Dual-Microphone Noise Removal for Keyword Spotting

Rayan Daod Nathoo[1], Frédéric Bischoff[1]

[1]Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland

***Abstract*—The aim of this project is to propose an adaptive noise cancellation (ANC) filter for a dual-microphone so that a wake-word uttered by an user, such as "Alexa" or "Hey Google", appears more intelligible and clear to the following keyword spotting algorithm. The noise source could be any background noise, but also music and dialog coming from an audio system, radio or TV.**

Keywords: Adaptive noise cancellation, Dual-microphone, wake word

I. Introduction

Voice user interfaces (VUI) are the primary way of interacting with virtual assistants. They make spoken human interaction with computers possible, using speech recognition to understand spoken commands and answer questions. Some concrete VUI are already present in our all-day life, like Google Assistant, Amazon Alexa, and Apple Siri.

Voice assistants must be usable in a wide range of acoustic conditions and keyword detection in a noisy environment remains a challenge. In this report we will focus on background noise removal for dual-microphones in order to enhance keyword spotting. Based on recent publications, we will first present a review of some of existing techniques before implementing our own solution.

II. Review of existing techniques

A. Coherence based techniques

Coherence is a statistic used to examine the relation between two signals. It is commonly used to estimate the power transfer between input and output of a linear system. Several publications rely on coherence for multi-channel microphone noise reduction applications. We can mention [5], a speech enhancement algorithm based on the coherence function. The algorithm developed in this paper relies on the fact that speech signals in the two channels are correlated, while the noise signals are uncorrelated, so if the magnitude of the coherence function between the noisy signals at the two channels is one (or close to one), the speech signal is predominant and thus should be passed without attenuation, and if the magnitude is close to zero speech is absent, and thus the input signals should be suppressed.

We can also mention a specific application for cell-phones developed in [6]. This dual-channel speech enhancement algorithm is derived from the coherence function and the Kalman filter. The latter is an algorithm that uses series of measurements observed over time, containing statistical noise, and produces estimates of clean speech that tends to be more accurate than those based on a single measurement alone by estimating a joint probability distribution over the variables for each time frame.

B. Machine learning techniques

An important number of recent publications rely on deep learning [2]. Different type of neural network structures are used like deep neural network [3] or convolutional neural network [4]. The drawback of all these machine learning techniques is that they require an important training dataset in order to be efficient.

III. Algorithm description

In this project, we are using two microphones separated by a small distance to capture the background noise from the surroundings and the keyword, which makes those microphones coherent. The setup is as follows:

$$x_1(t) = h_1(t) * (s(t) + n(t))$$

$$x_2(t) = h_2(t) * (s(t) + n(t))$$

with $s(t)$ the clean speech signal, $n(t)$ the noise, and $h_i(t)$, i=1,2 the two channels. In section A, we implement a Wiener filter in Short Time Fourier Transform (STFT) domain, and in section B we focus on an ANC filter with deferred filter coefficients, an algorithm derived in [1].

A. Wiener Filter

The Wiener filter, aims at estimating one signal s of the dual microphone from the other x. The filter coefficients are calculated in STFT domain for each frame i of the STFT with a given forgetting factor α taking into account previous frames. We suppose that the recording is essentially composed of noise and that the keyword lasts a very short time (less than 1s), as a consequence, the obtained filter should be able to remove only noise and keeps the keyword.

Fig.1 describes visually how the filter is obtained. First, we calculate the short time fourier transform (STFT) of two input signals s and x, then ϕ_{xs} and ϕ_{xx} are obtained as follow for each frame i of the STFT:

$$\phi_{xs}[i] = \alpha s[i] * x[i]^* + (1-\alpha)\phi_{xs}[i-1]$$

$$\phi_{xx}[i] = \alpha x[i] * x[i]^* + (1-\alpha)\phi_{xx}[i-1]$$

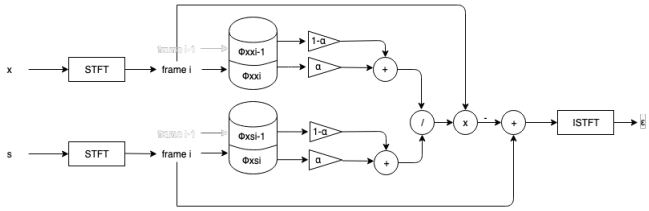

Fig. 1. STFT based Wiener filter with forgetting factor

The ratio of ϕ_{xs} over ϕ_{xx} allows to get the filter coefficients for all frequencies at frame i:

$$h[i] = \frac{\phi_{xs}[i]}{\phi_{xx}[i]}$$

When the auto-correlation between x and s is high (the signals are highly correlated), ϕ_{xs} and ϕ_{xx} are very close and $h[i]$ is close to 1, whereas when the auto-correlation between x and s is low (the signals are very distinct), ϕ_{xs} is close to zero and so is $h[i]$.

Then we can estimate the error signal ϵ:

$$\epsilon[i] = s[i] - h[i] * x[i]$$

The inverse short time fourier transform of ϵ gives us in return the cleaned signal.

B. Adaptive noise cancellation with deferred filter coefficients

A classical adaptive filter applied to one microphone signal with the result subtracted from the other microphone signal would cancel not only noise but also speech. This setup makes the problem interesting and different from some dual-microphone algorithms relying on the fact that the two microphones are not coherent, with one capturing more noise, and the other mainly capturing speech with a high Signal to Noise Ratio (SNR). Ideally in our case, the adaptation should be performed when speech is absent and must be halted otherwise. Here we propose to implement and explain the algorithm described in [1], which relies on the two following assumptions:

1) The short segment immediately preceding a keyword contains only noise, which allows us to estimate the noise before an utterance occurs.

2) A keyword has a short duration, typically less than 1 second, which will be important for the later parameter choices.

With these two assumptions the authors were able to derive an algorithm consisting on two processing layers, as illustrated by Fig. 2, with one layer on top of the other. The algorithm is presented in more details in Fig. 3.

The bottom layer works like a traditional ANC system that finds the filter coefficients that minimize the mean square of the prediction error $e(n)$ between the two inputs. It first computes the STFT of both signals with a window size W and an overlap percentage O, and computes the filter coefficients for the current frame by means of a Kalman gain vector taking into account the L last frames with a forgetting factor λ. These

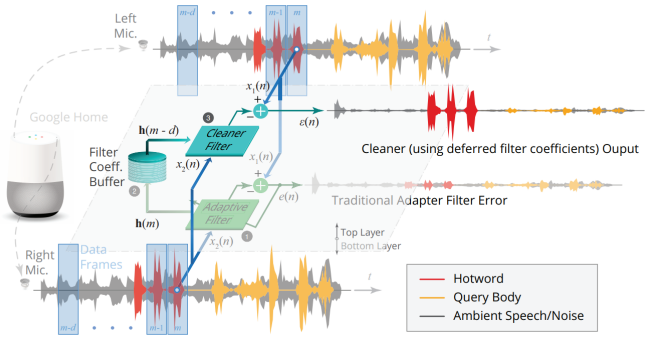

Fig. 2. ANC filter with deferred filter coefficients

Parameters:

d = size of the delay buffer in number of frames

L = filter length, λ = forgetting factor

δ = coefficient to initialize $\mathbf{P}(0)$

Initialization:

$\mathbf{h}(0) = \mathbf{0}, \mathbf{x}_2(0) = \mathbf{0},$

$\mathbf{P}(0) = \delta^{-1}\mathbf{I}$, where $\mathbf{I}$ is the identity matrix of rank L.

Adaptation: for frames $m = 1, 2, \cdots$

A-priori error:

$$E(m) = X_1(m) - \mathbf{h}^H(m-1)\mathbf{x}_2(m),$$

Kalman gain vector:

$$\mathbf{g}(m) = \frac{\mathbf{P}(m-1)\mathbf{x}_2(m)}{\lambda + \mathbf{x}_2^H(m)\mathbf{P}(m-1)\mathbf{x}_2(m)},$$

Update:

$$\mathbf{P}(m) = \lambda^{-1}\left[\mathbf{P}(m-1) - \mathbf{g}(m)\mathbf{x}_2^H(m)\mathbf{P}(m-1)\right],$$

$$\mathbf{h}(m) = \mathbf{h}(m-1) + \mathbf{g}(m)E^*(m),$$

Output:

$$\mathcal{E}(m) = X_1(m) - \mathbf{h}^H(m-d)\mathbf{x}_2(m).$$

Fig. 3. STFT domain fast RLS with deferred coefficients algorithm

estimated filter coefficients are then saved into a first-in-first-out (FIFO) buffer of size d.

