ACTIVE NOISE REDUCTION

Harrison Walters

University of Victoria

Ryan Hutton

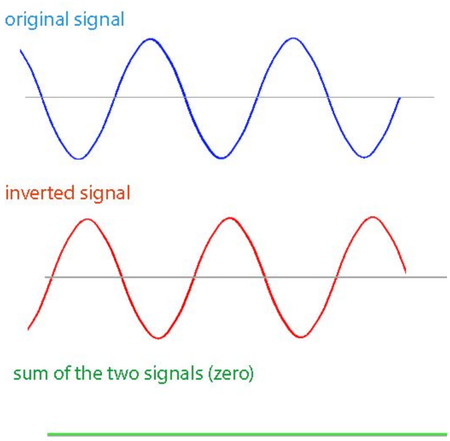
University of Victoria

ABSTRACT

Active noise reduction (ANR) is a means of reducing undesirable ambient noise, achieved by adding a secondary sound wave designed to cancel out the first. Within the context of this project, we will be using the input from a portable microphone to reduce periodic ambient noise around the user by playing a sound wave with inverted amplitude around the user's headphones, which will effectively cancel out sound waves generated by ambient sources (a process known as *destructive interference*). Our goal for this project is to produce a measurable amount of noise reduction, and to maximize the results via different strategies and tweaks.

1. BACKGROUND

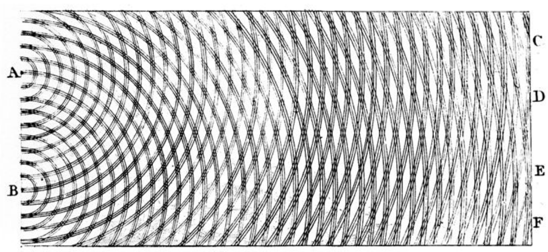
"Sound", as perceived by humans, is essentially the rapid and periodic compression and rarefaction of air, that travels in a wave-like form through an environment. The faster the sound wave oscillates (higher frequency) the higher the pitch of the sound and the taller the wave is (its amplitude) the louder that sound is [20]. If one were to able to perfectly emit a secondary signal with an amplitude that was the exact inverse of the original signal (called an "anti-signal"), the sum of the two signals would be zero and a person would perceive no sound [20]:



1.1. Noise reduction in detail:

In a "textbook implementation" of noise reduction, a microphone would detect incoming noise signals, and a speaker producing the anti-signal would emit a perfectly incoming noise signals instantly [20]. This would cancel-out all incoming noise perfectly, and the user would perceive absolute silence. However in reality, since noise signals change frequently, and there are is a delay from when the microphone detects noise, and the computed anti-signal is emitted. In addition to that, it is difficult to control noise in large regions of space. Problems also occur when trying to cancel high-frequency noise, since it is often difficult to match the phase of the noise signal, and the speaker emitting the anti-signal usually ends up adding **more noise** to the resulting signal. That being said, it is usually easier to cancel low-frequency sound using anti-signals, and high-frequency is limited to "soundproofing" an environment using padding or insulation.

Noise cancellation with one speaker fighting against one sound source is limited to a "one dimension zone" (a term used in practice), meaning that the destructive interference is only effective in one direction [20] (ie. a user sitting on the far right of the diagram below)



This kind of noise reduction can be very effective within a **very small region,** such as the region around the user's ears, however noise reduction within a large region (such as in a corner of the room, or a room in its entirety) is very complex to implement/achieve, and are beyond the scope of this project. These "3 dimensional zones" that cover a wide area requires many microphones and speakers. Real-world mechanism such as noise-cancelling headphones, active mufflers, and air-conditioning noise cancellers would fall under this “1 dimensional” category.

In addition to these issues, since the user’s left and right ear are in different locations, there will be a slight disparity between the incoming audio signals arriving at the left and right headphone buds. A software implementation of noise reduction would have to process each “ear piece” independently.

1.2. ANR filters:

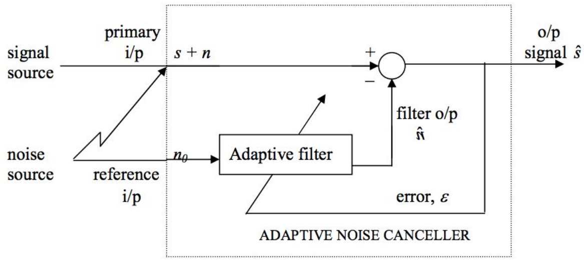
Part of the difficulty of noise cancellation is determining what noise to remove and creating an accurate filtering algorithm that can adapt to different situations.

