© 2019 Sheng Shen

ACTIVELY EXPLOITING PROPAGATION DELAY FOR ACOUSTIC SYSTEMS

BY

SHENG SHEN

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2019

Urbana, Illinois

Doctoral Committee:

Professor Romit Roy Choudhury, Chair
Professor Klara Nahrstedt
Professor Rayadurgam Srikant
Assistant Professor Haitham Al-Hassanieh

ABSTRACT

Propagation delay refers to the length of time it takes for a signal to travel from point A to point B. Many existing systems, including Global Positioning System (GPS) localization, vehicular imaging, and microphone array beamforming, have taken advantage of propagation delay. This dissertation revisits different properties of propagation delay to enable new acoustic techniques and applications. For instance: (1) We leverage the propagation delay difference between two very different frequencies – radio frequency (RF), and acoustics – to improve active noise cancellation. By "piggybacking" sound over RF, our proposed system is able to compute anti-noise signals more precisely, and ultimately attain better cancellation performance. (2) We develop solutions that exploit the propagation delays of multipath echoes to localize an indoor human speaker. By aligning the arrivals of the voice signal at different times, we compute user location within an optimization framework, serving as a valuable context for smart voice assistants like Amazon Echo and Google Home. (3) We design 3D directional sound by actively synthesizing different propagation delays at two ears using earphones. We develop algorithms that accurately track the 3D orientation of the head, a key enabler for designing 3D acoustics. In general, this dissertation shows that while propagation delay has been studied for a long time and for many applications, there is still opportunity for new techniques and systems, by carefully looking at different properties of the propagation delay, across frequencies, time, and space.

To my parents, and Shuanghui, for their love and support.

ACKNOWLEDGMENTS

I am deeply indebted to many:

To my advisor, Professor Romit Roy Choudhury, for his continuous guidance and support. I will never forget the many afternoons he spent with me in the coffee shop brainstorming new research ideas, or the late nights when he drove me home as we were working for paper deadlines. I have been, and will always be, learning from him – his research taste, presentation skills, writing style, and even life philosophy. I could not have imagined having a better advisor and mentor throughout my Ph.D. years. Deciding to join his lab six years ago was the best decision I have ever made.

To Professor Haitham Hassanieh, with whom I have been working closely since 2016. I have always been inspired and motivated by him through countless discussions.

To the University of Illinois at Urbana-Champaign, the Department of Electrical and Computer Engineering, and the Coordinated Science Lab, for all the support during my Ph.D. study. I especially thank Professor Klara Nahrstedt and Professor Rayadurgam Srikant for serving on my doctoral committee, and for their insightful comments. Klara has always amazed me with her breadth of knowledge, and Srikant is the person to go to whenever I have theoretical questions.

To Rufeng Meng and Professor Srihari Nelakuditi at the University of South Carolina, for their collaboration with me in research projects. They

both have helped and encouraged me a lot, especially at the beginning stage of my Ph.D. program.

To my past and present labmates in the SyNRG group. I especially thank He Wang, Mahanth Gowda, Nirupam Roy, Ashutosh Dhekne, Daguan Chen, Junfeng Guan, Zhijian Yang, and Yu-Lin Wei for their collaboration. Thanks, Puneet Jain, for your kind help in my job hunting.

To my parents, and my wife, Shuanghui, for their endless love, belief, and support.

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION 1
1.1 Proposed Techniques and Applications 3
1.2 Organization 6

CHAPTER 2 MUTE: BRINGING IOT TO ACOUSTIC NOISE CANCELLATION 7
2.1 Introduction 7
2.2 Noise Cancellation Primer 13
2.3 Lookahead Aware ANC 16
2.4 *MUTE*: System and Architecture 27
2.5 Evaluation 34
2.6 Current Limitations 44
2.7 Related Work 45
2.8 Conclusion 47

CHAPTER 3 VOLOC: SOURCE LOCALIZATION FROM VOICE SIGNALS 48
3.1 Introduction 48
3.2 Background and Formulation 54
3.3 System Architecture 58
3.4 Implementation, Evaluation 72
3.5 Limitations and Discussion 82
3.6 Related Work 83
3.7 Conclusion 85

CHAPTER 4 MUSE: IMU ORIENTATION TRACKING FOR 3D DIRECTIONAL SOUND 86
4.1 Introduction 86
4.2 Foundations of Tracking 90
4.3 Orientation Estimation 96
4.4 Orientation Evaluation 103
4.5 Points of Discussion 111
4.6 Related Work 112
4.7 Conclusion 114

CHAPTER 5 CONCLUSION 115

REFERENCES 116

CHAPTER 1

INTRODUCTION

Propagation delay is a well-known concept in physics, and refers to the amount of time it takes for a signal to travel from the sender to the receiver. Numerous acoustic and wireless applications have taken advantage of propagation delays to compute distances, perform beamforming, or image the environment [1, 2, 3, 4, 5, 6]. For instance, in GPS localization [7], the receiver measures the propagation delays from different GPS satellites, translates them into distances, and finally computes its own location via trilateration. In another example, microphone arrays in meeting rooms can beamform towards a specific human speaker [8]. This is essentially because voice signals travel in the air, and arrive at each microphone on the array with different propagation delays. After compensating for these relative delays, the energy can add up across multiple microphones, thereby enhancing the recorded SNR for that speaker. Finally, today's self-driving cars use different sensing modalities to transmit signals and receive their reflections. By measuring the round-trip propagation delay, they are able to image the surrounding environment, and compute the distances to nearby objects [9].

Recently, a variety of new acoustic technologies are surfacing, all centered around sensing and analyzing voice and sound. For example, voice assistants such as Amazon Echo and Google Home have begun to percolate our homes, and more than 100 million such devices have been sold [10]. Qualcomm is developing a completely new audio platform for earphones, aimed at combining music streaming, natural language processing, and wireless sharing [11].

A related startup named Nura has developed headsets that shine acoustic signals into the ear to profile its anatomical structure, so that sound can be carefully customized to the user [12]. This excitement around the acoustic landscape keeps growing, primarily due to the convergence of embedded sensing, computing, and communication.

This dissertation lies at the intersection of propagation delay and acoustics. We believe revisiting propagation delay will bring fresh opportunities to this exciting acoustic landscape. In particular, this dissertation exploits propagation delay along different dimensions to develop key acoustic primitives, that can be further leveraged to enable and improve human-centric applications. For instance:

- **Propagation delay across frequencies**: We leverage the propagation delay difference between radio frequency (RF) and acoustics to improve active noise cancellation. By having RF carry acoustic information, our proposed system is able to compute anti-noise signals more precisely, and ultimately achieve better cancellation performance.

- **Propagation delay across time**: We identify unique opportunities at the beginning of one's speech to localize the human speaker. The key idea is to decouple and exploit different propagation delays of multipath echoes at this beginning moment for reverse triangulation. This gives user location as an important context for smart voice assistants such as Amazon Alexa and Google Home.

- **Propagation delay across space**: We design 3D directional sound by carefully synthesizing the propagation delays of the acoustic signal played across the two earphones. By accurately tracking the device's spatial orientation, the sound can be designed to come from a specific 3D direction, enabling applications like acoustic augmented reality.

Exploiting propagation delay in these scenarios, however, entails unique challenges in both hardware and algorithm design. For example, cross-frequency propagation calls for hardware design with multiple sensing modalities and minimum processing delay; decoupling multipath propagation necessitates novel algorithms to extract each echo from the mixture in microphone recordings; and finally, creating artificial propagation delay requires accurate tracking of the device's rotational motion, an especially challenging task if the device constantly moves. This dissertation tackles these practical challenges by combining digital signal processing, optimization, and hardware engineering.

In general, this dissertation shows that while propagation delays have been looked at extensively in the past, there is still opportunity for new techniques and applications, by revisiting different physical properties of propagation delay, in both general as well as application-specific settings. We briefly elaborate on each of the acoustic techniques and systems, followed by the organization of this dissertation.

1.1 Proposed Techniques and Applications

1.1.1 Acoustic Noise Cancellation

Noise pollution is rising rapidly and becoming a health concern. Long term exposure to high noise levels can affect blood pressure and hypertension, leading to cardiovascular diseases; children's developmental cognition can also be affected [13, 14]. Noise from traffic, loud conversations in offices, continuous announcements in airports, large factories, and music playing in public places – all contribute to such trends. In developing regions, the problem is probably most pronounced because windows remain open due to limited air-conditioning. Hence loud music and chants from public speakers,

car honks, road construction, and just the urban cacophony can prevent a quiet lifestyle [15]. The accepted solution today has been to wear ear-plugs or ear-blocking headphones, both of which are uncomfortable for continuous use [16, 17, 18].

In Chapter 2, we propose *MUTE* [19], a system which considers breaking away from convention and aims to cancel complex sounds, including speech and music, without blocking the ear. The key idea is to place an IoT device in the environment that listens to ambient sounds and forwards the sound over its wireless radio to the ear-device. Since wireless signals travel much faster than sound, the ear-device can receive the sound in advance of its actual arrival. This *lookahead* is valuable for noise cancellation, since it (1) offers a much-needed time cushion for computation, (2) provides information for prediction, and (3) enables non-causal filtering for producing more precise anti-noise signals. Using custom-built IoT hardware, as well as lookahead-aware cancellation algorithms, we demonstrate a fully functional prototype that performs favorably against Bose's latest $QC35$ headphones. Importantly, our design does not need to block the ear like headphones or ear-buds. They can be fitted around the ear, or clipped to eyeglasses, making them comfortable (and healthier) for continuous use.

1.1.2 User Localization for Voice Assistants

Voice assistants such as Amazon Echo and Google Home continue to gain popularity with new "skills" getting continuously added to them. Towards enriching multiple dimensions of context-awareness, companies like Amazon, Google, and Samsung are also pursuing the problem of user localization [20, 21, 22, 23]. Location adds valuable context to the user's commands, allowing Alexa to resolve ambiguities. For instance, knowing the user's location could help in determining which light the user is referring to, when she says "turn

on the light" (naming every IoT device, and precisely remembering their names, is quickly becoming a memory overload for the users [24, 25]). More broadly, location could aid speech recognition by narrowing down the set of possible commands [26, 27, 28].

In Chapter 3, we develop *VoLoc*, a system which infers user location from voice signals received over a microphone array on Alexa. *VoLoc* accurately estimates the angles of arrival (AoAs) of multipath echoes, and traces back these AoAs to reverse triangulate the user's location. The key idea is to leverage the very beginning moment of one's speech when multipath echoes are sparse, and design an *iterative align-and-cancel* algorithm to compute each echo's AoA. We also estimate the geometry of a nearby wall reflection using an error-minimization technique. Together, the AoAs and geometric parameters are fused to achieve robust and useable localization accuracy.

1.1.3 Orientation Tracking for 3D Directional Sound

Human brains are capable of resolving the 3D direction of a sound, by sensing the slight difference in the propagation delay at two ears. We aim at actively synthesizing 3D directional sound by injecting artificial delays at two earphones. As one application, imagine Alice running to catch a train in a large train station. Her earphone can navigate her using a 3D voice that says "follow me". The voice is carefully designed and played across the two earphones (with synthesized propagation delays), so that it appears to come from the direction in which she should walk. Alice simply follows the *perceived direction* of the voice and reaches the platform in time; she does not pull out her phone, nor does she check for maps.

To generate the correct 3D directional sound, it is important to track the head's (earphone's) orientation over time accurately. In Chapter 4, we propose *MUSE* [29], an improved inertial measurement unit (IMU) orientation

tracking algorithm running on mobile devices, including smartphones, smartwatches, and earphones. The core observation is that conventional systems have trusted gravity more than the magnetic North to infer the 3D orientation of the object. We find that the reverse is more effective, as magnetometers, unlike accelerometers, are unpolluted by object motion. Real experiments across a range of uncontrolled scenarios show consistent improvement in orientation accuracy, without requiring any training or machine learning.

1.2 Organization

The rest of the dissertation expands on each of the applications in detail. We discuss acoustic noise cancellation in Chapter 2; user localization for voice assistants in Chapter 3; and orientation tracking for 3D directional sound in Chapter 4. Finally, we summarize the contributions and conclude the dissertation in Chapter 5.

CHAPTER 2

MUTE: BRINGING IOT TO ACOUSTIC NOISE CANCELLATION

Active noise cancellation (ANC) is the cancellation of noise in the environment by producing *anti-noise* signals near the human ears (e.g., in Bose's noise cancellation headphones). This chapter brings IoT to active noise cancellation by combining wireless communication with acoustics. The core idea is to place an IoT device in the environment that listens to ambient sounds and forwards the sound waveform over its wireless radio. Since wireless signals travel much faster than sound, our ear-device receives the signal before the actual sound arrives. This time difference serves as a glimpse into the future, which we call *lookahead*, and proves crucial for real-time noise cancellation, especially for unpredictable, wide-band sounds like music and speech. Using custom IoT hardware, as well as lookahead-aware cancellation algorithms, we demonstrate *MUTE*, a fully functional noise cancellation prototype that outperforms Bose's latest ANC headphone. Importantly, our design does not need to block the ear – the ear canal remains open, making it comfortable (and healthier) for continuous use.

2.1 Introduction

Ambient sound can be a source of interference. Loud conversations or phone calls in office corridors can be disturbing to others around. Working or napping at airports may be difficult due to continuous overhead announcements. In developing regions, the problem is probably most pronounced. Loud music or chants from public speakers, sound pollution from road traffic, or just

general urban cacophony can make simple reading or sleeping difficult. The accepted solutions have been to wear ear-plugs or ear-blocking headphones, both of which are uncomfortable for continuous use [16, 17, 18]. This chapter considers breaking away from convention and aims to cancel complex sounds without blocking the ear. We introduce our key idea next with a simple example.

Consider Alice being disturbed in her office due to frequent corridor conversations (Figure 2.1). Imagine a small IoT device – equipped with a microphone and wireless radio – pasted on the door in Alice's office. The IoT device listens to the ambient sounds (via the microphone) and forwards the exact sound waveform over the wireless radio. Now, *given that wireless signals travel much faster than sound*, Alice's noise cancellation device receives the wireless signal first, extracts the sound waveform from it, and gains a "*lookahead*" into the actual sound that will arrive later. When the actual sound arrives, Alice's ear-device is already aware of the signal and has had the time to compute the appropriate *anti-noise* signal. In fact, this lead time opens various other algorithmic and architectural opportunities, as will become clear in the subsequent discussions.

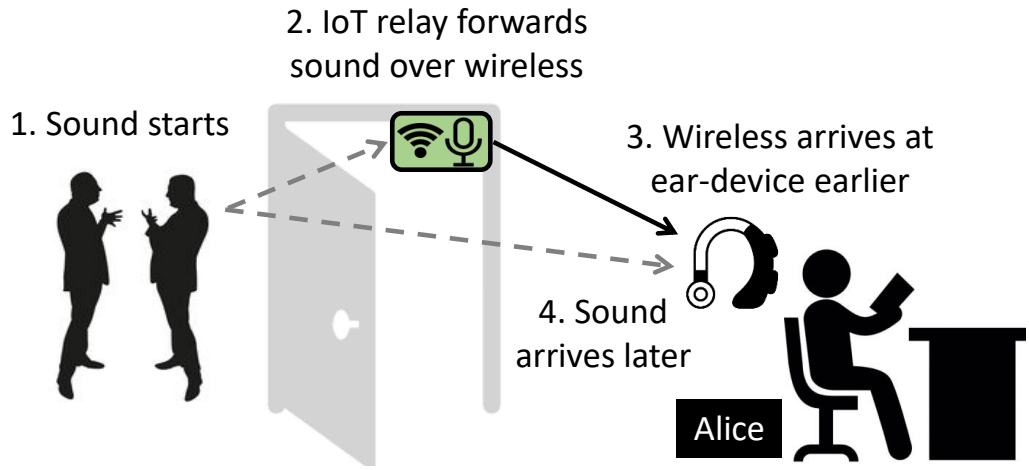


Figure 2.1: *MUTE* leverages the difference between wireless and acoustic propagation delay to provide a lookahead into the incoming sound signals.

In contrast, consider today's noise cancellation headphones from Bose [30, 31], SONY [32], Philips [33], etc. These headphones essentially contain a microphone, a DSP processor, and a speaker. The processor's job is to process the sound received by the microphone, compute the anti-noise signal, and play it through the speaker. This sequence of operations starts when the sound has arrived at the microphone; however, *it must complete before the same sound has reached the human's ear-drum.* Given the small distance between the headphone and the ear-drum, this is an extremely tight deadline ($\approx 30\ \mu$s [34]). The penalty of missing this deadline is a phase error, i.e., the anti-noise signal is not a perfect "opposite" of the actual sound, but lags behind. The lag increases at higher frequencies, since phase changes faster at such frequencies. This is one of the key reasons why current headphones are designed to only cancel low-frequency sounds below 1 kHz [35, 36], such as periodic machine noise. For high-frequency signals (e.g., speech and music), the headphones must use sound-absorbing materials. These materials cover the ear tightly and *attenuate* the sounds as best as possible [37, 30].

Meeting the tight deadline is not the only hurdle to real-time noise cancellation. As discussed later, canceling a sound also requires estimating the inverse of the channel from the sound source to the headphone's microphone. Inverse-channel estimation is a *non-causal* operation, requiring access to future sound samples. Since very few future samples are available to today's headphones, the anti-noise signal is not accurate, affecting cancellation quality.

With this background in mind, let us now return to our proposal of forwarding sound over wireless links. The forwarded sound is available to our cancellation device *several milliseconds* in advance of its physical arrival (as opposed to tens of microseconds in conventional systems). This lookahead presents three opportunities:

1 **Timing:** The DSP processor in our system can complete the anti-noise computation before the deadline, enabling noise cancellation for even higher frequencies. Hence, sound-absorbing materials to block the ear are not necessary.

2 **Profiling:** Lookahead allows the DSP processor to foresee macro changes in sound profiles, such as when Bob and Eve are alternating in a conversation. This allows for quicker multiplexing between filtering modes, leading to faster convergence at transitions.

3 **Channel Estimation:** Finally, much longer lookahead improves anti-noise computation due to better inverse-channel estimation, improving the core of noise cancellation.

Of course, translating these intuitive opportunities into concrete gains entails challenges. From an algorithmic perspective, the adaptive filtering techniques for classical noise cancellation need to be delicately redesigned to fully harness the advantages of lookahead. From an engineering perspective, the wireless relay needs to be custom-made so that forwarding can be executed in real-time (to maximize lookahead), and without storing any sound samples (to ensure privacy). This chapter addresses all these questions through a *lookahead-aware noise cancellation* (LANC) algorithm, followed by a custom-designed IoT transceiver at the 900 MHz ISM band. The wireless devices use frequency modulation (FM) to cope with challenges such as carrier frequency offset, non-linearities, and amplitude distortion.

Figure 2.2(b) shows the overall experimentation platform for our wireless noise cancellation system (*MUTE*). The custom-designed wireless relay is pasted on the wall, while the (crude) ear-device is laid out on the table. The ear-device has not been packaged into a wearable form factor; however, it is complete in functionality, i.e., it receives the wireless signals from the relay, extracts the audio waveform, and feeds it into a TI TMS320 DSP board run-

ning the LANC algorithm. Figure 2.2(a) visualizes the potential form-factor for such a wearable device (sketched in AutoDesk), while Figure 2.2(c) zooms into the relay hardware. To compare performance, we insert a "measurement microphone" into the ear position of the human head model – this serves as a virtual human ear. We place Bose's latest ANC headphone (QC35 [30]) over the head model, and compare its cancellation quality against *MUTE*, with different types of sounds, multipath environments, and lookahead times. Finally, we bring in five human volunteers to experience and rate the performance difference in noise cancellation. Our results reveal the following:

- *MUTE* achieves cancellation across $[0, 4]$ kHz, while Bose cancels only up to 1 kHz. Within 1 kHz, *MUTE* outperforms Bose by 6.7 dB on average.

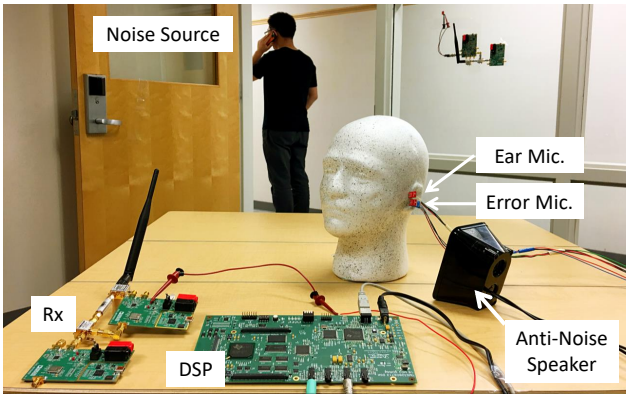
- Compared to Bose's full headphone (i.e., ANC at $[0, 1]$ kHz + sound-absorbing material for $[1, 4]$ kHz), our cancellation is 0.9 dB worse. We view this as a non-ear-blocking device with a slight compromise. With ear-blocking, *MUTE* outperforms Bose by 8.9 dB.

- *MUTE* exhibits greater agility for fast changing, intermittent sounds. The average cancellation error is reduced by 3 dB, and human volunteers consistently rate *MUTE* better than Bose for both speech and music.

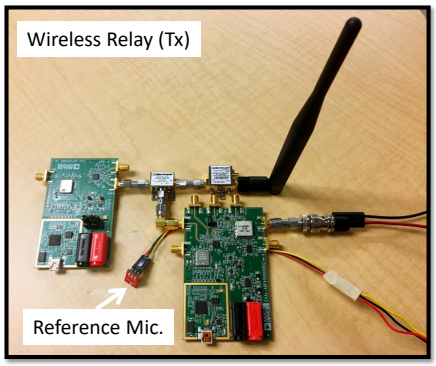
- Finally, Bose is advantaged with specialized microphones and speakers (with significantly less hardware noise); our systems are built on cheap microphone chips ($9) and off-the-shelf speakers ($19). Also, we have designed a mock ear-device to suggest how future earphones need not block the ear (Figure 2.2(a)). However, we leave the real packaging (and manufacturing) of such a device to future work.

In closing, we make the following contributions:

(a) Hollow Ear Device Model



(b) Overall WiNC Prototype



(c) Relay Hardware

Figure 2.2: *MUTE*'s experimental platform: Figure (a) shows our vision of the hollow ear-device, not covering the ear. Figure (b) shows the full system with the wireless IoT relay taped on the room's inside wall and the (crude) ear-device on the table (composed of a microphone on the human head model, an anti-noise speaker, and a DSP board). Figure (c) zooms into the relay hardware.

- We introduce *MUTE*, a wireless noise cancellation system architecture that harnesses the difference in propagation delay between radio frequency (RF) and sound to provide a valuable "lookahead" opportunity for noise cancellation.

- We present a *Lookahead Aware Noise Cancellation* (LANC) algorithm that exploits lookahead for efficient cancellation of unpredictable high frequency signals like human speech. Our prototype compares well with today's ANC headphones, but does not need to block the user's ears.

We expand on each of these contributions next, beginning with a brief primer on active noise cancellation (ANC), and followed by our algorithm, architecture, and evaluation.

2.2 Noise Cancellation Primer

An active noise cancellation (ANC) system has at least two microphones and one speaker (see Figure 2.3). The microphone placed closer to the ear-drum is called the *error microphone* M_e, while the one away from the ear is called the *reference microphone*, M_r. The speaker is positioned close to M_e and is called the *anti-noise speaker*. Ambient noise first arrives at M_r, then at M_e, and finally at the ear-drum. The DSP processor's goal is to extract the sound from M_r, compute the *anti-noise*, and play it through the speaker such that the *anti-noise* cancels the ambient noise at M_e.

Given that received sound is a combination of current and past sound samples (due to multipath), the DSP processor cannot simply reverse the sound samples from M_r. Instead, the various channels (through which the sound travels) need to be estimated correctly to construct the anti-noise signal. For this, the DSP processor uses the cancellation error from M_e as feedback and updates its channel estimates to converge to a better anti-

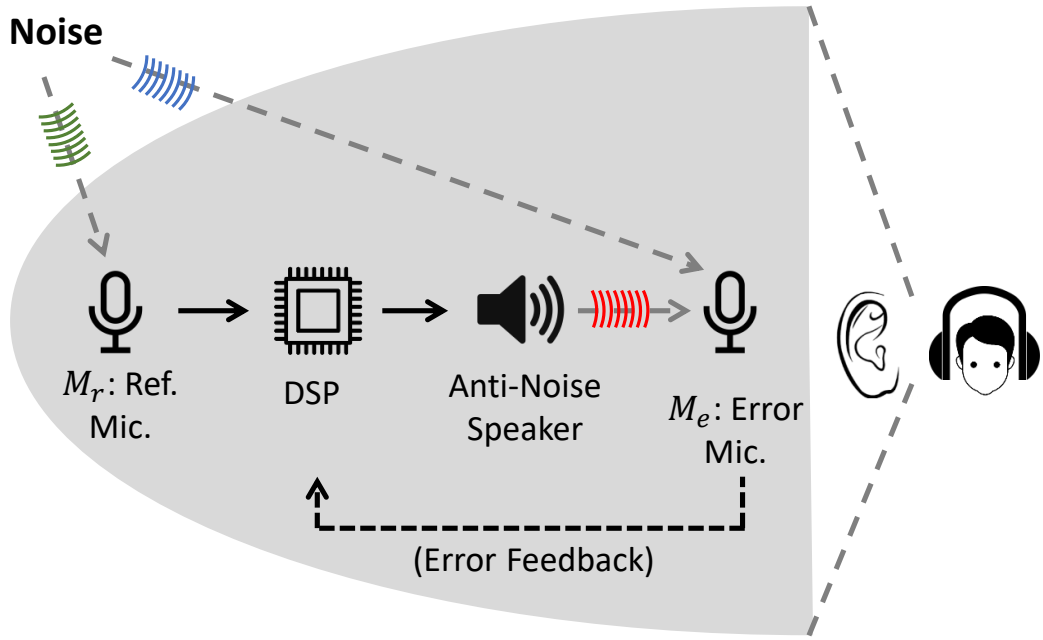


Figure 2.3: Basic architecture of an ANC headphone, currently designed for a single noise source.

noise in the next time step. Once converged, cancellation is possible at M_e regardless of the sound sample. So long as the ear-drum is close enough to M_e, the human also experiences similar cancellation as M_e.

■ **The ANC Algorithm:** Figure 2.4 recapitulates Figure 2.3 but from an algorithmic perspective. Observe that the error microphone M_e receives two signals, one directly from the noise source, say $a(t)$, and the other from the headphone's anti-noise speaker, say $b(t)$. The output of this microphone can be expressed as $e(t) = a(t) + b(t)$. For perfect cancellation, $e(t)$ would be *zero*.

Now, $a(t)$ can be modeled as $a(t) = h_{ne}(t) * n(t)$, where h_{ne} is the air channel from the noise source to M_e, $n(t)$ is the noise signal, and $*$ denotes convolution. Similarly, $b(t)$ can be modeled as:

$$b(t) = h_{se}(t) * \Big(h_{AF}(t) * \Big(h_{nr}(t) * n(t)\Big)\Big) \tag{2.1}$$

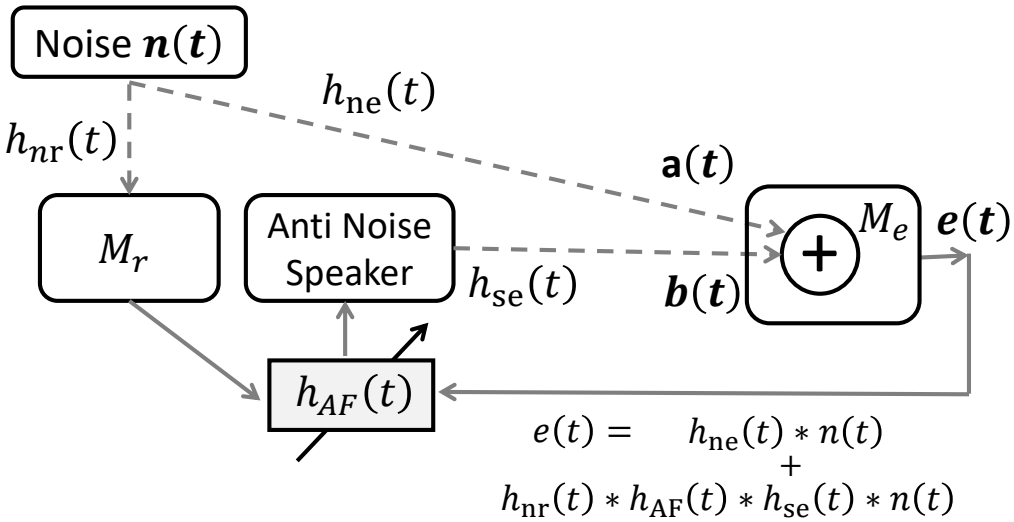


Figure 2.4: ANC block diagram.

Here, the inner-most parenthesis models the noise signal received by the reference microphone M_r over the channel $h_{nr}(t)$. The ANC algorithm in the DSP processor modifies this signal using an adaptive filter, $h_{AF}(t)$, and plays it through the anti-noise speaker. The speaker's output is distorted by the small gap between the speaker and the error microphone, denoted $h_{se}(t)$. Thus, the error signal $e(t)$ at the output of M_e is

$$\begin{aligned} e(t) &= a(t) + b(t) \\ &= h_{ne}(t) * n(t) + h_{se}(t) * \Big(h_{AF}(t) * \Big(h_{nr}(t) * n(t)\Big)\Big) \end{aligned}$$

For active noise cancellation, the ANC algorithm must design $h_{AF}(t)$ such that $e(t)$ is as close to 0 as possible. This suggests that $h_{AF}(t)$ should be set to:

$$h_{AF}(t) = -h_{se}^{-1}(t) * h_{ne}(t) * h_{nr}^{-1}(t) \tag{2.2}$$

In other words, ANC must estimate all three channels to apply the correct h_{AF}. Fortunately, h_{se}^{-1} can be estimated by sending a known preamble from the anti-noise speaker and measuring the response at the error microphone. However, h_{ne} and h_{nr}^{-1} cannot be easily estimated since: (1) the noise sig-

nal $n(t)$ does not exhibit any preamble-like structure, (2) the channels are continuously varying over time, and (3) the inverse channel requires future samples for precise estimation.

To cope with this difficulty in estimating h_{AF}, ANC uses adaptive filtering. The high-level idea is gradient descent, i.e., adjusting the values of the vector h_{AF} in the direction in which the residual error $e(t)$ goes down. Thus, ANC takes $e(t)$ as the feedback and feeds the classical least mean squares (LMS) technique [38, 39] – the output is an adaptive filter, $h_{AF}(t)$.

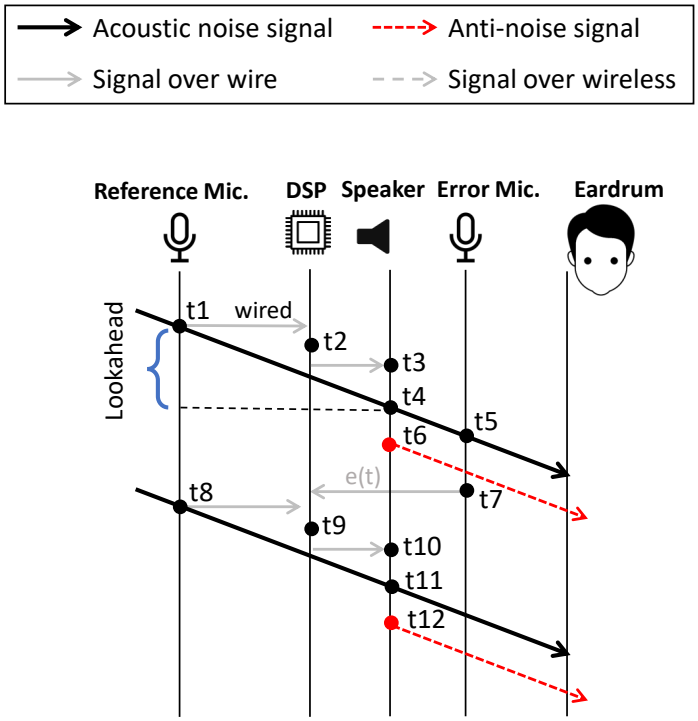
With this background, let us now focus on the lookahead advantage and corresponding design questions.

2.3 Lookahead Aware ANC

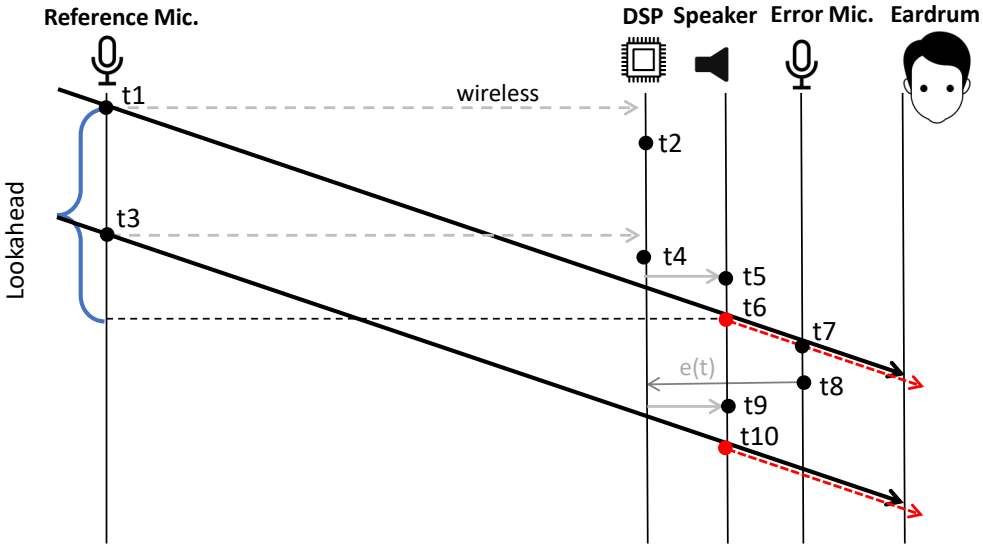
MUTE proposes a simple change to the conventional system architecture, namely: disaggregate the reference microphone M_r from the headphone, place M_r a few feet away towards the noise source, and replace the wired connection between M_r and the DSP processor with a wireless (RF) link. This separation significantly increases the lead time (or *lookahead*), translating to advantages in timing and cancellation. We detail the advantages next and then develop the Lookahead Award Noise Cancellation (LANC) algorithm.