In the top layer, the buffer outputs are used with a delay of d frames to generate the cleaned signal (n) from the estimated one. This deferred coefficients technique allows us with a well chosen set of parameters to remove the noise and enhance the speech signal since it only takes into account earlier statistics of the signal. The algorithm is explained in more details in [7].

IV. Results and Discussion

For our experiments, we used a Logitech C920s camera with two embedded microphone. Those two entries were separated by $6cm$ and the sampling rate was 32000Hz. The setup is illustrated in Fig 3. As you can see, there is no delay introduced regarding the speech but only for the noise. If we assume that the speed of sound is $340m.s^{-1}$, it corresponds to:

$$\frac{0.06}{340} * 32000 = 5\ samples$$

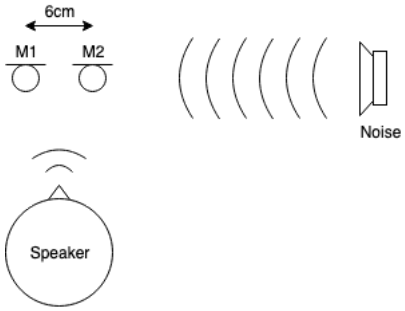

Fig. 4. Experimental setup

For each of the two algorithms, we ran four different tests:

1) Delayed inputs with speech added: this scenario represents a simple case in which the two inputs are only delayed one from another by the same number of samples as above, the speech signal has no delay (we suppose that the speaker is in front of both microphones as depicted in Fig.4), and the same channel is applied to both.

2) Delayed inputs with speech added and both signals filtered with a different Room Impulse Response (RIR): here we experiment the same scenario, except that we filter each of the two signals with a RIR, mainly a $close\_mic\_rir$ RIR, simulating a close microphone, and a far_mic_rir RIR, simulating a far microphone. This way we could simulate scenario a bit more realistic since as mentioned before with two different channels applied to our signals.

3) Original signals from the recording with speech added: this time we take the original recorded signals and add a pre-recorded speech signal on top. This is an interesting scenario too since real channels h_1 and h_2 can be observed. Notice that since the speech signal is recorded before, the noise signal and the speech signal are not under the same conditions/channels. We are still assuming no delay in the speech signals.

4) Original signals from the recording already containing the speech signal: finally we try our algorithm on a real case scenario illustrated in Fig. 4.

The plots of the results are available in the following notebooks:

STFT based Wiener filter with forgetting factor

STFT-domain fast RLS with deferred coefficients

V. Conclusion

We have implemented two different algorithms for noise reduction in the context of keyword spotting: an STFT based Wiener filter with forgetting factor algorithm, and an STFT-domain fast RLS with deferred coefficients. We did not have time to vectorize the algorithms which would make them a lot faster so this would be a possible future work. The logical continuation would then be to implement those algorithms in real-time and to combine them with actual keyword detection algorithms.

VI. Bibliography

[1] *Hotword cleaner: dual-microphone adaptive noise cancellation with deferred filter coefficients for robust keyword spotting*, Yiteng (Arden) Huang, Turaj Z. Shabestary, Alexander Gruenstein Google Inc., USA, 2019

[2] *Deep neural networks for acoustic modeling in speech recognition*, G. Hinton, L. Deng, Sep. 2012

[3] *Small-footprint keyword spotting using deep neural networks*, G. Chen, C. Parada, 2014

[4] *Convolutional neural networks for small-footprint keyword spotting*, T. N. Sainath and C. Parada, 2015

[5] *A Dual-Microphone Speech Enhancement Algorithm Based on the Coherence Function*, Nima Yousefian and Philipos C., 2012

[6] *Speech enhancement in dual-microphone mobile phones using Kalman filter*, Wahbi Nabi, Noureddine Aloui, 2015

[7] *Supervised noise reduction for multi-channel keyword spotting*, Y. Huang, T. Hughes, T. Z. Shabestary, and T. Applebaum, in Proc. IEEE ICASSP, 2018, pp. 5474–5478.