Lab 15a: Cochlear Implant Simulation with a Filter Bank

Pre-Lab and Warm-Up: You should read at least the Pre-Lab and Warm-up sections of this lab assignment and go over all exercises in the Pre-Lab section before going to your assigned lab session.

Verification: The Warm-up section of each lab must be completed **during your assigned Lab time** and the steps marked *Instructor Verification* must also be signed off **during the lab time**. One of the laboratory instructors must verify the appropriate steps by signing on the **Instructor Verification** line. When you have completed a step that requires verification, simply demonstrate the step to the instructor. Turn in the completed verification sheet to your instructor when you leave the lab.

Lab Report: Write a lab report on Section 4 with graphs and explanations. Please **label** the axes of your plots and include a title for every plot. In order to keep track of plots, include each plot *inlined* within your report. If you are unsure about what is expected, ask the instructor who will grade your report.

1 Introduction

~~A Cochlear Implant (CI) serves to overcome deafness by directly stimulating auditory nerve fibers with electrical current thereby conveying auditory cues. The goal of this lab is to emulate the speech processing function of a cochlear implant.[1,2] To better understand the internal functions of a cochlear implant device, it is useful to review the process of hearing. The ear is divided into three parts: outer, middle and inner as shown in Fig. 1a. During the process of normal hearing, the outer ear captures sound, which the middle ear then converts into mechanical vibrations. These vibrations travel into the cochlea, a coiled fluid-filled tube, located in the inner ear. Dividing the fluid-filled tube is a membrane, the basilar membrane that exhibits a variable mechanical stiffness. As a result, it is more sensitive to high frequencies at the base, and more sensitive to low frequencies near the apex (see Fig. 1b). The fluid displacement reveals (or encodes) the frequency information of the acoustic signal. Attached to the membrane are small hair cells lined with stereocilia that bend when the membrane is displaced. When bent, these hair cells activate neurons that fire signals to the brain relaying acoustic information. Deafness is caused by the disruption of this hearing process. In the majority of the cases, it is the hair cells that are damaged, not the auditory nerves. A cochlear implant functions by directly stimulating the auditory nerves with patterns of electrical current determined by speech processing.[3,4]~~

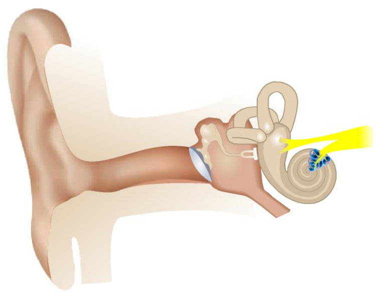
~~Figure 2a depicts the four external parts of a cochlear implant. Understanding the signal processing inside the speech processor is the focus of this lab. Figure 3 shows a block diagram of the major signal processing blocks.~~

~~[1]P. T. Bhatti and J. H. McClellan, "A Cochlear Implant Signal Processing Lab: Exploration of a Problem-Based Learning Exercise," *IEEE Transactions on Education,* Vol. 54, No. 4, pp. 628–636, 2011. DOI: 10.1109/TE.2010.2103317~~

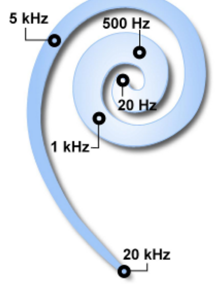
~~[2]C. Luo, J. H. McClellan and P. T. Bhatti, "Introductory signal processing labs based on filterbank applications," *2011 IEEE Digital Signal Processing Workshop and IEEE Signal Processing Education Workshop (DSP/SPE),* Sedona, AZ, pp. 90–94, 2011. DOI: 10.1109/DSP-SPE.2011.5739192~~

~~[3]P. C. Loizou, "Introduction to Cochlear Implants," *IEEE Engineering in Medicine and Biology Magazine,* Vol. 18, No. 1, pp. 32–42, 1999. DOI: 10.1109/51.740962.~~

~~[4]A tutorial article can be found here (April, 2016): `http://ecs.utdallas.edu/loizou/cimplants/tutorial/`~~
~~P. C. Loizou, "Mimicking the human ear," *IEEE Signal Processing Magazine,* Vol. 15, No. 5, pp. 101–130, 1998. DOI: 10.1109/79.708543~~

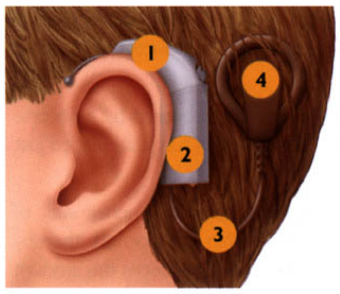


~~(a) Peripheral auditory system.~~

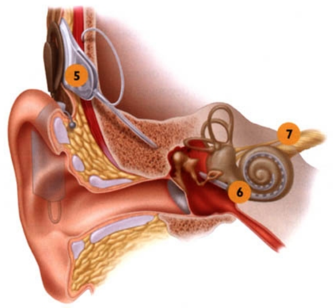


~~(b) Frequency-to-place mapping of the cochlea.~~

~~Figure 1: (a) The ear is divided into an outer, middle, and inner ear. The cochlea is located in the inner ear. (b) Different acoustic frequencies activate different parts of the basilar membrane, and therefore the cochlea. The acoustic frequency as a function of place is indicated in Hz.~~



~~(a) External parts of a cochlear implant.~~



~~(b) Internal parts of a cochlear implant.~~

~~Figure 2: Cochlear Prosthesis Components. (a) External components are (1) microphone, (2) external speech processor, (3) cable to transmitter, (4) transmitter coil. (b) Internal components are (5) receiver/stimulator, (6) electrode array, and (7) vestibulocochlear nerve.~~

2 Pre-Lab

In this lab, the student will hear sounds that emulate what a patient with a Cochlear Implant (CI) hears. A sound file will undergo speech processing similar to that of a cochlear implant. The following sections describe the major signal processing blocks needed to decompose a speech (or audio) signal into frequency components that are consistent with the frequency-to-place mapping of the cochlea.

