

POLITECNICO DI TORINO

LAUREA MAGISTRALE in INGEGNERIA DEL
CINEMA E DEI MEZZI DI COMUNICAZIONE



Tesi di LAUREA MAGISTRALE

Plugin audio per il posizionamento di
sorgenti sonore multiple in uno spazio 2D
virtuale

Supervisor

Prof. ANTONIO SERVETTI

Prof. MARCO CANAVESE

Candidato

ALBERTO ROLETTO

OTTOBRE 2023

Sommario

L'idea di questo progetto, dal nome di *Quaso Localization Panner*, nasce all'interno del Dipartimento Nuove Tecnologie presso la Fondazione Scuola di Alto Perfezionamento Musicale di Saluzzo. L'obiettivo è quello di semplificare e gestire, in un unico plugin e quindi in un'unica interfaccia grafica, alcune delle operazioni più importanti che si presentano durante le fasi del missaggio audio. In particolare quando è necessario far coesistere nel mix un numero elevato di elementi ripresi singolarmente, come nel caso di una registrazione d'orchestra. Tra gli interventi di missaggio, i prime due, ovvero il "Bilanciamento" e il "Posizionamento", sono sicuramente i più importanti e prevedono un processo iterativo di modifica e ascolto nel quale il tempo necessario e le operazioni richieste aumentano all'aumentare del numero di elementi presenti. Il bilanciamento prevede la regolazione del livello di ogni traccia nel mix, al fine di creare un equilibrio piacevole e/o realistico tra le varie parti, in modo che tutte siano udibili e nessuna risulti particolarmente dominante o troppo debole. Il posizionamento, invece, prevede di determinare la posizione di ogni traccia all'interno di uno spazio sonoro immaginario, che nel caso di un ascolto a due altoparlanti è detto "Panorama Stereofonico". All'interno di questo panorama la posizione di ogni strumento può rappresentare la disposizione originale dei musicisti durante la registrazione, oppure basarsi su canoni estetici già consolidati nella cultura musicale. Semplificare queste due fasi implica ridurre il numero di operazioni che il tecnico del suono deve svolgere mediante il software, per questo motivo, nel Quaso Localization Panner, ogni traccia viene ricondotta ad un'icona su un piano la cui ascissa ne definisce la posizione nel panorama stereofonico, mentre l'ordinata definisce il livello sonoro e la relativa modifica timbrica, al fine di ricreare la sensazione psicoacustica di una maggiore o minore distanza dall'ascoltatore. Realizzare dunque la disposizione di ognuno dei vari strumenti nell'ensemble complessivo, che sia quella dei musicisti in fase di registrazione o una completamente nuova, risulta particolarmente intuitivo in quanto è sufficiente muovere graficamente le singole tracce sul piano per avere un riscontro diretto tra ciò che l'occhio osserva e ciò che l'orecchio ascolta.

Indice

1	Introduzione	1
1.1	Localizzazione di una sorgente sonora	1
1.2	Configurazioni d'ascolto	2
1.3	Digital Audio Workstation	3
1.4	Plugin Audio	4
1.4.1	Tipologie	4
1.4.2	Estensioni	5
1.5	JUCE	5
1.6	Panner	6
2	Funzionamento	8
2.1	Posizione orizzontale (Panning)	8
2.2	Distanza	8
2.2.1	Livello	9
2.2.2	Risposta in frequenza	9
2.2.3	Riverberazione	11
3	Misure sperimentali	12
3.1	Strumentazione e scelte tecniche	13
3.2	Misure e risultati	13
4	Digital Signal Processing	15
4.1	DSP in JUCE	15
4.1.1	Buffer Audio	15
4.1.2	AudioProcessorValueTreeState (APVTS)	17
4.1.3	Moduli DSP base in JUCE	18
4.2	DSP nel Quaso Localization Panner	19
4.2.1	Parametri del Quaso Localization Panner	20
4.2.2	Gain Module	22
4.2.3	Peak Filter Module	23
4.2.4	Shelf Filter Module	24

4.2.5	Reverb Sends Module Module	25
4.2.6	Pan Module	27
4.2.7	Output Gain Module	28
5	Graphical User Interface	29
5.1	GUI in JUCE	29
5.2	GUI del Quaso Localization Panner	30
5.2.1	Draggable Component	31
5.2.2	Attachments	32
6	Implementazione nella DAW	34
7	Indagine statistica	37
7.1	Questionari	37
7.1.1	Prova d'ascolto guidata	37
7.1.2	Prova d'ascolto individuale	38
7.2	Risultati	39
7.2.1	Risultati prima prova	40
7.2.2	Risultati seconda prova	42
8	Conclusioni	44

Capitolo 1

Introduzione

1.1 Localizzazione di una sorgente sonora

La capacità di localizzare con precisione una sorgente sonora è fondamentale per la nostra percezione e comprensione dell'ambiente circostante. Il cervello, infatti, elabora una serie di informazioni acustiche che, tramite l'utilizzo di meccanismi fisici e psicoacustici complessi, permettono di determinare la posizione delle sorgenti sonore nello spazio. Di seguito sono brevemente descritti alcuni dei fattori che entrano in gioco:

- **Differenze di intensità e di tempo:** Questi due meccanismi, chiamati "Interaural Level Difference" (ILD) e "Interaural Time Difference" (ITD), definiscono le differenze di ampiezza e tempo, del segnale sonoro, percepite dalle due orecchie. Il segnale di una sorgente posta lateralmente all'ascoltatore, infatti, raggiungerà prima l'orecchio più vicino; il quale capterà un suono di intensità maggiore rispetto all'orecchio opposto, permettendo così al cervello di individuarne la direzione di provenienza. Bisogna però tenere a mente che le informazioni ricavate da questi meccanismi, e la prevalenza di uno piuttosto che l'altro, dipendono dalla frequenza del segnale stesso. Infine, per una comprensione totale del fenomeno, risulta necessario considerare ulteriori caratteristiche, tra cui l'ostacolo sonoro introdotto dal cranio dell'ascoltatore e la velocità di elaborazione del sistema uditivo.
- **Spettro sonoro:** In generale, le sorgenti sonore emettono un certo spettro di frequenze. Durante la propagazione delle onde acustiche, alcune frequenze vengono attenuate o modificate in base all'ambiente circostante e alla distanza che devono percorrere. Il cervello utilizza queste informazioni spettrali, o timbriche, per determinare la provenienza del suono.

- **Riverberazione:** La riverberazione è il fenomeno che descrive le riflessioni del suono sulle superfici dell'ambiente, come pareti e oggetti. L'orecchio umano è sensibile a questi riflessi, che permettono di stimare con maggiore precisione alcune caratteristiche come la distanza della sorgente e le dimensioni dell'ambiente.

1.2 Configurazioni d'ascolto

Al giorno d'oggi i contenuti audio vengono fruiti mediante configurazioni e supporti d'ascolto molto differenti tra loro. Alcune delle principali modalità comprendono:

- **Ascolto monofonico (adimensionale o puntiforme)** L'audio viene riprodotto attraverso un singolo diffusore, combinando tutti i canali del segnale audio in un'unica sorgente sonora. Questo sistema non crea alcun tipo di spazialità o di percezione della posizione, tantomeno della profondità, e viene comunemente usato nelle trasmissioni radiofoniche e nelle comunicazioni telefoniche.
- **Ascolto stereofonico (monodimensionale, segmento lineare orizzontale)** L'audio viene riprodotto attraverso due diffusori separati, uno a sinistra e uno a destra del fronte sonoro, posti in genere alla base di un triangolo isoscele al cui vertice si trova la testa dell'ascoltatore. Grazie alle differenze di ampiezza tra canale destro e sinistro viene creata una sensazione di spazialità e tridimensionalità sonora virtuale, permettendo di posizionare in fase di editing le singole sorgenti sonore all'interno di un panorama stereofonico diffuso. Per elaborazioni binaurali si considerano anche differenze di tempo e di timbro. Una particolarità dell'ascolto stereo con diffusori esterni e frontali è il fenomeno del "Crosstalk" (Diafonia), ovvero il segnale riprodotto dal diffusore sinistro è captato anche dall'orecchio destro e viceversa. Questo fenomeno tende a ridurre il grado di separazione tra i due canali, fornendo un'immagine stereo complessiva non del tutto focalizzata come nella realtà. Il sistema stereofonico è il più utilizzato in ambito musicale e nella maggior parte delle produzioni audiovisive.
- **Ascolto multicanale (bidimensionale, superficie/campo sonoro)** L'ascolto multicanale utilizza più canali audio e altoparlanti disposti intorno alla posizione d'ascolto per creare un'esperienza sonora più immersiva e spaziale. Questo include configurazioni come audio 5.1, 7.1, o audio a oggetti basato su un numero variabile di canali. In questa configurazione gli altoparlanti vengono posizionati in maniera strategica attorno all'ascoltatore: ad esempio nel caso del 5.1 si hanno due altoparlanti frontali sinistro e destro, due altoparlanti surround sinistro e destro dietro l'ascoltatore, un canale centrale frontale e un

subwoofer per le basse frequenze. L'ascolto multicanale è utilizzato in contesti come le sale cinematografiche, l'home theatre, le installazioni audiovisive immersive e in ambito videoludico.

- **Ascolto olofonico/binaurale o 10.2/Atmos (tridimensionale, sferico)**
L'ascolto in cuffia di materiale sonoro ripreso con sistemici microfonic olofonici o binaurali, oppure realizzato in post-produzione, con elaborazione e diffusione del segnale mediati da sistemi di altoparlanti disposti anche al di sopra e al di sotto dell'ascoltatore, permette di ricreare la completa spazialità sonora nelle tre direzioni, inclusa l'elevazione o azimuth. Si tratta del fronte di sviluppo più recente, sul quale vi è grande interesse da parte dell'industria videoludica.
- **Ascolto in cuffia** L'audio viene riprodotto attraverso due altoparlanti posti sui padiglioni auricolari, questo permette una completa separazione tra i due canali e un certo grado di isolamento dall'ambiente circostante che dipende dal tipo di cuffia utilizzata (con padiglioni aperti o chiusi). Questa configurazione d'ascolto offre una spazialità più precisa e dettagliata nella quale, a causa dell'assenza di crosstalk, si ha il centro del panorama stereo più definito, tuttavia percepito in asse e ed internamente alla testa dell'ascoltatore. L'ascolto in cuffia è sempre più diffuso al giorno d'oggi per quanto riguarda la fruizione di contenuti audiovisivi, in quanto è caratterizzato da un'ascolto più intimo e privo del disturbo causato dall'ambiente esterno.

1.3 Digital Audio Workstation

Una Digital Audio Workstation (DAW) è un ambiente software per la produzione musicale, che consente la registrazione, il missaggio, l'editing e la riproduzione di audio digitale (esempio in Figura 1.1). Questo strumento è progettato per fornire ad artisti e tecnici un ambiente completo per creare, modificare e gestire registrazioni audio e MIDI. Le DAW offrono una vasta gamma di funzionalità e strumenti, tra cui tracce audio multiple per i vari elementi, strumenti virtuali, sequencer MIDI, effetti audio, mixer, librerie di suoni e altro. Oltre alle funzionalità di base, le DAW supportano anche plugin di terze parti, che consentono di espanderne ulteriormente le capacità con effetti e strumenti aggiuntivi. Ciò offre agli utenti un'ampia scelta di suoni e opzioni creative per personalizzare e migliorare la propria produzione. Le DAW risultano quindi l'ambiente ottimale per verificare le funzionalità e la corretta implementazione dei plugin durante la loro fase di sviluppo e test.

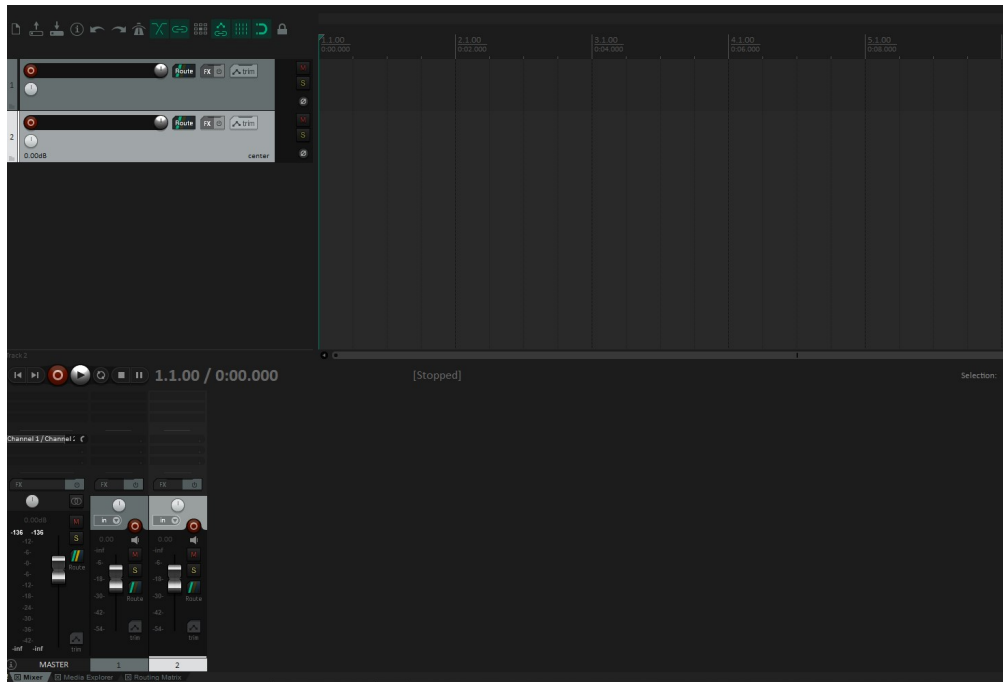


Figura 1.1: Finestra di REAPER, la DAW di Cockos. Sono visibili la timeline (a destra), le tracce (a sinistra) e il mixer (in basso).

