Simulation of Underwater Communications with Colored Noise Approximation and Mobility

M. Barbeau*, S. Blouin†, G. Cervera‡, J. Garcia-Alfaro§ B. Hasannezhad* and E. Kranakis*

* School of Computer Science, Carleton University, K1S 5B6, Ottawa, Ontario, Canada
Email: {barbeau,kranakis}@scs.carleton.ca

† Defence R&D Canada - Atlantic, Dartmouth, NS, Canada B2Y 3Z7
Email: stephane.blouin@drdc-rddc.gc.ca

‡ Universidad Tecnológica Metropolitana, 97279, Merida, Yuc., Mexico
Email: gimer.cervera@utmetropolitana.edu.mx

§ Telecom SudParis, CNRS Samovar UMR 5157, Evry, France
Email: joaquin.garcia-alfaro@acm.org

***Abstract*—We study the software simulation of underwater physical communications, namely, underwater acoustic waves and (de)modulation of underwater acoustic digital data signals. We take into account the mobility of nodes, such as in underwater sensor networks. We also consider the integration with protocol layers above the physical layer, i.e., the link and network layers. In this context, mobility is relevant because there are underwater vehicles and environmental conditions causing displacements of sensors. Attenuation is sensitive to transmitter-receiver separation distances. Because of mobility, distances change. Our solution is based on the work of Borrowski (2010). The physical layer is modeled as Matlab functions. As a function of distance and frequency, our physical layer model takes into account attenuation and noise and their effects on a phase-shift keyed signal. We use OMNeT++ to model link and network layer protocols. The Matlab functions and OMNet++ models are linked together. While Matlab does particularly well with signal processing, OMNeT++ addresses better the protocols placed in the link layer and above.**

Keywords*—Underwater sensor network, underwater communications, acoustic waves, simulation.

I. Introduction

Underwater communication builds upon acoustic waves. With the current off-the-shelf acoustic modem technology, underwater communications concentrate in the 5 to 80 kilohertz range. For instance, EvoLogics' modems operate, according to the model, from 7 to 78 kilohertz [1]. Teledyne Benthos' modems operate in the bands 9-14, 16-21 and 22-27 kilohertz [2].

We focus on software emulation of underwater acoustic digital data signal propagation, modulation, and demodulation. We take into account mobility and integration of protocol layers. Mobility is relevant because there are underwater vehicles and environmental conditions causing displacements of nodes. Because of mobility, transmitter-receiver separation distances are variable. Attenuation is sensitive to distance.

We have developed a solution leveraging the work of Borrowski [3]. Two tools are used in co-simulation: Matlab and OMNeT++. Matlab deals particularly well with signal processing issues, while OMNeT++ can better address network protocol problems. The physical layer is modeled using Matlab functions. They are compiled into a dynamic library. They become available in the OMNeT++ environment. Our physical layer model takes into account attenuation and colored noise, i.e., frequency dependent noise level. Attenuation is frequency and propagation distance dependent. We model attenuation, and its effect, on a Phase-Shift Keying (PSK) modulated signal as a function of frequency and distance. Noise is frequency dependent. We implement the effect of noise on a PSK signal as a function of frequency. Matlab and OMNeT++ complement each other. While OMNeT++ is an excellent discrete event simulator with a modular programming interface that facilitates integration of new mobility models and network protocols, it lacks functionality to conduct the signal processing routines. Matlab offers such functionalities (modulation, demodulation of signals, creation of filters) that, can be exported and integrated as an OMNeT++ module. Previous works in the related literature heads in the same direction, e.g., Borowski [3], Zhang et al. [4]. Mobility is simulated using the meandering current model of Caruso et al. [5].

Related work is reviewed in Section II. Section III discusses the attenuation and noise models adopted for this work. Section IV describes our physical layer simulation methodology in details. The integration within the OMNeT++ environment is described in Section V. Section VI covers mobility simulation. We conclude with Section VII.

II. Related Work

Stojanovic and Preisig [6] reviewed conditions that impair underwater acoustic communications. They include attenuation, time-varying multi-path propagation, low propagation speed (1500 meters per second), noise, delay spreading and Doppler effect. Attenuation is captured by the Thorp model [7], [8], further discussed in Section III. Coates proposed models for various sources of ambient noise (turbulence, shipping, waves and thermal) [9], also further discussed in Section III. Attenuation depends on the transmitter-receiver separation distance. Both attenuation and noise are not flat across the acoustic wave spectrum and depend on frequency. To cope with these conditions, Stojanovic [10] proposed a model where the transmission power is determined as a function of frequency.

It is distributed according to the water-filling principle [11].