2.1 Cochlear Implant Signal Processing

~~The external parts of a cochlear implant consist of a microphone, a speech processor, a transmitter/receiver, and an electrode array as shown in Fig. 2a. The speech processor is the function that can be emulated in MATLAB. There are four steps in cochlear speech processing: Pre-emphasis, multiple band pass filters (BPFs) in a *filter bank* structure, envelope detection, and bipolar pulse generation. Figure 3 shows the signal flow through these blocks. The last step, bipolar pulse generation, is needed to produce the appropriate signal to send to the current stimulator, and then onto the electrode array, causing nerve stimulation. This step (of current stimulation) will not be duplicated in the MATLAB code. Instead, the fourth step is envelope modulation and summation of the channels, which allows the processed sound to be heard.~~

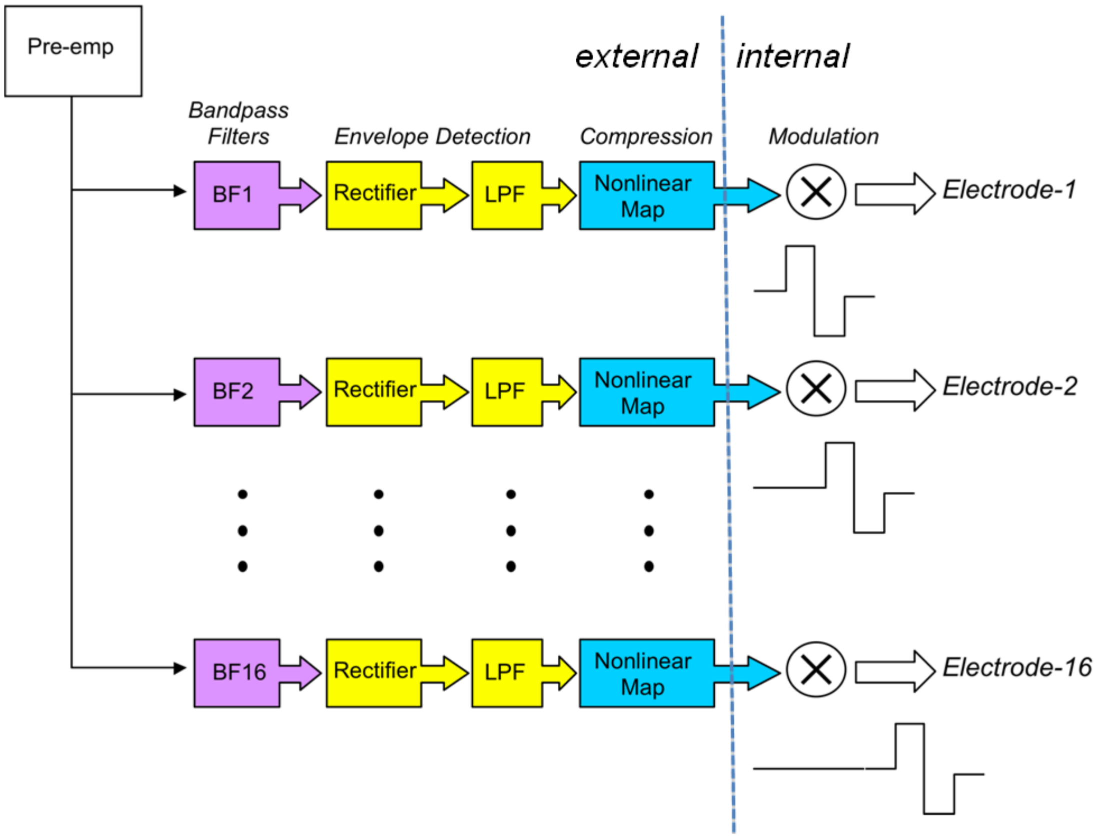


Figure 3: Multi-channel filter bank for analyzing speech; sixteen channels shown.

