Using Orthogonal Chirps Underwater for In-Band, Full-Duplex Communication with Minimal Self-Interference Cancellation

Bryson Schiel
schielb@byu.edu
Brigham Young University
Electrical and Computer Engineering
Provo, Utah, USA

Joshua Montierth
jmontie3@byu.edu
Brigham Young University
Electrical and Computer Engineering
Provo, Utah, USA

Eli Blattner
bla19012@byui.edu
Brigham Young University
Electrical and Computer Engineering
Provo, Utah, USA

Corey E. Dobbs
cedobbs@byu.edu
Brigham Young University
Physics and Astronomy
Provo, Utah, USA

Tracianne B. Nielsen
traci.neilsen@byu.edu
Brigham Young University
Physics and Astronomy
Provo, Utah, USA

Philip Lundrigan
lundrigan@byu.edu
Brigham Young University
Electrical and Computer Engineering
Provo, Utah, USA

ABSTRACT

Underwater acoustic networks (UANs) have the potential to benefit greatly from advances in in-band full-duplex (IBFD) communication, as this would reduce packet collisions and timing overhead from link layer protocols. A major direction of research to achieve IBFD has been through self-interference cancellation (SIC), a process that tries to minimize the effect of self-interference (SI) caused by a device's own transmission when it is trying to receive a packet from another device. Active SIC techniques, however, can be costly in computation and energy resources. The chirp spread spectrum (CSS) modulation scheme, which already provides noise- and multipath-resistant communication in UANs, can also provide a novel solution for simplified IBFD communication. In this paper, we use orthogonal CSS to enable IBFD communication that requires absolutely no SIC filtering when the signal of interest (SOI) is within -20 dB of the power of the self-interference (SI). For situations where the SOI is weaker than -20 dB of the power of the SI, we also provide a computationally light SIC filter that maintains clear communication.

KEYWORDS

In-band full-duplex, Chirp spread spectrum, Orthogonal CSS, Underwater acoustic network, Self-interference cancellation

ACM Reference Format:
Bryson Schiel, Joshua Montierth, Eli Blattner, Corey E. Dobbs, Tracianne B. Nielsen, and Philip Lundrigan. 2024. Using Orthogonal Chirps Underwater for In-Band, Full-Duplex Communication with Minimal Self-Interference Cancellation. In *The 18th ACM International Conference on Underwater Networks & Systems (WUWNET '24), October 28–31, 2024, Sibenik, Croatia.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3699432.3699464

1 INTRODUCTION

As underwater acoustic networks (UANs) continue to mature, their need for higher data rates and transmission fidelity increases, even as bandwidth and range limitations remain fixed. To compensate for these limitations, research is turning to in-band full-duplex (IBFD) communication to increase network efficiency. The ability for devices to communicate asynchronously on a shared part of the acoustic spectrum could double the rate of communication and eliminate time delays from link layer protocols, delays that are especially costly in underwater environments.

IBFD, however, is hindered in large part by self-interference (SI), the phenomenon where a device's own transmitter signal saturates its receiver, interfering with a signal of interest (SOI) from another device. Self-interference cancellation (SIC), therefore, has become a major focus of recent research into IBFD in UANs. Some work has focused on applying noise filters, either to the isolated SIC [4] or to the combination of SI and other noises [6]. Other work has attempted to use different modulation schemes for the sender and receiver [11] or even add channel estimation improvement to enhance SIC filtering [15]. Others have tried to use recursive methods such as recursive least squares as part of their SIC filtering [8, 9, 12].

Many of these SIC techniques, however, can add costly computation to the receiving device, and they often run for the entire time that SI is present. In this work, we explore how these issues of SIC can be offset. In particular, we focus on applying a version of the chirp spread spectrum (CSS) modulation scheme. CSS has an established history in terrestrial networks such as LoRa [16], and it has been proven to be an effective means of long distance, noise-tolerant communication. CSS has thus gained attention in the UAN space, and several studies have already explored its effectiveness in this environment. These studies have compared it to other underwater protocols such as frequency shift keying [13], a variant of orthogonal frequency division multiplexing [3], and even the JANUS protocol [14]. In each of these studies, CSS proved its strength as being resilient to harsh multipath conditions and achieving comparable transmission rates in normal conditions.

In this paper, we introduce the fundamentals of CSS, as well as those of orthogonal chirps [18]. Orthogonal CSS in particular provides unique ways of modulating signals between two devices that prevent the SI signal of one device from impacting its reception of the signal of another device. We show through experimental tank

WUWNET '24, October 28–31, 2024, Sibenik, Croatia

ACM ISBN 979-8-4007-1160-2/24/10
https://doi.org/10.1145/3699432.3699464

tests that this unique concept in CSS enables IBFD communication with minimal SIC filtering required.

1.1 Contributions

The contributions of this work are as follows:

(1) We use orthogonal CSS to enable IBFD communication **that requires no SIC filtering** when the power of the signal of interest (SOI) is within -20 dB of the power of the SI.
(2) For scenarios where the power of the SOI is less than a -20 dB the power of the SI, we introduce a **computationally light SIC filter** that is only required until a very small part of a packet in the SOI is detected, after which orthogonality alone facilitates IBFD communication again.

This paper is organized as follows. Section 2 explains some of the mathematical fundamentals of standard CSS modulation, while Section 3 explores how CSS can specifically be geared towards IBFD—primarily using orthogonal chirps, but also using SIC filtering in extreme cases. In Section 4, we present an experimental setup in a tank environment, and our results are presented in Section 5. Section 6 discusses this research and how it can be adapted in future studies, and finally Section 7 concludes the paper.

2 FUNDAMENTALS OF CSS COMMUNICATION

Over the course of this section, we explain already-established basics of chirp spread spectrum (CSS) modulation according to LoRa techniques, which can be further reviewed at [7]. These basics establish how CSS can be used for half-duplex communication, and in the next section we expand on these basics to build a system that enables full-duplex communication.

