

UNIVERSITÀ POLITECNICA DELLE MARCHE

FACOLTÀ DI INGEGNERIA

Corso di Laurea Triennale in Ingegneria Elettronica e delle Tecnologie Digitali

Modellazione fisica di macchine a nastro analogiche e implementazione real-time

Physical modeling of analog tape machines and real-time implementation

Tesi di Laurea di:
CHRISTIAN VESPERINI

Relatore: Chiar.mo
Prof. **LEONARDO GABRIELLI**

Correlatore:
Prof. **STEFANO SQUARTINI**

Anno Accademico 2025/2026

Abstract

Questo lavoro illustra la modellazione fisica e l'implementazione in tempo reale di un effetto che simula un tape delay vintage, basato sull'emulazione del registratore a nastro analogico Sony TC-260. Il cuore matematico del sistema sfrutta il modello di Jiles-Atherton per la simulazione accurata del ciclo di isteresi magnetica.

Il progetto si articola in due fasi. La prima consiste in un'implementazione offline del sistema in ambiente MATLAB; la seconda fase riguarda il *porting* del modello in linguaggio C++ per l'esecuzione in tempo reale all'interno della piattaforma modulare VCV Rack.

Il risultato finale è un modulo software stabile e interattivo, in grado di riprodurre fedelmente la saturazione, le imperfezioni e le fluttuazioni meccaniche della controparte analogica.

Indice

1 INTRODUZIONE **1**
1.1 Contesto e stato dell'arte 1

2 DESCRIZIONE DELL'ALGORITMO DI TAPE DELAY **5**
2.1 Architettura generale del sistema 5
2.2 Modello di Jiles-Atherton e ciclo di isteresi 7
2.3 Implementazione numerica del modello 9
2.3.1 Generazione del segnale di eccitazione e del campo magnetico 11
2.3.2 Risoluzione tramite metodo di Runge-Kutta 13
2.4 Modellazione della testina di riproduzione e delle perdite 15
2.4.1 Spacing loss 15
2.4.2 Thickness loss 16
2.4.3 Gap loss 16
2.4.4 Implementazione mediante filtro FIR 17
2.5 Debias e conversione alla frequenza di campionamento originale . . . 17

3 VALIDAZIONE MATLAB **20**
3.1 Validazione del ciclo di isteresi dinamica 21
3.2 Analisi spettrale e temporale della distorsione 22
3.3 Validazione delle perdite della testina (Loss Effects) 24
3.4 Analisi del Wow e Flutter 25

4 IMPLEMENTAZIONE IN C++ **28**
4.1 Porting del modello e integrazione in VCV Rack 28
4.2 Stabilizzazione del solutore numerico 29
4.3 Implementazione della linea di ritardo e del feedback 31
4.4 Generazione real-time di wow e flutter 33
4.5 Taratura dei parametri e adattamento del modello 34
4.6 Ottimizzazione del carico computazionale 36
4.6.1 Valutazione di un fattore di sovracampionamento ridotto . . . 36
4.6.2 Precalcolo e aggiornamento periodico dei coefficienti 37
4.7 Interfaccia del modulo VCV Rack 38
4.8 Verifica dell'implementazione real-time 39

5 CONCLUSIONI **43**
5.1 Sviluppi futuri 44

Bibliografia **46**

Capitolo 1

INTRODUZIONE

1.1 Contesto e stato dell'arte

La possibilità di modificare e caratterizzare il suono attraverso sistemi di elaborazione costituisce da sempre un elemento fondamentale della produzione audio. L'impiego di effetti audio per alterare le caratteristiche del segnale rappresenta una pratica consolidata, sia nelle esibizioni dal vivo sia nella produzione discografica in studio, durante le fasi di registrazione, mixing e mastering. Tradizionalmente, tali operazioni sono state realizzate mediante dispositivi analogici, il cui impiego comporta alcune limitazioni, tra cui l'ingombro fisico, gli elevati costi di manutenzione e la complessità operativa. L'introduzione dell'elaborazione digitale ha consentito di superare in parte tali vincoli, offrendo maggiore flessibilità nel trattamento dei segnali e semplificandone l'instradamento. In questo contesto, l'elaborazione numerica dei segnali (Digital Signal Processing, DSP) ha assunto un ruolo centrale nello sviluppo di sistemi per l'elaborazione audio.

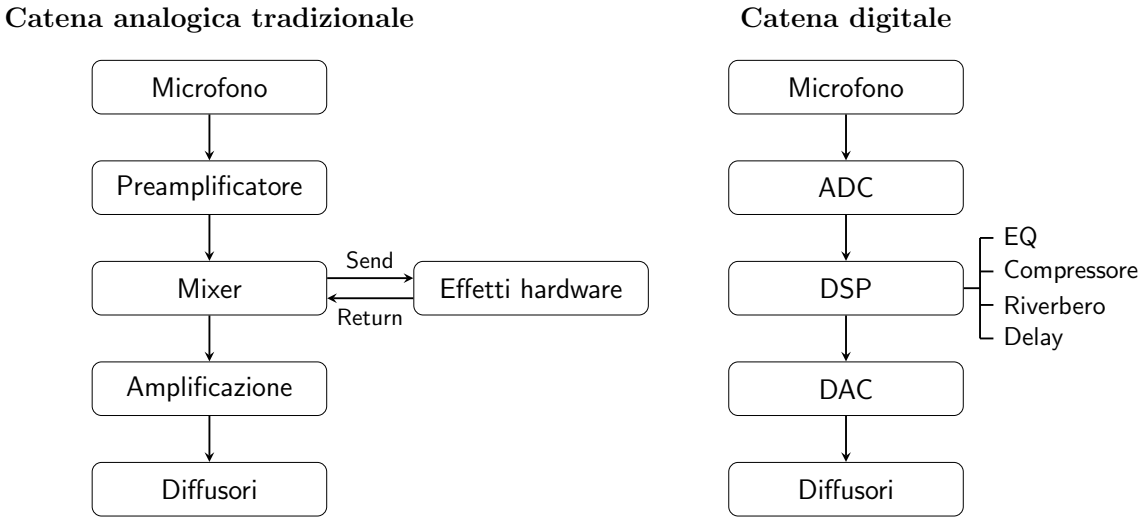


Figura 1.1: Confronto architetturale tra il flusso di segnale analogico e l'elaborazione digitale.

L'evoluzione dei mixer audio rappresenta un esempio significativo di questa transizione. Le moderne console digitali, sia per il live sia per lo studio, integrano la potenza di calcolo necessaria per gestire simultaneamente centinaia di processori di segnale, riducendo quasi completamente la necessità di unità di elaborazione esterne ingombranti e costose.

Nonostante i vantaggi in termini di riduzione del rumore, efficienza e praticità offerti dal dominio digitale, nel contesto musicale permane una forte esigenza estetica di recuperare le "imperfezioni" e il calore timbrico tipici delle apparecchiature storiche. In questo contesto si è sviluppato il paradigma del *Virtual Analog* (VA), volto a riprodurre mediante algoritmi digitali il comportamento di sistemi analogici. Una panoramica delle principali tecniche di modellazione Virtual Analog è proposta in [1], mentre [2] ne tratta l'applicazione alla simulazione di effetti e circuiti analogici.

Tra gli effetti di maggiore rilievo in questo settore si annovera il *tape delay*, un dispositivo elettroacustico progettato per generare effetti di eco. Il funzionamento si basa sulla registrazione del segnale audio su un anello di nastro magnetico e sulla successiva lettura tramite una testina di riproduzione posizionata a una certa distanza da quella di registrazione. La reiniezione di una porzione del segnale letto all'ingresso del sistema, tramite un circuito di feedback, produce le tipiche ripetizioni. Le imperfezioni fisiche del supporto determinano un degrado progressivo di ogni ripetizione, caratterizzato da distorsione armonica, perdita di alte frequenze e modulazioni temporali dovute alle instabilità del motore (wow e flutter).

Dispositivi iconici come il Maestro Echoplex e il Roland RE-201 Space Echo hanno storicamente definito il carattere distintivo di questo effetto.



Figura 1.2: A sinistra: l'unità hardware Maestro Echoplex. A destra: il Roland RE-201 Space Echo.

La modellazione specifica di tali sistemi ha stimolato approcci dedicati, come documentato dal lavoro di Arnardottir [3] sul ritardo tempo-variabile dell'Echoplex. Più in generale, in letteratura si distinguono due strategie principali di modellazione. Da un lato vi sono i modelli *black-box*, che utilizzano tecniche di apprendimento automatico e garantiscono costi computazionali prevedibili, ma senza un legame diretto con i parametri fisici reali; un esempio di questo approccio applicato ai registratori a nastro è il modello basato su reti neurali proposto da Mikkonen et al. [4]. Dall'altro lato, i modelli *white-box* si basano sulla modellazione fisica e permettono di associare l'algoritmo ai parametri costruttivi originali, a fronte di una maggiore complessità dovuta alla necessità di risolvere equazioni differenziali. In quest'ultimo ambito, il modello fenomenologico di Jiles-Atherton [5] si è affermato come riferimento per la descrizione dell'isteresi magnetica.
La modellazione fisica consente quindi di ottenere una rappresentazione maggiormente interpretabile dei fenomeni che caratterizzano il dispositivo reale. Tuttavia, l'impiego di modelli di questo tipo in applicazioni audio in tempo reale introduce un ulteriore vincolo progettuale: la complessità computazionale deve essere compatibile con i vincoli temporali imposti dall'elaborazione digitale del segnale. È pertanto necessario individuare un compromesso tra l'accuratezza della simulazione e il costo computazionale, intervenendo sia sull'algoritmo sia sulla sua implementazione.

In tale contesto, il presente lavoro prende le mosse dal modello fisico sviluppato da Chowdhury [6], il quale propone un'emulazione completa della macchina a nastro Sony TC-260. Tale emulazione è basata sulla derivazione analitica dei campi magnetici, su misurazioni hardware dirette per fenomeni quali il *wow e flutter*, e su un'implementazione in tempo reale nel framework JUCE/C++.
Rispetto a tale approccio, questo progetto di tesi si differenzia per il focus specifico sulla modellazione di un effetto tape delay con catena di retroazione. A tal fine, il modello di Jiles-Atherton viene isolato e impiegato come blocco non lineare centrale all'interno del percorso di segnale, mentre il guadagno del ramo di feedback e i parametri utilizzati per la generazione di wow e flutter sono stati definiti nell'ambito della simulazione. Inoltre, a livello metodologico, il lavoro si distingue per una fase di prototipazione algoritmica e validazione in ambiente MATLAB, propedeutica alla successiva implementazione *real-time* in C++ per la piattaforma VCV Rack. Per esporre in modo organico il lavoro svolto, la trattazione è strutturata come segue:

- Il **Capitolo 2** descrive l'architettura generale del sistema e le equazioni fenomenologiche alla base del tape delay. Vengono illustrati il modello di Jiles-Atherton per l'isteresi magnetica, l'integrazione numerica delle equazioni e la modellazione degli effetti di perdita della testina.

- Il **Capitolo 3** è dedicato alla validazione del modello offline in ambiente MATLAB. Vengono presentate le analisi temporali e spettrali necessarie per verificare il corretto funzionamento della catena di elaborazione, inclusa la corretta simulazione del ciclo di isteresi e le modulazioni di wow e flutter.

- Il **Capitolo 4** presenta l'evoluzione del modello verso l'implementazione in tempo reale in linguaggio C++ per la piattaforma VCV Rack. Vengono discusse la stabilizzazione del solutore numerico, la taratura dei parametri rispetto all'hardware di riferimento e le strategie algoritmiche impiegate per la gestione e l'ottimizzazione del carico sulla CPU.

- Infine, il **Capitolo 5** riassume i risultati ottenuti, valutando l'efficacia del sistema sviluppato e proponendo possibili sviluppi futuri.

Capitolo 2

DESCRIZIONE DELL'ALGORITMO DI TAPE DELAY

2.1 Architettura generale del sistema

A partire dai principi descritti nel Capitolo 1, il sistema sviluppato traduce i principali fenomeni della registrazione magnetica in una catena di elaborazione numerica. L'obiettivo è riprodurre, oltre al ritardo e al feedback tipici del tape delay, gli effetti non lineari e le alterazioni del segnale introdotte dal processo di registrazione e riproduzione su nastro.

Un elemento particolarmente importante nel processo di registrazione è il *bias*. Al segnale audio viene infatti sovrapposto un segnale sinusoidale ad alta frequenza, utilizzato per ridurre la distorsione introdotta dalla caratteristica non lineare del materiale magnetico. Il bias consente di ottenere condizioni di magnetizzazione più favorevoli per la registrazione del segnale audio e viene pertanto applicato prima del modello di isteresi.

Nel modello sviluppato, il bias è rappresentato mediante un segnale sinusoidale a 55 kHz, sommato al segnale audio prima dell'ingresso nel modello di Jiles–Atherton. Poiché tale componente non appartiene al segnale audio di interesse, al termine della catena viene applicato un filtraggio di *debias*, con lo scopo di attenuarne il residuo prima del ritorno alla frequenza di campionamento originale.

A partire da queste considerazioni, è stata definita la catena di elaborazione del segnale, traducendo i principali fenomeni fisici del registratore a nastro in algoritmi

operanti nel dominio del tempo discreto. Come illustrato in Figura 2.1, il percorso di elaborazione si articola nelle seguenti macro-fasi:

- **Oversampling e iniezione del Bias:** Il segnale audio in ingresso viene sovracampionato con un fattore $OS = 16$. Questo passaggio consente di ridurre l'aliasing introdotto dalle non linearità del modello durante l'elaborazione a frequenza sovracampionata. Al segnale sovracampionato vengono quindi sommati il segnale di feedback, ottenuto dalla reiniezione del segnale riprodotto, e il tono di bias ad alta frequenza.

