Sound Field Synthesis Toolbox

Hagen Wierstorf[1] Sascha Spors[2]

[1]Assessment of IP-based Applications, [2]Quality and Usability Lab
Telekom Innovation Laboratories, Technische Universität Berlin, Germany
{hagen.wierstorf,sascha.spors}@tu-berlin.de

Deutsche Telekom Laboratories

Abstract

An open source toolbox for Sound Field Synthesis (SFS) is introduced. The toolbox is able to numerically simulate sound fields synthesized by SFS methods like Wave Field Synthesis (WFS) or higher order Ambisonics (HOA). Various loudspeaker driving signals for the mentioned methods are provided for 2-, 2.5- and 3-dimensional synthesis. The toolbox allows mono-frequent as well as broadband excitation signals. The latter enables to generate snapshots of the spatio-temporal impulse response of a chosen reproduction technique. The toolbox furthermore includes the computation of binaural room impulse responses (BRIR) for a given SFS setup. These can be used to simulate different sound field synthesis methods via binaural resynthesis. The toolbox is provided for Matlab/Octave and comes with an online documentation.

Introduction

Sound field synthesis offers the possibility to create a determined sound field within an extended listening area. Common methods to reach this in reality are Wave Field Synthesis or higher order Ambisonics which apply different kinds of driving signals to a loudspeaker array in order to control the sound field within the listening area. For real loudspeaker setups the SoundScape Renderer (SSR) [1] can provide such driving signals. The SSR is an open source software, which is also developed by our group.

To investigate the properties of different sound field synthesis methods and implementations, it is preferable to simulate their behavior beforehand. To close this gap, the *Sound Field Synthesis Toolbox* provides functions to numerically simulate WFS and HOA for 2-, 2.5-, and 3-dimensional synthesis (see e.g. [2]). These functions provide simulations for the sound fields of mono-frequent as well as broadband virtual sources. The latter allows to generate snapshots of the spatio-temporal impulse response for a given SFS method.

As an alternative to applying expensive, real loudspeaker setups, dynamic binaural synthesis can simulate different loudspeaker setups to evaluate sound field synthesis methods (e.g. [3],[4]). Therefore, binaural room impulse responses (BRIRs) are convolved with driving signals to simulate a synthesized sound field via headphones. The *Sound Field Synthesis Toolbox* can compute these BRIR data sets by applying the driving functions that are used for the simulation of the sound fields.

Access

The *Sound Field Synthesis Toolbox* is freely available under the GNU General Public License v3 and can be downloaded at

http://dev.qu.tu-berlin.de/projects/sfs-toolbox/files

It comes with detailed built-in help, available via the `help` function within Matlab/Octave. An additional wiki page provides online help, a tutorial for the first steps and different use cases at

http://dev.qu.tu-berlin.de/projects/sfs-toolbox/wiki

References

[1] Geier M, et al. The SoundScape Renderer: A Unified Spatial Audio Reproduction Framework for Arbitrary Rendering Methods. In *124th AES Conv.* 2008.

[2] Ahrens J. *Analytic Methods of Sound Field Synthesis.* Springer, Berlin, 2012.

[3] Wierstorf H, et al. Perception of Focused Sources in Wave Field Synthesis. *J Audio Eng Soc*, accepted.

[4] Völk F, et al. Simulation of wave field synthesis. *J Acoust Soc Am*, 123(5):3159, 2008.

[5] Donoho DL, et al. Reproducible Research in Computational Harmonic Analysis. *Comput Sci Eng*, 11(1):8–18, 2009.

[6] Ince DC, et al. The case for open computer programs. *Nature*, 482(7386):485–488, 2012.

Driving functions

The theory of sound field synthesis assumes a listening area surrounded by elementary sound sources, referred to as secondary sources. The question of how to drive these secondary sources to get a determined sound field within the listening area can answered by solving the following equation as [2]

$$P(\mathbf{x},\omega) = \int_{\partial V} D(\mathbf{x}_0,\omega)\; G(\mathbf{x}-\mathbf{x}_0,\omega)\; dS \;,$$

with $\mathbf{x}$ a position within V, $\mathbf{x}_0$ the position of the secondary sources on surface ∂V, dS an infinitesimal surface element, P the desired sound field, driving function D and Greens function G, which describes the sound field emitted by a secondary source.

The following loudspeaker setup geometries and driving functions are implemented in the *Sound Field Synthesis Toolbox* so far.

