



Università degli studi di Cagliari

Dipartimento di Ingegneria Elettrica ed Elettronica
Corso di Laurea Triennale in Ingegneria Elettrica ed Elettronica

Algoritmi per il declipping di segnali audio digitali

Candidato:

Daniele Medda

Relatore:

Prof. Giuseppe Rodriguez

*alla mia famiglia,
che è stata sempre presente*

Indice

1	Introduzione	1
2	Richiami di algebra lineare	6
2.1	Fattorizzazione SVD	7
2.1.1	Fattorizzazione SVD generalizzata (GSVD)	9
2.2	Problemi ai minimi quadrati	11
2.3	Risoluzione di problemi ai minimi quadrati	12
2.3.1	Metodo delle equazioni normali	12
2.3.2	Risoluzione tramite fattorizzazione SVD	14
2.4	Regolarizzazione di Tikhonov	14
3	Metodi di declipping	17
3.1	Algoritmo CBAR (<i>Constrained Blind Amplitude Reconstruction</i>)	17
3.1.1	Risultati sperimentali CBAR	19
3.2	Algoritmo RBAR (<i>Regularized Blind Amplitude Reconstruction</i>)	33
3.2.1	Risultati sperimentali RBAR	36
4	Conclusioni	46

Capitolo 1

Introduzione

Il clipping di un segnale è un tipo di distorsione nel quale il segnale viene troncato in ampiezza. Le ragioni che portano ad avere un segnale distorto in questo modo possono essere varie; nel dominio analogico le cause principali sono:

- superamento dell'ampiezza massima permessa dalla circuitazione, in particolare dalle linee di alimentazione;
- non linearità degli apparati di registrazione, soprattutto se di tipo magnetico come le unità a nastro.

Nel dominio digitale, invece, la causa principale è il superamento del valore massimo rappresentabile con gli n bit a disposizione del convertitore analogico-digitale. Questo valore è pari a 2^n , che per una risoluzione di 16 bit, standard PCM per i CD audio, è pari a 65536.

Una distinzione particolare necessaria tra il clipping analogico e quello digitale è che il primo presenta una tosatura più smussata e morbida, dovuta alla natura dei componenti elettronici, e per questo è definito *soft clipping* (Figura 1.1a). Se indichiamo con x_o il segnale originale da ricostruire, è possibile ricondursi a $x_c = f(x_o)$, indicante il segnale misurato, affetto da soft clipping. Questo segnale è ottenuto applicando a x_o una funzione non lineare smooth $f(\cdot)$. Un esempio semplice di funzione non lineare può essere $f(x_o) = x_o^3$. Per quanto riguarda una possibile ricostru-

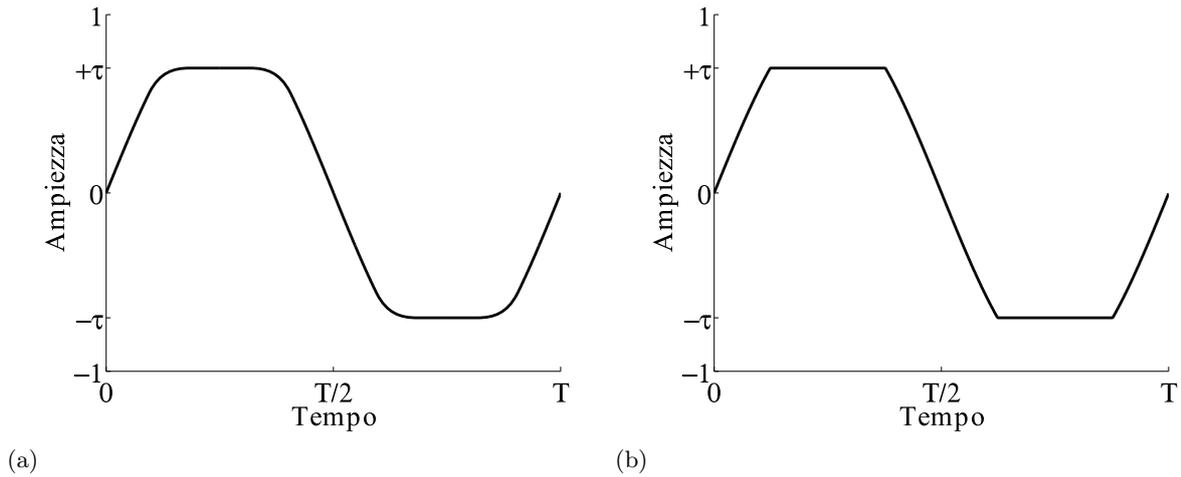


Figura 1.1: Soft clipping (a) e hard clipping (b).

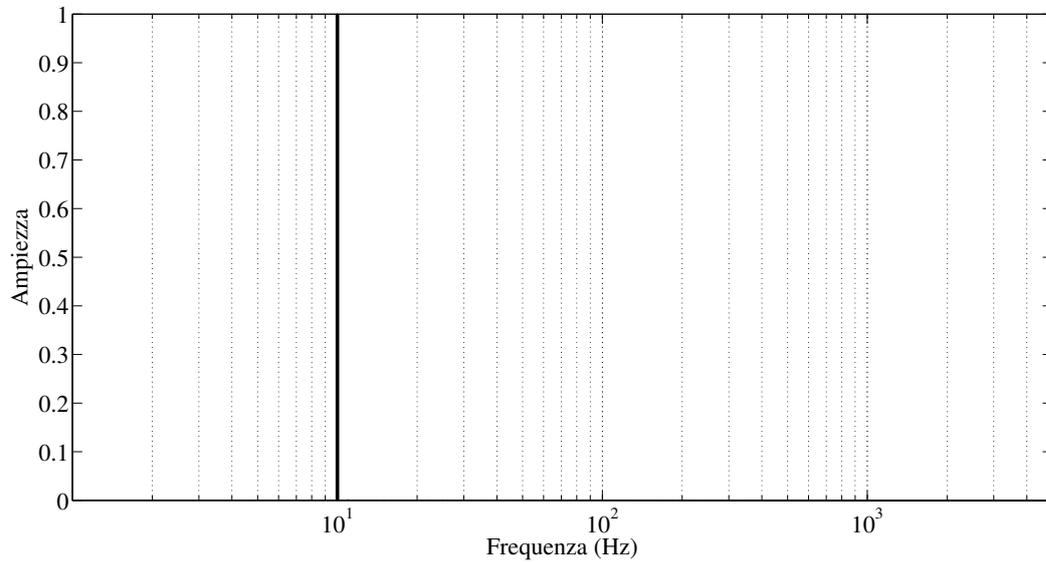
zione del segnale, è ragionevole pensare di costruire una funzione apposita $g(\cdot)$ per compensare la distorsione introdotta dalla funzione non lineare.[2]

Al contrario, il clipping digitale non ha a che fare con circuiti analogici poiché avviene a livello di bit e di conseguenza la sua distorsione è diretta: qualsiasi valore del segnale superiore alla soglia di clipping viene limitato al valore massimo di clipping. In modo simile al caso analogico, questo tipo di clipping è definito *hard clipping* (Figura 1.1b). Consideriamo un segnale digitale campionato a una frequenza di campionamento pari a f_s . Di conseguenza esso sarà rappresentato da un vettore di n elementi, con $n = f_s \cdot t_{max}$, dove t_{max} è la durata in secondi del segnale. Indichiamo con $x[n]$ il segnale originale privo di distorsione (ovvero quello che si tenterà di andare a ricostruire), con $x_c[n]$ il segnale misurato, affetto da hard clipping e con τ la soglia di clip, ovvero il valore assoluto di ampiezza oltre al quale le informazioni del segnale originale vengono perse. Si ha:

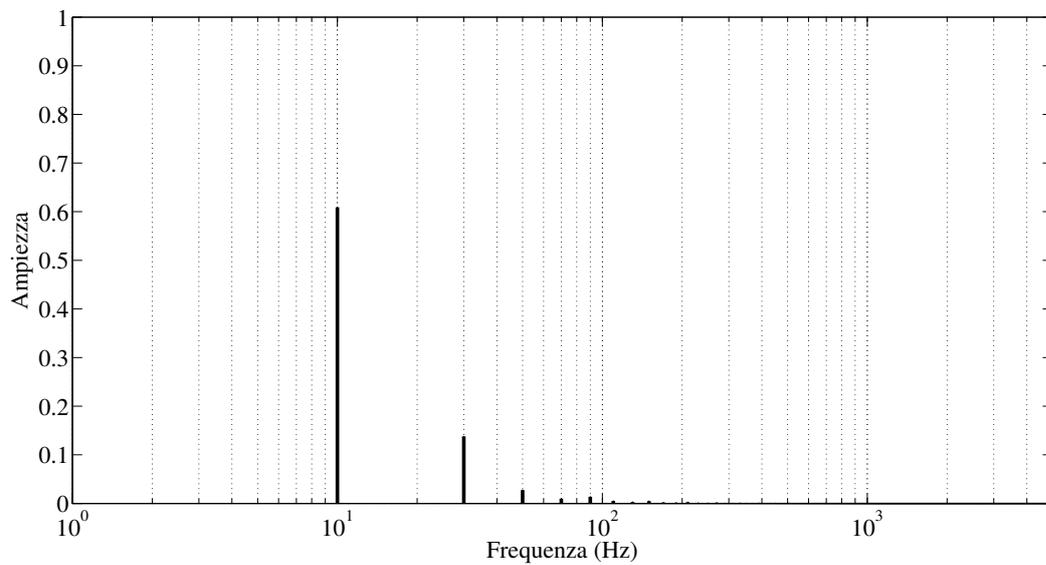
$$x_c[n] = \begin{cases} x[n] & \text{se } |x[n]| < \tau \\ \tau \cdot \text{sign}(x[n]) & \text{se } |x[n]| \geq \tau \end{cases} \quad (1.1)$$

Dal punto di vista armonico, generalmente il clipping aggiunge componenti in alta frequenza. Prendendo a esempio il caso di un semplice segnale

sinusoidale clippato con hard clipping, esso sarà scomponibile (tramite l'analisi di Fourier) in una serie infinita di sinusoidi. È possibile analizzare il contenuto armonico dei due tipi di segnali attraverso la trasformata veloce di Fourier (FFT), che fornisce direttamente lo spettro.



(a) Spettro di una sinusoide a 10 Hz intatta.



(b) Spettro di una sinusoide a 10 Hz clippata per $\tau = 0.50$.

Figura 1.2: Confronto degli spettri di una sinusoide intatta e di una clippata

In Figura 1.2a è raffigurato lo spettro di una sinusoidale di frequenza 10 Hz e ampiezza 1 intatta, senza nessun tipo di distorsione. Come si può osservare, l'unica componente armonica presente nello spettro è quella relativa alla sinusoidale stessa, quindi a 10 Hz. In Figura 1.2b invece è rappresentato lo spettro della stessa sinusoidale, la quale è stata però sottoposta a una operazione di hard clipping, appiattendolo le parti di segnale maggiori della soglia di clipping di 0.50 (tale valore è stato scelto arbitrariamente). È possibile notare, in questo caso, la presenza abbastanza prevedibile di armoniche superiori alla fondamentale di 10 Hz. In particolare sono presenti le componenti armoniche dispari a 30, 50, 70 e 90 Hz; chiaramente il numero di armoniche aggiunte è maggiore di quattro: in questo caso non sono state prese in considerazione le armoniche superiori sia perché fortemente attenuate, e quindi trascurabili, sia per motivi tecnici riguardanti l'impossibilità di rappresentare uno spettro abbastanza ampio con risoluzione sufficiente.

Nel caso di segnali audio, questo fenomeno si traduce in disturbi particolarmente sgradevoli all'ascolto. Come è facilmente prevedibile, un segnale hard clipped avrà un numero di componenti armoniche maggiore di uno soft clipped.

Nell'ambito della restaurazione sonora e delle comunicazioni telefoniche (dove ci si trova spesso a lavorare con segnali pesantemente degradati e clippati) sarebbe desiderabile poter riuscire a ricostruire la parte di segnale mancante. Nonostante questa operazione non sia possibile in maniera perfetta, poiché in ogni caso l'informazione non acquisita è una informazione persa, è possibile utilizzare degli algoritmi che tentano, con risultati variabili, un'approssimazione delle parti mancanti ricostruendo quanto sarebbe potuto essere il segnale originale.

