VibroComm: Using Commodity Gyroscopes for Vibroacoustic Data Reception

Robert Xiao
Human-Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, PA, United States & Department of Computer Science, University of British Columbia, Vancouver, British Columbia, Canada

Sven Mayer
Human-Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, PA, United States

Chris Harrison
Human-Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, PA, United States

ABSTRACT

Inertial Measurement Units (IMUs) with gyroscopic sensors are standard in today's mobile devices. We show that these sensors can be co-opted for vibroacoustic data reception. Our approach, called VibroComm, requires direct physical contact to a transmitting (i.e., vibrating) surface. This makes interactions targeted and explicit in nature, making it well suited for contexts with many targets or requiring and intent. It also offers an orthogonal dimension of physical security to wireless technologies like Bluetooth and NFC. Using our implementation, we achieve a transfer rate over 2000 bits/sec with less than 5% packet loss – an order of magnitude faster than prior IMU-based approaches at a quarter of the loss rate, opening new, powerful and practical use cases that could be enabled on mobile devices with a simple software update.

CCS CONCEPTS

• **Human-centered computing** → Sound-based input / output.;
• **Ubiquitous and mobile devices.**;

KEYWORDS

Vibroacoustics, gyroscopes, IMU, smartphone, data transmission, modem, sensing

ACM Reference Format:
Robert Xiao, Sven Mayer, and Chris Harrison. 2020. VibroComm: Using Commodity Gyroscopes for Vibroacoustic Data Reception. In *22nd International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '20), October 05–08, 2020, Oldenburg, Germany*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3379503.3403540

1 INTRODUCTION

With the rise of the Internet of Things (IoT), pairing and transmission schemes that gracefully scale to dense ecosystems of "smart" devices are going to be more important than ever. Unfortunately, wireless schemes (e.g., Bluetooth, NFC, and ultrasound) are ill-suited for this task, as their long range makes interactions less explicit. Mere proximity to a device should not be interpreted as an interaction request. For this reason, most systems often rely on physical buttons to trigger time-limited open connection windows (inherently unsafe), or temporary PINs to make sure others nearby do not connect (needlessly cumbersome).

Our transmission approach takes advantage of the unique characteristics of gyroscopic sensors to show that these ubiquitous sensors can be co-opted as vibroacoustic data receivers. This allows a smartphone (or, in principle, any IMU-equipped device) to be pressed to a surface and receive data encoded as structured vibrations induced by a low-cost transducer (*e.g.*, piezo, and voice coil), which can be embedded in an object. Unlike wireless transmission approaches, which cannot distinguish between devices that are near vs. touching, our approach requires direct physical contact (*i.e.*, acoustic coupling). This unique property makes our approach *targeted* and *explicit* in nature, requiring *intent* to interact. Our approach also requires physical *presence,* offering a useful security dimension that wireless methods cannot (cf. Nishihara *et al.* [25]). This gives our technique a substantially different feel than e.g., opening a camera app on a phone and aiming it towards an object or marker. Additionally, it excels in situations with many dense targets, and also apps wanting higher guarantees of physical presence (e.g., not capturing a QR code through a window or over someone's shoulder).

Our best performing transmission scheme achieved 2028 bits/sec when applying error correction for the 95th percentile of bit error rate. This performance is an order of magnitude faster than prior IMU-based approaches [21] [28] [33]. Our technique is also robust to ambient noise and vibrations – in addition to collecting study data on a static table, we also captured data in two extreme vibration conditions: an airplane in flight and when the receiving smartphone is playing music. We show that VibroComm can be used to improve security, but also rapidly transmit small payloads, like a device ID or URL (e.g., for convenient paring), see Figure 1. Overall, VibroComm has a unique set of properties that make it a valuable addition to the toolbox of data transmission techniques researchers and practitioners can draw upon in the creation of future interactive systems.

In summary, the contributions of this work are as follows:

- VibroComm is the first work to investigate data communication using gyroscope sensors, which are inexpensive, high-speed and common in smartphones today.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
MobileHCI '20, October 05–08, 2020, Oldenburg, Germany
© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-7516-0/20/10…$15.00
https://doi.org/10.1145/3379503.3403540

Figure 1: VibroComm uses the gyroscope sensor found in phones for vibroacoustic data communication. Objects hovering above surfaces receive no data (left). Only when physical contact is made is data received (right). This makes interactions targeted and explicit, requiring presence and intent.

- We implemented and tested four transmission schemes, and rigorously tested them in a head to head evaluation.
- VibroComm is more than 10x faster than the best previous IMU-based (i.e., accelerometer) approach in the literature [28].
- Our higher bitrate offers practically useful packet sizes for real world applications, of which we enumerate many in our Example Use Cases section.

2 RELATED WORK

There is a vast literature on data transmission, both wired and wireless techniques. After a brief note on gyroscopes, we review general data transmission approaches used in mobile devices, namely smartphones, and then focus more specifically on vibroacoustic methods in the literature.

2.1 Gyroscopes

Today's mobile computing devices contain a rich suite of sensors, including gyroscopes for sensing rotational acceleration. Gyros typically operate in concert with an accelerometer and magnetometer as part of an IMU, providing 9-axis orientation data to applications. Of these three sensors, gyroscopes almost always provide the highest raw bandwidth. For instance, modern IMU offerings from Bosch (*e.g.*, BHI160 [6]), InvenSense (*e.g.*, MPU6500 series [17]), and STMicroelectronics (*e.g.*, LSM6DSL [30]) have maximum accelerometer output data rates (ODR) of 1.6, 4.5, and 1.6 kHz respectively. In contrast, gyro ODR on these same sensors is 3.2, 9.0, and 12.6 kHz, respectively.