Table 1: Center frequencies and bandwidths for an 8-channel filter bank (frequencies in Hz)

n	1	2	3	4	5	6	7	8
f_c	394	692	1064	1528	2109	2834	3740	4871
B	265	331	413	516	645	805	1006	1257

2.1.1 Pre-emphasis Filter

~~A pre-emphasis filter is often used to amplify high frequencies and attenuate low frequencies prior to the rest of the processing. This filtering enhances the low-energy, high-frequency consonants with respect to~~ the high-energy, low-frequency vowels in a speech signal. A simple FIR filter such as a *first-difference* ~~filter can provide this capability. Recall that a first-difference filter would completely remove DC from the input speech waveform. The low frequency cutoff for human hearing is generally taken as 20 Hz and most microphones and speakers used on PCs are unable to record or play frequencies in these low ranges anyway.~~

2.1.2 Bandpass Filter Bank

At this point in the speech processing, the pre-emphasized sound waves are separated into parallel channels that are narrow frequency bands. Table 1 gives the center frequency and bandwidth values for an 8-channel filter bank. Each channel in the filter bank employs a bandpass filter (BPF) to separate out a small range of frequency components of sound signal, i.e., the frequencies within its pass band. The ~~pass band of each BPF~~ is from $f_c - \frac{1}{2}B$ to $f_c + \frac{1}{2}B$, so the width of the pass band is B. For example, the seventh channel in the ~~8-channel system given in Table 1 has a BPF that passes frequencies from 3237 Hz to 4243 Hz.~~

When designing the CI filter bank, a major question would be how many frequency channels to use, and how to choose the passband width of the channels. With too few channels the reconstructed sound will be unintelligible; however, with too many channels the computational load from the signal processing is too great which makes it difficult the implement a real-time system that is (relatively) inexpensive and has low power dissipation for longer battery life. The choice of center frequencies and bandwidths for the filter bank must be consistent with human perception that depends on the frequency-to-place mapping of the cochlea. Therefore, it turns out that the bandwidths (B) of the filter banks are logarithmically spaced, i.e., the difference between successive $\log(B)$ values is constant. Equivalently, the ratio between bandwidths of successive channels is constant.

2.1.3 FYI: Defining the Channel Bandwidths

The BPF specifications were derived in a study examining the number of channels needed to understand speech.[5] Here is a simple algorithm for choosing the bandwidths in a constant-bandwidth-ratio N-channel filter bank:

1. Determine the minimum and maximum frequencies to be covered by the filter bank, say $[f_a, f_b]$.

2. The bandwidths of neighboring channels are defined as $B_n = \rho B_{n-1}$, where ρ is a constant (greater than one) that defines the ratio of successive bandwidths. Decide on values for N and ρ, e.g., in the 8-channel filter bank of Table 1, $\rho = 5/4 = 1.25$, and $[f_a, f_b] = [262, 5500]$.
   Note: the numbers in the table were obtained by rounding f_c and B to integer values.
   The sixteen-channel case is $\rho = 9/8 = 1.125$, and $[f_a, f_b] = [156, 5500]$; while the four-channel case is $\rho = 2.07$, and $[f_a, f_b] = [300, 5500]$

3. Knowing N, ρ and $[f_a, f_b]$, we can solve for B_1 from the following:

$$f_b - f_a = \sum_{n=1}^{N} B_n = \sum_{n=1}^{N} \rho^{n-1} B_1 = B_1 \sum_{n=0}^{N-1} \rho^n = B_1 \frac{1-\rho^N}{1-\rho}$$

   Recall that the bandwidths are then generated from the recursion $B_n = \rho B_{n-1}$.

4. Finally, the center frequencies are generated via:

$$f_{c_1} = f_a + \tfrac{1}{2} B_1 \qquad \text{and} \qquad f_{c_n} = f_{c_{n-1}} + \tfrac{1}{2} B_{n-1} + \tfrac{1}{2} B_n$$

2.1.4 Envelope Detection

~~Once the pre-emphasized speech signal has been separated into sub-band channels by bandpass filtering, detection of the envelope for each sub-band signal is necessary. Each BPF output is modeled as a narrowband signal consisting of a slowly varying *envelope* signal that multiplies the high-frequency sinusoid whose frequency equals the center frequency of the sub-band channel. This model is, in effect, an AM (amplitude modulation) representation of the narrowband signal. Envelope extraction requires two steps: full-wave rectification and low-pass filtering. Full-wave rectification obtains the instantaneous magnitude of the sub-band signal and then low-pass filtering then smooths out the signal, and thus removes the high frequency nature of the sinusoid at the center frequency.~~
~~*Note:* The envelope is needed because it is a low-bandwidth signal that is used to modulate the current pulses that stimulate nerve fibers as described in Section 2.1.5.~~

[5]P. C. Loizou, et. al., "On the number of channels needed to understand speech," *J. Acoustical Soc. of America,* vol. 106, pp, 2097–2103, 1999.

