What is the Project title?
JACK: Joint Audio Correction Kit

Who is in the group?
Cooper Barth, Andrew Finke, Jack Wiig

Give a high-level description of what you want to do. (a couple of sentences) Are you solving a problem? Duplicating a system you think is cool? Running an experiment?
The objective of our project is to create an audio correction kit consisting of two parts: an interface for noise reduction in audio files and software for active noise-cancellation. The first portion of this project will mimic other existing noise-reduction solutions while providing a clean one-click interface for users to easily utilize. The second portion of this project solves an existing problem heavily explored in the hardware spaced but underexplored in software.

Explain why this is this an interesting/useful/cool thing to do (a paragraph)
The first part of the project, the audio file noise reduction, will act as an exercise in meaningful learning through development. Noise reduction software that works on audio files has already been integrated into software like Adobe Audition. However, building this software ourselves will give us a deeper understanding of the process behind noise detection and reduction. The second part of the project, software-based active noise cancellation, is an interesting problem simply because it hasn't been done effectively before. Developing an effective software for active noise cancellation would be very impressive for us as students if we could pull it off.

What prior art is there?
Noise reduction and cancellation are important qualities in higher quality headphones. Modern solutions perform this in real time but are often much costlier than their non-cancelling counterparts. Research on noise cancellation and reduction in real time has been ongoing, with some early sources being a paper that attempts to isolate the voice in a transmission from the high-ambient background[1]. This paper was published in 1980 and found that using different noise suppression factors and pre-filtering helped isolate the noise. In more modern times, researchers are looking for predictive noise cancellation[2]. Active noise control systems are not quick enough to do real-time cancellation, so they must be use predictive methods to be useful. We then looked at the hardware used noise cancelling headphones to understand what existing solutions use. The paper we found explored the circuit that controls the active noise cancellation[3]. In order to accomplish noise reduction, we plan to use the SD-ROM algorithm[4], which was selected by one study[5] as the most effective for noise cancellation in audio.

What is the EXACT TASK that your system will do?
For noise reduction, we will take an audio file with some noise as input on our website. The website's Python backend will then analyze the audio track for noise and remove it from the track. After doing so, we will display the waveforms for both the original audio track and the new, noise reduced track. The user will also be able to play and download an mp3 of the modified track.

For noise cancelling, because a client-server approach might be too slow for real-time noise cancelation, we will build a command-line tool that will take samples of the room's background noise from the headphones' microphone and cancel out the received noise.

What measure will you will use to evaluate performance of what you build?
For noise reduction and noise cancellation, we are going to use a logarithmic signal-to-noise ratio to evaluate the effectiveness of our project. A signal-to-noise ratio is computed by the average power of the signal divided by the average power of the noise associated with that signal. We can get the average power of each signal using the power spectrum, which is the square of the magnitude spectrum. Our goal will be maximize the ratio, meaning that we are successfully eliminating noise.

Is there a data set that your system can be tried on? If so, what is it (give links). If not, explain why not?
We will test our system on two samples of audio files. One of these sets will be generated via Audacity; we will take a variety of "clean" mp3 files and add white noise to them in Audacity and test them on the system. The other set will be a set of audio files that we make ourselves in noisy environments, to test less constant noise.

What is the baseline approach you will compare your system to?
We will be comparing our noise-reduction system to existing technologies that exist to apply noise reduction, such as Adobe Audition. For the real-time noise cancellation, we will be comparing our software solution to existing hardware solutions, such as noise-cancelling headphones.

Describe any software will you need to write.
We will need to write the web interface for uploading the audio track, the interface for displaying the original track and the adjusted one, and the python backend, for which we will need to implement a noise reduction algorithm and an algorithm for soft noise reduction.

What are potential obstacles to success?
One obstacle in our path is the practical implementation of the SD-ROM algorithm for noise reduction. We haven't worked with this algorithm before, so there may be challenges in its implementation. Another obstacle in our path to success is the challenge of tackling real-time noise cancellation through software alone. This is a problem that has not been solved effectively before, and may likely take more knowledge and time than we have to complete.

How will you QUICKLY determine if these obstacles will stop you?
We plan to execute our plans for development very early in the project timeframe. As such, we will know if we hit any serious roadblocks earlier, and will be able to reach out for help as soon as possible.

What other tasks (besides coding) will you need to do?
The main task that we will need to work on is a comprehensive reading of literature. As we will likely be applying many existing algorithms, we will need to source from other research which has developed these algorithms and software.

Milestones
By the first meeting we will have completed a basic web interface and Python backend that supports uploading audio tracks and displaying their waveforms.

By the second meeting we will have completed our noise reduction algorithm to allow the user to upload audio tracks and reduce their noise. We will also have the interface implementation that allows the user to play the resulting audio file and see its waveform and spectrogram.

By the final presentation we will have attempted to implement a soft noise cancellation algorithm. This would allow the user to play an audio track and then, using the last few seconds of microphone input, cancel out the audio of their environment during playback.

Given the tight integration of each of the components, all three team members will share equal responsibility of each milestone, programming each part as a group.

Citations

[1] Speech enhancement using a soft-decision noise suppression filter
McAulay, Robert, and Marilyn Malpass. "Speech enhancement using a soft-decision noise suppression filter." IEEE Transactions on Acoustics, Speech, and Signal Processing 28.2 (1980): 137-145.

[2] Prediction filter design for active noise cancellation headphones
Guldenschuh, Markus, and Robert Höldrich. "Prediction filter design for active noise cancellation headphones." (2013).

[3] Controller design for active noise cancellation headphones using experimental raw data
Yu, Shiang-Hwua, and Jwu-Sheng Hu. "Controller design for active noise cancellation headphones using experimental raw data." IEEE/ASME transactions on mechatronics 6.4 (2001): 483-490.

[4] Chandra, Charu, Michael S. Moore, and Sanjit K. Mitra. "An efficient method for the removal of impulse noise from speech and audio signals." *ISCAS'98. Proceedings of the 1998 IEEE International Symposium on Circuits and Systems (Cat. No. 98CH36187)*. Vol. 4. IEEE, 1998.

[5]Prasadh, S. Kshipra, Sai Sriram Natrajan, and S. Kalaivani. "Efficiency analysis of noise reduction algorithms: Analysis of the best algorithm of noise reduction from a set of algorithms." *2017 International Conference on Inventive Computing and Informatics (ICICI)*. IEEE, 2017.