

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

SCUOLA DI SCIENZE

Corso di Laurea Magistrale in Matematica

**Modellazione matematica
delle non linearità di sistemi acustici
mediante serie di Volterra modificate**

Tesi di Laurea in Acustica Musicale

Relatore:
Chiar.mo Prof.
Lamberto Tronchin

Correlatrice:
Chiar.ma Prof.ssa
Elena Loli Piccolomini

Correlatore:
Ing.
Andrea Venturi

Presentata da:
Vanna Lisa Coli

**III Sessione
Anno Accademico 2012/2013**

*L'arte nel suo mistero
le diverse bellezze insiem confonde.*

dal libretto di *Tosca*,
opera in tre atti di Giacomo Puccini.

*ai miei
ventott'anni:*

Indice

Ringraziamenti	v
Introduzione	vii
1 Segnali, sistemi e Digital Signal Processing	1
1.1 Trasformazioni sui segnali e segnali notevoli	3
1.2 Sistemi per l'elaborazione dei segnali	7
1.2.1 Composizione di sistemi	9
1.2.2 Spazi vettoriali di segnali	11
1.3 Sistemi lineari	15
1.3.1 Caratterizzazione di un sistema lineare	15
1.3.2 Sistemi tempo-invarianti	16
1.3.3 Integrale di convoluzione	17
1.4 Analisi in frequenza di segnali analogici	18
1.4.1 Segnali periodici	18
1.4.2 Risposta in frequenza dei sistemi LTI	20
1.4.3 Serie di Fourier	20
1.4.4 Trasformata di Fourier	21
1.4.5 Banda ed ampiezza di banda di un segnale	24
1.4.6 Trasformata di Fourier di funzioni reali	25
1.4.7 Risposta in frequenza dei sistemi lineari tempo-invarianti	27
1.4.8 Segnale analitico	28
1.5 Filtri	29
1.5.1 Filtri di Butterworth	33

1.6	Conversione analogico-digitale	34
1.6.1	Campionamento	35
1.6.2	Aliasing	36
1.6.3	Quantizzazione	37
1.7	Trasformata discreta di Fourier e FFT	38
1.7.1	Trasformata di Fourier a Tempo Discreto (DTFT)	38
1.7.2	Trasformata Discreta di Fourier (DFT)	41
1.7.3	Trasformata Discreta di Fourier e DSP	43
1.7.4	Fast Fourier Transform	45
2	Sistemi non lineari e serie di Volterra: il modello	47
2.1	Sistemi di Volterra lineari	48
2.1.1	Integrale di convoluzione	49
2.1.2	Causalità	51
2.1.3	Stabilità	52
2.2	Sistemi di Volterra non lineari	53
2.2.1	Causalità	56
2.2.2	Stabilità	56
2.3	Serie di Volterra	57
2.4	Il modello	59
2.4.1	Modello di Wiener-Hammerstein	59
2.4.2	Serie di Volterra diagonale	60
2.5	Modello numerico	62
3	Emulazione di un sistema audio	65
3.1	Synchronized Sine Sweep	67
3.1.1	Proprietà nel dominio delle frequenze	71
3.1.2	Filtro inverso	71
3.2	Nonlinear Convolution	74
3.2.1	Misura e deconvoluzione	75
3.2.2	Separazione delle risposte all'impulso	76
3.2.3	Calcolo dei kernel di Volterra	81

3.2.4	Emulazione del dispositivo non lineare	91
4	Sperimentazione effettuata	97
4.1	Risultati ottenuti sugli sweep	103
4.2	Risultati ottenuti sui segnali reali	108
4.2.1	Accordo di chitarra	109
4.2.2	File audio di chitarra solo	111
4.2.3	Prova d'ascolto	113
5	Conclusioni	115
	Bibliografia	117

Ringraziamenti

Ringrazio il Prof. Lamberto Tronchin per avermi proposto l'argomento di questa tesi, che mi ha permesso di ascoltare e di scoprire con "orecchio scientifico" il meraviglioso suono della musica.

Ringrazio l'Ing. Andrea Venturi per avermi guidata ed assistita nei miei primi e timorosi passi nel Digital Signal Processing, e per aver supportato e sopportato con pazienza le mie continue richieste di chiarimenti.

Ringrazio la Prof. ssa Elena Loli Piccolomini per l'interesse e la grande disponibilità dimostrate da subito nei confronti della mia tesi.

Ringrazio la mia Puffa, il mio Fratellone, la mia Sma' e il mio Babuuuu perché sì (che non è una risposta).

Introduzione

*La musica non esiste nel vuoto*¹.

Il suono (e quello che potrebbe essere considerato un suo caso particolare, cioè la musica) necessita di un mezzo elastico attraverso il quale propagarsi: l'aria è il primo degli strumenti. L'aria è anche lo strumento meno “invasivo”, quello che meno modifica il suono nel suo percorso, quello più *lineare*.

Il “mezzo” che mi ha fornito l'opportunità di sviluppare questo lavoro di tesi è stato il corso di Acustica Musicale tenuto dal Prof. Lamberto Tronchin, che mi ha permesso di studiare ed applicare la Matematica in un ambito che amo molto.

In questa tesi si è studiato un metodo per modellare e virtualizzare tramite algoritmi in Matlab le distorsioni armoniche di un dispositivo audio *non lineare*, ovverosia uno “strumento” che, sollecitato da un segnale audio, lo modifichi (con modalità che gli sono proprie ed, il più delle volte, ignote), introducendovi delle componenti non presenti in precedenza. In particolare, il dispositivo che si è scelto per questo studio è il pedale “BOSS SD-1 Super OverDrive” per chitarra elettrica e lo “strumento matematico” che ne fornisce il modello è lo *sviluppo in serie di Volterra*, in una sua forma modificata.

I sistemi lineari tempo-invarianti sono completamente caratterizzati dalla risposta che essi forniscono ad un segnale di tipo impulsivo: una volta che tale risposta sia stata misurata è possibile ricostruire la risposta che essi

¹Benjamin Britten, *On receiving the First Aspen Award*, Londra, Faber & Faber, 1964.

fornirebbero ad un segnale qualsiasi semplicemente mediante il procedimento di *convoluzione* con la risposta all'impulso.

Lo sviluppo in serie di Volterra viene diffusamente usato nello studio di sistemi fisici non lineari, nel caso in cui si abbia interesse a modellare un sistema che si presenti come una “black box”, ovvero sia un sistema del quale non si posseggano informazioni. Il metodo della *Nonlinear Convolution* presentato dall'Ing. Angelo Farina nel 2002 ha applicato con successo tale sviluppo anche all'ambito dell'acustica musicale: servendosi di una tecnica di misurazione facilmente realizzabile e del modello fornito dalla serie di Volterra Diagonale, il metodo permette di caratterizzare un dispositivo audio non lineare mediante le risposte all'impulso non lineari che il dispositivo fornisce a fronte di un opportuno segnale di test (denominato “Exponential Sine Sweep”). Le risposte all'impulso del dispositivo vengono utilizzate per ricavare i *kernel di Volterra* della serie.

L'utilizzo di tale metodo ha permesso al Prof. Lamberto Tronchin ed all'Ing. Andrea Venturi di realizzare un software che virtualizzasse in post-processing le non linearità di un sistema audio, per il quale l'Università di Bologna ha ottenuto un brevetto nel 2012. In questa tesi si è ripreso il lavoro che ha portato al raggiungimento del brevetto, apportandovi due innovazioni: in primo luogo, si è modificata la scelta del segnale utilizzato per testare il dispositivo (si è fatto uso del *Synchronized Sine Sweep*, presentato dal Dott. Antonín Novák nel 2009, in luogo dell'Exponential Sine Sweep); in secondo luogo, si è messo in atto un primo tentativo di orientare la virtualizzazione verso l'elaborazione in *real time*, implementando un procedimento “veloce” (ma ancora in post-processing) di creazione dei kernel in dipendenza dal volume dato in input al dispositivo non lineare.

La tesi è organizzata come segue: nel primo capitolo vengono presentati gli elementi di teoria ed elaborazione dei segnali che forniscono gli strumenti agli algoritmi utilizzati; nel secondo capitolo viene riportata la teoria di

Volterra per i sistemi non lineari, che fornisce il supporto matematico del modello; nel terzo capitolo si descrive il procedimento di misura e virtualizzazione delle non linearità del dispositivo audio in esame; nel capitolo finale viene esposta la sperimentazione messa in atto, con i risultati ottenuti.

Capitolo 1

Segnali, sistemi e Digital Signal Processing

Il flusso continuo di *segnali* è diventato uno degli aspetti caratterizzanti la quotidianità odierna. In qualsiasi ambito scientifico, l'analisi e l'elaborazione delle *informazioni* contenuti nei segnali e veicolate attraverso *sistemi* hanno assunto un'importanza via via crescente con il progredire della tecnica, a partire dall'avvento dei calcolatori. Il Digital Signal Processing (DSP) fornisce gli algoritmi e le tecniche per studiare e manipolare i segnali del mondo esterno, una volta trovata una loro espressione *digitale*. Di particolare interesse in questo lavoro, saranno i segnali di tipo *audio*.

Le nozioni e le figure presenti in questo capitolo sono riportate dai trattati [1] e [2], ulteriori approfondimenti si possono trovare nei trattati [3], [4].

Un *segnale* può essere considerato come la descrizione della variazione di un parametro in funzione di un altro ed è in generale un mezzo per veicolare informazioni che possano essere estratte, trasmesse ed elaborate.

Nella Figura 1.1 si ha un esempio di segnale acustico, il quale descrive la variazione della pressione acustica $p(t)$ in funzione del tempo t : è dunque un segnale *temporale*.

Dal punto di vista prettamente matematico un segnale può dunque esse-

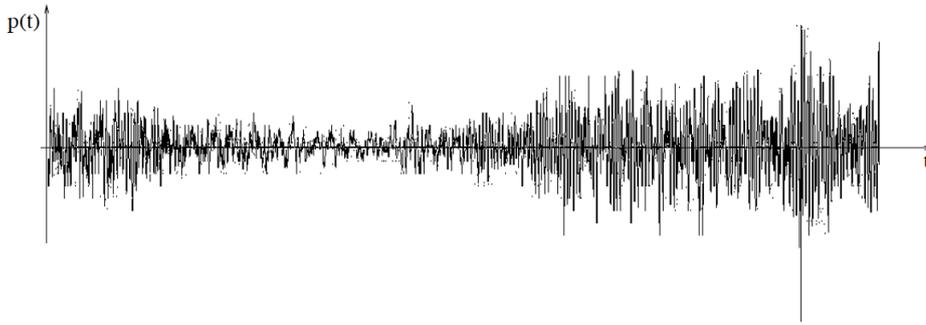


Figura 1.1: Segnale acustico

re rappresentato da una funzione $f : A \rightarrow B$ con A e B sottoinsiemi di \mathbb{R}^n : i valori di A si riferiscono in generale a variabili spaziali e/o temporali, mentre quelli di B a quelli assunti da grandezze fisiche quali pressione, tensione, correnti. I segnali che tratteremo saranno tutti segnali temporali ed, in particolare, dei due tipi seguenti:

- *segnali analogici*: a tempo continuo ($t \in A \subseteq \mathbb{R}$) e valori continui ($f(t) \in B \subseteq \mathbb{R}$)
- *segnali digitali*: a tempo discreto ($n \in A \subseteq \mathbb{Z}$) e valori discreti ($f[n] \in B \subseteq \mathbb{Z}$).

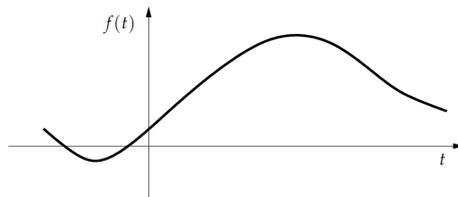


Figura 1.2: Segnale analogico.

Osservazione 1.1. In tutto il testo verranno utilizzate le notazioni di uso comune nell'ambito del Digital Signal Processing: un segnale analogico verrà

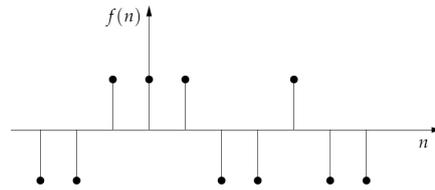


Figura 1.3: Segnale digitale.

dunque caratterizzato dall'utilizzo di parentesi tonde (ad esempio: “ $f(t)$ ”), mentre un segnale digitale dall'utilizzo di parentesi quadre (ad esempio: “ $f[n]$ ”).

1.1 Trasformazioni sui segnali e segnali notevoli

Sui segnali si possono operare le seguenti trasformazioni:

- *traslazione*: fissato un tempo t , la traslazione trasforma il segnale $f(t)$ nel segnale $f(t - t_0)$, ovvero si ritarda il segnale $f(t)$ di un tempo t_0 ;
- *scalatura*: fissato un numero reale $a > 0$, la scalatura trasforma il segnale $f(t)$ nel segnale $f(at)$; l'effetto ottenuto è quello di una compressione lineare nel caso in cui $a > 1$ oppure di un allungamento lineare nel caso in cui $a < 1$;
- *inversione*: il segnale $f(t)$ viene trasformato nel segnale $f(-t)$, che rappresenta il segnale letto al contrario.

Le trasformazioni appena definite possono essere estese naturalmente al caso dei segnali discreti.

Impulso rettangolare L'impulso rettangolare unitario $\text{rect}(t)$ è definito come:

$$\text{rect}(t) = \begin{cases} 1 & , \text{ se } |t| < \frac{1}{2} \\ 0 & , \text{ se } |t| > \frac{1}{2} \end{cases}$$

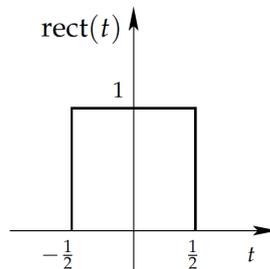


Figura 1.4: Impulso rettangolare unitario.

Gradino unitario Il gradino unitario $u(t)$ è definito come:

$$u(t) = \begin{cases} 1 & , \text{ se } t > 0 \\ 0 & , \text{ se } t < 0 \end{cases}$$

Il gradino unitario traslato in t_0 è invece definito da:

$$u(t - t_0) = \begin{cases} 1 & , \text{ se } t > t_0 \\ 0 & , \text{ se } t < t_0 \end{cases}$$

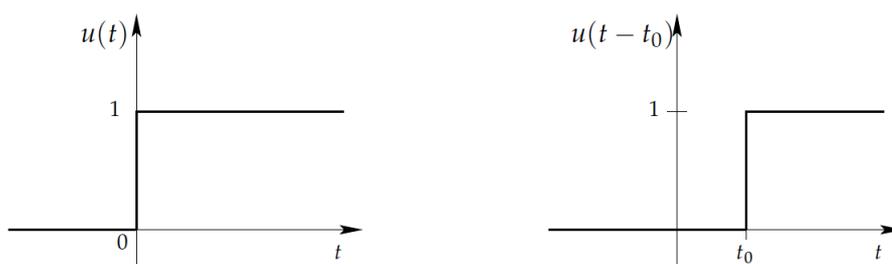


Figura 1.5: Gradino unitario e gradino unitario traslato in t_0 .

Impulso unitario La funzione impulso unitario $\delta(t)$ ha grande importanza nella teoria dei segnali e si è soliti riferirsi ad essa con il termine *delta*

di Dirac. Intuitivamente, si può pensare alla delta di Dirac come ad un rettangolo di base infinitesima Δ e di altezza infinita $\frac{1}{\Delta}$ e tale che $\int_{\mathbb{R}} \delta(t) dt = 1$. Dal punto di vista matematico, dunque, non è una funzione, bensì una funzione generalizzata, dal momento che il valore di $\delta(0)$ è $+\infty$. L'impulso $\delta(t)$ può essere visto come limite per $\Delta \rightarrow 0$ di una sequenza di funzioni $p_{\Delta}(t)$ non negative, con $p_{\Delta}(t)$ nulla al di fuori dell'intervallo $[-\Delta/2, \Delta/2]$ e tale che $\int_{\mathbb{R}} p_{\Delta}(x) dx = 1$.

La delta di Dirac $\delta(t)$ viene rappresentata con una freccia verticale di altezza 1 ed applicata in 0.

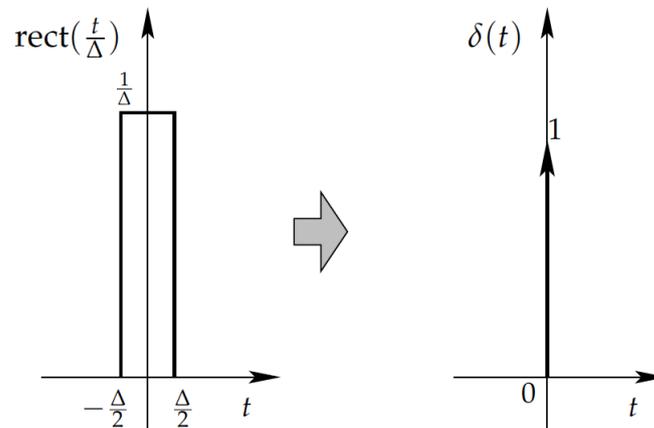


Figura 1.6: Gradino unitario e gradino traslato in t_0 .

Nella Figura 1.6 l'impulso è visto come il limite di un impulso rettangolare:

$$\delta(t) = \lim_{\Delta \rightarrow 0} \frac{1}{\Delta} \text{rect} \left(\frac{t}{\Delta} \right)$$

Nella Figura 1.7 viene mostrato l'impulso $A\delta(t-t_0)$ di ampiezza A che occorre al tempo $t = t_0$.

La delta di Dirac assume un senso più preciso solo all'interno di un integrale, da cui si ha la seguente definizione:

$$\langle f, \delta \rangle = \int_{\mathbb{R}} f(t) \delta(t) dt = f(0)$$

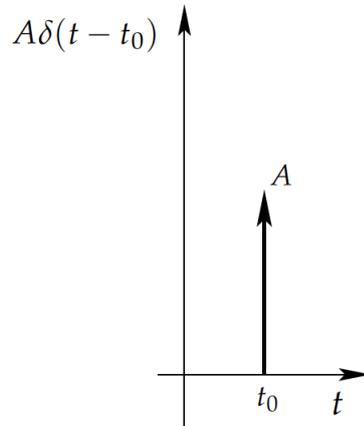


Figura 1.7: Impulso unitario scalato e traslato in t_0 .

dove $\langle f, \delta \rangle$ denota il prodotto scalare tra $f(\cdot)$ e $\delta(\cdot)$, sotto l'ipotesi che $f(\cdot)$ sia continua intorno a 0. Se (\cdot) è continua in $t = \tau$ allora

$$\int_{\mathbb{R}} f(t)\delta(t - \tau)dt = f(\tau)$$

ed anche

$$\int_{\mathbb{R}} f(\tau)\delta(t - \tau)d\tau = f(t)$$

Si riportano di seguito alcune proprietà della delta di Dirac:

- $\delta(t - \tau) = 0$ se $t \neq \tau$;
- $f(t)\delta(t - \tau) = f(\tau)\delta(t - \tau)$;
- $\delta(at) = \frac{1}{|a|}\delta(t)$ per $a \in \mathbb{R}$;
- $\int_{\mathbb{R}} \delta(t - \tau)dt = 1$.

Nella teoria dei segnali l'impulso ricorre sovente; i suoi due usi principali sono lo studio dei sistemi lineari tempo-invarianti e il campionamento ideale, come si vedrà nel resto del capitolo.

Impulso e gradino discreto Due segnali discreti che ricoprono un ruolo fondamentale nello studio dei sistemi a tempo discreto sono gli analoghi discreti della delta di Dirac e del gradino unitario.

L'*impulso unitario discreto* $\delta[n]$ è definito da

$$\delta[n] = \begin{cases} 1 & , \text{ se } n = 0 \\ 0 & , \text{ se } n \neq 0 \end{cases} \quad (1.1)$$

Il *gradino unitario discreto* $u[n]$ è invece definito da:

$$u[n] = \begin{cases} 1 & , \text{ se } n \geq 0 \\ 0 & , \text{ se } n < 0 \end{cases}$$

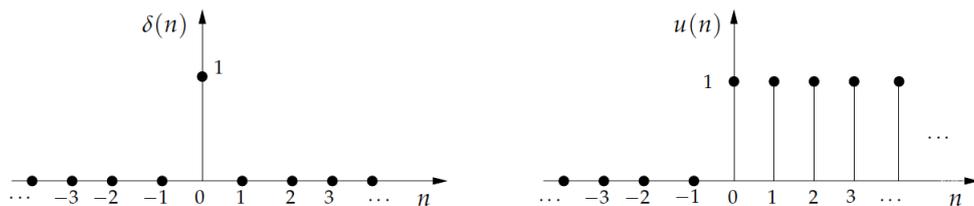


Figura 1.8: Impulso unitario discreto e gradino unitario discreto.

1.2 Sistemi per l'elaborazione dei segnali

Un sistema fisico è un apparato che, a fronte di un segnale in ingresso, produce in uscita un nuovo segnale. Un *sistema* è dunque un processo per il quale esiste una relazione di ingresso-uscita o anche di causa-effetto.

Nell'ambito di questa trattazione, la causa sarà rappresentata dal segnale in ingresso (*input*), l'effetto dal segnale di uscita (*output*) e la relazione dalle equazioni che determinano il *modello* del sistema.

In Figura 1.9 è data una rappresentazione grafica di un sistema S che, preso in input un segnale $f(t)$, produce in uscita il segnale $g(t)$; si è soliti

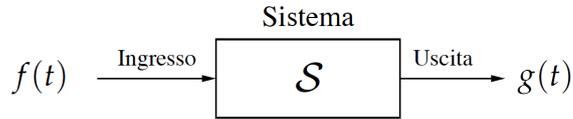


Figura 1.9: Sistema a tempo continuo.

riferirsi a $g(t)$ come alla *risposta del sistema* S all'ingresso $f(t)$:

$$g(t) = S[f(t)]$$

Osservazione 1.2. S rappresenta una *trasformazione*, ovvero sia tutte le operazioni che contribuiscono a modificare $f(t)$ per ottenere $g(t)$.

Analoga definizione viene estesa anche al caso dei sistemi a tempo discreto, Figura 1.10.

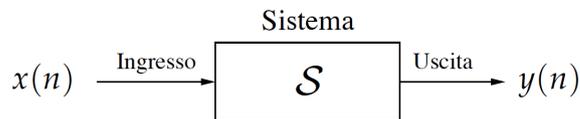


Figura 1.10: Sistema a tempo discreto.

Se indichiamo con \mathcal{F}_1 l'insieme dei possibili segnali di ingresso e con \mathcal{F}_2 l'insieme dei possibili segnali d'uscita, allora la seguente legge di trasformazione

$$S : \mathcal{F}_1 \rightarrow \mathcal{F}_2$$

fornisce una caratterizzazione completa del comportamento del sistema. Ovviamente, nella maggior parte dei casi di interesse, tale trasformazione è incognita.

1.2.1 Composizione di sistemi

Data la complessità della maggior parte dei sistemi, può essere utile suddividere questi ultimi in “sottosistemi” più semplici per agevolarne lo studio. Di particolare interesse in questa trattazione sono i seguenti tipi di composizione di sistemi:

- *composizione sequenziale* o *cascata*: dati due sistemi $S_1 : \mathcal{F}_1 \rightarrow \mathcal{F}_2$ ed $S_2 : \mathcal{F}_2 \rightarrow \mathcal{F}_3$, la loro composizione sequenziale è il sistema $S_3 : \mathcal{F}_1 \rightarrow \mathcal{F}_3$ ottenuto ponendo in ingresso al sistema S_2 l'uscita del sistema S_1 ;
- *composizione parallela*: dati due sistemi $S_1 : \mathcal{F}_1 \rightarrow \mathcal{F}_2$ ed $S_2 : \mathcal{F}_1 \rightarrow \mathcal{F}_2$, la loro composizione parallela è il sistema che ha come risposta la somma delle risposte di S_1 e di S_2 (nell'ipotesi che tale somma sia ancora un segnale in \mathcal{F}_2 , altrimenti la composizione non è definita).

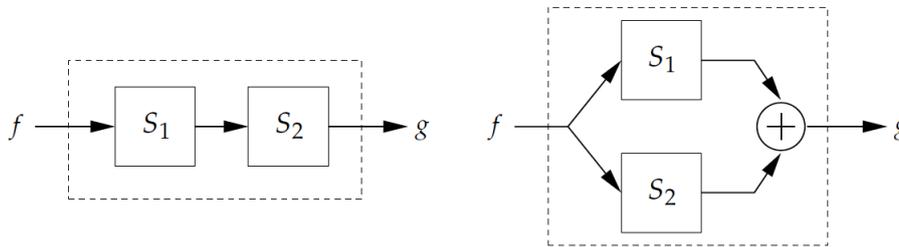


Figura 1.11: Cascata e composizione parallela.

Esempio 1.3. Conversione analogico-digitale. Dalla composizione sequenziale di un campionatore e di un quantizzatore si ottiene un convertitore analogico-digitale (ADC), che trasforma un segnale analogico in uno digitale.

Esempio 1.4. Conversione digitale-analogica (DAC). È la trasformazione inversa della precedente, in quanto trasforma un segnale digitale in uno analogico.

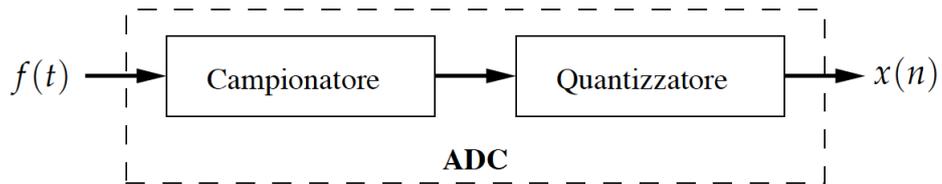


Figura 1.12: Elaborazione di segnali digitali.

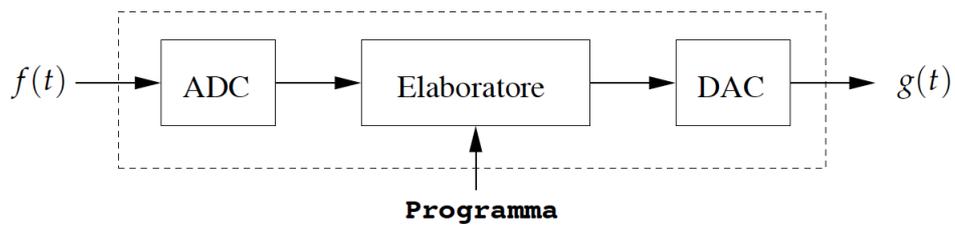


Figura 1.13: Elaborazione di segnali analogici.

Gli ADC ed i DAC realizzano dunque l'interfaccia tra il mondo "vero" analogico e quello digitale e verranno discussi approfonditamente nella Sezione 1.6.

1.2.2 Spazi vettoriali di segnali

Dati due segnali $f_1(t)$ ed $f_2(t) \in \mathcal{F}$, la loro *somma* è il segnale $f(t) \in \mathcal{F}$ ottenuto dalla sovrapposizione di $f_1(t)$ e di $f_2(t)$:

$$f(t) = f_1(t) + f_2(t)$$

L'operazione di somma è commutativa ed associativa.

Dato uno scalare $a \in \mathbb{R}$, il *prodotto* di $f_1(t)$ per tale scalare è il segnale $f(t)$ ottenuto amplificando $f_1(t)$ del fattore a , cioè

$$f(t) = af_1(t)$$

Il *segnale nullo* 0 è tale che

$$f(t) + 0 = f(t)$$

Il segnale $f(t)$ ammette l'*opposto* $-f(t)$ che soddisfa $f(t) - f(t) = 0$.

Dati due scalari $a, b \in \mathbb{R}$, vale inoltre che

- $a(bf(t)) = abf(t)$;
- $a(f_1(t) + f_2(t)) = af_1(t) + af_2(t)$.

Ne segue che l'insieme \mathcal{F} dei segnali, munito delle operazioni sopra definite, è uno spazio vettoriale.

Data una famiglia di segnali $\{f_1(t), \dots, f_r(t)\}$ possiamo dunque definirne la combinazione lineare:

$$f(t) = a_1f_1(t) + a_2f_2(t) + \dots + a_rf_r(t) = \sum_{i=1}^r a_if_i(t) \quad , \quad a_i \in \mathbb{R}$$

Introducendo opportune nozioni di limite, questa operazione può essere estesa ad infiniti segnali, dando senso ad espressioni del tipo:

$$\sum_{i=-\infty}^{+\infty} a_i f_i(t) \quad , \quad a_i \in \mathbb{R}$$

Data una famiglia di segnali $\{f_1(t), \dots, f_r(t)\}$, diremo che il segnale $f(t)$ è *linearmente dipendente dalla famiglia* $\{f_1(t), \dots, f_r(t)\}$ se esso può essere ottenuto come combinazione lineare degli elementi della famiglia; diversamente, viene detto *linearmente indipendente dalla famiglia* $\{f_1(t), \dots, f_r(t)\}$.

Sia \mathcal{A} un sottospazio vettoriale di \mathcal{F} contenente la famiglia $\{f_1(t), \dots, f_r(t)\}$; se ogni segnale $f(t) \in \mathcal{A}$ si può ottenere come combinazione lineare dei segnali della famiglia $\{f_1(t), \dots, f_r(t)\}$, ovverosia se vale che

$$\forall f(t) \in \mathcal{F} \quad f(t) = \sum_{i=-\infty}^{+\infty} a_i f_i(t) \quad , \quad a_i \in \mathbb{R}$$

allora la famiglia $\{f_1(t), \dots, f_r(t)\}$ è detta *insieme di generatori di* \mathcal{A} .

Se tutti gli elementi di $\{f_1(t), \dots, f_r(t)\}$ sono linearmente indipendenti tra loro, la famiglia $\{f_1(t), \dots, f_r(t)\}$ è una *base per* \mathcal{A} . Data una base per un sottospazio \mathcal{A} , si può dimostrare che per ogni $f(t) \in \mathcal{A}$ esiste una sola sequenza di coefficienti a_i tale che

$$f(t) = \sum_{i=1}^r a_i f_i(t)$$

In altre parole, ogni segnale $f(t) \in \mathcal{A}$ può essere determinato in maniera univoca da una opportuna combinazione lineare degli elementi di una base. Introducendo opportune nozioni di limite, la nozione di base può essere estesa a basi con infiniti elementi.

Per segnali continui definiti sull'intervallo $[a, b]$ possiamo definire la seguente operazione, detta *prodotto interno*:

$$(f(t), g(t)) := \int_a^b f(t)g(t)dt$$

Una base $\{f_1(t), \dots, f_r(t), \dots\}$ è detta *ortogonale* se risulta:

$$\forall i \neq j \quad (f_i(t), f_j(t)) = 0$$

Fatto 1.5. Se $\{f_1(t), \dots, f_r(t), \dots\}$ è una base ortogonale per uno spazio di segnali \mathcal{F} allora ogni segnale $f(t) \in \mathcal{F}$ è tale che

$$f(t) = \sum_i a_i f_i(t)$$

dove

$$a_i = \frac{(f(t), f_i(t))}{(f_i(t), f_i(t))}$$

Dimostrazione 1.6. Si ha che

$$(f(t), f_i(t)) = \left(\sum_k a_k f_k, f_i \right) = \sum_k a_k (f_i(t), f_i(t))$$

e ricavando il valore a_k si ottiene il risultato.

