ARCHEOTOPOLOGIE: IMPLEMENTAZIONE CRITICA DI MEMORIE SENZA COLORE

Conference Paper · December 2024

CITATION
1

READS
15

5 authors, including:

Francesco Vitucci
Conservatorio "Niccolò Piccinni" di Bari
2 PUBLICATIONS **1** CITATION
SEE PROFILE

Anthony Di Furia
Conservatorio Niccolò Piccinni di Bari
2 PUBLICATIONS **1** CITATION
SEE PROFILE

Francesco Scagliola
Conservatorio di Musica "Niccolò Piccinni", Italy, Bari
11 PUBLICATIONS **4** CITATIONS
SEE PROFILE

Giuseppe Silvi
Conservatorio N. Piccinni di Bari
7 PUBLICATIONS **2** CITATIONS
SEE PROFILE

ARCHEOTOPOLOGIE: IMPLEMENTAZIONE CRITICA DI MEMORIE SENZA COLORE

Daniele Giuseppe Annese
Conservatorio "N. Piccinni" di Bari
dan.annese1@gmail.com

Francesco Vitucci
Conservatorio "N. Piccinni" di Bari
francescovitucci1@gmail.com

Anthony Di Furia
Conservatorio "N. Piccinni" di Bari
anthonydifuria.sound@gmail.com

Francesco Scagliola
Conservatorio "N. Piccinni" di Bari
francesco.scagliola@gmail.com

Giuseppe Silvi
Conservatorio "N. Piccinni" di Bari
grammaton@me.com

ABSTRACT

Nello studio della musica elettronica l'osservazione critica di algoritmi e architetture informatiche archetipiche è un esercizio tecnico, stilistico, paragonabile allo studio di passi del repertorio per lo strumento musicale acustico. Nello specifico, si esercitano capacità di elaborazione del segnale a fini didattici e di sostenibilità del patrimonio musicale e culturale.

La letteratura di riferimento a cui si attinge è quella relativa alla nascita delle tecniche di riverberazione digitale mediante costruzione e relazioni di filtri. I testi di Manfred R. Schroeder rappresentano momenti di approfondimento matematico e fisico, la cui ri-costruzione offre inevitabili riflessioni di carattere scientifico, informatico e musicale.

L'implementazione mediante linguaggi informatici testuali e opportuni modelli di analisi del segnale, consente l'accesso alla letteratura storica in una chiave di comprensione contemporanea, garantisce continuità futura del testo e apre prospettive di costruzione algoritmica nuova.

1. INTRODUZIONE

Ad un approccio superficiale, considerato l'impiego di strumenti tecnologici nati nel secondo novecento, può accadere che la letteratura informatica ed elettroacustica siano ritenute molto più giovani della letteratura in fisica acustica. Tuttavia, nonostante l'attenzione architettonica a fenomeni sonori sia stata per millenni parte integrante delle capacità progettuali di strutture, la fisica acustica ha ottenuto una base scientifica solida solo dai primi anni del novecento, ad opera di Wallace Sabine. [1]

L'osservazione dei fenomeni di riflessione e riverberazione, la loro descrizione scientifica, fino alla definizione controllata dei tempi di riverberazione da parte di Sabine sono punti di partenza anche nella letteratura sulla modellazione digitale del riverberatore, fin dal principio, nell'opera di Manfred Schroeder. [2, 3] A tale proposito è stata realizzata una traduzione integrale del testo *Natural Sounding Artificial Reverberation* [1] [2] al quale il presente articolo fa riferimento per una implementazione critica che aiuti a comprendere la specificità dell'intervento di Schroeder. I due testi fanno da supporto didattico per la costruzione passo passo dei riverberatori con strumenti di analisi e grafici da confrontare con quelli proposti dall'autore e mirano alla riscrittura e all'ampliamento dei processi per dare vita a nuovi strumenti musicali. [4]

Come per Sabine, chiamato a risolvere problematiche di carattere acustico, Schroeder rende possibile un avanzamento scientifico legato al riverbero approcciando alla soluzione di alcuni problemi di stabilità e linearità in frequenza in relazione ai riverberi elettronici disponibili all'epoca. È consapevole delle criticità, che mette in chiaro fin dal principio: la diffusione di un riverbero necessita di un numero minimo di 1000 echi per secondo per non esprimere fastidiose fluttuazioni. Inoltre questa diffusione deve avvenire senza danneggiare il contributo timbrico della sorgente, cosa che accadeva con i riverberi dell'epoca:

> 1. Le loro *risposte in ampiezza e in frequenza* non sono piatte. Infatti, deviano da una risposta piatta cosi tanto che è possibile percepire una sgradevole "colorazione". [...]
> 2. La *densità degli echi* (*i.e.*, il numero di echi al secondo che l'output del riverberatore genera quando viene stimolato da un singolo impulso all'input) è troppo bassa comparata con la densità degli echi di un ambiente acustico reale. [...]

L'implementazione segue la struttura narrativa del testo di Schroeder costruendo i due blocchi basilari *Comb Filter* e *All-Pass Filter* e le successive elaborazioni ed interrelazioni tra questi. La scrittura su *Faust* delle funzioni consente una documentazione mediante diagramma a blocchi e l'analisi immediata della risposta all'impulso nei domini di tempo e frequenza. Il confronto di questi dati, con i materiali di analisi forniti dall'autore insieme alla documentazione matematica, costruiscono una visione ampia e robusta sui problemi di implementazione *samplerate*.

[1] https://github.com/s-e-a-m/2024-CIM-MSNSAR/blob/main/rsrc/msnsar-it/build/2024-msnsar-it.pdf - Daniele Giuseppe Annese, 2024.

