

TDoA Cross + Simulated Array Design Architecture

Project: exp-tdoa-cross-correlation Updated: 2026-01-28 Owner: Jenner

1. 目標與範圍

此架構文件定義在 單聲道 MIC 音檔上建立「模擬線性陣列」, 再用 Beamforming (DS/MVDR) 與 MUSIC 進行 DoA/TDoA 估計, 並與既有的 CC/NCC/GCC-PHAT 比較。重點在 方法比較的可重現性與指標一致性。

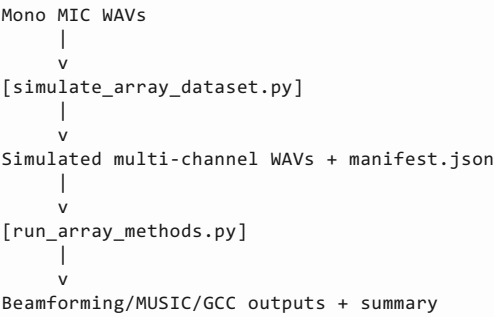
納入範圍 - 單聲道 WAV -> 模擬多聲道線性陣列 - DS Beamforming + MUSIC (wideband) - 以 GCC-PHAT 作為 array baseline - 結果輸出與統一評估

不納入範圍 - 真實陣列硬體資料與校正流程 - SRP-PHAT / FRIDA 等進階方法 (可後續擴展)

2. 假設與限制

- 目前資料為 單聲道 MIC WAV
- 模擬假設: 平面波 + 線性陣列 + 無混響 (可加噪)
- MUSIC 在 ULA 上有 $\pm\theta$ 對稱模糊, 需額外先驗處理

3. 資料流 (Data Flow)



4. 模組與職責

4.1 simulate_array_dataset.py

功能 - 將單聲道 WAV 轉為多聲道陣列 - 每個檔案指定 DOA angle - 可加噪 (SNR)、通道增益抖動

輸入 - --in_dir 單聲道 WAV 目錄 - --num_mics 陣列麥數 (預設 4) - --spacing_m 麥距 (預設 0.035 m) - --angle_* 角度設定 (random / grid) - --snr_db 模擬噪聲

輸出 - wavs/*.wav 多聲道 WAV - manifest.json (含 angle_gt, delays)

4.2 run_array_methods.py

功能 - DS beamforming - MUSIC (wideband) - GCC-PHAT (兩端麥 baseline)

輸入 - `--array_root` (wavs/ + manifest.json) - `--n_fft`, `--hop_length` - `--freq_min`, `--freq_max` - `--angle_min`, `--angle_max`, `--angle_step`

輸出 - `detailed_results.json` (per file) - `summary.json` (aggregate stats)

4.3 run_cross_correlation_tdoa.py (既有)

功能 - CC / NCC / GCC-PHAT on MIC-LDV

用途 - 作為 single-pair TDoA baseline - 與 array method 結果比較 (如有可對齊情境)

5. 評估指標

DoA / 空間頻譜 - `doa_error_deg` (vs. `angle_gt`) - `beamwidth_deg` (optional) - `spatial_psr`

TDoA / 時延 - `tau_ms` (GCC-PHAT) - `abs_diff_ms` (cross-method)

6. 實驗設計 (Simulation Protocol)

基準設定 - $M = 4$, $\text{spacing} = 0.035\text{ m}$ - DOA uniform random in $[-60, 60]$ deg - SNR = 30 dB

Sweep 建議 - SNR: 10 / 20 / 30 dB - 角度步進: 2, 5 deg - M : 2 / 4 / 6

7. 輸出結構

```
results/  
  array_sim_<timestamp>/  
    wavs/*.wav  
    manifest.json  
  array_eval_<timestamp>/  
    detailed_results.json  
    summary.json
```

8. 風險與緩解

風險	影響	緩解
MUSIC $\pm\theta$ 模糊	DoA 偏差	加角度先驗 / restrict scan range
模擬過度理想	結果過於樂觀	加混響 / noise / gain jitter
GCC vs Beamforming 不一致	指標分歧	在相同 baseline 下比較

9. 可擴展項目

- SRP-PHAT
- MVDR beamforming
- 加入真實 array 資料與校正流程
- 更完整的 multi-source MUSIC

10. 快速使用

1) 模拟 array

```
python -u scripts/simulate_array_dataset.py \  
  --in_dir "C:/Users/Jenner/Documents/SBP Lab/LDVReorientation/dataset/aud \  
  --out_dir "results/array_sim_<timestamp>" \  
  --num_mics 4 --spacing_m 0.035 --angle_mode random --snr_db 30
```

2) run beamforming + MUSIC

```
python -u scripts/run_array_methods.py \  
  --array_root "results/array_sim_<timestamp>" \  
  --out_dir "results/array_eval_<timestamp>" \  
  --num_mics 4 --spacing_m 0.035 --freq_min 300 --freq_max 3000
```

