Reliable Symbol Synchronization in Software-Driven Acoustic Sensor Networks

Raja Jurdak*, Antonio G. Ruzzelli*, Gregory M.P. O'Hare*, and Cristina Videira Lopes†

* University College Dublin Belfield D4 Ireland
{raja.jurdak, ruzzelli, gregory.ohare} @ucd.ie
†University of California, Irvine CA 92697 U.S.A
lopes@ics.uci.edu

***Abstract*—Symbol synchronization in traditional hardware-driven communication systems has relied on the transmission of training sequences of symbols just before the beginning of the frame symbols. The use of training sequences is not suitable for software-driven communication systems, such as lightweight acoustic underwater sensor networks [4, 5], in which the high symbol loss rate may cause the loss of training symbols, preventing accurate symbol synchronization. Software-driven communication networks require symbol synchronization that is resilient to a high loss environment, that does not represent large communication or processing overhead, and that is tunable to the noise profile of different environments. These requirements are emphasized for mote-based acoustic underwater sensor networks in which the bandwidth and processing capability are sparse. This paper proposes the use of a short signature synchronization symbol (S^4) as both a preamble and post-amble to enable receiver synchronization in mote-based acoustic communication systems that rely on software modems. To synchronize to an incoming signal, the receiver performs cross-correlation of N reference signature symbols with the incoming signal to identify the beginning of the preamble and post-amble. The output of the cross-correlation yields 2N peak values, from which the receiver chooses the sharpest and most symmetric for synchronization to the beginning of the frame. Empirical experiments confirm a synchronization accuracy within 5 ms in air within a range of 10.5 m, and 11 ms in water within a range of 15 m.**

I. Introduction

The availability of speakers and microphones in embedded and mobile devices, such as cell phones, PDA's, and sensor modules, makes acoustic communications a viable alternative to radio frequency (RF) in low bit rate applications, such as ubiquitous computing [1, 2] and underwater networking [3–5]. Acoustic hardware built-in to devices can be complemented with software modems that modulate acoustic signals with digital data to establish communication links [4, 5].

Essential to acoustic communication with software modems is the ability of the receiver to synchronize to the first symbol of an incoming data stream. Traditional symbol synchronization approaches rely on the transmission of a predefined sequence of symbols, often referred to as a training sequence. The conventional approach makes two assumptions about the communication channel that do not hold for acoustic communications, namely: (1) a bit rate at least in the order of tens of kilobits per second; (2) a bit error rate (BER) in the order of 10^{-6} or lower. Software-driven acoustic communication, both aerial and underwater, supports lower bit rates that range between tens to hundreds of bits per second [4, 5, 9, 10]. In addition, the bit error rate of software-driven acoustic communication is several orders of magnitude higher than the radio frequency bit error rate. The higher bit error rate in acoustic communications tends to cause loss of training sequence symbols, preventing proper symbol synchronization. Furthermore, providing high redundancy in the training sequence to mitigate training symbol losses is not an option for the narrow usable bandwidth of software-driven acoustic communications.

This paper proposes a short signature synchronization symbol (S^4) method that is tailored for the low bandwidth and high BER of software-driven acoustic communications systems. The method uses the same signature symbol as a preamble and post-amble to a transmitted data stream. An acoustic receiver correlates to both the preamble and post-amble to determine which of the two has a better correlation peak quality. The receiver then selects the signature symbol with the higher correlation peak quality and synchronizes to that symbol.

The design of the signature symbol aims at a high probability of correlation at the receiver even in cases of partial loss of the symbol and at low probability of false synchronization with ambient noise or data symbols. To this end, the symbol features include two square waves, with each square wave transmitted at a predefined frequency, separated by a predetermined guard time. The use of two frequencies for the signature symbol mitigates the effects of frequency selective fading or interference. The signature symbol guard time duration is chosen so that it is not equal to, and not a constant multiple of, the inter-symbol guard times to avoid high correlation with data symbols. In sum, the signature symbol features that promote high correlation at the receiver despite potential signal losses are the *two frequencies*, the *duration of the square wave signals*, and the *guard time*.