1.4 Plugin Audio

Un plugin audio è un tipo di software utilizzato per estendere le funzionalità di un'applicazione per la produzione musicale. Esso è progettato per lavorare all'interno di un ambiente ospite, consentendo all'utente di aggiungere effetti, strumenti virtuali o altre funzionalità alla propria produzione.

1.4.1 Tipologie

I plugin audio appartengono a diverse categorie, le quali possono spesso sovrapporsi e convivere all'interno della stessa sessione di lavoro, a seconda delle esigenze tecniche e creative dell'utente. Alcune di queste categorie includono:

- **Processori di segnale:** Sono utilizzati per modificare caratteristiche intrinseche e proprie del segnale audio in maniera tecnica, al fine di compensare e correggere problematiche dovute alla ripresa microfonica, o talvolta con operazioni legate ad aspetti creativi. Alcuni processori comuni includono filtri, equalizzatori, compressori o expander, distorsori, cancellatori di rumore e analizzatori di spettro.

- **Effetti audio:** Sono utilizzati per alterare e modulare il suono del segnale audio in modo creativo ed estetico, aggiungendo elementi, originariamente non presenti, nel suono delle varie sorgenti in un mix multitraccia; tipicamente ambiente, ma anche emulazione di caratteristiche speciali. Alcuni effetti comuni includono riverberi, echi e delay.
- **Strumenti virtuali:** Permettono di emulare strumenti reali o di generare suoni sintetizzati a partire da algoritmi di sintesi. Con questi plugin è possibile suonare e registrare in real-time le parti strumentali utilizzando la tastiera del computer o un controller MIDI.

1.4.2 Estensioni

I plugin audio possono avere estensioni di file diverse a seconda del formato o dello standard utilizzato per la loro implementazione e per la comunicazione con il software ospite. Solitamente, i plugin audio sono disponibili in diversi formati ed estensioni. in modo da garantire la compatibilità su piattaforme diverse. Alcuni esempi di estensioni di file sono:

- **.VST (Virtual Studio Technology):** Questa estensione rappresenta la categoria più comune e diffusa, basata sullo standard di comunicazione VST sviluppato e introdotto negli anni '90 da Steinberg, avente estensione .DLL su Windows e .VST su MacOS. A partire dal 2008 è stata sviluppata una versione più aggiornata del protocollo, con estensione .VST3, e avente l'obiettivo di migliorare le prestazioni e l'efficienza del plugin. Alcune di queste migliorie includono il supporto al multicanale, una migliore elaborazione a bassa latenza e una maggiore efficienza nell'utilizzo delle risorse di sistema. Ad oggi non tutti i software di produzione supportano però nativamente lo standard VST3.
- **.AU (Audio Unit):** Questa estensione è specifica per i plugin audio utilizzati sulle piattaforme Apple macOS e iOS. Gli Audio Unit sono supportati nativamente da software per la produzione musicale su dispositivi Apple.
- **.AAX (Avid Audio eXtension):** Questa estensione è utilizzata per i plugin audio sviluppati per il software di registrazione Pro Tools di Avid. I plugin AAX sono specifici per l'ambiente di produzione Pro Tools e offrono un'integrazione avanzata con il software.

1.5 JUCE

JUCE è un framework open-source e cross-platform, scritto in C++, utilizzato per lo sviluppo di software audio, applicazioni musicali e plug-in audio per desktop e

mobile. Data la sua natura, le applicazioni sviluppate con JUCE possono essere eseguite su diverse piattaforme, come Windows, macOS, iOS e Android, senza la necessità di scrivere codice specifico per ciascuna di esse. Il framework offre una vasta gamma di funzionalità per facilitare lo sviluppo di applicazioni audio. Tra queste vi sono classi e librerie per la gestione dei segnali, la creazione di interfacce utente grafiche, l'implementazione di effetti audio in tempo reale, il supporto per il protocollo MIDI e molto altro ancora, grazie anche al supporto di librerie esterne e plugin di terze parti. I progetti realizzati con JUCE possono essere gestiti da un IDE detto *Projucer*, che permette di definire i parametri di progetto e la creazione di file specifici per l'editing e la compilazione all'interno di IDE come Visual Studio, Xcode, CodeBlocks e altri ancora. Il codice di un generico plugin realizzato con JUCE può essere suddiviso a partire dai due file di codice sorgente creati dal Projucer:

- **Plugin Processor:** Definisce una classe che eredita da *AudioProcessor* e che si occupa del Digital Signal Processing (DSP), degli algoritmi che lo implementano e del "Wrapping" del codice nel formato di plugin desiderato.
- **Plugin Editor:** Definisce una classe che eredita da *AudioProcessorEditor* e si occupa della creazione, gestione e personalizzazione della Graphical User Interface (GUI) e di come i componenti di questa vengano collegati ai parametri del plugin.

1.6 Panner

Un panner è un effetto audio utilizzato per disporre una o più sorgenti sonore all'interno di un determinato spazio d'ascolto. Come accennato in precedenza, questo spazio è strettamente legato alla configurazione di altoparlanti utilizzata dall'utente. Per questo motivo, esistono diverse tipologie di plugin per il panning a seconda della necessità e dell'ambito di applicazione. In generale, nell'interfaccia grafica dei software di produzione audio, ogni traccia è già dotata di un potenziometro che permette di gestire la posizione di questa nel panorama stereofonico d'ascolto. I plugin esterni, nella maggior parte dei casi, consentono quindi di agire su parametri ulteriori quali livello, equalizzazione, quantità riverbero e immagine stereo. Un esempio è dato dal plugin *Panagement 2* di Auburn Sounds, mostrato in Figura 1.2, che permette di disporre una singola traccia mono o stereo all'interno della semicirconferenza centrata sull'ascoltatore. Nel software, inoltre, la traccia può essere disposta ad una certa distanza virtuale dall'utente. Questo risultato è ottenuto mediante una riduzione del volume e l'introduzione di un riverbero, dai controlli minimali, al fine di creare una dimensione di "profondità", basata su fenomeni fisici. Tra le altre caratteristiche, il plugin permette di osservare

l'immagine stereo della traccia, gestendone l'apertura tra i canali, con la possibilità di ruotare virtualmente l'angolo di emissione della sorgente sonora, grazie al taglio, o all'enfasi, delle alte frequenze. Questo e altri plugin, come il *3D Panner* di Avid o *NX* di Waves, sono principalmente orientati alla produzione di mix per l'ascolto binaurale e multicanale, destinati ad una fruizione in esperienze particolarmente immersive per l'ascoltatore, come quelle cinematografiche e videoludiche. Il Quaso Localization Panner, trattato in questa tesi, ha invece finalità diverse. L'obiettivo è infatti quello di realizzare uno strumento atto a semplificare il missaggio di più sorgenti all'interno di un mix per l'ascolto stereofonico, e non multicanale. Per questo motivo una delle differenze principali è la possibilità di gestire più di una singola traccia, o sorgente, all'interno di un'unica istanza del plugin, riducendo così le operazioni e il tempo necessario al missaggio. Inoltre, per garantire una maggiore flessibilità, il Quaso Localization Panner non implementa un algoritmo di riverbero, ma permette invece di interfacciarsi con un qualunque plugin di riverbero esterno in base alle esigenze dell'utente.



Figura 1.2: Interfaccia grafica del plugin Panagement 2 di Auburn Sounds.

Capitolo 2

Funzionamento

Per descrivere gli stadi di elaborazione del segnale audio è necessario comprendere, da un punto di vista psicoacustico e fisico, quali dovranno essere le funzionalità del plugin stesso. Il Quaso Localization Panner deve infatti consentire il posizionamento di più sorgenti audio monofoniche all'interno di uno spazio 2D virtuale, mediato dalla stereofonia, potendone gestire separatamente la posizione (reale) sull'asse orizzontale (Panning) e la distanza (virtuale) rispetto all'ascoltatore. Quest'ultima, da un punto di vista psicoacustico, può essere suggerita nel segnale stereofonico controllando tre caratteristiche distinte di ogni elemento sonoro nel missaggio, ovvero il livello, la risposta in frequenza e la quantità di riverberazione o ambienza.

2.1 Posizione orizzontale (Panning)

La posizione di una sorgente sull'asse orizzontale è la più semplice da analizzare, in quanto prevede la distribuzione del segnale audio nel campo sonoro stereofonico mediante la regolazione di un parametro, detto comunemente "Pan", che ripartisce il livello di ogni segnale tra canale destro e sinistro. La variazione di questo parametro, che dipende dall'algoritmo utilizzato, genera una percezione di provenienza del suono che rispecchia l'intero orizzonte compreso tra i due estremi del sistema di riproduzione, e tutte le posizioni intermedie. Per quanto riguarda l'elaborazione del segnale, dovrà quindi essere possibile la gestione separata del parametro Pan per ogni singolo canale, o sorgente, in ingresso al plugin, il quale attribuirà di conseguenza una proporzione adeguata del livello sui canali d'uscita destro e sinistro.

2.2 Distanza

La percezione della distanza di una sorgente sonora rispetto all'ascoltatore è legata a fattori acustici, psicoacustici e ambientali. I parametri su cui è necessario agire

sono infatti il livello, il variare del timbro nella gamma delle medio-alte e alte frequenze e la quantità di riverberazione.

2.2.1 Livello

Per quanto riguarda il livello, si considerano le equazioni sulla propagazione di un'onda acustica in campo libero, le quali dimostrano che il livello di intensità sonora, misurato in decibel, diminuisce di 6 dB (SPL) ad ogni raddoppio della distanza dalla sorgente. Questa conclusione, che rispecchia la nostra percezione del fenomeno, rende necessaria l'introduzione, all'interno dell'algoritmo del plugin, di uno stadio di elaborazione del guadagno, o Gain, che possa agire in maniera indipendente su ogni canale in ingresso, andando ad aumentare il livello in avvicinamento e a diminuirlo in allontanamento.

2.2.2 Risposta in frequenza

Per le considerazioni sulla risposta in frequenza vanno considerati due fenomeni distinti, il primo di natura fisica e il secondo di natura psicoacustica, basato sulla nostra percezione sonora come esseri umani:

Assorbimento dell'energia acustica

L'energia acustica emessa da una generica sorgente si irradia per onde sferiche diminuendo con il quadrato della distanza. Quando si considerano le alte frequenze può sembrare che queste si attenuino più rapidamente rispetto alle basse frequenze, fenomeno che risulta attribuibile all'assorbimento sonoro dovuto alle molecole del mezzo di trasmissione. In generale l'aria. Analizzando i suoni in natura, si può osservare che questi presentano una maggiore concentrazione di energia acustica nelle basse frequenze. Per questo motivo, quando una sorgente è lontana dall'ascoltatore, oltre ad essere ad un volume minore, questa sembrerà timbricamente più "ovattata" e meno definita; in quanto le sue alte frequenze, dovute agli armonici successivi, tenderanno a decadere prima rispetto alle basse frequenze. Questo fenomeno è particolarmente rilevante nella nostra percezione della distanza di una sorgente e, dovendolo implementare nell'algoritmo di elaborazione del segnale, è stata fatta la scelta di effettuare delle misure sperimentali di assorbimento acustico, come illustrato nel Capitolo 3.

Sensibilità dell'orecchio umano

La sensibilità dell'orecchio umano non è lineare, ma bensì variabile alle diverse frequenze e in base al livello di pressione sonora, come illustrato nelle isofoniche di Fletcher-Munson in Figura 2.1. Queste curve ottenute sperimentalmente, che

possono differire da individuo a individuo, dimostrano che, in seguito ad una variazione di volume reale di un segnale, la variazione percepita dall'orecchio umano sarà diversa a seconda della frequenza. Quello che è interessante notare nel grafico è la regione di massima sensibilità a 3.5 kHz, all'interno della quale, per ragioni legate all'evoluzione della specie, ricadono i suoni consonanti che garantiscono l'intelligibilità del parlato. Questa zona della risposta in frequenza può essere ricondotta alla forma di un filtro a campana con guadagno variabile. Il quale, se implementato all'interno del plugin, permette di conferire maggiore definizione alla sorgente quando è virtualmente vicina all'ascoltatore, andando ad aumentare il guadagno della campana. Lo stesso ragionamento può essere fatto considerando l'attenuazione dovuta ad una sorgente distante. Questa rappresentazione del fenomeno non è completamente corretta da un punto di vista fisico e, se l'escursione del guadagno del filtro è esagerata, il risultato sonoro tenderà ad essere poco realistico e poco piacevole all'ascolto. Regolando però nel modo corretto questi parametri si ottiene un effetto consistente e credibile, che simula bene i meccanismi percettivi dell'orecchio umano.

