

Reverb and Spatial Effects in the Context of Low-Latency Audio Streaming

Merten Schicht

Audio Communication Group

Technische Universität Berlin

Berlin, Germany

merten.schicht@campus.tu-berlin.de

Alexander Carôt

Department of computer science and languages

Anhalt University of Applied Sciences

Köthen, Germany

alexander.carot@hs-anhalt.de

Abstract—Network Music Performance (NMP) systems have established themselves as a workable means of distributed ensemble playing within tight latency budgets, but the streams they deliver to each participant remain dry and perceptually isolated, lacking the common room acoustic and the spatial source arrangement that contribute to the impression of joint musical presence. This paper presents an extension of the SoundJack low-latency audio streaming engine that integrates reverberation and spatial audio rendering into the receiver-side processing pipeline. The architecture exploits the perceptual asymmetry between latency-critical direct sound and latency-tolerant reverberant signal by running the dry and wet processing paths asynchronously at different time granularities, decoupled through a lock-free triple-buffer hand-off. Four complementary reverberation modes are integrated into the wet path, spanning a deliberate trade-off range between algorithmic room simulation, ambisonic and mono spatial room impulse response convolution, and direct binaural room impulse response rendering. The system is implemented in C++ as an extension of SoundJack, using the open-source Spatial Audio Framework (SAF) for the underlying spatial processing. A quantitative evaluation across three impulse-response lengths from 0.25 s to 2.8 s shows that all four modes meet the 10.67 ms wet-thread deadline on commodity hardware (Apple M1 Pro), with substantial headroom even at concert-hall-scale impulse-response lengths, while the audio callback itself remains unaffected by the reverb workload.

Index Terms—distributed music, reverb, room simulation, spatial audio, low-latency, head tracking

I. INTRODUCTION AND MOTIVATION

Network Music Performance (NMP) refers to the joint, real-time musical performance of spatially separated musicians over the internet. For rhythmically coordinated ensemble interaction to remain viable, the one-way transmission latency must not exceed the so-called Ensemble Performance Threshold (EPT) of approximately 25 ms [1], although in practice the acceptable range varies between roughly 15 ms and 50 ms depending on tempo, instrumentation, and musical style [2]. Within this budget, our SoundJack engine [3], JackTrip [4], Jamulus¹, and Sonobus² have all demonstrated that musical interaction across the network can be conducted in a manner consistent with conventional rehearsal conditions; a broader overview of the NMP landscape is given in [5].

Current NMP systems in productive use focus almost exclusively on the faithful, low-latency delivery of the dry source signal. While the acoustic context in which the joint performance takes place has been addressed in a growing body of research — notably by Cairns [6], Hupke et al. [7], and Genovese [8] — these systems have generally been realised as separate, purpose-built prototypes rather than integrated into an established NMP platform. In the tools that musicians actually use, the streams that reach each participant remain dry and perceptually isolated, lacking the common room acoustic that would otherwise bind the individual sources together into a coherent ensemble image. In conventional ensemble performance, the shared room acoustic plays a central perceptual role. Reflections and reverberation act as an acoustic “glue” that has been standard practice in studio mixing for decades, and the spatial positioning of co-performers around the listener contributes substantially to the impression of joint musical presence [9]. The Realistic Interaction Approach we have advocated for SoundJack and related systems [1], in which musicians are intended to perform “as if in the same room”, is therefore only partly supported by the dry-stream paradigm of current NMP systems.

The integration of reverberation and spatial rendering into NMP is technically demanding. In our earlier work on the Fast-Music project [3], summarising the state of low-latency audio streaming achieved by SoundJack, we explicitly observed that under typical operating conditions there remains “*no additional headroom for further DSP-related calculations such as Ambisonics and further audio effects*”. Convolution with concert-hall-scale impulse responses easily exceeds 144,000 samples per channel at 48 kHz [10]; geometric room simulation through the image-source method [11] grows rapidly in cost with the reflection order; and spatial encoding through Ambisonics [12] multiplies the per-source cost by the number of Ambisonic channels. All of these operations share the same hard real-time deadline imposed by the audio callback, and a single missed deadline produces an audible discontinuity.

Several recent studies have approached related problems from different angles. Cairns [6] explored ambisonic audio for immersive NMP across a range of practical use cases and evaluated the perceptual effect of auralised reverberation time in this context [13]; Schörkhuber et al. [14] demonstrated

¹<https://jamulus.io>

²<https://sonobus.net>

zero-delay binaural rendering through minimum-phase FIR filters; Hupke et al. [7] realised an extended-reality room extension environment for NMP with separately rendered early reflections and diffuse reverberation; and Genovese [8] investigated auditory virtual environments for distributed performance with respect to acoustics and copresence. Comparative studies of existing spatial audio tooling in this domain have been provided by Tomasetti et al. [15], who report systematic latency measurements of spatial audio plugins, and by Müller et al. [16], who survey telemersive audio systems for online jamming. What appears not to have been published, however, is a systematic comparison of multiple complementary reverberation strategies — spanning algorithmic room simulation, ambisonic and mono Spatial Room Impulse Response (SRIR) convolution, and direct Binaural Room Impulse Response (BRIR) rendering — implemented within a single production NMP engine and evaluated together under identical hard real-time constraints. The present work contributes such a comparison.

The system presented here extends our SoundJack engine [3] with reverberation and spatial audio rendering. It is built around three design choices that together address the challenges identified above: receiver-side processing of the entire spatial scene, asynchronous handling of the direct and reverberant components at different time granularities to exploit their perceptual delay asymmetry [9], and four complementary reverb modes that span a deliberate trade-off range between computational cost, acoustic faithfulness, and output-format flexibility. The architecture is implemented as a C++ extension of the SoundJack engine and evaluated quantitatively in terms of its computational behaviour. The measurements show that all four modes meet the wet-thread deadline on commodity hardware, even in the demanding case of concert-hall-scale impulse responses.

The remainder of this paper is organised as follows. Section II characterises the technical and conceptual challenges of integrating reverberation and spatial audio into low-latency NMP. Section III presents the proposed concept with its receiver-side architecture, dry/wet decoupling, and four reverberation modes. Section IV reports the implementation details. Section V presents the quantitative evaluation and the proof-of-concept listening test. Section VI concludes with a discussion of directions for future work.

II. PROBLEM

As introduced in Section I, integrating reverberation and spatial audio rendering into the low-latency NMP context is technically demanding. The challenges fall into three distinct problem areas which we discuss in the following.

A. Processing load under hard real-time constraints

As outlined in Section I, network propagation, jitter compensation, and audio codec processing already consume most of the EPT budget in non-local connections, leaving only a marginal time budget for additional local digital signal processing.