2.3.1 Timing Advantage from Lookahead

Figure 2.5(a) shows the timeline of operations in today's ANC systems and Figure 2.5(b) shows the same, but with a large lookahead. Note that time advances in the downward direction with each vertical line corresponding to different components (namely, reference microphone, DSP processor, speaker, etc.). The slanting solid arrow denotes the arrival of the noise signal, while



(a) Today's ANC processing timeline



(b) *MUTE* processing timeline with large lookahead

Figure 2.5: Global timeline with (a) limited lookahead and (b) large lookahead. Time advances in the downward direction, and the slanted arrows denote the sound samples arriving from a noise source to the human ear. With large lookahead in (b), *MUTE* has adequate time to subsume all delays and play the anti-noise (red arrow) in time.

the black dots mark relevant events on the vertical timelines. We begin by tracing the sequence of operations step-by-step in Figure 2.5(a).

The noise signal first arrives at the headphone's reference microphone at time t_1. This sample is conveyed via wire and reaches the DSP processor at time t_2, where $(t_2 - t_1)$ is the ADC (analog-to-digital converter) delay. The DSP processor now computes the anti-noise sample and sends it to the anti-noise speaker at t_3, which outputs it after a DAC (digital-to-analog converter) and playback delay. Ideally, the speaker should be ready to play the anti-noise at t_4 since the actual sound wave is also passing by the speaker at this time. However, meeting this deadline is difficult since the distance between the reference microphone and speaker is smaller than 1 cm. With sound traveling at 340 m/s, the available time window is $(t_4 - t_1)$, which is around 30 μs. Since ADC, DSP processing, DAC and speaker delay can easily be 3$\times$ more than this time budget, today's ANC systems miss the deadline. Thus, instead of t_4, the anti-noise gets played at a later time t_6, as shown by the red dashed line in Figure 2.5(a).

For low frequencies, this can still deliver partial noise cancellation, since the phases of the noise and anti-noise would be slightly misaligned. However, for higher frequencies (i.e., smaller wavelengths), the performance would degrade since the excess delay (past t_4) would cause larger phase misalignment. This is the core struggle in today's noise cancellation systems.

Figure 2.5(b) illustrates how *MUTE* naturally relieves this time pressure. By virtue of being farther away, the reference microphone captures the noise signal earlier and forwards it over wireless (as shown by the horizontal dashed arrow at time t_1). The lookahead is far greater now, offering adequate time to subsume the ADC, DSP, DAC, and speaker delays. Hence, *MUTE* can compute the anti-noise sample and be ready to play it exactly when the actual noise arrives at the speaker at t_6. The anti-noise now coincides with

the noise, as shown by the black and red arrows in Figure 2.5(b). It should therefore be possible to cancel higher frequencies too.

To summarize, the following is a necessary condition for overcoming the timing bottleneck in ANC systems:

$$\textit{Lookahead} \geq \text{Delay at } \{\text{ADC + DSP + DAC + Speaker}\} \tag{2.3}$$

This condition raises the natural question: **How much lookahead does *MUTE* provide in practice?** Let us assume that noise travels a distance d_r to reach the reference microphone at the IoT relay, and a distance $d_e > d_r$ to reach the error microphone at the ear device. Since wireless signals travel at the speed of light, a million times faster than the speed of sound, forwarding the noise signal from the IoT relay is almost instantaneous. Hence, lookahead can be calculated as:

$$T_{lookahead} = \frac{d_e}{v} - \frac{d_r}{v} = \frac{(d_e - d_r)}{v} \tag{2.4}$$

where v is the speed of sound in air ($\approx$ 340 m/s). Translating to actual numbers, when $(d_e - d_r)$ is just 1 m, lookahead is $\approx$ 3 ms, which is 100$\times$ larger than today's ANC headphones. This fact implies that Alice can place the IoT relay on her office table and still benefit from wireless forwarding. Placing it on her office door, or ceiling, only increases this benefit.

2.3.2 Lookahead Aware ANC Algorithm

The timing benefit discussed above is a natural outcome of lookahead. However, we now (re)design the noise cancellation algorithm to explicitly exploit lookahead. Two key opportunities are of interest:

1. Recall from Equation 2.2 that the adaptive filter $h_{AF}(t)$ depends on the inverse channel, $h_{nr}^{-1}(t)$. Since this inverse is non-causal, the con-

struction of the anti-noise signal would require sound samples from the future (elaborated soon). Today's systems lack future samples, hence they suffer from suboptimal cancellation. Large lookahead with *MUTE* can close this gap.

2. Lookahead will help foresee macro changes in sound profiles, such as when different people are taking turns speaking. While traditional ANC incurs latency to converge to new sound profiles, *MUTE* can cache appropriate filters for each profile and "load" them at profile transitions. With lookahead, profile transitions would be recognizable in advance.

We begin with the first opportunity.

(1) Adaptive Filtering with Future Samples

■ **Basic Filtering:** Observe that a filter is essentially a vector, the elements of which are used to multiply the arriving sound samples. Consider an averaging filter that performs the average of the three most recent sound samples – this filter can be represented as a vector $h_F = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$. At any given time t, the output of the sound passing through this filter would be: $y(t) = \frac{1}{3}x(t) + \frac{1}{3}x(t-1) + \frac{1}{3}x(t-2)$ (which is called the convolution operation "*"). This filter is called *causal* since the output sample only relies on past input samples.

■ **Non-Causality:** Now consider the inverse of this filter ${h_F}^{-1}$. This should be another vector which convolved with $y(t)$ should give back $x(t)$, i.e., $x(t) = {h_F}^{-1} * y(t)$. Filtering theory says that this inverse needs to be carefully characterized, since they are *non-causal*, *unstable*, or both [40, 41]. With a non-causal inverse, determining $x(t)$ would require $y(t+k)$ for $k > 0$. Thus estimating $x(t)$ in real time would be difficult; future knowledge of $y(t)$ is necessary. The physical intuition is difficult to convey concisely; however,

one way to reason about this is that $x(t)$ originally influenced $y(t+1)$ and $y(t+2)$, and hence, recovering $x(t)$ would require those future values as well. In typical cases where h_F is the room's impulse response (known to have non-minimum phase property [42]), the future samples needed could be far more [40, 43].

■ **Adaptive Filtering:** Now, let us turn to adaptive filtering (h_{AF}) needed for noise cancellation. The "adaptive" component arises from estimating the filter vector at a given time, convolving this vector with the input signal, and comparing the output signal against a target signal. Depending on the error from this comparison, the filter vector is *adapted* so that successive errors converge to a minimum. Since this adaptive filter is non-causal (due to its dependence on the inverse filter), it would need future samples of the input signal to minimize error. With partial or no future samples (i.e., a truncated filter), the error will be proportionally higher. With this background, let us now design the LANC algorithm to fully exploit the lookahead.

■ **LANC Design:** Recall from Section 3.2 that the adaptive filter needed for noise cancellation is $h_{AF}(t) = -h_{se}^{-1}(t) * h_{ne}(t) * h_{nr}^{-1}(t)$. This minimizes the error:

$$e(t) = h_{ne}(t) * n(t) + h_{se}(t) * h_{AF}(t) * x(t) \tag{2.5}$$

where $x(t)$ is the noise captured by the reference microphone, i.e., $x(t) = h_{nr}(t) * n(t)$. Now, to search for the optimal h_{AF}, we use steepest gradient descent on the squared error $e^2(t)$. Specifically, we adapt h_{AF} in a direction opposite to the derivative of the squared error:

$$h_{AF}^{(new)} = h_{AF}^{(old)} - \frac{\mu}{2}\frac{\partial e^2(t)}{\partial h_{AF}} \tag{2.6}$$

where μ is a parameter that governs the speed of gradient descent. Expanding

the above equation for each filter coefficient $h_{AF}(k)$, we have:

$$h_{AF}^{(new)}(k) = h_{AF}^{(old)}(k) - \mu e(t) h_{se}(t) * x(t-k) \tag{2.7}$$

In the above equation, $h_{se}(t)$ is known and estimated a priori, $e(t)$ is measured from the error microphone, and $x(t)$ is measured from the reference microphone.

This is where non-causality emerges. Since h_{AF} is actually composed of h_{nr}^{-1}, the values of k in Equation 2.7 can be negative ($k < 0$). Thus, $x(t-k)$ becomes $x(t+k)$, $k > 0$, implying that the updated $h_{AF}^{(new)}$ requires future samples of $x(t)$. With lookahead, our LANC algorithm is able to "peek" into the future and utilize those sound samples to update the filter coefficients. This naturally results in a more accurate anti-noise signal $\alpha(t)$, expressed as:

$$\alpha(t) = h_{AF}(t) * x(t) = \sum_{k=-N}^{L} h_{AF}(k) x(t-k) \tag{2.8}$$

Observe that the larger the lookahead, the larger the value of N in the subscript of the summation, indicating a better filter inversion. Thus, with a lookahead of *several milliseconds* in LANC, N can be large and the anti-noise signal can significantly reduce error (see pseudocode in Algorithm 1). In contrast, lookahead is *tens of microseconds* in today's headphones, forcing a strict truncation of the non-causal filter, leaving a residual error after cancellation.

(2) Predictive Sound Profiling

Another opportunity with lookahead pertains to coping with more complex noise sources, such as human conversation. Consider a common case where a human is talking intermittently in the presence of background noise – Figure 2.6(a) and (b) show example spectra for speech and background noise,

Algorithm 1 LANC: Lookahead Aware Noise Cancellation

```
while True do
    Play α(t) at anti-noise speaker
    t = t + 1
    Record the error e(t) at error mic.
    Record future sample x(t + N) at reference mic.
    for k = −N, k ≤ L, k++ do
        h_AF(k) = h_AF(k) − μe(t)h_se(t) ∗ x(t − k)
    end for
    α(t) = Σ_{k=−N}^{L} h_AF(k)x(t − k)
end while
```

respectively. Now, to cancel human speech, the adaptive filter estimates the channels from the human to the ear device. However, when the speech pauses, the filter must reconverge to the channels from the background noise source. Reconvergence incurs latency since the h_{AF} vector must again undergo the gradient descent process to stabilize at a new minimum. Our idea is to leverage lookahead to foresee this change in sound profile, and swap the filtering coefficients right after the speech has stopped. Hence, we expect our cancellation to not fluctuate even for alternating sound sources, like speech or music.

■ **Validation:** Figure 2.7 explains the problem by illustrating the convergence of a toy adaptive filter, h_{AF}, with 7 taps. Initially, the filter is $h_{AF}^{(1)}$, and since this vector is not accurate, the corresponding error in Figure 2.7(b) is large. The vector then gets updated to $h_{AF}^{(2)}$ based on Equation 2.7, in the direction that reduces the error. This makes $h_{AF}^{(2)}$ closer to the ideal filter and $e(t)^2$ closer to zero. The filter continues to get updated until the error becomes nearly zero – at this point, the filter is said to have converged, i.e., $h_{AF}^{(3)}$.

For persistent noise (like machine hum), the converged adaptive filter can

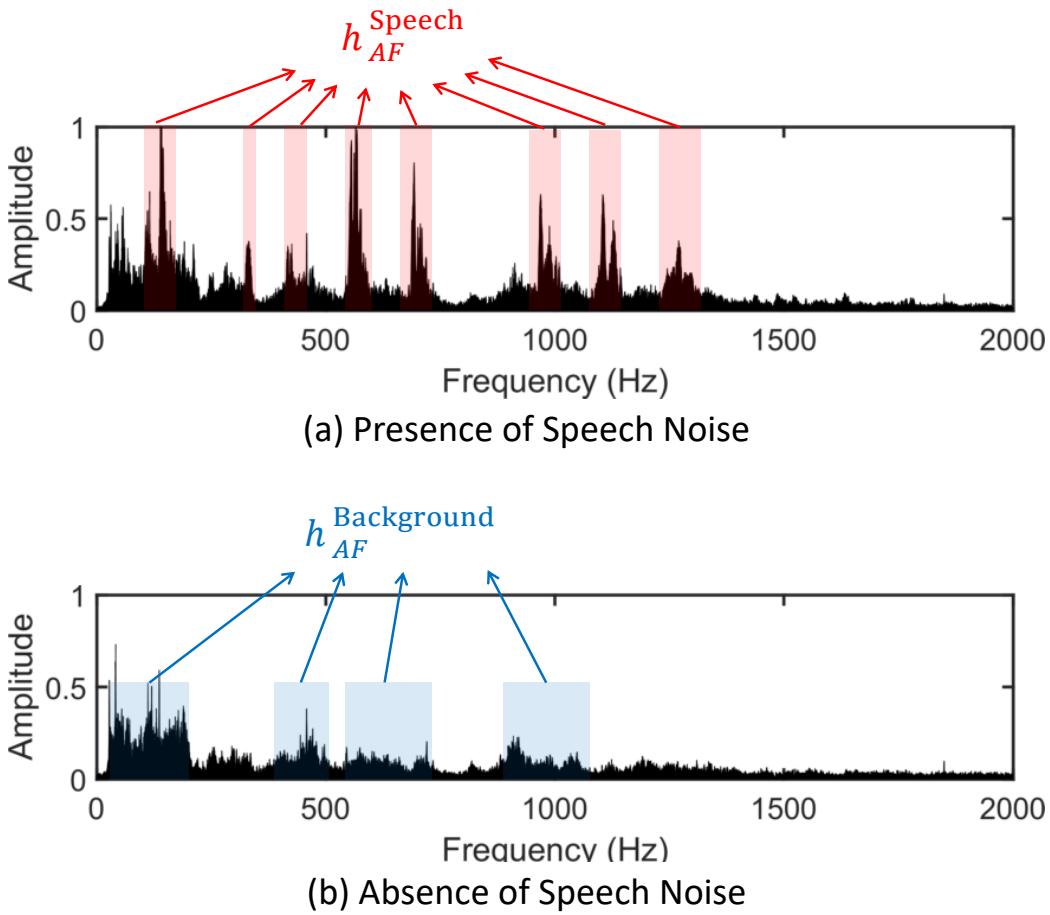


Figure 2.6: Acoustic spectrum in the (a) presence and (b) absence of speech. LANC recognizes the profile and pre-loads its filter coefficients for faster convergence.

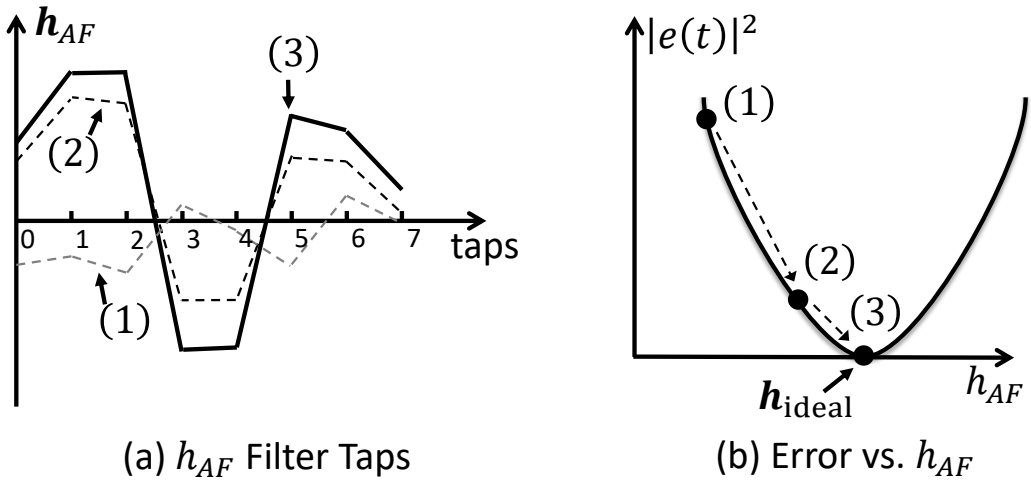


Figure 2.7: Convergence process of the adaptive filter, h_{AF}. (a) 7-tap h_{AF} filter changes from time (1) to time (3). (b) Residual error $e(t)$ converges to a minimum.

continue to efficiently cancel the noise, as shown in Figure 2.8(a). However, for intermittent speech signals with random pauses between sentences, the adaptive filter cannot maintain smooth cancellation as shown in Figure 2.8(b). Every time the speech starts, the error is large and the adaptive filter needs time to (re)converge again.

■ **Predict and Switch:** With substantial lookahead, LANC gets to foresee the start and stop of speech signals. Thus, instead of adapting the filter coefficients every time, we cache the coefficient vector for the corresponding sound profiles. A sound profile is essentially a statistical signature for the sound source – a simple example is the average energy distribution across frequencies. For two profiles – say speech and background noise – LANC caches two adaptive filter vectors, h_{AF}^{speech} and $h_{AF}^{background}$, respectively. Then, *by analyzing the lookahead buffer in advance, LANC determines if the sound profile would change imminently.* When the profile change is indeed imminent (say the starting of speech), LANC directly updates the adaptive filter with h_{AF}^{speech}, avoiding the overhead of reconvergence.

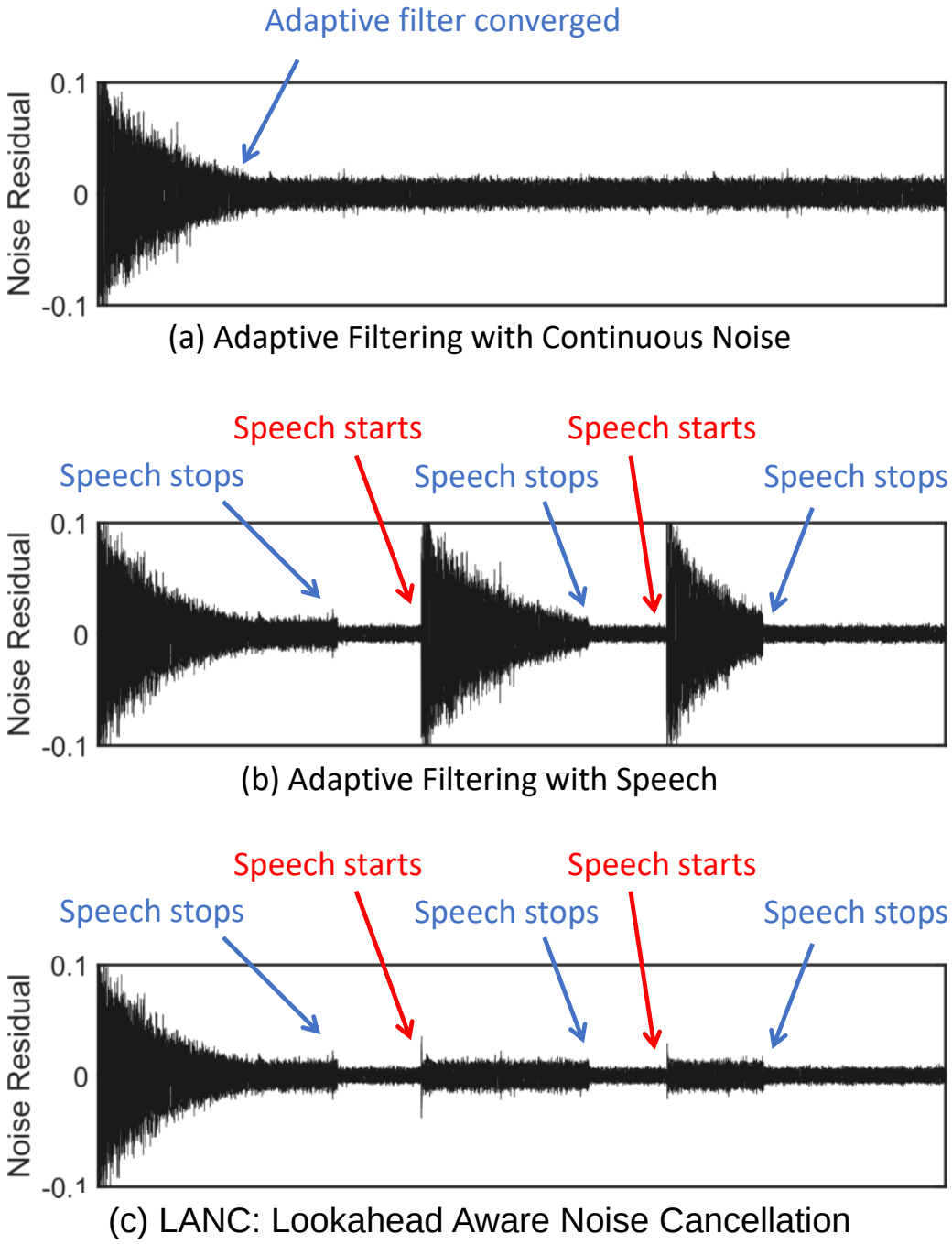


Figure 2.8: LANC's convergence timeline showing adaptive filtering with (a) continuous noise, (b) speech, (c) lookahead aware profiling. LANC converges faster due to its ability to anticipate profile transitions in advance.

To generalize, LANC maintains a converged adaptive filter for each sound profile, and switches between them at the right time. So long as there is one dominant sound source at any given time, LANC cancels it quite smoothly as shown in Figure 2.8(c). Without lookahead, however, the profile-change cannot be detected in advance, resulting in periodic reconvergence and performance fluctuations.

With the LANC algorithm in place, we now turn to bringing together the overall *MUTE* system.

2.4 *MUTE*: System and Architecture

Recall that our basic system requires an IoT relay installed near the user; the relay listens to the ambience and streams the acoustic waveform over its RF interface in real time. The receiver – a hollow earphone – receives the sound signal, applies the LANC algorithm to compute the anti-noise signal, and finally plays it through the speaker. Several components have been engineered to achieve a fully functional system. In the interest of space, we discuss three of these components, namely: (1) the wireless relay hardware, (2) automatic relay selection, and (3) privacy protection. Finally, as a conclusion to this section, we envision architectural variants of *MUTE* – such as noise cancellation as a service – to demonstrate a greater potential of our proposal beyond what is presented in this chapter. We begin with wireless relay design.

2.4.1 Wireless Relay Design

MUTE's RF forwarding consists of two main components: A *relay* that captures acoustic noise, converts it to RF and transmits on the wireless channel and a *receiver* that captures the wireless signals and converts them back to

acoustic signals. Our design leverages an analog architecture to eliminate delays from digitization and processing. We use frequency modulation (FM) in the 900 MHz ISM band. Figure 2.9 shows the high-level design of the relay and receiver.

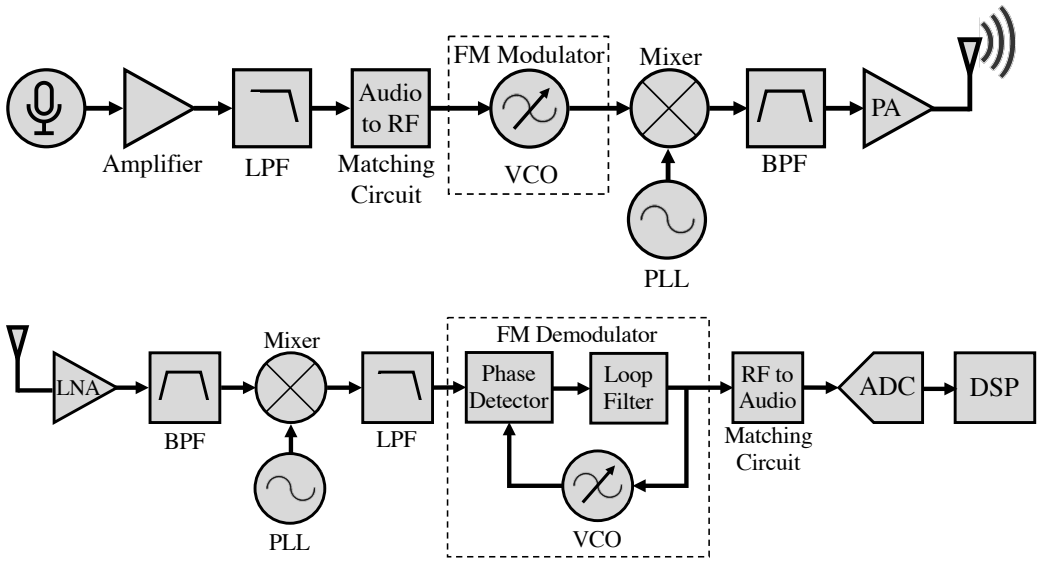


Figure 2.9: *MUTE*'s RF relay design: (a) Acoustic-to-RF relay, (b) RF-to-acoustic receiver.

Acoustic-to-Wireless Relay: This device consists of a (reference) microphone that captures the ambient noise signal, passes it through a low pass filter (LPF), and then amplifies it. An impedance matching circuit connects the audio signal to an RF VCO (voltage controlled oscillator). The VCO outputs a frequency modulated signal which is then mixed with a carrier frequency generated by a phase lock loop (PLL), and up-converted to the 900 MHz ISM band. The RF signal is then band pass filtered and passed to a power amplifier connected to a 900 MHz antenna. We ensure that the relay's transmit power complies with FCC regulations in the ISM band.

Given an audio signal $m(t)$ captured at the microphone, the transmitted

signal $x(t)$ is:

$$x(t) = A_p \cos\left(2\pi f_c t + 2\pi A_f \int_0^t m(\tau) d\tau\right) \tag{2.9}$$

where f_c is the carrier frequency, A_p is the gain of the RF amplifier, and A_f is the combined gain of the audio amplifier and FM modulator.

Wireless-to-Acoustic Receiver: Figure 2.9(b) shows the architecture of the *receiver*. This analog receiver captures the RF signal and passes it to a low noise amplifier. The signal is then band pass filtered, down-converted to a low intermediate frequency using an RF mixer, and then low pass filtered. The received signal can then be written as:

$$y(t) = h_w(t) * A_p \cos\left(2\pi f_i t + 2\pi A_f \int_0^t m(\tau) d\tau\right) + n(t) \tag{2.10}$$

where $h_w(t)$ is the wireless channel, f_i is a low intermediate frequency, and $n(t)$ is the noise. The signal is then demodulated using a phase detector, loop filter, and a VCO, to extract the audio signal $m(t)$. An impedance matching circuit then connects the audio signal to an ADC (analog-to-digital converter) where it is sampled at 16 MS/s before being passed to the DSP (digital signal processor).

Why Frequency Modulation (FM)? The significance of FM is three-fold. *First*, it delivers better audio quality because noise mainly affects amplitude, leaving the frequency of the signal relatively less affected. *Second*, since the bandwidth used is narrow, $h_w(t)$ is flat in frequency and hence can be represented with a single tap. As a result, there is no need to estimate the wireless channel since it will not affect the audio signal $m(t)$. *Finally*, any carrier frequency offsets between up-conversion and down-conversion appear as a constant DC offset in the output of the FM demodulator which can easily be averaged out. This precludes the need to explicitly compensate for

carrier frequency offset (CFO).

2.4.2 Automatic Relay Selection

MUTE is effective only when the wireless relay is located closer to the sound source than the earphone. This holds in scenarios such as Figure 2.1 – the relay on Alice's door is indeed closer to the noisy corridor. However, if the sound arrives from an opposite direction (say from a window), the relay will sense the sound *after* the earphone. Even though the relay forwards this sound, the earphone *should not* use it since the lookahead is negative now (i.e., the wirelessly-forwarded sound is lagging behind). Clearly, *MUTE* must discriminate between positive and negative lookahead, and in case of the latter, perhaps nudge the user to reposition the relay in the rough direction of the sound source.

- **How to determine positive lookahead:** *MUTE* uses the GCC-PHAT cross-correlation technique [44]. The DSP processor periodically correlates the wirelessly-forwarded sound against the signal from its error microphone. The time of correlation-spike tells whether the lookahead is positive or negative. When positive, the LANC algorithm is invoked. Correlation is performed periodically to handle the possibility that the sound source has moved to another location.

- **Multiple relays:** Observe that a user could place multiple relays around her to avoid manually repositioning the relay in the direction of the noise source. The correlation technique would still apply seamlessly in such a scenario. The relay whose correlation spike is most shifted in time is the one *MUTE* would pick. This relay would offer the maximum lookahead, hence the best cancellation advantage.

2.4.3 Architectural Variants

The basic architecture thus far is a wireless IoT relay (closer to the sound source) communicating to an ear-device around the human ear. We briefly sketch a few variants of this architecture aimed at different trade-offs and applications.

1. **Personal Tabletop:** *MUTE* removes the reference microphone from the headphone, which in turn eliminates the noise-absorbing material. As mentioned earlier, this makes the ear-device light and hollow. Following this line of reasoning, one could ask what else could be stripped off from the ear-device. We observe that even the DSP can be extracted and inserted into the IoT relay. In other words, the IoT relay could compute the anti-noise and wirelessly transmit to the ear-device; the ear-device could play it through the anti-noise speaker, and transmit back the error signal from its error microphone. Observe that the IoT relay can even become a portable tabletop device, with the ear-device as a simple "client". The user can now carry her personal *MUTE* tabletop relay (Figure 2.10(a)), eliminating dependencies on door or wall mounted infrastructure.

2. **Public Edge Service:** Another organization is to move the DSP to a backend server, and connect multiple IoT relays to it, enabling a *MUTE* public service (Figure 2.10(b)). The DSP processor can compute the anti-noise for *each* user and send it over RF. If computation becomes the bottleneck with multiple users, perhaps the server could be upgraded with multiple-DSP cores. The broader vision is an edge cloud [45] that offers acoustic services to places like call centers.

3. **Smart Noise:** A third architecture could be to attach IoT relays to noise sources themselves (and eliminate the relays on doors or ceilings). Thus, heavy machines in construction sites, public loudspeakers,

or lawn mowers, could broadcast their own sound over RF. Those disturbed by these noises can wear the *MUTE* ear-device, including the DSP. Given the maximal lookahead, high quality cancellation should be feasible.

We conclude by observing that the above ideas may be viewed as a "disaggregation" of conventional headphones, enabling new future-facing possibilities. This chapter is an early step in that direction.

2.4.4 Privacy Awareness

Two relevant questions emerge around privacy:

- **Will the IoT relay record ambient sounds and conversations?** We emphasize that the relays are analog and not designed to even hold the acoustic samples. The microphone's output is directly applied to modulate the 900 MHz carrier signal with no recording whatsoever. In this sense, *MUTE* is different from Amazon Echo, Google Home, and wireless cameras that must record digital samples for processing.

- **Will the wirelessly forwarded sound reach certain areas where it would not have been audible otherwise?** This may be a valid concern for some scenarios, e.g., a person outside a coffee shop may be able to "hear" inside conversations. However, with power control, beamforming, and sound scrambling, the problem can be alleviated. We leave a deeper treatment of this problem to future work. On the other hand, this may not be a problem in other scenarios. For instance, with personal tabletop devices, the wireless range can be around the user's table, resulting in almost no leakage. For smart noise, the noise need not be protected at all, while for call-center-like settings, acoustic privacy is relatively less serious.

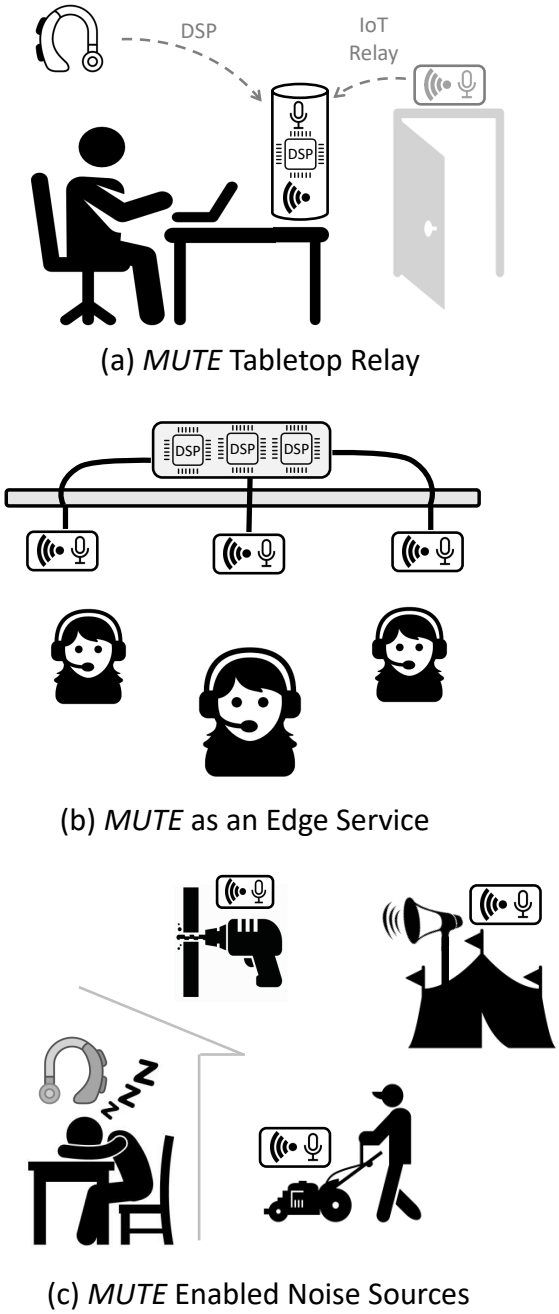


Figure 2.10: Architectural variants: (a) Personal tabletop device includes DSP and reference microphone; sends anti-noise signal to ear-device, which responds with error signal. (b) Noise cancellation as an edge service: the DSP server is connected to IoT relays on the ceiling and computes the anti-noise for all users. (c) Smart noise, where noise sources attach a IoT relay while users with *MUTE* ear-devices benefit.

