



UNIVERSITA' POLITECNICA DELLE MARCHE
FACOLTA' DI INGEGNERIA

Corso di Laurea triennale in Ingegneria Elettronica

Studio e sviluppo in ambiente Matlab di un algoritmo di Digital Signal Processing per la generazione di variazioni realistiche di suoni campionati tramite filtri evoluti.

Study and development in Matlab environment of a Digital Signal Processing algorithm for the generation of realistic variations of sampled sounds through advanced filtering.

Relatore: Chiar.mo
Prof. Stefano Squartini

Tesi di Laurea di:
Michele Frati

A.A. 2020 / 2021

ABSTRACT

In questo documento viene presentato un algoritmo, basato su particolari tipologie di filtraggio, che si propone di generare micro-variazioni casuali di suoni campionati a partire da una sola registrazione del suono desiderato.

Il metodo proposto utilizza un VNF (velvet noise filter) breve e pseudo-random in cascata ad un low-shelf filter per produrre variazioni timbriche mirate ad opportune regioni spettrali di singoli suoni di percussioni.

L'implementazione dell'algoritmo è stata effettuata in ambiente di programmazione MATLAB attraverso la creazione di appositi funzioni e script scritti dal programmatore e settati con parametri adatti alla tipologia di suono trattato.

I risultati ottenuti sono stati oggetto di analisi spettrale; inoltre sono stati effettuati dei test di ascolto per valutare se rispettassero o meno la realtà.

La qualità dei suoni ottenuti dal filtraggio risulta essere simile a variazioni ottenute da diverse registrazioni di suoni reali, a volte modellando in modo più che coerente la realtà.

INDICE

CAPITOLO 1: INTRODUZIONE	1
1.1 Stato dell'arte	1
1.2 Descrizione generale del lavoro	2
CAPITOLO 2: VELVET NOISE FILTER	3
2.1 Proprietà del velvet noise	3
2.2 Generazione di velvet noise	5
2.3 Implementazione e VNF	6
CAPITOLO 3 SHELIVING FILTER	8
3.1 Storia dell'equalizzazione	8
3.2 Equalizzatori parametrici	9
3.3 Low-shelf filter di primo ordine	10
CAPITOLO 4: ALGORITMO COMPLETO PER LA CONVERSIONE ONE-TO-MANY DI SUONI PERCUSSIONI	12
4.1 Filtro completo	12
4.2 Campioni di test	14
4.3 Implementazione MATLAB e parametri	15
CAPITOLO 5: ANALISI DEI RISULTATI E CONFRONTO CON PAPER DI RIFERIMENTO	17
5.1 Hi-hat	17
5.2 Snare drum	20
5.3 Tom tom	23
CAPITOLO 6: CONCLUSIONI	26
BIBLIOGRAFIA	27

Capitolo 1

INTRODUZIONE

1.1 Stato dell'arte

[1] Il processo di campionamento è largamente utilizzato come tecnica di sintesi sonora ed applicato non solo in molteplici strumenti, come i sintetizzatori acustici, ma anche in tutti quegli ambiti in cui ci si trova a dover riprodurre artificialmente suoni presi dalla realtà, come batterie elettriche o effetti sonori di videogiochi. In ogni caso la ripetizione consecutiva di un singolo suono risulta all'ascoltatore monotona ed artificiosa ("machine-gun effect").

Esistono numerose tecniche di processazione dei segnali che permettono di modificare profondamente il suono registrato, ma ottenere variazioni sottili e naturali di suoni consecutivi è stato oggetto di molti studi e si è rivelato molto difficoltoso.

Nel mondo dei videogiochi le variazioni desiderate spesso vengono ottenute tramite l'archiviazione di numerose registrazioni dello stesso tipo di suono, ad esempio colpi di armi da fuoco o rumori di passi di persone che camminano o corrono, per poi essere utilizzati alternativamente. Esistono anche tecniche che utilizzano l'effetto morphing per sintetizzare nuovi suoni con minime variazioni, ma hanno come problematica il fatto che bisogna predisporre buona parte della memoria per l'archiviazione di suoni diversi precedentemente registrati.

Per quanto riguarda tecniche che ottengono lo stesso risultato utilizzando un solo suono di partenza si possono citare tecniche di predizione lineare o di sintesi modale. La prima utilizza un filtro di previsione ed agisce riposizionando le eccitazioni del segnale ogni volta in ordine differente; per fare questo occorrono però filtri di ordine molto elevato, oltre $N = 1000$. Ad ogni modo queste tecniche possono lavorare esclusivamente offline, e questo rende notevolmente limitante il loro utilizzo.

1.2 Descrizione generale del lavoro

In questa trattazione viene presentato un algoritmo in grado di fornire una soluzione dal punto di vista dello spazio di archiviazione, che al giorno d'oggi si sa essere molto importante, ed al contempo offre il vantaggio non banale di poter essere utilizzato online. Per queste motivazioni il metodo di conversione presentato viene definito *one-to-many*, ovvero a partire da un singolo suono si possono produrre potenzialmente infinite variazioni del suono stesso, raggiungendo così l'obiettivo di naturalezza che ci si aspetta dall'ascolto di suoni consecutivi dello stesso tipo.

L'algoritmo proposto è principalmente composto da due filtri in cascata: un low-shelf filter ed un VNF basato su un tratto di velvet noise breve e sparso. Entrambi i filtri sono stati progettati per fornire cambiamenti spettrali in linea con variazioni naturali, prese in esame da un set di registrazioni dello stesso suono.

Nei prossimi capitoli vengono presentati:

- Capitolo 2: velvet noise, utilizzo, genesi del rumore ed implementazione del filtro;
- Capitolo 3: shelving filter, storia e funzione di trasferimento;
- Capitolo 4: filtro proposto, analisi del circuito e dei parametri, set di suoni utilizzati ed implementazione MATLAB;
- Capitolo 5: analisi spettrale dei risultati ottenuti e confronto con paper di riferimento;
- Capitolo 6: conclusioni riguardo i risultati ottenuti.

Capitolo 2

VELVET NOISE

2.1 Proprietà del velvet noise

[2] [3] Il velvet noise, introdotto per la prima volta nel 2007 da Karjalainen e Järveläinen, è un segnale sparso e pseudo-random, simile al rumore bianco, che possiede una minima percentuale di campioni diversi da zero. Questa caratteristica fornisce al velvet noise una notevole efficienza nel calcolo della convoluzione nel dominio temporale, da qui la possibilità di lavorare con tempi di latenza nulli. Allo stesso tempo permette di ridurre in modo significativo l'onere computazionale del processo di elaborazione audio, fornendo ulteriori vantaggi anche a livello economico.

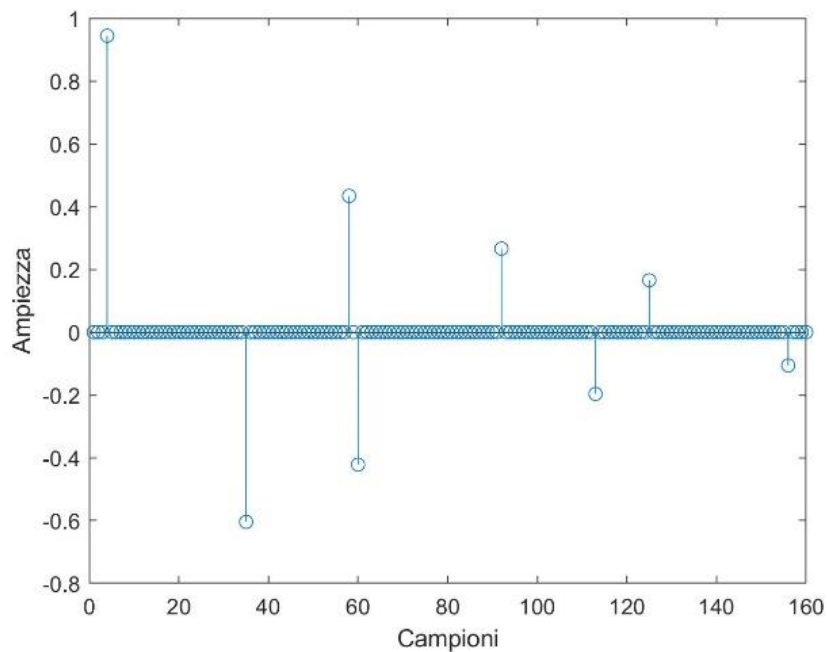


Figura 2.1 Esempio di sequenza di velvet noise con attenuazione esponenziale

Test di ascolto dimostrano che il velvet noise è il rumore percepito come il più uniforme e fluido tra almeno sei diversi algoritmi comparabili con densità di impulso medie pari o inferiori a 2000 impulsi/s, indicando ulteriormente che il velvet noise è un rumore sparso generato artificialmente che risulta realistico e naturale all'ascolto. Ulteriori test di ascolto hanno dimostrato che un velvet noise a banda larga con densità di impulso di circa 2000 impulsi/s suona in maniera più fluida di un rumore bianco gaussiano con lo stesso livello RMS.

