Audio effects for multi-source auralisations

Claus Lynge Christensen
Odeon A/S
Kgs. Lyngby, Denmark
clc@odeon.dk

George Koutsouris
Odeon A/S
Kgs. Lyngby, Denmark
gk@odeon.dk

Antoine Richard
Odeon A/S
Kgs. Lyngby, Denmark
ar@odeon.dk

Jens Holger Rindel
Odeon A/S
Kgs. Lyngby, Denmark
jhr@odeon.dk

***Abstract*—Implementing multi-source auralisations is essential for comprehensive acoustic studies of concert halls, operas, restaurants and other spaces, where many sound sources are active. Nowadays, room simulation programs and computers are fast enough to calculate the many impulse responses required for such auralisations, in short time. However, setting up the calculations for the entire ensemble of sources and corresponding signals requires a lot of time-consuming manual work and a large number of recorded source signals. To facilitate the process of creating multiple source signals, with sufficient variation for auralisation, a set of audio effects has been developed in the ODEON Room Acoustics Software [1]. The effects can be combined together and help to create a population of incoherent output signals by varying certain attributes of just a single input signal. Two examples are: a) A group of violins, composed of several signals that have been generated as variations of just one violin recording, using individually randomized Chorus, Spectrum and amplitude effects. b) A number of speech signals with a given output length, created by performing randomized pitch change, tempo shift and random amplitude modulation. The length of the output speech signal can be obtained by time shifting and repeating each of the processed signals with pauses of random length in-between until the desired total length is obtained.**

***Index Terms*—amplitude modulation, auralisation, audio effects, audio processing, chorus effect, phase vocoder, room acoustic simulations**

I. Introduction

Room acoustic simulations are commonly used in the acoustic design of spaces, and their results are typically presented in terms of objective parameters, such as reverberation time, sound pressure level, etc. However, auralisation offers a subjective approach to assessing the simulation results, where one can listen how a given signal would sound in the virtual space.

Two main inputs are needed for auralisation. First, a room impulse response should be obtained between a source position and a receiver position; this sound file describes the acoustic propagation between the two points in the room. The impulse response can either be simulated or measured in a real room. Second, an anechoic signal (i.e., without reflections) should be chosen as the signal produced by the source. These two signals are *convolved* together to obtain an audio file which recreates how a listener at the receiver position hears the input signal from the source [2].

Occasionally, auralisations with more than one source are desirable, for instance in the creation of soundscapes [3], [4], which can provide an immersive acoustic experience. In such a case, multiple sources are used in the acoustic model, and each source is assigned a specific anechoic file. Still, only one receiver location is used with a fixed direction, no matter how many sources are involved. In the ODEON Room Acoustic Software [1] an impulse response is first computed from every source to the receiver, and each anechoic file is convolved with its corresponding impulse response. Finally, all convolved signals are mixed together in a single output file. The resulting file includes impressions of spaciousness, source localization and distance, which would not have been perceivable with a single sound source. An intuitive Auralisation mixer and Multi source/signal auralisation expert is available in the Auditorium and Combined editions of ODEON, allowing auralisations with up to 300 independent sources. Several demonstrations of the tools can be found on Odeon's home page [5].

Auralisation with multiple sources requires many input files (one per source), especially for recreation of complex environments or soundscape scenarios. These anechoic files may be difficult to record or to obtain. For example, the anechoic recordings of symphonic music by Vigeant *et al.* [6][1] or later by Pätynen*et al.* [7] include only a few recordings of the string instruments in the orchestra, as it is challenging to record each musician performing the same passage separately. Consequently, typically one or two string players are recorded and the rest of the string section is filled with copies of them. In an early example of orchestra simulations [8], all instruments were modeled as omni-directional point sources and identical recordings were used for the instruments in the string section. However, using the exact same audio file for two or more distinct sources is not desirable and not realistic. As it happens with stereophonic or other loudspeaker arrangements, the result is close to a new phantom source location [9], but not to a spatially full string ensemble. Real musicians perform the same piece on different instruments and in slightly different ways, with small delays, pitch changes and dynamics, which should also be included in auralisations. Several attempts have been suggested for creating an ensemble out of few files, using various phase-shifting techniques [6], [10]. Variations could be added manually in a Digital Audio Workstation (DAW) software, but it may introduce bias in the resulting output files.

To address this issue, an Audio Effects tool is included in ODEON since version 15, as shown in Fig. 1. ODEON makes use of different audio effects to introduce a degree

[1]These recordings have been used extensively in various ODEON simulations (https://odeon.dk/brahms-aarhus-house-of-music/)

of decorrelation between similar sound sources. Each effect introduces variations in terms of pitch, speed, gain, spectrum, etc., governed by random parameters. These parameters are selected within a user-defined range. The Audio Effects tool allows to automatically create a population of similar files from a single anechoic recording, thus reducing the required number of recorded input signals.

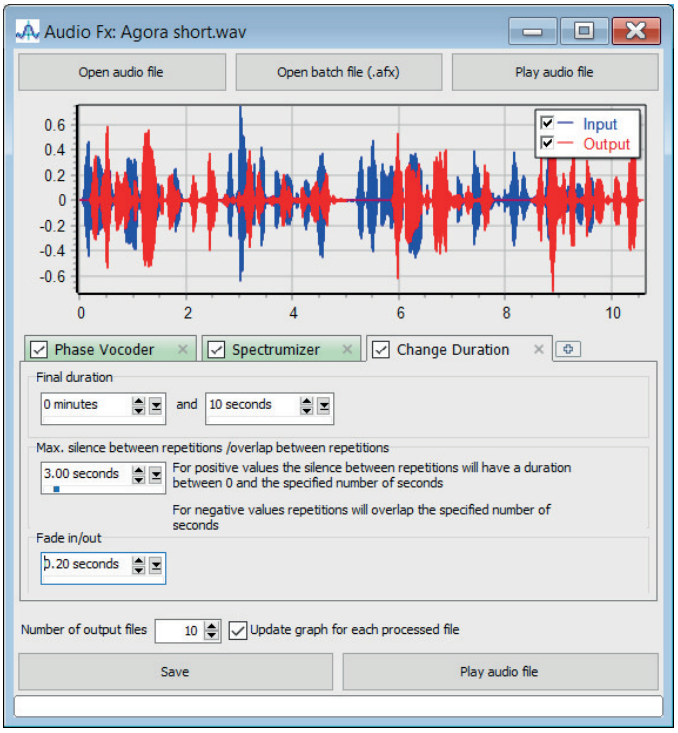

Fig. 1. ODEON Audio effects editor. In this example, three effects are included in the chain: Phase Vocoder, Spectrumizer and Change Duration.