2.5 Evaluation

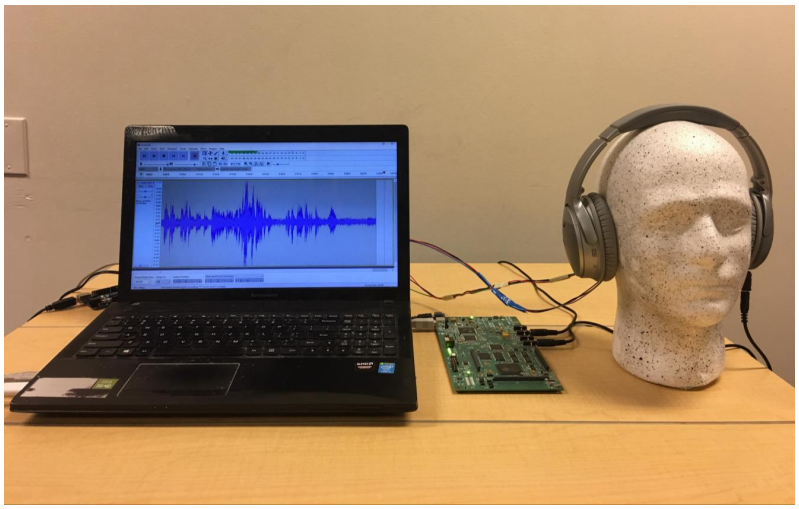
We begin with some details on experimental setup and comparison schemes, followed by performance results.

2.5.1 Experimental Setup

MUTE's core algorithms are implemented on the Texas Instrument's TMS-320C6713 DSP board [46], equipped with the TLV320AIC23 codec. The microphones are SparkFun's MEMS Microphone ADMP401 and the anti-noise speaker is the AmazonBasics computer speaker. Ambient noise is played from an Xtrememac IPU-TRX-11 speaker. All microphones and speakers are cheap off-the-shelf equipment. For performance comparison, we purchased Bose's latest ANC headphone, the QC35 [30] (pictured in Figure 2.11).

For experimentation, we insert a separate "measurement microphone" at the ear-drum location of a 3D head model (Figure 2.2(b)) – this serves as the approximation of what the human would hear. We play various sounds from the ambient speaker and measure the power level at this microphone. We then compare the following schemes:

- ***MUTE*_Hollow:** Our error microphone is pasted outside the ear while the anti-noise speaker and DSP board are placed next to it, as shown in Figure 2.2(b).
- **Bose_Active:** We place the Bose headphone on the 3D head model and measure cancellation, first with ANC turned OFF, and then with ANC turned ON. Subtracting the former from the latter, we get Bose's active noise cancellation performance.
- **Bose_Overall:** We turn on ANC for Bose and measure the net cancellation, i.e., the combination of its ANC and passive noise-absorbing material.



(a) Bose headphone on the head

(b) Measurement microphone inside the ear

Figure 2.11: *MUTE*+Passive: (a) Bose headphone on the 3D head model, with DSP output connected to the headset. (b) The measurement microphone inside the ear, and the reference microphone nearby.

Finally, we recruit human volunteers to compare Bose and *MUTE*. In the absence of a compact form factor for *MUTE*, we utilize Bose's headphone. Specifically, we feed the output of our DSP board into the AUX input of the Bose headphone (with its ANC turned OFF), meaning that our LANC algorithm is executed through Bose's headphone (instead of its native ANC module). Of course, the passive sound absorbing material now benefits both Bose and *MUTE*, hence we call our system ***MUTE*+Passive** (see Figure 2.11). We report cancellation results for various sounds, including machines, human speech, and music.

2.5.2 Performance Results

Our experiment aims to establish the following:

1. Comparison of overall noise cancellation for *MUTE*_Hollow, Bose_Active, Bose_Overall, and *MUTE*+Passive.
2. Performance comparison for various sound types.
3. Human experience for Bose_Overall and *MUTE*+Passive.
4. Impact of lookahead length on *MUTE*_Hollow.
5. Accuracy of relay selection for *MUTE*_Hollow.

■ Overall Noise Cancellation

Figure 2.12 reports comparative results when wide-band white noise (which is most unpredictable of all noises) is played from the ambient speaker. The noise level is maintained at 67 dB at the measurement microphone. Four main points are evident from the graph. (1) Bose_Active is effective only at lower frequency bands, implying that Bose must rely on passive materials

to cancel sounds from 1 kHz to 4 kHz. (2) The ear-blocking passive material is effective at higher frequencies, giving Bose_Overall a −15 dB average cancellation. (3) *MUTE*_Hollow is almost comparable to Bose_Overall even without passive materials, indicating that our LANC algorithm performs well (Bose_Overall is just 0.9 dB better on average). (4) When *MUTE*+Passive gains the advantage of passive materials, the cancellation is 8.9 dB better than Bose_Overall, on average.

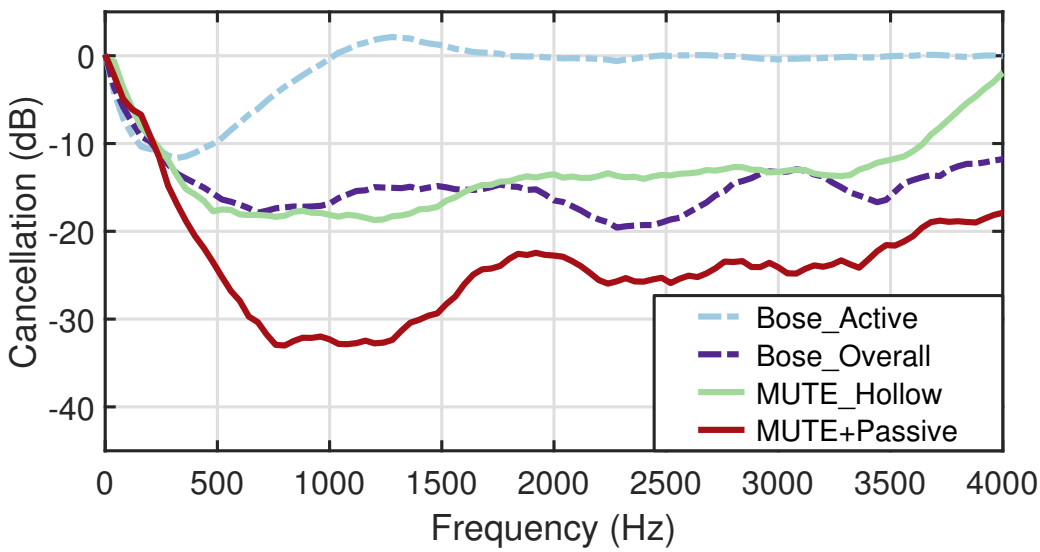


Figure 2.12: *MUTE* and Bose's overall performance.

In summary, *MUTE* offers two options in the cancellation versus comfort tradeoff. A user who values comfort (perhaps for long continuous use) can prefer lightweight, open-ear *MUTE* devices at a 0.9 dB cost compared to Bose, while one who cares more about noise suppression can experience 8.9 dB improvement over Bose.

We briefly discuss two technical details here: (1) *MUTE*'s cancellation is capped at 4 kHz due to limited processing speed of the TMS320C6713 DSP. It can sample at most 8 kHz to finish the computation within one sampling interval. A faster DSP will ease the problem. (2) The diminishing cancellation at very low frequencies ($<$ 100 Hz) is due to the weak response of our

cheap microphone and anti-noise speaker – Figure 2.13 plots the combined frequency response.

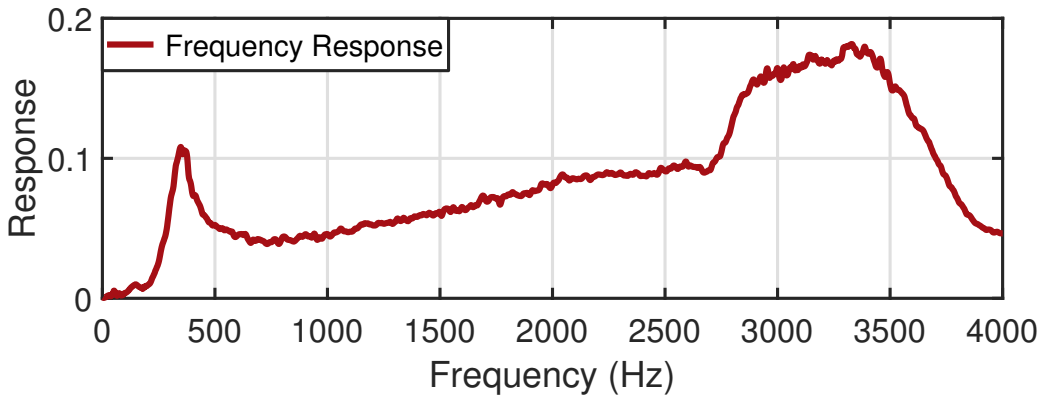


Figure 2.13: The combined frequency response of our anti-noise speaker and the microphone.

■ Varying Ambient Sounds (Speech, Music)

Figure 2.14 shows *MUTE*'s cancellation performance across 4 different types of real-world noises with different spectral characteristics: male voice, female voice, construction sound, and music. The results are a comparison between *MUTE*_Hollow and Bose_Overall. Our lookahead-aware ANC algorithm achieves mean cancellation within 0.9 dB to Bose's native ANC combined with its carefully perfected passive sound-absorbing materials [30].

■ Human Experience

We invited 5 volunteers to rate *MUTE*+Passive's performance relative to Bose_Overall. Recall that for *MUTE*+Passive, we use the Bose headset with ANC turned OFF. Now, since we have only one DSP board, we were able to run *MUTE*+Passive only on the right ear – for the left ear, we use both an earplug and the headset (with ANC turned OFF). For Bose_Overall, we

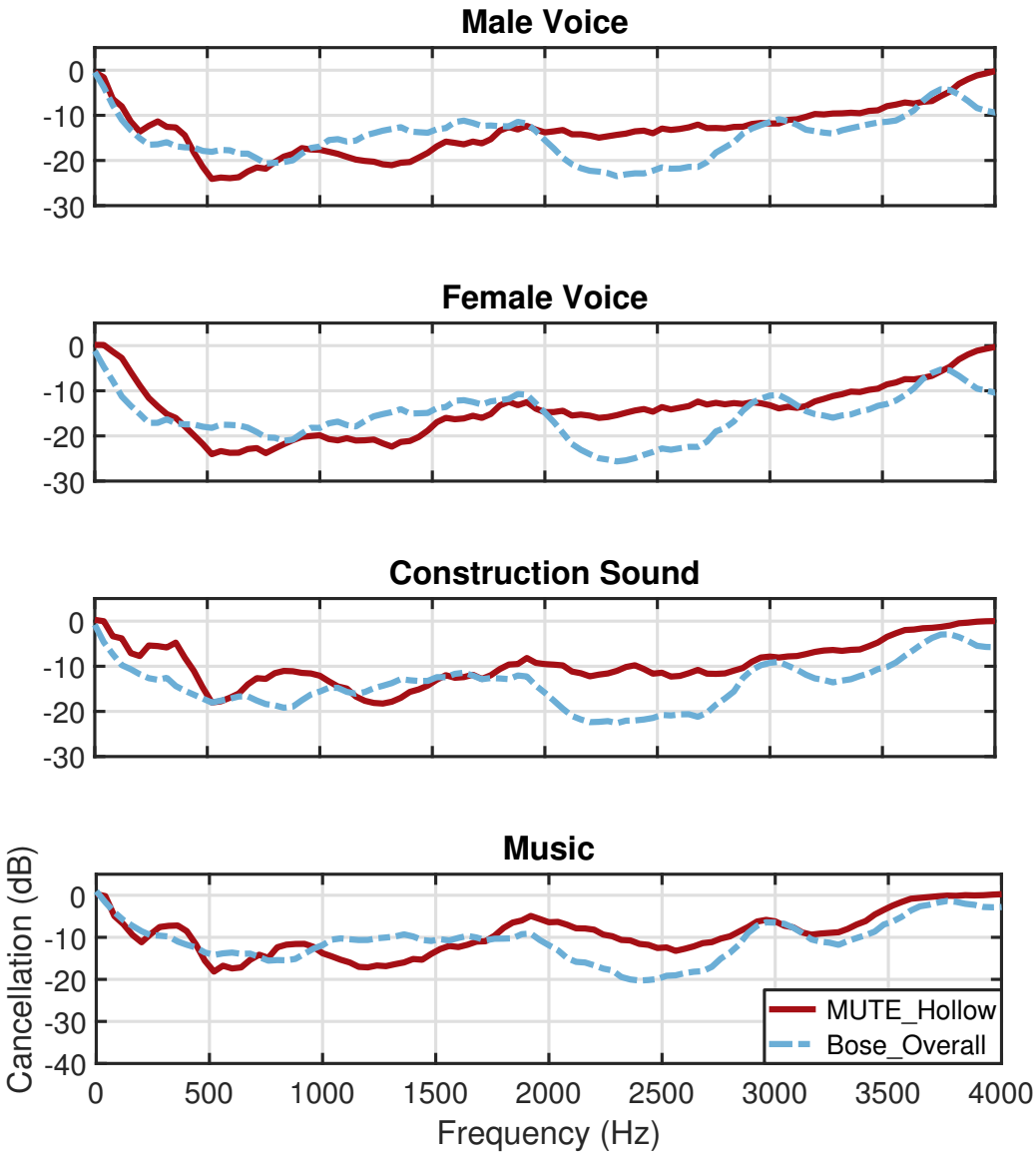


Figure 2.14: Comparison between *MUTE*_Hollow and Bose_Overall, measured for 4 types of ambient sounds.

turned ON native ANC on both ears. In this setup, we played various human voices and music through the ambient speaker. Since fine grained (per-frequency) comparison is difficult for humans, we requested an overall rating between 1 and 5 stars. We did not tell the volunteers when *MUTE* or Bose was being used for cancellation.

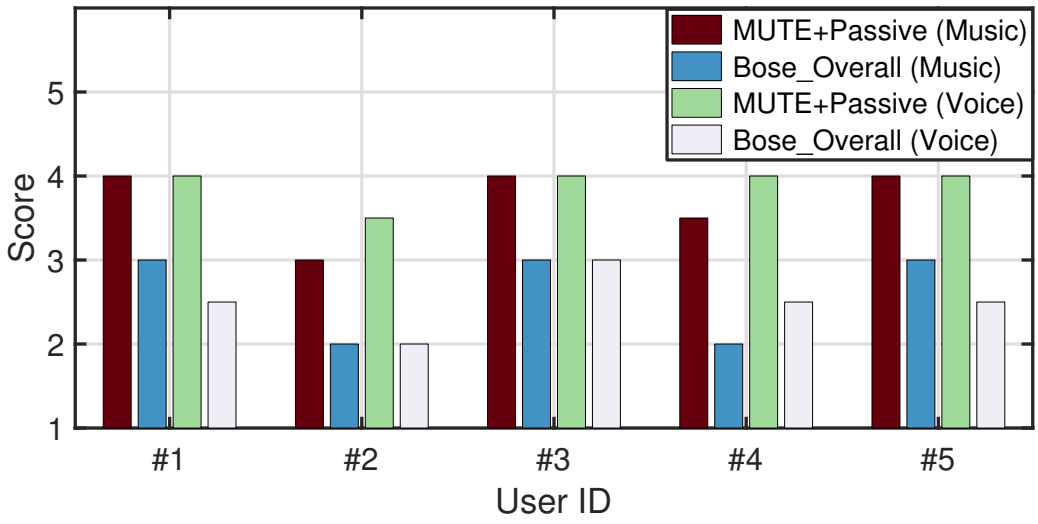


Figure 2.15: User feedback of music and voice noise.

Figure 2.15 shows the comparison for music and human voice. Every volunteer consistently rated *MUTE* above Bose. Their subjective opinions were also strongly positive. However, almost all of them also said that "Bose was superb at canceling hums in the environment", and *MUTE* did not perform as well. One reason is the weak response of the speaker and microphone at low frequencies, as mentioned before. Upon analyzing, we also realized that the background hums are from various sources. With Bose's microphone array, they are equipped to handle such scenarios, while our current system is aimed at a single noise source (the ambient speaker). We have left multi-source noise cancellation to future work, as discussed later in Section 3.5.

■ Impact of Shorter Lookahead

Lookahead reduces when the wireless relay gets closer to the user, or when the location of the noise source changes such that the time-difference between direct path and wireless-relay path grows smaller. For accurate comparison across different lookaheads, we need to ensure that the physical environment (i.e., multipath channel) remains identical. Therefore, instead of physically moving the noise source or the wireless relay (to vary lookahead time), we fix their positions, but deliberately inject delays into the reference signal within the DSP processor (using a delayed line buffer).

Figure 2.16 plots the results for *MUTE*_Hollow. The lookahead times are expressed relative to the "Lower Bound" from Equation 2.3 (recall that lookahead must be greater than ADC + DSP processing + DAC + speaker delay, as explained in Section 2.3.1). Evidently, as the lookahead increases, the performance improves due to better inverse filtering.

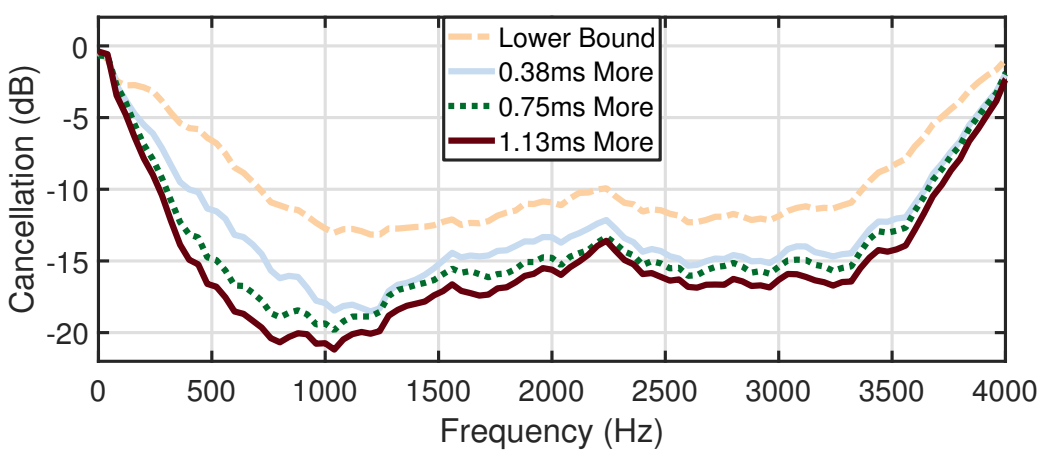


Figure 2.16: As lookahead becomes smaller, the system performance degrades.

■ Profiling and Cancellation

To highlight the efficacy of sound profiling and filter switching, we run a separate experiment where wide-band background noise is constantly being played from one ambient speaker, while mixed human voice (with pauses) is being played from another speaker. We compare the residual error of *MUTE*'s filter selection mechanism with that of using only one adaptive filter. Figure 2.17 shows the cancellation gain in *MUTE*_Hollow with profiling and switching turned ON. Evidently, the cancellation improves by 3 dB on average. We could not compare with Bose in this case since Bose uses at least 6 microphones to cope with scattered noise sources. Upgrading *MUTE* with that many microphones is bound to offer substantial advantage.

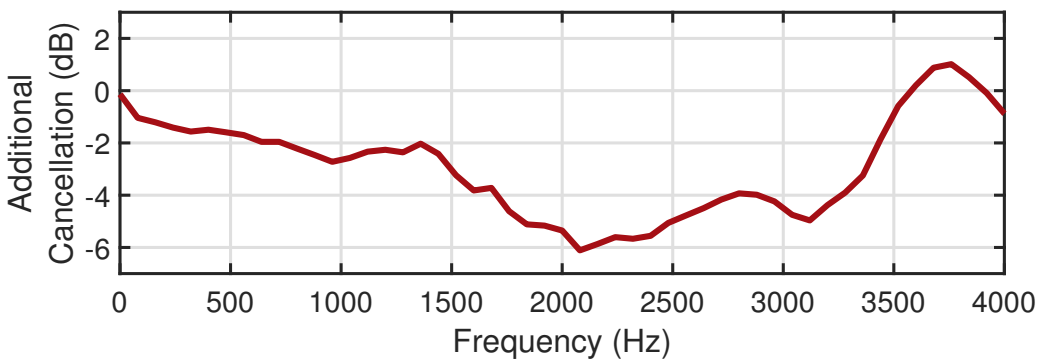


Figure 2.17: Lookahead enabled filter switching provides additional gain for intermittent noise cancellation.

■ Wireless Relay Selection

Does the correlation technique to identify (maximum) positive lookahead work in real environments? Figure 2.18 shows two typical examples of GCC-PHAT based cross-correlation between the forwarded sound waveform and the directly-received sound. Observe that one case is positive lookahead while the other is negative. *MUTE* was able to correctly determine these cases in every instance.

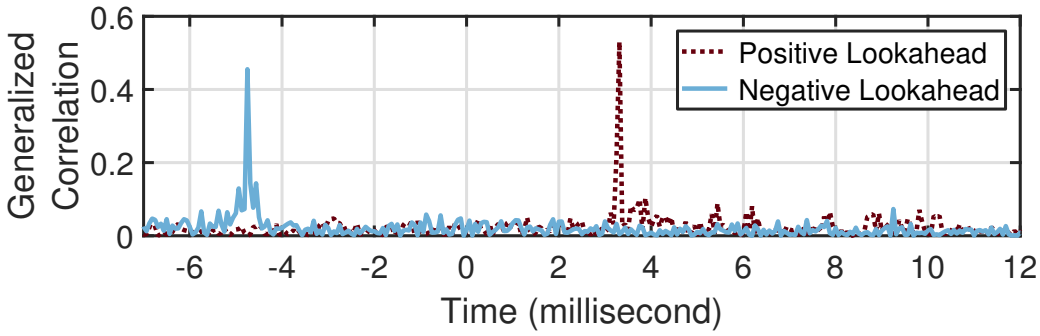


Figure 2.18: *MUTE* client chooses the relay with largest positive lookahead (i.e., earliest correlation).

Now consider multiple relays and different locations of the noise source. Figure 2.19 shows *MUTE*'s ability to correctly pick the wireless relay depending on the ambient speaker location in the room. We place the *MUTE* client at the center of the room, and three wireless relays around the edges and corners. We observe that when the ambient speaker is near the i-th relay, *MUTE* selects that relay consistently. We also observe that when the noise source is closer to the *MUTE* client location, no relay is selected because all of them offer negative lookahead.

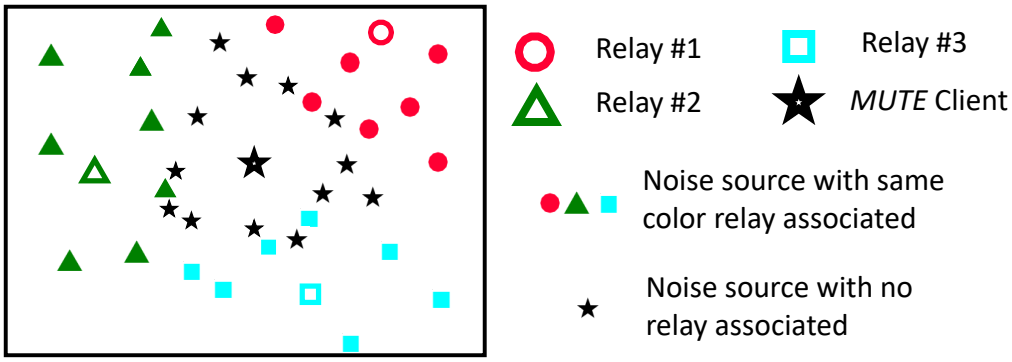


Figure 2.19: *MUTE* client associates with appropriate RF relays, depending on the location of the noise source.

2.6 Current Limitations

Needless to say, there is room for further work and improvement. We discuss a few points here.

- **Multiple Noise Sources:** Our experiments were performed in natural indoor environments, with a dominant noise source (such as a human talking on the phone, or music from an audio speaker). With multiple noise sources, the problem is involved, requiring either multiple microphones (one for each noise channel), or source separation algorithms that depend on statistical independence among sources. Today's ANC headphones utilize at least 6 microphones and source separation algorithms to mitigate such issues. We believe the benefits of looking ahead into future samples will be valuable for multiple sources as well – a topic we leave to future work.

- **Cancellation at the Human Ear:** We have aimed at achieving noise cancellation at the measurement microphone, under the assumption that the ear-drum is also located close to the error microphone. Bose, Sony, and other companies take a step further, i.e., they utilize anatomical ear models (e.g., KEMAR head [47]) and design for cancellation at the human ear-drum. Thus, Bose's performance may have been sub-optimal in our experiments. However, even without ear-model optimizations, our human experiments have returned positive feedback. Of course, a more accurate comparison with Bose would require *MUTE* to also adopt human ear-models, and then test with large number of human subjects. We have left this to future work. Finally, companies like *Nura* [12] are leveraging in-ear acoustic signals to build personalized ear models. Embracing such models is likely to benefit both *MUTE* and Bose.

- **Head Mobility:** We have side-stepped human head mobility, since our error microphone is static around the head model. Of course, head mobility will cause faster channel fluctuations, slowing down convergence. While this affects all ANC realizations (including Bose and Sony headphones), the issue has been alleviated by bringing enhanced filtering methods known to converge faster. We plan to also apply such mobility-aware LMS techniques in our future versions of *MUTE*.

- **Portability:** While Bose and Sony headphones are easily portable, *MUTE* requires the user to be around the IoT relay. While this may serve most static use cases (e.g., working at office, snoozing at the airport, sleeping at home, working out in the gym, etc.), headphones may be advantageous in completely mobile scenarios, like running on the road.

- **RF Interference and Channel Contention:** Our system will occupy the RF channel once the IoT relay starts streaming. However, it only occupies 8 kHz bandwidth, far smaller than the 26 MHz channel in the 900 MHz ISM band. Further, covering an area requires few relays (three for any horizontal noise source direction, and four for any 3D direction), hence, the total bandwidth occupied remains a small fraction. Even with multiple co-located users, channel contention can be addressed by carrier-sensing and channel allocation.

2.7 Related Work

The literature in acoustics and active noise control is extremely rich, with connections to various sub-fields of engineering [48, 39, 49, 50, 51, 52, 53, 54, 55, 56]. In the interest of space, we focus on two directions closest to *MUTE*: wireless ANC, and ANC with lookahead.

- **Wireless ANC:** An RF control plane has been proposed in the context of multi-processor ANC, mainly to cope with various sound sources in large spaces [57, 58, 59, 60, 61, 62]. In this body of work, distributed DSP processors communicate between themselves over wired/wireless links to achieve real-time, distributed, noise cancellation. The notion of "piggybacking" sound over RF, to exploit the propagation delay difference, is not a focus in these systems. Moreover, most of the mentioned systems are via simulations [57, 60, 61, 62].

- **ANC with Lookahead:** Certain car models [63, 64, 65] and airplanes [66, 67] implement ANC inside their cabins – reference microphones are placed near the engine and connected via wires to the DSP devices. While this offers promising lookahead, observe that the problems of inverse-channel estimation are almost absent, since the noise source positions are known, the noise signal is well structured, and the acoustic channel is stable. Moreover, these systems have no notion of at-ear feedback (from headphone microphones), since they are canceling broadly around the passenger's head locations. This is the reason why cancellation is feasible only at very low frequencies (< 100 Hz in Honda vehicles [64]). In contrast, *MUTE* introduces *wireless forwarding*, embeds lookahead-awareness in the ANC pipeline, and integrates a personal architecture for 4 kHz cancellation. Said differently, the intersection of "personal" ANC and "wireless" lookahead is both technically and architecturally new, to the best of our knowledge.

The idea of sound forwarding over RF has been applied to very different contexts, such as acoustic communication across sound-proof boundaries [68], wireless acoustic MIMO and beamforming [69], and even walkie-talkies and wireless microphones [70, 71]. However, sound forwarding has not been applied to noise cancellation. Finally, we should

mention that some systems have leveraged the propagation delay difference between RF and sound, albeit for other applications. *Cricket* [1], *AHLoS* [2], and *Dolphin* [3] have all used the time-of-arrival (ToA) difference between RF and sound for ranging and localization. Wang et al. [4] use RF signals as a tool to avoid acoustic collision in wireless sensor networks. Overall, this is similar to how earthquake and tsunami sensors [5, 6] work, by utilizing the fact that wireless signals travel much faster than ocean waves and tectonic vibrations.

2.8 Conclusion

This chapter exploits the velocity gap between RF and sound to improve active noise cancellation. By anticipating the sound milliseconds in advance, our proposed system is able to compute the anti-noise signal in time, better estimate sound channels, and ultimately attain wider-band cancellation. In addition, the core idea opens a number of architectural possibilities at the intersection of wireless networks and acoustic sensing. This chapter is a first step in these directions.

CHAPTER 3

VOLOC: SOURCE LOCALIZATION FROM VOICE SIGNALS

Voice assistants such as Amazon Alexa and Google Home use microphone arrays to estimate the angle of arrival (AoA) of the human voice. This chapter focuses on adding user localization as a new capability to voice assistants. For any voice command, we desire Alexa to be able to localize the user inside the home. The core challenge is two-fold: (1) accurately estimating the AoAs of multipath echoes without knowledge of the source signal, and (2) tracing back these AoAs to reverse triangulate the user's location.

We develop *VoLoc*, a system that proposes an *iterative align-and-cancel* algorithm for improved AoA estimation, followed by an *error-minimization* technique to estimate the geometry of a wall reflection. The AoAs and geometric parameters are then fused to reveal the user's location. Under modest assumptions, we report localization accuracy of 0.44 m across different rooms, clutter, and microphone locations. *VoLoc* runs in near real-time but needs to hear around 15 voice commands before becoming operational.

3.1 Introduction

Voice assistants such as Amazon Echo and Google Home continue to gain popularity with new "skills" being continuously added to them. A skill coming to Alexa is the ability to infer emotion and age from the user's voice commands [72, 73, 74]. More of such skills are expected to roll out, aimed at improving the contextual background of the human's voice command. For

instance, knowing a user's age may help in retrieving information from the web and personalizing human-machine conversations.

Towards enriching multiple dimensions of context-awareness, companies like Amazon, Google, and Samsung are also pursuing the problem of user localization [20, 21, 22, 23]. Location adds valuable context to the user's commands, allowing Alexa to resolve ambiguities. For instance, knowing the user's location could help in determining which light the user is referring to, when she says "turn on the light" (naming every IoT device, and precisely remembering their names, is quickly becoming a memory overload for the users [24, 25]). More broadly, location could aid speech recognition by narrowing down the set of possible commands [26, 27, 28]. For example, Google is working on "generating kitchen-specific speech recognition models", when its voice assistant detects "utterances made in or near kitchens" from the user [75].

These and other uses of location will emerge over time, and the corresponding privacy implications will also need attention. In this chapter, however, we focus on exploring the technical viability of the problem. To this end, let us begin by intuitively understanding the general problem space, followed by the underlying challenges and opportunities.

The central question in voice-source localization is that an unknown source signal must be localized from a single (and small) microphone array. Relaxing either one of the requirements brings up rich bodies of past work [76, 77, 78, 79, 80, 81, 82]. For instance, a known source signal can be localized through channel estimation and fingerprinting [83, 79, 84, 82], while scattered microphone arrays permit triangulation [78, 77, 80, 85]. However, *VoLoc*'s aim to localize arbitrary sound signals with a single device essentially inherits the worst of both worlds.

In surveying the space of solutions, we observe the following: (1) Signal strength based approaches that estimate some form of distance are fragile due to indoor multipath. Amplitude variations across microphones are also small due to the small size of Alexa. (2) Machine learning approaches to jointly infer the in-room reflections and per-user voice models seem extremely difficult, even if possible. Moreover, such training would impose a prohibitive burden on the users, making it unusable. (3) Perhaps a more viable idea is to leverage the rich body of work in *angle of arrival* (AoA). Briefly, AoA is the angular direction from which a signal arrives at a receiver. Voice assistants today already estimate the direct path's AoA and beamform towards the user [86, 87, 88, 89]. So one possibility is to detect additional AoAs for the multipath echoes and trace back the AoA directions to their point of intersection (via reverse triangulation).

Unfortunately, extracting AoAs for individual echoes, especially indoors, is difficult even in today's state-of-the-art algorithms [90, 91]. Even the direct path AoA is often erroneous/biased in today's systems, and small AoA offsets magnify localization error. Finally, tracing back the AoAs requires knowledge of reflectors in the room, a somewhat impractical proposition.

While the problem is non-trivial, application-specific opportunities exist:

- Perhaps not all AoAs are necessary; even two AoAs may suffice for reverse triangulation, so long as these AoAs are estimated with high accuracy. Of course, the reflector for the second AoA is still necessary.
- To connect to power outlets, Alexa is typically near a wall. If the AoA from the wall can be reliably discriminated from other echoes, and the wall's distance and orientation estimated from voice signals, then reverse triangulation may be feasible.
- Finally, the user's height can serve as an invariant, constraining the 3D

location search space.

All in all, these opportunities may give us adequate ammunition to approach the problem. Thus, the core algorithmic questions boil down to *accurate AoA detection* and *joint wall geometry estimation*. These two modules form the technical crux of *VoLoc* – we discuss our core intuitions next.

■ **Accurate AoAs:** Accurate AoA estimation is difficult in multipath settings because each AoA needs to be extracted from a mixture of AoAs, caused by echoes. Existing algorithms try to align (beamform) towards different directions to find energy maxima, but do not perform well because all the echoes are strongly correlated (elaborated in Section 3.2). We aim to break away from this approach, and our central idea is rooted in leveraging (1) slow velocity, and (2) *pauses* (or short silences) in acoustic signals. A voice command, for example, is preceded by silence. The ends of these silences are unique opportunities to observe the cascade of arriving signals, starting with the clean direct path first, followed by the first echo, second echo, and so on. This means that the direct path signal is essentially clean for a short time window, presenting an opportunity to accurately derive its AoA. Since the first echo is a delayed version of the direct path, this echo can be modeled and *cancelled* with appropriate alignment. This process can continue iteratively, and in principle, all AoAs and delays can be jointly extracted.