2.2 Smartphone Data Transmission

Smartphones primarily use wireless radio technologies, such as Bluetooth, WiFi, and cellular radio for data transmission. The most widely deployed smartphone standard for short-range communication is NFC (near-field communication), which uses a low-power radio signal to exchange data with a proximate receiver. Although NFC supports a high bit rate (424 kbits/sec in the fastest configuration), it does not require physical contact between the devices, and thus can be accidentally triggered. Capacitive Near-Field Communication [13] works at a range of about 15cm at 2 kbits/sec. For this reason, current implementations have users manually initiate NFC transmissions on their devices, adding interaction overhead. It has also been demonstrated that wireless approaches are vulnerable to interception by interlopers [7] [11].

Acoustic data transmission techniques, typically using inaudible ultrasonic frequencies, have also been explored in both the academic (*e.g.*, Dhwani [24]) and commercial (*e.g.*, Chirp [2], and AT&T [3]) spheres. Chirp is a commercially deployed approach that transmits data at around 25-100 bits/sec depending on ambient noise characteristics. Like NFC, these acoustic approaches do not require physical contact. Finally, optical techniques for proximate pairing have also been explored (*e.g.*, LightAnchors [1], FlashLight [15], Snap-To-It [9], and CapCam [31]), where the device's camera is used to receive a color-modulated signal. These techniques provide low bandwidth (e.g., 200 bits/sec for CapCam) and require a screen or color LED in order to present the data. Deshotels [10] established a connection using the speaker and microphone of two smartphones enabling a transmission rate with 345 bits/sec in the range above 17 kHz (inaudible to most people). In a similar manner, Nishihara *et al.* [25] achieved 980 bits/sec, while Nittala *et al.* [26] established an broadcasting service at 8 bits/sec.

2.3 Vibroacoustic Data Transmission

More closely related to VibroComm are methods that use vibration to transmit data. Hwang *et al.* [16] used on-off keying at 5 bits/sec to transmit data between two smartphones placed on a surface. Osteoconduct [34] used a similar 5 bits/sec on-off keying approach to transmit vibration-encoded data through the body via bone conduction.

Ripple [28] was the first system to demonstrate robust, high-bitrate vibrational data transmission, achieving 196 bits/sec from a vibrating motor to a mechanically-coupled, external accelerometer. In Ripple II [27], microphones are used for data reception, and the authors achieve up to 30 kbits/sec of bandwidth. The use of microphones necessitated significant amounts of noise filtering and processing to avoid ambient noise, whereas IMUs are inherently much less sensitive to acoustic interference. Of course, IMUs have been used in a wide variety of applications, such as Knocker [12], which can recognize tapped objects, and TapSense [14], which can recognize what part of a finger has touched a screen. EM-Comm [32] used direct touch in combination with modulated electromagnetic emissions enabling transmission *rates* between 22 bits/sec 5.8 kbits/sec depending on the objects. Finally, ViBand [21], SecureVibe [20], and Skin-MIMO [22] explored similar territory as VibroComm, using preexisting, commodity IMUs to receive data, but through a wearer's body.

3 IMPLEMENTATION

With VibroComm, a user presses their phone to an instrumented object, which is vibrating at inaudible ultrasonic frequencies via an internal transducer. Upon physical contact, the vibrations propagate through the phone's chassis, resulting in a signal at the IMU containing an accelerometer and gyroscope. We sample the IMU's

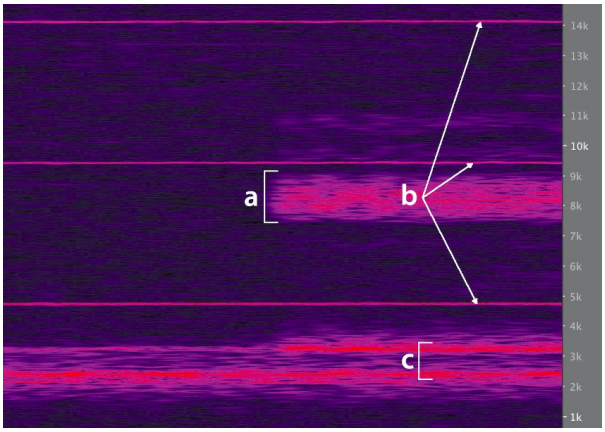

Figure 2: Spectrogram of the gyroscope at a 32.082 kHz sampling rate. (a) Start of a DPSK transmission (19.2 kHz carrier frequency, aliased to ~8.2 kHz). (b) Comb frequency at 4.7 kHz and two harmonics (9.4 and 14.1 kHz). (c) Gyroscope resonant band.

gyroscope at a high sample rate (32 kHz), enabling the device to pick up these vibrations and demodulate them into binary data.

3.1 Receiver Hardware

For the smartphone, we used an LG Nexus 5, a popular low-cost Android smartphone. This smartphone contains an InvenSense MPU-6515 (part of the MPU-6500 series [17], which are among the most common IMUs in the market). This IMU is capable of sampling its accelerometer at 4 kHz and its gyroscope at 32 kHz [18]. The default Linux kernel drivers for Android support a maximum sample rate of 100 Hz, so we wrote a custom kernel driver that configures the IMU into its 32 kHz high-speed sampling mode and disables a low-pass filter. Disabling this filter causes high-frequency vibrations (above the 16 kHz Nyquist frequency) to appear aliased in the sampled data, instead of being attenuated. The driver commands the IMU to store data to an internal First In, First Out (FIFO) buffer, which is emptied using a burst-read command every 20 ms by the kernel driver. The internal I^2C bus is not fast enough to transmit all three axes at 32 kHz. Therefore, we use only the z-axis of the gyro, which was determined to have the highest SNR in a series of informal tests (looking at standard SNR metrics with the phone lying flat on the transducer and also perpendicular with the corner touching, as well as transmission of sinusoids and structured data). However, we note that all three axes work, and the chief reason we selected just one axis (Z) was for limited I^2C bandwidth reasons. In a commercial implementation, it seems likely that fusing data from all three axes (as shown in e.g., [23]) could enable higher performance.