Harris and Zorzi [12] have developed a four-component underwater acoustic communication model for the network simulator NS2 [13]. The four components of the model are *propagation*, *channel*, *physical* and *modulation*. The propagation model follows the work of Stojanovic [10], with the following difference. The Signal-to-Noise Ratio (SNR) is calculated considering maximum attenuation, ambient noise and signal interference range. Uniform transmission power across the bandwidth is assumed. The SNR results from the difference between the transmission power and maximum attenuation and noise combination. Interference between stations is modeled. Xie et al. [14] have developed a simulation model in NS2, based on the model of Stojanovic [10]. Multi-path propagation and packet collisions are modeled.

EvoLogics S2CR is an underwater acoustic modem emulator [1], [15]. It does physical and data-link layer emulation. Other software emulation works include Nautilus, of Masiero et al. [16], a channel model with neighbor set calculation, collisions and propagation delays, of Cnar and Orencik [17], the World Ocean Simulation System (WOSS) library [18], a model developed by King [19] for OMNeT++ [20], and a model for the OPNET environment by Llor et al. [21], [22].

Mobility may be due to movable sensors or currents. Suzuki et al. have created an underwater vehicle simulator [23]. Detweiller et al. have developed underwater sensor network hardware [24]. The network includes mobile robots with a maximum speed of 1.5 meters per second. Arima et al. introduced an underwater glider simulation model [25]. Speed is assumed to be between 0.1 and 0.5 meter per second. Caruso et al. introduced a mobility model for underwater sensor nodes [5]. Causes of mobility are attributed to currents and vortices. The model is based on fluid dynamic theory. Zhou et al. proposed a scheme for localizing underwater sensors leveraging known mobility patterns [26]. Mobility is assumed to be caused by environmental conditions. The emphasis of the work is on predicting future mobility patterns based on previous measurements. In short, mobility support is relevant in underwater communications simulation.

III. Attenuation and Noise

Attenuation occurs during the underwater propagation of an acoustic wave. The main causes are conversion of acoustic energy into heat and geometrical spreading. Conversion of acoustic energy into heat is proportional to the signal frequency. Geometrical spreading component captures the fact that as sound wave travels away, the distance from the source increases. The surface covered covered by the waves expands. The sound energy intensity drops. These causes are represented in the Thorp model [6]–[8], [10],

$$A(d,f)_{dB} = k \cdot 10 \log d + d \cdot a(f). \tag{1}$$

Letter d is the transmitter-receiver separation distance (km). Letter k is a geometrical spreading coefficient, a value in the interval $[1,2]$. Symbol f is a frequency (kilohertz). Function $a(f)$ is a frequency-dependent absorption coefficient (dB/km). The path loss exponent depends on the geometry of propagation, which can be cylindrical (1), spherical (2) or *practical* (1.5). For a frequency f above a few hundreds hertz, the absorption coefficient is determined using the equation [8], [10]

$$a(f)_{dB/km} = \frac{0.11 \cdot f^2}{1+f^2} + \frac{44 \cdot f^2}{4100+f^2} + 2.75 \cdot 10^{-4} \cdot f^2 + 0.003.$$

There are two main types of noise affecting underwater acoustic communications: site-specific noise and ambient noise. Sources of site-specific noise are geographically dependent. They include noise made by breaking ice and sea creatures. Four sources of ambient noise have been identified: turbulence, shipping, waves, and thermal. Ambient noise is approximated using the equation [9][1]

$$N(f)_{\text{dB re } \mu\text{Pa per Hz}} = 50 - 18 \cdot \log f. \tag{2}$$

Frequency f is in kilohertz. Alternatively, equations do exist modeling turbulence, shipping, waves and thermal noise, individually [9]. The resulting ambient noise is the sum of all of them. This captures optimistic conditions, corresponding to a no wave-no wind situation. A more advanced noise model is given by Wenz [27].

IV. Simulation Methodology

Our model consists of a software modulator, a channel simulator, and a demodulator. We reuse the PSK modulator and demodulator originally developed by Borrowski [3]. The modulator and demodulator are implemented as Matlab functions `modulatePSK` and `demodulatePSK`. The formal parameters of function `modulatePSK` are `byteStream`, representing modulating data, `fs`, providing the sampling frequency (samples per second, sps), `fc`, corresponding to the carrier frequency (hertz), and `symbolsPerSecond`, specifying the symbol rate (baud). It returns an array of samples representing the PSK modulated signal.

The formal parameters of function `demodulatePSK` are `packetWaveform`, representing the samples of a modulated signal, `fs`, providing the sampling frequency (sps), `fc`, corresponding to the carrier frequency (hertz), and `symbolsPerSecond`, specifying the symbol rate (baud). It returns an array of bytes representing demodulated data.