In this paper the workflow of creating multi-source auralisations, using the tools in ODEON, is first presented. Then a brief introduction into some of the audio effects currently available in ODEON and how they can be combined together to transform the properties of multiple anechoic signals is given. Two major examples are discussed.

The first example is the simulation of a string chamber orchestra, consisting of violins, violas and cellos. All the violins are assumed to play the same melody. The audio effects tool is used to imitate the individual variations within a group of musicians, in terms of timing, pitch or strength, which sounds more realistic and closer to a group effect than using the same file for each instrument. The second example presents a simulation speech of a crowd noise in a cafeteria. The crowd is represented by a set of point sources to which speech signals are assigned. In reality, such a crowd consists of a large number of individuals, who act as uncorrelated sound sources and emit speech signals with different attributes, such as pitch, level and "content" itself. These attributes do not only vary from person to person but also with time. In this application, two speech signals (male and female) are used to create different speech files with varying attributes, so that their combination sounds convincingly like crowd noise.

II. Creating Multi-Source Auralisations

Fig. 2 shows a flow chart of the overall multi-source auralisation process in ODEON. One or more input signals are processed by a chain of Audio Effects, which produces a number of output files, corresponding to the number of sources. Each output file is convolved with a Room Impulse Response (RIR) from one of the sources to a fixed receiver position. The final mixing is performed in the Auralisation mixer, resulting in a single Binaural or B-format output file [1]. In this graph the chain of Audio Effects is considered a black box, that transforms the input (anechoic) signal and creates as output a population of files. The possible effects used will be described in Section III. It should be emphasized that one auralisation mixing process can include several populations of files, which originate from several corresponding input signals (eg. the different string sections in an orchestra that can be created from a single violin, a single cello etc.). Any combination of populations as well as individual, unprocessed files is possible.

In Fig. 3 the chain of the audio effects is shown in detail, illustrating the process of creating N incoherent files out of a single input signal, by applying random variation of the corresponding effect parameters. First, the input signal is copied N times. Then, each copy of the signal is passed through a chain of effects, that consists of any combination from those listed in Section III. No matter what effects are chosen in the chain, the Change duration effect is automatically performed at the end, to ensure that the specified duration for each output file is kept; as it is important to have one common signal length of all input signals when creating multi-source soundscapes.

The parameters of each effect vary randomly, within the limits specified by the user. The random variations of each effect follow a *uniform distribution*, except for the chorus effect, which follows a *normal distribution*.

III. List of Audio Effects in ODEON

Since ODEON 15, the audio effects library is continuously expanded to serve the needs of advanced multi-source auralisations. We will briefly explain five of the available effects, which are used in the application cases in Sections IV and V.

A. Change Duration

Creating scenes and soundscapes containing multiple speech signals, music, noise events etc. will typically require that the input signals to the multi source auralisation have the same length. Whether the original source file is shorter or longer than the required scene length, the duration effect is used to create a new file of a fixed length. When the final duration is longer than the input signal, the new file consists of repetitions of the original signal and silences between them. On the other hand, when the final duration is shorter, the new file is simply a cropped version of the original signal. The user determines the *final duration* of the file, the *length of silence* (or a range

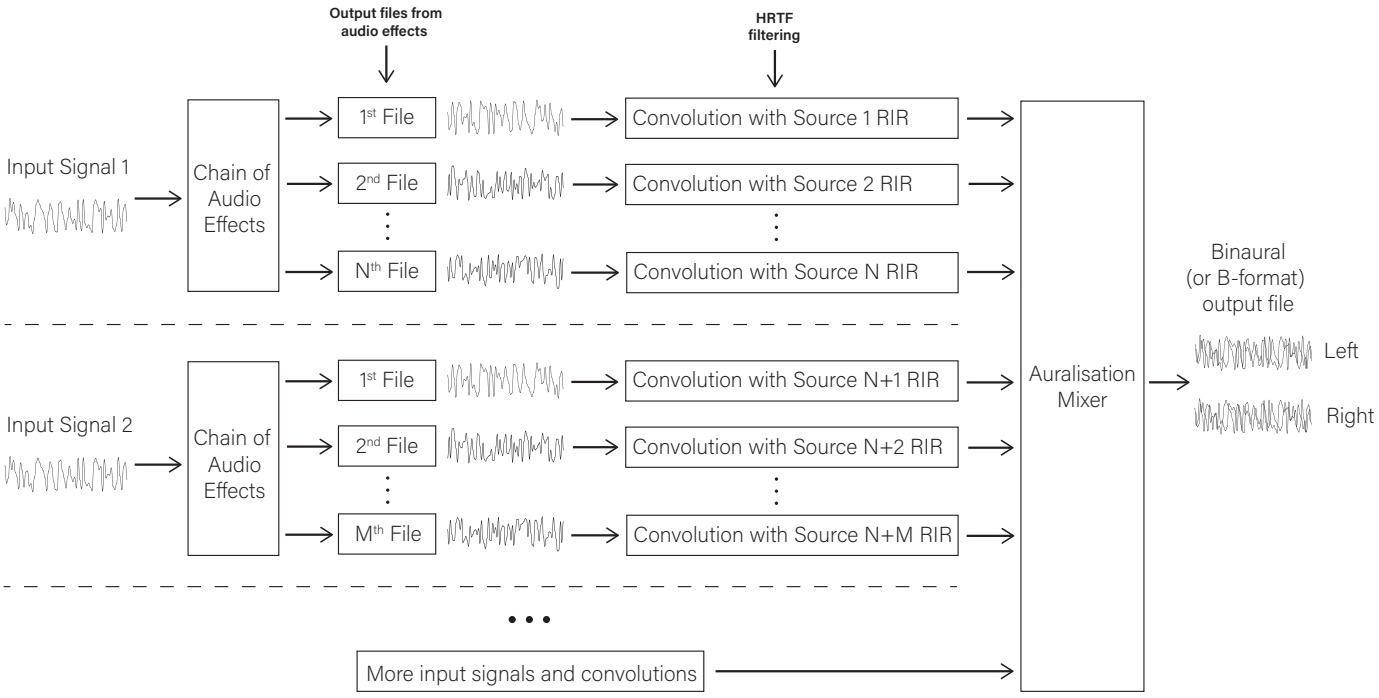

Fig. 2. Diagram of the overall multi-source auralisation process. In this example, two input files are processed through the chain of audio effects, giving N and M output files. These file populations are convolved with the RIR associated with an equal number of sources in the room. Finally, all convolutions are mixed together in the Auralisation mixer for a single binaural or B-format file.