2.2 Generazione di velvet noise

[4] Il velvet noise risulta essere un particolare treno di impulsi, dove la maggior parte di questi impulsi sono nulli, mentre quelli diversi da zero assumono segno e posizione casuali. In precedenza questo particolare tipo di rumore era stato definito come pseudo-random, infatti la posizione di ogni impulso diverso da zero non è completamente casuale, ma rimane sempre all'interno di un dato intervallo, definito dalla densità di impulsi desiderata.

In questo modo può essere definito, dati f_s frequenza di campionamento e ρ densità, T_d come distanza media tra due impulsi diversi da zero:

$$T_d = f_s/\rho, \quad (2.1)$$

La posizione di ogni impulso diverso da zero è data da:

$$k(m) = \lfloor mT_d + r_1(m)(T_d - 1) \rfloor, \quad (2.2)$$

dove $m = 0, 1, \dots, M-1$, $\lfloor \cdot \rfloor$ è l'operatore di arrotondamento all'intero più vicino e $r_1(m)$ è un numero casuale con distribuzione uniforme tra 0 e 1.

Il segno di ogni impulso diverso da zero è dato da:

$$s(m) = (2 \lfloor r_2(m) \rfloor - 1), \quad (2.3)$$

dove $m = 0, 1, \dots, M-1$, $\lfloor \cdot \rfloor$ è l'operatore di arrotondamento all'intero più vicino e $r_2(m)$ è un numero casuale con distribuzione uniforme tra 0 e 1.

2.3 Implementazione e VNF

[3] [4] L'implementazione di quanto detto in precedenza può essere descritta dal diagramma a blocchi in figura 2.2:

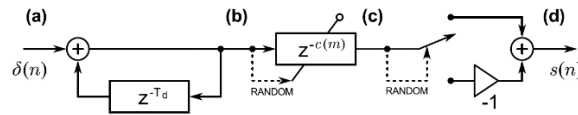


Figura 2.2 Diagramma a blocchi per generare velvet noise

In figura 2.3 viene mostrato, dalla sotto figura 'a' alla 'd', il segnale in ognuna delle quattro posizioni diverse del diagramma a blocchi in figura 2.2. In entrambe si nota che il segnale in ingresso è un singolo impulso (a) che viene fatto passare attraverso un ciclo con ritardo T_d , generando così un treno

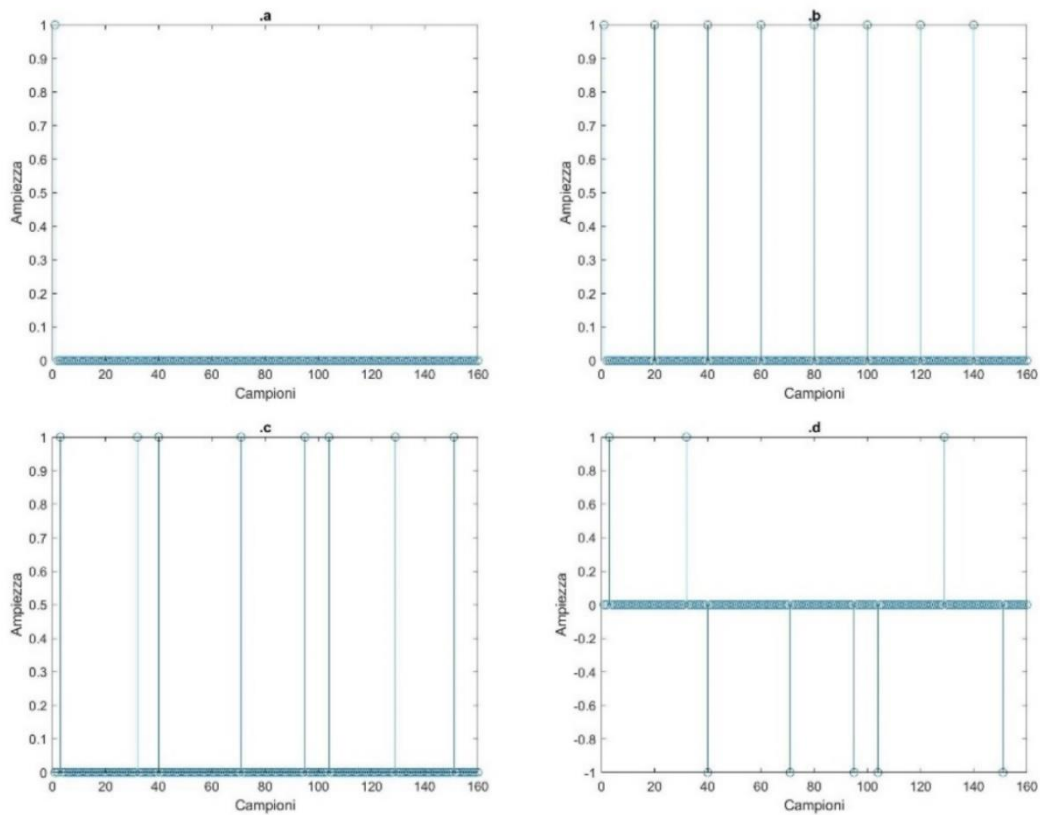


Figura 2.3 Generazione del velvet noise

di impulsi equidistanti (b); proseguendo si entra nella linea con ritardo variabile (c) dato da:

$$c(m) = \lfloor r_1(m)(T_d - 1) \rfloor, \quad (2.4)$$

ottenendo in questo modo la randomizzazione delle posizioni degli impulsi; infine un interruttore casuale (d) fa in modo di cambiare segno ad alcuni impulsi ricavando il segnale finale desiderato.

Il segnale rappresentato in figura 2.3 raffigura un esempio di velvet noise, di lunghezza pari a 160 campioni con una densità media di impulsi, $1/T_d$, di 2205 impulsi/s e frequenza di campionamento a 44.1 kHz; su 160 campioni totali solo 8 sono diversi da zero.

In seguito il segnale appena generato viene attenuato esponenzialmente per prevenire il disturbo del transitorio in fase di ascolto finale:

$$s_e(m) = e^{-\alpha m} s(m), \quad (2.5)$$

dove $\alpha > 0$ è il tasso di decadimento.

In conclusione il velvet noise filter (VNF) risulta:

$$v(n) = \begin{cases} s_e(m) & \text{per } k(m) = n \\ 0 & \text{altrimenti} \end{cases} \quad (2.6)$$

Corrispondente al segnale in figura 2.1.

Il decadimento totale di energia è dato da $L_{dB} = 20 \log_{10} e^{-\alpha M}$.

Utilizzando questo segnale viene creato un VNF (velvet noise filter), ovvero un filtro FIR parametrizzato da f_s frequenza di campionamento, M numero di campioni non nulli del VNF, L_{dB} decadimento totale di energia.

Capitolo 3

SHELVING FILTER

3.1 Storia dell'equalizzazione

[5] [6] [7] [8] Il termine *equalizzatore* (EQ) ha le sue origini nella prima ingegneria telefonica, quando le perdite ad alta frequenza su lunghe distanze dovevano essere corrette in modo che lo spettro del suono al ricevitore corrispondesse allo spettro sonoro inizialmente trasmesso.

