**Documentation for the UCD HRIR Files***

CIPIC Interface Laboratory
University of California at Davis
Davis, CA 95616

October 7, 1998

## 1. Introduction

This document describes the format for our MATLAB™ data files of compensated head-related impulse response (HRIR) measurements. The measurements were made at the CIPIC Interface Laboratory at the University of California at Davis. At the moment, the files have not been released for general use. Thus, please do not redistribute the data or this documentation further without approval by Prof. V. R. Algazi (algazi@ece.ucdavis.edu).

## 2. Coordinates

A head-related impulse response `h` is a function of azimuth, elevation and time. In the data files, `h` is sampled in space and time. The values of azimuth, elevation and time are specified by discrete indices, `naz`, `nel` and `nt`, and `h(naz, nel, nt)` is a $25 \times 50 \times 200$ 3-dimensional array. Thus, HRIR values are given for 25 different azimuths, 50 different elevations, and 200 instants in time.

The azimuth angle $\theta$ and the elevation angle $\phi$ are measured in a head-centered interaural-polar coordinate system (see Fig. 1). The azimuth is the angle between a vector to the sound source and the midsaggital or vertical median plane, and varies from $-90°$ to $+90°$. The elevation is the angle from the horizontal plane to the projection of the source into the midsaggital plane, and varies from $-90°$ to $+270°$. With these coordinates,

$(0°, 0°)$ corresponds to a point directly ahead
$(0°, 90°)$ corresponds to a point directly overhead
$(0°, 180°)$ corresponds to a point directly behind
$(0°, 270°)$ corresponds to a point directly below
$(90°, 0°)$ corresponds to a point directly to the right
$(-90°, 0°)$ corresponds to a point directly to the left

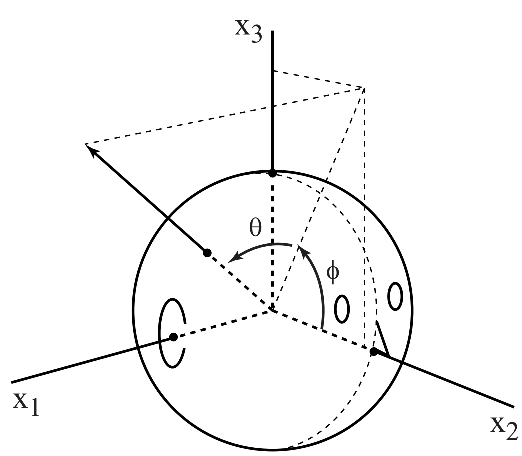


Fig. 1. The interaural-polar coordinate system.

The azimuth values are not uniformly sampled, but are more densely spaced near the midsaggital plane. The azimuth index `naz` is related to the azimuth angle $\theta$ as follows:

* This work was supported by the National Science Foundation under Grant IRI-9619339. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors, and do not necessarily reflect the views of the National Science Foundation.

| naz | $\theta$ | naz | $\theta$ | naz | $\theta$ | naz | $\theta$ | naz | $\theta$ |
|---|---|---|---|---|---|---|---|---|---|
| 1 | $-80°$ | 6 | $-35°$ | 11 | $-10°$ | 16 | $15°$ | 21 | $40°$ |
| 2 | $-65°$ | 7 | $-30°$ | 12 | $-5°$ | 17 | $20°$ | 22 | $45°$ |
| 3 | $-55°$ | 8 | $-25°$ | 13 | $0°$ | 18 | $25°$ | 23 | $55°$ |
| 4 | $-45°$ | 9 | $-20°$ | 14 | $5°$ | 19 | $30°$ | 24 | $65°$ |
| 5 | $-40°$ | 10 | $-15°$ | 15 | $10°$ | 20 | $35°$ | 25 | $80°$ |

In MATLAB, the azimuth angle corresponding to `naz` is the `naz`-th element of the vector

```
azimuths = [-80 -65 -55 -45:5:45 55 65 80] .
```

Both elevation and time are uniformly sampled. Elevations range from $-45°$ to $+230.625°$ in steps of $5.625°$. This angular increment divides the full circle into 64 equal parts, but only 50 values are used in our measurements. In MATLAB, the elevation angle corresponding to `nel` is the `nel`-th element of the vector

```
elevations = -45 + 5.625*(0:49) .
```

The temporal sampling frequency is $f_s = 44.1$ kHz.

## 3. Measurement and Compensation

We use a modified version of the Snapshot™ system by Crystal River Engineering (now part of Aureal Semiconductor) to measure the HRIR's. The subject is seated in the center of a 1-m radius hoop whose axis is aligned with the subject's interaural axis. The subject's ear canals are blocked, and Etymotic Research ER-7C probe microphones are used to pick up the sound that arrives at the entrance to the ear canal. (In the literature, this is called a blocked-meatus measurement.) The Snapshot system generates pseudo-random signals that are used sequentially to drive five 6.4-cm diameter Bose Acoustimass™ loudspeakers fastened to the hoop. The resulting sounds are picked up by the microphones, digitized, and processed to yield a raw HRIR. The records are limited to about 4 ms to delete most room reflections, although some floor and knee reflections can be seen in the data at low elevations, and some ceiling reflections can be seen in the data at high elevations.

