I Can Hear More: Pushing the Limit of Ultrasound Sensing on Off-the-Shelf Mobile Devices

Yuchi Chen*, Wei Gong*, Jiangchuan Liu* and Yong Cui†
*School of Computing Science, Simon Fraser University, Canada
†Department of Computer Science and Technology, Tsinghua University, China
Email: yuchi_chen@sfu.ca, gongweig@sfu.ca, jcliu@cs.sfu.ca, cuiyong@tsinghua.edu.cn

***Abstract*—Recent years have seen various acoustic applications on mobile devices, e.g. range finding, gesture recognition, and device-to-device data transport, which use near-ultrasound signals at frequencies around 18-24 kHz. Due to the fixed low sound sample rate and hardware limitation, the highest detectable sound frequency on commercial-off-the-shelf (COTS) mobile devices is capped at 24 kHz, presenting a daunting barrier that prevents high-frequency ultrasounds from benefiting acoustic applications. To bridge this gap, we present iChemo, a technology that enables COTS mobile devices to sense high-frequency ultrasound signals. Specifically, we demonstrate how to detect the power spectral density (PSD) of a high-frequency ultrasound signal by customizing the coprime sampling algorithm on COTS devices. Through our prototype and evaluation on extensive mobile devices, we demonstrate that iChemo can sense the PSD of ultrasound at frequency of 60 kHz, which is over twice of the current sensible frequency threshold.**

I. Introduction

Smart mobile devices, including smartphones, tablets and smart watches, have become increasingly indispensable for today's communication and entertainment. Besides the conventional applications, mobile devices have also been explored to support more novel functions with their sensing capabilities. One major exploration is sound sensing for other purposes beyond voice communications, such as range finding [1], movement tracking [2], in-door localization, and even near-field data transport [3], [4].

A fundamental step in sound sensing is measuring the *Power Spectrum Density* (PSD) of sound signals. PSD essentially illustrates the sound's power distribution throughout a certain frequency band. Since sounds can be easily interfered by the environment or even buried by ambient noises, PSD measurement is necessary to characterize sound signals of interest. Other analysis on the target sounds, such as extracting the signal waveform, detecting the power changes or measuring frequency shifts, are further enabled. Therefore, PSD is fairly useful in sound sensing apps, especially those built on amplitude or frequency modulation [2], [5].

Figure 1 gives an example of the PSD measurement, which is done through a Fourier analysis on a 40 kHz *ultrasound* signal. With ultrasounds at this frequency, commercial ultrasound range finders are able to reach a resolution of 1 mm. Since higher frequency typically means better resolution and accuracy, ultrasound has long been widely used in industry [6], and it is generally preferred in practice. Basically, sounds at frequencies below 18 kHz are audible to humans. When the frequency goes into the near-ultrasound band, i.e. 19-24 kHz, the sound becomes inaudible to most people, but it is still recordable by mobile devices. Many existing sound sensing apps work with sound signals in this band, in order to avoid distracting users while preserving good performance.

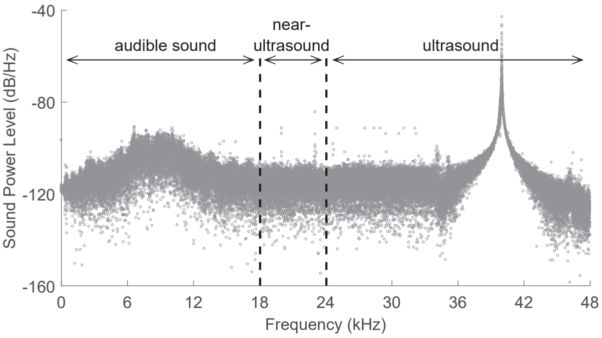


Fig. 1. PSD of a 40 kHz ultrasound signal

Unfortunately, ultrasounds at frequencies higher than 24 kHz are not yet available to existing sound sensing apps. The main reason is that, the highest sound sampling rate supported by most *commercial-off-the-shelf* (COTS) smart mobile devices is only 44.1 kHz or 48 kHz. According to the Nyquist-Shannon Sampling Theorem [7], they can not record any ultrasound at a frequency over 24 kHz.

For COTS mobile devices, their built-in sound recording system consists of two key components, i.e. a *Micro Electromechanical System* (MEMS) microphone and an *Analog-to-Digital Converter* (ADC). They are designed for recording audible sounds; hence the upper sample rate limit of 44.1 or 48 kHz comes. In addition, before the ADC can get the analog signal from the MEMS microphone, there is an *Anti-Aliasing Filter* (AAF) eliminating inaudible parts from the signal. This makes the sensitivity to sounds decay even more quickly before the sound frequency reaching 24 kHz, the theoretical upper limit. Figure 2 illustrates the *frequency response* of some popular mobile devices we have measured. This frequency response is basically a PSD measurement, reflecting how sensitive a mobile device can be to sounds at different frequencies. As is shown in the figure, their frequency response curves are far from being flat within the audible band. Particularly, their sensitivity significantly decreases from

978-1-5386-4128-6/18/$31.00 ©2018 IEEE

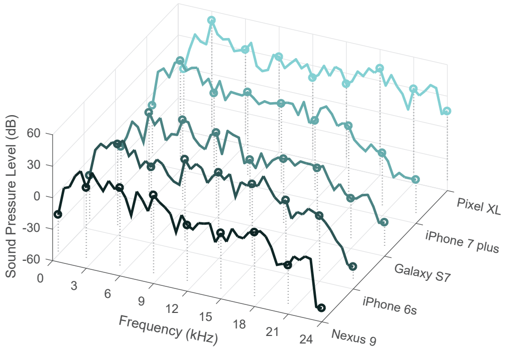


Fig. 2. Frequency response of several popular mobile devices

a certain point in the near-ultrasound band, such as 18 kHz for Google/HTC Nexus 9, and 20 kHz for Apple iPhone 6s. That said, existing apps that are based on inaudible sounds can only operate in a narrow near-ultrasound band, and the sharp decay not only hinders the app effectiveness but also complicates their implementations. In order to work with sounds in as larger band as possible, some projects even have no choice but to involve sounds in the audible band [8], [3], facing the risk of bringing auditory impact on users.

