Bleep: Motor-Enabled Audio Side-Channel for Constrained UAVs

Adeola Bannis
Carnegie Mellon University
Moffett Field, California
abannis@andrew.cmu.edu

Hae Young Noh
Stanford University
Palo Alto, California
noh@stanford.edu

Pei Zhang
Carnegie Mellon University
Moffett Field, California
peizhang@cmu.edu

ABSTRACT

Small unmanned autonomous vehicles (UAVs) swarms are becoming ubiquitous in a number of applications (e.g., surveying, monitoring, and situational awareness). Indoor environments may contain metal equipment that temporarily disrupts radio reception. During these momentary interruptions, a small UAV needs to be able to broadcast a ‘heartbeat’ to indicate that it is not damaged or lost. Considering alternative messaging modalities, we observe that light-based methods require line-of sight, which is not guaranteed when UAVs are moving through a cluttered environment, while a naive sound-based method is easily drowned out by the UAV’s own loud motor and propeller noise.

We present Bleep, a side-channel messaging system that modulates UAV motors sounds to enable multiple UAVs to communicate when radio is unavailable. Bleep accomplishes this by 1) embedding linear chirps within the PWM frequency while maintaining the controllability of the UAV and 2) decoding the received sounds utilizing chirp multiplication to enhance signal detection in loud motor noise. Our chirp-based encoding allows multiple simultaneous transmissions to be detected and decoded in the presence of loud sounds from multiple UAV motors. Through real-world experiments, we show that Bleep is able to achieve over 95% signal detection and decoding accuracy with simultaneous transmissions.

CCS CONCEPTS

- **Computer systems organization** → **Sensors and actuators;**
- **Networks** → *Mobile ad hoc networks.*

KEYWORDS

UAV, motor sound, side-channel

ACM Reference Format:

Adeola Bannis, Hae Young Noh, and Pei Zhang. 2020. Bleep: Motor-Enabled Audio Side-Channel for Constrained UAVs. In *MobiCom 2020 (MobiCom ’20)*, September 21–25, 2020, London, United Kingdom. ACM, New York, NY, USA, 13 pages. <https://doi.org/10.1145/3372224.3419183>

1 INTRODUCTION

Unmanned aerial vehicles (UAVs) can be used in indoor/spatially constrained applications such as urban search and rescue [30, 35], warehouse management [32], and 3D indoor sensor surveys [29].

This work is licensed under a Creative Commons Attribution International 4.0 License.
MobiCom ’20, September 21–25, 2020, London, United Kingdom
© 2020 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-7085-1/20/09.
<https://doi.org/10.1145/3372224.3419183>

Some of these environments, such as warehouses, contain equipment or metal walls which absorb or scatter wireless signals [6], making it difficult for UAVs to maintain radio contact throughout the entire sensing task. Loss of radio contact during rescue missions or in busy warehouses can lead to loss of control, equipment/personal damage, and even destruction of UAVs. Therefore, we explore the possibilities for secondary, short-term communication that can be used as a ‘heartbeat’, indicating that the UAV is still present despite loss of radio communication.

Besides RF, information can be transmitted between these small UAVs through other sensing modalities. Vibration or motion-based methods [21, 23] cannot be used without physical contact between devices. Light-based communication schemes [8, 9] offer high data rates similar to the radios on the nano-UAVs. However, when multiple UAVs are moving around a cluttered indoor area, they may not be able to establish line-of-sight links. This leads us to explore the other possibilities available for messaging between small UAVs. Sound [10, 15, 17, 18] is another modality that can operate without direct line of sight, but is difficult to implement for UAVs due to the loud sounds produced by their motors and propellers.

We present Bleep, a sound-based signalling method for UAVs in indoor environments. We explore the difficulties of adapting sound-based communications in the presence of interference from the devices themselves. This signalling method operates at 1-2 bits/s, which is sufficient to establish that the UAV is present and operational until the radio link is reestablished. We characterize the spectral properties of the motor and propeller noise produced by UAVs in relation to the operating principles of the motors. Using this insight, we are able to use the motor sound for signalling and design an encoding and decoding scheme that is able to overcome the two main challenges of (1) high noise floor created by motor and propeller noise, and (2) computational constraints for signal decoding on small UAVs. Bleep encodes information as linear chirps by varying the frequency, but not duty cycle, of the motors during flight. The chirps from the motors can then be decoded in parallel with low computational cost using chirp multiplication.

Our main contributions in this work are:

- A chirp-based motor-sound encoding scheme that encodes information in motor noise without disrupting flight,
- A chirp multiplication decoding algorithm that detects multiple simultaneous heartbeat messages at a rate of 2 bits/s,
- Real-world characterization of motor-produced sound and evaluation of motor-produced message decoding accuracy and the effect on flight characteristics.

The rest of the paper is organized as follows. Section 2 gives an overview of the hardware requirements and software components of Bleep. Section 3 explains the motor-produced chirp encoding scheme. Section 4 describes the parallel chirp decoding algorithm.

Section 5 evaluates the performance of Bleep in terms of single and multi-message decoding accuracy, as well as the speed of multiple transmission decoding. Section 6 discusses some related work in non-RF messaging. Finally, Section 7 summarizes the key elements and performance of Bleep.

2 SYSTEM OVERVIEW

Bleep modulates motor sounds to transmit messages between drones when one or more of them has lost radio connectivity. Figure 1 shows the major components and processes in the Bleep side-channel messaging system. Each transmitter loads a message, (e.g. "signal lost"). This message is then encoded in the sound produced by the transmitter's motors by modifying the motor pulse-width modulation (PWM) frequency of the control signals used for flight. The receivers are equipped with microphones that capture the loud motor and propeller sounds from multiple transmitters. The sound recorded by the microphone is used to detect and decode the messages from the transmitters. This detection process leverages the shape of the transmitted signal, a linear chirp, for fast, parallel decoding. The message encoding and decoding processes are described in more detail in Sections 3 and 4.

Small, resource-constrained UAVs in an indoor sensing system send collected data and receive coordination commands over radio [30]; when connectivity is lost, they may run out of storage space for data, lose localization ability, or even collide with the environment or nearby UAVs. Therefore, the loss of radio connectivity is a critical error which must be resolved quickly. At a minimum, UAVs should be able to communicate whether they are still operational, with momentary radio loss. This information can be used to design a higher-level side-channel protocol, e.g. if a majority of UAVs have lost connectivity, abort the sensing task and land, else have the disconnected UAVs move closer to the others. In order to support many messages, we use Bleep to encode messages as a series of bits.

The only additional hardware needed by the receivers is a microphone; the transmitters use their already present motors for signalling. This frees up the sensing payload of the UAVs for just the sensors needed for the required task, plus at most a microphone.

Since the sound channel is very noisy, we expect a relatively low bit rate will be needed to ensure reliable reception. The bitrate for our implementation of Bleep is ~ 1 -2 bits/s. For comparison, LoRaWAN networks use frequencies in the GHz range, with a minimum bandwidth per chirp of 125kHz and a bit rate between 11bps and 27kbps. Bleep is restricted to a bandwidth of 1-2kHz per chirp, or about 1/100, and the bitrate is correspondingly smaller. Bleep is designed as a broadcast system, i.e. one-to-many messaging, and so there are no encoded destination addresses. To allow the message source to be identified, each UAV uses its own channel with a known frequency range. The encoded messages are represented as linear chirps within the assigned frequency range. The use of linear chirps in assigned, separated frequency bands also allows receivers to separately decode messages from multiple UAVs at the same time. The encoding and decoding methods are discussed in more detail in Sections 3 and 4 respectively.

