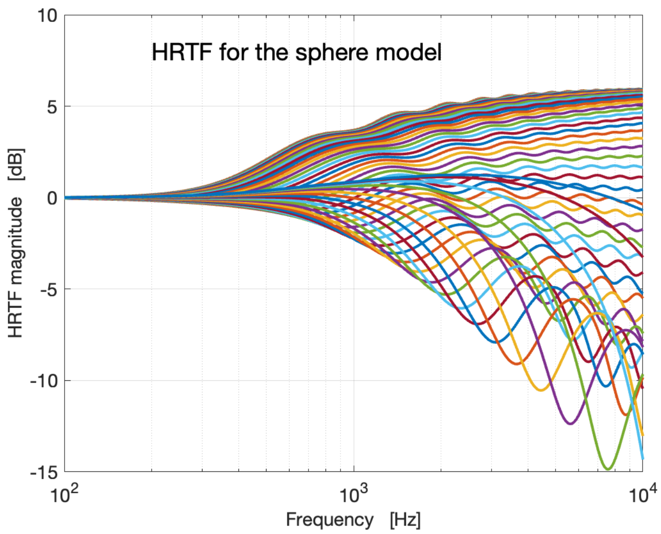
Assignment 2 - Directional hearing

Peter Svensson

September 25, 2025



All tasks are mandatory and should be part of your report. The deadline is specified in blackboard.

Introduction

In this assignment, you should generate some sound examples using the technique called *binaural synthesis*. This technique exploits the properties of humans directional hearing. Central to the binaural synthesis is the concept of *Head-Related Transfer Functions (HRTFs)*. The principles for binaural synthesis are quite simple and elegant:

- Our hearing has two input channels (we ignore vibrations in our bodies for now): the two sound pressure signals at the two eardrums. Therefore, if we *record* the two sound pressure signals at the eardrums, and reproduce them at a later instant, we should hear the same thing as at the original recording event. This is the *binaural recording technique.*

- As a consequence of the bullet point above, if we can *synthesize* the two eardrum sound pressure signals, for a real or virtual situation, then the listening person should get the same listening experience as in the simulated situation. This is the *binaural synthesis technique.*
- So, what is the sound pressure at the eardrum? That is described by the *Head-Related Transfer Functions (HRTFs)*, which is the TF of a filter, the output of which is the sound pressure at the eardrum, and the input signal of which is the sound pressure in an incoming sound wave (before it hits a listener's head). There will be one HRTF for each incidence angle, and one for each ear[1].

After the initial euphoria over the remarkably simple principle to generate 3D sound, one realizes that there are some challenges to solve:

1. The HRTFs will depend on the shape and size of a person's head, shoulders and outer ear properties. Thus the HRTFs are individual.
2. The HRTFs will be needed for a large number of incidence angles, since we can detect incidence angle differences down to 1 degree (in the most sensitive direction).
3. If we use headphones to generate the sound at the eardrums, we usually need to equalize the headphone's frequency response, to make sure that the sound pressure which arrives to the eardrum is as intended.
4. If we use headphones to generate the sound at the eardrums, we might need to use head-tracking and dynamically update the HRTFs used, to take into account that a listener deliberately and involuntarily moves the head more or less.
5. The ultimate input signal is the sound pressure at the eardrum of the listener, but it is very impractical to have to record at the eardrum (for capturing the sound, and to measure/check that the reproduced sound is as intended). A more common reference sound pressure is the sound pressure at the entrance to a blocked ear canal. With this choice, it turns out that the individual differences between HRTFs will be smaller than if the sound pressure at the eardrum is recorded.

HRTF alternatives

It is possible to choose more or less realistic HRTFs, which correspond to more or less complicated processing (for acquiring the HRTFs and/or for doing the filtering). Four levels of realism are:

[1] If the sound source is closer than approx. 1 m, the HRTFs will also be distance-dependent.

1. The most accurate way is to use *individual HRTFs* for each listening person. It is, however, complicated to measure the HRTFs in an accurate way.

2. A much simpler approach is to represent the listening person's head with a *rigid sphere in free air*, that is, without any torso or ear pinna (outer ears). The sound pressure at two points on the sphere can then represent the sound pressure which enters the ear canal. Importantly, these two sound pressures would have realistic magnitudes and phases (the phase of the two signals represents the interaural time difference). However, the mathematical expressions for the sound pressure at a rigid sphere are somewhat complicated.