- **Modello di Isteresi (Testina di Registrazione):** Il modello di isteresi di Jiles-Atherton descrive l'evoluzione non lineare della magnetizzazione del nastro, modellando il processo di registrazione magnetica e la conseguente saturazione.

- **Effetti di Perdita (Testina di Riproduzione):** La testina di lettura introduce un'attenuazione naturale dipendente dalla frequenza. Questo comportamento passa-basso è modellato tramite un filtro FIR che implementa analiticamente le perdite per spaziatura, spessore e traferro (*spacing, thickness e gap loss*).

- **Modulazione del Ritardo (Wow & Flutter):** Il transito fisico del nastro tra le due testine è simulato da una linea di ritardo digitale, la cui lunghezza è costantemente modulata da disturbi stocastici per emulare le imperfezioni meccaniche del motore (*wow*) e le vibrazioni di scorrimento (*flutter*).

- **Debias e Downsampling:** Il segnale globale attraversa un filtro passa-basso di pulizia per sopprimere il residuo ad alta frequenza del bias, venendo infine decimato per tornare alla frequenza di campionamento originale in fase di riproduzione.

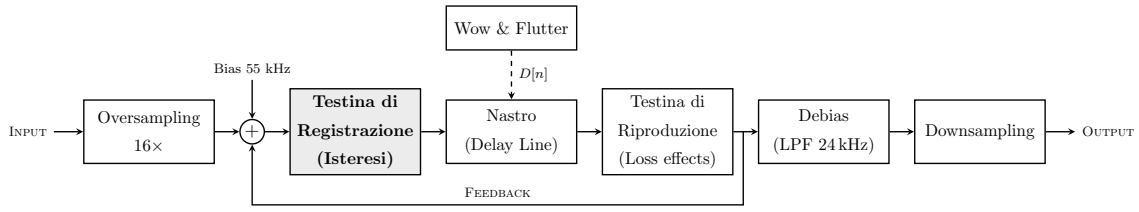


Figura 2.1: Architettura generale della catena di elaborazione. Il segnale audio sovracampionato viene sommato al segnale di feedback e al bias ad alta frequenza, quindi applicato al modello non lineare di Jiles-Atherton, che rappresenta il processo di registrazione magnetica. Il segnale registrato viene successivamente trasferito lungo una linea di ritardo, la cui lunghezza è modulata dalle componenti di *wow* e *flutter*, e viene quindi elaborato dal modello delle perdite della testina di riproduzione. Il segnale così ottenuto viene utilizzato sia per il feedback sia per il successivo filtraggio di debias e downsampling.

La catena descritta traduce quindi nel dominio discreto i principali fenomeni fisici coinvolti nel funzionamento del tape delay. Tra questi, il modello di isteresi costituisce il nucleo non lineare della simulazione, in quanto descrive il processo di registrazione magnetica e le non linearità introdotte dal supporto.

2.2 Modello di Jiles-Atherton e ciclo di isteresi

La modellazione fisica di un registratore a nastro analogico si fonda principalmente sulla simulazione del comportamento non lineare del supporto ferromagnetico, generalmente composto da ossido di ferro (γ-Fe_2O_3). Dal punto di vista microscopico, il nastro è costituito da domini magnetici. In assenza di un campo magnetico esterno, i momenti magnetici di questi domini sono orientati secondo direzioni differenti, producendo una magnetizzazione macroscopica prossima allo zero.
L'applicazione di un segnale audio alla testina di registrazione genera un campo magnetico H proporzionale alla corrente d'ingresso; all'aumentare di H, le pareti dei domini magnetici si spostano e i momenti magnetici tendono ad allinearsi parallelamente al campo applicato, incrementando la magnetizzazione complessiva M del nastro. Questa relazione è di natura non lineare e si manifesta attraverso un ciclo di isteresi, caratterizzato da alcune grandezze fondamentali:

- **Saturazione (M_s):** Aumentando il campo magnetico, la magnetizzazione tende a raggiungere un valore massimo oltre il quale ulteriori incrementi di H producono variazioni limitate di M.

- **Rimanenza (M_r):** Quando il campo applicato H viene riportato a zero, il nastro non si smagnetizza completamente, ma mantiene una quota di magnetizzazione residua. Questa magnetizzazione residua consente al supporto di conservare l'informazione magnetica anche in assenza di un campo applicato.

- **Forza Coercitiva (H_c):** valore del campo magnetico applicato nella direzione opposta, necessario per azzerare la magnetizzazione residua del materiale.

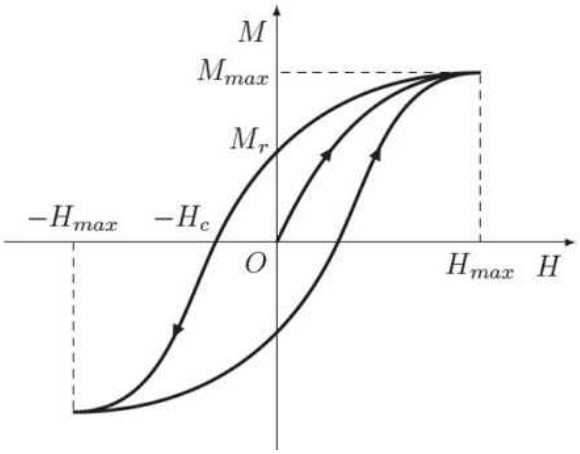


Figura 2.2: Rappresentazione teorica di un ciclo di isteresi magnetica. Sono evidenti la curva di prima magnetizzazione (uscente dall'origine O) e i punti chiave di saturazione (M_{max}), rimanenza (M_r) e coercitività ($-H_c$).

La simulazione matematica di questo comportamento si basa sul modello fenomenologico di Jiles-Atherton [5], ampiamente adottato per descrivere l'isteresi magnetica. L'efficacia del modello risiede nella scomposizione della magnetizzazione totale M in due componenti distinte: una reversibile (M_{rev}) e una irreversibile (M_{irr}).
La componente irreversibile rappresenta le perdite energetiche dovute all'attrito (effetto di *pinning*), causate dalle impurità presenti nel reticolo cristallino del materiale che ostacolano il movimento delle pareti dei domini. La componente reversibile descrive invece le variazioni di magnetizzazione recuperabili al variare del campo applicato.
L'equazione differenziale fondamentale del modello governa la variazione della magnetizzazione totale M rispetto alla variazione del campo magnetico H, ed è espressa come:

$$\frac{dM}{dH} = \frac{(1-c)\delta_M(M_{an}-M)}{(1-c)\delta_S k - \alpha(M_{an}-M)} + c\frac{dM_{an}}{dH} \tag{2.1}$$

dove i principali parametri del modello assumono il seguente significato:

- k rappresenta una misura dell'ampiezza del ciclo di isteresi ed è strettamente correlata alla densità dei siti di *pinning* e alla coercitività H_c del materiale;

- c quantifica la porzione reversibile del processo di magnetizzazione, ovvero il rapporto tra le suscettibilità iniziali normale e anisteretica;
- α è il parametro di campo medio, che modella l'accoppiamento magnetico e le interazioni reciproche tra i domini adiacenti del nastro;
- δ_S è una funzione segno direzionale (assume valore $+1$ se il campo H è in fase crescente, -1 se decrescente);
- δ_M è un parametro che garantisce la corretta dissipazione energetica, impostandosi su 1 se i gradienti hanno lo stesso segno o su 0 altrimenti, evitando comportamenti non fisici durante le inversioni del segnale.

Un elemento fondamentale dell'Equazione 2.1 è la magnetizzazione anisteretica M_{an}, che rappresenta lo stato magnetico termodinamico ideale verso cui tende la magnetizzazione in assenza dei meccanismi irreversibili associati al *pinning*.

$$M_{an} = M_s L\left(\frac{H + \alpha M}{a}\right) \tag{2.2}$$

dove M_s indica la magnetizzazione di saturazione, a è un parametro associato alla densità dei domini che determina la pendenza della curva anisteretica, mentre la funzione $L(x)$ è definita dalla relazione:

$$L(x) = \coth(x) - \frac{1}{x} \tag{2.3}$$

Nel modello utilizzato per l'emulazione del registratore Sony TC-260, i parametri k, c, α, a e M_s determinano il comportamento magnetico del supporto e vengono selezionati per riprodurre la risposta non lineare del sistema [6].
Poiché il segnale audio viene elaborato nel dominio temporale, l'Equazione 2.1, espressa in termini di variazione della magnetizzazione rispetto al campo magnetico, deve essere riformulata introducendo esplicitamente la dipendenza dal tempo t.
Tale trasformazione costituisce il passaggio necessario per ottenere una formulazione idonea alla discretizzazione numerica e alla successiva implementazione dell'algoritmo di elaborazione.

2.3 Implementazione numerica del modello

Applicando la regola della catena si ha:

$$\frac{dM}{dt} = \frac{dM}{dH}\frac{dH}{dt}. \tag{2.4}$$

Nel modello di Jiles-Atherton, tuttavia, la magnetizzazione anisteretica dipende anche dalla magnetizzazione M:

$$M_{an} = M_s L\left(\frac{H + \alpha M}{a}\right). \tag{2.5}$$

Di conseguenza, la derivata temporale di M_{an} dipende sia dalla variazione del campo applicato sia da quella della magnetizzazione. Introducendo

$$Q = \frac{H + \alpha M}{a}, \tag{2.6}$$

si ottiene:

$$\frac{dM_{an}}{dt} = \frac{M_s}{a} L'(Q)\left(\frac{dH}{dt} + \alpha \frac{dM}{dt}\right), \tag{2.7}$$

In questa espressione, $L'(Q)$ indica la derivata della funzione di Langevin rispetto al proprio argomento.

Sostituendo l'Equazione 2.7 nella 2.1 e raccogliendo i termini contenenti $\frac{dM}{dt}$, si ottiene:

$$\frac{dM}{dt} = \frac{\left[\frac{(1-c)\delta_M(M_{an}-M)}{(1-c)\delta_S k - \alpha(M_{an}-M)} + c\frac{M_s}{a}L'(Q)\right]\frac{dH}{dt}}{1 - c\alpha\frac{M_s}{a}L'(Q)}. \tag{2.8}$$

Questa rappresenta la forma temporale del modello utilizzata nell'implementazione MATLAB. In questo contesto, la magnetizzazione M costituisce la variabile di stato del sistema, mentre $H(t)$ rappresenta il campo magnetico applicato.

La derivata della funzione di Langevin utilizzata nel calcolo è:

$$L'(x) = \frac{1}{x^2} - \coth^2(x) + 1. \tag{2.9}$$

La valutazione diretta delle espressioni della funzione di Langevin e della sua derivata in prossimità di $x = 0$ può introdurre errori numerici a causa della sottrazione tra termini di grande ampiezza. Pertanto, nell'implementazione si adotta un'approssimazione tramite sviluppo di Taylor per valori di $|x|$ sufficientemente piccoli:

$$L(x) \approx \frac{x}{3} \qquad , \qquad L'(x) \approx \frac{1}{3}. \tag{2.10}$$

2.3.1 Generazione del segnale di eccitazione e del campo magnetico

Prima dell'integrazione del modello magnetico, il segnale audio viene elaborato alla frequenza di campionamento utilizzata internamente dal sistema. Nell'implementazione considerata viene adottato un fattore di oversampling pari a:

$$OS = 16, \tag{2.11}$$

per cui la frequenza di campionamento interna al modello è:

$$F_{s,up} = OS\, F_s. \tag{2.12}$$

Il campo magnetico applicato alla testina di registrazione viene ottenuto combinando il segnale audio, il segnale di feedback e il bias ad alta frequenza. Il feedback viene ottenuto ricircolando il segnale riprodotto attraverso una linea di ritardo digitale. Il ritardo nominale, espresso in campioni alla frequenza di campionamento sovracampionata, è definito come:

$$D_0 = \text{round}\left(T_d F_{s,up}\right), \tag{2.13}$$

dove T_d rappresenta il tempo di ritardo desiderato. Nell'implementazione considerata, $T_d = 0.3\,\text{s}$, mentre il coefficiente di feedback, ossia la frazione del segnale riprodotto che viene reiniettata all'ingresso del sistema, è posto a $g = 0.85$. Tale valore è stato scelto empiricamente durante la fase di sviluppo e taratura della simulazione, in modo da ottenere un decadimento progressivo delle ripetizioni del segnale.
Per simulare le variazioni della velocità di scorrimento del nastro, il ritardo viene modulato nel tempo mediante due componenti associate ai fenomeni di *wow* e *flutter*:

$$D_{\text{mod}}[n] = D_0 + D_{\text{wow}}[n] + D_{\text{flutter}}[n]. \tag{2.14}$$