A basic linear chirp signal is a modified sinusoidal wave,

$$x(t) = cos(\phi(t)) \tag{1}$$

where $x(t)$ corresponds to the acoustic pressure on a hydrophone, and $\phi(t)$ is the phase of the signal at any given time t. In a chirp, $\phi(t)$ is characterized by sweeping across a frequency band with an initial frequency (f_0) and an ending frequency (f_1) over some duration of time (T). When f_0 is less than f_1, this is reflected as a linear increase in frequency, which is referred to as an up chirp.

$$\phi(t) = f_0 t + \frac{(f_1 - f_0)}{2T} t^2 \tag{2}$$

We use this $\phi(t)$ in Equation (1) and put our signal into complex exponential form to get our base chirp for the paper:

$$x(t) = e^{2\pi j[\phi(t)]} \tag{3}$$

where 2π is inserted into the exponential because we are measuring angular velocity in Hz with f rather than radians per second with ω, and $\omega = 2\pi f$. An example of an up chirp is shown in Figure 1.

2.1 Modulo Chirps

One way to encode data into a chirp signal is through a cyclic shift in frequency or time [10, 16]. Rather than starting an up chirp with a cosine wave at f_0, the beginning frequency is some intermediate value f_i where $f_0 < f_i \leq f_1$. Starting at f_i, the frequency increases linearly, as shown before, until it reaches f_1; at this point the frequency changes to f_0, creating a discontinuity, and resumes its linear increase until returning to the starting frequency f_i. Since this acts similarly to the modulo operator in mathematics, we refer to this behavior as a "modulo chirp" (in other literature, it might be known as "frequency shift modulation" [16] or a "temporal cyclic shift" [10]; for brevity, we use "modulo").

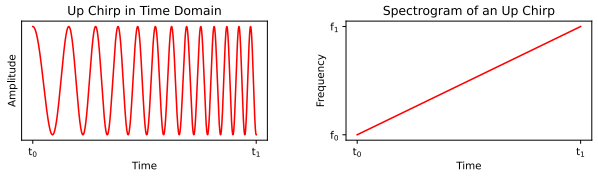

Figure 1: An up chirp signal in the time domain (left) and in time against frequency, in the form of a spectrogram (right).

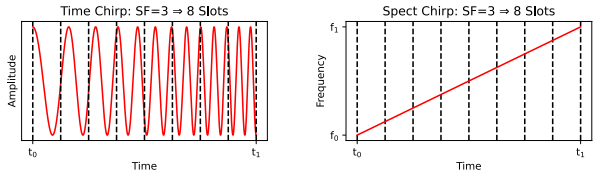

Figure 2: A chirp with a spreading factor of 3; set into 8 different starting slots, equidistant in both frequency and time.

Most chirp applications limit the number of distinct f_i starting frequencies to a power of 2, to facilitate binary encoding. This exponent of 2 is referred to in literature as the spreading factor (SF) [16], and it allows for 2^{SF} distinct starting frequencies. With more starting frequencies, the time for a single chirp must also increase. Thus, the spreading factor determines the chirp time and power; an increase in spreading factor increases both of these properties by a factor of 2.

The modulo scheme provides a robust way to encode binary data into a base chirp. For example, to encode a symbol of three bits of data into a single chirp, we set SF = 3 which gives us $2^3 = 8$ possible starting frequencies. An example of this is found in Figure 2[1].

For example, to encode the number 3 (011) in a chirp, the signal begins at the third frequency interval (f_{i3}), whereas encoding a 5 (101) begins at the fifth frequency interval (f_{i5}). In both cases, signal increases from f_i to f_1, wraps around to f_0, and increases back to the stating frequency f_i. The reference point in time that corresponds to the wrap-around discontinuity is denoted as s; as the encoded number increases, the s point occurs sooner (hence, s_5 would be earlier in time than s_3). All of these features and the difference between encoding a 3 and a 5 can be seen in Figure 3.

To encode this signal correctly in reference to the time $t = s$, we modify our signal equation in (3) as:

$$x(t,i) = \begin{cases} e^{2\pi j[\phi(s_i - t)]}, & t_0 < t \leq s_i \\ e^{2\pi j[\phi(t - s_i)]}, & s_i < t \leq t_1 \end{cases} \tag{4}$$

[1]Figure 2 and others show this segmentation in the time axis rather than frequency; due to the linear nature of chirps, however, equal divisions in time correspond to equal divisions of frequency. Segmenting in time is especially easier to visualize and track in later figures.

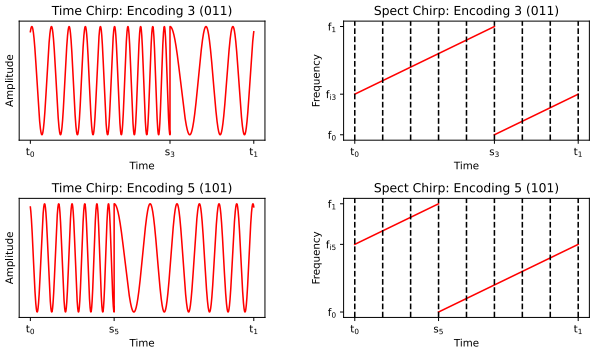

Figure 3: Two 3-bit modulo chirps, one encoding a 3 (top) and the other encoding a 5 (bottom).

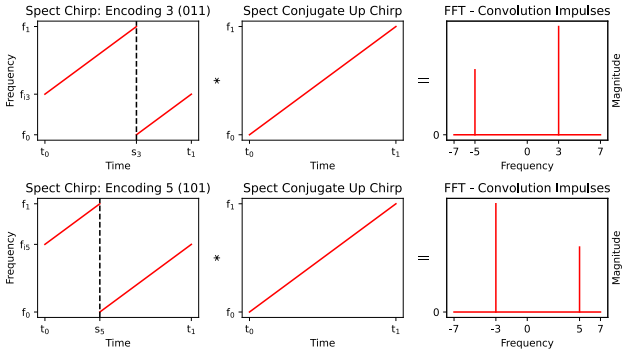

Figure 4: A modulo chirp convolved in the frequency domain with a conjugate chirp produces two tones of constant frequencies, presented in the Fourier domain as an impulses at the frequencies i and $(i - 2^{\mathrm{SF}})$. This is shown where $i = 3$ (top) and $i = 5$ (bottom).

where s_i in terms of the encoded value i is

$$s_i = \frac{t_1 - t_0}{2^{\mathrm{SF}}} i, \quad i \in [0, 2^{\mathrm{SF}} - 1]$$

It should also be noted that we are adding this value i as an input to our chirp function.

