta}_i^* - \boldsymbol{\eta}_j^*)^T (\boldsymbol{\eta}_i^* - \boldsymbol{\eta}_j^*))^{\frac{1}{2}}$ is the desired distance between vehicle i and j; and $\rho_{ij}(t) = ((\boldsymbol{\eta}_i(t) - \boldsymbol{\eta}_j(t))^T (\boldsymbol{\eta}_i(t) - \boldsymbol{\eta}_j(t)))^{\frac{1}{2}}$ is the actual inter-distance of a pair of vehicles $(i,\, j)$. Note that we do not include the virtual leader position in the computation.

2) Simulations: Using the proposed distributed framework, it is possible to assess the performance of the overall system varying several relevant parameters, such as packet error rate (PER). Vehicle drifts or formation geometry rigidity (by adjusting the parameters α_i) can also be varied to analyze the robustness of the approach. In what follows we mainly focus on the influence of the PER on the formation performance. We do not address here the effect of the drifts in terms of tracking performances since it is related with the low-level control law and is independent of the mean of communication, update rate and delays.

However, with the increase of the drifts, we expect larger tracking errors in some situations as a result of the increased effort of the tracking lower level control law, which in turn contribute for a slower evolution of the formation. Note that this happens regardless of the communication network used, i.e., the same is expected for Wi-Fi as for acoustic communications. Nonetheless, as long as the absolute individual errors are less than $\bar{g}_i/\alpha_i$, the evolution of the formation is guaranteed. Although independent of the communication characteristics, the formation rigidity parameters α_i can be set according to the desired performances of the formation, which include evolution rate and permitted deformation, and thus be balanced with the update rate and delay. We have conducted three sets of simulations in which we have kept the leader trajectory constant and varied the PER in the set $0, 0.15, 0.30$. Two ASVs have been considered acting as followers and one static node acting as leader.

Figure 5. displays the trajectories followed in our experiment when the PER is equal to 0%: Vehicles follow a smooth trajectory and are able to keep their relative positions. Even using the same leaders path, it is expectable that the formation geometry rigidity varies as a function of the information exchange rate and transmission errors. This effect is clearly shown in Figure 6, which displays the trajectories originated by both the vehicles for a PER equal to 30%, where we can notice small deviations from the original paths and a formation geometry deformation with respect to the case for a PER equal to 0% (Figure 5). The same remaining environmental conditions and parameters were kept constant in these simulations. The deviations are mainly due to the need to stabilize on points belonging to the path: When the reference sent by the virtual leader is delayed due to a packet loss, the vehicle will hold the last position reference received, which implies an effort to compensate the drift. Because of the non-holonomy of the robotic platforms considered in this work, the vehicles have to face the drift vector when they are required to

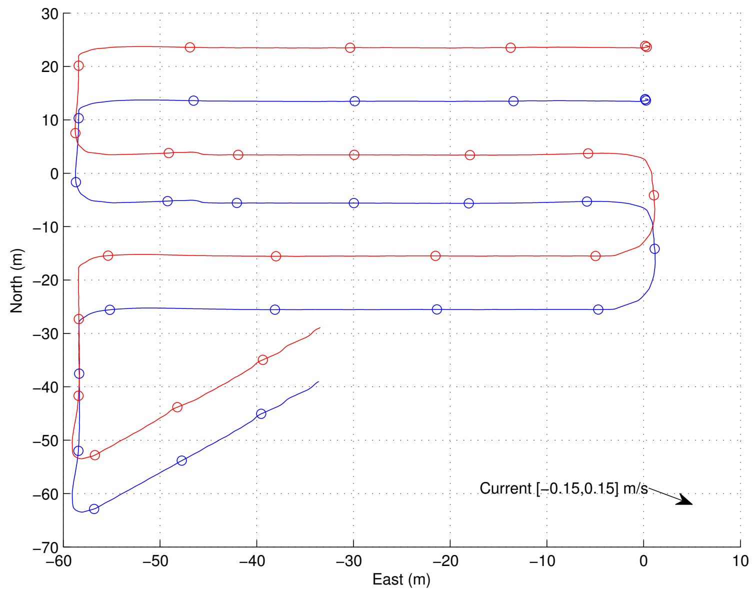

Figure 5: Simulated trajectories for two followers (PER = 0%; $\boldsymbol{v}_c = (-0.15, 0.15)$; $\alpha = \frac{1}{15}$; $v_{max} = 1.5$).