Reverberation is among the most demanding operations to place into this remaining window when the goal extends beyond a generic, diffuse reverb tail. Lightweight algorithmic methods such as Schroeder reverberators [17] or Feedback Delay Networks (FDNs) [18] can produce plausible reverberation at low computational cost and can model simplified rectangular-room acoustics, but they do not capture the directional structure of a specific measured space. The techniques relevant to the present work — those that render a defined, directionally consistent room — fall into two computationally heavier categories. Geometric room simulation synthesises reflection patterns from a parametric room description; its cost scales with the number of modelled reflections, growing rapidly with reflection order and the number of sources within the virtual room. Convolution-based approaches, by contrast, render reverberation by filtering the input signal with a measured or synthesised impulse response [10]; their cost scales with the impulse-response length and the number of channels involved.

Even the comparatively modest case of a BRIR, which captures the acoustic transfer from a single source position to a listener’s two ears, already requires real-time convolution of a multi-second filter. Comprehensive measured datasets for such impulse responses are publicly available [19]. SRIRs, which preserve the directional structure of the sound field through an Ambisonics [12] representation, multiply this cost by the number of Ambisonic channels — given by $N_{\text{ch}} = (M + 1)^2$ for Ambisonic order M , e.g., 4 channels for first order, 9 for second, and 16 for third — and easily exceed 144,000 samples per channel at 48 kHz for a concert hall of three seconds reverberation time. Real-time processing of such filters requires partitioned FFT-based convolution [10], whose efficiency degrades as the partition size decreases, since FFT setup and overlap-add overhead are amortised across fewer samples. Low-latency callbacks require small block sizes — down to 64 or even 32 samples — which convolution algorithms handle least efficiently; the requirements are inherently conflicting.

Spatial rendering further increases the computational load. Each source must be encoded into an Ambisonics representation; the resulting scene must, depending on the output target, be decoded either to a loudspeaker array or to binaural via HRTF convolution per ear; and, if head tracking is included, the scene must be rotated according to listener orientation. All of these stages share the same callback deadline. The audio callback itself is hard real-time: a single missed deadline causes an audible discontinuity. Unlike typical benchmarking contexts, average-case performance is not meaningful — system viability is determined by worst-case behaviour.

Finally, NMP is inherently a multi-source scenario; ensembles of four to eight peers are typical, and orchestra-scale scenarios involving several dozen performers have been demonstrated using our server-based streaming infrastructure [3]. Reverberant and spatial processing cost typically grows with the number of sources — in most cases at least linearly, though certain operations such as Ambisonic scene rotation

are independent of source count — so processing budgets that appear adequate for a single source can become untenable for a full ensemble.

B. Wet/dry latency asymmetry

The direct and reverberant components of an acoustic signal differ in their sensitivity to additional delay. The direct sound governs ensemble synchrony; any latency added to this path counts directly against the EPT budget. Reverberant energy, by contrast, is perceptually tolerant of modest delay relative to the direct sound, a phenomenon grounded in the precedence effect and in the slowly evolving nature of late reverberation [9]. Treating both components within a single uniform processing pipeline forces the entire signal to share the worst-case latency of the most expensive operation, leaving exploitable perceptual headroom unused.

Exploiting this asymmetry requires separating the direct sound from the simulated room response, so that early reflections and late reverberation reach the listener only through the asynchronous reverb path while the direct path carries no room colouration. This is acoustically a simplification, since in a physical room both originate from the same propagation; its perceptual acceptability rests on the dominance of the direct signal in localisation cues and on the comparatively slow perceptual integration of room response.

C. Plausibility and coherence of the rendered space

For reverberation to function as the perceptual “glue” familiar from studio production, all peers must be perceived as inhabiting a common acoustic space — a requirement that aligns with the *Realistic Interaction Approach* we have advocated for SoundJack and related systems [1], in which musicians are intended to perform “as if in the same room”. Independent per-source processing without a coherent shared room model breaks this illusion. At the same time, position-dependent rendering introduces additional challenges: time-domain crossfading between impulse responses for different source positions produces audible artefacts when the spatial grid is too coarse, since direct and early-reflection components are phase-sensitive and their misaligned superposition becomes audible [20]. More principled solutions — decomposing each impulse response into direct, early, and late components and treating each separately, as investigated perceptually by Cooper et al. [21] — are computationally costly and introduce their own seams.

Optional head tracking introduces a further real-time control loop: scene rotation must follow listener orientation continuously without introducing zipper noise or abrupt discontinuities.

III. CONCEPT

The system presented here extends our SoundJack low-latency audio streaming engine [3] with reverberation and spatial audio rendering. Our concept is built around three design choices that together address the problem areas identified in Section II: all rendering occurs at the receiver, the

direct and reverberant components of the signal are processed asynchronously at different time granularities, and four complementary reverb-and-spatialisation modes are offered to span the trade-off between computational cost, acoustic faithfulness, and flexibility of output format. In this work we rely on the open-source Spatial Audio Framework (SAF) [22] for the underlying spatial processing modules. SAF is well-established in the spatial audio research community, implemented in C/C++, actively maintained, and integrates straightforwardly with SoundJack’s existing real-time architecture. The present paper focuses on establishing the architectural and computational feasibility of this integration; the impulse-response datasets used here serve as a first working set to demonstrate the approach, and their extension to finer spatial grids or additional room types is a matter of dataset selection rather than architectural change. Framework-specific details are deferred to Section IV.

A. Receiver-side processing

A first architectural decision concerns where reverberation and spatial rendering occur. The sender transmits a dry mono PCM stream; all spatial and reverberant processing takes place at the receiving end. This receiver-side approach has three direct consequences. First, the transmitted payload remains a single mono channel, so the bandwidth behaviour of the underlying SoundJack stream is unaffected by the addition of reverb. Second, each listener can render the acoustic scene to suit their own monitoring situation, whether headphones, a loudspeaker array, or further downstream processing. Third, the shared room model is instantiated once per listener rather than once per sender, allowing individualised perspectives within a common acoustic space.

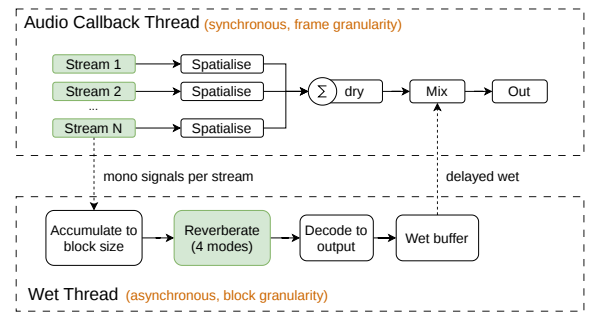


Fig. 1. Dry/wet thread architecture. The dry path is spatialised per-stream within the audio callback, preserving ensemble-critical latency. The wet path aggregates the mono signals of all N active streams, accumulates them to the convolution block size, and processes reverberation asynchronously in a dedicated thread before recombination with the dry mix at the output stage.

