

공간 재현을 위한 보정 Impulse Response 생성 및 공연에서 실시간 적용 프로젝트 보고서

유태빈 2025. 09. 30

1. 문제 인식 (Problem Statement)

Impulse Response(IR)는 공간이 소리에 어떻게 반응하는지를 나타내는 시스템 응답이다. 공연 환경에서 컨볼루션 리버브는 목표하는 타 공연장의 IR을 그대로 적용하는 방식이 일반적이다. 그러나 서로 다른 음향 특성을 가진 공연장에 동일한 IR을 적용할 경우, 청취자는 동일한 공간으로 인지하지 않는다. 예를 들어, 작은 재즈클럽에서 A콘서트홀의 IR을 컨볼루션 할 때의 넓은 체육관에서 같은 A콘서트홀의 IR을 컨볼루션할 때, 실제 청감상 같은 콘서트홀의 IR이 컨볼루션 되었다고 느껴지지 않는다. 이는 현재 공간의 고유한 잔향, 초기 반사, 주파수 응답이 고려되지 않기 때문이다. 본 프로젝트는 현재 공간과 목표 공간 간의 차이를 정량적으로 분석하고, 그 차이를 보정하는 IR을 생성함으로써 공연 환경에서도 일관된 공간 재현을 구현하는 것을 목표로 한다.

2. 시스템 개념 (Concept)

기존 방식

“ Input Signal * Target Hall IR ”

제안 방식

“ Input Signal * **Python으로 생성된 Corrective IR** $\approx$ Target Hall IR ”

Python으로 생성된 Corrective IR = Target Hall IR / My Hall IR

3. 실험 환경 및 데이터 수집

My Hall Venue



장소 : DIMA 지성관 콘서트홀

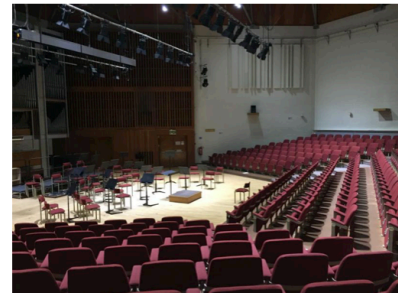
목적 : 라이브밴드 홀

좌석 수 : 약 200석

RT30 (500Hz) : 0.39s



Target Hall Venue



장소 : Jack Lyons Concert Hall,

University of York (UK)

목적 : Classic Concerts

좌석 수 : 약 350석

RT30 (500Hz) : 1.89s



IR 캡처 방식

- 서라운드 스피커 시스템 시뮬레이션, 얼라인먼트 및 튜닝
- 무대 상부 AB 스테레오 마이크 리깅
- 각 서라운드 스피커 채널별 IR 캡처
- 샘플레이트 통일: 48 kHz



4. 보정 IR 생성 알고리즘 (Python 스크립트 요약)

1. 정렬 및 전처리

- 두 IR의 직도달음 기준 시간 정렬 (peak / cross-correlation), Polarity 확인 및 게이팅

2. 대역 한정

- 125 Hz – 4 kHz 대역에 보정 가중, 지각적으로 중요한 범위에 초점

3. 주파수 영역 비율 계산

- 이상적 관계: $F(\omega) = B(\omega) / A(\omega)$, 발산 방지를 위한 정칙화 및 스무딩 적용

4. 초기·후기 반사 분리

- Early: 0–40 ms (이미징 보존), Late: 60–200 ms (과보정 방지)