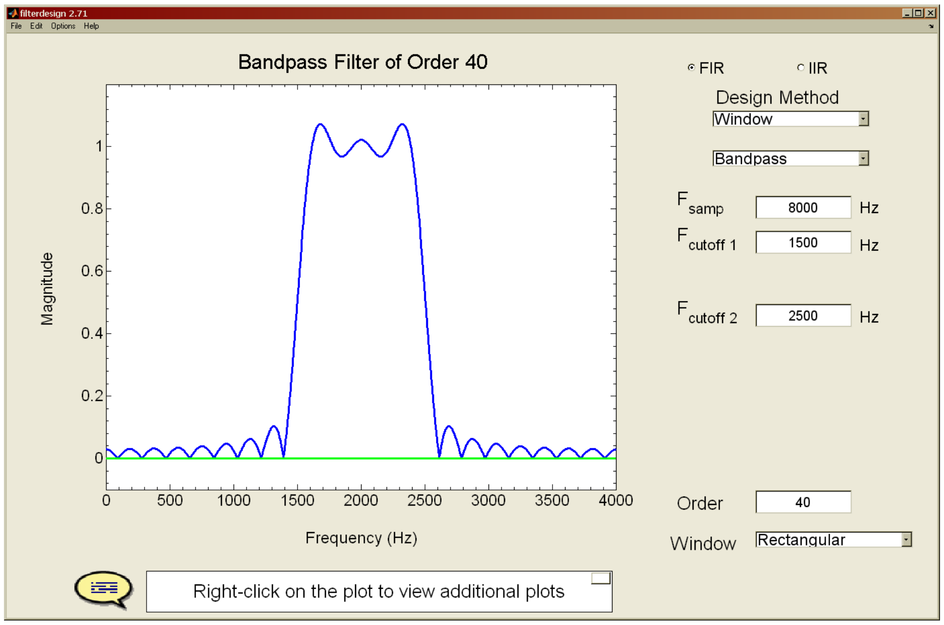
2.1.5 Envelope Modulation

The external signal processor for the cochlear implant produces the envelope signals, one for each channel, and then transmits these signals to the internal receiver which can then use the envelopes to stimulate auditory nerve fibers at different locations along the electrode array. Amplitude-modulated biphasic signals are used to stimulate with a current output at each electrode site. *Location along the cochlea maps to frequency, so each frequency band in the filter bank is assigned to an individual electrode position.* For example, the first channel (lowest frequency) in the filter bank corresponds to the most apical (deepest) electrode site, hence the lowest frequency for stimulation (see Fig. 1b). Similarly, the highest-frequency channel in the filter bank corresponds to the most basal electrode site, the highest frequency.

Signal processing blocks cannot simulate the interaction between the electrodes and the auditory nerve fibers that create the different frequency sounds we hear. Instead, we must create an acoustic simulation of a CI signal processor, meaning that our simulation should attempt to be "acoustically faithful." We use the slowly varying envelope signal from each channel to restore a signal at the center frequency of that channel, i.e., the envelope for the n-th channel modulates a sinusoid whose frequency is set to the center frequency of the n-th channel.[6] The sinusoids from all the channels are then summed to produce a speech signal that emulates what is heard by a patient with a cochlear implant.

2.2 GUI for Filter Design

For the CI system, many filters are needed. The MATLAB GUI called **`filterdesign`** illustrates several filter design methods for LPF, BPF and HPF filters. The interface is shown in Fig. 4. If you have installed the *SP-First* (or *DSP-First*) Toolbox, you should already have this demo on the `matlabpath`.



[✓ FIR] IIR

Design Method: Window

Bandpass

F_{samp} 8000 Hz

$F_{cutoff\ 1}$ 1500 Hz

$F_{cutoff\ 2}$ 2500 Hz

Order 40

Window Rectangular

Right-click on the plot to view additional plots

Figure 4: Interface for the **`filterdesign`** GUI. The default setting for the frequency axis is analog frequency f in Hz, but can be changed to $\hat{\omega}$ via the `Edit->Show Radian Frequency` menu selection. When the Filter Choice is set to $\boxed{\texttt{FIR Filters}}$, many different window types can be selected, including the Hamming window and the Rectangular window (i.e., no window).

[6]In reality "The amplitudes of the sinusoids are computed by estimating the root-mean- square (rms) energy of the envelopes every 4 msecs." Loizou ref.

Both FIR and IIR filters can be designed, but we are primarily interested in the FIR case, which is selected from the radio button in the upper right. Once $\boxed{\texttt{FIR}}$ is selected, the window type should be selected from the drop-down list in the lower right. Lastly, it is necessary to set the order of the FIR filter and the cutoff frequency; these parameters can be entered in the edit boxes. Use the **`filterdesign`** GUI to design some FIR filters with the Hamming window method. To exercise your filter design skills, produce a length-31 lowpass filter with its cutoff frequency at 1000 Hz when the sampling rate is 8000 Hz. Use the *export* feature of the GUI to transfer the filter coefficients into the workspace.

2.3 FIR Bandpass Filter Design

A bandpass filter design function is needed for this lab. Linear-phase FIR bandpass filters can be obtained directly from Hamming-sinc design formula by subtracting LPFs. In a previous lab, you may have experi-

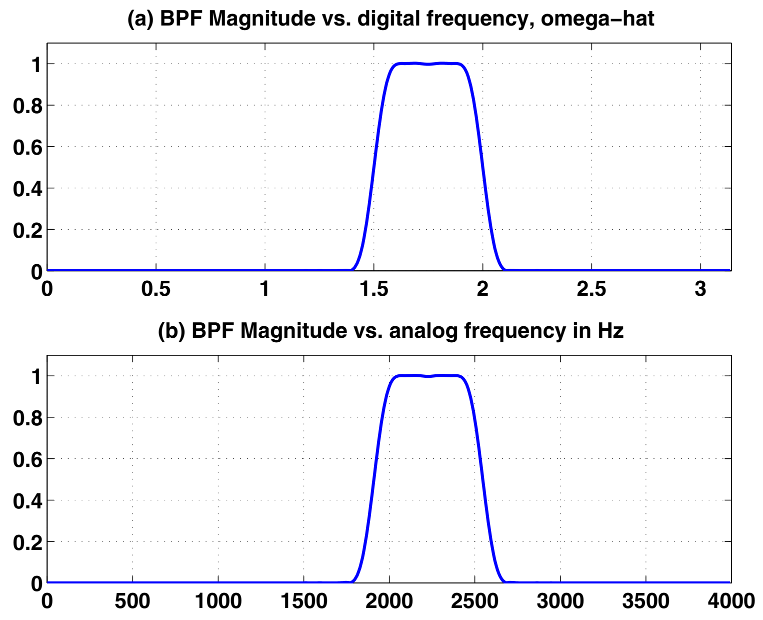


Figure 5: Frequency response of a length-101 bandpass filter (BPFs) created via the MATLAB call `designBPF(100,1.5,2)`. (a) $\left|H_B(e^{j\hat{\omega}})\right|$ plotted versus $\hat{\omega}$, and (b) the same frequency response versus analog frequency f (Hz), assuming that $f_s = 8000$ Hz.

