Weak Signal Underwater Communications in the Ultra Low Frequency Band

Michel Barbeau BARBEAU@SCS.CARLETON.CA
School of Computer Science, Carleton University, 1125 Colonel By Drive, Ottawa, ON, Canada K1S 5B6

Abstract

We investigate the problem of underwater communications with weak received signal strength in the ultra low frequency acoustic band. It is a band where attenuation is relatively low, but half-power bandwidth is thin. We leverage the work of Taylor and Walker (2017) on Weak Signal Propagation Reporter. It comprises narrow band multiple frequency-shift keying modulation, long synchronization bit sequences intertwined with data bits and high constraint convolutional forward error correction with probabilistic decoding. In contrast to the original work of Taylor and Walker, in our system frames can be sent anytime. No synchronization in reference to a universal clock is required. We review the mathematical foundations, software design and preliminary tests of our approach for underwater communications under weak received signal strength conditions.

1. Introduction

Underwater acoustic communications are used by subsurface activity sensors (Otnes et al., 2012), unmanned undersea vehicles (Button et al., 2009) and airplane underwater locator beacons (Wikipedia, 2017). Acoustic signals are subject to attenuation. They propagate in an environment that contains anthropogenic and naturogenic noise. We focus our attention on underwater weak receive signal strength, which results from long range communications or use of minimum power that prolongs the battery lifetime or intentionally low power acoustic signals aiming to reduce impact on marine life. For long range underwater communications, the Ultra Low Frequency (ULF) band is favoured because there is less attenuation at the lower end of the acoustic spectrum. For instance, Freitag et al. (Freitag, 2015) have been able to achieve communication over a 400 kilometer range at 900 Hertz. On the other hand, in the ULF the half-power bandwidth is relatively narrow. It means that solely extremely slow baud rates are possible.

Proceedings of the 7th GNU Radio Conference, Copyright 2017 by the author(s).

Using a GNU Radio software-defined approach (GNU Radio, 2017) and reusing previous software development work accomplished by Franke and Taylor (Franke & Taylor, 2017), we propose a solution for weak signal underwater communications. The highlights of the solution include: narrow band Multiple Frequency-Shift Keying (MFSK) modulation, long synchronization bit sequences intertwined with data bits and high constraint convolutional Forward Error Correction (FEC) with probabilistic decoding.

Section 2 introduces the challenge of underwater communications in the ULF band. The design of the sender is reviewed in Section 3, while the receiver side is discussed in Section 4. Section 5 covers in more depth the FEC aspect. Our hardware platform and tests that we conducted are reviewed in Section 6. We conclude in Section 7.

2. Underwater Communications

Underwater communications rely on sound waves. In contrast to waves used in classical wireless communications, they travel at much lower speed. Therefore, propagation delays are significant. Assuming a propagation speed of 1.5 km/second, for long range underwater communications, the propagation delays can easily be in the order of minutes. Typically, the sound speed is variable and depth dependent. This phenomenon is the cause of non negligible refraction. Other significant impairments are attenuation and numerous sources of noise. They are discussed in an article authored by Stojanovic (Stojanovic, 2007).

For long range communications, attenuation is an important issue. The main causes are conversion of acoustic energy into heat and geometrical spreading. The magnitude of underwater attenuation is represented in the Thorp's model (Thorp, 1967; 1965; Thorp & Browning, 1973). Figure 1 plots the attenuation as a function of distance for selected frequencies in the ULF band. Realistically, for long range solely the use of low frequencies can be envisioned. Another important fact is the gradient of the attenuation versus frequency, it limits the operating bandwidth. The half-power bandwidth is the difference between a reference frequency and the upper frequency where transmission loss is 3 dB up. The half-power concept is used to define cutoff frequencies and bandwidths of filters. Figure 2 shows the

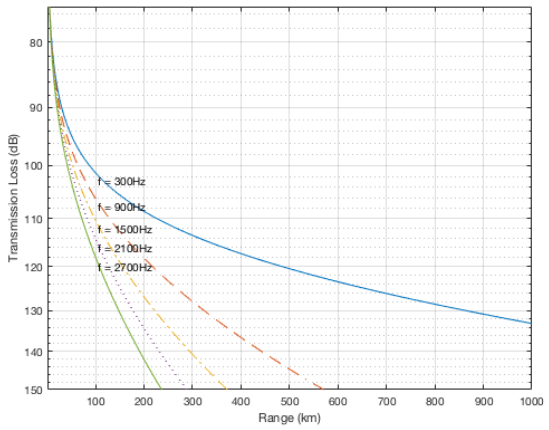

Figure 1. Underwater attenuation for selected ULF frequencies.