In practice, hardware noise becomes the limiting factor, hence cancellation errors accrue over time. Thus, *VoLoc* extracts accurate AoAs and delays for only the initial echoes and utilizes them for source localization.

■ **Wall Geometry Estimation:** Inferring location from AoA requires geometric knowledge of signal reflectors. The opportunity arises from the fact that the wall near Alexa serves as a stable echo, i.e., it is always present. If the wall's distance and orientation can be estimated with respect to Alexa,

then the echo's AoA and delay become a function of user location. This also helps in discriminating the wall echo from other echoes, say from objects on the table around Alexa. The algorithmic challenge lies in estimating the wall's ⟨distance, orientation⟩ tuple from the same voice signals.

We address this problem by gathering signals from recent voice commands and asking the following question: *At what distance d and orientation θ must a reflector be, such that its echo arrives early and is frequently present in voice command signals?* We formulate this as an optimization problem with the error function modeled in terms of $\langle d, \theta \rangle$. This error is summed across multiple recent voice commands, and the minimum error yields the $\langle d, \theta \rangle$ estimates. We over-determine the system by fusing AoA, $\langle d, \theta \rangle$, and user height h,[1] and converge to the user's indoor 2D location.

We implement *VoLoc* on an off-the-shelf hardware platform composed of a 6-microphone array, positioned in a circular shape like Amazon Echo (Figure 3.1). This was necessary to gain access to raw acoustic signals (commercially available Echo or Google platforms do not export the raw data). Our microphone array forwards the signal to a Raspberry Pi, which performs basic signal processing and outputs the data into a flash card, transmitted to our laptop over a WiFi direct interface. Experiment results span across AoA and location estimations in various environments, including student apartments, house kitchen, conference rooms, etc.

Our results reveal median localization accuracy of 0.44 m across a wide range of environments, including objects scattered around the microphone. In achieving this accuracy, the detected AoAs consistently outperform GCC-PHAT and MUSIC algorithms. *VoLoc* also estimates wall geometry (distance and orientation) with average accuracies of 1.2 cm and 1.4°, respectively. The results are robust across rooms, users, and microphone positions.

[1] We assume the heights are known for each family member.

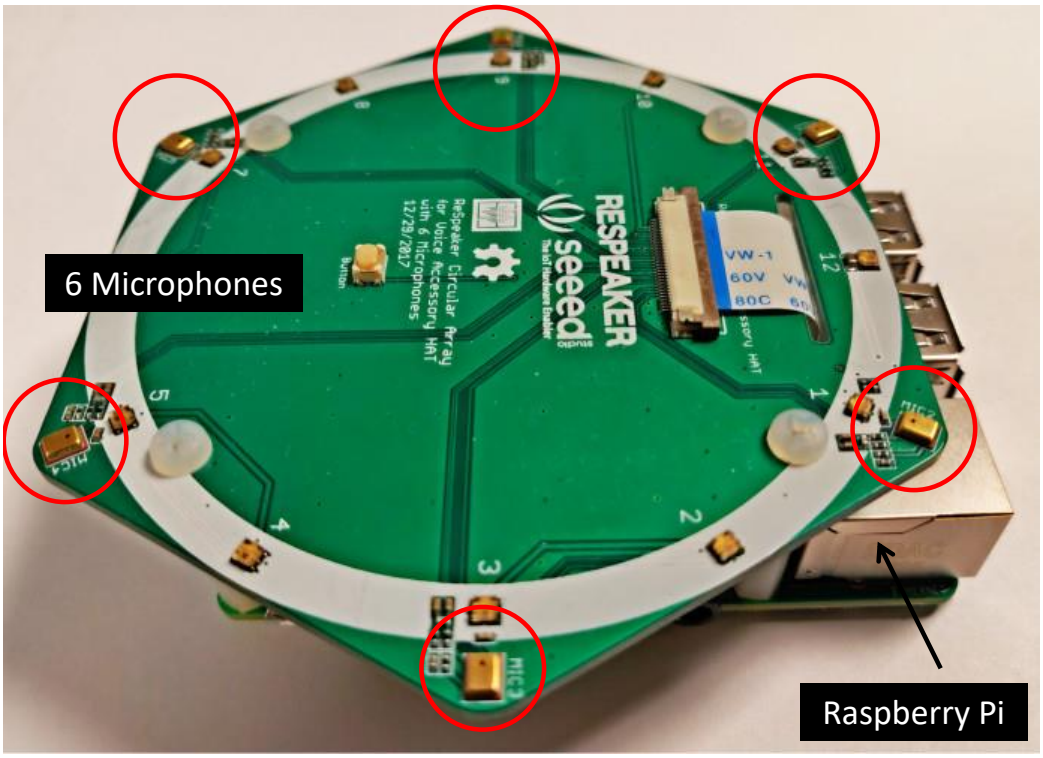


Figure 3.1: An off-the-shelf 6-microphone array, sitting on top of a Raspberry Pi.

In sum, we summarize the chapter's contributions as follows:

- A novel *iterative align-and-cancel algorithm* that jointly extracts initial AoAs and delays from sound pauses. The technique is generalizable to other applications.
- An *error minimization formulation* that jointly estimates the geometry of a nearby reflector using only the recent voice signals.
- A *computationally efficient fusion* of AoA, wall-reflection, and height to infer indoor 2D human locations.

We expand on each of these contributions next, starting with background on AoA.

3.2 Background and Formulation

This section presents relevant background for this chapter, centered around array processing, angle of arrival (AoA), and triangulation. The background will lead into the technical problems and assumptions in *VoLoc*.

3.2.1 Array Processing and AoA

Figure 3.2 shows a simple 3-element linear microphone array with d distance separation. Assuming no multipath, the source signal $s(t)$ will arrive at each microphone as $x_1(t)$, $x_2(t)$ and $x_3(t)$, after traveling a distance of D_1, D_2 and D_3, respectively. Usually $\{D1, D2, D3\} \gg d$, hence these sound waves arrive almost in parallel (known as the far field scenario). From geometry, if the signal's incoming angle is θ, then the signal wave needs to travel an extra distance of $\Delta d = d\cos(\theta)$ to arrive at microphone M_2 compared to M_1, and an extra $2\Delta d$ at M_3 compared to M_1.

When the additional travel distance is converted to phase, the phase difference between $x_2(t)$ and $x_1(t)$ is $\Delta\phi = 2\pi d\cos(\theta)/\lambda$, and between $x_3(t)$ and $x_1(t)$ is $2\Delta\phi$. On the other hand, the amplitudes of $x_1(t)$, $x_2(t)$ and $x_3(t)$ will be almost the same, due to very minute differences among D_1, D_2 and D_3.[2] Thus, in general, the received signal vector can be represented as:

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} x_1 \\ x_1 e^{j\Delta\phi} \\ \vdots \\ x_1 e^{j(n-1)\Delta\phi} \end{bmatrix} = \begin{bmatrix} e^{j0} \\ e^{j\Delta\phi} \\ \vdots \\ e^{j(n-1)\Delta\phi} \end{bmatrix} x_1$$

■ **AoA Estimation without Multipath:** In reality, we do not know the signal's incoming angle θ, hence we perform AoA estimation. One solu-

[2]Sound amplitude attenuates with $1/r$ where r is traveled distance. For two paths of r and $r+\Delta d$, the relative amplitude difference is $[1/r - 1/(r+\Delta d)]/(1/r) \approx \Delta d/r$. If Δd=2 cm and r=2 m, there would be a 1% amplitude difference.

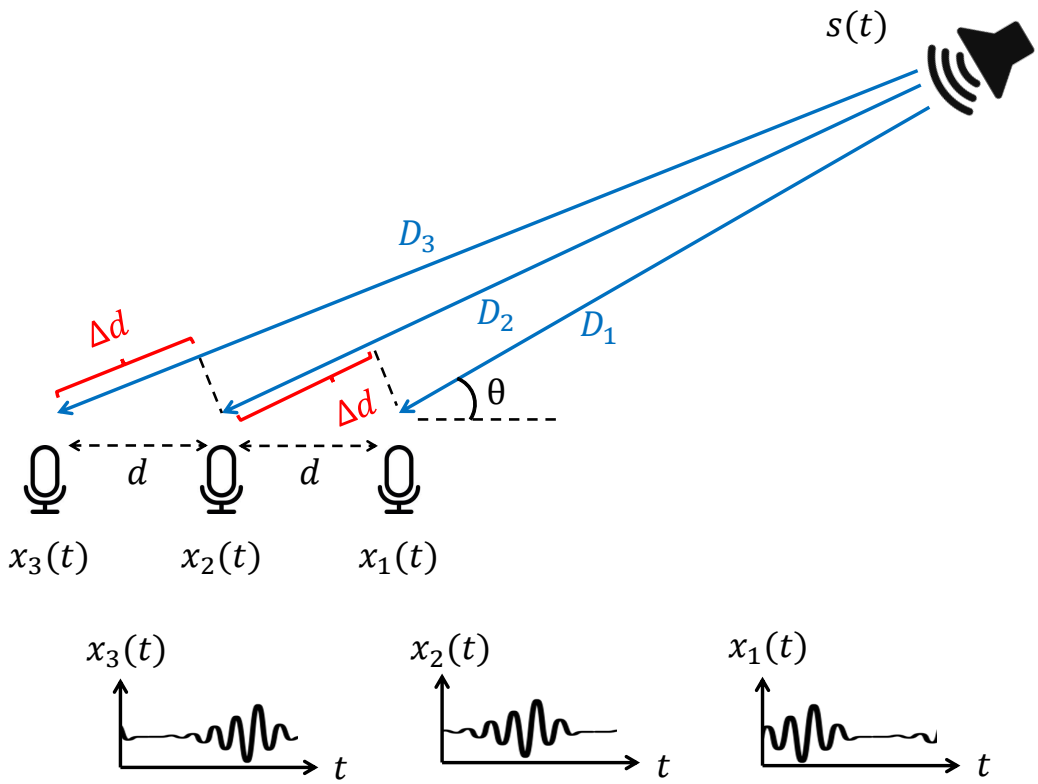


Figure 3.2: A simple 3-element microphone array.

tion is to consider every possible θ, compute the corresponding $\Delta\phi$, apply the appropriate negative phase shifts to each microphone, and add them up to see the signal energy. The correct θ should present a maximum energy because the signals will be perfectly aligned, while others would be relatively weak. This AoA technique essentially has the effect of steering the array towards different directions of arrival, computing an AoA energy spectrum, and searching for the maximum peak. For a single sound source under no multipath, this reports the correct AoA direction.

■ **Impact of Multipath Echoes:** Now consider the source signal $s(t)$ reflecting on different surfaces and arriving at different delays from different directions. Each arriving direction is from a value of θ_i, translating to a corresponding phase difference $\Delta\phi_i$. Thus the received signal at each microphone (with respect to microphone M_1) is a sum of the same source signal,

delayed by different phases. With k echoes, we can represent the received signal as:

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} e^{j0} & e^{j0} & & e^{j0} \\ e^{j\Delta\phi_1} & e^{j\Delta\phi_2} & \dots & e^{j\Delta\phi_k} \\ \vdots & \vdots & & \vdots \\ e^{j(n-1)\Delta\phi_1} & e^{j(n-1)\Delta\phi_2} & \dots & e^{j(n-1)\Delta\phi_k} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_k \end{bmatrix}$$

■ **Estimating AoA under Multipath:** The earlier AoA technique (of searching and aligning across all possible θ_i) is no longer accurate since phase compensation for an incorrect θ_i may also exhibit strong energy in the AoA spectrum (due to many strongly correlated paths). Said differently, searching on θ_i is fundamentally a cross-correlation technique that degrades with lower SNR. Since any path's SNR reduces with increasing echoes, AoA estimation is unreliable.

While many AoA-variants have been proposed [92, 93, 94, 44, 95, 96, 97, 91, 98, 99, 100], most still rely on cross-correlation. The most popular is perhaps GCC-PHAT [94, 44, 95, 101, 96] which compensates for the amplitude variations across different frequencies by whitening the signal. The improvement is distinct but does not solve the root problem of inaccurate alignment. Subspace based algorithms (like MUSIC and ESPRIT [97, 98, 102, 99, 100]) are also used, but they rely on the assumption that signal paths are uncorrelated. Multipath echoes exhibit strong correlation, so AoA estimation remains a difficult problem.

3.2.2 Reverse Triangulation

Even if AoAs are estimated correctly, localization would require knowledge of reflectors in the environment to reverse triangulate (Figure 3.3). While some past work has scanned the environment with depth cameras to create 3D

room models [103], this approach is largely impractical for real-world users.

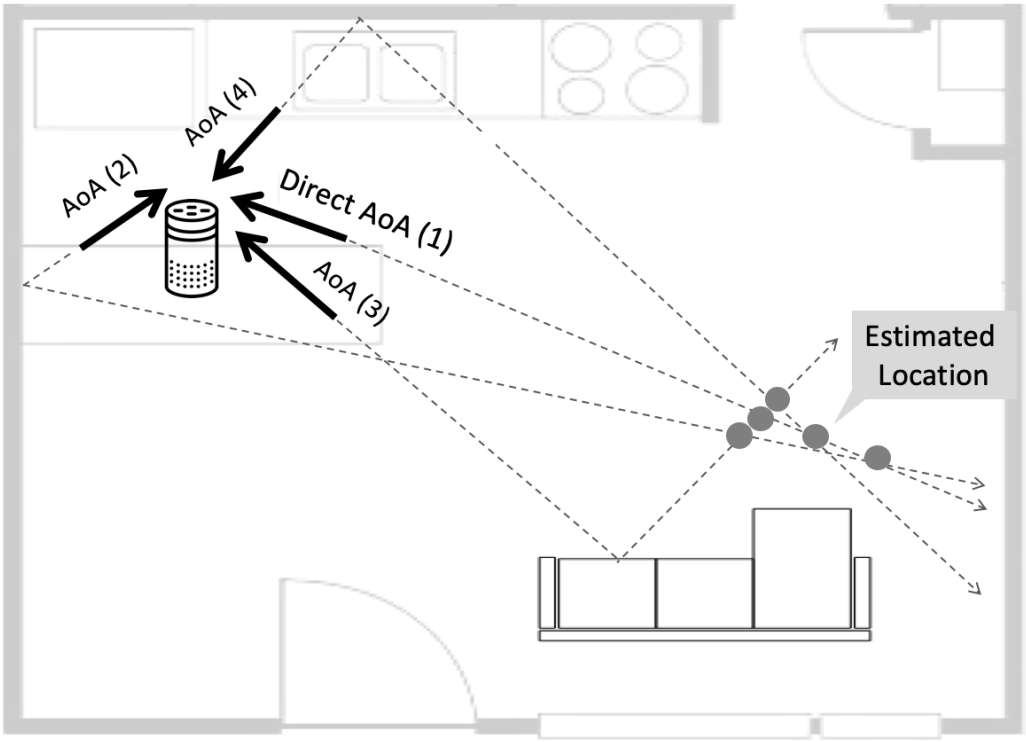


Figure 3.3: Reverse triangulation requires location of all reflector surfaces, making it impractical.

In principle, however, not all echoes are necessary for tracing back to the source. The direct path's AoA and one other AoA would be adequate: say, AoA(1) and AoA(2) in Figure 3.3. Of course, the location and orientation of AoA(2)'s reflector still needs to be inferred. The authors of [76, 83] have attempted a related problem. They attempt to infer the shape of an empty room; however, they use precisely designed wideband signals, scattered microphone arrays, and essentially solve compute-intensive inverse problems [76, 83]. *VoLoc* takes on the simpler task of estimating one reflector position, but under the more challenging constraint of unknown voice signals and near real-time deadlines (a few seconds[3]).

[3]The time granularity of a human voice command.

3.2.3 Problem Statement and Assumptions

With this background, the problem in this chapter can be stated as follows. Using a 6-microphone array, without any knowledge of the source signal, and under the assumptions

- Alexa located near a wall to connect to a power outlet
- User's height known

VoLoc needs to

- Precisely estimate AoA for two signal paths in multipath-rich environments.
- Estimate the distance and orientation of at least one reflector, and identify the corresponding AoA for reverse triangulation.
- Fuse the AoAs, reflector, and height to geometrically infer the user's indoor location.

The solution needs to be performed without any voice training, must complete in the order of seconds, and must handle clutter (such as various objects scattered on the same table as Alexa).

3.3 System Architecture

Figure 3.4 illustrates *VoLoc*'s overall architecture. When the user speaks a voice command, the *IAC (Iterative Align-and-Cancel) AoA* module takes the raw acoustic samples, identifies the "pause" moment, and extracts a few initial AoAs from the following signal. To translate AoAs into location, the *Fusion* module takes two initial AoAs and fuses them with three parameters: the distance and orientation $\langle d, \theta \rangle$ of the nearby wall reflector, and the user's height, h. Together, the AoA and geometric information over-determine the

user's 2D location for robustness to errors. The two parameters are separately estimated by the *Joint Geometric Parameter Estimation* module, by using the ensemble of *IAC*-estimated AoAs from recently received voice commands. This is a one-time estimation during initialization, meaning *VoLoc* is operational within the first $n = 15$ voice commands.

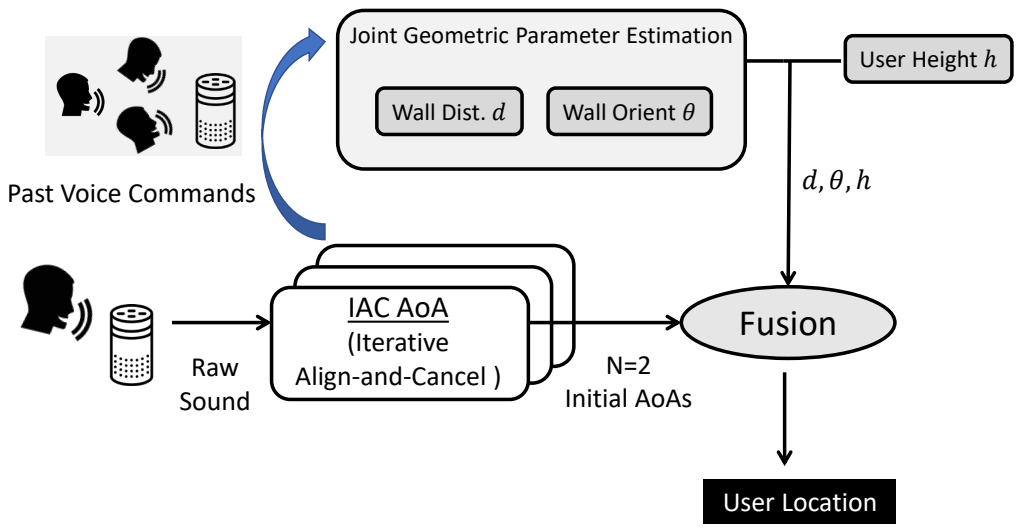


Figure 3.4: *VoLoc* system overview. When a user speaks a voice command, *IAC AoA* computes two initial AoAs. The direct path's AoA, when combined with height of the user, is ready to produce a basic 2D user location. To improve the estimate, the *Fusion* module fuses the two AoAs, the closest wall reflector, and the height information together to geometrically refine the location. The *Joint Parameter Estimation* module aims at computing the wall's relative distance and orientation by analyzing recent voice commands from the user.

We begin this section by describing our IAC (Iterative Align-and-Cancel) AoA algorithm, a general AoA technique that also applies to other applications.

3.3.1 IAC (Iterative Align-and-Cancel) AoA

The goal of the IAC AoA algorithm is to extract both angles of arrival and corresponding delays of a few early paths in the multipath environment. This

is very different from existing AoA algorithms which perform only *alignment* to find AoAs; we perform both *alignment and cancellation*, starting with a clean signal at the initial pause moment.

■ A Glance at the Initial Moment

Figure 3.5 zooms into the scenario when the voice signal is beginning to arrive at the microphones. The user starts to speak a sequence of voice samples, denoted as $x(t) =$ "ABCDE...". The signal travels along the direct (red) path, and arrives at the microphone array as early as time t_1. Note that due to the microphone arrangement in the array, mic $\#1, \#2, \cdots$ hear the first sample "A" at slightly different times: $t_1^{(1)}, t_1^{(2)}, \cdots$. These slight differences capture the AoA of the direct path.

With ensuing time, the same voice signal also arrives along the second (blue) path, known as the first echo. Since this second (blue) path is longer, the signal arrives at the microphone array at a later time, t_2, denoted as "abcdefg...". As a result, between t_1 and t_2, all the microphones hear clean, unpolluted direct path signal (tens of samples). Similarly, if t_3 is the time the third path arrives, then for $t \in [t_2, t_3]$, the microphones receive the signal from only the first two paths.

■ Detecting AoA_1 for the Direct Path

Recall that signals in time window $t \in [t_1, t_2]$ contain only the direct path signal, and its angle of arrival (denoted as AoA_1) is captured in the slight time offset across microphones: $t_1^{(1)}, t_1^{(2)}, \cdots$. To infer AoA_1, we first detect t_1 from the rise of signal energy, and select a small time window $[t_1, t_1 + \Delta]$ of signals after that. Then, we ask the following question: Given this window of data, among all possible AoAs, which AoA_1 best aligns with the actual time offsets across the three microphones?

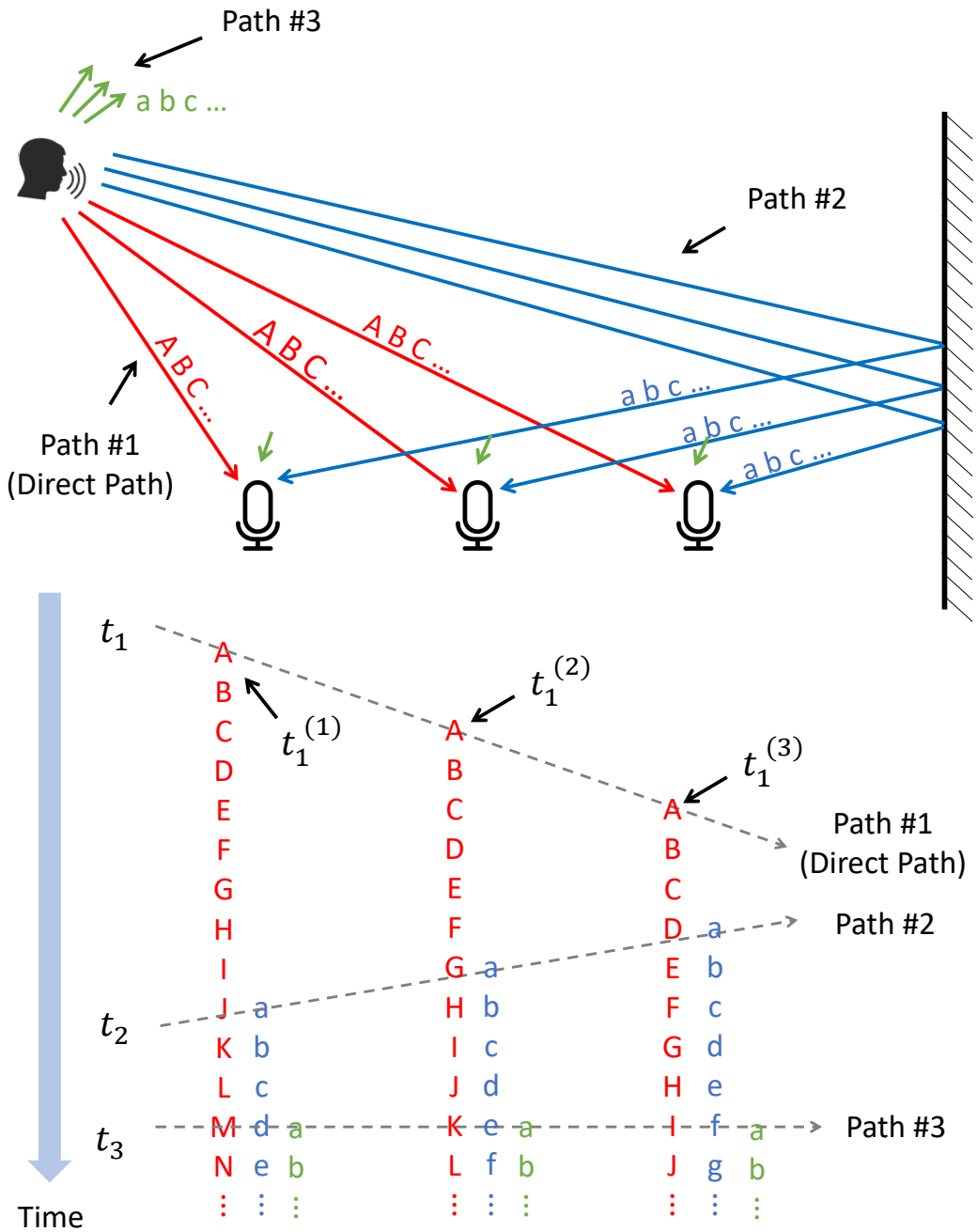


Figure 3.5: In a multipath environment, the voice signal travels along different paths and arrives at the microphone at different times.

We solve this problem by performing a one-step "align-and-cancel". Figure 3.6(a) shows the key idea. Assume we have found the correct AoA_1; then, for any given pair of microphones, we can align their signals based on this AoA, in order to "reverse" the offset effect. This alignment is done by simply applying a cross-delay, i.e., delaying microphone i's signal with j's delay, and j's signal with i's delay.[4] The aligned signals are now subtracted, meaning they fully cancel each other with zero cancellation error. Any cancellation residue quantifies the error in the alignment, which further indicates the error in AoA estimation. After searching across all possible AoAs, we choose the one which minimizes the sum of cancellation errors across all microphone pairs.

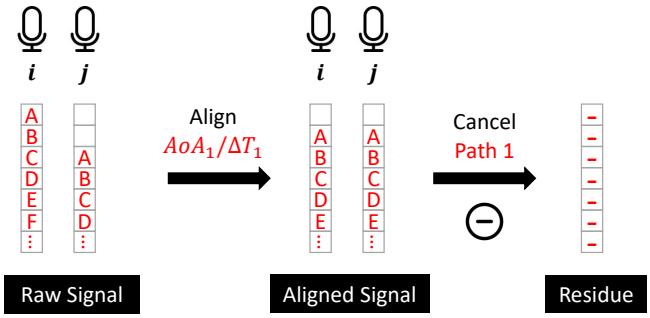
■ Detecting AoA_2 for the Second Path

Once we have found the right AoA_1, the align-and-cancel operation should maintain low error over time, until when the second path arrives at time t_2. Thus, once we observe growing error, it is time to estimate the second path's angle of arrival, AoA_2.

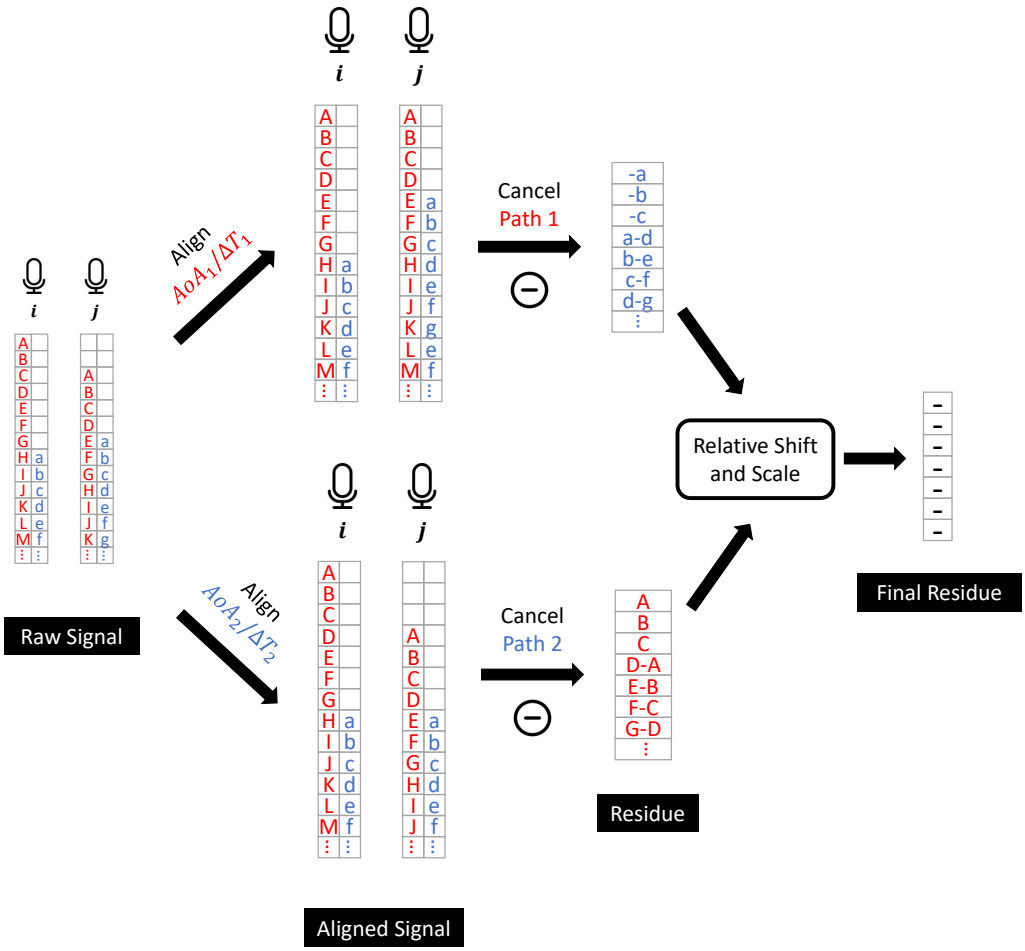
For this, we will again perform align-and-cancel for both AoA_1 and AoA_2, as shown in Figure 3.6(b). However, since the microphones are now receiving a mixture of two paths, we can align only one path at a time, meaning only one path gets canceled. In other words, after aligning the AoA_1 path and canceling the signals, the residue will be the difference between the "unaligned" second paths, and the vice versa. The middle column in Figure 3.6(b) shows both the alignments.

Fortunately, as shown in the third column of Figure 3.6(b), *the two cancellation residues are identical, except for a scaling factor* caused by the

[4]Its easy to understand this by imagining the microphones' delays as two numbers x and y. To align them, we just need to add x to y and y to x, making both microphone's delays $(x + y)$.



(a) Solving AoA_1 for the first path. After aligning with the correct AoA, the aligned signals will cancel each other.



(b) Solving AoA_2 for the second path. The residual signals after "align and cancel" can further cancel each other by aligning the relative shift and scale.

Figure 3.6: The idea of iterative delay-and-cancellation (IAC) algorithm, shown for $K = 1$ and $K = 2$.

amplitude difference between two paths. This similarity is because both residues are essentially the "unaligned" path signal minus a delayed copy of the same "unaligned" signal, and that delay (which is the delay caused by AoA1 plus the delay caused by AoA2) is the same for both alignments. A linear combination of the two residues will be zero, and the coefficients are exactly the amplitudes of each path.

Based on the observation above, we solve for the second path's AoA by doing the following: We search across all possible AoAs for path #2, and for each AoA candidate, we perform the operation in Figure 3.6(b), and run least squares (LS) over the two residues. The LS solution gives the estimated linear coefficients (which are the amplitudes), and the *fitting error* of LS indicates the cancellation error after alignment. We pick the candidate which has the smallest sum of fitting errors.

One point worth noting is that AoA only captures relative time offsets among microphones. To fully cancel the two residues, we also need to make sure the two cancellation residues (first path residue, and second path residue) are aligned in absolute time scale. This means the absolute delay t_2 has to be accurate as well. Since our t_2 detection may not be precise, we also jointly search for different t_2's around the time when the first path's cancellation error starts growing. We jointly pick the t_2 and AoA_2 that minimize the fitting error.

■ Detecting More AoAs

The same idea applies to getting more AoAs. If the signal contains K paths, and we have obtained the AoAs (and absolute delays) for the first $(K-1)$ paths, then we can search for the AoA (and absolute delay) of the K-th path by following operations similar to those in Figure 3.6(b). Theorem 3.3.1 states that when AoA_k is estimated correctly, the K cancellation residues

are linearly dependent, and a linear combination of them (with coefficients as each path's amplitude) will be a zero vector.

Theorem 3.3.1 (IAC AoA Decoding)**.** *For a given pair of microphones, the k residue vectors from aligning and canceling each of the k AoAs are linearly dependent.*

Proof. Denote the signal "ABCDEFG..." as $x[t]$, and the signal arriving along the k-th path at the i-th microphone as $x[t-t_{k,i}]$. Then, the total signal received by the i-th microphone can be written as $y_i[t] = \sum_{k=1}^{K} x[t - t_{k,i}]$. When we align the k'-th path's AoA, the aligned signals are $y_1[t - t_{k',2}]$ and $y_2[t - t_{k',1}]$, respectively. The cancellation residue is:

$$\begin{aligned}
&y_1[t - t_{k',2}] - y_2[t - t_{k',1}] \\
&= \sum_{k=1}^{K} x[t - t_{k,1} - t_{k',2}] - \sum_{k=1}^{K} x[t - t_{k,2} - t_{k',1}]
\end{aligned}$$

The sum of all the cancellation residues is:

$$\begin{aligned}
&\sum_{k'=1}^{K} \big(y_1[t - t_{k',2}] - y_2[t - t_{k',1}]\big) \\
&= \sum_{k'=1}^{K}\sum_{k=1}^{K} x[t - t_{k,1} - t_{k',2}] - \sum_{k'=1}^{K}\sum_{k=1}^{K} x[t - t_{k,2} - t_{k',1}] \\
&= \sum_{k=1}^{K}\sum_{k'=1}^{K} x[t - t_{k',1} - t_{k,2}] - \sum_{k'=1}^{K}\sum_{k=1}^{K} x[t - t_{k,2} - t_{k',1}] \\
&= 0
\end{aligned}$$

This proves that the sum of all cancellation residues is 0. Of course, we have ignored the amplitude of each path here, but it is easy to prove that amplitude is exactly the linear coefficient that makes the sum of these cancellation vectors zero. □

Explained differently, observe that we obtain K residues after aligning-and-canceling each AoA path. While the residues are identical for $K =$

2 (except for amplitude), in general, the K residues are not identical but linearly dependent. Thus, the best AoA_k can be found by minimizing the least squares fitting error. Algorithm 2 shows in detail how we compute this error.