B. Dry/wet thread architecture

The receiver-side processing pipeline is structured to exploit the wet/dry latency asymmetry described in Section II-B. Two processing paths are maintained in parallel and recombined at the output stage, as depicted in Fig. 1.

The *dry path* runs synchronously inside the audio callback. Each incoming stream is spatialised individually — placed at a defined direction relative to the listener via Ambisonics encoding and binaural (or multi-channel) decoding — and the resulting per-source contributions are summed into a dry mix. Because this path operates at the granularity of the audio callback’s frame size, the reverb extension itself introduces no buffering; the direct sound reaches the listener with the latency of the existing spatialisation stage, preserving the ensemble-critical timing that counts directly against the EPT budget.

The *wet path* runs asynchronously in a dedicated background thread. The mono signals of all N active streams are gathered at each callback and accumulated until a larger convolution block size is reached, at which point the chosen reverberation mode processes them. The resulting wet signal is then handed back to the audio callback and mixed with the dry signal at the output stage. By the time the audio callback consumes a wet block, the corresponding dry signal has already been delivered: the wet signal lags behind the dry signal by the accumulation period plus one full convolution block, amounting to approximately 21 ms in Mode D and 37 ms in the Ambisonic modes, where the binaural decoder adds a further filter-bank delay. This offset is expected to remain within the perceptual tolerance of reverberant energy.

This delay is not merely an artefact to be tolerated: in a physical room, the direct sound reaches the listener first and the first reflection arrives only after the initial time delay gap, which in concert halls typically falls in the range of 10 ms to 30 ms. In Mode D this offset falls within that range and can be understood as an approximate model of it; in the Ambisonic modes it exceeds it and acts more as a pre-delay. It should be noted, however, that the measured impulse responses used in Modes B, C, and D already contain their own initial time delay gap, so the two add up; a decomposition that removes the direct component from the wet path, as discussed in Section VI, would resolve this and make the modelling explicit.

This decoupling has two main benefits. The audio callback is relieved of the heavy convolution and room-simulation work, so its worst-case execution time remains short and predictable. The wet thread, in turn, is freed from the small block sizes imposed by the low-latency callback and can be configured for the requirements of the chosen reverberation mode.

C. Four complementary reverberation modes

For the sake of completeness and comparability, our concept considers four reverberation modes altogether, to be integrated into the wet path as illustrated in Fig. 2. They are deliberately chosen to span distinct trade-offs in cost, realism, and flexibility, allowing the same system to serve different use cases and to support direct comparison between rendering approaches.

Lightweight delay-line reverberation such as Schroeder reverberators or FDNs would provide acoustic “glue” at far lower cost, but cannot represent the directional structure of a specific measured space or render localisable contributions

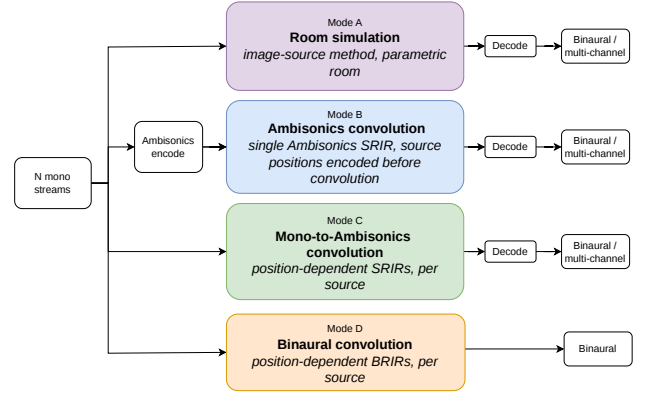


Fig. 2. Four reverberation modes integrated into the wet thread. Mode B encodes the N aggregated streams into a shared Ambisonics scene before convolution, whereas Modes A, C, and D operate on the source signals directly. Modes A–C produce an Ambisonics scene decoded to binaural or multi-channel output; Mode D folds room response and binaural rendering into a single position-dependent convolution and is therefore restricted to binaural output.

of individual sources within it. Three of our four modes are therefore organised around an Ambisonics intermediate representation; the fourth (Mode D) takes a different route.

Mode A — algorithmic room simulation. The acoustic space is described parametrically by its dimensions, source and listener positions, and per-wall absorption coefficients. Reflections are synthesised through an image-source model [11], producing an Ambisonic scene without requiring any measured data. Compared to measured SRIRs, this representation is a substantial simplification: only rectangular room geometries can be modelled, the reflection density is lower, and the reverberation length is bounded by the maximum modelled reflection order (10 in the present configuration; see Section V-A). Mode A is the only convolution-free option of the four and is well suited to exploration of arbitrary virtual rectangular rooms.

Mode B — ambisonic SRIR convolution. Each source is first Ambisonic-encoded at its assigned direction, and the individual encoded contributions are summed into a single shared B-format scene. This scene is then convolved with a measured ambisonic SRIR representing the acoustic response of a fixed listening position within the target space. Dynamic source motion is handled entirely by the Ambisonic encoder; the SRIR itself remains static. Because the SRIR is positional with respect to the listener but not with respect to individual sources, all sources are placed within the same shared virtual room, with their directional structure preserved through the B-format representation.

Mode C — mono SRIR convolution. A measured SRIR is treated as a one-to-many filter that maps each mono source directly into Ambisonics. Compared to Mode B, this approach folds the Ambisonics encoding into a single matrix convolution, reducing computational cost while still producing a directionally structured response. In Mode C, a set of

SRIRs measured at different points within the same space is selected per source, enabling distinct sources to occupy distinct locations within a coherent shared room.

Mode D — position-dependent BRIR rendering. Mode D bypasses the Ambisonics intermediate representation entirely. A BRIR is selected from a database indexed by source azimuth and convolved directly with the mono input to produce the final binaural stereo output. This combines room response and binaural rendering into a single operation, at the cost of fixing the output format to two-channel binaural.

Across all three convolution-based modes (B, C, D), the measured impulse responses are used in their full form, without prior decomposition into direct, early-reflection, and late-reverberation components. This preserves the complete measured signature of the target space but introduces two known simplifications. First, the direct-sound component of each impulse response is reproduced by the wet path in addition to the direct sound rendered by the dry path, producing a mild duplication of the direct signal. Second, the early reflections encoded in each impulse response reflect the acoustic geometry seen from the original measurement position. In Mode B this means that early reflections are decoupled from the encoded source directions; in Modes C and D, where impulse responses are positional per source, the early reflections belong to the correct source position but their measurement-specific structure contributes to the phase-sensitive crossfading artefacts noted in Section II-C. A more principled treatment based on Direct/Early/Late decomposition is discussed as future work in Section VI.