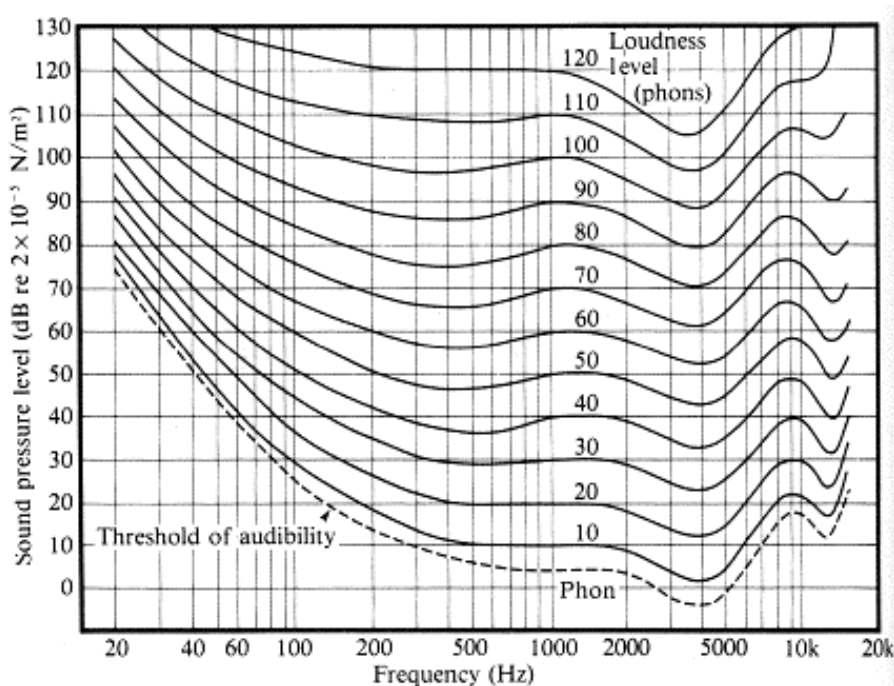


Figura 2.1: Diagramma di Fletcher-Munson. Rappresenta le curve di misura della pressione acustica in dB SPL, al variare della frequenza, ed alle quali l'ascoltatore percepisce la stessa intensità sonora. Per questo motivo vengono anche dette curve isofoniche.

2.2.3 Riverberazione

La riverberazione è responsabile di alcune caratteristiche fondamentali dell'esperienza uditiva, come la sensazione di spazialità, la percezione delle dimensioni dell'ambiente, nonché l'impressione di distanza e profondità del suono. Le onde sonore emesse da una sorgente, infatti, si propagano nell'ambiente colpendo le superfici circostanti, le quali in parte riflettono l'energia acustica incidente e in parte la assorbono. Queste riflessioni mescolandosi con il suono diretto creano un campo diffuso che definisce l'acustica propria dell'ambiente. Da un punto di vista tecnico e artistico i riverberi sono una necessità in ogni ambito della produzione audio, potendo conferire un'atmosfera particolare al suono o consolidare la sensazione di realismo di una registrazione effettuata in studio. Per questo motivo, esistono plugin dedicati che permettono sia la simulazione di storici processori hardware ma anche la possibilità di ricreare l'acustica di ambienti reali come sale da concerto e teatri, a partire dalle misure della risposta all'impulso. Risulta quindi necessario poter interfacciare il Quaso Localization Panner con un qualsiasi riverbero esterno, il cui livello per ogni canale in ingresso dovrà essere proporzionale al relativo parametro di distanza della sorgente. Maggiore è infatti la distanza rispetto all'ascoltatore, più facilmente si perderanno le informazioni binaurali sulla posizione della sorgente a cause delle molteplici riflessioni del campo sonoro diffuse su tutto il panorama stereofonico.

Capitolo 3

Misure sperimentali

Simulare, per mezzo di un filtro, l'assorbimento dell'energia acustica da parte dell'aria è indispensabile al fine di ottenere un effetto credibile della distanza di una generica sorgente dall'ascoltatore. Per questo motivo, sono state effettuate misure sperimentali di questo fenomeno all'interno della "Regia A" della Fondazione Scuola di Alto Perfezionamento Musicale. L'obiettivo delle misure è stato quello di verificare l'attenuazione delle medio-alte ealte frequenze all'aumentare della distanza dalla sorgente, in questo caso un diffusore, tramite registrazioni microfoniche e successiva analisi spettrale dei risultati.



Figura 3.1: Sorgente sonora e sistema di misura utilizzato, composto da microfono, fonometro e metro.

3.1 Strumentazione e scelte tecniche

- Microfono da misura (ECM8000 - Behringer)
- Preamplificatore microfonico DBX 576
- Diffusore (Genelec 1030A) - No Tilt, No Rolloff
- Fonometro Velleman DVM805
- DAW Pro Tools HD10
- VST Signal Generator Avid
- Pro Audio Analyzer Waves
- Linear Phase EQ Waves (Low Band)

Al fine di osservare con maggiore precisione il fenomeno si è scelto di mantenere il livello di pressione sonora alla capsula del microfono costante e pari a 65dB SPL per ogni misura, andando quindi a compensare ogni raddoppio della distanza con un aumento di emissione della sorgente di 6 dB. Per le misure è stato generato un rumore rosa a -18 dBFS che, avendo uno spettro di potenza inversamente proporzionale alla frequenza, permette di simulare meglio la percezione sonora non lineare dell'orecchio umano. Al fine di evitare rumble a bassa frequenza, ma soprattutto per compensare i problemi di cancellazione dovuti ai modi di distribuzione armonica della stanza, si è anteposto al generatore di segnale un filtro passa-alto a 200 Hz avente fase lineare e pendenza infinita.

3.2 Misure e risultati

Sono stati quindi eseguiti quattro rilievi acustici, partendo da una distanza tra capsula microfonica e cono del diffusore, di un metro e andandola a raddoppiare ad ogni misura fino ad una distanza di otto metri. Ogni registrazione è successivamente visualizzata e comparata alle altre attraverso un plugin per l'analisi spettrale. Dai risultati ottenuti in Figura 3.2 e in Figura 3.3 si osserva che all'interno della regione di attenuazione, che si estende dalla frequenza da 1 kHz fino a 16 kHz, è presente una diminuzione del livello di circa 4 dB tra la prima e la quarta misura. Il taglio ripido oltre i 16 kHz è dovuto in parte all'assorbimento energetico del mezzo, ma soprattutto alla risposta in frequenza non lineare del microfono utilizzato; fenomeno che risulta però poco rilevante ai fini della trattazione in quanto la sensibilità dell'orecchio umano in quella regione dello spettro è bassa, come mostrato in Figura 2.1.

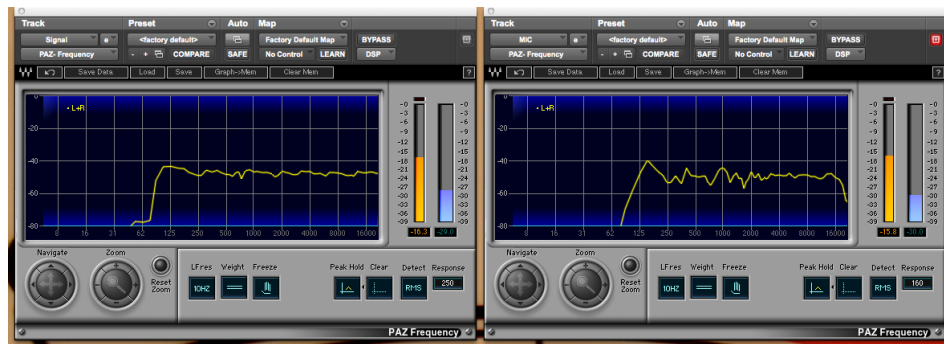


Figura 3.2: Spettro del segnale generato (a sinistra) e spettro del segnale misurato (a destra) alla distanza di 1m.

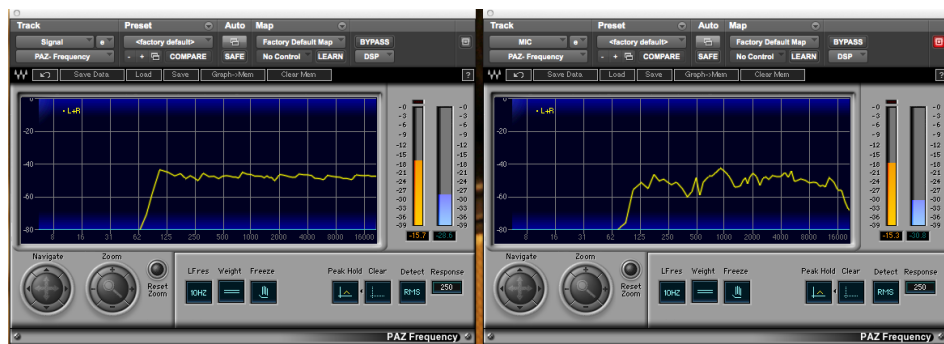


Figura 3.3: Spettro del segnale generato (a sinistra) e spettro del segnale misurato (a destra) alla distanza di 8m.

L'assorbimento dell'energia acustica può quindi essere simulato in due modi, tramite un filtro a "Scaffale", detto High Shelf Filter, o per mezzo di un filtro passa basso, detto Low Pass Filter. La versatilità del filtro shelf è duplice in quanto permette sia una enfasi costante dell'ampiezza, che un'attenuazione, all'interno della regione dello spettro designata. In questo modo può essere sfruttato in entrambi i casi di avvicinamento e allontanamento della sorgente sonora. A differenza del filtro shelf, un passa basso, attenua rapidamente le frequenze superiori a quella di taglio in base ad un parametro chiamato "Ordine del filtro". Un filtro del primo ordine attenua l'ampiezza di 6 dB ad ogni raddoppio della frequenza, o ottava, simulando, in parte in eccesso, l'effetto di assorbimento dell'energia acustica. È doveroso osservare che un passa basso non può effettuare un processo di enfasi, ma solo di attenuazione; inoltre la variazione del timbro introdotta anche solo da un filtro del primo ordine risulta eccessiva per le applicazioni tecniche e artistiche del plugin, portando ad un risultato troppo invadente nell'ambito del messaggio sonoro. L'utilizzo di un high shelf rappresenta comunque una soluzione credibile.

Capitolo 4

Digital Signal Processing

4.1 DSP in JUCE

Durante la creazione di un nuovo progetto di tipo Audio Plug-In, il Projucer genera una coppia di file header e sorgente chiamati rispettivamente *PluginProcessor.h* e *PluginProcessor.cpp*. All'interno di questi, viene definita una classe che eredita da `AudioProcessor`, ovvero la classe base di JUCE per l'analisi, l'elaborazione (DSP) e la generazione di segnali audio. Tra i metodi ereditati quelli più significativi sono `prepareToPlay()` e `processBlock()`. Il primo viene sempre chiamato prima della riproduzione (o playback) di un segnale e al suo interno, per ogni modulo della catena di DSP, vengono definiti tutti i parametri di elaborazione che non cambiano durante il playback, come la frequenza di campionamento, il numero di canali in input e in output e la dimensione del buffer, definita in campioni, o samples. Nel secondo metodo, invece, il buffer che contiene il segnale campionato viene elaborato in sequenza dai vari moduli; i quali lo scandiscono canale per canale e campione per campione, sovrascrivendo i dati in ingresso con quelli elaborati in uscita.

4.1.1 Buffer Audio

Gli algoritmi di elaborazione interni ai plugin prevedono una continua lettura e scrittura del buffer audio, ovvero una regione di memoria in cui sono immagazzinati i valori di ampiezza del segnale campionato. In generale, un buffer è rappresentabile come una matrice di vettori; ognuno di lunghezza pari alla dimensione del buffer, definita in campioni in campioni e in numero pari ai canali in ingresso, come mostrato in Figura 4.1.

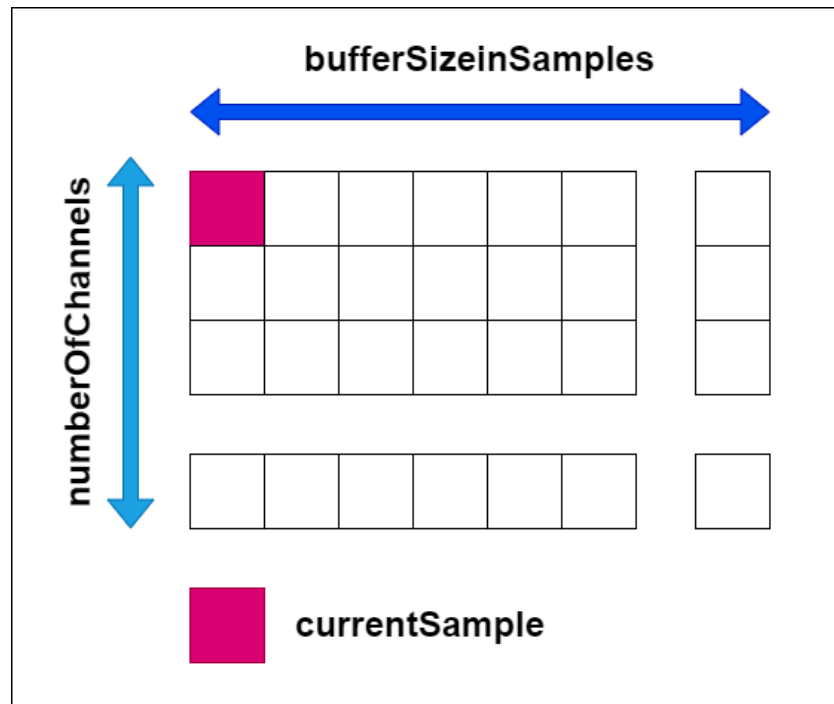


Figura 4.1: Rappresentazione matriciale di un buffer audio. Ogni cella, che individua uno dato campione in una data posizione, contiene il valore di ampiezza normalizzato del segnale campionato.