Algorithm 2 For a Given Set of K AoAs and Absolute Delays, Compute the Overall Cancellation Error $\mathcal{E}$

```
1: ℰ = 0
2: for all pairs of microphones do
3:     ResidueList = {}
4:     for k = 1 ··· K do
5:         Align the two signals using the k-th AoA
6:         Compute the difference of two aligned signals as the cancellation
           residue
7:         Delay the residue using the k-th absolute delay
8:         ResidueList.Add(residue)
9:     end for
10:    Run least squares on ResidueList to compute the best linear combi-
       nation, and get its fitting error e
11:    ℰ += e
12: end for
```

■ Can We Detect Infinite AoAs?

In practice, the number of AoAs we could obtain is limited for two reasons: (1) In multipath environments, the first few paths are sparse while the latter ones are dense. This means the time window $[t_k, t_{k+1}]$ will be very short as k grows larger, making it hard to find the k-th path's AoA without being influenced by the $(k+1)$-th path. Said differently, there is no strict time of arrival of a given echo, hence, shorter gaps between arriving echoes make them difficult to separate. (2) Voice energy ramps up slowly due to the way humans produce sound. This means the latter echoes of the early samples are considerably weaker than the direct path samples. Background noise adds

to this, further lowering the SNR of the latter echoes. This is why *VoLoc* conservatively uses only the first $N = 2$ AoAs.

■ **Simulation Results**

To compare IAC's AoA estimation accuracy with other AoA algorithms under different indoor reverberation and SNR conditions, as well as to obtain ground truth for higher-order AoAs, we run a simulation with the "Alexa" voice as the source signal, added with varying levels of echoes and noise. The simulation uses the image source model [104] to simulate room impulse responses. We compare with three AoA techniques discussed in Section 3.2: (1) delay-and-sum, (2) MUSIC, and (3) GCC-PHAT.

Figure 3.7 shows the accuracy performance of these four algorithms. In general, we observe that GCC-PHAT is robust to reverberation and can get the first path correctly, but the correlation will fail at higher order paths. MUSIC and delay-and-sum do not work well in indoor reverberated environments where the acoustic signals are highly correlated. IAC, in contrast, actively takes advantage of correlated signals to jointly estimate each path's AoA and delay, leading to improved performance in rooms. This is the reason why our algorithm is, we believe, a new contribution to the body of AoA algorithms.

3.3.2 User Localization via Fusion

The above estimated AoAs can be reverse-triangulated to the user's location when we already know where the nearby wall reflector is, i.e., its distance d and orientation θ with respect to Alexa. Moreover, the human height (h) constrains the location search space to 2D. Pretending we already know $\langle d, \theta, h \rangle$, we design an optimization technique to efficiently fuse all three parameters to infer user location. In the next section, we will discuss how

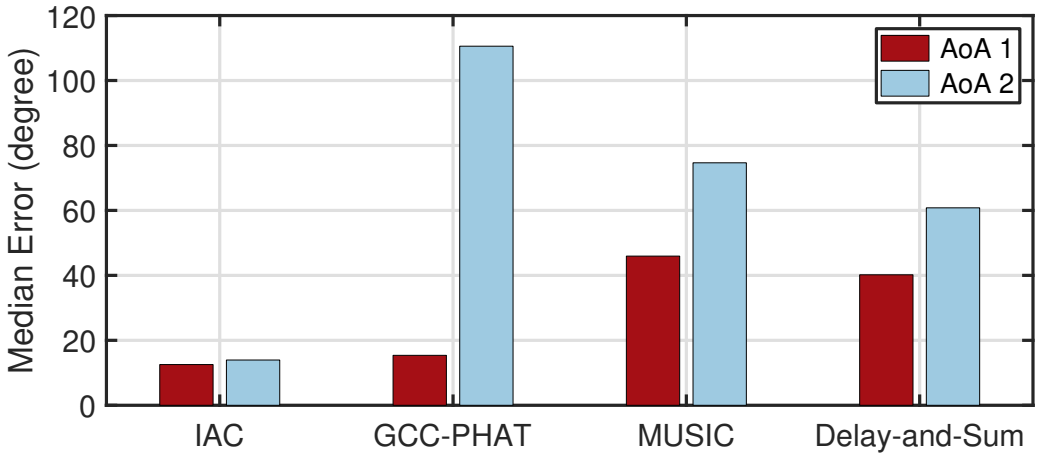


Figure 3.7: Accuracy comparison of four AoA techniques: IAC, GCC-PHAT, MUSIC, and Delay-and-Sum.

we jointly infer the $\langle d, \theta \rangle$ parameters from recent voice signals.

In ideal settings, the two AoAs and the wall's $\langle d, \theta \rangle$ are enough to analytically solve for the source location. In real environments, all the AoA and geometry estimates incur error, so over-determining the system with echo delay and human height h is valuable. In fusing all these and solving for user location, we make the following observations and decisions:

(1) First, not all AoAs/delays are feasible as the user is only moving in 2D with a fixed height. Therefore, searching for user location in this 2D plane will be more efficient (than searching for all AoAs and delays).

(2) Second, the direct path AoA from IAC, especially its azimuth, is accurate. This further reduces the search space to a beam in the 2D plane, as shown in Figure 3.8.

(3) Finally, for each possible location on this 2D beam, we can directly obtain the parameters for the direct path and wall reflection path using geometry (recall, we pretended to know all three parameters). This means we can directly compute the cancellation error using Algorithm 2 (using $K = 2$ echoes). The best location is determined by having the

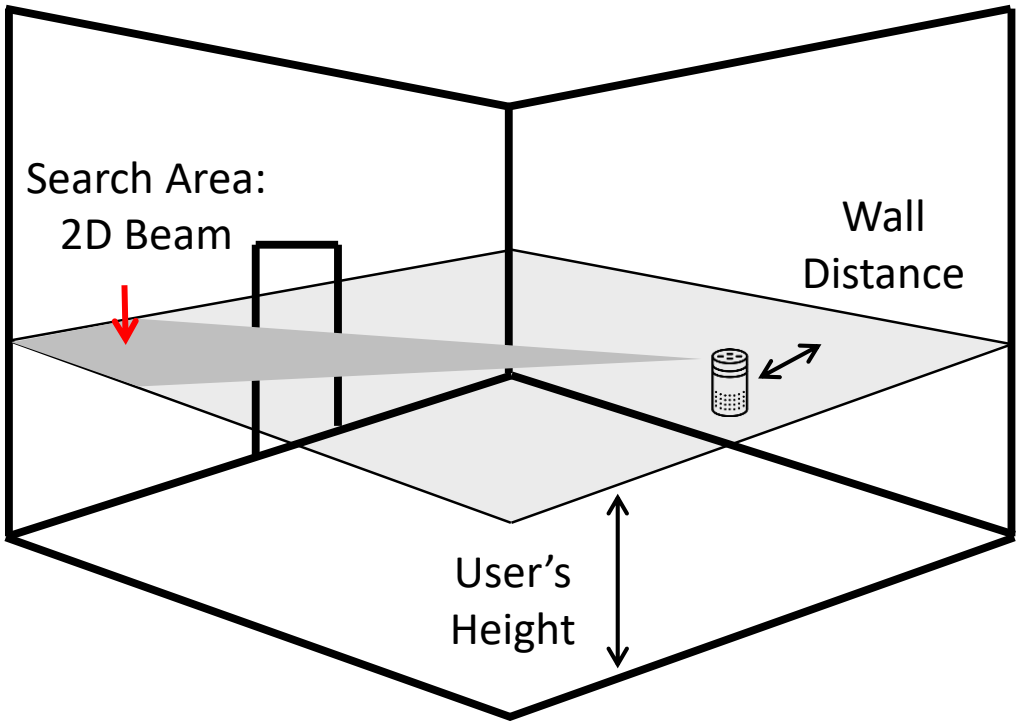


Figure 3.8: The reduced search space.

minimum cancellation error. Algorithm 3 summarizes our searching procedure.

3.3.3 Joint Wall Parameter Estimation

Finally, we describe our solution to estimate the two parameters (d, θ) from an ensemble of recent voice samples. Our core idea is the following. For each (past) voice command, we utilize the direct path AoA as a trustworthy estimate. We shortlist locations within a 3D cone around this AoA that satisfies our known height h. Now for each of these locations, and pretending the wall is d_i, θ_j from Alexa, we compute the corresponding wall AoA and delay. If $\langle d_i, \theta_j \rangle$ are the correct estimates, then the computed AoA and delay will align well with the measured signal, minimizing the cancellation error in Algorithm 2.

Of course, in the presence of other echoes from clutters around Alexa, the

Algorithm 3 Search for the Most Likely User Location in the Room

```
Run IAC to first obtain direct path's azimuth, azi
minError = +∞,   bestLoc = [ ]
for all Location loc on 2D plane do
    if loc's azimuth not close to azi then
        continue
    end if
    Compute AoA and absolute delay for both direct path and wall reflection path, using geometry
    Compute cancellation error ℰ using Algorithm 2
    if ℰ < minError then
        minError = ℰ,   bestLoc = loc
    end if
end for
Declare bestLoc as user location
```

wall echo may not match best, hence $\langle d_i, \theta_j \rangle$ may produce a higher (than minimum) cancellation error. However, when this operation is performed over multiple recent voice commands, and the cancellation errors summed up, we expect the optimal $\langle d_i^*, \theta_j^* \rangle$ to minimize this sum. The intuition is that different voice commands from different locations would consistently reflect from the wall, while reflections from other objects would come and go.[5] As a result, the correct values of $\langle d_i^*, \theta_j^* \rangle$ would eventually "stand out" over time.

Figure 3.9 shows one example of how the objective function (i.e., sum of cancellation errors) varies across the joint $\langle d, \theta \rangle$ variation. The X and Y axes of the graph are d and θ offsets from the ground truth, meaning the contour should minimize at $[0, 0]$. We search with a granularity of 2 cm and 1°, and the minimum point of the contour proves to be at $X = 2$ cm and $Y = 1°$.

[5]The table reflection may also be consistent; however, Alexa/Google microphones are designed with low gain towards the downward direction, and hence the energy of table reflection is weak.

This is promising and we evaluate this more in Section 3.4.

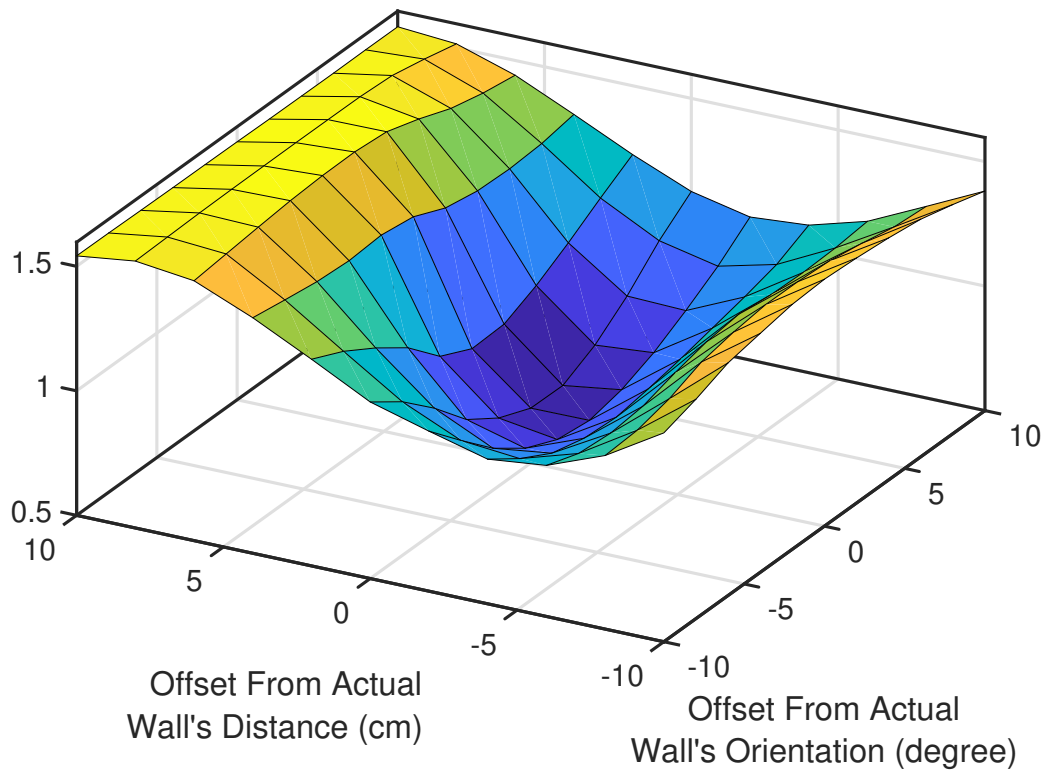


Figure 3.9: The sum of cancellation error minimizes at the nearly the true wall distance and orientation.

While joint parameter estimation is time consuming (in hours), we need to run this only during initialization. Once the estimates are ready, the fusion module uses these parameters and executes in a very short time.

3.3.4 Points of Discussion

- **Will echoes from furniture/environment affect the estimation of wall geometry?**

  Observe that the echo from the nearby wall is also close in time to the direct path. In fact, the echo's delay can be computed since $\langle d^*, \theta^*, h \rangle$ are all known. Because of this, echoes that bounce off farther away reflectors can be discounted, since all their delays arrive long after the

wall-echo. Confusion arises from reflectors that are closer to Alexa than the wall – like objects on the same table as Alexa. These unmodeled echoes prevent the cancellation errors from dropping sharply. Nonetheless, as we see in our evaluation, the error reduction is still a minimum for the correct user location. This is the reason why *VoLoc* is able to operate even in reasonably cluttered environments.

- **What happens when the wall echo is blocked by an object on the table?**

  This is the case where *VoLoc* will perform poorly, since the cancellation will be poor for the expected wall AoA. It may be possible to recognize the problem from the value of the cancellation error, such that we can gain some measure of confidence on the localization result. We have empirically observed increased cancellation errors; however, it is not clear how to develop a systematic confidence score from it (note that global thresholds or distributions would not scale well). Hence, we leave the design of a confidence metric to future work.

3.4 Implementation, Evaluation

This section discusses the experiment methodology and performance results of *VoLoc*.

3.4.1 Implementation

VoLoc is implemented on a Seeed Studio 6-microphone array [105], arranged in a circular shape similar to Amazon Echo. This is due to the lack of raw acoustic samples from commercial voice assistants. The acoustic signals on the array are sampled at 16 kHz, a sampling rate that covers most of the energy in voice frequencies. The array is connected to a Raspberry Pi to

forward its sound samples to a laptop over wireless. The laptop executes code written in MATLAB to compute user location, which takes $6 - 10$ seconds to finish.

3.4.2 Methodology

Our experiments were executed in four different indoor environments: (1) a studio apartment, (2) a kitchen, (3) a student office, and (4) a large conference room. The first two in-home places both have an Amazon Echo pre-installed, so we directly replace it with our microphone array. For the office and the conference room, we simply put the microphone array on a desk that is close to a power outlet. The distance to the wall ranges between 0.2 m and 0.8 m.

We recruited three student volunteers to speak different voice commands to the microphone array. Volunteers were asked to stand at marked positions, whose 2D locations (X, Y) have been measured beforehand (for ground truth) using a laser distance measurer. The voice commands start with either "Alexa, ..." or "Okay Google, ...", and are repeated five times at each location. We collected a total number of 2350 voice commands. Meanwhile, for in-home environments, we also recorded some other non-voice sounds and played them at different locations using a portable Bluetooth speaker. These sounds include the sound of cooking, the microwaves dings, or random sound clips from TVs. The goal is to test whether *VoLoc* has the potential to localize such arbitrary sounds from everyday objects.

3.4.3 Performance Results

The following questions are of interest:

(1) How well can *VoLoc* compute user locations in general? What is the break-up of gain from AoA and wall-estimation?

(2) How does *VoLoc*'s performance vary among different sounds (including non-voice sounds), varying clutter level, and varying ranges (near, medium, far)?

(3) How many recent voice samples are necessary for *VoLoc* to converge on the geometric parameters (d, θ)?

■ **Overall Localization Accuracy**

Figure 3.10 shows the CDF of *VoLoc*'s overall localization errors across all experiments, as well as the CDF of errors in each room. Overall, the median error is 0.44 m. We believe this accuracy makes it amenable to location-aware applications for in-home voice assistants like Alexa and Google Home.

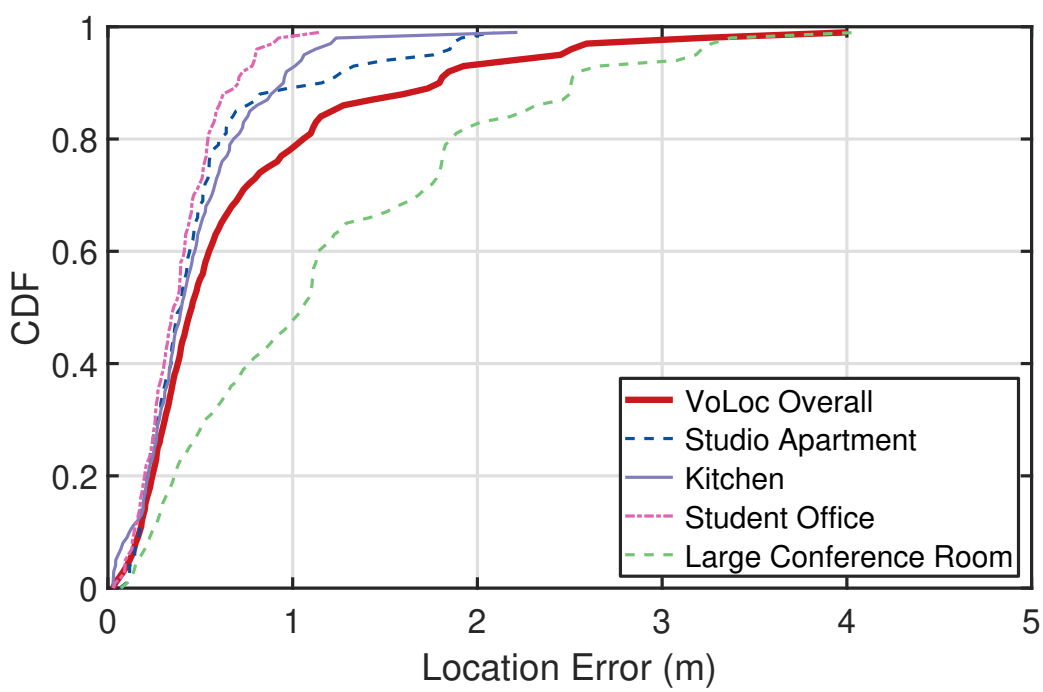


Figure 3.10: CDF of *VoLoc*'s overall localization accuracy, and the accuracy across different rooms.

Upon comparing the performance across the four rooms, we find that the conference room incurs significantly higher errors than the other three. Anal-

ysis shows that the conference room is large in size, meaning the user often stands far from Alexa, leading to increased location error. Said differently, far field errors are higher in triangulation algorithms because same angular error (in AoA, d, or θ) translates to larger location error.

Figure 3.11 compares *VoLoc*'s localization accuracy with the following two schemes: (1) *VoLoc*++, which assumes the two geometric parameters (wall's distance d and orientation θ) are perfectly known. Therefore, *VoLoc*++ will be a performance upper bound of *VoLoc*. (2) GCC-PHAT, which combines GCC-PHAT's direct path AoA with human height information (h) to compute human location. We choose GCC-PHAT as the baseline because it performs the best in Section 3.3.1.

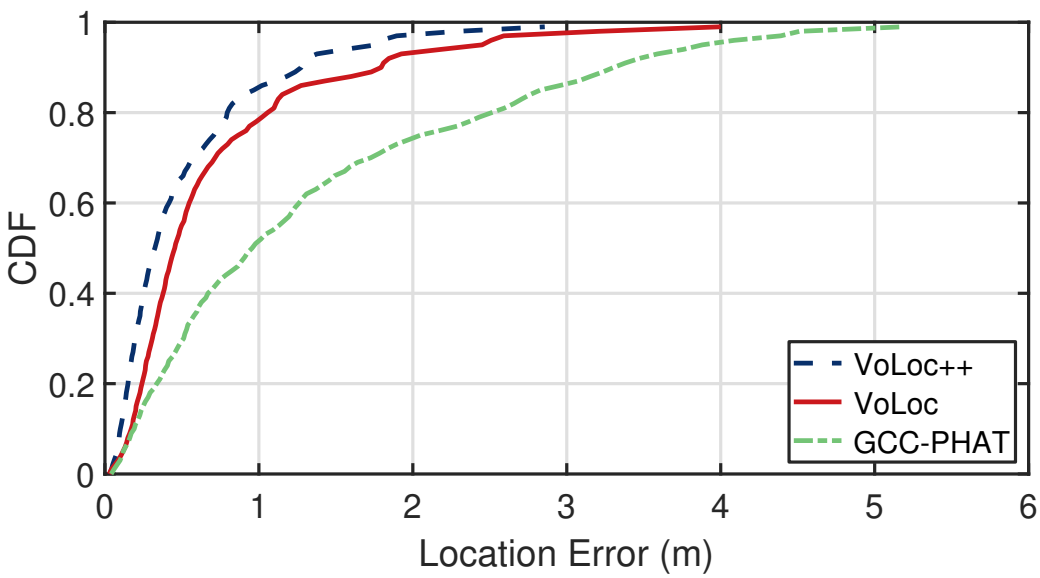


Figure 3.11: Performance comparison of *VoLoc*++, *VoLoc*, and GCC-PHAT.

Compared to GCC-PHAT's median location error of 0.93 m, *VoLoc*'s median error reduces by 52%. This demonstrates the value of precise 2-AoA estimation from our IAC algorithm. *VoLoc*++ further reduces the median error from 0.44 m to 0.31 m, assuming the geometric parameters are precisely known. This captures *VoLoc*'s efficacy to estimate the wall parameters

– there is a small room for improvement.

■ Accuracy Across Different Sounds

Figure 3.12 shows *VoLoc*'s median localization error across various kinds of sounds for in-home environments. The first two sounds are human voice commands, while the latter four are natural sounds from objects, such as microwave bell sound or music from TV. In general, we observe that localizing objects' sounds is easier than localizing the human voice. This is because most sounds made by objects have good energy ramp-up; i.e., unlike human voice, the energy of the sound quickly goes up within a very short time window. This means the SNR of the signal is strong for IAC to estimate AoA, leading to improved location results.

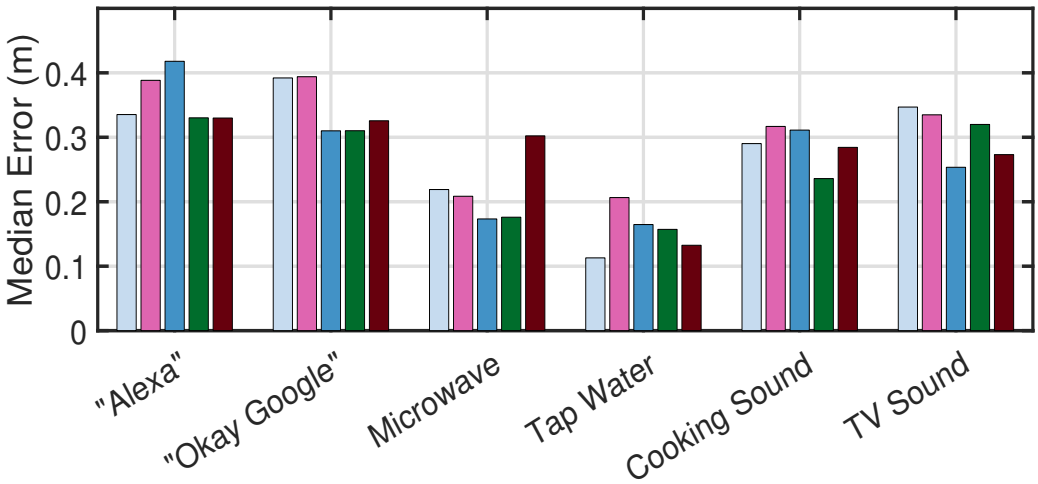


Figure 3.12: *VoLoc*'s localization accuracy across different kinds of sounds. Each cluster of bars represents one sound, and each bar within one cluster represents the median error across locations during one session.

■ Accuracy Over Distances to Alexa

VoLoc's localization accuracy will naturally go down as the user moves away from the microphone array. This is essentially because the resolution of AoA estimation limits the range of the user, i.e., a large movement at a

far away distance may only translate to a slight change in AoA. Figure 3.13 visualizes *VoLoc*'s localization error in the conference room. The microphone array is placed on a table towards the northeast side. Evidently, the location accuracy varies with the proximity to the microphone array.

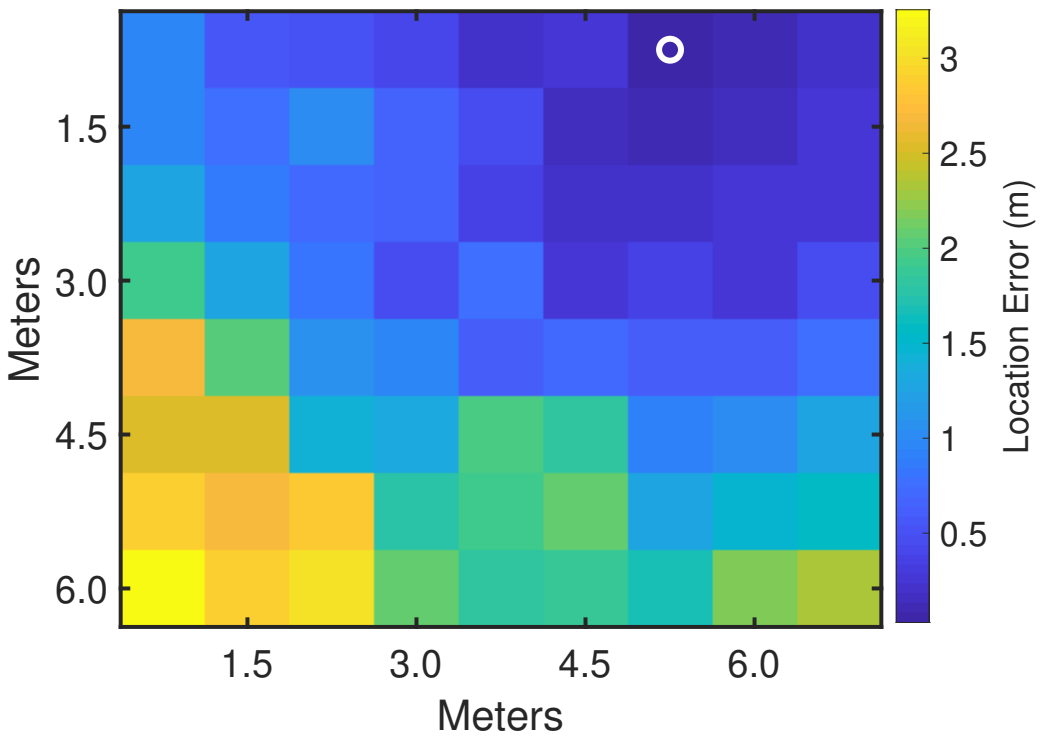


Figure 3.13: Heatmap of *VoLoc*'s localization error in the conference room (bird's eye view). Small white circle represents the microphone array location.

We classify all our measurements into three groups, based on the distance from the user to the microphone array: Near (< 2 m), Medium ($2 - 4$ m), and Far (> 4 m). Figure 3.14 shows that within 2 m, location error is almost always below 0.5 m, and within 4 m, the majority of the errors are still within 1 m.

■ Sensitivity to Different Users

To test *VoLoc*'s sensitivity to different users, we asked three volunteers

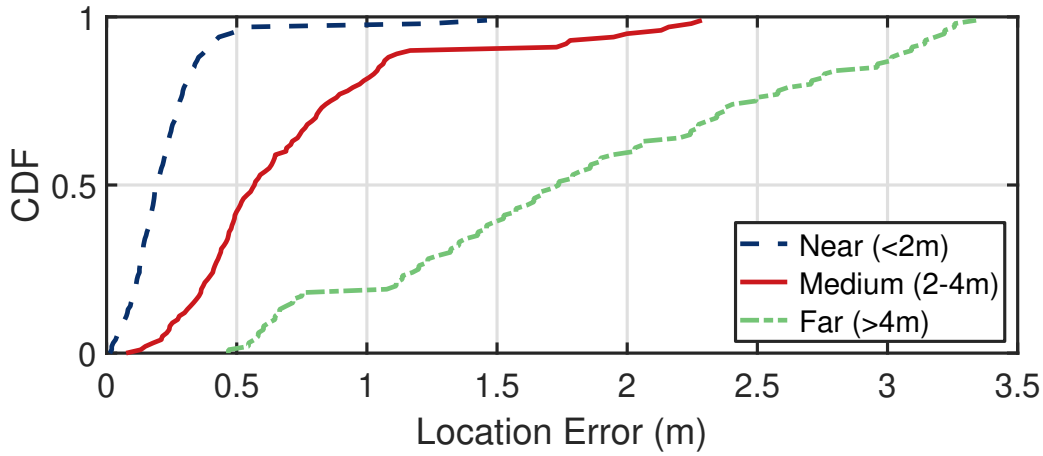


Figure 3.14: CDF of *VoLoc*'s localization error, according to the distance of the user from Alexa.

to enter the same room, stand at the same set of locations, and speak the same voice commands. Figure 3.15 shows the variation of median localization error across different locations. Evidently, localization error is much more correlated with the user's standing location (as would be expected), rather than the users voice or speaking patterns.

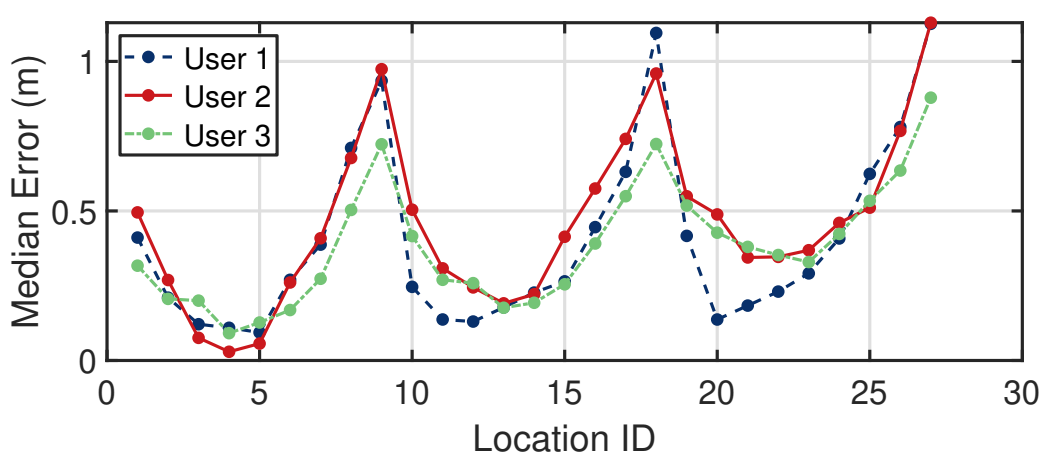


Figure 3.15: Variation of *VoLoc*'s localization error across different locations in the room, shown separately for each user.

■ Sensitivity to the Clutter Levels

Clearly, *VoLoc*'s performance will depend on the density of the multipath

signals due to other objects' reflections. Since we only look at the very beginning moment of the sound, most indoor reflections (like furniture) are not a problem for us. However, objects that are very close to the array may reflect sounds into the microphone even earlier than the wall, or even totally block the wall reflection, leading to higher cancellation residue and more location errors. In the extreme case where even the direct path is totally blocked, the localization error will go up dramatically.

Figure 3.16 shows the degradation of *VoLoc*'s localization accuracy, as we keep increasing the clutter level around the microphone array (i.e., putting objects on the same table as the array to add complexity to its multipath profile). Evidently, the error is low when there is no object nearby. Even when there are a few objects around and the clutter level is moderate, the location error is still acceptable. However, as more and larger objects start to fully block the wall reflection and even the direct path, the location error quickly increases.

■ Convergence of Geometric Parameter Estimation

Figure 3.17 shows how the parameter estimation is converging, with an increasing number of past voice commands. While more past samples are useful, with as few as 5 samples, our estimation has converged to < 1 cm and $< 2°$ fluctuation. This shows *VoLoc*'s ability to quickly converge to new wall parameters even after being moved around on the table. This experiment was performed at the medium clutter level (as per expectations, the estimation converges faster and slower for sparse and dense clutter levels, respectively).