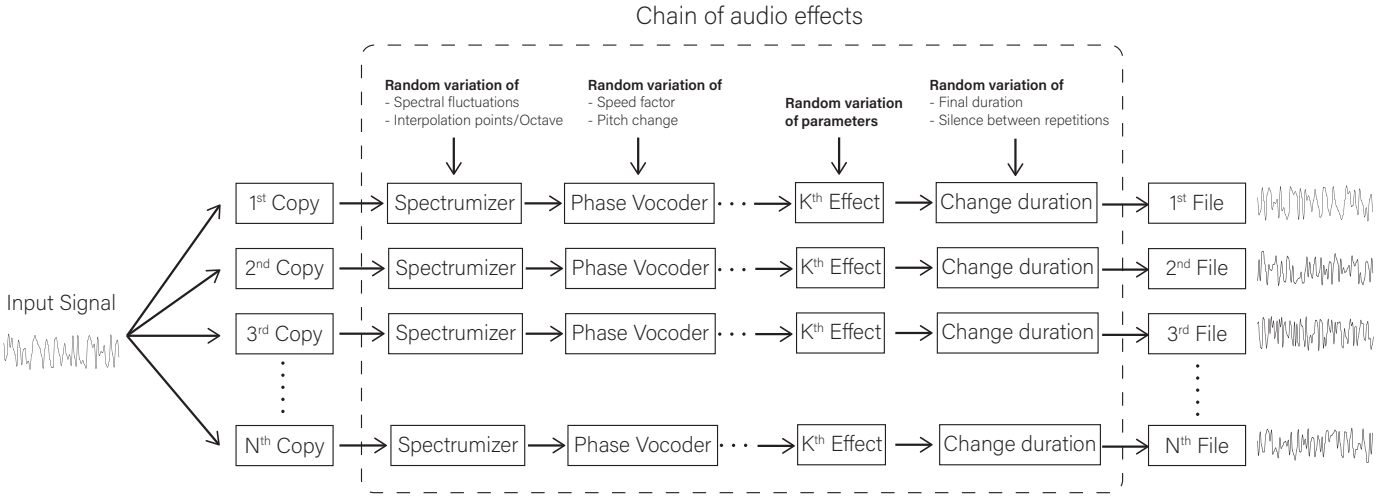

Fig. 3. The chain of audio effects in ODEON can be applied with random variations on a single input signal to create a bank of different files.

of random silences) between repetitions and the *length of fade in/out* for the whole output signal.

A population of output files with a longer final duration and random variations of silence between them is constructed according to the following algorithm :

1) A random location is determined between the start and end of the input signal. This location is not controlled by the user.
2) A random period of silence is added at the end of the file, according to the maximum limit given by the user (the default value is 3 s). This is useful especially when working with speech. A copy of the input signal will follow the silence period. If the user specifies a negative silence period, then the repetitions of the signal will overlap, without a silence. To accomplish this, ODEON applies cross fading between the repetitions. A negative period of silence is useful when attempting to extend the duration of a stochastic signal, such as noise.
3) The first two steps are repeated until the length of the output signal is exceeded and the output signal is truncated at the desired length.
4) Fade in/out is applied at the beginning and at the end of

the output signal, in order to avoid clicking sounds and give the desired duration to the output file.

Fig. 4 illustrates the process. Since other effects may change the length of the signal, the *Change Duration* effect is always added as the last effect in the chain, to ensure that the output will end up having the specified duration. An exception to this rule is when the *Amplitude modulation effect* (see Section III-E) is used in the chain. Two reasons justify this exception: 1) It might be desirable to use this effect after the Change Duration effect to vary the gain across the whole final length. 2) The *Amplitude modulation* is applied directly in the time domain signal and does not alter the original duration.

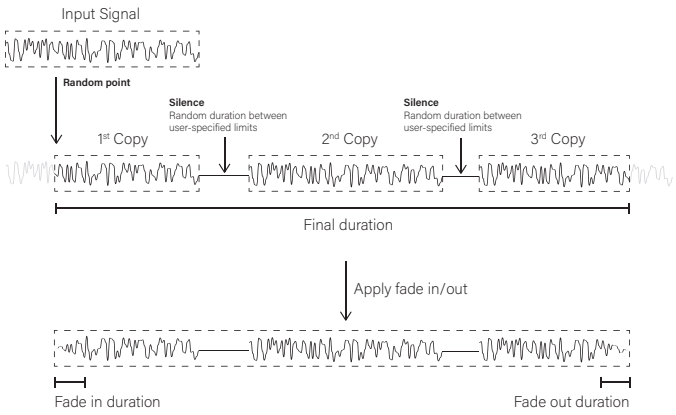

Fig. 4. Illustration of how the *Change Duration* effect works in ODEON.

B. Phase Vocoder

A common way to change the pitch of a signal is by resampling it in the time-domain. For example, when a signal is downsampled (fewer samples per second), then the pitch rises, but at the same time the duration becomes shorter (higher playback speed). On the other hand, if the signal is upsampled (more samples per second), then pitch becomes lower, but duration becomes longer.

However, it is often desirable to change the pitch and speed independently. The Phase Vocoder belongs to the time-frequency processing audio effects and among others it offers that particular function. It can be used for example to make a speech signal slower, without changing its pitch.

The effect transforms the time-domain signal into a time-frequency-domain representation, where its Magnitude and Phase can be modified. The modified frequency-domain representation is then converted back to the time-domain output signal. The process starts by segmenting the time-domain signal with a sliding window of finite length N. For each segment a Fast Fourier Transform (FFT) is applied to obtain the frequency-domain (spectral) representation. The result is a spectrum that varies with time. If n is the discrete time index and $k = 0, 1, \ldots, N-1$ are the frequency bins included in the sliding window, then the spectrum for each windowed segment is defined as: $X(n,k) = |X(n,k)|e^{j\phi(n,k)}$. The amplitude and phase of the short-time spectra of each segment can then be modified. Each segment is then converted back to the time-domain by applying an inverse FFT. All time-domain segments are finally windowed and overlapped to form the output signal [11].