In questa tesi si affronterà il caso del clipping digitale, poiché attualmente l'elaborazione di segnali audio avviene quasi esclusivamente per via digitale. Inoltre, nel dominio digitale è possibile applicare tecniche numeri-

che impossibili da mettere in pratica nel mondo analogico. È da notare che nel caso del clipping, a differenza del caso della distorsione di un segnale dovuta all'assenza di alcuni dei campioni, si è a conoscenza del segno dei campioni da ricostruire. Questa piccola differenza in realtà fornisce una informazione importantissima, che di fatto sarà al centro degli algoritmi di ricostruzione.

Il primo metodo affrontato in questa tesi basa il proprio funzionamento su un semplice problema ai minimi quadrati, in cui si cerca di valutare la parte di segnale mancante in base alle informazioni sulla pendenza assunta dal segnale nei momenti precedenti. Tale metodo, nonostante sia di facile implementazione, funziona in maniera soddisfacente con segnali semplici, mentre con altri più complessi l'errore risulta troppo ampio. Il secondo metodo rappresenta una versione modificata del primo attraverso una regolarizzazione di Tikhonov, che di fatto rende più efficiente il funzionamento.

Gli esperimenti sono stati eseguiti su un set di segnali matematici identico per tutti gli algoritmi e su due file audio mono di un secondo identici (contenenti della musica) a una frequenza di campionamento di 8 e 12 kHz. Le frequenze di 8 e 12 kHz sono appropriate per l'acquisizione e la trasmissione della voce, ma insufficienti per la musica, in quanto gli artefatti dovuti all'aliasing risultano troppo evidenti. La ragione della scelta di tali frequenze di campionamento è dovuta a motivi computazionali, in quanto le prove effettuate a una frequenza di 44.1 kHz (come da standard PCM) avrebbero richiesto un impiego di memoria eccessivo col calcolatore utilizzato per effettuare i calcoli.

Capitolo 2

Richiami di algebra lineare

Il problema di declipping è di fatto un problema **mal posto**. Ricordiamo la definizione di problema ben posto fornita da Hadamard nel 1923[7] un problema è detto ben posto quando esso possiede una e una sola soluzione ed essa dipende con continuità dai dati, all'interno di un determinato campo di definizione. In caso contrario, il problema è mal posto.

In particolare la dipendenza con continuità dai dati ha grande importanza nelle applicazioni reali, poiché nella realtà si ha perennemente a che fare con dati misurati e quindi affetti da errore. Di conseguenza, è possibile che una piccola perturbazione sui dati dovuta agli errori possa portare a una soluzione sensibilmente differente da quella corrispondente ai dati esatti. Quando accade questo, si dice che il problema è **instabile**. È dunque necessario fornire una misura quantitativa di come la soluzione di un problema (anche stabile) sia influenzata da una perturbazione sui dati dovuta a errori. Tale informazione è detta **condizionamento** del problema e viene misurata attraverso il **numero di condizionamento**. Si possono definire due tipologie di numero di condizionamento: assoluto e relativo.

Sia δd una perturbazione sui dati d di un problema e sia δx la perturbazione corrispondente sulla soluzione x . Si definisce numero di condizionamento assoluto $K = K(d)$ la relazione

$$\|\delta x\| \leq K \|\delta d\| ,$$

mentre si definisce numero di condizionamento relativo $k = k(d)$ la relazione

$$\frac{\|\delta x\|}{\|x\|} \leq k \frac{\|\delta d\|}{\|d\|}.$$

Il numero di condizionamento relativo è anche detto più semplicemente *numero di condizionamento*. Inoltre, è da notare che le relazioni appena viste sono valide per qualsiasi norma vettoriale (di fatto non è stato specificato a pedice se si trattasse di norma-1, norma-2 o norma-infinito, per fare alcuni esempi).

Pertanto il numero di condizionamento misura quanto un errore nella misurazione dei dati iniziali venga amplificato nei risultati del problema. In secondo luogo, la propagazione dell'errore in caso di un elevato numero di condizionamento è indipendente dal metodo di risoluzione e dall'utilizzo o meno di un calcolatore.[14]

2.1 Fattorizzazione SVD

La fattorizzazione SVD (*Singular Value Decomposition*) è uno strumento particolarmente adatto alla risoluzione di problemi mal posti.

Sia A una matrice quadrata o rettangolare tale che $A \in \mathbb{R}^{m \times n}$. La fattorizzazione SVD scompone la matrice A nella seguente forma

$$A = U \Sigma V^T = \sum_{i=1}^n u_i \sigma_i v_i^T$$

dove U e V sono matrici con colonne ortonormali tali che

$$U = (u_1, u_2, \dots, u_n) \in \mathbb{R}^{m \times n} \quad \text{E} \quad V = (v_1, v_2, \dots, v_n) \in \mathbb{R}^{n \times n}$$

e Σ è una matrice diagonale con elementi $\sigma_1, \sigma_2, \dots, \sigma_n$. Tali elementi sono detti **valori singolari** di A e sono sempre reali e positivi. Convenzionalmente vengono schematizzati in modo da decrescere in valore man mano

che aumenta l'indice, ottenendo quindi

$$\sigma_1 \geq \sigma_2 \geq \sigma_i \geq \dots \geq \sigma_n \geq 0 .$$

I valori singolari sono sempre ben condizionati, dal punto di vista delle perturbazioni d'errore. Se A viene perturbata da una matrice d'errore E , allora la norma-2 $\|E\|_2$ è limite superiore per la perturbazione di ogni valore singolare.

La fattorizzazione SVD di A è fortemente collegata alla fattorizzazione spettrale delle matrici semi definite positive simmetriche $A^T A$ e AA^T . Da questo deriva che la fattorizzazione SVD di una data matrice A è unica fino a un cambio di segno all'interno della coppia (u_i, v_i) , fatta eccezione per i vettori associati a valori singolari multipli, in quanto solo gli spazi vettoriali proiettati dai vettori sono unici.

All'interno di questa tesi, la fattorizzazione SVD è utilizzata per alcune sue proprietà che la rendono uno strumento potente per poter affrontare problemi mal posti discreti, come di fatto è il problema del declipping. In particolare, sono due le caratteristiche di cui ci serviremo:

- i valori singolari σ_i hanno un decadimento graduale verso zero senza nessun particolare intervallo tra loro nello spettro. L'aumentare delle dimensioni di A non produrrà nessun altro effetto se non quello di aumentare il numero di valori singolari di piccolo valore;
- i vettori singolari sinistro e destro u_i e v_i hanno la tendenza ad avere più cambiamenti di segno nei loro elementi con l'aumentare dell'indice i (e quindi col diminuire di σ_i).

Come appena detto, la SVD è utile per i problemi mal condizionati. È possibile sfruttare tale fattorizzazione per ottenere direttamente informazioni sul condizionamento della matrice A ; consideriamo le seguenti

relazioni:

$$\left. \begin{aligned} Av_i &= \sigma_i u_i, & \rightarrow & \quad \|Av_i\|_2 = \sigma_i \\ A^T u_i &= \sigma_i v_i, & \rightarrow & \quad \|A^T u_i\|_2 = \sigma_i \end{aligned} \right\} \text{ per } i = 1, 2, \dots, n. \quad (2.1)$$

Si può notare che un valore singolare piccolo σ_i , in confronto a $\sigma_1 = \|A\|_2$, indica che esiste una certa combinazione lineare delle colonne di A tale che $\|Av_i\|_2 = \sigma_i$ è piccolo. In modo identico si applica per u_i e per le righe di A . In definitiva, la presenza di uno o più valori singolari σ_i piccoli implica che la matrice A non ha un rango adeguato (comunemente viene detto che A è *rank deficient*) e che i vettori u_i e v_i associati agli σ_i piccoli sono vettori nulli rispettivamente di A^T e di A .

Detto questo, combinando le caratteristiche della SVD illustrate precedentemente e il fatto che essa ci permette di sapere se A è mal condizionata, si arriva alla conclusione che la matrice di un problema mal posto discreto è sempre mal condizionata.[8][3]

2.1.1 Fattorizzazione SVD generalizzata (GSVD)

La fattorizzazione GSVD è una generalizzazione della SVD per una coppia di matrici (A, L) . I valori singolari generalizzati della coppia (A, L) sono semplicemente le radici quadrate degli autovalori generalizzati della coppia $(A^T A, L^T L)$. Supponiamo che le due matrici $A \in \mathbb{R}^{m \times n}$ e $L \in \mathbb{R}^{p \times n}$ soddisfino la condizione $m \geq n \geq p$, come generalmente avviene in tutti i casi di problemi mal posti. Inoltre si richiede che $N(A) \cap L = 0$, dove $N(A)$ indica lo spazio nullo di A (*null space*), e che L sia a rango pieno. È quindi possibile definire la GSVD della coppia di matrici A e L nel seguente modo:

$$A = U \begin{pmatrix} \Sigma & 0 \\ 0 & I_{n-p} \end{pmatrix} X^{-1} \quad , \quad L = V(M, 0)X^{-1} \quad (2.2)$$

Le matrici $U \in \mathbb{R}^{m \times n}$ e $V \in \mathbb{R}^{p \times p}$ sono costituite da colonne ortonormali. Di conseguenza, $U^T U = I_n$ e $V^T V = I_p$. La matrice $X \in \mathbb{R}^{n \times n}$ è quadrata e non singolare, e le sue colonne sono ortogonali. Infine, $\Sigma \in \mathbb{R}^{p \times p}$ e

$M \in \mathbb{R}^{p \times p}$ sono due matrici quadrate diagonali: $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_p)$, $M = \text{diag}(\mu_1, \mu_2, \dots, \mu_p)$. Gli elementi sulle diagonali di Σ e di M sono non negativi e sono in un ordine tale che

$$0 \leq \sigma_1 \leq \dots \leq \sigma_p \leq 1 \quad , \quad 1 \geq \mu_1 \geq \dots \geq \mu_p > 0 \quad ,$$

inoltre sono normalizzati in modo che

$$\sigma_i^2 + \mu_i^2 = 1 \quad \text{per } i = 1, 2, \dots, p .$$

Ognuna delle due matrici A e L è fattorizzata in modo da avere ciascuna un proprio set di valori singolari. È possibile definire un insieme di valori singolari riguardanti l'intera fattorizzazione della coppia (A, L) . Questi valori sono detti **valori singolari generalizzati** della coppia (A, L) e sono definiti come il rapporto dei valori singolari di A e L

$$\gamma_i = \frac{\sigma_i}{\mu_i} \quad \text{per } i = 1, 2, \dots, p .$$

Al contrario dei valori singolari della matrice A , i valori singolari generalizzati sono in un ordine crescente (il motivo è puramente storico).

Analogamente alla fattorizzazione SVD, i valori singolari (σ_i, μ_i) sono ben condizionati rispetto alle perturbazioni presenti nelle matrici A e L ; inoltre anch'essa fornisce tre nuove basi vettoriali linearmente indipendenti (ovvero le colonne di U , V e X), in modo che le due matrici A e L divengano diagonali in corrispondenza a tali cambiamenti di base.

L'algoritmo maggiormente utilizzato per il calcolo della fattorizzazione SVD è stato descritto da Golub e Kahan nel 1965[6]. Per quanto riguarda la GSVD, invece, l'algoritmo più usato è quello presente all'interno della libreria di calcolo numerico LAPACK[1], che implementa il metodo presentato da Christopher C. Paige[13].

2.2 Problemi ai minimi quadrati

Spesso, quando si ha a che fare con applicazioni reali, ci si imbatte in sistemi lineari con un numero di incognite diverso dal numero di equazioni. Abbiamo quindi un sistema del tipo

$$A\mathbf{x} = \mathbf{b}$$

dove la matrice A ha dimensione $m \times n$, il vettore dei termini noti $\mathbf{b} \in \mathbb{R}^m$ e il vettore delle incognite $\mathbf{x} \in \mathbb{R}^n$.

Nell'ipotesi che il rango della matrice A sia massimo, si possono identificare due casi distinti:

- $m > n$: in questo caso si hanno a disposizione più equazioni che incognite; questo si traduce nella possibilità di non esistenza di una soluzione del problema. Il sistema è quindi detto sovradeterminato;
- $m < n$: in opposto al precedente caso, si hanno più incognite che equazioni, quindi il problema può avere infinite soluzioni. Un sistema di questo tipo è detto sottodeterminato.

