2022 Sixth Underwater Communications and Networking Conference (UComms) | 978-1-6654-7461-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/UComms56954.2022.9905700

A First-of-its-kind Low Size, Weight and Power Run-Time Reconfigurable Underwater Modem

Jared Hermans, *Student Member, IEEE*, George Sklivanitis, *Member, IEEE*
and Dimitris A. Pados, *Senior Member, IEEE*
FAU Center for Connected Autonomy and AI
Florida Atlantic University Boca Raton, FL 33431, USA
E-mail: {jhermans2018, gsklivanitis, dpados}@fau.edu

***Abstract*—Low size, weight, power and cost (SWaP-C), flexible programmability and rapid run-time reconfigurability are desired features for the design of adaptive, underwater wireless communication modems for micro autonomous underwater vehicle (μAUV) swarms. In this paper, we present a first generation software and hardware design of a new class of low SWaP-C underwater modems that can accommodate wideband acoustic front-ends and achieve μs-level reconfigurability during run-time. We investigate the benefits of dynamically mapping certain signal processing operations of the acoustic communication stack to the Field Programmable Gate Array (FPGA) and others to the embedded processing system of the modem's System-on-Chip and demonstrate data rates up to 2000 bps in ranges up to 50 m with the implementation of custom programmable logic. We test and evaluate power consumption, wireless link performance and speed of reconfiguration for two communications schemes: Binary Frequency Shift Keying (FSK) and Fast Frequency Hopping FSK both with simulations and field experiments.**

***Index Terms*—Underwater IoT, System on Chip, Reconfigurable Computing and Communications.**

I. Introduction

Fleets of intelligently coordinated macro/micro autonomous underwater vehicles (AUVs) can collect more data than a single ship/vehicle and offer the opportunity to study data from space-time scales not previously possible [1]–[4]. Furthermore, a group of tens/hundreds of cooperative, low-cost and flexible robots that can be easily deployed from a boat and/or aerial vehicles can offer performance benefits, for example in terms of mission completion time. Acoustic waves are the preferred signals for wirelessly coordinating such large fleets of mobile nodes. Unfortunately, commercially available underwater acoustic (UW-A) modems are: (i) large in size to fit small-size AUVs; (ii) prohibitively expensive for large-scale deployments; (iii) typically closed source, thus hampering their programmability and interoperability with other sensors and therefore their application in research [5].

Over the past few years, there has been a wide range of low-cost and low-power experimental modem prototypes developed by both academics and the industry. Seatrac [6], [7] offers a miniature acoustic modem solution that operates from 24 to 32 kHz and offers data rates from $100-1000$ bps using M-ary phase shift keying (MPSK) modulation and error correction coding. SeaModem [8] is also a low-cost and low-power UW-A modem that operates in similar frequencies using selectable 2-4-8 FSK modulation tones and error detection and correction algorithms. Seanet [9]–[11] provides acoustic energy harvesting and high data rate solutions of several kbit/s, however the modems are bulky and power-hungry due to the use of off-the-shelf development kits. Ahoi [5] is an open-source acoustic modem built around a micro-controller for μAUV operations that operates from 25 up to 87.5 kHz and can achieve data rates up to 4700 bps and supports bandwidths up to 37.5 kHz. The list of UW-A modems above –which is by no-means exhaustive– presents limitations and trade-offs in size, weight, power and cost (SWaP) optimization and run-time reconfigurability to accommodate truly environment-aware adaptive UW-A communications.

In this work, we present the software and hardware architecture of the first generation of a run-time reconfigurable SWaP-C optimized UW-A wireless modem that is capable of rapidly (at the μs level) adapting during run-time its acoustic communication stack, such as operating frequency, bandwidth, and physical/medium-access-control-layer modulation to mitigate multipath-induced and/or avoid other sources of environmental interference [12]–[14]. The modem demonstrates data rates up to 2000 bps in ranges up to 50 m. Maximum operational bandwidth can reach up to 75 kHz. In the next sections, we present the board design, and custom programmable logic that we developed and evaluated in a laboratory water tank, a swimming pool, and very shallow and noisy harbor environments, considering testing distances (from 2 m to 50 m) that would be realistic for micro-AUV operations.