This paper presents the empirical experiments to evaluate the synchronization accuracy of (S^4) in both aerial and underwater environments. The experiments use the Tmote Invent module [7] speakers as transmitters and a standard PC microphone as the receiver. The aerial experiments in an indoor environment yield a synchronization accuracy within 5 ms for distances up to 10.5 m, enabling the use of S^4

for synchronization in indoor acoustic location systems [1, 2] and ubiquitous computing applications [8]. Underwater experiments in the River Dodder reveal a high resilience of (S^4) to high interference aquatic environment with an average synchronization accuracy less than 11.7 ms within 15m., which motivate S^4's adoption in software-driven underwater acoustic sensor networks.

The remainder of the paper is organized as follows. Section III discusses the design details of the signature symbol. Section IV presents the performance evaluation experiments and results in both air and water. Section V discusses the results and sums up the paper.

II. Related Work

Recent research proposals have targeted the issue of synchronization in software-driven acoustic modems. Iltis et al. [9] propose an underwater acoustic telemetry modem that uses training symbols to perform symbol synchronization. Fu et. al [10] extend Iltis et al.'s work and consider software-driven acoustic modems that run, at least partially, on specialized transducers and DSP boards with a target bit rate of 161 bps. They propose the increase of redundancy of training symbols in order to reduce packets misalignments. For synchronization with two training symbols, their experiments yield synchronization shifts up to 40 ms in lab tests, 100 ms in hallway tests, and 10 ms in underwater tests at 12.7 m.

While increased redundancy of training symbols provides better synchronization, this technique is not well-suited for software modems running on off-the-shelf sensor modules. Sensor modules have lower processing power than dedicated DSP boards, so detecting more training symbols wastes more energy and time at the modules. Another issue with training symbols is that high BER situations, such as the case of software modems on off the shelf acoustic hardware, require an even higher degree of redundancy, resulting in less efficient bandwidth utilization and in energy consumption overhead. The S^4 method avoids redundant symbols by employing a single signature symbol (which limits processing requirements at the receiver to a single correlation) that has a short duration (which minimizes control overhead) and that is robust to partial losses due to high interference.

III. Signal Design

The proposed synchronization technique considers a signal composed of a preamble, data symbols, and a post-amble. The preamble and post-amble have the same structure, combining two square waves of frequencies f_1 and f_2, and separated by a silence of duration T. Figure 1 shows the synchronization symbol structure, as generated at a PC. Figure 2 shows the structure of the synchronization symbol generated by the Tmote Invent speaker, which is not as symmetric as the same symbol generated by the PC, due to the hardware imperfections of the mote speaker. Furthermore, the Tmote Invent speaker generates non-deterministic variations in signals captured in different media (e.g. air, water, etc..). These

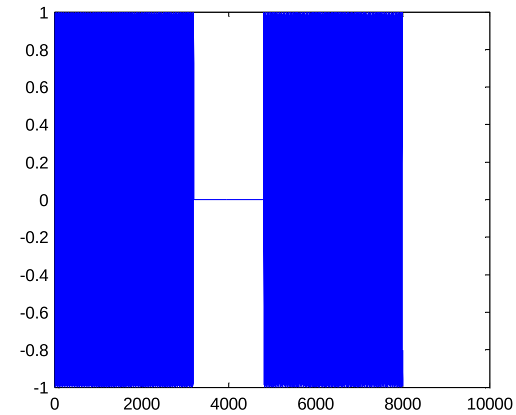

Fig. 1. Synchronization symbol structure as produced at the PC

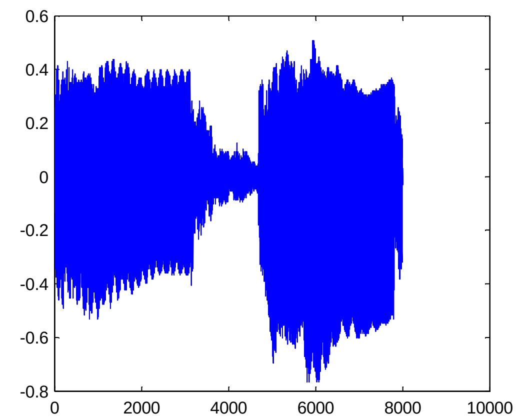

Fig. 2. First reference synchronization symbol captured from Tmote Invent speaker

properties of the Tmote Invent speakers have dictated the following signal design choices:

1) The signature of the preamble and post-amble symbol cannot rely solely on amplitude. As a result, the signal signature includes the two square waves of predetermined frequencies, the duration of the square wave signals, and a predetermined guard time between them. This signal shape allows the correlation function at the receiver to detect the surge at the beginning of the first square signal, the drop at the end of the first square signal, the surge at the beginning of the second square signal, and the drop at the end of the second the square signal. In addition, the synchronization technique selects a guard time duration that is not a constant multiple of the the inter-symbol guard times to avoid any correlation or misalignments with data symbol sequences.
2) The non-deterministic variation in signal amplitude within different media motivates matching the incoming signal to more than one reference signal at the receiver. The receiver can then correlate each of the N reference signals with the incoming signal, select the reference signal with the best correlation to the received signal, and synchronize to the sample with the highest cor-

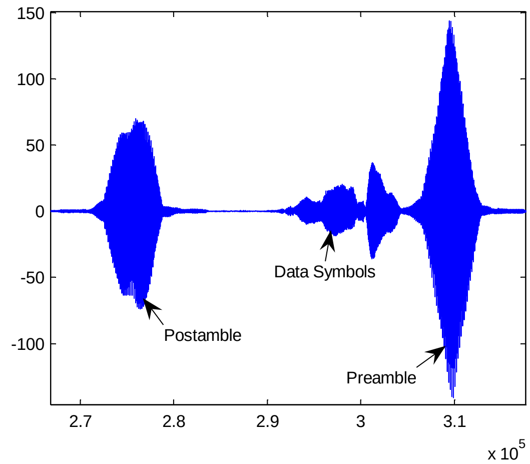

Fig. 3. Correlation output with the first reference signal

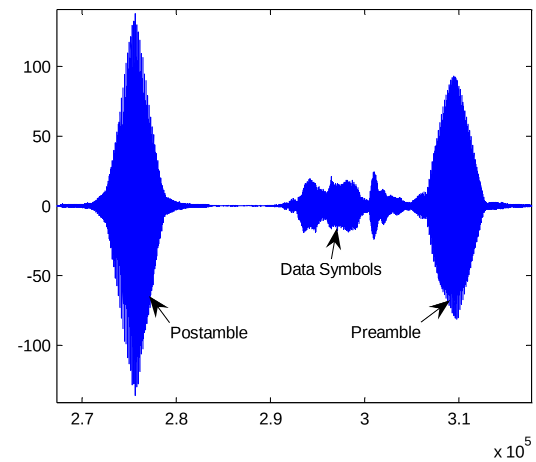

Fig. 4. Correlation output with the second reference signal

relation between the selected reference signal and the received signal. Each of N reference signals are captured from within the target deployment area. For example, prior to deploying the nodes in a river, a calibration phase is performed during which N reference signals are transmitted in the river and stored at the receiver, providing better correlation.

We illustrate the receiver synchronization process through an example with $N = 2$. Upon receiving an incoming signal, the algorithm correlates the received signal with the two reference signals. In our current example, Figures 3 and 4 represent the correlation outputs of the received signal with the two reference signals, yielding 4 correlation peaks: 2 peaks at the preamble, and 2 peaks post-amble. Visual inspection of the 4 peaks in these figures reveals that the preamble in the first correlation output and the post-amble in the second correlation output have the sharper peaks, but it does not indicate which of the 2 sharper peaks is a better reference for synchronization. The synchronization algorithm stores the amplitude of each peak a_i and the sample corresponding to each peak t_i, where i is the peak index. The algorithm then checks each peak's properties in order to determine the best peak to consider. First, the synchronization algorithm isolates each peak by storing P/2 signal samples around each peak into a vector S_i, where P is number of samples in each square signal of the preamble and post-amble. Next, the algorithm computes the envelope of the signal around each peak through the following expression:

$$Q_i = Hilbert(P_i) \tag{1}$$
$$R_i = Q_i * conj(Q_i)$$

which first takes the Hilbert transform of each peak and then retains the real part of the transform. The slope of the envelope around each peak is then computed through the following expressions:

$$R_i^1(i) = dR_i/dt \tag{2}$$
$$LR_i = average[R_i^1(t_i - \alpha)]$$
$$LF_i = average[R_i^1(\alpha - t_i)]$$