Da quanto appena visto, appare evidente che il sistema, di qualsiasi dei due casi esso sia, è un problema mal posto. Questo implica che non è possibile trovare una soluzione in un senso classico.

Possiamo tentare di risolvere questo inconveniente riformulando il problema in un problema ben posto. Richiediamo perciò che lo scarto quadratico medio tra il primo e il secondo membro del sistema sia minimo, definendo così il **problema ai minimi quadrati**:

$$\min \|A\mathbf{x} - \mathbf{b}\|_2$$

che risulta zero se il sistema iniziale ammette una soluzione in senso classico, altrimenti si ottiene una soluzione nel senso dei minimi quadrati.

2.3 Risoluzione di problemi ai minimi quadrati

2.3.1 Metodo delle equazioni normali

Sia $f(\mathbf{x})$ una funzione del vettore di incognite \mathbf{x} . Se $f(\mathbf{x}) = \mathbf{b}^T \mathbf{x}$, dove $\mathbf{x}, \mathbf{b} \in \mathbb{R}^n$, allora:

$$\nabla(\mathbf{b}^T \mathbf{x}) = \mathbf{b}$$

dove

$$\nabla f(\mathbf{x}) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right)^T$$

denota il gradiente della funzione f .

Inoltre, se A è una matrice quadrata ($n \times n$), si ha che:

$$\nabla(\mathbf{x}^T A \mathbf{x}) = A \mathbf{x} + A^T \mathbf{x}$$

infine, se A è una matrice simmetrica

$$\nabla(\mathbf{x}^T A \mathbf{x}) = 2A \mathbf{x} .$$

Poiché si ha la necessità di minimizzare una quantità sempre positiva, possiamo tranquillamente considerare il quadrato della norma del residuo

$$\begin{aligned} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 &= (\mathbf{A}\mathbf{x} - \mathbf{b})^T (\mathbf{A}\mathbf{x} - \mathbf{b}) \\ &= \mathbf{x}^T A^T A \mathbf{x} - \mathbf{x}^T A^T \mathbf{b} - \mathbf{b}^T A \mathbf{x} + \mathbf{b}^T \mathbf{b} \\ &= \mathbf{x}^T A^T A \mathbf{x} - 2\mathbf{x}^T A^T \mathbf{b} + \mathbf{b}^T \mathbf{b} . \end{aligned}$$

A questo punto è necessario minimizzare la norma; questo si traduce in una banale ricerca di minimo. Calcoliamo il gradiente e ne imponiamo l'annullamento

$$\nabla(\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2) = 2A^T A \mathbf{x} - 2A^T \mathbf{b} = 0$$

arrivando dunque al seguente sistema:

$$A^T A \mathbf{x} = A^T \mathbf{b} .$$

Giunti a questo punto, è necessario fare la distinzione tra due casi;
Se la matrice A è a rango pieno, allora $A^T A \in \mathbb{R}^{n \times n}$ è invertibile. Questo implica che la soluzione del sistema lineare è unica e vale:

$$\mathbf{x}_{LS} = (A^T A)^{-1} A^T \mathbf{b} .$$

Se invece la matrice A ha rango inferiore a n , non è possibile invertire $A^T A$, poiché singolare. Il sistema però rimane consistente perché il vettore $A^T \mathbf{b}$ appartiene all'immagine di A^T , che coincide con l'immagine di $A^T A$. Ora, i vettori in grado di soddisfare il sistema sono infiniti; comunemente si assume come soluzione quella dotata di minima norma euclidea.

Definiamo con A^\dagger la matrice pseudo inversa di A e vale

$$A^\dagger = (A^T A)^{-1} A^T .$$

A^\dagger è una matrice inversa sinistra ma non un'inversa destra di A :

$$\begin{aligned} A^\dagger A &= (A^T A)^{-1} (A^T A) = I && \text{INVERSA SINISTRA} \\ AA^\dagger &= A (A^T A)^{-1} A^T \neq I && \text{NON INVERSA DESTRA.} \end{aligned}$$

Se A è a rango pieno, $A^T A$ è simmetrica definita positiva. Quindi il sistema normale $A^T A \mathbf{x} = A^T \mathbf{b}$ può essere risolto utilizzando per esempio la fattorizzazione di Cholesky. Il vettore di soluzioni \mathbf{x}_{LS} viene quindi calcolato risolvendo i due sistemi triangolari:

$$\begin{cases} R^T \mathbf{y} = A^T \mathbf{b} \\ R \mathbf{x} = \mathbf{y} \end{cases} ,$$

dove R è una matrice triangolare superiore $n \times n$.

2.3.2 Risoluzione tramite fattorizzazione SVD

Il calcolo della matrice pseudo inversa A^\dagger è piuttosto complessa sul piano computazionale, specialmente se la matrice A ha rango non pieno. In questo si rivela particolarmente utile la fattorizzazione SVD, in quanto alleggerisce parecchio il calcolo.

Se A è invertibile, non è necessario il calcolo della matrice pseudo inversa di Moore-Penrose e dunque la sua inversa è data da

$$A^{-1} = \sum_{i=1}^n v_i \sigma_i^{-1} u_i^T ;$$

la soluzione del sistema $Ax = \mathbf{b}$ è banalmente $x = \sum_{i=1}^n \sigma_i^{-1} (u_i^T \cdot \mathbf{b}) v_i$

Nel caso invece in cui A non sia invertibile, è necessario il calcolo della matrice pseudo inversa. Questo è facilmente eseguibile tramite la SVD, e si ha

$$A^\dagger = \sum_{i=1}^{\text{rank}(A)} v_i \sigma_i^{-1} u_i^T ,$$

che porta quindi alla soluzione in termini dei minimi quadrati

$$x_{LS} = A^\dagger b = \sum_{i=1}^{\text{rank}(A)} \frac{u_i^T \cdot \mathbf{b}}{\sigma_i} v_i. \quad (2.3)$$

È da notare che in (2.3) la divisione per i valori singolari σ_i amplifica le componenti in alta frequenza del vettore dei termini noti \mathbf{b} .

2.4 Regolarizzazione di Tikhonov

La regolarizzazione è una tecnica spesso utilizzata per modificare i problemi ai minimi quadrati in modo che il vettore soluzione abbia delle determinate caratteristiche. Nel nostro caso utilizzeremo la cosiddetta **regolarizzazione di Tikhonov**. [15]

Un problema ai minimi quadrati regolarizzato secondo Tikhonov ha la

seguinte forma:

$$\min_x \|A\mathbf{x} - \mathbf{b}\|_2^2 + \lambda^2 \|L\mathbf{x}\|_2^2, \quad (2.4)$$

dove λ è detto *parametro di regolarizzazione* e ha la funzione di controllare il peso della minimizzazione del termine di regolarizzazione. Per $\lambda = 0$, come prevedibile, il problema regolarizzato equivale a un problema ai minimi quadrati non regolarizzato.

All'interno della definizione (2.4) è stato assunto che gli errori di misura presenti nella parte destra siano tra loro non correlati e che abbiano una matrice varianza - covarianza pari a $\sigma_0^2 I_m$. Supposto che la matrice varianza-covarianza abbia la forma più generale CC^T , con C a rango pieno m , allora è possibile scalare il residuo dei minimi quadrati attraverso la matrice C^{-1} per poi risolvere il problema scalato

$$\min_x \|C^{-1}(A\mathbf{x} - \mathbf{b})\|_2^2 + \lambda^2 \|L\mathbf{x}\|_2^2.$$

Il costo computazione dell'operazione di riscaldamento è spesso sufficientemente basso, poiché dipende dalla struttura della matrice varianza-covarianza C , e frequentemente nella pratica essa è una matrice a banda o direttamente diagonale. Perciò nella maggior parte dei casi reali l'aggiunta del riscaldamento non influisce nel costo computazione globale.[8]

Affinché il sistema lineare sia sufficientemente stabile, è richiesto che il parametro di regolarizzazione λ sia grande; questo però vorrebbe dire anche modificare in modo consistente il sistema, e di conseguenza allontanarci dalla soluzione del sistema originale. Ciò vuol dire che λ deve anche essere sufficientemente piccolo in modo che il sistema su cui si andrà a lavorare sia il più simile possibile a quello originale. Sono stati studiati numerosi metodi per stimare in modo ottimale il valore del parametro di regolarizzazione λ . In ogni caso, un metodo di facile comprensione e che permette di ottenere dei buoni risultati consiste nel fare delle prove variando il parametro e valutando quale determini un comportamento ottimale.

La risoluzione di un problema regolarizzato di Tikhonov può avvenire in

modi analoghi a quanto si fa con problemi non regolarizzati. È possibile utilizzare il metodo delle equazioni normali per poi passare alla soluzione tramite fattorizzazione di Cholesky. Nonostante in teoria questo metodo funzioni egregiamente, nella pratica è piuttosto sconveniente poiché nella realtà si lavora utilizzando calcolatori con aritmetica a precisione finita, il che porterebbe a una inevitabile perdita di informazioni, specialmente nel calcolo di $A^T A$. Inoltre, la presenza del parametro di regolarizzazione λ pone nella condizione di essere costretti a calcolare una fattorizzazione di Cholesky per ogni valore di λ utilizzato, rendendo quindi lungo il suo processo di stima. Il metodo più efficiente e numericamente stabile sarebbe l'algoritmo di bidiagonalizzazione di Eldén[5], che richiede la trasformazione del problema in una forma modificata in cui alla matrice A e al vettore \mathbf{b} vengono sostituiti rispettivamente \bar{A} e $\bar{\mathbf{b}}$. Quindi il problema viene ridotto a uno equivalente altamente strutturato e con matrici sparse. Per lo scopo di questa tesi, tale metodo è fin troppo complicato, perciò si è preferito utilizzare un metodo meno efficiente ma più semplice. Il metodo scelto si basa sull'utilizzo della fattorizzazione GSVD, che dal punto di vista computazionale è circa tre volte meno efficace del metodo di bidiagonalizzazione. Inoltre l'utilizzo di questo metodo permette di avere a disposizione maggiori informazioni sul problema di regolarizzazione.

Tutti i metodi affrontati in questa sezione sono implementati nel pacchetto *Regularization Tools*[9] per Matlab sviluppato da Per Christian Hansen. Questo pacchetto è stato utilizzato estensivamente per lo sviluppo di questa tesi, in particolare le funzioni `cgsvd` e `tikhonov`.

Capitolo 3

Metodi di declipping

3.1 Algoritmo CBAR (*Constrained Blind Amplitude Reconstruction*)

Si affronterà ora il primo metodo per la ricostruzione di un segnale presentante clipping digitale. Questo algoritmo, chiamato Constrained Blind Amplitude Reconstruction (CBAR) basa il suo funzionamento sull'interpolazione dei segmenti clippati minimizzando l'energia della derivata seconda del segnale ricostruito. L'utilizzo della derivata seconda è giustificato dal fatto che essa è in grado di fornire una stima della pendenza che avrebbe dovuto avere il segnale originale. Ora si vedrà la definizione formale.

Sia $\mathbf{x} \in \mathbb{R}^{L \times 1}$ un vettore colonna di lunghezza L . Tale vettore contiene tutti i campioni del segnale acquisito. Si può dunque affermare che $\mathbf{x} = x_c$, dove x_c è il segnale misurato definito in (1), affetto da clipping. Supponiamo che all'interno di \mathbf{x} ci siano R campioni non clippati e che $R \leq L$ (anche se è ovvio che con $R = L$ tutti i campioni sono intatti e non avrebbe senso eseguire il restauro). È dunque possibile partizionare \mathbf{x} in due vettori: $\mathbf{x}_r \in \mathbb{R}^{R \times 1}$, che contiene gli R campioni intatti, e $\mathbf{x}_c \in \mathbb{R}^{C \times 1}$, contenente invece $C = L - R$ campioni danneggiati.