From a communications standpoint, the frequency discontinuity at time s_i is how the transmitted data is encoded. Identifying s_i relative to the start of the chirp forms the basis for demodulation.

2.2 Demodulating a Modulo Chirp

The process of identifying the discontinuity in a modulo chirp can be accomplished deterministically by convolving this frequency domain signal with another signal, the complex conjugate of the base up chirp.

The equation for this conjugate chirp is

$$x^*(t) = e^{-2\pi j[\phi(t)]} \tag{5}$$

As discussed in [16], this function, if convolved with the modulo chirp, produces a signal of two principle frequencies: a positive frequency from $t_0 < t \leq s_i$ and a negative frequency from $s_i < t \leq t_1$. In the frequency domain, this result is reflected as two Dirac delta-like impulses, one at the positive frequency i, and the other at its negative frequency counterpart $i - 2^{\mathrm{SF}}$. An example is provided in Figure 4.

Since convolution in the frequency domain is equivalent to multiplication in the time domain, we can avoid a very costly convolution and instead multiply equations (4) and (5) and take the discrete Fourier transform ($\overset{\mathcal{F}}{\longrightarrow}$) of the resultant signal. The transform of the product function is simply a pair of delta-like impulses, one centered on i and another on the negative frequency counterpart $i - 2^{\mathrm{SF}}$.

The entire demodulation process is expressed as:

$$x(t,i) \times x^*(t) = m(t) \approx \begin{cases} e^{2\pi j[i]t}, & t_0 < t \leq s_i \\ e^{-2\pi j[i-2^{\mathrm{SF}}]t}, & s_i < t \leq t_1 \end{cases} \tag{6}$$

$$m(t) \overset{\mathcal{F}}{\longrightarrow} M(f) \approx \delta(i) + \delta(i - 2^{\mathrm{SF}}) \tag{7}$$

This process of multiplying the signals and taking the Fourier transform is akin to finding the convolution between these two signals, thus creating a matched filter. The temporal cyclic shift in the modulo chirp creates a constant frequency difference which can be readily measured during the entire duration of the signal.

Taking an argmax of this $M(f)$ function can then reveal the encoded i value. This argmax is bounded between 0 and the maximum value that i can be, which is $2^{\mathrm{SF}} - 1$. A receiver's estimate of i, denoted as $\hat{\imath}$, is thus calculated between these two values[2].

$$\hat{\imath} = \mathrm{argmax}(M(f)), \quad f \in [0, 2^{\mathrm{SF}} - 1] \tag{8}$$

We show this process of up chirp modulation and demodulation between two devices in the format of a block diagram in Figure 5.

2.3 Chirp Synchronization

An essential part of chirp demodulation is knowing the start time of the chirp, referred to previously as t_0. Without time synchronization, the demodulation of the chirp $x(t, i)$ would produce a clear and distinct value, but it would be incorrect, as the perceived frequency discontinuity is shifted in time. To account for this, we use a sync word, $\psi(t)$, which is a chirp where $i = 0$ and whose spreading factor is greater than the rest of the chirps in the message (we use $SF + 3$, increasing both the clarity and power of this signal). $\psi(t)$ is the first chirp sent in a series of chirps $x(t, i)$ to compose a packet (more on this in Section 3.2).

The receiving device monitors the incoming signal and cross-correlates it with a stored copy of the sync word $\psi(t)$. When an incoming packet, which starts with this sync word, arrives at the device, the device is able to identify the start time t_0 of the message.

Everything established thus far allows for the modulation and demodulation of a chirp packet between two devices in a half-duplex

[2]Due to properties of the discrete Fourier transform, the largest impulse in the frequency domain can occur in the range $[K - 2^{\mathrm{SF}}, K]$, where K is the total number of discrete samples collected. Since negative frequencies are aliased as high frequencies, the negative frequency impulse is present here. When i is greater than 2^{SF-1} (i.e. halfway between 0 and 2^{SF}), the spike in this higher band actually has more power than the spike in the lower band, because the negative frequency is seen for a longer portion of the chirp time. While this changes the best way to interpret $\hat{\imath}$ slightly, it is nevertheless a simple, deterministic process to calculate $\hat{\imath}$: there is a known set of frequencies in which the impulse spike occurs, and any spikes outside those frequencies can safely be ignored for ease of computation.

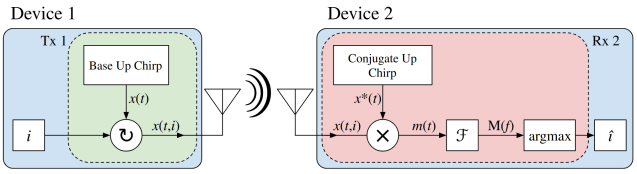

Figure 5: A block diagram showing the modulation (left) and demodulation (right) process for sending a single up chirp from Device 1 to Device 2 in a UAN. The "↻" symbol indicates the temporal shift that occurs when a modulo chirp encodes the data i.

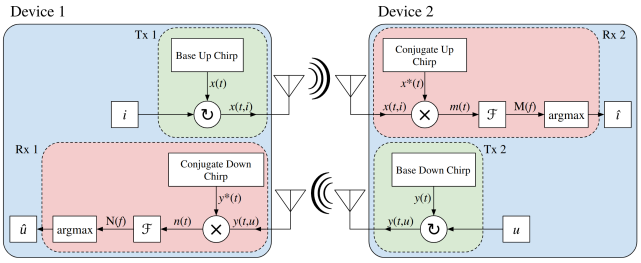

Figure 6: A block diagram showing the two-way modulation and demodulation process for sending a single up chirp from Device 1 to Device 2 in a UAN, and a single down chirp from Device 2 to Device 1.

configuration. In the following section, we take these principles and expand on them to enable full-duplex communication.