A feature shared by Modes C and D is that the spatial resolution of the impulse-response dataset is typically coarser than the resolution at which source direction can be specified in the Ambisonic encoder. In Mode C, this mismatch is most pronounced: the SRIR sets used in this work provide only three to four measurement positions per room, while the encoder places each source at a continuous azimuth and elevation. In Mode D, the situation is considerably more favourable in the horizontal plane: SOFA-based BRIR datasets typically cover a fine azimuthal grid, so the azimuthal resolution mismatch is comparatively small. The current implementation uses only the horizontal subset of the available BRIRs and does not yet exploit the elevation samples that many SOFA datasets provide. In both modes, source motion is realised by switching to the nearest available impulse response, with equal-power crossfading between adjacent choices to avoid audible clicks while accepting the comb-filter limitation discussed in Section II-C. The rendering fidelity of both modes is therefore bounded by the density of the effectively used impulse-response subset; extending Mode C datasets to a finer spatial grid, and extending Mode D to full-sphere lookup including elevation, are natural directions for improvement and are discussed in Section VI. Optional head tracking drives a continuous rotation of the rendered scene at the binaural output stage.

IV. IMPLEMENTATION

The system is implemented in C++ as an extension of the SoundJack audio engine [3]. Spatial audio processing is delegated to the open-source Spatial Audio Framework [22], which provides the Ambisonics encoders and decoders, the image-source room simulator, and the partitioned-convolution modules used in this work. Reading of SOFA-format [23] binaural impulse response files relies on libmysofa [24].

A. Mapping of conceptual modes to processing modules

Each of the four modes from Section III-C is realised through a specific combination of SAF modules. Mode A uses `ambi_roomsim` for the image-source simulation; Mode B combines `ambi_enc` for the encoding stage with `multiconv` for the B-format convolution; Mode C uses `matrixconv` in a one-to-four configuration; and Mode D uses `matrixconv` in a one-to-two configuration. For Modes A–C the binaural decoding stage is realised by SAF’s `ambi_bin` module, and multi-channel decoding to a loudspeaker array uses `ambi_dec`; Mode D bypasses these decoders entirely, since the BRIR convolution already produces the binaural output.

B. Frame size handling

A practical complication arises from the differing block-size expectations within the pipeline. The PortAudio callback driving SoundJack supports configurable frame sizes of 32, 64, 128, 256, or 512 samples per call. The SAF processing modules, by contrast, operate internally at fixed block sizes that are compiled in: 64 samples for the synchronous spatialisation path (`ambi_bin`, `ambi_enc`, `ambi_dec`, `binauraliser`), and 512 samples for the convolution-based modes (`multiconv`, `matrixconv`, `ambi_roomsim`). The convolution-based modules’ 512-sample block size — which contributes to the wet-path latency discussed in Section III-B — was chosen specifically to keep partitioned-convolution efficiency acceptable for impulse responses of several seconds in length; smaller block sizes were found to miss the audio-callback deadline for typical SRIRs.

The dry path resolves this mismatch by subdividing each audio frame into 64-sample sub-blocks within the callback and invoking `ambi_bin_process` once per sub-block. In the SoundJack architecture specifically, the binaural processing is invoked as part of pulling an audio packet from the network jitter buffer and decoding it, which occurs between two consecutive audio callbacks. As long as the total time of these receiver-buffer operations remains below the audio callback interval, glitch-free playback is preserved.

The wet path takes the opposite direction: incoming frames are accumulated until the convolution modules’ 512-sample block is filled, so that for smaller callback frame sizes the wet result becomes available only after several callbacks have elapsed. The resulting wet-path latency was discussed in Section III-B.

C. Asynchronous wet processing

The wet thread runs decoupled from the audio callback through a condition-variable-based hand-off. On each callback, the per-source mono signals are written to a shared input buffer; once the accumulator reaches 512 samples, the wet thread is signalled. It processes the block — the steps depending on the active mode — and writes the result into one of three output buffers. A triple-buffer scheme is used to avoid blocking either thread: the wet thread always writes to the buffer that is neither currently being read by the callback nor most recently completed, while the audio callback reads the buffer with the most recently completed result. Index management uses lock-free atomic operations (`std::atomic`), so neither thread waits on a mutex during audio processing. If the wet thread misses a deadline, the audio callback re-uses the most recent valid wet block rather than producing silence, which avoids audible discontinuities under transient CPU load.

The structure of the wet processing loop is sketched below:

```
while (running) {
    wait_for_signal(wetDataReady);
    int writeIdx = (lastReadIdx + 1) % 3;
    while (writeIdx == lastReadIdx
        || writeIdx == prevWriteIdx)
        writeIdx = (writeIdx + 1) % 3;
    process_block(mode, wetIn,
                  wetOut[writeIdx]);
    wetReadIdx.store(writeIdx);
    prevWriteIdx = writeIdx;
}
```

Listing 1. Triple-buffered wet-thread processing loop.

D. Source position handling and crossfading

In Modes A and B, source position is communicated to the SAF modules as continuous azimuth and elevation angles, which the modules interpolate internally. In Modes C and D, source position is necessarily discretised to the spatial grid of the impulse-response dataset: the system selects the nearest available position from the loaded set per source.

To avoid audible artefacts when a source’s nearest grid position changes, an equal-power crossfade is applied between the previous and the new impulse response over a fixed window of 40 ms. During the transition, both filters are evaluated in parallel and their outputs combined as

$$y(t) = \cos\left(\frac{\pi}{2} \alpha(t)\right) y_{\text{old}}(t) + \sin\left(\frac{\pi}{2} \alpha(t)\right) y_{\text{new}}(t), \quad (1)$$

where $\alpha(t) \in [0, 1]$ is the crossfade progress. This avoids the click that a hard switch would produce, but does not eliminate the comb-filtering between phase-misaligned early reflections noted in Section II-C. Nearest-neighbour selection with crossfading is a deliberately simple strategy chosen here for its predictable computational cost, and more sophisticated interpolation schemes are available in the literature: McKenzie et al. [25] propose perceptually informed interpolation and rendering of SRIRs for room transitions, and De Muynke et al. [26] characterise the influence of grid density and panning

method on perceived source stability in navigable auralisations. Either approach is likely to yield better perceptual results than the nearest-neighbour scheme adopted here, at the cost of additional processing in the wet path; integrating such methods is discussed as future work in Section VI.