3 TRANSMITTING MESSAGES AS PWM-DRIVEN MOTOR SOUNDS

The message encoding scheme to allow detection over the high amplitude background noise created by multiple UAV motors and propellers by using the motors themselves as a high-amplitude sound source, and using linear chirps at high frequencies where the propeller noise is less intense.

3.1 Designing Motor-based Sound Communication

3.1.1 Adapting PWM-produced sound. Bleep encodes the coordinating messages by modifying the frequency of the motor's pulse width modulation (PWM) signal, while leaving the duty cycle (for flight control) unaffected. PWM is used when a controller with digital (on/off) outputs needs to drive a motor or other device that expects continuous input. For example, a UAV makes fine adjustments to the power in each of its motors to control its three-dimensional pose and velocity. PWM control is especially effective for inductive loads such as motors, as the current in motor coils cannot switch fully on or off instantaneously.

In most control applications, the frequency of a PWM driving signal is held constant while the duty cycle varies to change the power delivered to the the motors, i.e., the average energy delivered per unit time. A sufficiently high frequency (on the order of kHz [7]) PWM signal will appear to be a smooth input, at a power level determined by the wave duty cycle. The driving frequency can have the effect of a whine or rumble as the motor casing and/or coils vibrate at this frequency. The main advantage of varying the PWM frequency is that it does not significantly change the control characteristics of the UAVs because duty cycle of the motors is not changed by Bleep. Figure 2 shows the effect of varying duty cycle or frequency on the power output by a PWM driver. The top and middle signal are driven at the same frequency but with different duty cycles, and so provide different amounts of output power. The top and bottom signals are driven at different frequencies and the same duty cycle, and so provide the same average output power. At very low frequencies, the time between pulses is long enough for the propellers to lose significant momentum before the next pulse, causing ripples in the power output, as shown in the bottom figure. These ripples are always present, but are less noticeable at higher frequencies. Therefore, the power output and therefore the flight of the small UAVs will not be affected by changes in the PWM frequency as long as the new frequency is high (on the order of 10kHz [7]). This means that Bleep is able to encode and transmit the message signal using the UAV motors without significantly disrupting flight or delaying the sensing task.

3.1.2 Chirp-based modulation scheme. The sounds produced by the motors must be chosen so that they can be easily and accurately produced without reducing the UAVs ability to fly. Since we are constrained to maintain flight control by not changing the duty cycle, we focus on messaging schemes that depend on frequency modulation. In particular, we choose linear chirps as the basis of our message encoding, which are commonly used to transmit signals over long distances or when SNR is low [3, 24, 33, 38]. A linear chirp used as message bits is defined by three parameters: chirp length,

Figure 1: System diagram for Bleep. Multiple UAV transmitters can send out chirp-encoded signals from their motors. Each receiver can collect these signals and estimate the proximity of the senders to control radio traffic, assign sensing groups, or for other swarm coordination tasks.

Figure 2: PWM signals of different duty cycles and frequencies. The effective power delivered to the motors is determined by the duty cycle of the motor, i.e. what proportion of time it is kept on. At very low frequencies, the long intervals between pulses can cause ripples in the power output.

chirp rate (slope/steepness) and center frequency. Each of them is limited in some way by the characteristics of motor-produced sounds or the noise from nearby UAV motors and propellers.

3.1.3 Encoding constraints and trade-offs. The usable center frequencies are mainly constrained by the characteristics of UAV motor and propeller noise at different duty cycles. Bleep modulates only the PWM frequency, while duty cycle of each motor changes frequently during flight for control and movement. Thus, we must choose parameters for the chirp so that Bleep signals are detectable at a variety of different duty cycles.

Duty cycle. Figure 4 shows the effect of duty cycle on motor-produced chirps with the same frequency parameters. The duty cycle of the UAV motors is varied from 5% to 75%, and the chirp is transmitted 0.5s after the motors turn on. The driving PWM frequency before and after the chirp is set to a very high value which is a multiple of the sampling frequency, so that does not appear in the spectrogram. The transmitted chirp has a center frequency of 7kHz, a chirp rate of 3200Hz/s, and a length of 1.25s; these parameters were chosen to emphasise the effect of duty cycle and the shape of the motor-produced chirp.

Figure 3: High-amplitude harmonics from PWM-driven UAV motors. A fixed PWM frequency (a) produces harmonics which appear in the spectrogram at multiples of the original frequency, or below it due to aliasing. These harmonics could be mistaken for signals from other UAVs. With a chirped PWM frequency (b), these harmonics have higher chirp rates (i.e. steeper slopes) than the original, so are less likely to be mistaken for real messages.

At very low duty cycles (a), the fundamental chirp and its harmonics are much stronger than the propeller noise. As the duty cycle increases (b-d), the propeller noise increases more quickly at low frequencies, so that the fundamental PWM frequency is harder to detect and eventually lost. Although the propeller noise is present at all frequencies, it is less intense at high frequencies, even with a very high duty cycle (d). Therefore, Bleep uses center frequencies that are as high as possible without crossing the Nyquist frequency of the receiver's audio recording hardware.

Figure 4: Spectrogram of the motor and propeller noise from a UAV transmitting a linear chirp. As the duty cycle increases from 5% in (a) to 75% in (d), the propeller noise at low frequencies becomes to intense for the chirp to be detected. These signals also clearly show the distinct frequency steps in a motor-produced chirp, and a slight increase in sound power when the chirp begins.

Chirp rate. The chirp rate is constrained by the physical characteristics of motors and propellers. In Figure 4, the entire signal becomes more intense as the chirp begins, and less intense when it ends. In theory, the presence of the linear chirp in the PWM frequency should not affect the power output of the motors. However, the signal output from UAV motors is not a smooth, linear chirp, but rather a series of frequency steps that rise or fall to approximate the chirp. The discontinuity between these steps causes power spikes; over time these power spikes average out to an increase in power output. The higher the chirp rate, the bigger this discontinuity and therefore the worse the distortion in the motor power. However, very low chirp rates mean that harmonics is not as easily distinguished from real chirp messages. Therefore a balance between high and low chirp rates must be found to maximise message detection and decoding accuracy.

Length. The chirp length is indirectly constrained by the background motor and propeller noise from the transmitter UAV as well as the other UAVs around it. For example, a short chirp may be lost in the case of sudden changes in the ambient noise (e.g. objects falling, more UAVs approaching). Increasing the chirp length gives more opportunity for the detection algorithm to register the presence of a chirp and verify its frequency and slope. However, longer chirps reduces the rate of information transfer in the messages, as it takes longer to transfer each single bit of information. Therefore, Bleep must balance faster information transfer with shorter chirps against more robust detection with longer chirps. The final parameter selection is described in Section 5.5.