	WFS			HOA		
	2D	2.5D	3D	2D	2.5D	3D
plane wave	x	x			x	
point source	x	x			x	
focused source	x	x				

Example: plane wave in 2.5D WFS

```
% Start Toolbox and get a config struct
SFS_start;
conf = SFS_config;
% Define geometry of listening area and array
conf.array = 'linear';  % array type
conf.dx0 = 0.15;        % loudspeaker distance
L = 3;                  % array length
X = [-2,2];             % simulated area
Y = [-0.15,3];
conf.xref = [0,2];      % WFS amplitude reference point
src = 'pw';             % source type (plane wave)
xs = [0.5,1];           % direction of plane wave
% Simulate mono-frequent wave field
f = 1000; % Hz
[x,y,P,ls_activity] = wave_field_mono_wfs_25d(X,Y,xs,L,f,src,conf);
% Simulate spatio-temporal impulse response
conf.frame = 200; % time index
[x,y,p,ls_activity] = wave_field_imp_wfs_25d(X,Y,xs,L,src,conf);
```

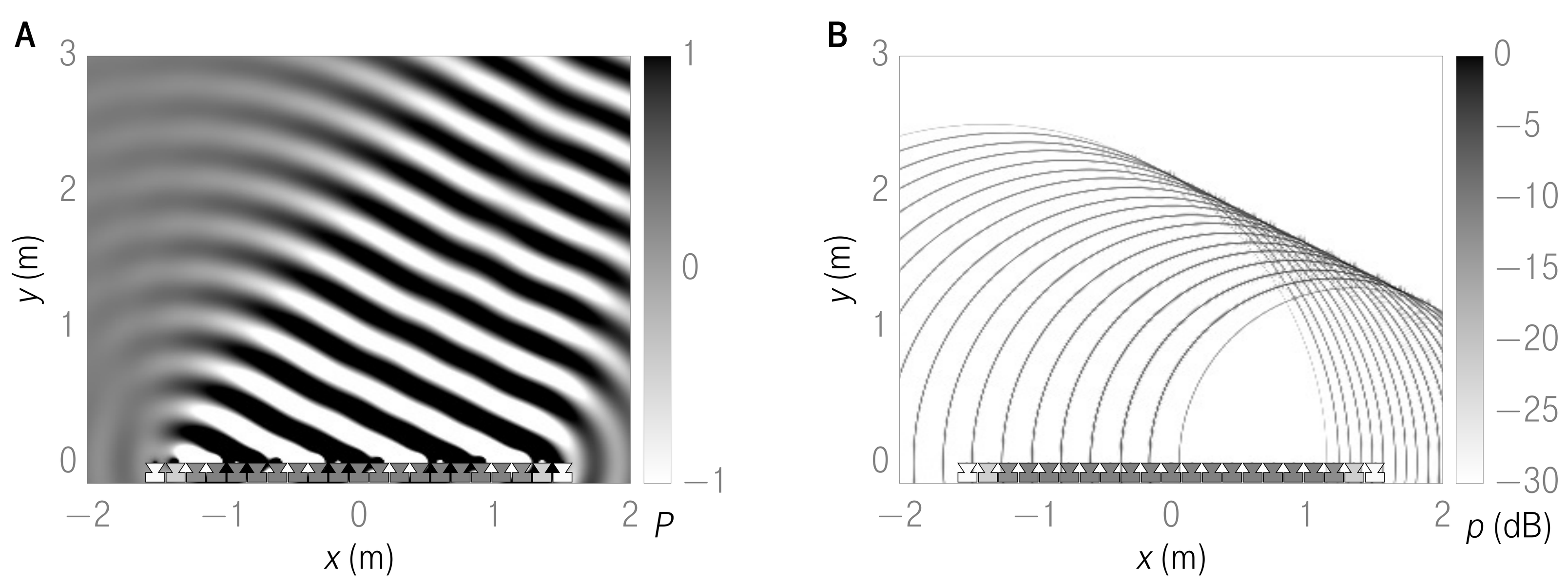

Figure: The mono-frequent (A) and a snapshot in time of the spatio-temporal impulse (B) response of a linear loudspeaker array for the 2.5D WFS driving function for a plane wave is shown. For the mono-frequent part the frequency was $f = 1000$ Hz. The direction of the plane wave is 63°.

Additional Features

- Binaural simulation of SFS using HRIRs/BRIRs (see Laptop demo)
  These are extremely useful to perform listening experiments
- Large number of functions to handle HRIRs, including extrapolation
- Fractional delay filter
- Matlab/Octave and Gnuplot plotting routines specialized for SFS

Reproducible Research

- Follows the principle of *Reproducible Research* [5, 6]
- Signal processing and related fields are vulnerable to implementation errors
- Release code and data with your paper

Acknowledgments

This work was supported by DFG RA 2044/1-1.