When a signal is corrupted with additive noise, the usual method of removing the noise while leaving the signal unchanged is to pass it through a filter that takes the signal and the noise, then creates the antinoise signal and sums all the signals together, leaving us with only the signal sans noise. This process is typically known as direct filtering and can be subdivided into 2 categories:

Fixed Filters - which require prior knowledge of the signal and the noise beforehand.

Adaptive Filters - which have the ability to adjust their response to filter out the correlated signal in the input. They require little to no prior knowledge of the signal and its noise.

Typically we are not interested in fixed filters as they are trivial, static and require prior knowledge we typically do not have. As the power of digital signal processors increased, adaptive filters became more and more common and can be found in everyday devices such as mobile phones, camcorders and cameras, as well as medical monitoring equipment. For ANR adaptive filtering means estimating the noise and if done improperly could mean an increase in noise in the outputting signal.



The figure above illustrates an Adaptive Noise Canceller. It takes a primary input which consists of a signal s that is corrupted by noise n which is uncorrelated with the signal and it takes a reference input that consists of a noise n_0 that is correlated in some way with noise n. The noise n_0 is put through the filter to produce a close estimate $\hat{n}$ of the primary input noise, which is then subtracted from the corrupted signal to produce an output signal $\hat{s}$ which is an estimated of the desired output signal.

1.3. Common Adaptive Filtering Algorithms:

Steepest Descent: Steepest Descent is an old, recursive, deterministic feedback system algorithm. It begins with some initial value for the filter coefficients vector and improves with each iteration. The deterministic feedback system allows the finding of the minimum point in the average error-surface without knowing that surface.

Least Mean Squares: The LMS algorithm is a linear stochastic gradient algorithm. It’s largely used in ANR due to its ability of requiring no knowledge of the exact signal statistics and contains methods that are able to track variations in the signal statistics. It utilizes a filtering process that computes the transversal filter output produced by the filter coefficient inputs and the compare that output with a desired response, generating an error estimation. With this error estimation it can then automatically adjust the filter coefficients.

Normalised Least Mean Squares: The main issue with LMS algorithms is that they are sensitive to the scaling of the input making it very difficult to choose a learning rate that guarantees stability. NLMS modifies LMS by normalising with the power of the input.

Recursive Least Squares: Contrasting to the LMS algorithm, the RLS algorithm's objective is to recursively find the filter coefficients that minimize the least square cost function. RLS has an extremely fast convergence, but a rather large computational complexity.

2. TECHNICAL SPECIFICS

2.1. Tools and libraries:

Our project will be written in Python, and use the PyAudio library to handle incoming/outgoing audio streams, and the NumPy package to process the signals.

2.2. Software Implementation:

Given the many problems with noise cancellation/reduction, such as processing delays and changes to the signal, our software will have to constantly sample ambient sound to build a synthetic signal from a **long-term approximation** of the ambient waveform, and emit that from the speaker (phase shifted by 180 degrees) as an anti-signal. We would like for our software to also be able to **simulate processing and output delays** that would be present in a real-world hardware implementation, and also be able to use a variety of different pre-recorded noise signals. In addition to this, we would like our software to calculate the amount of **total energy** present in the resulting waveform (to compare with original signal), as a measure of how well the noise reduction is working. We will also add a **negative feedback** mechanism, so that if the anti-signal is incorrect or too strong at any moment in time, it will reduce/enhance the strength of the anti-signal. Likewise, We also want to be sure that we use small buffers and little memory, so that our software implementation could be feasible in real-world applications.