3.2 Encoding Information as Linear Chirps

Given the three parameters of a linear chirp, center frequency, chirp rate and chirp length, we must vary at least one of them to represent the different bits in our message. We have chosen to use center frequency to represent different transmitters, leaving chirp rate and chirp length constant for all transmitters. Varying the chirp rate would require multiple decoding filters/chirp templates, which would increase the decoder memory usage if these are pre-calculated, otherwise it would increase the computational complexity. We use two different chirp rates, which are equal in magnitude but opposite in direction, with an upchirp (increasing frequency) representing **1** and a downchirp (decreasing frequency)

representing **0**. In the LoRa message format [1, 3, 36], which is also chirp based, the chirp length is also constant, but the start point of the chirp is used to encode multiple symbols, giving a higher bit rate with the same chirp length. However, the more symbols used, the greater the probability of decoding error in a noisy channel. Since the noise from UAV motors is so large, we chose to have only two symbols, with a correspondingly lower bitrate, in order to maximise decoding accuracy. Furthermore, we implement a basic encoding without error-correction and assess the baseline error of this physical layer. The addition of error-detection or error-correction codes over this baseline will greatly improve message delivery rates.

There was also no packet structure imposed on the transmissions. The main goal of this was to explore the feasibility of motor-produced sounds as an information channel between nano-UAVs and a listening receiver. We therefore transmitted only raw bit streams and assessed whether bits were recovered at the receiver side. Given the high losses from noise in this channel, we envision that a future protocol would have data rates similar to LoRa's 4/8 setting, i.e. 8 symbols to transmit four bits. In addition, a packet header to synchronize the transmitter and header is needed. This could be realized with a preamble of alternating bits, followed by synchronization bits, similar to Ethernet [34]. Unlike Ethernet, however, the preamble should be much shorter due to the lower data rate currently possible with motor-produced sound.

In summary, PWM-driven motors used in nano-UAVs can be used to encode message signals in the high-amplitude sound they already produce. The motor-produced signals can be detected in the loud background of propeller sound from the transmitter itself as well as other nearby UAVs. The linear chirps used can be separated in the frequency domain to allow for multiple transmitters, and are robust to self-interference from PWM harmonics. The decoding process for these motor-produced linear chirps completes the overall design of the messaging system, as we will now describe in Section 4.

4 PARALLEL DECODING OF MOTOR-PRODUCED SIGNALS

Bleep's decoding algorithm must work with motor-sound encoded messages which may be simultaneously transmitted by many UAVs that have lost connectivity, and in the presence of noise from other, non-transmitting UAVs. The messages should be decoded quickly,

Figure 5: Multiple overlapping Bleep messages

since loss of radio is a critical problem for small, constrained UAVs that rely on remote coordination. This means that parallel decoding of multiple messages, instead of sequential decoding, is the best case when multiple UAVs lose connectivity. Additionally, the decoding method should have low computational and storage requirements, so that nearby UAVs can receive and relay messages when the remote coordinating device is too far from the signalling UAVs.

In this section, we describe the use of chirp multiplication to demultiplex the parallel transmissions with low space and computational overhead. The main steps of the decoding algorithm are (1) audio capture, (2) chirp multiplication, (3) frequency tracking and (4) decoding.

4.1 Audio capture

The first step in decoding the UAV motor-produced signals from multiple transmitters is to capture audio with a microphone for processing. Figure 5 shows the spectrogram of captured audio containing multiple simultaneous messages, before windowing. The sampling rate of the audio capture system must be high enough to allow for messages with high enough center frequency to avoid the most intense band of propeller noise. The recorded data is then stored in a buffer, where it is divided into frames, i.e. short windows of audio data. Each audio frame can then be processed to detect and decode messages from other nearby UAVs. Before the next step, chirp multiplication, the audio frame is high-pass filtered to remove the highest-amplitude part of the motor and propeller noise, in order to more clearly detect embedded motor-produced signals.

4.2 Chirp Multiplication

4.2.1 Chirp multiplication vs cross-correlation. Bleep uses chirp multiplication [4, 28] instead of the more traditional cross-correlation methods [16, 25] for detecting chirps in a noisy environment. This is because of the specific requirements in the case of side-channel communication during radio loss. In a radio loss event, the total number of affected UAVs is not known beforehand, so the number of simultaneous transmissions is not predictable.

Additionally, the start times of these transmissions cannot be synchronized because the radio loss event occurs unexpectedly. In applications where there is only one transmitter, a single cross-correlation is able to detect and differentiate all the 0 and 1 bits in a captured audio window. In the case of a radio loss event, however, multiple transmitters will send out chirps with different center frequencies, meaning that cross-correlation must be run multiple times against each captured audio window, with a different chirp center frequency each time. This incurs a burden of either storage or computational space on the small UAVs that are listening for

messages: they can (1) store chirp templates for each possible center frequency assigned to the swarm members, or they can (2) store a single chirp template and modulate it (or the captured window) to each possible center frequency. Low-power processors suitable for small, indoor UAVs may have signal processing capabilities that are limited to short window sizes [22].

A better approach is to use a method that can detect the presence of chirps from multiple transmitters (i.e. with different center frequencies) within the same captured audio window, even when the length of the window is less than the length of the chirp. This is possible using chirp multiplication, which can be thought of as losing information about the time domain in order to locate chirps in the frequency domain, whereas cross-correlation loses information about the frequency domain content of a signal in order to precisely locate it in time. This dual relationship is shown in Fig. 6.

4.3 Frequency tracking

Since the signal window used on the nano-UAVs is much shorter than the transmitted chirp length, the frequency must be tracked over time to differentiate a real signal from transient interference. As the captured window slides forward in time, a chirp with the same starting frequency will appear at different frequency bins after chirp multiplication depending on its start time relative to the window, as shown in Fig. 7. The apparent frequency of an up-chirp will slowly increase as its start time becomes earlier in (or before) the captured audio window until it disappears. When the beginning of the received chirp is aligned to the beginning of the audio window, it will appear at its start frequency, and just before it falls out of the window, it will appear at its end frequency. Together with the length of the chirp, the tracked frequencies allow us to find the true center frequency of the received chirp, and therefore assign the chirp to the receiver.

The frequency is tracked by applying a fast Fourier transform (FFT) to the captured window after chirp multiplication, then determining which frequencies in the output are likely to be the result of a motor-produced chirp-encoded message. For a signal to be recognized as a chirp in a Bleep message, it must produce a strong frequency peak within one of the frequency bands assigned to transmitters, and then follow the chirp rate chosen at the start of the sensing task. This reduces the likelihood of impulses or other transient noises being falsely marked as message symbols. If the possible symbol is detected for the correct duration, it is marked as a 1 if it is an up-chirp or a 0 if it is a down-chirp.

Figure 8 shows how the peak frequency changes as windows of audio are captured containing a Bleep message. The top figures on the left and right show the spectrogram of the same audio snippet containing a chirp from the motors of a real UAV. The chirp parameters chosen were a chirp length of 1.0s, chirp rate of -2000Hz/s, and center frequency of 16kHz. The audio snippet is processed in frames (windows) to produce peak frequency value (middle) and amplitude (bottom) signals. On the left side, the windows are multiplied with a complex chirp template before the frequency content is tracked. The frequency value signal clearly follows a linear trend from 17kHz to 15 kHz over 1.0s, while the amplitude value is much higher during the chirp than before or after. In contrast, the right side shows the raw frequency information tracked in the signal

Figure 6: The effect of cross-correlation (b) and chirp multiplication (c) on a motor-produced approximation to a linear chirp (a). Cross-correlation concentrates signal energy in the frequency interval of the chirp near a single point in time, whereas chirp multiplication concentrates the signal energy in the time interval of the chirp near a single point in frequency.