La componente di *wow*, associata alle variazioni lente della velocità di scorrimento del nastro, viene generata a partire da una sequenza di valori casuali definiti in corrispondenza di un insieme ridotto di istanti temporali. Per ottenere una variazione continua del ritardo, tali valori vengono successivamente interpolati mediante il metodo *pchip* (*Piecewise Cubic Hermite Interpolating Polynomial*), che costruisce una funzione continua a tratti polinomiali cubici preservando l'andamento locale dei campioni senza introdurre le oscillazioni tipiche delle interpolazioni polinomiali globali. La componente di *flutter*, associata a variazioni più rapide, viene invece ottenuta filtrando un

rumore bianco mediante un filtro passa-banda Butterworth del secondo ordine, con frequenze di taglio comprese tra 3 Hz e 8 Hz.
Entrambe le componenti vengono normalizzate rispetto alla propria deviazione standard e successivamente scalate mediante i rispettivi fattori di profondità. Nel modello utilizzato, tali fattori costituiscono parametri di simulazione scelti per ottenere una modulazione temporalmente apprezzabile del ritardo e non rappresentano misure dirette effettuate sull'hardware di riferimento:

$$d_{\text{wow}} = 10\,OS, \qquad d_{\text{flutter}} = 5\,OS. \tag{2.15}$$

Il segnale prodotto dal modello di Jiles-Atherton viene normalizzato rispetto alla magnetizzazione di saturazione M_s, ottenendo il segnale $y_{raw}[n]$:

$$y_{raw}[n] = \frac{M[n]}{M_s} \tag{2.16}$$

Questo segnale rappresenta la magnetizzazione normalizzata impressa sul nastro.
Poiché il ritardo modulato può assumere valori non interi, il campione prelevato dalla linea di ritardo viene determinato mediante interpolazione lineare. Definendo la parte intera del ritardo come:

$$D_{int}[n] = \lfloor D_{mod}[n] \rfloor \tag{2.17}$$

e la parte frazionaria come:

$$\mu[n] = D_{mod}[n] - D_{int}[n], \tag{2.18}$$

il campione prelevato dalla linea di ritardo può essere espresso come:

$$y_{read}[n] = (1 - \mu[n]) y_{raw}[n - D_{int}[n]] + \mu[n] y_{raw}[n - D_{int}[n] - 1]. \tag{2.19}$$

Il campione $y_{read}[n]$ rappresenta il segnale magnetico raggiunto dalla testina di riproduzione dopo il transito sul nastro. A tale segnale viene successivamente applicato il modello delle perdite della testina, ottenendo il segnale $y_{filt}[n]$ descritto nella sezione successiva.
Il segnale di feedback viene quindi ottenuto applicando il coefficiente g al segnale riprodotto:

$$y_{fb}[n] = g\, y_{filt}[n] \tag{2.20}$$

Il campo magnetico associato al segnale audio e al feedback è calcolato come:

$$H_{sig}[n] = drive \cdot M_s \left(x_{up}[n] + y_{fb}[n] \right), \tag{2.21}$$

dove *drive* controlla il livello di eccitazione del modello. Nel caso considerato, viene utilizzato un valore pari a 0.3, mentre $M_s = 3.5 \cdot 10^5$ A/m.
Come anticipato nella Sezione 2.1, nel processo di registrazione magnetica al segnale audio viene sovrapposto un segnale sinusoidale ad alta frequenza, denominato bias. Esso ha lo scopo di superare la resistenza iniziale alla magnetizzazione dei domini magnetici del nastro, consentendo al segnale audio di operare in una regione più lineare della caratteristica di magnetizzazione. Viene definito come:

$$H[n] = H_{sig}[n] + A_b \sin\left(2\pi f_b \frac{n}{F_{s,up}}\right) \tag{2.22}$$

dove A_b e f_b rappresentano rispettivamente l'ampiezza e la frequenza del segnale di bias. Nell'implementazione considerata:

$$A_b = 0.01\, M_s, \quad f_b = 55 \text{ kHz}. \tag{2.23}$$

Il campo magnetico $H[n]$ così ottenuto costituisce il segnale di eccitazione utilizzato nella successiva integrazione numerica del modello di Jiles-Atherton.

2.3.2 Risoluzione tramite metodo di Runge-Kutta

Il modello di Jiles–Atherton è caratterizzato da una marcata non linearità, dovuta alla relazione non lineare tra il campo magnetico applicato H e la magnetizzazione M. In particolare, la magnetizzazione anisteretica è espressa mediante la funzione di Langevin, che introduce una relazione non lineare tra il campo efficace e la magnetizzazione. Inoltre, il modello descrive la saturazione del materiale e la dipendenza della magnetizzazione dalla storia del campo applicato, dando origine al comportamento di isteresi.
Dal punto di vista della risoluzione numerica, un'equazione differenziale viene definita *stiff*, o rigida, quando la presenza di dinamiche caratterizzate da scale temporali differenti rende necessario utilizzare passi temporali sufficientemente piccoli per mantenere la stabilità della soluzione. In un sistema rigido, infatti, anche quando la soluzione varia lentamente, un passo temporale troppo grande può causare instabilità numeriche. Nel caso del modello considerato, Chowdhury [6] osserva che i metodi di ordine inferiore comunemente utilizzati nei sistemi audio, come la regola del trapezio, possono diventare rapidamente instabili con ingressi ad alta frequenza, in particolare quando il segnale è modulato dal bias. Per questo motivo è stato adottato il metodo di Runge–Kutta del quarto ordine (RK4) per la risoluzione numerica dell'equazione differenziale.

Indicando sinteticamente l'Equazione 2.8 nella forma:

$$\frac{dM}{dt} = f\left(H(t), \frac{dH(t)}{dt}, M(t)\right) \tag{2.24}$$

La variabile di stato, al campione temporale n-esimo, viene aggiornata a partire dal campione precedente $n-1$ tramite la somma pesata di quattro coefficienti k_i, che valutano la pendenza della funzione in diversi punti dell'intervallo di integrazione $T = 1/F_{s,up}$:

$$M[n] = M[n-1] + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{2.25}$$

I coefficienti k_i per l'avanzamento dello stato magnetico sono definiti come:

$$k_1 = T \cdot f\left(H[n-1], H'[n-1], M[n-1]\right) \tag{2.26}$$

$$k_2 = T \cdot f\left(H_{half}, H'_{half}, M[n-1] + \frac{k_1}{2}\right) \tag{2.27}$$

$$k_3 = T \cdot f\left(H_{half}, H'_{half}, M[n-1] + \frac{k_2}{2}\right) \tag{2.28}$$

$$k_4 = T \cdot f\left(H[n], H'[n], M[n-1] + k_3\right) \tag{2.29}$$

Per calcolare le valutazioni intermedie del metodo RK4 (k_2 e k_3), sono necessari i valori del campo magnetico applicato e della sua derivata temporale nel punto medio dell'intervallo temporale tra i campioni $n-1$ e n.
Assumendo che il segnale eccitante vari in modo sufficientemente regolare nell'intervallo temporale T, si utilizza un'interpolazione lineare:

$$H_{half} = \frac{H[n] + H[n-1]}{2} \tag{2.30}$$

Inoltre, la derivata temporale del campo magnetico eccitante H', necessaria all'interno della funzione f, deve essere stimata numericamente a ogni passo. A tale scopo viene utilizzata una discretizzazione trapezoidale della derivata. Partendo dalla relazione

$$H[n] - H[n-1] = \frac{T}{2}\left(H'[n] + H'[n-1]\right), \tag{2.31}$$

Si ricava ricorsivamente:

$$H'[n] = \frac{2}{T}\left(H[n] - H[n-1]\right) - H'[n-1]. \tag{2.32}$$

Per calcolare le valutazioni intermedie del metodo RK4, anche la derivata temporale viene stimata al punto medio dell'intervallo tra i campioni $n-1$ e n mediante interpolazione lineare:

$$H'_{half} = \frac{H'[n] + H'[n-1]}{2}. \tag{2.33}$$

I valori di $H[n]$, $H'[n]$ e delle rispettive grandezze al punto medio costituiscono quindi gli ingressi necessari per le quattro valutazioni della funzione f previste dal metodo RK4. A partire da tali valutazioni, la magnetizzazione del nastro viene aggiornata a ogni campione secondo l'Equazione 2.25, ottenendo il nuovo valore $M[n]$.
L'algoritmo così definito consente pertanto di integrare numericamente il modello di Jiles-Atherton nel dominio discreto. Al termine di ogni iterazione, il valore di $M[n]$ viene normalizzato rispetto a M_s, ottenendo il segnale $y_{\text{raw}}[n]$ associato alla magnetizzazione del nastro.

2.4 Modellazione della testina di riproduzione e delle perdite

Durante la fase di riproduzione, la testina non restituisce idealmente il segnale magnetico presente sul nastro. La risposta del sistema di lettura è infatti caratterizzata da un'attenuazione dipendente dalla frequenza, dovuta principalmente alla distanza tra nastro e testina, allo spessore dello strato magnetico e alle dimensioni del traferro della testina. Nel modello implementato tali fenomeni vengono rappresentati mediante tre contributi distinti, indicati rispettivamente come *spacing loss*, *thickness loss* e *gap loss*.
La risposta complessiva della testina viene ottenuta come prodotto delle singole componenti:

$$H_{\text{loss}}(f) = H_{\text{spacing}}(f) H_{\text{thickness}}(f) H_{\text{gap}}(f). \tag{2.34}$$

Nel seguito vengono descritte le tre componenti utilizzate nell'implementazione.

2.4.1 Spacing loss

La *spacing loss* tiene conto dell'attenuazione dovuta alla distanza tra la superficie del nastro e la testina di riproduzione. Indicando con d tale distanza, con f la frequenza del segnale e con v la velocità di scorrimento del nastro, si definisce:

$$k(f) = \frac{2\pi f}{v}, \tag{2.35}$$

da cui la risposta associata alla distanza tra nastro e testina è:

$$H_{\text{spacing}}(f) = e^{-k(f)d}. \tag{2.36}$$

La relazione evidenzia che l'attenuazione aumenta con la frequenza e con la distanza tra nastro e testina.

2.4.2 Thickness loss

La *thickness loss* tiene conto del fatto che lo strato magnetico possiede uno spessore finito. Indicando con δ lo spessore dello strato magnetico, il contributo considerato nel modello è:

$$H_{\text{thickness}}(f) = \frac{1 - e^{-k(f)\delta}}{k(f)\delta}. \tag{2.37}$$

Anche questo contributo introduce un'attenuazione crescente al crescere della frequenza.

2.4.3 Gap loss

Il terzo contributo è associato al traferro della testina di riproduzione. Se g_{gap} rappresenta la dimensione del traferro, la relativa risposta viene modellata mediante:

$$H_{\text{gap}}(f) = \text{sinc}\left(\frac{k(f)g_{\text{gap}}}{2\pi}\right), \tag{2.38}$$

dove la funzione sinc è definita, secondo la convenzione utilizzata nell'implementazione MATLAB, come:

$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}. \tag{2.39}$$

Combinando i tre contributi si ottiene quindi la risposta complessiva $H_{\text{loss}}(f)$ della testina.

Nel modello utilizzato per l'emulazione del Sony TC-260, i parametri fisici adottati sono:

$$d = 20\,\mu\text{m}, \qquad \delta = 35\,\mu\text{m}, \qquad g_{\text{gap}} = 2\,\mu\text{m}, \tag{2.40}$$

mentre la velocità di scorrimento del nastro è:

$$v = 0.1905\,\text{m/s}, \tag{2.41}$$

corrispondente a 7.5 ips.

2.4.4 Implementazione mediante filtro FIR

La risposta $H_{\text{loss}}(f)$ viene calcolata alla frequenza di campionamento sovracampionata $F_{s,up}$. Per ottenere una rappresentazione nel dominio del tempo, la risposta in frequenza viene resa simmetrica e successivamente trasformata mediante la trasformata inversa discreta di Fourier:

$$h[n] = \text{IFFT}\left\{H_{\text{loss}}(f)\right\}. \tag{2.42}$$

La risposta impulsiva ottenuta viene quindi centrata temporalmente tramite *fftshift*. Poiché una risposta ottenuta tramite IFFT presenta una durata finita ma generalmente superiore a quella necessaria per l'elaborazione, viene effettuato un troncamento FIR di ordine 300.
Per ridurre le discontinuità introdotte dal troncamento, la risposta impulsiva viene moltiplicata per una finestra di Hann:

$$h_{\text{FIR}}[n] = h[n]\, w_{\text{Hann}}[n]. \tag{2.43}$$

Il filtro FIR così ottenuto viene applicato al segnale prelevato dalla linea di ritardo. In questo modo, il segnale che rappresenta la lettura del nastro tiene conto sia del ritardo dovuto alla separazione tra le testine sia dell'attenuazione indotta dai fenomeni fisici di lettura.
Il segnale risultante viene indicato con $y_{\text{filt}}[n]$ ed è successivamente utilizzato sia per il percorso di uscita sia per la generazione del feedback:

$$y_{\text{fb}}[n] = g\, y_{\text{filt}}[n]. \tag{2.44}$$

2.5 Debias e conversione alla frequenza di campionamento originale

Al termine della fase di riproduzione, il segnale $y_{\text{filt}}[n]$ contiene ancora la componente ad alta frequenza associata al bias utilizzato durante la registrazione. Prima di riportare il segnale alla frequenza di campionamento originale, è quindi necessario attenuare tale componente e limitare la banda del segnale per evitare fenomeni di aliasing durante la successiva decimazione.

Il primo filtraggio ha la funzione di rimuovere il residuo del bias e viene realizzato mediante un filtro passa-basso Butterworth del quarto ordine, con frequenza di taglio pari a:

$$f_c = 24\,\text{kHz}. \tag{2.45}$$

Indicando con $y_{\text{db}}[n]$ il segnale dopo il filtraggio di debias, si ha:

$$y_{\text{db}}[n] = \text{LPF}_{24\,\text{kHz}}\left\{y_{\text{filt}}[n]\right\}. \tag{2.46}$$

Nella simulazione offline in ambiente MATLAB, il filtro di debias viene implementato mediante una struttura IIR causale, utilizzando la funzione `filter`. In questo modo il segnale risultante mantiene la struttura temporale necessaria per la successiva elaborazione.