Table I: PER influence on trajectory error metric.

Packet Error Rate	M (m)	Completion Time (s)
0%	0.12	295
15%	0.83	350
30%	1.13	400

hold their positions. For the sake of illustration, suppose that the virtual leader transmits the individual references to the followers at a given time. Each vehicle will track its own reference and wait for the next one. If the next reference is not transmitted before a vehicle reaches the reference, it will stabilize on it and try to compensate the disturbances induced by the drift. Because of the vehicle non-holonomy, the only means to compensate the sideslip is to align the direction of the forward thrust vector with the direction of the drift. In practice, the vehicle has to face the drift vector to hold its position.

The geometry deformation is caused by asynchronous reception of the individual references, arising in part from the packet losses and also from the different distances between the vehicles and the leader node, thus the different propagation delays before individual references are received. In Table I, we show how the PER influences the error metric over approximately the same trajectories. Clearly, increasing the packet error rate degrades the formation geometry. Although the same trajectories are not reproduced

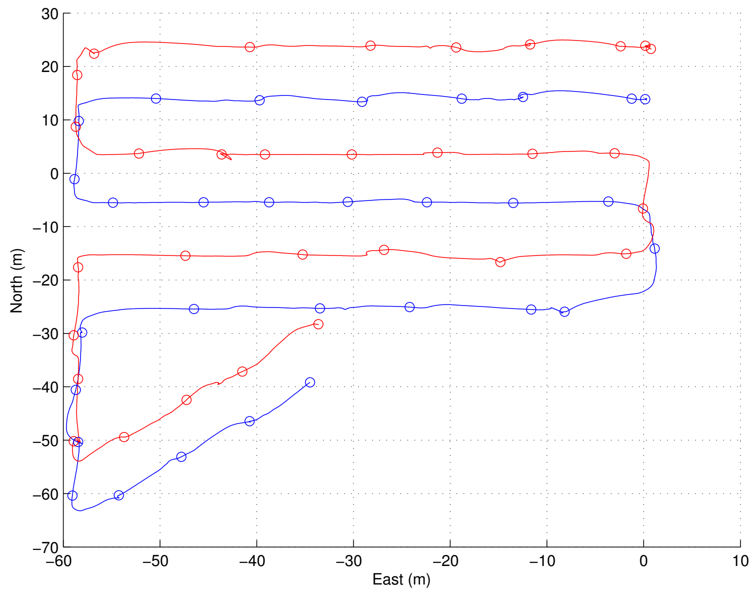

Figure 6: Simulated trajectories for two followers (PER = 30%; $\boldsymbol{v}_c = (-0.15, 0.15)$; $\alpha = \frac{1}{15}$; $v_{max} = 1.5$).

because of the randomness of the simulation, the completion time of a similar sub trajectory was included in the table for the sake of completeness. The results show an approximately linear relationship between the packet error rate and the traveled distance. When the number of packet losses increases due to the larger PER, the evolution of the formation slows down in the absence of new references from the leader.

3) In field results and validation of the simulation framework: In August 2012, in field tests have been conducted at the Douro River, close to Porto, where two ASVs have been considered acting as followers and one static node on shore acting as leader.