Figure 7: Effect of chirp multiplication on a time-frequency representation (e.g. a spectrogram) of a signal containing two chirps. The frequency of the resulting tone is distorted according to the timing of the chirps in the signal window.

without any chirp multiplication. The peak frequency value does not follow a linear chirp shape at all, and the peak frequency amplitude is much lower and does not clearly indicate the start and end of the chirp. This shows how effective chirp multiplication is for extracting the embedded motor-produced linear chirp messages.

This method of frequency tracking solves the main challenge of frequency distortion, but with important constraints on the captured audio window parameters and on the choice of chirp parameters for the transmitters.

4.3.1 Comparison to other spread spectrum techniques. A similar method is used to decode other chirp spread spectrum messages, such as LoRa symbols [37]. Typically, only upchirps are used in LoRa messages, and the starting frequency is varied to represent different symbols. These symbols are recovered after chirp multiplication according to the maximum frequency bin in the multiplied chirp signal. However, this decoding process must be modified for motor-produced chirp decoding for two reasons. Firstly, the digital signal processing (DSP) capabilities of the small UAVs are optimized for relatively small buffer sizes, meaning the final resolution of the Fourier transform used to decode the signal is lower. This makes it more difficult to distinguish different symbols based on the chirp start time. Instead, we only detect if a chirp fell into the correct frequency range or not. However, with only upchirps, it would not be possible to distinguish an “off” symbol from no transmission at all. This leads to the second modification, using a complex chirp template in order to separate the response to up- and down- chirps into different parts of the resulting spectrum with a single chirp

multiplication. These two modifications make the decoding of Bleep messages more robust to the very high amplitude noise from UAV motors and propellers.

4.3.2 Parameter constraints for decoding algorithm. Both the captured audio window parameters and the encoded chirp parameters affect the performance of the frequency tracking algorithm. The actual location of the maximum frequency bin from a multiplied audio buffer depends on the relationship between the true start time of the chirp and the start of the captured audio window. This can be seen as a distortion in the measured chirp center frequency. For a received chirp with slope k , the total distortion depends on the length of the audio window, T_{window} :

$$\text{distortion}_{\text{max}} = kT_{\text{window}} \quad (1)$$

Besides the length of the window itself, the overlap between windows affects the accuracy of frequency tracking. A large overlap means that the change in frequencies between windows is smaller; the tracking signal will be smoother and the true chirp start and end times are more likely to be captured. However, a large overlap requires faster processing, since a full window of captured audio must be processed before the small number of additional samples for the next window have been recorded. Therefore, the window overlap must be chosen so that the time taken to sample data for the next window is greater than the time needed to perform chirp multiplication and frequency analysis on the full window.

The choice of chirp length, slope, and center frequency also affect the frequency tracking algorithm. Longer chirps can be tracked over more windows, which increases the confidence in detection vs noise, as well as the accuracy of the extracted center frequency. However, shorter chirps allow messages to be completed more quickly, which is necessary during a critical event such as the loss of radio connectivity. The center frequency assigned to a single transmitter does not directly affect the tracking algorithm, but the spacing between the center frequencies of different transmitters interacts with the chirp slope and length to make frequency tracking more or less difficult. The frequency distortion is proportional to the delay to the start of the received chirp. If the center frequencies of two transmitters are too close, then their chirps can overlap to appear as one long chirp, which will be assigned to one or the other transmitter. An FFT on a captured window of N audio samples

sampled at f_s can only distinguish frequencies that are separated by more than $\frac{f_s}{N}$. In practice, however, frequencies are much easier to distinguish if they are even more separated. A small value n can be chosen as the minimum number of frequency bins there should be between signals from different transmitters. For a chirp slope of k , chirp length of T_{chirp} and a center frequency spacing Δf , two chirps from different transmitters appear as one long chirp if

$$|\Delta f| \leq kT_{\text{chirp}} + \frac{nf_s}{N} \quad (2)$$

Therefore, the chirp slope k and chirp length T_{chirp} determine the minimum spacing between center frequencies for different UAV transmitters, and therefore the maximum number of transmitters.

4.3.3 Frequency shift compensation. The decoding algorithm associates chirps to specific UAVs based on the frequency of the detected chirp. One phenomenon that may affect the received frequency is the Doppler effect, whereby motion causes the apparent frequency of a sound to increase or decrease depending on the relative motion of the source and observer. The frequency of the received signal, f , can be calculated as

$$f = f_0 \cdot \frac{c + v_o}{c + v_s} \quad (3)$$

where f_0 is the frequency as transmitted, c is the speed of sound, v_o is the velocity of the observer towards the source, and v_s is the velocity of the source away from the observer (i.e. in the same direction as v_o). We assume a maximum speed of 3m/s; although the UAVs may be capable of higher speed, in an indoor environment high speeds greatly increase the chance of collisions or crashes. This risk is especially high if an indoor localization system with positioning error ~10cm is used for navigation. Taking the speed of sound as 340m/s, we can find the frequency when moving together is $f = f_0 \cdot \frac{c-3}{c+3} = 0.98f_0$, and when moving apart as $f = f_0 \cdot \frac{c+3}{c-3} = 1.017f_0$. That is, there is an error of about 2% in the received frequency. This is small enough that most of the received energy for a particular frequency band lies in the expected range, and so no frequency shift compensation is done in Bleep.

4.4 Final decoding

The down- and up-chirps detected in the frequency tracking stages are converted to **0** and **1** bits, which are grouped to reconstruct the encoded message bits. A long pause (more than 2 symbols) is used to separate transmissions for decoding.

We implemented the encoding and decoding algorithms described in Sections 3 and 4 using real UAVs in order to assess their real-world performance. The next section details experiments and characterizations for Bleep evaluation.

5 EVALUATION

Bleep is designed to allow UAVs in a sensing swarm to send coordination messages to their nearest neighbors using linear chirps embedded in frequency modulated motor sounds. In this section, we first measure the effect of Bleep messages on the power output and flight of the quadcopters used, in order to determine any physical limitations. We also characterize the properties of motor and propeller noise from real quadcopters in order to select the optimal parameters for a Bleep deployment. We then evaluate the

performance of Bleep with real quadcopters in terms of decoding accuracy with a single transmitter and with multiple transmitters.

5.1 Experimental Hardware and Settings

The UAVs used for our deployment were Crazyflie 2.0[5] nano-quadcopters, which each have four brushed, coreless DC motors. The firmware of the Crazyflie was modified so that they could transmit messages by updating their PWM frequency every 50ms during transmission. When no message is in progress, the PWM frequency is kept at the high default value of 328kHz, to keep the audio bandwidth clear for other transmissions.

The firmware modification also included a decoder task, with a sampling rate of 48kHz, and a double audio buffer with 1024 samples per buffer. A microphone was attached to each Crazyflie to provide audio input. This receiver task was included to ensure that the small UAVs are computationally able to perform both encoding and decoding of Bleep messages simultaneously. However, the recorded audio and converted signals were not saved to the Crazyflie's storage, as that would involve additional computational and storage resources not directly related to encoding/decoding. Instead, a Pixel 3a smartphone was used to record and store the raw audio for characterization and evaluation. The recorded audio was transferred to a Windows PC for offline decoding. The message length chosen was 8 bits.