Il filtro a 24 kHz non costituisce, tuttavia, l'unico filtraggio impiegato prima del downsampling. Poiché il segnale viene elaborato alla frequenza $F_{s,up} = 16F_s$, è necessario limitarne ulteriormente la banda prima della decimazione. A tale scopo viene progettato un secondo filtro passa-basso FIR con frequenza di passband pari a 18 kHz e frequenza di stopband pari a 20 kHz:

$$f_{\text{pass}} = 18\,\text{kHz}, \qquad f_{\text{stop}} = 20\,\text{kHz}. \tag{2.47}$$

Il filtro è progettato con un'ondulazione massima in banda passante pari a 1 dB e un'attenuazione minima in banda di stop pari a 60 dB. Il suo compito è limitare le componenti spettrali che potrebbero ricadere all'interno della banda utile a seguito della decimazione.

Nella simulazione offline, il filtro anti-aliasing viene applicato tramite la funzione `filtfilt`. Tale procedura realizza un filtraggio a fase zero, applicando il filtro in entrambe le direzioni temporali, evitando l'introduzione di uno sfasamento netto del segnale. Questa modalità di elaborazione è adatta alla fase di validazione offline, ma non è causale e non può essere utilizzata direttamente in un'elaborazione in tempo reale.

Indicando con $y_{\text{AA}}[n]$ il segnale ottenuto dopo il filtraggio anti-aliasing, la conversione alla frequenza di campionamento originale viene effettuata mediante la funzione `resample`, con un rapporto di conversione pari a $1/OS$.

Nel caso considerato, con $OS = 16$, il segnale viene quindi riportato dalla frequenza di campionamento interna $F_{s,\text{up}}$ alla frequenza originale F_s.

Il segnale risultante costituisce l'uscita del modello di tape delay e viene infine normalizzato rispetto al massimo valore assoluto prima della riproduzione audio.

Capitolo 3

VALIDAZIONE MATLAB

Prima di procedere all'implementazione in tempo reale, l'intera catena di elaborazione è stata prototipata e verificata in ambiente MATLAB. L'obiettivo di questa fase è valutare il corretto funzionamento dei principali blocchi del modello e verificare la coerenza dei risultati ottenuti con il comportamento atteso e, dove possibile, con i risultati riportati in letteratura [6].

Per analizzare il comportamento dinamico del modello, il sistema è stato alimentato con un segnale audio di test (`audio.wav`), costituito dalla registrazione di un applauso. La Figura 3.1 mostra il confronto nel dominio del tempo tra il segnale originale e l'output complessivo generato dalla catena.

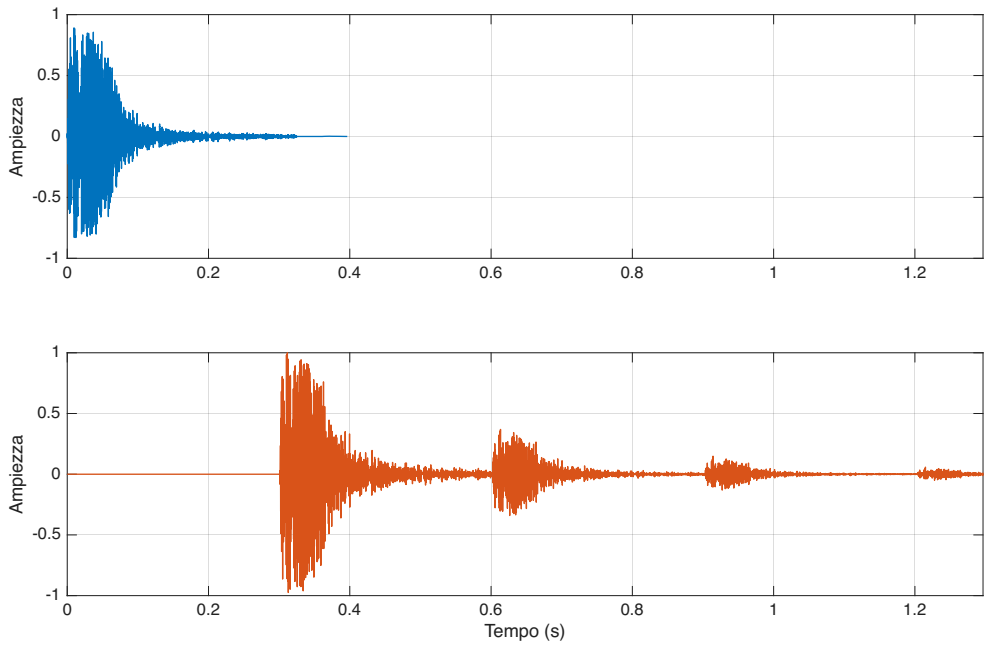


Figura 3.1: Confronto nel dominio del tempo tra il segnale audio di ingresso (in alto) e l'output processato dal tape delay (in basso).

Come si osserva nel pannello inferiore della Figura 3.1, la prima ripetizione (eco) a circa 0.3 secondi presenta un'ampiezza e una densità energetica rimodulate rispetto al segnale pulito originale. Questo fenomeno è la conseguenza diretta della sovrapposizione algebrica nel nodo d'ingresso tra il segnale fresco e il feedback ritardato, unita alla compressione e alla saturazione imposte dal modello di isteresi magnetica.

3.1 Validazione del ciclo di isteresi dinamica

Il primo test condotto sul modello numerico ha avuto lo scopo di verificare la corretta implementazione dell'equazione differenziale integrata tramite il metodo RK4, confrontando il ciclo di isteresi ottenuto con la letteratura di riferimento [6].
Al fine di riprodurre visivamente il ciclo di isteresi riportato nello studio originale e validare geometricamente la forma della curva nel range di ampiezze di test descritto (800 - 2000 A/m), è stato scelto di scalare temporaneamente i parametri fisici del materiale ($a = 1000$ A/m, $k = 400$ A/m). I valori reali del registratore Sony TC-260 ($a = 22 \times 10^3$ A/m e $k = 27 \times 10^3$ A/m) richiederebbero infatti campi di eccitazione nettamente superiori per raggiungere l'effettiva saturazione.
Come mostrato in Figura 3.2, eccitando il sistema con questa configurazione, l'algoritmo riesce a ricostruire fedelmente il ciclo di isteresi nel range considerato. L'andamento della curva evidenzia che, nel test considerato, il sovracampionamento adottato ($OS = 16$) consente al solutore RK4 di mantenere una traiettoria stabile e chiusa. Per la successiva elaborazione dei segnali audio, il sistema è stato ovviamente ripristinato con i parametri fisici reali, regolando di conseguenza il guadagno in ingresso (*Drive*).

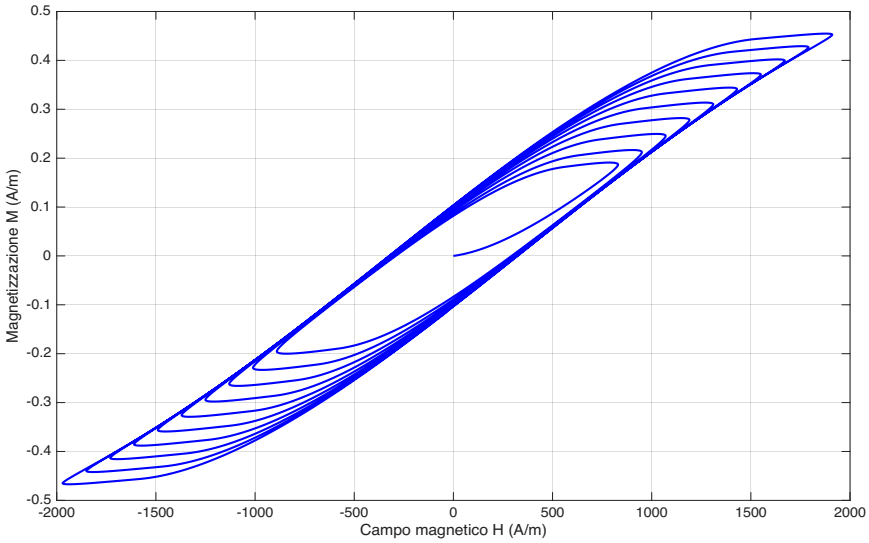


Figura 3.2: Ciclo di isteresi dinamica ottenuto tramite la simulazione numerica in MATLAB. I parametri di forma sono stati opportunamente scalati per consentire la visualizzazione della saturazione nel range di test.

3.2 Analisi spettrale e temporale della distorsione

Oltre all'analisi del ciclo di isteresi, è stato valutato l'effetto della non linearità sul contenuto spettrale del sistema. Prima del calcolo della Trasformata Discreta di Fourier (DFT), ai segnali è stata applicata una finestra di Hann al fine di mitigare il fenomeno della dispersione spettrale (*spectral leakage*) dovuto al troncamento temporale.
La Figura 3.3 mostra il confronto spettrale tra il segnale in ingresso e quello in uscita dal modello di isteresi. Il segnale di ingresso è costituito dal file `audio.wav`, contenente una registrazione di un applauso, precedentemente sottoposta a sovracampionamento con fattore $OS = 16$.
Nel segnale in uscita (curva arancione) si osserva una maggiore presenza di componenti alle frequenze più elevate, riconducibile alla natura non lineare del modello di Jiles-Atherton. È inoltre chiaramente distinguibile una componente in corrispondenza di 55 kHz, associata al tono di bias introdotto prima del modello di isteresi.

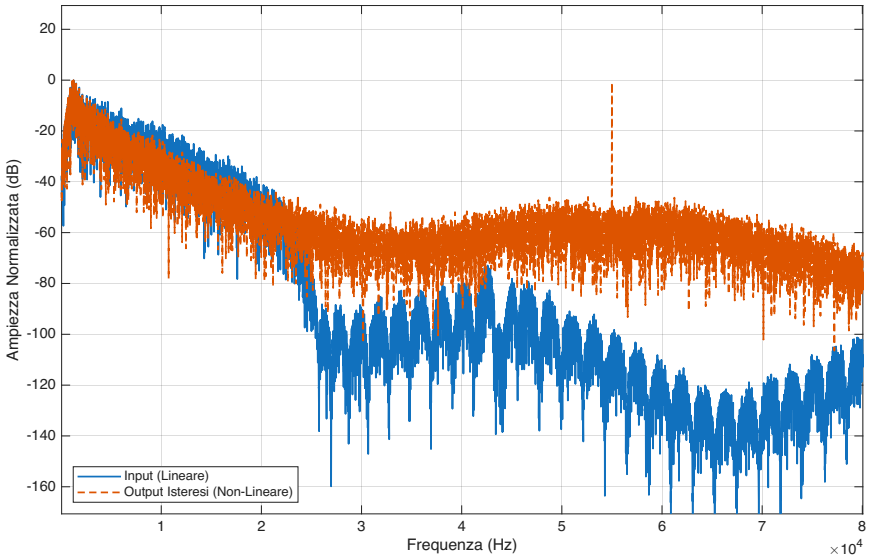


Figura 3.3: Validazione spettrale: confronto tra il segnale in ingresso (blu) e in uscita dal modello di isteresi (arancione), mediante DFT con finestra di Hann. Si osserva l'incremento di energia in alta frequenza dovuto alla non linearità e la componente a 55 kHz associata al bias.

Dal punto di vista temporale, il passaggio attraverso il ciclo di isteresi introduce uno smussamento dei picchi (*soft-clipping*), derivante dall'avvicinamento alla magnetizzazione di saturazione, visibile in Figura 3.4.

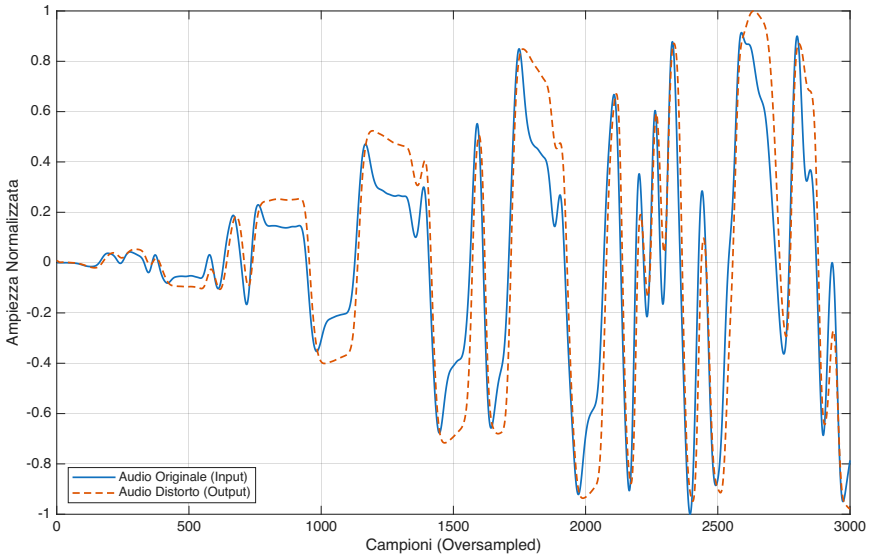


Figura 3.4: Validazione temporale nel dominio dei campioni. Si osserva lo smussamento dei picchi di maggiore ampiezza associato alla non linearità e alla saturazione del modello.