Il concetto di filtraggio delle frequenze audio era stato compreso fin dagli anni '70 dell'Ottocento ed era utilizzato nella progettazione dei telegrafi armonici. Attraverso dei tasti l'operatore del telegrafo poteva attivare delle ance elettromeccaniche vibranti, ciascuna assegnata ad una frequenza specifica. Sintonizzando un'ancia simile sulla stessa frequenza l'operatore ricevente poteva sintonizzare il segnale ricevuto, in questo modo si rendeva possibile la trasmissione di piú segnali su una singola linea telegrafica.

L'avvento del suono cinematografico ha visto l'emergere della necessità di un'equalizzazione variabile. In particolare, il design dell'equalizzatore esterno di John Volkman prevedeva una serie di frequenze selezionabili con boost e tagli, ed è talvolta considerato il primo equalizzatore a operatore variabile. I controlli di tono, boost e taglio applicati a alti e bassi, furono introdotti per la prima volta per l'uso con i grammofoni nel 1949. Nel 1952 Baxandall ideò i controlli di tono utilizzando potenziometri anziché interruttori, consentendo così il pieno controllo dell'utente.

Gli equalizzatori audio digitali sono emersi tra la fine degli anni '70 e l'inizio degli anni '80, portando alla diffusione commerciale dell'equalizzatore digitale DEQ7 Yamaha del 1987, il primo equalizzatore parametrico basato su DSP.

3.2 Equalizzatori parametrici

L'equalizzatore parametrico è il più potente e flessibile dei tipi di equalizzatore. Esso consente all'operatore di aggiungere un picco o una tacca (notch) in una posizione arbitraria nello spettro audio. Ad altre frequenze, lontane dal picco o dal notch, l'equalizzatore parametrico non modifica il contenuto spettrale, in quanto la sua risposta in ampiezza è unitaria (0 dB).

L'aggiunta di un picco può essere utile per aiutare uno strumento a essere ascoltato in un mix complesso o per aggiungere deliberatamente una colorazione al suono di uno strumento aumentando o riducendo una particolare gamma di frequenze. Le tacche possono essere utilizzate per attenuare i suoni indesiderati, inclusa la rimozione del ronzio della linea di alimentazione (50 Hz o 60 Hz e talvolta le loro armoniche) e la riduzione del feedback. Per rimuovere gli artefatti senza influenzare il resto del suono, verrebbe utilizzata una larghezza di banda ridotta.

Una singola sezione di un equalizzatore parametrico viene creata da un peaking/notch filter del secondo ordine o, in alcuni casi, da uno shelving filter del primo o del secondo ordine per le bande di frequenza più basse e più alte. Quando vengono utilizzate più sezioni, queste sono sempre collegate in cascata in modo che gli effetti di ciascun sotto filtro siano cumulativi su una scala di decibel.

3.3 Low-shelf filter di primo ordine

Shelving filters di primo ordine vengono utilizzati per creare transizioni fluide (circa 6 dB/ottava) tra le regioni di frequenza interessate e non interessate. In modalità shelving, il guadagno G viene applicato a tutte le frequenze al di sotto o al di sopra di una frequenza di crossover f_c .

Questi sono chiamati rispettivamente low-shelf filter o high-shelf filter e sono talvolta utilizzati nelle bande di frequenza inferiore o superiore.

Per definire la funzione di trasferimento del low-shelf filter si parte dalla considerazione che il filtro debba guadagnare G alle piú basse frequenze (in $z = 1$) e 1 alle frequenze piú alte ($z = -1$), ovvero:

$$H(1) = G, H(-1) = 1. \quad (3.1)$$

Per il guadagno alla frequenza di crossover viene scelto \sqrt{G} , che corrisponde alla media geometrica dei guadagni agli estremi (1 e G) e alla media aritmetica dei guadagni agli estremi in dB. Questo requisito può essere descritto in formule come:

$$|H(e^{j\omega_c})| = \sqrt{G}, \quad (3.2)$$

dove $\omega_c = 2\pi f_c/f_s$ è la frequenza di crossover in radianti, f_s è la frequenza di campionamento, j è l'unità immaginaria.

Il prototipo di low-shelf filter di primo ordine, con crossover a $\pi/2$, avrà una funzione di trasferimento che, dopo essere stata valutata in $\omega_c = \pi/2$ e seguendo il requisito (3.2), dovrà essere della forma

$$H_p(z) = \sqrt{G} \frac{z+p}{z-p}. \quad (3.3)$$

Combinando le equazioni (3.1) e (3.3) è possibile trovare la posizione del polo in funzione del guadagno G :

$$p = \frac{G - \sqrt{G}}{G + \sqrt{G}} \quad (3.4)$$

Sostituendo e applicando una trasformazione lowpass-to-lowpass si ottiene la funzione di trasferimento finale:

$$H_{LS}(z) = \frac{G \tan\left(\frac{\omega_c}{2}\right) + \sqrt{G} + [G \tan\left(\frac{\omega_c}{2}\right) - \sqrt{G}]z^{-1}}{\tan\left(\frac{\omega_c}{2}\right) + \sqrt{G} + [\tan\left(\frac{\omega_c}{2}\right) - \sqrt{G}]z^{-1}}. \quad (3.5)$$

In figura 3.1 viene illustrata la risposta in frequenza di un low-shelf filter di primo ordine con frequenza di crossover a 100 Hz e guadagno $G = -20$ dB.

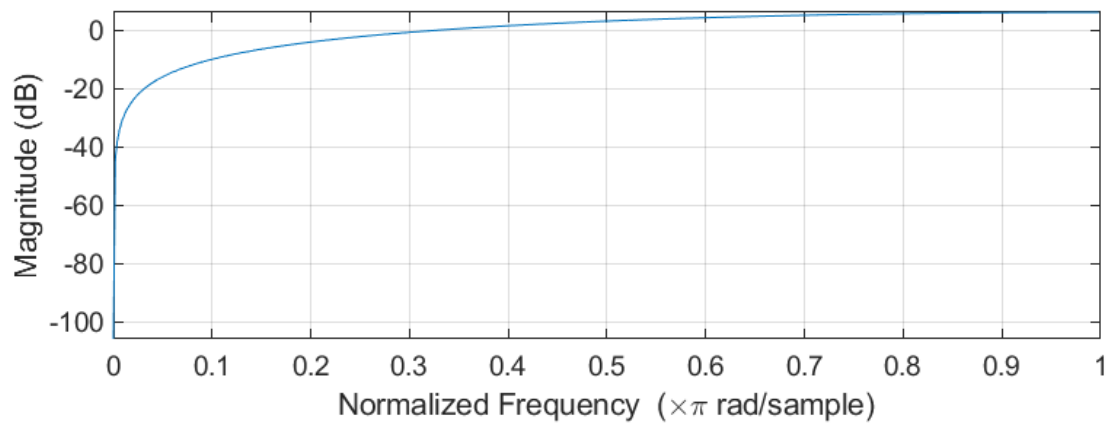


Figura 3.1 Risposta in frequenza di un low shelf filter con frequenza di crossover a 100Hz e guadagno $G = -20$ dB

Capitolo 4

ALGORITMO COMPLETO PER LA CONVERSIONE ONE-TO-MANY DI SUONI PERCUSSIONI

4.1 Filtro completo

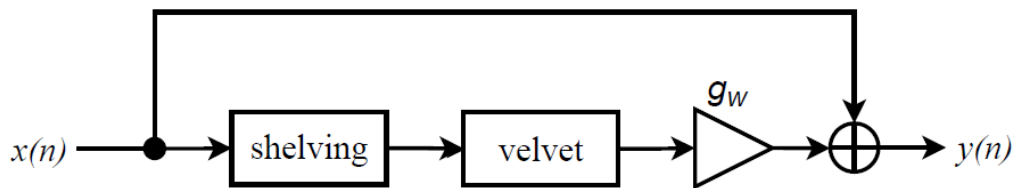


Figura 4.1 Schema a blocchi del filtro proposto

Nella figura 4.1 viene illustrato lo schema a blocchi rappresentante la struttura completa del filtro proposto per generare variazioni. Il filtro è composto da due rami: un ramo diretto e un ramo definito “filtrato”. Il ramo filtrato è il ramo che introduce le variazioni desiderate, e contiene il low-shelf filter del primo ordine in cascata al VNF (velvet noise filter); inoltre è stato inserito un guadagno g_w per fare in modo di controllare l’ampiezza delle variazioni introdotte in uscita.