2.3. Implementation specifics:

We are using functionality from the PyAudio library to read information from audio streams such as pre-recorded wav files or the user's microphone, and converting it into blocks of time-magnitude data. At the heart of our implementation is the **Discrete Fourier Transform**, which we can use this context to separate audio waveforms into the sine/cosine waves (of varying frequencies) that constitute it. Incoming audio data is then fed into the DFT (provided by NumPy), where frequency, magnitude, and phase data are collected and added to an "averaging set", which is used in an inverted DFT for synthesizing the anti-signal. There are a variety of strategies to be tested for synthesizing the anti-signal, such as :

- Outputting an inverted mean-average of the 'averaging set' as the anti-signal
- Outputting an inverted weighted-average of the 'averaging set' as the anti-signal (linearly interpolated / cubic interpolation)
- Simply outputting the inverted frequency-magnitude-phase data of the "most recently arrived" as the anti-signal

In addition to testing these different strategies, there are several other parameters we can configure, such as:

- size of the averaging set (will short-term averages or long-term averages work better?)
- amount of simulated processing delay (the software will hold on to "fresh" data for a specified amount of time before it becomes accessible to the rest of the software.)
- rate-of-correction of the negative feedback mechanism

We will compute the "total energy" of the original noise signal as the sum of the magnitude of its signal at each point in time, and compare it to the resulting waveform (original+anti-signal) to measure the 'effectiveness' of any given strategy. For example, a value of 1.50 would mean that the resulting signal has **more energy** and is louder than the original signal (meaning that a strategy is bad), and a value of 0.75 would mean that there is **less energy** in the signal and that the resulting waveform is quieter (and hence 'noise reduced') than the original unadulterated signal.

2.4. Hardware Implementation

Typically there are two types of setups for noise cancelling headphones. One involves putting the microphone outside the ear cup with an open-loop system (figure 1), while the other involves setting up the microphone inside the ear cup with a closed loop system. We opted for a closed loop system, housing our microphones within the ear cups.

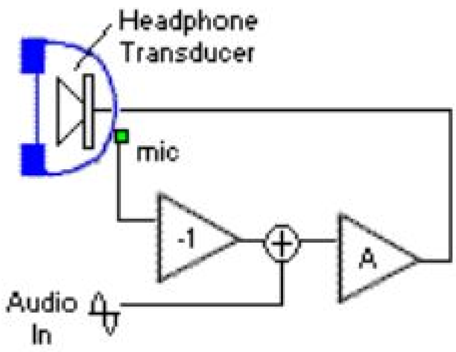


Figure 1.

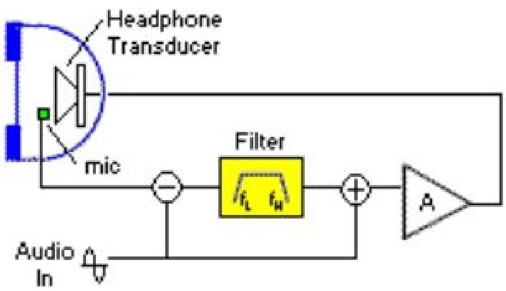
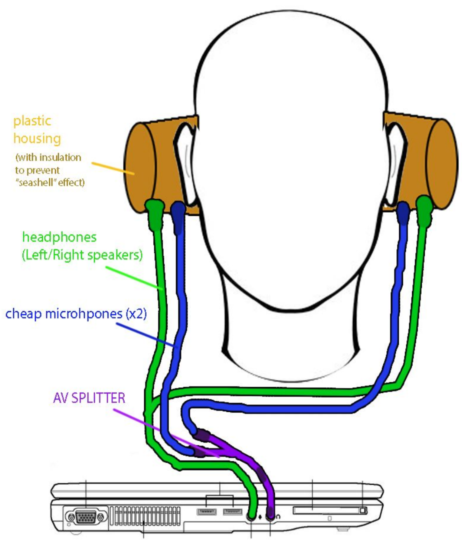


Figure 2.

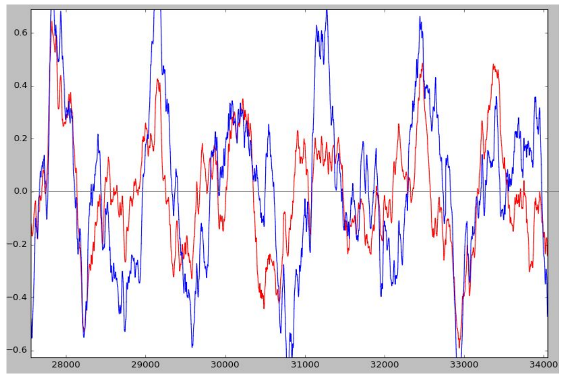
In addition to the core software implementation, we have built some primitive noise-cancelling headphones which we will attempt to get working with our software implementation using the following setup**:**