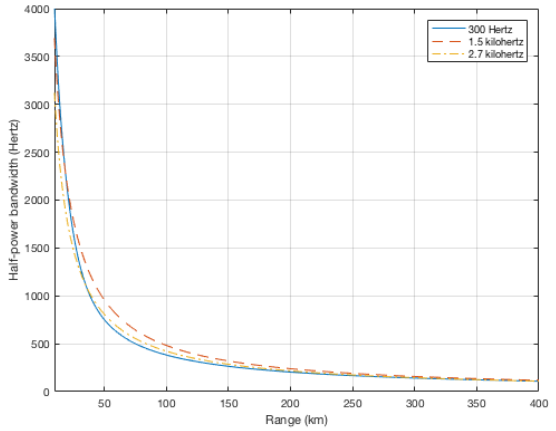

Figure 2. Half-power bandwidth for selected ULF frequencies.

half-power bandwidth for selected ULF frequencies versus distance.

That being said, theory is one thing and reality is another. Making an underwater contact is much harder to do than a classical radio contact (Dol et al., 2017; Blouin & Barbeau, 2017). Making an underwater contact, even for a distance in the order of kilometers is challenging. One of the practical difficulties is that underwater is an hostile environment to electronics. The failure or success of underwater communications depends on several factors compounded together.

3. Sender Side

Our work is based on the earlier software development efforts of Taylor and Walker (Taylor & Walker, 2010), on WSPR, and Franke and Taylor (Franke & Taylor, 2017), on Weak Signal Propagation Reporter (WSPR) implementation, Fano (Fano, 1963), on probabilistic convolutional decoding, and Karn (Karn, 1995), on convolutional decoder implementation. We use the frame format and modulation that are originally defined in WSPR (Taylor & Walker, 2010). The motivation is the ability of WSPR to communicate with a Signal-to-Noise Ratio (SNR) as low as -28 dB (in a 2.5 kilohertz reference bandwidth). Around the world communications are possible with five Watt-transmission power, or less.

It is a frame-oriented protocol. Each frame consists of 162 channel symbols. They comprises 50 information bits. Convolution coding is used, with a constraint (K) of 32 and a rate (r) of $1/2$. Convolutional coding FEC yields 162 bits data. Bit interleaving is used to increase robustness to burst errors. They are combined with with 162 binary synchronization bits s_i ($i = 1, \ldots, 162$). A pairing of each data bit with a synchronization bit is done to construct each channel symbol.

The least significant bit of each channel symbol is a synchronization bit. In WSPR, frame transmission is synchronous, from a second after every even Ultra Low Frequency (UTC) minute. Every frame fits into a two minute interval. A WSPR receiver searches for frames at the beginning of every even UTC minute. Underwater communications, with propagation delays in the order of several seconds and minutes, make application of the original WSPR synchronous scheme difficult. In our system, frames can be sent anytime. No synchronization in reference to a universal clock is required. Frame transmission is asynchronous.

Modulation is four-tone MFSK at 1.46 baud ($375/256$). The complex modulation envelope frequencies are -3, -1, 1 and 3 Hertz, corresponding to channel symbols 0, 1, 2 and 3. Hence, the signal bandwidth is six Hertz. The transmission time of a frame is:[1]

$$162 \text{ channel symbols}/1.46 \text{ baud} = 111 \text{ seconds}$$

4. Receiver Side

The complexity is in the receiver. It searches for frames in a spectrum of 300 Hertz, which is located anywhere in the ULF band. Processing of input is done according to a sliding window model, see Figure 3. Three processing iterations are shown, from top to bottom. Each iteration processes a window of 120 seconds of channel data. The next window slides in time for nine seconds. From window-to-window, there is an overlap of 111 seconds. Window overlapping insures that a frame (111 seconds) always fits within a window, although handling of duplicates is required when a frame happens to fit within exactly two consecutive windows because it is decoded twice. Processing

[1]The exact calculation is $162/(375/256)$ seconds.

