

EMasters School 2003, Barcelona
Auditory Coding of Vowels in Noise

Sarah Simpson

1 Introduction

How is speech coded in the auditory nerve? In the last 30 years two distinct theories have emerged. The **rate-place** theory claims that the average firing rate in the auditory nerve fibres is measured as a function of frequency, thereby creating an auditory spectrum. By contrast, the **timing** theory suggests that the intervals between successive spikes encode the frequency to which that fibre is responding, and that the ensemble of such dominant frequencies allows the spectrum to be recovered.

In this project, you will first explore processing by the auditory periphery using interactive demonstrations and simple computational models. Once you have familiarised yourself with these models, you'll implement processing schemes which will enable you to investigate the rate-place and timing theories of auditory coding. This project will involve some reading, the development of a small amount of MATLAB code, and some data analysis.

- Handel, S (1989) Listening, MIT Press, Chapter 12.
- Seneff, S (1990) A joint synchrony/mean-rate model of auditory speech processing, J. Phonetics, 16, 55-76.
- Gitzha, O (1990) Temporal non-place information in the auditory nerve firing patterns as a front-end for speech recognition in a noisy environment, J. Phonetics, 16, 109-124.

2 The project

2.1 Modelling the Basilar Membrane

Reading: Handel, pages 461-481.

In this stage you are going to investigate the shape of the gammatone filter, which is commonly used to simulate peripheral auditory filtering. In MATLAB, you can find out how to use the gammatone filter by typing `help gammatone`. The filter is linear, so it is completely characterised by its impulse response. Recall that the impulse response and frequency response of a linear system are related by the Fourier transform, so we can examine the filter shape in the frequency domain as follows:

1. Generate an impulse signal (ie the first sample is unity and the remaining samples are zero). We suggest your signal has a length of 2048 samples, eg `x=zeros(1,2048); x(1)=1;`
2. Process the impulse through a gammatone filter to obtain the impulse response (you should also plot the time-domain response to see what it looks like). We suggest you use a sampling frequency of 20 kHz and a centre frequency between 20 Hz and 8 kHz.
3. Plot the log magnitude of the Fourier transform of the impulse response.
4. Repeat steps (2) and (3) for a range of different centre frequencies. Use the command `hold on` so that all the curves appear on the same axis.
5. Compare the filter shapes you have derived with the auditory filter shapes shown in figure 1. How does the shape differ? How does the shape of the gammatone filter differ with centre frequency? Is this what you would expect?
6. Would you expect the gammatone to show the same level-dependent changes in filter shape and the auditory filters shown figure 1? How could you test this?

Run the interactive demonstration of the basilar membrane to explore the effects of different signals on the motion of the membrane, by typing `bm` at the MATLAB prompt. The demonstration uses a bank of gammatone filters to simulate the travelling wave behaviour of the basilar membrane. Answer the questions in the accompanying tutorial material, by selecting 'about `bm`...' from the 'Info' menu.

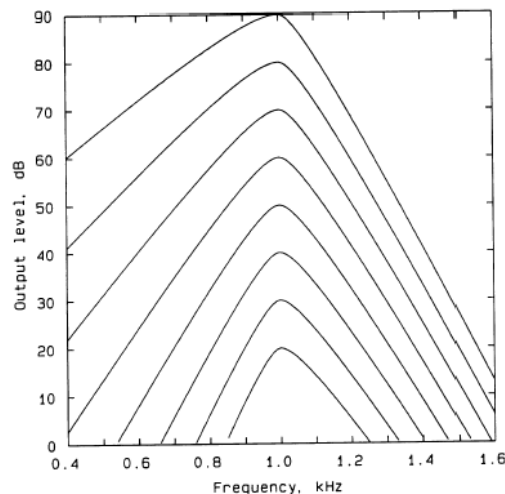


Figure 1: The shape of the auditory filter, measured psychophysically. Note how the filter become less sharply tuned with increasing sound levels. Reproduced from Moore (1997)

2.2 Modelling auditory nerve responses

Reading: Handel, pages 481-494.

In this task you will investigate the firing patterns in auditory nerve fibres by combining models of peripheral auditory filtering (as used in the previous task) and inner hair cell response.

2.2.1 Modelling adaption and saturation

Your first task is to plot a graph that relates the sound level of a tonal stimulus (in dB) to the average firing rate of an auditory nerve fibre which is responding to the stimulus. This is called a rate-intensity curve (see figure 2). You will again use the gammatone model of an auditory filter. The output of the gammatone filter is passed through the Meddis (1986) model of an inner hair cell to yield a representation of firing activity in the auditory nerve.