II. Transceiver Board Design

The first generation (GEN1) UW-A modem is built around a custom PCB developed in-house at the Center for Connected Autonomy and AI at Florida Atlantic University [15] and is shown in Fig. 1. The board is based on the Xilinx Zynq-7000 FPGA System-on-Chip (SoC), comprising both a processing system (PS), which contains a dual-core application processor unit (APU), interconnects and many peripherals such as UART, SPI, memory controller, etc., and a programmable logic (PL) block, that contains look-up tables (LUTs), registers, block RAM, and DSP blocks [16] under the same fabric. This architecture allows for a run-time re-configurable and re-programmable wireless system that can accommodate parallel processing of large data bandwidths from multiple front-ends with deterministic latency for next-generation underwater IoT.

The hydrophone used in GEN1 is a Teledyne RESON TC4013 with receiving sensitivity of -211 [dB re 1V/µPa at 1 m] that is relatively flat over the operational frequency range (1 Hz to 170 kHz) and transmitting sensitivity of 130 [dB re 1µPa/V at 1 m] at 100 kHz, omnidirectional horizontal and 270° vertical directivity patterns. Following the hydrophone to board connections shown in Fig. 2, the first component is the secondary winding of a parallel tuned transformer. A tap on the secondary winding follows the RX demodulation signal chain, while the primary winding follows the TX modulation

This work is supported in part by the National Science Foundation under Grant No. CNS 1753406, the Schmidt Family Foundation, and a seed funding award from the FAU College of Engineering and Computer Science and the FAU Institute for Sensing and Embedded Networks Systems Engineering.

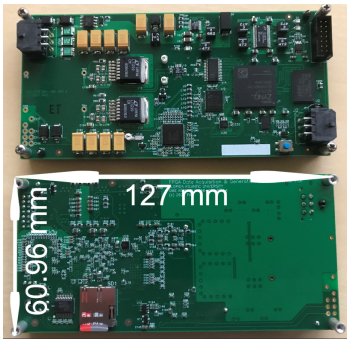

Fig. 1. Low SWaP-C Zynq-SoC-based transceiver board (127×60.96 mm).

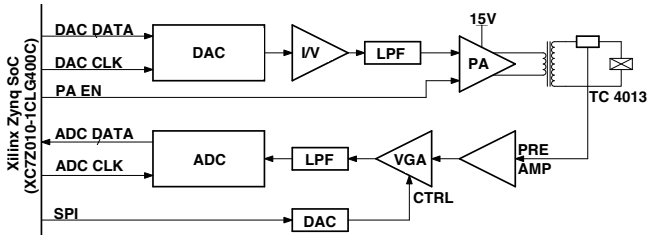

Fig. 2. UW-A modem analog front-end.

signal chain. From the secondary winding tap, there is a 19 dB pre-amplifier, a variable gain amplifier (VGA) (controlled via SPI), a 1 MHz low-pass filter and a 14-bit parallel 40 Msps analog-to-digital converter (ADC). After the 14-bit 10 Msps digital-to-analog converter (DAC), there is an I/V converter, 1 MHz low-pass filter, and a 5 W power amplifier to the primary winding of the transformer.

III. Programmable Hardware Design

The PL contains all signal processing hardware components necessary to operate either as TX or RX and a controlling Linux user space software program, on the PS, implements a high level protocol. Due to the low frequency nature of underwater acoustic channels, all signal processing is done via direct sampling. In the PL, standard Xilinx IP is used in combination with custom IP developed in hardware description language (HDL).

As the SoC was built around a custom platform, an embedded Linux system was built, configured and compiled, from scratch. Embedded Linux consists of a first-stage boot loader, a second-stage boot loader, a device tree, a kernel image and a root file system – all together referred to as the board support package (BSP) [17]. Great effort was spent on configuring the Linux kernel and root file system to include only necessary drivers and packages to keep the system as lightweight as possible. A shared repository [15] provides all custom software, hardware, BSP, and scripts pertaining to the modem design.