mented with FIR bandpass filters whose frequency response looks like Fig. 5. For this lab you must write a MATLAB M-file called `designBPF(100,1.5,2)` to design these filters.

3 Warm-up

3.1 Pre-emphasis Filter

Create a pre-emphasis filter to amplify the high-frequency sound content of a signal.

(a) Define a pre-emphasis filter as $H(z) = 1 - z^{-1}$, which is a first-difference FIR filter. Make a plot of its frequency response (magnitude only) by using `freqz` and `plot`. Notice what type of filter $H(z)$ is, i.e., FIR or IIR; and also desribe its frequency characteristic as LPF, HPF, BPF, or something else.

(b) Apply $H(z)$ to a speech signal, e.g., `catsdogs.wav`. To show the effect of the filter, make spectrograms of both the input signal and the output signal. Describe at least two features in the spectrograms that confirm the frequency response behavior of the pre-emphasis filter.

$\boxed{\textbf{Instructor Verification} \text{ (separate page)}}$

(c) Listen to the speech signal before and after pre-emphasis. Describe the difference that you hear in high-frequency spectral content? High-quality headphones would make this easier.

3.2 Envelopes via Full-Wave Rectification and Lowpass Filtering

The output of each BPF in a filter bank is a *narrowband* signal because it has frequency components only near the center frequency of the BPF. An approximate model for the narrowband output signal is the AM model that expresses the signal as the product of two components: a sinusoid at the center frequency of the channel and a *slowly varying (nonnegative) envelope* that changes the amplitude of the sinusoid versus time. This representation is approximate, but the AM (amplitude-modulated) sinusoid is well understood, so the model is easy to work with.

The objective in *envelope extraction* is to separate out the slowly varying envelope signal from the high frequency sinusoid (at the center frequency of the BPF channel). The envelope signal has its spectrum centered at zero frequency and is a signal with relatively low bandwidth. One common way to extract the envelope is to use a cascade system consisting of a full-wave rectifier (i.e., a magnitude device) followed by a lowpass filter. These two operations are treated in more detail in this part of the warm-up.

To demonstrate the processing we need a test signal that is the product of two components: a slowly varying component and a much higher frequency component. The simplest example of this sort is the *AM signal* studied in Chapter 3 and given below

$$b(t) = b_1(t)b_2(t) = (\beta + \cos(2\pi f_1 t + \varphi_1))\cos(2\pi f_2 t + \varphi_2) \qquad \text{where } f_1 \ll f_2$$

where $b_2(t)$ is a high-frequency sinusoidal signal. The slowly varying signal $b_1(t)$ controls the amplitude of the sinusoid $b_2(t)$. This behavior would be obvious from a plot of $b(t)$ versus t.

Extracting the envelope signal, which is $b_1(t)$ in this case, from $b(t)$ can be done with two simple processing steps: (1) a magnitude operator, and (2) a lowpass filter. In a hardware implementation, a full-wave *rectifier* would be used to perform the magnitude operation.

(a) Generate an amplitude-modulated sinusoid using

$$b(t) = (\beta + \cos(2\pi f_1 t))\cos(2\pi f_2 t)$$

Pick $f_1 = 215\,\text{Hz}$, $f_2 = 1064\,\text{Hz}$, and $\beta = 1.2$, which corresponds to the third channel in the 8-channel filter bank. Use a sampling rate of $f_s = 8000\,\text{Hz}$ to create the vector of signal values in MATLAB; call it `bt`. Create the signal vs. time for a duration of 1.3 s. In a plot of the signal, zoom in to find the feature of the plot that is the envelope of the signal.

(b) In MATLAB take the magnitude (i.e., absolute value) of `bt` and plot the magnitude signal versus time t. Once again zoom in to see details, and then point out the feature of the signal that is the envelope. Notice that you can also see the detailed high-frequency nature of the signal.

(c) Compare *two-sided* spectrograms of $b(t)$, $b_1(t)$ and $|b(t)|$ to see where these three signals have frequency content. *Note:* You might have to use a long window length, e.g., 512 or 1024, to see the individual spectrum lines. Use the *two-sided* spectrograms to justify that a LPF applied to the magnitude signal $|b(t)|$ does, in fact, yield the envelope $b_1(t)$. In addition, use the spectrograms to define where the passband and stopband edges need to be for the LPF. In particular, how does the stopband edge depend on center frequency of the BPF channel?

$\boxed{\textbf{Instructor Verification} \text{ (separate page)}}$

(d) Since lowpass filters are well-known for making signals smooth, we need a LPF that *passes* the frequency range where we expect to find the envelope signal, and *rejects* the frequency range where the high frequency components lie. The spectrograms done in the previous part should aid in choosing the passband and stopband edges of this LPF.