Sia ora $S_r \in \mathbb{R}^{R \times L}$ una matrice $R \times L$ ottenuta rimuovendo le righe corrispondenti ai campioni clippati dalla matrice identità $L \times L$. In modo analogo, sia $S_c \in \mathbb{R}^{C \times L}$ una matrice $C \times L$ ottenuta rimuovendo le righe corrispondenti ai campioni intatti dalla matrice identità $L \times L$. Infine, in-

dichiamo con D_2 l'operatore lineare di derivata seconda; questo operatore è rappresentabile come matrice quadrata $\in \mathbb{R}^{L \times L}$ di Toeplitz

$$D_2 = \begin{bmatrix} -2 & 1 & 0 & \cdots & 0 \\ 1 & -2 & 1 & \cdots & 0 \\ 0 & 1 & -2 & \cdots & 0 \\ \vdots & \vdots & \cdots & \cdots & 1 \\ 0 & 0 & 0 & 1 & -2 \end{bmatrix} .$$

È facilmente intuibile che

$$\mathbf{x} = S_r^T \mathbf{x}_r + S_c^T \mathbf{x}_c . \quad (3.1)$$

L'algoritmo CBAR tenta di ricostruire i campioni clippati risolvendo il seguente problema di minimizzazione:

$$\begin{aligned} \min_{\mathbf{x}_c} & \|D_2 (S_r^T \mathbf{x}_r + S_c^T \mathbf{x}_c)\|_2^2 \\ \text{subj} & \quad \mathbf{x}_c \circ \text{sgn}(S_c \mathbf{x}) \geq \tau \cdot \mathbf{1} , \end{aligned}$$

dove il simbolo \circ indica il prodotto di Hadamard (elemento per elemento) di due vettori o matrici.

Analizzando il problema di minimizzazione, è da notare che il prodotto $S_c \mathbf{x}$ è un vettore di dimensione $C \times 1$ contenente tutti i campioni clippati del segnale originale. Il vincolo del problema impone che se il campione clippato ha valore $+\tau$, allora il campione ricostruito dovrà necessariamente essere maggiore o uguale a $+\tau$. Al contrario, se il campione clippato vale $-\tau$, il campione ricostruito dovrà essere minore o uguale a $-\tau$. [10]

Implementazione in pseudo codice

L'implementazione di questo algoritmo è piuttosto semplice. Come primo passo, è necessario ottenere τ , ovvero il livello di clipping. Si pre-supporrà di essere già a conoscenza che il segnale che si vuole processare sia degradato. Questa è un'ipotesi semplificativa necessaria, in quanto in

questo modo la ricerca di τ si riduce a una banale ricerca di valore massimo. Senza tale ipotesi, la situazione diventa sensibilmente più complicata e sono stati presentati negli anni svariati metodi di ricerca del livello di clipping.[12] [4]

Algoritmo 1: Constrained Blind Amplitude Reconstruction

Dati: \mathbf{x} , τ , L

Risultato: $\mathbf{x}_{declipped}$

```

1 for  $i : 1$  to  $L$  do
2   if  $|\mathbf{x}[i]| < \tau$  then
3      $\mathbf{x}_r[i] = \mathbf{x}[i]$ ;
4   else
5      $\mathbf{x}_c[i] = \mathbf{x}[i]$ ;
6  $S_r = S_c = I \in \mathbb{R}^{L \times L}$ ;
7 Rimuovi da  $S_r$  le righe corrispondenti ai campioni clippati;
8 Rimuovi da  $S_c$  le righe corrispondenti ai campioni intatti;
9  $A = D_2 \cdot S_c^T$ ;
10  $\mathbf{b} = -D_2 S_r^T \mathbf{x}_r$ ;
11 Risolvi LSP  $\|A \mathbf{x}_c(\text{ricostruito}) - \mathbf{b}\|^2$ ;
12  $\mathbf{x}_{declipped} = S_r^T \mathbf{x}_r + S_c^T \mathbf{x}_c(\text{ricostruito})$ ;
```

Nel caso di implementazione di questo algoritmo con un linguaggio reale (non pseudo codice), è opportuno utilizzare strutture dati per matrici sparse per S_r , S_c e D_2 . Questo permette un notevole risparmio di memoria centrale, soprattutto se si lavora con segnali di grande dimensione. Nel caso di questa tesi, si è utilizzato Matlab, che possiede un'ottima gestione nativa delle matrici sparse.

3.1.1 Risultati sperimentali CBAR

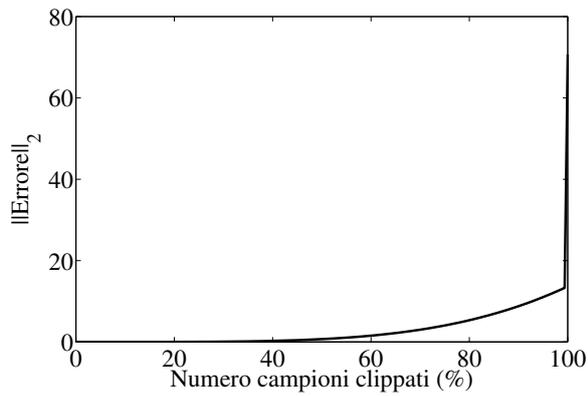
L'algoritmo CBAR è stato testato con il set di segnali seguente: sinusoide, onda triangolare, onda a dente di sega, un segnale poco complesso risultato somma di tre sinusoidi diverse, due segnali audio monofonici identici campionati alle frequenze di 8 kHz e 12 kHz. Un segnale puramente a onda quadra o rettangolare è stato completamente ignorato dai test poiché, come banalmente intuibile, il clipping su un segnale di quelle tipologie

ha solamente l'effetto di diminuire l'ampiezza, senza aggiungere nessuna componente in frequenza. Di conseguenza non ha senso applicare un algoritmo di declipping a un segnale quadro/rettangolare in quanto è sufficiente moltiplicarlo per un numero $G > 1$ per ripristinare la sua ampiezza originale.

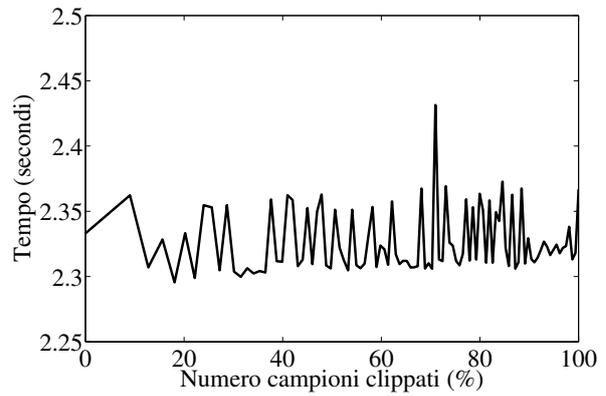
Declipping di una sinusoide

Il primo test effettuato è stato il declipping di un segnale sinusoidale, che è stato distorto con soglie di clipping via via più basse (attraverso uno script Matlab). Questo ha permesso di valutare l'errore commesso nella procedura di declipping al variare del numero di campioni clippati. In Figura 3.1a è rappresentata la norma-2 dell'errore commesso nel declipping tramite CBAR al variare della percentuale di campioni clippati. È possibile notare che, per il caso semplice di una sinusoide, il metodo ha un funzionamento abbastanza soddisfacente. L'errore inizia ad essere considerevole (circa 18) per percentuali di $x_c > 97\%$, che è un ottimo valore, visto che nella realtà difficilmente ci si imbatte in un segnale audio sinusoidale con una percentuale così alta di campioni clippati. In Figura 3.1c è mostrata la sinusoide oggetto del test. Distorcendo opportunamente il segnale per una soglia di clipping di $\tau = 0.2$, che corrisponde a una percentuale di campioni clippati pari a $x_c = 86.8\%$, si ottiene il segnale in Figura 3.1d. Successivamente, il segnale è stato sottoposto alla procedura di declipping mediante CBAR (Figura 3.1e), con un errore massimo di 0.135 (in Figura 3.1f è rappresentato l'intero andamento dell'errore).

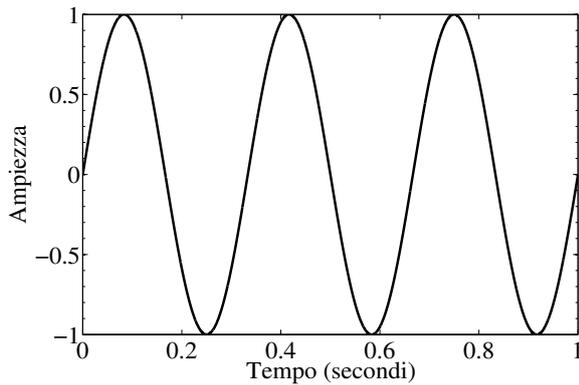
Si può notare che in questo caso la problematica maggiore riguarda la previsione dell'ampiezza avuta dal segnale originale e la possibilità di non poter intervenire in nessun modo per regolarla, in quanto il problema ai minimi quadrati prevede solamente il vincolo $\mathbf{x}_c \circ \text{sgn}(S_c \mathbf{x}) \geq \tau \cdot \mathbf{1}$. Questa è una delle limitazioni principali del CBAR.



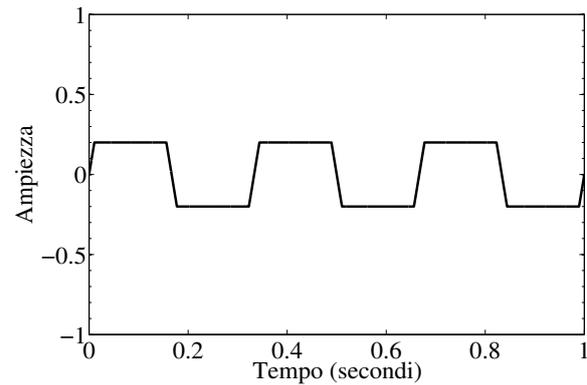
(a) Norma-2 dell'errore nel declipping di una sinusoide al variare di τ .



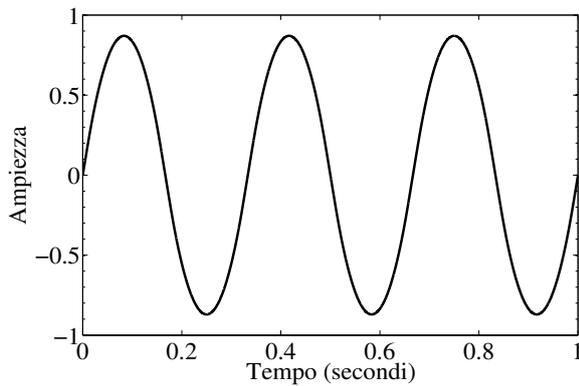
(b) Tempi di esecuzione del CBAR per una sinusoide al variare di τ .



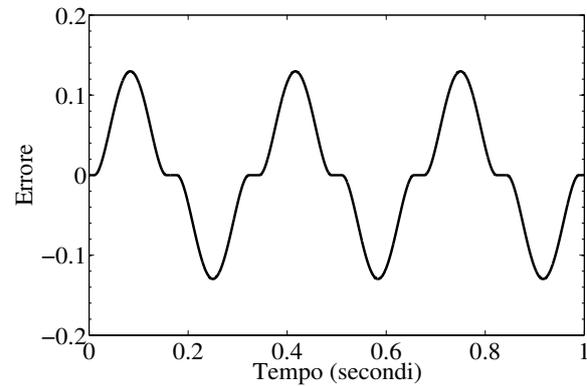
(c) Sinusoide oggetto del test.



(d) Sinusoide di Figura 3.1c clippata per $\tau = 0.2$.



(e) Declipping del segnale di Figura 3.1d.

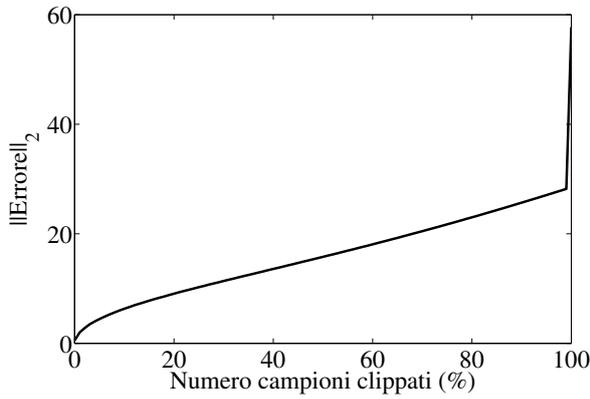


(f) Errore risultante dal declipping tramite CBAR del segnale di Figura 3.1c.