E. Head tracking integration

Optional head tracking is supported through two backends behind a common abstract interface. Its inclusion is motivated by evidence that musicians prefer head-tracked binaural rendering over static stereo in NMP contexts [27]. The first backend uses Apple CoreMotion to read orientation data from connected AirPods Pro or AirPods Max devices, yielding quaternions that are converted to yaw-pitch-roll on the audio side. The second accepts Open Sound Control (OSC) [28] messages over UDP, allowing the use of arbitrary external trackers — such as the Superware Head Tracker, hedrot, or DIY IMU-based solutions — through standardised message paths: `/head/yaw`, `/head/pitch`, `/head/roll`, or a bundled `/head/ypr` variant.

In the binaural output paths of Modes A–C, the resulting Euler angles are applied as scene rotations on the SAF binaural decoder (`ambi_bin`) at every wet-block boundary. In Mode D, head rotation can be realised by re-indexing the BRIR lookup according to the current orientation. Because both SAF modules and the head-tracker interface are queried synchronously within the audio thread, no further synchronisation primitives are needed.

V. EVALUATION

The evaluation of the system comprises two complementary parts. The first part, presented in this section, characterises the computational behaviour of the four reverberation modes under hard real-time constraints. A qualitative listening test, in which non-biased participants assessed the perceptual success of the reverberation and spatial rendering, is reported in Section V-D.

A. Experimental setup

All measurements were carried out on an Apple MacBook Pro (M1 Pro, 16 GB RAM) running macOS 15.7.3. The receiver streamed from a single peer over a localhost connection to isolate the receiver-side processing cost from any network-induced fluctuation. The audio configuration was 48 kHz sample rate, 32-sample audio callback frame size (0.67 ms callback deadline), a 512-sample wet-processing block (10.67 ms wet-thread deadline), a jitter buffer of one packet, a network-buffer size of 64 samples (approximately 750 packets per second), and Opus coding at 192 kbit s⁻¹. The 32-sample callback frame represents an aggressive low-latency configuration and a worst case for the dry-path sub-block dispatch.

The evaluation is deliberately confined to a single-source scenario in order to isolate the per-mode processing characteristics from the variability introduced by multiple concurrent sources. Since the wet-thread cost is dominated either by the

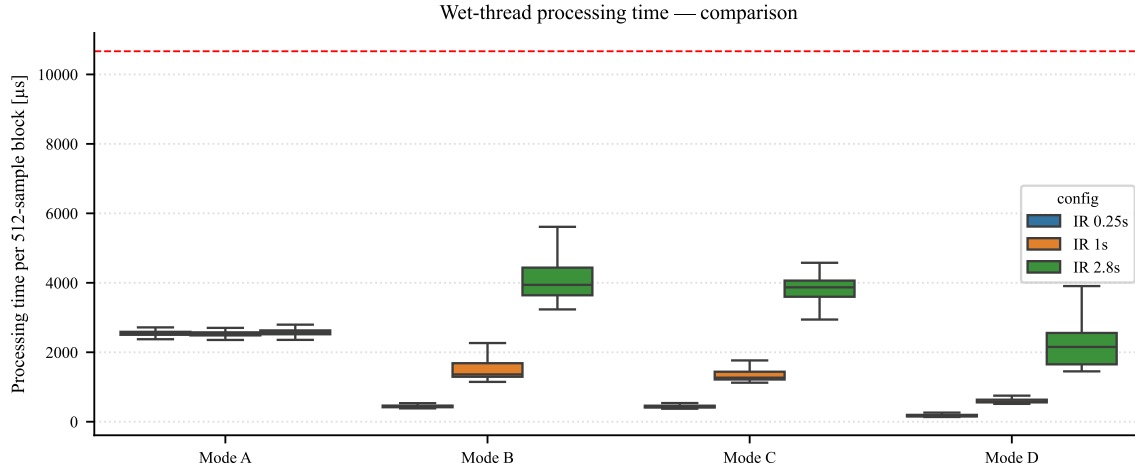


Fig. 3. Wet-thread processing time per 512-sample convolution block, for each reverb mode and three impulse-response lengths. The dashed line marks the 10.67 ms wet-thread deadline. Mode A is independent of impulse-response length, while the convolution-based modes scale sublinearly. The Mode B box aggregates over the four B-format channels processed in parallel; Modes A, C, and D show a single source.

encoding stage or by the per-source filter evaluation, scaling with source count is expected to be approximately linear for Modes C and D and sublinear for Mode B, where the shared Ambisonic scene is convolved with a single SRIR regardless of source count. A systematic multi-source characterisation remains an important subject for future work.

Within each measurement session the four reverb modes were activated in sequence with a single active source held at a fixed position. Each mode was active for approximately 30 s, yielding several thousand wet-thread blocks per mode. The first 50 wet-thread blocks following each mode change were excluded from the analysis to discard transient initialisation behaviour. Both the wet-thread processing time and the audio-callback execution time were measured using `std::chrono::high_resolution_clock` and logged to disk with microsecond resolution; all results below report the post-warm-up steady-state distribution.

Modes B and C used the four-position Hendrix Hall Derwent ambisonic SRIR dataset of the OpenAIR library [29]; Mode D used BRIRs from the WDR broadcast studio dataset [19] (Large Broadcast Studio, PA Center source position). All impulse responses were truncated to three test lengths — 0.25 s (12,000 samples), 1.0 s (48,000 samples), and the longest available BRIR length of 2.8 s (135,000 samples) — so that the modes are compared at identical lengths. Mode A was configured for a $20.1 \times 15.0 \times 6.0$ m room with reflection order 10 and uniform absorption coefficients of 0.25; since its cost is parametric, the same configuration served as a constant-cost baseline in all three series.

B. Wet-thread processing time

Fig. 3 and Table I report the wet-thread processing time per 512-sample convolution block across the four modes and three impulse-response lengths. Mode A (image-source room simulation) exhibits a near-constant median processing

time of approximately 2.55 ms; as expected, it is independent of impulse-response length, since its cost scales with the reflection order of the parametric room model rather than with any filter length.

TABLE I
WET-THREAD PROCESSING TIME PER 512-SAMPLE BLOCK, IN MICROSECONDS, ACROSS MODES AND IMPULSE-RESPONSE LENGTHS. VALUES ARE COMPUTED AFTER A 50-BLOCK SETTLING PHASE PER MODE CHANGE.

