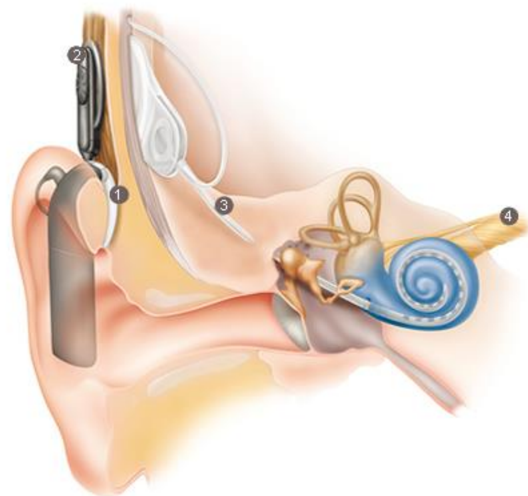
SYDE 252 project: Cochlear implant signal processing

Fall 2023, University of Waterloo
Instructor: Dr. Maftoon

In this project, you will design a rudimentary signal processor for cochlear implants. This document guides you in the design process and describes the project requirement. The project has 3 phases as will be described later.



Credit: www.cochlear.com

Learning objectives

You will follow the recipes, described in this document, and will perform some investigations. You will apply the knowledge you will acquire in the course to design a signal processor for a real biomedical device. In the design process you will be on your own to make lots of design choices, but you will need to evaluate your choices at the end. This is an easy and fun project but do not expect that everything you need will be given to you in the lectures or the textbook. This project was streamlined to cover graduate attributes Knowledge Base, Problem Analysis, Investigation and Design to name a few.

Signals and Systems in this project

Some background knowledge to start your project:

A. Your cochlear implant signal processor (a system) will read sound files as its input signals. (We don't design a real-time processor.)

B. It will process the input sound signal by dividing it into several frequency bands and simplifying it. (If you need more information about this, you can imagine that it will separate and sort pitches in the input sound signal). This will be done using filtering.

C. These segregated frequency bands will then be sent to the individual channels along the length of the cochlear implant to stimulate the nerves that naturally transmit stimulus at those frequencies. In our simplified version, the segregated frequency bands are modulated and combined to make the output signal. The output sound signal will be different from the input sound signal.

This document assumes that you will use Matlab, so all instructions are given in Matlab Syntax. However, you could use other languages if you wish. Your signal processor will cover many aspects of actual processors but not everything. In this project we only deal with sound between 100 Hz and 8 kHz.

Submit deliverables of each phase to Dropbox in Learn

Project description and deliverables

Phase 1: Preparations

Look at the schematic diagram at the end

Task 1- As it was explained in (C) above the output signal from the processor is different from the input. You are not expected to validate your design against specific sound-quality targets. However, you need to create an evaluation methodology to evaluate your various signal-processor design alternatives and parameters. This evaluation method should enable you to rank your designs and select your best design using tools such as evaluation matrices and ranking methodology. The metrics devised here will be used in your *Phase 3*.

Note: Do your best to put together a good evaluation method here because a poor one will risk your grades for this phase and Phase 3 because with a poor method here you cannot have the good evaluation expected in Phase 3. In this phase, you need to create a finalized definitive qualitative evaluation method that can generate numbers for ranking of the signal processor output. All design projects in the industrial world start with defining the evaluation process among other things and before starting the design, you should clearly know how you want to evaluate your design.

Hint 1: Although it is difficult to have an exhaustive list of parameters, pay attention to the overall goal in order to develop your evaluation metrics. Given an input audio, the aim of the signal processor is to obtain the best output signal in terms of clarity, word intelligibility and similarity as some examples. Also, it is important to also consider that the cochlear implant should perform well for a variety of sounds we hear and listening scenarios in our day to day lives. These include but not limited to sound types (male speech, female speech, child speech, etc), various listening environments (noisy, quiet, home, class, etc) and scenarios (single speaker, multi speaker, etc).

Hint 2: One way for generating numbers from a qualitative method can be using questionnaires, and a scoring scheme e.g., from 1 to 5 where 1 represents the worst and 5 represents the best scores.

Task 1 deliverable: Document your evaluation method (4 pages or shorter). What kind of metrics will you use in your Phase 3 to rank different signal processors that you will create? Here we want to make sure **you created an evaluation method** to assess the final product from the end user prospective (and not the process of developing the signal processor or the fundamental knowledge about it) which could be subjective rating provided by human listeners comparing the input audio signal with the generated output audio signal.

Task 2- Record or collect sound files (each should be 30 sec or less; avoid signals longer than 60 sec) to be used in your evaluation process (justify your choices using the deliverable of Task 1).

Suggestions: you can record sound using your cell phone, computer or other means. Pay attention to the appropriate format you will need in Task 3. Can you record directly in Matlab? You can use standardized sound files from the internet also.