Use the **`filterdesign`** GUI to create a LPF with its cutoff frequency at an appropriately chosen cutoff frequency. Try different FIR filter orders, but something less than $M = 30$ should work fine. Since the **`filterdesign`** GUI can take inputs that are analog frequencies, the cutoff frequency can be chosen directly in hertz. Use the GUI's plot of the frequency response to verify the correct locations of the passband and stopband. Finally, show the pole-zero plot (in the GUI) for the system function of the lowpass filter.

Instructor Verification (separate page)

(e) Export the filter coefficients from the GUI in order to use them with the `filter` function to process the magnitude signal $|b(t)|$.[7] Make a plot of the output signal from the LPF, which should be the filtered magnitude. Compare this output to the expected shape of the envelope. Point out similarities and differences; be aware that there might be a fixed scaling between the two plots. Notice the *transient* portion of the output signal at the beginning of the output signal—this is an interval where the output signal is unlike the rest of the waveform.

Instructor Verification (separate page)

3.3 M-file for FIR Bandpass Filter Design (Optional for Warm-up; Required for Lab Project)

(a) Write an M-file called `designBPF.m` that creates the impulse response of an FIR filter whose frequency response looks like Fig. 5. A template for `designBPF` is `bk = designBPF(M,wc1,wc2)`, where `M` is the filter order, and `wc1` and `wc2` are the lower and upper cutoff frequencies in the range $0 \le \hat{\omega} \le \pi$. When there are only two input arguments, i.e., `wc2` is missing, then the function should design a lowpass filter.

(b) Write a few lines of MATLAB code to make the plot in Fig. 5(a).

(c) Modify the plotting code in the previous part to plot the same frequency response versus a frequency axis (f) in Hz, i.e., convert $\hat{\omega}$ to f. This should display the plot in Fig. 5(b). Recall that the relationship between analog frequency and digital frequency is $\hat{\omega} = 2\pi(f/f_s)$.

Instructor Verification (separate page)

4 Lab Project

First, an observation about learning in the context of lab projects. For first-year or second-year students, labs are given with lots of step-by-step instructions. However, in the "real world" projects typically come with little or no detailed instructions. Thus, a valuable learning experience is to attempt occasionally a project without much hand-holding. If that were the case, this write-up could actually be no more than the following three-sentences.

[7]The GUI can export to the default names `num` and `den`; sometimes, the numerator coefficients are negated.

Implement the entire signal processing system shown in Fig. 6 for simulating the cochlear implant system. Once you have the system working, run tests on several types of input signals, including male/female speech and music. Finally, vary the number of channels in the filter bank and compare the results.

4.1 Special Considerations

The foregoing description might be a bit too terse, so here are some issues to consider.

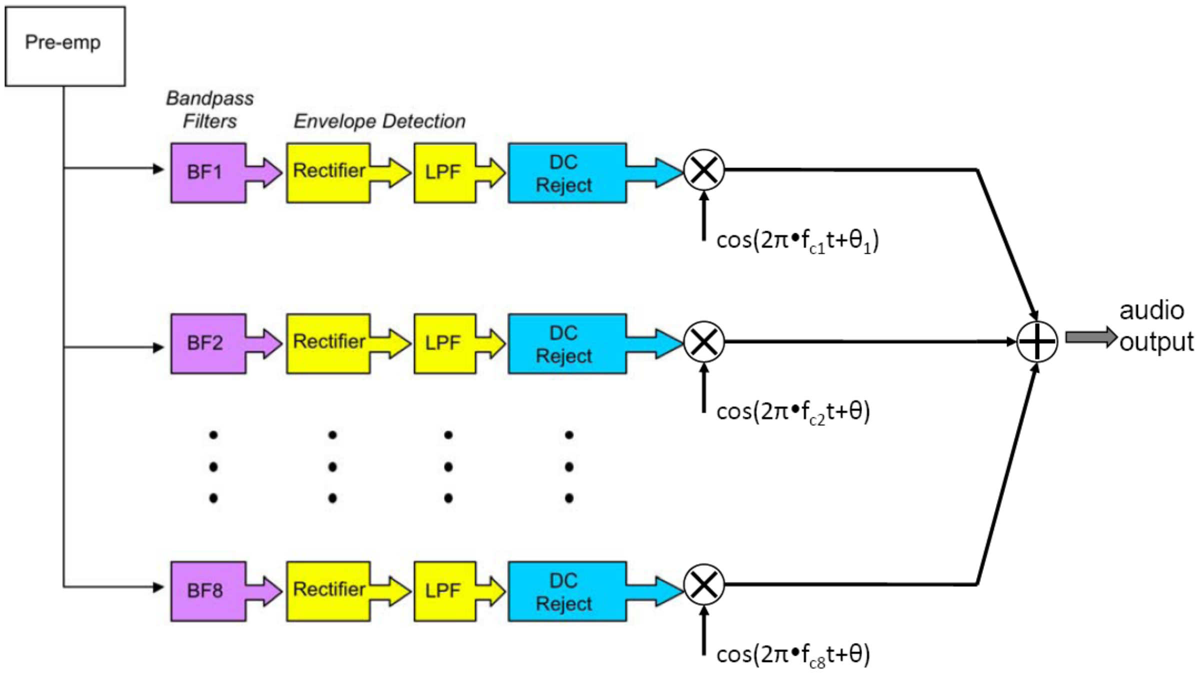


Figure 6: Acoustic simulation of the multi-channel filter bank for a cochlear prosthesis.

