Noise-free audio signal processing in noisy environment: a hardware and algorithm solution

Yarong Feng*, Zongyi Liu, Shunyan Luo, Yuan Ling, Shujing Dong, Shuyi Wang, Bruce Ferry
Customer Experience and Business Trends, Amazon.com
2121 7th Ave, Seattle, WA, 98121
yarongf@*, joeliu@, shunyl@, yualing@, shujdong@, wanshuyi@, bferry@amazon.com

Abstract

Dealing with background noise is a challenging task in audio signal processing, negatively impacting algorithm performance and system robustness. In this paper, we propose a simple solution that combines recording hardware modification and algorithm improvement to tackle the challenge. The proposed solution could produce clean and noise-free high quality audio recording even in noisy recording environment. Experiment results show that the proposed solution leads to better sound event detection accuracy and speech recognition results.

1. Introduction

Audio signal processing is an important field that deals with the manipulation and enhancement of audio signals, including everything from music and speech to environmental sounds. It plays a vital role in our daily lives, impacting various industries and technologies, such as telecommunications, automotive systems, voice assistants, and more.

Two widely used techniques in audio signal processing are sound event detection and speech recognition. Sound event detection focuses on the identification and categorization of distinct acoustic events within audio signals. It involves analyzing audio signals to determine when specific sounds or events occur, and classifying them into predefined categories, such as car horns, dog barks, or doorbells. Speech recognition is a technology that enables a computer or machine to convert spoken language into written text or commands, facilitating human-computer interaction and the automation of various tasks via voice input.

A challenging task in audio signal processing is dealing with noise. Noise, usually in the form of unwanted background sounds, introduces distortions and disruptions to audio signals, which degrade their quality. For example, noise can significantly impact sound event detection results by introducing false positive events or missing actual events. It can also limit the ability of automated speech recognition systems to accurately transcribe spoken words. Effective noise reduction techniques and robust algorithms are essential for improving the accuracy and performance of such systems in noisy conditions.

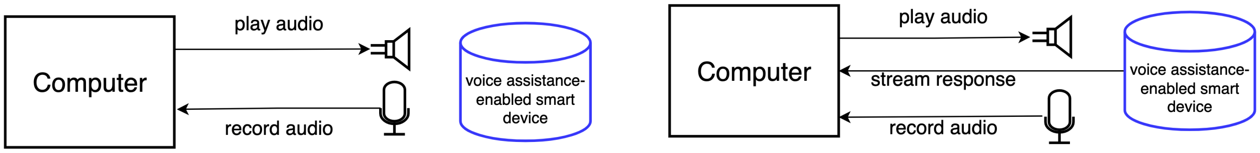
Depending on the time it is utilized, we can split noise reduction techniques into two types: at recording, and post recording. At-recording noise reduction techniques are applied during the recording phase aiming to minimize the introduction of noise into the audio signal in the first place. This involves selecting a quiet recording environment, using proper microphone techniques, employing soundproofing and acoustic treatment, and ensuring clean power sources. These practices can help capture cleaner audio with a higher signal-to-noise ratio. Post-recording noise reduction techniques are applied after the audio has been recorded. These methods include digital signal processing tools and software that can analyze the audio, identify noise components, and reduce or remove them while preserving the desired audio content. Noise reduction algorithms, spectral subtraction, and noise filtering are examples of post-processing techniques. Post-recording noise reduction is valuable when the recording environment is less than ideal or when dealing with unexpected noise.

In this paper, we propose a novel and simple solution combining at-recording and post-recording techniques to improve recorded audio quality. We show that the solution leads to improved performance on both sound event detection and speech recognition tasks.

Organization of the paper is as follows: Section 2 introduces related work. Section 3 explains the proposed methodology in details. Experimental results are shown in Section 4, and Section 5 concludes the paper.

2. Related Work

Sound Event Detection Recent research efforts focus on developing advanced algorithms, such as deep learning models and neural networks, to enhance the performance



(a) Diagram of existing hardware setup.

(b) Diagram of modified hardware setup.

Figure 1. Logical diagram of existing(left) and modified(right) hardware setup.

of sound event detection [5–7, 9]. These algorithms are designed to handle challenging scenarios like background noise, overlapping events, and low signal-to-noise ratios [8, 13, 15]. Additionally, there is active research on the integration of sound event detection into a wide array of applications, from smart home technology [10], security systems [4], to automated facility maintenance [11].