Se in una base ortogonale $\{f_1(t), \dots, f_r(t), \dots\}$ vale che $(f_i(t), f_i(t)) = 1$ per ogni i , tale base viene detta *ortonormale*. Per le basi ortonormali vale il seguente

Fatto 1.7. Se $\{f_1(t), \dots, f_r(t), \dots\}$ è una base ortonormale per uno spazio di segnali \mathcal{F} allora ogni segnale $f(t) \in \mathcal{F}$ è tale che

$$f(t) = \sum_i a_i f_i(t)$$

dove

$$a_i = (f(t), f_i(t))$$

Base per spazio di segnali a tempo continuo. Nello spazio dei segnali a tempo continuo, l'insieme delle delta di Dirac $\{\delta(t-x) : x \in \mathbb{R}\}$ rappresenta una base per i segnali a tempo continuo. Infatti ogni segnale $f(t)$ può essere espresso come combinazione lineare generalizzata di impulsi dove il coefficiente moltiplicativo di $\delta(t-x)$ è proprio $f(x)$, ovvero

$$f(t) = \int_{\mathbb{R}} f(x) \delta(t-x) dx$$

Si può dimostrare osservando che $f(t)\delta(t-x) = f(x)\delta(t-x)$, dal momento che $\delta(t-x) = 0$ per $x \neq t$. Allora

$$\int_{\mathbb{R}} f(x) \delta(t-x) dx = \int_{\mathbb{R}} f(t) \delta(t-x) dx = f(t) \int_{\mathbb{R}} \delta(t-x) dx = f(t)$$

Base per spazio di segnali a tempo discreto. Nel cercare una base per lo spazio dei segnali $x[n]$ a tempo discreto dobbiamo aggiungere la condizione $\sum_{n=-\infty}^{+\infty} x^2[n] < \infty$. Richiamiamo la definizione (1.1) di impulso unitario:

$$\delta[n] = \begin{cases} 1 & , \text{ se } n = 0 \\ 0 & , \text{ se } n \neq 0 \end{cases}$$

La famiglia dei segnali

$$\{\delta[n-k] : k \in \mathbb{Z}\} \quad (1.2)$$

è una base per lo spazio dei segnali a tempo discreto. Infatti, fissato un generico k , per $n = k$ risulta $\delta[n-k] = 1$, mentre $\delta[n-k] = 0$ per $n \neq k$. Preso dunque un segnale generico $x[n]$ vale che

$$x[n] = \sum_{k=-\infty}^{\infty} x[k] \delta[n-k]$$

Ogni segnale discreto è dunque esprimibile come combinazione lineare dei segnali di base $\delta[n-k]$ ($-\infty < k < \infty$).

Per i segnali a tempo discreto un'utile nozione di prodotto esterno è la seguente:

$$(x[k], y[n]) = \sum_{n=-\infty}^{\infty} x[k] y[n]$$

Ne ricaviamo che (1.2) è una base ortonormale. Infatti,

$$(\delta[n-k], \delta[n-k]) = \sum_{n=-\infty}^{\infty} \delta^2[n-k] = 1$$

Se, invece, $k \neq j$ vale

$$(\delta[n-k], \delta[n-j]) = \sum_{n=-\infty}^{\infty} \delta[n-k] \delta[n-j] = 0$$

Osservazione 1.8. La nozione di combinazione lineare può essere estesa ad un numero finito di segnali ed, attraverso opportune definizioni di limite, anche ad infiniti segnali.

1.3 Sistemi lineari

In questa sezione viene esposta la teoria dei sistemi lineari che ha un interesse specifico nell'ambito della teoria dei segnali; la trattazione verrà ripresa nel capitolo successivo sui sistemi non lineari.

La proprietà di *linearità* di un sistema è esprimibile mediante due proprietà più semplici: omogeneità ed additività .

Siano \mathcal{F}_1 ed \mathcal{F}_2 due spazi di segnali. Un sistema $S : \mathcal{F}_1 \rightarrow \mathcal{F}_2$ è detto *lineare* se soddisfa le seguenti proprietà :

- **omogeneità** : per ogni segnale $f(t) \in \mathcal{F}_1$ e per ogni $a \in \mathbb{R}$ vale

$$S [af(t)] = aS [f(t)] = ag(t) \quad , \quad g(t) \in \mathcal{F}_2$$

- **additività** : per ogni coppia di segnali $f_1(t), f_2(t) \in \mathcal{F}_1$ vale

$$S [f_1(t) + f_2(t)] = S [f_1(t)] + S [f_2(t)] = g_1(t) + g_2(t) \quad , \quad g_1(t), g_2(t) \in \mathcal{F}_2$$

Questa classe possiede l'importante proprietà della *sovrapposizione*: se l'ingresso consiste di una somma pesata di diversi segnali, la risposta del sistema è la somma pesata (sovrapposizione) delle risposte del sistema ai singoli segnali di ingresso. Tale proprietà vale per i sistemi continui e per i sistemi discreti.

1.3.1 Caratterizzazione di un sistema lineare

Sia l'insieme di vettori $\{b_1, \dots, b_n\}$ una base per lo spazio sul quale è definito il sistema lineare S . Allora *il sistema S è univocamente definito conoscendo le risposte di tale sistema agli elementi della base*. Infatti, per ogni ingresso x ottenuto come combinazione lineare degli elementi della base ($x = \sum_i \alpha_i b_i$) vale

$$S [x] = S \left[\sum_i \alpha_i b_i \right] = \sum_i \alpha_i S [b_i]$$

Conoscendo dunque le risposte $S[b_i]$ di tutti gli elementi della base si possono ricostruire le risposte $S[x]$ di ogni segnale x dello spazio.

Queste considerazioni possono essere estese anche a spazi con basi non finite.

Vediamo come questa caratterizzazione possa essere specializzata al caso dei sistemi lineari per segnali a tempo continuo.

Fatto 1.9. Se S è un sistema lineare per segnali a tempo continuo, allora

$$S[f(t)] = \int_{\mathbb{R}} f(x)M(t, x)dx$$

dove $M(t, x) = S[\delta(t - x)]$ è la risposta del sistema S all'impulso $\delta(t - x)$.

Dimostrazione 1.10.

$$S[f(t)] = S\left[\int_{\mathbb{R}} f(x)\delta(t - x)dx\right] = \int_{\mathbb{R}} f(x)S[\delta(t - x)]dx = \int_{\mathbb{R}} f(x)M(t, x)dx$$

dove $M(t, x)$ è la funzione risposta del sistema S alla funzione impulsiva $\delta(t - x)$.

1.3.2 Sistemi tempo-invarianti

Un'altra importante classe di sistemi è quella dei sistemi tempo-invarianti, per i quali una traslazione temporale nel segnale di ingresso produce la medesima traslazione temporale nel segnale di uscita.

Un sistema $S : \mathcal{F}_1 \rightarrow \mathcal{F}_2$ è *tempo-invariante* se, per ogni ingresso $f(t)$ tale che $g(t) = S[f(t)]$, allora $g(t - t_0) = S[f(t - t_0)]$ per ogni t_0 .

Notazione. Nel seguito i sistemi lineari tempo-invarianti saranno denotati come **sistemi LTI**.

Se S è un sistema LTI il suo comportamento è *completamente determinato* dalla sua risposta alla funzione impulsiva $\delta(t)$.

Fatto 1.11. Se S è un sistema lineare tempo-invariante, allora

$$S[f(t)] = \int_{\mathbb{R}} f(x)h(t - x)dx$$

dove $h(t) = S[\delta(t)]$ è la risposta del sistema S all'impulso unitario $\delta(t)$.

Dimostrazione 1.12. Poiché il sistema S è lineare, vale:

$$S[f(t)] = S \left[\int_{\mathbb{R}} f(x) \delta(t-x) dx \right] = \int_{\mathbb{R}} f(x) S[\delta(t-x)] dx$$

Dal momento che è anche tempo-invariante, se $h(t) = S[\delta(t)]$ allora $h(t-x) = S[\delta(t-x)]$ da cui

$$S[f(t)] = \int_{\mathbb{R}} f(x) h(t-x) dx$$

Un risultato analogo vale per i sistemi a tempo discreto.

Fatto 1.13. Se S è un sistema lineare tempo-invariante, allora

$$S[x[n]] = \sum_{k=-\infty}^{\infty} x[k] h[n-k]$$

dove $h[n] = S[\delta[n]]$ è la risposta del sistema S all'impulso unitario $\delta[n]$.

1.3.3 Integrale di convoluzione

La legge che associa a due segnali a tempo continuo f ed h il segnale $S[f(t)]$ di Fatto 1.11 è nota come *prodotto* (od *integrale*) di *convoluzione* di f ed h e viene denotata $f * h$:

$$(f * h)(t) \equiv f(t) * h(t) = \int_{\mathbb{R}} f(x) h(t-x) dx \quad (1.3)$$

Analogamente per i segnali a tempo discreto $x[n]$ ed $h[n]$ viene definita la *somma di convoluzione* come

$$(x * h)[n] \equiv x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k] h[n-k] \quad (1.4)$$

L'operazione di convoluzione gode di interessanti proprietà, che vengono riportate di seguito.

Commutativa

- Caso continuo: $(f * h)(t) = (h * f)(t) = \int_{\mathbb{R}} h(x) f(t-x) dx$;

- Caso discreto: $(x * h)[n] = (h * x)[n] = \sum_{k=-\infty}^{\infty} h[k] x[n - k]$.

Associativa

$$\left(f(t) * h_1(t)\right) * h_2(t) = f(t) * \left(h_1(t) * h_2(t)\right)(t) = f(t) * \left(h_2(t) * h_1(t)\right)(t)$$

Distributiva

$$f(t) * h_1(t) + f(t) * h_2(t) = f(t) * \left(h_1(t) + h_2(t)\right)$$

Elemento neutro

L'elemento neutro della convoluzione è la delta di Dirac:

$$f(t) * \delta(t) = f(t)$$

Se si esprimono i risultati Fatto 1.11 e Fatto 1.13 in termini della convoluzione si può dunque affermare che *la risposta di un sistema lineare tempo-invariante ad un qualsiasi ingresso è ottenuta dalla convoluzione del segnale di ingresso con la risposta all'impulso del sistema.*

1.4 Analisi in frequenza di segnali analogici

1.4.1 Segnali periodici

Un segnale $f(t)$ è detto *periodico di periodo* T ($T \in \mathbb{R}$) se si verifica che

$$f(t) = f(t + T) \quad , \quad \forall t \in \mathbb{R}$$

Se $f(t)$ è periodico di periodo T allora per ogni $k \in \mathbb{Z}$ è anche periodico di periodo kT , cioè $f(t) = f(t + kT)$. Il minimo valore del periodo $T > 0$ che soddisfa la definizione di periodicità viene chiamato *periodo fondamentale* e denotato con T_0 . Se la funzione $f(t)$ è periodica di periodo T risulta univocamente determinata dalla sua restrizione all'intervallo $-T/2 \leq t \leq T/2$.

Si verifica che la combinazione lineare di funzioni periodiche di periodo T è ancora una funzione periodica dello stesso periodo. Se $f(t)$ è periodica

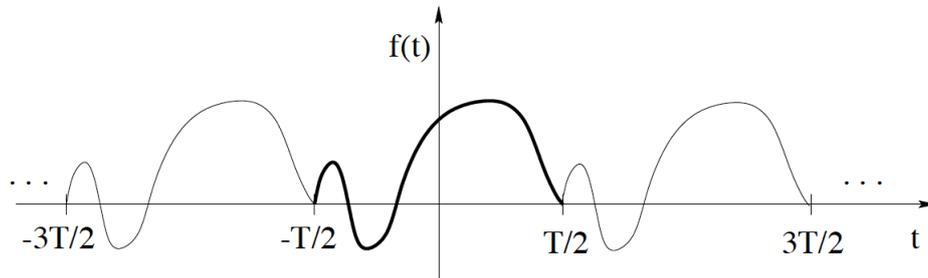


Figura 1.14: Esempio di funzione periodica di periodo T .

di periodo T allora $f(\alpha T)$ ($\alpha \in \mathbb{R}$) è periodica di periodo T/α . In generale, tutte le funzioni periodiche di periodo $T > 0$ possono essere ottenute da quelle periodiche di un diverso periodo $S > 0$ mediante un cambiamento di scala.

Frequenza. Dato un segnale periodico $f(t)$, si definisce la sua *frequenza* come il numero di ripetizioni del periodo per unità di tempo. Se l'unità di misura del tempo è il secondo (sec), la frequenza (denotata ν) può essere misurata in cicli al secondo (hertz, [Hz]) oppure radianti al secondo (rad/sec), nel qual caso è detta anche *pulsazione* (e denotata ω).

Un segnale $f(t)$ di periodo T ha dunque frequenza

$$\nu = \frac{1}{T} \text{ Hz} \quad \Longleftrightarrow \quad \omega = 2\pi\nu = \frac{2\pi}{T} \text{ rad/sec}$$

La *frequenza fondamentale* è la frequenza $\nu_0 = 1/T_0$ che corrisponde al periodo fondamentale T_0 .

Segnale sinusoidale. Un *segnale sinusoidale* si ottiene per dilatazione/compressione e/o traslazione di una funzione seno ed è definito dalla seguente espressione:

$$f(t) = A \sin(\omega_0 t + \phi) \quad , \quad -\infty < t < \infty \quad (1.5)$$

dove A indica l'ampiezza (detto anche *gain*, guadagno) del segnale, ω_0 la frequenza e ϕ la fase iniziale, ossia nell'istante $t = 0$.

Segnale esponenziale complesso. Un *segnale esponenziale complesso*

(detto anche *fasore*) è un segnale della forma

$$f(t) = Ae^{i(\omega_0 t + \phi_0)} \quad , \quad -\infty < t < \infty \quad (1.6)$$

dove $A > 0$ rappresenta l'ampiezza, $\omega_0 \in \mathbb{R}$ è la pulsazione misurata in rad/sec e ϕ_0 la fase iniziale misurata in rad. Il segnale $e^{i(\omega_0 t)}$ è periodico con periodo $T_0 = \frac{\omega_0}{2\pi}$ e frequenza $\frac{\omega_0}{2\pi}$ Hz (equivalentemente, ω_0 rad/sec); il suo modulo è pari ad 1 e la fase è $\omega_0 t$.

1.4.2 Risposta in frequenza dei sistemi LTI

L'importanza delle funzioni esponenziali complesse nello studio dei sistemi LTI risiede nel fatto che la risposta di un sistema LTI sollecitato da questo tipo di segnali è lo stesso esponenziale complesso variato in ampiezza, vale a dire:

$$e^{i\omega t} \longrightarrow H(\omega)e^{i\omega t}$$

La funzione $H(\omega)$ viene detta *risposta in frequenza* o *funzione di trasferimento* del sistema. Come si vedrà in seguito, $H(\omega)$ è la trasformata di Fourier della risposta all'impulso $h(t)$, il che mostra l'interesse dello studio della trasformata di Fourier per l'analisi dei sistemi LTI.

1.4.3 Serie di Fourier

Si è visto come i segnali possano essere rappresentati come combinazione lineare di impulsi opportunamente traslati; viene ora descritta una rappresentazione dei segnali periodici basata sulla combinazione lineare di segnali esponenziali complessi.

Ogni funzione $f_T(t)$ periodica di periodo T che soddisfi le seguenti *condizioni di Dirichlet* può essere espansa in serie di Fourier:

1. $f_T(t)$ deve avere al più un numero finito di discontinuità nel periodo;
2. $f_T(t)$ deve avere al più un numero finito di massimi e minimi nel periodo;

3. $f_T(t)$ deve essere limitata; per poter includere anche le funzioni impulsive questa richiesta viene estesa a quella di assoluta integrabilità per la funzione, ossia:

$$\int_T |f_T(t)| dt < \infty$$

Si considerino i seguenti fasori di modulo unitario:

$$e^{in\omega_0 t} = e^{in\frac{2\pi}{T}t}, \quad n = 0, \pm 1, \pm 2, \dots$$

aventi come frequenza fondamentale $n\omega_0$ e periodo $T = \frac{2\pi}{n\omega_0}$. Si può dimostrare che una funzione $f(t)$ periodica di periodo T che soddisfi le condizioni di Dirichlet (che sono verificate per la maggior parte dei segnali di interesse pratico) può essere ottenuta combinando linearmente le funzioni $e^{in\omega_0 t}$ ($-\infty < n < \infty$):

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{in\omega_0 t} \quad (1.7)$$

dove l'espressione dei coefficienti c_n è la seguente:

$$c_n = \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{-in\omega_0 t} dt \quad (1.8)$$

La serie (1.7) è detta *sviluppo in serie di Fourier* di $f(t)$ ed i coefficienti c_n sono detti *coefficienti di Fourier*; i termini corrispondenti a $n = \pm 1$, entrambi con frequenza ω_0 , sono detti *prime armoniche*, quelli corrispondenti ad $n = \pm k$ vengono chiamati *k-sime armoniche*. Le equazioni (1.7) e (1.8) prendono rispettivamente i nomi di *equazione di sintesi* ed *equazione di analisi* della serie di Fourier: la prima consente di costruire (*sintetizzare*) il segnale $f(t)$ sovrapponendo i singoli fasori della rappresentazione, la seconda permette di decomporre (*analizzare*) il segnale calcolando i coefficienti complessi c_n come somma di fasori.

1.4.4 Trasformata di Fourier

In questa sezione viene esposto il metodo per rappresentare mediante funzioni trigonometriche segnali non periodici (vale a dire: la quasi totalità dei

segnali): dal momento che non possono essere ottenuti come sovrapposizione di funzioni con frequenza multipla di una frequenza data (come nel caso dei segnali periodici) si ricorre alla *trasformata di Fourier* (chiamata anche, nel seguito, *spettro*). La trasformata di Fourier può essere vista come limite di una serie di Fourier (1.7) applicabile a funzioni non periodiche.

Si consideri una funzione $f(t)$ e si denoti con $f_T(t)$ la funzione periodica di periodo T che coincide con $f(t)$ sull'intervallo $[-T/2, T/2]$, il che ci consente di scrivere

$$f(t) = \lim_{T \rightarrow \infty} f_T(t)$$

Si ponga ora

$$\omega_n = n \frac{2\pi}{T} \quad \text{e} \quad \Delta\omega = \omega_{n+1} - \omega_n = \frac{2\pi}{T}$$

da cui si ricava

$$\frac{1}{T} = \frac{\Delta\omega}{2\pi}$$

Dalle equazioni (1.7) e (1.8) si ottiene

$$f_T(t) = \sum_{n=-\infty}^{\infty} c_n e^{in\omega_n t}$$

dove

$$c_n = \frac{\Delta\omega}{2\pi} \int_{-T/2}^{T/2} f_T(t) e^{-in\omega_n t} dt$$

Passando al limite per $T \rightarrow +\infty$ e combinando le espressioni di analisi e sintesi per $f_T(t)$ si ha:

$$f(t) = \lim_{T \rightarrow \infty} f_T(t) = \lim_{T \rightarrow \infty} \frac{1}{2\pi} \sum_{n=-\infty}^{\infty} \left(\int_{-T/2}^{T/2} f_T(t) e^{-in\omega_n(\tau-t)} d\tau \right) \Delta\omega$$

Si ha che $\lim_{T \rightarrow \infty} \omega_n = \omega$, da cui consegue che l'integrale che appare nell'espressione precedente può essere scritto come:

$$\lim_{T \rightarrow \infty} \int_{-T/2}^{T/2} f_T(t) e^{-in\omega_n(\tau-t)} d\tau = \left(\int_{-\infty}^{+\infty} f(\tau) e^{-i\omega\tau} d\tau \right) e^{i\omega t}$$

Denotando con $F(\omega) = \int_{-\infty}^{+\infty} f(\tau)e^{-i\omega\tau}d\tau$ e facendo tendere T all'infinito, si ricava la coppia:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} F(\omega)e^{i\omega t}d\omega \quad (1.9)$$

e

$$F(\omega) = \int_{-\infty}^{+\infty} f(t)e^{-i\omega t}dt \quad (1.10)$$

La (1.10) viene chiamata *trasformata di Fourier* (equazione di analisi) di $f(t)$ e viene denotata $\mathcal{F}\{f(t)\}$; la (1.9) viene detta *antitrasformata* o *trasformata inversa di Fourier* e denotata $\mathcal{F}^{-1}\{F(\omega)\}$. La corrispondenza

$$f(t) \quad \xleftrightarrow{\mathcal{F}} \quad F(\omega)$$

è biunivoca e lineare.

La trasformata di Fourier $F(\omega)$ fornisce lo *spettro di frequenze* del segnale $f(t)$, individuato dal suo modulo $|F(\omega)|$ (chiamato, nel seguito, anche *spettro di ampiezza* o *magnitude*) e dalla sua *fase* $\angle F(\omega)$

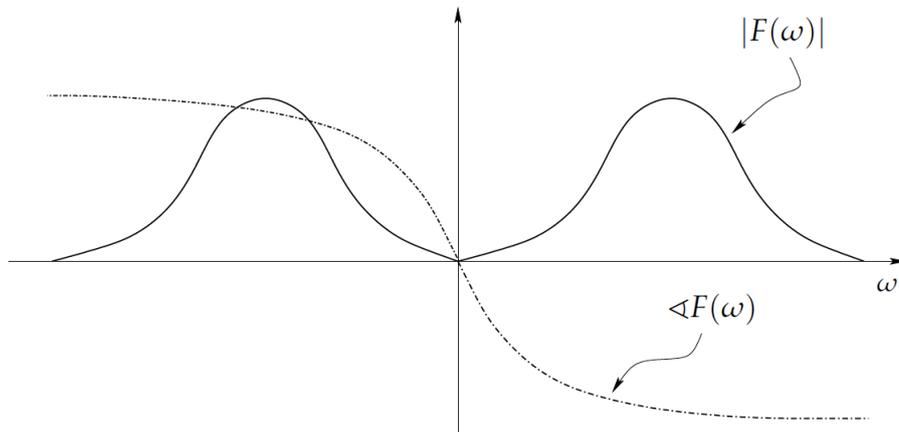


Figura 1.15: Modulo $|F(\omega)|$ e fase $\angle F(\omega)$ della trasformata di Fourier $F(\omega)$ di un segnale reale $f(t)$.

Condizioni di esistenza della trasformata di Fourier

Si riportano di seguito le condizioni sufficienti per l'esistenza della trasformata di Fourier di un segnale $f(t)$.

1. $\int_{\mathbb{R}} f(t)e^{-i\omega t} dt < +\infty \quad \forall t \in \mathbb{R}$;
2. $\exists M > 0$ tale che $|f(t)| \leq M \quad \forall t \in \mathbb{R}$;
3. $f(t)$ deve avere al più un numero finito di discontinuità nel periodo;
4. $f(t)$ deve avere al più un numero finito di massimi e minimi nel periodo;
5. $f(t)$ deve essere assolutamente integrabile:

$$\int_T |f(t)| dt < \infty$$

1.4.5 Banda ed ampiezza di banda di un segnale

Quasi tutti i segnali di interesse pratico possono essere rappresentati nel dominio delle frequenze mediante la trasformata di Fourier. Dall'equazione di sintesi (1.10) si osserva che ogni segnale è dato dalla sovrapposizione di fasori ad ampiezza complessa $F(\omega)$, corrispondenti ad una scelta continua di frequenze ω su \mathbb{R} .

L'intervallo di frequenze su cui lo spettro di ampiezza $|F(\omega)|$ è non nulla viene chiamato *banda* ed è dato dall'insieme $\mathcal{W}_F := \{\omega \in \mathbb{R} : |F(\omega)| \neq 0\}$; la sua misura B_F viene detta *larghezza di banda* (*bandwidth* in inglese). Come verrà esposto nel seguito, lo spettro di ampiezza dei segnali reali è una funzione pari ed è dunque sufficiente considerarne solo le frequenze positive, dimezzando così la banda cosiddetta *bilatera* e facendo riferimento alla banda *monolatera* in cui $\omega \in [0, +\infty)$. Un segnale è detto *a banda limitata dalla frequenza B* quando $\mathcal{W}_F = (-B, B)$ e quindi $|F(\omega)| = 0 \quad \forall \omega > B$.

Osservazione 1.14. Segnali aventi spettri con bande disgiunte sono univocamente ricostruibili dalla loro somma, come si vede in Figura 1.16, nella quale vengono evidenziate le bande monolatero \mathcal{W}_F e \mathcal{W}_G dei due spettri $F(\omega)$ e $G(\omega)$.

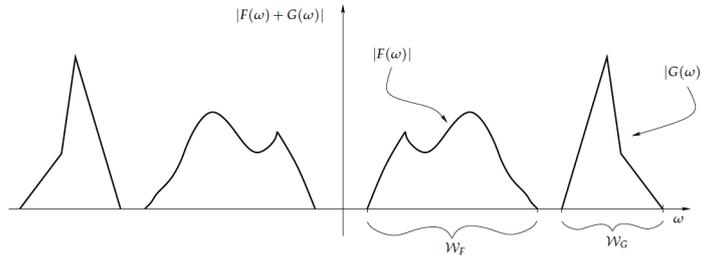


Figura 1.16: Segnali aventi spettri (e banda) disgiunti.

1.4.6 Trasformata di Fourier di funzioni reali

La trasformata di Fourier $F(\omega)$ di un segnale reale $f(t)$ è in generale complessa. Per poter visualizzare graficamente le informazioni contenute nello spettro si considerano separatamente la magnitudine $|F(\omega)|$ e la phase $\angle F(\omega)$ della trasformata, le cui espressioni analitiche sono mostrate di seguito:

$$|F(\omega)| = \sqrt{(\operatorname{Re}\{F(\omega)\})^2 + (\operatorname{Im}\{F(\omega)\})^2} \quad (1.11)$$

$$\angle F(\omega) = \arctan\left(\frac{\operatorname{Im}\{F(\omega)\}}{\operatorname{Re}\{F(\omega)\}}\right) \quad (1.12)$$

Dato un segnale $f(t)$ che ammette trasformata di Fourier $F(\omega)$, dall'identità trigonometrica $e^{i\omega t} = \cos(\omega t) + i \sin(\omega t)$ si ricava la parte reale $\operatorname{Re}\{F(\omega)\}$ e la parte immaginaria $\operatorname{Im}\{F(\omega)\}$ della trasformata:

$$\operatorname{Re}\{F(\omega)\} = \int_{\mathbb{R}} f(t) \cos(\omega t) dt \quad (1.13)$$

$$\operatorname{Im}\{F(\omega)\} = \int_{\mathbb{R}} f(t) \sin(\omega t) dt \quad (1.14)$$

Dalle proprietà delle funzioni seno e coseno si ottiene:

$$\operatorname{Re}\{F(\omega)\} = \operatorname{Re}\{F(-\omega)\} \quad (1.15)$$

$$\operatorname{Im}\{F(\omega)\} = -\operatorname{Im}\{F(-\omega)\} \quad (1.16)$$

Di conseguenza valgono:

$$|F(\omega)| = |F(-\omega)| \quad (1.17)$$

$$\angle F(\omega) = -\angle F(-\omega) \quad (1.18)$$

Le equazioni (1.17) e (1.18) sono equivalenti alla

$$\overline{F(\omega)} = F(\omega) \quad (1.19)$$

Tale proprietà viene detta *simmetria hermitiana* e una condizione necessaria e sufficiente affinché sussista è che il segnale $f(t)$ sia reale.

Notazione 1.15. Nel seguito risulterà utile esprimere, come già per il segnale reale $f(t)$, anche trasformata di Fourier $F(\omega)$ in termini di *ampiezza spettrale* $B(\omega)$ e *fase spettrale* $\Psi(\omega)$:

$$F(\omega) = B(\omega)e^{i\Psi(\omega)}$$

Proprietà della trasformata di Fourier

Si riportano nella Tabella 1.1 le principali proprietà della trasformata di Fourier.

Osservazione 1.16. La proprietà di linearità mostra che lo spettro di frequenza di un segnale periodico è una sovrapposizione di funzioni impulsive nel dominio delle frequenze situate nei multipli interi (*armoniche*) della frequenza fondamentale della funzione $f(t)$.

Osservazione 1.17. La traslazione di un segnale nel tempo preserva il modulo della rispettiva trasformata di Fourier ma introduce una traslazione lineare nella fase pari a $-t_0\omega$.

Proprietà	$\mathbf{f}(t)$	\leftrightarrow	$\mathbf{F}(\omega)$
Linearità	$af(t) + bg(t)$		$aF(\omega) + bG(\omega)$
Coniugazione	$\overline{f(t)}$		$\overline{F(-\omega)}$
Traslazione (tempo)	$f(t - t_0)$		$e^{-i\omega t_0} F(\omega)$
Traslazione (frequenza)	$e^{-i\omega_0 t} f(t)$		$F(\omega - \omega_0)$
Convoluzione (tempo)	$f(t) * g(t)$		$F(\omega)G(\omega)$
Convoluzione (frequenza)	$f(t)g(t)$		$\frac{1}{2\pi} F(\omega) * G(\omega)$
Modulazione	$f(t) \cos(\omega t)$		$\frac{1}{2} [F(\omega + \omega_0) + F(\omega - \omega_0)]$
Scalatura	$f(at)$		$\frac{1}{ a } F\left(\frac{\omega}{a}\right)$

Tabella 1.1: Proprietà della trasformata di Fourier

1.4.7 Risposta in frequenza dei sistemi lineari tempo-invarianti

Si è visto come il comportamento dei sistemi LTI possa essere completamente determinato dalla risposta all'impulso $\delta(t)$. Ricordiamo che, se S è un sistema LTI, $h(t)$ la risposta $S[\delta(t)]$ del sistema all'impulso $\delta(t)$ e $g(t)$ la risposta $S[f(t)]$ del sistema ad un ingresso $f(t)$, vale

$$g(t) = \int_{\mathbb{R}} f(x)h(t-x)dx \equiv f(t) * h(t)$$

In altri termini, la risposta di un sistema LTI ad un ingresso qualsiasi $f(t)$ si può ottenere tramite *convoluzione* dell'ingresso $f(t)$ con la risposta all'impulso $h(t)$. Utilizzando la proprietà della convoluzione riportata in Tabella 1.1 e denotando $F(\omega) = \mathcal{F}[f(t)]$, $H(\omega) = \mathcal{F}[h(t)]$ e $G(\omega) = \mathcal{F}[g(t)]$ le trasformate di Fourier dei segnali $f(t)$, $h(t)$ e $g(t)$ rispettivamente, si ottiene:

$$G(\omega) = F(\omega)H(\omega)$$

La trasformata di Fourier $H(\omega)$ della risposta $h(t)$ all'impulso viene chiamata *funzione di trasferimento* del sistema S .

Fatto 1.18. La risposta $G(\omega)$ di un sistema LTI nel dominio delle frequenze con funzione di trasferimento $H(\omega)$ è il prodotto della trasformata di Fourier $F(\omega)$ dell'ingresso per la funzione di trasferimento $H(\omega)$.

Per quanto riguarda il modulo delle trasformate in gioco, vale che

$$|G(\omega)| = |H(\omega)||F(\omega)|$$

1.4.8 Segnale analitico

Dato un segnale reale $f(t)$ è possibile definire il suo segnale analitico¹ come un segnale complesso la cui parte reale coincida con il segnale $f(t)$ stesso. La definizione di *segnale analitico* $z_f(t)$ è dunque la seguente:

$$z_f(t) = f(t) + i \frac{1}{\pi} \int_{\mathbb{R}} \frac{f(\tau)}{t - \tau} d\tau \equiv f(t) + iH[f(t)] \quad (1.20)$$

dove $H[f(t)]$ è la trasformata di Hilbert del segnale $f(t)$.

Si può esprimere segnale analitico può anche nei termini della sua ampiezza $a_z(t)$ e della sua fase $\varphi_z(t)$:

$$z_f(t) = a_z(t)e^{i\varphi_z(t)}$$

Lo spettro $Z(f) := \mathcal{F}\{z_f(t)\}$ del segnale $z_f(t)$ può essere scritto in termini dell'ampiezza $B_Z(f)$ e della fase $\Psi_Z(f)$ come

$$Z(f) = B_Z(f)e^{i\Psi_Z(f)}$$

La *frequenza istantanea* $f_i(t)$ e il *group delay* $t_f(t)$ sono definiti come

$$f_i(t) = \frac{1}{2\pi} \frac{d\varphi_f}{dt}(t) \quad (1.21)$$

$$t_f(t) = -\frac{1}{2\pi} \frac{d\Psi_z}{df}(f) \quad (1.22)$$

¹Le funzioni complesse di questo tipo soddisfano la condizione di Cauchy-Riemann per la differenziabilità, ref. [17].

Segnali asintotici

Si definisce *segnale asintotico*² un segnale invertibile la cui frequenza istantanea $f_i(t)$ sia invertibile.