Visto che il VNF produce un’ampia gamma di variazioni, soprattutto in bassa frequenza, viene utilizzato a monte il low-shelf filter per fare in modo di concentrare le variazioni nelle frequenze desiderate, ovvero nella banda passante del low-shelf filter.

Il ramo diretto somma al segnale filtrato il segnale originale ed il suo contributo è visibile in figura 4.2, dove viene illustrata una risposta impulsiva del filtro proposto.

In $t = 0$ è evidente l’effetto del ramo diretto, mentre subito dopo entrano in gioco gli effetti del ramo filtrato.

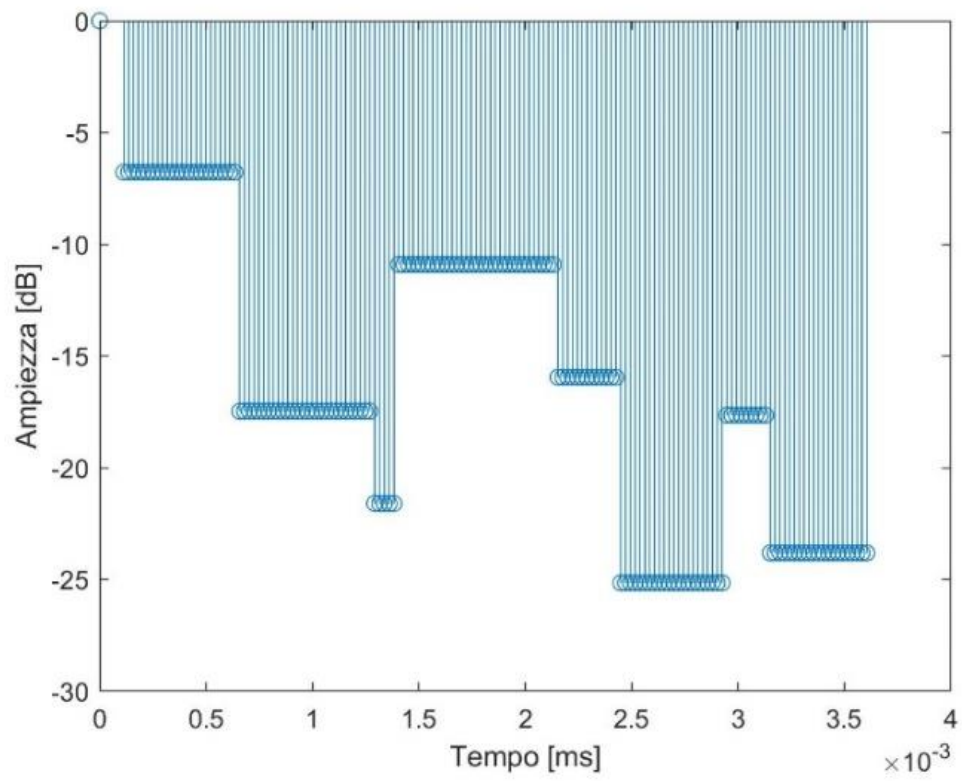


Figura 4.2 Risposta impulsiva del filtro proposto completo

4.2 Campioni di test

Il filtro proposto è stato testato su tre diverse tipologie di suoni di batteria: il charleston (hihat), il rullante (snare) ed il tomtom.

Nel paper preso come riferimento [1] i suoni delle percussioni sono stati registrati dal vivo, mentre dei musicisti professionisti suonavano gli strumenti; invece in questo studio per ogni tipo di suono sono state prese 7 diverse registrazioni da archivi online [9] [10] e le variazioni di ogni set di registrazioni sono state usate come termine reale nella comparazione con le variazioni ottenute dal filtraggio.

Per quanto riguarda le variazioni generate dal filtro sono state generati 50 diversi suoni per ogni tipologia, in modo da ottenere un risultato generale abbastanza per poter valutare oggettivamente la veridicità del lavoro svolto.

Per l'analisi spettrale i risultati ottenuti sono stati filtrati tramite un banco filtri di Bark, in modo da rendere la comparazione più semplice agevole. In figura 4.3 viene mostrata la risposta in frequenza del banco filtri utilizzato.

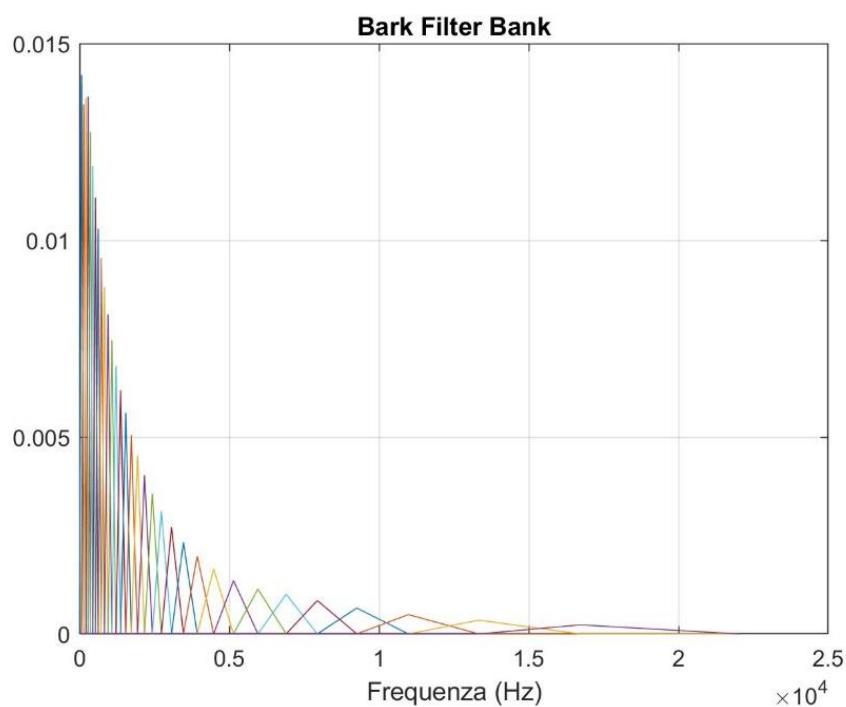


Figura 4.3 Risposta in frequenza del banco filtri di Bark

4.3 Implementazione MATLAB e parametri

Lo script Matlab che è stato creato per l'implementazione di questo filtro è diviso in più parti:

- Vengono inizializzate variabili, matrici e viene preso in lettura il file audio da cui si parte;
- Tramite un ciclo for viene lanciata la funzione `segnalefinale.m` che prende in input:
 - Audio -> traccia audio da elaborare
 - F_s -> frequenza di campionamento
 - M -> numero di campioni non nulli desiderati nel VNF
 - L_{dB} -> attenuazione totale di energia del VNF
 - F_c -> frequenza di taglio del low shelf filter
 - G -> guadagno del low shelf filter
 - G_w -> guadagno di amplificazione nel ramo filtrato

La funzione `segnalefinale.m` restituisce come output il segnale audio filtrato;

quindi viene creata una matrice in cui ogni colonna rappresenta un segnale differente.

- Vengono calcolati media (nero), massimo (blu), minimo (rosso) e deviazione standard (verde) per ogni frequenza di Bark ed ognuna di queste viene plottata in scala semilogaritmica.

Nella tabella di figura 4.4 vengono mostrati i parametri utilizzati per settare il filtro proposto:

	HIHAT	SNARE	TOMTOM
f_s	44.1 kHz	44.1 kHz	44.1 kHz
M	8	8	8
f_c	50 Hz	100 Hz	75 Hz
G_w	0.5	0.2	0.2
G	-20	-5	-5
L_{dB}	20	20	20

Figura 4.4 Parametri per la configurazione del filtro per ogni tipologia di suono

Capitolo 5

ANALISI DEI RISULTATI E CONFRONTO CON PAPER DI RIFERIMENTO

In questo capitolo viene presentata l'analisi spettrale dei risultati ottenuti, inoltre viene effettuato un confronto qualitativo con i risultati mostrati nel paper [1] utilizzato come riferimento in questo lavoro.