3.3 Validazione delle perdite della testina (Loss Effects)

Il segnale letto dal nastro subisce un'alterazione in frequenza dovuta alla geometria della testina di riproduzione. Per validare il filtro FIR a fase lineare progettato nella Sezione 2.4.4, è stata condotta una duplice analisi.

In primo luogo, la risposta in frequenza del filtro sintetizzato (tramite IFFT e finestratura di Hann) è stata confrontata direttamente con la curva di attenuazione teorica continua, ricavata dai parametri fisici di *spacing*, *thickness* e *gap loss*. Come si evince dalla Figura 3.5, l'approssimazione FIR si sovrappone al modello ideale, dimostrando la correttezza matematica della sintesi.

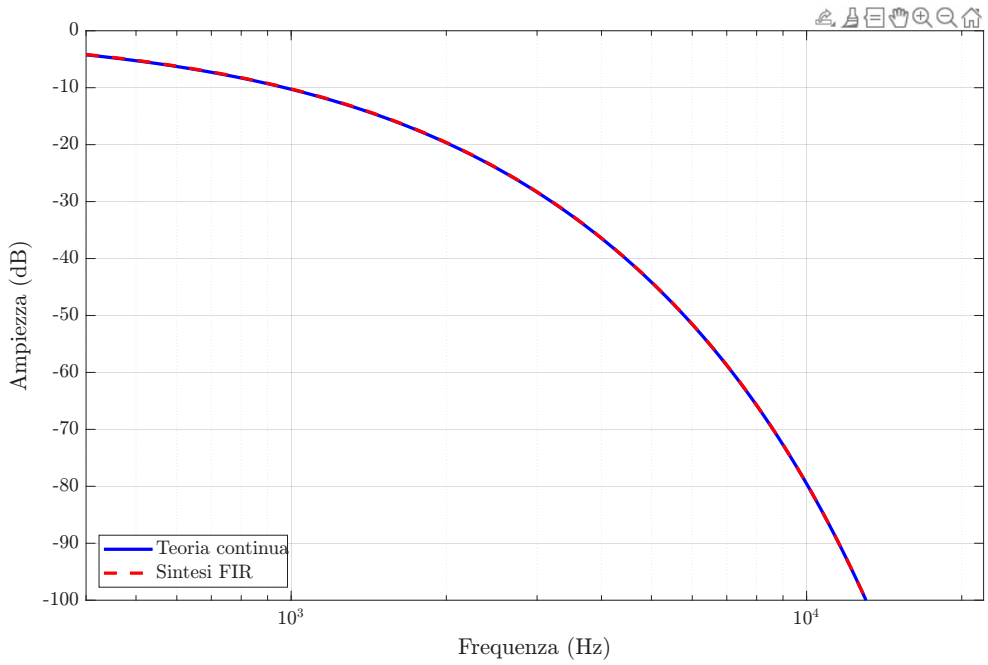


Figura 3.5: Confronto in scala logaritmica tra la risposta in frequenza teorica della testina (curva blu continua) e l'approssimazione ottenuta tramite il filtro FIR (curva rossa tratteggiata).

In secondo luogo, per valutare l'impatto del filtraggio sul segnale audio, è stato calcolato lo spettro del segnale prima e dopo l'applicazione del filtro FIR. La Figura 3.6 mostra l'attenuazione progressiva delle componenti alle alte frequenze introdotta dal modello della testina di riproduzione.

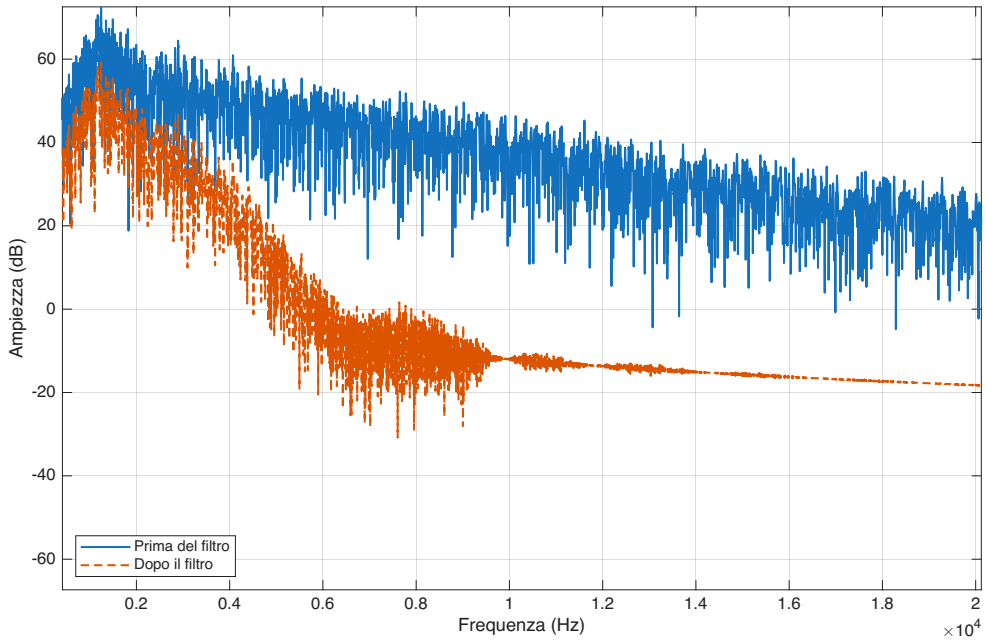
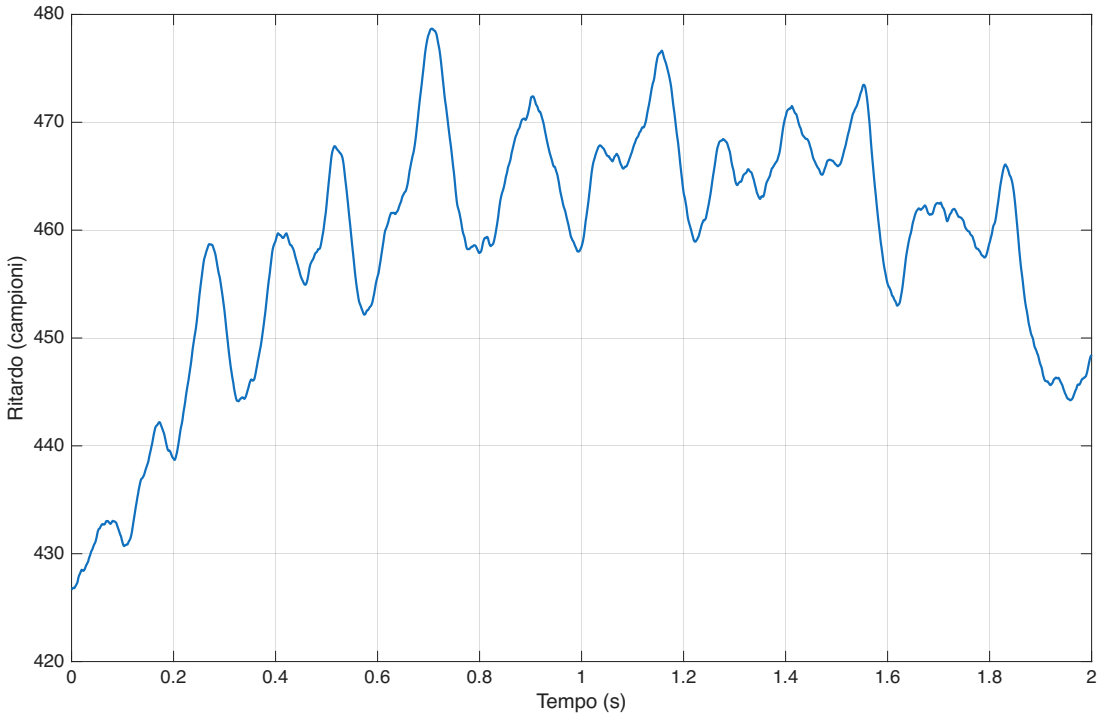


Figura 3.6: Effetto del filtro FIR sul contenuto spettrale di un segnale audio. Il grafico mostra lo spettro prima dell'applicazione del filtro (in blu) e dopo (in arancione), evidenziando il taglio netto alle alte frequenze introdotto dal modello della testina di riproduzione.

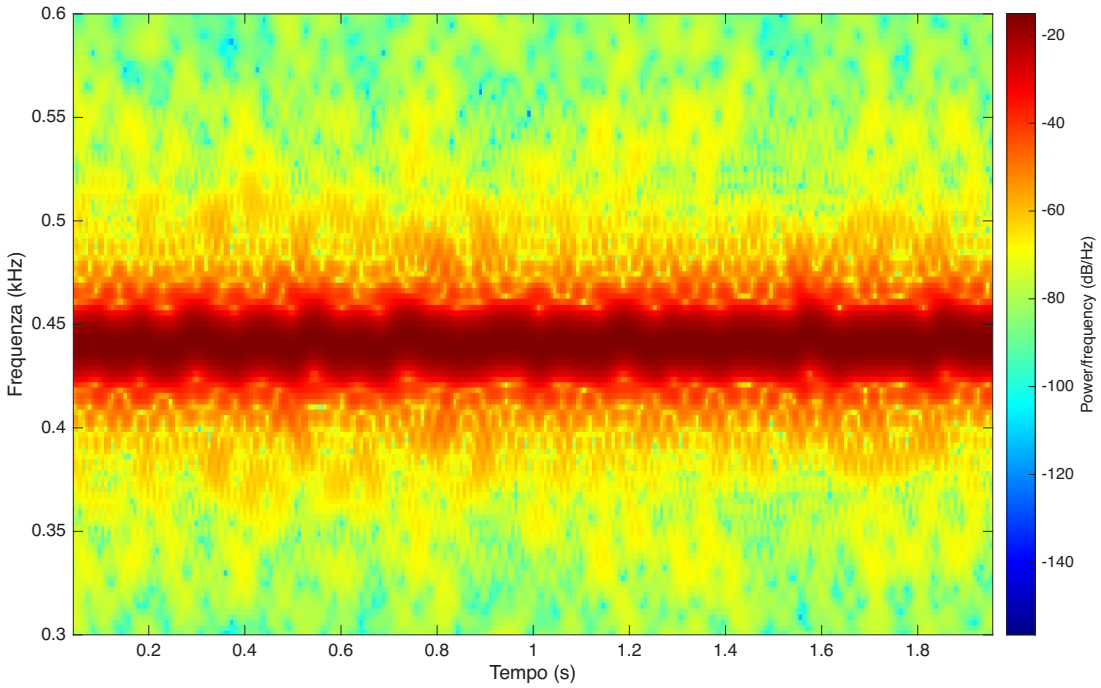
3.4 Analisi del Wow e Flutter

Per verificare il comportamento della modulazione temporale del ritardo, è stata utilizzata una sinusoide a 440 Hz come segnale di test. La Figura 3.7a mostra l'andamento del ritardo modulato nel tempo rispetto al valore nominale. La modulazione osservata è dovuta alla combinazione delle componenti di Wow e Flutter generate dal modello.

L'effetto della modulazione sul segnale risultante è analizzato mediante lo spettrogramma riportato in Figura 3.7b. Si osserva una variazione nel tempo della componente a 440 Hz, associata alla variazione temporale del ritardo introdotto dalle due componenti di modulazione.



(a) Variazione temporale del ritardo rispetto al valore nominale.



(b) Variazione della componente a 440 Hz nel dominio tempo-frequenza.

Figura 3.7: Analisi dell'effetto delle modulazioni di Wow e Flutter sulla linea di ritardo.

Nel complesso, i test effettuati hanno permesso di verificare separatamente i principali comportamenti introdotti dal modello MATLAB. Il ciclo di isteresi è stato verificato

sia dal punto di vista qualitativo sia mediante il confronto con il riferimento bibliografico, mentre l'analisi spettrale e temporale ha evidenziato gli effetti della non linearità sul segnale audio. La risposta delle perdite della testina è risultata coerente con il modello analitico da cui è stato ricavato il filtro FIR, e l'analisi del Wow e Flutter ha mostrato l'effetto della modulazione temporale sulla linea di ritardo.

La versione MATLAB così verificata costituisce quindi la base per la successiva fase di sviluppo in tempo reale, nella quale vengono affrontati i problemi di stabilità numerica, costo computazionale e adattamento dei blocchi di elaborazione ai vincoli di esecuzione in C++.