The raw HRIR measurements must be compensated to correct for limitations of the loudspeakers and the microphones employed. The basic correction is made by removing the subject, placing the microphones at the center of the hoop, and measuring the free-field response. The free-field-corrected response is the inverse Fourier transform of the ratio of the Fourier transform of the raw HRIR to the Fourier transform of the free-field response. However, this procedure breaks down at low frequencies, partly because of the low signal-to-noise ratio and partly because of the limited duration of the impulse response, and an additional correction is applied to obtain the proper low-frequency response.*

* Some subtle issues arise in low-frequency compensation of measured data. In theory, the zero-frequency or DC response should be close to unity (0 dB), since the head does not significantly diffract the sound waves when the wavelength is much longer than the head circumference. In the time domain, this is equivalent to saying that the integral of the head-related impulse response should be unity. In fact, the low-frequency HRTF is not exactly 0 dB, because (except for the median plane) the source is always somewhat closer to one ear than the other. Thus, the low-frequency gain is a function of azimuth. In addition, because A/D conversion scales the impulse response value to 16-bit integers, an arbitrary scale factor is applied to any set of HRIR's, which offsets the responses by a constant number of dB. However, because the amplifiers are AC coupled and the loudspeaker output vanishes at low frequencies, the DC gain for experimental data should theoretically be zero, which is $-\infty$ dB. In fact, noise in the measurements will result in some meaningless measured DC-gain. Since the free-field response is also zero at DC, free-field compensation actually exacerbates the problem, and the DC gain in the compensated data has no physical meaning. The low-frequency gain is meaningful, however, and the data are compensated so that the amplitude response is well behaved below about 400 Hz.

The resulting HRIR's described the sound pressure at the blocked entrance of the ear canal, and are missing the ear-canal resonance that would be present in a transfer function from the source to the ear drum. In addition, if a sound source is filtered by a blocked-meatus HRIR and heard through headphones, it is also important to compensate for the headphones. The compensation for headphones and the missing ear-canal resonance has been treated in detail by Møller (H. Møller, "Fundamentals of Binaural Technology," *Applied Acoustics*, **36**, pp. 171-218, 1992). Suffice it to say that these corrections, although important for reproducing sounds with a natural timber, are independent of direction, and merely add spectral coloration to the source.

## 4. Accessing the Data with MATLAB 5.x

In MATLAB 5.x, you can load the data by connecting to the directory containing the file and loading the file. That will create two arrays, `hrir_l` for the left-ear data and `hrir_r` for the right-ear data. If you want the left-ear HRIR for azimuth index `azn` and elevation index `eln`, evaluate `h = squeeze(hrir_l(azn,eln,:))`. Then, if you want to filter a signal `x` by `h`, evaluate `y = filter(h,1,x)`.

It is harder to retrieve the HRIR for a particular azimuth and elevation, and interpolation is required if the azimuth and/or elevation are not one of the sampled values. Interpolation is nontrivial. A simpler procedure is merely to find the HRIR for the nearest measured location. The following utility function called `getnearestUCDpulse.m` provides this retrieval ability.

```
function [pulse, azerr, elerr] = getNearestUCDpulse(azimuth,elevation,h3D);
% [pulse, azerr, elerr] = getNearestUCDpulse(azimuth,elevation,h3D);
%
% retrieves the impulse response from h3D that is closest to the
% specified azimuth and elevation (in degrees);

if nargin < 1,
   fprintf('Format: [pulse, azerr, elerr] = getNearestUCDpulse(azimuth,elevation,h3D)\n');
   return;
end;

azimuth = pvaldeg(azimuth);
if (azimuth < -90) | (azimuth > 90),
   error('Invalid azimuth');
end;
elevation = pvaldeg(elevation);

elmax = 50;
elindices = 1:elmax;
elevations = -45 + 5.625*(elindices-1);
el = round((elevation+45)/5.625 + 1);
el = max(el,1);
el = min(el,elmax);
elerr = pvaldeg(elevation - elevations(el));

azimuths = [-80 -65 -55 -45:5:45 55 65 80];
[azerr, az] = min(abs(pvaldeg(abs(azimuths - azimuth))));

pulse = squeeze(h3D(az,el,:));
```

Here the function pvaldeg finds a principal value of an angle, converting it to the range $[-90^\circ, 270^\circ)$:

```
function angle = pvaldeg(angle)
% angle = pvaldeg(angle)
%
% Maps angle (in degrees) into the range [-90, 270)

if nargin < 1,
   fprintf('Format: angle = pvaldeg(angle)\n');
   return;
end;

dtr = pi/180;
angle = atan2(sin(angle*dtr),cos(angle*dtr))/dtr;
if angle < -90,
   angle = angle + 360;
end;
```