The MPU-6515's gyroscope is a "vibrating-comb" design – a common type of gyroscope in which a micro-scale comb is vibrated at a fixed frequency f_c near a second fixed comb [19] [29], and the capacitance between the two combs is continuously measured. External rotations of the vibrating comb cause it to move out-of-plane, changing the capacitance between the two combs. The comb's fundamental vibration frequency and two of its harmonics are visible in the Spectrogram in Figure 2 b.

When an external oscillation of frequency f is picked up by the gyroscope, the combination of this frequency and the comb frequency produces a pair of *beat frequencies* at $f_c + f$ and $f_c - f$. For instance, in Figure 2 a, a transmission at a frequency of 19.2 kHz is received at an apparent center frequency of 8.2 kHz (19.2 + 4.7 = 23.9 kHz, which becomes 32.1 - 23.9 = 8.2 kHz after Nyquist aliasing). A spectrogram view of the received data is shown in Figure 2. Note that the frequency response of the gyroscope is not flat. The response is strongest around 28 kHz, which appears around 2-3 kHz in the received data (Figure 2 c) due to Nyquist aliasing, and drops off strongly ±8 kHz from this resonant frequency.

Finally, note that the sampling rate of the device is not exactly 32 kHz. In fact, the sampling rate can vary by up to ±750 Hz depending on manufacturing imperfections and the device temperature (the datasheet [17] specifies up to a 1% clock drift). In order to correct for this sample rate variation, the driver reports the nanosecond timestamp and total accumulated byte count when reading from the internal FIFO, allowing software side correction by computing the average sample rate over a period of time. The comb frequency also varies in approximately a 1:1 fashion with the sample rate, affecting the frequency of the received data.

These effects (sample rate instability, beat frequency, resonant band, and vibrating-comb noise) must be compensated for in order to provide the best possible data transmission rate. In particular, note that these effects are not specific or unique to this model of gyroscope – similar effects exist in all gyroscopes of this type.

3.2 Data Transmission

We experimented with several transducer types, including voice coils, piezoelectric discs and linear resonant actuators. We ultimately selected piezo discs for their small size, ability to produce accurate ultrasonic vibrations, extreme low cost (pennies at scale), low power consumption (milliwatts) and thin form factor, allowing them to be integrated into everyday objects. We drive the transducers with modulated ultrasound from the audio jack of a MacBook Pro computer. This consumes 4mW of power at the ports maximum rating of 2 VRMS.

Data is transmitted by outputting one or more waves at known frequencies, called carrier frequencies. The binary data is converted into a stream of *symbols*, which are then modulated onto these carrier waves at discrete intervals. To achieve this modulation, we experimented with several well-known approaches, including amplitude-shift keying (ASK), phase-shift keying (PSK), differential phase-shift keying (DPSK), frequency-shift keying (FSK), and quadrature amplitude modulation (QAM) [4]. Each of these techniques is capable of sending multiple bits per symbol by using different levels; for example, with four amplitude levels, ASK can transmit two bits per symbol. To improve robustness to bit errors, the symbols are encoded into levels using binary Gray code, so that receiving a level as its neighboring level will only induce a single bit-flip.

We also tried using orthogonal frequency division multiplexing (OFDM) to improve resilience and channel utilization, but eventually discarded this approach in favor of a rapidly modulated single-carrier approach because the unpredictable shifts in the gyro's sample rate precluded clean recovery of the orthogonal subcarriers. Similarly, PSK and QAM approaches proved unreliable, as shifts in

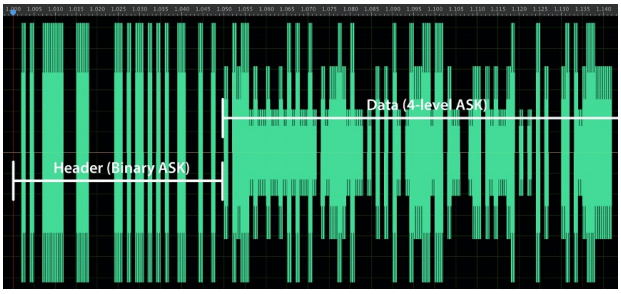

Figure 3: 4-ASK-modulated data, showing the 2-level packet preamble (left) and main 4-level payload.

the sample rate during a message resulted in phase drifts that accumulated over time. However, DPSK, which naturally compensates for phase drift, worked quite well, albeit at the cost of doubling the noise level.

For ASK, PSK, and DPSK, we chose a carrier frequency of 19.2 kHz. For FSK, we chose the frequency range of 17.2-21.2 kHz. In both cases, the goal was to use ultrasound frequencies for inaudibility (above ~17 kHz) and to avoid the resonant band (which is excited by a 22-24 kHz external oscillation). We packetize the data to be sent into one-second duration packets and prepend a header consisting of a fixed 50-symbol sequence; see Figure 3 left. This header allows the start of the message to be detected with very high probability and provides a basis reference (*e.g.*, phase and amplitude references) for decoding the remainder of the message.

3.3 Data Reception

We implemented receive logic for ASK, DPSK, and FSK techniques. Each receiver consists of two parts: a header-search portion to find the beginning (preamble) of the message, and a decoder that decodes the subsequent message using the channel properties identified in the header.

Data reception starts by converting the received data to a constant 32 kHz sample rate using a phase-preserving sinc-based resample filter [8]. Then, for each carrier frequency (1 carrier for ASK, DPSK, 1-8 carriers for FSK), we estimate the received frequency using a device-specific constant for the comb frequency and the received sample rate. Finally, we design and apply a fourth-order Chebyshev band-pass filter to select the carrier. These steps correct for the received frequency offsets due to the comb frequency, and the sample rate drift.