Given that the optimization of audible sound recording and cost saving is the first priority for mobile device manufacturers, we do not expect to see great improvement on the physical ultrasound sensing capability, even in the long run. However, through novel sampling and processing strategies, in particular, coprime sampling [9] with non-linearity feature and time-domain alignment, the theoretical and physical limits will not prevent COTS mobile devices from estimating the PSD of high-frequency ultrasounds, which are beyond their capacity to fully capture.

To this end, we present *iChemo* [1] that pushes the limit of ultrasound detection on COTS smart mobile devices to a much higher level. Our contributions in this paper can be summarized as follows:

- For the first time we enable the PSD measurement of sparse high-frequency ultrasound signals on COTS mobile devices, without any hardware extension. The highest sensible frequency can be as high as 60 kHz, which is over 2× of 24 kHz, the upper limit of existing sound sensing apps.
- To the best of our knowledge, for the first time we employ the coprime sampling algorithm on COTS mobile devices to achieve reliable PSD measurement on high-frequency ultrasounds.
- Although coprime sampling has been used on sparse electromagnetic signals [10], it is still challenging for practical use since multiple synchronized ADCs are required, which is also the key limit in our earlier work [11]. Yet, observing that the non-linearity feature can result in a *phantom signal* of the target sound, we reveal that it is similar to the alias signal caused by sampling at sub-Nyquist rate. Particularly, by resolving the time-domain alignment of the phantom and alias signals, we show that coprime sampling is feasible on a standalone COTS mobile device, where only one built-in ADC is available.

As a proof-of-concept, we have implemented the prototype of iChemo on two popular Android devices, the Nexus 9 tablet and the Pixel XL smartphone. We have also conducted extensive tests on iOS devices, such as Apple iPhone 6s and iPhone 7 plus. Through prototyping and evaluations, we show that iChemo is readily applicable on a wide range of mobile devices, while preserving good performances. In particular, iChemo is able to measure the PSD of an ultrasound at 60 kHz with a mean accuracy of 96.63%. By pushing the limit of sensing high-frequency ultrasounds, iChemo can instantly boost the performance of sound sensing apps that are built on PSD measurement. We also discuss the potential extension of iChemo on phase measurement, which is another foundation in many sound sensing apps.

II. Background and Related Work

We focus on pushing the limit for PSD measurement, which is a key step in a wide range of sound sensing apps. Based on the difference of sound sources, these apps fall into two categories: (1) sensing designated sound signals, and (2) sensing ambient sounds. It is however worth noting that our solution can also be extended for phase measurement, which we will further discuss in Section VI.C.

A. Sensing Designated Sound Signals

PSD measurement, by its nature, directly reflects the power distribution of sound signals. Therefore, one major usage of PSD measurement is to estimate the power change of certain sounds. For example, in an early effort [12], PSD measurement is applied to monitor the power change of multiple sparse sound signals, of which the power is modulated to carry data bits. This method is also known as the Amplitude Shift Keying (ASK). Similarly, Dolphin [3] measures the PSD of ASK-modulated signals, and it is able to achieve higher data rate with wider bandwidth. However, the power change of signals may not reliably detected if the ambient noise is too strong. Realizing this fact, authors of [13] uses the appearance and absence of certain signals to denote data bits.

Power change is not only feasible for data transport. By monitoring the power change in the PSD of the structure-borne sound, which is caused by changes in the framework of smartphones, ForcePhone [14] is able to measure the pressure force of people's finger touch on the screen.

Beside of being used to estimate the power change of certain sound signals, PSD is also useful in detecting the frequency shifts. For instance, the change in the propagation of sounds will result in a frequency shift, which is also known as the Doppler Shift. In AAMouse [15], a smartphone is used to measure the PSD of sound signals sourced from two

[1] Stands for I can hear more!

speakers, and detect the Doppler Shift of received sounds to determine its relative movement with the speakers. Since the position of speakers is known to the smartphone, it can then easily determine its spatial motion. However, realizing the fact that Doppler Shift may not precisely reflect the motion of the smartphone, CAT [5] employs multiple FMCW (Frequency Modulated Continuous Wave) signals, detecting their frequency shift through PSD measurement. Since the frequency of the FMCW signal naturally changes through the time, CAT is able to precisely determine the propagation distance of the received signal.

B. Sensing Ambient Sounds

Instead of generating and sensing dedicated sound signals, there are also some projects measure the PSD of ambient sound signals, using it to derive fingerprints or discover and monitor certain patterns. For instance, EchoTag [16] utilizes smartphones to measure the PSD of environment sounds, and fingerprints on the PSD to distinguish between locations.

Nevertheless, sensing and fingerprinting the PSD of all ambient sounds is not always necessary. For example, Keystroke Snooping [17] focuses on measuring the PSD of sounds caused by keystroking, derives the time-difference-of-arrival (TDoA) information to distinguish different keys, then snoops on the key input. Similarly, DopEnc [18] profiles on the PSD of human voices to determine if two persons are encountering and interacting with each other. V-Sense [19] measures the PSD and monitors on a low-frequency range to detect the turning sound, thereby determines the steering status of the vehicle.

All of the previous efforts, as described above, can only measure the PSD of sounds at frequency no higher than 24 kHz, due to the limited sample rate of mobile devices. As mentioned in [3], this limit fairly restricts the performance of these efforts, such as the throughput of data transport.

In this paper, we focus on pushing this limit to a much higher level, thus the performance of such sound sensing apps can be instantly boosted. We for the first time realize coprime sampling on mobile devices to enable PSD measurement. Also different from BackDoor [20], which applies non-linearity to mask audible sounds with inaudible ultrasounds, we use this inherent feature to yield additional sample sequence for coprime sampling, so that no hardware extension is necessary.

III. iChemo: Foundations

The PSD measurement essentially requires a time-domain sample sequence of the sound signal. Suppose that for the target signal $x(t)$, a sample sequence $x(nT)$ of length N is acquired at a sample rate of f_s. Its PSD, denoted by $S_x(\omega)$, can be then calculated from $X(\omega)$, the corresponding Discrete Fourier Transform (DFT) of the sample sequence:

$$S_x(\omega) = \frac{|X(\omega)|^2}{N f_s}. \tag{1}$$

PSD of a certain signal is reliably measurable if its sample sequence is acquired by sampling at a rate over twice of its frequency. Yet, due to the limit of sample rate, current mobile devices are not able to correctly sample the ultrasound over 24 kHz. We refer to sampling at a rate below Nyquist threshold as *sub-Nyquist sampling*.