We developed a new channel simulator that implements attenuation, according to Equation (1), and noise, according to Equation (2). The channel simulator is implemented as the Matlab function `simulateChannel`. The formal parameters are `signal`, corresponding to the samples of a modulated signal, `fs`, specifying the sampling rate (sps), `sps`, defining the symbol rate (baud), d, determining the distance (km), `lpf`, providing a low-pass filter used to model colored noise, and `P`, indicating the transmission power (μPa). The channel simulator returns an attenuated and noisy signal, according to the actual values assigned to the formal parameters.

A typical packet transmission script is as follows:

```
% assign values to fs, fc, sps, d, lpf and P
...
% build the packet to send
packet=...; l=length(packet)
% modulate
```

[1] The term *re μPa* stands for *relative to a reference pressure 1 micropascal*. The *pascal* is a unit of force per unit area, i.e., pressure. It is equal to one newton per square meter.

```
[s, nr] = modulatePSK(packet, fs, fc, sps);
% simulate the channel
[r nr]=simulateChannel(s, fs, sps, d, lpf, P);
% demodulate
receivedPacket=demodulatePSK(r, l, fs, fc, sps);
```

Values are assigned to the actual parameters `fs`, `fc`, `sps`, `d`, `lpf` and `P`. The transmitted packet is assembled and stored in variable `packet`, with length in `l`. The three key functions, of the implementation of the physical layer model, are successively invoked. Hereafter, we discus in details the implementation of the attenuation and noise simulations, i.e., function `simulateChannel`.

A. *Attenuation Simulation*

The input signal is amplified to the specified power level. Afterwards, it is attenuated.

A Fast Fourier Transform (FFT) is done to generate a frequency domain representation of the transmitted signal. Using distance d, the signal is attenuated according to Equation (1) in the frequency domain. After attenuation, the signal is converted back in the time domain using an inverse FFT.

The number of FFT bins n is the smallest power of two, greater than or equal to the number of samples in the transmitted signal. Let f_s be the sampling rate. According to the definition of the FFT, the bin frequencies are $\pm\frac{i \cdot f_s}{n}$ hertz, with $i = 0, 1, \ldots, n/2$. As a function of distance d, the frequency corresponding to index i is attenuated dividing by the factor

$$A\left(d \text{ km}, \frac{i \cdot f_s \text{ samples/second}}{n \text{ samples} \cdot 1000 \text{ Hz/KHz}}\right),$$

converted in linear form. This procedure captures the facts that attenuation is sensitive to distance and frequency, i.e., is non-flat across the band. The Thorp's model does not capture multi-path effects on an underwater acoustic signals. It is major communication impairment. The Bellhop ray-tracer models multipath effects [28]. Such a model will be integrated in future versions of our work.

B. *Noise Simulation*

To model the fact that noise is sensitive to frequency, we proceed in two steps. Noise is generated across the band using the maximum noise density. This maximum noise density level is at the lower edge of the band. The noise is fed through a low-pass filter to capture sensitivity to frequency. Afterward, noise is added to the signal.

According to Equation (2), highest noise power density is 50 dB re μPa per Hz. The bandwidth of the input signal corresponds to two times the sampling rate f_s. The initial noise power N results from the product of the highest noise power density, in linear form, and bandwidth. The concept of normalized frequency is used in the sequel. Let f be the frequency of a signal, in hertz, and f_s be the sampling frequency, in sps. The *normalized frequency* refers to the ratio

$$\omega = \frac{2f}{f_s} \text{ half-cycles/sample.}$$

Assuming a sampling frequency f_s of 100 kilosps, Figure 1 plots the power of the noise density as a function of the normalized frequency (i.e., the frequency actually goes from zero to 100 kilohertz). The underwater acoustic noise is simulated using an Additive White Gaussian Noise (AWGN) generator filtered by a low-pass filter with a profile similar to the profile of the underwater acoustic noise. The profile of such a filter is shown in Figure 2. It is a Butterworth Infinite Impulse Response (IIR) filter with normalized pass band frequency 0.05 half-cyle/sample, normalized stop band frequency 0.9 half-cyle/sample and attenuation in stop band 30 dB.