Note: Collect/generate well planned sound files that will be used in your evaluation methodology. Do not write about what is good and what is bad. You should collect everything now as in Phase 3 there will be no time to do this.

Task 2 deliverable: Submit your sound files. These files will be used to verify your deliverables of Task 3.

Task 3- Phase-1 programming:

3.1. Create a program to read these files (or some of these files) into Matlab and find their sampling rate. (**Hint:** You can read the Matlab help pages about the command `audioread`[1].) (see 3.6 below before starting this task).

3.2. Use `size` to check whether the input sound is stereo (2 channels/columns) or mono (1 channel/column). Use `if`: If stereo, add the two columns to make it single channel (or a 1-column array).

3.3. Play the sound in Matlab.

3.4. Write the sound to a new file.

3.5. Plot the sound waveform of one of your sound files as a function of the sample number.

3.6. If the sampling rate of the input signal is not 16 kHz, downsample it to 16 kHz (`resample`[2]). (If the original sampling rate is less than 16 kHz, it is much better to redo 3.1)

3.7. Generate a signal using the cosine function that oscillates at 1 kHz with the same time duration and array length (Task 3.2) as the input signal (you know the sampling rate, right?). Play the sound generated by this signal and plot two cycles of its waveform as a function of time.

Suggestion: You could put all the above steps in a `function` and then apply the same processing routine to several sound files by calling your function.

Task 3 deliverables: Your Matlab file (.m file) (or in other languages) and include the plots of Tasks 3.5 and 3.7 in your report.

---

[1] `Courier` font indicates a Matlab command.

[2] Install Signal Processing Toolbox of Matlab

Phase 2: Filter design

Comment out the lines you programmed for the cosine function of Task 3.7 for now (you will get back to it in Phase 3).

Look at the schematic diagram at the end

Task 4- Programming bandpass filter bank: design a bank of bandpass filters that filters (splits) the sound to N frequency bands between 100 Hz and 8 kHz. These N frequency bands represent the N channels distributed along the length of the cochlear implant.

Suggestion: Think about your choices for the number of channels (N) and how you want to distribute the frequency range among the channels.

The Matlab filter examples in your book can give you some ideas. Matlab has some filter design tools that you can use also (e.g., `filterDesigner`). Here are some example choices to consider in your design:

- IIR vs FIR filters
- Various filter types

Hint: If you use one of the Matlab built-in filters, pay attention to the normalized frequency (π radian/sample) vs the physical frequency in Hz. To convert Hz to normalized frequency divide it to half the sampling rate and vice versa.

Suggestion: Instead of repeatedly typing commands for channels, try to define your filters in a function and call the function with different parameters and bands in `for` loops.

Task 5- Filter the sound with the passband bank.

Task 6- Plot the output signals of the lowest and highest frequency channels.

Task 7- Envelop extraction step 1: rectify the output signals of all bandpass filters. (**Hint:** Use `abs`)

Task 8- Envelop extraction step 2: detect the envelopes of all rectified signals using a lowpass filter with 400 Hz cutoff. In the design of this filter pay attention to the choices you have (similar to Task 4).

> **Note: You should implement envelop extraction yourself. Using pre-written or built-in MATLAB functions (or in other languages) for envelop extraction is not acceptable. You are expected to do envelop extraction with the above method to understand effects of filters on signals.**

Task 9- Plot the extracted envelope of the lowest and highest frequency channels.

Deliverables of Phase 2:

- Submit a short report (6 pages or shorter) documenting your filter design activities in Tasks 4 and 8. Include plots of Tasks 6 and 9 in this report.
- Submit your Matlab file (.m file) or your file in other languages.

Phase 3: Final product and wrap up

Task 10- Start this task based on the exercise that you had in Task 3.7. For each channel, generate a signal using a cosine function that oscillate with the central frequency of each of the bandpass filters (instead of the original 200 Hz in Task 3.7) with the length of the rectified signals. (**Hint:** Use `for` loop)

Task 11- In a real cochlear implant, the output of each of the bandpass filters are used to make the signal passed to each of the electrode channels. However, in this project we will use these outputs rather differently to be able to hear the output without implanting our ears! For each channel, amplitude modulate the cosine signal of Task 10 using the rectified signal of that channel (Task 8).

Task 12- Add all signals of Task 11 together. This is your output signal. (**Suggestion:** Normalize this signal by the maximum of its absolute value.)

Task 13- Play the output sound in Matlab. Write the sound to a new file.

Task 14- Use the material you prepared in Task 1 and 2 in Phase 1 to evaluate your design.