3.4 ASK Decoding

Amplitude-shift keying (ASK) modulates the amplitude of the carrier frequency. The header is modulated using binary (two-level) ASK for ease of recognition. To find the header, we scan across the filtered data, fitting sliding windows to this ideal ASK-modulated preamble. To fit a sliding window, we split the window into symbol-sized units of length N and compute the amplitude of the received carrier frequency f_{cr}:

$$amplitude = \left| \sum_{n=0}^{N} x_n \, e^{2\pi n i / f_{cr}} \right|$$

where x_n is the n^{th} sample in the symbol. We use the median of the amplitudes within the window as a threshold to produce a sequence of binary bits and compare this sequence to the preamble. If fewer than 8 errors (16%) are found - a 1 in 9.5 million chance of occurring at random - we assume we have located a header. We then refine the header position by looking for the nearby window with the highest separation in amplitudes between '0' bits and '1' bits. Decoding then proceeds to walk over sliding windows of the data portion, computing amplitudes relative to the reference amplitudes from the header, and outputting the relevant discrete symbol (after undoing the Gray code).

3.5 PSK & DPSK Decoding

Phase-shift keying (PSK) modulates the phase of the carrier frequency. Differential phase-shift keying (DPSK) also codes data using phase, but as *differences* between phases in adjacent symbols. The latter is significantly more robust to phase drift issues than ordinary PSK, but it reduces the signal-to-noise ratio (SNR) by compounding errors from adjacent symbols.

Because the sample rate drifts with time, and the signal is aliased across the (drifting) Nyquist boundary, the received phase of the carrier frequency drifts with time. We use a first-order temporal approximation to the phase drift, *i.e.,* a linear model $\theta'_i = \theta_i + i\Delta\theta$ where θ_i is the original phase at symbol i, θ' is the received phase, and $\Delta\theta$ is the average phase drift. The presence of this accumulating phase drift is what causes uncorrected PSK data reception to fail. Even if it is carefully estimated, small variances will compound over the length of a message. In contrast, small variances in the phase drift have only small impacts on DPSK due to its differential nature.

To find a PSK-encoded header, we scan across the filtered data and fit sliding windows to the ideal preamble. To fit a sliding window to the PSK preamble, we compute the phase of each symbol:

$$\phi'_i = arg\left(\sum_{n=0}^{N} x_n \, e^{2\pi n i / f_{cr}} \right)$$

We then compute the estimated phase drifts $\phi_i = \phi'_i - \theta_i$ (where θ_i are the phases of the preamble). If these drifts were not angles, one could use least-squares to estimate $\Delta\theta$ and θ_0 using the linear relation $\phi_i = i\Delta\theta + \theta_0$. However, circular wraparound prevents this approach from working. Instead, we make an initial estimate using

$$\Delta'\theta = \left(\sum_{i=1}^{k-1} \phi_i - \phi_{i-1} \right) / (k-1)$$

where $k = 50$ (the length of the preamble), and then apply linear least-squares to

$$\phi_i + 2\pi \left[\frac{i\Delta'\theta - \varphi_i}{2\pi} \right] = i\Delta\theta + \theta_0$$

where [ ] is the rounding operator. This reconstructs the "unwrapped" phase values, allowing us to recover the phase drift accurately. It works as long as the initial estimate is off by no more than $2\pi/k$ radians. Once we have recovered an estimated phase drift, we can binarize the adjusted phases in the window and match them to the PSK payload. As before, we declare that we have found a header

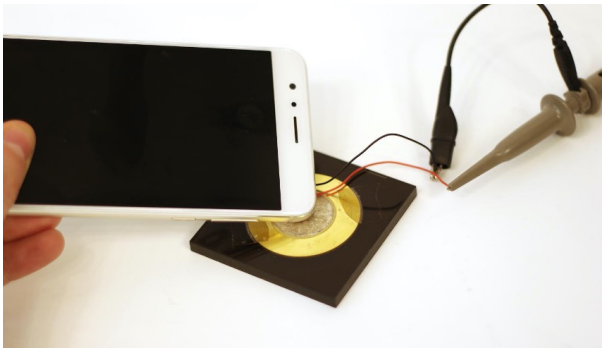

Figure 4: A phone being touched to a piezo transducer.

if fewer than 8 bits are in error. Decoding a DPSK transmission consists of walking over the sliding windows of the data portion, computing phases relative to the prior window, and outputting the corresponding symbol.

3.6 FSK Decoding

Frequency shift keying (FSK) switches between different carrier frequencies depending on the symbol. The FSK header consists of a fixed random permutation h of [0, 1, 2, ..., 49] modulo the number of levels, ensuring an even distribution of frequencies for sampling purposes. To find the header, we fit sliding windows to the preamble by comparing the amplitudes of each carrier frequency c in each symbol i, producing a sequence of vectors v_i. To account for variable frequency sensitivity and non-uniform background noise, we construct a set of *reference vectors*

$$r_{c,c} = \overline{v_{i,c} \text{ where } h_i = c}$$

$$r_{c,d} = \overline{v_{i,d} \text{ where } h_i \neq c} \text{ for } c \neq d$$

where $\bar{x}$ indicates the mean of x. Then, to extract FSK symbols from an amplitude vector, we simply find the closest r_c vector (by Euclidean distance) and output c, which we do for both the header and the payload.

4 EVALUATION

We ran an experiment to validate the robustness and performance of VibroComm (Figure 4). In order to ensure we were not over-fitting the system, we tested the system with three different Nexus 5s, one of which was never used during the development of the system. We calibrated the three devices by obtaining their baseline comb-sample rate offsets but otherwise did not engage in any device-specific configuration.

We used a 27 mm piezo bender (PUI Audio, 668-1013-ND) as a proof-of-concept transmitter. As noted earlier, VibroComm is intended for physical interactions with presence and intent, and thus the phone was placed flat, centered and on top of the transmitter coupled by gravity. In general, we found an extremely light touch will not transmit enough signal, while pressing too hard can prevent the transducer from flexing. We found the typical range of how one would press a phone to a surface to work, with only unnatural corner cases remaining.