5.1 Hi-Hat

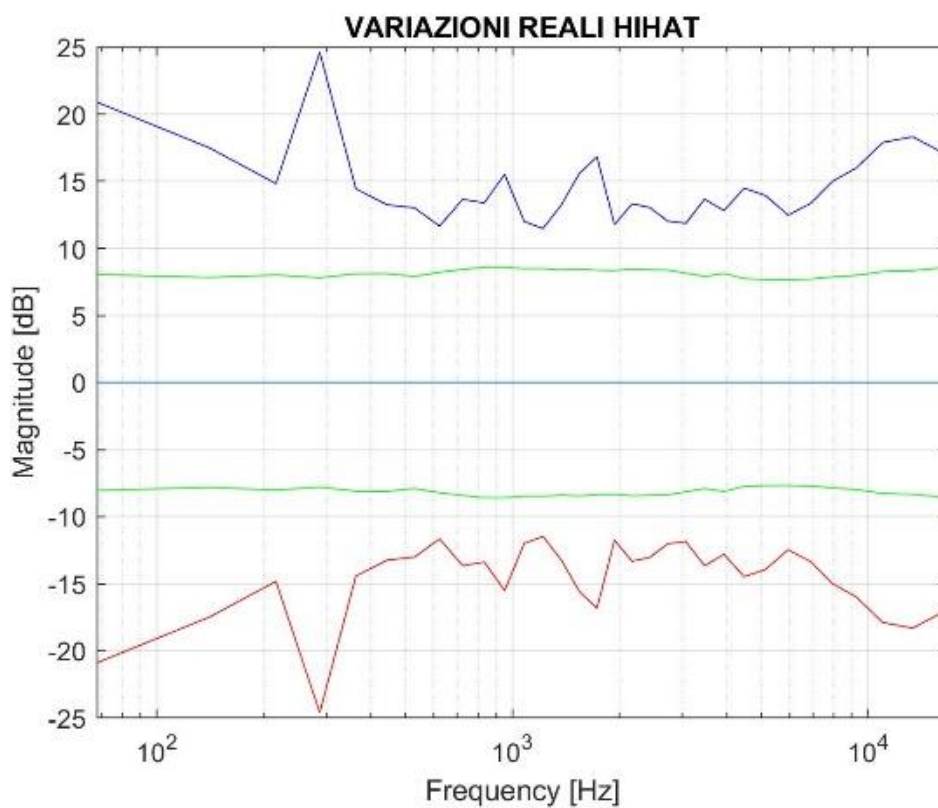


Figura 5.1.1 Analisi statistica dello spettro di ampiezza di suoni di charleston in cui vengono mostrate le variazioni reali. Vengono riportati massimo (blu), minimo (rosso), media (azzurro) e deviazione standard (verde).

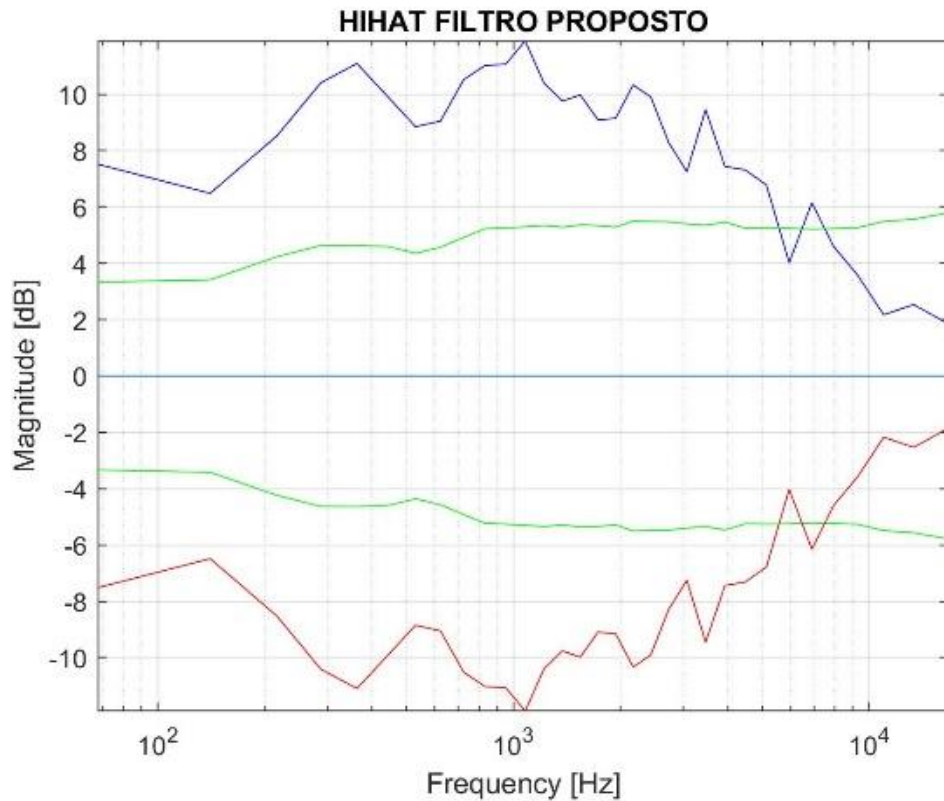


Figura 5.1.2 Analisi statistica dello spettro di ampiezza di suoni di charleston in cui vengono mostrate le variazioni ottenute dal filtraggio tramite l'algorithm proposto. Vengono riportati massimo (blu), minimo (rosso), media (azzurro) e deviazione standard (verde).

Le variazioni spettrali generate dal filtraggio proposto vengono messe a confronto con le variazioni presenti nei campioni audio reali di hi-hat.

Nelle figure 5.1.1 e 5.1.2 si nota come il filtro riesca a modellare in modo coerente le variazioni reali a medie frequenze, nell' intervallo di circa 400 Hz ÷ 5 kHz; per frequenze piú basse di 400 Hz mancano alcuni dettagli ed è evidente una perdita di potenza, mentre per frequenze superiori a 5 kHz il filtro perde di precisione mancando completamente armoniche presenti nella realtà.

In figura 5.1.3 viene riportata la stessa analisi condotta nel paper di riferimento: nella sotto figura (a) si osservano variazioni reali, nella (b) variazioni ottenute dal filtraggio tramite l'algorithm studiato.

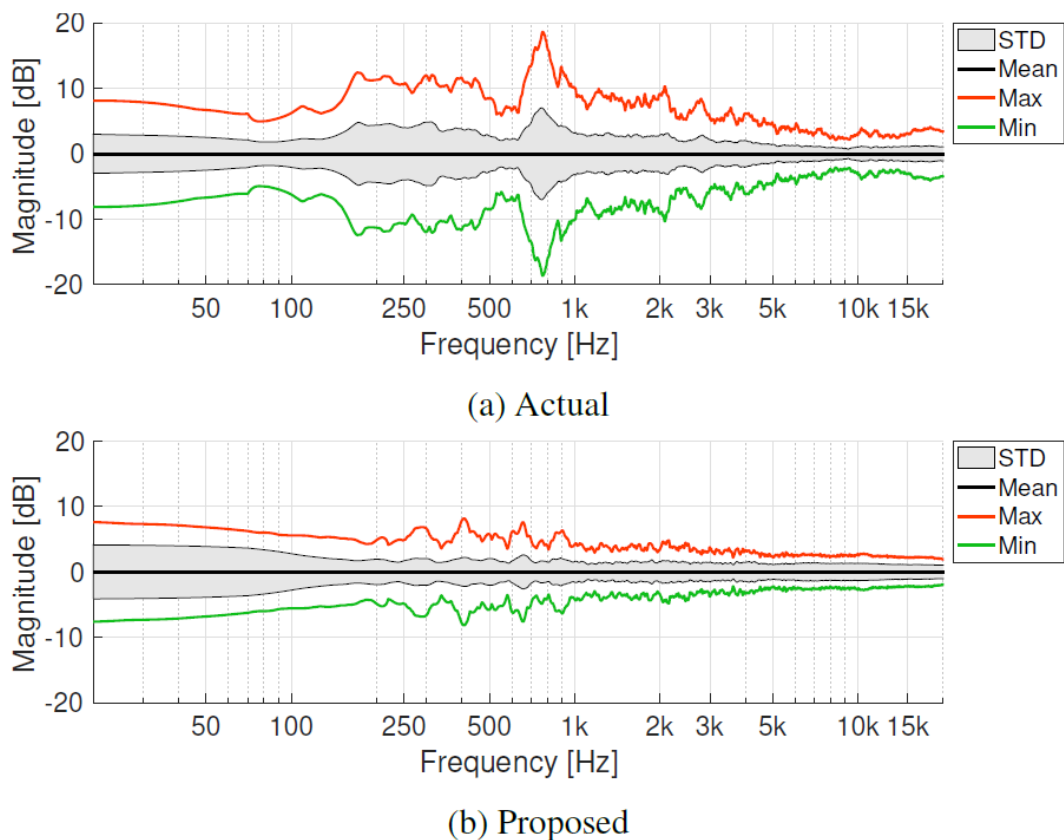


Figura 5.1.3 Analisi statistica dello spettro di ampiezza di suoni di charleston riportata nel paper di riferimento, utilizzata per il confronto dei risultati