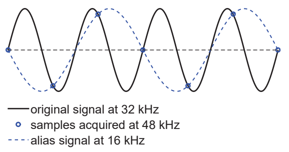


Fig. 3. An example of aliasing

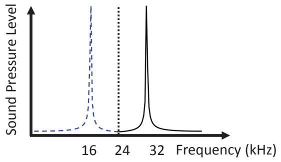


Fig. 4. 32 kHz signal aliases as a 16 kHz signal

A. Sub-Nyquist Sampling and Aliasing

Sub-Nyquist sampling will result in insufficient samples of the original signal. Such samples appear as they are acquired from another signal at a different frequency, which is usually lower than half of the Nyquist threshold. This phenomenon is called *aliasing*. Figure 3 illustrates a simple example that how aliasing goes. The 32 kHz signal is sampled at a rate of 48 kHz, resulting in a sample sequence of a 16 kHz signal, of which the phase is exactly shifted by π. Since the highest detectable frequency is 24 kHz, which is below 32 kHz, the original signal will appear as a 16 kHz signal that falls within the detectable range, as illustrated in Figure 4.

Suppose that a signal at frequency f_c is sampled at a rate of f_s. The frequency $f_(a)$ of the alias signal can be calculated by the Equation (2):

$$f_a = |nf_s - f_c|, n \in \mathbb{N}. \tag{2}$$

As such, even though we can not directly recover the original ultrasound signal, we can still acquire a portion of its samples, and further measure the PSD from its alias signal.

However, there is an essential drawback of this naive method: different ultrasound signals may result in the identical alias signal. For example, consider that two ultrasound signals, at frequencies of 40 kHz and 56 kHz respectively, are to be sampled at a rate of 48 kHz. In this case, both ultrasounds will appear as an alias signal of 8 kHz in the frequency domain, since $|48-40| = |48-56| = 8$. If the two signals are sampled simultaneously, there is no way to distinguish them from the resulting 8 kHz signal.

This drawback can be resolved if we can do the sampling at different rates. Specifically, if there are two sample sequences acquired at coprime rates, the two ultrasounds that appear as the identical alias signal in one sequence, will not result in the same alias signal in another sequence. We refer to this as *coprime sampling* [10].

B. Basics of Coprime Sampling

Coprime sampling was initially presented by Vaidyanathan et al [9] for efficient signal sampling and processing. The key idea is to sample the original signals using samplers that operate on mutually coprime time intervals. With coprime sampling the autocorrelation of sampled signals can be derived, therefore it is quite useful in signal analysis such as Direction of Arrival (DoA) estimation [21].

Formally, suppose the sampling interval is T, i.e. the sample rate is $(1/T)$ Hz, and the discrete samples of the original signal are denoted with $x(n) \equiv x(nT)$, $n = 1, 2, \ldots$. Sample sequences from two samplers, with sampling interval MT and NT, are denoted with $x_1(n_1)$ and $x_2(n_2)$ respectively, where M and N are mutually coprime [2]. As such, the relationship among them are as follows:

$$\begin{aligned} x_1(n_1) &= x(Mn_1), \\ x_2(n_2) &= x(Nn_2). \end{aligned} \tag{3}$$

By combining the two coprime sample sequences, the autocorrelation of the original signal can be derived, enabling further PSD measurement.

C. PSD measurement From Coprime Sample Sequences

The classic coprime sampling algorithm is readily usable for deriving the autocorrelation information $R_{xx}(k)$ of $x(t)$:

$$R_{xx}(k) = \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} x(n)x(n+k), \tag{4}$$

where $k \in \mathbb{Z}$ represents the lag for estimation. Since M and N are coprime integers, from Euclid's theorem, any integer k can be derived from some integer n_1 and n_2, so that

$$k = Mn_1 - Nn_2. \tag{5}$$

Note that any k within range $0 \leq k \leq MN - 1$ can be obtained by some pairs of n_1 and n_2 within range $0 \leq n_1 \leq 2N - 1$ and $0 \leq n_2 \leq M - 1$ respectively. Moreover, recall that $k = M(n_1 + Nl) - N(n_2 + Ml)$ for any integer l. As such, even though we can not directly obtain the autocorrelation $R_{xx}(k)$ from signal samples $x(n)$ that are obtained at sample rate as high as $\frac{1}{T}$ Hz, we can still estimate $R_{xx}(k)$ from two sample sequences $x_1(n_1)$ and $x_2(n_2)$ that are acquired at lower sample rates:

$$R_{xx}(k) = \frac{1}{L} \sum_{l=0}^{L-1} x_1(n_1 + Nl)x_2^*(n_2 + Ml), \tag{6}$$

where L can be arbitrary positive integer. As for the negative lags $-MN + 1 \leq k < 0$, the autocorrelation $R_{xx}(k)$ can be simply derived by taking $-n_1$ and $-n_2$. Furthermore, since $R_x x(k)$ and $R_x x(-k)$ are not necessarily equal for arbitrary signal sample $x(n)$, the final estimation of autocorrelation $\hat{R_{xx}}(k)$ can be taken as the average

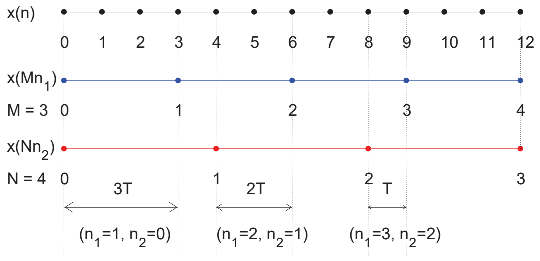


Fig. 5. An example of autocorrelation estimation. The value of autocorrelation on the lag of $3T$ can be derived from samples $x_1(1) = x(3)$ and $x_2(0) = x(0)$.

$$\hat{R_{xx}}(k) = \frac{R_{xx}(k) + R_{xx}(-k)}{2}, \tag{7}$$

so that $\hat{R_{xx}}(k) = \hat{R_{xx}}(-k)$. From this estimation of autocorrelation, the PSD of signal $x(t)$ can be derived. To this end, denoting $\hat{R_{xx}}(k)$ as $\hat{R}(k)$, a typical method is to form the correlation matrix as follows [22]:

$$\begin{bmatrix} \hat{R}(0) & \hat{R}(-1) & \ldots & \hat{R}(-MN+1) \\ \hat{R}(1) & \hat{R}(0) & \ldots & \hat{R}(-MN+2) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{R}(MN-1) & \hat{R}(MN-2) & \ldots & \hat{R}(0) \end{bmatrix}. \tag{8}$$

Since $\hat{R}(k) = \hat{R}(-k)$, the correlation matrix shown in (8) is basically a full-rank Toeplitz matrix.