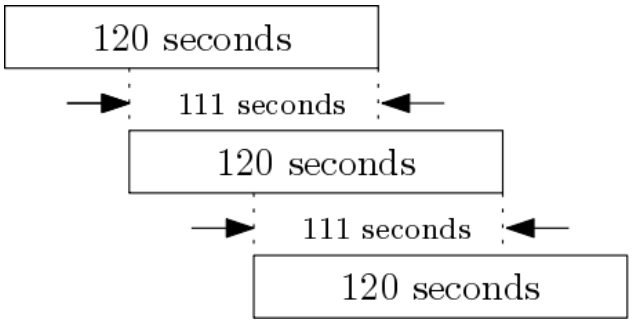

Figure 3. Sliding window on channel data.

of each 120-second channel data chunk is dispatched to a decoding thread. The decoding time is proportional to the number of frames contained in the channel data. Indeed, because of the narrow signal bandwidth (six Hertz), several frames can be packed in a 300 Hertz bandwidth. On a mini PC, decoding takes between a fourth of a second and three seconds, which fits comfortably within the nine-second window-to-window interval. Decoding is done using a time domain and a frequency domain representations of channel data.

4.1. Time Domain Representation

Audio is captured using an analog to digital converter, e.g., a sound card. It is band pass filtered and translated to the zero Hertz-center frequency. Every two-minute of channel data is represented in the time domain by a series of discrete complex samples $x_0, x_1, \ldots, x_{n-1}$, where n is 45,000. The sampling rate f_s is $n/120 = 375$ samples per second (sps). Each channel symbol is represented by 256 samples. There are 162 channel symbols in a frame. A frame consists of 256 samples per symbol times 162 channel symbols, i.e., 41,472 samples. The 120-second of channel data is searched for 111-second frames. Because of the sampling rate and Nyquist criterion, the spectrum represented by the samples is $\pm\lfloor f_s/2 \rfloor = \pm 187$ Hertz. The time domain representation is used to generate a frequency domain representation and signal demodulation.

4.2. Frequency Domain Representation

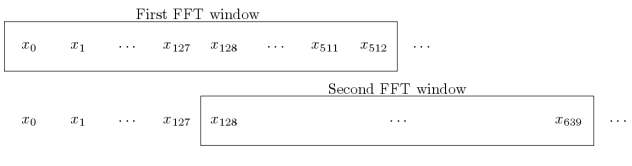

Figure 4. Windowed DFT.

For every two-minute interval, a coarse search for candidate signals is done using a frequency domain representation of the channel data, obtained with a Discrete Fourier Transform (DFT) of the time domain representation. Windowed DFTs are calculated, see Figure 4. Each DFT represents a time interval corresponding to the duration of two symbols, i.e., 512 samples. The size N of each DFT is 512 bins. The DFTs are calculated from the beginning of a two-minute interval, in steps of half symbol (128 samples). The number of DFTs is:

$$\texttt{nffts} = \left\lfloor \frac{n}{128} \right\rfloor - 3 = 348 \text{ DFTs}$$

The term "-3" is present because calculations of windowed DFTs stop before the third to last half-sample. Every coefficient of the DFTs is denoted as $X_{m,k}$, with the DFT window index m in the range $0, \ldots, \texttt{nffts} - 1 atimatom$. The windowing function (w) is

$$w(t) = \sin\left(\frac{\pi}{512} \cdot t\right), \text{ t=0,...,N-1.}$$

It is a half-sine wave cycle, in N steps.

