Spatial Audio toolbox v.1.0

Ambisonics

Documentation

1. **Function AMorder(n).**

This function returns the order of Ambisonics which can be used with **n** number of speakers.

- **Example 1:**
  - *input*

```
n=41;
order=AMorder(n);
```

  - *output*

```
order = 6;
```

- **Example 2:**
  - *input*

```
n=16;
order=AMorder(n);
```

  - *output*

```
order = 3;
```

2. **Function AMspeakers(order).**

This function returns minimum speakers required for a given Ambisonics order.

- **Example 1:**
  - *input*

```
order=4;
num_speakers=AMspeakers(order);
```

  - *output*

```
num_speakers=25
```

- **Example 2:**
  - *input*

```
order=3;
num_speakers=AMspeakers(order);
```

  - *output*

```
num_speakers=16
```

3. **Function AMbisonicsCF(phi,theta,order)**

This function returns coefficients for each encoded channel. Using natural encoding so-called SN3D where each encoded channel has coefficient derived with formulas (1) and (2)

$$Y_{\ln}^{k}(\theta,\phi) = P_{\ln}(\sin\phi)\cos(n\theta) \quad \text{if} \quad k = 1 \qquad \textbf{(1)}$$

$$Y_{\ln}^{k}(\theta,\phi) = P_{\ln}(\sin\phi)\sin(n\theta) \quad \text{if} \quad k = -1 \qquad \textbf{(2)}$$

- *Input parameters:*

phi – sound source elevation angle
theta – sound source azimuth angle
order – ambisonics order

- **Example:**
  - *input*

```
phi=25;
theta=40;
order=3;
coef_sp_array=AMbisonicsCF(phi,theta,order);
```

  - *output*

```
coef_sp_array =

Columns 1 through 9

1.0000    0.4226    0.6943    0.5826   -0.2321    0.5082    0.4264    0.1235    0.7005

Columns 10 through 16

-0.4452   -0.0455   -0.0382    0.1167    0.6620   -0.2943    0.5097
```

4. **Function AMencodechannel(f_source,f,coef_sp_array ,num_channel,filename);**

This function creates encoded wave files from a given input mono file.

- *Input parameters:*

f_source – original wave file 1 channel mono
f – Sample rate
coef_sp_array – array with ambisonics coefficients for each channel
num_channel – number of encoded channels
filename – main file name. For each channel the number of the channel will be added at the end.

- **Example:**
  - *input*

```
phi=25;
theta=40;
order=3;
coef_sp_array=AMbisonicsCF(phi,theta,order);
num_channels=length(coef_sp_array);
[f_source, f] = wavread('C:\work\source\signal2.wav');
filename='C:\work\enc_';
AMencodechannel(f_source,f,coef_sp_array, num_channels,filename);
```

- *output*

The output files will be saved in 'C:\work\' with filenames:
enc_1.wav – for the first channel
enc_2.wav – for the second
…..

5. **Function AMsp_position(order,type,fix_angle,zero_e);**

This function generates speakers' position on a perfect sphere. It returns azimuth and elevation angle of each speaker in two different arrays.

- *Input parameters:*

order – Ambisonics order
type_fix – defines way of panning speakers.
If 1 – rotation of azimuth angle, elevation angle showed on figure 1
If 2 or other value – rotation of elevation angle, azimuth angle fixed showed on figure 2
Fix_angle – degrees of rotation, defining the distance between the rings
zero_e – Number of rotational rings if zero_e = 1 – 2 rings if zero_e = 0 – 3 rings. Both figure 1 and 2 are with zero_e = 0.