C. Chorus

The *Chorus effect* is the result of multiple signals of similar pitch and timing played together and perceived as one. Unlike the classic chorus effect - where one output file is produced, the chorus effect in ODEON produces a separate audio file for each signal in the group. Older implementations of the *Chorus effect* employed the PSOLA method [10] or manual delay applied to the overall signal [6]. To achieve a more realistic variation of pitch and timing, the implementation of the *Chorus effect* in ODEON is based on *fractional delay lines* [11] that have random variations as a function of time.

Typical chorus effects include the section of strings in an orchestra or a choir of voices within the same vocal range. Although musicians in a string section as well as the singers in a choir try to perform the exact same musical piece, there are always slight variations in pitch and time of musical notes. This is because the timing and performance of the individual musician or singer are different from other musicians/singers, while the instruments used can differ significantly from each other.

According to Pätynen *et al.* [12], the onset time of musicians in a violin section follows a normal distribution function with a standard deviation (σ) of 40 ms. Assuming that the actual onset time (the mean of the normal distribution) is 0.00 ms, approximately 68.2 % of the onset times will occur within $\pm$ 40 ms ($\pm$ 1 σ), 27.2 % within $\pm$ 80 ms ($\pm$ 2 σ) and 4.2 % within $\pm$ 120 ms ($\pm$ 3 σ). We assume that a normal distribution can be used for any kind of signals generated by ODEON's Chorus effect implementation - not only for violins, but also the rest of the strings, as well as voices. The chorus effect in ODEON is implemented according to the following algorithm, utilizing a fractional *delay line*:

1) A vector of random delays as a function of time is constructed with the same length as the input signal. The vector is constructed in two steps: a vector of control points is made with the points placed at regular intervals specified by the *frequency parameter* of the effect. Each control point is assigned a random delay which varies around a central delay between $\pm$ a *depth parameter*, given in percent. A final delay vector is constructed by interpolation between the control points.
2) The delay times in ms are converted to non-integer sampling intervals, in order to keep continuity in the delay vector. For example, if the sampling frequency f_s is 44100 Hz, a delay of 100.309 ms is converted to 4423.627 samples.
3) A sampling interval *delay line* x with length equal to the maximum number of samples in the delay vector is constructed.
4) Each sample of the input signal is moved by the exact number of samples of the corresponding cell in the

delay line. Since the movement is done by a non-integer number of samples, fractional delay has to be utilized in the following way:

- The integer part of the delay in samples is stored as i.
- The decimal part of the delay in samples is stored as $frac$.
- Each sample n of the output signal is calculated by linear interpolation between the integer part of the current delay i and the previous $i-1$:

$$y[n] = x[i] \cdot frac + x[i-1] \cdot (1 - frac) \quad (1)$$

- Each (integer) sample of the input signal is stored into the first cell of the *delay line*, while the rest samples in the *delay line* are pushed one cell forward.

5) All samples in the delay line are pushed further by one cell.

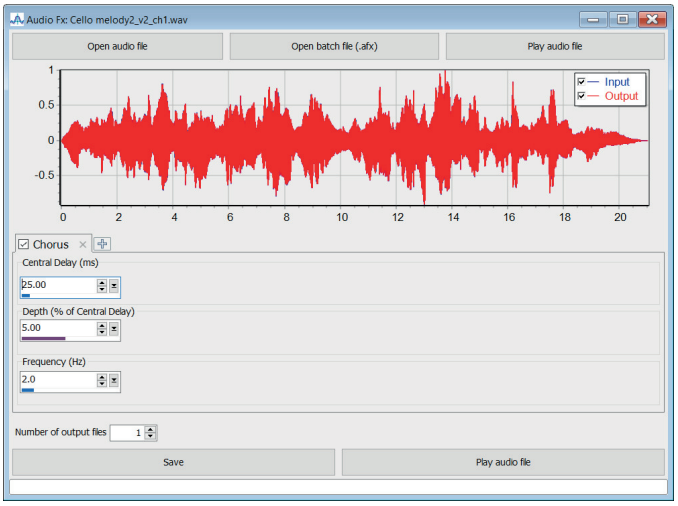

Fig. 5. Example of settings used for the Chorus Effect, applied on a 20 s long music signal by a Cello. Input signal is in blue and output signal is in red.

Fig. 5 shows an example of a 20 s music passage, by a cello. The central delay is set at 25 ms, the chorus depth at 5 %, while the frequency of control points is at 2 Hz. The corresponding vector of random delays is given in Fig. 6, as a function of time. It can be seen that the changes of the output signal are not easily noticeable in the waveform (in red colour), when compared to the input signal (in blue colour). This is because the effect parameters are typically chosen to cause only slight variations of the music passage, which will make the file sound as if it was played slightly differently, but not distorted.

D. Spectrumizer

The spectrumizer makes random changes to the spectrum of the input, similar to an Equalization (EQ) filter. The main input parameter is the Spectral Random fluctuations in percent which is by default 50 % meaning the level may be decreased

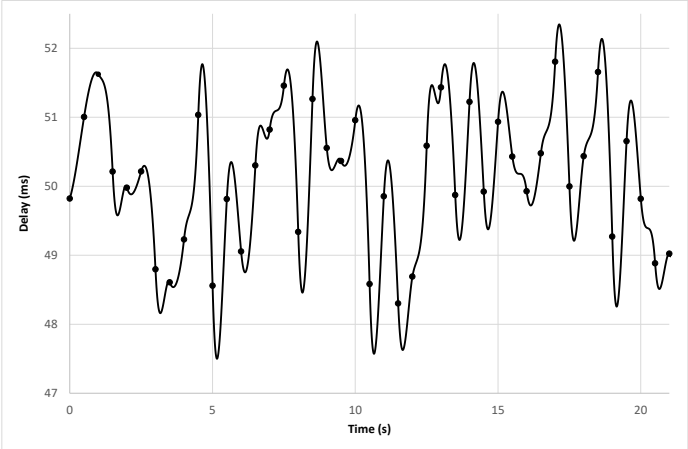

Fig. 6. Random variation of delay as a function of time for a cello music passage of about 20 s length. The central delay is specified at 25 ms. The chorus depth is ± 5 %, while the frequency of the control points is 2 Hz, which corresponds to a change every 0.5 s.

up to 6 dB at some frequencies and increased up to 3 dB at some frequencies.

