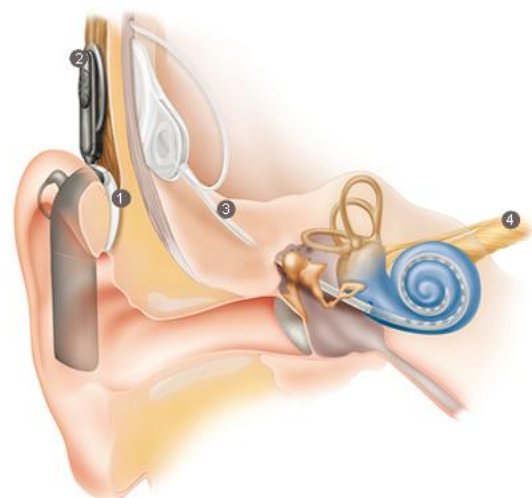
BME 252 project: Cochlear implant signal processing

Summer 2018, 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 described later.



Credit: www.cochlear.com

Learning objectives

In this project, you will follow the recipes described in this document and perform some investigations. You will apply the knowledge you 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 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:

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

II. 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.

III. These segregated frequency bands are then send to the individual channels along the length of the cochlear implant and the cochlea to stimulate 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 is different from the input sound signal.

You will create the signal processor in Matlab. 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 III 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 but does not need to be comprehensive or perfect.

> **Hint:** Pay attention to various sound types and various listening environment and scenarios. **Task 1 deliverable:** document your evaluation method. Discuss about included and excluded criteria in your evaluation.

Task 2- Record or find sound files 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.

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`). (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 (`cos`) 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 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) and plot file of the Tasks 3.5 and 3.7.

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

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).

Task 4- Programming bandpass filter bank: design a bank of bandpass filters that filters 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. 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 in Phase 3. The Matlab filter examples in your book can give you some idea. Matlab has some filter design tools that you can use also. Here are some example choices to think about in your design:

- IIR vs FIR filters
- Various filter types

Hint: if you use one of 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 dived it to half the sampling rate and vice versa.

Suggestion: Instead of repeatedly typing commands for channels, try using `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).

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

Deliverables of Phase 2:

- Submit a short report 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).

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- 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.

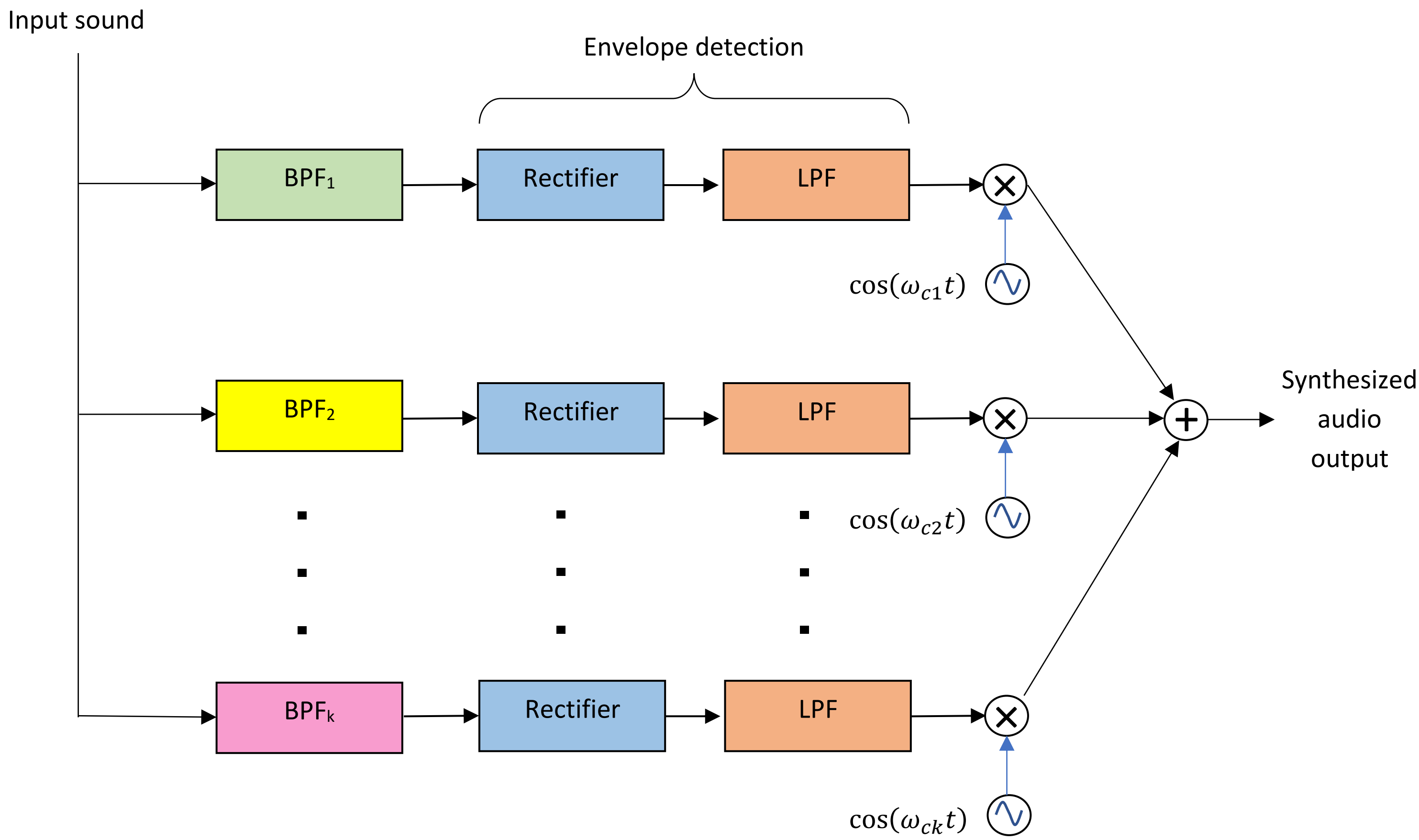
Task 15- Iterate your design by varying the parameters of your signal processor and altering the design choices you made in Tasks 4 and 8. Here are some more example choices to think about in your design:

- 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 index 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. 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