3. An even simpler approach than the rigid-sphere model is to use a *first-order IIR filter* for the magnitude of the (simplified) HRTFs, *combined with FIR-filters* to generate the correct interaural time delay. This idea was presented by C. P. Brown and R. O. Duda as described below.

4. The simplest approach is to only use interaural time differences (ITD) as HRTFs. This gives a completely unrealistic spectrum, and no interaural level differences, but the important ITD will be correct.

Brown and Duda's simplified HRTFs

A first-order filter that modifies the magnitude for the "sunny-side ear" as well as the "shadow-side ear" was suggested by C. P. Brown and R. O. Duda in "An efficient HRTF model for 3D-sound," in Proc. of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), 1997. They suggested a filter response which can act as anything between a *shelving filter* which boosts high frequencies, to a low-pass filter:

$$H_{L/R}(\omega) = \frac{\alpha_{L/R} \cdot j\omega + \beta}{j\omega + \beta} \tag{1}$$

where

$$\beta = \frac{2c}{a},$$

$a =$ the radius of a sphere that represents the head, $c =$ the speed of sound, and

$$\alpha_L = 1 - \sin\theta, \qquad \alpha_R = 1 + \sin\theta, \tag{2}$$

where θ is the incidence angle *relative to the frontal direction*.

Thus, if the sound is coming from straight ahead,

$$\theta = 0 \Rightarrow \begin{cases} \alpha_L = 1 \\ \alpha_R = 1 \end{cases}$$

If the sound is coming from the right side,

$$\theta = \frac{\pi}{2} = 90^\circ \Rightarrow \begin{cases} \alpha_L = 0 \\ \alpha_R = 2 \end{cases}$$

and if the sound is coming from the left side,

$$\theta = -\frac{\pi}{2} = -90^\circ \Rightarrow \begin{cases} \alpha_L = 2 \\ \alpha_R = 0 \end{cases}$$

etc for other angles $\theta \in [-90^\circ, 90^\circ]$.

How can we construct a digital filter from the frequency response expression above? The bilinear transform converts a Laplace transform, $H(s)$, to a z-transform, $H(z)$, via:

$$s = \frac{2}{T} \cdot \frac{(z-1)}{(z+1)} \tag{3}$$

where $T = 1/f_S$. Our given frequency response on the $j\omega$-form can be converted to the Laplace transform via $s = j\omega$:

$$H_{L/R}(s) = \frac{\alpha_{L/R} \cdot s + \beta}{s + \beta}$$

Using the bilinear transform above, you can get $H_{L/R}(z)$, and if you write it on this form,

$$H_{L/R}(z) = \frac{B_0 + B_1 z^{-1} + B_2 z^{-2} + ...}{1 + A_1 z^{-1} + A_2 z^{-2} + ...} \tag{4}$$

you have IIR-filter coefficients directly!

The interaural time difference can be handled separately with an FIR filter with a delayed pulse: all frequencies will be delayed with the same amount. The delay can be found using the simple formula called Woodworth's formula:

$$\Delta t = \frac{a}{c}(\theta + \sin\theta) \tag{5}$$

where θ is the angle relative to straight ahead, that is, the same as in Eqs. (2). Note that you get positive and negative delays for positive and negative θ with Eq. (5) - make sure that the shadow side of the head gets the sound later than the sunny side!

Tasks

In all your tasks, the sampling frequency should preferably be 44100 Hz. For the speed of sound you could choose 343 m/s, and for the head radius you could choose $a = 9$ cm.

Tasks

1. **The interaural time delay (ITD)**

   The simplest possible version of Head-Related Impulse Response introduces just an interaural time delay, with no modification of the magnitude. This is clearly not a great model (number 4 in the list of HRTF alternatives above) but it implements the dominating factor - the ITD.

   In this exercise you should implement a Matlab function (or in Python or ...) which gives two impulse responses, one for the left ear and one for the right ear, with an interaural time delay according to Woodworth's formula, for a given incidence angle. You need to consider only sources in the horizontal plane. Also, you can let the ear which is "on the sunny side" always get a pulse of amplitude 1 in the first sample (which is sample no. 1 in Matlab).