Let $j = \sqrt{-1}$. Every DFT coefficient is defined as:

$$X_{m,k} = \sum_{t=0}^{N-1} x_{128m+t} \cdot w(t) \cdot e^{-j2\pi it/N}$$

It represents the relative amplitude, and phase, of frequency:

$$\frac{k \cdot f_s \text{ sps}}{N \text{ samples}} \text{ Hertz}$$

From frequency-bin-to-frequency-bin, there is an offset Δf of $375/512 = 0.73$ Hertz. At 375 sps and according to Nyquist criterion, the frequency range of each DFT is ± 187 Hertz. The coefficient index k is in the range $\pm 187 \cdot N/f_s = \pm 255$. Figure 5 illustrates the spectrum

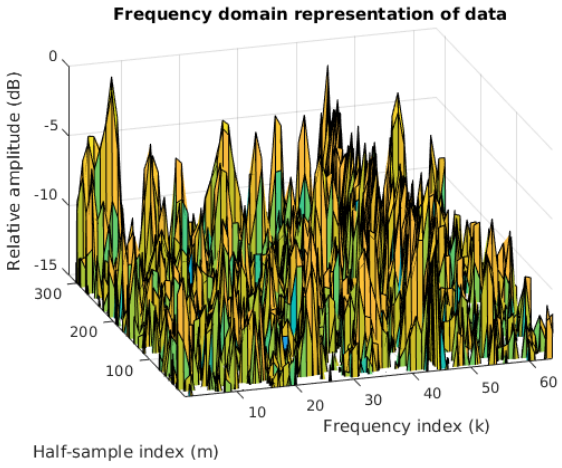

Figure 5. Frequency domain representation (noisy channel data).

of a noisy signal that contains a frame at frequency index 45.

4.3. Coarse Search for Candidate Frequencies

This step identifies frequencies that may potentially contain frames. The frequency domain representation is used for a coarse signal search. The procedure looks for frequencies where there is a *local SNR maximum*, across the two minute interval. Over all DFTs, the power is summed for each frequency:

$$S_k = \sum_{m=0}^{\texttt{nffts-1}} X_{m,k}$$

To reduce fluctuations, the spectrum is smoothed using a seven point spectral window ($7 \cdot \Delta f = 5$ Hertz) and the following summation:

$$M_k = \sum_{l=k-3}^{k+3} S_l$$

The noise level is calculated using a ± 150 Hertz spectrum, ($k = -205$ to 205). The M_k coefficients are sorted, from lowest to largest. Let M'_k denote a sorted coefficient. The noise level is determined by the power level at the 30th percentile, i.e., in the sorted list the power level at rank:

$$\kappa = \lfloor 30\% \cdot 205 - (-205) \rfloor$$

Using this reference noise level, the SNR for each frequency is calculated:

$$SNR_k = \frac{M_k}{M_\kappa}$$

A frequency becomes a candidate when there is a local SNR maximum. That is, index k is a candidate when

$$SNR_{k-1} < SNR_k > SNR_{k+1}.$$

4.4. Coarse Resolution of Timing

For each candidate frequency, this step finds a coarse time offset, from the start of a two minute interval. Each candidate frequency is examined. A complete frame can start anywhere from the beginning to a time delay corresponding to nine seconds (120 minus 111 seconds) into the interval. In half-symbols:

$$\delta_{max} = \left\lfloor \frac{f_s \text{ sps}}{128 \text{ samples/half-symb.}} \cdot 9 \text{ s.} \right\rfloor = 26 \text{ half-symb.}$$

Let $W_{m,l} = |X_{m,l}|$ denote the magnitude spectrum at indices m and l. Let k denote the index of a candidate frequency. The timing offset δ is the value in the range $0, \ldots, \delta_{max}$ that maximizes the ratio:

$$\sum_{i=1+\delta}^{162} (2s_i - 1) \left[\frac{(W_{i,k-4} + W_{i,k+1}) - (W_{i,k-1} + W_{i,k+4})}{\sum_{l=k-4,k-1,k+1,k+4} |W_{i,l}|} \right]$$

The summation measures the correlation of the spectrum power around frequency index k with the synchronization bit-string s. The multiplicand $2s_i - 1$ maps the synchronization bit s_i,which is 0 or 1, to value -1 or 1. The term $W_{i,k-4} + W_{i,k+1}$ is the sum of the power at the frequencies of synchronization bit value 1, while the term $W_{i,k-1} + W_{i,k+4}$ is the sum of the power at the frequencies of synchronization bit value 0. The denominator represents the sum of all power around candidate frequency k. In other words, power at synchronization bits is relativized to all the power at the candidate frequency. This *offline* approach to synchronization contrasts with the *online* timing recovery techniques commonly used in classical wireless communications (Mueller & Muller, 1976).