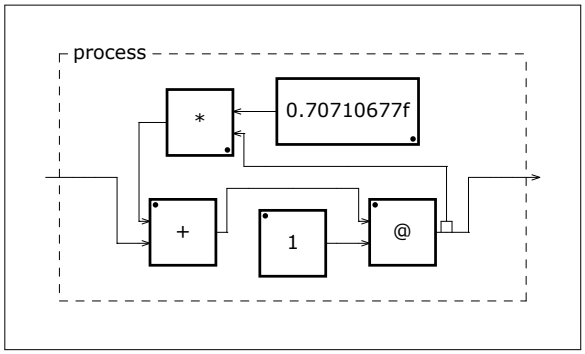

Figure 1. Diagramma a blocchi generato con la funzione `dfl` in *Faust*, apparentemente identica a quella proposta da Schroeder.

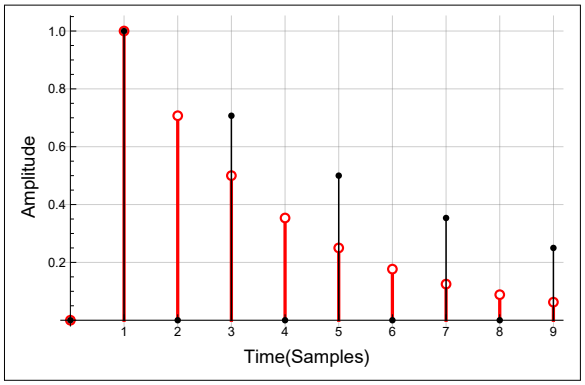

Figure 2. Comparazione tra le risposte all'impulso della funzione `dfl` in *Wolfram Mathematica*. Il diagramma in rosso rappresenta la risposta originale di Schroeder, caratterizzata da riscalamenti dell'impulso a periodo t, quello in nero rappresenta la risposta generata dal codice *Faust*, in cui i riscalamenti avvengono a periodo $2t$.

2. IMPLEMENTAZIONE CRITICA DEI COMPONENTI PRINCIPALI

L'articolo [2] propone un'analisi matematica dei processi e descrive implementazioni informatiche *archetipiche* mediante diagrammi a blocchi e stampe di risposte all'impulso. *Faust (Functional Audio Stream)* [2] è un linguaggio di programmazione funzionale per la sintesi del suono e l'elaborazione audio. La sua sintassi consente di scrivere algoritmi nel dominio del tempo che vengono rappresentati come diagrammi a blocchi. La scrittura in ambienti *samplerate* come *Faust* produce quindi diagrammi immediatamente confrontabili con gli orginali ma deve tener conto di problematiche di elaborazione numerica esperibili solo con un metodo di analisi dei risultati confrontabile con quello proposto dall'autore. A tal proposito si è scelto di elaborare i dati delle risposte all'impulso provenienti da *Faust* nell'ambiente di calcolo numerico *Wolfram Mathematica* [3] al fine di ottenere rappresentazioni grafiche chiare e comprensibili. Ciascuna delle funzioni implementate è disponibile nell'omonima libreria *Faust* di *SEAM - Sustainable Electro-Acoustic Music* [4], progetto fondato sul concetto di sostenibilità della pratica musicale elettroacustica dal vivo con l'idea di far crescere la consapevolezza dei problemi implementativi elettronici e informatici che questa pratica ha affrontato nel tempo.

2.1 Delay in Feedback Loop

Il percorso di costruzione di un sistema di riverberazione in grado di dare densità alle riflessioni e risposta timbrica a magnitudine lineare parte, per Manfred Schroeder, da una linea di ritardo, capace di restituire un singolo eco dopo un tempo di ritardo t. Un oggetto semplice, per cui ancora insufficiente a soddisfare gli obiettivi desiderati.

> Per produrre echi multipli senza usare più delay (costosi), la linea di ritardo è inserita in un ciclo di feedback [...], con gain g, minore di uno (in maniera tale da mantenere stabile il ciclo).

[2] `https://faust.grame.fr`
[3] `https://www.wolfram.com/mathematica/`
[4] `https://github.com/s-e-a-m/faust-libraries`

La topologia che Schroeder descrive in queste righe è quella del più semplice filtro a struttura ricorsiva *IIR*. Il cuore attorno a cui ruota la linea di retroazione è un ritardo variabile, che cambia il comportamento temporale e quindi spettrale del filtro in funzione dell'unità di ritardo.

Schroeder battezza da subito questa struttura, dichiarandone l'individualità: *Delay in Feedback Loop* è il mattone elementare su cui si poggeranno tutte le architetture riverberanti successive.

```
dfl(t,g) = (+ : @(t))~*(g);
```

Il codice *Faust* produce il diagramma a blocchi in Figura 1. Questo, coerente con il modello di Schroeder, può essere smontato e descritto in più parti per comprenderne la sintassi. Il blocco di ritardo è realizzato utilizzando la funzione `@`, una primitiva di *Faust* che esprime un tempo di ritardo t, intero e positivo, in quanto unità al campione. L'uscita del blocco di ritardo viene prelevata prima dell'uscita dal filtro mediante la composizione ricorsiva $\sim$ e reindirizzata alla sua entrata, in somma con l'entrata del filtro; questo circuito di retroazione è controllato dal coefficiente $g < 1$. Il filtro prevede un valore di ritardo $t >= 1$ applicato a tutti i campioni in entrata, situazione che può produrre diversi risultati sonori. La Figura 2 compara, sovrapponendole, le due risposte all'impulso: quella originaria di Schroeder e quella generata dalla funzione `dfl`. Entrambe condividono la stessa tendenza, per cui ogni nuova occorrenza dell'impulso viene attenuata di un coefficiente g, e questo conferisce al segnale un decadimento esponenziale. Pur avendo un diagramma a blocchi apparentemente identico, le risposte all'impulso dei filtri si comportano in modo diverso, motivo per cui meritano uno sguardo di analisi e un'attenta riflessione.