Le considerazioni fatte in precedenza sono in linea con quello che si può notare analizzando questi risultati; anche in questo caso il filtro modella in modo coerente le variazioni a partire dai 300/400 Hz fino a qualche migliaio di Hz, per poi perdere di precisione. In questo caso il filtro non riesce a modellare bene le armoniche posizionate intorno agli 800 Hz dovute ad una forte risonanza.

5.2 Snare Drum

Per quanto riguarda l'analisi del secondo tipo di suono di percussione, lo snare drum (rullante), vengono riportate nelle figure 5.2.1 e 5.2.2 le analisi statistiche dello spettro di ampiezza.

Osservando le variazioni ottenute da suoni reali salta all'occhio una notevole presenza di informazione a bassa frequenza, fino a 500/600 Hz, che il filtro proposto in questo studio non riesce a seguire in maniera adeguata.

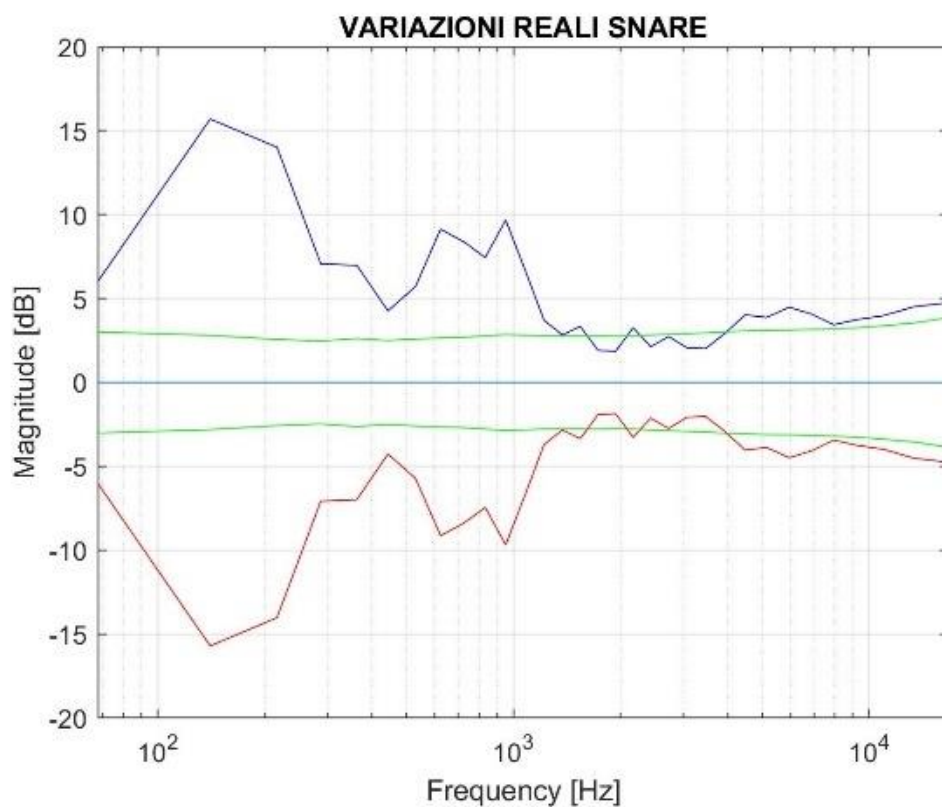


Figura 5.2.1 Analisi statistica dello spettro di ampiezza di suoni di rullante in cui vengono mostrate le variazioni reali. Vengono riportati massimo (blu), minimo (rosso), media (azzurro) e deviazione standard (verde).

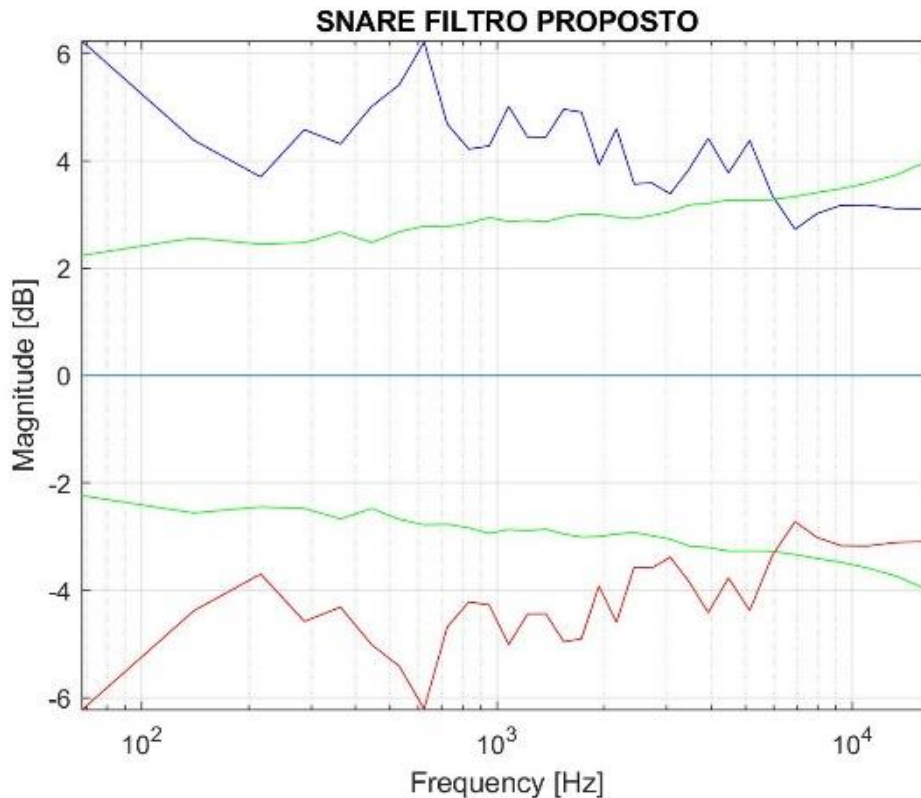
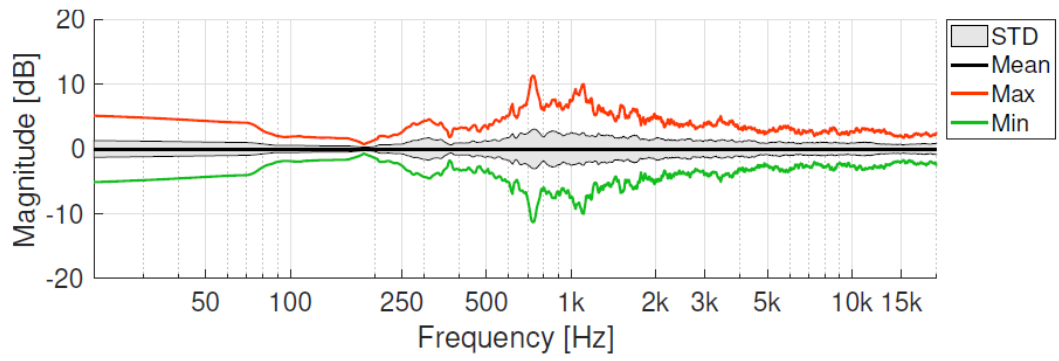