Se $f(t)$ egr un segnale asintotico, le equazioni (1.21) e (1.22) descrivono due curve nel piano tempo-frequenza che possono essere viste l'una come l'inversa dell'altra, vale a dire che la frequenza istantanea $f_i(t)$ ed il group delay $t_f(t)$ sono funzioni inverse.

Queste proprietà permettono di calcolare lo spettro del segnale $z_s(t)$ senza bisogno di calcolarne la trasformata di Fourier. L'ampiezza $a_z(t)$ e la fase $\varphi_z(t)$ nel dominio del tempo sono legate all'ampiezza $B_Z(f)$ e alla fase $\Psi_Z(f)$ ($f > 0$) nel dominio delle frequenze dalle seguenti relazioni:

$$B_Z(f) = \frac{a_z(t_f)}{\sqrt{\frac{1}{2\pi} |\varphi_z''(t_f)|}} \quad (1.23)$$

$$\Psi_Z(f) = \varphi_z(t_f) - 2\pi f t_f + \frac{\pi}{4} \text{sign}\left(\frac{df_i}{dt_f}(t_f)\right) \quad (1.24)$$

1.5 Filtri

Un sistema *senza distorsione* è un sistema che riproduce in uscita la stessa forma del segnale in ingresso, a meno di un eventuale fattore amplificativo e/o di un eventuale ritardo temporale. Un tale sistema può essere descritto dalla trasformazione

$$g(t) = A f(t - t_0)$$

Passando alle trasformate di Fourier ed applicando la proprietà di traslazione temporale si ha:

$$G(\omega) = A e^{-i\omega t_0} F(\omega)$$

La funzione di trasferimento $H(\omega)$ del sistema è dunque

$$H(\omega) = A e^{-i\omega t_0}$$

²ref. [19].

Si noti che il modulo della funzione di trasferimento è costante ($|H(\omega)| = A$) mentre la fase è lineare ($\angle H(\omega) = -t_0\omega$)

Si definisce *filtro lineare* un sistema che annulla le componenti armoniche in un determinato intervallo di frequenze e si comporta come un sistema senza distorsione sulle frequenze rimanenti.

Un esempio di filtro lineare è il *filtro passa-basso* o *lowpass*, il quale “passa” (riproduce in uscita con guadagno costante e fase lineare) le componenti con frequenza non superiore ad una certa *frequenza di taglio* ω_0 ed elimina quelle con frequenza superiore a questa soglia. La funzione di trasferimento di un filtro passa-basso è

$$H(\omega) = \begin{cases} Ae^{-i\omega t_0} & , \text{ se } |\omega| < \omega_c \\ 0 & , \text{ se } |\omega| > \omega_c \end{cases}$$

Viceversa, un *filtro passa-alto* o *highpass* elimina le componenti alle frequenze inferiori ad ω_c e passa quelle superiori.

Infine, un *filtro passa-banda* o *bandpass* passa una *banda* (od *intervallo*) di componenti in frequenza $\omega_a < \omega < \omega_b$ ed elimina quelle in frequenze inferiori ad essa. La banda che si interessa preservare viene detta *banda passante* mentre quella di cui si richiede l’eliminazione *banda proibita*.

In Figura 1.17, Figura 1.18 e Figura 1.19 vengono riportati i grafici di magnitudine e phase della tipica funzione di trasferimento di un filtro ideale passa-basso, passa-alto e passa-banda rispettivamente.

Osservazione 1.19. I filtri ideali non sono realizzabili nella pratica e vengono approssimati mediante filtri (analogici o digitali) realizzabili.

La funzione di trasferimento $H(\omega)$ di un filtro ideale passa-basso possiede le seguenti caratteristiche:

1. la magnitudine $|H(\omega)|$ è costante nella banda passante ed identicamente nulla nella banda proibita;
2. la banda passante e la banda proibita sono confinanti;

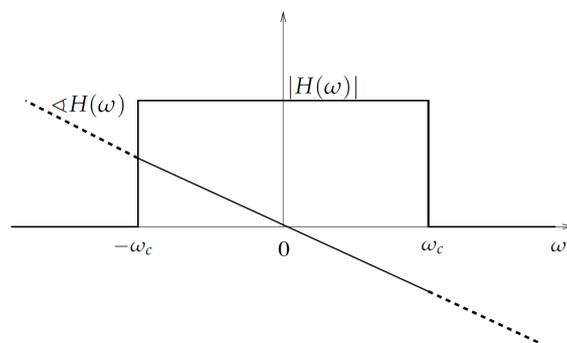


Figura 1.17: Magnitude e phase di un filtro passa-basso ideale.

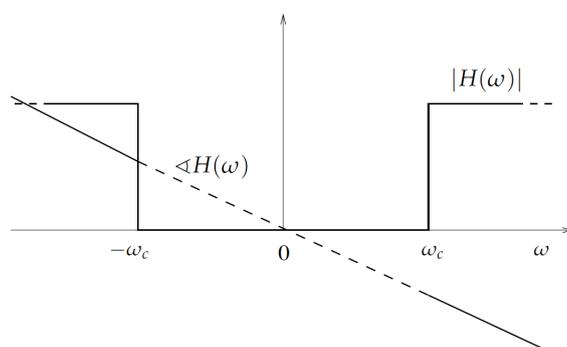


Figura 1.18: Magnitude e phase di un filtro passa-alto ideale.

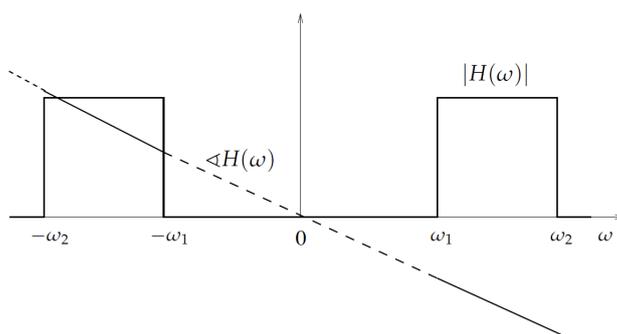


Figura 1.19: Magnitude e phase di un filtro passa-banda ideale.

3. la risposta in fase $\angle H(\omega)$ è lineare, vale a dire tutte le componenti armoniche nella banda passante hanno lo stesso ritardo temporale.

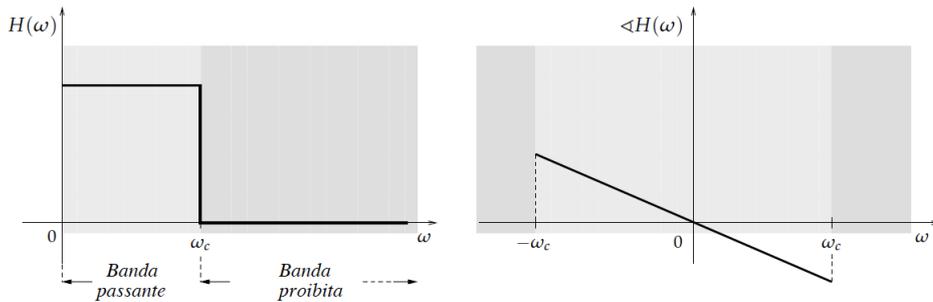


Figura 1.20: Magnitude e phase di un filtro passa-basso ideale con frequenza di taglio ω_c .

Un filtro ideale è un sistema non causale, motivo per il quale le tre condizioni sopra esposte non possono essere verificate contemporaneamente ed un tale filtro è dunque irrealizzabile: si progettano dunque filtri realizzabili con i quali approssimarli.

Indicando con $H(\omega)$ la funzione di trasferimento di un dato filtro passa-basso realizzabile, si è visto come $H(\omega)$ sia completamente specificata dalla sua magnitude $|H(\omega)|$ e dalla sua phase $\angle H(\omega)$. In Figura 1.21 si vedono le forme tipiche di $|H(\omega)|$ e $\angle H(\omega)$ per un filtro passa-basso realizzabile.

Rispetto al caso dei filtri ideali si rilevano le seguenti differenze:

1. la magnitude $|H(\omega)|$ non è costante nella banda passante e non è identicamente nulla nella banda proibita; si possono rilevare inoltre oscillazioni (dette *ripple*) di ampiezza non trascurabile sia nella banda passante che in quella proibita; un parametro importante per quantificare la bontà dell'approssimazione è l'ampiezza della massima oscillazione nella banda proibita oppure, equivalentemente, l'*attenuazione* $-20\log_{10}\delta_2$ dB;

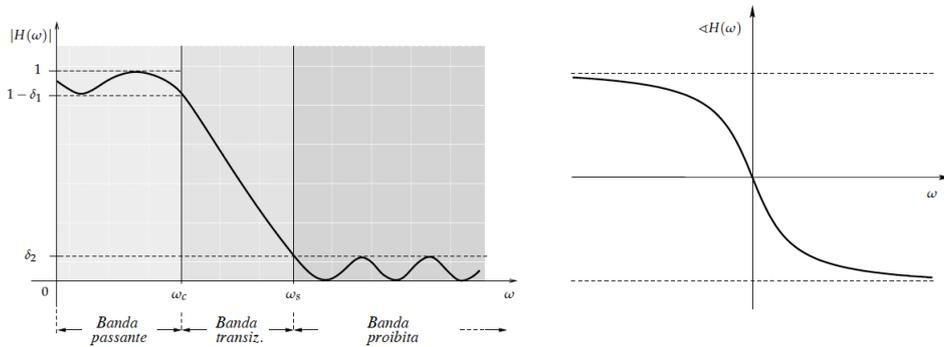


Figura 1.21: Magnitude e phase di un filtro passa-basso realizzabile con frequenza di taglio ω_c .

2. la banda passante e la banda proibita non confinano, ma sono separati da una banda detta *banda di transizione*; parametri importanti sono la frequenza di taglio a 3 dB ω_c , la frequenza di stop ω_s e la dimensione della banda di transizione $\omega_s - \omega_c$;
3. la fase $\angle H(\omega)$ non risulta essere lineare.

1.5.1 Filtri di Butterworth

I filtri di Butterworth costituiscono una famiglia di filtri che soddisfa bene il guadagno in banda passante e meno bene in banda di transizione. Sebbene non esibiscano una fase lineare in banda passante, l'approssimazione non è troppo cattiva e sono tra i più semplici filtri elettronici da realizzare. Un filtro di Butterworth è caratterizzato dall'*ordine* N e dalla *frequenza di taglio* ω_c .

La forma generale del modulo della funzione di trasferimento di tale filtro è

$$|H(\omega)| = \frac{1}{|B_N(i\frac{\omega}{\omega_c})|} = \frac{1}{\sqrt{1 + (\frac{\omega}{\omega_c})^{2N}}}$$

dove $B_N(s)$ è un opportuno polinomio detto N -simo polinomio di Butterworth.

th, mentre la scalatura $\frac{\omega}{\omega_c}$ rispetto ad ω denota la frequenza normalizzata alla frequenza di taglio.

Si riporta in Figura 1.22 la risposta in frequenza di alcuni filtri di Butterworth.

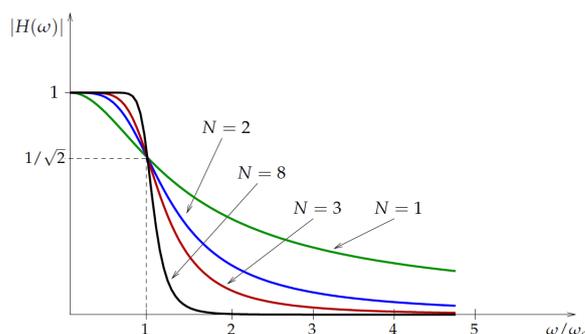


Figura 1.22: Risposta in frequenza di filtri di Butterworth.

Si possono fare le seguenti osservazioni:

- la frequenza di taglio a 3 dB ω_c è indipendente dall'ordine N del filtro;
- l'attenuazione nella banda proibita dipende fortemente dall'ordine N : risulta un'attenuazione di $20N$ dB per decade;
- non è presente ripple né in banda passante né in banda proibita: il filtro di Butterworth è quello che presenta la maggior "flatness" in banda passante.

In fase di progettazione di un filtro Butterworth il parametro ω_c viene scelto come frequenza di taglio desiderata, mentre l'ordine N viene scelto in modo da soddisfare l'attenuazione desiderata in banda proibita.

1.6 Conversione analogico-digitale

I segnali del mondo reale sono analogici, mentre un elaboratore digitale è in grado di memorizzare e trattare esclusivamente sequenze finite di

bit. Per trattare con tecniche digitali i segnali analogici è dunque necessario approssimare questi ultimi con segnali digitali. I sistemi che trasformano un segnale analogico nel corrispondente segnale digitale sono detti *convertitori analogico-digitali* (ADC, Analog-to-Digital Converter), mentre quelli che realizzano l'operazione inversa sono detti *convertitori digitale-analogico* (DAC, Digital-to-Analog Converter). I principi di base che sovrintendono tale processo di conversione sono il *campionamento* e la *quantizzazione*: essi determinano la precisione con cui l'informazione contenuta in un segnale analogico viene trasferita in un segnale digitale.

1.6.1 Campionamento

Campionare un segnale a tempo continuo significa rilevare le ampiezze del segnale su un insieme discreto di istanti. Ad esempio, fissato un intervallo di tempo di lunghezza τ , un campionamento (o *sampling*) uniforme con periodo τ di un segnale $f(t)$ corrisponde all'osservazione del segnale ai tempi $n\tau$ ($-\infty < n < \infty$); il segnale campionato può essere interpretato come il segnale a tempo discreto $f[n\tau]$.

Il sistema *campionatore uniforme* con frequenza di campionamento $\nu_s = 1/\tau$ trasforma un segnale a tempo continuo $f(t)$ nel segnale a tempo discreto $f[n\tau]$. La frequenza di campionamento $\nu_s = 1/\tau$ la frequenza con la quale il segnale originario viene "osservato", ovvero quanti istanti del segnale originario verranno memorizzati in un periodo di segnale campionato. Il campionamento converte la variabile indipendente temporale da continua in discreta.

Teorema del campionamento di Shannon

Dato un segnale $f(t)$, è importante stabilire con quale frequenza esso debba essere campionato affinché il segnale campionato $f[n\tau]$ contenga la stessa informazione di $f(t)$, vale a dire sia possibile ricostruire $f(t)$ a partire dalla sequenza $f[n\tau]$.

Vale il seguente:

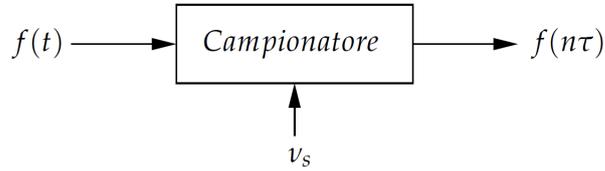


Figura 1.23: Sistema campionatore uniforme.

Teorema 1.20 (Teorema del campionamento di Shannon). *Un segnale $f(t)$ a banda limitata da ν_B Hz (la cui trasformata di Fourier $F(\omega)$ è nulla per $|\omega| > 2\pi\nu_B$) può essere univocamente ricostruito dai suoi campioni $f[n\tau]$ ($-\infty < n < \infty$) presi a frequenza $\nu_s = \frac{1}{\tau}$, se $\nu_s \geq 2\nu_B$. La frequenza $2\nu_B$ è detta tasso o frequenza di Nyquist.*

Sotto l'ipotesi del Teorema di Shannon (segnale a banda limitata da ω_B e frequenza di campionamento $\omega_s \geq 2\omega_B$), un segnale può dunque essere ricostruito senza perdita di informazione partendo dalla sua versione campionata.

1.6.2 Aliasing

Rimane da chiarire che cosa accade quando non sono soddisfatte le ipotesi del Teorema 1.20, ovverosia quando la banda del segnale è illimitata oppure il segnale è a banda limitata ma la frequenza di campionamento è inferiore al tasso di Nyquist ($\nu_s < 2\nu_B \Leftrightarrow \omega_s < 2\omega_B$).

La ricostruzione non risulta più possibile a causa delle sovrapposizioni (in grigio in Figura 1.24) che si creano nella ripetizione periodica del segnale trasformato.

La sovrapposizione di alcune componenti in frequenza viene detto *aliasing*: tale fenomeno rende impossibile la ricostruzione del segnale di partenza da quello campionato, dal momento che frequenze non presenti nel segnale originale vengono introdotte nello spettro. Per evitare tale fenomeno, un sistema campionatore a frequenza ω_s viene generalmente fatto precedere da un filtro passa-basso con frequenza di taglio ω_c pari al più a $\omega_s/2$ (filtro *anti-aliasing*).

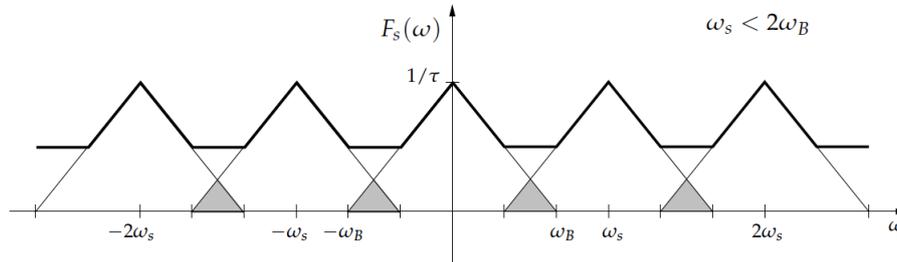


Figura 1.24: Manifestazione del fenomeno dell'aliasing.

1.6.3 Quantizzazione

La *quantizzazione* è il processo che permette di trasformare un segnale a valori continui in un segnale che assume un numero finito di valori. Un modo semplice di quantizzare consiste nel prefissare un insieme finito di l valori numerici $\{x_1, \dots, x_l\}$ ed associare ad ogni numero x il valore numerico x_k che meglio approssima x . Successivamente si *codificano* i valori dell'insieme $\{x_1, \dots, x_l\}$ in parole binarie che possano essere gestite dal calcolatore.

Se i segnali che vengono presi in considerazione hanno ampiezze comprese tra $-\frac{V}{2}$ e $\frac{V}{2}$, si divide l'intervallo $\left[-\frac{V}{2}, \frac{V}{2}\right]$ in l sotto-intervalli, detti *livelli*, e si attribuisce ad un punto $x \in \left[-\frac{V}{2}, \frac{V}{2}\right]$ il centro del livello in cui x si trova. La quantizzazione converte la variabile dipendente da continua in discreta.

La quantizzazione è rappresentata da un numero di *bit*, che descrive il numero di livelli di ampiezza codificabili: se n è il numero di bit usati nella codifica, si possono codificare 2^n livelli di ampiezza.

Il processo di quantizzazione introduce un errore irreversibile nel segnale quantizzato: se si conosce solo il segnale quantizzato, non è possibile ricostruire in maniera univoca il segnale di origine. Tale errore si manifesta sotto forma di *rumore*, tanto più grande quanto più basso è il numero di bit usati per la codifica: per esempio, una codifica a 12 bit aumenta il rumore dello 0.25%. La prassi comune prevede la codifica a 16 bit oppure 24 bit.

Osservazione 1.21. I file utilizzati nel corso del lavoro sono stati codificati a 32 bit.

1.7 Trasformata discreta di Fourier e FFT

La trasformata di Fourier permette di trattare i segnali a tempo continuo dal punto di vista delle loro componenti armoniche (analisi in frequenza). Analogamente, la trasformata di Fourier a tempo discreto permette l'analisi in frequenza dei segnali a tempo discreto.

I segnali a tempo discreto possono essere approssimati mediante segnali digitali, il che permette anche la loro successiva elaborazione tramite calcolatore. L'analisi in frequenza di tali segnali viene effettuata mediante la Trasformata Discreta di Fourier (DFT). In tempi recenti è stata progettata una classe di algoritmi veloci (FFT, Fast Fourier Transform) che permettono di calcolare la DFT in un tempo quasi lineare nella dimensione dei dati.

Il calcolo diretto della DFT è piuttosto costoso, dal momento che richiede $O(N^2)$ operazioni di prodotto (N è la dimensione dello spazio su cui viene applicata la trasformata); si possono ottenere algoritmi FFT per il calcolo della DFT in $O(N \log N)$ operazioni di moltiplicazione.

La FFT viene utilizzata nell'elaborazione dei segnali per:

- analisi spettrale di segnali digitali, con applicazioni alla sintesi di filtri digitali;
- calcolo veloce della convoluzione.

1.7.1 Trasformata di Fourier a Tempo Discreto (DTFT)

La trasformata e l'antitrasformata di Fourier introdotte nella Sezione 1.4.4 operano su segnali continui, sia nel dominio dei tempi che delle frequenze; ne vengono riportate di seguito le espressioni:

$$F(\omega) = \int_{-\infty}^{+\infty} f(t)e^{-i\omega t} dt, \quad f(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} F(\omega)e^{i\omega t} d\omega$$

Fissato un intervallo di ampiezza τ , si consideri il segnale $f_s(t)$ ottenuto campionando un segnale continuo $f(t)$ ai tempi $n\tau$ ($-\infty < n < \infty$); mediante la funzione impulsiva $\delta(t)$ tale segnale può essere riscritto nel seguente modo:

$$f_s(t) = \sum_{n=-\infty}^{\infty} f(n\tau)\delta(t - n\tau)$$

In Figura 1.25 si può vedere un esempio di segnale campionato $f_s(t)$.

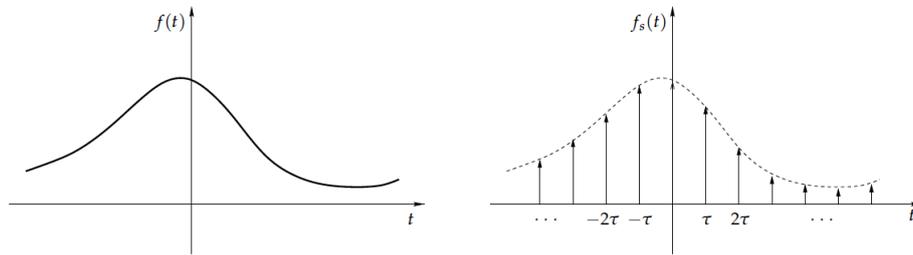


Figura 1.25: Segnale continuo $f(t)$ e segnale campionato $f_s(t)$.

Ricordando che $\mathcal{F}\{\delta(t - t_0)\} = e^{-i\omega t_0}$, per la proprietà di linearità la trasformata di Fourier di $f_s(t)$ risulta essere

$$F_s(\omega) = \sum_{n=-\infty}^{\infty} f(n\tau)e^{-i\omega n\tau}$$

Si consideri ora il cambio di variabile $\omega\tau = \Omega$: denotando con $\nu_s = \frac{1}{\tau}$ la frequenza di campionamento in Hz, risulta che $\Omega = \frac{\omega}{\nu_s}$. La variabile Ω è la frequenza normalizzata alla frequenza di campionamento: poiché ω è data in rad/sec e ν_s è data in cicli/sec, Ω è data in radianti.

Introducendo le notazioni

$$x[n] = f(n\tau) \quad , \quad X(\Omega) = F_s\left(\frac{\Omega}{\tau}\right)$$

si può definire la *trasformata di Fourier di un segnale a tempo discreto* come

$$X(\Omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-i\Omega n}$$

La trasformata $X(\Omega)$ rappresenta la decomposizione di $x[n]$ nelle componenti in frequenza $e^{-i\Omega n}$.

Vale dunque il seguente

Fatto 1.22. Sia $f_s(t)$ il segnale ottenuto campionando $f(t)$ con passo τ e sia $F_s(\omega)$ la sua trasformata di Fourier. Posto $x[n] = f(n\tau)$ e $X(\Omega) = F_s\left(\frac{\Omega}{\tau}\right)$ vale che

$$X(\Omega) = \sum_{n=-\infty}^{\infty} x[n] e^{-i\Omega n} \quad (1.25)$$

$$x[n] = \frac{1}{2\pi} \int_0^{2\pi} X(\Omega) e^{i\Omega n} d\Omega \quad (1.26)$$

Le trasformazioni (1.25) e (1.26) sono dette rispettivamente *trasformata* ed *antitrasformata di Fourier a tempo discreto*; la corrispondenza tra tali funzioni è biunivoca:

$$x[n] \quad \xleftrightarrow{\mathcal{F}_d} \quad X(\omega)$$

Convergenza della DTFT

Come già visto nel caso della serie di Fourier, spesso la condizione di assoluta sommabilità di $x[n]$ viene meno, non garantendo così la convergenza uniforme della (1.25). Dato che tale condizione è comunque solo sufficiente, vi sono sequenze non assolutamente sommabili per le quali la trasformata di Fourier è ugualmente definita: è il caso di segnali che sono quadrato-sommabili ($\sum_{n=-\infty}^{\infty} |x[n]|^2 < \infty$). Per tali segnali si impone la condizione di convergenza in media quadratica:

$$\lim_{N \rightarrow \infty} \int_{-\pi}^{\pi} |X(\Omega) - X_N(\Omega)|^2 d\Omega = 0$$

dove $X_N(\Omega)$ rappresenta la somma parziale delle prime N armoniche.

Proprietà della Trasformata di Fourier a Tempo Discreto

Le proprietà della DTFT risultano di particolare utilità nella semplificazione del processo di analisi di segnali e sistemi.

Traslazione temporale.

$$x[n] \xleftrightarrow{\mathcal{F}_d} X(\omega) \implies x[n-m] \xleftrightarrow{\mathcal{F}_d} e^{-i\Omega m} X(\omega)$$

Traslazione in frequenza.

Se $\mathcal{F}_d\{x[n]\} = X(\Omega)$ allora

$$e^{i\Omega_0 n} x[n] \xleftrightarrow{\mathcal{F}_d} X(\Omega - \Omega_0)$$

Teorema di convoluzione Se

$$x[n] \xleftrightarrow{\mathcal{F}_d} X(\omega) \quad \text{e} \quad y[n] \xleftrightarrow{\mathcal{F}_d} Y(\omega)$$

la convoluzione tra i due segnali è espressa come

$$x[n] * y[n] = \sum_{k=-\infty}^{\infty} x[k] y[n-k]$$

Allora

$$\mathcal{F}_d\{x[n] * y[n]\} = X(\Omega)Y(\Omega)$$

Osservazione 1.23. Con il supporto degli algoritmi veloci per il calcolo della trasformata, il teorema di convoluzione suggerisce che lo studio della relazione di ingresso-uscita di sistemi LTI possa essere effettuato tramite moltiplicazione nel dominio delle frequenze (attraverso l'uso della trasformata e dell'antitrasformata) piuttosto che tramite convoluzione nel dominio del tempo.

1.7.2 Trasformata Discreta di Fourier (DFT)

La trasformata di Fourier a tempo discreto (DTFT) è applicabile a segnali campionati: tali segnali sono a tempo discreto e con frequenza normalizzata nel continuo $[0, 2\pi)$. Per poter trattare opportune approssimazioni di tali segnali con tecniche digitali è necessario ulteriormente:

1. considerare solo un numero finito di campioni nel tempo;

2. effettuare un campionamento anche in frequenza, così da considerare solo un numero finito di frequenze anziché l'intervallo continuo $[0, 2\pi)$.

Si approssima dunque l'informazione contenuta in un segnale $f(t)$ con quella ottenuta dal vettore x formato da N campioni del segnale campionato a passo τ

$$x = [x[0], \dots, x[N]], \quad \text{con} \quad x[n] = f(n\tau), \quad n = 0, \dots, N-1$$

Il vettore x "cattura" quindi l'informazione del segnale contenuta nell'intervallo temporale $[0, (N-1)\tau]$.

In secondo luogo si considera il vettore X formato da N campioni della trasformata a tempo discreto $X(\Omega)$, campionata a intervalli di ampiezza $\frac{2\pi}{N}$

$$X = [X(0), \dots, X(N)], \quad \text{con} \quad X(k) = X\left(\frac{2\pi}{N}k\right), \quad k = 0, \dots, N-1$$

dove, con abuso di notazione, si è denotato con la medesima variabile il vettore $X(k)$ (il quale rappresenta la DFT) e la funzione complessa che rappresenta la trasformata di Fourier a tempo discreto $X(\Omega)$.

Sotto l'ipotesi che l'energia del segnale sia essenzialmente contenuta negli N campioni $x[0], \dots, x[N]$ vale che

$$X\left(\frac{2\pi}{N}k\right) = \sum_{n=-\infty}^{\infty} x[n] e^{-i\frac{2\pi}{N}kn} \approx \sum_{n=0}^{N-1} x[n] e^{-i\frac{2\pi}{N}kn} = X(k)$$

Si può dunque definire la *Trasformata Discreta di Fourier (DFT)* come la trasformazione $\mathcal{F}_{DFT} : \mathbb{C}^N \rightarrow \mathbb{C}^N$ che associa al vettore $x[n]$ ($n = 0, \dots, N-1$) il vettore $X(k)$ ($k = 0, \dots, N-1$) nel modo seguente:

$$X(k) = \sum_{n=0}^{N-1} x[n] e^{-i\frac{2\pi}{N}kn}, \quad k = 0, \dots, N-1 \quad (1.27)$$

La trasformazione \mathcal{F}_{DFT} è lineare ed invertibile e la sua inversa \mathcal{F}_{DFT}^{-1} , detta *antitrasformata*, è data da

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{i\frac{2\pi}{N}kn}, \quad n = 0, \dots, N-1 \quad (1.28)$$

Proprietà della DFT

Nello spazio dei vettori ad N componenti complesse si possono introdurre le seguenti operazioni:

- prodotto: $(f \cdot g)[n] = f[n]g[n]$;
- traslazione ciclica: $(\text{Shift}_a f)[n] = f[\langle n - a \rangle_N]$;
- convoluzione ciclica: $(f \otimes g)[n] = \sum_{k=0}^{N-1} f[k] \cdot g[\langle n - k \rangle_N]$

dove con $\langle s \rangle_N$ si intende il resto della divisione intera di s con N ed il simbolo \otimes denota la convoluzione ciclica.

Si riportano in Tabella 1.2 le proprietà della Trasformata Discreta di Fourier, denotando con $X(k)$ la trasformata di Fourier di $x[n]$ e con $Y(k)$ la trasformata di Fourier di $y[n]$.

Proprietà	$\mathbf{x}[n]$	\leftrightarrow	$\mathbf{X}(k)$
Linearità	$ax[n] + by[n]$		$aX(k) + bY(k)$
Traslazione ciclica	$x[\langle k - a \rangle_N]$		$e^{-i2\pi an} X(k)$
Convoluzione ciclica	$x[n] \otimes y[n]$		$X(k)Y(k)$

Tabella 1.2: Proprietà della trasformata discreta di Fourier

1.7.3 Trasformata Discreta di Fourier e DSP

Nell'ambito del Digital Signal Processing la trasformata discreta di Fourier ricopre un ruolo fondamentale, dal momento che permette di spostare l'analisi e l'elaborazione dei segnali dal dominio del tempo a quello delle frequenze, dove i calcoli risultano essere molto più performanti (si ricordi, a titolo di esempio, che la complessa operazione di convoluzione nel dominio del tempo viene risolta mediante semplice moltiplicazione nel dominio delle frequenze).

La trasformata discreta di Fourier che viene normalmente implementata sui calcolatori³ trasforma un segnale digitale di N campioni (nel dominio del tempo) in una coppia di segnali di $N/2 + 1$ campioni ciascuno nel dominio delle frequenze, come si vede in Figura 1.26. Tale coppia di segnali rappresentano la parte reale e quella immaginaria del segnale complesso che è il risultato della trasformata di Fourier e si è soliti riferirsi ad essi come alle *magnitude* e alla *phase*, rispettivamente.

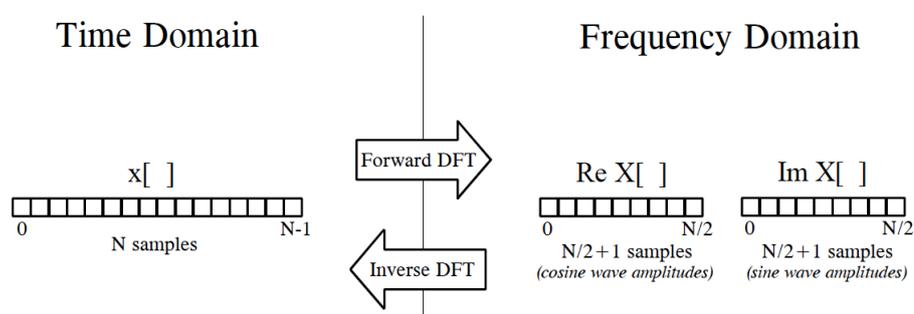


Figura 1.26: Passaggio dal dominio del tempo al dominio delle frequenze mediante la DFT.