By adjusting the levels in percentage, rather than in a dB scale, it is ensured that the average level of an output signal is the same as that of the input signal which is important for a controlled design of soundscapes. The spectral changes are made at a number of frequency points specified by the user. The points are distributed logarithmically leading to the same degree of randomness per octave. The maximum number of random points per octave sets a lower limit to the lowest frequency that will be modified.

E. Amplitude Modulation

The Amplitude Modulation effect changes the amplitude of the signal by a random envelope that is defined between a minimum and a maximum change in dB. Fig. 7 shows an example of random amplitude modulation applied to a speech signal of 10 s. The range for gain change is specified between 50 % and 80 %. However, the algorithm finds a single random percentage value, within this range so that the power is increased or reduced up to an equal amount. Similarly, a random period is selected between the specified range from 0.1 s and 0.6 s. Like in the Chorus Effect (Section III-C), this period determines the spacing of control points that are assigned random gain changes between the minimum and maximum percentage. The final random amplitude modulation curve is constructed by interpolation between the points.

Fig. 8 shows a typical random variation of the gain applied to the signal throughout its duration. We can see that the gain varies approximately between 0.4 and 1.6, which corresponds to a maximum of ± 60 % change of the original power. A series of random-gain change control points is constructed at a fixed period of 0.5 s, which corresponds to the random period that the algorithm selected between the specified range of 0.1 s and 0.6 s.

According to Fig. 7, from the resulting waveform (in red colour) it can be seen that the new amplitude is reduced significantly at the time between 6 and 7 s, in correspondence with the major dip of the gain function in Fig. 8. Conversely, a major boost is happening at the time between 1 and 2 s. However, this is not so obvious in the waveform of Fig. 7, because the input signal (waveform in blue colour) is much lower at this region and it is overlapped by the output waveform.

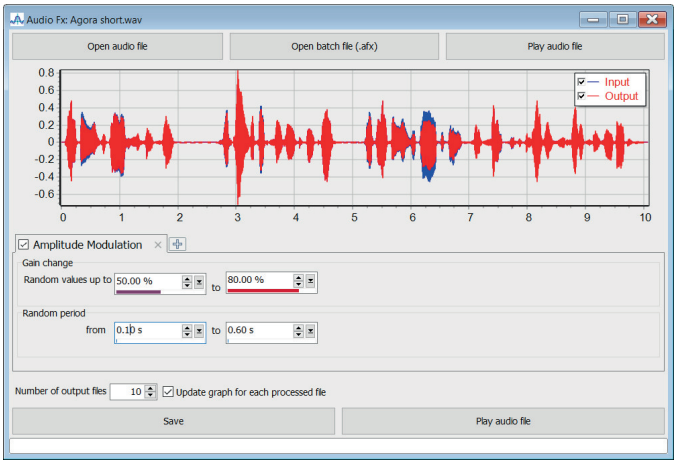

Fig. 7. Example of settings used for the Amplitude Modulation Effect, applied on a 10 s long speech signal. Input signal is in blue and output signal is in red.

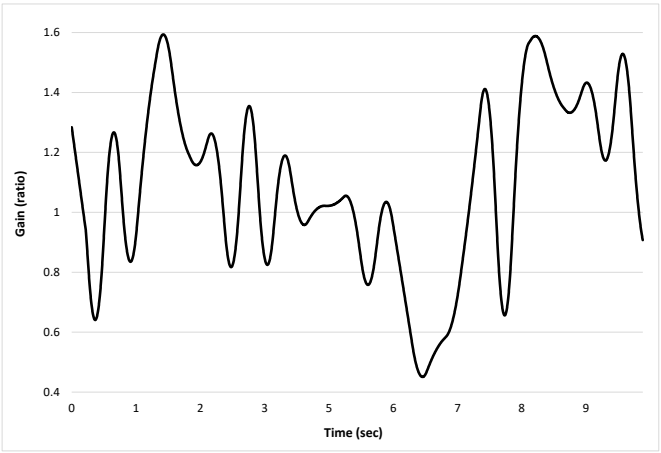

Fig. 8. Random variation of gain as a function of time for a signal of 10 s length. In this case, the gain varies between ± 60 %, which corresponds to a ratio from 0.4 to 1.6, relative to the original power/sample of the signal.

IV. Application 1: String Section in Orchestra

In this application we create a large string orchestra out of a string quartet, that performs an excerpt from J. S. Bach's second movement of *Orchestral Suite No. 3 in D major*, BWV 1068, the well-known *Air*. The input anechoic signals are the first 6 bars played by a first violin, a second violin, a viola and a cello. [14].

The orchestra is simulated in a room model of the Concertgebouw concert hall in Amsterdam, which has an average of 2.4 s Reverberation Time at mid frequencies. The *European* seating arrangement is followed, as suggested by Mayer in [13]. In this arrangement the typical number of instruments for our ensemble would be 12 first violins, 10 second violins, 8 violas and 6 cellos. However, in order to compare the colouration of the original quartet with the colouration of the full orchestra as in a A-B test, we choose to multiply each instrument by the same number, i.e. 12 times.

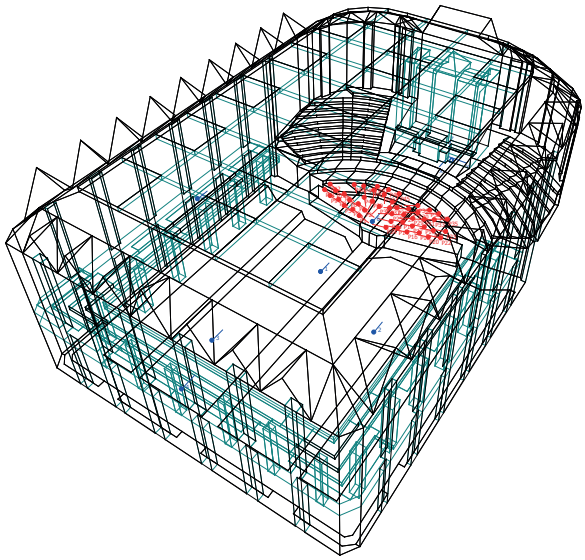

Fig. 9. The Concertgebouw model with the full string orchestra, conisting of 14 1st Violins, 12 2nd Violins, 10 Violas and 8 Cellos.

All four anechoic signals of a first violin, a second violin, a viola and a cello are processed in the chain of audio effects using the *Chorus*, *Spectrumizer* and *Amplitude Modulation* effects. The output files are then assigned to all sources of Fig. 10 and converted to auralisations using the process illustrated in Fig. 2.