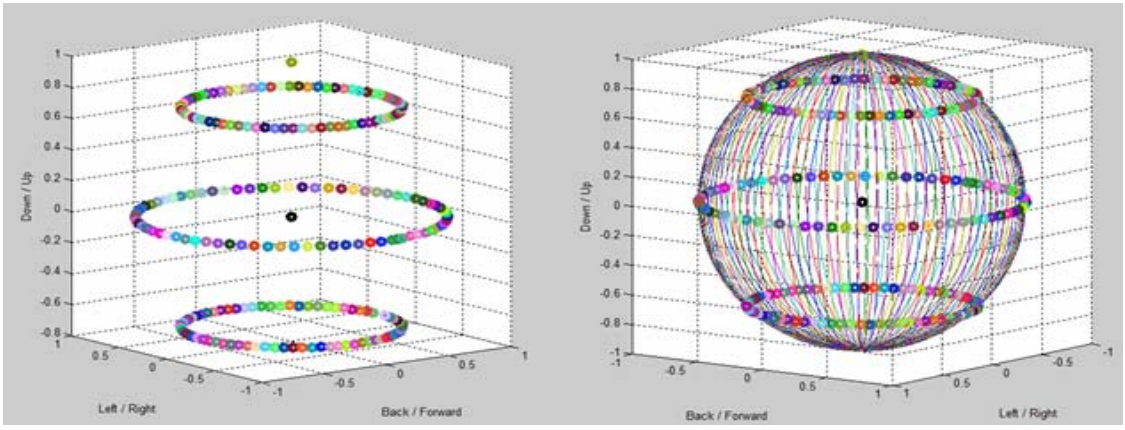

Figure 1

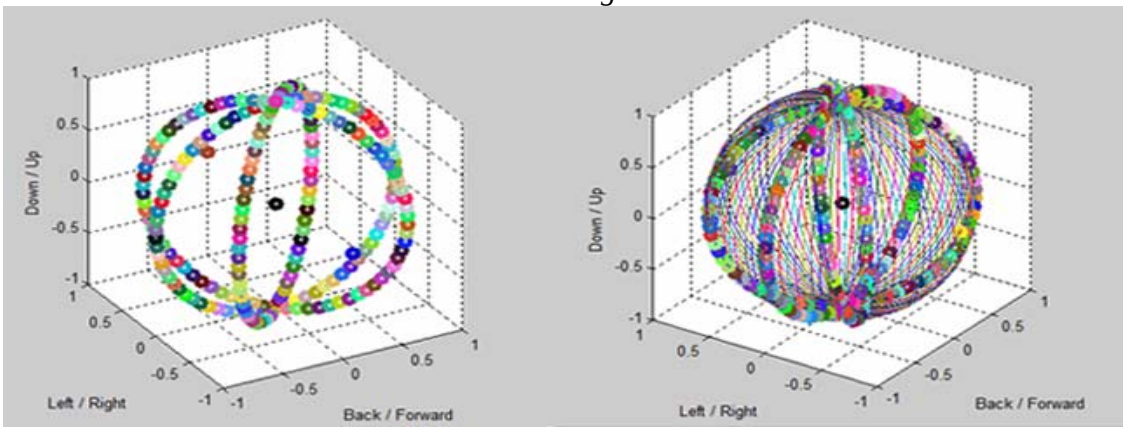

Figure 2

- **Example:**
  - *input*

```
[speaker_array_azimuth,speaker_array_elevation]=AMsp_position(3,2,5,0);
```

  - *output*

```
speaker_array_azimuth =0   90    0  275   45  225   45  225   45  225  135  315  135  315  135
315
speaker_array_elevation =90    0    0    0  -30  -30   30   30   90   90  -30  -30   30   30   90   90
```

6. **Function AMspeaker_channels(order,speaker_array_azimuth,speaker_array_elevation,filename,output_f);**

This function generates decoded files for each speaker, from given array of encoded files and given speaker's array.

- *Input parameters:*

order – Ambisonics order
speaker_array_azimuth – azimuth angle of each speaker
speaker_array_elevation – elevation angle of each speaker
filename – basic file name. For example if encoded files are:
enc_1 – encoded channel 1
enc_2 – encoded channel 2
enc_3 – encoded channel 3
filename must be: filename = "enc_".
output_f – basic file name for decoded channels. For example:
decsp_1.wav – decoded channel 1 (for speaker number 1)
decsp_2.wav – decoded channel 2 (for speaker number 2)
decsp_3.wav – decoded channel 3 (for speaker number 3)
output_f must be: output_f="decsp_".