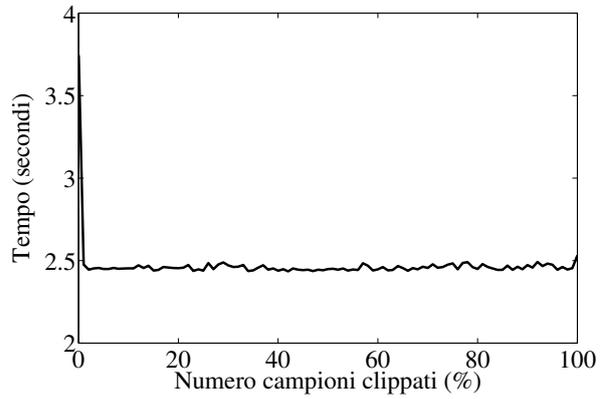
Figura 3.1: Risultati sperimentali del CBAR su delle sinusoidi.

Declipping di un'onda triangolare

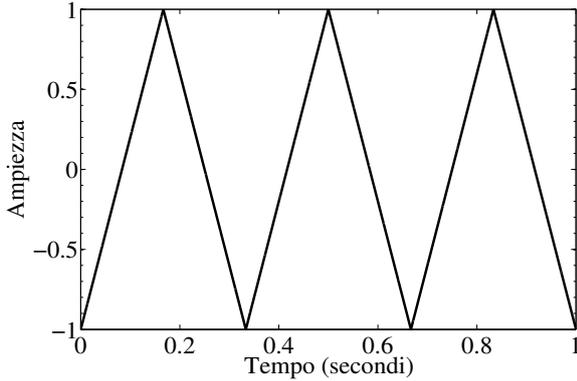
Il successivo segnale con cui si è andati ad effettuare i test dell'algoritmo CBAR è un segnale triangolare, di ampiezza 1 e nuovamente alla frequenza di 3 Hz. La procedura d'analisi è la medesima già affrontata per il caso della sinusoide.



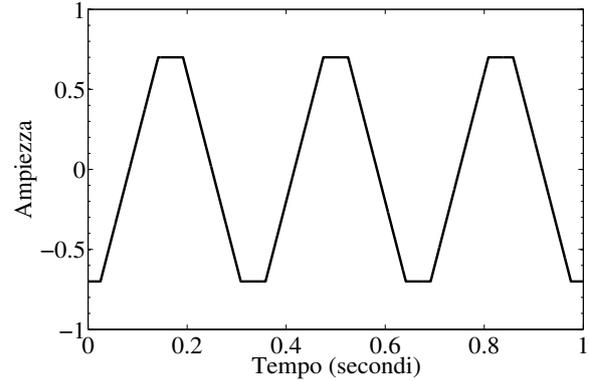
(a) Norma-2 dell'errore nel declipping di un'onda triangolare al variare di τ .



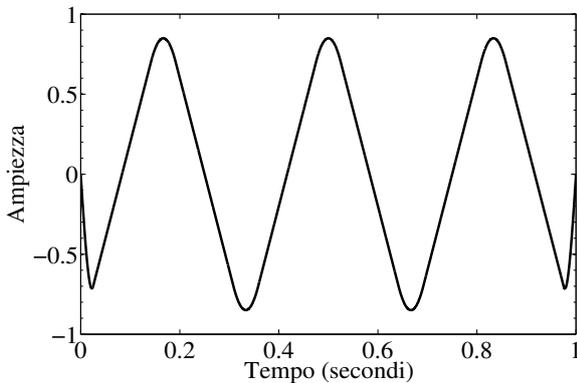
(b) Tempi di esecuzione del CBAR per un'onda triangolare al variare di τ .



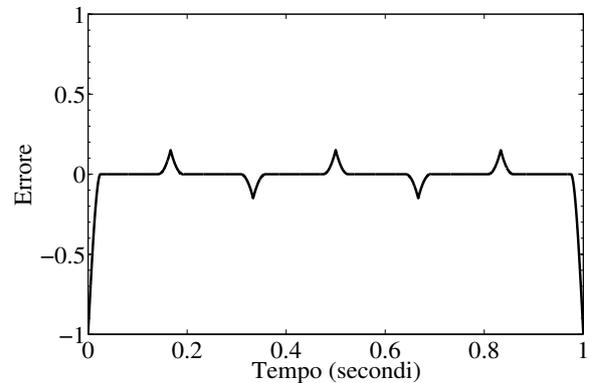
(c) Onda triangolare oggetto del test.



(d) Segnale triangolare di Figura 3.2c clippata per $\tau = 0.7$.



(e) Declipping del segnale di Figura 3.2d.



(f) Errore risultante dal declipping del segnale di 3.2d.

Figura 3.2: Risultati sperimentali del CBAR su delle onde triangolari.

Dall'analisi della Figura 3.2a si può affermare che l'andamento della norma-2 dell'errore è inizialmente logaritmico (anche se in modo molto leggero), mentre successivamente diviene in gran parte lineare. La spezzata risultante dall'analisi evidenzia un innalzamento brusco dell'errore per percentuali di campioni clippati molto alte, mentre l'errore è sensibilmente alto con una percentuale dell'80% di campioni clippati.

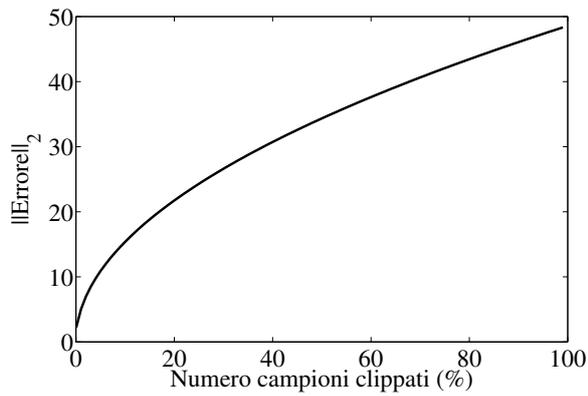
Per quanto concerne i tempi di calcolo, essi sono sufficientemente indipendenti dal numero di campioni distorti. Nella Figura 3.2b è visibile il fatto che i tempi di calcolo restano nell'intorno dei 2.5s, fatta eccezione per il picco iniziale a 3.55s, dovuto sostanzialmente ai tempi di setting di Matlab per l'esecuzione dello script.

Analizziamo ora uno dei casi di declipping di un'onda triangolare eseguiti. L'onda triangolare in Figura 3.2c è stata clippata per $\tau = 0.7$, che corrisponde a 30.14% di campioni clippati (il segnale risultante è visibile in Figura 3.2d). Il risultato dell'operazione di declipping è mostrato in Figura 3.2e. Ovviamente la ricostruzione risultante è piuttosto *smooth*, perché è impossibile ricostruire una punta come quella caratteristica di un segnale triangolare utilizzando una singola funzione sinusoidale, come avviene risolvendo il problema ai minimi quadrati presente all'interno del CBAR. Di conseguenza, il segnale ricostruito presenta una forte similitudine con una senoide (la cosa è più evidente se la soglia di clipping τ è bassa).

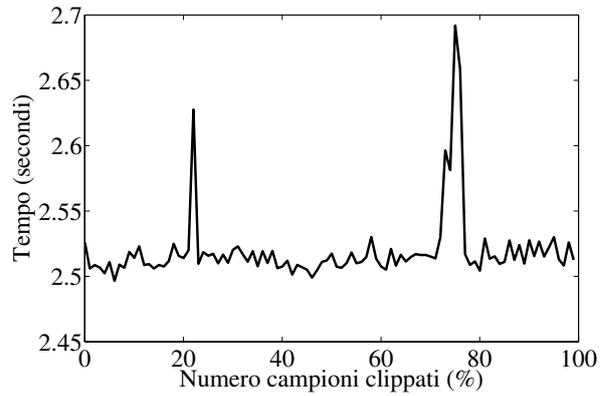
Declipping di un'onda a dente di sega

L'ultimo segnale semplice che si è andato ad analizzare ha una forma d'onda a dente di sega. Possiamo ritenere di essere al limite, per quanto riguarda il tipo di segnale semplice che è possibile declippare col CBAR, perché, come è visibile in 3.3a, fin da percentuali di campioni clippati abbastanza basse si ha un errore alto. La ragione di questo errore così grande è dovuta alla discontinuità naturale del segnale a dente di sega originale. In un segnale con una discontinuità come quella presente in questo caso, i campioni distorti positivamente e negativamente risultano contigui. L'algoritmo CBAR non fa distinzione tra campioni clippati positivamente e negativamente e quindi tenta di ricostruire, con pessimi risultati, anche la discontinuità, nonostante questo non sia necessario.

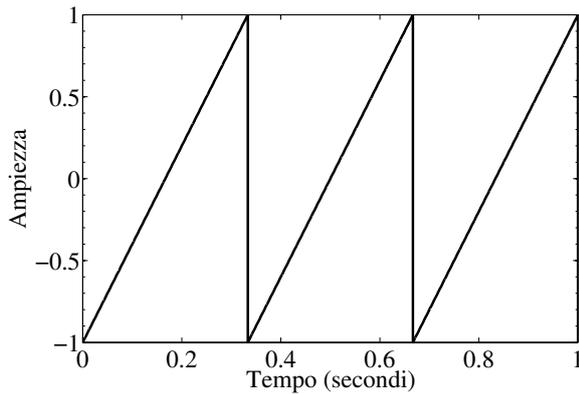
È necessario puntualizzare però che il segnale a dente di sega analizzato in questa sede è praticamente identico alla forma matematica. Nella



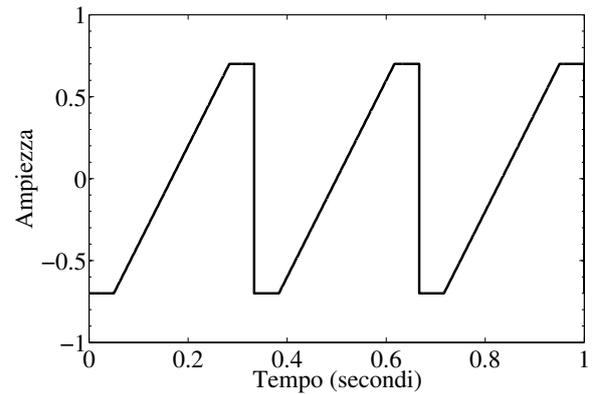
(a) Norma-2 dell'errore nel declipping di un'onda a dente di sega al variare di τ .



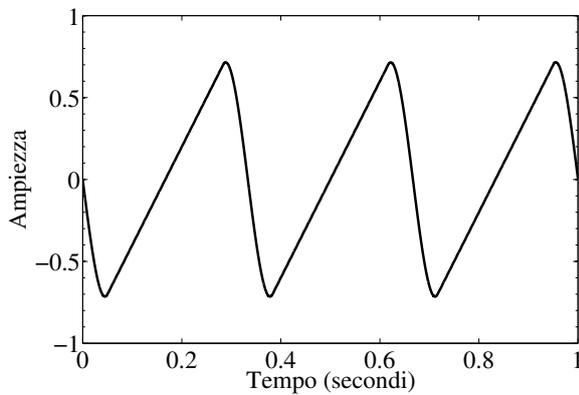
(b) Tempi di esecuzione del CBAR per un'onda a dente di sega al variare di τ .



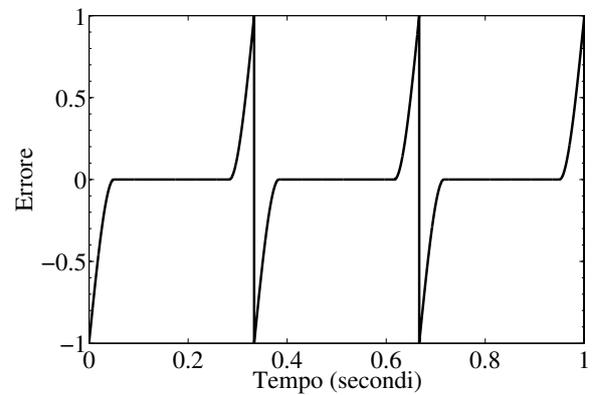
(c) Onda a dente di sega oggetto del test.



(d) Segnale a dente di sega di Figura 3.3c clippata per $\tau = 0.7$.



(e) Declipping del segnale di Figura 3.3d.



(f) Errore risultante dal declipping del segnale di 3.3d.

Figura 3.3: Risultati sperimentali del CBAR su delle onde a dente di sega.

realtà, è molto difficile (per non dire impossibile) che si abbia a che fare con un segnale a dente di sega con una discontinuità netta come in questo caso. Nel caso reale, la discontinuità avrà sempre una pendenza (dovuta a parametri di non idealità dei componenti elettronici, come ad esempio lo slew-rate degli amplificatori operazionali) e, compatibilmente

con una frequenza di campionamento sufficiente, essa sarà apprezzabile e di conseguenza il declipping presenterà un errore minore.