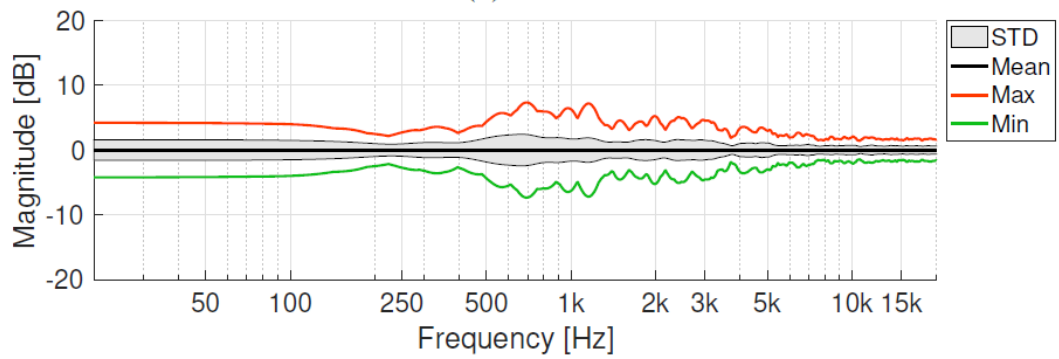
Figura 5.2.2 Analisi statistica dello spettro di ampiezza di suoni di rullante in cui vengono mostrate le variazioni ottenute dal filtraggio tramite l'algoritmo proposto. Vengono riportati massimo (blu), minimo (rosso), media (azzurro) e deviazione standard (verde).

Al contrario, per frequenze più alte viene riscontrata una maggiore coerenza sia a livello di potenza sia a livello di dettagli. Nonostante questo, anche in questo caso l'algoritmo non riesce a replicare l'andamento reale in presenza di risonanze importanti, come si può notare attorno ai 1000 Hz.

I risultati ottenuti nel paper preso come riferimento in figura 5.2.3 mettono alla luce problematiche alle basse frequenze, ma con differenze ridotte rispetto alla situazione mostrata nelle figure 5.2.1 e 5.2.2. Nonostante questo l'andamento reale viene ben modellato dal filtro, seppur con scarsa qualità nei dettagli, mantenendo però le stesse mancanze nelle situazioni in cui è presente risonanza.



(a) Actual



(b) Proposed

Figura 5.2.3 Analisi statistica dello spettro di ampiezza di suoni di rullante riportata nel paper di riferimento, utilizzata per il confronto dei risultati

5.3 Tom Tom

Le variazioni reali e derivate dal filtraggio di un singolo suono di tom sono riportate nelle figure 5.3.1 e 5.3.2.

Analizzando le variazioni reali ci si accorge facilmente che, al contrario delle altre due tipologie di percussioni, in questo caso la maggior parte delle armoniche sono concentrate a frequenze medio-alte. Questa peculiarità non viene evidenziata nell'analisi spettrale delle variazioni generate dall'algoritmo che, sebbene riesca a modellare in modo abbastanza coerente l'andamento fino a circa 5 kHz, ad alte frequenze non risulta essere fedele alla realtà come dovrebbe.

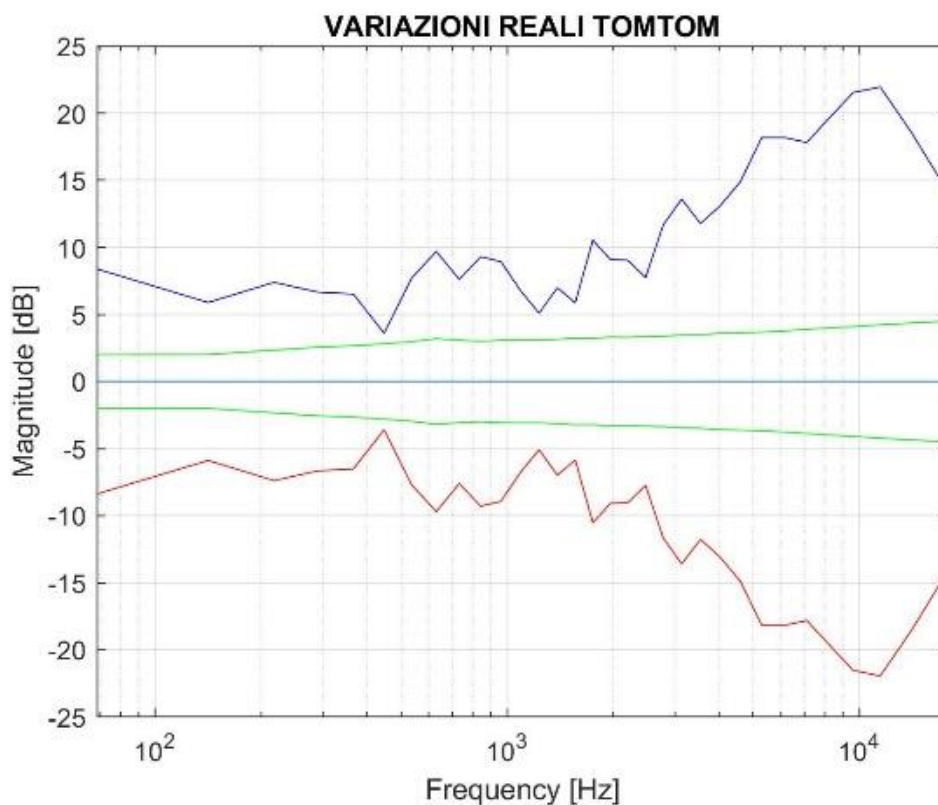


Figura 5.3.1 Analisi statistica dello spettro di ampiezza di suoni di tom tom in cui vengono mostrate le variazioni reali. Vengono riportati massimo (blu), minimo (rosso), media (azzurro) e deviazione standard (verde).

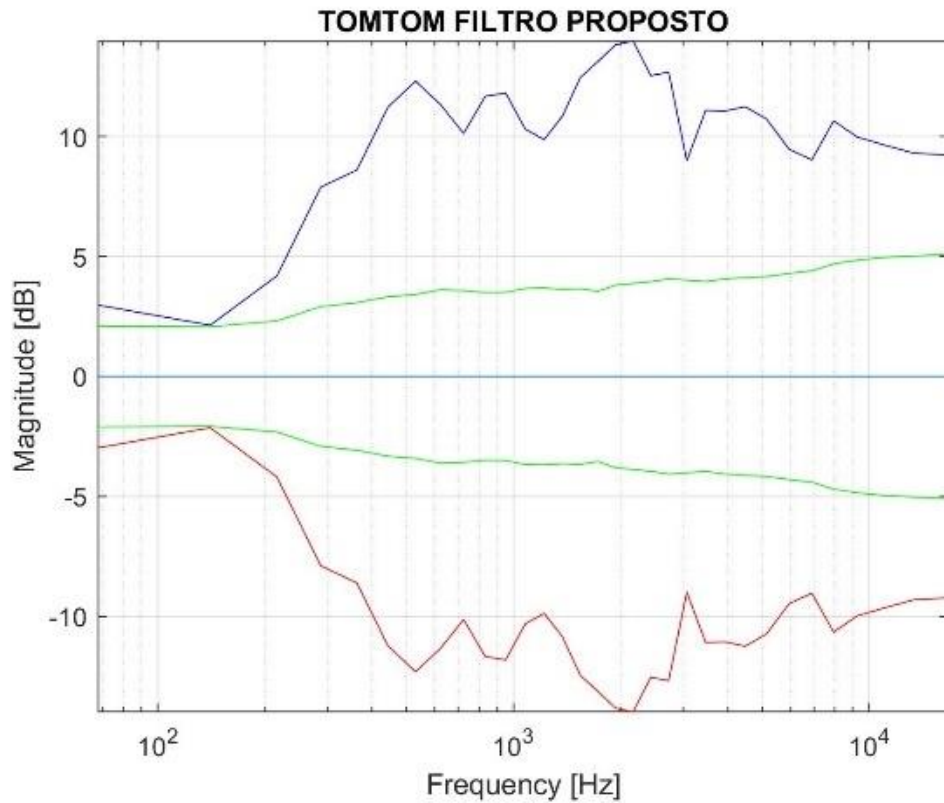


Figura 5.3.2 Analisi statistica dello spettro di ampiezza di suoni di rullante in cui vengono mostrate le variazioni ottenute dal filtraggio tramite l'algoritmo proposto. Vengono riportati massimo (blu), minimo (rosso), media (azzurro) e deviazione standard (verde).