Il codice per leggere e processare i campioni all'interno di un buffer audio è simile a quello usato per la scansione riga per riga di una matrice:

```

1 int numberOfChannels = buffer.getNumChannels();
2 int bufferSizeInSamples = buffer.getSize();
3 for(int channel = 0; channel < numberOfChannels; ++channel)
4 {
5     for(int sample = 0; sample < bufferSizeInSamples; ++sample)
6     {
7         //lettura del campione alla posizione "sample" nel canale "channel"
8         float currentSample = buffer.getSample(channel, sample);
9
10        //generica funzione di processamento di un campione
11        currentSample = process(currentSample);
12
13        // riscrittura del campione nel buffer
14        buffer.setSample(channel, sample, currentSample);
15    }
16 }

```

In JUCE vengono utilizzate due strutture principali per memorizzare e manipolare dati audio:

- **AudioBuffer:** Questa classe viene utilizzata per memorizzare ed elaborare un segnale campionato e rappresenta un buffer audio di dimensioni fisse. Una volta creato, un `AudioBuffer` mantiene la sua dimensione e i dati possono essere scritti o letti all'interno di esso utilizzando gli operatori `getSample()` e `setSample()`.
- **AudioBlock:** Questa classe rappresenta un riferimento a una porzione di dati audio, inclusi i metadati associati come il numero di canali e il numero di campioni. Un `AudioBlock`, quindi, non possiede direttamente i campioni, ma punta ai dati contenuti in un altro oggetto `AudioBuffer` o a un array di dati audio esistente. Questo significa che un `AudioBlock` può essere utilizzato per manipolare dati senza dover copiare l'intero buffer, il che lo rende un oggetto particolarmente efficiente in termini di gestione della memoria e per garantire maggiore velocità quando si opera con grandi blocchi.

4.1.2 AudioProcessorValueTreeState (APVTS)

La classe `AudioProcessorValueTreeState`, abbreviata in `APVTS`, è una delle funzionalità fondamentali di JUCE per lo sviluppo di software audio, in quanto semplifica notevolmente la gestione dei parametri all'interno di un plugin o di una generica applicazione. Questa classe si basa su un oggetto `ValueTree`, il quale rappresenta l'albero dei valori, dove ogni nodo corrisponde ad un parametro di elaborazione e in cui sono salvate informazioni tra cui, il tipo di parametro (ad esempio `int`, `bool` o `float`), il valore corrente, il range di valori consentiti e il tipo di interpolazione da utilizzare. Questa struttura gestisce automaticamente l'aggiornamento dei valori di ogni parametro permettendo un'connessione diretta con i componenti che li indentificano all'interno dell'interfaccia grafica. Utilizzando l'`APVTS`, è possibile collegare facilmente i parametri audio a controlli UI come slider, bottoni e menu a discesa. Inoltre, questa struttura permette di salvare e ripristinare gli stati dei parametri per poter memorizzare le impostazioni degli utenti e ripristinarle in sessioni future. L'`APVTS` è fondamentale per la programmazione di una qualsiasi applicazione audio, di seguito sono elencati i passi fondamentali per la sua implementazione:

- Creazione di un oggetto `APVTS` all'interno del `PluginProcessor.h`, ovvero il file che definisce la classe base del plugin che si sta realizzando.
- Definizione di un metodo, solitamente chiamato `createParameterLayout()`, che restituisce una struttura di tipo `ParameterLayout`, ovvero un vettore di puntatori ai parametri audio che sono implementati nell'applicazione.

- Inizializzazione dell'APVTS chiamando il metodo `createParameterLayout()` all'interno del costruttore della classe base del plugin.
- Creazione dei parametri audio all'interno del metodo `createParameterLayout()`. Ogni parametro è definito da una classe che ne rappresenta il tipo, come `AudioParameterBool` o `AudioParameterFloat`, ed è univocamente indentificato da due stringhe, una detta "ID", e l'altra contenente il nome. Inoltre, per ogni parametro è necessario definire il valore di default, il valore massimo e il valore minimo. Tutti i puntatori ai parametri devono successivamente essere inseriti all'interno di un vettore che verrà restituito dal metodo `createParameterLayout()`.

Dopo aver creato e definito i parametri è necessario garantire la coerenza tra i valori assunti e il componente associato nell'interfaccia grafica:

- La classe del plugin deve ereditare da `AudioProcessorValueTreeState::Listener`, in modo da poter ricevere le notifiche quando avviene la modifica di un componente sull'interfaccia grafica.
- Nel costruttore della classe del plugin è necessario aggiungere per ogni parametro, identificato dal suo ID, un listener utilizzando il metodo ereditato `createAndAddParameter()`.
- Tramite il metodo ereditato `parameterChanged()`, ovvero la callback chiamata quando avviene una modifica di un parametro, si leggono i nuovi valori dall'APVTS che possono essere utilizzati come argomenti per i moduli di elaborazione del segnale.

4.1.3 Moduli DSP base in JUCE

Il framework JUCE implementa di default alcuni moduli per l'elaborazione dei segnali all'interno del namespace "DSP". Questi moduli non sono altro che classi create per processare dati audio come filtri, distorsori, panner, riverberi e altri ancora. Tra i metodi comuni ad ogni modulo, oltre a quelli per passare i valori letti dall'APVTS, hanno particolare rilevanza il metodo `prepare()` e il metodo `process()`. Il primo, solitamente invocato all'interno di `prepareToPlay()`, prende come argomento un oggetto di tipo `ProcessSpec`, ovvero una struttura contenente frequenza di campionamento, lunghezza del buffer in samples e numero di canali, al fine di passare al modulo DSP tutti quei parametri che rimarranno costanti durante il playback del segnale. Il secondo, solitamente invocato all'interno di `processBlock()`, prende come argomento un `AudioBlock` e, in base ai valori del relativo parametro dell'APVTS, processa il segnale leggendo, elaborando e riscrivendo i dati audio nel buffer.

4.2 DSP nel Quaso Localization Panner

Grazie alle considerazioni e alle misure sperimentali fatte si può osservare come, per ognuno dei sedici canali in ingresso al plugin, debba essere implementata la seguente catena (in Figura 4.2) di moduli per l'elaborazione del segnale:

- **Gain Module:** Questo modulo agisce sul livello del segnale in decibel, aumentandolo in caso di avvicinamento della sorgente e diminuendolo in caso di allontanamento.
- **Peak Filter Module:** Questo modulo agisce sulla risposta in frequenza del segnale inserendo un filtro a campana che modella il picco di massima sensibilità dell'orecchio umano.
- **Shelf Filter Module:** Questo modulo agisce sulla risposta in frequenza del segnale inserendo un filtro shelf che modella l'assorbimento dell'energia acustica da parte dell'aria.
- **Reverb Sends Module:** Questo modulo gestisce il collegamento, all'interno della DAW, tra il Quaso Localization Panner e un plugin di riverbero esterno; agendo sul livello del segnale riverberato in base alla distanza della sorgente dall'ascoltatore.
- **Pan Module:** Questo modulo posiziona ogni sorgente nel panorama stereofonico, gestendo i contributi di ogni canale in ingresso, sui due canali in uscita del sistema di ascolto.
- **Output Gain Module:** Questo modulo, presente nella maggior parte dei software audio, gestisce il livello d'uscita complessivo del plugin.

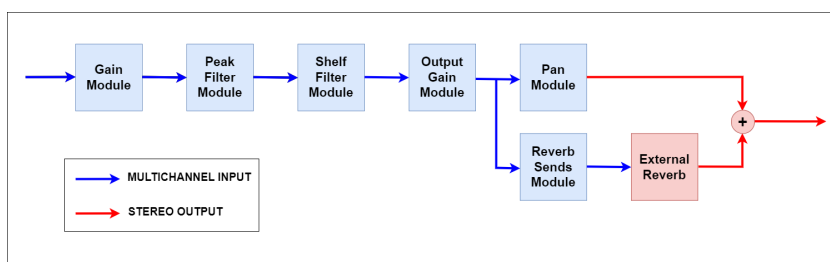


Figura 4.2: Rappresentazione schematica dei vari moduli di processamento del segnale, partendo dai sedici canali in ingresso fino ai due canali sinistro e destro in uscita

4.2.1 Parametri del Quaso Localization Panner

Per poter fornire i dati di elaborazione ai vari moduli, è necessario definire quali sono i parametri dell'APVTS associati ad ogni canale:

Pan

Parametro di tipo float, compreso tra -1 e 1, che definisce la posizione orizzontale della sorgente, pilotando l'algoritmo del Pan Module, che gestisce la differenza di ampiezza tra i canali destro e sinistro in uscita. Il valore -1 corrisponde ad una sorgente completamente a sinistra, il valore +1 ad una sorgente completamente a destra e il valore 0, di default, ad una sorgente perfettamente centrale, dove l'ampiezza del segnale è uguale per entrambi i diffusori.

Distance

Parametro di tipo float, compreso tra 0 e 1, definisce la distanza virtuale della sorgente rispetto all'ascoltatore e che pilota, tramite opportune funzioni di mappatura, i moduli Peak Filter Module, Shelf Filter Module e Reverb Sends Module. La scelta dell'intervallo, nel quale 0 rappresenta la massima distanza e 1 la minima, è conseguenza dell'utilizzo del metodo `jmap()` di JUCE, che permette di mappare un intervallo tra 0 a 1 su un altro intervallo. Visto l'obiettivo di realizzare un plugin che si adatti ad esigenze tecniche e artistiche, nelle quali non vi è spesso la necessità di alzare il livello di un segnale correttamente registrato di più di 6 dB, si è scelta una suddivisione asimmetrica della distanza; che conta tre incrementi in allontanamento e un solo incremento in avvicinamento, a partire dalla posizione di partenza. Per questo motivo è stato scelto un valore di default del parametro pari a 0.75, per il quale non vi è alcuna colorazione timbrica del segnale in ingresso, dove l'escursione tra 0.75 e 1 corrisponde ad un avvicinamento e quella da 0.75 a 0 ad un

l'allontanamento. Nella Figura 4.3, sono mostrate le relazioni che intercorrono tra i valori del parametro di Distance e i valori di guadagno dei tre moduli pilotati.

	DISTANCE	GAIN	PEAK GAIN	SHELF GAIN	REVERB GAIN
LONTANO	0	-18 dB	-4.5 dB	-3 dB	-16 dB
	0.25	-12 dB	-3 dB	-2 dB	-32 dB
	0.5	-6 dB	-1.5 dB	-1 dB	-48 dB
DEFAULT	0.75	0 dB	0 dB	0 dB	-64 dB
VICINO	1	+6 dB	+1.5 dB	+1 dB	-64 dB

Figura 4.3: Rappresentazione degli incrementi del parametro Distance in avvicinamento e allontanamento e il loro effetto sui valori dei moduli di processing del segnale associati. L'interpolazione tra questi valori è lineare.

Track

Parametro di tipo bool, che permette di attivare e disattivare il contributo del relativo canale sull'uscita; consentendo quindi di ascoltare unicamente le sorgenti scelte. Questo parametro consente di evitare di dover ricorrere ai bottoni "Solo" e "Mute" delle singole tracce all'interno della DAW, per escluderle o includerle in fase di ascolto; concentrando quindi questa operazione, frequente durante il missaggio, all'interno del plugin. Il valore di default è true e indica che il canale associato è attivo.

Output

Parametro di tipo float, compreso tra -64 dB e +12 dB, che permette di regolare il livello di ogni canale in uscita dal plugin, pilotando il modulo Output Gain Module. Il suo valore di default è 0 dB.

4.2.2 Gain Module

La classe CustomGain è basata sulla classe Gain, già implementata nel namespace DSP in JUCE, e ne estende il funzionamento ad un numero numero arbitrario di canali, in questo caso sedici. Questo modulo, a partire dai valori di guadagno, in decibel, letti dall'APVTS, modifica il valore di ampiezza di ogni campione all'interno del buffer. Il parametro Distance di ogni canale è mappato su un intervallo da -18 dB a +6 dB, con un valore di default di 0 dB. Questo si traduce, da un punto di vista acustico, in tre raddoppi (da 0dB a -18dB) e in un dimezzamento (da 0dB a +6dB) della distanza virtuale tra la sorgente e l'ascoltatore. Il metodo process() implementato in questo modulo è il seguente:

```

1  void process(juce::dsp::AudioBlock<float>& block) noexcept
2  {
3      std::vector<juce::dsp::AudioBlock<float>> inputBlocks;
4
5      for (int channel = 0; channel < __inputChannels; ++channel)
6      {
7          inputBlocks.push_back(block.getSingleChannelBlock(channel));
8      }
9
10     for (int channel = 0; channel < __inputChannels; ++channel)
11     {
12         for (int sample = 0; sample < block.getNumSamples(); ++sample)
13         {
14             block.setSample(channel, sample, inputBlocks[channel].getSample(0, sample) *
15                 gains[channel].getNextValue());
16         }
17     }

```

Il vettore inputBlocks contiene tutti i puntatori ai dati audio dei canali da processare, svincolando i campioni in ingresso e quelli in uscita e permettendo la riscrittura, di volta in volta, dei valori processati direttamente sull'AudioBlock passato come argomento al metodo process(). Il vettore gains, invece, contiene tutti i nuovi valori di guadagno di ogni canale, letti dall'APVTS e convertiti da decibel a valori normalizzati tra 0 e 1, tramite il metodo setGainsDecibels(). Per evitare "clicks" e artefatti durante la modifica dei parametri i valori contenuti nel vettore gains sono di tipo SmoothedValue e vengono interpolati linearmente su un tempo di 20 millisecondi tramite il metodo getNextValue().

4.2.3 Peak Filter Module

La classe CustomPeakFilter si basa sul processamento dei sedici canali attraverso sedici filtri a campana a guadagno variabile, implementati a partire alla classe Filter definita nel modulo DSP di JUCE. Per creare un filtro a campana è necessario impostarne i parametri tramite un metodo dedicato, che richiede come argomento un oggetto di tipo `dsp::IIR::Coefficients`; ovvero una struttura che contiene i coefficienti del filtro, definiti a partire da frequenza di centro banda, frequenza di campionamento, fattore di merito e guadagno espresso in decibel.