4.5. Demodulation and Refined Timing

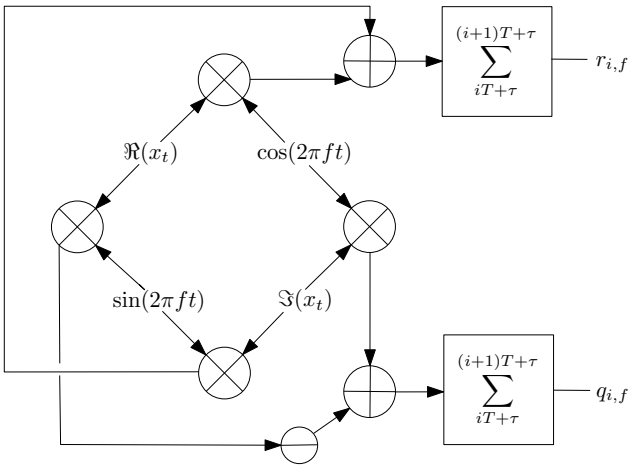

Figure 6. Demodulator: $f \in -3, -1, 1, 3$ and $i = 1, \ldots, 162$.

A key element of the demodulator is depicted in Figure 6. The terms $\Re(x_t)$ and $\Im(x_t)$ represent the real and imaginary parts of a complex sample at time t. It is a non-coherent type of demodulation, i.e., the carrier phase is ignored (Proakis & Salehi, 2008; Proakis et al., 2013). There are four signal waveforms, at frequencies $f = -3, -1, 1, 3$, and four correlators per waveform (diagonally crossed circles). The two rectilinearly crossed circles represent adders. The horizontally split circle represents a minus operation. Let $j = \sqrt{-1}$. The received signal is correlated with values resulting from the evaluation of the expression:

$$\begin{aligned} & x_t \cdot e^{-j2\pi ft} \\ &= [\Re(x_t) + j\Im(x_t)] \left[\cos(2\pi ft) - j \sin(2\pi ft)\right] \\ &= \Re(x_t) \cos(2\pi ft) + \Im(x_t) \sin(2\pi ft) \\ &+ j \left[\Im(x_t) \cos(2\pi ft) - \Re(x_t) \sin(2\pi ft)\right] \end{aligned}$$

Over a symbol interval T, the power is summed to obtain

the energy:

$$r_{i,f} + jq_{i,f} = \sum_{iT+\tau}^{(i+1)T+\tau} x_t \cdot e^{-j2\pi ft} \quad (1)$$

Which is mapped to a magnitude:

$$P_{i,f} = |r_{i,f} + jq_{i,f}| \quad (2)$$

The four magnitudes P_f ($f = -3, -1, 1, 3$) are used to calculate soft symbols. A hard symbol represents a numerical value, for instance 0 or 1. In contrast, a soft symbol represents a value and its quality. Receive quality metrics are associated with the symbol. This information is used in the decoding process. The most likely symbols are selected first. More details about soft symbols can be found in Appendix A.

The timing offset τ is chosen in the interval $[-128, 128]$ that maximizes the correlation of the synchronization soft symbols with the synchronization binary pattern $s_1, \ldots, s_{162}$. A deinterleaving procedure reorders the 162 data soft symbols. The resulting 162 soft symbols are passed to FEC.

5. Forward Error Correction

FEC is based on convolutional coding. Convolutional coding is popular because of its relative simplicity. The coding process has two parameters: a constraint (K) and a rate (r). The data is processed bit-by-bit. The K most recent bits are used to generate r codeword bits. Let n be the frame size (bits), $n \geq K$. The first $n-1$ bits generate $r(n-1)$ codeword bits. The last bit generates rK codeword bits, for a total of $r(n+K-1)$ codeword bits. For a given FEC and with respect to an uncoded signal, the concept of *coding gain* refers to the reduction of SNR per bit[2] required to achieve a given BER. It turns out that the coding gain is proportional to K to r.