La risposta all'impulso di Figura 2 è stata generata impostando le variabili $t = 1$ (un campione di ritardo) e $g = 1/\sqrt{2}$ (ampiezza scalata di $1/\sqrt{2}$ ad ogni ciclo di feedback). Ci si aspetta quindi un comportamento molto simile a quello descritto da Schroeder: per una successio-

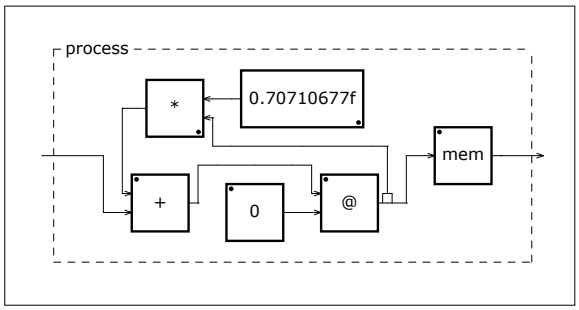

Figure 3. Diagramma a blocchi generato con la funzione `dflc` in *Faust*. La correzione temporale determina un $t - 1 = 0$ per $t = 1$. Il campione sottratto all'unità di ritardo viene ripristinato fuori dal ciclo di reiterazione che già conta, intrinsecamente, un campione di ritardo.

ne di multipli di 1 dovremmo avere il primo impulso nella posizione $n[1]$ con ampiezza $a = 1$ (non scalata, l'impulso non è ancor transitato nel ciclo di feedback). Il secondo impulso dovrebbe essere posizionato subito dopo il primo, nella posizione $n[2]$ con ampiezza $a = 1/\sqrt{2}$, e così proseguendo. Tuttavia, osservando l'indicizzazione dei campioni sull'asse delle ascisse si può constatare che in realtà il nostro filtro sta impiegando due campioni per ogni ciclo impulsivo in luogo di uno, restituendo gli impulsi a periodo $2t$. La motivazione di questa differenza sta nella natura degli ambienti di programmazione *samplerate* come *Faust*. Poichè qui il tempo scorre ad un intervallo minimo di un campione, qualsiasi tipo di operazione richiede almeno un campione per essere eseguita. Per cui, la composizione $\sim$ produce, nel momento in cui la linea di retroazione ritorna in input, un inevitabile campione di ritardo. Per adeguare quindi, i risultati prodotti dall'implementazione *samplerate* del filtro a quelli numerici prospettati da Schroeder, occorre definire una nuova funzione `dflc`, corretta:

```
dflc(t,g) = (+ : @(t-1))~*(g) : mem;
```

Sottraendo il campione di ritardo, prodotto dalla ricorsione, alla variabile t con $t - 1$ si produrranno, per $t = 1$, zero campioni di ritardo all'uscita e un campione scalato di g all'entrata della somma in input. Questo, però genera un offset temporale in quanto la sequenza di campioni ritardati, ora a periodo corretto, si trova in uscita un campione in anticipo a causa del ritardo zero. Anche questa problematica è risolvibile inserendo un ulteriore campione di ritardo, utilizzando la primitiva `mem`, all'uscita del filtro, in modo da bilanciare l'intera sequenza temporale.

La Figura 4 mostra la risposta all'impulso del filtro corretto, ora coerente con le aspettative. Nel descrivere le caratteristiche del filtro nel dominio del tempo e della frequenza, Schroeder commenta:

> La risposta in ampiezza-frequenza ha le sembianze di un pettine con massimi e minimi periodici [...]. È proprio questo sistema di picchi e valli che conferisce l'indesiderata qualità "colorata" al suono [...].

È da sottolineare il carattere sperimentale e pionieristico

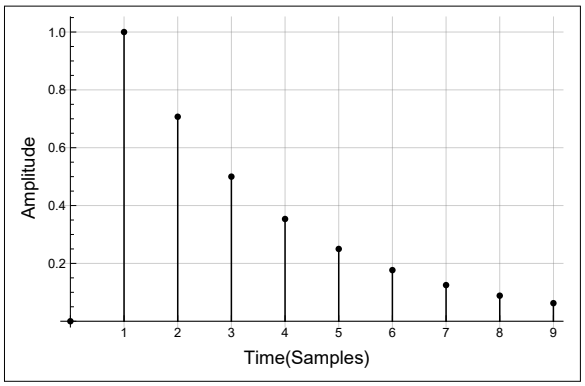

Figure 4. Risposta all'impulso della funzione `dflc` in *Wolfram Mathematica*, coerente con la rappresentazione di Schroeder, e caratterizzata da riscalamenti dell'impulso a periodo t.

delle implementazioni di Schroeder di quel periodo, al punto che la dicitura ≪ha le sembianze un pettine≫ poi si trasforma linguisticamente in *comb-filter*, filtro a *pettine*.

Il filtro appena costruito può quindi portare a diversi risultati sonori: dal semplice ritardo di una quantità di campioni (per $g = 0$) ad un filtraggio *low-pass* (con $t = 1$), ad un filtro *comb* (con $t > 1$).

2.2 All-Pass Filter

Sebbene il filtro *comb* sia sufficiente a generare echi multipli, la colorazione del segnale in uscita rappresenterebbe un ostacolo alla progettazione di un riverberatore artificiale con una risposta in frequenza piatta. Per eliminare questo problema Schroeder propone quindi di miscelare il segnale ritardato uscente dal filtro *comb* con una certa quantità di segnale diretto.

> In altre parole, l'aggiunta di un percorso non ritardato opportunamente proporzionato, ha convertito il filtro *comb* [...] in un filtro *all-pass* [5] [...]. Ciò non è un semplice risultato accademico. La conversione [...] è accompagnata da un miglioramento marcato nella qualità del suono riverberato dal carattere cavo del precedente alla qualità del successivo, perfettamente "incolore".

Consultando i diagrammi a blocchi originali, è evidente l'intenzione di Schroeder, nella rappresentazione del filtro *comb* di predisporlo anche solo graficamente per essere inserito nel futuro *all-pass*. Per questo motivo, l'implementazione *Faust*, segue la stessa logica di stratificazione (tipica dei linguaggi di programmazione funzionale) per cui, la definizione di funzioni semplici (`dflc`) viene poi utilizzata all'interno di funzioni più complesse (`apf`).