Figure 5 shows an example of how coprime sampling estimates the autocorrelation of the signal samples $x(n)$ that are supposed to be obtained at sample rate $(1/T)$ Hz. The two coprime sample sequences are aligned to the same beginning sample $x(0)$. Therefore the two sequences also share common samples where the equation $Mn_1 = Nn_2$ is met, such as $x(12)$, $x(24)$, and so forth. But other than these common samples, there is no overlap between two sequences, as such these samples can be used to derive autocorrelation of the original signal sample $x(n)$. For example, the autocorrelation at lag T, i.e. $R_{xx}(1)$, can be calculated by setting $n_1 = 3$ and $n_2 = 2$, since $k = Mn_1 - Nn_2 = 3 \times 3 - 4 \times 2 = 1$. For the autocorrelation at negative lag $R_{xx}(-1)$, samples at $n_1 = 4 + (-3) = 1$ and $n_2 = 3 + (-2) = 1$ can be used. As such, the final estimation of autocorrelation at lag T is calculated by equation (7).

With the correlation matrix, the PSD can then be estimated by applying the MUSIC algorithm [23]. Formally, suppose that $F_k(\omega)$ be the Fourier Transform of the k-th column in the correlation matrix, where ω is the estimated frequency, then the PSD measurement $S_x(\omega)$ is derived as follows:

$$S_x(\omega) = \frac{1}{\sum_{k=p+1}^{N} |F_k(\omega)|^2}, \tag{9}$$

[2] In the following section these two sequences are referred to as **coprime sample sequences**, and two factors M and N are referred to as **coprime factors**.

where p is the double of the number of target signals, and N is the dimension of the eigenvector of the correlation matrix.

IV. Implementing iChemo on Mobile Devices

Coprime sampling essentially requires two sample sequences which are acquired at different sampling rates. To fulfill this requirement, there must be at least two ADCs operating simultaneously but at different sampling rates. Unfortunately, on most current mobile devices there is but one ADC, even if some of them have multiple microphones. For those devices that support USB On-The-Go (OTG) ADCs, the built-in ADC is disable when OTG ADCs are operating. Another challenge comes with the built-in *Anti-Aliasing Filter* (AAF). In order to suppress inaudible noises in the input of the microphone, mobile devices usually employ a AAF on the recorded analog sound signal, before it further goes to the ADC [20]. This issue further makes the implementation of iChemo no easy task.

To resolve these two challenges, we employ an inherent feature of the sound recording system on COTS mobile devices: the non-linearity feature.

A. Basics of the Non-linearity Feature

An ideal sound recording system is linear, that is, the recorded signal is proportional to its input signal. Assume that the input signal $S(t)$ and corresponding output is $\hat{S}(t)$. The ideal linear relationship between the input and output is:

$$\hat{S}(t) = aS(t), \tag{10}$$

where a is a measurable factor.

However, this linearity does not perfectly hold in the real built-in sound recording system due to the imperfect implementation, such as the capacity mismatch and finite operational amplifier DC gains [24]. The actual output $\hat{S}'(t)$ is not proportional to the input $S(t)$. Theoretically, $\hat{S}'(t)$ is in a form of an infinite power series:

$$\hat{S}'(t) = \sum_{i=0}^{\infty} a_i S(t)^i, \tag{11}$$

where $S(t)$ and $\hat{S}'(t)$ are the input and output signal respectively, and a_i is the factor of the ith order polynomial.

This non-linearity can be explicitly affecting the result when the sound recording system is heavily loaded, i.e. taking multiple signals as input. Suppose that the input $S(t)$ is composed with two components $s_1(t)$ and $s_2(t)$:

$$S(t) = s_1(t) + s_2(t), \tag{12}$$

where $s_1(t)$ and $s_2(t)$ are sines signals:

$$\begin{aligned} s_1(t) &= A_1 sin(2\pi f_1 t + \phi_1), \\ s_2(t) &= A_2 sin(2\pi f_2 t + \phi_2). \end{aligned} \tag{13}$$

Suppose that the highest non-linearity order is 2. According to the equation (11), before processed by any low-pass filter, the output $\hat{S}'(t)$ will be as follows:

$$\begin{aligned} \hat{S}'(t) &= a_1 S(t) + a_2 S(t)^2 \\ &= a_1 \big[A_1 sin(2\pi f_1 t + \phi_1) + A_2 sin(2\pi f_2 t + \phi_2)\big] \\ &\quad + \frac{a_2}{2}\big[A_1^2 + A_2^2 - A_1^2 cos(4\pi f_1 t + 2\phi_1) \\ &\qquad - A_2^2 cos(4\pi f_2 t + 2\phi_2)\big] \\ &\quad - a_2 A_1 A_2 cos\big[2\pi(f_1 + f_2)t + (\phi_1 + \phi_2)\big] \\ &\quad + a_2 A_1 A_2 cos\big[2\pi(f_1 - f_2)t + (\phi_1 - \phi_2)\big] \end{aligned} \tag{14}$$

By applying the AAF, which generally suppresses signals at frequency over half of the sample rate f_s, the high-frequency components will be removed from the output. Assume that $f_s/4 < f_1 < f_s/2$ and $f_s/4 < f_2 < f_s/2$, after processed by the AAF, the actual output becomes as follows:

$$\begin{aligned} \hat{S}'(t) &= a_1 S(t) + a_2 S_p(t), \\ S_p(t) &= A_1 A_2 cos\big[2\pi(f_1 - f_2)t + (\phi_1 - \phi_2)\big]. \end{aligned} \tag{15}$$

Apparently the output is not proportional to the input. The two components $s_1(t)$ and $s_2(t)$ "create" a signal $S_p(t)$ at frequency $|f_1 - f_2|$. We refer to $S_p(t)$ the *phantom signal*, and we call $s_2(t)$ the *mask signal* of $s_1(t)$.