Before running the in field trials we had also performed simulations through our distributed framework varying PER and drift values to investigate the expected performance of the coordination algorithm during the river trial. We have considered values in a range similar to the one experienced during the previous tests of July 2012. In what follows we compare simulation and in field results considering in case of simulations approximately the same parameters experienced at the river: A constant drift vector ($\boldsymbol{v}_c = [-0.15, 0.15]^T$ m/s) and a PER equal to 15%. According to the TDMA protocol used, the update rate is of approximately 4 seconds for both the followers and the virtual leader. The relative 2D positions of the two followers in the formation was set so that $\boldsymbol{\eta}_Z - \boldsymbol{\eta}_G = [18, 1]^T$ m, where the subscripts

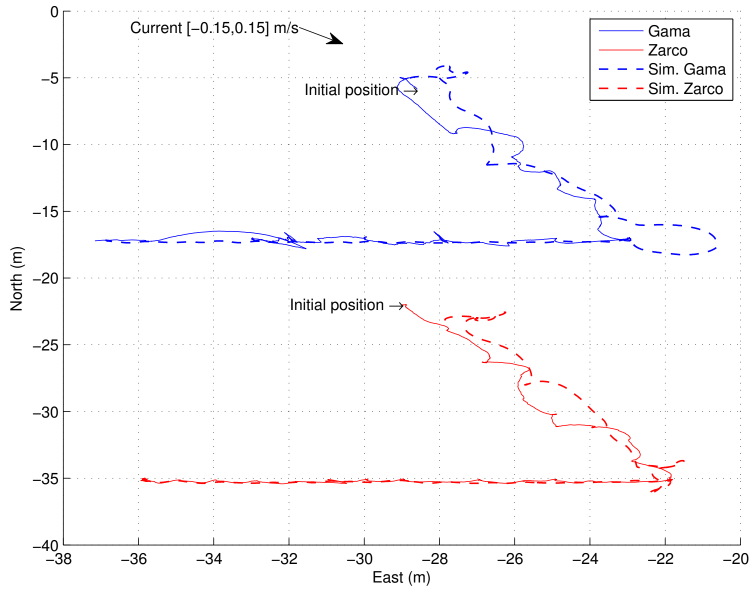

Figure 7: Real and simulated trajectories (PER = 15%; $\boldsymbol{v}_c = (-0.15, 0.15); \alpha = \frac{1}{5}; v_{max} = 1.0$).

Z and G stand for Zarco and Gama ASVs. Figure 7 shows the real and simulated trajectories of the vehicles. The former were obtained from vehicles logged data using the onboard GPS. Ideally, in both cases, the connection of the three waypoints should originate line segments. However, the non-holonomy of the vehicle, the low update rate and the drifts result in trajectories that deviate from the ideal ones, as described in Section V-B1. Without extensive parameter tuning, the formation performed the overall trajectory at a mean speed slightly above 0.2 m/s. Although slow, this mean speed is expected due to low data rate. As stated in Section III-B2, a faster motion can be obtained by relaxing the formation rigidity but at the cost of larger allowed errors.

Figure 8 depicts the metric absolute error given by $e = |\rho_{12}^* - \rho_{12}(t)|$ for the coordinated trajectories shown in Figure 7. As expected from Figure 7, the errors do not coincide for three main reasons: 1) The random nature of the simulation and of the packet loss imply different initial conditions and different reference evolutions; 2) The disturbances in real conditions are variable with variable orientation and some of them, such as wave induced forces, were neglected thus mismatching the initial assumption of constant disturbances induced by a constant drift; 3) biases in actuation and sensing, and corrections in position measurements modify the trajectories with respect to one another. We can observe the effect of