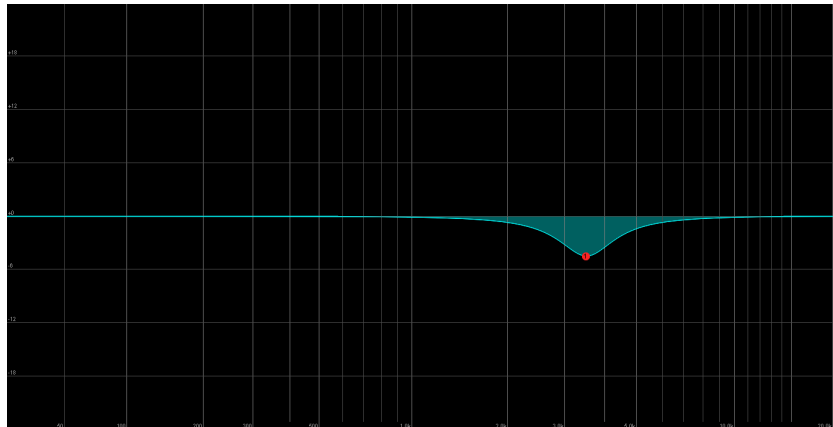


Figura 4.4: Rappresentazione del filtro a campana implementato con guadagno di -4.5 dB. Realizzato con il plugin ReaEQ di REAPER (Cockos).

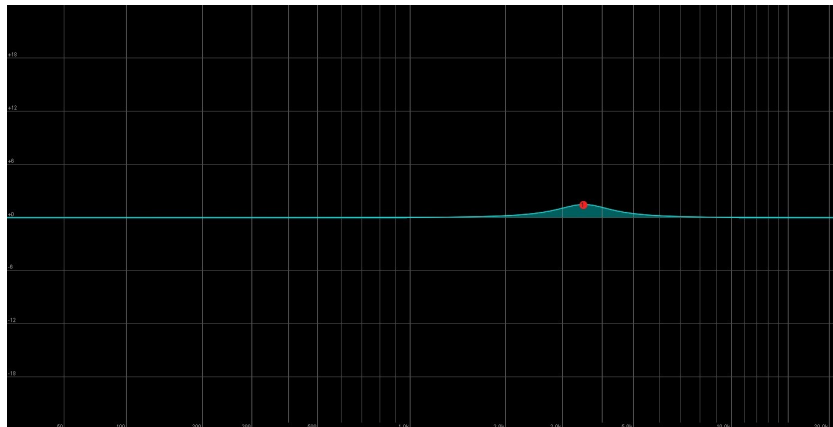


Figura 4.5: Rappresentazione del filtro a campana implementato con con guadagno di +1.5 dB. Realizzato con il plugin ReaEQ di REAPER (Cockos).

Il parametro Distance di ogni canale viene mappato su un intervallo in decibel e inserito in un vettore che li contiene tutti. In questo modo, questi, possono

essere passati come argomento al metodo che imposta il guadagno di ogni filtro all'aggiornamento dell'APVTS. Visto che la campana è centrata sul picco di massima sensibilità dell'orecchio umano, l'escursione del guadagno non deve essere troppo accentuata o si corre il rischio di rendere il risultato sonoro poco naturale e "gracchiante", ma soprattutto si perde la credibilità dell'effetto di avvicinamento o allontanamento della sorgente. Per questo motivo, dopo diversi tentativi e ascolti, sisono stati scelti i seguenti valori per ogni filtro:

- **Frequenza di centro banda:** 3500 Hz.
- **Fattore di merito (Q):** 2.
- **Escursione del guadagno:** Da -4.5 dB (sorgente più distante) a +1.5 dB (sorgente più vicina).
- **Valore di default del guadagno:** 0 dB.

4.2.4 Shelf Filter Module

La classe CustomShelfFilter si basa sul processamento dei sedici canali attraverso sedici filtri high shelf con guadagno variabile, implementati a partire dalla classe Filter definita nel modulo DSP di JUCE. Come per i filtri a campana descritti in precedenza, è necessario generare i coefficienti del filtro utilizzando il metodo createShelfFilter(), anzichè createPeakFilter(), e fornendo come argomento i valori di frequenza di campionamento, frequenza di taglio, fattore di merito e guadagno.

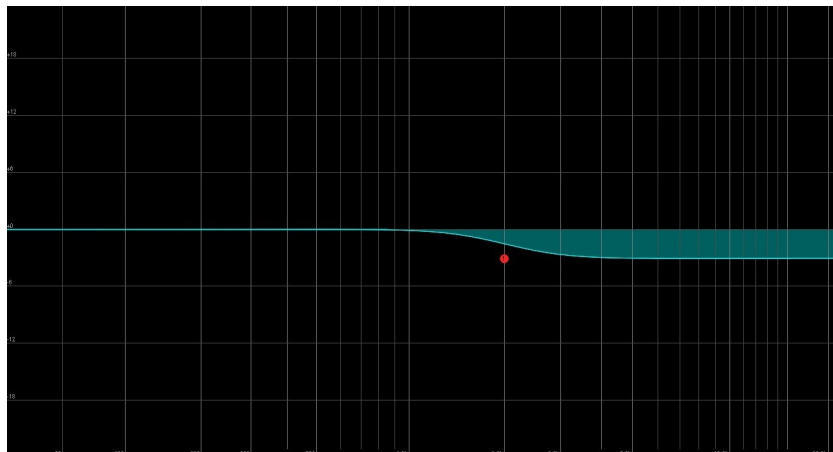


Figura 4.6: Rappresentazione del filtro shelf implementato con guadagno di -3 dB. Realizzato con il plugin ReaEQ di REAPER (Cockos).

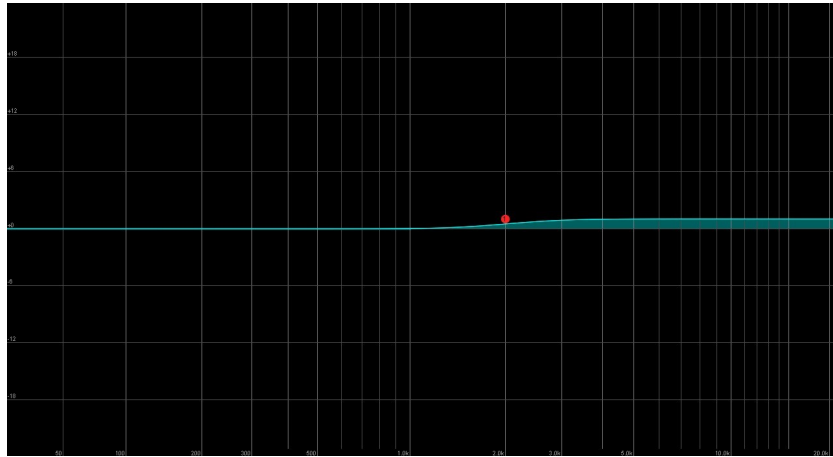


Figura 4.7: Rappresentazione del filtro shelf implementato con guadagno di +1 dB. Realizzato con il plugin ReaEQ di REAPER (Cockos).

Dai risultati ottenuti nel Capitolo 3, in Figura ?? e Figura 3.2, si osserva che la "coda" del filtro shelf deve estendersi fino ad 1 kHz, frequenza oltre alla quale vi è una progressiva ai limiti tecnici del microfono utilizzato. Per questo motivo, per modellare il fenomeno di assorbimento dell'energia acustica dovuto al mezzo di trasmissione, sono stati scelti i seguenti valori per ogni filtro:

- **Frequenza di centro banda:** 2000 Hz.
- **Fattore di merito (Q):** 1.5.
- **Escursione del guadagno:** Da -3 dB (sorgente più distante) a +1 dB (sorgente più vicina).
- **Valore di default del guadagno:** 0 dB.

Come per il modulo precedente, il parametro Distance di ogni canale è mappato su un intervallo in decibels e inserito all'interno di un vettore che li contiene tutti. Questo vettore può così essere passato, come argomento, al metodo che imposta il guadagno di ogni filtro all'aggiornamento dell'APVTS.

4.2.5 Reverb Sends Module Module

La classe CustomReverbSends gestisce il livello dei sedici canali in uscita, escludendo l'output stereo, dopo essere stati processati dai moduli descritti in precedenza. Qui il livello di ogni canale viene pesato in base al valore del parametro Distance, per consentire una corretta interazione con un riverbero esterno all'interno della DAW. Vista infatti la necessità di dover interfacciare il plugin con un qualunque

riverbero a scelta dell'utente, si sfrutta il meccanismo dei "Sends" o "Mandate". Per cui un segnale audio viene mandato parallelamente verso un'altra unità di elaborazione, o una traccia della DAW, per poi sommarsi al segnale di partenza. I riverberi sono definiti effetti paralleli, in quanto processano il segnale diretto, o "Dry Signal", generando la "coda" di riverbero e le riflessioni, in modo da ottenere il "Wet Signal". Questi due segnali vengono poi sommati in uscita come mostrato in Figura 4.8. Maggiore è la distanza della sorgente, maggiore dovrà essere il livello

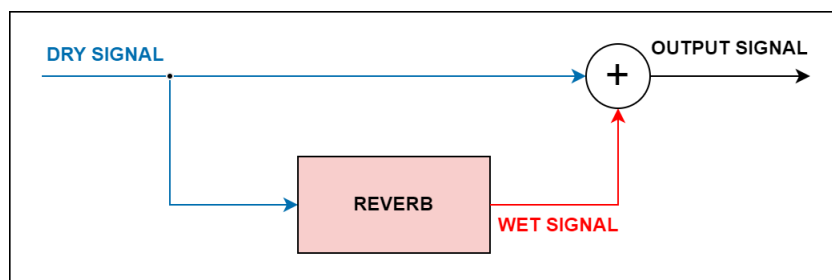


Figura 4.8: Rappresentazione schematica di un riverbero parallelo. L'Output Signal è ottenuto dalla somma del Dry Signal e del Wet Signal in uscita al riverbero.

del suo Wet Signal, in quanto, il campo sonoro diretto tende a diventare sempre più diffuso e ricco di riflessioni dovute all'ambiente. All'interno di questo modulo, il parametro Distance, definito tra 0 e 1, invertito e mappato su un nuovo intervallo in decibel, da -64 dB a -16 dB. Il risultato così ottenuto fa sì che, all'allontanarsi della sorgente si ottenga un livello, sul relativo canale in uscita, sempre maggiore; il quale, in seguito, può essere mandato ad un'altra traccia della DAW contenente un qualunque plugin di riverbero. Questo riverbero deve però generare solo il segnale Wet, in quanto il segnale Dry è già ottenuto dall'uscita stereo del Quaso Localization Panner.

Come accennato all'inizio di questo capitolo, in JUCE, l'elaborazione del segnale attraverso i moduli DSP è sequenziale e nel buffer, o AudioBlock, quelli che erano i campioni in ingresso vengono continuamente sovrascritti con i nuovi dati audio processati. Per questo motivo, a valle del Pan Module, i canali 0 e 1, che corrispondono alle uscite stereo del sistema di ascolto, contengono rispettivamente, la somma delle componenti di tutti i segnali a sinistra e la somma delle componenti di tutti i segnali a destra. Per questa ragione, non è possibile mandare le uscite 0 e 1 ad un riverbero parallelo per ottenere il segnale Wet delle sorgenti in ingresso sui canali 0 e 1. Si rende dunque necessario, all'interno del PanModule, un collegamento, o "Routing", che consiste nel copiare il buffer dei canali 0 e 1 in ingresso, già processati dai filtri, per incollarlo su due canali ausiliari in uscita, ovvero il 16 e 17. Le corrispondenze tra ingressi e uscita sono mostrate in Figura 4.9.

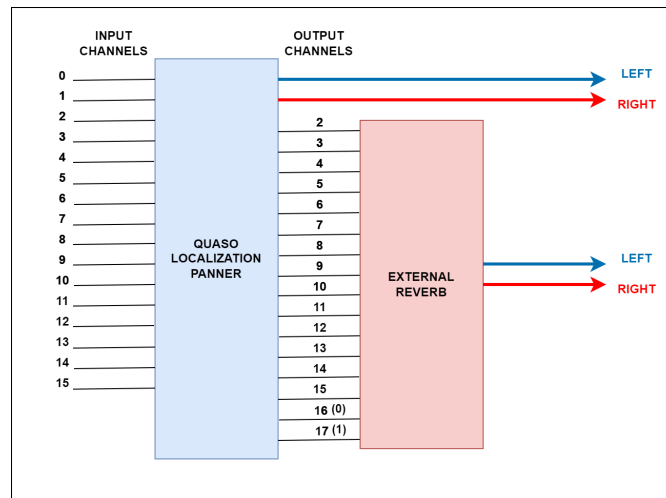


Figura 4.9: Schema dei canali in ingresso e uscita al plugin e del collegamento ad un riverbero esterno. Per poter riportare singolarmente i segnali 0 e 1 in ingresso su due uscite, i loro buffer vengono copiati sui due canali ausiliari 16 e 17.