Capitolo 4

IMPLEMENTAZIONE IN C++

4.1 Porting del modello e integrazione in VCV Rack

La versione MATLAB descritta nei capitoli precedenti ha permesso di sviluppare e validare offline la catena di elaborazione del tape delay. La fase successiva ha riguardato il porting del modello in linguaggio C++, con l'obiettivo di ottenere un'implementazione eseguibile in tempo reale all'interno dell'ambiente VCV Rack.
Il passaggio da un'implementazione offline a una real-time introduce un vincolo fondamentale: il segnale deve essere elaborato progressivamente durante l'esecuzione, senza poter disporre anticipatamente dell'intera sequenza di campioni. Ogni chiamata al ciclo di processamento deve quindi produrre l'uscita corrispondente all'ingresso ricevuto, mantenendo tra campioni successivi tutte le informazioni necessarie a descrivere lo stato del sistema. Nel porting effettuato, questa esigenza ha portato alla riorganizzazione del modello in blocchi di elaborazione con memoria di stato, ciascuno dedicato alla gestione di una specifica parte della catena.
Il modulo principale è implementato dalla struttura `MioTapeDelay`, derivata dalla classe `Module` fornita dal framework VCV Rack [7]. Il modulo dispone di un ingresso e di un'uscita audio e mette a disposizione sei parametri regolabili dall'utente: *Delay Time*, *Feedback*, *Drive*, *Dry/Wet*, *wow & flutter* e *Tape Loss*. I relativi intervalli e valori iniziali vengono definiti nella fase di configurazione del modulo. L'adozione di controlli interattivi permette quindi di modificare durante l'esecuzione i principali parametri dell'effetto, trasformando il modello offline in un sistema utilizzabile direttamente come modulo audio.
Per i filtri passa-basso impiegati nel percorso real-time è stata inoltre definita una

struttura `Biquad`, che implementa un filtro IIR del secondo ordine attraverso i coefficienti del numeratore e del denominatore e due variabili di stato. I coefficienti vengono determinati a partire dalla frequenza di taglio, dalla frequenza di campionamento e dal fattore di qualità Q.
La modellazione della risposta della testina di riproduzione è invece gestita dalla struttura `PhysicalPlaybackFilter`. Quest'ultima calcola la risposta in frequenza teorica a partire dai parametri fisici di spacing, thickness e gap loss e costruisce successivamente una risposta FIR finita da utilizzare durante l'elaborazione del segnale. La separazione di questi blocchi consente di mantenere distinta la modellazione fisica del sistema dalle operazioni necessarie alla sua elaborazione in tempo reale. È stata inoltre introdotta una struttura dedicata alla pre-enfasi del segnale di registrazione, applicata prima dell'ingresso nel modello magnetico. Tale blocco mantiene uno stato relativo al campione precedente e modifica il segnale audio prima della conversione nel campo magnetico.
La struttura `MioTapeDelay` mantiene inoltre le variabili di stato necessarie al funzionamento continuo dell'algoritmo. In particolare, il modello magnetico conserva il valore precedente della magnetizzazione M_{prev}, del campo applicato H_{prev} e della sua derivata temporale $\dot{H}_{\text{prev}}$. A queste si aggiungono la fase del segnale di bias, l'ultimo campione prodotto dal ramo wet e le variabili utilizzate per la generazione delle modulazioni di wow e flutter. Il sistema dispone infine di un buffer dedicato alla linea di ritardo, implementato come vettore di campioni a dimensione fissa.
Questa organizzazione consente di conservare lo stato del sistema tra successive chiamate alla funzione di processamento, condizione necessaria per l'esecuzione corretta del modello in tempo reale.

4.2 Stabilizzazione del solutore numerico

Uno degli aspetti più critici del porting del modello riguarda la risoluzione numerica dell'equazione differenziale di Jiles–Atherton. Come descritto nella Sezione 2.3.2, il modello viene integrato mediante il metodo di Runge–Kutta del quarto ordine (RK4). Nell'implementazione real-time è necessario mantenere un passo temporale sufficientemente piccolo, poiché l'equazione presenta una dinamica non lineare e deve inoltre essere elaborata in presenza del segnale di bias ad alta frequenza.
Per questo motivo, il modello C++ non viene valutato direttamente alla sola frequenza di campionamento del segnale audio. Il segnale viene innanzitutto sovracampionato di un fattore

$$OS = 16, \tag{4.1}$$

portando l'elaborazione interna a una frequenza pari a

$$F_{s,\mathrm{up}} = 16F_s. \tag{4.2}$$

Il blocco di sovracampionamento utilizza due filtri FIR distinti: il primo viene impiegato nella fase di aumento della frequenza di campionamento e svolge la funzione di anti-imaging, mentre il secondo viene utilizzato nella fase di riduzione della frequenza e svolge la funzione di anti-aliasing. L'upsampling viene realizzato producendo i campioni intermedi attraverso l'inserimento di valori nulli e il successivo filtraggio FIR, mentre nella fase di downsampling i campioni prodotti internamente vengono filtrati prima della selezione del campione di uscita.

Il solo sovracampionamento non costituisce tuttavia l'unico accorgimento adottato per la stabilizzazione dell'integrazione. Ogni campione prodotto nel dominio sovracampionato viene ulteriormente suddiviso in quattro sotto-intervalli temporali all'interno della funzione `processHysteresisSample`. Indicando con T_s il periodo di campionamento alla frequenza audio, il passo temporale utilizzato dal solutore risulta quindi:

$$\Delta t = \frac{T_s}{OS\, N_{\mathrm{sub}}}, \tag{4.3}$$

dove, nel modello implementato,

$$OS = 16, \qquad N_{\mathrm{sub}} = 4. \tag{4.4}$$

Il passo interno corrisponde pertanto a una suddivisione complessiva dell'intervallo di campionamento audio in 64 sotto-intervalli.

All'interno di ciascun sotto-intervallo il campo magnetico viene fatto variare linearmente tra il valore precedente e quello successivo. Se H_{prev} rappresenta il campo al termine del sotto-passo precedente e H_{target} il valore complessivo da raggiungere, l'incremento utilizzato è:

$$\Delta H = \frac{H_{\mathrm{target}} - H_{\mathrm{prev}}}{N_{\mathrm{sub}}}. \tag{4.5}$$

Il valore successivo del campo viene quindi calcolato come

$$H_{i+1} = H_i + \Delta H. \tag{4.6}$$

La derivata del campo viene ottenuta dal rapporto tra la variazione di H e il passo temporale interno. Per le valutazioni intermedie di RK4 vengono inoltre utilizzati i valori medi del campo e della sua derivata tra l'inizio e la fine del sotto-intervallo.
Per ogni sotto-intervallo vengono quindi calcolati i quattro coefficienti del metodo RK4:

$$k_1 = \Delta t\, f(H_i, \dot{H}_i, M_i), \tag{4.7}$$

$$k_2 = \Delta t\, f(H_{i+1/2}, \dot{H}_{i+1/2}, M_i + k_1/2), \tag{4.8}$$

$$k_3 = \Delta t\, f(H_{i+1/2}, \dot{H}_{i+1/2}, M_i + k_2/2), \tag{4.9}$$

$$k_4 = \Delta t\, f(H_{i+1}, \dot{H}_{i+1}, M_i + k_3). \tag{4.10}$$

La magnetizzazione viene quindi aggiornata secondo:

$$M_{i+1} = M_i + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6}. \tag{4.11}$$

Al termine dei quattro sotto-passi, i valori aggiornati di M, H e $\dot{H}$ vengono memorizzati come stato iniziale per il successivo campione. L'introduzione del sovracampionamento e del sub-stepping consente quindi di aumentare la risoluzione temporale con cui viene integrato il modello magnetico, riducendo la variazione del campo considerata a ogni singolo passo del solutore. Questa strategia è particolarmente rilevante nel contesto real-time, nel quale il modello deve gestire contemporaneamente la dinamica non lineare della magnetizzazione e il segnale di bias ad alta frequenza.
L'approccio adottato rappresenta pertanto un compromesso tra accuratezza numerica e costo computazionale: l'impiego di quattro valutazioni interne per ciascun sotto-passo RK4, ripetute nel dominio sovracampionato, aumenta il numero di operazioni richieste, ma permette di mantenere una risoluzione temporale più elevata rispetto all'elaborazione diretta alla frequenza audio.

4.3 Implementazione della linea di ritardo e del feedback

Una volta completata l'elaborazione del segnale attraverso il modello di isteresi, è necessario ricostruire in ambiente digitale il comportamento della linea di ritardo che caratterizza il tape delay. Nell'implementazione C++, il segnale magnetizzato ottenuto dopo il downsampling viene memorizzato all'interno di un buffer circolare, dal quale viene successivamente letto con un ritardo variabile.

A differenza della fase di integrazione del modello magnetico, che viene eseguita alla frequenza sovracampionata, la linea di ritardo opera alla frequenza di campionamento audio. Il campione prodotto dal modello di isteresi viene quindi prima riportato alla frequenza originale e successivamente scritto nel buffer della linea di ritardo. La memoria utilizzata per la linea di ritardo è organizzata come buffer circolare di dimensione fissa. L'indice di scrittura viene incrementato a ogni campione e, raggiunta la fine del buffer, riparte automaticamente dall'inizio. Questa struttura permette di realizzare una linea di ritardo continua senza dover spostare fisicamente i campioni in memoria.

Il tempo di ritardo viene determinato dal parametro *Delay Time* e convertito in un numero di campioni mediante la relazione:

$$D[n] = T_d[n]F_s, \tag{4.12}$$

dove $T_d[n]$ rappresenta il tempo di ritardo istantaneo e F_s la frequenza di campionamento del segnale audio. La posizione di lettura all'interno del buffer viene quindi ottenuta sottraendo tale quantità all'indice corrente di scrittura. Poiché il valore del ritardo può non corrispondere a un numero intero di campioni, la posizione di lettura può assumere un valore frazionario. Per evitare una quantizzazione del tempo di ritardo, il campione viene ricostruito mediante interpolazione lineare tra i due campioni adiacenti.

Indicando con $y_1[n]$ e $y_2[n]$ i campioni corrispondenti alle due posizioni intere e con $\mu[n]$ la parte frazionaria dell'indice di lettura, il campione ritardato è calcolato come:

$$y_{\text{del}}[n] = y_1[n] + \mu[n]\,(y_2[n] - y_1[n])\,. \tag{4.13}$$

L'interpolazione lineare consente quindi di ottenere una variazione continua del tempo di ritardo e risulta particolarmente importante quando la posizione della lettura viene modificata nel tempo dalle modulazioni di wow e flutter. Il comportamento tipico del tape delay è ottenuto introducendo un ramo di retroazione che reinietta nell'ingresso una parte del segnale precedentemente riprodotto. Nel modello implementato, il segnale utilizzato per la retroazione è il campione *wet* prodotto al passo precedente. Il segnale effettivamente inviato al processo di registrazione è pertanto:

$$x_{\text{rec}}[n] = x[n] + g\, y_{\text{wet}}[n-1], \tag{4.14}$$

dove $x[n]$ è il segnale di ingresso, $y_{\text{wet}}[n-1]$ rappresenta l'uscita elaborata al campione precedente e g è il coefficiente di feedback.

Nel modulo C++, il valore del parametro *Feedback* viene inoltre moltiplicato per un fattore di sicurezza pari a 0.95, così da mantenere un margine rispetto alla condizione teorica di feedback unitario. Il segnale risultante viene quindi inviato al blocco di sovracampionamento e successivamente al modello magnetico. La combinazione tra linea di ritardo, interpolazione e retroazione consente di ricostruire il comportamento fondamentale del tape delay nel dominio digitale. A ogni iterazione il campione elaborato viene registrato nel buffer, mentre una versione temporalmente ritardata viene riprodotta e reinserita nel percorso di elaborazione, generando una successione di ripetizioni con ampiezza progressivamente decrescente.

4.4 Generazione real-time di wow e flutter

Nel funzionamento di un registratore a nastro, le variazioni della velocità di trascinamento del nastro producono piccole variazioni temporali del segnale riprodotto. Nel modello digitale tale comportamento viene simulato modificando dinamicamente la posizione di lettura della linea di ritardo.
Nell'implementazione real-time non viene generata una forma d'onda deterministica con frequenza costante, ma viene introdotta una modulazione stocastica del tempo di ritardo. A questo scopo vengono mantenuti separatamente due inviluppi, associati rispettivamente a wow e flutter, caratterizzati da differenti intervalli di aggiornamento. Per ciascuna componente vengono memorizzati un valore iniziale e un valore obiettivo. Una volta trascorso l'intervallo temporale prefissato, il valore obiettivo viene aggiornato mediante una nuova estrazione casuale e l'interpolazione tra il vecchio e il nuovo valore viene effettuata progressivamente durante l'intervallo successivo.
Indicando con x_0 il valore iniziale, con x_1 il nuovo valore obiettivo e con u la variabile normalizzata nell'intervallo $[0, 1]$, il valore interpolato viene calcolato mediante la funzione *smoothstep*:

$$s(u) = u^2(3 - 2u), \tag{4.15}$$

da cui:

$$x(u) = x_0 + s(u)(x_1 - x_0). \tag{4.16}$$

L'utilizzo della funzione *smoothstep* permette di ottenere una variazione graduale tra due valori successivi, evitando discontinuità improvvise nella modulazione del ritardo. Nel modello implementato, l'inviluppo associato al wow viene aggiornato

con un intervallo di 0.65 s, mentre quello associato al Flutter viene aggiornato con un intervallo di 0.10 s. La diversa velocità di aggiornamento permette di ottenere due componenti caratterizzate da variazioni temporali differenti.
La componente di flutter viene inoltre attenuata mediante un fattore pari a 0.30, così da limitarne il contributo rispetto alla componente più lenta del wow. Le due componenti vengono quindi combinate per ottenere una modulazione complessiva del tempo di ritardo. Indicando con $w[n]$ e $f[n]$ i valori degli inviluppi di wow e flutter e con A_{WF} il parametro di intensità controllato dall'utente, la modulazione viene espressa come:

$$\Delta T[n] = (w[n] + f[n])\ 0.010\, A_{\mathrm{WF}}. \tag{4.17}$$

Il tempo di ritardo utilizzato dalla linea di ritardo risulta quindi:

$$T_d[n] = T_{d,0} + \Delta T[n], \tag{4.18}$$

con un limite inferiore imposto al valore risultante per evitare tempi di ritardo non validi.
La posizione di lettura del buffer viene aggiornata a ogni campione sulla base del nuovo valore di $T_d[n]$. In questo modo le variazioni di velocità del nastro vengono tradotte direttamente in una modulazione temporale del segnale riprodotto. L'approccio adottato privilegia quindi una generazione compatibile con l'elaborazione real-time, evitando la necessità di precalcolare l'intera traiettoria temporale della modulazione. Il carattere stocastico e la variazione graduale degli inviluppi permettono di introdurre una perturbazione continua della linea di ritardo, integrandola direttamente nel ciclo di elaborazione del modulo.