The parameters for the *Chorus effect* are set as follows:

- Standard deviation for random delays: 40 ms.
- Random depth up to: 5 %.
- Random frequency up to: 3 Hz.

The parameters for the *Spectrumizer* effect are the following:

- Spectral random fluctuations: 90 %.
- Maximum number of random points per octave: 3.
- Random spectrum length in points: 256.

Finally, the parameters for the *Amplitude Modulation* effect are set as follows:

- Random gain change: 50% to 70%.
- Random period: 0.5 to 2 s.

The audio results are available on the Odeon's online archive [15]. Together with the results from the chorus effect, an auralisation of the quartet with four musicians playing in the

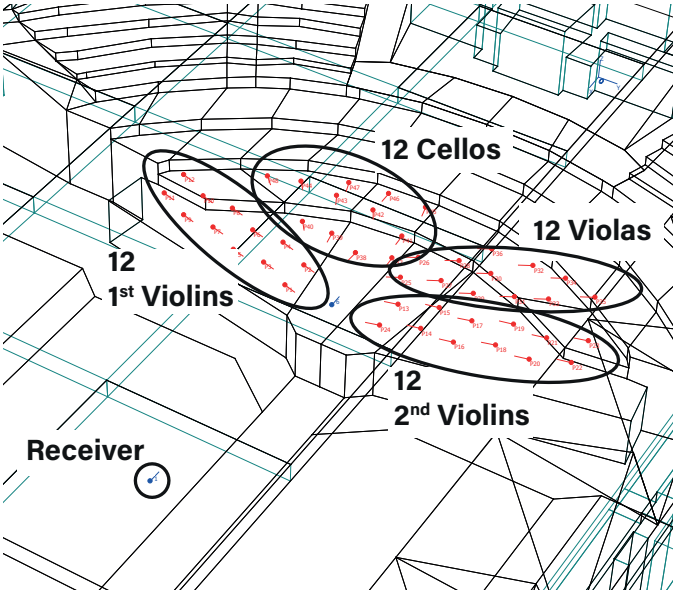

Fig. 10. The Concertgebouw model with the full string orchestra, conisting of 12 1st Violins, 12 2nd Violins, 12 Violas and 12 Cellos.

concert hall is available as a reference. The overall impression of the full orchestra simulation is a rich orchestral sound that seems to be balanced well towards low frequencies, with the cellos dominating. Although the configuration of an equal number of strings provides a good A-to-B comparison with the colouration balance of the original quartet, in a realistic setup there would be far less violas and cellos. As a next step, we perform the same auralisation with the typical modern orchestra arrangement of 12 first violins, 10 second violins, 8 violas and 6 cellos, as mentioned in the beginning of the section. Therefore, this time we deactivate part of the strings in each section. The result seems more balanced between low and high frequencies in this case. It should be noted that during the Baroque period (when Bach made his compositions) the orchestras were even smaller - therefore it would be valid to argue that using even less instruments in each section would be more realistic.

It should be noted that the quartet recordings used so far have been performed by amateur musicians. This might have several negative effects when a single recording is converted to a chorus in the chain of effects: any mistakes in pitch and timing are multiplied and exaggerated, because they happen at similar times. On the contrary, real musicians would make mistakes (if any) at different times, and the overall result would sound smoother. Another remark is that typically anechoic orchestra recordings - like the ones in the present example - are made using a single microphone. Therefore only part of the instrument sound is included. This might not be audible in the listening of the quartet alone, but it seems to be a reason why the chorus examples sound a bit 'narrow' and not as 'full' as we would expect in real life. Vingeant *et al.* [6] have experimented with multiple recording microphones for each musical instrument (the multi-channel multi-source technique). During auralisations, each instrument can be represented by several radiation sources, providing a richer result.

As a last example, we attempt to convert an anechoic file from a professional performance into a chorus. We apply the same settings for the chain of effects as for the previous example to generate a section of 12 violins. This time the file is an excerpt from the first violin of Mozart's 40^{th} Symphony. The recordings were performed at the Technical University of Denmark [6]. As it can be heard in the online examples, the chorus effect is less rich for the professional musician. Using the same effect settings as above the audio outputs sound still too similar. We try to increase the parameters of the different effects a bit and re-apply the audio process.

For this example, the parameters for the *Chorus effect* are set as follows:

- Standard deviation for random delays: 40 ms.
- Random depth up to: 7 %.
- Random frequency up to: 3 Hz.

The parameters for the *Spectrumizer* effect are the following:

- Spectral random fluctuations: 90 %.
- Max number of random points per octave: 3.
- Random spectrum length in points: 256.

Finally, the parameters for the *Amplitude Modulation* effect are set as follows:

- Random gain change: 60% to 80%.
- Random period: 0.5 to 2 s.

These last settings seem to have a more dramatic effect on the excerpt from the professional player. This shows that apparently there is not a single optimal setting for all types of recordings and musical instruments.

V. Application 2: Cafeteria Example

In this application we simulate a crowd inside the *cafeteria of building 381 at DTU Science Park*, Kgs. Lyngby, Denmark. The space has an average reverberation time equal to approximately 0.4 s at the mid frequencies. The corresponding ODEON model is shown in Fig. 11. A total of 31 sources are used, with one of them being the source that radiates a speech signal in focus at a time, while the rest represents crowd noise. We will examine two cases: the first with the source in focus being on the left of a fixed receiver and the second with the source in focus being in front of the same receiver. Both cases represent a person speaking next to a receiver in a situation where the cocktail-party effect can be assumed [16].

First we create two sets of populations: one with 15 variations of single male speech signal and one with 15 variations of a single female speech signal. The male speech signal is a 10 s long excerpt from the 'Definition of Decibels' (in English) written and performed by Mario Alfredo M. Sandoval, while the female speech signal is a 12.5 s long excerpt from 'Aninha e suas pedras' by Cora Coralina (in Brazilian Portuguese), performed by Caroline Gaudeoso.