Declipping di un segnale complesso

Ora si passerà ad analizzare i risultati ottenuti effettuando dei test su un segnale mediamente complesso, ottenuto dalla somma di tre sinusoidi. Il segnale ha la seguente espressione

$$s(t) = 4.06 \sin(2\pi \cdot 13t) + 1.45 \sin(2\pi \cdot 73t) + 2.37 \sin(2\pi \cdot 46t) \quad (3.2)$$

ed è stato ottenuto generando in modo pseudo casuale i valori ampiezza e frequenza, affinché fosse possibile avere di fatto un segnale più simile a uno audio acquisito nella realtà. Graficamente il segnale appare come in Figura 3.1.1.

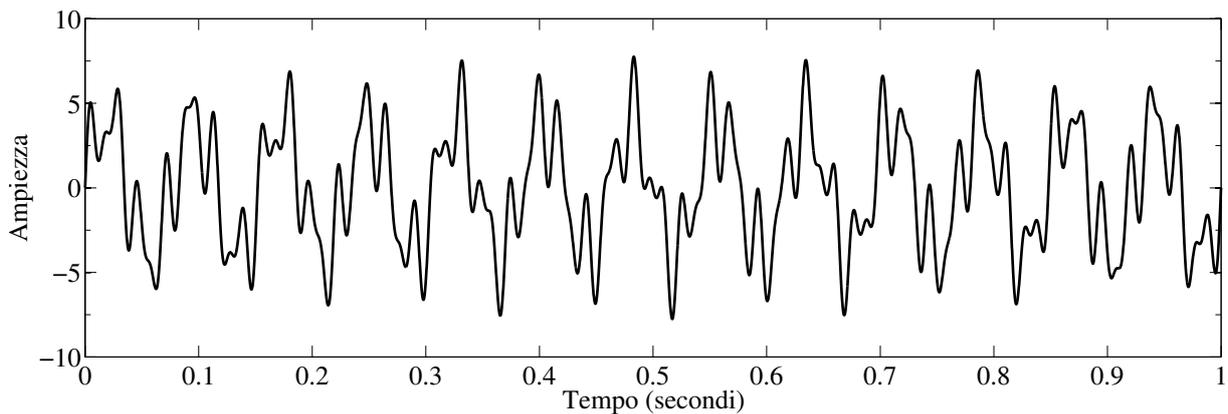


Figura 3.4: Rappresentazione grafica del segnale (3.2).

Poiché ci si avvicina al tipo di segnale che si andrà a trattare nella realtà, si ritiene necessario analizzare anche il comportamento in frequenza dell'algoritmo CBAR. Il segnale originale ha lo spettro armonico di Figura 3.1.1. È possibile distinguere chiaramente le tre componenti, rispettivamente a 13, 46 e 73 Hz.

Dal punto di vista dell'errore introdotto dal restauro di questo segnale, come si può vedere dalla Figura 3.6a, esso ha un andamento grosso modo lineare fino a una percentuale di campioni clippati pari al 64%. Successivamente l'andamento dell'errore aumenta sensibilmente con numerose

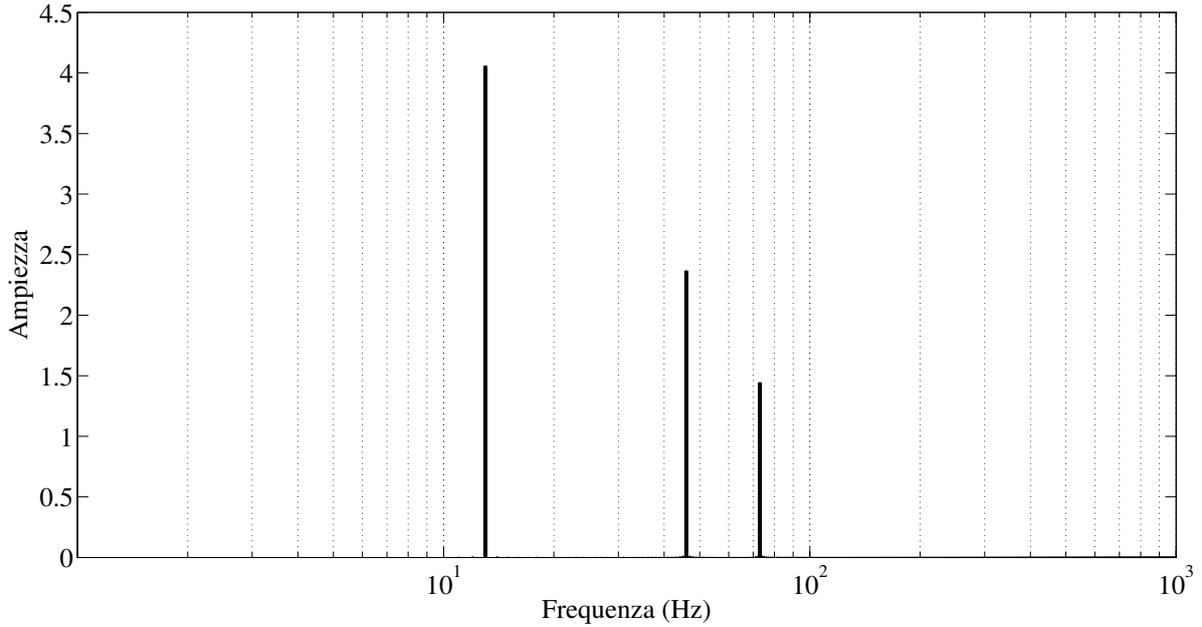
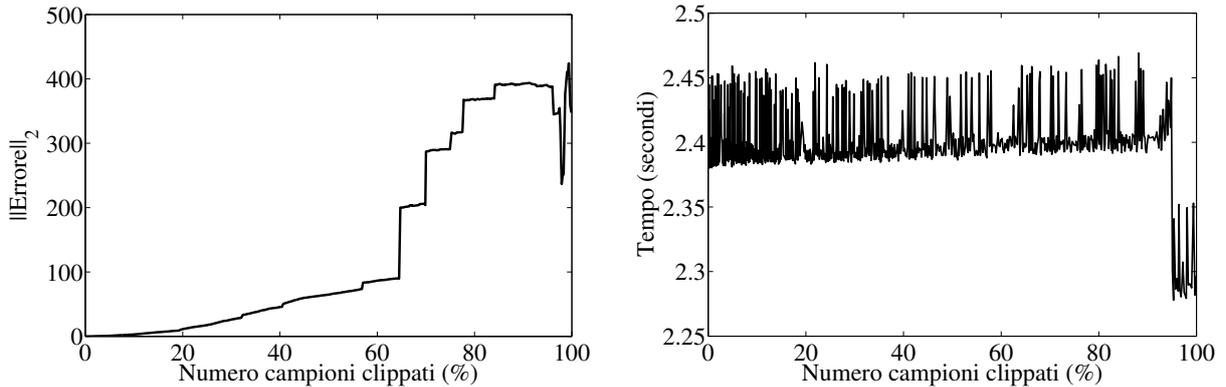


Figura 3.5: Spettro del segnale (3.2).

discontinuità, arrivando a un picco di errore in norma-2 pari a 400, in corrispondenza di percentuali di campioni clippati nell'intervallo tra il 85% e il 98%. Di conseguenza, si può ritenere che il limite massimo per cui può essere utilizzato il CBAR con tale segnale corrisponde al 64% di campioni clippati, che è un risultato estremamente buono, considerando come sempre che difficilmente ci si imbatte in segnali distorti in questo modo.



(a) Norma-2 dell'errore nel declipping del segnale di Figura 3.1.1 al variare di τ .

(b) Tempi di esecuzione del CBAR per il segnale di Figura 3.1.1 al variare di τ .

Figura 3.6: Risultati sperimentali del CBAR sul segnale composto di Figura 3.1.1

Per quanto riguarda invece i tempi di calcolo, questi rimangono abbastanza costanti al variare di τ (anche se presentano una leggera tenden-

za ad aumentare col diminuire di τ) con un valore medio di circa 2.40 s. L'andamento completo è raffigurato in Figura 3.6b.

Affinché fosse possibile vedere nel dettaglio un risultato di declipping per il segnale (3.2), esso è stato clippato per $\tau = 4$ (Figura 3.7). Questa distorsione, dal punto di vista armonico, introduce delle componenti a frequenze più alte chiaramente visibili in Figura 3.8, che anche se di ampiezza abbastanza piccola, sono comunque udibili in campo audio.

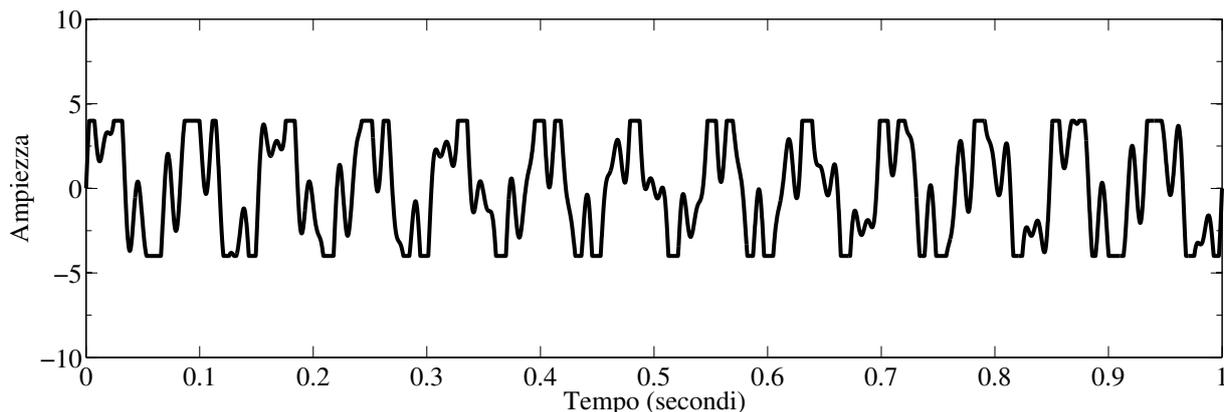


Figura 3.7: Segnale (3.2) clippato per $\tau = 4$.

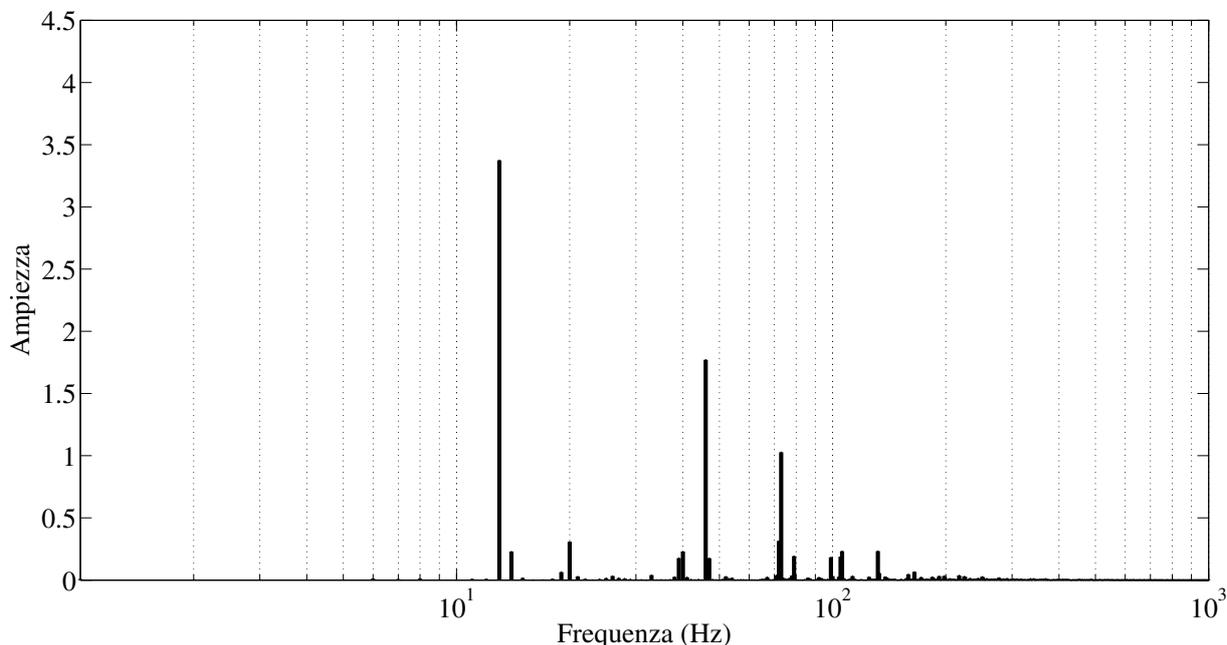


Figura 3.8: Spettro del segnale (3.2) successivamente al clipping per $\tau = 4$.