3 CSS FOR IBFD

The goal of this paper is to introduce a method of IBFD using CSS. This becomes possible with chirp orthogonality [18].

3.1 Orthogonal Chirps

There are two primary types of linear chirps: those that chirp up (increasing in frequency) and those that chirp down (decreasing in frequency; $f_0 > f_1$).

The internals of Equation (2) do not change for down chirps, but for clarity, we refer to down chirp signals with the letter y. A base down chirp is $y(t)$, a modulo down chirp is $y(t, i)$, a conjugate down chirp is $y^*(t)$, their product is $n(t)$, and its Fourier transform is $N(f)$. Beyond that, however, the computation for demodulating a down chirp is analogous to that of demodulating an up chirp.[3]

A unique feature of up and down chirps is how they each respond to a convolution with the conjugate chirp of the other. As noted before, when we convolve a modulo up chirp with a conjugate up chirp, the spike in the frequency domain happens somewhere in the bounds of 0 and $2^{\mathrm{SF}} - 1$. However, the convolution of a down chirp with the conjugate up chirp is near zero for the entire frequency domain. The same happens with an up chirp and a conjugate down chirp—no spike is observed in the frequency domain. The presence of one class of chirp thus has a near-zero effect on the demodulation of the other. This phenomenon is often referred to as chirp orthogonality or quasi-orthogonality [18], and we identify it as an essential basis for IBFD communication in UANs.

In the context of the two underwater devices from Figure 5, we add a receiver onto Device 1 and a transmitter onto Device 2. This second pair of modules would transmit and receive with down chirps, whereas the first pair communicates with up chirps. This approach is shown in Figure 6. To clarify moving forward, we label the value being sent by Device 2 as u rather than i; Device 1's approximation of this value is $\hat{u}$.

In Section 5, it will be shown that these two pairs, one sending up chirps and the other sending down chirps, can asynchronously transmit their orthogonal chirps and still demodulate the incoming signal, despite the presence of SI. In that section, this holds true up to the point where the ratio between the powers of the remote transmitter and the local SI is as low as -20 dB. In essence, *orthogonal CSS provides a very clear basis for enabling IBFD communication in UANs with no SIC filtering required.*

3.2 Two-Way Packet Transfer

Now that IBFD with orthogonal chirps has been introduced, we wish to give a more explicit definition of how traffic might flow between Devices 1 and 2.

For the concept of a transmission, we define a packet to be a signal composed of a sync word and multiple chirps. This entire packet is referred to as $a(t)$, and it is composed symbolically as

$$a(t) = [\psi(t),\ x(t, i_0),\ x(t, i_1),\ \ldots,\ x(t, i_k)]$$

for a packet with the sync word followed by k chirps. This is the packet being transmitted by Device 1. Device 2, transmitting down chirps, would have a separate packet signal function, in this case referred to as $b(t)$.

We then define the asynchronous combination of the two packet signals as $z(t)$, where

$$z(t) = a(t) + b(t)$$

We refer to the perceived power of a packet on a device's receiver as $P(\cdot)$. In Device 2's frame of reference, the signal from Device 1 is attenuated due to the distance between them; thus, we can expect that $P(b(t)) > P(a(t))$.

Orthogonal CSS alone will be shown to handle IBFD communication when this ratio of $P(a(t)) : P(b(t))$ is -20 dB, but for even weaker signals, a simple SIC filter may become necessary for IBFD. The following subsection introduces a simple SIC filter that we use for these lower power ratios.

3.3 SIC Filtering for Improved IBFD in CSS

As distance between Device 1 and Device 2 increases, the power ratio between $a(t)$ and $b(t)$ decreases below -20 dB. At this point, the $b(t)$ portion of this signal generates noise interference in our cross-correlation results. The normal spike from cross-correlating the sync word in $a(t)$ with a stored copy of $\psi(t)$ on Device 2 may become occluded by this noise, such that synchronization becomes impossible.

[3]The spike in $N(f)$ actually occurs at $2^{\mathrm{SF}} - 1 - i$. When estimating the value of i, we use $\hat{i} = 2^{\mathrm{SF}} - 1 - \mathrm{argmax}(N(f))$. The major principle stays the same, however: we can identify the encoded value i by looking for a spike in a known range and ignoring everything outside that range.

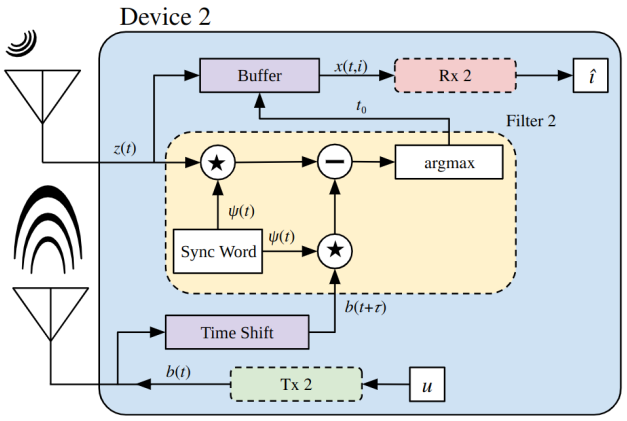

Figure 7: A block diagram showing the filter design to cancel out self-interference in full-duplex communication. The ⋆ symbol represents the cross-correlation operation.