A. Transmitter Architecture

Transmitter functionality is implemented almost entirely in the PS. Creating the modulating waveform in software in the PS instead of the PL allows for greater flexibility in waveform design, at the expense of performance. The PS is responsible for encoding the digital data –received over UART– into a waveform $\Phi(t)$. The PS generates the digital samples at 1/10th of the sample rate of the DAC (due to PS-PL communication performance limitations): $\Phi[L \cdot n \cdot T_s]$, where $\mathrm{n} = \{0, 1, ...\mathrm{M}\}$, $\mathrm{L} = 10$, $\mathrm{T_s} = 1/\mathrm{F_s}$. As a result a $\times 10$ interpolation FIR filter in the PL is required. The first stage of interpolation is zero packing of $L = 10$ which adds 9 zero samples in between every sample. The spectrum of the zero-packed signal contains $\frac{L}{2} - 1$ repetitions of the original spectrum every $L \cdot \omega_b$, where ω_b is the bandwidth of the original spectrum and $0.5 \cdot L \cdot \omega_s/2$ is the new Nyquist rate. The signal is then reconstructed with the new sample rate $L \cdot F_s$ by applying a low-pass filter with cutoff frequency $\omega_b \leq \omega_c \leq \omega_b + (0.5 \cdot \omega_f/L - \omega_b)$.

Waveform samples are sent from the PS to the PL via direct memory access (DMA). The DMA is configured in direct register mode so no Linux kernel driver is necessary to operate it –control and status registers within the DMA engine are memory mapped directly from the Linux user-space application. The device tree was also configured to allocate a large block of physically contiguous memory in which the buffers accessed by the DMA reside and where the kernel has no access, too.

GEN1 demonstrates the flexibility in waveform design in the PS with the implementation of two communications schemes: 1) Binary Freqeuncy Shift Keying (B-FSK), and Fast Frequency Hopping Frequency Shift Keying (FH-FSK) as these are described in Section IV.

B. Receiver Architecture

Custom IP was created to convert the digital data received from the ADC into a standardized data stream [18]. After that, a $\times 40$ decimator is implemented to limit the number of samples coming from the ADC. This converts the data rate from 40 Msps to 1 Msps. Following clock conversion, the data stream is then digitally spit into one data stream that follows the path of demodulation and another data stream that is sent to the PS via DMA to be saved to the SD card. The latter can be used to estimate SNR among other things.

1) Coefficient Re-Loadable FIR Filter: GEN1 considers the implementation of two 512-tap FIR correlators –using Xilinx FIR Compiler IP [19]. The input to both correlators is the data stream of the downsampled received data $\Phi \downarrow^{40} [n]$. We denote the matched filtered output of each FIR as $x[n]$. Each FIR has the capability of updating its coefficients from the PS via DMA and a custom IP as shown in Fig. 3.

Hardware simulation shows exact timing to reload both 512 tap filters and start filtering with the new coefficient set. Both filters are re-loaded in parallel. Each coefficient takes

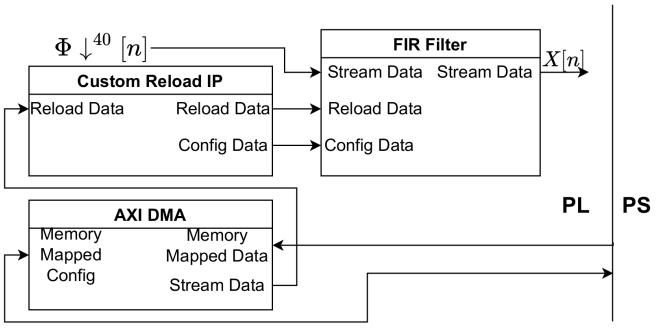

Fig. 3. Custom IP for FIR coefficient reloading.

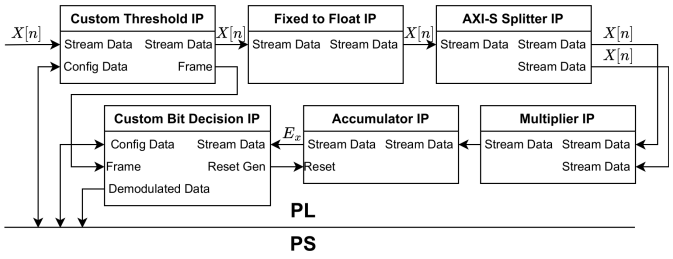

Fig. 4. Custom IP for frame synchronization, energy detector and bit decision.

a single clock cycle to send. Due to the optimization and implementation of the FIR filter within the DSP columns, there are actually 516 coefficients implemented and require a specific out-of-order reload sequence [19]. A single configuration packet is also required, which is handled by the custom IP and takes a single clock cycle. With 516 clock cycles for re-loading coefficients, 1 clock cycle for configuration and 1 clock cycle to update new coefficients into the filter it takes exactly 5.18 μs to re-load both filters. In a field test, where the PS (runs Linux OS) sends new coefficients, it was measured to take less than 6 μs.