5.2 On-device timing

The RAM and time constraints of the UAVs used were important considerations in the choice of decoding algorithm. The Crazyflie uses a STM32F405 processor with a processor speed of 168MHz and 192kb of SRAM storage. A microphone and amplifier were attached to a grounded Crazyflie to give the audio input used to evaluate the on-device requirements performance of the Bleep decoding algorithm. The setup is shown in Figure 9. The Fourier transforms needed for Bleep were done using a hardware-supported digital signal processing (DSP) library for the STM32F405 processor. The FFT implementation is limited to buffers of 4096 32-bit values, which would occupy 16kB of RAM. However, in practice, the Bleep algorithm could only be implemented with a buffer of 2048 samples before the RAM size was exceeded. This is due to the memory overhead for storing the data, performing the transform and processing the results, as shown in Table 1. The flight control, sensor

Function	Bytes per buffer element
Audio buffering	4
Chirp template	8
FFT buffer	8
FFT support structure	12
Total	32

Table 1: Memory overhead for Bleep chirp decoding

drivers, and other existing firmware already occupies 54kB or 41% of the total SRAM, leaving 77kB available. A buffer size of 2048 audio samples requires 65.6kB for processing, or 85% of all the remaining available SRAM. Therefore, we limited our on-device implementation to a maximum buffer size of 2048 samples. We then

Figure 8: An example of the tracking signals produced in the detection stage. Each point of the tracking signals is calculated using a windowed FFT centered at that time. Chirp multiplication (top) focuses the energy in the chirp while dispersing the background noise, giving a clean, well-defined linear frequency signal. If the chirp multiplication is omitted (bottom), the chirp is much more difficult to identify.

Figure 9: Microphone and amplifier used to test algorithm run time on a stationary Crazyflie

ran the chirp decoding algorithm with different buffer sizes to find the tradeoff between accuracy and speed/memory use. The results are given in Table 2. Although the maximum buffer size of 2048 samples uses almost all the available memory, it gives much better performance than smaller buffers. It may be possible to reduce the memory requirements of the algorithm using downsampling, in order to free RAM for additional tasks.

Buffer size	Memory (kB)	Time (ms)	Accuracy (%)
256	8	0.18	9.4
512	16	0.52	53.1
1024	32	1.43	62.5
2048	64	3.97	96.9

Table 2: Processing speed vs accuracy for Bleep decoding

5.3 Effect on Motor Power Output

5.3.1 Experiment Setup. Bleep is intended to allow small UAVs to transmit messages by modulating the PWM frequency of their motors without degrading flight control. Therefore, we evaluate the effect of the different chirp parameters used in Bleep on the power output of the UAV motors. A digital scale was used to measure the thrust created by a single Crazyflie on a weighted stand, as shown in Fig. 10a. The stand both ensured that the Crazyflie did not takeoff from the scale at high thrust (i.e. high duty cycle), and that the airflow was not distorted by ‘ground effect’ [12] of being too close

Figure 10: Experimental setup for real device evaluations. To evaluate the effect on thrust (a), a Crazyflie was mounted on a scale. To evaluate decoding accuracy (b) multiple Crazyflies are arranged in a line, with a smartphone for off-line data storage. All motors are on for the duration of the experiments, for the full effect of motor noise interference.

to a solid surface. The weight of the Crazyflie and stand without any motor power was taken at the beginning of the experiment as W_0 , and then the weight $W(t)$ was tracked over the course of multiple chirps and used to determine the thrust output as $T(t) = (W_0 - W(t)) * g$, where g is the acceleration due to gravity, 9.81m/s^2 . The default parameters chosen were a chirp length of 1.0s, a chirp slope of 1500 Hz/s, center frequency 18kHz, and 0.2 duty cycle. Each of these parameters was varied to measure the effect of changes on the motor output. To find the default values of the motor thrust without chirps, the Crazyflie motors were run at each different duty cycle at the original 328kHz frequency used in the Crazyflie firmware before starting the chirp message transmissions.

5.3.2 Results. Figure 11 shows the results of varying the length, slope, center frequency of the encoding chirps, as well as the performance with the default chirp parameters at different duty cycles. The periodic increases in motor PWM frequencies used to simulate linear chirps introduces extra energy to the driven motors, which shows itself as an increased power output during the chirp messages vs without chirps for all the parameters.

Chirp length. Increasing the chirp length slightly increases this offset (Fig 11a), which is expected as there are more frequency

Figure 11: The effect on motor output power of chirp parameters and duty cycle. Increasing chirp length and decreasing chirp frequency appear to make the offset larger, while chirp slope did not have a significant effect.

jumps, and therefore more impulse-like changes in the driving motor power.

Chirp rate. Interestingly, the slope of the chirp did not significantly affect the offset compared to length or center frequency over the range tested in Fig. 11b. It is expected that increasing the slope will increase the power of the motor output signal, since the frequency content of a linear chirp becomes closer to an impulse as the slope (chirp rate) increases. However, if a pure tone is treated as having an angle of 0° and an impulse as having an angle of 90° , the highest slope tested here corresponds to an angle of only 4° . This seemingly small angle occupies a large amount of the available audio bandwidth that must be shared between UAVs. For the sampling rate of 48kHz used, this chirp rate already limits the number of possible transmitters to 13 or less, without accounting for the extremely high interference of propeller noise at low frequencies, which effectively halves the estimate. Therefore, higher chirp slopes are not practical for swarm communications, and the effect of chirp slope in a real deployed Bleep system is much less than the effect of the discontinuities in PWM frequency from creating the chirps.

Center frequency. The thrust offset decreases slightly as the center frequency of the chirp increases, which was not initially expected. One possible explanation is that the size of the frequency steps made by motors during the chirp relative to the current frequency is what creates the impulsive energy jump. Using the default chirp slope of 1500Hz/s and chirp length of 1.0s, we note that the frequency change is 15% of the center frequency at 10kHz, but only 6.8% at 22kHz, and this percentage decreases exponentially similar to the exponential decrease in the thrust offset seen in Fig. 11c.

Duty cycle. Finally, we note that the thrust offset is not the same at different duty cycles. Instead, it increases as the duty cycle increases from 0.2 to 0.5, then decreases again as the duty cycle increases. The duty cycle of a PWM signal does not only change the average power of the wave, but also the relative amplitude of different frequency components. One possible explanation for the offset frequency-dependence with duty cycles is that the high frequencies created by the step discontinuities in the motor noise create constructive or destructive interference with the harmonics making up the PWM signal. This interference changes with duty cycle because the frequency content of the harmonics also changes.

5.4 Effect on Flight Stability

5.4.1 Experimental Setup. To evaluate the effect of the changing PWM frequency on flight stability, a Crazyflie was made to hover in place using a tracking marker, as seen in Figure 12. The Crazyflie was allowed to hover for 10 seconds, then a chirp message was sent while still hovering. The transmitted chirps had a center frequency of 16kHz, chirp rate of 1000Hz/s, and chirp length of 0.5s. The displacement from the hover setpoint was used as a measure of flight stability, with a high displacement indicating poor stability.