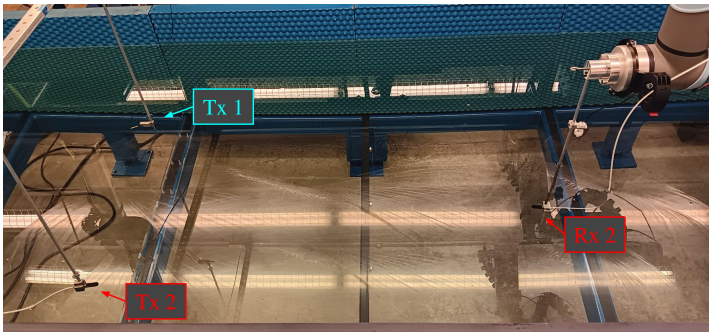

Figure 8: Our setup using hydroacoustic transceivers in a water tank. Device 2's receiver module is on the right, and its transmitter module is on the bottom left. Also on the left is Device 1's transmitter module.

As discussed previously, synchronization of a chirp is key to CSS demodulation. In the case of a severe power imbalance, SIC filtering becomes necessary. If SIC can be applied to $z(t)$ up until the sync word from $a(t)$ can be identified, Device 2's receiver can synchronize with the incoming $a(t)$ packet. After this point, chirp orthogonality can filter out the SI from device 2, and interpret the $x(t, i)$ chirps correctly without any more need of SIC filtering. Thus, SIC filtering is only required for a limited amount of time. Our work strives to do this by using a simple SIC filter design, shown in Figure 7.

The first step in the filter is to find the time lag τ that Device 2 experiences between sending an impulse and receiving the same impulse as self-noise. This step can be done well before communication between the two devices begins. Once τ is known, Device 2 begins sampling incoming signals and storing these samples in a buffer. There are two scenarios from this point.

Scenario 1: In the first scenario, Device 2 is not transmitting when $a(t)$ arrives, and a cross-correlation between the incoming samples and the sync chirp $\psi(t)$ reveals the start of $a(t)$ from Device 1. The time t_0 is determined within the buffer, and the subsequent samples are pushed to the receiver module.

Scenario 2: In the second scenario, Device 2 is transmitting first, and problems of self-interference during the sync word correlation arise. To address this, Device 2 creates a copy of the signal it is transmitting, $b(t)$, delayed by the lag time τ. Two cross-correlations are then calculated. The first is of this new signal, $b(t + \tau)$, with the sync word $\psi(t)$, which represents the contribution from the self-noise that will be present in the correlation of the incoming signal. The second is the correlation of the actual measured signal $z(t)$ with $\psi(t)$. We then subtract this first correlation from the second to remove any self-interference. Once done, any remaining spike in the correlation, even if at a very low power, indicates the presence of an incoming packet $a(t)$ as well as its associated start time t_0. t_0 is then determined within the buffer, and the subsequent samples are pushed to the receiver module.

Once t_0 is known, the receiver module is automatically synchronized with the start of the incoming chirps in the $a(t)$ packet. If, while buffering the samples, the correlation between $z(t)$ and the stored sync word $\psi(t)$ never passes a certain threshold, then samples in the buffer can be dropped, since it can be safely assumed that there is no attempted transmission located in them.

This design is similar to other SIC filter designs [4], but its mathematical complexity is reduced due to the simple mathematics of CSS. Additionally, this filter is only required up until the point that the packet $a(t)$ is synchronized with the first chirp, $\psi(t)$; after this point, orthogonal CSS is sufficient to decipher the incoming chirps $x(t, i)$. Other SIC filtering techniques, by contrast, must remain active the entire time that the SI is present. Thus, *our CSS SIC filter design is computationally light and is only needed for a fraction of the packet time when a remote packet signal is inbound.*

4 EXPERIMENTAL SETUP

In order to prove that chirp orthogonality and our self-interference filter are effective for IBFD, we generate, transmit, collect, and process signals in a tank test environment. The Hydroacoustics Lab at Brigham Young University has a water tank for producing and measuring signals in the water [17]. We use this tank and its acoustic transducers to test our method.

Figure 8 shows our setup in relation to the setup shown in Figure 7. The hydrophone on the right represents the Rx 2 module, and on the bottom left we have the Tx 2 module, transmitting down chirps; both of these transducers are the B&K Miniature Hydrophone Type 8103. In the top left is the Tx 1 module transmitting up chirps, in this case using a Teledyne Reson TC 4038 hydrophone[4].

As in Figure 7, we transmit a strong signal from Tx 2, representing the strong SI that Device 2 would receive from its own transmitter. A weaker signal is transmitted by Tx 1. Both Tx 1 and Tx 2 transmit chirps in the 20-25 kHz band. With each trial, we adjust the power level on the transmitters to make Tx 1 weaker and weaker compared to Tx 2, thereby mimicking the effect of greater and greater distances between Device 1 and Device 2.

[4]We acknowledge that the hydrophones for Tx 1 and Tx 2 are different. This is explained and discussed further in Section 6

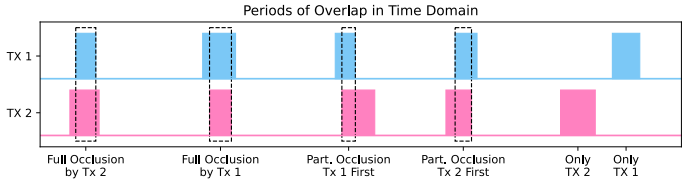

Figure 9: As Tx 1 and Tx 2 are transmitting, we set their signals to have very specific overlaps in time, giving us a broad range of full-duplex situations.

Custom LabVIEW software transmits pre-generated messages on Tx 1 and Tx 2 and records the incoming signal on Rx 2. For each trial, a single binary file is saved with the incoming signal strength on the Rx 2 hydrophone. This file includes the signal from both transmitters.

The Tx 1 transducer sends a signal containing multiple $a(t)$ packets, each one starting with the up chirp version of the sync word. Tx 2 transmits a signal that contains multiple $b(t)$ packets. We explicitly set these signals to have periods of overlap between the packets they are sending: some $b(t)$ packets start transmitting before the corresponding $a(t)$ packet arrives, and at other times the opposite occurs. This is done to help explore the two scenarios from Section 3.3. For control, we have individual packets in $a(t)$ and $b(t)$ that do not overlap at all. These transmitted packets and their periods of overlap are shown in Figure 9. The goal of this strict time setting is to test multiple asynchronous transmission situations that might arise during IBFD communication. As part of our results, we analyze how well each scenario handles full-duplex communication.