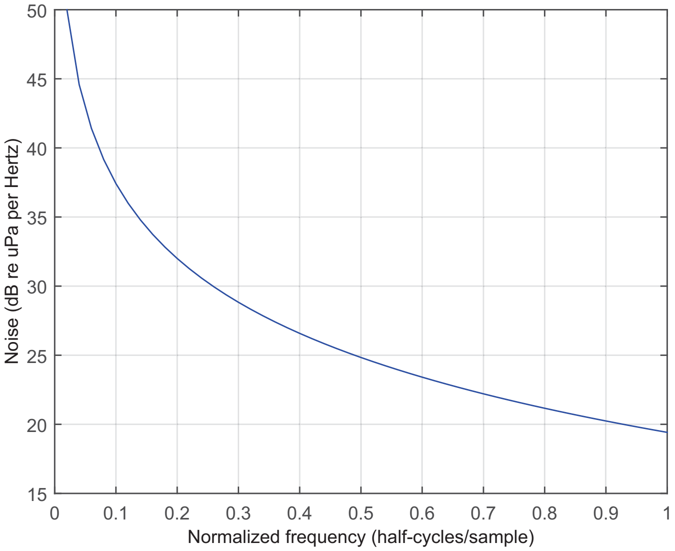


Fig. 1. Noise power density versus normalized frequency, in half-cycles/sample (f_s is 100 kilosps).

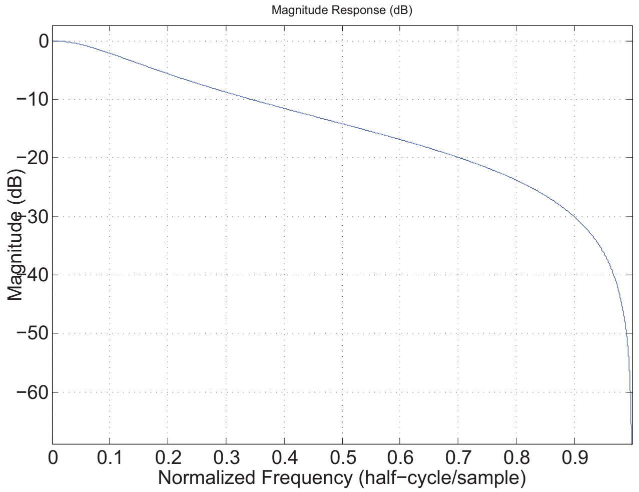


Fig. 2. Profile of low-pass filter (filter response, in dB, as a function of normalized frequency, in half-cycles/sample).

Figure 3 shows an example PSK signal of one kilobaud with center frequency in the very low frequency band, before passing through the simulated channel. The Root Mean Square (RMS) power of the signal is 80 dB re μPa. Figure 4 shows the signal after passing through the simulated channel, with a distance of one km. The upper part shows the original signal

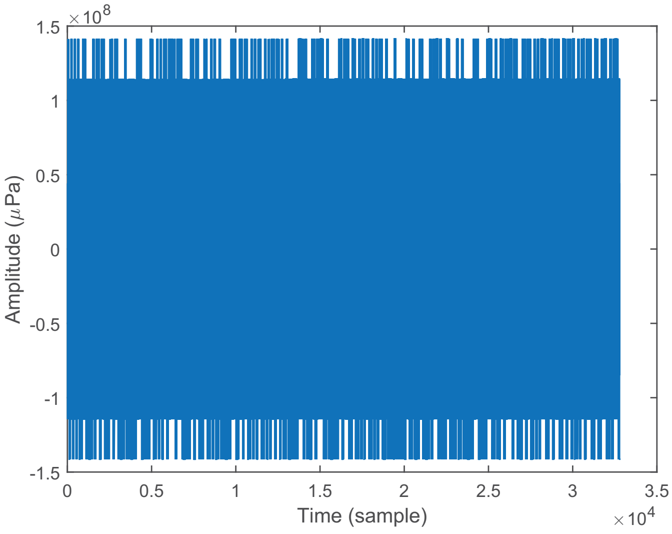


Fig. 3. Example input of simulated channel, signal amplitude (μPa) as a function of time (sample).

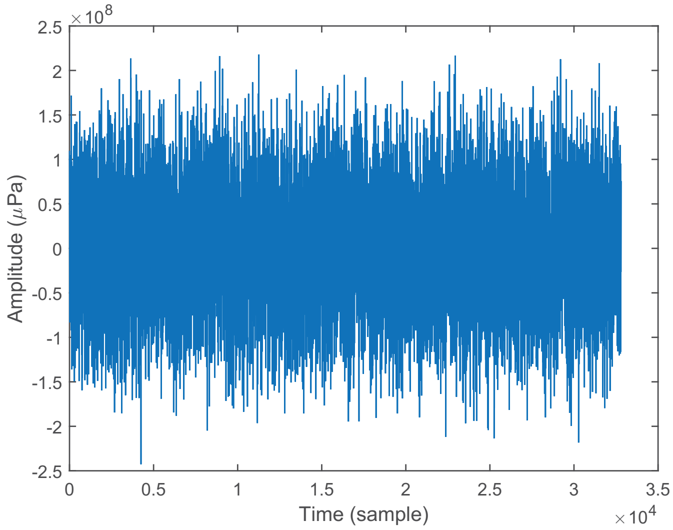


Fig. 4. Example output of simulated channel, with a distance of one km (signal amplitude (μPa) as a function of time (sample).

amplitude (μPa) as a function of time (sample) at the input of the channel simulator.