B. Coprime Sampling With Non-linearity

The aforementioned phantom signal $S_p(t)$ does not actually exist. Yet it appears like an alias signal in the record, where the original signal is at frequency f_1 and the sample rate is f_2. Making use of this fact, it becomes possible to acquire two coprime sample sequences by sampling for once.

Taking the two sine signals in equation (13) as an example. Suppose that the sample rate is f_s, and the input consists of only $s_1(t)$. Then, we intentionally generate the mask signal $s_2(t)$, and we choose f_2 to be coprime with f_s. As such, the input is exactly identical to the one in equation (12).

According to the equation (15), the output $\hat{S}'(t)$ will be a weighted sum of two components $S(t)$ and $S_p(t)$. Since $s_2(t)$ is a known signal intentionally generated, it can be easily removed from the output by conducting a subtraction. In addition, since $s_1(t)$ is a high-frequency ultrasound signal, its frequency f_1 is expected to be higher than $f_s/2$, meaning $S(t)$ will cause an alias signal $S'(t)$. Suppose that $|f_1 - f_2| < f_s/2$, the two components $S'(t)$ and $S_p(t)$ will become as follows:

$$\begin{aligned} S'(t) &= A_1 sin[2\pi(f_s - f_1)t + \phi_1 + \Phi], \\ S_p(t) &= A_1 A_2 sin\big[2\pi(f_1 - f_2)t + (\phi_1 - \phi_2) - \pi/2\big], \end{aligned}$$

where Φ equals to either π or 0, depending on the relationship between f_1 and f_s. Basically, $\Phi = 0$ if f_1 is $[N, N{+}1/2]$ times of f_s, where N is a non-negative integer. Similarly, $\Phi = \pi$ if f_1 is $[N + 1/2, N + 1]$ times of f_s.

As such, we can get two signals from the output: one is an alias signal $S'(t)$, and the another is a phantom signal

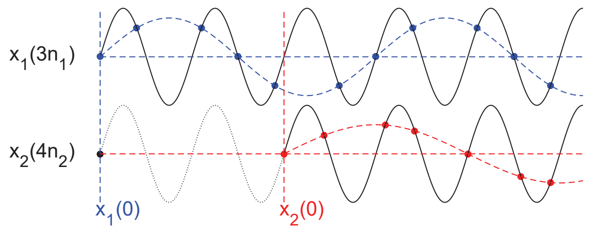


Fig. 6. An example of two sample sequences where there is a latency of 2 periods of the target sound signal.

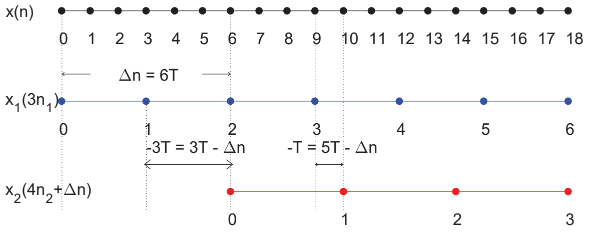


Fig. 7. A scenario where there is a lag of 6 unit intervals between two coprime sample sequences.

$S_p(t)$. However, $S_p(t)$ can not be simply taken as another alias signal. The reason is that, the alias signal is composed by the samples of the original signal, meaning that they are naturally aligned in the time domain. However, the phantom signal is basically the subtraction of the original signal and the mask signal. That is, the samples composing the phantom signal are only at the positions where the samples of the mask signal are acquired. Since the mask signal may not naturally align with the original signal, there is an unknown random latency between their time-domain waveform. To use both the phantom signal and the mask signal as paired alias signals, a decent sample sequence alignment method is a must.

C. Approaching Ideal Sample Sequence Alignment

The original signal and the mask signal do not naturally align with each other, causing the latency between the alias signal and the phantom signal. Yet, an insight here is that, this latency can be reduced on two conditions: (1) the latency is a multiple of the period of target ultrasound, or (2) the latency is a multiple of the target sampling interval. The soundness of condition (1) is obvious. Since the ultrasound signal transits on the same pattern during every period, it is obvious that the delayed sampler will witness the same beginning sample (although technically speaking the two samples are within different periods) as the one without being delayed. Figure 6 shows an example, where the second sample sequences $x_2(4n_2)$ starts two period of target signal after the first sample sequences $x_1(3n_1)$. Since the latency is integer times of the period of target signal, both sample sequences starts at the same interval, meaning they shall witness the same time-domain waveform of the signal.

The soundness of condition (2) comes as follows. Suppose that the alias signal is $x_1(n_1)$ and the phantom signal is $x_2(n_2)$. The target sample rate is f_s, so the target unit sampling interval is $T = 1/f_s$. The coprime factors of $x_1(n_1)$ and $x_2(n_2)$ are M and N, meaning that their sampling intervals are MT and NT respectively. Now suppose that a latency of ΔnT is between the two signals. That is, the phantom signal starts at interval ΔnT instead of interval 0. Since this latency is Δn times of the target unit sampling interval T, when calculating the lag k using equation (5), the latency just introduces a constant shift of $-\Delta n$ to the value of k:

$$k' = Mn_1 - (Nn_2 + \Delta n) = Mn_1 - Nn_2 - \Delta n. \qquad (16)$$

Figure 7 shows an example of this situation. The unit sampling interval is T, coprime factors are $M = 3$ and $N = 4$, and the latency introduced to the second sampling process is $\Delta n = 6T$. When $n_1 = 1$ and $n_2 = 0$, the value of k will become $3 - 6 = -3$, meaning that the autocorrelation $R_{xx}(3)$ becomes $R_{xx}(-3)$. Similarly, autocorrelations $R_{xx}(k)$ on every other feasible lags k will become $R_{xx}(k-6)$.

Therefore, combining both conditions (1) and (2), a latency of arbitrary length can be approximated to just a couple times of the period of the target sampling interval. Formally, suppose that the frequency of the original signal is f_c, and the target sample rate is f_s. By Nyquist-Shannon Sampling theorem, f_c shall be no longer than half of f_s, meaning that the period of the target signal $1/f_c$ is at least twice longer than $1/f_s$. Let ΔT be the latency between the alias and phantom signal, it can be approximated as follows:

$$\Delta T \approx \frac{a}{f_c} + \frac{b}{f_s}. \qquad (17)$$

where a is a non-negative integer and b is an arbitrary real number. According to condition (1), the portion of a/f_c is ignorable, as such b shall be within range $[0, \frac{f_s}{f_c}]$. For instance, suppose that the target sample rate is 96 kHz and the frequency of target ultrasound signal is 20 kHz, then b is within range $[0, 4.8]$. This means that latency of arbitrary length ΔT can be approximately eliminated by letting Δn vary from 0 to 5.