Noise reduction Noise reduction for audio signal processing involves the development of techniques and algorithms aimed at removing or mitigating unwanted noise from audio recordings while preserving the quality of the desired signal. Traditional approaches usually involve filtering [1], spectral subtraction [2], statistical methods [16], etc. More recently, data-driven approaches leveraging deep learning and machine learning are developed [3, 12, 18].

Hardware and sensor technology Various tools and devices have been invented to remove or mitigate noise during the capturing and processing of audio data. For instance, high-quality microphones and sensors sensitive to the target audio signals can minimize the capture of unwanted noise [14]. Advancements in signal processing hardware, including dedicated digital signal processors (DSPs) and specialized integrated circuits, also play a significant role in noise reduction [17].

3. Methodology

The proposed method consists of two parts: at-recording hardware modification and post-recording algorithm improvement for sound event detection. We first introduce the existing audio recording setup in Section 3.1, and move on to the details of each part in Section 3.2 and Section 3.3.

3.1. Existing audio recording setup

The existing audio recording setup we use includes a voice assistant-enabled smart device, a speaker, and a microphone, as shown in Figure 1a. The speaker and microphone are connected to and controlled by a computer. In a typical audio recording experiment, we play synthesized wake word and question/request to the voice assistant-enabled smart device through the speaker, wait for the response from it, and use the microphone to record the entire conversation, which usually has a duration of 20 $\sim$ 30 seconds. Some example conversations are shown in Figure 2, with an Alexa-enabled device in the setup.

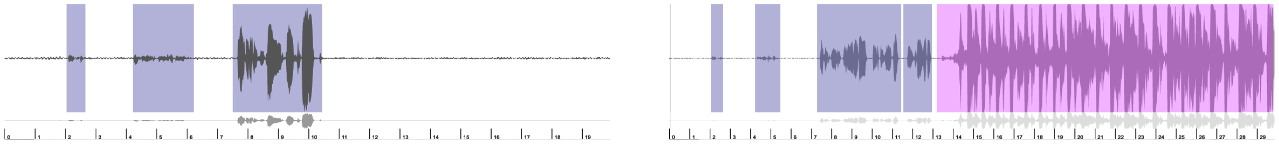
Audios collected using this setup can be used to evaluate a voice assistant's performance, leveraging the appropriate technique. For instance, to measure the response latency, we can perform sound event detection on the waveform to find the gap between the end of the question/request and the start of the response. Similarly, to measure the response quality, we can first perform speech recognition to extract the response text, and then assess the quality of answer manually or using an automated system.

However, audios recorded using the setup in Figure 1a are subject to unpredictable and uncontrolled background noise, leading to degraded audio quality. As pointed out in Section 1, it will negatively impact the performance of downstream algorithms and systems that take these audios as input, such as the sound event detection algorithm and the speech recognition system mentioned above, and in turn lead to inaccurate performance measurement and questionable conclusion.

3.2. At-recording Hardware Modification

We make one simple modification to the setup in Figure 1a: the voice assistant-enabled device is modified such that its response is directly streamed out using a wire and sent to the computer, as shown in Figure 1b. Audios collected in this way is completely free of background noise, because the signal now travels through the wire instead of air. Note that we make this modification in such a way that it does not affect the device's capability of producing audible response to human. In other words, the modified setup captures the device response in two ways in parallel: the usual audio response audible to human and will be recorded by the microphone, and the additional audio response not audible to human that is directly streamed out by a wire, as shown in Figure 1b. We shall refer to the first type as "recorded audio" and the second type as "streamed audio" in the remainder of the paper.

We show the waveform of such a pair of recorded and streamed audio in Figure 3. There are three main differences between them: first, the recorded audio includes the wake word and question/request played to the device through the speaker, while the streamed audio does not contain them, because the wire is only connecting the device and the computer; second, the gap between the wake word,



(a) Example conversation 1.

Synthesized user: Alexa, what's today's date?

Alexa: Today is November 1st, 2021.

(b) Example conversation 2.

Synthesized user: Alexa, play baby shark.

Alexa: Baby Shark by Pink Fong, on Amazon Music.

Alexa: (music playing).)