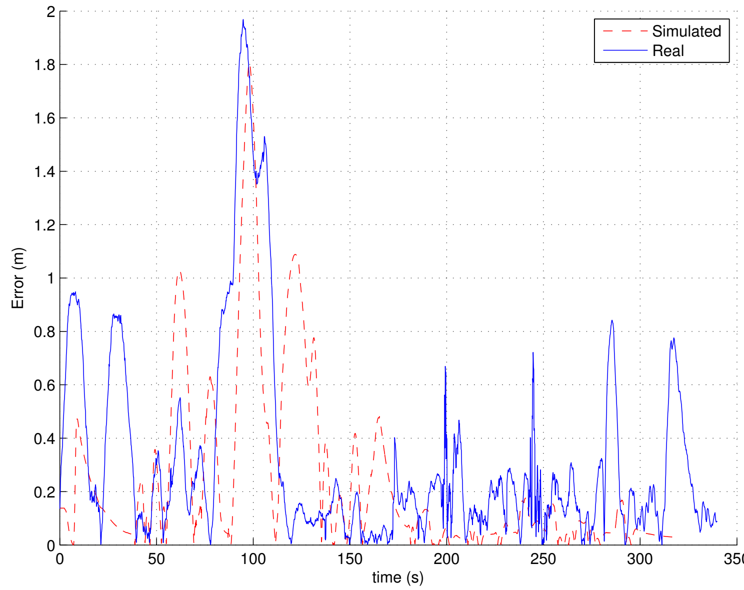

Figure 8: Relative position error.

the latter in abrupt changes in the error obtained from field trials ($t \approx 200$ s and $t \approx 245$ s), which are likely caused by well-known GPS fixes.

The pattern shown in both errors is similar and reinforces the validity of the simulators along with the results shown in the Figure 7. Mainly due to packet losses and low data rate, the vehicles are required to hold their positions at several points. We note that the largest peaks occur in the interval $t \in [0, 170]$ s, which is coincident with the time that the vehicles are moving over the first leg of the path, where they have to rotate large angles in order to face the drifts when they are required to hold their positions. Nevertheless, the error remains less than two meters during all the coordinated operation. The results of the metric M for the considered interval are 0.24 m for the simulation and 0.33 m for the field trials. These results are in agreement with the reasoning discussed above, implying a slightly greater error in the field trials. It is important to highlight that these values are below the achievable precision of 0.5 m announced by the GPS vendor. Simulations are used in this context not only to characterize the performance of the coordinated operation but are also used as an important way to tune the different parameters before field trials. In the present scenario, we consider that a metric M of the same magnitude of the GPS precision ensures a satisfactory performance of the overall system.

4) Extension to N vehicles: Results presented above were considering formation with only two vehicles. However, the solutions developed can be applied to larger formations. In what follows, we aim at demonstrating that our approach is extensible to as many vehicles as desired with ultimately no efforts in parameter tuning. A direct comparison with the results above is not useful, since a change in the number of vehicles affects the performance metrics, even if the geometry of the formation shows a similar dynamic behavior. Instead, we show the robustness of our implementation when subject to different environmental conditions.

In order to prove the flexibility and scalability of the presented algorithm and distributed framework, we have simulated the case of four vehicles following the same ideal formation trajectory displayed in Figure 5. A constant drift vector ($\boldsymbol{v}_c = [-0.4, 0]^T$ m/s) and a random constant packet error rate of 15% have been assumed. The resulting simulated trajectories of the four vehicles are shown in Figure 9. The error metric M in this case has a value equal to 0.66 m, even in the presence of stronger drifts, which increase the individual tracking errors.

One may notice the deformation of the formation: Due to stronger drifts, non-null packet error rate, a limited deformation is expected. Such a deformation grows with the packet error rate, with the communication delay and stronger drifts as they induce tracking errors. It is possible to limit the errors of the formation by properly tuning the parameters α_i, which are directly related to the formation geometry rigidity and also the allowed tracking error. For example, increasing α_i, the deviation of the i-th vehicle from its reference will penalize more the evolution of the formation. Therefore, formation geometry deviations can be as small as desired. However, this is not without consequences since the speed of the overall formation will be directly affected by tracking errors. Moreover, due to communication constraints, such as large delays and low bit rate, the evolution of the formation is even more penalized and the choice of α_i has also to be balanced according to these characteristics to meet the desired performances.