Important note: the Meddis hair cell model assumes that an input signal level of 1 corresponds to 30 dB. In the following, you will need to experiment with various tone levels to ensure that your inputs to meddis are in the appropriate range. Meddis responds at its spontaneous rate for signals below about 30 dB, and saturates at around 80-90 dB.

1. Create a tone stimulus using the `tone` function. We suggest that you use a tone duration of 200 ms, a frequency of 1 kHz and a sampling frequency of 10 kHz. Start with a level of 10 dB.
2. Process the tone stimulus through a gammatone filter using the same sampling frequency, and having the same centre frequency as the frequency of the tone (1 kHz in our example).
3. Process the output of the gammatone filter through the Meddis model (type `help meddis` to find out more). Then, compute the average firing rate by taking the mean result (MATLAB has a function `mean`). Also, try plotting the output of the Meddis model; this should resemble the plots of firing rate against time in figure 12.13 on page 490 of Handel. How might you show the spontaneous rate of the fibre?
4. Plot the average firing rate you computed in step 3 against the level of the tone (10 dB in our example).
5. Now repeat steps 1 through 4 for tone levels of 20, 30, . . . , 100 dB. It would be easiest to do this using a `for` loop. In the end, you should have a curve that looks like the one of the traces in figure 2. If you don't, please re-read the important note on signal levels at the head of this section.
6. What is the threshold of this stimulated auditory nerve fibre? Does the model show the same saturation property as real auditory nerve fibres?

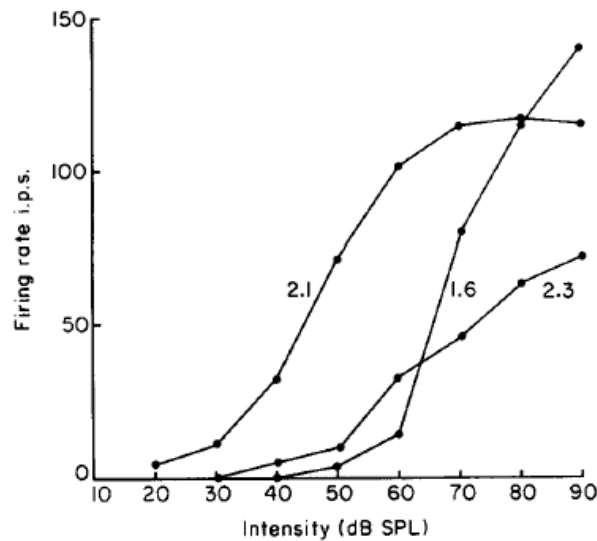


Figure 2: Rate-intensity curves for three groups of auditory nerve fibres. The firing rate as a function of stimulus intensity shows a sigmoidal shape. Note the dynamic range of most fibres is small (approx 30 dB) while the range of normal hearing subjects is much wider (approx 140 dB). From Pickles (1988)

2.2.2 Modelling period histograms and tuning curves

Figure 12.12(b), page 488 of Handel shows period histograms for an individual fibre. In this task you will produce period histograms for a number of different tonal stimuli.

1. Create a tone as before.
2. Process the tone through a gammatone filter with a centre frequency equal to the frequency of the tone. Then process the output of the tone through the Meddis hair cell model.
3. Compute the sampling period (in number of samples) required, as a function of the filter centre frequency and the sampling frequency you are using.
4. Next compute the mean rate at each sampling period. You might find this is best done using a `for` loop.
5. Use the `bar` function to plot a bar chart of the means of the sample periods.

A tuning curve shows how an individual fibre responds to stimulation at different frequencies and different intensities (see figure 3).

To produce a tuning curve, you need to compute the mean firing rate for a range of frequencies and intensities. Typically, you would probe the fibre with a short tone, record the mean firing rate, then produce another tone with a different intensity or frequency. By varying the frequency and intensity over a specified range you can build-up a tuning curve similar to the ones in figure 12.9. You will find it useful to use two `for` loops to vary the tone level and frequency. If you store the results in a matrix, use the supplied function `spect` to display the result.

2.3 A rate-place model

Using the models explored in the previous stages, you will construct a scheme for producing rate-place representations of synthetic speech sounds.