Il numero N di campioni del segnale nel dominio del tempo viene preferibilmente scelto pari ad una potenza di 2, dal momento che gli algoritmi più efficienti⁴ per il calcolo della DFT operano di norma con segnali che abbiano tali lunghezze. Qualora il segnale in questione sia costituito da un numero diverso di campioni, si è soliti “paddare⁵ di zeri” il segnale, ovvero si aggiunge dei campioni alla fine del segnale fino a raggiungere la lunghezza necessaria, ciascuno dei campioni aggiunti avente valore 0: in tal modo l’informazione contenuta nel segnale non viene modificata e si ottimizza il procedimento di trasformazione.

³In alcuni testi tale processo viene indicato come “Trasformata discreta di Fourier reale”, per distinguerla da quella complessa introdotta nella precedente sezione.

⁴detti *Fast Fourier Transform*, si veda la Sezione 1.7.4.

⁵dall’inglese *to pad*, “imbottire”.

Il grafico della magnitude risulta molto utile quando si lavora con segnali di tipo audio, dal momento che permette una rapida visualizzazione delle frequenze presenti nel segnale: si usa “plottare” i risultati in scala logaritmica, dal momento che l’ampiezza (chiamata anche *gain*⁶) viene di norma espressa in decibel⁷.

1.7.4 Fast Fourier Transform

La principale ragione che rende importante la DFT è l’esistenza di algoritmi veloci per il suo calcolo: tali algoritmi richiedono $O(N \log N)$ operazioni invece di $O(N^2)$ e per tale motivo sono detti *algoritmi FFT (Fast Fourier Transform)*.

Nell’elaborazione dei segnali, la FFT ha tre principali settori di utilizzo:

1. analisi spettrale di segnali analogici e digitali, con applicazioni alla sintesi di filtri digitali;
2. calcolo veloce della convoluzione;
3. compressione di dati per la memorizzazione e la trasmissione efficiente degli stessi.

Per quanto riguarda la relazione tra la trasformata di Fourier $F(\omega)$ e la trasformata discreta di Fourier, se le approssimazioni fatte (passo di campionamento τ , numero N di campioni) risultano ragionevolmente buone, allora

$$x[n] = f(n\tau), \quad n = 0, \dots, N-1 \quad \Longrightarrow \quad X(k) \approx F\left(\frac{2\pi k}{N\tau}\right), \quad k = 0, \dots, N-1$$

Tali approssimazioni risultano accettabili quando l’intervallo di campionamento del dominio temporale consente di avere un aliasing trascurabile nel dominio delle frequenze.

⁶dall’inglese, “guadagno”.

⁷Si veda Nota 2, Sezione 3.1.

Capitolo 2

Sistemi non lineari e serie di Volterra: il modello

Nella prima parte di questo capitolo viene presentata la teoria di Volterra per lo studio dei sistemi non lineari, argomento matematico che è oggetto di crescente interesse in diversi ambiti scientifici applicativi, non da ultimo quello dell'acustica musicale. Nella seconda, viene descritta la serie di Volterra *diagonale*, che fornisce il modello utilizzato per l'emulazione del dispositivo non lineare in interesse.

La teoria riguardante i sistemi non lineari viene riportata dai trattati[5] (del quale vengono adottate le notazioni), [6], [7] e [10]. La descrizione del modello utilizzato viene ripresa dai trattati [31], [28], [27], [29], [30], [25].

Nel seguito, un *sistema* che prende in ingresso un input $x(t)$ e restituisce la risposta $y(t)$ sarà denotato nel modo seguente:

$$y(t) = \mathbf{T}[x(t)], \quad t \in \mathbb{R} \tag{2.1}$$

dove \mathbf{T} è un *operatore*.

Osservazione 2.1. I sistemi che considereremo nel corso di questa trattazione saranno tutti tempo-invarianti.

Un *sistema* è detto *causale* se la risposta al tempo t del sistema ad un generico input $x(t)$ dipende solamente dai valori che tale input assume in t e prima di t e non da quelli che assumerà in futuro.

Un *sistema* è altresì detto *stabile* se l'output del sistema ad un input $x(t)$ limitato sarà limitato a sua volta. Si è soliti fare riferimento a questa definizione anche con il termine di stabilità *BIBO* (Bounded Input - Bounded Output).

La teoria sviluppata nel corso di questo capitolo vale generalmente per funzioni input $x : \mathbb{R} \rightarrow \mathbb{R}$ continue a tratti nella variabile temporale t e per corrispondenti funzioni output $y : \mathbb{R} \rightarrow \mathbb{R}$ continue.

2.1 Sistemi di Volterra lineari

Viene presentata innanzitutto la teoria di Volterra per i sistemi di primo ordine, dal momento che la teoria per i sistemi di ordine superiore può essere vista come una sua generalizzazione. Viene qui ripresa la teoria già esposta nel capitolo precedente ed inserita nell'ambito della teoria dei sistemi di Volterra.

Un *operatore* \mathbf{T}_1 è detto *del primo ordine* se soddisfa il principio di sovrapposizione, ovvero sia dati

$$y_n(t) = \mathbf{T}_1 [x_n(t)] \quad , \quad n = 1, 2, \dots$$

l'operatore \mathbf{T}_1 è del primo ordine se, a fronte di un input della forma

$$x(t) = \sum_{n=1}^N c_n x_n(t) \quad , \quad c_n \in \mathbb{R} \quad , \quad N \in \mathbb{N}$$

produce la risposta

$$\begin{aligned} y(t) = \mathbf{T}_1 [x(t)] &= \mathbf{T}_1 \left[\sum_{n=1}^N c_n x_n(t) \right] = \sum_{n=1}^N c_n \mathbf{T}_1 [x_n(t)] \\ &= \sum_{n=1}^N c_n y_n(t) \end{aligned}$$

Si denota \mathbf{H}_1 l'operatore di primo ordine tempo-invariante (od *operatore di Volterra del primo ordine*) cosicché per un sistema lineare tempo-invariante vale che

$$y(t) = \mathbf{H}_1[x(t)]$$

2.1.1 Integrale di convoluzione

Un sistema è completamente caratterizzato nel momento in cui si conosce la trasformazione \mathbf{T} che fornisce l'uscita di tale sistema a fronte di un generico ingresso. Come già visto, nel caso di un sistema lineare il concetto di basilare importanza è la possibilità di conoscere, nel momento in cui si conosca la risposta $q(t)$ ad un input $v(t)$, anche le risposte del sistema che siano combinazioni lineari dell' input $v(t)$. In termini matematici, se l'input è

$$x(t) = \sum_n c_n v(t - \tau_n) \quad (2.2)$$

e denotando $q(t)$ la risposta del sistema lineare a tale input

$$q(t) = \mathbf{H}_1[v(t)] \quad (2.3)$$

la risposta del sistema alla combinazione lineare è

$$y(t) = \sum_n c_n q(t - \tau_n)$$

ed il sistema lineare è dunque *caratterizzato* dall'equazione (2.3) per input del tipo $v(t)$ e sue combinazioni lineari.

Per ottenere una caratterizzazione *completa* del sistema lineare occorre individuare un input (che chiameremo anche *waveform*) “di base” con il quale poter rappresentare, mediante opportune combinazioni lineari, tutti gli input di interesse.

La scelta della waveform di base è orientata dall'osservazione che tutte le waveform di interesse possano essere approssimate da una funzione a gradini. Si definisce dunque la waveform di base $u_\delta(t)$ nel modo seguente:

$$u_\delta(t) = \begin{cases} \frac{1}{\delta} & \text{se } |t| < \frac{1}{2}\delta \\ 0 & \text{altrimenti} \end{cases} \quad (2.4)$$

Tale funzione possiede l'interessante proprietà di avere integrale su \mathbb{R} pari ad 1, indipendentemente da δ :

$$\int_{-\infty}^{+\infty} u_\delta(t) dt = 1$$

Si può dunque esprimere l'approssimazione di una generica waveform $x(t)$ in termini della funzione $u_\delta(t)$:

$$x_\delta(t) = \sum_{k\delta=-\infty}^{+\infty} \delta x(k\delta) u_\delta(t - k\delta) \quad (2.5)$$

Tale approssimazione migliora al rimpicciolirsi di δ , di modo che, al tendere di δ a 0, $x_\delta(t)$ tende ad $x(t)$:

$$x(t) = \lim_{\delta \rightarrow 0} x_\delta(t)$$

Si ottiene la caratterizzazione generale di un sistema lineare tempo-invariante determinando la risposta a $x_\delta(t)$ e facendo tendere δ a 0. La risposta del sistema lineare a $u_\delta(t)$ sarà denotata $h_1(t, \delta)$

$$h_1(t, \delta) = \mathbf{H}_1[u_\delta(t)]$$

La risposta del sistema all'input $x_\delta(t)$ sarà dunque

$$\begin{aligned} y_\delta(t) &= \mathbf{H}_1[x_\delta(t)] \\ &= \sum_{k\delta=-\infty}^{+\infty} \delta x(k\delta) h_1(t - k\delta, \delta) \end{aligned} \quad (2.6)$$

Mandando δ al limite si ottiene

$$\lim_{\delta \rightarrow 0} y_\delta(t) = y(t) = \mathbf{H}_1[x(t)]$$

Una conseguenza di questa azione è che l'ampiezza di $u_\delta(t)$ sia infinitesimale, cosicché possiamo considerare $\delta = d\sigma$ e $k\delta = \sigma$. Al tendere di δ a 0 la sommatoria in (2.6) tende dunque al valore dell'integrale

$$y(t) = \int_{-\infty}^{+\infty} x(\sigma) h_1(t - \sigma) d\sigma \quad (2.7)$$

nel quale si è definito l'ente

$$h_1(t) = \lim_{\delta \rightarrow 0} h_1(t, \delta) = \mathbf{H}_1[u_0(t)]$$

che chiameremo *kernel di Volterra del primo ordine*. L'input $u_0(t)$ è detto *impulso unitario* e, di conseguenza, $h_1(t)$ è la *risposta all'impulso* del sistema lineare in considerazione.

Osservazione 2.2. Per la proprietà commutativa della convoluzione l'equazione (2.7) è equivalente alla

$$y(t) = \int_{-\infty}^{+\infty} h_1(\tau)(t - \tau)d\tau \quad (2.8)$$

Portando avanti il processo di passaggio al limite si ottiene

$$\begin{aligned} x(t) &= \lim_{\delta \rightarrow 0} \int_{-\infty}^{+\infty} x(\sigma)u_\delta(t - \sigma)d\sigma \\ &= \int_{-\infty}^{+\infty} x(\sigma)u_0(t - \sigma)d\sigma \end{aligned} \quad (2.9)$$

Come già visto, l'equazione (2.7) è il noto integrale di convoluzione tra il generico input $x(t)$ e la risposta all'impulso $h_1(t)$. La risposta di un sistema lineare ad un qualsiasi input può essere ottenuta semplicemente tramite convoluzione, una volta che sia nota la sua risposta all'impulso $h_1(t)$. Un sistema lineare è *completamente caratterizzato* dalla sua risposta all'impulso.

2.1.2 Causalità

La caratteristica dell'output al tempo t di un sistema *causale* è quella di dipendere solamente dai valori che ha assunto l'input fino a quel momento e non da quelli che assumerà in futuro. Si può ricavare ora una condizione necessaria e sufficiente affinché un sistema lineare tempo-invariante possa essere considerato causale.

Si riporta di seguito per comodità l'integrale di convoluzione (2.7)

$$y(t) = \int_{-\infty}^{+\infty} x(\sigma)h_1(t - \sigma)d\sigma$$

Se si considera t come l'istante attuale, i valori passati dell'input sono tutti i valori $x(\sigma)$ per cui $\sigma < t$ cosicché l'output al tempo t derivante dal passato dell'input è

$$y(t) = \int_{-\infty}^t x(\sigma)h_1(t - \sigma)d\sigma$$

Il futuro dell'input è $x(\sigma)$ per $\sigma > t$ al quale corrisponde l'output

$$y(t) = \int_t^{+\infty} x(\sigma)h_1(t - \sigma)d\sigma$$

Per far sì che la risposta del sistema non sia funzione dei valori futuri dell'input si richiede che quest'ultimo integrale sia nullo per ogni funzione $x(\sigma)$. Questo sarà vero solo se $h_1(t - \sigma) = 0$ per $\sigma > t$ oppure, equivalentemente, $h_1(\tau) = 0$ per $\tau < 0$. Tale condizione è necessaria e sufficiente per la causalità di un sistema lineare tempo-invariante. La richiesta che $h_1(\tau)$ sia nullo per $\tau < 0$ è soddisfatta dal kernel di un sistema causale: ricordiamo che $h_1(t) = \mathbf{H}_1[u_0(t)]$ cosicché l'istante in cui l'impulso arriva al sistema definisce l'istante iniziale $t = 0$ della funzione $h_1(t)$; inoltre, prima di tale istante è impossibile che l'impulso abbia effetto sull'output di un sistema causale. Di conseguenza, dato che si suppone che il sistema sia inizialmente in quiete, $h_1(t)$ deve essere nullo per $t < 0$.

2.1.3 Stabilità

La nozione di stabilità BIBO per un sistema lineare tempo-invariante sottoposto ad un input limitato ($|x(t)| < M$ per $M \in \mathbb{R}$) è equivalente alla seguente condizione:

$$\int_{-\infty}^{+\infty} |h_1(\tau)|d\tau \quad (2.10)$$

La condizione (2.10) è sufficiente: se $|x(t)| < M$ e (2.10) è soddisfatta, usando (2.8) si ottiene che

$$\begin{aligned} |y(t)| &= \left| \int_{-\infty}^{+\infty} h_1(\tau)x(t-\tau)d\tau \right| \\ &\leq \int_{-\infty}^{+\infty} |h_1(\tau)||x(t-\tau)|d\tau \\ &< M \int_{-\infty}^{+\infty} |h_1(\tau)|d\tau \end{aligned}$$

Per provare che la condizione è anche necessaria, si mostra che se (2.10) non è soddisfatta esiste almeno un input limitato per il quale l'output è illimitato in almeno un istante temporale. Tale istante viene scelto in $t = 0$. Si ha che

$$y(0) = \int_{-\infty}^{+\infty} h_1(\tau)x(-\tau)d\tau$$

Si consideri il seguente input

$$x(-t) = \begin{cases} +1 & \text{per } t \text{ tale che } h(t) \geq 0 \\ -1 & \text{per } t \text{ tale che } h(t) < 0 \end{cases}$$

Tale input è limitato dal momento che $|x(t)| = 1$ per ogni t . La risposta del sistema a tale input nell'istante $t = 0$ è

$$y(0) = \int_{-\infty}^{+\infty} |h_1(\tau)|d\tau = +\infty$$

dato che per ipotesi (2.10) non è soddisfatta. La condizione di stabilità (2.10) è dunque necessaria e sufficiente per un sistema lineare tempo-invariante.

2.2 Sistemi di Volterra non lineari

La rappresentazione appena descritta per i sistemi lineari può essere generalizzata ed estesa al caso dei sistemi non lineari tramite gli operatori di Volterra di ordini superiori.

Analogamente al caso lineare, si dice che un operatore \mathbf{T}_p è di ordine p ($p \in \mathbb{N}$) se la sua risposta ad un input del tipo

$$x(t) = \sum_{n=1}^N c_n x_n(t) \quad , \quad c_n \in \mathbb{R} \ , \ N \in \mathbb{N}$$

è della forma

$$\begin{aligned} y_\delta(t) &= \mathbf{H}_p [x_\delta(t)] \\ &= \sum_{n_1=1}^N \cdots \sum_{n_p=1}^N \mathbf{T}_p \{c_{n_1} x_{n_1}(t), \dots, c_{n_p} x_{n_p}(t)\} \\ &= \sum_{n_1=1}^N \cdots \sum_{n_p=1}^N c_{n_1} \cdots c_{n_p} \mathbf{T}_p \{x_{n_1}(t), \dots, x_{n_p}(t)\} \end{aligned} \quad (2.11)$$

L'operatore \mathbf{T}_p è p -lineare (cioè lineare nei suoi p argomenti) e inoltre, se $x_1 = x_2 = \cdots = x_n \equiv x$ allora $\mathbf{T}_p \{x_{n_1}(t), \dots, x_{n_p}(t)\} = \mathbf{T}_p [x]$.

La risposta di un operatore di ordine p è generalmente determinata da operazioni su p segnali in input alla volta.

Se l'operatore $\mathbf{T}_p[\cdot]$ è tempo-invariante viene detto *operatore di Volterra di ordine p* e denotato $\mathbf{H}_p[\cdot]$. La regola che lega la risposta $y_p(t)$ del sistema non lineare all'input $x(t)$ è dunque

$$y_p(t) = \mathbf{H}_p [x(t)] \quad (2.12)$$

Si ottiene la caratterizzazione funzionale dell'operatore di Volterra di ordine p con un procedimento simile a quello del caso lineare: utilizzando la waveworm $u_\delta(t)$ (definita in (2.4)) determiniamo in prima istanza la risposta del sistema all'input $x_\delta(t)$ (la funzione a gradini con la quale si approssima l'input $x(t)$) e otteniamo la risposta ad $x(t)$ passando al limite $\delta \rightarrow 0$.

Dall'equazione (2.5) si ottiene

$$x_\delta(t) = \sum_{k\delta=-\infty}^{+\infty} \delta x(k\delta) u_\delta(t - k\delta)$$

Dalla (2.11) si ricava la risposta ad $x_\delta(t)$

$$\begin{aligned} y_\delta(t) &= \mathbf{H}_p [x(t)] = \\ &= \sum_{m_1\delta=1}^N \cdots \sum_{m_p\delta=1}^N \mathbf{H}_p \{ \delta x(m_1\delta) u_\delta(t - m_1\delta), \dots, \delta x(m_p\delta) u_\delta(t - m_p\delta) \} \\ &= \sum_{m_1\delta=1}^N \cdots \sum_{m_p\delta=1}^N \delta x(m_1\delta), \dots, \delta x(m_p\delta) \mathbf{H}_p \{ u_\delta(t - m_1\delta), \dots, u_\delta(t - m_p\delta) \} \end{aligned}$$

Si utilizza la seguente notazione per indicare la componente funzionale:

$$h_p(t - m_1\delta, \dots, t - m_p\delta) = \mathbf{H}_p \{ u_\delta(t - m_1\delta), \dots, u_\delta(t - m_p\delta) \}$$

Proseguiamo nell'espressione di $y_\delta(t)$ si ottiene

$$y_\delta(t) = \sum_{m_1\delta=1}^N \cdots \sum_{m_p\delta=1}^N \delta x(m_1\delta), \dots, \delta x(m_p\delta) h_p(t - m_1\delta, \dots, t - m_p\delta)$$

Come nel caso lineare, si definisce

$$\begin{aligned} h_p(t) &= \lim_{\delta \rightarrow 0} h_p(t - m_1\delta, \dots, t - m_p\delta) \\ &= \mathbf{H}_p \{ u_0(t - m_1\delta), \dots, u_0(t - m_p\delta) \} \end{aligned} \quad (2.13)$$

Di nuovo la sommatoria (p -dimensionale, in questo caso) tende all'integrale

$$\begin{aligned} y_p(t) &= \mathbf{H}_p [x(t)] \\ &= \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} x(\sigma_1) \cdots x(\sigma_p) h_p(t - \sigma_1, \dots, t - \sigma_p) d\sigma_1 \cdots d\sigma_p \end{aligned} \quad (2.14)$$

che, operando il cambio di variabile $\tau_n = t - \sigma_n$ per $n = 1, \dots, p$, può essere espresso anche nella seguente forma:

$$\begin{aligned} y_p(t) &= \mathbf{H}_p [x(t)] \\ &= \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} h_p(\tau_1, \dots, \tau_p) x(t - \tau_1) \cdots x(t - \tau_p) d\tau_1 \cdots d\tau_p \end{aligned} \quad (2.15)$$

L'integrale (2.15) è detto *funzionale di Volterra di ordine p* e $h_p(\tau_1, \dots, \tau_p)$ è il *kernel di Volterra di ordine p* dell'operatore di Volterra \mathbf{H}_p di ordine p .

2.2.1 Causalità

Come già nel caso lineare, l'ipotesi di causalità (indipendenza della risposta di un sistema dai valori futuri dell'input) per i sistemi non lineari è equivalente all'annullarsi del kernel di Volterra negli istanti passati.

Viene riportata di seguito la rappresentazione dell'operatore di Volterra di ordine p mediante integrale di convoluzione:

$$y_p(t) = \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} h_p(t - \sigma_1, \dots, t - \sigma_p) x(\sigma_1) \dots x(\sigma_p) d\sigma_1 \dots d\sigma_p$$

Affinché il sistema sia causale, la risposta $\mathbf{H}_p(\mathbf{t})$ non può essere funzione del futuro dell'input. Se t è l'istante presente, i valori futuri della funzione input sono $x(\sigma_1), \dots, x(\sigma_p)$ per ogni $\sigma_n > t$ con $n = 1, \dots, p$. Si richiede, dunque, che $h_p(t - \sigma_1, \dots, t - \sigma_p) = 0$ per ogni $\sigma_n > t$ con $n = 1, \dots, p$. Questa ipotesi può essere espressa in una forma più utile utilizzando il cambio di variabili $\tau_n = t - \sigma_n$ per $n = 1, \dots, p$, da cui si ricava che un operatore di Volterra di ordine p è causale se e solo se

$$h_p(\tau_1, \dots, \tau_p) = 0 \quad \text{per ogni } \tau_n < 0 \quad , \quad n = 1, \dots, p$$

In altri termini, $h_p(\tau_1, \dots, \tau_p)$ può essere non nullo solo se ciascuno dei suoi argomenti è non negativo. Questa condizione è necessaria e sufficiente affinché l'operatore \mathbf{H}_p sia causale.

2.2.2 Stabilità

La definizione di *stabilità* che si utilizza per gli operatori di Volterra è chiamata anche, come già indicato, criterio BIBO (ogni input limitato produce una risposta limitata).

Come già nel caso lineare, un operatore di Volterra H_p è BIBO stabile se

$$\int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} |h_p(t - \sigma_1, \dots, t - \sigma_p)| d\sigma_1 \dots d\sigma_p < +\infty \quad (2.16)$$

Questa condizione è sufficiente ma non necessaria: sia $|x(t)| < M$ ($M \in \mathbb{R}$) un input limitato, supponendo che la (2.16) valga, risulta che

$$\begin{aligned} |\mathbf{H}_p[x(t)]| &= \left| \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} h_p(\tau_1, \dots, \tau_p) x(t - \tau_1) \dots x(t - \tau_p) d\tau_1 \dots d\tau_p \right| \\ &\leq \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} |h_p(\tau_1, \dots, \tau_p)| |x(t - \tau_1)| \dots |x(t - \tau_p)| d\tau_1 \dots d\tau_p \\ &< M \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} |h_p(\tau_1, \dots, \tau_p)| d\tau_1 \dots d\tau_p < +\infty \end{aligned}$$

A differenza del caso lineare (in cui la condizione di stabilità (2.10) è necessaria e sufficiente), la condizione (2.16) è solamente sufficiente: non sono note condizioni che siano anche necessarie per gli operatori di Volterra di ordine $p \geq 2$.

2.3 Serie di Volterra

Dal punto di vista fisico un sistema può essere visto come una “black box” sulla quale non si hanno informazioni, salvo la conoscenza del suo output $y(t)$ a fronte di un input noto $x(t)$. Un sistema non-lineare tempo-invariante può essere modellato tramite una somma infinita di integrali di convoluzione multidimensionale, che prende il nome di *sviluppo in serie di Volterra*¹:

¹Tale serie venne presentata nel 1925 dal matematico Vito Volterra nel suo trattato [7] e, data l'importanza che ha assunto in numerosissimi ambiti scientifici, porta oggi il suo nome.

$$\begin{aligned}
y(t) &= \sum_{n=1}^{\infty} \int_{R^n} h_n(\tau_1, \dots, \tau_n) x(t - \tau_1), \dots, x(t - \tau_n) d\tau_1 \dots d\tau_n \\
&= \int_{\mathbb{R}} h_1(\tau_1) x(t - \tau_1) d\tau_1 \\
&\quad + \int_{\mathbb{R}^2} h_2(\tau_1, \tau_2) x(t - \tau_1) x(t - \tau_2) d\tau_1 d\tau_2 \\
&\quad + \dots \\
&\quad + \int_{R^n} h_n(\tau_1, \dots, \tau_n) x(t - \tau_1) \dots x(t - \tau_n) d\tau_1 \dots d\tau_n \\
&\quad + \dots
\end{aligned} \tag{2.17}$$

nel quale per $n = 1, 2, \dots$

$$h_n(\tau_1, \dots, \tau_n) = 0 \quad \forall \tau_j < 0 \quad , \quad j = 1, 2, \dots, n \tag{2.18}$$

I (2.18) sono detti *kernel di Volterra* del sistema .

La serie (2.17) può essere espressa tramite gli operatori di Volterra descritti in precedenza:

$$y(t) = \mathbf{H}_1[x(t)] + \mathbf{H}_2[x(t)] + \mathbf{H}_3[x(t)] + \dots + \mathbf{H}_n[x(t)] + \dots \tag{2.19}$$

operatori dei quali riportiamo di seguito l'espressione

$$\mathbf{H}_n[x(t)] = \int_{R^n} h_n(\tau_1, \dots, \tau_n) x(t - \tau_1) \dots x(t - \tau_n) d\tau_1 \dots d\tau_n \tag{2.20}$$

Osservazione 2.3. Una forma equivalente della (2.17) che verrà utilizzata nel seguito è la seguente:

$$\begin{aligned}
y(t) &= \sum_{n=1}^{\infty} \int_{R^n} h_n(t - \tau_1, \dots, t - \tau_n) x(\tau_1) \dots x(\tau_n) d\tau_1 \dots d\tau_n \\
&= \int_{\mathbb{R}} h_1(t - \tau_1) x(\tau_1) d\tau_1 \\
&\quad + \int_{\mathbb{R}^2} h_2(t - \tau_1, t - \tau_2) x(\tau_1) x(\tau_2) d\tau_1 d\tau_2 \\
&\quad + \dots \\
&\quad + \int_{R^n} h_n(t - \tau_1, \dots, t - \tau_n) x(\tau_1) \dots x(\tau_n) d\tau_1 \dots d\tau_n \\
&\quad + \dots
\end{aligned} \tag{2.21}$$

A seconda del sistema che si voglia rappresentare, gli integrali possono essere calcolati su intervalli temporali finiti od infiniti: il supporto dei kernel di Volterra 2.18 definisce la *memoria* del sistema in considerazione, ovverosia l'intervallo temporale nei quali i valori passati dell'input possono influenzare il valore attuale dell'output. La serie di Volterra può infatti essere vista come una serie di Taylor con memoria, dal momento che può caratterizzare sistemi le cui risposte dipendono anche da valori passati dell'input (e non solo dall'istante presente come nel caso della serie di Taylor).

Le funzioni di input sono tipicamente definite su spazi di Hilbert reali e separabili come $L^2[a, b]$, quelle di output sullo spazio $C[a', b']$ delle funzioni continue limitate. Come per la serie di Taylor, la convergenza di una serie di Volterra può essere garantita solo per un range limitato di ampiezza dell'input e l'insieme delle funzioni di input deve quindi essere ristretto ad un opportuno sottoinsieme. Per esempio, se le funzioni di input sono definite su un sottoinsieme compatto del dominio, si può applicare il teorema di Stone-Weierstrass per mostrare che ogni sistema non lineare continuo può essere uniformemente approssimato ad un grado arbitrario di precisione da una serie di Volterra di grado finito opportuno.

2.4 Il modello

Si procede ora con la descrizione del modello utilizzato, inserendolo nel contesto numerico utilizzato nell'applicazione.

2.4.1 Modello di Wiener-Hammerstein

Molti sistemi non lineari possono essere modellati tramite uno delle seguenti possibilità

- modello di Wiener: sistema lineare senza memoria seguito da uno non lineare senza memoria;

- modello di Hammerstein: sistema non lineare senza memoria seguito da uno lineare senza memoria;
- modello di Wiener: sistema lineare senza memoria seguito da uno non lineare senza memoria seguito a sua volta da un altro sistema lineare con memoria.

Ciascuno dei tre modelli è caratterizzato dal fatto che la memoria del sistema viene gestita dalla parte lineare, mentre le non linearità sono nella parte senza memoria.

Il modello che sarà utilizzato è quello di Hammerstein, del quale vengono descritti di seguito input ed output.

- il segnale $x(t)$ dato in input alla parte non lineare del modello produce l'output

$$w(t) = a_0 + \sum_{n=1}^{\infty} a_n [x(t)]^n \quad (2.22)$$

dove gli a_i sono i coefficienti dello sviluppo in serie di Taylor; tale valore è giustificato dal fatto che il sistema lineare qui ipotizzato è senza memoria, quindi ogni valore in uscita dipende solamente dal valore attualmente in ingresso e può essere modellato dallo sviluppo in serie di Taylor;

- l'output $w(t)$ diventa a sua volta input per la parte lineare, producendo dunque l'output

$$y(t) = h(t) * w(t) = \int_{\mathbb{R}} h(t - \tau) w(\tau) d\tau \quad (2.23)$$

dove $h(t)$ è la risposta all'impulso del sistema lineare (la quale, come già visto, lo caratterizza completamente).

2.4.2 Serie di Volterra diagonale

Il kernel di Volterra $h_n(t)$ di ordine n è una funzione di n variabili: all'aumentare dell'accuratezza del modello cresce dunque notevolmente anche la

complessità computazionale, dal momento che ciascun kernel dovrebbe essere rappresentato sul calcolatore da una matrice multidimensionale di dimensione pari all'ordine del kernel. Se si suppone di ridurre la “portata” dei kernel assumendo che la parte del sistema con memoria risieda nella parte lineare del modello, si può ottenere un modello più adatto alle esigenze computazionali. Si definisce *serie di Volterra diagonale* il caso particolare di (2.21) con la seguente ipotesi sui kernel (2.18):

$$\forall n \in \mathbb{N} \exists j \in \{2, \dots, n\} \text{ tale che } \tau_j \neq \tau_1 \implies h_n(t - \tau_1, \dots, t - \tau_n) = 0$$

In altri termini, gli unici valori non nulli che possono assumere i kernel sono quelli per cui $\tau_1 = \tau_2 = \dots = \tau_n$, cioè quelli sulla diagonale principale del kernel. In linea di principio, la semplificazione qui adottata potrebbe sembrare molto forte; nella pratica, la maggior parte dei valori “esterni alla diagonale” sono molto vicini a zero. Nella maggior parte dei casi pratici di simulazione, si preferisce accettare l'errore dovuto alla semplificazione del modello a fronte dei vantaggi derivanti dalla possibilità di estendere la simulazione ad ordini superiori di non linearità .

Si introduce la notazione $\tau := \tau_1 = \dots = \tau_n$ per tutti i valori “diagonali” dei kernel, da cui risulta la seguente forma dello sviluppo in serie di Volterra diagonale:

$$\begin{aligned} y(t) &= \int_{\mathbb{R}} h_1(t - \tau) x(\tau) d\tau \\ &+ \int_{\mathbb{R}} h_2(t - \tau) [x(\tau)]^2 d\tau \\ &+ \dots \\ &+ \int_{\mathbb{R}} h_n(t - \tau) [x(\tau)]^n d\tau \\ &+ \dots \end{aligned} \tag{2.24}$$

Dalle formule (2.22), (2.23) e (2.24) si vede come il modello di Hammerstein

coincida con la serie di Volterra diagonale:

$$\begin{aligned}
 y(t) &= \int_{\mathbb{R}} h(t - \tau) w(\tau) d\tau \\
 &= \int_{\mathbb{R}} h(t - \tau) \left(a_0 + \sum_{n=1}^{\infty} a_n [x(t)]^n \right) d\tau \\
 &= \int_{\mathbb{R}} h(t - \tau) a_0 d\tau + \sum_{n=1}^{\infty} \int_{\mathbb{R}} a_n h(t - \tau) [x(t)]^n d\tau \\
 &\equiv h_0 + h_1(t) * x(t) + h_2(t) * [x(t)]^2 + h_3(t) * [x(t)]^3 + \dots
 \end{aligned} \tag{2.25}$$

Questo è il modello utilizzato in questo lavoro per dare una forma matematica alle non linearità dei sistemi audio.