In the convolutional approach, the challenge is in decoding. The are two options: parallel or sequential. Given a received frame f', decoding determines the corresponding probable transmitted frame f. Parallel decoding explores all hypothetical frames simultaneously. At the end, it returns the most probable sequence of bits. Viterbi is the main parallel algorithm (Viterbi, 1967). On the one hand, parallel decoding is optimal and runs at constant speed. On the other hand, parallel decoding cannot deal with large constraints because its memory complexity is exponential, i.e., in the order of 2^{K-1}. Therefore, a typical Viterbi decoder used for classical wireless communications operates with a low constraint, e.g., seven, and rate $1/2$.

[2]The ratio E_b/N_0.

Table 1. Decoding performance of sequential relative to Viterbi decoding, according to Ref. (Karn, 1995).

Decoder	Constraint (K)	Rate (r)	Decoding rate (bits/time units)
Viterbi	7	$1/2$	1
Sequential	32	$1/2$	0.22

Sequential decoding tries one hypothetical frame f at a time. It stops when a *good* match is resolved or when allocated search time expires, which means a failure to decode. A hypothetical frame f corresponds to a search path. Not all paths are examined entirely, because their probability falls below a threshold before reaching their end. In such cases, the decoder backtracks and tries another hypothesis. When a processed frame contains relatively few errors, sequential decoding resolves the case quickly. However, in the presence of a relatively large number of errors, sequential decoding performs worst because it explores several alternatives before finding a good match. For sequential decoding there is number of options such as the Fano algorithm (Fano, 1963). Fano decoding can be tuned to favor good run time on frames with high numbers of errors, at the expense of slower run time on frames with less errors. An alternative is the Jelinek algorithm (Jelinek, 1969). Faster run time than the Fano algorithm is claimed. Sequential decoding is typically used for space probe communications and weak signal communications. Sequential decoding is not optimal. On the other hand, it is significantly stronger than parallel decoding. For instance, a non FEC Binary Phase-Shift Keying (BPSK) signal achieves a bit error probability of 10^{-5} with a SNR per bit of 9.6 dB. Karn is the author of an open source implementation of Fano decoding (Karn, 1995). He has analyzed the decoding performance of sequential ($K = 32$) and Viterbi decoding ($K = 7$). The key result is summarized in Table 1. Within these parameters, the decoding rate of sequential decoding is 22% the one of Viterbi. In this project, we use a convolution coding and a Fano decoder with the constraint (K) and rate (r) $1/2$. The lower rate of sequential decoding has little impact because of the low frame arrival time, which gives ample time to decode them.

6. Hardware and Trials

Modems, and accompanying protocols, are implemented in software using GNU Radio (GNU Radio, 2017), see Figures 7 and 8. They run in the Linux environment on a mini PC. To interface with the acoustic world, a recording studio-quality sound card is used (AudioBox USB). The hydrophone (DolphinEar PRO) plugs in. The speaker (Ocean Technology System - Diver Recall System-100) has its own amplifier (up to 132 Watts) and a line-in input connected to a laptop line-out. Both the hydrophone and

Figure 7. Receiver system.

Figure 8. Sender system.

speaker have long cables that allow operation in depths of up to eight meters. We tested our system at the Hartwell's Lockstation. The longest distance we could test was 90 meters, see Figure 9. The test is interesting because of the echoic environment, due to the canal concrete walls. A few Watts were used, although we cannot calibrate the exact transmission power yet. We used a carrier at 1.5 kilohertz. All communications were perfect.

We also did tests in the Ottawa river in an area called Lac Déchênes. This test is interesting because we were able to plunge the speaker in open water. The transmitter was on board the Sassy Gaffer sailboat. Figure 11 shows the Sassy Gaffer. The blue marker on the map of Figure 10 indicates the sailboat test location. Figure 12 shows the on board test setup. The hydrophone and receiver stayed on shore (yellow marker on Figure 10). We were expecting the presence of a dock, where we could plunge the hydrophone also in a kind of open water condition. There was no such dock. We could only put it in a couple of feet of water (may be three). Theoretically, due to the shallow water much of the acoustic energy is absorbed by seabed before reaching the shore so close. Given these far from optimal conditions, were able to make a 2.6 km contact.