Note that this method may not totally remove the latency, as any latency of length less than $1/f_s$ is not erasable. However, suppose that the maximum latency is ρ, the impact of such kind of latency is theoretically bounded as the proportion of $(\frac{1}{f_c\rho})^2$, which is controllable [25].

D. Resolving Built-In AAF

As aforementioned, AAF is commonly employed in the built-in sound recording system on mobile devices. The AAF for analog signals is a physical circuit placed between the microphone and the ADC. This AAF, unfortunately, can neither be disabled or bypassed. Yet, an insight here is that,

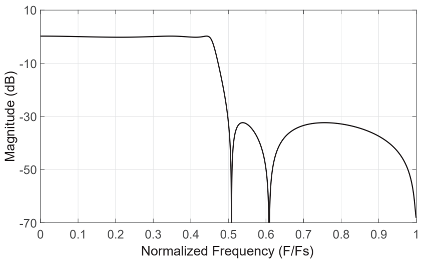


Fig. 8. Frequency response of a typical analog AAF

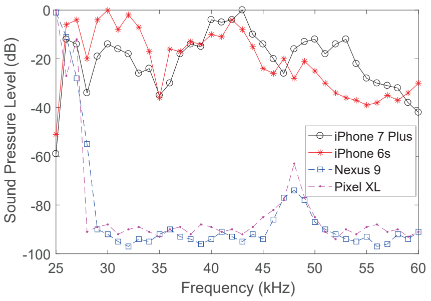


Fig. 9. Frequency responses from 24 to 60 kHz

even if the AAF for analog signals is inevitable, its impact to the ultrasound sensitivity is still controllable. Even though the AAF is designed to only allow the sound below *cutoff frequency* to pass, in reality they can not completely remove all high-frequency ultrasounds from the record. In fact, there are always a series of peaks in the frequency response, outside the passband of the AAF. Figure 8 shows the frequency response of a typical analog AAF. As is shown by the curve, although the cutoff frequency is $0.5F_s$, the signals at frequency around $0.54F_s$ and $0.75F_s$ are still not fully removed (yet the power is reduced by about 33 dB).

If we can get the exact frequency response of the built-in AAF on mobile devices, we can expect to get a strong enough read of high-frequency ultrasound. Unfortunately, due to that the native sample rate is limited to 48 kHz, it is not possible to directly measure the frequency response over 24 kHz, the near-ultrasound threshold. Therefore, we can only take the power read of corresponding alias signal as the power response to the high-frequency ultrasound.

Figure 9 illustrates the frequency response of some popular mobile devices we have measured, including Apple iPhone 6s, iPhone 7 plus, Google Nexus 9 and Pixel XL, within the ultrasound range of 24-60 kHz. Through this extensive tests, we can still observe strong enough alias signals (at power over -40 dB) on the two Apple smartphones, when feeding them with the ultrasound at frequency over 28 kHz. This means that we can directly get the necessary alias signal from these two models. However, for the two Android models, they are not able to get strong enough alias signal when the ultrasound is over 29 kHz. Therefore, we need to use the non-linearity feature to strengthen their read, i.e. apply additional speakers to feed them with a mask signal at 48 kHz.

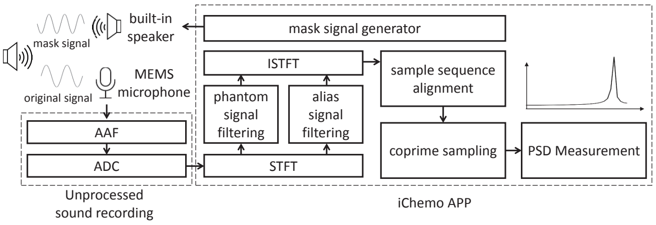


Fig. 10. Overview of iChemo prototype on Android

V. Prototype and Evaluations

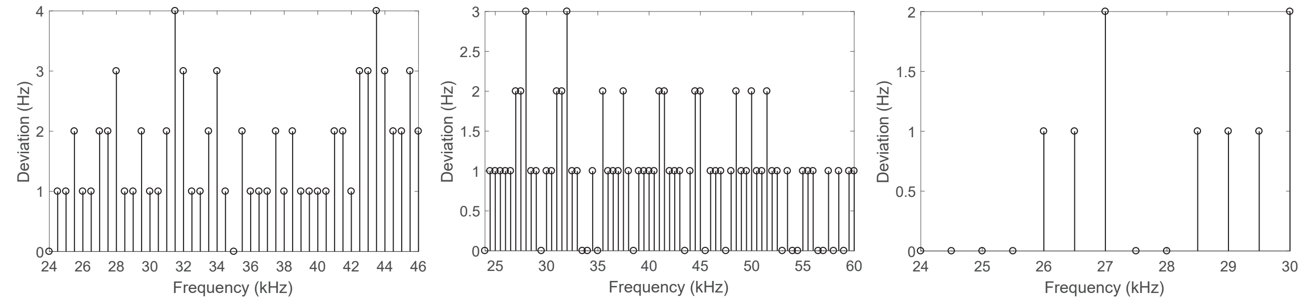
In this section we introduce the evaluations on the performance of iChemo on both Android and iOS platforms. On Android we evaluate the prototype of iChemo, which is an app deployed on the Google Nexus 9 tablet and Pixel XL smartphone. As for the iOS platform, we collect sound records from iPhone 6s and iPhone 7 plus smartphones, and then process the sound records on a separate desktop for evaluation.

A. Prototype Overview

Although it is mostly ideal to prototype iChemo on both platforms, the inherent sound signal processing on iOS platform can not be disabled. On the other hand, Android supports raw sound sampling since its version 7.0. Therefore, we implement the iChemo prototype as a user-layer app on Android 7.1. Figure 10 illustrates an overview of iChemo prototype.