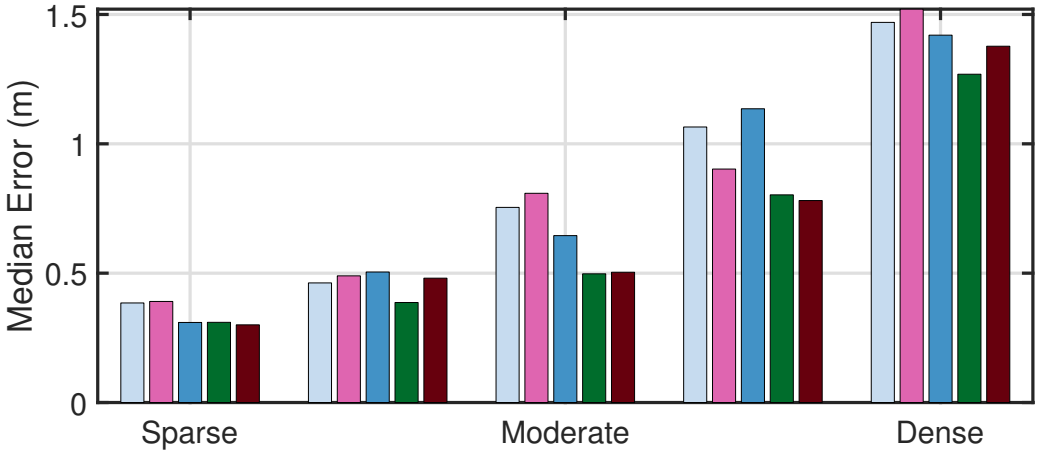


Figure 3.16: *VoLoc*'s localization accuracy across increasing clutter levels (from left to right). Each cluster of bars represents one environment, and each bar within one cluster represents the overall median error across locations during one visit. The three pictures correspond to the measurement in the #1, #3 and #5 environments.

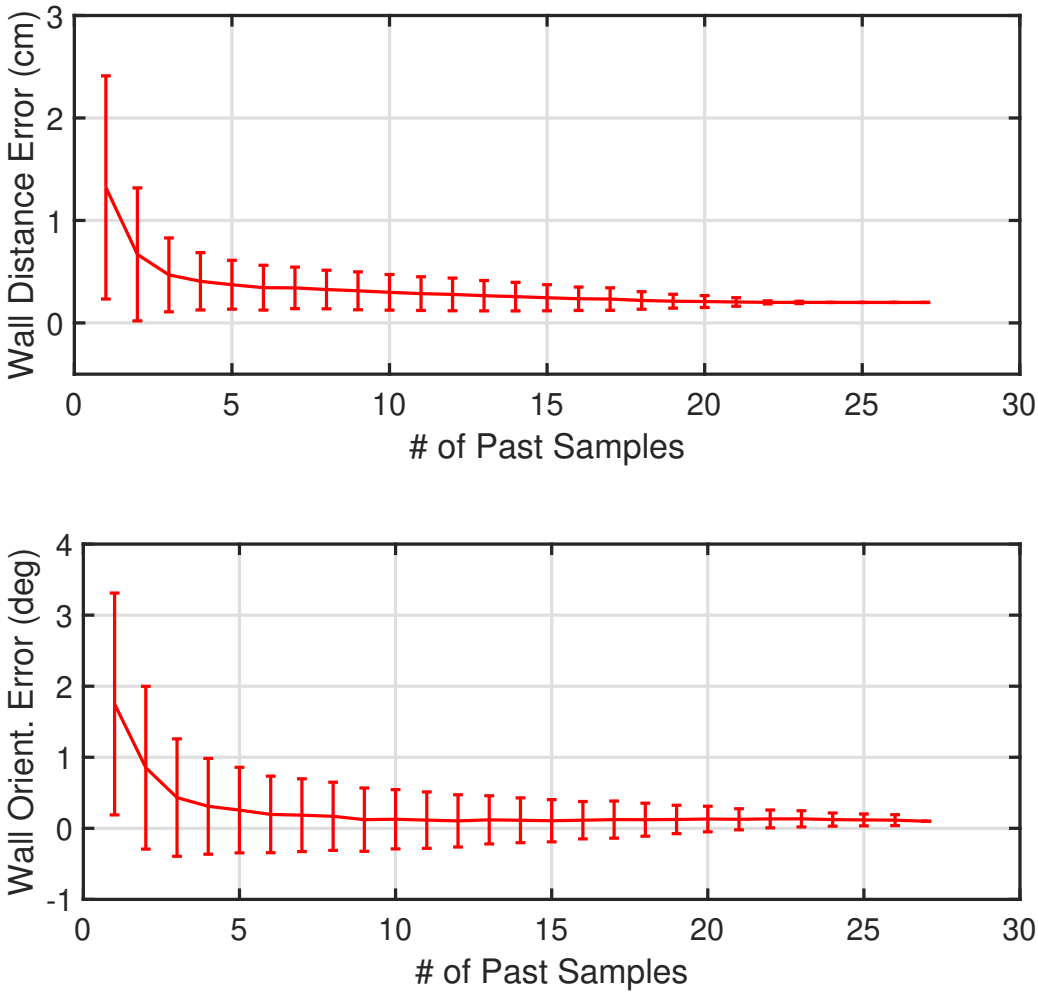


Figure 3.17: How *VoLoc*'s parameter estimation converges for (1) wall distance d, and (2) wall orientation θ, with increasing number of past voice samples. The red line and its error bar represent the average and standard deviation of estimation errors for a total number of 1000 runs.

3.5 Limitations and Discussion

In this section, we discuss limitations and room for improvement.

- **Semantic interpretation of location:** *VoLoc* infers the user and wall location in Alexa's reference frame. To be semantically meaningful (i.e., the user is at the laundry room) the inferred locations need to be superimposed on a floorplan. Alternatively, Alexa could localize other sounds, understand their semantics, and transfer those semantics to location. For instance, knowing that the washer/dryer sound arrives from the same location as a voice command can reveal that the user is at the laundry room. Building such a semantic layer atop localization is an important follow-up work.

- **Coping with variations in height:** A family will likely have multiple users with different heights (including children). *VoLoc* needs the height of each user and some form of voice fingerprinting to apply the corresponding height during computation. We have not implemented such per-user adaptation. We also do not cope with scenarios where the user is sitting or lying down (we assume standing users).

- **Mobile users:** *VoLoc* has been designed and evaluated with static users. When a user issues a voice command while walking, the echo patterns will likely "spread" in space. Our current formulation does not model the effects of mobility – the algorithms will need to be revisited.

- **Many pause opportunities:** A voice command offers at least two pause opportunities, one before the command, and one after the word "Alexa". Naively averaging location, derived from each pause, will improve accuracy. However, averaging in the signal space (i.e., optimizing the wall parameters using all the post-pause signals) could offer greater benefits. We leave such refinements to future work.

- **Privacy:** How applications use the location information in the future remains an open question. On one hand, we see context-aware Alexas and Googles becoming crucial support technologies to old age independent living; sharing the user's height may be worthwhile in such cases. On the other hand, for everyday users, we see Amazon and Google peering even more closely into our homes and daily lives, a stronger erosion of privacy. Regardless of utility or abuse, we believe awareness of such capabilities is critical. We hope this chapter aids in developing this awareness.

3.6 Related Work

In the interest of space, we give a heavily condensed summary of the vast literature in localization and acoustic signal processing, with bias towards work more related to *VoLoc*.

- **Multiple arrays or known sound signals**: Distributed microphone arrays have been used to localize (or triangulate) an unknown sound source, such as gun shots [78], wildlife [77], and mobile devices [85]. Many works also address the inverse problem of localizing microphones with speaker arrays playing known sounds [106, 107, 108, 109]. Ishi et al. [80] report modeling the room multipath to improve multi-array localization results. Xiong and Jamieson [81] and Michalevsky et al. [110] demonstrate localization using multiple RF-based landmarks. On the other hand, when the source signal is known, localization has been accomplished by estimating the channel impulse response (CIR). For instance, [76] uses an acoustic sine sweep to localize room boundaries and compute the shape of a room; reverbs captured by multiple microphones reveal the room impulse responses (RIR), which stipulate the locations of reflecting surfaces. In RF (like WiFi), CIR and SNR based

fingerprinting has been used extensively [111, 79, 112, 113, 114, 82]. As mentioned earlier, *VoLoc* must cope with single array and unknown signals.

- **Unknown signal, single array**: Perhaps closest to *VoLoc* are [103, 84]. In [103], a robot scans a 3D model of the room with a Kinect, identifies AoAs with a microphone array, then performs 3D inverse ray-tracing to localize sound sources. Besides mapping room geometry, the robot also relies on specific impulse-like sounds, such as clapping. Ribeiro et al. [84] simulate localization using a microphone array and present results from three carefully chosen locations in an empty, rectangular room. In comparison, our solution is designed for real-world, multipath-rich, uncontrolled environments. In [115], a single microphone is used to classify a speaker's distance into a set of discrete values, but needs per-room, per-distance training.

- **AoA estimation**: Rich bodies of work have focused on acoustic AoA using microphone arrays [116, 117, 118, 119, 120]. Some are best suited for different sound sources, some for specific signal types and frequencies, some for certain environments. Examples include delay-and-sum [121, 92, 122], GCC-AoA [94, 44, 95, 101, 96], MUSIC [97, 98, 102], and ESPRIT [99, 100]. However, in multipath-rich environments, blind AoA estimation struggles, especially for successive AoAs.

- **Blind channel estimation**: Blind channel estimation (BCE) describes the process of deducing a channel from received signals without knowing the source. BCE is a useful tool for estimating indoor acoustic channels [123, 124, 125, 126]. We consider IAC to be a particular and powerful realization of BCE with significant computation gains. IAC was also inspired by ZigZag decoding for RF networks [127], which decodes packets by exploiting interference-free segments.

3.7 Conclusion

This chapter shows the feasibility of inferring user location from voice signals received over a microphone array. While the general problem is extremely difficult, we observe that application-specific opportunities offer hope. Instead of inferring and triangulating all signal paths in an indoor environment, we observe that estimating a few AoAs and reflector surfaces is adequate for the localization application. We design an iterative cancellation algorithm (IAC) for AoA estimation, followed by a joint optimization of wall distance and orientation. When fused together, the localization accuracies are robust and usable.

Location-awareness in Alexa and Google may entail important privacy implications. We emphasize that we are not proponents of in-home localization, except perhaps in special scenarios like old age independent living. Nonetheless, we believe this work is still important to spread awareness of what is possible from voice signals. We hope such awareness helps in shaping policies around voice assistants of the future, both for utility and abuse in smart indoor environments.

CHAPTER 4

MUSE: IMU ORIENTATION TRACKING FOR 3D DIRECTIONAL SOUND

We aim at generating 3D directional sound using earphones, by synthesizing the (slightly different) propagation delays at two ears. Applications include acoustic navigation, acoustic augmented reality, etc. The key challenge is to track the head's (earphone's) orientation accurately. A rich body of work has focused on orientation tracking techniques using inertial sensors, namely accelerometers, gyroscopes, and magnetometers. This chapter identifies room for improvement over today's orientation tracking techniques. The core observation is that conventional systems have trusted *gravity* more than the *magnetic North* to infer the 3D orientation of the object. We find that the reverse is more effective, especially when the object is in continuous fast motion. We leverage this opportunity to design *MUSE*, a magnetometer-centric sensor fusion algorithm for orientation tracking. Real experiments across a wide range of uncontrolled scenarios show consistent improvement in orientation and location accuracy, without requiring any training or machine learning. We believe these results constitute an important progress in the otherwise mature field of IMU-based motion tracking.

4.1 Introduction

Inertial motion units (IMUs) serve as the bedrock to a large number of mobile systems and applications. Smartphones and smartwatches have already utilized IMUs to infer human activities and gestures, while drones and robots have classically employed IMUs to guide their motion-stabilization and con-

trol algorithms. More recently, IMUs are playing a role in everyday objects. A start-up called Grush [128] proposed a fascinating idea that won the "2016 America's Greatest Makers" contest. The company tracks the motion of *IMU-embedded toothbrushes* and feeds this motion into a smartphone game where monsters need to be killed. When brushing teeth, a child must move to different corners of his mouth to kill the scattered monsters in the smartphone screen. In a more serious context, health rehabilitation centers are increasingly giving motion-trackers to patients so their progress can be monitored even at home. Needless to say, any improvement to IMU-based motion tracking will impact a range of systems and applications.

Let us begin by intuitively understanding the core problems in IMU-based motion tracking. Observe that all the inertial sensors, i.e., accelerometer, gyroscope, and magnetometer, operate in their local frames of reference. For instance, if the accelerometer measures motion along its X axis, it is not clear what this motion means in the global reference frame. As an analogy, imagine a friend calling from inside a flying airplane and saying that she is turning "right". Without knowing the 3D orientation of her plane in the earth's reference frame, there is no way to infer which way she is turning. Put differently, tracking the motion of any object first requires the knowledge of the object's 3D orientation in the *global framework*. Then, the motion sensed locally by the IMU sensors can be appropriately projected onto the global framework, ultimately enabling a meaningful motion tracking solution.

Precisely estimating the object's 3D orientation in the global reference frame (GRF) is non-trivial. Conventional systems solve this by utilizing gravity and magnetic North as "anchor" directions. Loosely, the accelerometer measures components of gravity along its three axes and infers how tilted the object is from the horizontal plane. Once this tilt is compensated, the magnetometer can measure the object's heading angle by comparing against

the magnetic North direction. This indeed yields the 3D orientation of the object, *although only in the case when the object is completely static.* When the object is moving, the accelerometer measures the "*mixture*" of both gravity and linear motion, making gravity isolation difficult. As a consequence, the object can no longer be tilted precisely to become flat on the horizontal plane, which further pollutes the estimation of the magnetic North direction. In summary, estimating the global 3D orientation of a moving object is the key bottleneck to IMU based motion tracking. Worse, this 3D orientation needs to be continuously tracked, since local sensor measurements need to be continuously projected onto these global directions.

Of course, we are not the first to look into this problem. Many techniques exist in the literature [129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141], of which A^3 [142] from MobiCom 2014 is probably the most effective. Importantly, A^3 sidesteps the problem of disentangling gravity and linear motion; instead, it opportunistically searches for moments when the object is static. The object's 3D orientation is re-calibrated at these static moments (using gravity and North), and for all times between these moments, A^3 uses the gyroscope to track the local changes in orientation. Referring back to our analogy, if our friend from the airplane could periodically stop and tell us her global orientation, we could utilize her local measurements of "turning right" to interpolate her global orientations at all times.

Unfortunately, objects may not pause often, and even if they do, determining those moments produces false positives and false negatives. All in all, A^3's performance upholds only in certain types of movements, and even then, some components of IMU information remain unused (as elaborated later).

This chapter, inspired by A^3, finds room for improvement. Our core insight is that the *earth's 3D magnetic North vector* could serve as a better global

"anchor" than gravity, especially during motion. This is because magnetometers are mostly unpolluted by the device's motion and can, as a result, always measure the magnitude of the global 3D North vector. The direction is still unknown since it is a function of the object's own orientation. Nonetheless, if we can measure the direction of this North vector just one time, we can utilize it thereafter as a trusted anchor for tracking orientation, even when the object is moving. Gravity is still necessary, but only as a secondary anchor to complete the orientation estimation.

This sensor fusion opportunity ultimately results in *MUSE*, an iterative algorithm that uses gravity from a static moment to estimate the global 3D North vector, and thereafter uses this North vector as the primary anchor. Gravity is still used, although with varying trust, depending on whether the object is moving. Finally, the gyroscope measurements are also used to track rotation, and then fused with the accelerometer + magnetometer measurements to overdetermine the system. The net result is improved 3D orientation in the global reference frame (GRF), in turn helping the localization aspect of motion tracking.

We implement and evaluate *MUSE* on an off-the-shelf Samsung Galaxy S6 smartphone. Orientation tracking is evaluated across various human activities and object motions. Ground truth is obtained by periodically bringing the device to a pre-specified orientation. Comparison with A^3 shows an average of 2.9X performance gain for various natural activities, and higher when the motion is continuous without pauses.

We summarize *MUSE*'s key contribution as follows:

- We identify that the *global 3D magnetic North vector can serve as a better anchor compared to gravity.*
- We design an efficient magnetometer-centric sensor fusion algorithm to

track the 3D orientation of a moving object.

The rest of this chapter elaborates on this contribution; however, we need to begin with the foundations of IMU based motion tracking. While this makes the next section long, we believe the material is necessary to appreciate the problems and solutions. We also believe the material is easy to follow, since it starts from first principles.

4.2 Foundations of Tracking

- **Reference Frames:** Consider the general case where an object's motion, i.e., sequence of 3D locations and 3D orientations, needs to be tracked in a global reference frame, say $\langle$*North*, *East*, and *Up*$\rangle$. The sequence of 3D locations, when differentiated twice, gives acceleration; the accelerometer measures this acceleration, but in its local reference frame (LRF). The sequence of 3D orientations, when differentiated once, gives angular velocity; the gyroscope measures this angular velocity, but again in the LRF. Tracking 3D motion in the global reference frame (GRF) requires a continuous translation between the two coordinate frames. Specifically, at every time instant, the object's LRF needs to be rotated and aligned to the GRF, and the acceleration needs to be computed in this aligned framework. Thus, the first question in tracking degenerates to constantly estimating this LRF-to-GRF rotation.

- **Understanding 3D Orientation:** The LRF-to-GRF rotation is essentially the phone's 3D orientation. To understand, consider a plane taking off in Figure 4.1, with its true 3D heading direction as 45° North-East and 30° to the vertical direction. However, for a passenger inside the plane, the heading direction is always along its local Y axis. Thus, computing the global heading direction entails rotating the plane $-30°$ around its local X axis (which is along the wing of the plane), and then

rotating the plane $-45°$ around its local Z axis. These rotations align the local and global axes, capturing the orientation mismatch between the LRF and GRF. Mathematically, this net mismatch can be modeled as a single 3D rotation matrix, and the inverse of this matrix (i.e., the mismatch from GRF to LRF) is defined as the object's 3D orientation.

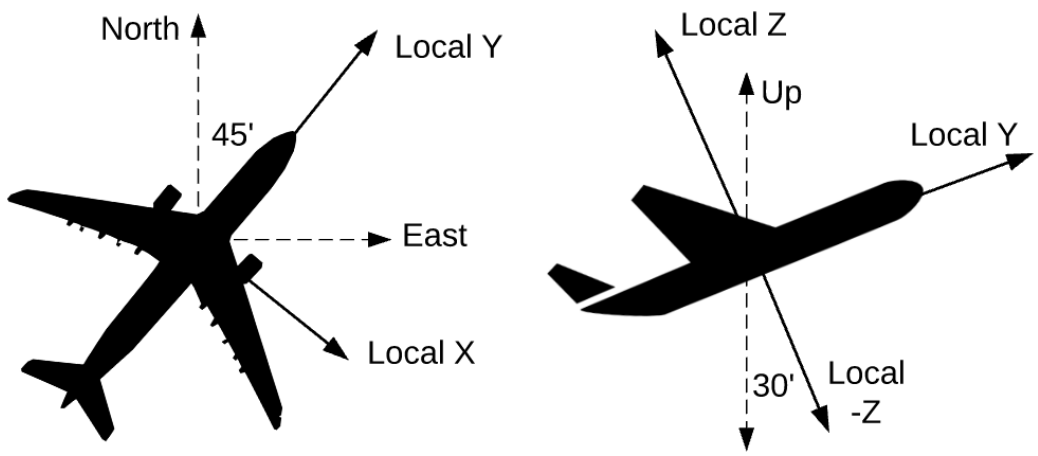


Figure 4.1: The top view (left) and side view (right) of a plane taking off. The 3D orientation of the plane is the net rotation needed to align the plane's local $\langle X,\ Y,\ Z\rangle$ axes with the global $\langle North,\ East,\ Up\rangle$ axes.

- **Computing 3D Orientation:** In reality, how would a passenger inside the plane compute its global heading direction? One opportunity is to measure quantities, such as gravity and magnetic North, whose directions are universally known in the global reference frame (GRF). Specifically, an accelerometer would be able to identify that the universal gravity vector is tilted by $30°$ from its local $-Z$ axis. Similarly, a magnetometer (or compass) should be able to recognize that the Earth's magnetic North direction is offset by $45°$ from its local Y axis. Thus, using local measurements of gravity and North from the accelerometer and the magnetometer respectively, an object should be able to compute its own LRF-to-GRF mismatch, which is exactly the 3D orientation of the object.[1]

[1]Note that only gravity or only North is inadequate for determining 3D orientation.

- **Basic Motion Tracking:** Now, tracking the object's *3D orientation* over time can be achieved in two ways: (1) performing the above LRF-to-GRF alignment (using gravity and North) at every instant of time, or (2) performing the alignment once to get initial orientation, and then integrating the gyroscope data thereafter to obtain subsequent orientations. Tracking the object's *3D location* over time is slightly more involved. For each time step, the object's 3D orientation needs to be estimated and the accelerometer data needs to be projected to this global reference frame. This projected accelerometer data contains both linear acceleration and a gravitational component (Figure 4.2(a)). After removing gravity, linear acceleration is now double integrated to compute the next location. The orientation and location estimation repeats for every time step, ultimately producing the 3D orientation and 3D location of the object at any given time.

4.2.1 Why Theoretical Tracking Falls Short

The above tracking method, although conceptually complete, does not scale to real world situations. We discuss three main issues:

(1) **Gravity Pollution:** In describing how the accelerometer uses gravity to compute its vertical misalignment, we need to assume that the object is static. Otherwise, the object's motion will mix with gravity measurements, yielding an incorrect tilt. Put differently, computing the vertical tilt of a moving object is difficult.

(2) **Magnetic Interference:** In indoor environments, magnetometer measurements of the earth's magnetic North can be polluted by nearby ferromagnetic materials. This again derails the estimate of 3D orientation, impacting 3D location as well.

For instance, even if gravity is perfectly aligned along the $-Z$ axis, the object can still be in many possible orientations on the horizontal plane (with different heading directions).

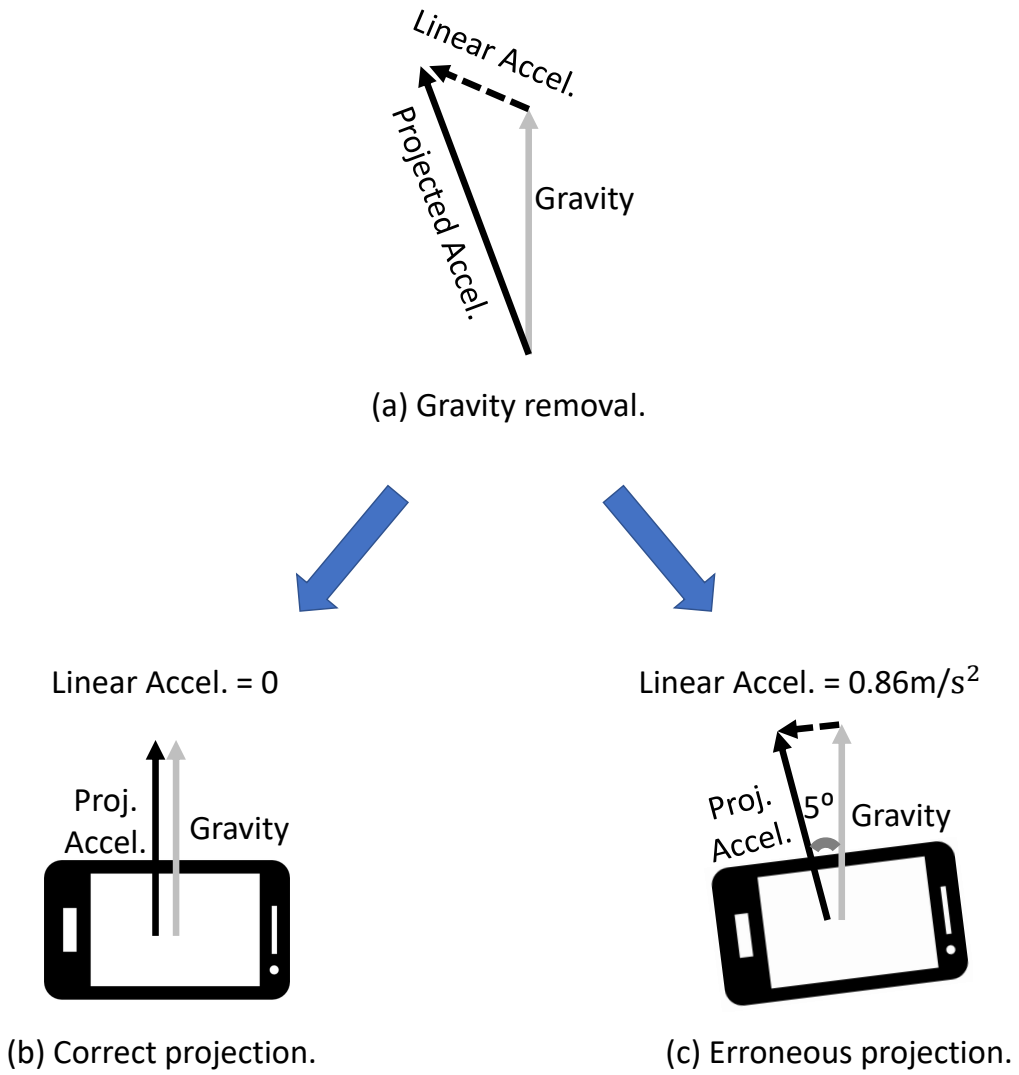


Figure 4.2: (a) Accelerometer projected to GRF contains both linear acceleration and gravity. (b) Take a static object as an example, projection with correct orientation removes gravity perfectly, making no errors in linear acceleration, but (c) slight offset in orientation can cause large projection error, leading to wrong linear acceleration.

(3) **Inherent Sensor Noise:** Finally, hardware noise is inherent in all IMU sensors. Any integration operation accumulates this noise and the problem is pronounced for location tracking with accelerometers. This is because accelerometers need to be integrated *twice* to obtain location, and further, any orientation error directly translates to accumulated location-error over time. Gyroscopes also drift, but less since they require a single integration. Magnetometers do not drift, but experience a random high frequency noise in their measurements.

We emphasize again the perils of 3D orientation error on location. Observe that the accelerometer data would get projected erroneously from LRF to GRF (Figure 4.2(b) and (c)), and as subsequent velocities and locations get computed in GRF (using single and double integrations), the error will diverge over time. The analogy is in orienting a gun slightly off from the direction of the target – the margin by which the bullet misses the target increases with the distance of the target from the gun. Thus, precisely estimating 3D orientation is crucial and challenging, especially for a moving object in an indoor (ferromagnetic) environment.

With this background on practical challenges, we focus on today's techniques and identify the room for improvement.

4.2.2 State-of-the-Art Method

Classical motion tracking spans control theory, robotics, signal processing, and graphics [136, 137], and is difficult to cover limited space. The recent work called A^3 [142] from MobiCom 2014 is perhaps the most practical solution today. As mentioned earlier, A^3 recognizes that accelerometers measure the mixture of gravity and motion, that isolating gravity is difficult, and that 3D orientation is difficult to estimate. In light of this, their proposal is to identify opportunities when gravity measurement is unpolluted. This

happens in two cases: (1) when the object is static, or (2) when the object is in pure rotational motion (i.e., only rotating but not moving). In both these cases, the accelerometer only measures gravity, allowing for estimating the global orientation.

The static case is easy to detect, but for pure rotational motion, A^3 showed that the gyroscope measurement correlates well with the accelerometer data, since gravity will spin similarly in the object's local reference frame. Falling back to the plane analogy, consider Alice sitting inside a plane that is not moving but only spinning in the air (and assume Alice has a gyroscope in her hand). Also assume that Alice can track how the direction of the sun is changing (perhaps because the plane is made of transparent glass). A^3 points out that Alice would see the sun spinning around her, and this spin should correlate with the spin measured by her gyroscope. However, if the plane was both spinning and moving linearly, the correlation would break down. Thus, strong correlation is an indication of unpolluted gravity, offering an opportunity to align LRF to GRF, and ultimately infer the device's 3D orientation.

We believe that A^3 is an elegant contribution; however, the shortcoming is that such opportunities are infrequent. Figure 4.3 shows a case of running, where the accelerometer magnitude is constantly varying. Given that running activities can easily last for far longer time durations, A^3 may not be able to utilize the pausing opportunity at all. For many real-world movements, the state of "rotation but no acceleration" also occurs rarely – for the running case in Figure 4.3, as well as many other cases evaluated later, we did not find a single opportunity. As a result, 3D orientation tracking remains an elusive problem, hindering accurate location tracking. In light of this, we present our proposal on magnetometer fusion, *MUSE*.

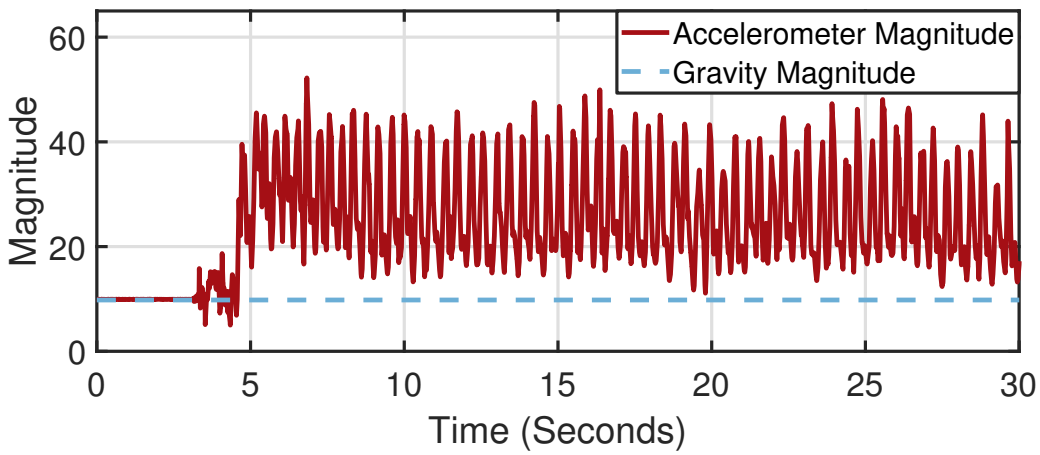


Figure 4.3: The magnitude of accelerometer measurement vs. the constant magnitude of gravity, as a user picks up a phone in hand and starts running. Clearly, the accelerometer cannot measure gravity properly in this case.

4.3 Orientation Estimation

This section proposes improvements to 3D orientation estimation. Note that A^3 and prior methods have always viewed gravity as the primary "anchor" for estimating orientation, and since gravity gets mixed with linear motion (in the accelerometer), it is difficult to extract a precise global orientation. We break away from this approach and observe that the 3D magnetic North vector can be a more effective anchor. The advantage arises from magnetometers being unaffected by linear motion of the object. However, the tradeoff is that the intensity and direction of magnetic North may vary across locations (unlike gravity). In light of this, *MUSE* requires the object to start from a static moment, utilizes the unpolluted gravity to precisely estimate the 3D North vector, and thereafter uses the North vector as the anchor for orientation. We elaborate the algorithm next.

4.3.1 MUSE Overview

Figure 4.4 shows the *MUSE* orientation estimation pipeline. IMU data (accelerometer) from the initial static time window offers unpolluted gravity,

used to determine the vertical tilt of the object. The magnetometer, on the other hand, measures the 3D magnetic vector in its local reference frame (LRF), and projects it to the horizontal plane to compute the object's heading. These two anchors together fully determine the 3D orientation of the object in the global reference frame (GRF).[2] Now, once this initial 3D orientation $\boldsymbol{O}(t_0)$ is known, the current magnetometer's local measurement, $\boldsymbol{N}^L(t_0)$, can be projected back onto the GRF, leading to the global 3D magnetic North vector, $\boldsymbol{N}^G$. This gives us the anchor we need:

$$\boldsymbol{N}^G = \boldsymbol{O}(t_0)\boldsymbol{N}^L(t_0) \tag{4.1}$$

As the object starts moving, the magnetometer tracks $\boldsymbol{N}^G$ in its LRF, leading to 2 DoFs (degrees of freedom) of global orientation. In a parallel thread, gyroscope tracks all 3 DoFs of rotation. Together, this is an overdetermined system, with 5 DoFs of information available for 3 DoFs of changing orientation. To avoid pollution from linear motion, gravity estimation will be opportunistically used when the object stops or moves slowly. Thus, our task at hand is to solve this overdetermined system via sensor fusion.

4.3.2 Magnetometer + Gyroscope Fusion

Since gyroscopes measure changes in 3-DoF orientation, and magnetometers measure 2 DoFs of global orientation (which is $\boldsymbol{N}^G$) directly, the two sensors can be combined to better track (2 DoFs of) orientation. Thus, while gyroscope drift accumulates over time, the magnetometer can be used for recalibration (achieving better noise properties than either of the individual sensors). We use the complementary filter for this combining operation.

[2]In other words, roll and pitch angles are computed from gravity, while the yaw angle is computed from compass. This is a standard operation, which is also implemented by the Android SensorManager.getRotationMatrix() API: `https://android.googlesource.com/platform/frameworks/base/+/master/core/java/android/hardware/SensorManager.java`

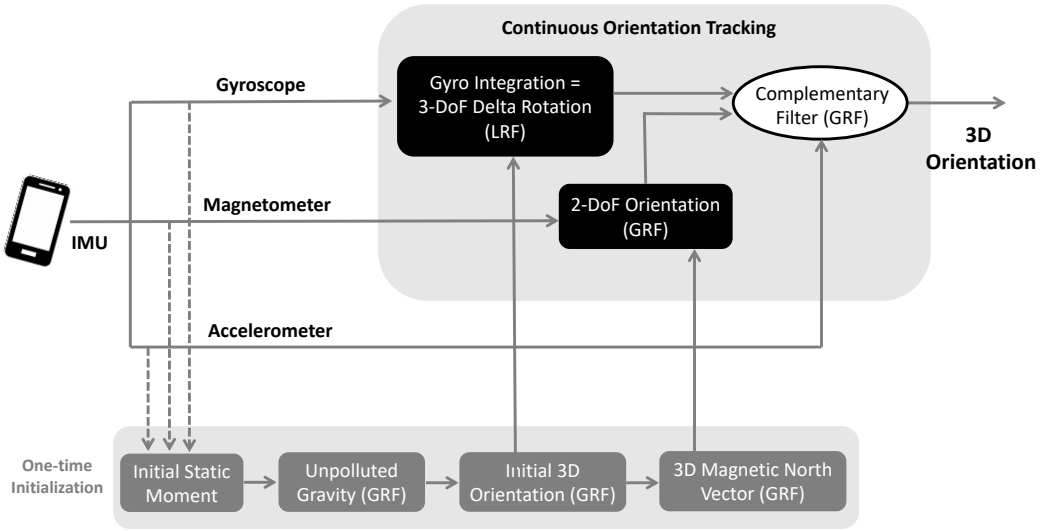


Figure 4.4: *MUSE* processing pipeline: The IMU data is processed in stages to compute the 3D orientation in the global reference frame. The gyroscope is integrated to provide 3-DoF information on orientation, while the magnetometer produces extra information on 2 DoFs. These two sources of information are fused in a complementary filter. The accelerometer is opportunistically used to refine 3D orientation. A one-time initialization step is necessary to bootstrap the system, during which the initial orientation and the 3D magnetic North vector are computed.