We collected data for each combination of *modulation* (FSK, ASK, PSK, and DPSK), *bits per symbol* (1, 2, 3, 4, with the exception of FSK which only went up to 3), and *symbol rate* (500, 1000, and 1250 symbols/sec). In total, there were 45 conditions. For each condition, we generated 25 different codes, and tested each code 6 times, for a total of 150 data points per condition per device. Each sample consisted of a header plus one second of data, followed by silence for segmentation. In total, we collected 20,250 instances over the course of 28 hours of data transfer, representing 44,550,000 bits of vibro-acoustically transmitted data.

We additionally collected 2,334 instances of data (~5M bits) in flying aircraft (Boeing 787 Dreamliner). An analysis of the bit error rate in this data set showed no significant differences, so it was merged with the main study set. We also collected 270 instances of data with music playing at maximum volume out of the phone's internal speakers. An analysis of this data showed an increase in bit errors for ASK decoding, but no significant effect on FSK or DPSK techniques.

5 RESULTS

We found no difference in accuracy across our three test devices, and so the data were merged for analysis. Of the 22,584 packets transmitted, 78 (0.3%) were not detected at all (*i.e.* the header was not found). The vast majority (73) of these errors were in the 1250 Hz symbol rate condition. In the 500 Hz and 1000 Hz conditions combined, there were only 5 missed packets out of 15,032 (a detection rate of 99.96%).

Table 1 shows the 95th percentile bit error rates across all conditions (*i.e.* 95% of packets had lower error rates). For PSK, most conditions quickly yield bit error rates approaching 50%, which is as bad as random noise. This is primarily due to the problem of phase drift. DPSK modulation performs significantly better.

The 95th percentile bit error rates are useful for calculating the *effective* bit rate, after subtracting the necessary error correction overhead to receive 95% of all packets successfully. A typical analysis uses a correction percentage of twice the error rate, and coding schemes such as Turbo Codes [5] achieve this rate in practice.

The best overall condition was FSK modulation at 3 bits per symbol, 1000 symbols per second, which is a raw bit rate of 3000 bits/sec. With a 95th percentile BER of 16.20%, the effective bit rate is 2028 bits/sec. In other words, we can transmit error-corrected data at 2028 payload bits/sec and expect that 95% of packets will be received successfully. For DPSK, the best condition was 4 bits per symbol and 1000 symbols per second. With a BER of 26.65%, this is an effective bit rate of 1868 bits/sec. With ASK, the best condition was 2 bits per symbol and 1250 symbols/sec (BER 17.80%), for an effective bit rate of 1627 bits/sec.

For reference, Ripple [28] – the fastest IMU-based transmission technique we could find in the literature – used 80th percentile BER to obtain an effective bit rate of 196.6 bits/sec. This latter was achieved using an IMU placed at the end of a rigid, cantilevered arm to amplify vibrations. Additionally, (although not a direct point of comparison, having to transmit vibrations through the human body), ViBand [21] was 155 bits/sec at 80th percentile BER and SecureVibe [20] ran at 20 bits/sec.

Table 1: 95th percentile bit error rates.

bits/symbol @ symbols/sec	ASK	DPSK	FSK	PSK
1 @ 500	0.00%	0.20%	0.00%	0.00%
1 @ 1000	2.20%	0.20%	4.40%	0.10%
1 @ 1250	2.16%	6.64%	15.20%	47.68%
2 @ 500	4.90%	0.30%	0.00%	38.10%
2 @ 1000	19.05%	10.05%	8.25%	46.75%
2 @ 1250	17.80%	23.04%	13.60%	58.24%
3 @ 500	17.87%	4.40%	1.40%	51.07%
3 @ 1000	30.00%	19.60%	16.20%	52.67%
3 @ 1250	29.92%	32.99%	23.76%	58.24%
4 @ 500	26.50%	11.40%	-	55.20%
4 @ 1000	35.08%	26.65%	-	54.20%
4 @ 1250	35.70%	39.04%	-	57.52%

5.1 Transducer Integration

Although we initially envisioned that piezo transducers would have to be integrated into devices to support VibroComm, we discovered that some devices are capable of producing controlled vibrations using their existing internal speakers (e.g., wireless speakers, television sets, conference room speakerphones). Figure 7 offers proof-of-concept signals from two uninstrumented devices. We also suspect that devices with high-quality vibration actuators (i.e., linear resonant actuators, but not eccentric rotating mass vibration motors) would also work. We look forward to exploring this promising avenue in future work.

We also had some limited success in actuating large surfaces, such as a sheet of architectural glass, meaning that a whole window could be VibroComm-capable (Figure 11). However, we note that transducer power was so high that it produced audible harmonics, which is undesirable. In general, we found the best performance when pressing phones directly to a transducer or when attaching a transducer to the underside of the interactive area (*e.g.*, in the reader head of a payment terminal, Figure 5 and Video Figure).

In tests placing transducers elsewhere (*e.g.*, corner of a wood table, and side of an office printer), we found that damping and multipath obfuscated the signal significantly, dramatically increasing bit error rate. We were most successful when using the resonant frequency of the gyroscope, though even still bandwidth was reduced by an order of magnitude. Nonetheless, this suggests that resonance-based transmission schemes might be viable and merits future work. Example Use Cases

Applications requiring one-way transfers of small data payloads are ideal for VibroComm. Additionally, scenarios requiring stronger guarantees of physical presence could also use VibroComm as an extra layer of defense. We now summarize some example uses, see Video Figure.

5.2 Payments

Payments are perhaps the most straightforward example use (e.g., groceries, and parking; Figure 5). One option would be to combine VibroComm with, e.g., NFC, and Bluetooth, offering a combined method that is fast and explicit, with greater security. However,

Figure 5: Devices could emit a unique transaction identifier, which could be received by VibroComm-enabled devices, allowing for rapid payment.

VibroComm could also be used in lieu of technologies like NFC. Even a short tap (~100ms, or around 200 bits) would be sufficient to transfer a unique transaction identifier and ephemeral encryption key, allowing the transaction to be completed securely over a WiFi or cellular connection. Although security is not the focus of our paper, we hypothesize that VibroComm would be significantly harder to spoof compared to wireless techniques like NFC (interception of signals is possible, but would require specialized equipment for long range vibrometry, e.g., Zhang *et al.* [33]).