Figure 2. Waveform and text of example conversations collected in audio recording experiments. Blue color blocks in the waveform represent speech segments, and pink ones represent music.

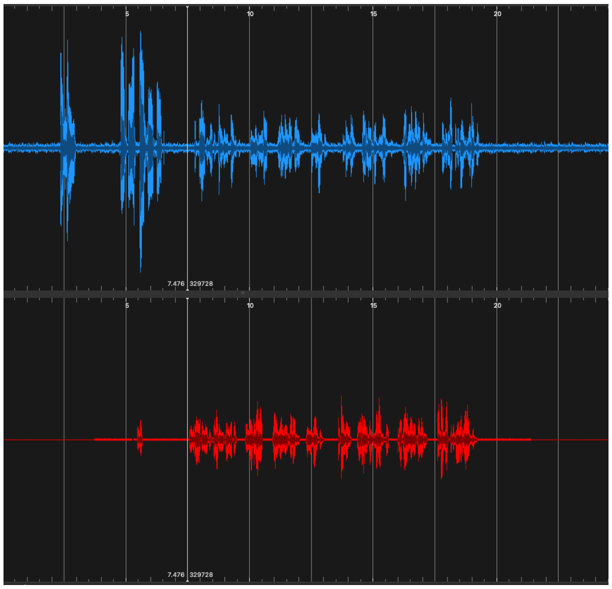


Figure 3. Waveform of recorded(top, blue) and streamed audios(bottom, red) for the same conversation.

question, and response segments in the recorded audio contains ambient noise, while in streamed audio it's completely free of noise(represented by an almost flat line in between sound events); third, the recorded audio and the streamed audio may not be in perfect sync. Because the two are collected in different ways, there is a non-negligible chance that they are out of sync in the generated audio files. For instance, in Figure 3, the streamed audio is slightly ahead of the recorded audio, with the difference most visible around second 7.5.

3.3. Post-recording Algorithm Improvement for Sound Event Detection

For the purpose of speech recognition, the streamed audio collected in the modified setup can be used directly to obtain high quality response text. However, to measure performance latency using sound event detection, streamed audio is insufficient because it only contains the response part of the conversation. The benefit of it, though, is running sound event detection on it can be achieved using simple off-the-shelf algorithms without advanced noise reduction or separation, because it can be considered completely noise-free. On the contrary, recorded audio contains the complete conversation, but requires more advanced and robust algorithms to ensure the sound event detection results are not impacted negatively by the background noise. Moreover, as explained in Section 3.2, one can not directly combine the recorded audio and the streamed audio, because the two can be out of sync.

Based on these observations, we propose a simple algorithmic improvement that brings the best of the two worlds together, and addresses the out of sync issue at the same time. We first align the recorded audio and the streamed audio using cross-correlation, bringing them in sync. Given two real-valued one-dimensional arrays **x** and **y** of length n_x and n_y respectively, their cross-correlation is an array **z** such that

$$\mathbf{z}[k] = \sum_{l=0}^{n_x-1} \mathbf{x}_l \mathbf{y}_{l-k+N-1}, k = 0, 1, ..., n_x + n_y - 2,$$

where $N = max(n_x, n_y)$, and $\mathbf{y}_m = 0$ when m is outside the range of **y**, and k is the lag index. To align **x** and **y**, we simply take $k^* = argmax_k \mathbf{z}[k]$, and shift **x** by k^*.

Then we run an off-the-shelf sound event detection algorithm on both the recorded audio and the streamed audio, and refer to the results as recorded sound events and streamed sound events, respectively. Lastly, we update the recorded sound events using the more accurate streamed sound events. Note that in the last step, only the recorded sound events in the response part are updated, with the wake word and question part intact. This is feasible because we know the rough time when the question/request ends in the recorded audio, as we control the start play time of the question and also know its length. The complete proposed algo-