4.1.1 Sampling Rate

Although many of the previous labs have used speech signals sampled at $f_s = 8000$ Hz, for this lab a higher sampling rate must be used because the highest frequency BPF in the filter bank has a passband that extends up to 5500 Hz. Furthermore, the magnitude operator used in envelope extraction is a nonlinear system that produces even higher-frequency spectral components. The best choice would be $f_s = 22$ kHz or greater, so make sure that your implementation works for these sampling rates.

These rates are a convenient choice because it is relatively easy to acquire a recording at $f_s = 22.05$ kHz or $f_s = 44.1$ kHz which are common rates related to the sampling rate used on CDs. You can increase/decrease the sampling rate by using the MATLAB M-file called `resample`, e.g., you could change from $f_s = 8000$ Hz to $f_s = 22$ kHz by using a rate-conversion factor of $\frac{11}{4}$. From a previous lab you might have a recording of your own voice that you may have already studied, so that signal could be used for some of the evaluations below. Other recordings are available from the resource area of the *DSP First Companion Website*.

4.1.2 Components Available

In the warm-up, the pre-emphasis filter and the envelope extraction processes were studied. Assuming that you have successful implementations of these components, only minor modifications should be needed to use them in the final system simulation.

4.2 Complete Simulation of the Cochlear Implant System

The CI system requires many filters, so filter design is the core activity in this lab project.

4.2.1 Implement the Filter Bank with FIR BPFs

~~The filter bank consists of many BPFs running in parallel. Looking at Table 1, we see that the passband widths of the channel filters are all different. Thus it is necessary to write a loop that takes the center frequency and bandwidth of each channel, determines the cutoff frequencies of the passband, and then does the filter design for each BPF to obtain the filter coefficients. In the FIR case, the Hamming-sinc filter design~~ function (`designBPF`) would be used to get the filter coefficients. Since the outputs of the channels are ~~summed together, the lengths of all the BPFs must be the same. The reason for this is that symmetric FIR filters have a delay, and all channels must have the same delay. The Hamming-sinc formula gives an FIR~~ filter with symmetric coefficients, so the resulting filter has linear phase with a slope of $-\frac{1}{2}(L-1)$, where L is the filter length. Recall that phase slope in the frequency domain corresponds to time delay.

~~To simplify matters, the bandedges of the BPFs (in Table 1) can be used as the cutoff frequencies in the Hamming-sinc design method. One justification for this choice comes from the fact that you would like the sum of all the BPFs to be flat across frequency. Consider that the channels of the filter bank are splitting the signal into eight subsignals. A desirable property of the filter bank would be for the sum of those eight subsignals to be equal to the original signal before filtering. A restatement of this idea in the frequency domain is that the sum of the frequency responses of the eight BPFs be equal to one (in magnitude) across the entire frequency band covered by all the filters. If you design BPFs by subtracting LPFs, as suggested with the Hamming-sinc method, then you should be able to prove (or verify) that the sum of the BPFs is~~ *approximately* equal to one (across the entire bandwidth range of the filter bank).

The primary decision during the filter design process is choosing the filter length (L). In this lab, the ~~simulation has to be done with FIR filters designed via the Hamming-sinc method. Thus the only design~~ choices are cutoff frequencies and filter length (L). Once you start doing the designs, you should see that ~~very long filters are required. Thus the FIR filter bank is actually a very high complexity implementation for the speech processing because long FIR filters involve lots of multiplications. The number of multiplications~~ per output point is proportional to L. In reality, there are better filter design methods for this application; ~~ones that have lower complexity, e.g., IIR filters. However, for the task of writing the simulation in this lab, it is easier to work with the FIR case.~~

4.2.2 Envelope Extraction

The output of the BPFs are narrowband signals that can be viewed as the product of a slowly varying envelope signal with a high-frequency sinusoid at the center frequency of the BPF. The slowly varying envelope signal can be extracted in a two-step process by taking the magnitude and then averaging with a smoothing LPF.

(a) A lowpass filter (LPF) is used because the valuable frequency content of the slowly varying envelope signal is concentrated in the low-frequency region. The LPF should be designed as a Hamming-sinc FIR LPF with the passband chosen to be flat over the frequency bandwidth of each channel. One question that must be addressed is whether to use a different LPF for each channel, i.e., different passband edges. The following approaches could be studied: (1) use the same LPF in each channel, (2) use several different LPFs, e.g., one for the very narrow low-frequency BPFs, one for the wider mid-range BPFs, and one for the very high-frequency BPFs, (3) use a different LPF for each channel.

In this lab project, you must use option #3, i.e., a different LPF for each channel. There are two factors that constrain the cutoff frequency of the smoothing LPF, and these factors vary across the set of BPFs. The bandwidth of the channel BPF determines the bandwidth of the usable envelope signal; the nonlinear magnitude operator produces higher-frequency copies of the desired spectrum at integer multiples of the center frequency.