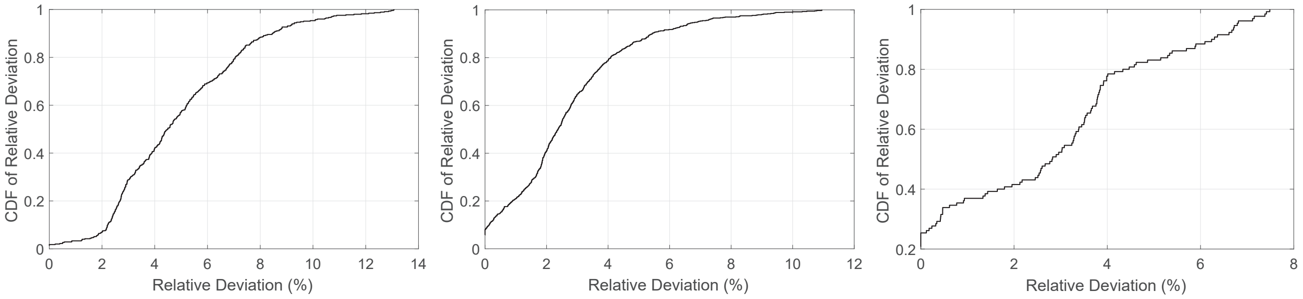
Firstly, iChemo takes the raw sample of sounds from the built-in microphone, at a native sample rate of 48 kHz. iChemo applies the unprocessed sound sampling procedure through Android AudioTrack, which allows the sound samples to bypass the pre-processing in the Hardware Abstraction Layer (HAL). Then iChemo derives the spectrum of the raw samples through the Short-Time Fourier Transform (STFT), in a time interval of 10 ms. From the spectrum, iChemo extracts the phantom signal and alias signal component by frequency-domain filtering, and restore the signal through a typical Inverse-STFT (ISTFT) process. After both signals acquired, iChemo conducts sample sequence alignment described in Section IV.C, and then derive the autocorrelation through the coprime sampling algorithm. Based on the autocorrelation information, iChemo applies the MUSIC algorithm to yield the PSD measurement, as is described in Section III.B.

Since both the phantom signal and the alias signal are to be obtained from the same spectrum, a subtle frequency-domain filtering strategy is necessary. For instance, if the mask signal is at 32 kHz, and the original signal is at a frequency within the range of 24-40 kHz, then the frequency of alias signal will vary from 24 down to 8 kHz. This is because the sample rate is 48 kHz, and according to equation (2), it is equally the sum of the frequencies of the original and alias signal. Similarly, the frequency of the phantom signal will only be in a range of 0-8 kHz. Therefore, to filter out the phantom signal, we only



(a) With mask signal at 32 kHz (b) With mask signal at 40 kHz (c) With mask signal at 20 kHz

Fig. 11. Absolute Frequency Deviation in PSD measurement



(a) With mask signal at 32 kHz (b) With mask signal at 40 kHz (c) With mask signal at 20 kHz

Fig. 12. Relative Frequency Deviation in PSD measurement

take the spectrum between 0 Hz and 8 kHz, while for the alias signal, we use the spectrum from 8 kHz to 24 kHz.

Note that most current mobile devices can emit a near-ultrasound signal at 20 kHz. This means that iChemo can individually work without any external mask signal being emitted, yet the highest measurable frequency is then limited.

B. Absolute Frequency Deviation

Sound sensing apps that take advantage of Doppler Shifts are sensitive to the absolute frequency deviation from the truth. Therefore, we evaluate iChemo on this metric under three main setups: (1) an external mask signal at 32 kHz is present, (2) an external mask signal at 40 kHz is present, and (3) a mask signal at 20 kHz, emitted by iChemo, is present.

With the native sample rate being 48 kHz, the first setup forms a coprime factor pair where $M = 3$ and $N = 2$, enabling iChemo to measure the ultrasounds at a frequency up to 48 kHz. We use this setup for signals at frequencies vary from 24 to 46 kHz. In the second setup, the chosen mask signal frequency forms a coprime factor pair where $M = 6$ and $N = 5$. With this setup, the highest measurable frequency is 120 kHz in theory. Unfortunately, the frequency of phantom signal must not pass half of the frequency of mask signal; otherwise the phantom signal will fall into the band reserved for alias signal, which will interfere the outcome. As such, we only use this setup to measure the PSD at frequencies up to 60 kHz. Similarly, in the third setup, even the coprime factor pair is $M = 12$ and $N = 5$, only ultrasounds at frequencies no higher than 30 kHz are measurable. Figure 11 presents the evaluation results, where the maximal absolute frequency deviation is 4 Hz under all the three setups.

C. Relative Frequency Deviation

For sound sensing apps that are built on ASK, a frequency deviation in the PSD measurement is ignorable if it is within a certain threshold. This threshold is mostly related to the target spectrum bandwidth, which is determined by the frequency of target signals.

To evaluate the performance of iChemo in such apps, we assess a metric called *relative deviation*, which is the ratio of the deviation to the target signal frequency. Figure 12 illustrates the Cumulative Distribution Function (CDF) of the evaluated relative deviations, under the same experiment setups as above. Among all the tests, the maximum relative deviation is 13.07%, and the mean relative deviation is 3.37%. This means the overall accuracy of PSD measurement is over 86.93%, and the mean accuracy is 96.63%.

VI. Discussions

A. The Limit of AAF

As aforementioned, most current mobile devices employ an analog AAF to remove ultrasound from the sound records. The cutoff frequency of the AAF varies among different devices, but most of them suppress any sound at frequency higher than 24 kHz. Since this analog AAF can not be bypassed, it becomes the major challenge of putting iChemo into practice.

In this paper, we present a resolution to this issue, i.e. producing a phantom signal of the target one by introducing a proper mask signal. This method basically takes advantage of the inherent non-linearity feature on mobile devices. It however has two limitations: (1) this feature may not be explicit in high-end ADC chips, and (2) there is the risk of overlapping the phantom signal and the alias signal.

B. PSD measurement on Multiple Concurrent Sound Signals

iChemo requires that the target ultrasound signal is sparse, meaning it occupies a very limited bandwidth around its center frequency. This is the most basic case in sound sensing apps that are built on ASK or Doppler Shift measurements. However, iChemo still supports measuring the PSD of multiple ultrasound signals, if these signals are all sparse and there are adequate gaps among the bands they occupy. Recall that in equation (9), the parameter p is the double of the number of target sound signals in the PSD measurement. That is, if this number is known to iChemo, it can still measure the PSD of such sound signals by choosing a proper value of p.