   The syntax of the function could be like this Matlab example (with some help text to clarify):

```
function [irl,irr] = hrir1(incidenceangle,headradius,fs,cair)
% Input parameters:
%   incidenceangle The incidence angle, in degrees, in the horizontal plane.
%                  0 deg.  is straight ahead and the angle increases
%                  in the clockwise direction when the head is viewed from above.
%   headradius     The head radius, in m
%   fs             The sampling frequency, in Hz
%   cair           The speed of sound, in m/s
%
% Output parameters:
%   irl, irr             The HRIR of the left and right ear, respectively
```

   Hint: you can give the HRIRs a length which is given by the longest possible interaural delay, which happens for a 90 degree incidence.

   Option: A more precise timing definition would be needed if you want to use this function to add several reflections. In principle, the HRIR should give a delay relative to the center position of the head. *This does, however, lead to that the IR for the sunny side ear arrives before the time zero, which can not be implemented. Negative arrival times can be implemented, if one adds a constant number of samples at the beginning of all IRs, a pre-zeropadding!*
   Thus, an optional task could be to implement the correct delay, for all incidence angles, relative to the center of the head, by introducing a constant number of pre-padding zeros.

2. **The interaural time delay + simple filter**

In this task, you should calculate the frequency response of the shelving filter according to Brown and Duda, as described above in Eq. (1).

You should then plot the resulting frequency responses for a few incidence angles, and plot for the sunny side and the shadow side, for each incidence angle.

The syntax of the function, and the help text, could be this:

```
function [tfl,tfr,fvec] = hrtf1(incidenceangle,headradius,fs,cair,nfft)
% Input parameters:
%   incidenceangle The incidence angle, in degrees, in the horizontal plane.
%                  0 deg.  is straight ahead and the angle increases
%                  in the clockwise direction when the head is viewed from above.
%   headradius     The head radius, in m
%   fs             The sampling frequency, in Hz
%   cair           The speed of sound, in m/s
%   nfft           The fft size
%
% Output parameters:
%   tfl, tfr       The HRTF of the left and right ear, respectively
%   fvec           The frequency vector, of length nfft/2+1, starting with zero and
%                  ending with fs/2
```

Hint: you can create a frequency vector via this command (inside the function):

```
fvec = fs/nfft*[0:nfft/2-1].';
```

Then you can plot the frequency responses like this:

```
semilogx(fvec,20*log10(abs([tfl tfr]))) grid
```

3. The simple filter implemented as a digital IIR filter

In the previous task, you implemented a function which gives the *frequency response* of a shelving filter - but now you should implement the first-order IIR filter according to Brown and Duda.
Derive the expression for the IIR filter coefficients on the form in Eq. (4), and write a Matlab function which gives you the IIR filter coefficients for the left and right ear according to Eq. (4).

Hint: the IIR filter should just require three coefficients: B_0, B_1 *and* A_1.

The syntax of the function, and the help text, could be this:

```
function [BL,AL,BR,AR] = hrtfiir(incidenceangle,headradius,fs,cair)
% Input parameters:
%   incidenceangle The incidence angle, in degrees, in the horizontal plane.
```

```
%                  0 deg.  is straight ahead and the angle increases
%                  in the clockwise direction when the head is viewed from above.
%   headradius     The head radius, in m
%   fs             The sampling frequency, in Hz
%   cair           The speed of sound, in m/s
%
% Output parameters:
%   BL, AL       The IIR filter coefficients of the left ear.  AL(1) is always 1.
%   BR, AR       The IIR filter coefficients of the right ear.  AR(1) is always 1.
```

4. The simple IIR filter combined with the ITD

The IIR filter in task 3 gives the right frequency response magnitude, but not the ITD. You could combine the result in task 3 with the result in task 1 to generate a complete HRIR simulator!

Write a Matlab function which combines the IRs in task 1 and task 3.

5. The final sound demonstration

Generate a sound example where you take a sound signal (music, noise, or whatever you prefer) and simulate short bursts that arrive from different incidence angles. It is quite effectful to let short noise bursts move around your head in steps of e.g., 30 degrees.

Hint: it is preferable to use pink noise bursts rather than white noise bursts, because pink noise bursts are perceived as having a more even spectrum than white noise (which is perceived as sharp, or high-frequency dominated).

Hint 2: Beware of the filter *function in Matlab. If you use this function for filtering with an FIR filter, please note that* the output signal is the same length as the input signal. *A convolution with an FIR filter should give an output signal which is longer than the input signal, and the effective truncation caused by the* filter *function might or might not have an unwanted effect for you. It is, however, very easy to avoid any potential problems by making sure that all input signals that you will filter with will have a number of zeros at the end which is larger than the length of the FIR filters.*