5.3 Device Pairing

Today, pairing devices using technologies like Bluetooth is a cumbersome affair, often requiring expiring PINs to increase security. Part of the problem with wireless methods is the lack of guarantees that devices are actually next to one another (and not e.g., in a neighbor's apartment). Even "near field" technologies such as NFC have trouble disambiguating between true touch and being nearby, which is problematic in e.g., busy retail settings. VibroComm offers stronger guarantees of physical proximity, as well as a pairing interaction that is much more explicit and intuitive: devices must be physically held together (e.g., a smartphone, and Bluetooth speaker; Figure 6).

5.4 Ephemeral Pairing

Permanent device-device pairing is not required for many use cases, especially interactions that might only last for minutes or hours.

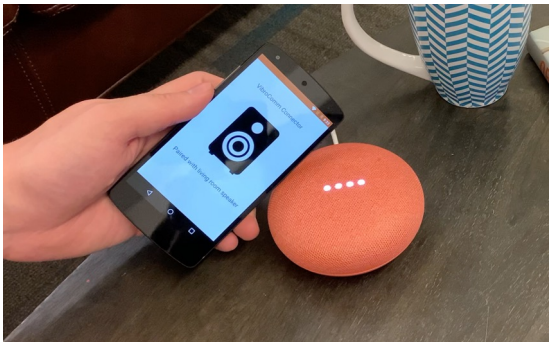

Figure 6: Devices could be intuitively paired via VibroComm.

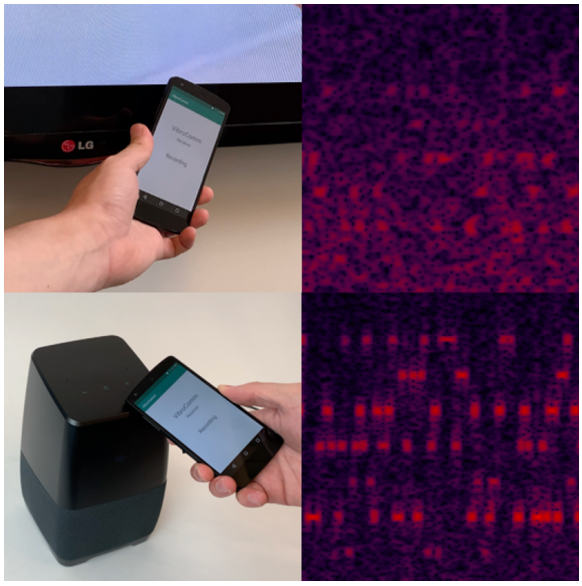

Figure 7: Devices containing speakers – such as televisions (top) and wireless speakers (bottom) – can be used to emit structured vibrations without modification, though the signals are generally noisier and weaker, as seen in these examples.

One example of this type of short-lived pairing is guest WiFi access, where a user could touch the access point to confirm they are inside e.g., a cafe, which transmits a one-time token for temporary internet access; see Figure 8

5.5 Sharing Connection Details

Another class of pairing occurs with semi-public, shared devices. A canonical example are office printers – with VibroComm, the front panel could repeatedly transmit all necessary model and network details to setup printing, including extra security credentials if needed; see Figure 9. Such an approach would inherently only allow those with physical access to the premise access to the printer.

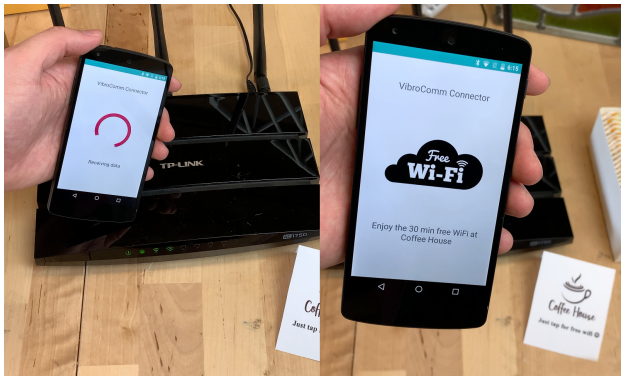

Figure 8: Temporary guest WiFi access could be as easy as tapping one's phone to a café's router.

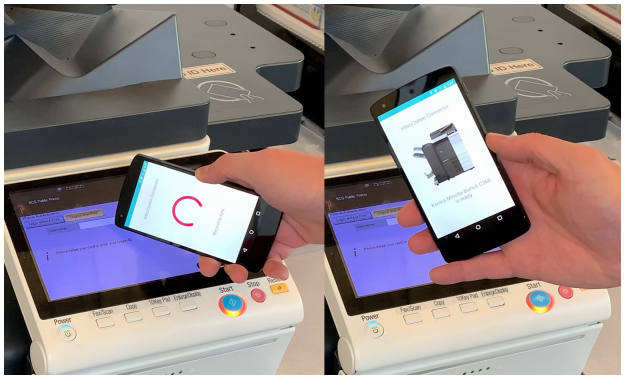

Figure 9: Devices, such as printers, could share their model and networking details, facilitating easy setup.

5.6 Device Handoff

There are also instances where a function or process on a device might wish to be transferred to another. For example, transferring a call one took on a smartphone to a conference room speakerphone (Figure 10, left) or transferring a video stream to a larger screen.

5.7 Launching Companion Apps

VibroComm could be especially useful for rapidly launching companion apps on smartphones. For example, touching one's smartphone to a smart thermostat launches the manufacturers app, offering touchscreen controls and advanced settings; see Figure 10, right.