Dal diagramma a blocchi di Figura 5: mediante la composizione divisiva `<:`, ciascun impulso in ingresso $n[x]$

[5] *ndt* si usa il nome *all-pass* non tradotto per definire il tipo di filtro, ideato in quel periodo storico dallo stesso autore, con la caratteristica "passa tutto".

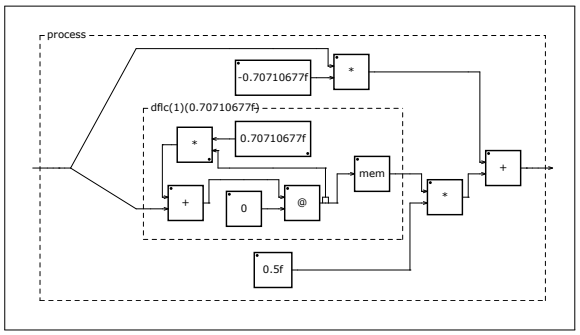

Figure 5. Diagramma a blocchi generato con la funzione `apf` in *Faust*, con aggiunta dei coefficienti di bilanciamento $-g$ per il segnale diretto e $1 - g^2$ per il segnale ritardato.

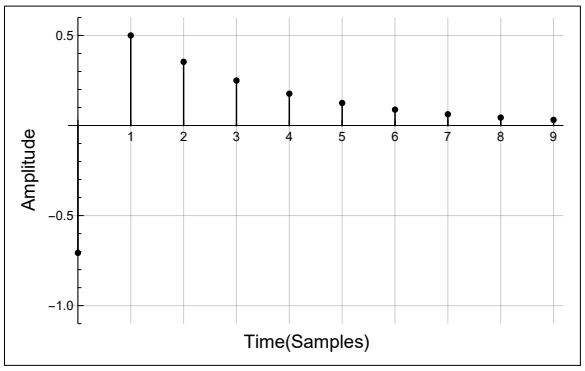

Figure 6. Risposta all'impulso della funzione `apf` in *Wolfram Mathematica*, caratterizzata dal riscalamento e inversione di fase del primo campione in uscita e dal decadimento esponenziale dei successivi.

viene biforcato in corrispondenza del punto di prelievo. Di questi, uno viene scalato di un coefficiente $-g$ (negativo, per cui viene anche invertito di fase), l'altro transita all'interno di `dflc`, dove, dopo ciascun ciclo di retroazione, a periodo t e coefficiente di riscalamento g, viene ulteriormente attenuato di $1 - g^2$. Questi due processi, simultanei, vengono poi sommati prima dell'uscita del filtro.

```
apf(t,g) = _ <: *(-g)+(dflc(t,g)*(1-(g*g)));
```

Il che significa che a posizione $n[1]$, in cui il campione $x[1]$ non ha ancora percorso il ciclo di retroazione, $y[1]$ equivale a $x[1]$ scalato di $-g$; al ciclo successivo in posizione $n[2]$, il campione $x[2]$ viene sommato al precedente $x[1]$, uscente dal dal ciclo di retroazione. La somma di ogni valore $x[n]$ con il precedente $x[n-1]$ procede fino ad estinzione del segnale e questo determina la condizione per cui

> [...] il riverberatore all-pass [...] condivide con gli altri riverberatori che usano delay e feedback la proprietà del decadimento esponenziale dell'energia sonora come avviene negli ambienti acusticamente ben progettati.

La presenza simultanea di segnale diretto e segnale ritardato ha come risultato il fatto che il filtro *all-pass* si comporti in maniera differente al variare dei valori assunti dal coefficiente di bilanciamento g: per $g = 1$ la componente ritardata `dflc` viene completamente annullata limitando l'intervento del filtro all'inversione di fase del segnale in ingresso; per $g = 0$ la componente diretta e il coefficiente di feedback di `dflc` vengono annullate, limitando l'intervento del filtro al ritardo del segnale in ingresso; per $g = 1/\sqrt{2}$ le componenti diretta e ritardata contribuiscono a quantità bilanciate.

2.3 Tempo di Riverberazione T60

Circa sessant'anni prima della pubblicazione di *Natural Sounding Artificial Reverberation*, *Wallace Sabine*, padre della fisica acustica, definì il tempo di riverbero o *T60* come la quantità di tempo in cui, in un determinato ambiente chiuso, la densità di energia sonora della riverberazione decade di $60dB$.

$$T = [60/(-20 \cdot log|g|)] \cdot \tau = (3/log|1/g|) \cdot \tau \qquad (1)$$

Schroeder, di fatto, eredita la relazione in totale continuità con Sabine, introducendo, però, un discrimine fondamentale: quella che per Sabine era una formula di misurazione delle caratteristiche acustiche di determinato ambiente fisico, diventa per Schroeder un vero e proprio algoritmo di sintesi numerica nella riproduzione di quelle stesse caratteristiche acustiche, in un ambiente digitale.

Per facilitare l'implementazione dei componenti successivi, si è deciso di affiancare a questa relazione anche le relazioni inverse, per il calcolo di t e g.

$$\tau = [(-20 \cdot log|g|) \cdot T]/60 \qquad (2)$$

$$g = 10^{(60 \cdot t)/(-20 \cdot T)} \qquad (3)$$

È possibile confrontare i risultati numerici delle tre relazioni facendo riferimento all'esempio presente nel repositorio.

```
process = sms.t60(11,1/sqrt(2)),
  tau(219.247253417968,1/sqrt(2)),
  gain(219.247253417968,11);
```

3. INCREMENTO DELLA DENSITÀ DI ECHI

Schroeder conosceva a pieno i requisiti di una buona riverberazione, ma era altrettanto consapevole che, nel raggiungimento di questi obiettivi, l'utilizzo di linee di ritardo in feedback, avrebbe generato diverse problematiche, di carattere computazionale (per restituire la densità di echi richiesta sarebbe servito un numero eccessivo di linee di ritardo *in parallelo*), ma anche di carattere qualitativo (connettere *in serie* anche solo due linee di ritardo in feedback avrebbe danneggiato il segnale in ingresso determinando una scarsa qualità del segnale riverberato).