Both files are processed in the chain of audio effects (Fig. 3) using *Phase Vocoder* and *Change Duration* and *Amplitude Modulation*, for a final duration of **1 min**. Even though the

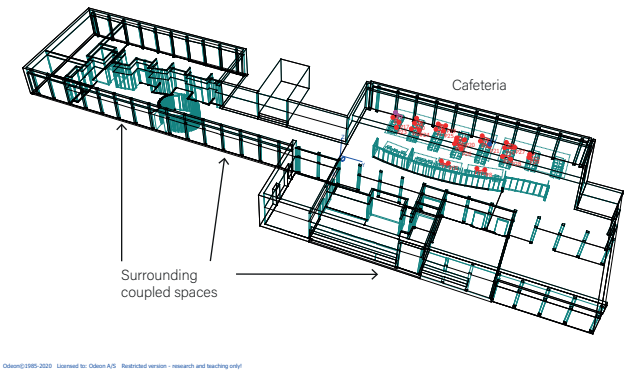

Fig. 11. The model of the cafeteria at DTU Science Park with 30 crowd sources, one receiver and an extra source (P21) next to the receiver.

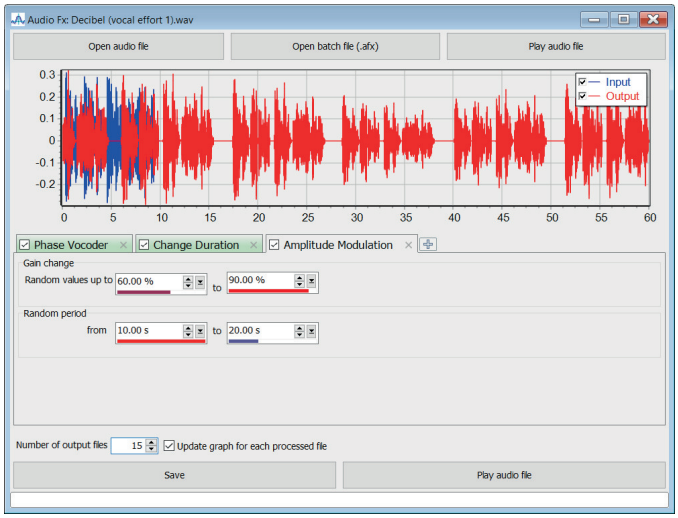

Fig. 12. The chain of audio effects used to generate a total of 30 output files (15 male variations, 15 female variations) for the crowd of the cafeteria example.

original length of the two input signals is different, all output files have the same fixed duration, according to the process that was illustrated in Section III-A.

The parameters for the *Phase Vocoder* effect are set as follows:

- Random values for Speed factor: 0.66 to 1.33.
- Random values for Pitch change [semitones]: -3.00 to 3.00

The parameters for the *Change Duration* effect are the following:

- Final duration: 1 min and 0 s.
- Max silence between repetitions: 3 s.
- Fade in/out: 0.2 s.

Finally, the parameters for the *Amplitude Modulation* effect are set as follows:

- Random gain change: 60% to 90%.
- Random period: 10 to 20 s.

A screenshot of the chain of effects with the parameters used here is shown in Fig. 12. It can be seen that the output signal (in red colour) has a new duration of 1 min, while the overall amplitude is reduced between about 30 and 45 s, corresponding to the slow random period between 10 and 20 s, chosen in this case.

All sources in the model use the **BB93_RAISED_NATURAL.SO8** [17] directivity file, which corresponds to a human speaking with raised voice. Each file from the crowd population is assigned to a source in the model using the *Auralisation Mixer* and the *Multi-source auralisation expert*. Before making the assignment, the order of the 15 male and 15 female variations are shuffled so that they are evenly distributed among the sources in space. The shuffling is currently performed manually. This process makes it apparent that there is no specific source assigned to a specific file, other than that sufficient randomness is maintained.

In Fig. 13, the location and direction of the receiver and the two source positions in focus are shown. Initially auralisations are performed using an average-person Head Related Transfer Function (HRTF) in ODEON. This provides an adequate binaural listening impression.

First, we assign the speech signal in focus on source **P21**, which is to the left side of the receiver in a distance of 0.5 m. The signal is an excerpt from the literary folktale *The Emperor's New Clothes* by Hans Christian Andersen. The excerpt is performed by Claus Lynge Christensen. The audio results are available on the Odeon's online archive [18]. It can be heard that the signal from source **P21** is extremely clear on top of the crowd sounds by the left ear.

Second, we assign the speech signal in focus on source **P13**, which is in front of the receiver, while source **P21** is assigned one of the crowd signals from the audio effects population. In the beginning, source **P13** is at approximately 1 m from the receiver. This results to a much lower level comparing to source **P21**, which is at 0.5 m. The speech signal in focus (*The Emperor's New Clothes*) is almost inaudible. Assuming that in a real crowd situation people tend to approach each other to facilitate their communication, we bring **P13** gradually closer to the receiver. First at 0.75 m and then at 0.5 m. The signal becomes more and more audible, but is not as equally loud as in the case of source **P21**, even when the distance from the receiver is exactly the same (0.5 m). The reason for that discrepancy is the use of HRTF, which represents our directional hearing mechanism. Indeed, source **P13** is located in front of the receiver so the SPL is almost the same as if the receiver point was in free-field. However, when sound arrives to the listener from the side direction, which is the case for source **P21**, the SPL at the nearest ear is approximately 6 dB higher than in free-field, as shown in Fig. 9 and Fig. 14 of [19].

This analysis is confirmed when we set the HRTF to a *unity* function in ODEON. Such a setting basically bypasses any ear and spatial-related attributes in the auralisation. In this case, the signals from both sources **P21** and **P13** sound almost

equally loud.

Simulating the crowd using speech signals created according to the process above sounds quite realistic. It is not obvious that it is the same signals that are repeated multiple times and the auralisation does give the impression that the receiver is surrounded by speaking people (crowd). At the same time, depending on the distance between the source and the receiver the main speech signal might sound more or less masked by the crowd. After careful listening, some repetitions among the voices become audible, but this artefact can be mitigated with the use of one or a few more different input files.

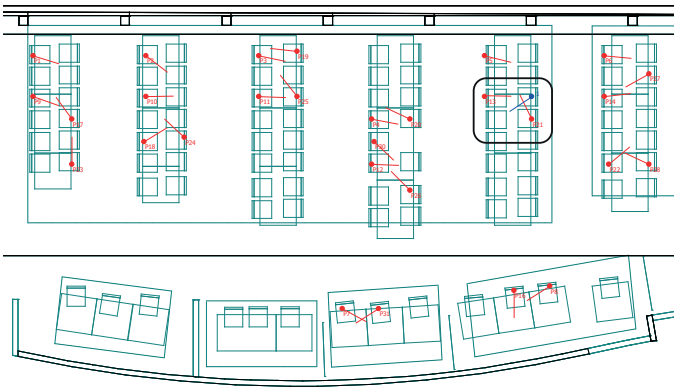

Fig. 13. Detail of Fig. 11 with 31 sources and one receiver. Two simulations are examined with one of the sources (**P13** and **P21**) next to the receiver treated as the speech in focus. Source **P13** is gradually placed closer to the receiver, following the distances 1.0 m, 0.75 m and 0.5 m.