4.5 Taratura dei parametri e adattamento del modello

Il passaggio dal prototipo MATLAB al modulo real-time richiede non soltanto la trasposizione delle equazioni, ma anche l'adattamento dei parametri alle esigenze di un sistema interattivo. Nella versione C++, alcuni parametri che nella fase MATLAB erano fissati durante la simulazione sono stati resi controllabili dall'utente mediante l'interfaccia del modulo.
I parametri messi a disposizione sono il tempo di ritardo (*Delay Time*), il guadagno di retroazione (*Feedback*), il livello di eccitazione del modello (*Drive*), il rapporto

Dry/Wet, l'intensità di *Wow & Flutter* e il parametro *Tape Loss*. Gli intervalli di variazione e i valori iniziali sono definiti nella configurazione del modulo. Il parametro *Drive* modifica l'ampiezza del campo magnetico associato al segnale audio prima dell'ingresso nel modello di isteresi. Indicando con D il valore normalizzato del controllo, il fattore utilizzato per il campo audio è:

$$S_H = (0.2 + 1.8D)a, \tag{4.19}$$

dove a è il parametro del modello di Jiles–Atherton associato alla magnetizzazione anisteretica.
Il campo magnetico dovuto al segnale audio risulta quindi:

$$H_{\text{audio}}[n] = x_{\text{eq}}[n]S_H, \tag{4.20}$$

dove $x_{\text{eq}}[n]$ rappresenta il segnale dopo il blocco di pre-enfasi. L'incremento del parametro *Drive* porta il modello a operare su una regione più ampia della caratteristica di magnetizzazione, modificando di conseguenza la quantità di non linearità e saturazione introdotta. Un secondo adattamento riguarda la modellazione delle perdite della testina di riproduzione. Nella versione C++ il comportamento della testina viene rappresentato mediante un filtro FIR di lunghezza finita, i cui coefficienti sono ottenuti a partire dalla risposta teorica delle perdite per spaziatura, spessore e traferro.
Per rendere il comportamento regolabile durante l'utilizzo del modulo, il parametro *Tape Loss* modifica i valori utilizzati per il calcolo della risposta:

$$d = 0.1 \cdot 10^{-6} + 14.9 \cdot 10^{-6}L, \tag{4.21}$$

$$g = 1.5 \cdot 10^{-6} + 3.5 \cdot 10^{-6}L, \tag{4.22}$$

$$v = 0.1905 - 0.09525L, \tag{4.23}$$

dove L rappresenta il valore normalizzato del controllo, d la distanza tra nastro e testina, g la larghezza del traferro e v la velocità di scorrimento del nastro. La risposta della testina non viene tuttavia ricalcolata a ogni campione. Durante l'elaborazione il parametro *Tape Loss* viene controllato periodicamente e, quando necessario, vengono aggiornati i coefficienti del filtro FIR. In questo modo è possibile mantenere il controllo interattivo del parametro senza introdurre il costo computazionale della ricostruzione della risposta a ogni campione.

4.6 Ottimizzazione del carico computazionale

L'esecuzione del modello in tempo reale introduce vincoli più stringenti rispetto alla simulazione offline. Tutte le operazioni necessarie all'elaborazione devono infatti essere completate entro il tempo disponibile per il campione corrente. Il costo computazionale risulta particolarmente rilevante nel modello sviluppato, poiché l'isteresi viene calcolata con un fattore di sovracampionamento pari a 16 e mediante quattro sotto-passi RK4 per ciascun campione sovracampionato.
La versione C++ è stata quindi organizzata in modo da ridurre le operazioni eseguite nel percorso audio e da spostare, quando possibile, le elaborazioni più costose in fasi di inizializzazione o di aggiornamento periodico.

4.6.1 Valutazione di un fattore di sovracampionamento ridotto

Tra le possibili strategie di riduzione del carico computazionale, è stata considerata anche la variazione del fattore di sovracampionamento. Durante lo sviluppo è stata valutata, in alternativa alla configurazione utilizzata nel modello MATLAB, una configurazione con $OS = 4$.
La riduzione del fattore di sovracampionamento comporta una diminuzione diretta del numero di campioni elaborati internamente dal modello e, conseguentemente, del numero di valutazioni richieste dal solutore numerico. Poiché il segnale di bias viene generato alla frequenza di elaborazione interna, nella configurazione $OS = 4$ è stata inoltre valutata una riduzione della sua frequenza da 55 kHz a 40 kHz.
La configurazione $4 \times /40\,\text{kHz}$ consente quindi di ridurre il carico computazionale rispetto alla configurazione $16 \times /55\,\text{kHz}$. Tuttavia, la versione definitiva del modulo mantiene il fattore di sovracampionamento pari a $OS = 16$ e il bias a 55 kHz, in continuità con la configurazione utilizzata durante la fase di prototipazione e validazione in ambiente MATLAB.
La valutazione della configurazione a fattore ridotto ha permesso di analizzare il compromesso tra costo computazionale e risoluzione temporale del modello, evidenziando una possibile direzione di ottimizzazione per eventuali sviluppi futuri senza modificare la configurazione adottata nella versione finale.

4.6.2 Precalcolo e aggiornamento periodico dei coefficienti

Per contenere il costo computazionale è stato inoltre adottato il precalcolo dei coefficienti dei filtri. I coefficienti del blocco di sovracampionamento e dei filtri di debias vengono inizializzati durante la configurazione del modulo, mentre la risposta della testina di riproduzione viene aggiornata periodicamente durante l'esecuzione.
In particolare, la risposta della testina viene campionata su un numero limitato di punti spettrali e successivamente utilizzata per ricostruire un filtro FIR di 31 coefficienti. Il calcolo completo della risposta non viene quindi effettuato a ogni campione, ma solo negli aggiornamenti periodici previsti dall'algoritmo.
Un'ulteriore ottimizzazione riguarda la gestione della linea di ritardo. Il buffer circolare utilizzato per memorizzare i campioni ha una dimensione fissa pari a

$$262144 = 2^{18}, \tag{4.24}$$

scelta in modo da poter gestire il riavvolgimento degli indici mediante una maschera binaria definita come

$$\text{mask} = 2^{18} - 1. \tag{4.25}$$

Gli indici del buffer possono quindi essere ricondotti all'intervallo valido mediante un'operazione bit a bit, evitando il ricorso all'operazione di modulo nei principali accessi circolari. Questa soluzione semplifica la gestione del buffer e contribuisce a ridurre il costo delle operazioni ripetute all'interno del ciclo audio.
Anche i filtri impiegati nel percorso audio utilizzano strutture a dimensione fissa per memorizzare coefficienti e stato. I filtri FIR mantengono in memoria gli ultimi campioni elaborati attraverso array preallocati, mentre i filtri IIR memorizzano un numero limitato di variabili di stato.
Questa organizzazione evita allocazioni dinamiche durante il processamento del segnale e mantiene prevedibile il comportamento del percorso audio. Inoltre, le funzioni richiamate con elevata frequenza, come le operazioni di filtraggio e la valutazione del modello di isteresi, sono definite come `inline`, riducendo l'overhead associato alle chiamate di funzione.
Le ottimizzazioni introdotte non modificano quindi la struttura fondamentale del modello, ma intervengono principalmente sulla modalità con cui le operazioni vengono organizzate nel tempo. Il sovracampionamento e il sub-stepping mantengono elevata la risoluzione numerica del modello magnetico, mentre il precalcolo dei coef-

ficienti, l'aggiornamento periodico dei filtri e l'utilizzo di strutture a memoria fissa contribuiscono a contenere il costo computazionale.
Il risultato è un'implementazione nella quale le operazioni strettamente dipendenti dal campione corrente rimangono all'interno del ciclo audio, mentre quelle che possono essere calcolate preventivamente o aggiornate con una frequenza inferiore vengono eseguite al di fuori del percorso critico.

4.7 Interfaccia del modulo VCV Rack

Il modello sviluppato è stato integrato all'interno di un modulo per VCV Rack, con l'obiettivo di rendere direttamente accessibili all'utente i principali parametri dell'effetto e di consentirne la regolazione durante l'elaborazione del segnale.
Il modulo implementato, denominato `MioTapeDelay`, dispone di un ingresso audio e di un'uscita audio e presenta sei controlli dedicati alla configurazione del comportamento del sistema. I parametri disponibili sono: *Delay Time*, *Feedback*, *Drive*, *Dry/Wet*, *Wow & Flutter* e *Tape Loss*.
Il controllo *Delay Time* determina il tempo di ritardo nominale della linea di memoria. Il parametro *Feedback* controlla invece la quantità di segnale riprodotto che viene reinserita nel percorso di registrazione, determinando il numero e il decadimento delle ripetizioni.
Il controllo *Drive* regola il livello del campo magnetico associato al segnale audio prima dell'ingresso nel modello di isteresi e consente pertanto di modificare il grado di non linearità e saturazione introdotto dal modello.
Il parametro *Dry/Wet* determina il bilanciamento tra il segnale diretto e quello elaborato. Indicando con $x[n]$ il segnale in ingresso, con $y_{\text{wet}}[n]$ il segnale elaborato e con m il valore del controllo, l'uscita del modulo è calcolata come:

$$y[n] = (1 - m)x[n] + m\, y_{\text{wet}}[n]. \tag{4.26}$$

Il controllo *Wow & Flutter* determina l'ampiezza della modulazione temporale applicata alla linea di ritardo, mentre *Tape Loss* controlla i parametri utilizzati per la modellazione delle perdite della testina di riproduzione.
L'interfaccia grafica del modulo riprende la struttura tipica dei moduli di VCV Rack e dispone i sei controlli in due colonne, con ingresso e uscita audio collocati nella parte inferiore del pannello. La separazione tra interfaccia e logica di elaborazione consente di mantenere indipendente la struttura grafica dal percorso DSP vero e proprio.

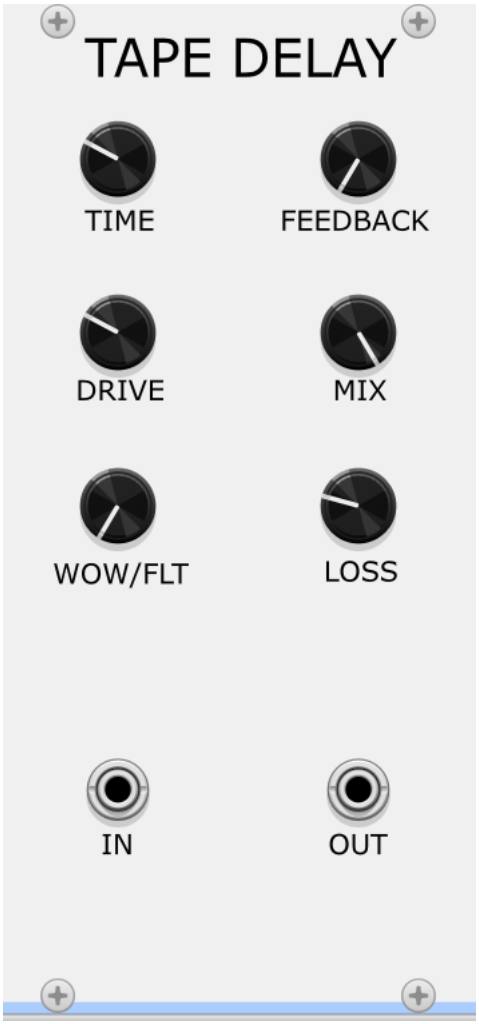


Figura 4.1: Interfaccia grafica del modulo `MioTapeDelay` implementato per VCV Rack.

4.8 Verifica dell'implementazione real-time

La fase conclusiva dello sviluppo è stata dedicata alla verifica del comportamento del modello all'interno dell'ambiente VCV Rack. Dopo aver appurato qualitativamente la corretta risposta del sistema alle variazioni interattive dei parametri esposti sull'interfaccia (descritti nella Sezione 4.7), l'analisi si è concentrata sulla validazione numerica dell'algoritmo DSP in tempo reale, confrontandolo direttamente con il prototipo offline.

Per verificare la coerenza del porting dall'ambiente MATLAB alla successiva implementazione real-time, è stato effettuato un confronto diretto tra le due versioni del modello. A tale scopo è stato utilizzato un segnale di test costituito da un burst sinusoidale a 1 kHz. Il confronto è stato effettuato considerando la prima ripetizione

del burst, corrispondente al primo eco prodotto dalla linea di ritardo.
Per rendere il confronto maggiormente significativo, nelle due implementazioni sono state utilizzate condizioni equivalenti, con *Feedback* e *Wow & Flutter* disattivati. Il confronto nel dominio del tempo è riportato in Figura 4.2. Per facilitare il confronto della forma d'onda, i due segnali sono stati normalizzati separatamente rispetto al proprio valore massimo esclusivamente ai fini della rappresentazione grafica.