where LR_i represents the average slope of the envelope prior to peak i, LF_i represents the average slope of the envelope after peak i, and α is a constant that specifies the number of samples to check around the peak. The receiver then estimates the sharpness of the peak PS_i through the following equation:

$$PS_i = \frac{a_i}{average(S_i)} \tag{3}$$

which is essentially the ratio of the peak amplitude to the average amplitude of the samples in S_i. The variable PS_i is proportional to the peak sharpness, since the higher the value of PS_i, the sharper the peak i. Next, the receiver computes the symmetry of each peak through the following equation:

$$SYM_i = \frac{LR_i + LF_i}{LR_i} \tag{4}$$

The SYM_i variable is inversely proportional to the symmetry of a peak i. In the extreme case, LR_i and $-LF_i$ are equal and SYM_i is zero. An ideal peak has a high degree of sharpness and a high degree of symmetry, so the peak quality can be quantified by the following expression:

$$PQ_i = \frac{PS_i}{SYM_i} \tag{5}$$

After determining each PQ_i, the algorithm selects the peak with the highest peak quality and uses the peak sample t_i to synchronize to the received signal. In case t_i corresponds to the post-amble, the algorithm subtracts the fixed frame length from t_i to compute the start of the data symbols.

In short, upon receiving an acoustic signal through its microphone, the receiver executes the following steps in the S^4 algorithm:

Correlate signal to each of the 2 reference signals
ForEach: Peak i
  Record t_i and a_i
  Store S_i
  Determine envelope
  Compute R_i^1
  Compute LR_i and LF_i
  Compute PS_i,SYM_i, and PQ_i
Select Peak m with highest PQ_i
Synchronize to Peak m

IV. Empirical Experiments

This section presents the empirical experiments in air and water in order to evaluate S^4's synchronization accuracy. All of the experiments use $f_1 = 800\ Hz$ and $f_2 = 900\ Hz$, with the duration of 200ms for each of the two square waves in the S^4 symbol, and the guard time duration is 100 ms.

A. Synchronization for Aerial Acoustic Transmissions

The first experiment set evaluates the S^4 method's synchronization accuracy in an aerial environment. In these experiments, the Tmote Invent speaker is the transmitter of an acoustic signal that is received by a Dell XPS M1210 laptop with a built-in microphone. The aerial experiment set covers a transmission range between 1.5m and 10.5m at 1.5m increments within an indoor Line-Of-Sight (LOS) environment. Experiments for each distance were conducted 4 times.

Figure 5 illustrates the results of the aerial acoustic experiment set, plotting the synchronization shift versus the transmission distance. The synchronization shift, measured in milliseconds, is the difference in time between the S^4 algorithm output time sample and the real starting sample of the received signal. Each point in Figure 5 represents the synchronization shift of a single transmission instance. The continuous line plots the average synchronization shift.

The synchronization shift for all of the experiments ranges between 0 and 4.7 ms. The repetition of experiments at the same distance exhibit limited variance, with the lowest standard deviation of 0.2 ms observed at 2m and the highest standard deviation of 1.4 ms at 6m. The average synchronization shift, which ranges between 1 and 3 ms, does not appear to depend on transmission distance at distances from 4.5m and up. The relatively small synchronization shift for the aerial experiments is encouraging for the adoption of S^4 in indoor acoustic location systems and ubiquitous computing applications. Additionally, the aerial experiments serve as an initial validation of S^4 in a controlled environment prior to testing in harsher real-world settings.

B. Synchronization for Underwater Acoustic Transmissions

The next experiment set evaluates the accuracy of S^4 in an aquatic environment, in particular, in the River Dodder, which is in the vicinity of University College Dublin. Figure 6 shows the location of the testing site for these experiments. The tests in the river present several practical real-world challenges to underwater communication, namely:

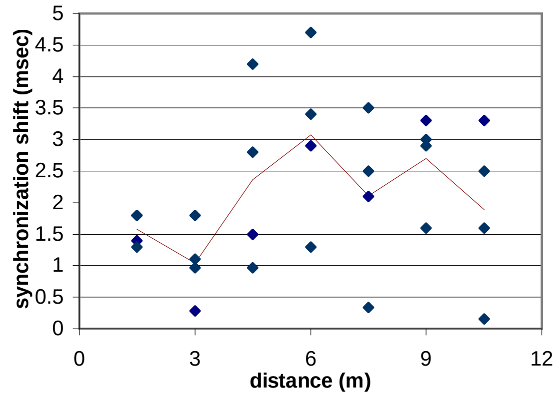

Fig. 5. The receiver symbol synchronization shift for acoustic signal transmitted in air by the Tmote Invent speaker.