> Per quanto riguarda il problema della *bassa densità di echi*, abbiamo scoperto che servono approssimativamente 1000 echi al secondo per una riverberazione libera da fluttuazioni. [...] Sfortunatamente, le densità di echi

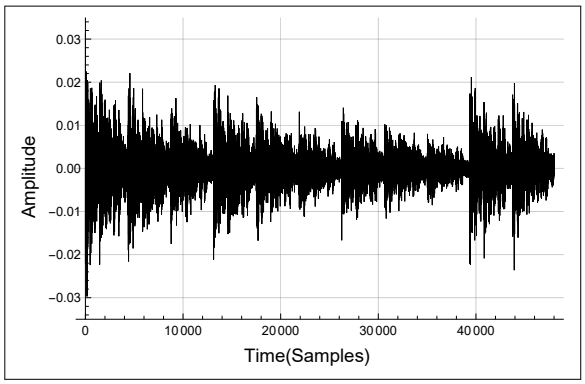

Figure 7. Risposta all'impulso della funzione `apfs` in *Wolfram Mathematica*.

> di 1000 per secondo non sono facilmente raggiungibili da dispositivi di ritardo monodimensionali.

3.1 Serie di All-Pass

Il filtro *all-pass*, grazie alla sua risposta in frequenza piatta, permette di superare queste problematiche e costituisce di fatto una singola unità riverberante da poter collegare in serie per un numero variabile di volte.

> [...] In questa maniera, ciascun'unità *moltiplicherà* effettivamente il numero di echi prodotti dalle unità precedenti. Supponendo che ogni impulso sia "propagato" in 3 di uguale ampiezza, il fattore di moltiplicazione per ogni unità addizionale è all'incirca 3. Ricominciando con un'unità che produce 25 echi al secondo, sono richieste solo all'incirca tra le 3 e 4 unità addizionali per raggiungere la desiderata densità di echi.

Sulla base di queste considerazioni teoriche, Schroeder costruisce così la serie: imposta il ritardo del primo *all-pass* $t_1 = 0.1\,\text{sec}$ e ognuno dei successivi a $1/3$ della sezione precedente; i coefficienti sono tutti uguali al valore di quadratura $g = 0.7$. Istanziandone 5 con queste impostazioni, ottiene 810 echi al secondo: il massimo raggiungibile con le tecnologie che ha a disposizione. Un risultato empirico, non identico a quello teorico desiderato, ma comunque sufficiente a soddisfare delle buone condizioni di riverberazione. Inoltre, per evitare cancellazioni e sovrapposizioni di echi, consiglia di scegliere i tempi di ritardo secondo rapporti incommensurabili, anzichè secondo multipli interi.

La struttura sintattica di *Faust*, capace di mettere in relazione programmazione funzionale e algebra dei diagrammi a blocchi, facilita l'implementazione di questo algoritmo. Le iterazioni in *Faust*, funzionano in maniera analoga ai cicli `for(...)` negli altri linguaggi di programmazione. La funzione `seq`, itera espressioni connettendole in serie; accetta tre argomenti: il primo costituisce il nome della variabile contenente il numero dell'iterazione corrente (analogamente alla variablile `i` nei cicli `for(...)`, indicizzata a partire da 0), il secondo costituisce il numero totale delle iterazioni espresso come una costante intera, il terzo costituisce l'espressione da iterare.

```
apfnp(md,t,m) = seq(i,8,sms.apfv(md,t*m^i : max(
    sff.np,1), 1/sqrt(2-(i/10))));
```

La funzione `apfnp` è appunto un iterazione del filtro `apf`, mediante l'operatore `seq`. Questa, ha come variabili: N, l'ordine dell'iterazione (da istanziare a tempo di compilazione), md, la massima allocazione di memeoria del blocco di ritardo, t_1, il primo tempo di ritardo, m, la base della relazione esponenziale che regola il rapporto tra tempi di ritardo consecutivi della serie, $g = 1/\sqrt{2}$, il coefficiente di feedback degli *all-pass*, identico per ogni filtro della serie.

Per quanto riguarda la costruzione della serie si è deciso di assegnare il tempo di ritardo minimo t_1 al primo *all-pass* della catena e di calcolare i tempi crescenti successivi mediante la relazione impostata nell'iteratore `seq`. A partire da t_1 la relazione $t * m^i : max(np, 1)$ restituisce numeri primi in proporzione, in maniera tale che, per ciascuna iterazione il tempo di ritardo sia un numero primo in rapporto di all'incirca $3:1$ con il precedente.

```
np = ffunction(int next_pr(int), "../h/nextprime
    .h","");
```

Dato un numero intero la funzione `np` restituisce il numero primo immediatamente successivo. È stata scritta utilizzando la primitiva `ffunction`, mediante la quale è possibile richiamare una funzione esterna scritta in linguaggio *C* indicandone il tipo di dato (`int`), il nome della funzione `nextpr_(int)` e il file `include` contenente il codice *C* sotto forma di stringa (`"nextprime.h"`).

Le indicazioni fornite da Schroeder, insieme alle tecnologie di cui disponeva, sono strumenti su cui oggi poter ragionare creativamente. Per questo motivo, nell'implementazione *Faust* di questo algoritmo si è pensato di aumentare il numero di *all-pass* della serie superando quello da lui utilizzato. Il rapporto $3:1$ però comporta una crescita esponenziale del tempo di ritardo che, per un valore di $t_1 = 2$ raggiunge il dato proposto da Schroeder ($0.1\,\text{sec}$) già all'ottava unità. La Figura 7 indica la risposta all'impulso della serie.