The program `template.m` processes an input signal with a bank of gammatone filters and an array of Meddis hair cell models. Modify the program to produce an excitation pattern which is made from the mean firing rate in each channel. For simplicity, use a time window of only the final 25 ms of the signal for your analysis. Plot the output for the different vowel signals in the EMasters directory.

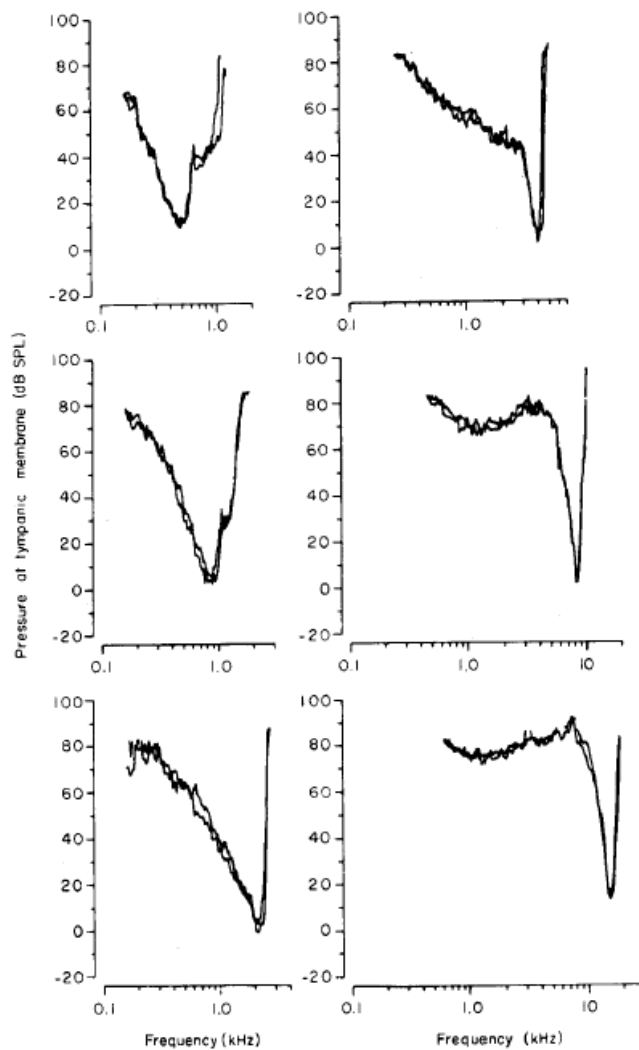


Figure 3: Auditory nerve fibre tuning curves. When stimulated by pure tones, auditory nerve fibres show a maximal sensitivity to a particular frequency (the so-called characteristic frequency) and a fairly rapid fall off in sensitivity to other frequencies. Hence, the neuron acts like a bandpass filter. The tuning curves shown here are for fibres with a variety of characteristic frequencies, reproduced from Liberman and Kiang (1978).

What happens when you increase the amplitude of the vowels? Try plotting the rate-place representation of a vowel at a range of different amplitudes.

When using a bank of gammatone filters, it is possible to vary the number of filters used. What happens to the representation of the vowel as you vary the number of filters?

2.4 A temporal model: The generalised synchrony detector

Reading: Seneff (1988).

Proposed by Stephanie Seneff in the 1980's, the generalised synchrony detector (GSD) measures the degree to which a model auditory filter is synchronised (phase-locked) to a stimulus frequency near to its centre frequency. If the centre frequency of an auditory filter is close to a spectral dominance (such as a formant peak), the response of the GSD for that channel will be large; otherwise the response will be small. Plotting the GSD for each centre frequency gives a 'pseudo-spectrum', an estimate of the spectrum derived from timing information.

2.4.1 Implementing the GSD

For *each* auditory filter channel, compute:

$$\text{GSD}(f) = \arctan \frac{\{\|u + v\|\}}{\{\|u - v\|\}} \quad (1)$$

Where:

- $\{\dots\}$ indicates a time average. For these tasks, using vowels, you should average the quantity inside the curly brackets over a time window of about 25 ms (use the last 25 ms of the vowel). Experiment with different window lengths. A rectangular window is acceptable, but a shaped window is better — try a Hamming window (use the `myHamming` function).
- $\|\dots\|$ indicates magnitude (full wave rectification). Take the absolute value (using the function `abs`).
- f is the centre frequency (in Hz) of the auditory filter channel (the centre frequencies are returned by the `MakeErbCFs` function).
- u is the auditory nerve activity in the channel (i.e. the output of the `meddis` function, which in turn gets its input from the `gammatone` function).
- v is the auditory nerve activity delayed by time τ , where $\tau = 1/f$ (i.e. the reciprocal of centre frequency of the filter).