Because the Tx 1 and Tx 2 hydroacoustic transducers are equidistant to the Rx 2 hydrophone, their signals arrive as scheduled in Figure 9. This allows us to only need to change the timings in the transmission files rather than modify the positions of the transducers in the tank to specify the timings of packet overlap in the figure.

As noted before, chirp synchronization is essential to identifying the start of a chirp packet on $a(t)$. We test how well this is achieved using orthogonal CSS alone, and we compare that to the effect of adding the SIC filter. Since the Hydroacoustics Lab setup only allows for signal collection, we post-process the signals after collection on Rx 2. We run the analysis in Python on the saved binary file, once with the SIC filtering functions, and once without them. This allows us to run our post-process analyses on identical data, simulating reception with and without the SIC filter.

In order to test the effects of chirp orthogonality alone, we run the received $z(t)$ signal through our Rx 2 module from Figure 5. If a cross-correlation spike occurs with the stored sync word $\psi(t)$, we have synchronized a packet and can begin decoding the message.

In order to test the effect of our filter from Figure 7, we need to use a time-shifted version of our $b(t)$ signal as an input to the filter. Since we have the pre-generated signal from Tx 2, once we collect the signal on Rx 2, we apply our filter using a copy of our transmitted packets, shifted in time by τ. We take the cross-correlation of this signal, $b(t+\tau)$, with our stored sync word $\psi(t)$, and we routinely subtract it from the cross-correlation of the received $z(t)$ with $\psi(t)$. This removes the SI that Rx 2 experiences from Tx 2. Any time we then transmit a packet as part of the $a(t)$ signal on Tx 1, this produces a spike in the cross-correlation with the stored sync word, informing us of the start of a packet from Tx 1.

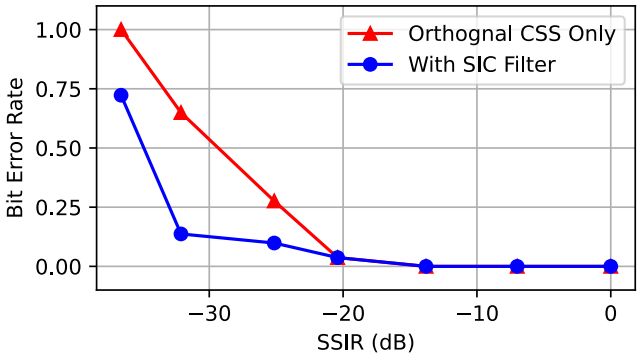

Figure 10: Post processing of our received signal on Rx 2. All measurements for each SSIR trial are averaged.

5 RESULTS

During our tank tests, we transmit Tx 1 and Tx 2 at seven different power ratios; for each power ratio trial, we transmit and record data eight distinct times. This ends up totaling about 2400 bits being transmitted by Tx 1 for each trial. We test at various voltage ratios between Tx 1 and Tx 2, starting by keeping them equal and then attempting to halve the voltage ratio for each stage of the experiment. In practice, however, these voltage ratios are not perfectly executed by the system. For accurate measurement, we record the average voltage for the duration of each non-overlapping packet in Tx 1 and Tx 2, get their voltage ratio, and convert it to a power ratio. **Our actual measured power ratios are 0, -7.0, -13.8, -20.4, -25.1, -32.1, and -36.6 dB.** Throughout this section, this ratio is referred to as the Signal-to-Self-Interference Ratio (SSIR).

Over the course of our experiment, we identify three different effects of chirp orthogonality and our SIC filter toward IBFD communication. These effects make up the bulk of this section.

5.1 Bit Error Rate

Using the collected data, we calculate the Bit Error Rate (BER) between bits we are able to interpret compared to bits we expect during the packet transmissions times for Tx 1. The comparison between our two post-processing methods over our range of SSIRs is found in Figure 10.

The results of this experiment clearly indicates the power of orthogonal CSS—without any SIC filtering whatsoever, it achieves an extremely low BER, even as the SSIR falls to -20 dB. Beneath the -20 dB mark, SIC filtering provides some benefits to the BER, but it also supports other aspects of IBFD communication, which we describe below.

	Orthog. Chirps Only		With SIC Filter	
SSIR (dB)	T+	F+	T+	F+
0	40	0	40	0
-7.0	40	0	40	0
-13.8	40	0	40	0
-20.4	40	0	40	0
-25.1	**24**	**14**	**40**	**0**
-32.1	**9**	**114**	**24**	**0**
-36.6	**0**	**127**	**11**	**8**

Table 1: The True Positive (T+) and False Positive (F+) counts for each trial in our experiment. As the SSIR decreases between trials, false spikes occur in the cross-correlation of $z(t)$ with the stored $\psi(t)$, causing false packets to be reported to Device 2. An ideal analysis should contain 40 True Positives and 0 False Positives; the bold-font rows indicate where this does not occur. Note the improved performance when adding the SIC filter.

	Orthog. Chirps Only		With SIC Filter	
SSIR (dB)	S1	S2	S1	S2
0	16	16	16	16
-7.0	16	16	16	16
-13.8	16	16	16	16
-20.4	16	16	16	16
-25.1	**8**	**8**	**16**	**16**
-32.1	**1**	**0**	**16**	**0**
-36.6	**0**	**0**	**2**	**0**

Table 2: A measurement of packets that are detected based on how they are classified using the scenarios from Section 3.3. An ideal analysis should contain 16 packets in each cell; the bold-font rows indicate where this does not occur. Note the improved performance when adding the SIC filter.

5.2 Appearance of False Packets

Over the course of transmitting these packets at different SSIRs, we identify an unexpected behavior at lower SSIRs. As noted before, the self-interference from `Tx 2` can produce some noise when being processed using the `Rx 2` module in Figure 6. Occasionally, when this noise is not filtered out, false positives occur when cross correlating $z(t)$ with the sync word $\psi(t)$. During the post-processing of our received signals, the `Rx 2` module may identify a correlation with the sync word $\psi(t)$ and indicate the start of a packet that is not actually being transmitted.