IR	Mode	Median	p95	p99	Max
0.25 s	A	2545	2701	2947	3283
	B	432	796	1857	7695
	C	425	787	1795	8321
	D	164	237	479	4112
1.0 s	A	2531	2650	2775	3400
	B	1362	2124	2289	2742
	C	1268	1541	2099	2814
	D	575	859	1713	7639
2.8 s	A	2573	2912	3148	3420
	B	3941	5173	5468	5795
	C	3870	4261	4348	4577
	D	2153	3648	5258	8611

The three convolution-based modes (B, C, D) scale with impulse-response length, but more efficiently than a naive linear cost model would predict. For Modes B and C, an 11.2-fold increase in impulse-response length (0.25 s to 2.8 s) translates to only a 9.1-fold increase in median processing time, confirming the partitioned-FFT efficiency reported by Wefers [10]. Mode D shows a steeper apparent scaling at the median (a factor of 13.1) but a near-linear scaling when measured at the 99th percentile (factor 11.0); the difference reflects the dominance of per-invocation fixed overhead at very short impulse-response lengths, where Mode D requires only 164 μs per block — close to the noise floor of the SAF `matrixconv` implementation.

One caveat applies to the direct comparison between Mode B and Modes C and D: Mode B is built on SAF’s `multiconv` module, which performs channel-wise convolution of the B-format scene, whereas Modes C and D use `matrixconv`, which evaluates a one-to-many filter matrix. The reported differences therefore reflect not only the conceptual cost of each rendering strategy but also the characteristics of two distinct partitioned-convolution implementations. Mode B additionally carries the cost of the preceding Ambisonic encoding stage, which Modes C and D fold into the convolution itself.

A notable feature of the data is that Mode A is, perhaps counter-intuitively, more expensive than all three convolution-based modes at short impulse-response lengths: at 0.25 s, Mode A requires 2.55 ms, while Modes B, C, and D remain at or below 0.5 ms. The cross-over point at which Modes B and C become more expensive than Mode A lies, for this hardware, between 1.0 s and 2.8 s; Mode D remains less expensive than Mode A at the median across the full tested range.

Across all tested configurations, every processing-time sample remained below the 10.67 ms wet-thread deadline. The worst observed value across all configurations was 8.6 ms (Mode D at 2.8 s, occurring at the upper tail of the distribution), corresponding to 81 % of the deadline. While the system therefore did not rely on the triple-buffer recovery mechanism described in Section IV-C during these measurements, that mechanism remains in place as a safety net for transient load spikes outside the tested envelope.

C. Audio-callback impact

A central architectural claim of this work is that the asynchronous wet thread relieves the audio callback from the computational cost of reverberation. The measurements directly confirm this: across all tested modes and impulse-response lengths, the audio-callback execution time at 32-sample frame size remained below 40 μ s — corresponding to less than 6 % of the 0.67 ms callback deadline (Fig. 4). The distribution shows two characteristic clusters: a denser cluster around 3 μ s to 5 μ s, corresponding to callbacks that only accumulate input samples into the 512-sample buffer, and a sparser cluster around 15 μ s to 25 μ s, corresponding to those callbacks that additionally trigger the wet-thread hand-off and consume a completed wet block. The relative populations of the two clusters — approximately 15:1 — are consistent with the structural ratio of $512/32 = 16$ dictated by the block-size mismatch.

Crucially, neither cluster scales with impulse-response length: the distribution remains structurally identical across the 0.25 s, 1.0 s, and 2.8 s test configurations. This confirms that the wet-thread isolation succeeds in its primary architectural goal — keeping the audio callback unaffected by the cost of reverb processing, regardless of the chosen mode or impulse response.

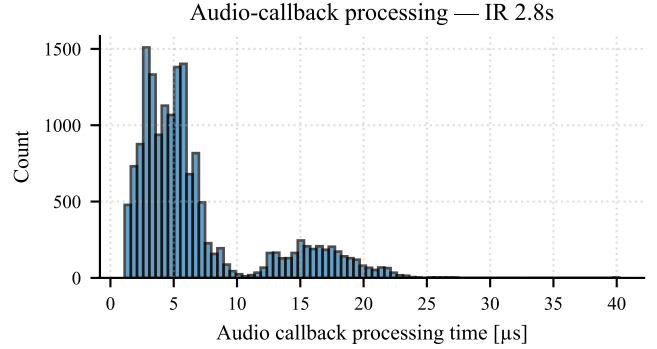


Fig. 4. Audio-callback execution time at 32-sample frame size with active Mode D reverb processing on a 2.8 s BRIR. The bimodal distribution reflects callbacks that only accumulate input samples (smaller cluster, 3 μ s to 5 μ s) versus those that additionally trigger the wet-thread hand-off (larger cluster, 15 μ s to 25 μ s). All values remain well below the 0.67 ms callback deadline.

D. Qualitative listening test

To complement the technical evaluation, a compact listener-based proof-of-concept study was conducted. While a formal perceptual evaluation following the MUSHRA methodology [30] would provide statistically robust data across a wider range of conditions, the scope of the present paper is limited to verifying whether the approach is perceived as useful in the first place; a full MUSHRA-based study across all four modes is planned as a follow-up. For this initial assessment, we integrated the reverb mode into the public version of SoundJack and asked five non-biased users (not involved in this work) to try it and provide feedback. In order to keep this process as simple as possible and in turn to allow ideal headphone usage, we used the binaural implementation of our approach and offered three binaural room impulse responses from the WDR dataset [19], representing a small, medium and large room. For the trial we instructed participants (a) to use headphones and to adjust two playback channels, (b) to use a microphone and to adjust one capture channel, (c) to set their audio and network buffers to 128 samples, (d) to enter the SoundJack public room and to establish a localhost connection to themselves, (e) to check the reverb checkbox and to leave the choice of the room to “large”, (f) to drag the red general volume and the orange reverb fader to the full right and eventually to talk into the mic. During talking, the participants were supposed to change the position of the spatial panner and repeat the listening test with medium and small rooms by changing the content of the choice box. After the trial the participants were asked the following questions: (1) Do you identify the applied effect as a regular room-size dependent reverb? (2) Can you distinguish between the three room sizes (small, medium, large)? (3) Does the 3D panner give you a realistic left/right orientation? (4) Does the 3D panner give you a realistic front/back orientation? Apart from the fact that three persons had problems with the front/back orientation of the spatial panner, all remaining questions were answered 100% positively. A limitation of this initial proof-of-

concept study is that participants tested the system with speech signals rather than musical performance material; a follow-up evaluation involving actual ensemble playing is planned as part of the MUSHRA-based study described in Section VI.