7. Conclusion

We proposed a solution for weak signal underwater communications. For long-range communications, important attenuation results into weak receive signal strength. For short-range communications, the use of low transmission power has potentially lower environmental impact, but also results into weak signal. Our solution builds mainly upon the work of Franke and Taylor (Franke & Taylor, 2017) on WSPR and has been developed within the GNU Radio environment (GNU Radio, 2017). Source code, examples and flowgraphs are available online: https://github.com/michelbarbeau/gr-uwspr.

Acknowledgments

We acknowledge financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC).

A. Soft Symbols

The concept of soft symbol is used to determine the precise timing offset, data bit values and their likelihood. Relative to the start of a window of channel data, refined calculation of the timing offset of a frame is based on the total energy in the symbols of a candidate frame, that is,

$$\sigma = \sum_{\substack{i=1,\ldots,162 \\ f=0,1,2,3}} P_{i,f},$$

using Equations 1 and 2 to calculate the $P_{i,f}$ terms. The timing offset τ is selected in the interval $[-128, 128]$ such that the following ratio is maximized:

$$\sum_{i=1}^{162} (2s_i - 1) \frac{[(P_{1,f} + P_{3,f}) - (P_{0,f} + P_{2,f})]}{\sigma}$$

Figure 9. Hartwell's Lockstation viewed from the air.

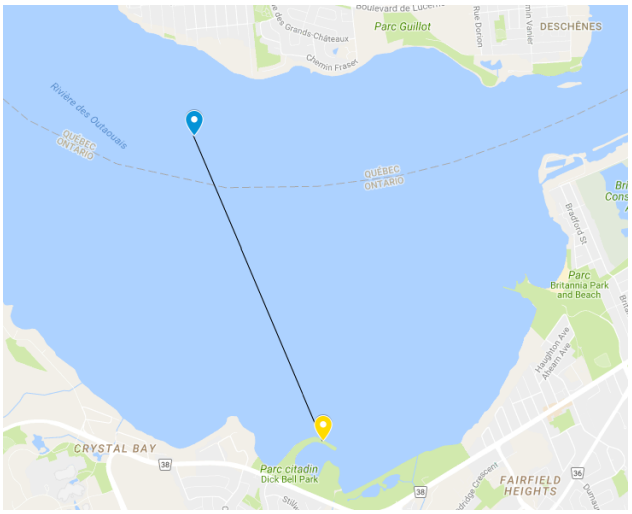

Figure 10. Ottawa river, Lac Déchênes test.

The expression $P_{1,f} + P_{3,f}$ represents the energy at synchronization bit value 1, while the sum $P_{0,f} + P_{2,f}$ corresponds to the energy at synchronization bit value 0. According to the value of the expected synchronization bit s_i at position i in the frame, the numerator selects the corresponding energy.

The data soft symbol at position i in a frame is calculated using a metric that reflects the difference between the energy associated with data bits 1 and 0:

$$b_i = s_i(P_3 - P_1) + \overline{s_i}(P_2 - P_0)$$

If the synchronization bit is 1 (condition s_i is verified), the expression $P_3 - P_1$ measures the difference of energy associate with data bits 1 and 0. If the synchronization is 0 (condition $\overline{s_i}$ is verified), this difference is measured by the expression $P_2 - P_0$. Using a divisor γ and a multiplicand β, each value b_i is normalized and mapped to a value in the interval $[-128, 127]$, i.e., the soft data symbol is:

$$d_i = \max\left[\min\left(\beta\frac{b_i}{\gamma}, 127\right), -128\right] \qquad (3)$$

The $\min$ and $\max$ operators clip the normalized value in the range $[-128, 127]$. The closer the value of d_i to -128, the stronger the belief that the data bit at position i is a zero. The closer to 127, the stronger the belief that it is a 1. The normalization multiplicand β is 50. The normalization divisor is:

$$\gamma = \sqrt{\frac{1}{162}\sum_{i=1,\ldots,162} b_i^2 - \left(\frac{1}{162}\sum_{i=1,\ldots,162} b_i\right)^2}$$

References

Blouin, S. and Barbeau, M. An experimental baseline for underwater acoustic broadcasts. In *To appear in: IEEE 86th Vehicular Technology Conference (VTC2017-Fall)*, Toronto, Canada, 2017.

Button, Robert W., Kamp, John, Curtin, Thomas B., and Dryden, James. A survey of missions for unmanned undersea vehicles, 2009. RAND National Defense Research Institute.