5.8 File Transmission

Although VibroComm is not suitable for transmitting large files, it could transmkit server details (e.g., IP/port/password) or a URL that allows for download over faster means, such as WiFi of cellular. This could allow, for instance, a user to download a menu from a restaurant window; see Figure 11

6 LIMITATIONS

The most immediate limitation of our technique is that it can only receive data. Thus, it would have to be combined with another transmission method (e.g., WiFi, Bluetooth, or cellular) for many

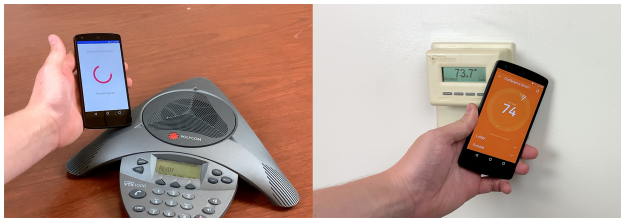

Figure 10: Left: VibroComm could be used to handoff functionality from one device to another, as seen with this conference room speaker. Right: Full-featured apps could be automatically launched in response to a user touching their phone to a smart device, for example, smart thermostats.

Figure 11: VibroComm can be used on large, ridged surfaces, such as glass windows. In this example, a URL is transmitted, which opens a restaurant's menu.

applications. We also note that the vibratory signals we use are very subtle and poor acoustic coupling will render the signal unrecoverable. For example, we found that smartphone cases can reduce the signal by 10 dB or more (especially if made from a very compliant material like silicone; hard plastic cases are less impacted). Also, as there are innumerable IMU models and vendors, it is hard to predict how well our approach scales to different hardware. From our experiences playing with perhaps a dozen phones, we did not see a dramatic effect on IMU location inside the device. Corners, often being more rigid, did appear to better propagate the signal across the entire chassis than the middle of the phone back, but this again is likely to be phone dependent. However, the fundamental principles of operation should be applicable to all gyroscopes in any mobile device (as they are designed to faithfully digitize a physical phenomena), and we believe any IMU that offers sufficiently high sampling rate should be applicable.

7 CONCLUSION

We demonstrated that we can transmit error-corrected data at 2028 bits/sec with the expectation that 95% of packets will be successfully received, using nothing more than a commodity gyroscope sensor found in today's smartphones. Above all, we believe this work brings to light a curious new feature for a now ubiquitous sensor and could unlock new experiences on devices with little more than a software update. While VibroComm will never compete with technologies like WiFi, Bluetooth, and NFC, it offers an orthogonal dimension that make it a valuable addition to the toolbox of data transmission techniques available for enriching future interactive systems. We concluded our paper showcasing a variety of powerful use cases enabled VibroComm.

REFERENCES

[1] Karan Ahuja, Sujeath Pareddy, Robert Xiao, Mayank Goel, and Chris Harrison. 2019. LightAnchors: Appropriating Point Lights for Spatially-Anchored Augmented Reality Interfaces. In Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19). Association for Computing Machinery, New York, NY, USA, 189–196. DOI: https://doi.org/10.1145/3332165.3347884

[2] Asio Ltd. "Chirp". https://chirp.io

[3] AT&T. Bio-acoustic Data Transfer: Key Transmission through Bone Conduction. https://www.att.com/gen/press-room?pid=22683.

[4] Bellamy, J. C. Digital Telephony, 3rd Edition. Chapter 6.1 Digital Modulation, pp. 279-308.

[5] Claude Berrou, Alain Glavieux, and Punya Thitimajshima. 1993. Near Shannon limit error-correcting coding and decoding: Turbo-codes. *Communications, 1993. ICC '93 Geneva. Technical Program, Conference Record, IEEE International Conference on*, Geneva. DOI: https://doi.org/10.1109/ICC.1993.397441

[6] Bosch SensorTec. BHI160 datasheet. https://www.mouser.com/datasheet/2/783/BST-BHI160B-DS000-01-Datasheet-1220916.pdf

[7] Tim WC Brown, Thomas Diakos, and Johann A. Briffa. 2013. Evaluating the eavesdropping range of varying magnetic field strengths in NFC standards. In Proceedings of the 7th European Conference on Antennas and Propagation (EuCAP '13), IEEE.

[8] de Castro Lopo, E. libsamplerate (software). http://www.mega-nerd.com/SRC/index.html

[9] de Freitas, Adrian A., Michael Nebeling, Xiang "Anthony" Chen, Junrui Yang, Akshaye Shreenithi Kirupa Karthikeyan Ranithangam, and Anind K. Dey. 2016. Snap-To-It: A User-Inspired Platform for Opportunistic Device Interactions. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16). ACM, New York, NY, USA, 5909–5920. DOI: https://doi.org/10.1145/2858036.2858177

[10] Luke Deshotels. 2014. Inaudible sound as a covert channel in mobile devices. In 8th {USENIX} Workshop on Offensive Technologies ({WOOT} 14).

[11] Thomas P. Diakos, Johann A. Briffa, Tim WC Brown, and Stephan Wesemeyer. 2013. Eavesdropping near-field contactless payments: a quantitative analysis. The Journal of Engineering 2013, no. 10 (2013): 48-54.

[12] Taesik Gong, Hyunsung Cho, Bowon Lee, and Sung-Ju Lee. 2019. Knocker: Vibroacoustic-based Object Recognition with Smartphones. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 3, 3, Article 82 (September 2019), 21 pages. DOI: https://doi.org/10.1145/3351240

[13] Tobias Grosse-Puppendahl, Sebastian Herber, Raphael Wimmer, Frank Englert, Sebastian Beck, Julian von Wilmsdorff, Reiner Wichert, and Arjan Kuijper. 2014. Capacitive near-field communication for ubiquitous interaction and perception. In Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '14). Association for Computing Machinery, New York, NY, USA, 231–242. DOI: https://doi.org/10.1145/2632048.2632053

[14] Chris Harrison, Julia Schwarz, and Scott E. Hudson. 2011. TapSense: enhancing finger interaction on touch surfaces. In Proceedings of the 24th annual ACM symposium on User interface software and technology (UIST '11). Association for Computing Machinery, New York, NY, USA, 627–636. DOI: https://doi.org/10.1145/2047196.2047279