The distance is one kilometer. The lower part shows the signal at the output of the channel simulator.

V. Integration in OMNeT++

Our physical layer model is integrated into the OMNeT++ framework [20], [29]. We leverage the shared libraries of OMNeT++/INET [30], OMNeT++/MiXiM [31] and Matlab. The OMNeT++ framework aims for the development of discrete event simulations. In OMNeT++, a simulation model is described in the C++ language [32] and a network topology specification language called NED (NEtwork Description) [33]. In a nutshell, OMNeT++ consists of simulation modules that communicate with each other through message passing (either via OMNeT++ gates, or through direct messages). The use of add-ons, e.g., via shared libraries, makes possible complementing OMNeT++ code with functionality from other simulation engines, e.g., by using dynamic or static Matlab shared libraries and linked to the final OMNeT++ simulation code, as suggested in works like [3], [4], [34].

The use of OMNeT++/INET provides existing modules for protocols above the link layer (e.g., HTTP, DHCP, UDP, TCP/IP), as well as support for dynamic connection management and routing (e.g., link state routing protocols such as OLSR [35] and BATMAN [36]). In terms of mobility, OMNeT++/INET provides several types of mobility models, such as trace-based models (e.g., to replay previously recorded traces in either real or simulated environments) or synthetic models (e.g., random and mass mobility models). More information about extending OMNeT++/INET to provide new mobility models is available in [37], [38]. Our simulation of mobility is further dicussed in Section VI.

The use of OMNeT++/MiXiM provides existing modules for the link and physical layers. Moreover, OMNeT++/MiXiM provides additional communication structures to connect simulated entities via four new module types: (1) the world utility module, (2) the connection manager, (3) the playground, and (4) network interface card (NIC) modules. The world utility module stores global information about the simulation. The connection manager provides the means for setting up the communications between the simulation nodes and region where nodes can be placed (the playground). The NIC modules equip the nodes with the link and physical layers. The NIC modules communicate via message passing. It attempts to be a close representation of the functionality expected by the media access control (MAC) and physical hardware transmission of the simulated technologies.

The work presented in Section IV is combined with the OMNeT++/MiXiM existing models for wireless sensor networks, by extending those four aforementioned module types into a new communication structure according to the acoustic model. Hence, we rely on the OMNeT++/MiXiM functionality up to the link layer, and on the Matlab functionality (cf. the Matlab functions reported in Section IV) for the physical layer. As a result, our simulation model can use modules implementing different NICs, such as terrestrial and underwater NICs. It allows, as well, the use of mobility and routing.

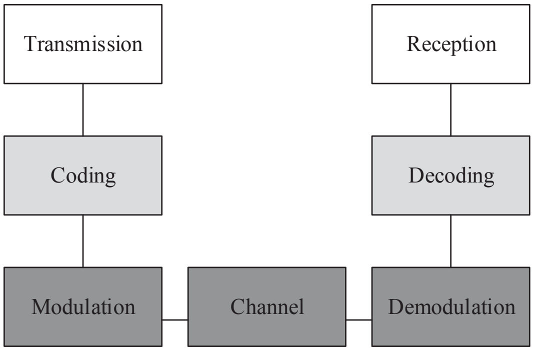


Fig. 5. Simulation model architecture. Three protocol layers are shown. From top to bottom, they correspond to the network, link and physical. Clear and light-gray rectangles are protocols implemented in OMNeT++. Gray rectangles are protocols implemented in Matlab.

Figure 5 shows our simulation mode architecture. There are three protocol layers: physical, link and network. The modules of OMNeT++/INET are represented in white. The OMNeT++/MiXiM modules are in light gray. Matlab functions are in dark grey. Matlab-OMNeT++ code integration is discussed further in an Appendix [39].

VI. Mobility Simulation

In addition to the physical layer model, we integrated in OMNeT++ the meandering current mobility model by Caruso et al. [5]. The necessary code is available as an open source library in C [40]. It provides a time-dependent, one-degree-of-freedom Hamiltonian, stream mobility function. In our simulation, the physical positions of sensor nodes are computed using the Caruso et al. library. The sensors are then managed by OMNeT++ base mobility instances. The sensor positions are updated at each simulation time step, in order to move the nodes according to the velocity field defined by the stream function included in the library. The initial positions of the nodes are defined using the OMNeT++ NED language, for designing network topologies [33].