The count of True and False Positives is shown in Table 1. For the eight transmissions as part of each trial, five packets were sent from `Tx 1` for each transmission, so the ideal True Positive value should be 40. Additionally, the term "Positive" in the table indicates only that a packet has been detected by the presence of a sufficient spike in the cross-correlation of $z(t)$ with the sync word $\psi(t)$; in this table, a "True Positive" has no bearing on the BER of that packet in the analysis.

Similar to the behavior observed in the previous section, orthogonal CSS results in perfect packet identification when SSIR > -20 dB in our experiment. Below this power ratio threshold, however, adding the SIC filter provides clear benefits to performance as it removes almost all false positives that Device 2 might incorrectly detect. The combination of these two techniques can support IBFD communication in extreme UAN environments.

5.3 Scenario-Dependent Packet Identification

One major goal of our experiments was to see how behavior between the different scenarios listed in Section 3.3 might vary. In Figure 9, the second and third packets from `Tx 1` are received by `Rx 2` before the corresponding packets in `Tx 2` have started to be heard by the receiver. This matches the situation described by Scenario 1. In contrast, the first and fourth packets transmitted by `Tx 1` reach the receiver after the corresponding packets from `Tx 2`; this matches Scenario 2.

During the course of our experiments, we track which packets are dropped or received more consistently to see if a pattern emerges about an IBFD scenario that is more tolerant to SI. Table 2 shows how many packets were collected from Scenarios 1 and 2 as described previously. Once again, a correct packet identification has no bearing on the BER of that packet.

As before, we see orthogonal CSS perfectly identifying packets as the SSIR falls toward -20 dB. Beneath this threshold, the combination with the simple SIC filter helps maintain this performance.

It should be noted, however, that packets transmitted under the conditions of Scenario 2 are universally dropped after a certain SSIR threshold. This indicates that a perfectly asynchronous network of IBFD devices may not be achievable at these lower SSIR values using only the methods we test. UANs may have to adopt an alternate strategy if IBFD is to be attained at these lower SSIR values.

6 DISCUSSION AND FUTURE WORK

Our results indicate that orthogonal chirps provide a very strong basis for performing IBFD communication in UANs. In addition, they require no additional SIC computation when the power of the SOI is within -20 dB of the SI signal. Below this power level, a very simple and computationally light SIC filter can be applied to the receiver module for greatly enhanced performance. This performance is due in large part to the resilience of the CSS protocol in harsh communication environments and strongly indicates the need for more research on CSS in UANs and IBFD communication.

Future work will involve taking this technique outside of a lab setting. In Section 4, we indicate that different hydrophones are used for `Tx 1` and `Tx 2`, and this is due to limitations within the Hydroacoustics Lab setup itself: they currently only have the hardware to transmit on one of each of their hydrophones. Additionally, the tank, while having some soundproofing, still provides a harsh multipath environment for our messages. We determined during experimentation that our packets needed to be relatively short to avoid these severe multipath effects, and thus extra trials were run to compensate for the loss of data per packet. Future experiments will take place using software defined modems (SDMs), such as those presented in [1, 2, 5], which will use the same hydrophones between devices. Additionally, tests in a natural body of water, such

as a lake, will provide more realistic multipath scenarios, which will inform more realistic packet limitations.

Lake tests with equivalent hydroacoustic transducers will also allow us to see the effects of actual distance on power ratios in IBFD, rather than manually adjusting power levels to simulate distance. This will better inform how network throughput is affected by node separation and mobility.

Using SDMs will also allow for real-time analysis on the transmitted signals. Whereas signals were recorded in the lab and processed elsewhere, an SDM could both process the data and measure network throughput in real-time. An SDM will also generate and transmit packets automatically, creating a more natural traffic flow in the network and allowing for more randomization in how the IBFD scenarios are encountered in a live network.

Future experiments would also require a more-specialized analog-to-digital conversion (ADC) process designed for this full-duplex implementation. In the experiments shown, the ADC is performed by the built-in hardware of the Hydroacoustics Lab; tests on an SDM could use a custom analog implementation of our SIC filter to further improve signal reception and better inform the effect of orthogonal CSS on IBFD communication.

More work also needs to be done to study how chirps could be further modified to improve network performance. In this work, elements such as the chirp spreading factor were kept constant; in future tests, increasing the spreading factor might prove to make IBFD more resilient as the SSIR decreases. Alternatively, smaller spreading factors would decrease the time required for a chirp, increasing the data rate in the network, but potentially raising the SSIR threshold for accurate detection. This trade-off between transmission speed and signal clarity can be further studied.

7 CONCLUSION

This paper presents a new avenue for achieving IBFD communication is UANs. Orthogonal CSS innately filters out a device's own signal, and our in-tank experiments indicate that it can operate with no separate SIC computation up to an SSIR of -20 dB. Even when the SSIR decreases below this threshold, the process of adding SIC is computationally simple and lasts only until a packet sync word symbol is detected, after which orthogonality can take over for enabling IBFD.

As this techniques gains further study, it has the potential to enable robust IBFD communication in all manner of UANs, from stationary networks to fleets of underwater vehicles. The doubling of bandwidth that IBFD offers could lead to many new innovations in underwater communications, with applications in research, industry, and military fields. Orthogonal CSS provides an extremely viable option to achieve this.

ACKNOWLEDGMENTS

This work was funded in part by the National Science Foundation award #2322058. We also express gratitude to undergraduate students Madilyn Moore and Natalie Bickmore in the Hydroacoustics Lab for their help in data collection.