5. 시간 영역 FIR 생성

- IFFT → 창 적용 → 길이 및 피크 제어

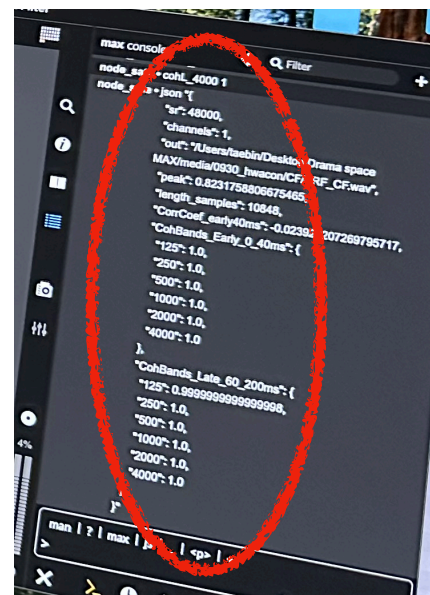
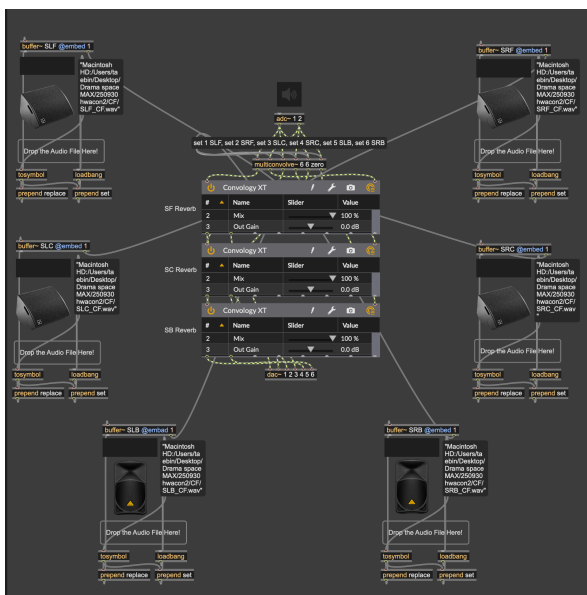
```

59 const args = {
60   String(script),
61   "--club", String(club),
62   "--hall", String(hall),
63   "--out", String(outPath),
64   "--mode", "mono",
65   "--band_low", "125", "--band_high", "4000",
66   "--transition_hz", "900",
67   "--reg_db", "-32", "--smooth_bins", "121",
68   "--gate_db", "-33", "--gate_pre_ms", "4", "--gate_tail_ms", "16", "--gate_fade_ms", "5",
69   "--early_ms", "40", "--cover_ms", "16", "--late_ms", "200",
70   "--reg_db_late", "-28", "--smooth_bins_late", "129",
71   "--late_phase", "keep",
72   "--initial_gain", "1",
73   "--peak_limit_ir", "0.95",
74   "--export_full", "1"
75 };
76
77 let out = "", err = "";
78
79 const child = spawn(String(py), args, { shell:false });
80
81 child.stdout.on("data", d => out += d.toString());
82 child.stderr.on("data", d => err += d.toString());
83
84 child.on("close", code => {
85   const text = (out + "\n" + err).trim();
86   if (code !== 0 || /usage/mi.test(text)) {
87     Max.outlet(["error", text || "python exited $(code)"]);
88     return;
89   }
90   try {
91     const j = JSON.parse(text);
92     if (j && j.error) Max.outlet(["error", text]); return;
93     Max.outlet(["done", outPath]); // 최종 파일 경
94     if (j.CorrCoef_early40ms !== undefined)
95       Max.outlet(["corrcoef", j.CorrCoef_early40ms]);
96     const bands = ["125", "250", "500", "1000", "2000", "4000"];
97     for (const k of bands)
98       if (j.CohBands_Early_0_40ms[k] !== undefined)
99         Max.outlet(["cohE_s(k)", j.CohBands_Early_0_40ms[k]]);
100     for (const k of bands)
101       if (j.CohBands_Late_60_200ms[k] !== undefined)
102         Max.outlet(["cohL_s(k)", j.CohBands_Late_60_200ms[k]]);
103   } catch(e){
104     Max.outlet(["error", "JSON parse failed\n"+text]);
105   }
106 }

```

5. 실시간 적용 시스템 (Max/MSP)

생성된 보정 IR은 Max/MSP 기반 실시간 컨볼루션 프로세서로 구현하였다. 각 서라운드 스피커 채널에 독립적으로 생성된 개별 보정 IR을 적용하여, 공연 중 실시간으로 동일한 공간 재현이 가능하도록 설계하였다.



6. 결과 및 검증

건축음향학적으로 공간의 색채를 결정하는 초기 반사 구간(0-40 ms)의 주파수 및 위상 응답을 비교 분석한 결과, 목표 효과의 일치도를 나타내는 Coherence(상관계수)가 1.0에 도달함을 검증하였다. 본 프로젝트는 공연 음향을 단순한 새로운 효과 적용이 아닌 실제 다른 공간의 건축음향 특성을 재현하는 측정·분석·보정이 가능한 공학적 시스템으로 접근한 시도이다. 현장 음향 경험을 연구 문제로 확장한 사례로서 의미를 갖는다.