Osservazione 2.4. I kernel di Volterra sono la parte del modello che caratterizza il sistema: una volta che siano noti gli h_n , fissato un opportuno livello di accuratezza n , è possibile ricostruire l'output $y(t)$ a fronte di qualsiasi ingresso $x(t)$ tramite la (2.24). Nella prossimo Capitolo verrà descritto il metodo per ricavare i kernel a partire dalla risposte all'impulso del sistema.

2.5 Modello numerico

La versione a tempo discreto dello sviluppo in serie di Volterra (2.17) è la seguente:

$$\begin{aligned}
 y[m] &= h_0 + \sum_{i_1=0}^{\infty} h_1[i_1] x[m - i_1] \\
 &\quad + \sum_{i_1=0}^{\infty} \sum_{i_2=0}^{\infty} h_2[i_1, i_2] x[m - i_1] x[m - i_2] \\
 &\quad + \dots \\
 &\quad + \sum_{i_1=0}^{\infty} \sum_{i_2=0}^{\infty} \dots \sum_{i_p=0}^{\infty} h_p[i_1, i_2, \dots, i_p] x[m - i_1] x[m - i_2] \dots x[m - i_p] \\
 &\quad + \dots
 \end{aligned} \tag{2.26}$$

dalla quale si ricava la versione finita da utilizzare per le applicazioni numeriche, supponendo che la non linearità del sistema venga modellata fino all'ordine p e la memoria fino all'ordine M :

$$\begin{aligned}
y[m] = & h_0 + \sum_{i_1=0}^M h_1[i_1] x[m - i_1] \\
& + \sum_{i_1=0}^M \sum_{i_2=0}^M h_2[i_1, i_2] x[m - i_1] x[m - i_2] \\
& + \dots \\
& + \sum_{i_1=0}^M \sum_{i_2=0}^M \dots \sum_{i_p=0}^M h_p[i_1, i_2, \dots, i_p] x[m - i_1] x[m - i_2] \dots x[m - i_p]
\end{aligned} \tag{2.27}$$

Dalle formule (2.24) si ricava la versione finita della serie di Volterra diagonale:

$$y[m] = h_0 + \sum_{i=0}^{M-1} h_1[i] x[m - i] + \sum_{i=0}^{M-1} h_2[i] x^2[m - i] + \sum_{i=0}^{M-1} h_3[i] x^3[m - i] + \dots \tag{2.28}$$

Osservazione 2.5. La prima sommatoria della serie discreta modella la parte lineare del sistema, mentre gli addendi successivi aggiungono le non linearità di ordine via via crescente: la simulazione sarà dunque tanto più accurata quanto più alto sarà l'ordine dei kernel calcolati.

Capitolo 3

Emulazione di un sistema audio

In questo capitolo vengono presentati gli algoritmi e le tecniche di misura utilizzate per riprodurre virtualmente tramite algoritmi in Matlab gli effetti di distorsione di un segnale audio compiuti da un dispositivo non lineare.

Le informazioni riguardanti il segnale di test vengono riportate dai trattati [12],[13],[14],[15], [16]. Il metodo di misura è stato ripreso dai trattati [27], [29], [28], [30], [31], [23], [24], [25] e dal Brevetto [39]. Le note sui segnali audio sono prese dai trattati [8], [9].

Nota. In alcuni casi, in assenza di una traduzione italiana adeguata dei termini impiegati, verrà utilizzata la terminologia inglese.

Il sistema audio non lineare che si vuole emulare è un pedale distorsore per chitarra elettrica denominato “BOSS SD-1 Super Overdrive” (Figura 3.1); le registrazioni sono state effettuate con l’ausilio della scheda audio per computer MOTU Traveler-mk3 (Figura 3.2) e del software audio REAPER v.4.52/64; l’editor audio utilizzato è Adobe Audition CS5.5.



Figura 3.1: Pedale BOSS SD-1 Super Overdrive per chitarra elettrica.



Figura 3.2: Scheda audio MOTU Traveler-mk3.

3.1 Synchronized Sine Sweep

Un *segnale di tipo sine sweep*¹ è un segnale la cui frequenza istantanea varia nel tempo. Viene comunemente usato come segnale di test in diversi ambiti, tra cui quello dell’acustica. Un segnale sine sweep può essere definito come

$$s(t) = a(t) \sin(\varphi(t)) \quad (3.1)$$

dove $a(t)$ rappresenta l’ampiezza² e $\varphi(t)$ la fase del segnale.

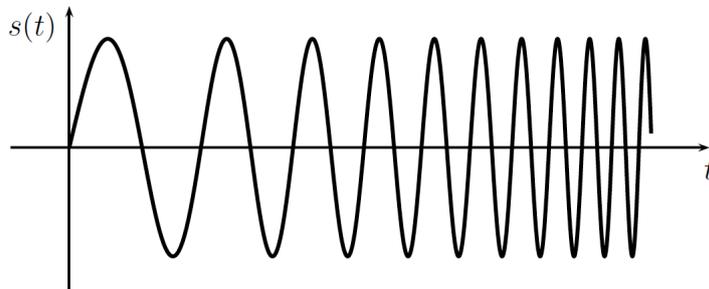


Figura 3.3: Esempio di sine sweep.

Il segnale di input che viene utilizzato per misurare il sistema da modellizzare è il Synchronized Sine Sweep presentato da Antonín Novák³ nel 2009

¹dall’inglese, “spazzare”.

²L’ampiezza di un segnale audio ne determina il *volume*. Con il termine volume si indica il *livello di pressione sonora* (SPL, dall’inglese *Sound Pressure Level*), la cui misura è data dal rapporto tra la pressione del suono p e la pressione di un suono convenzionale di riferimento p_{rif} , espresso in *decibel* (dB): $SPL = 20 \log_{10}(p/p_{rif})$. Tali livelli variano logicamente da 0 dB (soglia di udibilità) a 120 dB (soglia del dolore). Negli editor e nella strumentazione audio (e in questo lavoro) si è soliti esprimere il volume non più in termini assoluti ma in termini relativi, ovverosia comparato ad un altro livello sonoro (detto “fondo scala”): per esempio, un suono a 0 dB ha possiede lo stesso livello sonoro del fondo scala, mentre un suono a -6 dB possiede un livello sonoro pari a metà del fondo scala. Il rapporto tra l’ampiezza α di un segnale audio e il suo volume “relativo” vol_{db} è dato dalla relazione $vol_{db} = 20 \log_{10}(\alpha)$ (cit. [8]).

³nella sua tesi di dottorato [12].

come una modifica del segnale Exponential Sine Sweep⁴. Le caratteristiche di tale segnale permettono di utilizzare le risposte all'impulso non lineari che si ottengono dalla misurazione del dispositivo "così come sono", mentre utilizzando l'Exponential Sine Sweep si rendeva necessario progettare dei filtri che ne "riallineassero" le fasi.

Il *Synchronized Sine Sweep* (SySS) è un segnale la cui frequenza istantanea aumenta esponenzialmente col tempo e nel quale le fasi delle frequenze armoniche⁵ sono sincronizzate. La sua espressione⁶ è

$$\begin{aligned} s(t) &= a(t) \sin \left[2\pi f_1 \int_0^t \exp\left(\frac{t'}{L}\right) dt' \right] \\ &= \sin \left\{ 2\pi f_1 L \left[\exp\left(\frac{t}{L}\right) - 1 \right] \right\} \end{aligned}$$

dove f_1 è la frequenza iniziale del segnale al tempo $t = 0$ ed L è il tasso di crescita esponenziale della frequenza. Il parametro L dipende dalla durata temporale del segnale e dalla frequenza f_2 cui arriva lo sweep nell'istante finale. Il segnale $s(t)$ può essere espresso in termini della (3.1) con ampiezza costante $a(t) = 1$ e fase istantanea

$$\varphi(t) = 2\pi f_1 L \left[\exp\left(\frac{t}{L}\right) - 1 \right] \quad (3.2)$$

La frequenza istantanea (1.21) è conseguentemente

$$f_i(t) = f_1 \exp\left(\frac{t}{L}\right) \quad (3.3)$$

Il segnale $s(t)$ e la sua frequenza istantanea $f_i(t)$ sono invertibili sull'intervallo temporale $[0, T]$, il che fa di $s(t)$ un segnale asintotico: il group delay t_f è dunque la funzione inversa della frequenza istantanea ed ha la seguente

⁴presentato da Angelo Farina nel 2000 [23] e base del metodo della Nonlinear Convolution esposto nella Sezione 3.2.

⁵Data una *frequenza fondamentale* f , si definiscono le sue *frequenze armoniche* f_n ($n = 2, 3, \dots$) come le frequenze che sono multiple intere di tale frequenza fondamentale: $f_n = n f$.

⁶dal momento che l'ampiezza $a(t)$ del segnale non influisce sul calcolo dei parametri dello sweep verrà per il momento posta uguale ad 1.

espressione:

$$t_f(f) = L \ln \left(\frac{f_i}{f_1} \right) \equiv L \ln \left(\frac{f}{f_1} \right) \quad (3.4)$$

La durata T dello sweep può essere espressa come il lasso di tempo tra la frequenza iniziale f_1 e quella finale f_2 :

$$T = L \ln \left(\frac{f_2}{f_1} \right)$$

e dunque il coefficiente L è definito come

$$L = \frac{T}{\ln \left(\frac{f_2}{f_1} \right)} \quad (3.5)$$

L'espressione del parametro L viene di seguito ridefinita in modo da consentire la sincronizzazione delle fasi delle risposte in frequenza non lineari. Sia Δt_m l'istante in cui la frequenza istantanea $f_i(\Delta t_m)$ dello sweep è data da

$$f_i(\Delta t_m) = m f_1, \quad m \in \mathbb{N} - \{0\}$$

vale a dire l'istante in cui la frequenza istantanea è l' m -sima armonica della frequenza iniziale f_1 . Dalla (3.4) si ricava

$$\Delta t_m = L \ln(m) \quad (3.6)$$

e la fase istantanea al tempo Δt_m è dunque data da

$$\varphi(\Delta t_m) = 2\pi f_1 L(m - 1)$$

Come mostrato in Figura 3.4, si richiede che lo sweep $s(t)$ agli istanti Δt_m sia nullo ($s(\Delta t_m) = 0$), con il vincolo aggiuntivo che la sua derivata prima sia positiva ($s'(\Delta t_m) > 0$).

Tali condizioni implicano che

$$\varphi(\Delta t_m) = 2k\pi, \quad k \in \mathbb{Z}$$

da cui si ricava che

$$f_1 L(m - 1) = k$$

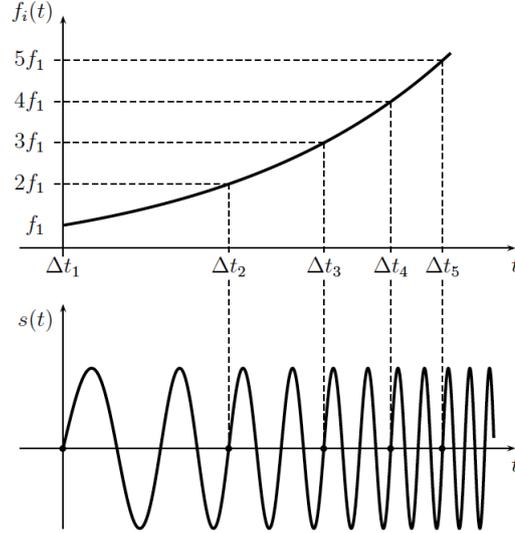


Figura 3.4: Sincronizzazione delle fasi nel SySS.

Una condizione sufficiente affinché valga l'equazione precedente è che $f_1 L \in \mathbb{Z}$. Utilizzando tale condizione e la definizione (3.5) si può scrivere

$$L = \frac{1}{f_1} \text{Round} \left(\frac{\tilde{T} f_1}{\ln \left(\frac{f_2}{f_1} \right)} \right)$$

dove “Round” rappresenta l'arrotondamento all'intero più vicino e \tilde{T} è la durata “indicativa” (cioè a meno degli arrotondamenti) che si vorrebbe avesse lo sweep. La durata effettiva T dello sweep $s(t)$ risente degli arrotondamenti necessari per la sincronizzazione delle fasi ed ha dunque l'espressione seguente:

$$T = \frac{1}{f_1} \text{Round} \left(\frac{\tilde{T} f_1}{\ln \left(\frac{f_2}{f_1} \right)} \right) \ln \left(\frac{f_2}{f_1} \right)$$

In definitiva, l'espressione completa del Synchronized Sine Sweep $s(t)$ è la seguente

$$s(t) = \sin \left\{ 2\pi \text{Round} \left(\frac{\tilde{T} f_1}{\ln \left(\frac{f_2}{f_1} \right)} \right) \left[\exp \left(\frac{f_1 t}{\text{Round} \left(\frac{\tilde{T} f_1}{\ln \left(\frac{f_2}{f_1} \right)} \right)} \right) \right] \right\} \quad (3.7)$$

3.1.1 Proprietà nel dominio delle frequenze

Come ricordato in precedenza, lo spettro di un segnale asintotico può essere ricavato dalle proprietà nel dominio delle frequenze senza bisogno di calcolare la trasformata di Fourier. Per ricavare tali proprietà nel caso del segnale $s(t)$ sopra definito si utilizza la sua espressione analitica

$$z_s(t) = s(t) + iH[s(t)] = a_s(t)e^{i\varphi_s(t)}$$

dove $H[\cdot]$ è la trasformata di Hilbert di $s(t)$, $a_s(t)$ l'ampiezza e $\varphi_s(t)$ e la fase di $z_s(t)$.

Lo spettro $Z_s(f)$ del segnale $z_s(t)$ può essere scritto in termini dell'ampiezza $B_s(f)$ e della fase $\Psi_s(f)$ come

$$Z_s(f) = B_s(f)e^{i\Psi_s(f)}$$

Il segnale $s(t)$ è di tipo asintotico, poiché la frequenza istantanea f_i ed il group delay t_f sono funzioni inverse l'una dell'altro. In tali condizioni, lo spettro di ampiezza di $z_s(t)$ può essere definito come

$$B_s(f) = \frac{a_s(t_f)}{\sqrt{\frac{1}{2\pi}|\varphi''(t_f)|}} \quad (3.8)$$

dove, utilizzando (3.2) e (3.4), la derivata seconda della fase $\varphi''(t_f)$ viene riscritta come

$$\varphi_s''(t) = \frac{2\pi}{L} f_1 \exp\left(\frac{t}{L}\right) = \frac{2\pi f_i(t)}{L} \quad (3.9)$$

Usando le equazioni (3.8), (3.9) e l'equivalenza $f \equiv f_i$, lo spettro di ampiezza di $s(t)$ può essere scritto nel modo seguente (per $f > 0$):

$$B_s(f) = \sqrt{\frac{L}{f}} \quad (3.10)$$

3.1.2 Filtro inverso

Per mettere in atto il procedimento di Nonlinear convolution⁷ è necessario trovare un'espressione per il segnale $\tilde{s}(t)$ inverso di $s(t)$.

⁷Sezione 3.2.

Il filtro inverso $\tilde{s}(t)$ è definito come quel segnale che, convoluto con il segnale “diretto” $s(t)$, fornisce come risultato la funzione delta di Dirac $\delta(t - t_0)$ traslata nel tempo:

$$s(t) * \tilde{s}(t) = \delta(t - t_0)$$

Si prendono in considerazione i segnali analitici $z_s(t)$ e $\tilde{z}_s(t)$ associati rispettivamente ai segnali $s(t)$ ed $\tilde{s}(t)$. Il segnale analitico $\tilde{z}_s(t)$ del filtro inverso è

$$\tilde{z}_s(t) = \tilde{s}(t) + iH[\tilde{s}(t)]$$

mentre la trasformata di Fourier $Z_{\tilde{s}}(f)$ del segnale analitico $\tilde{z}_s(t)$ è l'inverso della trasformata del segnale analitico $z_s(t)$:

$$Z_{\tilde{s}}(f) = \frac{1}{Z_s(f)}$$

Per dare fornire un'espressione al segnale analitico del filtro inverso si ricavano le ampiezze e le fasi delle trasformate di Fourier dei due segnali, nel modo seguente:

$$Z_{\tilde{s}}(f) = \frac{1}{B_s(f)e^{i\Psi_s(f)}} = \frac{1}{B_s(f)}e^{-i\Psi_s(f)}$$

da cui si ricava

$$B_{\tilde{s}}(f) = |Z_{\tilde{s}}(f)| = \frac{1}{B_s(f)} \quad (3.11)$$

$$\Psi_{\tilde{s}}(f) = -\Psi_s(f) \quad (3.12)$$

Il filtro inverso analitico di un segnale asintotico è a sua volta un segnale asintotico:

$$z_{\tilde{s}(t)} = a_{\tilde{s}}(t)e^{i\tilde{\varphi}(t)}$$

La fase $\tilde{\varphi}(t)$ viene ricavata dall'espressione di $\tilde{t}_f(f)$: utilizzando le equazioni (1.22) e (3.12) si ottiene che

$$\tilde{t}_f(f) = -\frac{1}{2\pi} \frac{d\Psi_{\tilde{s}}}{df}(f) = \frac{1}{2\pi} \frac{d\Psi_s}{df}(f)$$

Di conseguenza, si ha

$$\tilde{t}_f(f) = -t_f(f) \quad (3.13)$$

$$\tilde{\varphi}_s(t) = \varphi_s(-t) \quad (3.14)$$

Per ricavare l'ampiezza $a_{\tilde{s}}(t)$ del filtro inverso si utilizza (1.23)

$$B_{\tilde{s}}(f) = \frac{a_{\tilde{s}}(\tilde{t}_f)}{\sqrt{\frac{1}{2\pi}|\tilde{\varphi}''(\tilde{t}_f)|}}$$

Dalle equazioni (3.13) e (3.14) si ha che $\tilde{\varphi}''(\tilde{t}_f) = \varphi_s''(-t)$ e sostituendo tale espressione nella (3.9) si ottiene

$$B_{\tilde{s}}(f) = \frac{a_{\tilde{s}}(\tilde{t}_f)}{\sqrt{\frac{f}{L}}} \quad (3.15)$$

Dalle equazioni (3.10), (3.11) e (3.15) si ricava

$$a_{\tilde{s}}(\tilde{t}_f) = \frac{f}{L}$$

Utilizzando (3.3), l'ampiezza $a_{\tilde{s}}(t)$ è data da

$$a_{\tilde{s}}(t) = \frac{f_1}{L} \exp\left(-\frac{t}{L}\right)$$

ed il filtro analitico è infine espresso come

$$z_{\tilde{s}}(t) = \frac{f_1}{L} \exp\left(-\frac{t}{L}\right) e^{i\varphi_s(-t)}$$

o, in una forma più breve,

$$z_{\tilde{s}}(t) = \frac{f_1}{L} \exp\left(-\frac{t}{L}\right) z_s(-t)$$

Il filtro inverso $\tilde{s}(t)$ è dunque espresso come una copia a tempo invertito del segnale diretto $s(t)$ con una modulazione in ampiezza:

$$\tilde{s}(t) = \frac{f_1}{L} \exp\left(-\frac{t}{L}\right) s(-t) \quad (3.16)$$

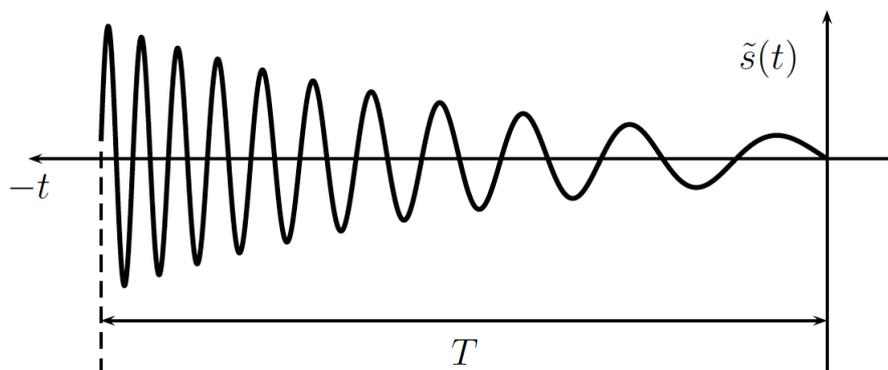


Figura 3.5: Filtro inverso $\tilde{s}(t)$ del Synchronized Sine Sweep $s(t)$.

3.2 Nonlinear Convolution

La *Nonlinear Convolution* è un metodo⁸ per processare segnali audio, al fine di ricreare una simulazione udibile delle modificazioni prodotte su un segnale da un sistema. Questo metodo rappresenta un'estensione del procedimento di *auralizzazione* basato sulla convoluzione lineare del segnale con la risposta all'impulso di un sistema (modellato come) lineare: fornisce dunque la possibilità di emulare anche sistemi (modellati come) nonlineari, caratterizzati nei termini delle distorsioni armoniche a diversi ordini.

Il diagramma a blocchi del metodo è mostrato in Figura 3.6. Di seguito la descrizione del procedimento:

1. si genera un segnale di test $s(t)$ di tipo sine sweep⁹ che viene utilizzato come segnale di input per il sistema che si vuole testare;
2. si registra l'output distorto $y(t)$;
3. si genera il segnale $\tilde{s}(t)$ inverso¹⁰ del segnale $s(t)$, costituito dal segnale $s(t)$ a tempo invertito e modulato in ampiezza, dimodoché la

⁸presentato dall'Ing. Angelo Farina nel corso della 110a Convention della Audio Engineering Society nel 2001, [23].

⁹in questo lavoro verrà utilizzato il segnale "Synchronized Sine Sweep", Sezione 3.1.

¹⁰"filtro inverso" in [23].

convoluzione tra $s(t)$ ed $\tilde{s}(t)$ produca l'elemento neutro delta di Dirac $\delta(t)$;

4. si effettua la convoluzione tra il segnale di output $y(t)$ e il filtro inverso $\tilde{s}(t)$ (tale procedimento verrà anche chiamato *deconvoluzione*), ottenendo così le risposte all'impulso non lineari;
5. si calcolano i kernel della serie di Volterra diagonale.

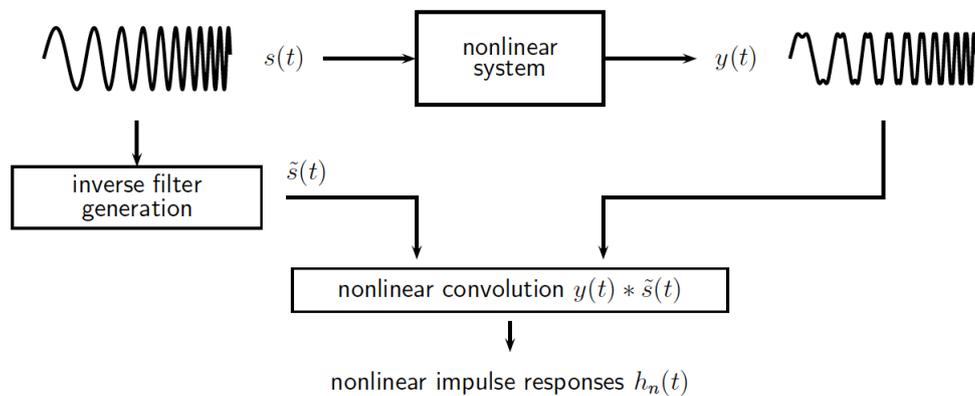


Figura 3.6: Diagramma a blocchi per la Nonlinear Convolution.

3.2.1 Misura e deconvoluzione

Il segnale utilizzato per ottenere le risposte all'impulso non lineari che caratterizzano il sistema da virtualizzare è un Synchronized Sine Sweep generato con Matlab che possiede le seguenti caratteristiche:

- intervallo di frequenze “spazzate”: $[f_1, f_2] = [20\text{Hz}, 20000\text{Hz}]$ ¹¹;
- durata: $\tilde{T} = 15$ secondi + 5 secondi di silenzio;
- ampiezza: costante, $a(t) = \alpha \leq 1$;

¹¹l'intero spettro delle frequenze udibili da un orecchio umano.

- frequenza di campionamento f_s : 192000 Hz;

ed ha la seguente espressione matematica:

$$s(t) = \alpha \sin \left\{ 2\pi \text{Round} \left(\frac{\tilde{T} f_1}{\ln \left(\frac{f_2}{f_1} \right)} \right) \left[\exp \left(\frac{f_1 t}{\text{Round} \left(\frac{\tilde{T} f_1}{\ln \left(\frac{f_2}{f_1} \right)} \right)} \right) \right] \right\}. \quad (3.17)$$

Quando un segnale $s(t)$ viene introdotto in un dispositivo non lineare, l'output $y(t)$ che viene prodotto contiene anche le distorsioni armoniche generate dal sistema: Le risposte all'impulso delle nuove componenti non lineari possono essere ottenute mediante convoluzione dell'output con il filtro inverso del segnale.

Il risultato della deconvoluzione può essere espresso come

$$y(t) * \tilde{s}(t) = \sum_{n=1}^{\infty} \delta(t + \Delta t_n) \quad (3.18)$$

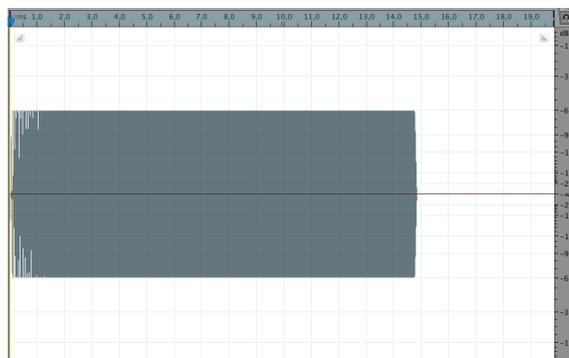
dove $\delta_n(t) := \delta(t + \Delta t_n)$ sono le risposte all'impulso delle nonlinearità di ordine superiore e Δt_n è il ritardo temporale tra la prima risposta all'impulso (lineare) e l' n -sima risposta all'impulso (non lineare).

Il risultato della deconvoluzione consiste dunque in una sequenza di risposte all'impulso non lineari separate nel tempo (Figura 3.14).

Osservazione 3.1. La funzione impulsiva $\delta(t)$ è caratterizzata dall'averne stessa energia su tutte le frequenze, il che si può visualizzare mediante uno spettro delle magnitudine che risulta piatto, nel quale cioè tutte le frequenze del dominio hanno la stessa ampiezza. Dal momento che i segnali considerati in questo lavoro sono di tipo audio, le loro frequenze rimangono nel solo spettro dell'udibile, ovverosia $[20\text{Hz}, 20000\text{Hz}]$: nel seguito si indicherà dunque con $\delta(t)$ una funzione impulsiva ristretta al dominio delle frequenze udibili.

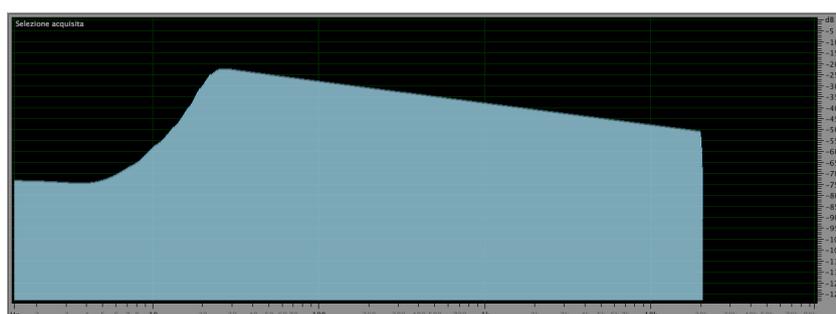
3.2.2 Separazione delle risposte all'impulso

Si può ricavare una relazione tra le risposte all'impulso non lineari e i kernel della serie di Volterra diagonale che si è assunto come modello per il



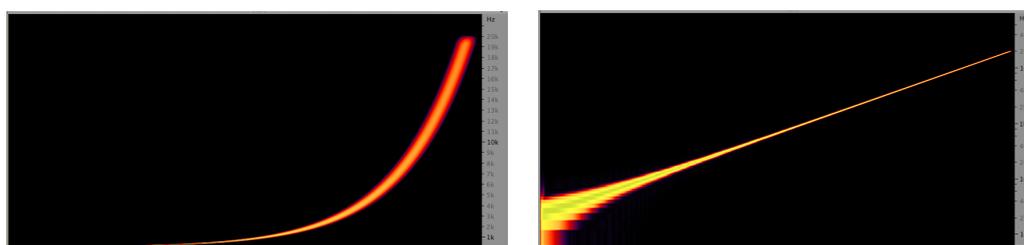
(a)

Figura 3.7: Waveform di un Synchronized Sine Sweep $s(t)$ di 15 secondi + 5 secondi di silenzio con ampiezza $\alpha = -6$ dB.



(a)

Figura 3.8: Spettro di ampiezza di un Synchronized Sine Sweep $s(t)$.



(a) Scala lineare (frequenza di campionamento $f_s = 44000$ Hz)

(b) Scala logaritmica (frequenza di campionamento $f_s = 192000$ Hz)

Figura 3.9: Spettrogrammi di un Synchronized Sine Sweep $s(t)$.

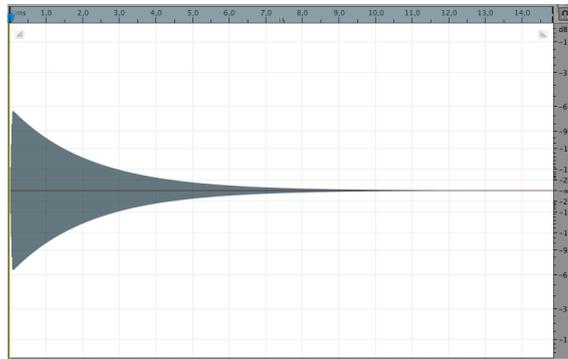


Figura 3.10: Waveform di un Synchronized Sine Sweep inverso $\tilde{s}(t)$ di 15 secondi.

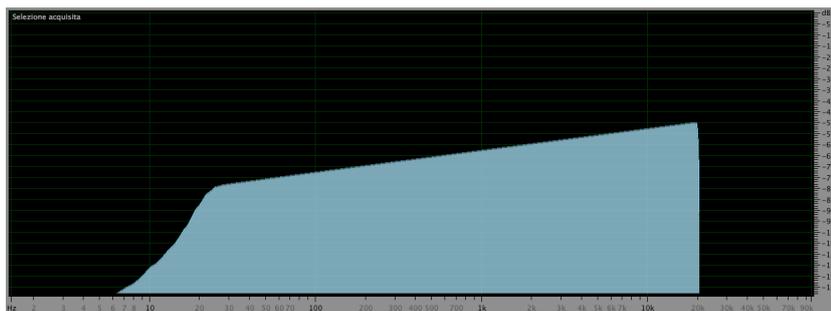
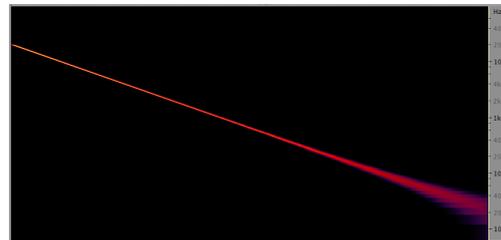


Figura 3.11: Spettro di ampiezza di un Synchronized Sine Sweep inverso $\tilde{s}(t)$.



(a) Scala lineare (frequenza di campionamento $f_s = 44000\text{Hz}$)



(b) Scala logaritmica (frequenza di campionamento $f_s = 192000\text{Hz}$)

Figura 3.12: Spettrogrammi di un Synchronized Sine Sweep inverso $\tilde{s}(t)$.

