BUNKERVIK SPATIAL REVERB DEMO

Michele Ducceschi and Craig J. Webb

Department of Industrial Engineering
University of Bologna
Bologna, Italy
michele.ducceschi@unibo.it

ABSTRACT

This paper accompanies a demonstration of a real-time audio plug-in for a dynamic spatial reverb. The reverb is based on acoustic measurements of the Bunkervik creative arts space in Brescia, Italy. A modal synthesis reverberation engine was created based on measured impulse responses from three locations in the tunnel. Using a common set of modal frequencies the position of the receiver can be dynamically moved through the space by interpolating between data sets of residue weights and FIR filter taps. The audio plug-in also allows real-time manipulation of the high-frequency content, damping, and microphone rotation, all of which can be modulated using two LFOs.

1. INTRODUCTION

The acoustic measurements were conducted in November 2025 within the Bunkervik tunnel, a former WWII air-raid shelter in Brescia, Italy, comprising a vaulted gallery approximately 30 metres in length with a roughly uniform semi-circular cross-section. Two views of the gallery are reproduced in Fig. 1. The geometry is well approximated as one-dimensional: a listener walking along the central axis experiences a smooth, monotonic evolution of the reverberant signature.

Impulse responses were acquired using the exponential sine sweep (ESS) method [1], with a Genelec studio monitor serving as the excitation source and a Beyerdynamic free-field omnidirectional condenser microphone as the receiver, with audio recorded at 48 kHz and 24-bit resolution via a portable USB audio interface. To compensate for the inherent directionality of the loudspeaker, the source was successively oriented along six cardinal directions (Up, Down, Front, Back, Left, Right) at constant gain and signal level, and the resulting sweeps were time-aligned, trimmed to a common length, and summed to approximate an omnidirectional source characteristic.

The receiver was positioned at three locations along the central axis of the tunnel at distances of approximately 8 m, 10 m, and 25 m from the source, yielding three independent impulse responses spanning the perceptually relevant range of the space. The geometry is summarised schematically in Fig. 2.

Each combined sweep was converted to an impulse response via Wiener/Tikhonov-regularised frequency-domain deconvolution, with a regularisation factor applied to ensure numerical stability at the spectral extremes of the sweep.

(a) Longitudinal view. **(b)** Acquisition rig.

Figure 1: Bunkervik tunnel - the vaulted gallery is approximately 30 m long, with cross-section roughly uniform along the longitudinal axis. The Genelec studio monitor was rotated through six cardinal orientations to approximate an omnidirectional source.

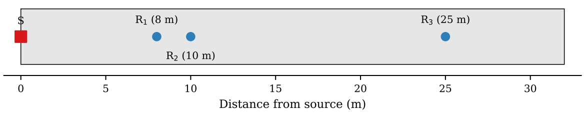

Figure 2: Schematic of the source–receiver geometry. A single Genelec source S at one end of the tunnel was rotated through six orientations; the omnidirectional receiver was placed in turn at three positions R_1, R_2, R_3 along the longitudinal axis.

2. DATA PROCESSING

The data processing stage translates the impulse responses into a modal synthesis reverberation engine [2] that uses 3 data sets of equally sized arrays, thereby enabling direct and efficient interpolation between them. The processing is loosely based on the methodology described in [3]. The first stage of the processing pipeline involves independently decomposing each impulse response into a modal representation, in which the acoustic transfer function is expressed as a superposition of damped sinusoidal components, each characterised by a frequency, a damping coefficient, and a complex residue. Modal frequencies are identified by peak-picking on the magnitude spectrum within a set of overlapping sub-bands, and per-mode damping values are derived from band-wise reverberation time estimates, with a global fallback applied in bands where reliable decay estimates cannot be obtained.

Once the poles are established, the corresponding complex residues are determined by banded least-squares fitting against the measured spectrum [4], supplemented by a 256-tap FIR correction filter designed to absorb any residual spectral colouration in the early portion of the impulse response.

A direct consequence of performing these fits independently at each position is that the same physical acoustic mode may be identified at slightly different frequencies and damping values across the three pole sets, rendering straightforward residue interpolation ill-defined. To resolve this, a common modal basis is constructed by concatenating the three pole sets and clustering them along the frequency axis using a frequency-dependent tolerance, with candidate poles within each cluster merged into a single representative pole via a residue-magnitude-weighted average.