E. Discussion

Taken together, the technical measurements and the proof-of-concept listening test support three claims that motivate the proposed architecture. First, all four reverb modes can be operated in real time on commodity hardware with substantial deadline headroom, even at the aggressive 32-sample callback frame size and with concert-hall-scale impulse responses. Second, the convolution-based modes scale sublinearly with impulse-response length, confirming the suitability of partitioned-FFT convolution for the present domain. Third, the asynchronous wet thread effectively decouples the audio callback from the reverb pipeline: the callback execution time is determined entirely by the sub-block dispatch and accumulation logic of Section IV-B and is unaffected by the much heavier wet-thread workload running in parallel. The small remaining overhead — a few microseconds per callback to copy samples into the accumulation buffer, plus the occasional triple-buffer index update — is independent of both the reverb mode and the impulse-response length.

The present work is positioned as a proof-of-concept: the impulse-response datasets serve as a first working set, and their extension is a matter of dataset selection that does not affect the architecture. The single-source evaluation and the informal listening test are deliberate scoping choices, complemented by the follow-up studies in Section VI.

The proof-of-concept listening test complements these technical findings by indicating that the approach is perceived as useful in practice: the room-size-dependent reverberation was clearly distinguishable across the three tested rooms. While participants were not explicitly asked about latency or switching artefacts, none reported them, which is at least consistent with the perceptual acceptability assumption underlying the dry/wet decoupling. The front/back confusions reported by three of the five participants are a well-known limitation of binaural rendering without individualised HRTFs and will be addressed in future work.

No deadline overrun was observed during the entire measurement series. The aggressive configuration (32-sample callback frame, 64-sample network buffer, single-packet jitter buffer) thus represents a viable everyday operating point for SoundJack-based NMP with full spatial-audio rendering, not merely an extreme corner case.

VI. CONCLUSIONS AND FUTURE WORK

In this paper we have presented a novel approach for integrating reverberation and spatial audio rendering into the SoundJack low-latency audio streaming engine. The key architectural decision is the separation of the latency-critical dry path from a slightly delayed wet path containing the reverberant signal. Both paths are decoupled through a lock-free triple-buffer hand-off and function independently, so that

the ensemble-critical latency of the direct sound is preserved while the wet path is free to process impulse responses of concert-hall-scale length.

Four complementary reverberation modes were implemented and evaluated on a technical quantitative basis. The results demonstrate that all four modes meet the wet-thread deadline across the tested range of impulse-response lengths, including the demanding case of 2.8 s filters on commodity hardware, and that the audio callback itself remains unaffected by the reverb workload. This confirms that parallel processing in independent threads works as intended and that the proposed dry/wet decoupling is a viable operating point for SoundJack-based NMP with full spatial-audio rendering.

To complement the technical evaluation, we conducted a compact user-based proof-of-concept study. The results indicate that the approach is perceived as useful and that the room-size-dependent reverberation is clearly distinguishable across small, medium and large rooms. The front/back localisation of the spatial panner emerged as the primary point requiring refinement.

Based on these findings, several directions for future work naturally arise. First, the perceptual evaluation will be extended into a comprehensive formal listening study following the MUSHRA methodology [30], providing statistically robust perceptual data across all four reverberation modes and comparing musically and non-musically trained listeners.

Second, the front/back spatial rendering will be improved. While the BRIR dataset used here [19] provides high-quality artificial-head measurements, the residual front/back confusions observed suggest that additional cues — such as individualised HRTFs, parametric anthropometric adaptation, or dynamic spectral shaping — warrant further investigation.

Third, the diversity and spatial density of the effectively used impulse-response material will be extended, and the position-dependent rendering itself will be refined. This is most pressing for the SRIR sets underlying Mode C, which currently provide only a handful of measurement positions per room; denser measurement grids, or perceptually informed interpolation between existing positions along the lines proposed by McKenzie et al. [25], would allow finer position-dependent rendering and reduce the audibility of transitions between neighbouring impulse responses. For Mode D, the immediate extension is to expand the current horizontal-only BRIR lookup to the full spherical grid provided by the SOFA datasets, adding elevation as a further source dimension. Additional room types and higher-order Ambisonic SRIRs for Modes B and C are further natural directions, at a proportionally higher convolution load that the present measurements suggest to be within reach.

Fourth, a more principled treatment of the SRIR- and BRIR-based modes would decompose each measured impulse response into its direct, early-reflection, and late-reverberation components and process each separately: direct sound rendered through the low-latency dry path, early reflections through a lightweight position-dependent stage, and the diffuse late reverberation through an efficient omni-to-Ambisonic or

decorrelated multi-mono convolution. A comparable separation of early reflections and diffuse reverberation has been realised by Hupke et al. [7] in an extended-reality NMP context. Such a decomposition would eliminate the residual direct-sound duplication currently present in the wet path, resolve the acoustic mismatch between encoded source directions and measurement-position-specific early reflections in Mode B, mitigate the crossfading artefacts in Modes C and D, and reduce the computational overhead of full Ambisonic-to-Ambisonic convolution for what is largely diffuse energy.

Fifth, the multi-source scaling behaviour of the wet thread under realistic ensemble conditions will be characterised, in particular for quartet-scale and larger ensembles that go beyond the single-source configuration reported in the present evaluation.

Finally, we consider the integration into immersive virtual and augmented reality environments a particularly promising direction, in which the spatial-audio infrastructure developed here would form a natural foundation for VR-based ensemble rehearsals and performances in which musicians not only hear but also see each other within a shared virtual acoustic space. First steps in this direction have been taken by Hupke et al. [7] through projection-based visual room extension, and architectural considerations for such remote immersive musical practices, including the role of emerging network infrastructure, have been outlined by Turchet et al. [31].

ACKNOWLEDGMENT

The authors used Claude (Anthropic) for language editing and for assistance with parts of the C++ implementation. All content was reviewed and verified by the authors.