These results provide the hardware framework to support real-time, software-controlled modulation adaptation between B-FSK, FH-FSK in this first generation of the modem. Existing state-of-the-art designs for FIR filter reloading show "slow" reloading times not suitable for fast frequency hopping or real time adaptation. One such design uses partial-reconfiguration to re-program a new filter with updated coefficients, which requires 4.57 ms for a 10-tap filter [20].

2) Frame Synchronization: In a non-coherent FSK system, frame synchronization is a critical component to accurately define symbol timing in order to avoid artificially high BER [21]. The following design is a simple hardware implementation of a frame synchronizer, requiring only 51 LUTs and 46 registers for 32-bit data.

At the output of both FIR correlators, there is custom IP, shown in Fig. 4, responsible for calculating a moving window average. This essentially detects the beginning of an FSK symbol. The IP has software defined values for window size W and threshold size H. The beginning of a frame is detected when the following condition is satisfied: $\frac{1}{W} \cdot \sum_{n=0}^{W-1} |x[n]| \geq H$ Once the beginning of a frame is detected, symbol timing within the frame is established and window average is not calculated until the end of the frame. This approach compared to other implementations [21] has drawbacks in that optimal values for W and H must be calculated for an estimated received signal strength but benefits in keeping bit rate unaffected and resource utilization low.

3) Energy Detector: At the output of the FIR correlators, the total energy is calculated by integrating each FIR output over the duration of every symbol as follows: $E_x = \sum_{n=0}^{T \cdot F_s} |x[n]|^2$ where T is the symbol length and F_s is the sampling frequency. In the PL this is implemented using floating point operations as they allow for a greater range than fixed-point. Fig. 4 shows the implementation of the energy detector. The accumulator is reset at every symbol.

4) Bit Decision: Custom IP was developed to perform a simple comparison between the output of each energy detector. Bit 0 or bit 1 is selected for the current symbol based on the largest energy value. The IP also generates a reset to the accumulator in the energy detector based on the values of symbol length and frame length; which are programmable.

The final recovered message is continuously written into dual-port BRAM, which is then accessed by the PS.

IV. Gen1 Supported Communications Schemes

A. Binary Frequency Shift Keying

Utilizing binary FSK (B-FSK) as the modem's base modulation scheme allows for some simplicity in the design. A B-FSK signal is created by shifting the carrier signal, f_c, by Δf based upon the bit $\{0,1\}$ in the digital message being transmitted. A modulated B-FSK symbol has the form:

$$\Phi_i^{B-FSK}(t) = g(t)\cos(2\pi(f_c + m_i\Delta f)t), \quad m_i \in \{0,1\}, \quad i = 1,\ldots,N \tag{1}$$

where $g(t)$ is a rectangular pulse of duration T, and $m_i \in \{0,1\}^N$ is the digital message signal. Guard band – which is defined as the time in between bits where no signal is transmitted – is defined as T_g. It is known that the minimum frequency separation between $f_0 = f_c$ and $f_1 = f_c + \Delta f$ to minimize error probability for B-FSK is when $\Delta f = {}^{0.715}/_{T}$ and that the minimum frequency separation for orthogonality is $\Delta f = {}^{1}/_{2T}$ [22, p. 206].

B. Fast Frequency Hopping – Frequency Shift Keying

For the proposed SWaP UW-A modem, we allow parameters f_c, T, T_g and Δf to be tunable to a wide range of values to mitigate inter-symbol interference (ISI) induced by multipath and interference from other users operating in the same frequency band. Real-time tuning of f_c implements a frequency hopping FSK (FH-FSK) system and the changing of f_c for every symbol implements a fast FH-FSK (FFH-FSK) system [23]. The modulated FFH-FSK symbol becomes similar to Eq. 1 where the carrier is updated to f_j for M hops as follows: $f_j = f_c + j\Delta f, \quad j = 1,\ldots,M$. The spectrogram in Fig. 5 demonstrates rapid switching between modulation types and modulation parameters as well as accurate demodulation by the receiver. The switching pattern is pre-loaded to on-chip LUTs at both the transmitter and the receiver. For future network deployments, we will consider the implementation of AI-assisted receivers that autonomously identify modulation types/parameters adopted by the transmitter, and thus could avoid network synchronization overheads.