VI. Conclusion

In this paper we have described the implementation of an underwater acoustic network to support the operation of heterogeneous systems, including AUVs, ASVs, and moored devices. Our approach for such an implementation involved:

1) Identification of the requirements for the network architecture and communications tools;
2) A detailed analysis of the physical devices to be used;

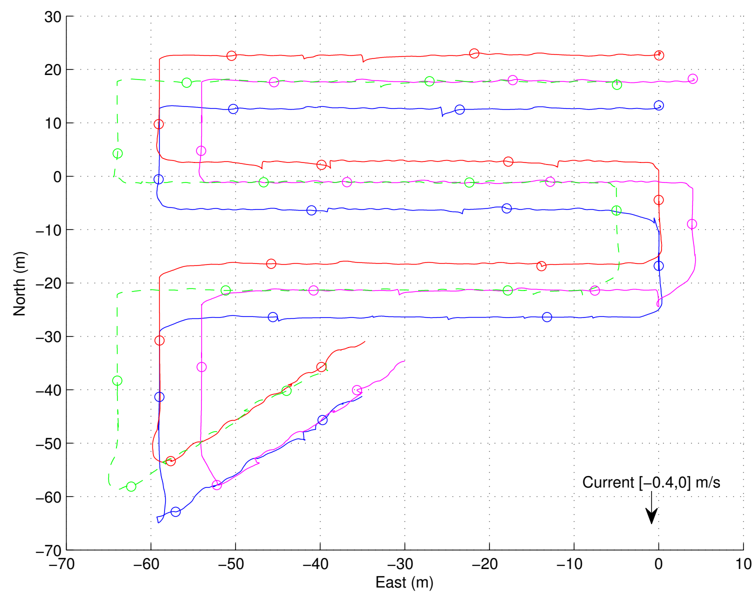

Figure 9: Simulated trajectories for four followers (PER = 15%; $\boldsymbol{v}_c = (-0.4, 0); \alpha = \frac{1}{5}; v_{max} = 1.5$).

3) The implementation of the complete system;
4) The definition of suitable performance metrics, to allow for quantitative evaluations;
5) An emulation infrastructure to allow for efficient testing of multiple scenarios;
6) Validation of a selected set of scenarios in field trials.

The SUNSET framework has been used to provide acoustic communication and networking capabilities to a set of robotic platforms and a distributed infrastructure has been developed to allow for the emulation of all the dynamic systems involved in a coordinated operation of marine vehicles. Such an infrastructure is paramount to perform extensive testing of all possible conditions found in operational scenarios, before actual deployment in the field.

As a practical example, we have addressed the effect of the PER on the performances of a vehicle formation by keeping a constant rate of information between nodes. We have demonstrated that communication failures and low update rates can be accommodated with the current implementation. We expect that increasing the update rate for both position feedback and references would improve the overall performance of our approach with respect to the metric defined and the formation evolution rate. The comparison between real data and the output of the distributed simulation environment demonstrated that realistic simulations can be obtained, thanks to the accuracy of the emulator modules. This motivates the future exploitation of the presented infrastructure, as the communication network is being expanded to include a much larger and varied number of systems, both fixed and mobile. Results of in field experiments also shows the effectiveness of the proposed solution which enables reliable effective cooperation and vehicle formation maintenance.

VII. Acknowledgments

This work was supported in part by the EU FP 7 STREP project CLAM "CoLlAborative EMbedded Networks for Submarine Surveillance". Bruno Ferreira is supported by the Portuguese Foundation for Science and Technology through the grant SFRH/BD/60522/2009. This work is financed by the ERDF European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project FCOMP - 01-0124-FEDER-022701.

References

[1] N. Baldo, F. Maguolo, M. Miozzo, M. Rossi, and M. Zorzi, "ns2-MIRACLE: A modular framework for multi-technology and cross-layer support in network simulator 2," in *Proceedings of the 2nd International Conference on Performance Evaluation Methodologies and Tools, ValueTools 2007*, Nantes, France, October 23–25 2007, pp. 1–8.