5.4.2 Results. The results of the flight experiment are shown in Figure 13. Without the chirp transmissions, the Crazyflie tended to stay within 0.5m of the setpoint, with a median displacement of 0.22m. During the chirp transmission, the Crazyflie remained within 0.45m of the setpoint, with a median displacement of 0.3m. The displacement from the setpoint was slightly higher with the chirp, but was still within the same range as without the chirp messages. Additionally, the variation of position is smaller during the chirp signalling than with no transmissions. Therefore, the use of the motors for chirping did not reduce the controllability of the UAV. This is expected, as the frequency of the PWM driver does not determine the power being sent to the motors. The flight control is still able to adjust the motor duty cycle independently of frequency to stabilize the Crazyflie. A video of the Crazyflie signalling in flight is available online ¹.

5.5 Characterization and Parameter Selection

In a real-world Bleep deployment, the specific parameters of the linear chirps must be set: length, slope and center frequency. As described in Sections 3 and 4, each of these parameters is subject to trade-offs that affect flight control, message transmission rate, decoding accuracy, and the number of simultaneous transmissions possible. Furthermore, the parameters that are chosen for a real-world deployment should be robust to small variations in duty cycle, as it cannot be directly controlled during the message transmission.

5.5.1 Simulated parameter search. The interdependent effects of the linear chirp parameters make manual selection with real hardware difficult and time-consuming. In order to narrow the search space for linear chirp parameters, we first simulated some motor-produced chirps from a single quadcopter over a wide range of parameters, given in Table 3. The simulated motor-produced chirps

¹<https://youtu.be/ioDUebBijSM>

Figure 12: A Crazyflie hovering in place with a tracking marker. The displacement of the Crazyflie from its hover setpoint was tracked with and without chirp transmissions from its motors. The bottom figure shows the view from the tracking camera during the flight.

Figure 13: The displacement of the Crazyflie from its hovering setpoint with no chirp and during chirp message transmission. The displacement from the setpoint is slightly higher during the chirp, but completely within the range of the variation without the chirp.

were created by combining background noise with generated step-like chirp signals. The background noise was created recording the sound of the Crazyflie motors and propellers at 50% duty cycle and default PWM frequency of 328kHz (i.e. with no embedded chirps), and the amplitude of this noise was scaled up and down to reflect different values of duty cycle. The chirps themselves were produced by generating a PWM signal as a square pulse train with the desired duty cycle, and increasing or decreasing the frequency every 50ms (the update rate in the real Crazyflie firmware) to match the desired chirp length and chirp rate.

The best chirp parameters were determined using the bit error ratio (BER) of the decoded messages, calculated as the ratio of missing, wrong or extra bits to the length of the true message. A

Parameter	Range	Default
Symbol length (s)	0.1 - 0.9	0.75
Chirp rate (Hz/s)	500 - 3000	1500
Duty cycle (%)	10 - 90	50
Minimum frequency (Hz)	9000 - 21000	15000

Table 3: Search space for linear chirp parameters to be used in a real-world deployment of Bleep. The values selected for verification after simulation are given in the rightmost column.

BER of 0 signifies a perfect transmission and decoding; a BER of 1 signifies that all the bits were lost or flipped. During a sensing deployment, the duty cycle and chirp center frequency are not directly controllable: the duty cycle changes during flight, while the center frequency is assigned to a UAV so that each UAV has a separate frequency. Therefore, we focus on changing the chirp slope and the chirp length and finding the effect on BER for different duty cycles and center frequencies. The best choice of chirp length and chirp rate is the combination that provides the lowest BER over the widest range of duty cycle and center frequency. For the simulated parameter selection, we generated 50 8-bit messages to be encoded as motor-produced chirps, then decoded the simulated audio signals with the decoding algorithm described in Section 4.

Simulation results. Figure 14 shows partial results of the performance of the decoding algorithm on these simulated messages. From the selected results, we note that increasing the chirp slope improves accuracy. In Fig. 14e, the chirp slope is very low, and the error rate is much higher than with higher slopes and the same chirp length (Fig. 14c, 14d). The BER at 1500Hz/s (Fig. 14d) is nearly as low over the whole testing range as 3500Hz/s (Fig. 14c), and so we prefer the lower chirp slope, where each UAV occupies less bandwidth and therefore more UAVs can be supported in a swarm. The length of the chirp also affected accuracy, with shorter chirps (Fig. 14a, 14b) having higher BER over the whole range compared to longer chirps (Fig. 14c, Fig. 14d). Shorter chirps are preferable for fast messaging, but fast messages are not useful if they cannot be received correctly. Based on the results with simulated motor chirps, we selected a chirp rate of 1500Hz/s and a chirp length of 0.75 as the optimal values for our real-world deployment.

5.5.2 Real-world characterization. The real-world performance of the parameters chosen by simulation was evaluated to ensure that the simulated motor chirps were a good approximation to real chirps. For this verification test, we used a Pixel 3a smartphone as the receiver, recording at a sampling rate of 48kHz, and a single Crazyflie as the transmitter, located 1m away. We varied one parameter at a time while keeping the others at the selected values in the last column of Table 3. Each parameter was tested with the same set of 50 8-bit messages used in simulation.

Real vs simulated performance. Figure 15 shows the real and simulated bit error ratio when (a) chirp length, (b) chirp rate, (c) duty cycle, and (d) center frequency were varied. In general, the results agree well except for the extreme values of chirp rate and for a duty cycle of 75%. In the case of chirp rate, the real-world chirp

Figure 14: A subset of the bit error rates found when decoding simulated motor-produced chirp messages with different chirp parameters. Each figure is labelled with (chirp length, chirp slope), and shows the variation in error at different duty cycles and center frequencies. Long chirp length (c,d,e vs a,b) increases accuracy, but has less of an effect than high chirp rate (a,c vs b,d vs e).

Figure 15: Comparison of decoding accuracy of real-world and simulated Bleep messages. In most cases, the real-world motor chirps matched or outperformed the simulation. In particular, real-world chirps were much more robust to changes in chirp rate (b).

performed much better at low values than expected, while the chirp rate of 2500Hz/s did not perform well at all. This may be due to some filtering behavior caused by induction in the actual motor and casing, which distorted the chirp signal at specific resonant high frequencies. In the case of high duty cycle, the real results are actually more in line with expectation than the simulated results. The non-linear relationship between the propeller noise frequencies and the motor duty cycle was not captured effectively in simulation, causing overly optimistic results. In general, however, the simulation proved to be a close approximation to the real performance of the motor-encoded sound messaging system, allowing us to select reasonable parameters without a full evaluation in real hardware.

5.6 Single Transmitter Accuracy

5.6.1 Experiment Setup. For the single transmitter experiments we used two Crazyflies, one as transmitter, and one as receiver, with a smartphone below it for offline data collection. This is represented

by Figure 10b, with a single transmitter at distance d_1 from the transmitter. The receiving Crazyfly had a fixed motor duty cycle of 50%, which is approximately the power level needed to hover with no additional sensor payload [2]. The transmitter’s motor duty cycle was varied from 10% to 75%, while the transmitter-receiver distance was varied from 0.5m to 4.0m. The chirp center frequency was held constant at 22.0kHz. Based on the results with simulated motor chirps, we selected a chirp rate of 1500Hz/s and a chirp length of 0.75 as the optimal values.

We evaluated the decoding accuracy for Bleep messages with a single transmitter and a single receiver, separated by 0.5m to 4.0m and different transmitter duty cycles. Our metric for performance was BER, as defined in Section 5.5. The results are shown in Figure 16.