Example: point source in 2.5D WFS

```
% Start Toolbox and get a config struct
SFS_start;
conf = SFS_config;
% Define geometry of listening area and array
conf.array = 'circle';  % array type
conf.dx0 = 0.15;        % loudspeaker distance
L = 3;                  % array diameter
X = [-2,2];             % simulated area
Y = [-2,2];
conf.xref = [0,0];      % WFS amplitude reference point
src = 'ps';             % source type (point source)
xs = [0.5,2];           % position of point source
% Simulate mono-frequent wave field
f = 1000; % Hz
[x,y,P,ls_activity] = wave_field_mono_wfs_25d(X,Y,xs,L,f,src,conf);
% Simulate spatio-temporal impulse response
conf.frame = 100; % time index
[x,y,p,ls_activity] = wave_field_imp_wfs_25d(X,Y,xs,L,src,conf);
```

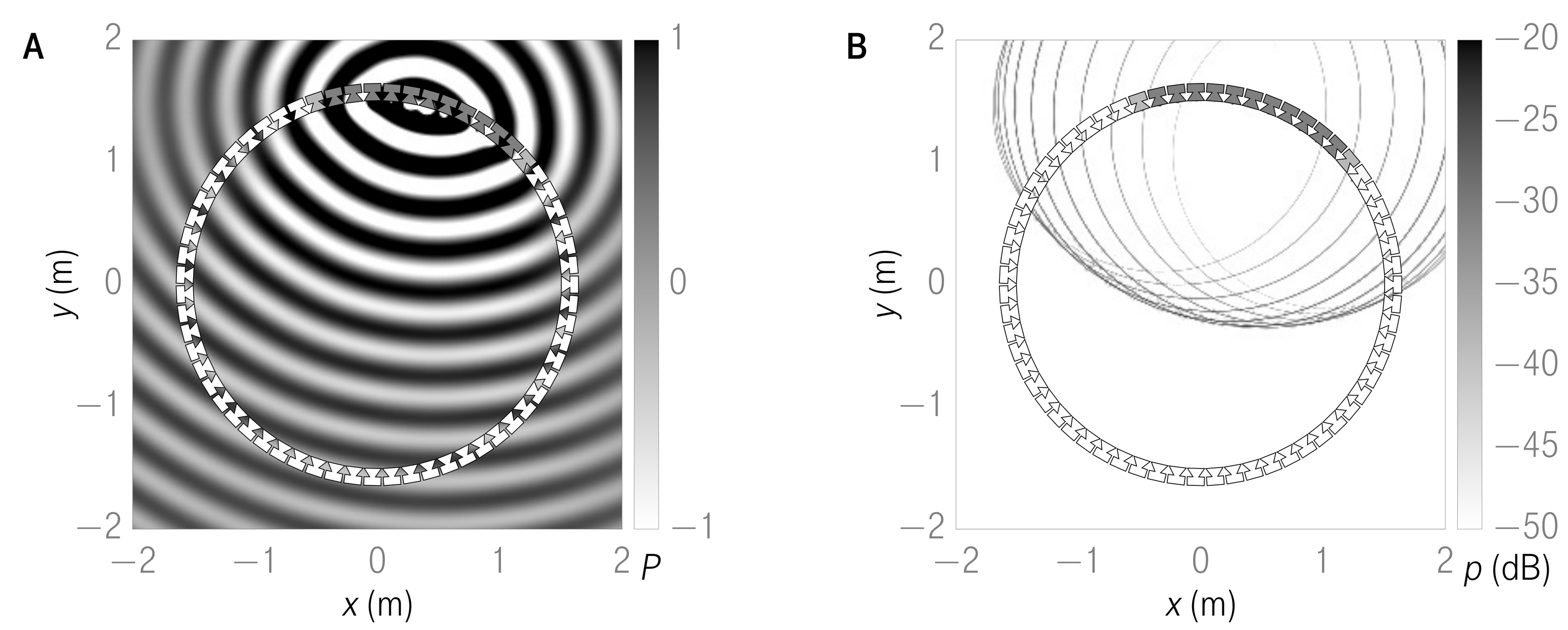

Figure: The mono-frequent (A) and a snapshot in time of the spatio-temporal impulse (B) response of a circular loudspeaker array for the 2.5D WFS driving function for a point source is shown. For the mono-frequent part the frequency was $f = 1000$ Hz. The position of the point source is $(0.5, 2)$ m.