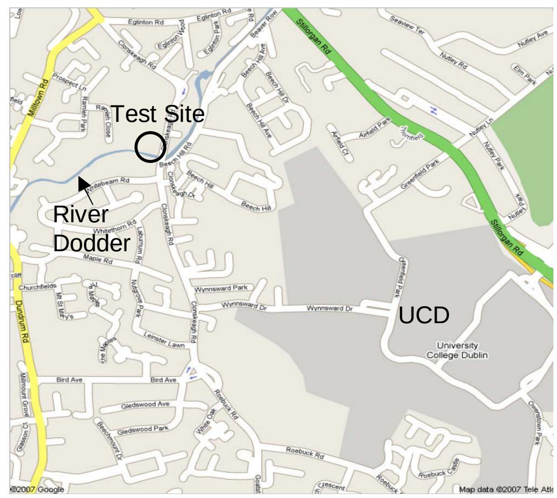

Fig. 6. Location of testing site. (courtesy of Google Maps)

- The river exhibited medium current strength at the time of testing, constituting a high ambient noise environment.
- At the test site, the river contains floating tree branches, plants, and man-made garbage. The presence of these objects in the river results in a high multi-path environment.
- Noise sources external to the river include ducks swimming in the vicinity of the test site, and cars passing by, causing potential vibrations in the water underneath.

The underwater experiments use the same setup as the aerial experiments, with the addition of a Labtec PC microphone to avoid immersing the laptop computer in the water. Latex membranes waterproof the microphone and the Tmote Invent module that are placed in the water. The microphone remains connected to a laptop outside the water at the bank of the river. The experiments evaluate the synchronization accuracy of S^4 at transmission distances between 1.5m and 15m at increments of 1.5m in a real world environment. At each distance, the experiments were repeated four times.

Figure 7 plots the results of the underwater synchronization experiments. Initial observation of Figure 7 reveals a steadily increasing trend in synchronization shift between 4.5m and

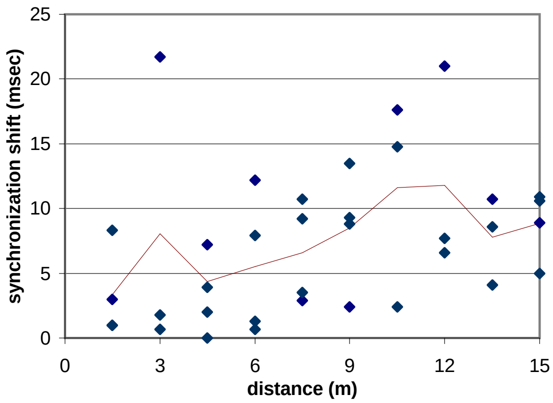

Fig. 7. The receiver symbol synchronization shift for acoustic signal transmitted in the River Dodder by the Tmote Invent speaker.

10.5m. At 12m, the synchronization shift stabilizes, only to drop again at 13.5m and 15m. The drop in synchronization shift above 12m indicates that the physical testing site is the primary source of synchronization shift rather than distance. As a result, we conclude that the steady increase in synchronization shift between 4.5m and 10.5m was caused by the test site topology, namely a clear bend in the river path that is most pronounced between 4.5m and 10.5m. The bend in the river path changes the signal propagation characteristics, increasing multiple path propagation that produces a shadowing effect at the receiver and reduces synchronization accuracy.

Another observation during data analysis is that S^4 is robust to partial losses in the preamble and post-amble signals. In several instances, the strong current in the river caused large impulses in the received signal at the microphone. The occurrence of large impulses during reception of a preamble or post-amble leads to partial or total loss of that signature symbol. S^4 proved capable of synchronizing to the partially intact signature symbols, validating its robustness in harsh underwater environments.