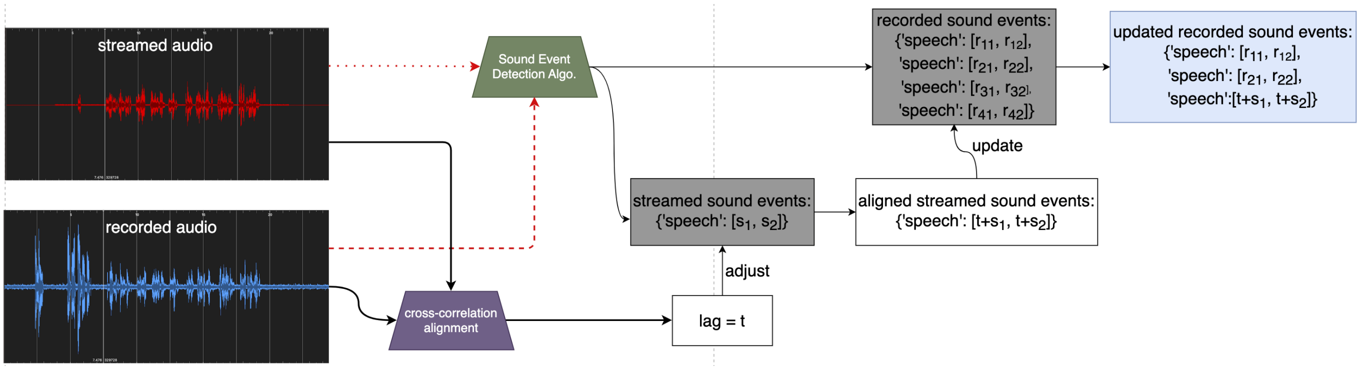

streamed audio
recorded audio
Sound Event Detection Algo.
cross-correlation alignment
lag = t
adjust
streamed sound events: {'speech': [s_1, s_2]}
aligned streamed sound events: {'speech': [t+s_1, t+s_2]}
update
recorded sound events: {'speech': [r_{11}, r_{12}], 'speech': [r_{21}, r_{22}], 'speech': [r_{31}, r_{32}], 'speech': [r_{41}, r_{42}]}
updated recorded sound events: {'speech': [r_{11}, r_{12}], 'speech': [r_{21}, r_{22}], 'speech':[t+s_1, t+s_2]}

Figure 4. Proposed algorithm diagram.

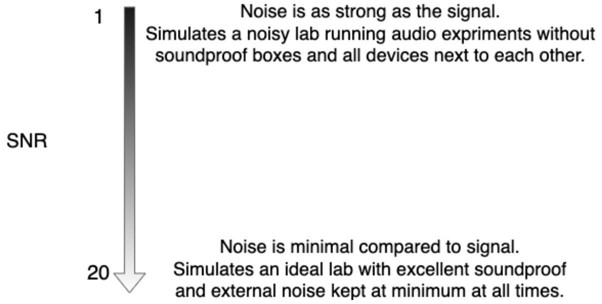

SNR
1
Noise is as strong as the signal.
Simulates a noisy lab running audio expriments without soundproof boxes and all devices next to each other.
20
Noise is minimal compared to signal.
Simulates an ideal lab with excellent soundproof and external noise kept at minimum at all times.

Figure 5. An intuitive guide on the scale of SNR and mapping to real-world scenarios.

rithm is shown in Figure 4. Note that when the recording environment is noisy, alignment between the streamed audio and recorded audio may not be accurate. In such cases, one can optionally pass the recorded audio through a bandpass filter to remove some noise, before running alignment.

4. Experiment Results

To demonstrate the effectiveness of the proposed solution, we run experiments to collect data across a variety of questions/requests using the modified setup in Section 3.2. We use one Apple home pod mini as the voice assistant-enabled device in all experiments for convenience. In reality, one can use any voice assistant-enabled device to run the experiment, as long as the device can be modified in the same way as described in Section 3.2. In each experiment, the wake word ”Hey Siri” and a randomly selected question/request from a predefined question/request set are played. We perform sound event detection using the proposed algorithm in Section 3.3 and speech recognition using AWS Transcribe. We use the existing setup in Section 3.1 with recorded audio only as the baseline. We introduce the detailed experiment settings in the following sections.