4.2.6 Pan Module

La classe CustomPanner estende le funzionalità della classe Pan di JUCE, al fine di poter posizionare fino a sedici sorgenti sonore all'interno del panorama stereofonico. Il vettore contenente il parametro Pan di ogni canale viene passato al modulo, in modo da calcolare le componenti di segnale sui sinistro e destro del sistema d'ascolto in base alla posizione desiderata. L'algoritmo utilizzato è detto "Balanced Panning", in quanto permette di mantenere costante il livello sonoro complessivo durante il movimento panoramico della sorgente. All'interno di questo modulo, inoltre, viene anche passato il parametro Track relativo ad ogni canale, al fine di escludere dall'ascolto le tracce non attive, aventi valore "false" e andando quindi a riempire il loro buffer con campioni nulli. Di seguito è rappresentato il codice per ottenere i volume delle componenti a sinistra e a destra in base al valore del parametro Pan:

```

1  for (int channel = 0; channel < _inputChannels; ++channel)
2  {
3      float normalisedPan = 0.5 * ( _pans[channel] + 1.0 );
4      float leftValue = 1.0 - normalisedPan;
5      float rightValue = normalisedPan;
6      _leftVolumes[channel] = leftValue;
7      _rightVolumes[channel] = rightValue;
8  }

```

Dopo aver salvato i valori di ampiezza per ogni canale nei rispettivi vettori, viene chiamato il metodo `process()`, che effettua il posizionamento andando a:

- Copiare il buffer dei canali 0 e 1 sui canali ausiliari 16 e 17 al fine di avere le mandate per il riverbero degli ingressi 0 e 1, come descritto e illustrato in Figura 4.9.
- Mutare i canali aventi parametro `Track` pari a "false", andando a moltiplicare i loro campioni per zero.
- Moltiplicare i campioni di ogni canale per i rispettivi volumi sinistro e destro.
- Sommare tutti i campioni a sinistra per scriverli sul canale di uscita 0 (altoparlante sinistro) e sommare tutti i campioni a destra per scriverli sul canale di uscita 1 (altoparlante destro).

4.2.7 Output Gain Module

Questo modulo è una seconda istanza della classe `CustomGain` e permette, di modificare, con unico parametro chiamato `Output`, il livello di tutte le uscite del plugin. Questo comprende i canali 0 e 1 dell'uscita stereo principale, ma anche tutti quelli utilizzati per le mandate al riverbero.

Capitolo 5

Graphical User Interface

5.1 GUI in JUCE

Le interfacce grafiche in JUCE sono realizzate a partire da oggetti grafici elementari derivanti dalla classe Component, i quali possono essere imparentati tra di loro secondo determinate gerarchie che semplificano la programmazione dell'interfaccia. Tutti i componenti vivono all'interno della finestra principale dell'applicazione, creata dalla classe AudioProcessorEditor, la quale presenta tra i suoi dati un riferimento ad AudioProcessor, in modo da poter accedere ai parametri dell'APVTS che andranno collegati ai componenti grafici. In Figura 5.1 è presente l'interfaccia grafica del Quaso Localization Panner.



Figura 5.1: Interfaccia grafica del Quaso Localization Panner in cui sono presenti i macro elementi principali: Panning Window, Active Tracks e Output Slider

5.2 GUI del Quaso Localization Panner

Per poter semplificare e ottimizzare le operazioni svolte dal tecnico del suono durante le fasi di bilanciamento e posizionamento del missaggio audio, è necessario ridurre il numero di azioni da svolgere, cercando di concentrarle all'interno di un'unica finestra. Il Bilanciamento richiede un'azione iterativa, sul volume di ogni singola traccia, il quale è solitamente rappresentato e gestito nelle DAW da uno slider verticale, chiamato anche "Fader". Per il posizionamento, invece, per ogni traccia, si agisce su un potenziometro le cui posizioni estreme corrispondono alle posizioni totalmente a sinistra o a destra del sistema di ascolto stereo. Prendendo in considerazione le fasi successive del missaggio, in cui entrano in gioco anche equalizzazione, controllo dinamico ed effettistica, si può affermare che la complessità del lavoro e il numero di operazioni da svolgere crescono rapidamente con l'aumentare degli elementi da introdurre nel mix. Per questo motivo, nel Quaso Localization Panner, ogni traccia viene ricondotta ad un'icona su un piano, per le quali i parametri Pan e Distance, creati nell'algoritmo di DSP, sono definiti a partire alle coordinate orizzontale e verticale. L'utente può quindi muovere e disporre le icone nell'interfaccia grafica, al fine di realizzare la disposizione acustica desiderata per i vari elementi presenti. L'interfaccia del Quaso Localization Panner può essere suddivisa nei seguenti elementi principali:

- **Panning Window:** All'interno di questa finestra, mostrata in Figura 5.2, tutti i canali in ingresso, sono rappresentati come icone quadrate definite a partire dalla classe `DraggableComponent`. La quale è stata creata ad-hoc per questo plugin. Ogni sorgente può essere posizionata all'interno della Panning Window tramite il movimento del mouse. Le coordinate X e Y di ogni icona nella finestra vengono poi mappate per definire i parametri Pan e Distance. L'intersezione tra la linea verticale e orizzontale all'interno della Panning Window rappresenta la posizione di default, nella quale il segnale audio si trova al centro del panorama stereofonico e non vi è alcun tipo di elaborazione dovuta ai moduli di DSP.
- **Active Tracks:** Questo elemento, mostrato in Figura 5.3, contiene al suo interno sedici bottoni, definiti a partire dalla classe `Button` di JUCE. Ogni bottone è associato ad un canale in ingresso e ha la funzione di attivarlo o disattivarlo, andando a modificarne il parametro `Track` salvato nell'APVTS. Il colore e il nome di ogni bottone riflettono quello del canale attivo associato. Il colore grigio, invece, indica un canale disattivato.
- **Output Slider:** Questo slider verticale, posto sulla destra dell'interfaccia, è creato a partire dalla classe `Slider` di JUCE. Ad esso è associato il parametro `Output` dell'APVTS.

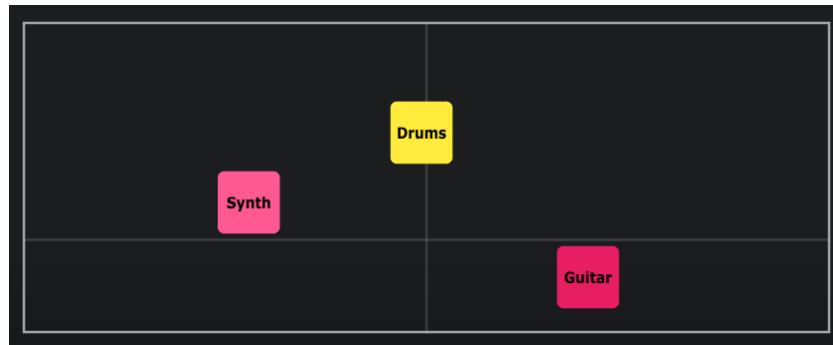


Figura 5.2: Immagine della Panning Window, in cui si possono notare le tre sorgenti attualmente attive.

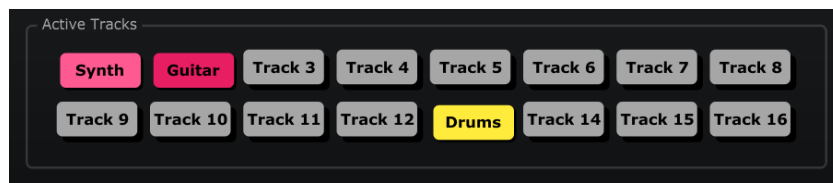


Figura 5.3: Immagine della sezione Active Tracks, in cui si possono notare i bottoni colorati per le tracce attive e grigi per quelle disattivate.

5.2.1 Draggable Component

Ogni sorgente in ingresso al plugin è implementata graficamente da un oggetto di tipo `DraggableComponent`. Questi componenti ereditano dalla classe `Component` di JUCE e hanno tra i loro dati un'istanza di un oggetto di tipo `ComponentDragger` e di uno di tipo `ComponentBoundsConstrainer`. Il primo di questi permette il movimento del componente tramite click e trascinarsi del mouse, mentre il secondo impone una restrizione su questo movimento all'interno di un'area definita, in questo caso la superficie della Panning Window. Per questo motivo i sedici `Draggable Component` sono posti, all'interno della struttura gerarchica, come componenti figli della Panning Window. In questo modo si ha un accesso diretto ai dati di ogni figlio, al fine di effettuare letture o modifiche dei valori. Di seguito sono elencate le caratteristiche più importanti dei `Draggable Component`:

- **Movimento:** Come già accennato, ogni oggetto di questo tipo può essere posizionato all'interno della Panning Window afferrandolo con un click e trascinandolo verso la posizione desiderata.
- **Posizione di default:** La posizione di default di ogni `Draggable Component` è data dall'intersezione tra le righe orizzontale e verticale presenti nella Panning Window. Effettuando un doppio click sull'oggetto è possibile riportarlo

automaticamente in questa posizione.

- **Etichetta:** Ogni Draggable Component ha tra i suoi dati un oggetto di tipo Label, che contiene e visualizza il nome del componente, come mostrato sull'interfaccia grafica. Di default ogni nome è del tipo "Track N", dove N è il numero compreso tra 1 e 16 del canale associato. Tramite un click sull'etichetta l'utente può modificare il nome della sorgente e di conseguenza quello visualizzato sul relativo bottone.
- **Colore:** Ogni Draggable Component è caratterizzato da un colore univoco, non modificabile, che viene impostato nel costruttore dell'oggetto. Questo colore, come per l'etichetta, è lo stesso del suo bottone.
- **Bottone associato:** Come accennato, ogni Draggable Component ha un bottone associato, che permette di attivare o disattivare il relativo canale in ingresso. Quando un bottone è disattivato, il suo Draggable Component non è più visibile sull'interfaccia grafica e l'utente non ha modo di interagirci. In questo modo è possibile ridurre il numero di elementi visibili in base alle necessità e al numero di sorgenti utili.

5.2.2 Attachments

Per poter collegare i componenti della UI ai parametri di funzionamento salvati nell'APVTS, è necessario utilizzare apposite classi dette "Attachment". Per configurare correttamente gli Attachment, la classe padre, che contiene i componenti da "ascoltare", deve ereditare dalla classe associata di tipo "Listener". Nel Quaso Localization Panner, infatti, la finestra principale eredita da Button::Listener e da Slider::Listener; mentre la Panning Window, che contiene i Draggable Component, eredita dalla generica classe Component::Listener. Queste classi padre, tra i loro dati, contengono un'istanza degli Attachment associati ad ogni componente, che vengono inizializzati nel costruttore al fine di passare i parametri dell'APVTS; i quali sono letti dall'istanza del PluginProcessor. Gli oggetti di tipo Draggable-Component non sono componenti default di JUCE e per questo motivo il relativo Attachment è definito, a partire da quello generico, nel seguente modo:

- Nel metodo mouseDown(), chiamato quando il tasto del mouse viene premuto sul componente, vengono inizializzate le "Gesturew" degli attachments relativi ai parametri Pan e Distance. Una Gesture rappresenta un'interazione continua tra componente e parametro, in questo caso sottoforma di trascinamento del mouse.
- Nel metodo mouseDrag(), chiamato quando il componente è stato afferrato e trascinato, si aggiornano i parametri di Pan e Distance dell'APVTS. Questo

viene svolto andando a mappare le coordinate X e Y del componente all'interno della Panning Window sull'intervallo dei parametri da aggiornare.

- Nel metodo `MouseUp()`, chiamato quando il tasto del mouse viene rilasciato, si terminano le due Gesture precedentemente attive.

Capitolo 6

Implementazione nella DAW

Il Quaso Localization Panner è un plugin multicanale definito da 16 ingressi e 18 uscite, se si considerano le due uscite ausiliare descritte nel Capitolo 4. Per questo motivo è stato scelto di effettuare il "Wrapping" del plugin nel formato VST3; che, oltre a garantire un supporto su buona parte delle DAW in commercio, è adatto alla realizzazione di plugin multicanale, a differenza dello standard VST. Per la sua implementazione all'interno di una DAW è quindi necessario che questa supporti sia l'estensione del file che la creazione e gestione di tracce multicanale. Per questo motivo, durante tutto lo sviluppo del plugin si è scelto di utilizzare REAPER, DAW di proprietà di Cockos, che fa dei suoi punti di forza il supporto completo verso la maggior parte dei formati e la possibilità di personalizzazione tramite scripting e software di terze parti. Per una corretta implementazione del Quaso Localization Panner in REAPER, vanno seguiti alcuni passaggi:

- Creare una nuova traccia e, aprendone il menù di routing, impostare il numero di canali a 18. Aggiungere su questa traccia il plugin del Quaso Localization Panner e, aprendo il suo menù "Plugin pin connector", connettere i 16 canali in ingresso e i 18 in uscita, come mostrato in Figura 6.1.
- Creare un numero di tracce aventi un singolo canale, e fino ad un massimo di sedici, per ogni sorgente che si intende gestire con il plugin. All'interno del menù di routing di ognuna disattivare la voce *Master Send*, in questo modo non si ha un doppio contributo di ogni sorgente sull'uscita. Sempre all'interno di questo menù è necessario impostare una nuova mandata verso uno dei canali della traccia contenente il Quaso Localization Panner, come mostrato in Figura 6.2.
- Creare una traccia stereo in cui inserire il plugin di riverbero che si intende usare e modificarne il valore potenziometro Dry/Wet, che cambia a seconda del software utilizzato, per avere un'uscita unicamente Wet, ovvero contenente

solo le riflessioni dell'ambiente e senza segnale diretto. Il segnale diretto è infatti quello in uscita da primi due canali del Quaso Localization Panner.

- All'interno del menù di routing della traccia contenente il Quaso Localization Panner, creare, per ogni canale in ingresso utilizzato, una mandata verso traccia contenente il riverbero, come mostrato in Figura 6.3.