Il declipping ha notevolmente migliorato il segnale, sia per quanto riguarda l'ampiezza (Figura 3.9) sia per il contenuto armonico (Figura 3.11).

Ovviamente le armoniche spurie superiori generatesi a causa del clipping non possono essere del tutto eliminate, ma vengono comunque parecchio attenuate, mentre l'ampiezza delle armoniche del segnale originale viene ripristinata, con più o meno errore (Figura 3.10). Questo si traduce in un rapporto segnale/distorsione chiaramente migliore rispetto al segnale distorto. È possibile però notare l'effetto di *smoothing* sulle armoniche introdotto dal declipping, visibile soprattutto nell'intorno dell'armonica fondamentale di 10 Hz. Tale effetto fa quindi comparire delle armoniche, specialmente a frequenze più basse rispetto alla fondamentale, che inizialmente non erano presenti né sul segnale originale non distorto né sul segnale clippato.

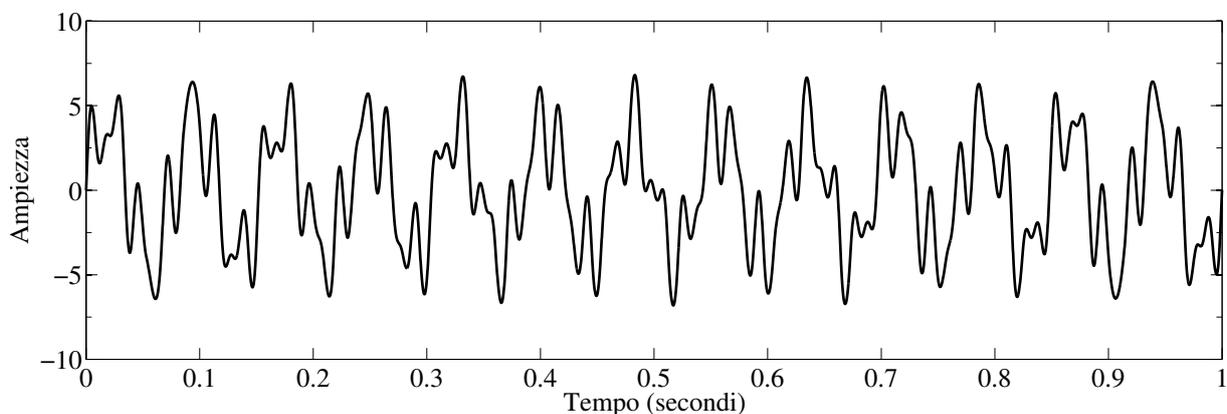


Figura 3.9: Segnale di Figura 3.7 declippato tramite CBAR.

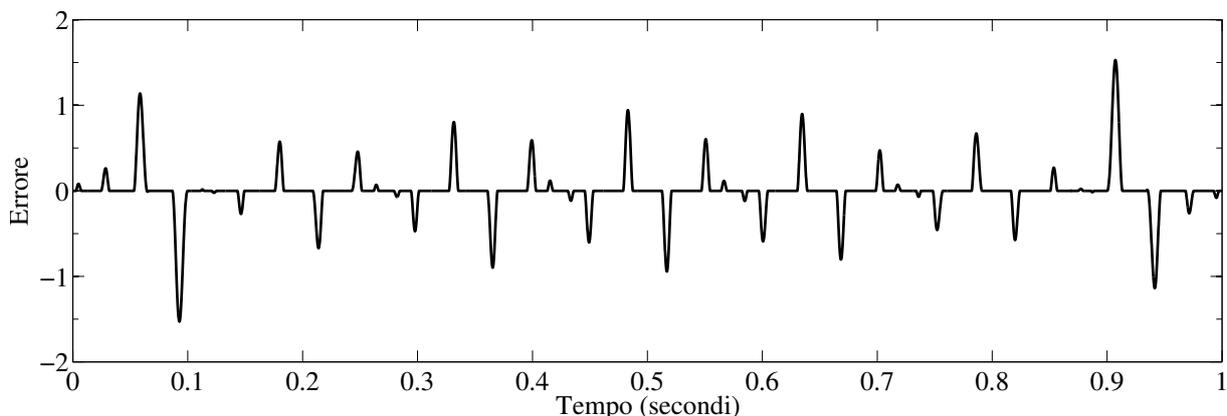


Figura 3.10: Errore risultante dal declipping tramite CBAR.

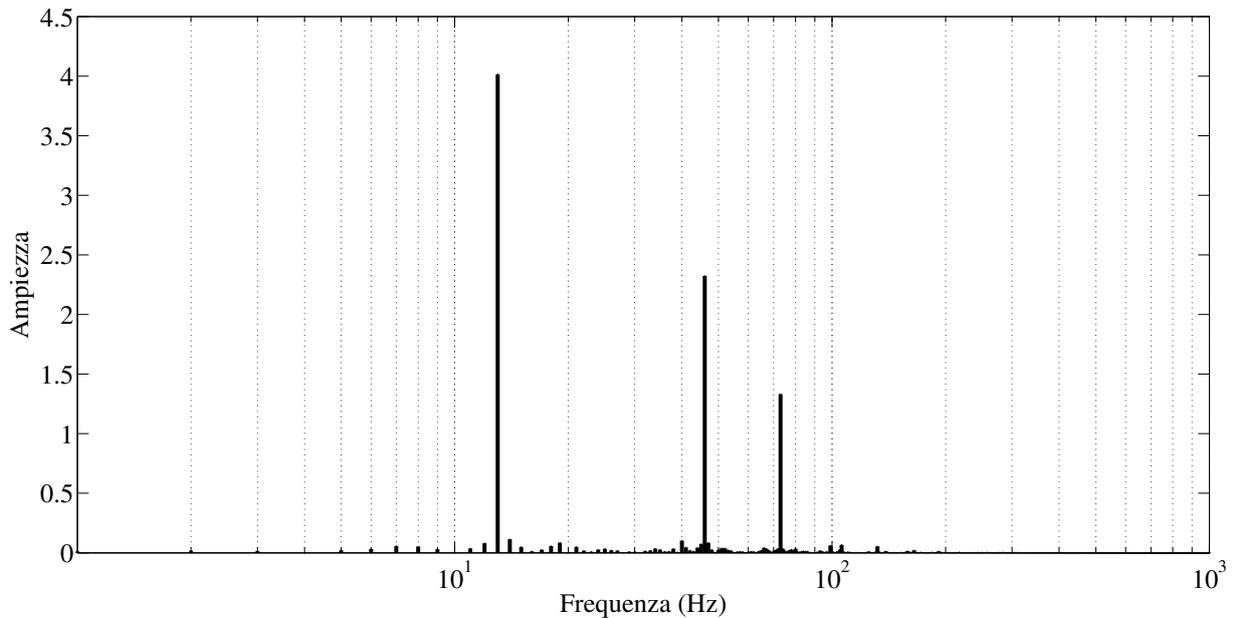


Figura 3.11: Spettro del segnale di Figura 3.7 successivamente al declipping tramite CBAR.

Declipping di un segnale audio musicale

L'ultimo test svolto si è concentrato sul declipping di un segnale audio monofonico di un secondo contenente della musica (rappresentato in Figura 3.12). Il segnale audio originale è stato campionato alla frequenza standard PCM di 44.1 kHz, ma per effettuare le prove la frequenza di campionamento è stata ridotta a 8 e 12 kHz. Per frequenze di campionamento maggiori è necessario ricorrere, almeno per quanto riguarda l'utilizzo di un calcolatore comune, all'elaborazione per frames, poiché le matrici coinvolte nel problema inverso divengono troppo grandi.

Come già fatto per tutti i test precedenti, sono stati valutati l'errore in norma-2 risultante dal declipping tramite CBAR e il tempo medio di esecuzione della ricostruzione in funzione del numero di campioni clippati.

Gli esperimenti hanno rivelato un errore in norma-2 sostanziale già per percentuali di campioni clippati basse, con un andamento dell'errore che può ricordare vagamente una curva logaritmica (Figura 3.13a).

Allo scopo di analizzare la bontà dell'algoritmo CBAR, si è andati a clippare il segnale audio di Figura 3.12 per $\tau = 0.3$. Tale valore è stato scelto perché garantisce la presenza di campioni degradati sia positivi che

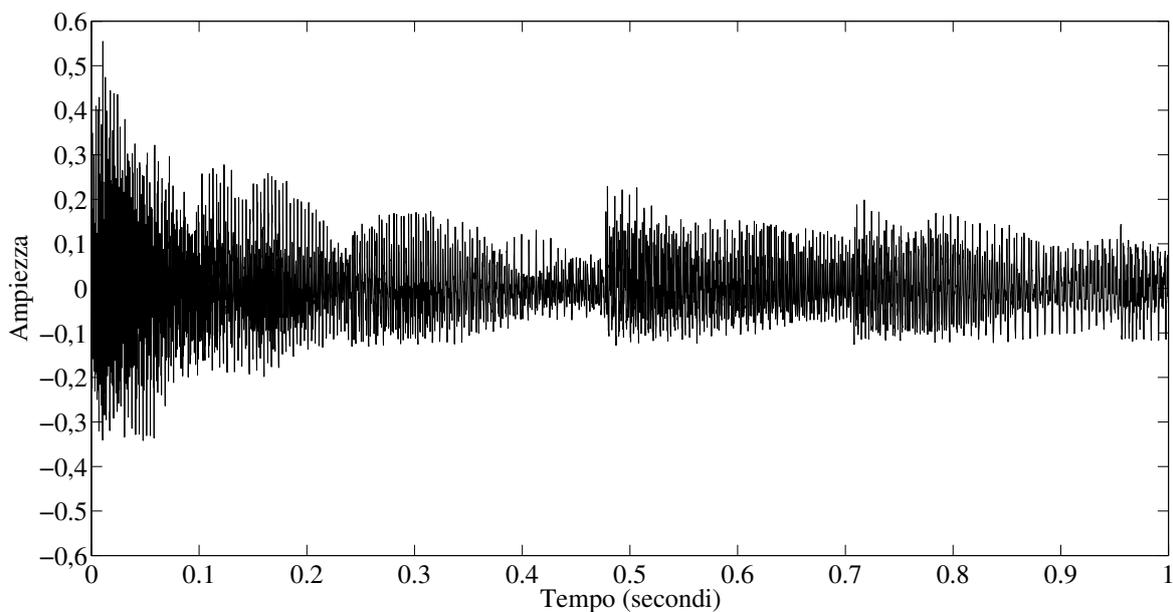
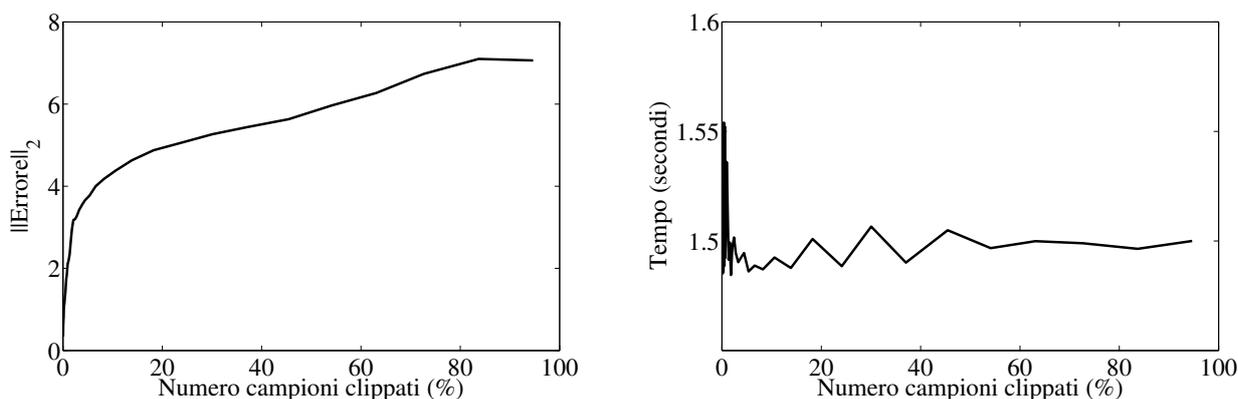


Figura 3.12: Segnale audio musicale utilizzato per i test.