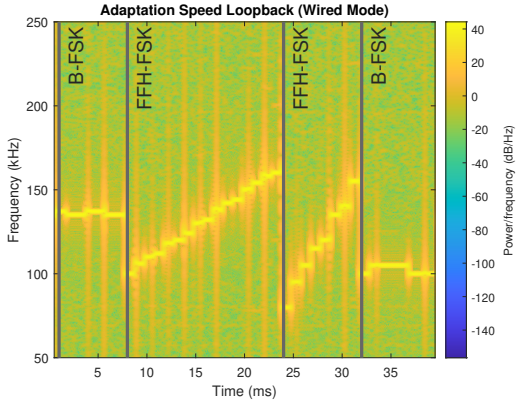

Fig. 5. Demonstration of run-time reconfigurability, by dynamically switching from B-FSK to FFH-FSK and vice-versa.

V. Modem Performance Evaluation

We characterize the performance of the acoustic communication stack first in simulation. Bit error rate (BER) vs signal to noise ratio (SNR) plots were generated using the exact same architecture implemented in the modem hardware. B-FSK simulations were performed with different symbol lengths, guard bands and frequency spacing to guide the selection of optimized values pertaining to the implementation.

A. Underwater Multipath Simulation

We considered 100,000 transmissions of B-FSK symbols over additive white Gaussian noise and underwater multipath. The relationship between SNR and SNR per bit, E_b/N_0 is given by $\mathrm{SNR}(dB) = E_b/N_0(dB) - 10\log_{10}(n_{samp})$ where n_{samp} is the number of samples in a symbol. The simulation generates a multipath UW-A channel using the acoustical method of images [24]. The optimal parameter values determined for the B-FSK implementation were used in an underwater multipath channel with 1, 2, and 5 m between hydrophones as shown in Fig. 6. We observe that increasing hydrophone distance (in greater than 2 m) results in poor channel conditions for the selected parameters. This is due to longer multipath delays that result in ISI. Increasing guard bands between symbols can resolve ISI at the expense of bit rate. An alternative is to use FFH-FSK which keeps the bit rate the same at the expense of bandwidth.

B. Modem Wired Loop-back and Wireless Experiments

We tested BFSK and FFH-FSK in both loop-back (i.e, transmit and receive are connected with a cable) and wireless modes. Fig. 7 shows the spectrogram of a 16-hop and 32-hop FFH-FSK waveform at the receive side of the modems with a direct transmit-receive wired connection and wirelessly with 2 m separation between the hydrophones in a tank. Both modulations occupy the same bandwidth from 100 kHz to 164 kHz, where the 16-hop has 32 total f_0 and f_1 frequencies and the 32-hop has 64 frequencies. We observe that it takes around 25 ms for multipath arrivals to die down before it is ISI-free to transmit and receive at the same frequency again.

C. Field Testing

Initial field testing started in a $1.5\,\mathrm{m} \times 2.4\,\mathrm{m} \times 0.85\,\mathrm{m}$ laboratory water tank, then to a $7.62\,\mathrm{m} \times 9.14\,\mathrm{m} \times 4.57\,\mathrm{m}$ pool and finally to the FAU Seatech harbor. The sea bottom is a combination of mud and sand and the water depth was about 2 m. The transmit and receiver transducers were placed about 1 m below the surface. Acoustic spectrum sensing during harbor tests showed significant interference in the band of operation. Bit error rates for different modulation parameters were calculated – without error correction coding – by a software program (written in C) running on the PS of both transmit and receive modems in which a known 10,000 bit random message was compared to the recovered message. Results are shown in Table I. In parallel, raw samples were saved to the SD card of the modem and plotted in Fig. 7.

D. Hardware Utilization and Power

The full PL design consisting of both transmit and receive hardware components only took 7,919 (45%) LUTs, 13,021 (37%) registers, 30.5 (50.8%) BRAM, and 49 (61.25%) DSP blocks of a Zynq-7010 FPGA SoC. Power analysis of the SoC shows total on-chip power of 1.806 W.

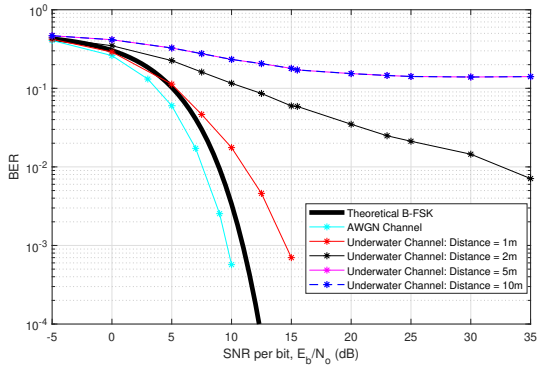

Fig. 6. B-FSK simulation results with UW-A multipath channel.