3. PRELIMINARY RESULTS

We will be testing with 3 .wav files (a fan noise, a fridge humming noise, and a washing machine) instead of real-time microphone input, and measure the "total energy" before/after an anti-signal is applied to measure the effectiveness of our different noise-reduction strategies in a deterministic/objective way,

A screenshot from our digital environment (using our "most-recent-frame" strategy)



Blue: original, unadulterated signal
Red : resultant signal with an anti-signal applied (notice : the magnitude of the waveform is significantly lower)

3.1. Most-Recent-Frame Strategy

Our "most-recent-frame" strategy is the one that mirrors most real-world implementations. Noise-cancelling headphones rely on processing incoming signals immediately through specialized high-speed hardware with small or non-existent memory. In our implementation, it uses the most recent "chunk" of data (which would be stored in a buffer of some sort in a real-world implementation), and synthesizes a generalized anti-signal from that. Results were optimal across the 3 test wav files when the strength of the delayed anti-signal was set to **0.8**. The corresponding reduction of "total energy" of the waveforms for this value was:

- fan-wav : 0.75880989267
- fridge-wav : 0.75032214688
- washer-wav : 0.74818417987
- average: 75% of original intensity

3.2. Frame-Averaging Strategy

Our "frame-averaging" strategy takes a mean average of **n** previously seen FFTs, stored in a list. The computed average is used to generate the anti-signal. Results were optimal across the 3 test files when the averaging-set size was set to 3, and was significantly better than the MRF strategy.

- fan-wav : 0.58163341549
- fridge-wav : 0.6199657470
- washer-wav : 0.65880989267
- average: 62% of original intensity

Implementing this method in a real-world hardware application would require more memory and would be more expensive to produce.

3.3. Weighted-Frame-Averaging Strategy

Our final strategy was to do a weighted average of n previously seen FFTs. The weights would be determined via linear interpolation. For example, if n was set to 3 (use 3 previous FFT frames), then frame-1 (newest/most-recent frame) would get a weight of 1.0, whereas frame-3 (oldest frame) would get a weight of 0.25. Results were slightly improved over the blind mean-average tried before, but results were still optimal when using an averaging-set of size 3.

- fan-wav : 0.6043550128
- fridge-wav : 0.594550299
- washer-wav : 0.574588391
- average: 59% of original intensity

3.4. Hardware Attempt

Upon further experimentation with the microphones we had purchased for this project, we discovered that none of them were sensitive enough to pick up ambient noise. Even with a fan running at maximum speed, and placed right next to the microphone, the microphones would not pick up any audio data whatsoever. In practice, a pre-amplifier is used in noise-cancelling headphones to amplify the weak signals produced by ambient sound, along with a very sensitive microphone, but in our case, the microphones we used produced no data whatsoever. Unfortunately, due to time constraints, we were unable to pursue this extended segment.

3.5. Results, Conclusions, Future Work

We were hoping to achieve a >50% reduction of ambient noise (in our simulated software environment) using the handful of primitive techniques we used, but were only able to achieve ~57% at best using the weighted-frame-averaging strategy. Still, that is a considerable reduction of undesirable ambient noise using only simple techniques. If we had more time to work on this project, we would definitely try to extend our software to an actual real-world environment (ie. build a physical realization of an active-noise-reduction environment). In addition, we would probably attempt some of the more "intense" math-heavy ANR filters.

4. PROJECT OUTLINE AND DATES

March 4th : skeletal code, working libraries.
March 11th : ability to synthesize/replay periodic signals recorded from the environment
March 18th : anti-signal computation and phase adjustment/minimization functionality
March 25th : experimentation with various noise reduction techniques

5. RELATED WORK

Dr. Amar Bose is widely credited as the inventor of noise cancelling headphones. He coined the idea while on a transatlantic flight and being unable to enjoy his music over the roar of the engines. The patent for the headphones [18] would then go to be applied to pilots of aircraft to protect hearing in the first ever unrefueled non-stop around-the-world flight.

While Bose may have coined the idea of noise cancelling headphones in the 70's, noise cancellation itself is much older. Lueg Paul filed a patent in 1934 for the process of silencing sound oscillations by subjecting these oscillations to a displacement of phases. "In these processes only the source of the oscillations is used to cause the displacement of phases, so that the superposition of phases takes place in a purely mechanical manner" [15].