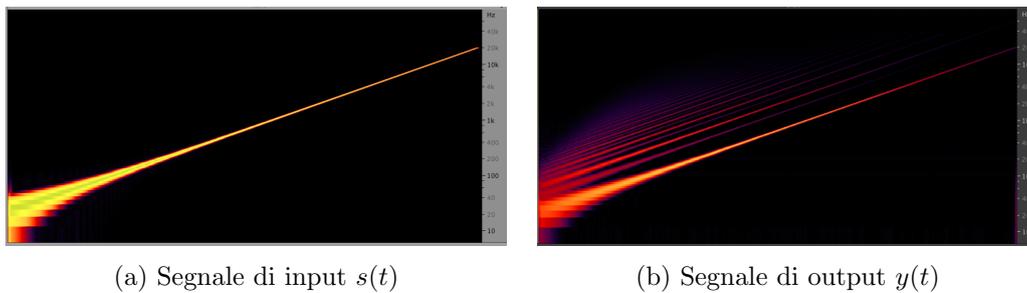
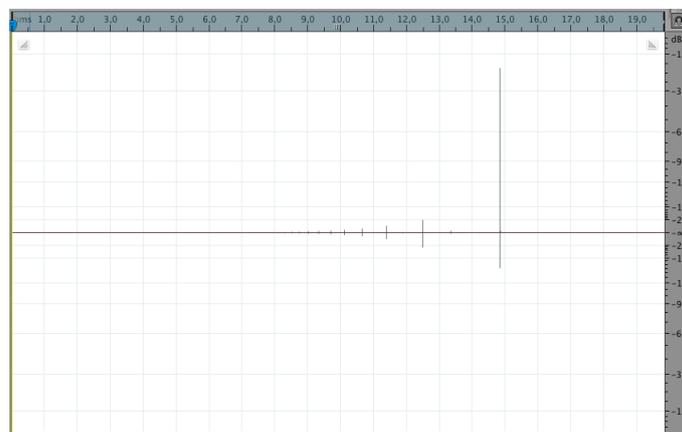


Figura 3.13: Spettrogrammi dell'output del sistema nonlineare.

Figura 3.14: Risultato della deconvoluzione $y(t) * \tilde{s}(t)$.

dispositivo non lineare. In primo luogo, è necessario isolare ciascuna risposta all'impulso non lineare h_n .

In Figura 3.15 viene mostrata l'estensione teorica per $t < 0$ dello spettrogramma dell'output $y(t)$ di un sistema nonlineare sottoposto ad un Synchronized Sine Sweep $s(t)$.

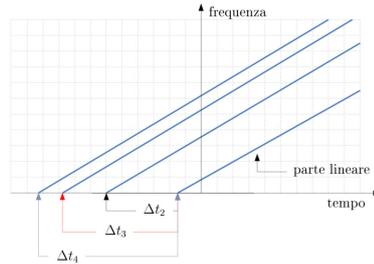


Figura 3.15: Distanze temporali Δt_n .

Il primo impulso a destra in Figura 3.14 rappresenta la risposta lineare del sistema e si manifesta al tempo $t = T$, gli impulsi successivi rappresentano le risposte non lineari di ordine n e si manifestano agli istanti $T - \Delta t_n$.

I gap temporali Δt_n rimangono invariati dopo la deconvoluzione. Ciascuno dei Δt_n rappresenta il divario temporale tra ciascuna frequenza istantanea $f(t)$ ($t > 0$) dello sweep e la sua n -sima frequenza armonica $nf(t)$. Dall'equazione (3.6) si ha che

$$\Delta t_n = \ln(n) \frac{T}{\ln \frac{f_2}{f_1}}$$

Si può dunque ricavare l'istante temporale esatto t_n in cui si manifesta l'impulso non lineare $\delta_n(t)$

$$t_n = T - \Delta t_n = T \left[1 - \frac{\ln(n)}{\ln\left(\frac{f_2}{f_1}\right)} \right] \quad (3.19)$$

ed isolare così le risposte all'impulso non lineari in modo da utilizzarle per ricavare i kernel di Volterra.

3.2.3 Calcolo dei kernel di Volterra

Si riscrive per semplicità l'equazione (3.17) come

$$s(t) = \alpha \sin(\omega(t))$$

e si utilizza tale espressione come “input” dello sviluppo in serie di Volterra diagonale (2.25) troncata all'ordine N , ottenendo così l'approssimazione dell'output $y(t)$ del dispositivo non lineare tramite il modello:

$$y(t) \approx h_0 + h_1(t) * \alpha \sin(\omega(t)) + h_2(t) * [\alpha \sin(\omega(t))]^2 + h_3(t) * [\alpha \sin(\omega(t))]^3 + \dots + h_N(t) * [\alpha \sin(\omega(t))]^N \quad (3.20)$$

L'approssimazione¹² adottata in questo lavoro raggiunge l'ordine $N = 11$; di seguito si sviluppa l'equazione (3.20) con l'ausilio delle identità per gli

¹²già descritta in [31] fino al quinto ordine, viene qui estesa.

elevamenti a potenza delle funzioni trigonometriche:

$$\begin{aligned}
y(t) \approx h_0 & \\
& + \alpha h_1(t) * \sin(\omega(t)) \\
& + \alpha^2 h_2(t) * \left[\frac{1}{2} - \frac{1}{2} \cos(2\omega(t)) \right] \\
& + \alpha^3 h_3(t) * \left[\frac{3}{4} \sin(\omega(t)) - \frac{1}{4} \sin(3\omega(t)) \right] \\
& + \alpha^4 h_4(t) * \left[\frac{3}{8} - \frac{4}{8} \cos(2\omega(t)) + \frac{1}{8} \cos(4\omega(t)) \right] \\
& + \alpha^5 h_5(t) * \left[\frac{10}{16} \sin(\omega(t)) - \frac{5}{16} \sin(3\omega(t)) + \frac{1}{16} \sin(5\omega(t)) \right] \\
& + \alpha^6 h_6(t) * \left[\frac{10}{32} - \frac{15}{32} \cos(2\omega(t)) + \frac{6}{32} \cos(4\omega(t)) - \frac{1}{32} \cos(6\omega(t)) \right] \\
& + \alpha^7 h_7(t) * \left[\frac{35}{64} \sin(\omega(t)) - \frac{21}{64} \sin(3\omega(t)) + \frac{7}{64} \sin(5\omega(t)) - \frac{7}{64} \sin(7\omega(t)) \right] \\
& + \alpha^8 h_8(t) * \left[\frac{35}{128} - \frac{56}{128} \cos(2\omega(t)) + \frac{28}{128} \cos(4\omega(t)) \right. \\
& \quad \left. - \frac{8}{128} \cos(6\omega(t)) + \frac{1}{128} \cos(8\omega(t)) \right] \\
& + \alpha^9 h_9(t) * \left[\frac{126}{256} \sin(\omega(t)) - \frac{84}{256} \sin(3\omega(t)) + \frac{36}{256} \sin(5\omega(t)) \right. \\
& \quad \left. - \frac{9}{256} \sin(7\omega(t)) + \frac{1}{256} \sin(9\omega(t)) \right] \\
& + \alpha^{10} h_{10}(t) * \left[\frac{126}{512} - \frac{210}{512} \cos(2\omega(t)) + \frac{120}{512} \cos(4\omega(t)) \right. \\
& \quad \left. - \frac{45}{512} \cos(6\omega(t)) + \frac{10}{512} \cos(8\omega(t)) - \frac{1}{512} \cos(10\omega(t)) \right] \\
& + \alpha^{11} h_{11}(t) * \left[\frac{462}{1024} \sin(\omega(t)) - \frac{330}{1024} \sin(3\omega(t)) + \frac{165}{1024} \sin(5\omega(t)) \right. \\
& \quad \left. - \frac{55}{1024} \sin(7\omega(t)) + \frac{11}{1024} \sin(9\omega(t)) - \frac{1}{1024} \sin(11\omega(t)) \right]
\end{aligned}$$

$$\begin{aligned}
y(t) \approx h_0 &+ \frac{\alpha^2 h_2(t)}{2} + \frac{3\alpha^4 h_4(t)}{8} + \frac{5\alpha^6 h_6(t)}{16} + \frac{35\alpha^8 h_8(t)}{128} + \frac{63\alpha^{10} h_{10}(t)}{256} \\
&+ \left[\alpha h_1(t) + \frac{231\alpha^{11} h_{11}(t)}{512} + \frac{3\alpha^3 h_3(t)}{4} + \right. \\
&\quad \left. \frac{5\alpha^5 h_5(t)}{8} + \frac{35\alpha^7 h_7(t)}{64} + \frac{63\alpha^9 h_9(t)}{128} \right] * \sin(\omega(t)) \\
&+ \left[-\frac{\alpha^2 h_2(t)}{2} - \frac{\alpha^4 h_4(t)}{2} - \frac{15\alpha^6 h_6(t)}{32} + \right. \\
&\quad \left. -\frac{7\alpha^8 h_8(t)}{16} - \frac{105\alpha^{10} h_{10}(t)}{256} \right] * \cos(2\omega(t)) \\
&+ \left[-\frac{\alpha^3 h_3(t)}{4} - \frac{5\alpha^5 h_5(t)}{16} - \frac{21\alpha^7 h_7(t)}{64} + \right. \\
&\quad \left. -\frac{21\alpha^9 h_9(t)}{64} - \frac{165\alpha^{11} h_{11}(t)}{512} \right] * \sin(3\omega(t)) \\
&+ \left[\frac{\alpha^4 h_4(t)}{8} + \frac{3\alpha^6 h_6(t)}{16} + \frac{7\alpha^8 h_8(t)}{32} + \frac{15\alpha^{10} h_{10}(t)}{64} \right] * \cos(4\omega(t)) \\
&+ \left[\frac{\alpha^5 h_5(t)}{16} + \frac{7\alpha^7 h_7(t)}{64} + \frac{9\alpha^9 h_9(t)}{64} + \frac{165\alpha^{11} h_{11}(t)}{1024} \right] * \sin(5\omega(t)) \\
&+ \left[-\frac{\alpha^6 h_6(t)}{32} - \frac{\alpha^8 h_8(t)}{16} - \frac{45\alpha^{10} h_{10}(t)}{512} \right] * \cos(6\omega(t)) \\
&+ \left[-\frac{\alpha^7 h_7(t)}{64} - \frac{9\alpha^9 h_9(t)}{256} - \frac{55\alpha^{11} h_{11}(t)}{1024} \right] * \sin(7\omega(t)) \\
&+ \left[\frac{\alpha^8 h_8(t)}{128} + \frac{5\alpha^{10} h_{10}(t)}{256} \right] * \cos(8\omega(t)) \\
&+ \left[\frac{\alpha^9 h_9(t)}{256} + \frac{11\alpha^{11} h_{11}(t)}{1024} \right] * \sin(9\omega(t)) \\
&+ \frac{\alpha^{10} h_{10}(t)}{512} * \cos(10\omega(t)) \\
&+ \frac{\alpha^{11} h_{11}(t)}{1024} * \sin(11\omega(t))
\end{aligned}$$

I coefficienti $h_n(t)$ sono i kernel di Volterra e costituiscono le incognite da ricavare.

Convolvere l'approssimazione dell'output $y(t)$ con il filtro inverso $\tilde{s}(t) = \widetilde{\sin(\omega(t))}$ si ottiene:

$$\begin{aligned}
y(t) * \widetilde{\sin}(\omega(t)) \approx & \left[h_0 + \frac{\alpha^2 h_2(t)}{2} + \frac{3\alpha^4 h_4(t)}{8} + \frac{5\alpha^6 h_6(t)}{16} + \right. \\
& \left. \frac{35\alpha^8 h_8(t)}{128} + \frac{63\alpha^{10} h_{10}(t)}{256} \right] * \widetilde{\sin}(\omega(t)) \\
& + \left[\alpha h_1(t) + \frac{231\alpha^{11} h_{11}(t)}{512} + \frac{3\alpha^3 h_3(t)}{4} + \right. \\
& \left. \frac{5\alpha^5 h_5(t)}{8} + \frac{35\alpha^7 h_7(t)}{64} + \frac{63\alpha^9 h_9(t)}{128} \right] * \sin(\omega(t)) * \widetilde{\sin}(\omega(t)) \\
& + \left[-\frac{\alpha^2 h_2(t)}{2} - \frac{\alpha^4 h_4(t)}{2} - \frac{15\alpha^6 h_6(t)}{32} + \right. \\
& \left. -\frac{7\alpha^8 h_8(t)}{16} - \frac{105\alpha^{10} h_{10}(t)}{256} \right] * \cos(2\omega(t)) * \widetilde{\sin}(\omega(t)) \\
& + \left[-\frac{\alpha^3 h_3(t)}{4} - \frac{5\alpha^5 h_5(t)}{16} - \frac{21\alpha^7 h_7(t)}{64} + \right. \\
& \left. -\frac{21\alpha^9 h_9(t)}{64} - \frac{165\alpha^{11} h_{11}(t)}{512} \right] * \sin(3\omega(t)) * \widetilde{\sin}(\omega(t)) \\
& + \left[\frac{\alpha^4 h_4(t)}{8} + \frac{3\alpha^6 h_6(t)}{16} + \frac{7\alpha^8 h_8(t)}{32} + \right. \\
& \left. \frac{15\alpha^{10} h_{10}(t)}{64} \right] * \cos(4\omega(t)) * \widetilde{\sin}(\omega(t)) \\
& + \left[\frac{\alpha^5 h_5(t)}{16} + \frac{7\alpha^7 h_7(t)}{64} + \frac{9\alpha^9 h_9(t)}{64} + \right. \\
& \left. \frac{165\alpha^{11} h_{11}(t)}{1024} \right] * \sin(5\omega(t)) * \widetilde{\sin}(\omega(t)) \\
& + \left[-\frac{\alpha^6 h_6(t)}{32} - \frac{\alpha^8 h_8(t)}{16} - \frac{45\alpha^{10} h_{10}(t)}{512} \right] * \cos(6\omega(t)) * \widetilde{\sin}(\omega(t)) \\
& + \left[-\frac{\alpha^7 h_7(t)}{64} - \frac{9\alpha^9 h_9(t)}{256} - \frac{55\alpha^{11} h_{11}(t)}{1024} \right] * \sin(7\omega(t)) * \widetilde{\sin}(\omega(t)) \\
& + \left[\frac{\alpha^8 h_8(t)}{128} + \frac{5\alpha^{10} h_{10}(t)}{256} \right] * \cos(8\omega(t)) * \widetilde{\sin}(\omega(t)) \\
& + \left[\frac{\alpha^9 h_9(t)}{256} + \frac{11\alpha^{11} h_{11}(t)}{1024} \right] * \sin(9\omega(t)) * \widetilde{\sin}(\omega(t)) \\
& + \frac{\alpha^{10} h_{10}(t)}{512} * \cos(10\omega(t)) * \widetilde{\sin}(\omega(t)) \\
& + \frac{\alpha^{11} h_{11}(t)}{1024} * \sin(11\omega(t)) * \widetilde{\sin}(\omega(t))
\end{aligned}$$

(3.21)

Osservazione 3.2.

Il termine $A := h_0 + \alpha^2 h_2(t) * \frac{1}{2} + \alpha^4 h_4(t) * \frac{3}{8} + \alpha^6 h_6(t) * \frac{5}{16} + \alpha^8 h_8(t) * \frac{35}{128} + \alpha^{10} h_{10}(t) * \frac{63}{256}$ è costante; infatti, dall'ipotesi di stabilità BIBO si ha che tutti gli integrali seguenti sono convergenti:

$$h_0 = \text{costante};$$

$$\alpha^2 h_2(t) * \frac{1}{2} = \int_{\mathbb{R}} \frac{1}{2} \alpha^2 h_2(\tau) d\tau = \frac{1}{2} \alpha^2 \int_{\mathbb{R}} h_2(\tau) d\tau < +\infty$$

$$\alpha^4 h_4(t) * \frac{3}{8} = \int_{\mathbb{R}} \frac{3}{8} \alpha^4 h_4(\tau) d\tau < +\infty$$

$$\alpha^6 h_6(t) * \frac{5}{16} = \int_{\mathbb{R}} \frac{5}{16} \alpha^6 h_6(\tau) d\tau < +\infty$$

$$\alpha^8 h_8(t) * \frac{35}{128} = \int_{\mathbb{R}} \frac{35}{128} \alpha^8 h_8(\tau) d\tau < +\infty$$

$$\alpha^{10} h_{10}(t) * \frac{63}{256} = \int_{\mathbb{R}} \frac{63}{256} \alpha^{10} h_{10}(\tau) d\tau < +\infty.$$

Dal momento che $\widetilde{\sin}(\omega(t))$ è non nullo solo sull'intervallo compatto $[0, T]$, anche $A * \widetilde{\sin}(\omega(t))$ è costante e rappresenta uno scostamento¹³ del segnale rispetto allo 0. Poiché costante, non produce nessun suono, diventando così ininfluenza ai fini di un'emulazione: può essere azzerato mediante un filtro opportuno.

Si trasferiscono ora i calcoli nel dominio delle frequenze, per poter sfruttare le proprietà¹⁴ della trasformata di Fourier. Si utilizzano le seguenti notazioni:

- $X(\omega) = \mathcal{F}\{\sin(\omega(t))\}$ per la trasformata di Fourier del Synchronized Sine Sweep;
- $\widetilde{X}(\omega) = \mathcal{F}\{\widetilde{\sin}(\omega(t))\}$ per la trasformata di Fourier del Synchronized Sine Sweep inverso;

¹³DC offset.

¹⁴Tabella 1.1, Sezione 1.4.6.

- $H_n(\omega) = \mathcal{F}\{h_n(t)\}$ per la trasformata di Fourier dell' n -simo kernel di Volterra.

Osservazione 3.3. Dal momento che¹⁵

$$\sin(\omega(t)) * \widetilde{\sin}(\omega(t)) = \delta(t - t_0)$$

per la proprietà della convoluzione risulta che anche

$$\mathcal{F}^{-1}\{X(\omega)\widetilde{X}(\omega)\} = \delta(t - t_0).$$

Per la proprietà di scalatura vale che, se $X(\omega) = \mathcal{F}\{g(\omega(t))\}$ con $a \in \mathbb{R}$, allora

$$\frac{1}{|a|}X\left(\frac{\omega}{a}\right) = \mathcal{F}\{g(a\omega(t))\}.$$

Per la proprietà di traslazione nel tempo ed essendo valida la relazione $\sin(\omega(t) + \frac{\pi}{2}) = \cos(\omega t)$ si ottiene che

$$\mathcal{F}\{\cos(\omega(t))\} = \mathcal{F}\{\sin(\omega(t) + \frac{\pi}{2})\} = e^{-i\omega\frac{\pi}{2}}\mathcal{F}\{\sin(\omega(t))\} \equiv e^{-i\omega\frac{\pi}{2}}X(\omega)$$

dove il fattore moltiplicativo $J(\omega) = e^{-i\omega\frac{\pi}{2}}$ modifica di $\frac{\pi}{2}$ la fase di ciascuna componente frequenziale del sine sweep.

Osservazione 3.4. Il segnale $s(t)$ è un segnale reale e quindi la sua trasformata di Fourier gode della proprietà di simmetria Hermitiana¹⁶: nei calcoli verrà dunque considerata solo la parte positiva dello spettro, potendo ricavare quella negativa per simmetria.

Si sviluppa dunque la (3.21) nel dominio delle frequenze, dopo aver eliminato

¹⁵per definizione di filtro inverso, Sezione 3.1.2

¹⁶ovverosia, lo spettro della magnitudine $|X(\omega)|$ è una funzione pari, mentre lo spettro della fase $\angle X(\omega)$ è una funzione dispari (cfr. Equazione (1.19), Sezione 1.4.6).

la parte costante:

$$\begin{aligned}
\mathcal{F}\{y(t) * \widetilde{\sin}(\omega(t))\} \approx & \left[\alpha H_1(\omega) + \frac{231\alpha^{11}}{512} H_{11}(\omega) + \frac{3\alpha^3}{4} H_3(\omega) + \right. \\
& \left. \frac{5\alpha^5}{8} H_5(\omega) + \frac{35\alpha^7}{64} H_7(\omega) + \frac{63\alpha^9}{128} H_9(\omega) \right] X(\omega) \widetilde{X}(\omega) \\
+ & \left[-\frac{\alpha^2}{2} H_2(\omega) - \frac{\alpha^4}{2} H_4(\omega) - \frac{15\alpha^6}{32} H_6(\omega) + \right. \\
& \left. -\frac{7\alpha^8}{16} H_8(\omega) - \frac{105\alpha^{10}}{256} H_{10}(\omega) \right] J(\omega) \frac{1}{|2|} X\left(\frac{\omega}{2}\right) \widetilde{X}(\omega) \\
+ & \left[-\frac{\alpha^3}{4} H_3(\omega) - \frac{5\alpha^5}{16} H_5(\omega) - \frac{21\alpha^7}{64} H_7(\omega) + \right. \\
& \left. -\frac{21\alpha^9}{64} H_9(\omega) - \frac{165\alpha^{11}}{512} H_{11}(\omega) \right] \frac{1}{|3|} X\left(\frac{\omega}{3}\right) \widetilde{X}(\omega) \\
+ & \left[\frac{\alpha^4}{8} H_4(\omega) + \frac{3\alpha^6}{16} H_6(\omega) + \frac{7\alpha^8}{32} H_8(\omega) + \right. \\
& \left. \frac{15\alpha^{10}}{64} H_{10}(\omega) \right] J(\omega) \frac{1}{|4|} X\left(\frac{\omega}{4}\right) \widetilde{X}(\omega) \\
+ & \left[\frac{\alpha^5}{16} H_5(\omega) + \frac{7\alpha^7}{64} H_7(\omega) + \frac{9\alpha^9}{64} H_9(\omega) + \right. \\
& \left. \frac{165\alpha^{11}}{1024} H_{11}(\omega) \right] \frac{1}{|5|} X\left(\frac{\omega}{5}\right) \widetilde{X}(\omega) \\
+ & \left[-\frac{\alpha^6}{32} H_6(\omega) - \frac{\alpha^8}{16} H_8(\omega) \right. \\
& \left. -\frac{45\alpha^{10}}{512} H_{10}(\omega) \right] J(\omega) \frac{1}{|6|} X\left(\frac{\omega}{6}\right) \widetilde{X}(\omega) \\
+ & \left[-\frac{\alpha^7}{64} H_7(\omega) - \frac{9\alpha^9}{256} H_9(\omega) \right. \\
& \left. -\frac{55\alpha^{11}}{1024} H_{11}(\omega) \right] \frac{1}{|7|} X\left(\frac{\omega}{7}\right) \widetilde{X}(\omega) \\
+ & \left[\frac{\alpha^8}{128} H_8(\omega) + \frac{5\alpha^{10}}{256} H_{10}(\omega) \right] J(\omega) \frac{1}{|8|} X\left(\frac{\omega}{8}\right) \widetilde{X}(\omega) \\
+ & \left[\frac{\alpha^9}{256} H_9(\omega) + \frac{11\alpha^{11}}{1024} H_{11}(\omega) \right] \frac{1}{|9|} X\left(\frac{\omega}{9}\right) \widetilde{X}(\omega) \\
+ & \frac{\alpha^{10}}{512} H_{10}(\omega) J(\omega) \frac{1}{|10|} X\left(\frac{\omega}{10}\right) \widetilde{X}(\omega) \\
+ & \frac{\alpha^{11}}{1024} H_{11}(\omega) \frac{1}{|11|} X\left(\frac{\omega}{11}\right) \widetilde{X}(\omega)
\end{aligned}$$

(3.22)

Poiché

$$\mathcal{F}^{-1}\{X(\omega)\tilde{X}(\omega)\} \approx \delta(t - t_1)$$

$$\mathcal{F}^{-1}\left\{\frac{1}{|2|}X\left(\frac{\omega}{2}\right)\tilde{X}(\omega)\right\} \approx \delta(t - t_2)$$

$$\mathcal{F}^{-1}\left\{\frac{1}{|3|}X\left(\frac{\omega}{3}\right)\tilde{X}(\omega)\right\} \approx \delta(t - t_3)$$

$$\mathcal{F}^{-1}\left\{\frac{1}{|4|}X\left(\frac{\omega}{4}\right)\tilde{X}(\omega)\right\} \approx \delta(t - t_4)$$

$$\mathcal{F}^{-1}\left\{\frac{1}{|5|}X\left(\frac{\omega}{5}\right)\tilde{X}(\omega)\right\} \approx \delta(t - t_5)$$

$$\mathcal{F}^{-1}\left\{\frac{1}{|6|}X\left(\frac{\omega}{6}\right)\tilde{X}(\omega)\right\} \approx \delta(t - t_6)$$

$$\mathcal{F}^{-1}\left\{\frac{1}{|7|}X\left(\frac{\omega}{7}\right)\tilde{X}(\omega)\right\} \approx \delta(t - t_7)$$

$$\mathcal{F}^{-1}\left\{\frac{1}{|8|}X\left(\frac{\omega}{8}\right)\tilde{X}(\omega)\right\} \approx \delta(t - t_8)$$

$$\mathcal{F}^{-1}\left\{\frac{1}{|9|}X\left(\frac{\omega}{9}\right)\tilde{X}(\omega)\right\} \approx \delta(t - t_9)$$

$$\mathcal{F}^{-1}\left\{\frac{1}{|10|}X\left(\frac{\omega}{10}\right)\tilde{X}(\omega)\right\} \approx \delta(t - t_{10})$$

$$\mathcal{F}^{-1}\left\{\frac{1}{|11|}X\left(\frac{\omega}{11}\right)\tilde{X}(\omega)\right\} \approx \delta(t - t_{11})$$

risulta che la trasformata inversa della (3.22) ha l'espressione seguente:

$$\begin{aligned}
y(t) * \widetilde{\sin}(\omega(t)) \approx & \mathcal{F}^{-1} \left\{ \alpha H_1(\omega) + \frac{3\alpha^3}{4} H_3(\omega) + \frac{5\alpha^5}{8} H_5(\omega) + \right. \\
& \left. \frac{35\alpha^7}{64} H_7(\omega) + \frac{63\alpha^9}{128} H_9(\omega) + \frac{231\alpha^{11}}{512} H_{11}(\omega) \right\} * \delta(t - t_1) \\
+ & \mathcal{F}^{-1} \left\{ j \left[-\frac{\alpha^2}{2} H_2(\omega) - \frac{\alpha^4}{2} H_4(\omega) - \frac{15\alpha^6}{32} H_6(\omega) + \right. \right. \\
& \left. \left. - \frac{7\alpha^8}{16} H_8(\omega) - \frac{105\alpha^{10}}{256} H_{10}(\omega) \right] \right\} * \delta(t - t_2) \\
+ & \mathcal{F}^{-1} \left\{ -\frac{\alpha^3}{4} H_3(\omega) - \frac{5\alpha^5}{16} H_5(\omega) - \frac{21\alpha^7}{64} H_7(\omega) + \right. \\
& \left. - \frac{21\alpha^9}{64} H_9(\omega) - \frac{165\alpha^{11}}{512} H_{11}(\omega) \right\} * \delta(t - t_3) \\
+ & \mathcal{F}^{-1} \left\{ j \left[\frac{\alpha^4}{8} H_4(\omega) + \frac{3\alpha^6}{16} H_6(\omega) + \frac{7\alpha^8}{32} H_8(\omega) + \right. \right. \\
& \left. \left. \frac{15\alpha^{10}}{64} H_{10}(\omega) \right] \right\} * \delta(t - t_4) \\
+ & \mathcal{F}^{-1} \left\{ \frac{\alpha^5}{16} H_5(\omega) + \frac{7\alpha^7}{64} H_7(\omega) + \frac{9\alpha^9}{64} H_9(\omega) + \right. \\
& \left. \frac{165\alpha^{11}}{1024} H_{11}(\omega) \right\} * \delta(t - t_5) \\
+ & \mathcal{F}^{-1} \left\{ j \left[-\frac{\alpha^6}{32} H_6(\omega) - \frac{\alpha^8}{16} H_8(\omega) - \frac{45\alpha^{10}}{512} H_{10}(\omega) \right] \right\} * \delta(t - t_6) \\
+ & \mathcal{F}^{-1} \left\{ -\frac{\alpha^7}{64} H_7(\omega) - \frac{9\alpha^9}{256} H_9(\omega) - \frac{55\alpha^{11}}{1024} H_{11}(\omega) \right\} * \delta(t - t_7) \\
+ & \mathcal{F}^{-1} \left\{ j \left[\frac{\alpha^8}{128} H_8(\omega) + \frac{5\alpha^{10}}{256} H_{10}(\omega) \right] \right\} * \delta(t - t_8) \\
+ & \mathcal{F}^{-1} \left\{ \frac{\alpha^9}{256} H_9(\omega) + \frac{11\alpha^{11}}{1024} H_{11}(\omega) \right\} * \delta(t - t_9) \\
+ & \mathcal{F}^{-1} \left\{ j \frac{\alpha^{10}}{512} H_{10}(\omega) \right\} * \delta(t - t_{10}) \\
+ & \mathcal{F}^{-1} \left\{ \frac{\alpha^{11}}{1024} H_{11}(\omega) \right\} * \delta(t - t_{11})
\end{aligned} \tag{3.23}$$

Gli addendi della (3.23) trovano corrispondenza nei primi undici (da destra a sinistra) impulsi della deconvoluzione (3.18) visualizzati in Figura 3.14, dei quali conosce il punto iniziale t_n grazie alla (3.19). Prima di ritornare nel dominio delle frequenze si introduce la notazione $R_k(\omega) = \mathcal{F}\{\delta(t - t_k)\}$

($k = 1, \dots, 11$) per le trasformate di Fourier delle risposte all'impulso non lineari.

Ciascuno dei termini $R_k(\omega)$ corrisponde ordinatamente ad uno dei termini dell'Equazione (3.23); si ottiene dunque il sistema:

$$\left\{ \begin{array}{l} K_1(\omega) = \alpha H_1(\omega) + \frac{3\alpha^3}{4} H_3(\omega) + \frac{5\alpha^5}{8} H_5(\omega) + \frac{35\alpha^7}{64} H_7(\omega) + \\ \quad \frac{63\alpha^9}{128} H_9(\omega) + \frac{231\alpha^{11}}{512} H_{11}(\omega) \\ K_2(\omega) = j \left[-\frac{\alpha^2}{2} H_2(\omega) - \frac{\alpha^4}{2} H_4(\omega) - \frac{15\alpha^6}{32} H_6(\omega) + \right. \\ \quad \left. -\frac{7\alpha^8}{16} H_8(\omega) - \frac{105\alpha^{10}}{256} H_{10}(\omega) \right] \\ K_3(\omega) = -\frac{\alpha^3}{4} H_3(\omega) - \frac{5\alpha^5}{16} H_5(\omega) - \frac{21\alpha^7}{64} H_7(\omega) + \\ \quad -\frac{21\alpha^9}{64} H_9(\omega) - \frac{165\alpha^{11}}{512} H_{11}(\omega) \\ K_4(\omega) = j \left[\frac{\alpha^4}{8} H_4(\omega) + \frac{3\alpha^6}{16} H_6(\omega) + \frac{7\alpha^8}{32} H_8(\omega) + \frac{15\alpha^{10}}{64} H_{10}(\omega) \right] \\ K_5(\omega) = \frac{\alpha^5}{16} H_5(\omega) + \frac{7\alpha^7}{64} H_7(\omega) + \frac{9\alpha^9}{64} H_9(\omega) + \frac{165\alpha^{11}}{1024} H_{11}(\omega) \\ K_6(\omega) = j \left[-\frac{\alpha^6}{32} H_6(\omega) - \frac{\alpha^8}{16} H_8(\omega) - \frac{45\alpha^{10}}{512} H_{10}(\omega) \right] \\ K_7(\omega) = -\frac{\alpha^7}{64} H_7(\omega) - \frac{9\alpha^9}{256} H_9(\omega) - \frac{55\alpha^{11}}{1024} H_{11}(\omega) \\ K_8(\omega) = j \left[\frac{\alpha^8}{128} H_8(\omega) + \frac{5\alpha^{10}}{256} H_{10}(\omega) \right] \\ K_9(\omega) = \frac{\alpha^9}{256} H_9(\omega) + \frac{11\alpha^{11}}{1024} H_{11}(\omega) \\ K_{10}(\omega) = j \frac{\alpha^{10}}{512} H_{10}(\omega) \\ K_{11}(\omega) = \frac{\alpha^{11}}{1024} H_{11}(\omega) . \end{array} \right.$$

che può infine essere risolto nelle incognite $H_k(\omega)$:

$$\left\{ \begin{array}{l} H_1(\omega) = \frac{K_1(\omega) + 3K_3(\omega) + 5K_5(\omega) + 7K_7(\omega) + 9K_9(\omega) + 11K_{11}(\omega)}{\alpha} \\ H_2(\omega) = j \frac{2K_2(\omega) + 8K_4(\omega) + 18K_6(\omega) + 32K_8(\omega) + 50K_{10}(\omega)}{\alpha^2} \\ H_3(\omega) = - \frac{4K_3(\omega) + 20K_5(\omega) + 56K_7(\omega) + 120K_9(\omega) + 220K_{11}(\omega)}{\alpha^3} \\ H_4(\omega) = -j \frac{8K_4(\omega) + 48K_6(\omega) + 160K_8(\omega) + 400K_{10}(\omega)}{\alpha^4} \\ H_5(\omega) = \frac{16K_5(\omega) + 112K_7(\omega) + 189K_9(\omega) + 1232K_{11}(\omega)}{\alpha^5} \\ H_6(\omega) = j \frac{32K_6(\omega) + 256K_8(\omega) + 1120K_{10}(\omega)}{\alpha^6} \\ H_7(\omega) = \frac{-64K_7(\omega) - 576K_9(\omega) - 2816K_{11}(\omega)}{\alpha^7} \\ H_8(\omega) = -j \frac{128K_8(\omega) + 1280K_{10}(\omega)}{\alpha^8} \\ H_9(\omega) = \frac{256K_9(\omega) + 2816K_{11}(\omega)}{\alpha^9} \\ H_{10}(\omega) = j \frac{512K_{10}(\omega)}{\alpha^{10}} \\ H_{11}(\omega) = - \frac{1024K_{11}(\omega)}{\alpha^{11}} \end{array} \right. \quad (3.24)$$

Ricordando che $H_k(\omega) = \mathcal{F}\{h_k(t)\}$ ($k = 1, \dots, 11$) e antitrasformando le (3.24) si ottengono infine le espressioni dei kernel di Volterra $h_k(t)$ ($k = 1, \dots, 11$) desiderati.