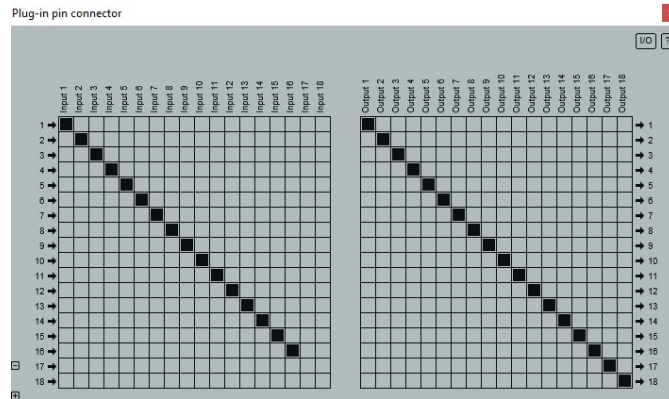


Figura 6.1: Menù in cui si connettono ingressi e uscite della traccia di REAPER ai canali definiti all'interno del codice del plugin.

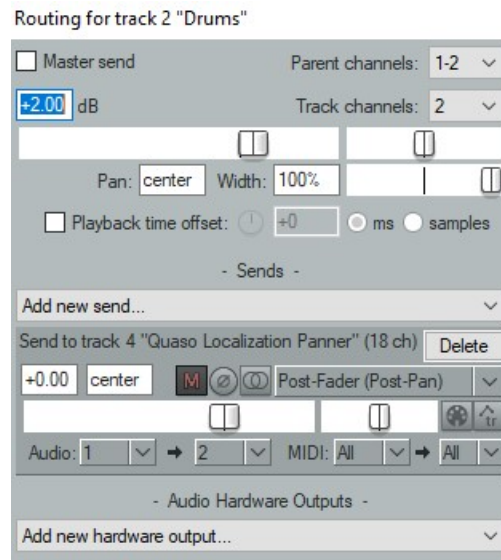


Figura 6.2: Menù di routing di una generica sorgente che si intende gestire con il Quaso Localization Panner.

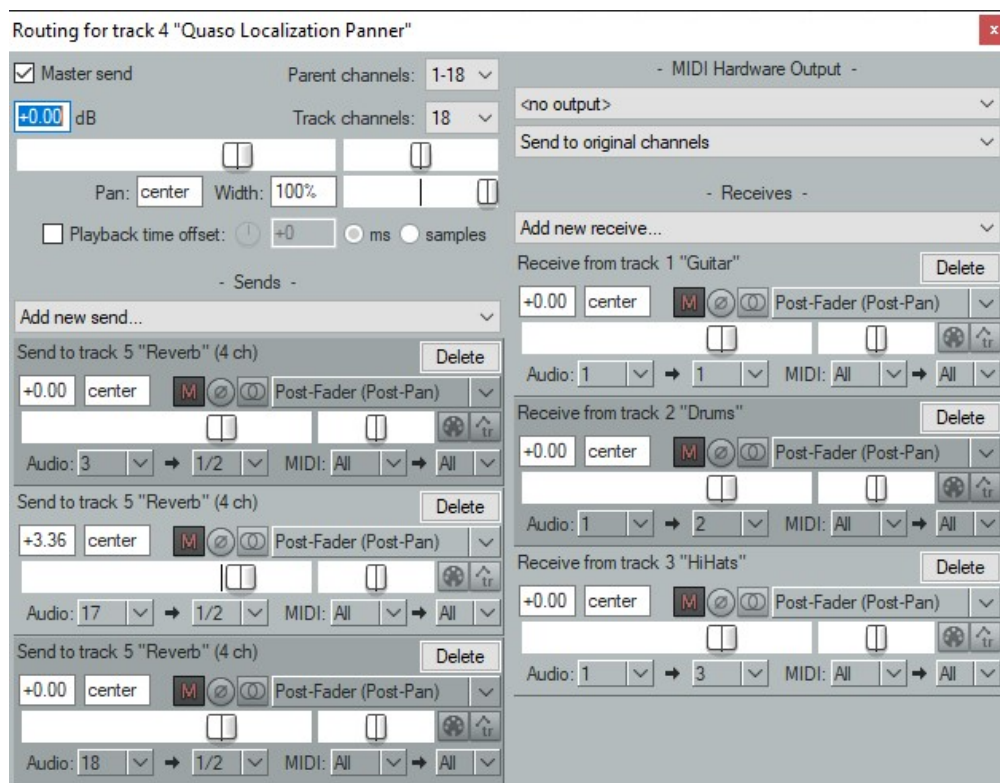


Figura 6.3: Menù di routing del Quaso Localization Panner, in cui si osservano le sorgenti in ingresso e le mandate verso il riverbero esterno.

Capitolo 7

Indagine statistica

7.1 Questionari

Come illustrato in precedenza, la percezione della posizione di una sorgente sonora nello spazio è legata a fattori acustici, ambientali e psicoacustici, i quali, oltre alle diverse modalità con cui si presentano, possono variare da individuo a individuo. Per questo motivo, al fine di verificare la compatibilità tra i risultati ottenuti per mezzo del Quaso Localization Panner e la percezione umana del fenomeno, sono stati realizzati due questionari. Ogni questionario contiene, in una prima parte, domande di carattere generale aventi l'obiettivo di verificare il sistema d'ascolto utilizzato dall'utente e la sua eventuale competenza con DAW e software di produzione audio. La seconda parte, invece, è costituita da due prove d'ascolto, una guidata e una individuale, al fine di raccogliere dati sulla percezione soggettiva del fenomeno studiato. Entrambe le prove d'ascolto utilizzano come sorgente una registrazione di doppiaggio cinematografico, fornita dalla Scuola APM. Nel parlato, infatti, le caratteristiche acustiche dei suoni consonanti si trovano nell'intervallo dello spettro che contiene la zona di massima sensibilità dell'orecchio umano.

7.1.1 Prova d'ascolto guidata

La prima prova d'ascolto guidata, proposta come video, serve a introdurre l'intervistato al funzionamento del software, evidenziando lo stretto legame tra la posizione della sorgente nell'interfaccia grafica e il risultato sonoro che ne consegue. Per questo motivo, si è scelto di utilizzare una singola sorgente, muovendola dalla posizione di default verso gli estremi della Panning Window (come illustrato in Figura 7.1), per poi evidenziare brevemente la possibilità di disporre l'elemento in tutte le posizioni intermedie dell'interfaccia.

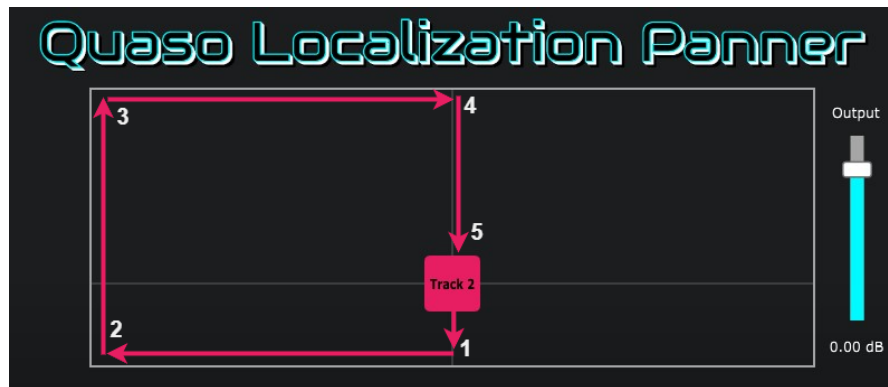


Figura 7.1: Rappresentazione del movimento della sorgente durante la prova d'ascolto guidata.

In seguito alla visione del video vengono poste all'intervistato delle domande, la cui risposte sono da una parte scale di valutazione numerica e dall'altra scelte multiple. Nel primo caso viene chiesto di valutare, con valori da 1 a 5, quanto si percepiscono le sensazioni di avvicinamento, allontanamento e spostamento laterale della sorgente. Nel secondo caso, invece, viene chiesto di selezionare quali effetti vengono percepiti quando la sorgente si allontana e si avvicina. In quest'ultima modalità, il set di possibili scelte è uguale per entrambe le domande in modo da poter verificare quale componenti, tra variazione del livello, variazione del timbro e presenza di riverberazione, sono più dominanti nella percezione del fenomeno; ma anche l'eventuali risposte errate da parte dell'intervistato.

7.1.2 Prova d'ascolto individuale

La prova d'ascolto individuale, è un video a schermo nero in cui la stessa traccia audio viene disposta e riprodotta in quattro posizioni diverse dell'interfaccia grafica. Due agli estremi e due intermedie. All'intervistato non viene mostrata la finestra del Quaso Localization Panner, perchè in questo modo si può verificare il processo percettivo inverso, ovvero la ricostruzione grafica della posizione della sorgente a partire da una fase di ascolto. Successivamente, viene chiesto all'intervistato di scaricare un'immagine dell'interfaccia del plugin e di modificarla, disegnando e numerando le posizioni della sorgente in base all'ascolto effettuato. I due questionari presentano unicamente differenze in questa prova d'ascolto, in cui le tracce 1, 2, 3 e 4, presenti nel primo questionario (Figura 7.2), vengono proposte nel secondo nell'ordine 2, 1, 4 e 3 (Figura 7.3).

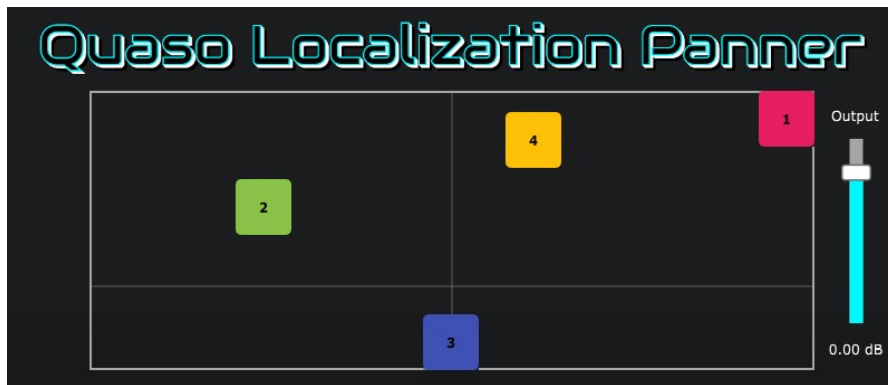


Figura 7.2: Posizioni della sorgente nella prova d'ascolto individuale del primo questionario.

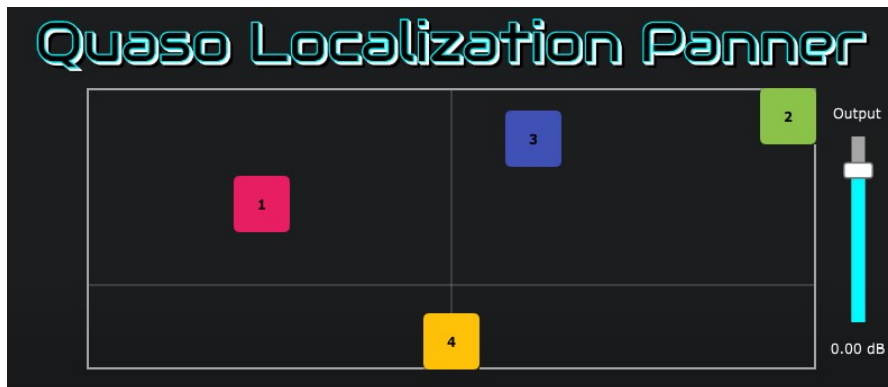


Figura 7.3: Posizioni della sorgente nella prova d'ascolto individuale del secondo questionario.

7.2 Risultati

I test sono stati proposti ad un campione di persone composto in parte da professionisti del settore audio, in particolare ex allievi della scuola APM, e in parte da persone senza competenze nell'ambito. Considerando entrambi i questionari sono state ottenute 33 risposte totali e dalle domande di carattere generale, poste all'inizio, è stato possibile caratterizzare il campione di persone come segue:

- L'80% dei partecipanti ascolta musica quotidianamente.
- Il 60% dei partecipanti ha esperienza professionale con l'utilizzo di software di produzione audio. Il 10% utilizza questi programmi in modo amatoriale e il restante 30% non li ha mai utilizzati.

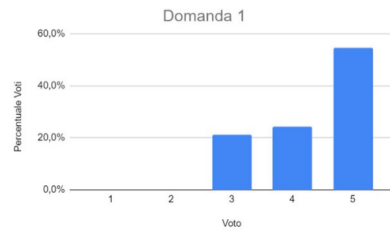
- Il 70% dei partecipanti ha utilizzato, per la compilazione del questionario e le relative prove d'ascolto, cuffie o auricolari, mentre il restante 30% ha utilizzato una coppia di monitor da studio.