Figure 6 shows a sample OMNeT++ simulation combining meandering current mobility and our physical layer model. Figures (a), (b) and (c) depict position progression at three different time scale events of eighty sensors randomly dispersed on the axis of a meandering current. Figures (d), (e) and (f) plot a bidimensional communication map. Every square represents the associated bit error ratio during the data transmission of any two sensors at a given time step. At each time step, a group of eighty sensors broadcast 320-bit beacons. They compute the bit error ratio associated with the reception of every beacon from any sensor to any other sensor. The results are depicted as eighty-by-eighty sensor color maps. Three error classes (i.e., gray levels) are assumed. White squares represent data transfers with a bit error ratio below 33%. Gray squares represent bit error ratios between 33% and 77%. Black squares represent bit error ratios higher than 77%. Notice that the simulation does not filter self-communication measures. Therefore the diagonal of white squares represent a data exchange from any sensor to itself with a 0% bit error ratio. The error rates in the communications maps, Figures (d), (e) and (f), are consistent with the sensor position progression, Figures (a), (b) and (c). Indeed, as time progresses, the meandering jet current disperses sensors. Sensor-to-sensor separation distances grow. According to the model of Section IV, attenuation grows as well as relative noise power. Quality of links drops, while the error rates increase.

VII. Conclusions

We have developed an underwater acoustic communication physical layer simulation. It consists of a modulator, a channel, and a demodulator. The system architecture, modulator and demodulator are borrowed from the work of Borrowski [3]. The channel simulator models attenuation, as a function of distance and frequency, and noise, according to frequency. Distance is a variable that changes due to the mobility of sensor nodes and curents. The physical layer is simulated as Matlab functions. The Matlab functions are compiled into a dynamic library. It is loaded in the OMNeT++ environment, where link and network layer protocols are simulated, further discussed in a companion paper [41].

VIII. Acknowledgments

We would like to thank Brian S. Borowski for sharing his OMNeT++ and Matlab code with us. We acknowledge financial support from Public Works and Government Services Canada (PWGSC contract # W7707-145688/001/HAL), Natural Sciences and Engineering Research Council of Canada (NSERC), Spanish Ministry of Science (project TIN2011-27076-C03-02 CO-PRIVACY) and Innovation and Ministry of Education of Mexico (PROMEP). Thank you to Jie Huang for helpful comments on the work.

References

[1] EvoLogics, “Underwater acoustic modems,” EvoLogics GmbH, Ackerstrasse 76, 13355 Berlin, Germany, Product Information Guide, 2013.

[2] Teledyne Benthos, “Acoustic modems,” November 2013. [Online]. Available: http://www.benthos.com/index.php/product_dashboard/acoustic_modems

[3] B. S. Borowski, “Application of channel estimation to underwater acoustic communicaton,” Ph.D. dissertation, Stevens Institute of Technology, Castle Point on Hudson, Hoboken NJ, 2010.

[4] Z. Zhang, Z. Lu, Q. Chen, X. Yan, and L.-R. Zheng, “COSMO: CO-simulation with MATLAB and OMNeT++ for indoor wireless networks,” in *2010 IEEE Global Telecommunications Conference (GLOBECOM 2010)*. IEEE, 2010, pp. 1–6.

[5] A. Caruso, F. Paparella, L. Vieira, M. Erol, and M. Gerla, “The meandering current mobility model and its impact on underwater mobile sensor networks,” in *The 27th IEEE Conference on Computer Communications (INFOCOM)*, April 2008, pp. 771–779.

[6] M. Stojanovic and J. Preisig, “Underwater acoustic communication channels: Propagation models and statistical characterization,” *IEEE Communications Magazine*, vol. 47, no. 1, pp. 84–89, 2009.

[7] W. Thorp, “Deep ocean sound attenuation in the sub and low kilocycle per second region,” *The Journal of the Acoustical Society of America*, vol. 38, no. 4, pp. 648–654, 1965.

[8] ——, “Analytic description of the low frequency attenuation coefficient,” *The Journal of the Acoustical Society of America*, vol. 42, p. 270, 1967.

[9] R.Coates, *Underwater Acoustic Systems*. New-York: Wiley, 1989.

[10] M. Stojanovic, “On the relationship between capacity and distance in an underwater acoustic communication channel,” *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 11, no. 4, pp. 34–43, Oct. 2007. [Online]. Available: http://doi.acm.org/10.1145/1347364.1347373

[11] J. Proakis, *Digital Communications*. New York, NY, USA: Mc-Graw Hill, 2001.

[12] A. F. Harris and M. Zorzi, “Modeling the underwater acoustic channel in NS2,” in *Proceedings of the 2nd International Conference on Performance Evaluation Methodologies and Tools*, ser. ValueTools ’07, 2007, pp. 18:1–18:8.