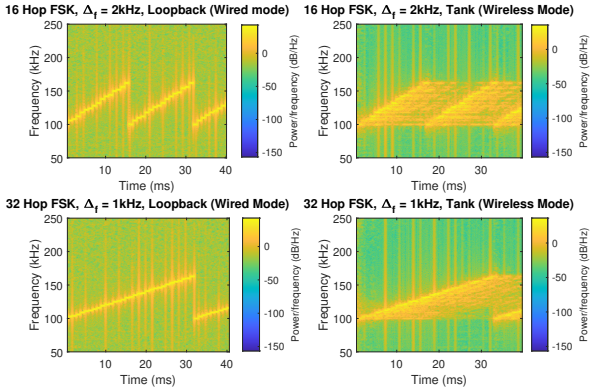

Fig. 7. FFH-FSK spectrogram from loopback communication (left), where the modems communicate over a wire and in a water tank (right), where modems communicate wirelessly.

Link Distance (m)	Data rate (bps)	Freq.-Hops j	T_g (ms)	$f_j = f_c + j\Delta f$ (Hz)	Δf (Hz)	BW $= \Delta f + 1/T$ (Hz)	BER
2 (Tank)	66.7	0	14	150k	2k	4k	0.0049
2 (Tank)	66.7	0	14	100k	2k	4k	10^{-4}
2 (Tank)	500	16	1	100k	2k	64k	0
2 (Tank)	1000	32	0.5	100k	1k	64k	0
2 (Tank)	2000	32	0	100k	1k	64k	0.1037
5 (Pool)	58.9	0	16	100k	2k	4k	0.00178
5 (Pool)	500	16	1	100k	2k	64k	0.001185
5 (Pool)	500	32	1	100k	1k	64k	0.0047
5 (Pool)	1000	8	0.5	88k	5k	75k	0
5 (Pool)	2000	8	0	88k	5k	75k	0.086
8 (Pool)	25	0	39	100k	2k	4k	0.0082
8 (Pool)	500	16	1	100k	2k	64k	0.0065
8 (Pool)	1000	8	0.5	88k	5k	75k	0.052
10 (Pool)	23.8	0	41	100k	2k	4k	0.006
10 (Pool)	500	16	1	100k	2k	64k	0.042
10 (Pool)	1000	8	0.5	100k	5k	75k	0.35
50 (Harbor)	100	16	9	100k	2k	64k	0.014

TABLE I
Modem Tests.

VI. Conclusion

We design, test and evaluate a new class of low-SWaP UW-A modems, that demonstrates in its first generation, μs-level communication switching between FSK and FFH-FSK for adaptive communications between micro-AUV nodes. The UW-A modem is based on a custom credit-card sized SoC which provides an unmatched flexibility, low power, low cost, high-speed, and practically unlimited customizability for next-generation connected underwater autonomous systems.

REFERENCES

[1] A. Hackbarth, E. Kreuzer, and E. Solowjow, "Hippocampus: A micro underwater vehicle for swarm applications," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 2258–2263.

[2] J. Jaffe, P. Franks, D. Mirza, C. Schurgers, R. Kastner, and A. Boch, "A swarm of autonomous miniature underwater robot drifters for exploring submesoscale ocean dynamics," *Nature Communications*, vol. 8, 2017.

[3] J. Shao, J. Yu, and L. Wang, "Formation control of multiple biomimetic robotic fish," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 2503–2508.

[4] T. Schmickl, R. Thenius, C. Moslinger, J. Timmis, A. Tyrrell, M. Read, J. Hilder, J. Halloy, A. Campo, C. Stefanini, L. Manfredi, S. Orofino, S. Kernbach, T. Dipper, and D. Sutantyo, "Cocoro – the self-aware underwater swarm," in *2011 Fifth IEEE Conference on Self-Adaptive and Self-Organizing Systems Workshops*, 2011, pp. 120–126.