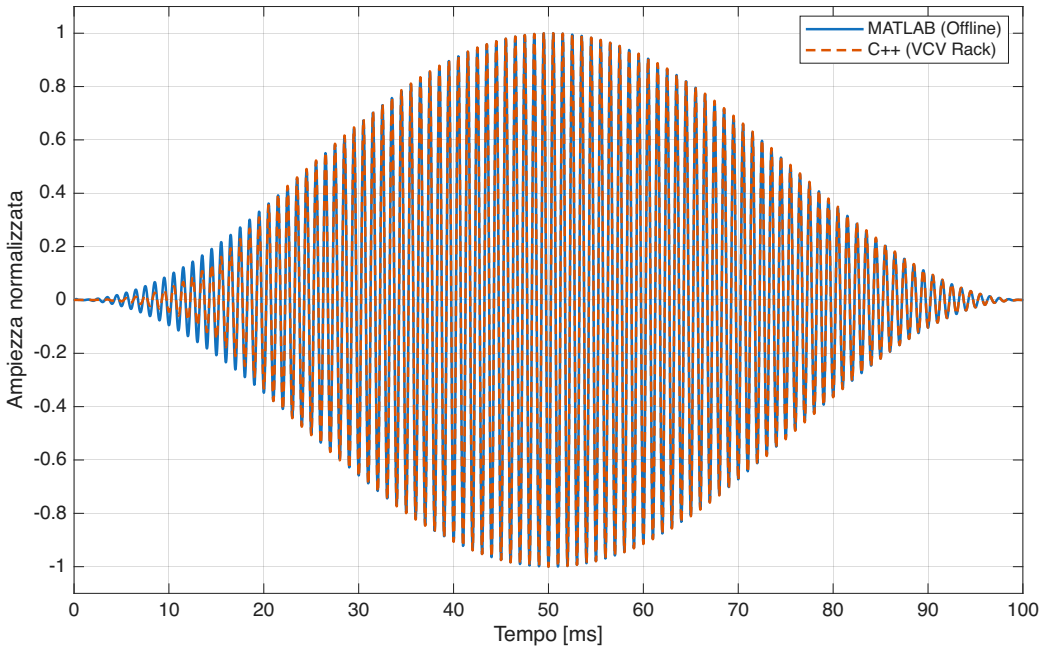


Figura 4.2: Confronto nel dominio del tempo tra la prima ripetizione del segnale di test ottenuta dal modello MATLAB e dall'implementazione C++ in VCV Rack. I segnali sono normalizzati rispetto al proprio massimo esclusivamente ai fini del confronto visivo.

Come si osserva dalla figura, le due implementazioni presentano un andamento temporale complessivamente simile, con una corrispondenza della frequenza del segnale e dell'inviluppo della prima ripetizione. La diversa ampiezza assoluta non costituisce in questo caso l'oggetto principale del confronto, poiché le due implementazioni adottano differenti modalità di scalatura e alcuni blocchi di adattamento specifici della versione real-time. Per completare la verifica è stato effettuato anche un confronto nel dominio della frequenza.

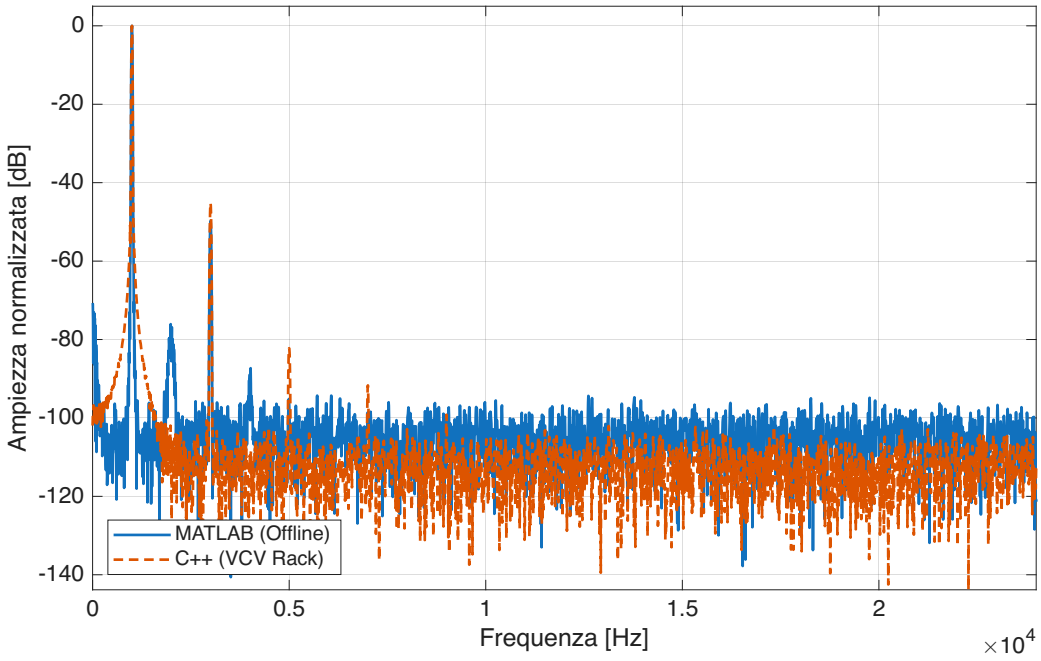


Figura 4.3: Confronto spettrale della prima ripetizione del segnale di test tra il modello MATLAB e l'implementazione C++. Lo spettro è stato calcolato mediante DFT con finestra di Hann e normalizzato rispetto al proprio valore massimo.

La prima ripetizione del burst è stata sottoposta a una finestra di Hann e successivamente alla Trasformata Discreta di Fourier (DFT). Gli spettri sono stati infine espressi in decibel e normalizzati rispetto al rispettivo valore massimo, così da confrontarne la forma indipendentemente dalla differenza globale di guadagno.
Il confronto evidenzia una buona corrispondenza della componente fondamentale, prossima a 1 kHz, presente in entrambe le implementazioni. Sono inoltre osservabili componenti armoniche associate alla non linearità del modello. Alle frequenze superiori si riscontrano alcune differenze nell'ampiezza delle componenti spettrali, riconducibili agli adattamenti introdotti nella realizzazione real-time dei blocchi di elaborazione. Particolare attenzione è stata inoltre riservata alla gestione dello stato interno del modello. I valori della magnetizzazione, del campo magnetico e della relativa derivata vengono conservati tra campioni consecutivi, così come gli stati dei filtri e gli indici della linea di ritardo. In occasione del reset del modulo, tali variabili vengono inizializzate nuovamente e il contenuto del buffer della linea di ritardo viene azzerato.
Questa gestione esplicita dello stato consente di preservare la continuità della simulazione durante l'esecuzione e di riportare il sistema a condizioni iniziali definite in seguito a un reset. Nel complesso, l'implementazione real-time ha consentito di trasferire il modello sviluppato e validato in ambiente MATLAB all'interno di un modulo

interattivo per VCV Rack. Il porting ha richiesto l'adattamento del solutore numerico, la gestione degli stati interni, la realizzazione della linea di ritardo in tempo reale e la sostituzione di alcune procedure offline con algoritmi aggiornabili durante l'esecuzione.

Il confronto tra le due implementazioni ha evidenziato una coerenza complessiva sia nel dominio temporale sia in quello spettrale, pur in presenza delle differenze introdotte dalle scelte implementative adottate per il funzionamento real-time.

Capitolo 5

CONCLUSIONI

Il presente lavoro ha portato allo sviluppo di un modello digitale di tape delay basato sulla modellazione fisica dell'isteresi magnetica e alla sua successiva implementazione in tempo reale in linguaggio C++ per la piattaforma VCV Rack.

La fase di prototipazione e validazione in ambiente MATLAB ha permesso di verificare il comportamento dei principali fenomeni considerati nel modello, tra cui la non linearità magnetica, le perdite della testina di riproduzione e le modulazioni temporali del ritardo. La successiva implementazione in C++ ha richiesto l'adattamento dell'algoritmo alle condizioni di elaborazione real-time, con particolare attenzione alla stabilità del solutore numerico, alla gestione dello stato e al contenimento del carico computazionale.

La versione finale del modulo mantiene un fattore di sovracampionamento pari a $OS = 16$ e utilizza il metodo RK4 con quattro sotto-passi per campione sovracampionato. Parallelamente, sono state introdotte strategie di ottimizzazione basate sul precalcolo dei coefficienti, sull'aggiornamento periodico dei filtri e sulla gestione efficiente della memoria.

Il confronto tra il modello MATLAB e l'implementazione C++ ha mostrato una coerenza complessiva sia nel dominio temporale sia nel dominio spettrale. In particolare, la componente fondamentale del segnale di test e la struttura generale delle componenti armoniche risultano presenti in entrambe le implementazioni. Le differenze osservate alle frequenze superiori sono invece riconducibili agli adattamenti introdotti nella realizzazione real-time.

Il risultato finale è un modulo interattivo per VCV Rack nel quale il modello di isteresi, la linea di ritardo, il feedback, gli effetti di perdita e le modulazioni temporali sono integrati all'interno di un unico percorso di elaborazione. I principali parametri dell'effetto possono inoltre essere modificati durante l'esecuzione, rendendo il modello

utilizzabile in un contesto audio interattivo.

5.1 Sviluppi futuri

Il lavoro svolto rappresenta una base per ulteriori sviluppi sia dal punto di vista dell'efficienza computazionale sia dal punto di vista della fidelizzazione dei fenomeni fisici modellati.

Un primo possibile sviluppo riguarda la riduzione del costo computazionale dell'algoritmo. La valutazione della configurazione con $OS = 4$ ha mostrato la possibilità di ridurre significativamente il numero di campioni elaborati dal modello magnetico. Un'analisi più approfondita potrebbe consentire di individuare il miglior compromesso tra fattore di sovracampionamento, frequenza del bias, accuratezza dell'integrazione e utilizzo della CPU. In questa direzione potrebbero essere valutate anche ulteriori strategie di ottimizzazione del solutore e dell'organizzazione del percorso DSP.

Un secondo ambito di sviluppo riguarda la riduzione delle differenze tra il prototipo MATLAB e l'implementazione real-time. Alcuni blocchi della versione C++ sono stati infatti adattati alle esigenze computazionali del funzionamento in tempo reale. Un'ulteriore evoluzione potrebbe quindi prevedere una maggiore corrispondenza tra le due implementazioni, ad esempio attraverso una rappresentazione più accurata della risposta della testina di riproduzione e mediante una maggiore uniformità nelle procedure di filtraggio e di conversione della frequenza di campionamento.

Un ulteriore sviluppo può riguardare la modellazione delle imperfezioni meccaniche ed elettriche del registratore. Il modello attuale introduce Wow e Flutter mediante una modulazione stocastica del tempo di ritardo; questa rappresentazione potrebbe essere ulteriormente raffinata attraverso modelli più dettagliati delle variazioni di velocità del nastro e delle loro caratteristiche temporali. Analogamente, potrebbe essere approfondita la modellazione del rumore e di altre imperfezioni associate al supporto magnetico e alla macchina reale.

Infine, potrebbe essere sviluppata una fase di validazione più strettamente legata all'hardware di riferimento, attraverso un confronto sistematico tra le caratteristiche del modello e misure effettuate direttamente su un registratore reale. Ciò permetterebbe di migliorare la taratura dei parametri fisici e di valutare quantitativamente la corrispondenza tra il comportamento del modello e quello della macchina analogica.

Nel complesso, questi sviluppi potrebbero consentire di mantenere la flessibilità e l'efficienza richieste da un'applicazione real-time, incrementando al tempo stesso il

livello di dettaglio della modellazione fisica e la corrispondenza con il comportamento del sistema di riferimento.

Bibliografia

[1] J. Pakarinen, D. T. Yeh, D. Berners, and J. O. Smith, "Review of virtual analog audio effects," *Journal of the Audio Engineering Society*, vol. 59, no. 1/2, pp. 63–76, 2011.

[2] V. Välimäki, S. Bilbao, J. O. Smith, J. S. Abel, J. Pakarinen, and D. Berners, "Virtual analog effects," in *DAFX: Digital Audio Effects*, 2nd ed. John Wiley & Sons, 2011.

[3] S. Arnardottir, J. S. Abel, and J. O. Smith III, "A digital model of the echoplex tape delay," in *Proceedings of the 125th AES Convention*, San Francisco, CA, USA, 2008.

[4] O. Mikkonen, A. Wright, E. Moliner, and V. Välimäki, "Neural modeling of magnetic tape recorders," in *Proceedings of the 26th International Conference on Digital Audio Effects (DAFx-23)*, Copenhagen, Denmark, 2023.

[5] D. C. Jiles and D. L. Atherton, "Theory of ferromagnetic hysteresis," *Journal of Magnetism and Magnetic Materials*, vol. 61, no. 1-2, pp. 48–60, 1986.

[6] J. Chowdhury, "Real-time physical modelling for analog tape machines," in *Proceedings of the 22nd International Conference on Digital Audio Effects (DAFx-19)*, Birmingham, UK, 2019.

[7] L. Gabrielli, *Developing Virtual Synthesizers with VCV Rack*. Focal Press, 2020.

Ringraziamenti

Il mio primo ringraziamento va ai professori Leonardo Gabrielli e Stefano Squartini, per la fiducia e la supervisione che hanno reso possibile questo lavoro. Inoltre, un grazie sincero va alla mia famiglia, agli amici, ai coinquilini e a tutti i compagni di viaggio che, in modi diversi, hanno contribuito a rendere indimenticabile questo traguardo.

Christian