- **Example**
  - *input*

```
filename='C:\work\enc_';
output_f='C:\work\decsp_';
order=3;
speaker_array_azimuth =[0   90    0  275   45  225   45  225   45  225  135  315  135  315  135
315];
speaker_array_elevation =[90    0    0    0  -30  -30   30   30   90   90  -30  -30   30   30   90
90];
AMspeaker_channels(order,speaker_array_azimuth,speaker_array_elevation,filename);
```

  - *output*

```
The output files will be saved in 'C:\work\' with filenames:
decsp_1.wav – for the first channel
decsp_2.wav – for the second
…..
```

7. **Function AMplot(speaker_array_azimuth,speaker_array_elevation,type_s);**

This function visualizes the position of each speaker. The virtual sphere can be plot also.

- *Input parameters:*

speaker_array_azimuth – azimuth angle of each speaker
speaker_array_elevation – elevation angle of each speaker
type_s – if 1 virtual sphere will be plotted, if 0 without virtual sphere.

8. **Function AMvectorgain(speaker_array_azimuth,speaker_array_elevation, filename,distance,azimuth,elevation);**

This function is for moving listener. It corrects the gains for each decoded channel according to the new listener's position and save the corrections in the files. It also returns the new gain values for each speaker.

- *Input parameters:*

speaker_array_azimuth – azimuth angle of each speaker
speaker_array_elevation – elevation angle of each speaker
filename – basic file name. For example if decoded files are:
decsp _1 – encoded channel 1
decsp _2 – encoded channel 2
decsp _3 – encoded channel 3
….
distance – the distance between center of the sphere and new position.
azimuth – the azimuth angle of the new position
elevation – the elevation angle of the new position

- **Example**
  - *input*

```
filename='C:\work\decsp_';
speaker_array_azimuth =[0   90    0  275   45  225   45  225   45  225  135  315  135  315  135
315];
speaker_array_elevation =[90    0    0    0  -30  -30   30   30   90   90  -30  -30   30   30   90
90];
distance=0.5l
azimuth=180;
elevation=0;
new_coef_sp_array=AMvectorgain(speaker_array_azimuth,speaker_array_elevation,filename,
distance,azimuth,elevation);
```

  - *output*

```
new_coef_sp_array =
    Columns 1 through 9

    0.7454    0.7454    1.0000    0.7709    0.9098    0.5323    0.9098    0.5323    0.7454

    Columns 10 through 16
```

0.7454 0.5323 0.9098 0.5323 0.9098 0.7454 0.7454

The output files will be saved in 'C:\work\' with filenames:
decsp_1.wav – for the first channel
decsp_2.wav – for the second
…

9. **Function AMvectordelay(speaker_array_azimuth,speaker_array_elevation,distance,azimuth,elevation,fs,SoundSpeed,filename);**

This function adds a delay to each decoded channel. It re-writes the files by adding 0 to the first n samples. n is the delay for each channel in samples. It also returns the delay for each channel in array.

- *Input parameters:*

speaker_array_azimuth – azimuth angle of each speaker
speaker_array_elevation – elevation angle of each speaker
filename – basic file name. For example if decoded files are:
decsp _1 – encoded channel 1
decsp _2 – encoded channel 2
decsp _3 – encoded channel 3
distance – the distance between center of the sphere and new position.
azimuth – the azimuth angle of the new position
elevation – the elevation angle of the new position
fs – sample rate
SoundSpeed – The speed of sound

- **Example**
  - *input*

```
filename='C:\work\decsp_';
distance=0.5l
azimuth=180;
elevation=0;
fs=44100;
SoundSpeed=340.25;
speaker_array_azimuth =[0   90    0  275   45  225   45  225   45  225  135  315  135  315  135
315];
speaker_array_elevation =[90    0    0    0  -30  -30   30   30   90   90  -30  -30   30   30   90
90];
new_sp_delay=AMvectordelay(speaker_array_azimuth,speaker_array_elevation,
distance,azimuth,elevation,fs,SoundSpeed,filename);
```

  - *output*

```
new_sp_delay =

    Columns 1 through 9

    49.5068   49.5068         0   44.5401   17.5379   90.9198   17.5379   90.9198   49.5068
```

Columns 10 through 16

49.5068 90.9198 17.5379 90.9198 17.5379 49.5068 49.5068

The output files will be saved in 'C:\work\' with filenames:
decsp_1.wav – for the first channel
decsp_2.wav – for the second
…