[5] B.-C. Renner, J. Heitmann, and F. Steinmetz, "Ahoi: Inexpensive, low-power communication and localization for underwater sensor networks and auvs," *ACM Trans. Sen. Netw.*, vol. 16, no. 2, jan 2020. [Online]. Available: https://doi.org/10.1145/3376921

[6] J. A. Neasham, G. Goodfellow, and R. Sharphouse, "Development of the "Seatrac" miniature acoustic modem and USBL positioning units for subsea robotics and diver applications," in *OCEANS 2015 - Genova*, 2015, pp. 1–8.

[7] J. Neasham, A. Chawla, B. Sherlock, and I. Lončar, "A multi-user underwater acoustic positioning system – experiments using asvs and miniature acoustic transceivers," in *OCEANS 2021: San Diego – Porto*, 2021, pp. 1–5.

[8] G. Cario, A. Casavola, M. Lupia, and C. Rosace, "Seamodem: A low-cost underwater acoustic modem for shallow water communication," in *OCEANS 2015-Genova*. IEEE, 2015, pp. 1–6.

[9] E. Demirors, J. Shi, A. Duong, N. Dave, R. Guida, B. Herrera, F. Pop, G. Chen, C. Casella, S. Tadayon, M. Rinaldi, S. Basagni, M. Stojanovic, and T. Melodia, "The SEANet Project: Toward a programmable Internet of Underwater Things," in *2018 Fourth Underwater Communications and Networking Conference (UComms)*, 2018, pp. 1–5.

[10] R. Guida, E. Demirors, N. Dave, and T. Melodia, "Underwater ultrasonic wireless power transfer: A Battery-less platform for the Internet of Underwater Things," *IEEE Transactions on Mobile Computing*, vol. 21, no. 5, pp. 1861–1873, 2022.

[11] R. Guida, E. Demirors, N. Dave, J. Rodowicz, and T. Melodia, "An acoustically powered battery-less Internet of Underwater Things platform," in *2018 Fourth Underwater Communications and Networking Conference (UComms)*, 2018, pp. 1–5.

[12] E. Demirors, G. Sklivanitis, T. Melodia, S. N. Batalama, and D. A. Pados, "Software-defined underwater acoustic networks: toward a high-rate real-time reconfigurable modem," *IEEE Communications Magazine*, vol. 53, no. 11, pp. 64–71, 2015.

[13] A. Gannon, S. Balakrishnan, G. Sklivanitis, D. A. Pados, and S. N. Batalama, "Short data record filtering for adaptive underwater acoustic communications," in *2018 IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, 2018, pp. 316–320.

[14] G. Sklivanitis, E. Demirors, S. N. Batalama, T. Melodia, and D. A. Pados, "Receiver configuration and testbed development for underwater cognitive channelization," in *2014 48th Asilomar Conference on Signals, Systems and Computers*, 2014, pp. 1594–1598.

[15] J. Hermans, G. Sklivanitis, and D. Pados, "Center for Connected Autonomy and AI (CA-AI) Underwater Modem," https://github.com/C2A2-at-Florida-Atlantic-University/UnderwaterModem, 2022.

[16] G. Xilinx and S. Guide, "Zynq-7000 All programmable SoC technical reference manual (ug585)," Tech. rep., Xilinx, 2014, https://www. xilinx. com/support/documentation, Tech. Rep., 2014.

[17] Xilinx, "Petalinux tools documentation, reference guide," 2021.

[18] ——, "Vitis high-level synthesis user guide," https://docs.xilinx.com/r/en-US/ug1399-vitis-hls/Customizing-AXI4-Lite-Slave-Interfaces-in-IP-Integrator, 2022.

[19] I. LogiCORE, "FIR Compiler v5.0, Xilinx," *Inc., San Jose, CA, USA*, 2010.

[20] G. Zhao, Q. Ge, and Y. Zhang, "Dynamically reconfigurable FIR filter design based on FPGA," in *2015 Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC)*. IEEE, 2015, pp. 1631–1635.

[21] Y. Li, X. Zhang, B. Benson, and R. Kastner, "Hardware implementation of symbol synchronization for underwater FSK," in *2010 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*. IEEE, 2010, pp. 82–88.

[22] J. G. Proakis and M. Salehi, *Digital communications*. McGraw-hill New York, 2001, vol. 4.

[23] J. H. Schmidt, "Using fast frequency hopping technique to improve reliability of underwater communication system," *Applied Sciences*, vol. 10, no. 3, p. 1172, 2020.

[24] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *The Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 1979.