[15] Tobias Hesselmann, Niels Henze, and Susanne Boll. 2010. FlashLight: optical communication between mobile phones and interactive tabletops. In ACM International Conference on Interactive Tabletops and Surfaces (ITS '10). Association for Computing Machinery, New York, NY, USA, 135–138. DOI: https://doi.org/10.1145/1936652.1936679

[16] Inhwan Hwang, Jungchan Cho, and Songhwai Oh. 2012. Privacy-aware communication for smartphones using vibration. In Proceedings of the 18th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA '12). IEEE. DOI: https://doi.org/10.1109/RTCSA.2012.43

[17] InvenSense. MPU6500 Datasheet. https://www.invensense.com/wp-content/uploads/2015/02/MPU-6500-Datasheet2.pdf

[18] InvenSense, Inc. MPU-6500 Register Map And Descriptions Revision 2.1. September 16, 2013. https://www.invensense.com/wp-content/uploads/2015/02/MPU-6500-Register-Map2.pdf

[19] Ville Kaajakari. 2009. Practical mems: Design of microsystems, accelerometers, gyroscopes, RF MEMS, optical MEMS, and microfluidic systems. Las Vegas, NV: Small Gear Publishing .

[20] Younghyun Kim, Woo Suk Lee, Vijay Raghunathan, Niraj K. Jha, and Anand Raghunathan. 2015. Vibration-based secure side channel for medical devices. In Proceedings of the 52nd Annual Design Automation Conference (DAC '15). Association for Computing Machinery, New York, NY, USA, Article 32, 1–6. DOI: https://doi.org/10.1145/2744769.2744928

[21] Gierad Laput, Robert Xiao, and Chris Harrison. 2016. ViBand: High-Fidelity Bio-Acoustic Sensing Using Commodity Smartwatch Accelerometers. In Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16). Association for Computing Machinery, New York, NY, USA, 321–333. DOI: https://doi.org/10.1145/2984511.2984582

[22] Dong Ma, Yuezhong Wu, Ming Ding, Mahbub Hassan, and Wen Hu. 2020. Skin-MIMO: Vibration-based MIMO Communication over Human Skin. arXiv preprint arXiv:2001.11574.

[23] Denys J.C. Matthies, Marian Haescher, Suranga Nanayakkara, and Gerald Bieber. 2018. Step Detection for Rollator Users with Smartwatches. In Proceedings of the Symposium on Spatial User Interaction (SUI '18). Association for Computing Machinery, New York, NY, USA, 163–167. DOI: https://doi.org/10.1145/3267782.3267784

[24] Rajalakshmi Nandakumar, Krishna Kant Chintalapudi, Venkat Padmanabhan, and Ramarathnam Venkatesan. 2013. Dhwani: secure peer-to-peer acoustic NFC. In Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM (SIGCOMM '13). Association for Computing Machinery, New York, NY, USA, 63–74. DOI: https://doi.org/10.1145/2486001.2486037

[25] Saki Nishihara, Naoki Shinmen, Tadashi Ebihara, Koichi Mizutani, and Naoto Wakatsuki. 2017. Design of secure near-field communication for smartphones using sound and vibration. In Proceedings of the 6th Global Conference on Consumer Electronics (GCCE '17). IEEE. DOI: https://doi.org/10.1109/GCCE.2017.8229356

[26] Aditya Shekhar Nittala, Xing-Dong Yang, Scott Bateman, Ehud Sharlin, and Saul Greenberg. 2015. PhoneEar: interactions for mobile devices that hear

high-frequency sound-encoded data. In Proceedings of the 7th ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '15). Association for Computing Machinery, New York, NY, USA, 174–179. DOI: https://doi.org/10.1145/2774225.2775082

[27] Nirupam Roy, and Romit Roy Choudhury. 2016. Ripple II: faster communication through physical vibration. In *Proceedings of the 13th Usenix Conference on Networked Systems Design and Implementation* (NSDI'16). USENIX Association, Berkeley, CA, USA, 671-684.

[28] Nirupam Roy, Mahanth Gowda, and Romit Roy Choudhury. 2015. Ripple: communicating through physical vibration. In *Proceedings of the 12th USENIX Conference on Networked Systems Design and Implementation* (NSDI'15). USENIX Association, Berkeley, CA, USA, 265-278.

[29] Joe Seeger, Martin Lim, and Steve Nasiri. Development of High-Performance, High-Volume consumer MEMS Gyroscopes (whitepaper). InvenSense, Inc. Sunnyvale, CA, USA.

[30] STMicroelectronics. LSM6DSL datasheet. http://www.st.com/content/ccc/resource/technical/docment/datasheet/ee/23/a0/dc/1d/68/45/52/DM00237456.pdf/files/DM00237456.pdf/jcr:content/translations/en.DM00237456.pdf

[31] Robert Xiao, Scott Hudson, and Chris Harrison. 2016. CapCam: Enabling Rapid, Ad-Hoc, Position-Tracked Interactions Between Devices. In Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces (ISS '16). Association for Computing Machinery, New York, NY, USA, 169–178. DOI: https://doi.org/10.1145/2992154.2992182

[32] Chouchang Jack Yang and Alanson P. Sample. 2017. EM-Comm: Touch-based Communication via Modulated Electromagnetic Emissions. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 1, 3, Article 118 (September 2017), 24 pages. DOI: https://doi.org/10.1145/3130984

[33] Yang Zhang, Gierad Laput, and Chris Harrison. 2018. Vibrosight: Long-Range Vibrometry for Smart Environment Sensing. In Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology (UIST '18). ACM, New York, NY, USA. DOI: https://doi.org/10.1145/3242587.3242608

[34] Lin Zhong, Dania El-Daye, Brett Kaufman, Nick Tobaoda, Tamer Mohamed, and Michael Liebschner. 2007. OsteoConduct: wireless body-area communication based on bone conduction. In Proceedings of the ICST 2nd international conference on Body area networks (BodyNets '07). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Brussels, BEL, Article 9, 1–8.