REFERENCES

- [1] A. Carôt, Musical Telepresence—A Comprehensive Analysis Towards New Cognitive and Technical Approaches, Ph.D. dissertation, Institute of Telematics, University of Lübeck, Lübeck, Germany, 2009.
- [2] A. Carôt, C. Werner, and T. Fischinger, “Towards a Comprehensive Cognitive Analysis of Delay-Influenced Rhythmical Interaction,” in *Proc. Int. Computer Music Conf. (ICMC)*, Montreal, QC, Canada, 2009.
- [3] A. Carôt, C. Hoene, H. Busse, and C. Kuhr, “Results of the Fast-Music Project—Five Contributions to the Domain of Distributed Music,” *IEEE Access*, vol. 8, pp. 47925–47951, 2020, doi: 10.1109/ACCESS.2020.2979362.
- [4] J.-P. Cáceres and C. Chafe, “JackTrip: Under the hood of an engine for network audio,” *J. New Music Research*, vol. 39, no. 3, pp. 183–187, 2010.
- [5] C. Rottondi, C. Chafe, C. Allocchio, and A. Sarti, “An overview on networked music performance technologies,” *IEEE Access*, vol. 4, pp. 8823–8843, 2016, doi: 10.1109/ACCESS.2016.2628440.
- [6] P. Cairns, “Immersive audio and network music performance: practical applications and user testing,” Ph.D. dissertation, Univ. of York, York, U.K., 2025.
- [7] R. Hupke, S. Preihs, and J. Peissig, “Immersive room extension environment for networked music performance,” in *Proc. 153rd AES Convention*, New York, NY, USA, Oct. 2022.
- [8] A. F. Genovese, “Acoustics and copresence: Towards effective auditory virtual environments for distributed music performances,” Ph.D. dissertation, New York Univ., New York, NY, USA, 2023.
- [9] J. Blauert, *Spatial Hearing: The Psychophysics of Human Sound Localization*, revised ed. Cambridge, MA: MIT Press, 1997.
- [10] F. Wefers, *Partitioned Convolution Algorithms for Real-Time Auralization*, Ph.D. dissertation, RWTH Aachen University, Berlin, Germany: Logos Verlag, 2015.
- [11] J. B. Allen and D. A. Berkley, “Image method for efficiently simulating small-room acoustics,” *J. Acoust. Soc. Am.*, vol. 65, no. 4, pp. 943–950, 1979, doi: 10.1121/1.382599.
- [12] M. A. Gerzon, “Periphony: With-height sound reproduction,” *J. Audio Eng. Soc.*, vol. 21, no. 1, pp. 2–10, 1973.
- [13] P. Cairns, H. Daffern, and G. Kearney, “Immersive network music performance: Evaluation of the effect of auralized reverberation time,” *J. Audio Eng. Soc.*, vol. 73, no. 12, pp. 862–871, Dec. 2025, doi: 10.117743/jaes.2022.0235.
- [14] C. Schörkhuber, M. Zaunschirm, and L. Turchet, “Zero-delay spatial audio rendering for immersive networked music performances,” in *Proc. IEEE 6th Int. Symp. on the Internet of Sounds (IS2)*, 2025, pp. 1–9, doi: 10.1109/IS264627.2025.11284677.
- [15] M. Tomasetti, A. Farina, and L. Turchet, “Latency of spatial audio plugins: A comparative study,” in *Proc. Immersive and 3D Audio: From Architecture to Automotive (I3DA)*, Bologna, Italy, Sep. 2023, pp. 1–10, doi: 10.1109/I3DA57090.2023.10289279.
- [16] P. Müller, R. Haefeli, J. Schütt, and M. Ziegler, “Telemersive audio systems for online jamming,” in *Proc. IEEE 5th Int. Symp. Internet of Sounds (IS2)*, Erlangen, Germany, 2024, pp. 1–15, doi: 10.1109/IS262782.2024.10704096.
- [17] M. R. Schroeder, “Natural sounding artificial reverberation,” *J. Audio Eng. Soc.*, vol. 10, no. 3, pp. 219–223, 1962.
- [18] J.-M. Jot and A. Chaigne, “Digital delay networks for designing artificial reverberators,” in *Proc. 90th AES Convention*, Paris, France, 1991.
- [19] P. Stade, B. Bernschütz, and M. Rühl, “A spatial audio impulse response compilation captured at the WDR broadcast studios,” in *Proc. 27th Tonmeisteragung — VDT International Convention*, Cologne, Germany, 2012, pp. 551–567.
- [20] A. Lindau and S. Weinzierl, “On the Spatial Resolution of Virtual Acoustic Environments for Head Movements in Horizontal, Vertical and Lateral Direction,” in *EAA Joint Symposium on Auralization and Ambisonics*, Berlin, Germany, 2012.
- [21] J. Cooper, H. Daffern, and G. Kearney, “Perceptual thresholds for angular decomposition of direct and reverberant sound in Spatial Room Impulse Responses,” in *Proc. AES Int. Conf. Spatial and Immersive Audio*, Huddersfield, U.K., Aug. 2023, paper 38.
- [22] L. McCormack and A. Politis, “SPARTA & COMPASS: Real-time implementations of linear and parametric spatial audio reproduction and processing methods,” in *Proc. AES Int. Conf. Immersive and Interactive Audio*, York, UK, 2019.
- [23] “AES69-2020: AES standard for file exchange — Spatial acoustic data file format,” Audio Engineering Society, 2020.
- [24] C. Hoene, “libmysofa,” GitHub repository, 2024. [Online]. Available: <https://github.com/hoene/libmysofa>
- [25] T. McKenzie, N. Meyer-Kahlen, R. Daugintis, L. McCormack, S. J. Schlecht, and V. Pulkki, “Perceptually informed interpolation and rendering of spatial room impulse responses for room transitions,” in *Proc. 24th Int. Congr. Acoustics (ICA)*, Gyeongju, Korea, Oct. 2022, pp. 1–11.
- [26] J. De Mynke, D. Poirier-Quinot, and B. F. G. Katz, “Creating navigable auralisations using RIR convolution: Impact of grid density and panning method on perceived source stability,” in *Proc. AES Int. Conf. Spatial and Immersive Audio*, Huddersfield, U.K., Aug. 2023, pp. 1–8.
- [27] M. Tomasetti and L. Turchet, “Playing with others using headphones: Musicians prefer binaural audio with head tracking over stereo,” *IEEE Trans. Human-Mach. Syst.*, vol. 53, no. 3, pp. 501–511, Jun. 2023, doi: 10.1109/THMS.2023.3270703.
- [28] M. Wright and A. Freed, “Open Sound Control: A new protocol for communicating with sound synthesizers,” in *Proc. Int. Computer Music Conf. (ICMC)*, Thessaloniki, Greece, 1997.
- [29] R. Cully, “The Virtual Drum Room (VDR): Simulating Room Acoustics through Real-Time Auralisation,” MSc Project Report, School of Physics, Engineering, and Technology, University of York, U.K., 2023. Hendrix Hall Derwent SRIR dataset available via OpenAIR, <https://www.openair.hosted.york.ac.uk/>. Licensed under CC BY 4.0.
- [30] ITU-R, “BS.1534-1: Method for the subjective assessment of intermediate quality levels of coding systems,” International Telecommunication Union, Recommendation, 2003.
- [31] L. Turchet, C. Rinaldi, C. Centofanti, L. Vignati, and C. Rottondi, “5G-enabled internet of musical things architectures for remote immersive musical practices,” *IEEE Open J. Commun. Soc.*, vol. 5, pp. 4691–4709, 2024, doi: 10.1109/OJCOMS.2024.3407708.