7.2.1 Risultati prima prova

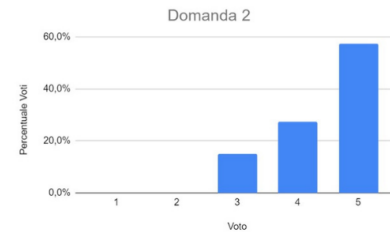
Nella prova d'ascolto guidata, le prime quattro domande prevedono una risposta tramite scala di valutazione numerica con valori da uno a cinque; dove uno indica una percezione del fenomeno bassa, o assente, mentre cinque indica una percezione molto marcata. A partire dalle tabelle e dai grafici in Figura 7.4, per le prime tre domande su avvicinamento, allontanamento e spostamento laterale (da vicino), più della metà dei partecipanti ha indicato un'ottima percezione dei fenomeni e nessuno di questi ha fornito una valutazione pari a due o inferiore. Questo risultato è ancora più evidente nella terza domanda, dove il 69% degli intervistati ha indicato la valutazione massima. Questo risultato più marcato è attribuibile in particolare all'ascolto in cuffia o tramite auricolare, dove la completa separazione tra i canali destro e sinistro rende più evidenti gli spostamenti sull'asse orizzontale. La quarta domanda, invece, riguarda sempre uno spostamento laterale, ma questa dall'estremo sinistro verso il centro, nella condizione in cui la sorgente si trova alla distanza massima. Qui il segnale diretto, contenente le informazioni sulla posizione, si somma alle riflessioni generate dal riverbero, le quali sono diffuse su tutto il panorama stereo, rendendo di conseguenza più difficile la localizzazione. Riassumendo quanto detto nei capitoli precedenti, una sorgente sonora in avvicinamento all'ascoltatore è caratterizzata da un'aumento di volume e da una leggera enfasi delle medio-alte e alte frequenze. Il suono tende ad essere più "brillante" e definito dal punto di timbrico, e la quantità di ambienza è ridotta, in quanto il campo diretto prevale su quello diffuso. Una sorgente in allontanamento, invece, prevede una riduzione del volume, a cui è associata un'attenuazione nell'intervallo delle frequenze medio-alte e alte. Il suono risulta quindi più "scuro" e sfocato e si ha una prevalenza del campo diffuso e delle riflessioni rispetto al campo diretto. Nelle ultime due domande, all'intervistato, viene chiesto di scegliere, tramite risposta multipla, quali fattori vengono percepiti durante le fasi di avvicinamento e allontanamento della sorgente. Dalla Figura 7.5, si può osservare che i due fattori percepiti maggiormente riguardano l'aumento o diminuzione del volume e la presenza del riverbero, un risultato che è conforme alla teoria sulla percezione umana della distanza di una sorgente sonora. Le variazioni timbriche dovute all'assorbimento energetico del mezzo e alle curve isofoniche, nonostante il contributo poco marcato, sono comunque state percepite da più di metà degli intervistati; con una maggiore prevalenza tra coloro che hanno indicato di avere un'esperienza professionale nell'utilizzo di DAW. Individuare queste variazioni nel timbro della sorgente richiede infatti una certa

attenzione verso l'ascolto critico, come quello sviluppato dai tecnici del suono durante la loro professione.

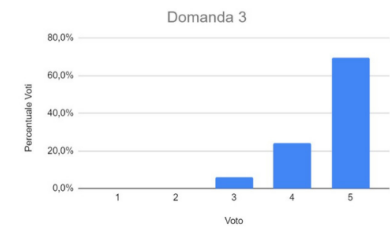
Domanda 1			
Quando la sorgente è spostata verso l'alto, quanto si percepisce l'effetto di allontamento?			
Intervistati	Voto	Numero Voti	Percentuale Voti
33	1	0	0,0%
	2	0	0,0%
	3	7	21,2%
	4	8	24,2%
	5	18	54,5%



Domanda 2			
Quando la sorgente è spostata verso il basso, quanto si percepisce l'effetto di avvicinamento?			
Intervistati	Voto	Numero Voti	Percentuale Voti
33	1	0	0,0%
	2	0	0,0%
	3	5	15,2%
	4	9	27,3%
	5	19	57,6%



Domanda 3			
Quando la sorgente è spostata verso sinistra, quanto si percepisce l'effetto di spostamento laterale?			
Intervistati	Voto	Numero Voti	Percentuale Voti
33	1	0	0,0%
	2	0	0,0%
	3	2	6,1%
	4	8	24,2%
	5	23	69,7%



Domanda 4			
Quando la sorgente è lontana ed è spostata verso il centro, quanto si percepisce l'effetto di spostamento laterale?			
Intervistati	Voto	Numero Voti	Percentuale Voti
33	1	0	0,0%
	2	0	0,0%
	3	14	42,4%
	4	7	21,2%
	5	12	36,4%

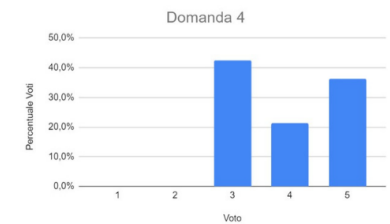


Figura 7.4: Risultati delle prime quattro domande relative alla prova d'ascolto guidata.

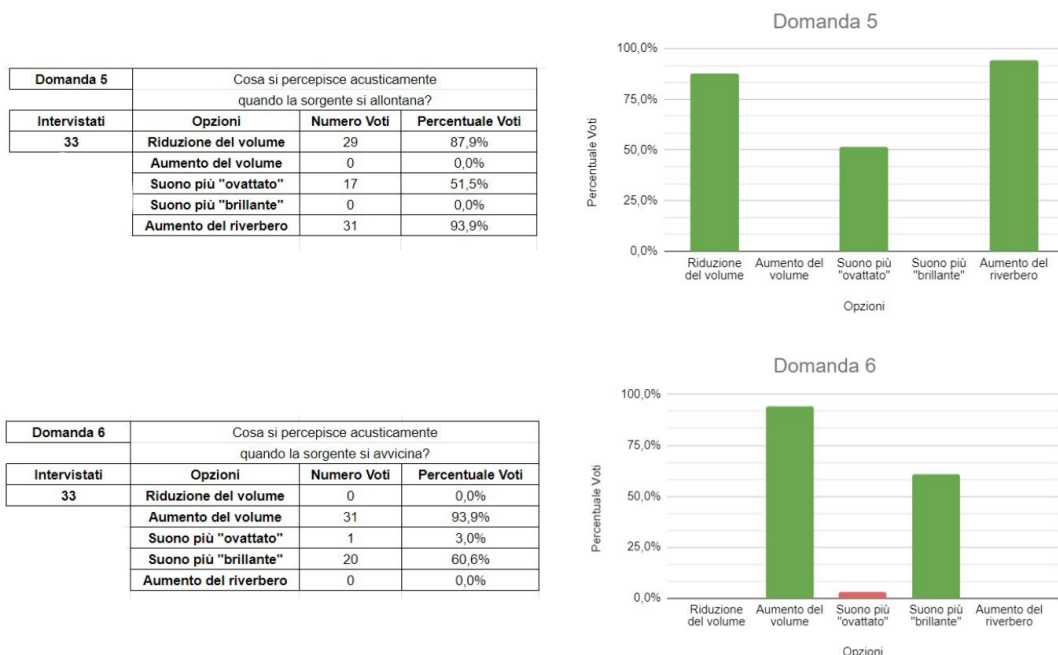


Figura 7.5: Risultati delle ultime due domande relative alla prova d’ascolto guidata.

7.2.2 Risultati seconda prova

Nella prova d’ascolto individuale i risultati sono stati raccolti sottoforma di immagini modificate dagli intervistati. Ognuna di queste contiene una rappresentazione della posizione della sorgente, disegnata sull’interfaccia grafica del Quaso Localization Panner, nei quattro punti diversi in cui viene fatta ascoltare. Tutte le risposte sono state successivamente rappresentate in un’unica immagine cercando di evitare sovrapposizioni tra i risultati. Come si nota in Figura 7.6 e in Figura 7.7, in entrambi i questionari, le due posizioni agli estremi dell’interfaccia sono state percepite con maggiore precisione, mentre quelle intermedie sono caratterizzate da una nuvola di punti più diffusa e imprecisa. Entrambi i risultati sono conformi alle aspettative; in quanto, nel primo caso, il sistema uditivo degli intervistati si è abituato al suono delle posizioni estreme, grazie al loro precedente ascolto nella prova guidata. Nel secondo caso, invece, l’individuazione delle posizioni intermedie risulta più ingannevole. Le piccole differenze di ampiezza tra i due diffusori, infatti, definiscono una posizione orizzontale, che risulta più difficile da attribuire con precisione. Inoltre, come già accennato, la presenza di un riverbero contribuisce in parte alla perdita di informazioni utili alla localizzazione.

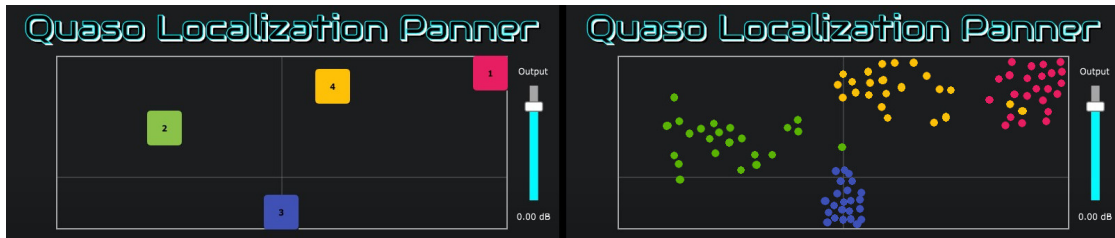


Figura 7.6: A sinistra, sono rappresentate le quattro posizioni della sorgente nella prova d'ascolto individuale del primo questionario. A destra, sono rappresentate le ventitre risposte raccolte.

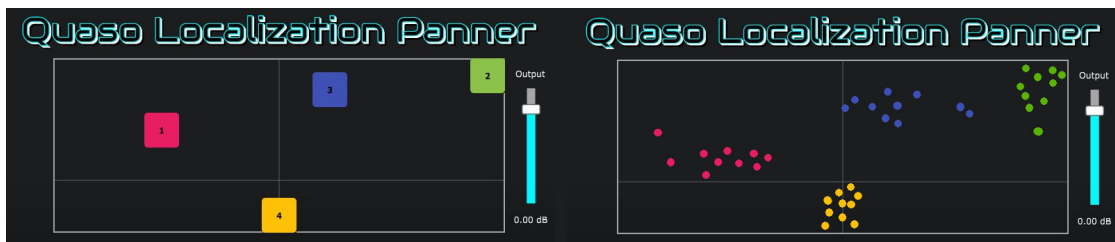


Figura 7.7: A sinistra sono rappresentate le quattro posizioni della sorgente nella prova d'ascolto individuale del secondo questionario. A destra sono rappresentate le dieci risposte raccolte.

Capitolo 8

Conclusioni

La presente tesi si è posta l'intento di realizzare un plugin audio che permetta il posizionamento di sorgenti monofoniche multiple all'interno di uno spazio 2D virtuale. L'obiettivo principale è stato quello di semplificare le molteplici operazioni svolte in tal senso dal tecnico del suono durante le fasi di missaggio, offrendo uno strumento capace di correlare la posizione di ogni elemento presente sull'interfaccia grafica con la posizione che questo occupa all'interno del panorama stereofonico rappresentato e la relativa distanza virtuale dall'ascoltatore. Grazie ai risultati raccolti nella prima prova di indagine, si è concluso che la maggior parte degli intervistati ha percepito in modo evidente i fenomeni di allontanamento, avvicinamento e spostamento laterale della sorgente sonora. Nella seconda prova, invece, gli intervistati sono riusciti ad individuare con buona precisione le quattro posizioni della sorgente sull'interfaccia del plugin, basandosi unicamente sui quattro ascolti associati. I questionari hanno permesso quindi di verificare, da una parte, la corretta modellizzazione dei processi psicoacustici che entrano in gioco durante la localizzazione di una sorgente sonora e, dall'altra, lo stretto legame che intercorre tra posizione sull'interfaccia grafica e posizione nello spazio virtuale ricostruito dal segnale stereofonico così ottenuto. Per quanto riguarda gli aggiornamenti futuri, è prevista un'implementazione che consenta al plugin di supportare anche sorgenti stereofoniche in ingresso, gestendone inoltre l'apertura tra i canali destro e sinistro ed il posizionamento nel panorama totale. Il codice sorgente del Quaso Localization Panner è disponibile su GitHub, come indicato in bibliografia, insieme al file di installazione del VST3 per sistema operativo Windows.

Bibliografia

- [1] sadmemelord. *Quaso Localization Panner Repository on GitHub*. 2023. URL: <https://github.com/sadmemelord/QuasoLocalizationPanner>.
- [2] JUCE. *JUCE Documentation*. 2023. URL: <https://docs.juce.com/>.
- [3] Auburn Sounds. *Panagement 2 User Manual*. 2023. URL: <https://www.auburnsounds.com/downloads/Panagement%20User%20Manual.pdf>.
- [4] Cockos. *REAPER User Guide*. 2023. URL: <https://dlz.reaper.fm/userguide/ReaperUserGuide681c.pdf>.
- [5] JUCE. *JUCE Repository on GitHub*. 2023. URL: <https://github.com/juce-framework/JUCE>.
- [6] Brian CJ Moore. *An introduction to the psychology of hearing*. Brill, 2012.
- [7] William A Yost. *Fundamentals of hearing: An introduction*. Acoustical Society of America, 2001.
- [8] David Miles Huber, Robert E Runstein e David Miles Huber. *Modern recording techniques*. Focal Press/Elsevier Burlington, MA, 2005.
- [9] Tomlinson Holman. *Surround sound: up and running*. CRC Press, 2014.
- [10] F Alton Everest e Ken C Pohlmann. *Master handbook of acoustics*. McGraw-Hill Education, 2022.
- [11] Bruce Bartlett e Jenny Bartlett. *Practical Recording Techniques: The step-by-step approach to professional audio recording*. CRC Press, 2016.
- [12] Vincenzo Lombardo e Andrea Valle. *Audio e multimedia*. Maggioli Editore, 2014.
- [13] Harvey Fletcher e Wilden A Munson. «Loudness, its definition, measurement and calculation». In: *Bell System Technical Journal* 12.4 (1933), pp. 377–430.
- [14] Nicholas J Wade e Diana Deutsch. «Binaural hearing—before and after the stethophone». In: *Acoustics Today* 4.3 (2008), pp. 16–27.