To elaborate, assume that at current time t, the actual orientation of the object is $\boldsymbol{O}(t)$. $\boldsymbol{O}(t)$ is a 3×3 rotation matrix that rotates the object's GRF to LRF, and is not known to the object. Since the magnetometer measures the (constant) global 3D magnetic North vector $\boldsymbol{N}^G$ in its local reference frame (LRF), we can write its local measurement at time t as:

$$\boldsymbol{N}^L(t) = \boldsymbol{O}^{-1}(t)\boldsymbol{N}^G \tag{4.2}$$

where $\boldsymbol{O}^{-1}(t)$ denotes the inverse of $\boldsymbol{O}(t)$. Of course, we do not know the actual orientation $\boldsymbol{O}(t)$, and our current *estimated* orientation, $\hat{\boldsymbol{O}}(t)$, derived from the gyroscope, may be erroneous. One way to check how large this error would be, is to use estimated orientation $\hat{\boldsymbol{O}}(t)$ to *infer* what the magnetometer should measure (as $\hat{\boldsymbol{N}}^L(t)$), i.e.,

$$\hat{\boldsymbol{N}}^L(t) = \hat{\boldsymbol{O}}^{-1}(t)\boldsymbol{N}^G \tag{4.3}$$

The difference between the inferred measurement $\hat{\boldsymbol{N}}^L(t)$, and the actual magnetometer measurement $\boldsymbol{N}^L(t)$, immediately reveals the drift in orientation,[3] as illustrated in Figure 4.5.

■ **Now, how do we update orientation estimation using this disparity?** The key observation here is that the noise properties of magnetometers and gyroscopes are different, allowing for informed sensor fusion. Specifically, gyroscopes exhibit a long-term integration drift that grows with time[4]; however, in the short term, they are quite accurate. Magnetometers, on the other hand, have short-term noise from environmental fluctuations and sensor imperfection; however, they do not drift in the long run since they are always measuring the same global North vector and no integration

[3]To be precise, it reveals the drift in 2 of the 3 DoFs of orientation (the DoFs parallel to the 3D North vector direction).

[4]This is because the gyroscope is an inertial sensor that measures angular velocity in its local frame of reference, and has no opportunity to correct its own integration.

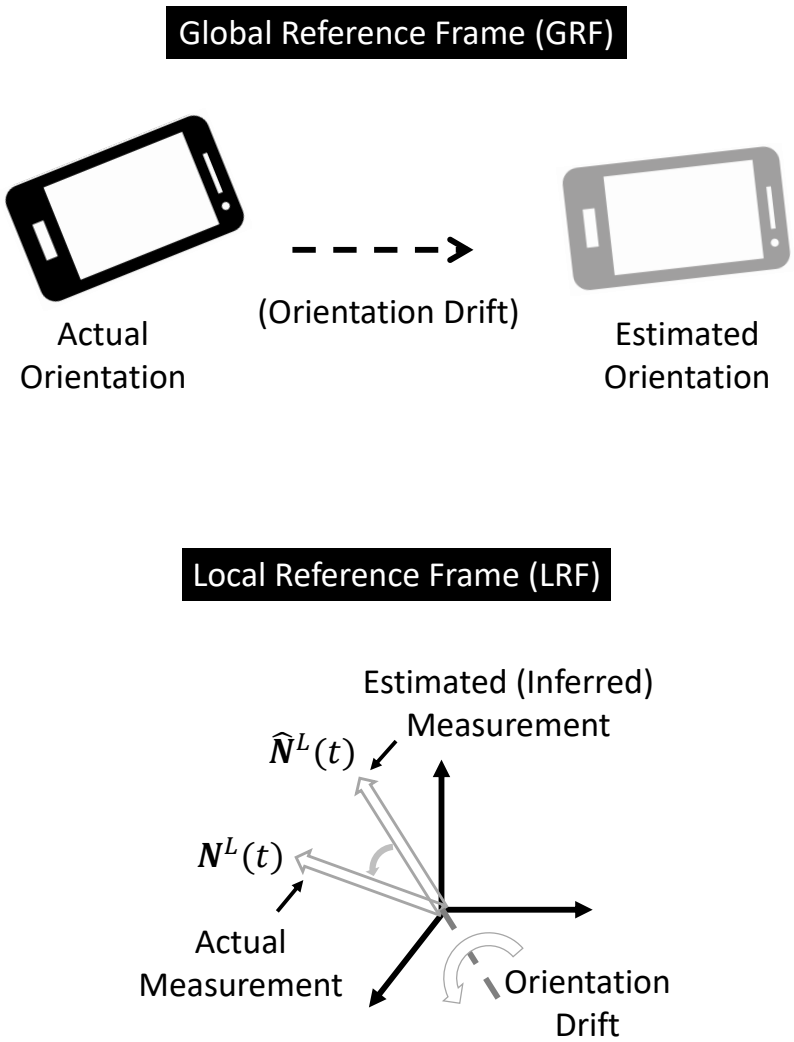


Figure 4.5: Magnetometer measures a globally constant vector, which helps correct the drift of orientation, reflected as the disparity between the inferred magnetometer measurement and the actual measurement.

is needed. To fuse the best of both sensors, we employ a *complementary filter* [133, 134, 135].

The complementary filter essentially computes a weighted combination of the two. Larger weight is assigned to the gyroscope, so that high frequency components are drawn from the gyroscope (since the gyroscope drifts less in the short term), and low frequency components from the magnetometer (since it is stable in the long run). The net output is a single orientation at each time instant – the estimate of 3D orientation in the GRF. Specifically, we first look for a delta rotation matrix ($\Delta\boldsymbol{R}$) that can align our *inferred* magnetometer measurement ($\hat{\boldsymbol{N}}^L(t)$) to the actual one ($\boldsymbol{N}^L(t)$):

$$\begin{aligned} \textit{Rotation Axis } \boldsymbol{e} &= \hat{\boldsymbol{N}}^L(t) \times \boldsymbol{N}^L(t) \qquad \text{(Cross Product)} \\ \textit{Rotation Angle } \boldsymbol{\theta} &= \measuredangle\big(\hat{\boldsymbol{N}}^L(t)\,, \boldsymbol{N}^L(t)\big) \\ \Delta\boldsymbol{R} &= \textit{AxisAngle2RotMat}\ (\boldsymbol{e}, \boldsymbol{\theta}) \end{aligned}$$

The inverse of this rotation matrix (i.e., $\Delta\boldsymbol{R}^{-1}$), when applied to our orientation estimation, will eliminate this disparity. Since we adopt a complementary filter design to reduce noise, we set a small coefficient α ($0 < \alpha \ll 1$) for this operation, i.e.,

$$\Delta\boldsymbol{R}(\alpha) = \textit{AxisAngle2RotMat}(\boldsymbol{e}, \alpha\boldsymbol{\theta}) \tag{4.4}$$

And the updated orientation is

$$\hat{\boldsymbol{O}}_{\text{new}}(t) = \Delta\boldsymbol{R}^{-1}(\alpha) \cdot \hat{\boldsymbol{O}}_{\text{old}}(t) \tag{4.5}$$

As a technical detail, we are using axis-angle and rotation matrix representation of rotation, rather than Euler angles (roll/pitch/yaw), in order to avoid gimbal lock and $\pm 180°$ ambiguity. In our implementation, we simply set α as 0.01. The final result from this filter is the convergence on 2 DoFs of

orientation, while the 3rd DoF, i.e. rotation around the 3D magnetic North vector, is still tracked but prone to drift.

4.3.3 Implementation Details

We briefly mention a few details here.

- **Gyroscope Bias**: IMU sensors are known to have bias (DC offset), of which gyroscope bias harms orientation estimation the most (because of integration). At the initial static moment where we perform one-time initialization of tracking, we also calibrate the bias by taking a time average of gyroscope readings, and remove the bias from subsequent gyroscope measurements.

- **Static Recalibration**: Even though *MUSE* addresses the gyroscope drifting issue (due to integration) in 2 of the 3 DoFs of orientation, there is still 1 DoF whose error cannot be corrected during motion. Therefore, *MUSE* also opportunistically detects static or slow-moving opportunities (by looking at time windows in which the accelerometer measures roughly 9.8 m/s^2), if any, to address the drift in this dimension. Unlike A^3 which simply replaces current orientation estimation with the one from gravity + North, we again use a complementary filter to update the estimation, which turns out to be more robust to accelerometer noise and have fewer false positives. We also leverage this opportunity to update our estimation of the 3D magnetic North anchor, $\boldsymbol{N}^G$.

- **Magnetometer Accuracy**: MEMS magnetometers in mobile devices typically have lower resolution than accelerometers and gyroscopes. Luckily, we do not require as high resolution for magnetometers as for gyroscopes, because we are integrating the gyroscope readings but averaging the magnetometer readings (using the complementary filter).

However, there might be ferromagnetic materials in the environment. Depending on the error distribution, in certain cases, it might be even better not to use the magnetometer. We evaluate the sensitivity of the algorithm to magnetic field fluctuations in the next section, and leave further investigation to the future work.

In sum, Algorithm 4 below shows the high-level pseudo code of *MUSE*'s orientation estimation algorithm.

Algorithm 4 *MUSE* Orientation Tracking

```
Opportunistically detect initial orientation and global 3D magnetic North
  anchor, using Equation (4.1)
while True do
    Integrate gyroscope to obtain new orientation
    if Accelerometer roughly measures 9.8m/s^2 then
        Recalibrate orientation estimation, and
        Update 3D magnetic vector estimation
    else
        Update orientation using Equation (4.3) - (4.5)
    end if
end while
```

We evaluate the accuracy of orientation estimation next.

4.4 Orientation Evaluation

4.4.1 Experiment Design

- **Platform and Test Scenarios:** *MUSE* uses the raw IMU data from a Samsung Galaxy S6 smartphone. It includes an InvenSense MPU6500 6-axis accelerometer + gyroscope, and a Yamaha YAS537 3-axis magnetometer. The same chips are also embedded in many other mobile

and wearable devices, including other phone models (iPhone 6s, Amazon Fire Phone, Samsung Galaxy S5, Samsung Note 5), tablets (Kindle Fire HD), smartwatches (Samsung Gear Fit), VR headsets (HTC Vive, Oculus Rift), gaming controller (Oculus Touch, Steam Controller), etc. Tracking various motion patterns is of interest, for both *humans* and *things*. For humans, we begin with controlled activities, like pure linear motion, pure rotation, and their mixtures. Then, we generalize to real-world natural motions, including running, eating, basketball, gaming, etc. For these activities, we recruit volunteers to carry/wear the phone in different positions, such as in-hand, wrist, arm, and legs. We offer no guidance to volunteers; they perform the activities completely naturally. Finally, for object motion, we insert/paste the phone on various "things", including tennis racquets, soccer balls, bicycle wheels, etc.

- **Metric:** Our main metric of interest is *3D orientation error* of the phone. Observe that this error need not be shown as separate errors around X, Y, and Z axes, respectively, but can be shown as a single orientation error (i.e., the minimal amount of rotation needed to align the estimated 3D orientation to the ground truth 3D orientation). Of course, this raises the question of determining the ground truth orientations.

- **Ground Truth:** *MUSE* adopts A^3's technique of measuring orientation ground truth. We start by placing the phone at a known orientation, using a printed protractor; then we use the phone for the test motion or activity; and then we bring the phone back to this known orientation. Since this end point is naturally a static moment for *MUSE*'s recalibration, we deliberately pause our algorithm and only use gyroscope integration for the last few seconds of motion. This ensures that the true motion tracking error is measured (without an artificial orientation reset at the end). Finally, since many of our motion track-

ing sessions will be long (5+ minutes), we will periodically bring the phone to the ground-truth orientation, pause, and then continue natural motion again. However, we will not use these artificial pauses for re-calibrating 3D orientation, but only for measuring the ground truth at intermediate points during motion. This will offer insights into the intermediate moments while motion is in progress.

- **Comparison Baselines:** Last but not least, we will compare *MUSE*'s performance against three other techniques. **(1) A3** from MobiCom 2014; **(2) GyroOnly**, indicating rotation estimation from 3-DoF gyroscope integration alone, with no gravity or magnetometer; and **(3) ComplemFilter**, indicating the traditional use of complementary filter for IMU sensor fusion, which constantly combines gyroscope integration with the estimation from gravity + North. Across all these cases, the algorithms are executed in MATLAB, using the same IMU data supplied by the *SensorManager* API from Android.

4.4.2 Results

We begin with the discussion of basic (controlled) motions, and then evaluate natural activities and gestures.

Basic Controlled Motions

Figure 4.6 plots 3D orientation error for pure translation, pure rotation, and mixtures of translational and rotational motions. For pure translation, the phone is continuously moved in different straight-line directions (not necessarily horizontal or vertical, but other possible diagonal lines). For pure rotation, the phone is located at roughly the same position, but rotated around various axes (not just X, Y, and Z). For instance, the phone could be rotated around an axis defined by the vector $\vec{V} = \vec{X} + \vec{Y}$). Finally,

for motion mixtures, we perform random actions involving both linear and rotational motion.

MUSE consistently performs well, while other techniques falter in some scenario or other. Complementary filter, for instance, gets affected in the presence of translational motion since gravity is polluted. Gyroscope integration incurs error when phone rotation is dominant. Finally, A^3 is better but still considerably worse than *MUSE* due to the lack of static moments for resetting orientation. In other words, the strong trust in the magnetometer serves *MUSE* well in estimating the phone orientation.

Figure 4.7 compares the overall CDF of 3D orientation error across all controlled scenarios. While A^3 performs better than GyroOnly and ComplemFilter, *MUSE* exhibits a consistent improvement over A^3 (3.5X gain at median and 4X gain at 90th percentile), closing the gaps for IMU based motion tracking.

Natural Human and Object Motions

Figure 4.8 plots orientation error across various natural movements of humans and objects. For instance, an object motion like "Tennis" refers to the phone taped on the strings of the tennis racquet and a user pretending to play with it; the "Ball" refers to a user playing with a ball with a phone tightly pasted to it. Evident from Figure 4.8, *MUSE* outperforms the other methods almost across all activities. A^3 is comparable when the motion is naturally slow such that the accelerometer pollution is not excessive (e.g., eating) or when the motion has natural pauses (e.g., 3D mouse), but falls short when the motion is continuous and without stops. On average, *MUSE* achieves 2.9X smaller orientation error than A^3. As for GyroOnly and ComplemFilter, the mixture of linear and rotational motion in natural activities affects their performance.

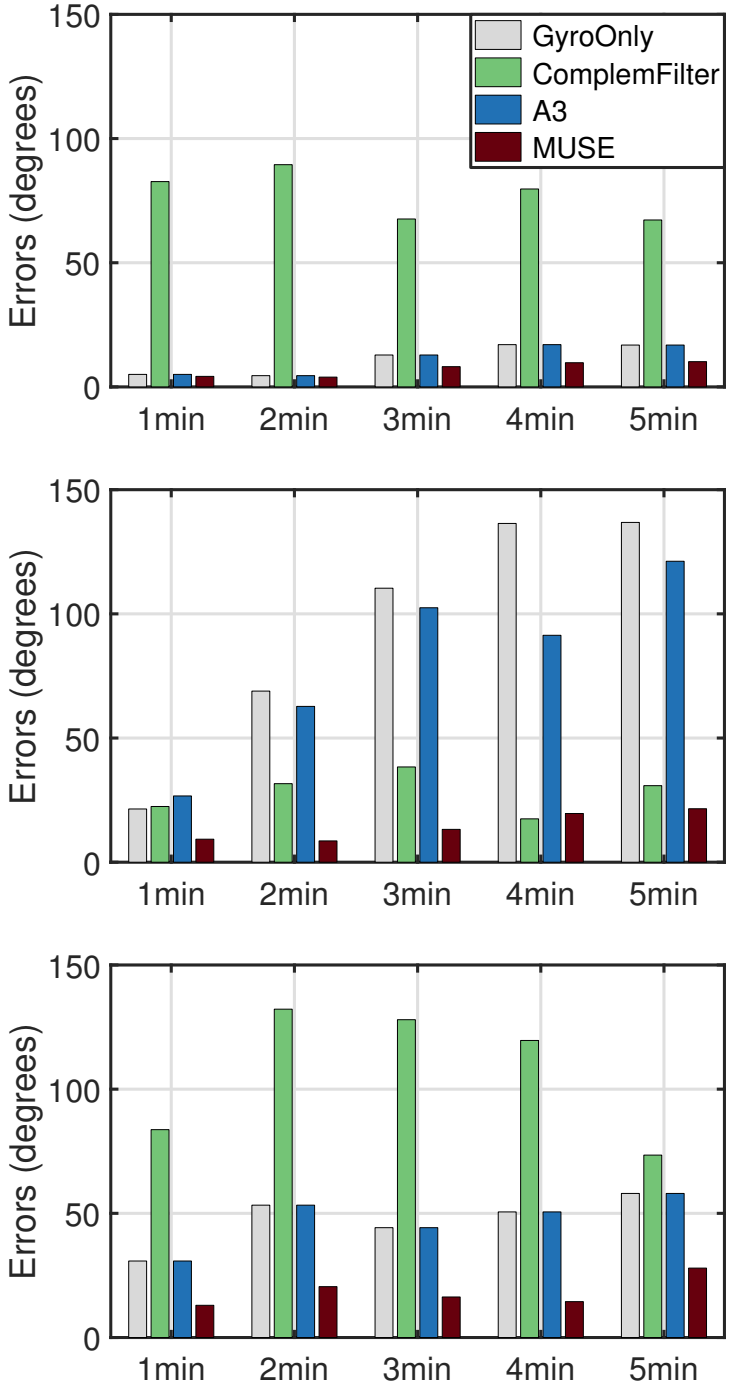


Figure 4.6: Orientation tracking error for three basic controlled motions: (a) pure translation, (b) pure rotation, and (c) mixtures of translational and rotational motions.

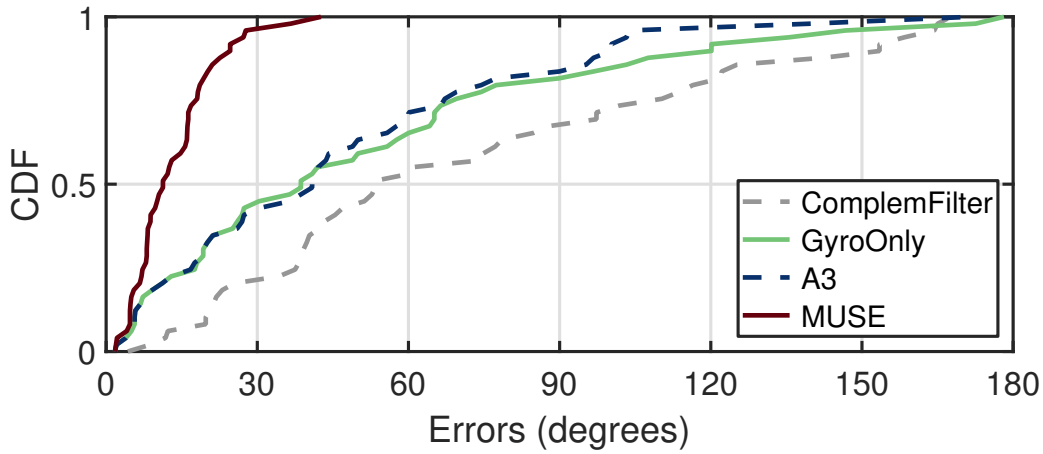


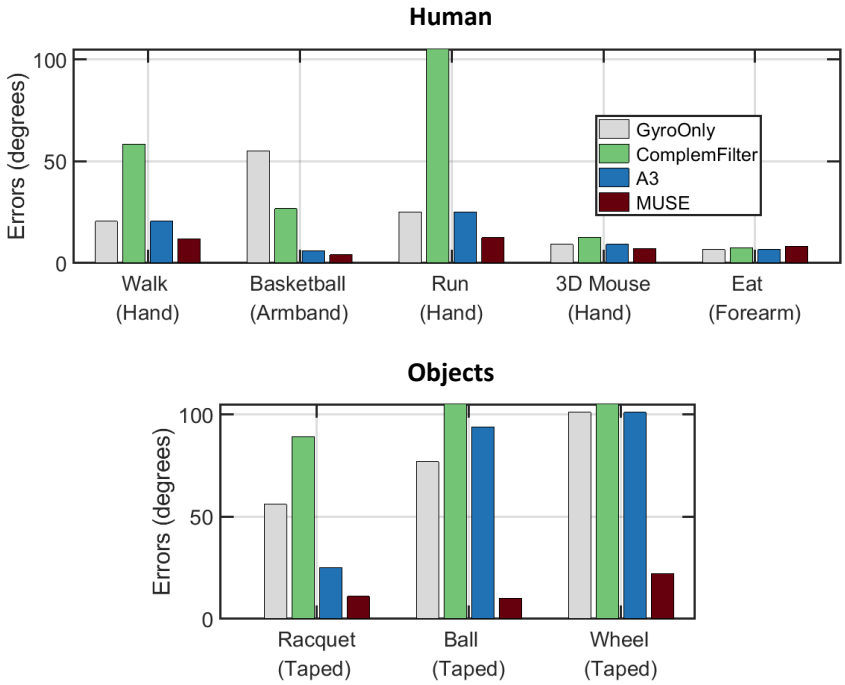
Figure 4.7: Overall orientation error CDF for controlled motions.

Finally, Figure 4.9 plots a sample trace of the 3D orientation error over time to demonstrate how the error can grow if A^3 is unable to find adequately frequent pause moments, or makes mistakes in identifying them. In contrast, *MUSE* maintains a low error, mainly due to magnetometer noise and the gyroscope's drift.

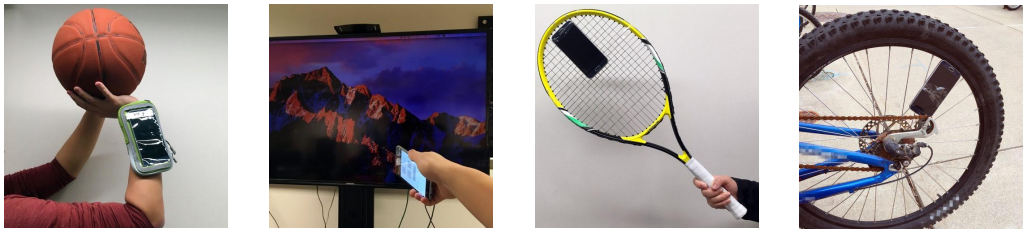
4.4.3 When Will MUSE Fail?

- **Rotation Only in 3rd DoF:** Utilizing the 3D magnetic North vector as an anchor, *MUSE* provides an overdetermined system in 2 of the 3 DoFs of orientation. This means that *MUSE* will not be useful, if the object's rotational motion happens to be only in the 3rd DoF, i.e. the object is exactly rotating around the global anchor direction (and will not change its axis of rotation thereafter). Of course, this rarely happens in practice, and even if it does, any static recalibration opportunity will mitigate this problem.

- **No Opportunity for Initialization:** While methods such as A^3 rely on frequent "pauses", *MUSE* gets rid of this assumption but still needs one static moment to bootstrap the tracking. If there is not even a single pause or slow-moving moment from the beginning, then *MUSE*



(a) Orientation error across different activities.



(b) Phone position for some of the activities.

Figure 4.8: (a) Orientation error across natural activities, including both human and object motions. (b) The phone position for some of the motion experiments.

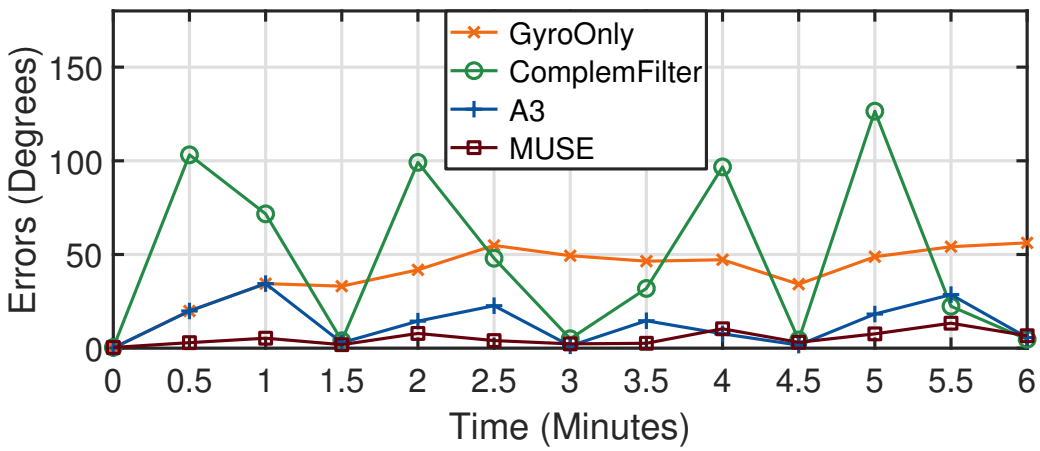


Figure 4.9: A sample trace of orientation error over time for different techniques. Intermediate ground-truth probing happens every half minute, and the device's motion pauses on purpose every 1.5 minutes (for recalibration).

will not be able to find an opportunity to compute initial orientation and the 3D magnetic North anchor.

- **Ferromagnetic Materials:** Since *MUSE* relies on the magnetic North vector as a global anchor, it would degrade in performance when the magnetic interference is strong (which is also the case for other systems). While deeper treatment is necessary in the future, we bring *MUSE* to more challenging environments to test its sensitivity to ferromagnetic materials.

Figure 4.10 shows the orientation tracking accuracy when we run *MUSE* at different places, ranging from outdoor and large indoor open space (with least interference), to crowded engineering buildings and labs with lots of computers and cables (with strongest interference). Each dot in the figure represents one trial, and its X value describes how fluctuated the magnetic field is. We measure the X value by moving and rotating the device around in this area, and taking the standard deviation of the magnitudes of magnetometer measurements. Clearly, *MUSE*'s performance decreases as the magnetic field density variation

increases. Techniques such as magnetic field profiling may help mitigate this issue – a topic we leave to future work.

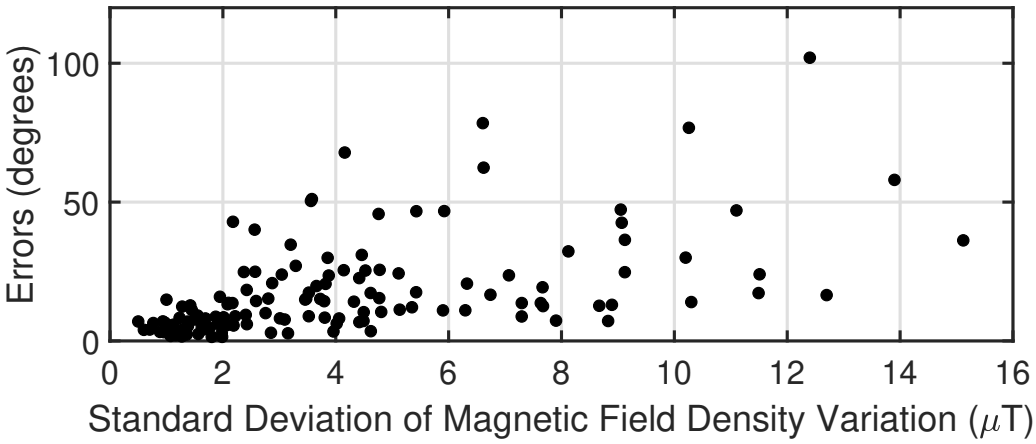


Figure 4.10: *MUSE*'s tracking accuracy as it runs at different places with varying levels of magnetic field fluctuations.

4.5 Points of Discussion

MUSE leaves room for further investigation, as discussed briefly in the interest of space:

- **Optimality:** We have not been able to comment on the optimality of IMU based tracking. This needs a deeper signal processing treatment via models of random sensor noise (and bias). Of course, when ignoring noise, we know that 3D orientation is solvable given at least 5 DoFs of information (2 from magnetometer and 3 from gyroscope). Location is also solvable, although more sensitive since the system is just adequately determined (3 DoFs of accelerometer for 3D location). For real systems, however, the interplay of hardware noise and restrictions of motion models will together determine the system's error. We leave this analysis to future work.

- **Running in Real Time:** *MUSE* runs a lightweight complementary filter, which has less computational complexity than related works that

can already run on mobile and wearable devices in real time [142, 143, 144].

4.6 Related Work

- **IMU Orientation Estimation:** As mentioned earlier, this problem has been well studied in aerodynamics and robotics, and various algorithms have been proposed to derive efficient sensor fusion algorithms under specific error models [136, 137]. While some of these algorithms [145, 142, 146, 147, 131, 148, 149, 132] also use all the available information of the magnetometer, they assume that the object's motion is slow or has intermittent stops. Table 4.1 summarizes some of the important related works, and classifies them based on their sensor fusion techniques, and the key assumptions they have made. These assumptions include:

  (A): The linear motion is slow so that the average of accelerometer is gravity.

  (B): The rotational motion is slow and the error model is Gaussian, in order to preserve the linearity of the system model.

  (C): The motion has frequent pauses (static moments) for resetting the gravity estimation.

  However, these assumptions may easily break down for continuous motion from human-wearable devices.

- **Other Tracking Modalities:** Many other modalities can track the position and motion of objects, including IR technology [165, 166], computer vision [167, 168], wireless sensing [169, 170, 171, 172, 173, 174], RFID [175, 176], visible light [177], acoustics [54, 55, 178], etc.

Table 4.1: Related works on IMU orientation tracking, classified based on their sensor fusion techniques and key assumptions made.

Related Works	Sensor Fusion Techniques	Key Assumptions
[134, 132, 150, 151, 133, 135]	Complementary Filter	(A)
[152, 153, 154, 155, 145]	Kalman Filter	(A) (B)
[156, 157]	Kalman Filter	(B) (C)
[158, 159, 160, 161, 162, 130, 146, 147, 131]	EKF Filter	(A) (B)
[129]	EKF Filter	(B) (C)
[148, 163, 149]	UKF Filter	(A)
[142]	Opportunistic Replacement	(C)
[164] (Android APIs)	Gravity+Compass Only	(A)

However, the core inertial nature of the MEMS IMU sensors presents unique challenges distinct from other sensing modalities.

- **3D Acoustics**: The fact that the human brain is capable of resolving the direction of the incoming sound is well known [179]. Past works have utilized this binaural effect for different purposes, including sound recording and reproduction [180, 181, 182], entertainment [183, 184, 185], and localization [186, 187, 188]. We instead track head motion to adapt 3D sound in real-time, so as to enable an immersive acoustic AR experience.

4.7 Conclusion

This chapter shows improvements to orientation tracking by recognizing that magnetometers, unlike accelerometers, are unpolluted by object motion. We believe this will benefit a range of systems and applications, including designing 3D directional acoustics.

CHAPTER 5

CONCLUSION

This dissertation shows various unconventional ways to exploit propagation delay for acoustic techniques and applications. For example, cross-frequency propagation delay can be exploited for acoustic noise cancellation; cross-time multipath delays can be utilized to compute sound source location; and finally, synthesized propagation delays at two ears can be leveraged to enable 3D directional sound.

We believe the space of acoustics and sensing is opening up, mainly for two reasons. First, human beings are now communicating with machines through acoustics, thanks to the advance in speech recognition and natural language processing. Second, many IoT devices, including microphones and speakers, are becoming cheap and scattered around us. The combination of these two is opening up space of new opportunities, which in turn will open new applications. This dissertation, which leverages propagation delay to enable acoustic systems, is only the starting point.

REFERENCES

[1] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*. ACM, 2000, pp. 32–43.

[2] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*. ACM, 2001, pp. 166–179.

[3] M. Minami, Y. Fukuju, K. Hirasawa, S. Yokoyama, M. Mizumachi, H. Morikawa, and T. Aoyama, "DOLPHIN: A practical approach for implementing a fully distributed indoor ultrasonic positioning system," in *International Conference on Ubiquitous Computing*. Springer, 2004, pp. 347–365.

[4] Q. Wang, R. Zheng, A. Tirumala, X. Liu, and L. Sha, "Lightning: A hard real-time, fast, and lightweight low-end wireless sensor election protocol for acoustic event localization," *IEEE Transactions on Mobile Computing*, vol. 7, no. 5, pp. 570–584, 2008.

[5] "Earthquake early warning," 2018. [Online]. Available: https://earthquake.usgs.gov/research/earlywarning/nextsteps.php

[6] "Deep ocean tsunami detection buoys," 2019. [Online]. Available: http://www.bom.gov.au/tsunami/about/detection_buoys.shtml

[7] D. Wells, N. Beck, A. Kleusberg, E. J. Krakiwsky, G. Lachapelle, R. B. Langley, K.-p. Schwarz, J. M. Tranquilla, P. Vanicek, and D. Delikaraoglou, "Guide to GPS positioning," in *Canadian GPS Assoc.* Citeseer, 1987.

[8] J. Benesty, J. Chen, and Y. Huang, *Microphone Array Signal Processing*. Springer Science & Business Media, 2008.

[9] J. Hecht, "Lidar for self-driving cars," *Optics and Photonics News*, vol. 29, no. 1, pp. 26–33, 2018.