The synchronization shifts in Figure 7 are notably higher than in the aerial experiments, with the average synchronization shift ranging between 3.25 ms at 1.5 m to 11.77 ms at 12 m. The higher synchronization shift in the underwater case is attributed to the high noise environment (due to currents, birds, passing cars). The variance in synchronization accuracy, highlighted by the outliers at 4.5m and 12m, is due to changes in the river current that slightly shift the Tmote Invent module while transmitting, leading to changes in the signal propagation along new paths.

V. Discussion

This paper has presented a short signature synchronization symbol (S^4) method for aerial and underwater acoustic communications. The main contribution of S^4 is the replacement of the traditional training symbol synchronization method with short signature symbols to which the receiver can reliably align itself while avoiding correlation with noise or data symbols. The performance evaluation of S^4 has included aerial indoor experiments and underwater experiments in the River Dodder. The aerial experiments have shown an average synchronization shift that is less than 5 ms for distances between 1.5m and 10.5m. The relatively small synchronization shift for aerial acoustic transmissions render S^4 an suitable symbol synchronization mechanism for acoustic indoor location systems [1, 2] and ubiquitous computing applications [8].

Underwater experiments in the higher noise environment at the River Dodder test site has shown that the synchronization shift is relatively independent of distance. The higher mean and variance of the synchronization shift in the underwater experiments motivates the adoption of adaptive fidelity data communication. High current or high noise situations cause a drop in signal quality, affecting both synchronization accuracy and data transfer capability. Assuming the synchronization accuracy drops to 22 ms (the largest recorded value in the River Dodder experiments) for high noise environments, the receiver can request a decrease in data transfer rate to keep up with the higher noise environment. The decrease in data transfer rate corresponds to an increase in data symbol duration, which makes the data symbols less vulnerable to larger variance in synchronization shifts.

This paper has used two reference signals for the S^4 algorithm. An interesting direction for future work include the investigation of the effect of the number of stored reference signals N on the synchronization accuracy.

The S^4 mechanism is adaptive to suit the requirements of the deployment environment. Our brief tests of S^4 in bodies of water that include the Ranelagh Gardens pond, the Grand Canal in Dublin, and the UCD pond have indicated that the noise and signal propagation profile of each of these environments differ significantly due to the water conditions and the surroundings. Providing the receiver with a sample received signature symbol from the deployment environment in question enables precise correlation at the receiver, highlighting the flexibility of software-driven symbol synchronization, modulation, and communication.

References

[1] R. Jurdak, C.V. Lopes, and P. Baldi. "An Acoustic Identification Scheme of Location Systems," In Proc. ICPS'04, Beirut, Lebanon. 2004.

[2] A. Mandal, C.V. Lopes, T. Givargis, A. Haghighat, R. Jurdak and P. Baldi, "Beep: 3D Indoor Positioning Using Audible Sound," In proc. *IEEE CCNC,* Jan. 2005.

[3] R. Jurdak, C. V. Lopes and P. Baldi. "Battery Lifetime Estimation and Optimization for Underwater Sensor Networks," *Sensor Network Operations*, Wiley-IEEE Press, pp. 397-420. May, 2006.

[4] R. Jurdak, C.V. Lopes, and P. Baldi. "Software Acoustic Modems for Short Range Mote-based Underwater Sensor Networks," Proc. *IEEE/OES Oceans*. May, 2006.

[5] R. Jurdak, P.M.Q. Aguiar, P. Baldi, and C.V. Lopes. "Software Modems for Underwater Sensor Networks," Proc. *IEEE/OES Oceans*. June, 2007.

[6] California Sustainable Wetland/Watershed Information Manager. available at: www.calswim.org

[7] MoteIV Corporation available at: www.moteiv.com

[8] C. V. Lopes and P. Aguiar. "Acoustic Modems for Ubiquitous Computing," *IEEE Per. Comp.,* Vol. 2, No. 3, pp 62-71, 2003.

[9] R. A. Iltis et al. "An Underwater Acoustic Telemetry Modem for Eco-Sensing," In proc. *MTS/IEEE Ocean'05*, September, 2005.

[10] T. Fu et al. "Design and Development of a Software-Defined Underwater Acoustic Modem for Sensor Networks for Environmental and Ecological Research," In proc. *MTS/IEEE Oceans*, September, 2006.