In figura 5.3.3 vengono riportate le analisi spettrali presenti sul paper di riferimento: anche in questo caso possiamo notare che il filtro modella in modo adeguato le variazioni fino a circa 6 kHz, per poi perdere di precisione e di potenza del segnale.

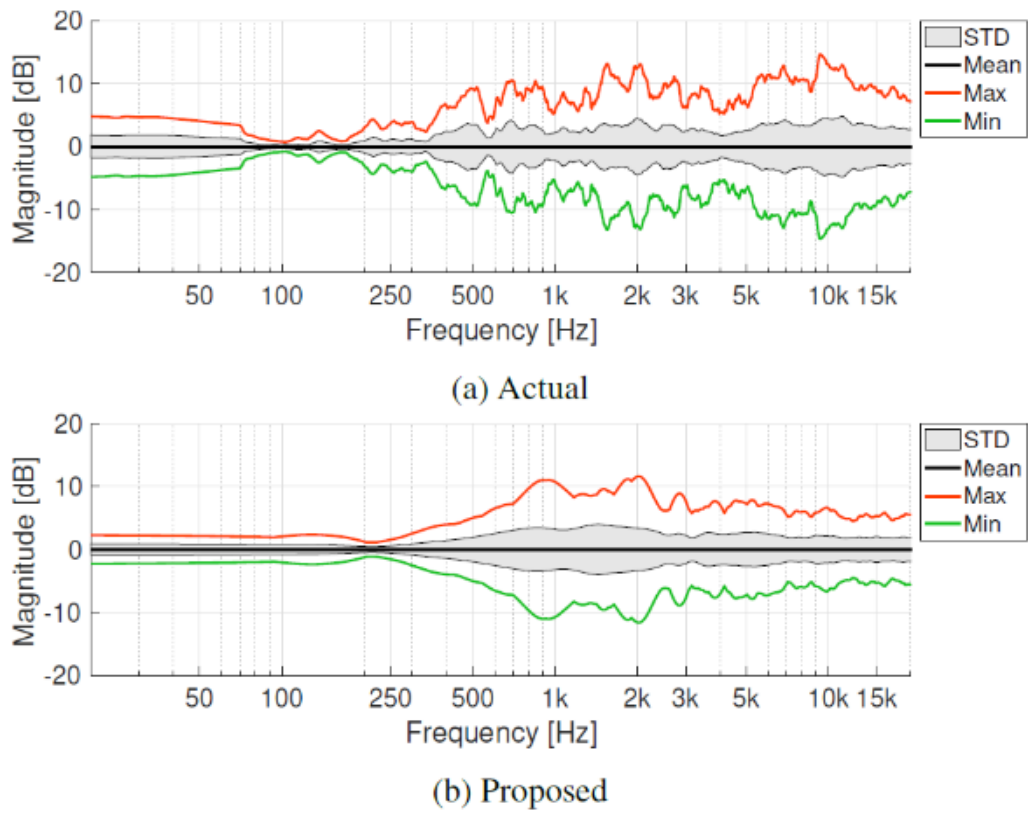


Figura 5.3.3 Analisi statistica dello spettro di ampiezza di suoni di rullante riportata nel paper di riferimento, utilizzata per il confronto dei risultati

Capitolo 6

CONCLUSIONI

In questo studio è stato presentato un algoritmo per la creazione di variazioni realistiche di uno singolo campione di percussione.

Viene utilizzato un filtro che si basa sull'utilizzo di un VNF in serie ad un low shelf filter di primo ordine, per creare variazioni casuali e posizzionarle alle frequenze desiderate. Il segnale filtrato viene attenuato e sommato a quello originale.

L'algoritmo viene testato su 3 diversi tipi di percussioni e per ogni tipo vengono usate le variazioni reali come comparazione del risultato ottenuto.

Il risultato ottenuto è da considerarsi discreto; nonostante non sempre si sia raggiunto un buon livello di dettaglio, i segnali dopo il filtraggio rimangono abbastanza fedeli alla realtà.

Inoltre ripetendo consecutivamente l'ascolto dei suoni generati a partire dallo stesso campione si nota un maggiore realismo rispetto alla ripetizione del campione stesso non filtrato.

Infine l'algoritmo si presta ad un semplice uso e modellazione in base ai suoni da elaborare avendo pochi parametri da settare.

BIBLIOGRAFIA

- [1] J. Fagerstrom, S. J. Schlecht, V. Valimaki. "One-to-many conversion for percussive samples" Proceedings of the 24th International Conference on Digital Audio Effects (DAFx20in21), Vienna, Austria, September 8-10, 2021
- [2] M. Karjalainen and H. Jarvelainen, "Reverberation modeling using velvet noise," in Proc. Audio Eng. Soc. 30th Int. Conf. Intell. Audio Environ., Saariselka, Finland, Mar. 2007.
- [3] B. Holm-Rasmussen, H.-M. Lehtonen, and V. Valimaki, "A new reverberator based on variable sparsity convolution," in Proc. Int. Conf. Digital Audio Effects (DAFx-13), Maynooth, Ireland, Sept. 2013, pp. 344-350.
- [4] V. Välimäki, H.-M. Lehtonen, and M. Takanen, "A perceptual study on velvet noise and its variants at different pulse densities," IEEE Transactions on Audio, Speech, and Language Processing, vol. 21, no. 7, pp. 1481–1488, July 2013.
- [5] Lundheim, L. On Shannon and Shannon's formula. *Teletronikk* 2002, 98, 20–29.
- [6] V. Valimaki and J. Reiss, "All about audio equalization: Solutions and frontiers," *Applied Sciences*, vol. 6, no. 5, May 2016, paper 129.
- [7] Williamson, D.T.N. Design of tone controls and auxiliary gramophone circuits. *WirelessWorld* 1949, 55, 20–29.
- [8] Berkovitz, R. Digital equalization of audio signals. In Proceedings of the Audio Engineering Society 1st International Conference on Digital Audio, Rye, NY, USA, 3–6 June 1982.
- [9] <https://www.fesliyanstudios.com/royalty-free-sound-effects-download/tom-tom-drums-275>
- [10] <https://www.musicradar.com/news/drums/1000-free-drum-samples>

A conclusione di questo elaborato, desidero menzionare tutte le persone senza le quali questo lavoro di tesi non esisterebbe nemmeno.

Ringrazio il mio relatore Stefano Squartini, che in questi mesi di lavoro è stato sempre presente, puntuale e disponibile. Grazie al percorso intrapreso insieme ho sviluppato maggiormente la mia capacità di analisi e di problem solving.

Ringrazio di cuore la mia famiglia. Grazie per avermi sempre sostenuto e per avermi permesso di portare a termine questo percorso di studio. Un abbraccio speciale va a nonna Maria per aver sempre creduto in me.

Ringrazio la mia ragazza Lucia per avermi trasmesso la sua immensa forza e il suo coraggio. Grazie per tutto il tempo che mi hai dedicato. Grazie perché ci sei sempre stata.

Ringrazio tutti i miei amici, in particolare tutti i ragazzi de "La Villa". Grazie per essere stati sempre presenti anche durante i momenti più difficili e grazie per tutti i momenti di spensieratezza. Vorrei ringraziare inoltre tutti i miei compagni di squadra e colleghi di lavoro con cui ho avuto il piacere di condividere momenti davvero importanti.

Un ringraziamento speciale va a tutti i colleghi conosciuti durante questo percorso, che hanno condiviso con me gioie e fatiche di tutti questi anni. In particolare Giorgio, Angelo e Leonardo, abbiamo affrontato e vinto molte sfide insieme.

Infine, vorrei dedicare questo piccolo traguardo a me stesso, che possa essere l'inizio di un lungo e brillante percorso.