3.2 Innesto di All-Pass

La serie di filtri *all-pass* illustrata nel paragrafo precedente costituisce già un algoritmo di riverbero autosufficiente. Nonostante questo, Schroeder propone diversi raffinamenti con lo scopo di conferire, al riverbero, un alto livello di realismo. Alcuni di questi sono:

- miscelazione del suono riverberato con suono diretto.
- introduzione di un intervallo temporale tra suono diretto e riverbero.

Questi raffinamenti, però, devono essere effettuati a condizione che nessuno dei due introduca alcun tipo di colorazione o fluttuazione del segnale riverberato, ovvero, che non violino il *principo all-pass*. Schroeder propone quindi

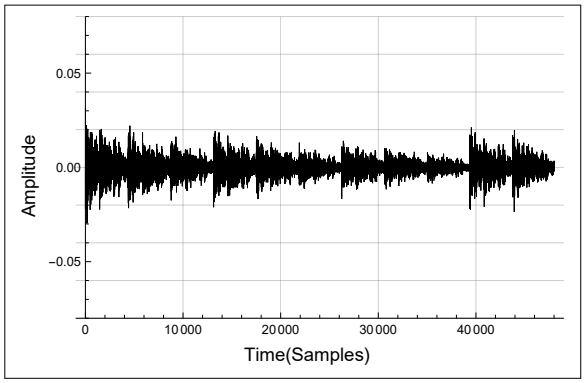

Figure 8. Risposta all'impulso della funzione `mdrned` in *Wolfram Mathematica*. L'incapsulamento della serie di *all-pass* all'interno di un *all-pass* determina il decadimento lineare anzichè esponenziale dell'impulso.

di incapsulare la serie di filtri *all-pass* in un'ulteriore filtro *all-pass*:

```
mdrned(t,g) = _ <: *(-g)+(((+ : de.delay(ma.SR
    /2, int(t-1)) :
          sms.apfnp(ma.SR,2,3))~*(g) : mem)*(1-(
              g*g)));
```

La miscelazione è ottenuta grazie ai coefficienti di bilanciamento ($-g$ per il segnale diretto, $1-g^2$ per il segnale ritardato) mentre, il ritardo tra suono diretto e suono riverberato (quello che nella letteratura successiva verrà definito *pre-delay*) è ottenuto mediante il blocco `delay`. Tra questo e il punto di prelievo del ciclo di retroazione viene inserita la funzione `apfnp`, responsabile del suono riverberato.

Un elemento che contraddistingue quest'architettura dalle precedenti (e dalle successive), evidente nella risposta all'impulso in Figura 8, è il decadimento lineare del segnale riverberato.

3.3 Rete di All-Pass e Comb

In ogni metodo scientifico, il feedback tra osservazione della realtà e produzione di modelli teorici è fondamentale. L'uno influenza l'altro in una relazione costante. Cosi Schroeder, sebbene abbia prodotto un modello teorico impeccabile, non perde di vista la constatazione empirica e pone un problema cruciale:

> Ci si potrebbe chiedere: "Perchè insistere su risposte in frequenza perfettamente piatte se gli ambienti reali hanno risposte in frequenza altamente irregolari?".

Dagli esperimenti psicoacustici svolti nei Bell Telephone Laboratories aveva appreso che, con la sufficiente densità di echi al secondo, le risposte frastagliate erano sostanzialmente indistinguibili dalle risposte piatte. Per un ambiente con un $T60 = 1\,\text{sec}$ ci sono 15 picchi in ogni intervallo di $100cps$. La riflessione sull'esperienza, lo spinge a tornare a ragionare sul modello:

> Quindi si potrebbe sperare che se un riverberatore artificiale avesse un numero simile di picchi alla risposta, potrebbe suonare bene tanto quanto un ambiente reale.

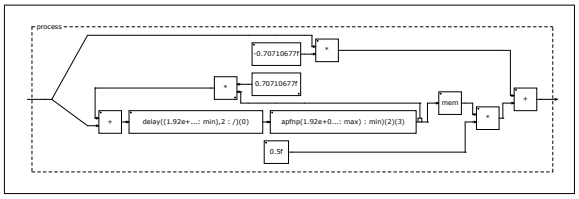

Figure 9. Diagramma a blocchi generato con la funzione `mdrned` in *Faust*, con incapsulamento della funzione `apfnp` all'interno di un filtro *all-pass*, nello specifico tra blocco di ritardo e punto di prelievo del ciclo di retroazione.

Per ottenere un simile risultato, Schroeder propone di utilizzare diversi filtri *comb*. Poichè con un ritardo di $0.04\,\text{sec}$ si generano 4 picchi ogni $100cps$, per approssimare il numero di picchi alla risposta di una stanza con $T60 = 1\,\text{sec}$ richede tra le 3 e 4 unità collegate in parallelo, ciascuna con tempi di ritardo incommensurabili e $g = 0.85$, per prevenire ogni tipo di fluttuazione. Ciascun filtro *comb* di questa configurazione produce 25 echi al secondo. Di conseguenza tutti 4 ne producono 100 (un decimo dei 1000 richiesti); poichè ogni unità riverberante *all-pass* moltiplica il numero di echi al secondo per 3, ne sono richieste due per arrivare quasi a quello richiesto in concordanza con le premesse teoriche dell'inizio.

```
dflapf = _ <: dflc(673,0.89), dflc(1447,0.78),
    dflc(1811,0.74), dflc(2111,0.63) :>
          apf(173,1/sqrt(2)) : apf(229,1/sqrt(2)
              );
```

3.4 Tempo di Riverbero Dipendente dalla Frequenza

L'ultimo raffinamento che Schroeder propone per l'architettura illustrata nel paragrafo precedente è la dipendenza del tempo di riverbero dalla frequenza. Per realizzare questa condizione, ciascuno dei guadagni g dei filtri *comb* può essere reso dipendente dalla frequenza mediante l'aggiunta di un filtro *low-pass* ad un polo (in sostituzione del filtro RC da lui proposto) all'interno di ogni ciclo di feedback.

```
dflf(md,t,g,d) = (+ : de.delay(md,t-1))~sfi.
    eavg2(d)*(g) : mem;
```

Qui, `eavg2` è controllato mediante coefficiente $0 < a < 1$, tuttavia la libreria dispone anche di un filtro `lp1p`, il cui controllo avviene mediante frequenza di taglio `fc`.