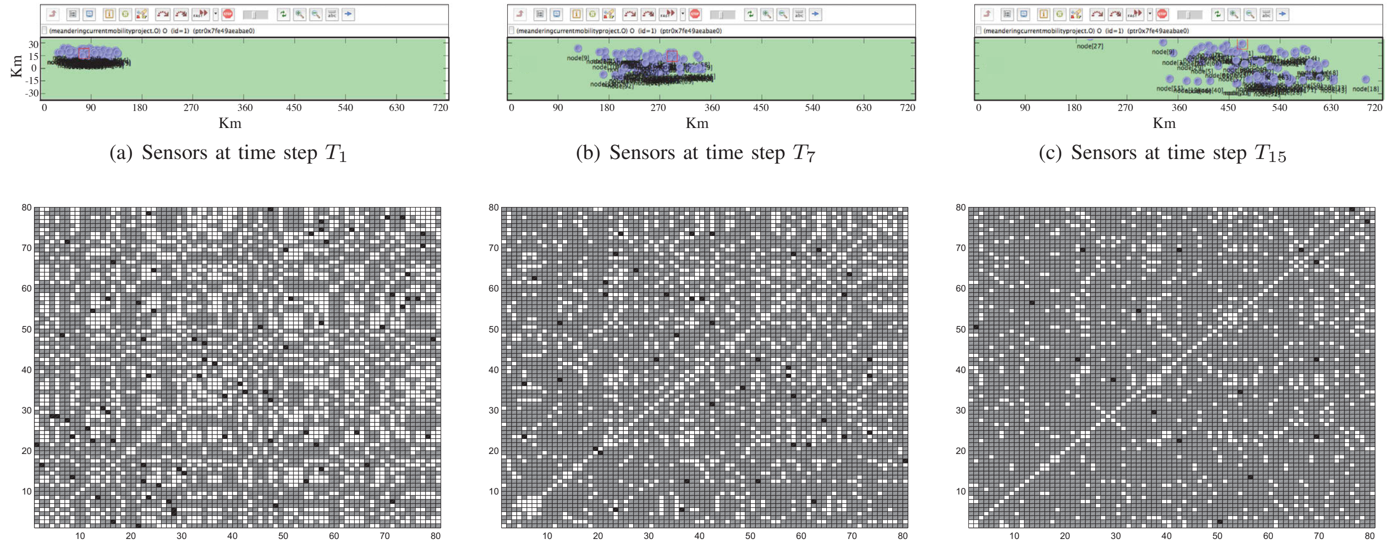
[13] S. McCanne, S. Floyd, K. Fall, K. Varadhan *et al.*, “The network simulator: NS-2,” Software package retrieved from http://www.isi.edu/nsnam/ns/, 1997.

[14] P. Xie, Z. Zhou, Z. Peng, H. Yan, T. Hu, J.-H. Cui, Z. Shi, Y. Fei, and S. Zhou, “Aqua-Sim: An NS-2 based simulator for underwater sensor networks,” in *MTS/IEEE Biloxi - Marine Technology for Our Future: Global and Local Challenges (OCEANS)*, 2009, pp. 1–7.

[15] EvoLogics, “S2C hydroacoustic modems,” EvoLogics GmbH, Ackerstrasse 76, 13355 Berlin, Germany, Product Information Guide, 2013.

[16] R. Masiero, S. Azad, F. Favaro, M. Petrani, G. Toso, F. Guerra, P. Casari, and M. Zorzi, “DESERT underwater: An NS-Miracle-based framework to design, simulate, emulate and realize test-beds for underwater network protocols,” in *Yeosu (OCEANS)*, 2012, pp. 1–10.

[17] T. Cnar and M. Orencik, “An underwater acoustic channel model using ray tracing in NS-2,” in *Wireless Days (WD), 2009 2nd IFIP*, 2009, pp. 1–6.



(a) Sensors at time step T_1

(b) Sensors at time step T_7

(c) Sensors at time step T_{15}

(d) Bit error results at time step T_1

(e) Bit error results at time step T_7

(f) Bit error results at time step T_{15}

Fig. 6. Sample OMNeT++ mobility simulation. Subfigures (a), (b) and (c) depict the position progression at three different time scale events of eighty sensors randomly dispersed on the axis of a meandering current. Subfigures (d), (e) and (f) plot a bidimensional communication map, in which every square represents the associated bit error ratio during the data transmission of any two sensors at a given time step. White squares represent data transfers with a bit error ratio below 33%. Gray squares represent bit error ratios between 33% and 77%. Black squares represent bit error ratios higher than 77%.

[18] F. Guerra, "World ocean simulation system (WOSS) library," Regents of Patavina Technologies, Tech. Rep., February 2013.

[19] T. King, "A simulator for marine wireless sensor networks," Ph.D. dissertation, Department of Computing Sciences, Texas A & M University, Corpus Christi, Texas, Spring 2011, graduate project.