[10] "More than 100 million Alexa devices have been sold," 2019. [Online]. Available: https://techcrunch.com/2019/01/04/more-than-100-million-alexa-devices-have-been-sold/

[11] "Qualcomm: Smart audio products and high quality audio technology," 2019. [Online]. Available: https://www.qualcomm.com/products/audio

[12] "Nura: Headphones that learn and adapt to your unique hearing," 2019. [Online]. Available: https://www.kickstarter.com/projects/nura/nura-headphones-that-learn-and-adapt-to-your-uniqu

[13] M. Klatte, K. Bergström, and T. Lachmann, "Does noise affect learning? A short review on noise effects on cognitive performance in children," *Frontiers in Psychology*, vol. 4, p. 578, 2013.

[14] S. A. Stansfeld, B. Berglund, C. Clark, I. Lopez-Barrio, P. Fischer, E. Öhrström, M. M. Haines, J. Head, S. Hygge, and I. Van Kamp, "Aircraft and road traffic noise and children's cognition and health: A cross-national study," *The Lancet*, vol. 365, no. 9475, pp. 1942–1949, 2005.

[15] "Guidelines for community noise," 2019. [Online]. Available: http://www.who.int/docstore/peh/noise/introduction.htm

[16] J. Mobley, C. Zhang, S. Soli, C. Johnson, and D. O'Connell, "Pressure-regulating ear plug," Oct. 13, 1998, US Patent 5,819,745. [Online]. Available: https://www.google.com/patents/US5819745

[17] J. G. Casali and J. F. Grenell, "Noise-attenuating earmuff comfort: A brief review and investigation of band-force, cushion, and wearing-time effects," *Applied Acoustics*, vol. 29, no. 2, pp. 117–138, 1990.

[18] S. N. Gerges, "Earmuff comfort," *Applied Acoustics*, vol. 73, no. 10, pp. 1003–1012, 2012.

[19] S. Shen, N. Roy, J. Guan, H. Hassanieh, and R. R. Choudhury, "MUTE: Bringing IoT to noise cancellation," in *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*. ACM, 2018, pp. 282–296.

[20] L. P. Heck, M. Chinthakunta, D. Mitby, and L. Stifelman, "Location based conversational understanding," Jan. 26, 2016, US Patent 9,244,984.

[21] X. Lei, G.-H. Tu, A. X. Liu, C.-Y. Li, and T. Xie, "The insecurity of home digital voice assistants-vulnerabilities, attacks and countermeasures," in *2018 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 2018, pp. 1–9.

[22] "'Take your pill!' Ultra-precise Dot beacons trigger alerts that matter," 2019. [Online]. Available: https://finance.yahoo.com/news/pill-ultra-precise-dot-beacons-181110596.html

[23] "Beco focuses on developing a spatially-aware Alexa skill," 2019. [Online]. Available: https://developer.amazon.com/blogs/alexa/post/Tx1BPHXBLZV5ZVN/beco-focuses-on-developing-a-spatially-aware-alexa-skill

[24] "Alexa, lights! How I turned my home into a sci-fi dream," 2019. [Online]. Available: https://www.theguardian.com/technology/2016/dec/23/alexa-lights-how-i-turned-my-home-into-a-sci-fi-dream

[25] "How do you organize your devices? Naming conventions? (2018)," 2018. [Online]. Available: https://community.smartthings.com/t/how-do-you-organize-your-devices-naming-conventions-2018/121688

[26] "Siri gets smarter at recognizing names of local businesses and attractions," 2018. [Online]. Available: https://voicebot.ai/2018/08/13/siri-gets-smarter-at-recognizing-names-of-local-businesses-and-attractions/

[27] T. N. Sainath, R. J. Weiss, K. W. Wilson, A. Narayanan, M. Bacchiani et al., "Speaker location and microphone spacing invariant acoustic modeling from raw multichannel waveforms," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2015, pp. 30–36.

[28] L. Lugosch, M. Ravanelli, P. Ignoto, V. S. Tomar, and Y. Bengio, "Speech model pre-training for end-to-end spoken language understanding," *arXiv Preprint arXiv:1904.03670*, 2019.

[29] S. Shen, M. Gowda, and R. Roy Choudhury, "Closing the gaps in inertial motion tracking," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. ACM, 2018, pp. 429–444.

[30] "Bose QuiteComfort 35 wireless noise cancelling headphone," 2019. [Online]. Available: https://www.bose.com/en_us/products/headphones/over_ear_headphones/quietcomfort-35-wireless-ii.html

[31] "Bose QuietControl 30 (QC30) wireless noise cancelling earbuds," 2019. [Online]. Available: https://www.bose.com/en_us/products/headphones/earphones/quietcontrol-30.html

[32] "Sony MDR-1000X bluetooth over-ear noise canceling headphone," 2019. [Online]. Available: https://www.sony.com/electronics/headband-headphones/mdr-1000x

[33] "Wireless noise cancelling headphones SHB8850NC/27 — Philips," 2019. [Online]. Available: https://www.download.p4c.philips.com/files/s/shb8850nc_00/shb8850nc_00_dfu_eng.pdf

[34] "Here active listening - change the way you hear the world by Doppler Labs," 2019. [Online]. Available: https://www.kickstarter.com/projects/dopplerlabs/here-active-listening-change-the-way-you-hear-the/description

[35] "Review and measurements: Bose QC25 noise-cancelling headphone," 2017. [Online]. Available: https://www.lifewire.com/bose-qc25-review-specs-3134560

[36] B. Rudzyn and M. Fisher, "Performance of personal active noise reduction devices," *Applied Acoustics*, vol. 73, no. 11, pp. 1159–1167, 2012.

[37] Y.-L. Hsu, C.-C. Huang, C.-Y. Yo, C.-J. Chen, and C.-M. Lien, "Comfort evaluation of hearing protection," *International Journal of Industrial Ergonomics*, vol. 33, no. 6, pp. 543–551, 2004.

[38] S. Haykin and B. Widrow, *Least-Mean-Square Adaptive Filters*. John Wiley & Sons, 2003.

[39] E. Bjarnason, "Analysis of the filtered-X LMS algorithm," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 6, pp. 504–514, 1995.

[40] J. N. Mourjopoulos, "Digital equalization of room acoustics," *Journal of the Audio Engineering Society*, vol. 42, no. 11, pp. 884–900, 1994.

[41] D. G. Manolakis and V. K. Ingle, *Applied Digital Signal Processing: Theory and Practice*. Cambridge University Press, 2011.

[42] S. T. Neely and J. B. Allen, "Invertibility of a room impulse response," *The Journal of the Acoustical Society of America*, vol. 66, no. 1, pp. 165–169, 1979.

[43] B. D. Radlovic and R. A. Kennedy, "Nonminimum-phase equalization and its subjective importance in room acoustics," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 6, pp. 728–737, 2000.

[44] M. S. Brandstein and H. F. Silverman, "A robust method for speech signal time-delay estimation in reverberant rooms," in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, vol. 1. IEEE, 1997, pp. 375–378.

[45] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Computing*, vol. 8, no. 4, 2009.

[46] "TMS320C6713 DSP starter kit," 2019. [Online]. Available: http://www.ti.com/tool/TMDSDSK6713

[47] "KEMAR head and torso simulators, for ear and headphone test, 2 channels," 2019. [Online]. Available: https://www.gras.dk/products/head-torso-simulators-kemar/kemar-for-ear-headphone-test-2-ch

[48] S. M. Kuo and D. Morgan, *Active Noise Control Systems: Algorithms and DSP Implementations*. John Wiley & Sons, Inc., 1995.

[49] Y. Song, Y. Gong, and S. M. Kuo, "A robust hybrid feedback active noise cancellation headset," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 4, pp. 607–617, 2005.

[50] S. M. Kuo, S. Mitra, and W.-S. Gan, "Active noise control system for headphone applications," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 2, pp. 331–335, 2006.

[51] W. S. Gan and S. M. Kuo, "An integrated audio and active noise control headset," *IEEE Transactions on Consumer Electronics*, vol. 48, no. 2, pp. 242–247, 2002.

[52] S. J. Elliott and P. A. Nelson, "Active noise control," *IEEE Signal Processing Magazine*, vol. 10, no. 4, pp. 12–35, 1993.

[53] L. Eriksson, "Development of the filtered-U algorithm for active noise control," *The Journal of the Acoustical Society of America*, vol. 89, no. 1, pp. 257–265, 1991.

[54] W. Mao, J. He, and L. Qiu, "CAT: High-precision acoustic motion tracking," in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. ACM, 2016, pp. 69–81.

[55] S. Yun, Y.-C. Chen, and L. Qiu, "Turning a mobile device into a mouse in the air," in *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2015, pp. 15–29.

[56] N. Roy, S. Shen, H. Hassanieh, and R. R. Choudhury, "Inaudible voice commands: The long-range attack and defense," in *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*, 2018, pp. 547–560.

[57] S. J. Elliott, "Distributed control of sound and vibration," *Noise Control Engineering Journal*, vol. 53, no. 5, pp. 165–180, 2005.

[58] V. Kulathumani and B. Lemon, "Sufficiency of local feedback for sensor-actuator network-based control systems with distance sensitivity properties," *Journal of Sensor and Actuator Networks*, vol. 2, no. 3, pp. 409–423, 2013.

[59] K. D. Frampton, "Advantages and challenges of distributed active vibro-acoustic control," *The Journal of the Acoustical Society of America*, vol. 118, no. 3, pp. 1950–1950, 2005.

[60] K. D. Frampton, "The control of rocket fairing interior noise with a networked embedded system," *The Journal of the Acoustical Society of America*, vol. 113, no. 4, pp. 2251–2251, 2003.

[61] K. D. Frampton, "Decentralized vibration control in a launch vehicle payload fairing," in *ASME 2002 International Mechanical Engineering Congress and Exposition*. American Society of Mechanical Engineers, 2002, pp. 155–160.

[62] M. Ferrer, M. de Diego, G. Piñero, and A. Gonzalez, "Active noise control over adaptive distributed networks," *Signal Processing*, vol. 107, pp. 82–95, 2015.

[63] "Cadillac ELR takes active noise cancelling to the limit," 2013. [Online]. Available: http://media.gm.com/media/us/en/cadillac/vehicles/elr/2014.detail.html/content/Pages/news/us/en/2013/Oct/1017-elr-anc.html

[64] "Honda's active noise cancellation," 2014. [Online]. Available: https://www.honda.co.nz/technology/driving/anc/

[65] "2017 Buick LaCrosse Quiet Tuning," 2016. [Online]. Available: http://media.buick.com/media/us/en/buick/bcportal.html/currentVideoId/5161313633001/currentChannelId/Running%20Footage/Cars.gsaOff.html

[66] "Flybe takes delivery of its 50th Bombardier Q400 airliner," 2019. [Online]. Available: https://www.bombardier.com/en/media/newsList/details.942-flybe-takes-delivery-of-its-50th-bombardier-q400-airliner.bombardiercom.html

[67] "Bombardier Q300 DHC-8 Dash 8," 2018. [Online]. Available: https://www.aerospace-technology.com/projects/bombardier_q300/

[68] L. Liu and S. M. Kuo, "Wireless communication integrated active noise control system for infant incubators," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 375–378.

[69] "WS800 wireless microphone system," 2018. [Online]. Available: http://www.clearone.com/products_wireless_microphone_system

[70] "Walkie-talkie," 2018. [Online]. Available: https://en.wikipedia.org/wiki/Walkie-talkie

[71] "Wireless microphone," 2018. [Online]. Available: https://en.wikipedia.org/wiki/Wireless_microphone

[72] "Amazon files for patent to detect user illness and emotional state by analyzing voice data," 2019. [Online]. Available: https://voicebot.ai/2018/10/10/amazon-files-for-patent-to-detect-illness-by-analyzing-voice-data/

[73] "Here are the upcoming Alexa features we're most excited about," 2019. [Online]. Available: https://thenextweb.com/artificial-intelligence/2018/09/20/here-are-the-upcoming-alexa-features-were-most-excited-about/

[74] H. Jin and S. Wang, "Voice-based determination of physical and emotional characteristics of users," Oct. 9, 2018, US Patent App. 15/457,846.

[75] G. Taubman and B. Strope, "Speech recognition models based on location indicia," Sep. 9, 2014, US Patent 8,831,957.

[76] I. Dokmanić, R. Parhizkar, A. Walther, Y. M. Lu, and M. Vetterli, "Acoustic echoes reveal room shape," *Proceedings of the National Academy of Sciences*, vol. 110, no. 30, pp. 12 186–12 191, 2013.

[77] T. C. Collier, A. N. Kirschel, and C. E. Taylor, "Acoustic localization of antbirds in a Mexican rainforest using a wireless sensor network," *The Journal of the Acoustical Society of America*, vol. 128, no. 1, pp. 182–189, 2010.

[78] J. Sallai, W. Hedgecock, P. Volgyesi, A. Nadas, G. Balogh, and A. Ledeczi, "Weapon classification and shooter localization using distributed multichannel acoustic sensors," *Journal of Systems Architecture*, vol. 57, no. 10, pp. 869–885, 2011.

[79] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Communications Surveys and Tutorials*, 2019.

[80] C. T. Ishi, J. Even, and N. Hagita, "Using multiple microphone arrays and reflections for 3d localization of sound sources," *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3937–3942, 2013.

[81] J. Xiong and K. Jamieson, "Arraytrack: A fine-grained indoor location system," in *Presented as part of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*, 2013, pp. 71–84.

[82] S. Sen, R. R. Choudhury, B. Radunovic, and T. Minka, "Precise indoor localization using phy layer information," in *Proceedings of the 10th ACM Workshop on Hot Topics in Networks*. ACM, 2011, p. 18.

[83] I. Dokmanic and M. Vetterli, "Listening to distances and hearing shapes: Inverse problems in room acoustics and beyond," *EPFL, Lausanne*, 2015.

[84] F. P. Ribeiro, D. E. Ba, C. Zhang, and D. A. F. Florêncio, "Turning enemies into friends: Using reflections to improve sound source localization," *2010 IEEE International Conference on Multimedia and Expo*, pp. 731–736, 2010.

[85] I. Constandache, S. Agarwal, I. Tashev, and R. R. Choudhury, "Daredevil: Indoor location using sound," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 18, no. 2, pp. 9–19, 2014.

[86] "Qualcomm delivers a fully integrated far-field smart audio reference platform for the Amazon Alexa voice service," 2018. [Online]. Available: https://www.qualcomm.com/news/releases/2018/01/08/qualcomm-delivers-fully-integrated-far-field-smart-audio-reference-platform

[87] "Audio hardware configurations," 2018. [Online]. Available: https://developer.amazon.com/docs/alexa-voice-service/audio-hardware-configurations.html

[88] J. Heymann, M. Bacchiani, and T. N. Sainath, "Performance of mask based statistical beamforming in a smart home scenario," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 6722–6726.

[89] A. Chhetri, P. Hilmes, T. Kristjansson, W. Chu, M. Mansour, X. Li, and X. Zhang, "Multichannel audio front-end for far-field automatic speech recognition," in *2018 26th European Signal Processing Conference (EUSIPCO)*, Sep. 2018, pp. 1527–1531.

[90] T.-J. Shan, M. Wax, and T. Kailath, "On spatial smoothing for direction-of-arrival estimation of coherent signals," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, no. 4, pp. 806–811, 1985.

[91] S. Kumar, S. Gil, D. Katabi, and D. Rus, "Accurate indoor localization with zero start-up cost," in *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*. ACM, 2014, pp. 483–494.

[92] K. M. Varma, "Time delay estimate based direction of arrival estimation for speech in reverberant environments," Ph.D. dissertation, Virginia Tech, 2002.

[93] J. H. DiBiase, *A High-Accuracy, Low-Latency Technique for Talker Localization in Reverberant Environments Using Microphone Arrays*. Brown University Providence, RI, 2000.

[94] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 4, pp. 320–327, 1976.

[95] B. Van Den Broeck, A. Bertrand, P. Karsmakers, B. Vanrumste, M. Moonen et al., "Time-domain generalized cross correlation phase transform sound source localization for small microphone arrays," in *2012 5th European DSP Education and Research Conference (EDERC)*. IEEE, 2012, pp. 76–80.

[96] J. Benesty, J. Chen, and Y. Huang, "Time-delay estimation via linear interpolation and cross correlation," *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 5, pp. 509–519, 2004.

[97] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[98] H. Tang, "DOA estimation based on music algorithm," Linnaeus University, Department of Physics and Electrical Engineering, Tech. Rep., 2014.

[99] R. Roy and T. Kailath, "ESPRIT-estimation of signal parameters via rotational invariance techniques," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 7, pp. 984–995, 1989.

[100] S. Shahbazpanahi, S. Valaee, and M. H. Bastani, "Distributed source localization using esprit algorithm," *IEEE Transactions on Signal Processing*, vol. 49, no. 10, pp. 2169–2178, 2001.

[101] I. L. Freire et al., "DOA of gunshot signals in a spatial microphone array: Performance of the interpolated generalized cross-correlation method," in *2011 Argentine School of Micro-Nanoelectronics, Technology and Applications*. IEEE, 2011, pp. 1–6.

[102] Y. L. Sit, C. Sturm, J. Baier, and T. Zwick, "Direction of arrival estimation using the MUSIC algorithm for a MIMO OFDM radar," in *2012 IEEE Radar Conference*. IEEE, 2012, pp. 0226–0229.

[103] I. An, M. Son, D. Manocha, and S.-E. Yoon, "Reflection-aware sound source localization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 66–73.

[104] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *The Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 1979.

[105] "ReSpeaker 6-mic circular array kit for Raspberry Pi," 2019. [Online]. Available: http://wiki.seeedstudio.com/ReSpeaker_6-Mic_Circular_Array_kit_for_Raspberry_Pi/

[106] P. Lazik, N. Rajagopal, O. Shih, B. Sinopoli, and A. Rowe, "ALPS: The acoustic location processing system," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2015, pp. 491–492.

[107] F. J. Álvarez, T. Aguilera, and R. López-Valcarce, "CDMA-based acoustic local positioning system for portable devices with multipath cancellation," *Digital Signal Processing*, vol. 62, pp. 38–51, 2017.

[108] T. Aguilera, J. A. Paredes, F. J. Álvarez, J. I. Suárez, and A. Hernandez, "Acoustic local positioning system using an iOS device," in *International Conference on Indoor Positioning and Indoor Navigation*. IEEE, 2013, pp. 1–8.

[109] J. N. Moutinho, R. E. Araújo, and D. Freitas, "Indoor localization with audible sound - towards practical implementation," *Pervasive and Mobile Computing*, vol. 29, pp. 1–16, 2016.

[110] Y. Michalevsky, A. Schulman, G. A. Veerapandian, D. Boneh, and G. Nakibly, "Powerspy: Location tracking using mobile device power analysis," in *24th USENIX Security Symposium (USENIX Security 15)*, 2015, pp. 785–800.

[111] P. Bahl and V. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No. 00CH37064)*, vol. 2. IEEE, 2000, pp. 775–784.

[112] R. Nandakumar, K. K. Chintalapudi, V. Padmanabhan, and R. Venkatesan, "Dhwani: Secure peer-to-peer acoustic NFC," in *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4. ACM, 2013, pp. 63–74.

[113] X. Zhang, L. Zhong, and A. Sabharwal, "Directional training for FDD massive MIMO," *IEEE Transactions on Wireless Communications*, vol. 17, no. 8, pp. 5183–5197, 2018.

[114] D. Bharadia, E. McMilin, and S. Katti, "Full duplex radios," in *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4. ACM, 2013, pp. 375–386.

[115] E. Georganti, T. May, S. van de Par, A. Harma, and J. Mourjopoulos, "Speaker distance detection using a single microphone," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 1949–1961, 2011.

[116] P. Svaizer, A. Brutti, and M. Omologo, "Environment aware estimation of the orientation of acoustic sources using a line array," in *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*. IEEE, 2012, pp. 1024–1028.

[117] K. Nakadai, H. Nakajima, K. Yamada, Y. Hasegawa, T. Nakamura, and H. Tsujino, "Sound source tracking with directivity pattern estimation using a 64 ch microphone array," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2005, pp. 1690–1696.

[118] H. Nakajima, K. Kikuchi, T. Daigo, Y. Kaneda, K. Nakadai, and Y. Hasegawa, "Real-time sound source orientation estimation using a 96 channel microphone array," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 676–683.

[119] K. Niwa, Y. Hioka, S. Sakauchi, K. Furuya, and Y. Haneda, "Estimation of sound source orientation using eigenspace of spatial correlation matrix," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2010, pp. 129–132.

[120] A. Brutti, M. Omologo, and P. Svaizer, "Inference of acoustic source directivity using environment awareness," in *2011 19th European Signal Processing Conference*. IEEE, 2011, pp. 151–155.

[121] A. L. Ramos, S. Holm, S. Gudvangen, and R. Otterlei, "Delay-and-sum beamforming for direction of arrival estimation applied to gunshot acoustics," in *Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense X*, vol. 8019. International Society for Optics and Photonics, 2011, p. 80190U.

[122] G. W. Elko, "Microphone array systems for hands-free telecommunication," *Speech Communication*, vol. 20, no. 3-4, pp. 229–240, 1996.

[123] Y. Lin, J. Chen, Y. Kim, and D. D. Lee, "Blind sparse-nonnegative (BSN) channel identification for acoustic time-difference-of-arrival estimation," in *2007 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. IEEE, 2007, pp. 106–109.

[124] Y. Lin, J. Chen, Y. Kim, and D. D. Lee, "Blind channel identification for speech dereverberation using l1-norm sparse learning," in *Advances in Neural Information Processing Systems*, 2008, pp. 921–928.

[125] K. Kowalczyk, E. A. Habets, W. Kellermann, and P. A. Naylor, "Blind system identification using sparse learning for TDOA estimation of room reflections," *IEEE Signal Processing Letters*, vol. 20, no. 7, pp. 653–656, 2013.

[126] M. Crocco and A. Del Bue, "Room impulse response estimation by iterative weighted l1-norm," in *2015 23rd European Signal Processing Conference (EUSIPCO)*. IEEE, 2015, pp. 1895–1899.

[127] S. Gollakota and D. Katabi, *Zigzag Decoding: Combating Hidden Terminals in Wireless Networks*. ACM, 2008, vol. 38, no. 4.

[128] "Grush smart oral care," 2019. [Online]. Available: https://www.grushgamer.com/

[129] B. Barshan and H. F. Durrant-Whyte, "Inertial navigation systems for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 3, pp. 328–342, 1995.

[130] J. L. Marins, X. Yun, E. R. Bachmann, R. B. McGhee, and M. J. Zyda, "An extended Kalman filter for quaternion-based orientation estimation using MARG sensors," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, vol. 4. IEEE, 2001, pp. 2003–2011.

[131] A. M. Sabatini, "Quaternion-based extended Kalman filter for determining orientation by inertial and magnetic sensing," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 7, pp. 1346–1356, 2006.

[132] S. O. Madgwick, A. J. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," in *2011 IEEE International Conference on Rehabilitation Robotics*. IEEE, 2011, pp. 1–7.

[133] H.-J. Lee and S. Jung, "Gyro sensor drift compensation by Kalman filter to control a mobile inverted pendulum robot system," in *2009 IEEE International Conference on Industrial Technology*. IEEE, 2009, pp. 1–6.

[134] M. Euston, P. Coote, R. Mahony, J. Kim, and T. Hamel, "A complementary filter for attitude estimation of a fixed-wing UAV," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 340–345.

[135] E. R. Bachmann, R. B. McGhee, X. Yun, and M. J. Zyda, "Inertial and magnetic posture tracking for inserting humans into networked virtual environments," in *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*. ACM, 2001, pp. 9–16.

[136] J. L. Crassidis, F. L. Markley, and Y. Cheng, "Survey of nonlinear attitude estimation methods," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 1, pp. 12–28, 2007.

[137] H. Chao, C. Coopmans, L. Di, and Y. Chen, "A comparative evaluation of low-cost IMUs for unmanned autonomous systems," in *2010 IEEE Conference on Multisensor Fusion and Integration*. IEEE, 2010, pp. 211–216.

[138] S. Shen, "Arm posture tracking using a smartwatch," in *Proceedings of MobiSys 2016 PhD Forum*. ACM, 2016, pp. 9–10.

[139] S. Shen, H. Wang, and R. Roy Choudhury, "I am a smartwatch and I can track my user's arm," in *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2016, pp. 85–96.

[140] M. Gowda, A. Dhekne, S. Shen, R. R. Choudhury, S. X. Yang, L. Yang, S. Golwalkar, and A. Essanian, "IoT platform for sports analytics," *GetMobile: Mobile Computing and Communications*, vol. 21, no. 4, pp. 8–14, 2018.

[141] M. Gowda, A. Dhekne, S. Shen, R. R. Choudhury, L. Yang, S. Golwalkar, and A. Essanian, "Bringing IoT to sports analytics," in *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*, 2017, pp. 499–513.

[142] P. Zhou, M. Li, and G. Shen, "Use it free: Instantly knowing your phone attitude," in *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*. ACM, 2014, pp. 605–616.

[143] J. Gośliński, M. Nowicki, and P. Skrzypczyński, "Performance comparison of EKF-based algorithms for orientation estimation on Android platform," *IEEE Sensors Journal*, vol. 15, no. 7, pp. 3781–3792, 2015.

[144] M. Nowicki, J. Wietrzykowski, and P. Skrzypczyński, "Simplicity or flexibility? Complementary filter vs. EKF for orientation estimation on mobile devices," in *2015 IEEE 2nd International Conference on Cybernetics (CYBCONF)*. IEEE, 2015, pp. 166–171.

[145] N. H. Q. Phuong, H.-J. Kang, Y.-S. Suh, and Y.-S. Ro, "A DCM based orientation estimation algorithm with an inertial measurement unit and a magnetic compass," *Journal of Universal Computer Science*, vol. 15, no. 4, pp. 859–876, 2009.

[146] E. Edwan, J. Zhang, J. Zhou, and O. Loffeld, "Reduced DCM based attitude estimation using low-cost IMU and magnetometer triad," in *2011 8th Workshop on Positioning, Navigation and Communication*. IEEE, 2011, pp. 1–6.

[147] H. F. Grip, T. I. Fossen, T. A. Johansen, and A. Saberi, "Globally exponentially stable attitude and gyro bias estimation with application to GNSS/INS integration," *Automatica*, vol. 51, pp. 158–166, 2015.

[148] M. Romanovas, L. Klingbeil, M. Trachtler, and Y. Manoli, "Efficient orientation estimation algorithm for low cost inertial and magnetic sensor systems," in *2009 IEEE/SP 15th Workshop on Statistical Signal Processing*. IEEE, 2009, pp. 586–589.

[149] B. Huyghe, J. Doutreloigne, and J. Vanfleteren, "3D orientation tracking based on unscented Kalman filtering of accelerometer and magnetometer data," in *2009 IEEE Sensors Applications Symposium*. IEEE, 2009, pp. 148–152.

[150] R. Mahony, T. Hamel, and J.-M. Pflimlin, "Nonlinear complementary filters on the special orthogonal group," *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1203–1218, 2008.

[151] S. Madgwick, "An efficient orientation filter for inertial and inertial/magnetic sensor arrays," *Report x-io and University of Bristol (UK)*, vol. 25, pp. 113–118, 2010.

[152] S. Yean, B. S. Lee, C. K. Yeo, and C. H. Vun, "Algorithm for 3D orientation estimation based on Kalman filter and gradient descent," in *2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. IEEE, 2016, pp. 1–6.

[153] V. Bistrov, "Performance analysis of alignment process of MEMS IMU," *International Journal of Navigation and Observation*, vol. 2012, 2012.

[154] L. Lou, X. Xu, J. Cao, Z. Chen, and Y. Xu, "Sensor fusion-based attitude estimation using low-cost MEMS-IMU for mobile robot navigation," in *2011 6th IEEE Joint International Information Technology and Artificial Intelligence Conference*, vol. 2. IEEE, 2011, pp. 465–468.

[155] W. Li and J. Wang, "Effective adaptive Kalman filter for MEMS-IMU/magnetometers integrated attitude and heading reference systems," *The Journal of Navigation*, vol. 66, no. 1, pp. 99–113, 2013.

[156] Z. Ercan, V. Sezer, H. Heceoglu, C. Dikilitas, M. Gokasan, A. Mugan, and S. Bogosyan, "Multi-sensor data fusion of DCM based orientation estimation for land vehicles," in *2011 IEEE International Conference on Mechatronics*. IEEE, 2011, pp. 672–677.

[157] D. Jurman, M. Jankovec, R. Kamnik, and M. Topič, "Calibration and data fusion solution for the miniature attitude and heading reference system," *Sensors and Actuators A: Physical*, vol. 138, no. 2, pp. 411–420, 2007.

[158] R. Zhu, D. Sun, Z. Zhou, and D. Wang, "A linear fusion algorithm for attitude determination using low cost MEMS-based sensors," *Measurement*, vol. 40, no. 3, pp. 322–328, 2007.

[159] E. Foxlin, "Inertial head-tracker sensor fusion by a complementary separate-bias Kalman filter," in *Proceedings of the IEEE 1996 Virtual Reality Annual International Symposium*. IEEE, 1996, pp. 185–194.

[160] R. Munguia and A. Grau, "Attitude and heading system based on EKF total state configuration," in *2011 IEEE International Symposium on Industrial Electronics*. IEEE, 2011, pp. 2147–2152.

[161] D. Gebre-Egziabher, R. C. Hayward, and J. D. Powell, "Design of multi-sensor attitude determination systems," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, no. 2, pp. 627–649, 2004.

[162] N. S. Kumar and T. Jann, "Estimation of attitudes from a low-cost miniaturized inertial platform using Kalman filter-based sensor fusion algorithm," *Sadhana*, vol. 29, no. 2, pp. 217–235, 2004.

[163] H. G. De Marina, F. J. Pereda, J. M. Giron-Sierra, and F. Espinosa, "UAV attitude estimation using unscented Kalman filter and TRIAD," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4465–4474, 2012.

[164] "Android developers: position sensors," 2019. [Online]. Available: https://developer.android.com/guide/topics/sensors/sensors_position#sensors-pos-orient

[165] "Vicon: award winning motion capture systems," 2019. [Online]. Available: https://www.vicon.com/

[166] "OptiTrack: motion capture systems," 2019. [Online]. Available: https://optitrack.com/

[167] "Microsoft Kinect," 2019. [Online]. Available: https://developer.microsoft.com/en-us/windows/kinect

[168] "Intel RealSense technology," 2019. [Online]. Available: http://www.intel.com/content/www/us/en/architecture-and-technology/realsense-overview.html

[169] F. Adib, Z. Kabelac, D. Katabi, and R. C. Miller, "3D tracking via body radio reflections," in *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*, 2014, pp. 317–329.

[170] Q. Pu, S. Gupta, S. Gollakota, and S. Patel, "Whole-home gesture recognition using wireless signals," in *Proceedings of the 19th Annual International Conference on Mobile Computing and Networking*. ACM, 2013, pp. 27–38.

[171] F. Adib, C.-Y. Hsu, H. Mao, D. Katabi, and F. Durand, "Capturing the human figure through a wall," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 6, p. 219, 2015.

[172] D. Vasisht, S. Kumar, and D. Katabi, "Decimeter-level localization with a single WiFi access point," in *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, 2016, pp. 165–178.

[173] R. Meng, S. Shen, R. R. Choudhury, and S. Nelakuditi, "Autolabel: Labeling places from pictures and websites," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2016, pp. 1159–1169.

[174] R. Meng, S. Shen, R. Roy Choudhury, and S. Nelakuditi, "Matching physical sites with web sites for semantic localization," in *Proceedings of the 2nd Workshop on Physical Analytics*. ACM, 2015, pp. 31–36.

[175] L. Shangguan, Z. Zhou, and K. Jamieson, "Enabling gesture-based interactions with objects," in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2017, pp. 239–251.

[176] T. Wei and X. Zhang, "Gyro in the air: tracking 3D orientation of batteryless Internet-of-Things," in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. ACM, 2016, pp. 55–68.

[177] T. Li, Q. Liu, and X. Zhou, "Practical human sensing in the light," in *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2016, pp. 71–84.

[178] R. Nandakumar, V. Iyer, D. Tan, and S. Gollakota, "FingerIO: Using active sonar for fine-grained finger tracking," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 2016, pp. 1515–1525.

[179] H. Møller, "Fundamentals of binaural technology," *Applied Acoustics*, vol. 36, no. 3-4, pp. 171–218, 1992.

[180] D. Hammershøi and H. Møller, "Methods for binaural recording and reproduction," *Acta Acustica United with Acustica*, vol. 88, no. 3, pp. 303–311, 2002.

[181] "3Dio: Professional binaural microphones," 2019. [Online]. Available: https://3diosound.com/

[182] C. P. Brown and R. O. Duda, "A structural model for binaural sound synthesis," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 5, pp. 476–488, 1998.

[183] "Sennheiser's short film shows the power of binaural audio," 2019. [Online]. Available: https://thenextweb.com/plugged/2018/09/17/sennheisers-short-film-shows-the-power-of-binaural-audio/

[184] D. R. Begault, *3-D Sound for Virtual Reality and Multimedia*. Academic Press Professional, Inc., 1994.

[185] T. Lentz, D. Schröder, M. Vorländer, and I. Assenmacher, "Virtual reality system with integrated sound field simulation and reproduction," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, no. 1, p. 070540, 2007.

[186] D. R. Perrott and A. Musicant, "Minimum auditory movement angle: Binaural localization of moving sound sources," *The Journal of the Acoustical Society of America*, vol. 62, no. 6, pp. 1463–1466, 1977.

[187] P. Minnaar, S. K. Olesen, F. Christensen, and H. Møller, "Localization with binaural recordings from artificial and human heads," *Journal of the Audio Engineering Society*, vol. 49, no. 5, pp. 323–336, 2001.

[188] L. A. Jeffress, "A place theory of sound localization." *Journal of Comparative and Physiological Psychology*, vol. 41, no. 1, p. 35, 1948.