Un procedimento analogo può essere svolto nel caso discreto per ottenere il kernel $h_k[n]$ ($k = 1, \dots, 11$) da utilizzare per implementare il modello di Volterra diagonale (2.28): i kernel $h_k[n]$ rappresentano la forma “virtuale” del dispositivo non lineare.

3.2.4 Emulazione del dispositivo non lineare

L'emulazione $y_{virt}(t)$ delle distorsioni prodotte dal dispositivo non lineare sullo sweep $s(t)$ avviene tramite il modello di Volterra diagonale esteso

all'undicesimo ordine:

$$\begin{aligned}
 y_{virt}(t) = & h_1(t) * s(t) + h_2(t) * [s(t)]^2 + h_3(t) * [s(t)]^3 + h_4(t) * [s(t)]^4 + \\
 & h_5(t) * [s(t)]^5 + h_6(t) * [s(t)]^6 + h_7(t) * [s(t)]^7 + h_8(t) * [s(t)]^8 + \\
 & h_9(t) * [s(t)]^9 + h_{10}(t) * [s(t)]^{10} + h_{11}(t) * [s(t)]^{11}
 \end{aligned}
 \tag{3.25}$$

Ciascuno degli addendi dell'Equazione (3.25) viene ottenuto elevando ad opportuna potenza il segnale $s(t)$, filtrando il risultato con un filtro passa-basso settato sul tasso di Nyquist per evitare l'aliasing ed infine convolvendolo con il corrispettivo kernel. Per una maggiore scorrevolezza nella lettura, tale procedimento di emulazione verrà unitariamente chiamato *Volterra-convoluzione*.

NOTA BENE. Ove non diversamente specificato, tutti i segnali visualizzati avranno ampiezza normalizzata per consentire un corretto confronto fra le caratteristiche.

Nelle Figure 3.16 e 3.17 vengono messe a confronto diverse caratterizzazioni dell'output $y(t)$ distorto dal dispositivo non lineare e del segnale $y_{virt}(t)$ creato virtualmente tramite il modello di Volterra: in Figura 3.16 vengono visualizzati i segnali e in Figura 3.17 i relativi spettrogrammi.

Nella Figura 3.18 viene mostrato lo spettro delle ampiezze delle frequenze presenti nei segnali $y(t)$ ed $y_{virt}(t)$ al secondo 09:00; si può osservare come lo spettro della parte lineare dello sweep (primo "picco" a sinistra) e dei successivi 10 armonici coincida per ambo i segnali.

Si riportano di seguito alcuni esempi di confronto tra le caratteristiche del segnale distorto misurato e della rispettiva virtualizzazione.

Nota. Le componenti frequenziali presenti nello spettro del segnale $y(t)$ al di sotto dei -100 dB sono dovute alla normalizzazione (e quindi all'aumento di volume) del rumore di fondo.

Virtualizzazione di un brano musicale

Scopo finale del progetto è quello di arrivare all'emulazione del dispositivo non lineare nel modo più fedele possibile. Il modello di Volterra diagonale

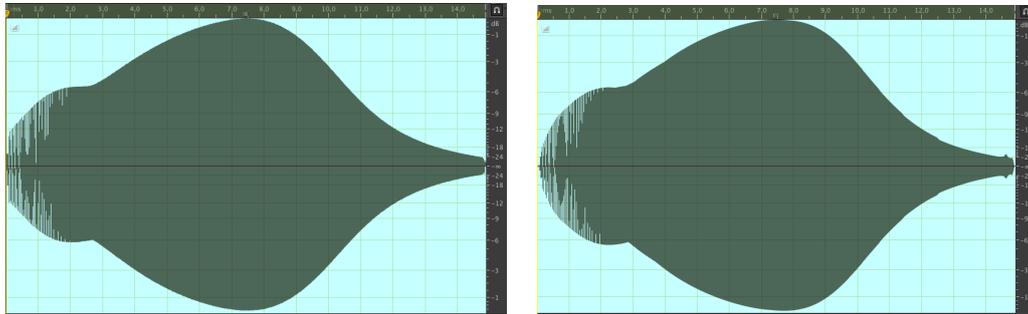
(a) Output $y(t)$ distorto registrato.(b) Virtualizzazione $y_{virt}(t)$ dell'output registrato.

Figura 3.16: Forme d'onda (normalizzate) dei risultati.

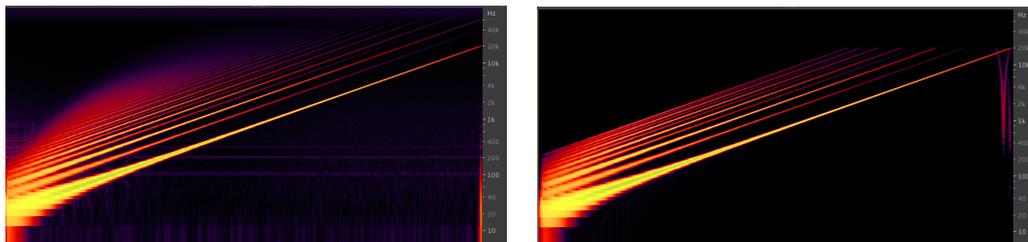
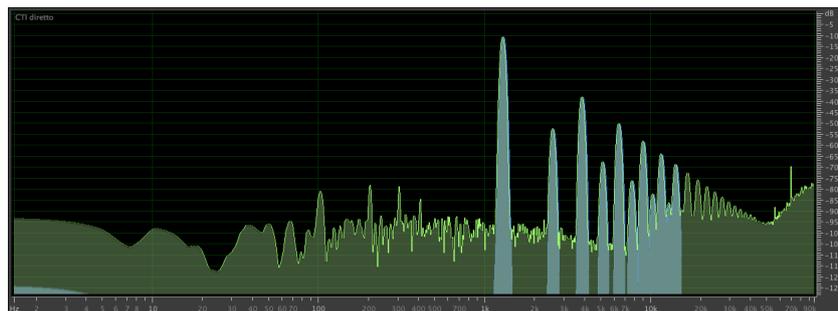
(a) Output $y(t)$ distorto registrato.(b) Virtualizzazione $y_{virt}(t)$ dell'output registrato.

Figura 3.17: Spettrogrammi dei risultati.

Figura 3.18: Spettro delle frequenze dell'output registrato $y(t)$ (in giallo) a confronto con la sua virtualizzazione $y_{virt}(t)$ (in blu) a 09:00 sec.

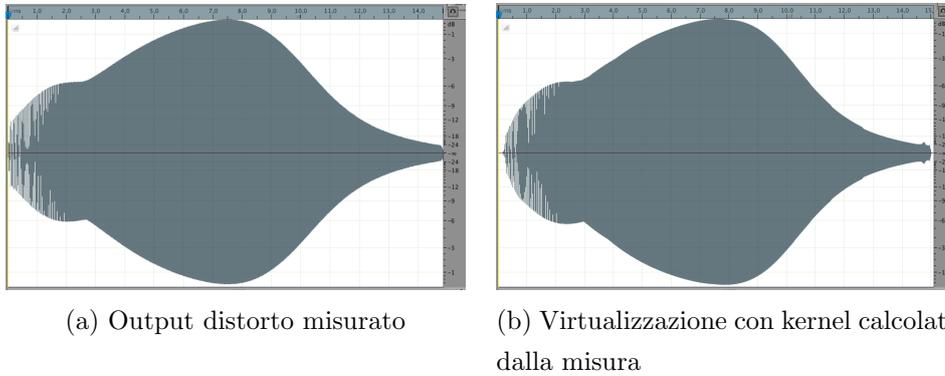


Figura 3.19: Confronto waveform misura - virtualizzazione per lo sweep a -18 dB.

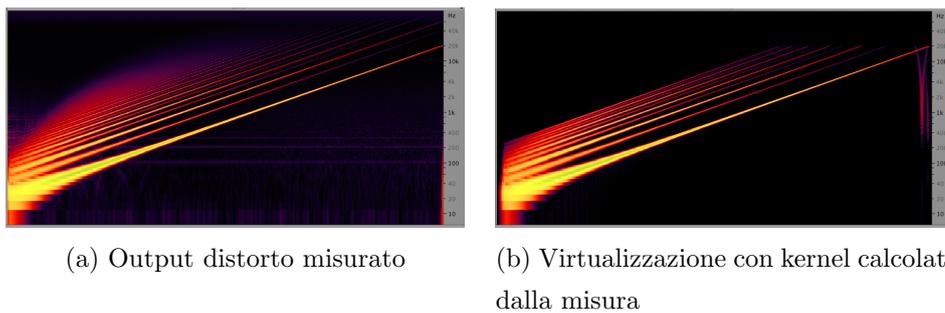


Figura 3.20: Confronto spettrogrammi misura - virtualizzazione per lo sweep a -18 dB.

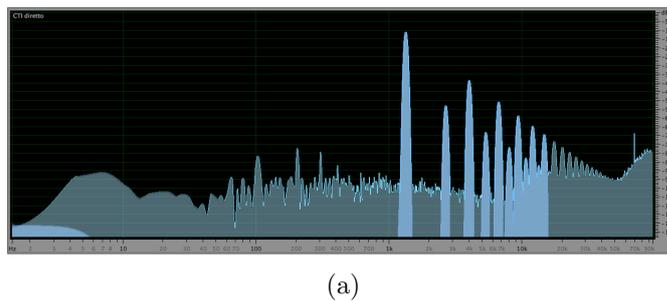


Figura 3.21: Spettro delle frequenze dell'output registrato $y(t)$ (in blu) a confronto con la sua virtualizzazione $y_{virt}(t)$ (in azzurro) a 09:00 sec.

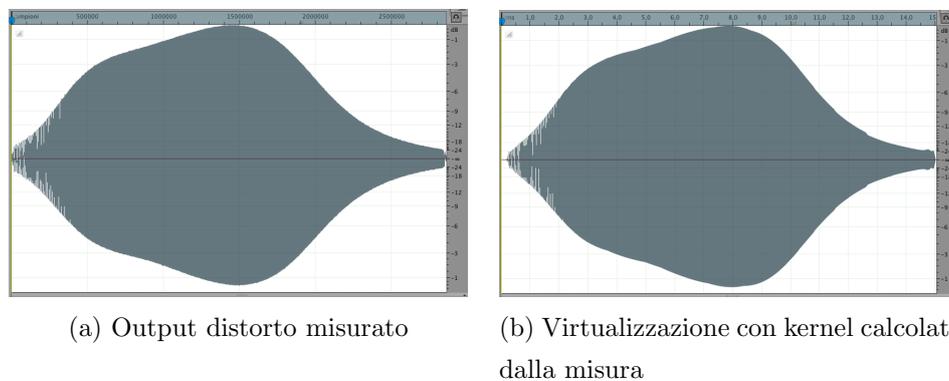


Figura 3.22: Confronto waveform misura - virtualizzazione per lo sweep a -33 dB.

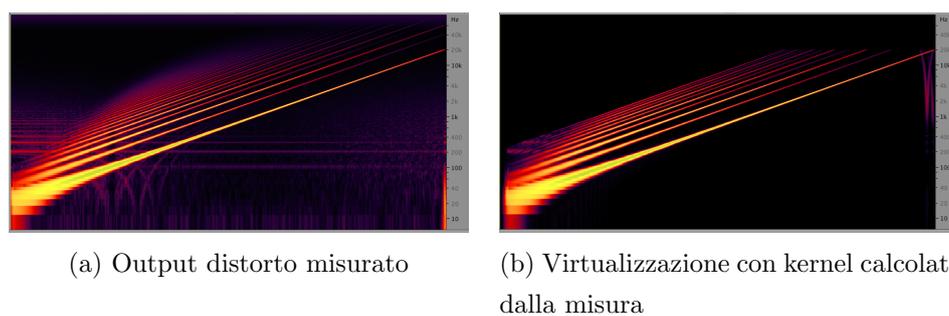
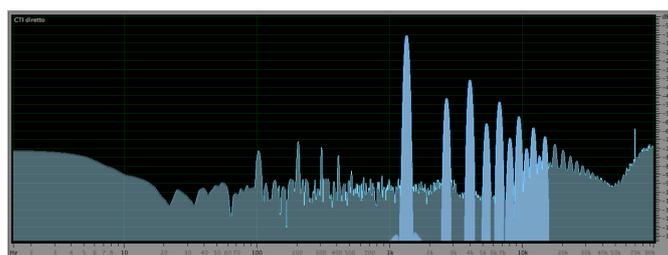


Figura 3.23: Confronto spettrogrammi misura - virtualizzazione per lo sweep a -33 dB.



(a)

Figura 3.24: Spettro delle frequenze dell'output registrato $y(t)$ (in blu) a confronto con la sua virtualizzazione $y_{virt}(t)$ (in azzurro) a 09:00 sec.

che si è utilizzato è valido per segnali che abbiano, in ogni istante, un'unica componente frequenziale (ipotesi verificata dallo sweep). Un segnale audio "reale" non soddisfa questa caratteristica, poiché i cosiddetti "toni puri" (ovverosia, senza distorsioni armoniche intrinseche) possono essere generati solamente al calcolatore e non da altri strumenti fisici¹⁷. Nella prospettiva futura di ricerca di un'estensione del modello di Volterra a segnali multi-frequenziali, si utilizza per il momento la forma virtuale del dispositivo fornita dai kernel $h_k[n]$ anche per riprodurre le distorsioni generate su un segnale audio reale. La procedura di virtualizzazione di un brano musicale verrà dettagliata nel prossimo capitolo.

¹⁷Uno stesso tono puro, nel momento in cui venga riprodotto tramite un qualsivoglia dispositivo, risente degli effetti prodotti dal suddetto dispositivo e quindi manifesta a sua volta dei suoni armonici.

Capitolo 4

Sperimentazione effettuata

Le modalità di virtualizzazione di dispositivi non lineari mediante modello di Volterra diagonale implementate finora¹ prevedono l'emulazione “off-line” delle non linearità : le distorsioni vengono generate in post-processing su un segnale di cui si sia verificato precedentemente il volume e generati i kernel di conseguenza. Nella prospettiva futura di emulazione delle non linearità *in tempo reale* di un segnale (il quale comprende, quindi, variazioni eventualmente considerevoli nella dinamica del suono lungo tutta la durata), viene qui presentato un primo tentativo di selezione “veloce” dei kernel in dipendenza dal volume del segnale dato in input.

In questo capitolo verrà dunque esposta la parte di sperimentazione progettata e messa in atto durante il lavoro di tesi.

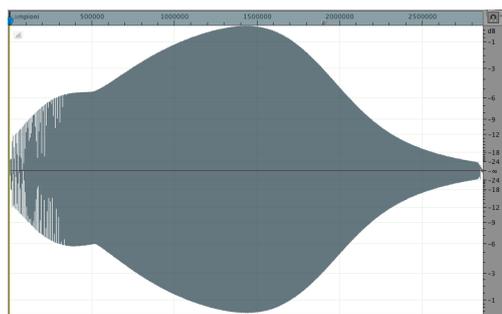
Notazioni.

- Con il termine *n-simo kernel di ampiezza α* si intenderà “il kernel di ordine n costruito a partire dalla Volterra-deconvoluzione di un Synchronized Sine Sweep di ampiezza α ”.
- Analogamente, con il termine *set di kernel di ampiezza α* si intenderà “l'insieme degli 11 kernel costruiti a partire dalla Volterra-deconvoluzione di un Synchronized Sine Sweep di ampiezza α ”.

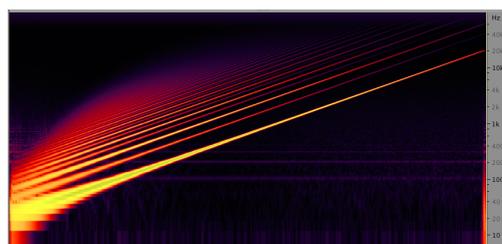
¹si fa riferimento al software realizzato dal Prof. Tronchin e dall'Ing. Venturi, per il quale l'Università di Bologna ha ottenuto un brevetto nel 2012, [39].

- Con *sweep* si indicherà il segnale di test Synchronized Sine Sweep descritto nella Sezione 3.1.

Dalle formule (3.24) si può vedere come l'espressione dei kernel del modello di Volterra diagonale dipenda intrinsecamente dal volume α dello sweep in ingresso: un set di kernel di ampiezza α_1 produrrà dunque i risultati migliori nel caso in cui si vogliano ricreare gli effetti di distorsione su uno sweep a volume α_1 , mentre risulterà inadeguato nel caso in cui si voglia mettere in atto il medesimo procedimento su uno sweep a volume $\alpha_2 \neq \alpha_1$. A titolo di esempio, si riporta in Figura 4.1 l'output distorto $y(t)$ di uno sweep a volume -18 dB ed in Figura 4.2 le virtualizzazioni effettuate con il set di kernel a -18 dB e -35 dB.



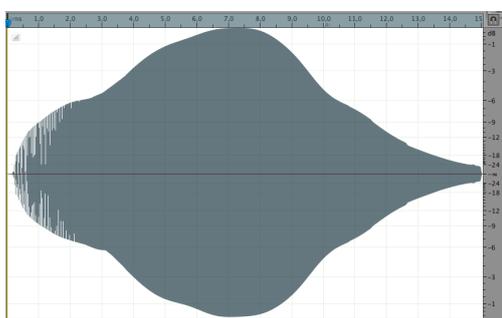
(a) Forma d'onda



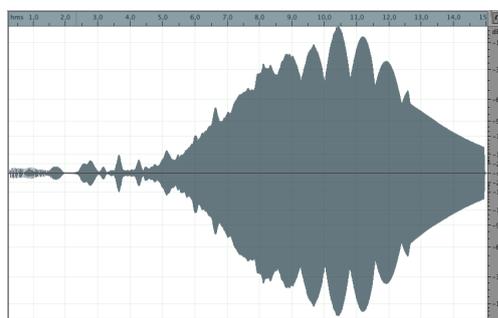
(b) Spettrogramma

Figura 4.1: Output distorto $y(t)$ dello sweep $s(t)$ a volume -18 dB.

Nell'ottica di una virtualizzazione “real-time” delle distorsioni, tale dipendenza presupporrebbe di dover avere a disposizione un set di kernel per ogni volume α raggiungibile dal segnale che si vuole distorcere: scopo di questo lavoro di tesi è stato quello di tentare di “snellire” tale procedimento sfruttando



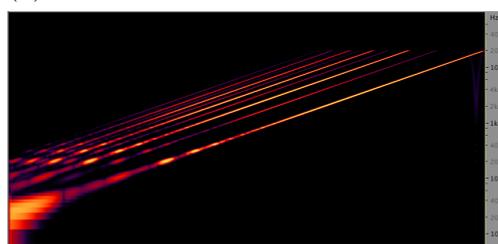
(a) Forma d'onda con set di kernel a -18 dB



(b) Forma d'onda con set di kernel a -35 dB



(c) Spettrogramma con set di kernel a -18 dB



(d) Spettrogramma con set di kernel a -35 dB

Figura 4.2: Virtualizzazioni dello sweep a volume -18 dB con differenti set di kernel.

le caratteristiche dei kernel, ovvero sia trovando mediante interpolazione i set di kernel di uno sweep ad un arbitrario volume α del quale non si debbano necessariamente avere le misure.

Partendo da uno sweep a volume α , la relazione che sussiste tra l'ampiezza del kernel $h_n(t)$ di ordine n e tale volume è di proporzionalità inversa:

$$\text{ampiezza}(h_n(t)) \propto \frac{1}{\alpha^n} .$$

Tale fatto si ripercuote anche sulle magnitudes nello spettro delle ampiezze

$$|H_n(\omega)| \propto \frac{1}{\alpha^n} ,$$

come si può vedere in Figura 4.3 dove sono visualizzate le magnitudes dei kernel $h_1(t)$, $h_2(t)$, $h_3(t)$, $h_4(t)$, $h_8(t)$ ed $h_9(t)$ per 31 sweep i cui volumi variano linearmente da -15 dB a -45 dB.

L'idea che si è voluta realizzare prevede dunque i seguenti passaggi:

1. registrare gli output di una sequenza di sweep a diversi volumi $\alpha_1, \dots, \alpha_k$: in particolare, $k = 31$ e l'insieme dei valori dei volumi è $\{-15 \text{ dB}, -16 \text{ dB}, \dots, -44 \text{ dB}, -45 \text{ dB}\}^2$;
2. creare i set di kernel per ciascuno dei volumi $\alpha_1, \dots, \alpha_k$;
3. mettere in atto un procedimento di interpolazione sulle magnitudes dei kernel, nel modo seguente:
 - (a) si è fissata una frequenza \bar{f} nello spettro delle magnitudes;
 - (b) per ogni ordine di kernel n si è costruita con Matlab una funzione $p_n(\alpha)$ [Figura 4.4 (b), (d), (e)] di terzo ordine che interpola a tratti di quattro nodi i valori delle k magnitudes alla frequenza fissata \bar{f} di ciascuno dei kernel di ordine n di ampiezza $\alpha = -15\text{dB}, \dots, -45\text{dB}$;

²Il valore massimo del volume che può essere raggiunto da una chitarra elettrica non supera i -15 dB.

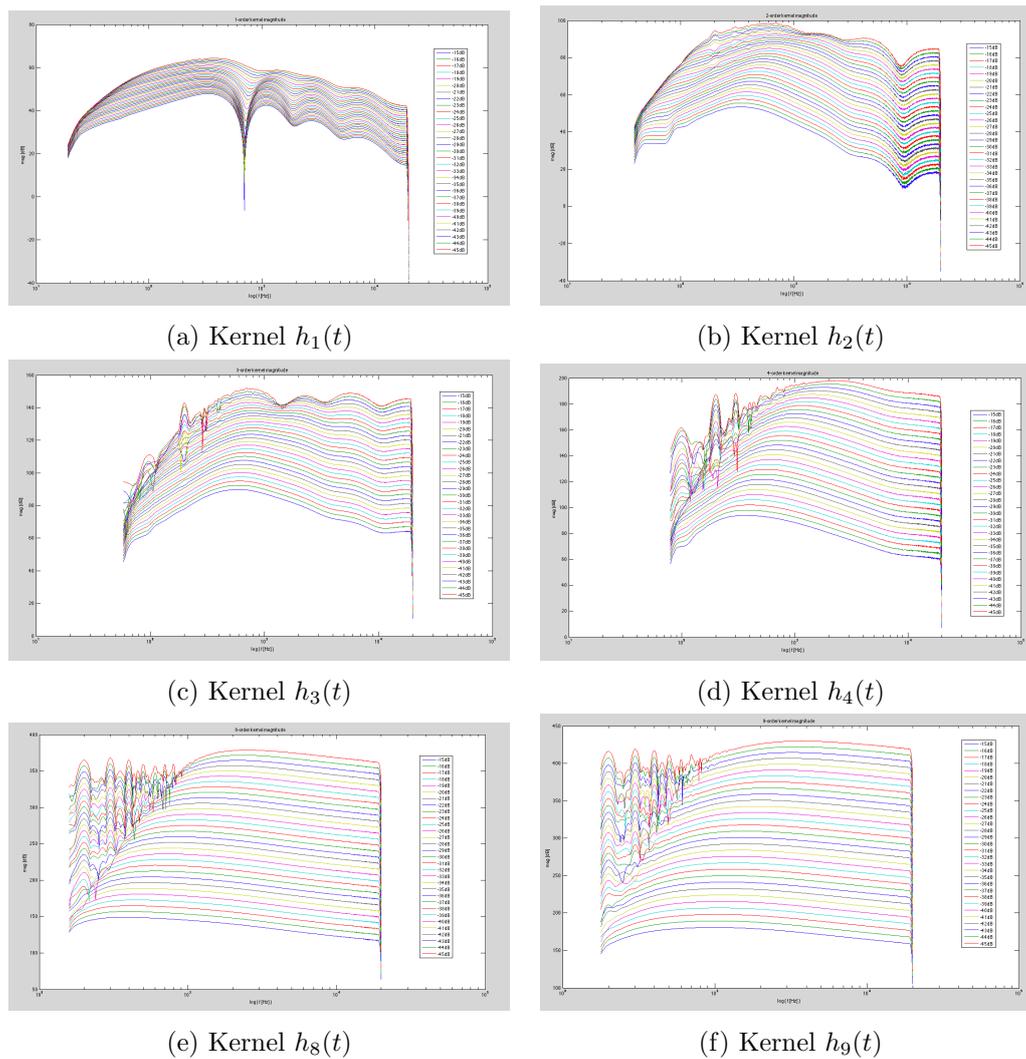


Figura 4.3: Magnitudes $|H_n(\omega)|$ a differenti volumi α dello sweep.

4. individuare un set di kernel “di riferimento” ad un’ampiezza $\bar{\alpha}$ da utilizzare per la ricostruzione del set di kernel ad un’ampiezza arbitraria $\alpha \in \{\alpha_1, \dots, \alpha_k\}$ con i valori trovati mediante l’interpolazione: il set di kernel scelto in questo lavoro è quello ad ampiezza -20 dB, dal momento che valori più bassi del volume avrebbero reso impossibile la ricostruzione dei kernel di ordine elevato a causa del bassissimo rapporto segnale rumore delle risposte all’impulso.

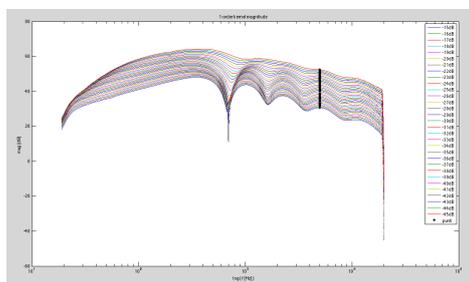
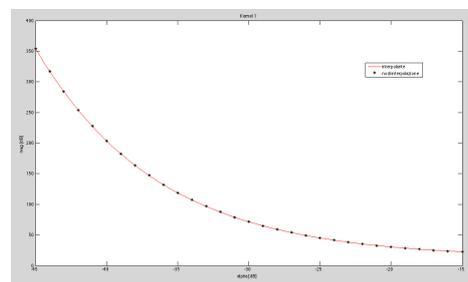
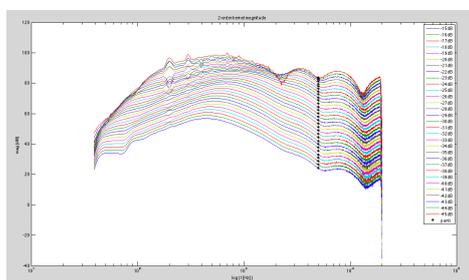
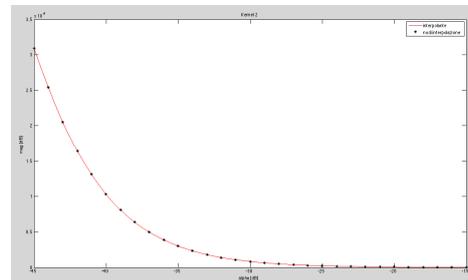
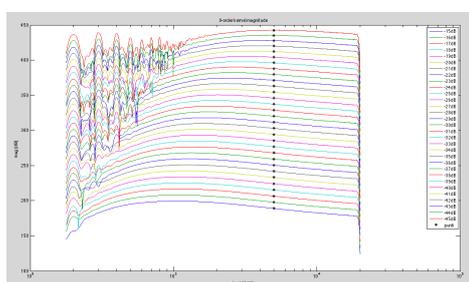
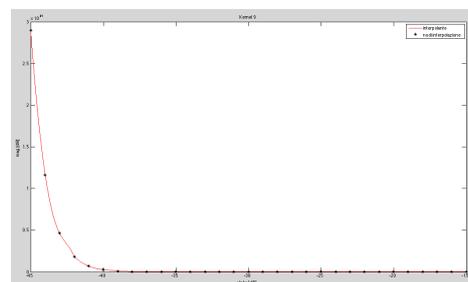
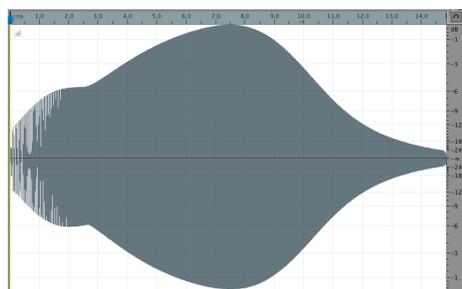
(a) Kernel $h_1(t)$ (b) Kernel $h_1(t)$ (c) Kernel $h_2(t)$ (d) Kernel $h_2(t)$ (e) Kernel $h_9(t)$ (f) Kernel $h_9(t)$

Figura 4.4: Magnitudes $|H_n(\omega)|$ e relativo polinomio interpolante $p_n(\alpha)$.

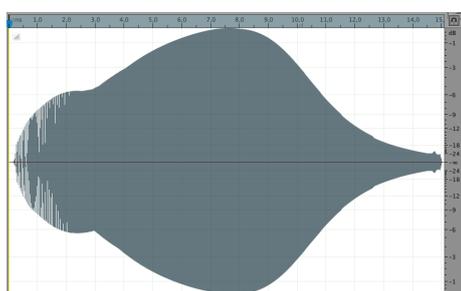
4.1 Risultati ottenuti sugli sweep

Si sono testati i set di kernel “interpolati” a diverse ampiezze α per la virtualizzazione dell’output dello sweep.

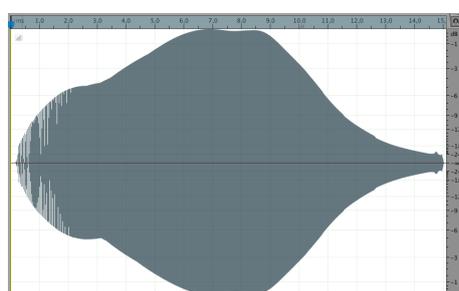
Come di evince dalle figure, il metodo di interpolazione messo in atto fornisce buoni risultati per ampiezze vicine a quella scelta come riferimento, mentre all’abbassarsi del volume la forma d’onda della virtualizzazione differisce progressivamente da quella dell’output misurato.