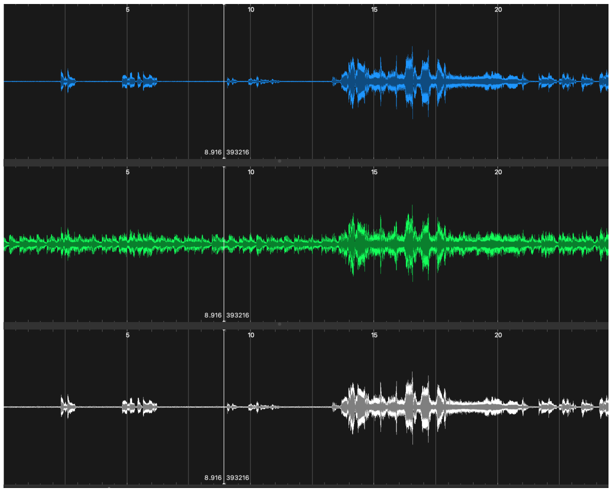


Figure 6. Example audios waveforms at different SNR values. Top(blue): original recorded audio, middle(green): SNR=1(high noise), bottom(white): SNR=20(low noise).

4.1. Sound Event Detection Results

To simulate a realistic acoustic environment for recording, we manually added background noise into the recorded audios, at different Signal-Noise-Ratio(SNR) levels. SNR is a common measure for audio quality and is computed as the ratio of the signal power and noise power. For our experiments, we simulated 20 distinct SNR settings(SNR ranging from 1 to 20) to measure the impact of different levels of noise. Figure 5 gives an intuitive guide on the SNR settings we choose. Figure 6 shows some example audios at different SNR levels and their waveforms.

We run 140 experiments and collect one recorded audio and one streamed audio from each. In the baseline method, for each recorded audio, we obtain its noisy version under each SNR setting(such as recorded_snr1, recorded_snr2, etc.), and run an off-the-shelf sound event detection algorithm on it to compute the response latency. In the proposed

SNR	**MAE(in seconds)**	
	streamed(proposed)	**recorded(baseline)**
1	0.08	1.25
2	0.08	0.85
3	0.08	0.46
4	0.08	0.23
5	0.08	0.19
6	0.08	0.18
7	0.08	0.20
8	0.08	0.15
9	0.08	0.16
10	0.08	0.14
15	0.08	0.26
20	0.08	0.19

Table 1. MAE using streamed audio versus recorded audio, in different SNR settings.

solution, for each audio pair, we simply use the algorithm in Section 3.3 to get the updated sound events and compute response latency. Under each SNR setting, the recorded audio in the proposed solution is replaced with the corresponding noisy version.

We use Mean Absolute Error(MAE) to measure the difference between the computed response latency and ground truth in each setting. As shown in Table 1, response latency computed using streamed audio has an MAE of 80ms in all SNR settings. This is to be expected because streamed audio is completely noise-free and the sound event detection results on them are not affected by the noise level. In contrast, response latency computed using recorded audio produces larger MAE as the noise gets stronger(SNR gets lower). At $SNR = 5$ and higher, the MAE stabilizes around $200ms$. Note that even the smallest MAE using recorded audio among all SNR settings($140ms$ at $SNR = 10$) is much higher than the MAE using streamed audio. This suggests that our proposed solution can still improve algorithm performance even in acoustic environment that has been optimized for noise reduction(with high SNR).

4.2. Speech Recognition Results

We collect data for 80 experiments. For each experiment, we run AWS Transcribe on the recorded audio and streamed audio respectively to obtain the response text. As no ground truth is available for this task, we compute the Word Error Rate(WER) between the recorded response text and streamed response text, to illustrate the difference between the two. Across 80 experiments, the average un-normalized WER between recorded response text and streamed response text is 18. It means on average, in each conversion, there are 18 words that are different. "Different" could mean word displacement, substitution, insertion, or deletion. During manual inspection, we have found that many of these differences are caused by transcription errors on recorded audios. We provide a few examples in Table 2 with the transcription error highlighted in red.

5. Conclusion

We propose a hardware modification solution to record noise-free and high quality audio even in noisy environment. In addition, we also propose an algorithm to perform accurate sound event detection using the recordings collected in the proposed setting. Experiment results show that the proposed solution could produce very accurate sound event detection results even in environments with low SNR. There is also qualitative evidence that the proposed solution leads to better speech recognition results.