(b) *There is a complication.* The magnitude operation produces a signal that is nonnegative, so it has a nonzero average value. In other words, there is a constant offset, because the DC level is a positive value. If we view the envelope as containing the interesting temporal variations in the signal, we should argue that this constant offset conveys no information. For the filter design, this suggests that we need a filter that is lowpass-like, but one that rejects DC—maybe another BPF. However, direct design of such a modified LPF could be difficult. Instead, we can cascade two filters to do the job: the Hamming-sinc LPF from the previous part followed by a *notch filter* that removes DC.

A simple, but effective, notch filter can be produced with an *IIR system function that has a zero exactly on the unit circle together with a pole nearby.*

$$H(z) = \tfrac{1}{2}(1+a)\frac{1-z^{-1}}{1-az^{-1}}$$

The pole location ($z = a$) can be varied to sharpen the *notch* by moving the pole very close to the zero. However, the pole must remain inside the unit circle; why? Plotting a frequency response should demonstrate the *notch capability* of this filter. Plot four magnitude responses in the same figure, for values of $a = 0.95, 0.98, 0.99, 0.995$, showing only the region $0 \le \hat{\omega} \le 0.25\pi$. For each of these four filters, determine the band edge frequency $\hat{\omega}_a$ where $|H(e^{j\hat{\omega}})| \ge 0.9$ for $\hat{\omega}_a \le \hat{\omega} \le \pi$.

(c) When the notch filter is used, DC is eliminated but the notch has a bandwidth, i.e., the width measured by band edge $\hat{\omega}_a$ at the 0.9 level. Assuming that the bandwidth of the notch must be 100 Hz or less when $f_s = 22.050$ kHz, determine the value of the pole a.

(d) Pick one channel and plot three frequency responses: the Hamming-sinc LPF from part (a), the DC-notch filter from part (c), and the cascade system consisting of these two.

4.2.3 Envelope Modulation

The envelope modulation is done by multiplying the slowly varying extracted-envelope signal by a high frequency signal that represents the channel (Fig. 6). For the acoustic simulation done in this lab, the appropriate choice would be a sinusoid at the center frequency of the channel.

Note-1: Excellent sound quality is *not* the primary objective when implementing the CI filter bank. Instead, the objective is to create a sound output that portrays how the user of a CI device would experience sound. On the other hand, the sound output should be intelligible even if the quality is poor.

Note-2: The sinusoidal modulation is an easy way to combine the envelope signals, but it is sensitive to the DC offset of the envelope. Hence, the DC-notch filter should make the output sound better.

4.3 Testing

The objective of testing is twofold: verify that the system works, and find conditions that "break" the system. You should perform at least two types of tests with your working system. First, vary the input signal by using male and female speakers. Describe any differences that you hear by finding cases where the system works well and other cases where you can observe serious degradations. Show a few spectrograms to help explain how the system works. Spectrograms would also be useful to compare the final output to the original—a fair comparison would be to compare to the pre-emphasized original.

Second, testing with music input would also be interesting since a patient with a cochlear implant would have the device optimized for speech, but would be hearing all sorts of other sounds. One issue would be the bandwidth coverage of the filter bank with respect to the expected range of frequencies in music. Spectrograms would help in comparing the output to the original input. In your tests you should listen primarily for intelligibility rather than high quality.

The ZIP file `CochlearTestSignals.zip` contains three signals for testing:

1. `greasyWW22k.wav`	Female speaker
2. `catsdogs22k.wav`	Male speaker
3. `furElise22k.wav`	Piano music

4.3.1 Debugging

Here is one hint on debugging: use a single sinusoid as the input signal rather than a speech signal. Also, vary the frequency of the sinusoidal test input. When the frequency of the test sinusoid equals the center frequency of one of the channels, it seems that you should get (almost) perfect reconstruction at the output. In any event, debugging is simplified because you can analyze how the sinusoid goes through the system, i.e., there is only one spectral line to track.

4.3.2 Demonstration

Even if you are writing a lab report, a short demonstration of the working system can be performed. The purpose of the demo would be to show how your system handles music, as well as speech. In a lab report write-up, you should be able to explain the design decisions that were made when creating the bandpass filters for the filter bank.

Lab: Cochlear Implant Simulation with an FIR Filter Bank
INSTRUCTOR VERIFICATION PAGE

For each verification, be prepared to explain your answer and respond to other related questions that the lab TA's or professors might ask. Turn this page in at the end of your lab period.

Name: ______________________________ Date of Lab: __________

Part 3.1 Demonstrate the pre-emphasis filter by showing spectrograms of a speech signal before and after filtering. Compare the spectrograms and point out changes after filtering.

Verified:____________________ Date/Time:__________

Part 3.2(c) Show the magnitude of the AM signal vs. time. Point out the feature on the plot that would be called the envelope. Explain the frequency content of $b(t)$ and $b_1(t)$ versus that of $|b(t)|$.

Verified:____________________ Date/Time:__________

Part 3.2(d) Explain how you designed the lowpass filter with the **`filterdesign`** GUI, or with **`designBPF`**. To show how you used the filter coefficients, write the call to the MATLAB function `filter` below.

```
yy = filter(
```

Verified:____________________ Date/Time:__________

Part 3.2(e) Plot the output signal from the LPF, and compare to the known envelope. Explain differences.

Verified:____________________ Date/Time:__________

Part 3.3 *Optional:* Write the MATLAB filter design function, **`designBPF.m`**.

Verified:____________________ Date/Time:__________