



Audio Communication Group  
2024

## Documentation of Pure Data Applications for Dynamic Binaural Synthesis

First draft: 10.07.2024  
Latest Updated: 19.07.2024

Author:

Konstantin Fontaine — [k.fontaine@tu-berlin.de](mailto:k.fontaine@tu-berlin.de)

Supervision:

Emanuele Porcinai — [emanuele.porcinai@tu-berlin.de](mailto:emanuele.porcinai@tu-berlin.de)

Fabian Brinkmann — [fabian.brinkmann@tu-berlin.de](mailto:fabian.brinkmann@tu-berlin.de)

# Contents

<b>1</b>	<b>General Information</b>	<b>1</b>
<b>2</b>	<b>Components</b>	<b>1</b>
2.1	Source Input . . . . .	1
2.2	Source Player . . . . .	1
2.3	File Loader . . . . .	2
2.4	OSC Receiver . . . . .	2
2.5	Binaural Convolution (BRIRs) . . . . .	2
2.6	Binaural Sampler . . . . .	3
<b>3</b>	<b>Folder and Data Structure</b>	<b>3</b>
3.1	Dynamic BRIR Convolution . . . . .	4
3.2	Dynamic Sampler . . . . .	5
<b>4</b>	<b>Main Applications</b>	<b>5</b>
4.1	main_convolver_single.pd (Single Player) . . . . .	5
4.2	main_convolver_multiple.pd (Multiple Players) . . . . .	5
4.3	main_player_single.pd (Single Player) . . . . .	5
4.4	main_player_multiple.pd (Multiple Players) . . . . .	6
4.5	main_hybrid_single.pd (Single Player) . . . . .	6

# 1 General Information

In the following a documentation of Pure Data (PD) applications is provided, which cover several use-cases to conduct listening experiments by the means of a dynamic binaural synthesis for single or multiple player usage. It covers a presentation of the core components, followed by binding setup requirements and a brief explanation of the four main applications. The implementation hereby covers a partitioned real-time convolution of live inputs, based on the SoundScape Renderer (SSR) PD externals<sup>1</sup>, as well as a dynamic binaural playback of pre-convolved source materials for various head orientations.

## 2 Components

### 2.1 Source Input

The *source-input* subpatch serves as a graphical user interface to handle the amplitudes of incoming source signals. The first four interface input channels are assigned automatically as depicted in the GUI. A master gain serves to control the overall amplitude of incoming signals. To amend the input channel numbers, open the subpatch and edit its *dac* ~ object, which contains the corresponding interface channel numbers.

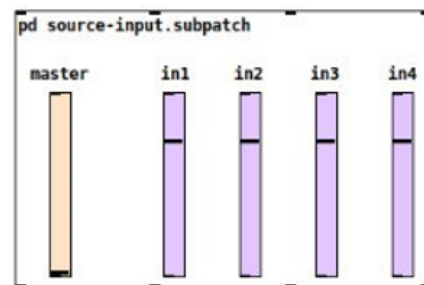


Figure 1: Source Input Mixer

### 2.2 Source Player

For the playback of mono files, the *source-player* provides one of four audio files to be played through its outlet. The GUI hereby offers options to start playback from a determined sample, loop the file or stop playback on demand. The four samples (drums, speech, guitar, noise)



Figure 2: Sample Player Interface

<sup>1</sup><https://ssr.readthedocs.io/en/0.6.1/use-cases.html>

are automatically loaded from the disk when running the main patch and can be selected from the GUI. To customize files, open the subpatch and adjust the fileloader strings.

### 2.3 File Loader

For the dynamic binaural playback of convolved sound files, the *file-loader* subpatch must be setup accordingly. Hereto the angular resolution (e.g.  $2^\circ$  steps) as well as the bipolar range (e.g. 60 for  $-60^\circ$  to  $60^\circ$  coverage) must be defined. Once set, a file container is generated containing a pre-convolved representation of the sound field. To assure its correct file management, refer to chapter 3.2.