Results. For a duty cycle of 50%, which corresponds approximately to power needed for hovering, there is almost no decoding error up at distances up to 4.0m. This suggests that when a UAV is hovering or moving with a low velocity, which is most likely during active data collection, it is able to communicate with other UAVs as much as 4.0m away if necessary. For higher and lower duty cycles, the decoding accuracy suffers as distance increases. The accuracy at a lower duty cycle of 25% is still very good, but the accuracy at 10% falls off quickly with distance. This is because the total energy in the signal is lower, so the sound is not detectable at long distances. In the 75% duty cycle messages, the energy in the chirps is much smaller proportional to the background noise created by the air moving over the propellers, and so the error increases.

5.7 Multiple Transmitter Accuracy

5.7.1 Experiment Setup. For the multiple transmitter experiments, four Crazyflies were used with their propellers on throughout the experiments. Groups of 1 to 4 transmitters were chosen to transmit overlapping messages, so that the receiver need separate and decode multiple IDs at once. The multi-transmitter experiment setup is shown in Figure 10b with various distances d_1 to d_4 , a maximum separation of 2.0m and a minimum separation of 0.25m between transmitters. Figure 5 shows a spectrogram of the unsynchronized, overlapping messages as recorded by the receiver.

We evaluated the decoding accuracy for multiple transmitters transmitting simultaneously. The transmitter UAVs were arranged as shown in Figure 10b. All motors for each of the four copters

Figure 16: The effect of range and duty cycle (power) on decoding of real quadcopter signals. The hovering power used is about 50% duty cycle, which can be accurately identified beyond 4m. At more extreme duty cycles, the error increases quickly with distance.

Figure 17: Decoding accuracy for multiple concurrent Bleep signals. There is a weak relationship between distance and decoding accuracy, but good results are possible at long distances.

were kept on at 50% duty cycle, whether or not a transmission was in progress. The quadcopters were assigned center frequencies between 19kHz and 23kHz, with a symbol (bit) length of 0.75s and a chirp rate of 1500Hz/s. The separation between UAVs was varied from 0.25 to 1.5m, so that d_1 to d_4 varied from 0.25 to 4.0m. In each of these configurations, the UAVs transmitted a series of 8-bit messages, with a short pause of random length (uniform between 1.0-2.0 seconds) between messages, so that all the transmissions overlapped. In total, 15 configurations were tested.

Results. The transmission error rate with multiple transmissions for different distances is shown in Figure 17. The decoding accuracy generally decreases with distance, but there were fluctuations similar to the fluctuations seen in the single transmitter case at high duty cycles in Section 5.6. In order to obtain more consistent decoding accuracy, it may be necessary to characterize and model the interference effect of the propeller noise and duty cycle harmonics on the embedded chirp signals before choosing center frequencies.

In general, however, these decoding results show that messages from multiple UAVs can be decoded despite the combined effect of their high-amplitude propeller noise.

6 RELATED WORK

Bleep provides a side-channel messaging system for small, resource-constrained UAVs that can be used when radio connectivity is lost during a sensing task. Available non-RF modalities include light [8, 9], vibration or motion [21, 27], or sound [10, 15, 17, 18].

Sound is suitable for indoor environments where equipment, walls, or other obstacles frequently block line of sight. However, UAVs themselves are known to produce high amplitude sounds during their normal operation [20], which makes sound-based communication difficult. Previous works have tried to characterize this noise in order to suppress it [19], or to uniquely identify a UAV by its sound [31], but have not explored the motor and propeller noise as carriers for communication. In this paper, we extensively characterize and analyze the properties of motor and propeller noise in small UAVs in order to design an accurate communication system with modest computation and storage requirements.

The chirp encoding chosen for the messages in Bleep are common to many communication system, not only for sound-based systems [15], but also in RF [3, 13] and radar [16, 24, 38] systems. This is because chirps are a simple form of frequency modulation; the simplicity makes them easier to detect and extract for decoding, while frequency modulation makes messages more robust to channel noise than if they were directly transmitted with amplitude modulation. Frequency modulation schemes can be broadly classified as narrow-band frequency- or phase-shift keyed methods [10, 17, 18], and wide-band linear chirp modulation methods [3, 13, 15, 33, 36]. Narrow-band methods use two or more distinct frequencies to represent message symbols, but during a given symbol, the frequency of the output is confined to a very narrow frequency band. Linear chirp modulation uses chirped signals whose frequency increases or decreases linearly over time, and each bit occupies a wider band of frequencies. Wide-band (or spread spectrum) techniques are used when SNR is low; therefore wide-band chirp signals are a good fit for sound communication in a noisy channel dominated by UAV motor and propeller noise. Thus, linear chirps were chosen as the frequency modulation method in Bleep.

For detecting and decoding linear chirp signals, there are alternatives to chirp multiplication, such as the fractional Fourier transform (FrFT) [11, 26], which does not have the frequency distortion problem. The FrFT rotates rather than shears the time-frequency bases of a signal, keeping them well separated in time and frequency, but is more computationally demanding and requires a final conversion step to obtain the original center frequency. Chirp multiplication is related to the Fourier and fractional Fourier transforms [4, 14, 28] but is more computationally efficient for detecting and extracting chirps.

7 CONCLUSION

In this paper, we present Bleep, a lightweight sound-based signalling system for small UAVs. We explored using the motors of the UAV to produce the message signals, as the motors and propellers are a source of high-amplitude sound that would otherwise interfere with transmission. We designed a chirp-based encoding scheme after analysis and characterization of motor and propeller sound properties. We also show that this frequency-based encoding does not prevent flight, as the power delivery to the motors is determined by duty cycle, which varies independently. In the presence of noise from multiple UAVs, Bleep allows messages to be transmitted and received in the presence of loud motor and propeller noise with a decoding accuracy of over 95% and a range of over 4.0m.