References

[1] A.S. Abutaleb. An adaptive filter for noise cancelling. *IEEE Transactions on Circuits and Systems*, 35(10):1201–1209, 1988. 2

[2] Steven Boll. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on acoustics, speech, and signal processing*, 27(2):113–120, 1979. 2

[3] Sebastian Braun and Ivan Tashev. Data augmentation and loss normalization for deep noise suppression. In *International Conference on Speech and Computer*, pages 79–86. Springer, 2020. 2

[4] S. Chandrakala and S. L. Jayalakshmi. Environmental audio scene and sound event recognition for autonomous surveillance: A survey and comparative studies. *ACM Comput. Surv.*, 52(3), jun 2019. 2

[5] Yarong Feng, Zongyi Joe Liu, Yuan Ling, and Bruce Ferry. A two-stage lstm based approach for voice activity detection with sound event classification. In *2022 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–6, 2022. 2

[6] Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D. Plumbley. Panns: Large-scale pretrained audio neural networks for audio pattern recognition, 2020. 2

[7] Shunyan Luo, Yarong Feng, Zongyi Joe Liu, Yuan Ling, Shujing Dong, and Bruce Ferry. High precision sound event detection based on transfer learning using transposed convolutions and feature pyramid network. In *2023 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–6, 2023. 2

[8] Annamaria Mesaros, Aleksandr Diment, Benjamin Elizalde, Toni Heittola, Emmanuel Vincent, Bhiksha Raj, and Tuomas Virtanen. Sound event detection in the dcase 2017 challenge. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(6):992–1006, 2019. 2

streamed response text	**recorded response text**
hey siri. what's the weather tomorrow? expect cloudy skies tomorrow. daytime temperatures will hover around 75 degrees with overnight lows around 62.	hey siri. what's the weather tomorrow? expect? cloudy skies tomorrow. daytime temperatures will hover around 75 degrees with overnight lows around 62.
hey siri. cancel all timers. there are no timers on home pod.	hey siri, cancel. all timers. there are no timers on home pie.
hey siri. what's fifteen hundred times twelve. 1500 times 12 is 18,000.	hey siri. what's 1500 times 12, 1000, 500 times 12 is 18,000.

Table 2. Qualitative speech recognition results comparison.

[9] Annamaria Mesaros, Toni Heittola, Tuomas Virtanen, and Mark D. Plumbley. Sound event detection: A tutorial. *IEEE Signal Processing Magazine*, 38(5):67–83, sep 2021. 2

[10] Sharnil Pandya and Hemant Ghayvat. Ambient acoustic event assistive framework for identification, detection, and recognition of unknown acoustic events of a residence. *Advanced Engineering Informatics*, 47:101238, 2021. 2

[11] Harsh Purohit, Ryo Tanabe, Kenji Ichige, Takashi Endo, Yuki Nikaido, Kaori Suefusa, and Yohei Kawaguchi. Mimii dataset: Sound dataset for malfunctioning industrial machine investigation and inspection, 2019. 2

[12] Chandan KA Reddy, Vishak Gopal, Ross Cutler, Ebrahim Beyrami, Roger Cheng, Harishchandra Dubey, Sergiy Matusevych, Robert Aichner, Ashkan Aazami, Sebastian Braun, et al. The interspeech 2020 deep noise suppression challenge: Datasets, subjective testing framework, and challenge results. *arXiv preprint arXiv:2005.13981*, 2020. 2

[13] Francesca Ronchini and Romain Serizel. A benchmark of state-of-the-art sound event detection systems evaluated on synthetic soundscapes. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1031–1035, 2022. 2

[14] Thomas Stark Ilona Anderson Peter Nopp Ernst Aschbacher Alexander Möltner Yassaman Khajehnouri Rudolf Hagen, Andreas Radeloff and Kristen Rak. Microphone directionality and wind noise reduction enhance speech perception in users of the med-el sonnet audio processor. *Cochlear Implants International*, 21(1):53–65, 2020. PMID: 31524107. 2

[15] Romain Serizel, Nicolas Turpault, Ankit Shah, and Justin Salamon. Sound event detection in synthetic domestic environments. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 86–90, 2020. 2

[16] Volker Stahl, Alexander Fischer, and Rolf Bippus. Quantile based noise estimation for spectral subtraction and wiener filtering. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)*, volume 3, pages 1875–1878. IEEE, 2000. 2

[17] Akihiko Sugiyama, Ryoji Miyahara, and Kouji Oosugi. A noise robust hearable device with an adaptive noise canceller and its dsp implementation. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2722–2726, 2019. 2

[18] Hao Zhang, Ke Tan, and DeLiang Wang. Deep learning for joint acoustic echo and noise cancellation with nonlinear distortions. In *Interspeech*, pages 4255–4259, 2019. 2