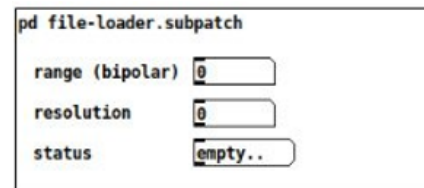


Figure 3: Grid Definition Interface

### 2.4 OSC Receiver

Tracking data is handled by the *osc-receiver* subpatch, which offers an OSC port selection and activation toggle, to allow data transmission within the main application. Head orientation is depicted by yaw, pitch and roll, while incoming OSC messages must have the shape "string1 string2 string3 <yaw> <pitch> <roll>", whereby the strings are discarded during processing.

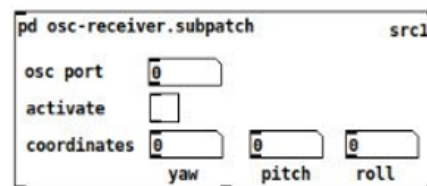


Figure 4: OSC Interface

### 2.5 Binaural Convolution (BRIRs)

The dynamic convolution of multiple sources per receiver is realized by the *binaural-convolution* subpatch, which automatically receives OSC data and

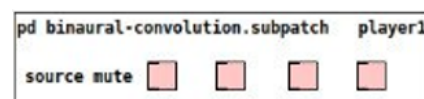


Figure 5: Binaural Convolution Setup

source inputs. The OSC receiver that is labeled as *src1* hereby sends its tracking data to the *binaural-convolution* subpatch labeled as *player1* and so forth. The four input sources of the live mixer are assigned to the four inlets of the binaural convolution subpatch and can be muted in the GUI for testing purposes. Within that subpatch two *ssr\_brs* ~ objects<sup>2</sup> convolve each incoming signal with the according BRIR set, in either a dynamic or static manner. While the dynamic part aligns the filter with corresponding head orientation, the static part convolves the unaltered frontal head orientation. To assure its correct file management, which is bound to naming convention and folder structure, refer to chapter 3.1.

## 2.6 Binaural Sampler

In case of pre-convolved sound fields, the dynamic playback is realized by the *binaural-player* subpatch. Also here, the player labeled as *player1* obtains its OSC data from the OSC receiver *src1*. Dynamically changing head orientations then pick the according reference convolution from the file container, that is described in section 2.3. Furthermore a start sample can be selected, the playback be looped and the volume be adjusted. Additionally the desired sample overlap resp. blend time can be set within the GUI, whereby an exponent of 2 is recommended. A turquoise signal feedback at the bottom indicates incoming OSC data.

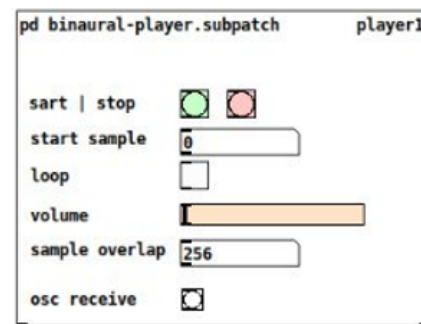


Figure 6: Binaural Player Setup

## 3 Folder and Data Structure

For a proper usability of the applications, certain conventions must be met to assure a correct assignment of files. In general, indices below 10 are represented by single digit numbers within filenames. The following describes conventions for the *binaural-convolution.subpatch* and the *binaural-player.subpatch*.

<sup>2</sup><https://ssr.readthedocs.io/en/latest/use-cases.html>

### 3.1 Dynamic BRIR Convolution

The BRIR sets for the dynamic convolution are automatically read from disk and assigned to the renderers accordingly, once the patch is started. The folder structure in which the sets are located should therefore look as depicted in figure 7.