4. RIVERBERAZIONE AMBIOFONICA

Alla fine del racconto di Schroeder ci si trova a leggere e ragionare attorno ad un'ipotesi che oggi risulta piuttosto lontana dal senso comune di riverbero artificiale. Schroeder invoca un'ultima modifica dei suoi riverberatori affinchè possano ≪produrre un realistico riverbero tridimensionale [...] Ovvero renderlo "ambiofonico"≫. Il termine ambiofonico viene riferito da Schroeder a Roelof Vermulen [5] ma il testo citato non riporta il termine. Risalendo alle fonti, appare l'introduzione del neologismo *ambiophonic* ad

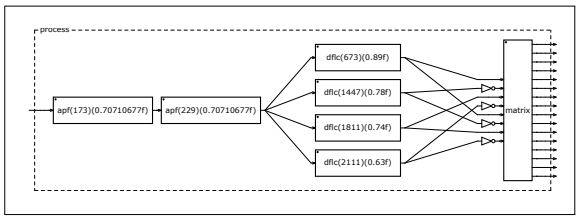

Figure 10. Diagramma a blocchi generato con la funzione `ambiorev` in *Faust*.

opera di Derk Kleis, ingegnere Philips di cui Vermeulen era Direttore della Ricerca Acustica dal 1947.

> A new term is introduced to replace the formerly used "stereo-reverberation". It appears that the latter is sometimes confused with stereophony - often abreviated to stereo. The risk of confusion is all the greater since the advent of the stereophonic gramophone record wich is making stereophony - or stereo - familiar to the genral public. It is hoped the introduction of the term *ambiophony* will put an end to this confusion.

Il testo a cui si riferisce Schroeder [5] è una elaborazione di un precedente lavoro di Vermeulen [6] al quale Kleis fa riferimento introducendo il termine prima di descrivere la necessità di correggere l'acustica di ambienti il cui tempo di riverbero è troppo breve introducendo suono indiretto artificiale mediante, appunto, la tecnica ambiofonica.

Il dato importante della parte finale del testo di Schroeder è che chiarisce l'esigenza di introdurre passo passo i riverberatori digitali illustrati nelle sezioni precedenti al fine di poter entrare nel pieno della ricerca sulla riverberazione artificiale che infuocava le discussioni di quel periodo con una proposta completamente nuova: sostituire i riverberi elettromeccanici con i digitali per la conversione elettroacustica delle sale da concerto.

5. CONCLUSIONI

Il percorso di approfondimento critico nei confronti dei testi di Manfred Schroeder ha permesso una progressiva comprensione delle architetture dei processori di riverbero archetipici corroborando l'importanza della letteratura storica in quanto materiale vivo, attuale, accessibile.

La ri-costruzione di queste architetture mediante l'utilizzo di *Faust* e *Wolfram Mathematica* ha permesso la risoluzione delle problematiche relative a implementazioni in ambienti *samplerate* mediante una struttura di analisi solida corredata di diagramma a blocchi, supporto matematico e risposte all'impulso dettagliate.

La comprensione a basso livello dei principi di funzionamento ha stimolato implementazioni creative di questi stessi processori che sono state documentate in un ulteriore articolo dello stesso gruppo di ricerca. [4]

In ultimo, la redazione di una libreria di funzioni inserita nel progetto *SEAM - Sustainable Electro-Acoustic Music* fa si che i risultati di questa ricerca contribuiscano alla

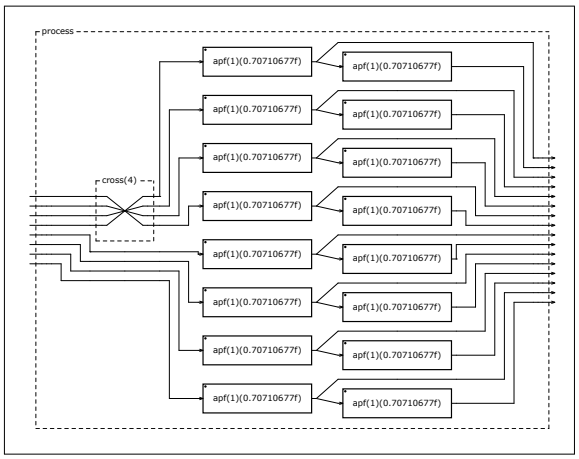

Figure 11. La matrice di resistenze indicata da Schroeder è stata sostituita con una rete di all-pass innestati che riproducono il movimento del fronte d'onda.

crescita di un ambiente condiviso, garante dell'accessibilità e della tutela di buona parte del patrimonio informatico elettroacustico ad oggi conosciuto.

6. REFERENCES

[1] W. C. Sabine, *Collected papers on acoustics*. 1921.

[2] M. R. Schroeder, "Natural sounding artificial reverberation," *J. Audio Eng. Soc*, vol. 10, no. 3, pp. 219–223, 1962.

[3] M. R. Schroeder, "New method of measuring reverberation time," *Acoustical Society of America*, 1964.

[4] F. Vitucci, D. Annese, A. Di Furia, F. Scagliola, and G. Silvi, "Architettura aperta come strumento creativo: il riverberatore," *Atti del XXIV Colloquio di Informatica Musicale*, 2024, under review.

[5] R. Vermeulen, "Stereo-reverberation," *J. Audio Eng. Soc*, vol. 6, no. 2, pp. 124–130, 1958.