This procedure yields a unified set of 6,300 poles spanning the band from 70 Hz to 12 kHz. Per-position residues and FIR correction filters are subsequently re-fitted against this shared pole set through a joint alternating least-squares optimisation, producing a representation in which residues at all three positions are anchored to identical poles and can therefore be meaningfully interpolated. The accuracy of the resulting fit is confirmed by time-domain normalised RMSE values below 3.5% at all three positions, with the combined IIR/FIR optimisation reducing error by a factor of two to three relative to the modal-only baseline.

3. AUDIO ENGINE

The audio engine consists of per-sample processing across two independent channels for stereo output. Separate data sets exist for each channel, with core data stored in a header file and loaded into class members at setup. The arrays of modal frequencies and residue weights contain 6,300 elements each, whilst the small FIR correction filters contain 256 elements. The modal reverberation scheme is supplemented by a fractional delay line to give the correct time delay for sound to travel from the source to the receiver, which can be up to 30 meters away.

The position of this microphone receiver can be dynamically changed during runtime, allowing the user to 'walk-through' the space. This movement needs to be smoothed in order to reduce audible artifacts. Whilst a simple linear interpolation produces high-quality results for the data sets of residue weights and FIR filter taps, a cubic lagrange interpolation is used for the fractional delay. The re-calculation of the state-size arrays of residue weights does not need to be performed at each time-step, in practice every eighth step gives smooth output at minimal computational overhead. This is an important consideration, as an LFO may be applied to the microphone position.

In addition to the dynamic receiver, we also implement manipulation of the high-frequency components by stretching the modal frequencies and adjusting the weights above a user-controlled cut-off frequency. As the original data only contains modal frequencies up to 12 kHz, this procedure allows higher-frequency content to be rendered, up to 20 kHz. This operates as a 'brilliance' control, which is labeled as 'Aura' in the plug-in itself. Both the cut-off and maximum frequencies are variable in real-time, as is the amount of boost given to the residue weights. The effect can be bypassed in the plug-in through the disable button, which resets the system to it's original data state.

The complete process method of the engine requires six stages, as shown in Figure 3. First the microphone position is smoothed to approach its target value, and then if a recalculation of the weights and FIR taps is required it is performed on an eight sample clock.

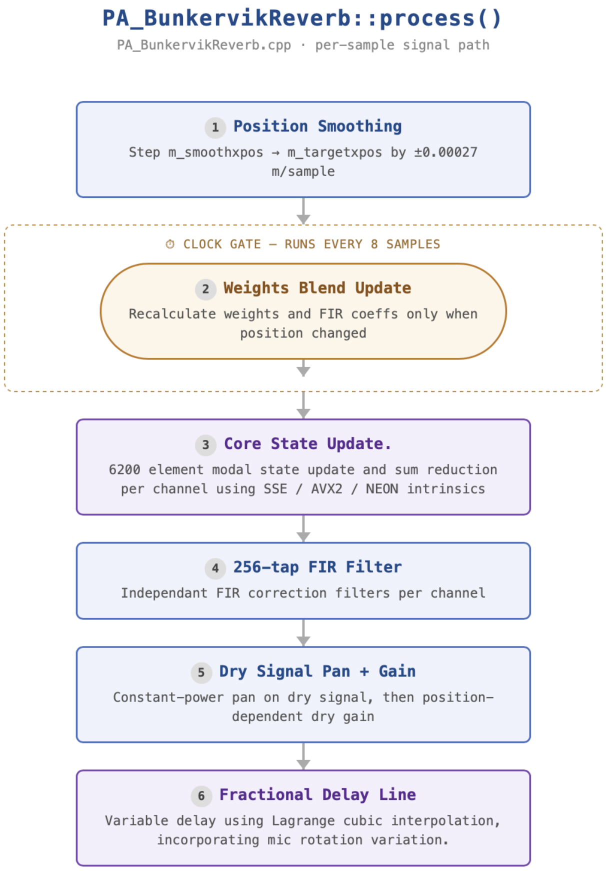

Figure 3: Flow chart of the process() method of the audio engine.