Finally, plot $\text{GSD}(f)$ for each channel against channel number. This gives a 'pseudo-spectrum' which can be compared with spectra estimated from the average firing rate information. As before, use the final 25 ms segment of the signal for your analysis.

2.4.2 How does it work?

The inputs to the GSD are the auditory nerve activity, u , and a delayed version of this activity, v . When the input is nearly periodic with the delay period, τ , the value of the denominator is close to zero, and hence the output of the GSD is large. Note that in the case of perfect synchrony, the denominator will be zero; a saturating nonlinearity (\arctan) is needed to contain the value of the GSD output.

2.4.3 Other things to try

The basic GSD described above can be modified in several ways. For example, we would prefer the GSD not to respond to very low amplitude events, and this can be fixed by subtracting a constant from the numerator, e.g. compute:

$$\text{GSD}(f) = \arctan \frac{\{\|u + v\|\} - \delta}{\{\|u - v\|\}} \quad (2)$$

The constant δ , should have a value slightly above the spontaneous rate of the auditory nerve channel. Earlier, you produced a plot which showed the spontaneous rate of the auditory nerve fibres in the Meddis simulation. You could try different kinds of compression too. An alternative to the \arctan compression is to add a constant, C , to the denominator:

$$\text{GSD}(f) = \arctan \frac{\{\|u + v\|\}}{\{\|u - v\|\} + C} \quad (3)$$

Try $C = 1$ to start with. What happens when you increase the value of C ?

The GSD is inaccurate at centre frequencies near to the Nyquist rate, since the time separation between u and v cannot be approximated well by a whole number of samples. How might you fix this?

2.5 A temporal model: The ensemble interval histogram

Reading: Ghitza (1988).

Proposed by Oded Ghitza in 1988, the Ensemble Interval Histogram (EIH) is a frequency-domain representation which gives fine low-frequency detail and a greater degree of robustness than conventional spectral representations. The representation is formed from the ensemble histogram of inter-spike intervals in an array of auditory nerve fibres. Rather than using a model of hair cell discharge, the output of each auditory filter is passed through a multi-level crossing detector.

2.5.1 Implementation

The modelling of hair cell transduction is achieved by implementing a multi-level detector in each auditory filter channel. You should modify the program `template.m` to produce your EIH representations. The output of the gammatone filter should first be half wave rectified, then processed by a bank of level detectors.

The level detectors operate in each channel, and detect 7 levels. In each channel, when the level of the input crosses the threshold of a detector, the detector should produce a spike. The threshold levels of the detectors should be equally spaced on a log scale. The function `find` might be useful for implementing these level detectors. You should try to determine the threshold levels yourself. Why not try testing your level detectors for a tone?

The spike output from the level detectors is converted into an inverse-interval histogram (the function `hist` might be useful). This is simply a count of the intervals between the spikes. These histograms can then be summed with the output of the other level detectors in the channel. Finally, the output of each channel is summed to produce the ensemble interval histogram.

2.6 Representations of vowels in noise

Reading: Handel, pages 499-521.

You have now have implemented three different schemes for representing sounds processed by a model of the auditory periphery. In this final exercise, you will explore the limitations of these schemes for producing robust representations of vowels mixed with noise.

Previously, you will have used your models to produce spectra of the synthetic vowels provided in the EMasters directory. To produce representations of vowels in noise, you will need to add the waveform of the vowel to a noise waveform. In these exercises, we will use white noise, as this is the most straightforward to synthesise (use the MATLAB function `rand` to generate the random numbers for the waveform). When synthesising the noise it is necessary to produce a waveform of the same length as the vowel waveform you wish to add it too. Determine the length of the vowel waveform using the function `length`.

When describing a speech signal which has been added to noise, it is usual to describe the mixture in terms of its signal-to-noise ratio (SNR). If the vowel waveform is s , and the noise waveform is n , and both waveforms are of length N samples, then the SNR in decibels (dB) is given by the formula:

$$\text{SNR} = 10 \log_{10} \left(\frac{\sum_{i=1}^N s^2(i)}{\sum_{i=1}^N n^2(i)} \right) \quad (4)$$

Using this equation, you should write a function which will allow you to add the two waveforms together at a specified SNR. Finally, produce plots of the different vowels with added noise at a range of SNRs for the three processing schemes you have implemented.