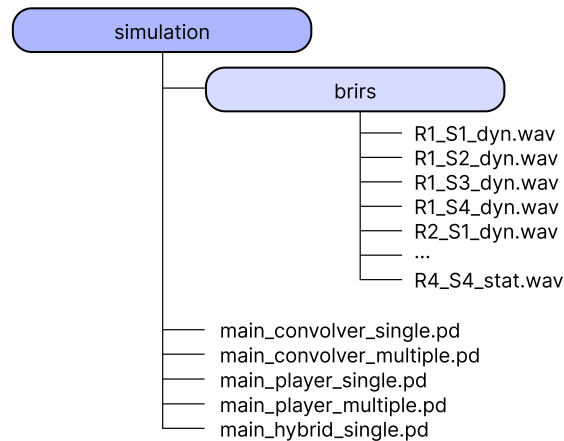


Figure 7: Folder Structure for the BRIR Simulation

To pick and assign the BRIR sets automatically, all file names should follow the convention of naming the source index followed by an underscore, the receiver index, another underscore and an indicator for the static (*stat*) or dynamic (*dyn*) BRIR component (e.g. *S1\_R2\_dyn.wav* for the dynamic part of the second receiver with source index one). While the wav files labeled as "...*dyn.wav*" are assigned to the dynamically adaptive renderer, sets labeled as "...*stat.wav*" are loaded by the statically behaving renderer. The folders should thus each contain 32 wav files covering all receiver-source-combinations, for the case of four simultaneously performing musicians. Each dynamic BRIR set must then contain 720 channels that are ordered as presented in the SSR binaural room simulation renderer documentation.<sup>3</sup>

**All BRIR sets must be provided in order to successfully instantiate the renderers. In case the entire BRIR should be dynamically convolved, the static files can be discarded, while the occurring initialization error at patch start can be neglected. If fewer sources are desired, either edit the subpatch or leave input channels unused.**

<sup>3</sup><https://ssr.readthedocs.io/en/0.6.1/renderers.html>

### 3.2 Dynamic Sampler

For dynamically playing back already convolved sound streams, the *file-loader* subpatch requires files by the name convention *convolved\_Rx<sub>1</sub>\_x<sub>2</sub>azim\_x<sub>3</sub>elev\_left.wav*, while  $x_1$  denotes the receiver index,  $x_2$  the yaw value and  $x_3$  the pitch value followed by the corresponding ear signal. Once the range and resolution are set, the *binaural-player.subpatch* component automatically obtains the buffer indices accordingly and becomes available for dynamic playback.

## 4 Main Applications

In case of the live convolution patches, DSP setup and activation is recommended prior starting the patches, to ensure a correct object initialization. The PD sampling rate must match with the sampling rate of the BRIR sets. For the dynamic player, maximum blend time in samples is recommended to be equal to the DSP block size.

### 4.1 main\_convolver\_single.pd (Single Player)

In the single player live convolver, an automatic assignment of four BRIR sets (each representing distinct source positions for one receiver) is applied when starting the patch. Usage thus only requires OSC port settings and activation. When input streams are muted or equal to zero, computation is compensated and thus more efficient. The four mixer inlets offer alternative source inputs to be passed through to the renderer, here exemplary the mono stream from the source player output can be convolved.

### 4.2 main\_convolver\_multiple.pd (Multiple Players)

The multiple player main application, enables four live convolutions of each four receiver-source-combinations. Each of them convolve the dynamic early component and a subsequent static part in alignment to the head orientation of the individual player. Tracking data is automatically committed to the corresponding convolvers once their ports are selected and activated.

### 4.3 main\_player\_single.pd (Single Player)

When dynamically simulating pre-convolved sound fields by the means of the *main\_player* patch, the file container must be setup initially, as described in chapter 2.3. In a subsequent step OSC port selection and tracking activation then enable the simulation.

#### **4.4 main\_player\_multiple.pd (Multiple Players)**

For multiple players the same processing is applied. The file containers are synchronized in regards to angular range and resolution, thus solely a single file loader must be defined. To assure coherence amongst the players, the read pointer positions and playback-related GUI parameters are kept in synchronization to one another, while the head orientation remains individualized for each musician.

#### **4.5 main\_hybrid\_single.pd (Single Player)**

The hybrid mode combines a dynamic convolution of four live inputs with a synchronized dynamic playback of pre-convolved files. For its use file loader and OSC receiver have to be set up as described above. The mode is solely offered for single player use yet.