REFERENCES

- [1] Ferran Adelantado, Xavier Vilajosana, Pere Tuset-Peiro, Borja Martinez, Joan Melia-Segui, and Thomas Watteyne. 2017. Understanding the limits of LoRaWAN. *IEEE Communications Magazine* 55, 9 (2017), 34–40.
- [2] Tobias Antonsson. 2017. Measuring Propeller RPM, Part 3. <https://www.bitcraze.io/2015/02/measuring-propeller-rpm-part-3/>.
- [3] Aloÿs Augustin, Jiazi Yi, Thomas Clausen, and William Townsley. 2016. A study of LoRa: Long range & low power networks for the internet of things. *Sensors* 16, 9 (2016), 1466.
- [4] Richard G Baraniuk and Douglas L Jones. 1993. Shear madness: New orthonormal bases and frames using chirp functions. *IEEE Transactions on Signal Processing* 41, 12 (1993), 3543–3549.
- [5] Bitcraze. 2017. CrazyFlie 2.0. <https://www.bitcraze.io/crazyflie-2/>.
- [6] Javier Ferrer Coll, Per Ångskog, Carl Karlsson, José Chilo, and Peter Stenumgaard. 2010. Simulation and measurement of electromagnetic radiation absorption in a finished-product warehouse. In *2010 IEEE International Symposium on Electromagnetic Compatibility*. IEEE, 881–884.
- [7] Reston Condit. 2010. *Brushed DC Motor Fundamentals*. Microchip Technology, Inc.
- [8] Giorgio Corbellini, Kaan Aksit, Stefan Schmid, Stefan Mangold, and Thomas Gross. 2014. Connecting networks of toys and smartphones with visible light communication. *IEEE Communications Magazine* 52, 7 (2014), 72–78.
- [9] Jarir Fadlullah and Mohsen Kavehrad. 2010. Indoor high-bandwidth optical wireless links for sensor networks. *Journal of lightwave technology* 28, 21 (2010), 3086–3094.
- [10] Thomas Hosman, Mark Yearly, John K Antonio, and Brent Hobbs. 2010. Multi-tone FSK for ultrasonic communication. In *2010 IEEE Instrumentation & Measurement Technology Conference Proceedings*. IEEE, 1424–1429.
- [11] Roshen Jacob, Tessamma Thomas, and A Unnikrishnan. 2009. Fractional Fourier transform based chirp detector versus some conventional detectors. In *2009 International Symposium on Ocean Electronics (SYMPOL 2009)*. IEEE, 56–65.
- [12] Xinyue Kan, Justin Thomas, Hanzhe Teng, Herbert G Tanner, Vijay Kumar, and Konstantinos Karydis. 2019. Analysis of Ground Effect for Small-Scale UAVs in Forward Flight. *IEEE Robotics and Automation Letters* 4, 4 (2019), 3860–3867.
- [13] Junghwan Kim. 1988. Ph.D. Dissertation. Virginia Polytechnic Institute and State University.
- [14] Aykut Koc, Haldun M Ozaktas, Cagatay Candan, and M Alper Kutay. 2008. Digital computation of linear canonical transforms. *IEEE Transactions on Signal Processing* 56, 6 (2008), 2383–2394.
- [15] Patrick Lazik and Anthony Rowe. 2012. Indoor pseudo-ranging of mobile devices using ultrasonic chirps. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems (Sensys '12)*. ACM, Toronto, ON, Canada, 99–112.
- [16] Jun Liu, Hongbin Li, and Braham Himed. 2015. On the performance of the cross-correlation detector for passive radar applications. *Signal Processing* 113 (2015), 32–37.
- [17] Kaikai Liu, Xinxin Liu, and Xiaolin Li. 2012. Acoustic ranging and communication via microphone channel. In *2012 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 291–296.
- [18] Cristina Videira Lopes and Pedro MQ Aguiar. 2003. Acoustic modems for ubiquitous computing. *IEEE pervasive computing* 2, 3 (2003), 62–71.
- [19] Patrick Marmaroli, Xavier Falourd, and Hervé Lissek. 2012. A UAV motor denoising technique to improve localization of surrounding noisy aircrafts: proof of concept for anti-collision systems. In *Acoustics 2012*.
- [20] Jack E Marte and Donald W Kurtz. 1970. *A review of aerodynamic noise from propellers, rotors, and lift fans*. Jet Propulsion Laboratory, California Institute of Technology.
- [21] Rene Mayrhofer and Hans Gellersen. 2007. Shake well before use: Authentication based on accelerometer data. In *International Conference on Pervasive Computing*. Springer, 144–161.
- [22] ST Microelectronics. 2019. *STM32 Cortex®-M4 MCUs and MPUs programming manual*. Retrieved March 12, 2020 from https://www.st.com/content/ccc/resource/technical/document/programming_manual/6c/3a/cb/e7/e4/ea/44/9b/DM00046982.pdf/files/DM00046982.pdf/jcr:content/translations/en.DM00046982.pdf
- [23] Mostafa Mirshekari, Shijia Pan, Adeola Bannis, Yan Pui Mike Lam, Pei Zhang, and Hae Young Noh. 2015. Step-level person localization through sparse sensing of structural vibration. In *proceedings of the 14th international conference on information processing in sensor networks*. ACM, 376–377.
- [24] Michael J Nowak, Zhiping Zhang, Lorenzo LoMonte, Michael Wicks, and Zhiqiang Wu. 2016. Mixed-modulated linear frequency modulated radar-communications. *IET Radar, Sonar & Navigation* 11, 2 (2016), 313–320.
- [25] Mohamed Abou Bakr Othman, John Belz, and Behrouz Farhang-Boroujeny. 2017. Performance analysis of matched filter bank for detection of linear frequency modulated chirp signals. *IEEE Trans. Aerospace Electron. Systems* 53, 1 (2017), 41–54.
- [26] Haldun M Ozaktas and M Alper Kutay. 2001. The fractional Fourier transform. In *Control Conference (ECC)*, 2001 European. IEEE, 1477–1483.
- [27] Shijia Pan, Carlos Ruiz, Jun Han, Adeola Bannis, Patrick Tague, Hae Young Noh, and Pei Zhang. 2018. UniverSense: IoT Device Pairing through Heterogeneous Sensing Signals. In *Proceedings of the 19th International Workshop on Mobile Computing Systems & Applications*. ACM, 55–60.
- [28] Soo-Chang Pei and Jian-Jiun Ding. 2002. Eigenfunctions of linear canonical transform. *IEEE Transactions on Signal Processing* 50, 1 (2002), 11–26.
- [29] Sixu Piao, Zhongjie Ba, Lu Su, Dimitrios Koutsonikolas, Shi Li, and Kui Ren. 2019. Automating CSI Measurement with UAVs: from Problem Formulation to Energy-Optimal Solution. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2404–2412.
- [30] Aveek Purohit, Zheng Sun, Frank Mokaya, and Pei Zhang. 2011. SensorFly: Controlled-mobile Sensing Platform for Indoor Emergency Response Applications. In *ACM/IEEE IPSN*.
- [31] Soundarya Ramesh, Thomas Pathier, and Jun Han. 2019. SoundUAV: Towards Delivery Drone Authentication via Acoustic Noise Fingerprinting. In *Proceedings of the 5th Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*. 27–32.
- [32] PINC Solutions. 2020. Warehouse Drones | PINC | Elevate Your Operation for Better Efficiency. <https://www.pinc.com/warehouse-drone-inventory-management/>. Accessed: 2020-02-06.
- [33] A Springer, W Gugler, M Huemer, L Reindl, CCW Ruppel, and R Weigel. 2000. Spread spectrum communications using chirp signals. In *IEEE/AFCEA EUROCOMM 2000. Information Systems for Enhanced Public Safety and Security* (Cat. No. 00EX405). IEEE, 166–170.
- [34] Charles E Spurgeon. 2000. *Ethernet: the definitive guide*. "O'Reilly Media, Inc".
- [35] Teodor Tomic, Korbinian Schmid, Philipp Lutz, Andreas Domel, Michael Kassecker, Elmar Mair, Iris Lynne Grix, Felix Ruess, Michael Suppa, and Darius Burschka. 2012. Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue. *IEEE robotics & automation magazine* 19, 3 (2012), 46–56.
- [36] Lorenzo Vangelista. 2017. Frequency shift chirp modulation: The LoRa modulation. *IEEE Signal Processing Letters* 24, 12 (2017), 1818–1821.
- [37] Xianjin Xia, Yuanqing Zheng, and Tao Gu. 2019. FTrack: Parallel decoding for LoRa transmissions. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*. 192–204.
- [38] Weiwen Zou, Hao Zhang, Xin Long, Siteng Zhang, Yuanjun Cui, and Jianping Chen. 2016. All-optical central-frequency-programmable and bandwidth-tailorable radar. *Scientific reports* 6 (2016), 19786.