C. Extension for Phase Measurement

PSD measurement basically reveals the frequency-domain feature of ultrasound signals. Yet sometimes the time-domain feature of signals is also necessary, and the need for the phase measurement comes. The functionality of iChemo is potentially able to be extended to support the phase measurement. Specifically, given the phases of both the mask signal and the phantom signal, the phase of the target signal can be derived by summing them up, as is shown in the equation (15). As a preliminary result, for ultrasounds at frequencies from 28 to 42 kHz, we find that iChemo is able to measure their phase with a mean accuracy of 97%.

VII. Conclusion

In this paper, we introduced iChemo, a technology that can enable the ability of COTS mobile devices to sense high-frequency ultrasounds that is beyond their capacity to fully capture. Particularly, we introduced how the non-linearity feature can be used to realize coprime sampling algorithm on mobile devices, with only one available built-in ADC. Also, we tackle the two realistic challenges against the implementation of coprime sampling on COTS mobile devices, including the sample sequence alignment and resolution of AAFs. By evaluating iChemo on both Android and iOS devices, we showed that it can sense the ultrasound at 60 kHz, which is over twice of the current upper limit of 24 kHz. Our future work includes extending iChemo to achieve phase shift measurement on high-frequency ultrasounds.

VIII. Acknowledgment

This work was supported by a Canada Technology Demonstration Program (TDP) grant, a Canada NSERC Discovery Grant, and an NSERC E.W.R. Steacie Memorial Fellowship.

Reference

[1] Chunyi Peng, Guobin Shen, Yongguang Zhang, Yanlin Li, and Kun Tan. Beepbeep: a high accuracy acoustic ranging system using cots mobile devices. In *ACM SenSys*, 2007.

[2] Rajalakshmi Nandakumar, Vikram Iyer, Desney Tan, and Shyamnath Gollakota. Fingerio: Using active sonar for fine-grained finger tracking. In *ACM CHI*, 2016.

[3] Qian Wang, Kui Ren, Man Zhou, Tao Lei, Dimitrios Koutsonikolas, and Lu Su. Messages behind the sound: real-time hidden acoustic signal capture with smartphones. In *ACM MobiCom*, 2016.

[4] Google Inc. Google tone. https://g.co/tone, 2017.

[5] Wenguang Mao, Jian He, and Lili Qiu. Cat: high-precision acoustic motion tracking. In *ACM MobiCom*, 2016.

[6] EB Steinberg. Ultrasonics in industry. *Proceedings of the IEEE*, 53(10):1292–1304, 1965.

[7] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE*, 5(1):3–55, 2001.

[8] G Enrico Santagati and Tommaso Melodia. U-wear: Software-defined ultrasonic networking for wearable devices. In *ACM MobiSys*, 2015.

[9] Palghat P Vaidyanathan and Piya Pal. Sparse sensing with co-prime samplers and arrays. *IEEE Transactions on Signal Processing*, 59(2):573–586, 2011.

[10] Haitham Hassanieh, Lixin Shi, Omid Abari, Ezzeldin Hamed, and Dina Katabi. Ghz-wide sensing and decoding using the sparse fourier transform. In *IEEE INFOCOM*, 2014.

[11] Yuchi Chen, Wei Gong, Jiangchuan Liu, and Yong Cui. Fine-grained ultrasound range finding for mobile devices: Sensing way beyond the 24 khz limit of built-in microphones. In *IEEE INFOCOM SmartCity*, 2017.

[12] Hyewon Lee, Tae Hyun Kim, Jun Won Choi, and Sunghyun Choi. Chirp signal-based aerial acoustic communication for smart devices. In *IEEE INFOCOM*, 2015.

[13] Soonwon Ka, Tae Hyun Kim, Jae Yeol Ha, Sun Hong Lim, Su Cheol Shin, Jun Won Choi, Chulyoung Kwak, and Sunghyun Choi. Near-ultrasound communication for tv's 2nd screen services. In *ACM MobiCom*, 2016.

[14] Yu-Chih Tung and Kang G Shin. Expansion of human-phone interface by sensing structure-borne sound propagation. In *ACM MobiSys*, 2016.

[15] Sangki Yun, Yi-Chao Chen, and Lili Qiu. Turning a mobile device into a mouse in the air. In *ACM MobiSys*, 2015.

[16] Yu-Chih Tung and Kang G Shin. Echotag: accurate infrastructure-free indoor location tagging with smartphones. In *ACM MobiCom*, 2015.

[17] Jian Liu, Yan Wang, Gorkem Kar, Yingying Chen, Jie Yang, and Marco Gruteser. Snooping keystrokes with mm-level audio ranging on a single phone. In *ACM MobiCom*, 2015.

[18] Huanle Zhang, Wan Du, Pengfei Zhou, Mo Li, and Prasant Mohapatra. Dopenc: acoustic-based encounter profiling using smartphones. In *ACM MobiCom*, 2016.

[19] Dongyao Chen, Kyong-Tak Cho, Sihui Han, Zhizhuo Jin, and Kang G Shin. Invisible sensing of vehicle steering with smartphones. In *ACM MobiSys*, 2015.

[20] Nirupam Roy, Haitham Hassanieh, and Romit Roy Choudhury. Backdoor: Making microphones hear inaudible sounds. In *ACM MobiSys*, 2017.

[21] Siavash Shakeri, Dyonisius Dony Ariananda, and Geert Leus. Direction of arrival estimation using sparse ruler array design. In *IEEE SPAWC*, 2012.

[22] Si Qin, Yimin D Zhang, and Moeness G Amin. High-resolution frequency estimation using generalized coprime sampling. In *SPIE Sensing Technology+ Applications*, pages 94970K–94970K. International Society for Optics and Photonics, 2015.

[23] Ralph Schmidt. Multiple emitter location and signal parameter estimation. *IEEE Transactions on antennas and propagation*, 34(3):276–280, 1986.

[24] Rongzong Kang, Pengwu Tian, and Hongyi Yu. Effects of adc nonlinearity on the spurious dynamic range performance of compressed sensing. *The Scientific World Journal*, 2014, 2014.

[25] Ali Koochakzadeh and Piya Pal. On the robustness of co-prime sampling. In *Signal Processing Conference (EUSIPCO), 2015 23rd European*, pages 2825–2829. IEEE, 2015.