REFERENCES

[1] Filippo Campagnaro, Roberto Francescon, Emanuele Coccolo, Antonio Montanari, and Michele Zorzi. 2023. A Software-Defined Underwater Acoustic Modem for Everyone: Design and Evaluation. *IEEE Internet of Things Magazine* 6, 1 (2023), 102–107. https://doi.org/10.1109/IOTM.001.2200221

[2] Emrecan Demirors, Bharatwaj G. Shankar, G. Enrico Santagati, and Tommaso Melodia. 2015. SEANet: A Software-Defined Acoustic Networking Framework for Reconfigurable Underwater Networking. In *Proceedings of the 10th International Conference on Underwater Networks & Systems* (Arlington, VA, USA) *(WUWNet '15)*. Association for Computing Machinery, New York, NY, USA, Article 11, 8 pages. https://doi.org/10.1145/2831296.2831316

[3] Kerem Enhos, Emrecan Demirors, Deniz Unal, and Tommaso Melodia. 2022. Coexistence of Multi-Dimensional Chirp Spread Spectrum in Underwater Acoustic Networks *(WUWNet '22)*. Association for Computing Machinery, New York, NY, USA, Article 2, 8 pages. https://doi.org/10.1145/3567600.3568152

[4] Zheng Guo, Aijun Song, Mohammad Towliat, Leonard J. Cimini, and Xiang-Gen Xia. 2023. Lake Experimentation of in-Band Full-Duplex Underwater Acoustic Communications With a Receiving Array. *IEEE Access* 11 (2023), 20741–20754. https://doi.org/10.1109/ACCESS.2023.3250444

[5] Jared Hermans, George Sklivanitis, and Dimitris A. Pados. 2022. A First-of-its-kind Low Size, Weight and Power Run-Time Reconfigurable Underwater Modem. In *2022 Sixth Underwater Communications and Networking Conference (UComms)*. 1–5. https://doi.org/10.1109/UComms56954.2022.9905700

[6] Bilal A. Jebur, Cornelius T. Healy, Charalampos C. Tsimenidis, Jeffrey Neasham, and Jonathon Chambers. 2019. In-Band Full-Duplex Interference for Underwater Acoustic Communication Systems. In *OCEANS 2019 - Marseille*. 1–6. https://doi.org/10.1109/OCEANSE.2019.8867207

[7] Alireza Maleki, Ha H. Nguyen, Ebrahim Bedeer, and Robert Barton. 2023. A Tutorial on Chirp Spread Spectrum for LoRaWAN: Basics and Key Advances. arXiv:2310.10503 [eess.SP]

[8] Gang Qiao, Shuwei Gan, Songzuo Liu, Lu Ma, and Zongxin Sun. 2018. Digital Self-Interference Cancellation for Asynchronous In-Band Full-Duplex Underwater Acoustic Communication. *Sensors* 18, 6 (2018). https://doi.org/10.3390/s18061700

[9] Gang Qiao, Shuwei Gan, Songzuo Liu, and Qingjun Song. 2018. Self-Interference Channel Estimation Algorithm Based on Maximum-Likelihood Estimator in In-Band Full-Duplex Underwater Acoustic Communication System. *IEEE Access* 6 (2018), 62324–62334. https://doi.org/10.1109/ACCESS.2018.2875916

[10] Marwane Rezzouki, Mohamed Amine Ben Temim, and Guillaume Ferré. 2021. Differential Chirp Spread Spectrum to perform Acoustic Long Range Underwater Localization and Communication. In *OCEANS 2021: San Diego – Porto*. 1–9. https://doi.org/10.23919/OCEANS44145.2021.9706010

[11] Lu Shen, Benjamin Henson, and Yuriy Zakharov. 2022. Full-Duplex UWA Communication System with Two Iterations. In *2022 Sixth Underwater Communications and Networking Conference (UComms)*. 1–5. https://doi.org/10.1109/UComms56954.2022.9905675

[12] Lu Shen, Yuriy Zakharov, Benjamin Henson, Nils Morozs, and Paul D. Mitchell. 2020. Adaptive Filtering for Full-Duplex UWA Systems With Time-Varying Self-Interference Channel. *IEEE Access* 8 (2020), 187590–187604. https://doi.org/10.1109/ACCESS.2020.3031010

[13] Fabian Steinmetz, Jan Heitmann, and Christian Renner. 2018. A practical guide to chirp spread spectrum for acoustic underwater communication in shallow waters. In *Proceedings of the 13th International Conference on Underwater Networks & Systems* (Shenzhen, China) *(WUWNet '18)*. Association for Computing Machinery, New York, NY, USA, Article 1, 8 pages. https://doi.org/10.1145/3291940.3291964

[14] Fabian Steinmetz and Bernd-Christian Renner. 2022. Taking LoRa for a Dive: CSS for Low-Power Acoustic Underwater Communication. In *2022 Sixth Underwater Communications and Networking Conference (UComms)*. 1–5. https://doi.org/10.1109/UComms56954.2022.9905674

[15] Mohammad Towliat, Zheng Guo, Leonard J. Cimini, Xiang-Gen Xia, and Aijun Song. 2022. Enhanced Accuracy of Self-Interference Cancellation in Underwater Acoustic Full-Duplex Communication. In *MILCOM 2022 - 2022 IEEE Military Communications Conference (MILCOM)*. 781–786. https://doi.org/10.1109/MILCOM55135.2022.10017822

[16] Lorenzo Vangelista. 2017. Frequency Shift Chirp Modulation: The LoRa Modulation. *IEEE Signal Processing Letters* 24, 12 (2017), 1818–1821. https://doi.org/10.1109/LSP.2017.2762960

[17] Cameron T. Vongsawad, Tracianne B. Neilsen, Adam D. Kingsley, John E. Ellsworth, Brian E. Anderson, Kaylyn N. Terry, Corey E. Dobbs, Scott E. Hollingsworth, and Gabriel H. Fronk. 2022. Design of an underwater acoustics lab. *Proceedings of Meetings on Acoustics* 45, 1 (03 2022), 070005. https://doi.org/10.1121/2.0001540

[18] Zhenqiang Xu, Shuai Tong, Pengjin Xie, and Jiliang Wang. 2020. FlipLoRa: Resolving Collisions with Up-Down Quasi-Orthogonality. In *2020 17th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. 1–9. https://doi.org/10.1109/SECON48991.2020.9158432