In an undergraduate research project conducted at the University of Washington St. Louis by Shizhang Wu where a noise canceling headphone was built with a negative feedback system [2]. The actual headphones cups were found to be able to block out noise 1 KHz and above (most headphones are capable of this without any sort of negative feedback system). The design was largely based around Bose's patent [18] and focused on eliminating/reducing noise at less than 500 Hz frequency. It was able to attain a noise attenuation well above 10 dB with a stable system.

Microphones require noise cancellation technology to block out acoustic noise with greater or equal energy to speech. This can be done by adaptively filtering a separately recorded correlated version of the noise signal and then subtracting it from the speech waveform. Up to 20 dB can be reduced without distortion to speech audio [9].

6. REFERENCES

[1] K. C. Chin, C. H. Khoh, S. P. Chong, and J. N. Ong. Noise cancelling headphone. Sony Emcs (Malaysia) Sdn. Bhd., assignee. US Patent No. 9117437. 25 Aug. 2015. Print.

[2] S. Wu. *Noise Canceling Headphones*. Department of Electrical and Systems Engineering Washington University in St. Louis , Fall 2008. Web.

[3] A. Milosevic and S. Urs. *Active Noise Control*. Thesis. Hochschule für Technik Rapperswil, 2005. N.p.: n.p., n.d. Print.

[4] S. J. Elliott and P. A. Nelson, "Active noise control," in *IEEE Signal Processing Magazine*, vol. 10, no. 4, pp. 12-35, Oct. 1993.
doi: 10.1109/79.248551

[5] L. E. Rees and S. J. Elliott, "Adaptive algorithms for active sound-profiling," in *IEEE*

Transactions on Audio, Speech, and Language Processing, vol. 14, no. 2, pp. 711-719, March 2006.
doi: 10.1109/TSA.2005.855828

[6] H. Pham. *PyAudio*. Computer software. Vers. 0.2.10. MIT, n.d. Web.

[7] J. N. Denenberg, "Anti-noise," in *IEEE Potentials*, vol. 11, no. 2, pp. 36-40, April 1992.
doi: 10.1109/45.127730

[8] R. Pettai. *Noise in Receiving Systems*. New York: Wiley-Interscience, 1984. Print.

[9] S. Boll and D. Pulsipher, "Suppression of acoustic noise in speech using two microphone adaptive noise cancellation," in *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 6, pp. 752-753, Dec 1980.
doi: 10.1109/TASSP.1980.1163472

[10] A. Sibbald, M. Howle. Ambient noise reduction. Wolfson Microelectronics Plc., assignee. US Patent No. 0105447. 29 Apr. 2010. Print.

[11] S. M. Short, W. A. Hartselle, V. Meadows. Ambient noise cancellation for voice communication device. AT&T Intellectual Property I, L.P., assignee. US Patent No. 9369799. 14 Jun. 2016. Print.

[12] Y. Kaneda and J. Ohga, "Adaptive microphone-array system for noise reduction," in *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 6, pp. 1391-1400, Dec 1986. doi: 10.1109/TASSP.1986.1164975

[13] Y. Kaneda, "Adaptive microphone array system for noise reduction (AMNOR) and its performance studies," *1988., IEEE International Symposium on Circuits and Systems*, Espoo, 1988, pp. 2395-2398 vol.3.
doi: 10.1109/ISCAS.1988.15425

[14] V. Pokorný and S. Mišák, "Elimination the noise of electrical transformers by antiphase," *2011 10th International Conference on Environment and Electrical Engineering*, Rome, 2011, pp. 1-4.
doi: 10.1109/EEEIC.2011.5874628

[15] H. H. Hall, "Analysis of Sound Waves," in *Journal of the Society of Motion Picture Engineers*, vol. 27, no. 4, pp. 396-408, Oct. 1936.
doi: 10.5594/J14740

[16] L. Paul. Process of silencing sound oscillations. Lueg Paul, assignee. US Patent No. 2043416. 9 June 1936. Print.

[17] J. Rehmeyer. "Equation: The Formula Behind Noise-Cancelling Headphones." *Wired*., 31 May 2011. Web.

[18] A. Bose, Headphoning. US Patent No. 4,455,675. 19 June 1984. Print.

[19] R. Falcão. *Adaptive Filtering Algorithms for Noise Cancellation*. Diss. Universidade do Porto, 2012. Print.

[20] Active Noise Control, December 2005. Aleksandar Milosevic and Urs Schaufelberger