Dol, H. S., Casari, P., van der Zwan, T., and Otnes, R. Software-defined underwater acoustic modems: Historical review and the NILUS approach. *IEEE Journal of Oceanic Engineering*, PP(99):1–16, 2017.

Fano, R. A heuristic discussion of probabilistic decoding. *IEEE Transactions on Information Theory*, 9(2):64–74, April 1963.

Franke, S. and Taylor, J. WSPR, 2017. URL `http://physics.princeton.edu/pulsar/K1JT/wspr.html`. [Online; accessed 9-May-2017].

Freitag, L. et al. Long range acoustic communications and navigation in the Arctic. In *OCEANS 2015 - MTS/IEEE Washington*, pp. 1–5, October 2015.

GNU Radio. GNU Radio - The free and open software radio system, 2017. URL `https://www.gnuradio.org`. [Online; accessed 6-June-2017].

Jelinek, F. Fast sequential decoding algorithm using a stack. *IBM Journal of Research and Development*, 13 (6):675–685, Nov 1969.

Figure 11. The transmitter was carried by the Sassy Gaffer.

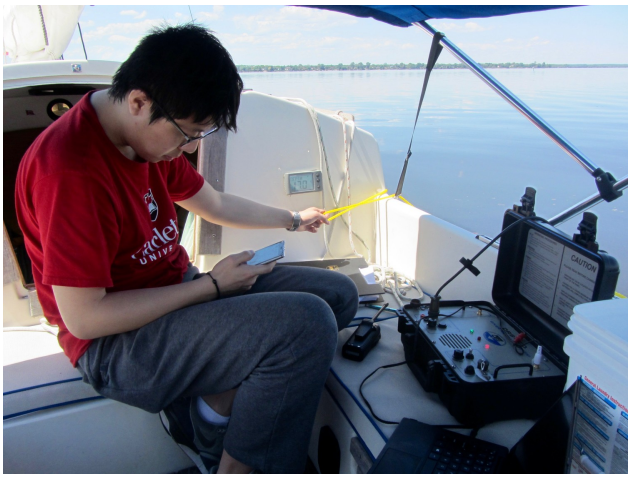

Figure 12. Test setup setup on the Sassy Gaffer.

Karn, P. Convolutional decoders for amateur packet radio. In *ARRL Digital Communications Conference*, pp. 45–50, 1995.

Mueller, K. and Muller, M. Timing recovery in digital synchronous data receivers. *IEEE Transactions on Communications*, 24(5):516–531, May 1976.

Otnes, Roald, Asterjadhi, Alfred, Casari, Paolo, Goetz, Michael, Husøy, Thor, Nissen, Ivor, Rimstad, Knut, Van Walree, Paul, and Zorzi, Michele. *Underwater acoustic networking techniques*. Springer Science & Business Media, 2012.

Proakis, J.G. and Salehi, M. *Digital Communication*. McGrah-Hill Higher Education, fifth edition, 2008.

Proakis, J.G., Salehi, M., and Bauch, G. *Contemporary Communication Systems Using MATLAB*. Cengage Learning, third edition, 2013.

Stojanovic, M. On the relationship between capacity and distance in an underwater acoustic communication channel. *SIGMOBILE Mob. Comput. Commun. Rev.*, 11(4): 34–43, October 2007. ISSN 1559-1662.

Taylor, J. and Walker, B. WSPRing around the world. *QST*, 94(10):30–32, November 2010.

Thorp, W.H. Deep ocean sound attenuation in the sub and low kilocycle per second region. *The Journal of the Acoustical Society of America*, 38(4):648–654, 1965.

Thorp, W.H. Analytic description of the low frequency attenuation coefficient. *The Journal of the Acoustical Society of America*, 42:270, 1967.

Thorp, W.H. and Browning, D.G. Attenuation of low frequency sound in the ocean. *Journal of Sound and Vibration*, 26:576–578, February 1973.

Viterbi, A. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, April 1967. ISSN 0018-9448.

Wikipedia. Underwater locator beacon, 2017. URL `https://en.wikipedia.org/wiki/Underwater_locator_beacon`. [Online; accessed 19-June-2017].