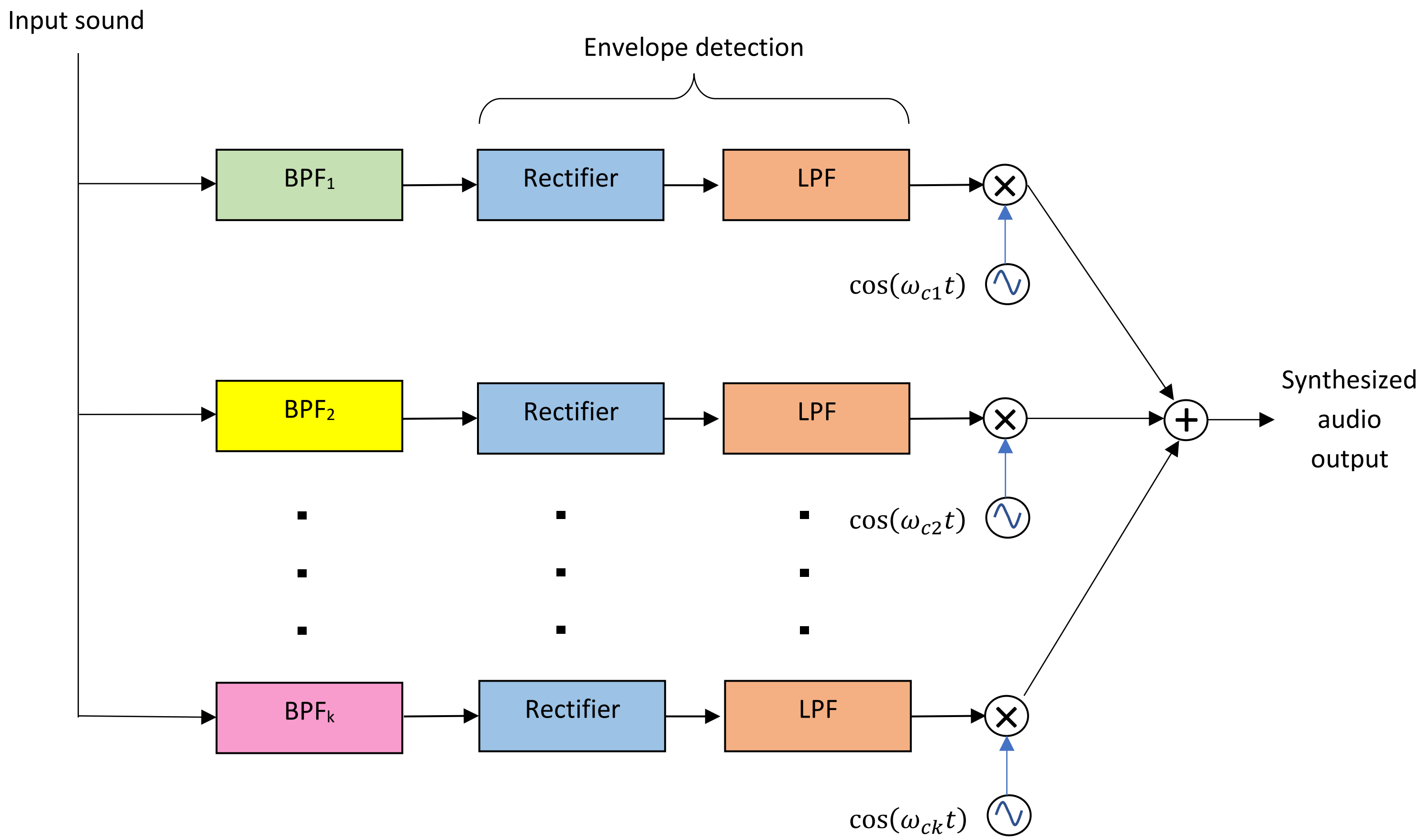
Task 15- Use your evaluation method and iterate your design by varying the parameters of your signal processor and altering the design choices you made in Tasks 4 and 8. Here are some more example choices to think about in your design:

- Linear spacing of sub bands between 100 Hz and 8 kHz vs other types of spacing
- Overlapping sub bands
- IIR vs FIR filters
- Various filter types
- What happens if the cutoff frequency of the envelop detecting lowpass filter (Task 8) is set at higher or lower frequencies?

Task 16- Study the effects of the number of channels in Task 4. Increasing the number of channels, increases the processing. This translates into bulkier and more expensive hardware and worse battery life. What is the optimum number of channels according to your evaluation?

Bonus Task- Now that you know a lot about signals and systems can you add another evaluation metric to your method that evaluates your design quantitatively?

Deliverables of Phase 3: Write a report that briefly discusses your activities and conclusions in Tasks 14, 15 and 16 (+ Bonus Task) and your recommended best design based on your evaluations and design iterations (6 pages or shorter). Talk about the advantages and also the setback of the specific filtering technique. Submit your final Matlab file of your best design, input signals (sound files) and output signals (sound files).



Schematic diagram of the cochlear implant signal processor
BPF: band-pass filter, LPF: low-pass filter