VI. Discussion

The tools described above offer the possibility to convert single audio files into populations of files for realistic auralisations. The process is largely automatic, which facilitates the creation of complex auditory scenes in ODEON.

In the string section example, the chorus effect is used together with the Spectrumizer and the Amplitude Modulation effects to synthesize a group of strings out of individual inputs. The largest advantage of the method is the creation of realistic variations of pitch and phase between the strings in the population, which lead to a convincing impression of ensemble in the orchestra. However, several other attributes present in a real string section are not included in the current method and could be incorporated in future updates of the ODEON's chorus effect. An example is the vibrato effect on string instruments. Vibrato is a regular slight change in pitch, caused by the musicians intentionally to make the performance more emotional. Each musician introduces his/her own vibrato, at different speeds, degrees and times during the performance.

As for the cafeteria example, the chain of employed effects makes it possible to create sufficiently varying audio files to represent a crowd, from only two different speech signals – one male and one female. The series of effects applied, introduces changes in pitch, speed, phase (time shift), as well as amplitude (gain). However, the amount of variation between the speech signals can still be limited. Depending on the location of the receiver relatively to the sources, repetitions of the speech signal are still audible. If only one input speech signal had been used, these repetitions would be even more prominent. Moreover, all files created from a given gender sound from the same gender, which is why both a male and a female voice were used as input for a more realistic result. Another interesting observation is that many of the manipulated crowd files sound rather artificial, when listening to them separately. Four random files are available for listening on the online archive [18]. Despite this artificial impression, the result of the mixed files still sounds realistic enough for a crowd auralisation. Although the creation of the files is automatic, the choice of the parameters for the audio effects is still left to the user. It can be challenging to properly select them. This has been illustrated in the string section example, in which a sense of harmony between the violins should also be preserved. In both examples, the parameters were determined subjectively by the authors. However, as part of future work, a series of listening experiments could be conducted to derive these parameters in a more objective manner.

Finally, due to the nature of the ODEON calculations, the auralisations assume fixed source and receiver positions. Investigating the effect of movement, particuarly when modeling crowds, could also be the subject of future research.

VII. Conclusion

In this paper the importance of multi-source auralisations has been addressed as a useful tool for exploring complex simulated soundscape scenarios, both for research and for consultancy purposes. Creating multi-source auralisations requires a large number of anechoic signals to be convolved with the sources involved in the simulation. Obtaining many anechoic files can be a demanding task, but for several applications this can be accomplished by creating less correlated variations of the original signal. Therefore only one available anechoic signal can be enough to create a population of different anechoic signals.

In the past, such a process was exclusively manual, being extremely time consuming and leaving the risk of biased results. Nowadays, with the computation power available, creating large sets of anechoic signals can be done automatically, with sufficient randomness included by the varying attributes of the signals.

We explored two examples: a string orchestra and a crowd in a cafeteria. In the first example, a recording of a quartet was transformed into an auralisation of a full orchestra using mainly the Chorus Effect in ODEON. In the second example, a male and a female speech signal were transformed into 30 sources in a crowd, using pitch, speed and repetition, together with slowly-varying random amplitude modulation. In both cases, the results are rather satisfying. Using the appropriate settings in the different effects involved, it has been shown that artificial multiplications of single files can be used successfully in auralisation scenarios, despite some minor artefacts introduced in the process.

REFERENCES

[1] Odeon A/S, ODEON User's Manual, Denmark, 2020.

[2] M. Vorländer, Auralization, fundamentals of Acoustics, Modelling, Simulation, Algorithms and Acoustic Virtual Reality, 1st edition. Springer, 2008, pp. 3 and pp. 141-142.

[3] ISO 12913-1:2014, "Acoustics — Soundscape — Part 1: Definition and conceptual framework," 2014.

[4] ISO/TS 12913-2:2018, "Acoustics — Soundscape — Part 2: Data collection and reporting requirements," 2018.

[5] Odeon A/S, Auralisation Tutorials, 2021.

[6] M. C. Vigeant, L. M. Wang and J. H. Rindel, "Investigations of orchestra auralizations using the multi-channel multi-source auralization technique," Acta Acust. united Ac., vol. 94, pp. 866-882, 2008.

[7] J. Pätynen, V. Pulkki and T. Lokki, "Anechoic recording system for symphony orchestra," Acta Acust. united Ac., vol. 94, nr. 6, pp. 856-865, November/December 2008.

[8] J. H. Rindel and C. L. Christensen, "Auralisation of concert halls using multi-source representation of a symphony orchestra," Proceedings of the Institute of Acoustics, Vol. 30, pp. 333 - 338, 2008.

[9] F. E. Toole, Sound Reproduction: Loudspeakers and Rooms. Focal Press, 2008.

[10] T. Lokki, "How many point sources is needed to represent strings in auralization?," International Symposium on Room Acoustics, Sevilla, September 2007.

[11] U. Zölzer, DAFX: Digital Audio Effects. United Kingdom, John Wiley & Sons, 2011.

[12] J. Pätynen, S. Tervo and T. Lokki, "Simulation of the violin section sound based on the analysis of orchestra performance," IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, NY, October 2011.

[13] J. Meyer, Acoustics and the performance of music, 5th edition. Springer, 2010, pp. 263 - 265.

[14] D. Thery and B. Katz, "Anechoic audio and 3D-video content database of small ensemble performances for virtual concerts," International Congress on Acoustics, Aachen, 2019.

[15] Odeon A/S, Bach's Air suite in Concertgebouw, Denmark, 2021.

[16] E. C. Cherry, "Some Experiments on the Recognition of Speech, with One and with Two Ears," J. Acoust. Soc. Am., vol. 25, pp. 975-979, 1953.

[17] Department for Education (DfE), Acoustic design of schools - performance standards, United Kingdom, 2015.

[18] Odeon A/S, The DTU cafeteria crowd auralisation example, Denmark, 2021.

[19] H. Møller, M. Friis Sørensen, D. Hammershøi and C. Boje Jensen, "Head-Related Transfer Functions of Human Subjects," J. Audio Eng. Soc., vol. 43, no. 5, pp. 300 - 321, 1995.