[2] O. Kebkal, K. Kebkal, and R. Bannasch, "Long-baseline hydro-acoustic positioning using d-mac communication protocol," in *Proceedings of MTS/IEEE OCEANS 2012*, Yeosu, Korea, May, 21–24 2012, pp. 1–7.

[3] N. A. Cruz and A. C. Matos, "The mares auv, a modular autonomous robot for environment sampling," in *Proceedings of MTS/IEEE OCEANS 2008*, 2008, pp. 1–6.

[4] R. Almeida, N. Cruz, and A. Matos, "Synchronized intelligent buoy network for underwater positioning," in *Proceedings of MTS/IEEE OCEANS 2010*, Seattle, WA, September, 20–23 2010, pp. 1–6.

[5] N. Cruz, L. Madureira, A. Matos, and F. Pereira, "A versatile acoustic beacon for navigation and remote tracking of multiple underwater vehicles," in *Proceedings of MTS/IEEE Conference and Exhibition OCEANS 2001*, vol. 3, Honolulu, HI, November, 5–8 2001, pp. 1829–1834.

[6] N. Cruz, A. Matos, S. Cunha, and S. Silva, "Zarco - an autonomous craft for underwater surveys," in *Proceedings of the 7th Geomatic Week, Barcelona, Spain, February 2007.*, 2007.

[7] A. Matos and N. Cruz, "Coordinated operation of autonomous underwater and surface vehicles," in *Proceedings of MTS/IEEE OCEANS 2007*, September 29–October 4 2007, pp. 1–6.

[8] B. M. Ferreira, A. C. Matos, and N. A. Cruz, "Modeling and control of trimares auv," in *Robotica 2012: 12th International Conference on Autonomous Robot Systems and Competitions*, E. Bicho, F. Ribeiro, and L. Louro, Eds. Guimarães: Universidade do Minho, 2012, pp. 57–62.

[9] B. Ferreira, J. Jouffroy, A. Matos, and N. Cruz, “Control and guidance of a hovering auv pitching up or down,” in *Proceedings of MTS/IEEE OCEANS 2012*, Hampton Roads, VA, October 2012, pp. 1–7.

[10] B. Ferreira, A. Matos, N. Cruz, and M. Pinto, “Modeling and control of the MARES autonomous underwater vehicle,” *Marine Technology Society Journal*, vol. 44, pp. 19–36, March/April 2010.

[11] N. A. Cruz, A. C. Matos, R. M. Almeida, B. M. Ferreira, and N. Abreu, “Trimares - a hybrid auv/rov for dam inspection,” in *Proceeding of MTS/IEEE OCEANS 2011*, Waikoloa, HI, September, 19–22 2011, pp. 1–7.

[12] M. Egerstedt and X. H., “Formation constrained multi-agent control,” *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 947–951, December 2001.

[13] O. Kebkal, K. Kebkal, and M. Komar, “Development of upper-layer protocols with S2CR acoustic modems emulator,” in *Proceedings of MTS/IEEE UComms 2012*, Sestri Levante, Italy, September, 12–14 2012, pp. 1–6.

[14] K. Kebkal, O. Kebkal, S. Yakovlev, and R. Bannasch, “Experimental performance of a hydro-acoustic usbl-aided lbl positioning and communication system,” *Proceedings of the IFAC Workshop Navigation, Guidance and Control of Underwater Vehicles (NGCUV)*, vol. 3, pp. 249–254, 2012.

[15] S. Basagni, C. Petrioli, R. Petroccia, and M. Stojanovic, “Optimized packet size selection in underwater WSN communications,” *IEEE Journal of Oceanic Engineering*, vol. 37, no. 3, pp. 321–337, July 2012.

[16] N. A. Cruz, B. M. Ferreira, A. C. Matos, C. Petrioli, R. Petroccia, and D. Spaccini, “Implementation of an underwater acoustic network using multiple heterogeneous vehicles,” in *Proceedings of MTS/IEEE OCEANS 2012*, Hampton Roads, Virginia, October, 14–19 2012.