[6] R. Vermeulen, "Stereo-reverberation," *Philips Technical Review*, vol. 17, no. 9, pp. 258–268, 1956.

[7] M. R. Schroeder, "An artificial stereophonic effect obtained from a single audio signal," *J. Audio Eng. Soc*, vol. 6, no. 2, pp. 74–79, 1958.

[8] M. R. Schroeder and B. F. Logan, "'colorless' artificial reverberation," *J. Audio Eng. Soc*, vol. 9, no. 3, pp. 192–197, 1961.

[9] M. R. Schroeder, "Listening with two ears," *Music Perception: An Interdisciplinary Journal*, vol. 10, no. 3, pp. 255–280, 1993.

[10] J. Dattorro, "Effect design, part 1: Reverberator and other filters," *J. Audio Eng. Soc*, vol. 45, no. 9, pp. 660–684, 1997.

A. LIBRERIA

```
declare name "SEAM Schroeder - Library";
declare version "0.2";
declare author "Daniele Annese";
declare author "Anthony Di Furia";
declare author "Giuseppe Silvi";
declare author "Francesco Vitucci";
declare license "CC3";

import("seam.lib");

//==============================================================================
//============= SCHROEDER - NATURAL SOUNDING ARTIFICIAL REVERBERATION - 1962 ===
//==============================================================================
sms = library("seam.schroeder.lib");
//================================================== DELAY FEEDBACK IN LOOP ---
// direct implementation, wrong impulse response
dfl(t,g) = (+ : @(t))~*(g);
//process = os.impulse : dfl(1,1/sqrt(2));

//------------------------------------------ DELAY FEEDBACK IN LOOP - FIXED ---
// delay compensation, right impulse response
dflc(t,g) = (+ : @(t-1))~*(g) : mem;
//process = os.impulse : dflc(1,1/sqrt(2));

//-------------------------------- DELAY FEEDBACK IN LOOP - FIXED - VARIABLE ---
// runtime variable delay - max del required
dflcv(md,t,g) = (+ : de.delay(md, int(t-1)))~*(g) : mem;
//process = os.impulse : dflcv(ma.SR,1,1/sqrt(2));

//============================================================== ALL-PASS FILTER ===
apf(t,g) = _ <: *(-g)+(dflc(t,g)*(1-(g*g)));
//process = os.impulse : apf(1,1/sqrt(2));

//------------------------------------------------ ALL-PASS FILTER - VARIABLE ---
// runtime variable delay - max del required
apfv(md,t,g) = _ <: *(-g)+(dflcv(md,t,g)*(1-(g*g)));
//process = os.impulse : apfv(ma.SR,1,1/sqrt(2));

//================================================================================ T60 ===
t60(t,g) = (60/(-20*log10(g)))*t;
//process = msT60(0.1,1/sqrt(2)); // 2 seconds

//================================================ INCREASE OF ECHO DENSITY ===
// direct implementation of five all-pass in serie.
ied5 = apf(5507,1/sqrt(2)) : apf(1831,1/sqrt(2)) : apf(613,1/sqrt(2)) :
     apf(199,1/sqrt(2)) : apf(67,1/sqrt(2));
//process = os.impulse : ied5;

//------------------------------ INCREASE OF ECHO DENSITY WITH VARIABLE TIMES ---
apfs(N,md,t,g) = seq(i,N,apfv(md,ba.take(i+1,t),g));
//process = os.impulse <: apfs(9,(spn.p3t),1/sqrt(2));

//------------------------ INCREASE OF ECHO DENSITY WITH NEXT PRIME FUNCTION ---
// N = order
// t = smallest t
// m = base for m^i => step (3 for 1/3 ratio)
apfnp(md,t,m) = seq(i,8,sms.apfv(md,t*m^i : max(sff.np,1), 1/sqrt(2-(i/10))));
//process = os.impulse : apfnp(ma.SR,2,3);

//========== MIXING OF DIRECTSOUND AND REVERBERATION - NON-EXPONENTIAL DECAY ===
mdrned(t,g) = _ <: *(-g)+(((+ : de.delay(ma.SR/2, int(t-1)) :
           sms.apfnp(ma.SR,2,3))~*(g) : mem)*(1-(g*g)));
//process = os.impulse : mdrned(1,1/sqrt(2));

//================================================ THE COMB FILTER APPROACH ===
dflapf = _ <: dflc(673,0.89), dflc(1447,0.78), dflc(1811,0.74), dflc(2111,0.63) :>
          apf(173,1/sqrt(2)) : apf(229,1/sqrt(2));

apfdfl = apf(173,1/sqrt(2)) : apf(229,1/sqrt(2)) <:
  dflc(673,0.89), dflc(1447,0.78), dflc(1811,0.74), dflc(2111,0.63);

//================================== FREQUENCY-DEPENDENT REVERBERATION TIME ===
dflf(md,t,g,d) = (+ : de.delay(md,t-1))~sfi.eavg2(d)*(g) : mem;
//process = os.impulse : dflf(ma.SR,1,1/sqrt(2),0.9);

//================================================ AMBIOPHONIC-REVERBERATION ===
//------------------------------------------------------------ NESTED ALL-PASS ---
apfn(0,t,g) = _;
apfn(1,t,g) = apf(t,g);
apfn(n,t,g) = apf(t,g) <: _,apfn(n-1,t,g);
//------------------------------------------------------------------- MATRIX ---
matrix = ro.cross(4), si.bus(4) : par(i,8, apfn(2,1,1/sqrt(2)));
//---------------------------------------- 16 CH AMBIOPHONIC REV EXAMPLE ---
ambiorev = apf(173,1/sqrt(2)) : apf(229,1/sqrt(2)) <:
 dflc(673,0.89), dflc(1447,0.78), dflc(1811,0.74), dflc(2111,0.63) <:
 par(i,4,_,0-_) : matrix;
```