Having set the correct data for our arrays we can then perform the core state update and sum reduction to compute the initial output for each channel of the modal reverb scheme (stage 3). This stage takes around 90% of the computational effort, so manual vector intrinsics were employed to optimize this section (SSE/AVX/NEON) on both macOS and Windows platforms.

The two output samples are then processed through the FIR filters (stage 4) before mixing and panning is applied. The final stage is to send the samples through the fractional delay lines, using small variations in their lengths in combination with the panning to give a microphone rotation effect. The only additional processing required is to update the damping and aura parameters, which manipulate the high-frequency modes and weights. This is applied once per-buffer and so has minimal effect on the overall computational costs. Initial testing on Apple Silicon M2 Pro and M4 Max showed a single-core CPU usage of < 25%. A silence check was also included, to cut processing of the engine when the output amplitude drops to less than -60dB.

4. USER INTERFACE

The audio plug-in was created using the JUCE framework, with a user interface built on live-drawn vector graphics as shown in Figure 4. The top section contains a visualization of the three Aura

Figure 4: Full user interface of the audio plug-in, showing aura and spatial controls, and modulation matrix.

parameters for display purposes only. The remainder of the UI is split into 4 sections of controls.

- *Aura controls* - Two large knobs and a slider control a high-frequency modal warping of the pole frequencies. Cutoff and Max Freq define the frequency range, whilst Boost scales the slope. This allows control of brilliance without altering the modal density or the spatial residue structure.
- *Spatial controls* - The core parameters that control the spatial reverb. Microphone Position from 0 - 30m, Microphone Rotation +/-90 degrees, and Damping. There is also a visualization of these on the left.
- *Modulation section* - This contains the 2 LFOs and a modulation matrix that allows a choice of 6 targets to be modified across 3 rows. Rates can be sync'd to the host, and there are sine, triangle and square waveforms available.
- *Output controls* - The final reverb Gain and Dry/Wet mix, with level meters. There is also a Bass parameter for controlling the low frequency content.

The rotary controls that are available for modulation have outer rings that animate according to their LFO-modified values. Other animation effects are the Aura visualizer and the Spatial diagram element. Each of these are drawn and updated as individual opaque components to minimize the CPU cost of the animation. The plug-in window is also fully re-sizable via affine transformation of the outer component window.

5. CONCLUSIONS

A spatial extension of the modal-reverberation framework has been demonstrated through acoustic measurement of a space and the development of a real-time audio plug-in that allows dynamic positioning of the output receiver. The core concept is the construction of a common modal basis, a single set of position-invariant poles against which per-position residues and FIR correction filters are fitted. This allows the output receiver to be dynamically positioned along the axis of the space. The engine also demonstrates real-time manipulation of high-frequency content using the Aura controls, stretching the modal frequencies and adjusting weights to allow control over brilliance and creative applications.

A direct extension of this work would be the use of data sets representing 2-dimensional or even 3-dimensional arrangements of recorded impulse responses. Interpolating through these larger data sets would allow a greater freedom of receiver position movement and reproduction of spatial presentation, achievable with comparable levels of computational cost.

The prototype plug-in is available in Audio Unit, VST3 and AAX formats for macOS, and VST3 and AAX formats for Windows. They can be obtained from the NEMUS wesite at: https://nemusproject.eu/resources/software-downloads/.

6. REFERENCES

[1] A. Farina, "Simultaneous measurement of impulse response and distortion with a swept-sine technique," in *Proceedings of the 108th AES Conv.*, Paris, France, 2000.

[2] J. S. Abel, S. Coffin, and K. Spratt, "A modal architecture for artificial reverberation with application to room acoustics modeling," in *Proceedings of 137th AES Conv.*, 2014.

[3] M. Ducceschi, C. Webb, and G. Fratoni, "Efficient synthesis of large room impulse responses in the modal domain," in *Proceedings of I3DA*, Bologna, Italy, Sept. 10–12, 2025.

[4] P. Guillaume, P. Verboven, S. Vanlanduit, H. V. der Auweraer, and B. Peeters, "A poly-reference implementation of the least-squares complex frequency-domain estimator," in *Proceedings of IMAC*, vol. 21, 2003.