[20] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," in *1st International conference on Simulation tools and techniques for communications, networks and systems & workshops (Simutools)*, 2008.

[21] J. Llor, E. Torres, P. Garrido, and M. P. Malumbres, "Analyzing the behavior of acoustic link models in underwater wireless sensor networks," in *Proceedings of the 4th ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks*, ser. PM2HW2N '09. New York, NY, USA: ACM, 2009, pp. 9–16. [Online]. Available: http://doi.acm.org/10.1145/1641913.1641915

[22] J. Llor, M. P. Malumbres, and P. Garrido, "Performance evaluation of underwater wireless sensor networks with opnet," in *Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques*, ser. SIMUTools '11. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2011, pp. 19–26. [Online]. Available: http://dl.acm.org/citation.cfm?id=2151054.2151058

[23] H. Suzuki, N. Kato, T. Katayama, and Y. Fukui, "Motion simulation of an underwater vehicle with mechanical pectoral fins using a cfd-based motion simulator," in *Symposium on Underwater Technology and Workshop on Scientific Use of Submarine Cables and Related Technologies*, April 2007, pp. 384–390.

[24] C. Detweiller, I. Vasilescu, and D. Rus, "An underwater sensor network with dual communications, sensing, and mobility," in *OCEANS 2007 - Europe*, June 2007, pp. 1–6.

[25] M. Arima, N. Ichihashi, and Y. Miwa, "Modelling and motion simulation of an underwater glider with independently controllable main wings," in *EUROPE OCEANS 2009*, May 2009, pp. 1–6.

[26] Z. Zhou, Z. Peng, J.-H. Cui, Z. Shi, and A. Bagtzoglou, "Scalable localization with mobility prediction for underwater sensor networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 3, pp. 335–348, March 2011.

[27] G. M. Wenz, "Acoustic ambient noise in the ocean: Spectra and sources," *The Journal of the Acoustical Society of America*, vol. 34, no. 12, 1962.

[28] O. C. Rodriguez, "General description of the BELLHOP ray tracing program," June 2008. [Online]. Available: http://oalib.hlsresearch.com/Rays/GeneralDescription.pdf

[29] The OMNeT++ Network Simulation Framework, Last Access: 2014, available at http://www.omnetpp.org/.

[30] The OMNeT++/INET Framework, Last Access: 2014, available at http://inet.omnetpp.org/.

[31] The OMNeT++/MiXiM Modeling Framework, Last Access: 2014, available at http://mixim.sourceforge.net/.

[32] J. Heijmans, A. Paalvast, and R. V. der Leij, "OMNeT++ Discrete Event Simulation System — User Manual," Last Access: 2014, available at http://www.ewh.ieee.org/soc/es/Nov1999/18/manual/usman.htm.

[33] ——, "NED language overview," Last Access: 2014, available at http://www.ewh.ieee.org/soc/es/Nov1999/18/ned.htm.

[34] C. Mayer and T. Gamer, "Integrating real world applications into OMNeT++," Institute of Telematics, University of Karlsruhe, Karlsruhe, Germany, Tech. Rep. TM-2008-2, Tech. Rep., 2008.

[35] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," RFC 3626 (Experimental), Internet Engineering Task Force, Oct. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3626.txt

[36] Freifunk, "The B.A.T.M.A.N. routing protocol (Better Approach To Mobile Adhoc Networking)," Last Access: 2014, available at http://www.open-mesh.org/projects/open-mesh/wiki.

[37] M. Löbbers and D. Willkomm, "A mobility framework for OMNeT++ — User Manual - Mobility model implementation for more realistic simulations," Last Access: 2014, available at http://pm-awn.googlecode.com/files/mf-manual.pdf.

[38] G.-J. Herbiet, "Mobility model implementation for more realistic simulations," Last Access: 2014, available at http://herbiet.gforge.uni.lu/files/omnet-lab04.pdf.

[39] M. Barbeau, S. Blouin, G. Cervera, J. Garcia-Alfaro, and E. Kranakis, "Appendix to simulation of mobile underwater communications," September 2014. [Online]. Available: http://people.scs.carleton.ca/~barbeau/Publications/2015/CCECE_2015_Appendix.pdf

[40] A. Caruso, F. Paparella, L. Vieira, M. Erol, and M. Gerla, "Simple Jet Meandering Model (SMM)," Last Access: 2014, available at https://github.com/antoniocaruso/smm.

[41] M. Barbeau, S. Blouin, G. Cervera, J. Garcia-Alfaro, and E. Kranakis, "Location-free link state routing protocol for underwater acoustic sensor networks," in *Submitted for publication*, 2014.