(a) Output distorto misurato

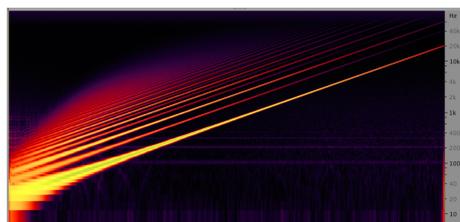


(b) Virtualizzazione con kernel calcolati dalla misura

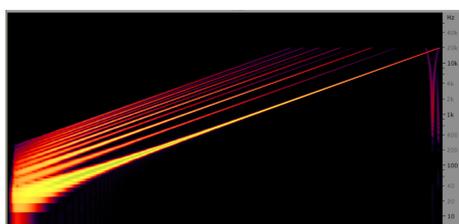


(c) Virtualizzazione con kernel calcolati dall'interpolazione

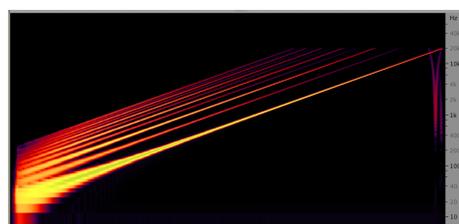
Figura 4.5: Confronto waveform misura - virtualizzazioni per lo sweep a -18 dB.



(a) Output distorto misurato

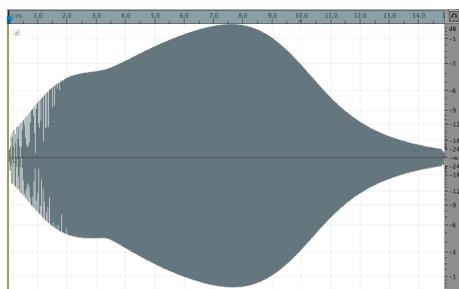


(b) Virtualizzazione con kernel calcolati dalla misura

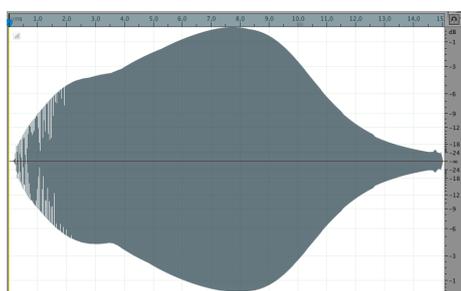


(c) Virtualizzazione con kernel calcolati dall'interpolazione

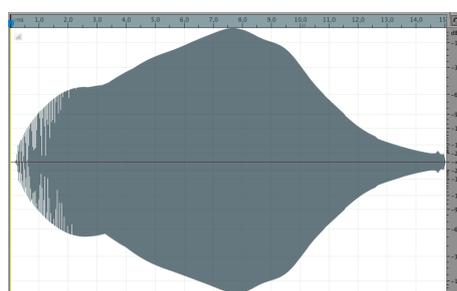
Figura 4.6: Confronto spettrogrammi misura - virtualizzazioni per lo sweep a -18 dB.



(a) Output distorto misurato

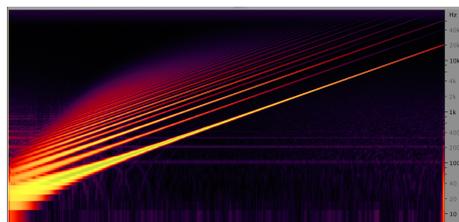


(b) Virtualizzazione con kernel calcolati dalla misura

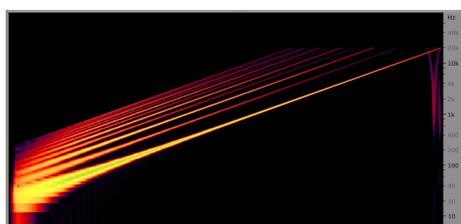


(c) Virtualizzazione con kernel calcolati dall'interpolazione

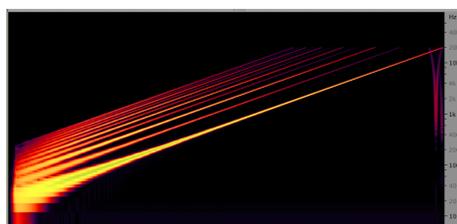
Figura 4.7: Confronto waveform misura - virtualizzazioni per lo sweep a -23 dB.



(a) Output distorto misurato

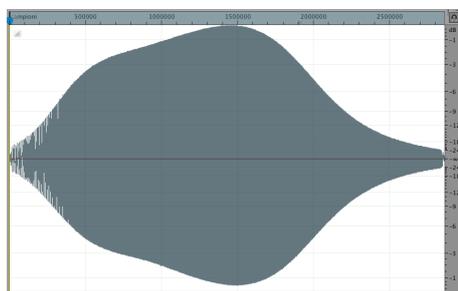


(b) Virtualizzazione con kernel calcolati dalla misura

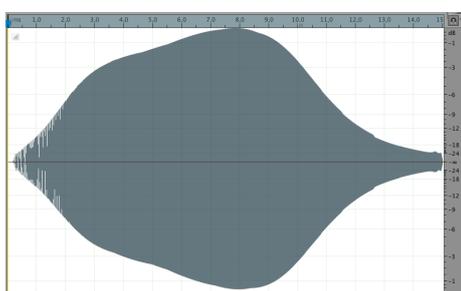


(c) Virtualizzazione con kernel calcolati dall'interpolazione

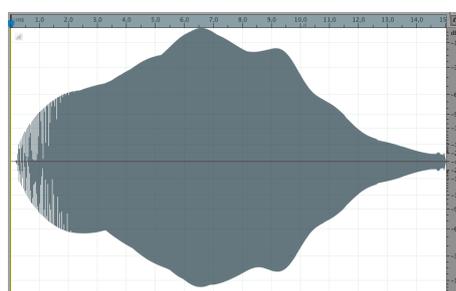
Figura 4.8: Confronto spettrogrammi misura - virtualizzazioni per lo sweep a -23 dB.



(a) Output distorto misurato

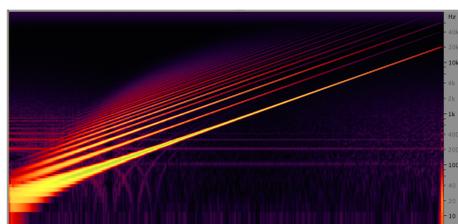


(b) Virtualizzazione con kernel calcolati dalla misura

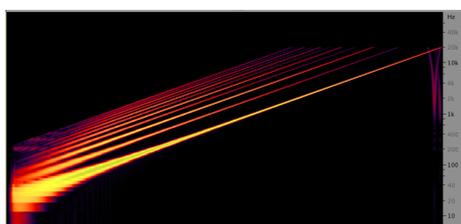


(c) Virtualizzazione con kernel calcolati dall'interpolazione

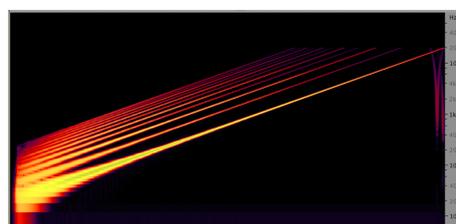
Figura 4.9: Confronto waveform misura - virtualizzazioni per lo sweep a -33 dB.



(a) Output distorto misurato

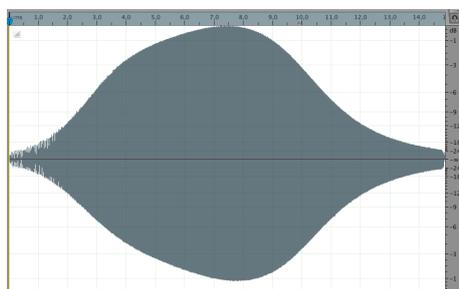


(b) Virtualizzazione con kernel calcolati dalla misura

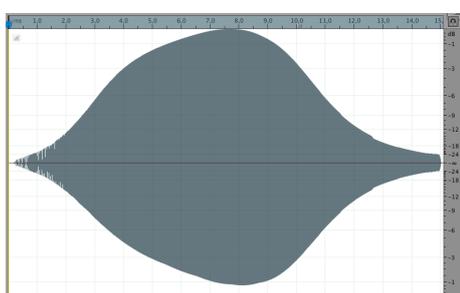


(c) Virtualizzazione con kernel calcolati dall'interpolazione

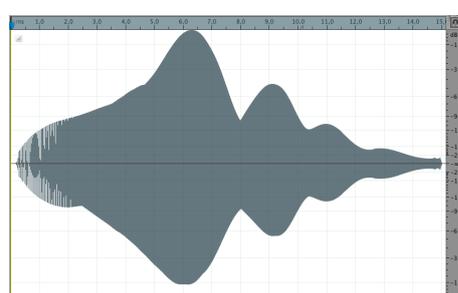
Figura 4.10: Confronto spettrogrammi misura - virtualizzazioni per lo sweep a -33 dB.



(a) Output distorto misurato

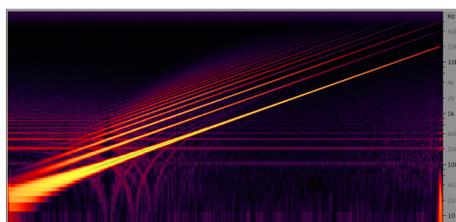


(b) Virtualizzazione con kernel calcolati dalla misura

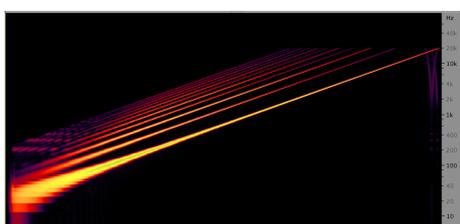


(c) Virtualizzazione con kernel calcolati dall'interpolazione

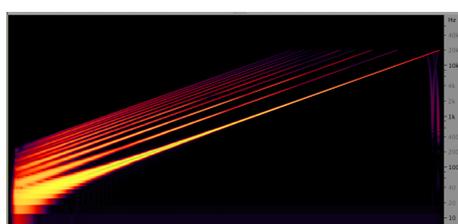
Figura 4.11: Confronto waveform misura - virtualizzazioni per lo sweep a -43 dB.



(a) Output distorto misurato



(b) Virtualizzazione con kernel calcolati dalla misura



(c) Virtualizzazione con kernel calcolati dall'interpolazione

Figura 4.12: Confronto spettrogrammi misura - virtualizzazioni per lo sweep a -43 dB.

4.2 Risultati ottenuti sui segnali reali

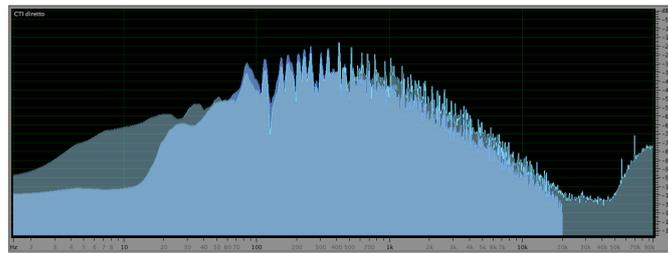
Per utilizzare i set di kernel ottenuti dal procedimento di Nonlinear Convolution anche sui segnali reali si è reso necessario individuare un parametro caratteristico del segnale che permettesse di individuare l'ampiezza del set di kernel da utilizzare: il parametro acustico che ha fornito i risultati migliori è quello di *picco*³

Una volta misurato il valore di picco di ciascuno dei segnali registrati, si è processata la virtualizzazione delle distorsioni sia con i kernel costruiti direttamente dalle risposte all'impulso non lineari che con quelli ricavati dal procedimento di interpolazione. Si riportano di seguito i risultati ottenuti su due segnali reali.

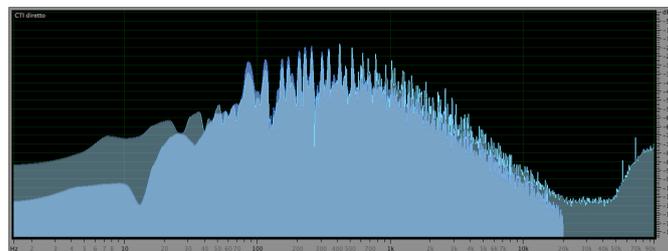
³ovverosia, la massima ampiezza che viene raggiunta dal segnale.

4.2.1 Accordo di chitarra

Il primo segnale che si è scelto di testare è un accordo di chitarra elettrica registrato in presa diretta mediante collegamento dello strumento alla scheda audio Motu Traveler MK-3 mediante cavo non bilanciato. Il segnale “pulito” così registrato è stato successivamente “suonato” attraverso il pedale BOSS SD-1, registrandone l’output distorto (come già nel caso dello sweep) per poter testare la “bontà ” della distorsione virtualizzata coi kernel. Il volume di picco (misurato dall’editor Adobe Audition CS5.5) per tale segnale è -21 dB.

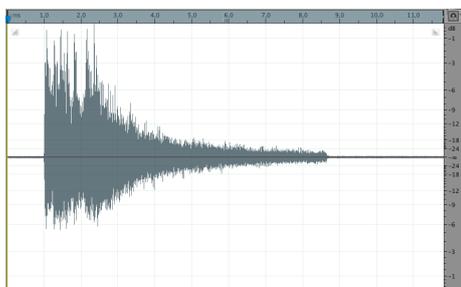


(a) Output distorto (in blu) e virtualizzazione con kernel calcolati dalla misura (in azzurro)

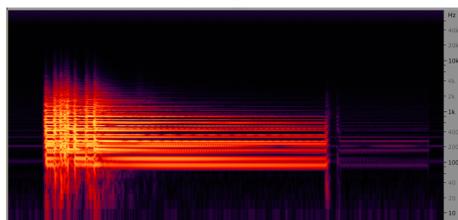


(b) Output distorto (in blu) e virtualizzazione con kernel calcolati dall’interpolazione (in azzurro)

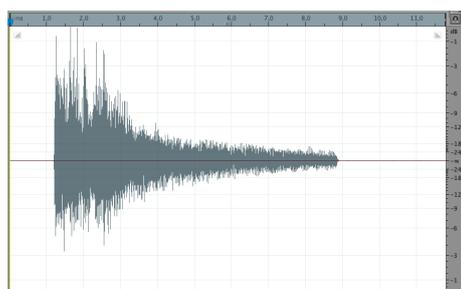
Figura 4.13: Analisi in frequenza misura - virtualizzazioni per il segnale “accordo” al secondo 01:20.



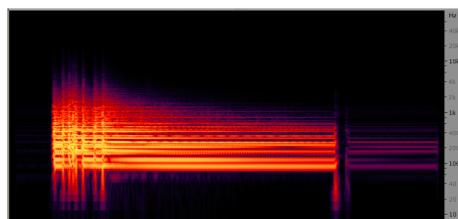
(a) Waveform output distorto misurato



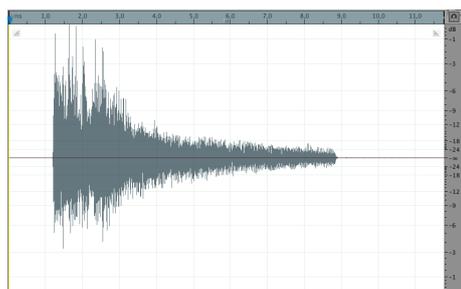
(b) Spettrogramma output distorto misurato



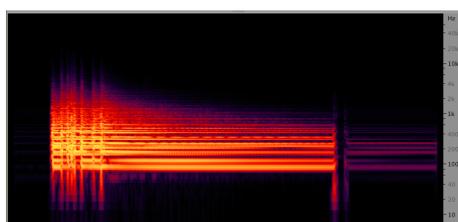
(c) Waveform virtualizzazione con kernel calcolati dalla misura



(d) Spettrogramma virtualizzazione con kernel calcolati dalla misura



(e) Waveform virtualizzazione con kernel calcolati dall'interpolazione

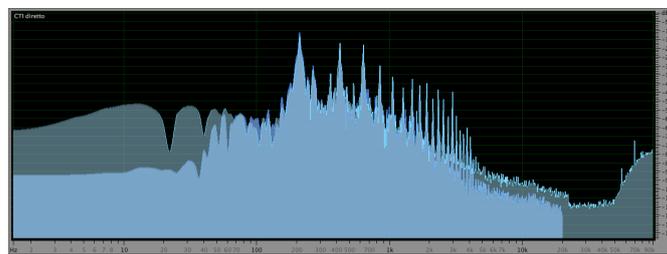


(f) Spettrogramma virtualizzazione con kernel calcolati dall'interpolazione

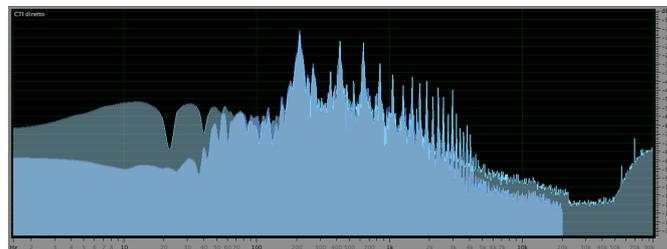
Figura 4.14: Confronto misura - virtualizzazioni per il segnale "accordo".

4.2.2 File audio di chitarra solo

Il secondo segnale che si è scelto di testare contiene gli 11 secondi iniziali del brano “Bang Bang (My baby shot me down)”⁴, nel quale si ascolta una chitarra in a solo. Il volume di tale segnale è stato abbassato di 14 dB per riportarlo nel range volumetrico abituale di una chitarra e successivamente “suonato” attraverso il pedale BOSS SD-1, registrandone l’output distorto come in precedenza. Il volume di picco di tale segnale è -17 dB.



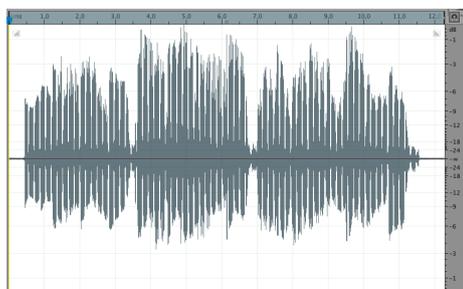
(a) Output distorto (in blu) e virtualizzazione con kernel calcolati dalla misura (in azzurro)



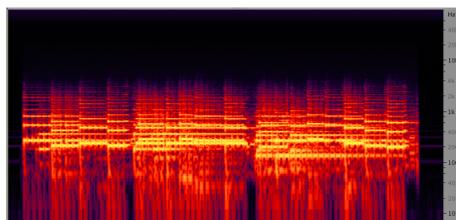
(b) Output distorto (in blu) e virtualizzazione con kernel calcolati dall’interpolazione (in azzurro)

Figura 4.15: Analisi in frequenza misura - virtualizzazioni per il segnale “Bang Bang” al secondo 03:00.

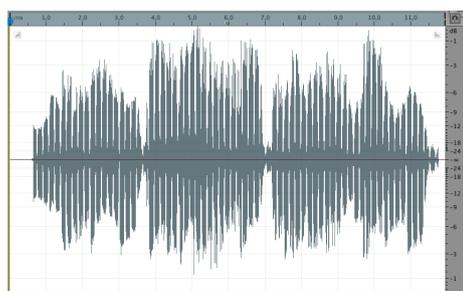
⁴di Nancy Sinatra, contenuto nella colonna sonora del film “Kill Bill Vol.1” di Quentin Tarantino



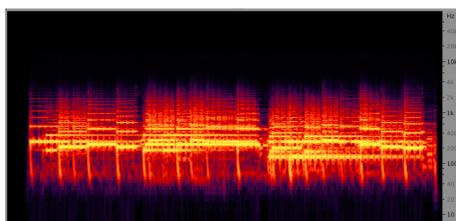
(a) Waveform output distorto misurato



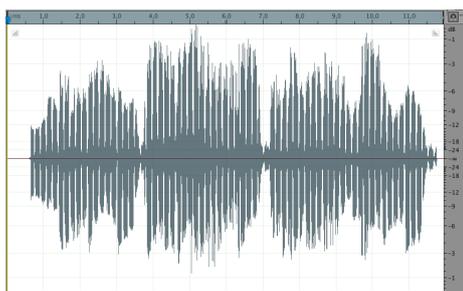
(b) Spettrogramma output distorto misurato



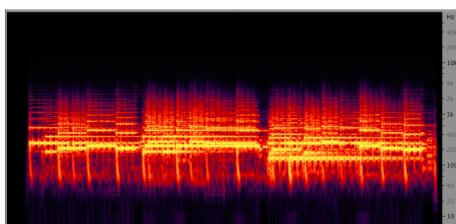
(c) Waveform virtualizzazione con kernel calcolati dalla misura



(d) Spettrogramma virtualizzazione con kernel calcolati dalla misura



(e) Waveform virtualizzazione con kernel calcolati dall'interpolazione



(f) Spettrogramma virtualizzazione con kernel calcolati dall'interpolazione

Figura 4.16: Confronto misura - virtualizzazioni per il segnale “Bang Bang”.

4.2.3 Prova d'ascolto

Una prova d'ascolto operata sui segnali elaborati ha (soggettivamente) riscontrato i fatti seguenti:

- la differenza timbrica tra il segnale registrato e le virtualizzazioni è percepibile (ad esempio, nel segnale “Accordo” il timbro caratteristico della chitarra rimane più nitido nel segnale di output registrato);
- le distorsioni introdotte dalle virtualizzazioni sul segnale “pulito” sono presenti e riscontrabili;
- non si sono notate differenze significative tra la virtualizzazione con i kernel costruiti a partire dalle risposte all'impulso non lineari e quella con i kernel costruiti a partire dai dati ricavati con l'interpolazione.

Capitolo 5

Conclusioni

In questo lavoro di tesi è stato presentato un metodo con il quale è possibile generare un'approssimazione delle non linearità di un sistema audio ad un arbitrario grado di precisione. Rispetto alle modalità descritte nel Brevetto ottenuto dall'Università di Bologna, si sono qui apportate due modifiche:

- il segnale di test utilizzato è il Synchronized Sine Sweep, che non richiede la progettazione di filtri correttivi per le risposte all'impulso non lineari (come era il caso dell'Exponential Sine Sweep);
- è stato messo in opera un primo tentativo di portare l'emulazione verso la progettazione in *real-time*, presentando un modo di poter disporre di set di kernel approssimati a differenti livelli di volume.

Numerose idee sono sorte nel corso del lavoro di tesi con il Prof. Tronchin e l'Ing. Venturi e possono costituire una base importante per futuri sviluppi del metodo:

- l'emulazione in *real-time* delle non linearità mediante caricamento di un buffer di un segnale in corso di esecuzione ed elaborazione di un parametro che permetta la scelta del migliore set di kernel da utilizzare per la virtualizzazione;

-
- l'emulazione in *real-time* delle non linearità mediante caricamento di un buffer di un segnale in corso di esecuzione ed elaborazione di un parametro che permetta la scelta del migliore set di kernel da utilizzare per la virtualizzazione;
 - raffinamento della procedura di interpolazione mediante l'ottimizzazione della scelta della frequenza di interpolazione e del set di kernel di riferimento che minimizzino l'errore commesso nell'approssimazione della virtualizzazione;
 - utilizzo di una scheda di acquisizione dati in luogo di una scheda audio per risolvere i problemi legati allo scarso rapporto segnale/rumore quando si effettuino misure a volumi bassi;
 - raffinamento del modello con l'introduzione di un maggior numero di kernel alle basse frequenze, dove le non linearità sono molto più marcate e presenti che non alle alte frequenze;
 - estensione del modello a kernel di dimensioni superiori ad 1 mediante lo studio delle intermodulazioni, ovverosia delle combinazioni lineari di due (o più) segnali suonati contemporaneamente.

Bibliografia

- [1] Bertoni A., Grossi G., *Dispensa del corso di Elaborazione Numerica dei Segnali*, Anno Accademico 2009-2010.
Link: http://ens.di.unimi.it/dispensa/dispensa_ENS_2009.pdf (consultata il 5 maggio 2013).
- [2] Smith J. O. III, *The Scientist and Engineer's Guide to Digital Signal Processing*.
Link: <http://www.dspguide.com> (consultata il 18 marzo 2013).
- [3] Cariolaro G., Pierobon G., Calvagno G., *Segnali e sistemi*, McGraw-Hill, Milano, 2005.
- [4] Argenti F., Mucchi L., Del Re E., *Elaborazione numerica dei segnali - Teoria, esercizi ed esempi al calcolatore*, McGraw-Hill, Milano, 2011.
- [5] Schetzen M., *The Volterra & Wiener theories of nonlinear systems*, Krieger Publishing Company, Malabar, Florida, 2006.
- [6] Rugh W. J., *Nonlinear system theory - The Volterra/Wiener approach*, versione web del 2002 reperibile all'indirizzo <http://worldtracker.org/media/library/Science/Advanced%20Calculus/Nonlinear%20System%20Theory%20-%20W.%20Rugh.pdf> (consultata il 23 novembre 2012); originariamente pubblicato da The John Hopkins University Press, 1981.
- [7] Volterra V., *Theory of functionals and of integral and integro-differential equations*, Dover Publications Inc, Mineola, New York, 2005.

-
- [8] Frova A., *Fisica nella musica*, Zanichelli, Bologna, 1999.
- [9] Azzaroni L., *Canone infinito. Elementi di teoria della musica*, Clueb, Bologna, 1997.
- [10] Cherry J. A., *Distortion Analysis of Weakly Nonlinear Filters Using Volterra Series*, Thesis for the degree of Master of Engineering, Ottawa-Carleton Institute for Electrical Engineering, Carleton University, Ottawa, Ontario, Canada, December 6, 1994.
- [11] Franz M. O., Schölkopf B., A unifying view of Wiener and Volterra theory and polynomial kernel regression, *Neural computation*, 2006; 18(12): 3097 - 3118.
- [12] Novák A., *Identification of nonlinear systems in Acoustics*, Doctoral Thesis for the degree of Doctor of Philosophy, Specialization Acoustic, École Doctorale, Académie de Nantes, Université du Maine, Le Mans, France, April 6, 2009.
- [13] Novák A., Simon L., Lotton P., Kadlec F., Modeling Of Nonlinear Audio Systems Using Swept-Sine Signals: Application To Audio Effects, Proceedings of the 12th International Conference on Digital Audio Effects (DAFx-9), Como, Italy, September 1 - 4, 2009.
- [14] Novák A., Simon L., Kadlec F., Lotton P., Nonlinear System Identification Using Exponential Swept-Sine Signal, *IEEE Transactions on instrumentation and measurement*, Vol. 59, nr. 8, pp. 2220 - 2229, 2010.
- [15] Novák A., Simon L., Lotton P., Analysis, Synthesis, and Classification of Nonlinear Systems Using Synchronized Swept-Sine Method for Audio Effects, *EURASIP Journal on Advances in Signal Processing*, Vol. 2010, Article ID 793816, 8 pages, 2010.

-
- [16] Novák A., Simon L., Lotton P., Gilbert J., Chebyshev model and Synchronized Swept Sine Method in nonlinear audio effect modeling, Proceedings of the 13th International Conference on Digital Audio Effects (DAFx-10), Graz, Austria , September 6 - 10, 2010.
- [17] Cohen L., *Time-frequency analysis*, Prentice Hall PTR, Upper Saddle River, New Jersey, 1995.
- [18] Cohen L., Instantaneous frequency and group delay of a filtered signal, *Journal of the Franklin Institute*, Vol. 337, pp 329 - 346, 2000.
- [19] Boashash B., Estimating and Interpreting the Instantaneous Frequency of a Signal - Part 1: Fundamentals, Proceedings of the IEEE, Vol. 80, No. 4, pp. 520 -538, April, 1992.
- [20] Boashash B., Estimating and Interpreting the Instantaneous Frequency of a Signal - Part 2: Algorithms and Applications, Proceedings of the IEEE, Vol. 80, No. 4, pp. 540 - 568, April, 1992.
- [21] Farina A., Simultaneous measurement of impulse response and distortion with a swept-sine technique, Audio Engineering Society 108th Convention, Paris, France, February 19 - 22, 2000.
- [22] Farina A., Advancements in impulse response measurements by sine sweeps, Audio Engineering Society 122nd Convention, Vienna, Austria, May 5 - 8, 2007.
- [23] Farina A., Bellini A., Armelloni E., Non-linear convolution: a new approach for the auralization of distorting systems, Proceedings of the 110th Audio Engineering Society Convention, Amsterdam, The Netherlands, May 12 -15, 2001.
- [24] Farina A., Armelloni E., Emulation of not-linear, time-variant devices, by the convolution technique, Audio Engineering Society Italian Section, Annual Meeting, Como, Italy, November 3 - 5, 2005.

-
- [25] Farina A., Farina A., Realtime auralization employing a not-linear, not-time-invariant convolver, Audio Engineering Society 123rd Convention, New York, USA, October 5 - 8, 2007.
- [26] Farina A., Capra A., Campanini S., *La misura della risposta all'impulso per la caratterizzazione di sistemi acustici e vibrazionali*, Strumenti e metodi di misura per l'acustica e le vibrazioni, Seminario in ricordo di Eugenio Mattei, Ancona, 21 settembre 2008
- [27] Tronchin L., Tarabusi V., Improvements of sound 3D auralisation by means of nonlinear convolution, *Advances in Sensors, Signals, Visualization, Imaging and Simulation*, pp. 29-34, 2011.
- [28] Tronchin L., Venturi A., The use of Volterra series for simulating the nonlinear behaviour of musical instruments, Proceedings of the Second Vienna Talk, University of Music and Performing Arts Vienna, Austria, September 19-21, 2010.
- [29] Tronchin L., Non-linear convolution applied to acoustic signals, *Recent Researches in Circuits & Systems*, pp. 485 - 490, 2012.
- [30] Tronchin L., The emulation of nonlinear time-invariant audio systems with memory by means of Volterra series, *Journal of Audio Engineering Society*, Vol. 60, No. 12, pp. 985 - 995, December, 2012.
- [31] Venturi A., *Sviluppo di un software per ricostruzione sonora di sistemi non lineari mediante serie di Volterra*, Tesi di laurea in Fisica Tecnica Ambientale, Facoltà di Ingegneria, Università degli Studi di Bologna, Sessione I, A.A. 2007/2008.
- [32] Kemp J., Primack H., Impulse response measurement of non-linear systems: properties of existing techniques and wide noise sequences, *Journal of Audio Engineering Society*, Vol. 59, No. 12, December, 2011.

- [33] Kibangou A. Y., Favier G., Wiener - Hammerstein Systems Modeling Using Diagonal Volterra Kernels Coefficients, *IEEE Signal Processing Letters*, Vol. 13, No. 6, pp. 381 - 384, June, 2006.
- [34] Cauduro Dias de Paiva, R., Pakarinen, J., Välimäki, V., Reduced-complexity modeling of high-order nonlinear audio systems using swept-sine and principal component analysis, Audio Engineering Society Conference: 45th International Conference: Applications of Time-Frequency Processing in Audio, March, 2012.
- [35] Carassale L., Kareem A., Modeling Nonlinear Systems by Volterra Series, *Journal of Engineering Mechanics*, Vol. 136, No. 6, pp. 801 - 818, June 1, 2010.
- [36] Martynyuk S., *Evaluation of the Dynamic Convolution Principle for simulating a nonlinear behavior of audio equipment*, Diplomarbeit zur Erlangung des Akademischen Grades Magister artium, Universität für Musik und darstellende Kunst, Wien.
- [37] Li F., Zhao Z. J., Zhao Z. D., An adaptive Volterra filtering algorithm with reduced parameters and kernels combination, IEEE 11th International Conference, Signal Processing (ICSP), Beijing, China, October 21 - 25, 2012.
- [38] Zhang X., Bao C., Liu X., Zhang L., A Blind Bandwidth Extension Method of Audio Signals based on Volterra Series, Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC) Asia-Pacific, Hollywood, CA, December 3 - 6, 2012.
- [39] Brevetto nr. US 2012/0328128 A1, *Method for artificially reproducing an output signal of a nonlinear time-invariant system*, Patent Application Publication, December 27, 2012, assegnato ad Alma Mater Studiorum, Università degli Studi di Bologna, inventori Lamberto Tronchin e Andrea Venturi.