(a) Norma-2 dell'errore nel declipping del segnale audio con $f_s = 8$ kHz al variare di τ .

(b) Tempi di calcolo medi al variare di τ per il segnale con $f_s = 8$ kHz.

Figura 3.13: Risultati sperimentali del CBAR su un segnale audio campionato a 8 kHz.

negativi. Il segnale distorto risultante è rappresentato in Figura 3.14.

Il risultato del declipping (Figura 3.15) mostra nuovamente il limite più grande dell'algoritmo CBAR, ovvero l'impossibilità di orientare la soluzione verso un valore desiderato. Infatti i campioni ricostruiti non riescono a superare di molto in ampiezza la soglia di clipping (arrivando massimo a un'ampiezza di 0.32). Questo di conseguenza porta a un errore abbastanza consistente, perlomeno per quanto riguarda l'ampiezza. In Figura 3.16 è rappresentato l'andamento dell'errore risultante, ristretto ai primi 60 ms,

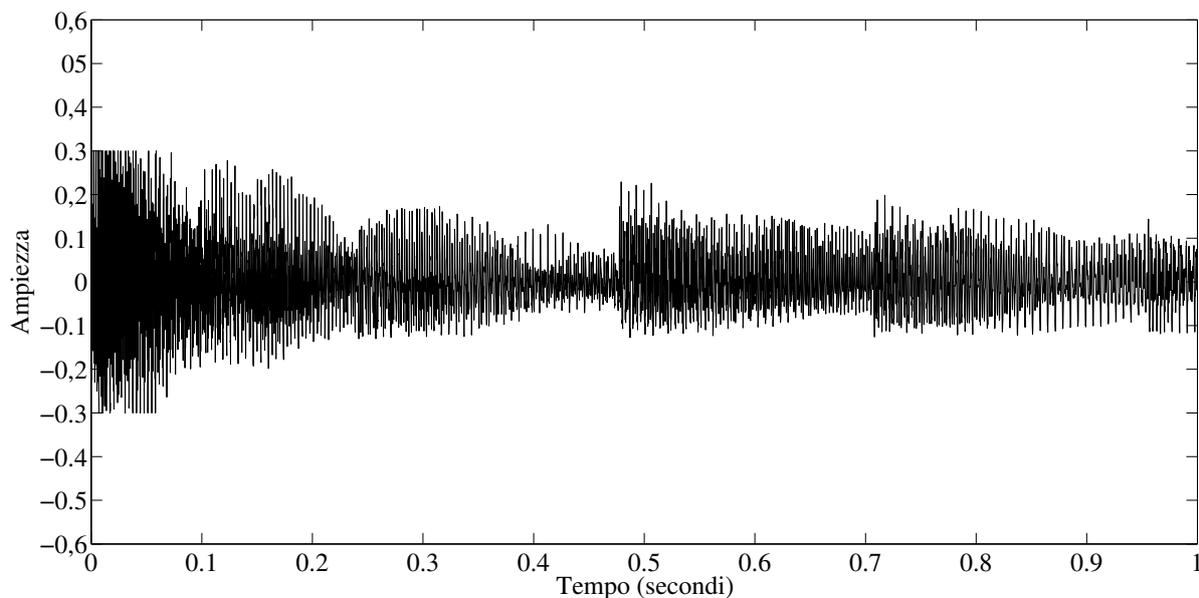


Figura 3.14: Segnale audio musicale di Figura 3.12 clippato per $\tau = 0.3$.

essendo l'intervallo di tempo in cui sono presenti i campioni clippati.

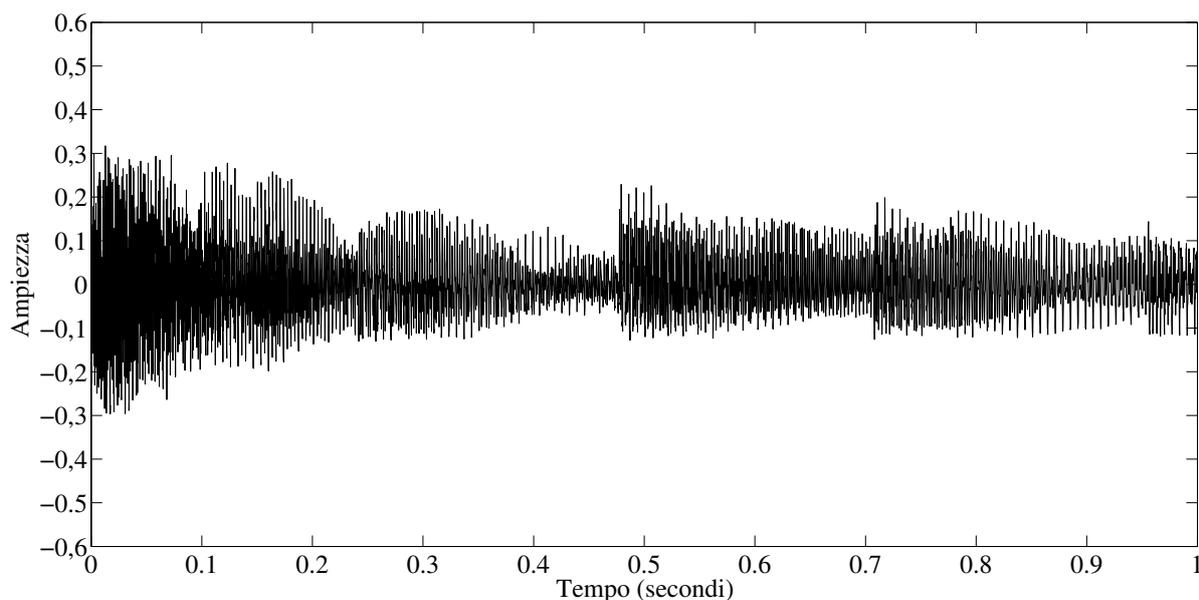


Figura 3.15: Risultato del declipping tramite CBAR del segnale di Figura 3.14.

L'utilizzo di un segnale con una frequenza di campionamento maggiore causa un aumento della norma-2 dell'errore. In Figura 3.17a è rappresentato il suo andamento al variare di τ per lo stesso segnale ma a una frequenza di campionamento di 12 kHz. Si può notare che la norma dell'errore raggiunge un valore massimo di circa 8.87, mentre la curva di Figura 3.13a

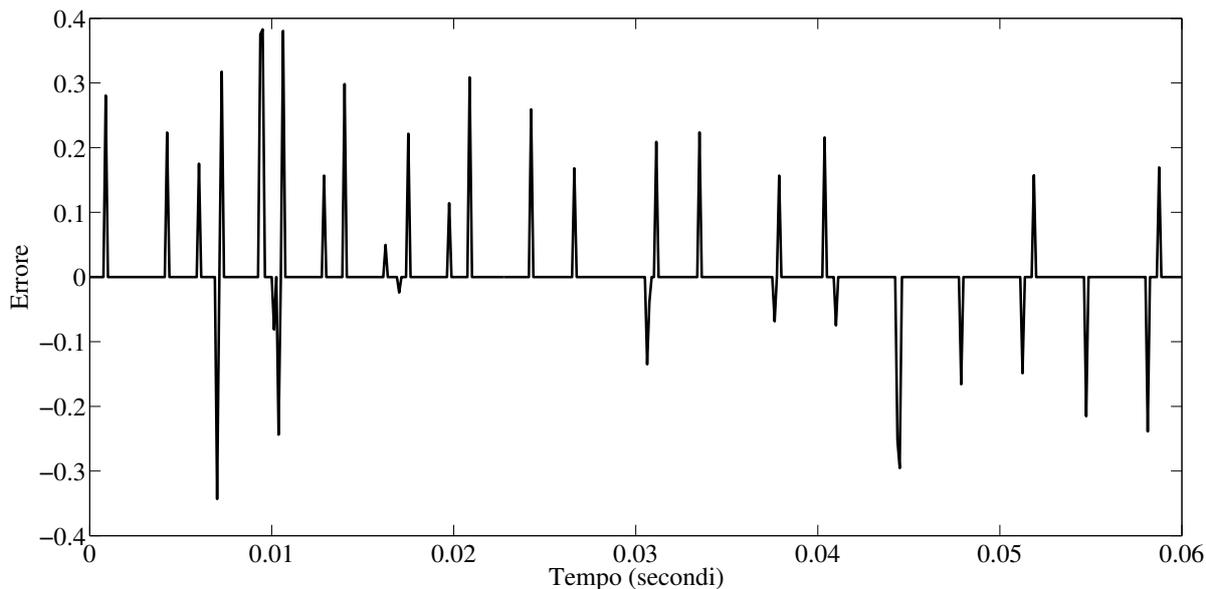
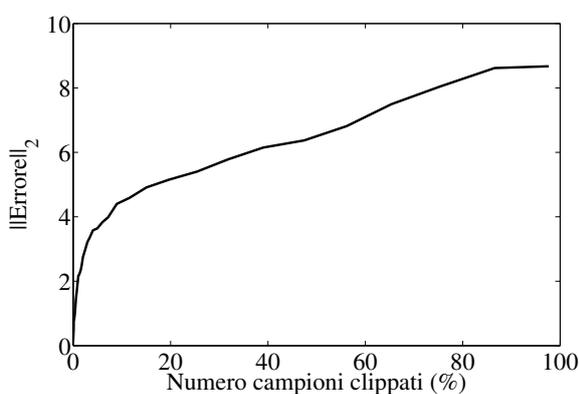
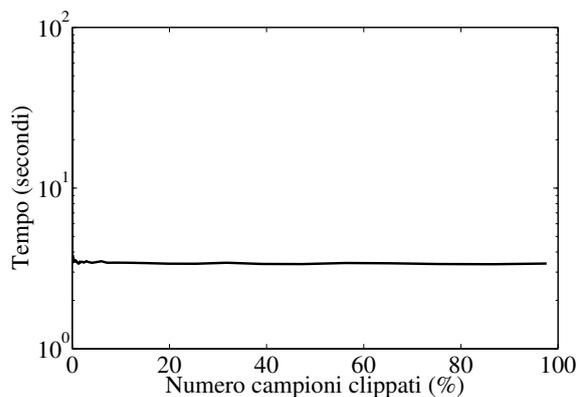


Figura 3.16: Errore risultante dal declipping tramite del segnale di Figura 3.14 ristretto ai primi 60 ms.

per il segnale campionato a 8 kHz mostrava una norma massima di 7.12 circa. Di fatto però l'aumento della frequenza di campionamento aumenta l'errore introdotto dal declipping, perché l'algoritmo ha più dati a disposizione per la risoluzione del problema inverso. Quindi possiamo affermare che maggiore è la frequenza di campionamento del segnale, minore sarà l'errore introdotto dal declipping tramite CBAR.



(a) Norma-2 dell'errore nel declipping del segnale audio con $f_s = 12$ kHz al variare di τ .



(b) Tempi di calcolo medi al variare di τ per il segnale con $f_s = 12$ kHz.

Figura 3.17: Risultati sperimentali del CBAR su un segnale audio campionato a 12 kHz.

L'aumento della frequenza di campionamento del segnale provoca, come prevedibile, un aumento dei tempi di calcolo. Come è visibile in Figura 3.17b, il tempo medio di calcolo è di circa 3.4 secondi, quindi rispetto

al segnale campionato a 8 kHz i tempi sono grosso modo raddoppiati.

3.2 Algoritmo RBAR (*Regularized Blind Amplitude Reconstruction*)

L'algoritmo RBAR (*Regularized Blind Amplitude Reconstruction*) può essere definito come una versione migliorata del CBAR, analizzato precedentemente. Come già detto, il CBAR ha il problema di essere particolarmente lento e per questo non utilizzabile per applicazioni real-time. La causa di questa lentezza è riconducibile al fatto che all'interno del problema ai minimi quadrati viene utilizzato un vincolo fisso (*hard constrain*). Si può girare intorno a questo problema introducendo all'interno del CBAR un termine di regolarizzazione, rendendo così non più necessario ricorrere a un vincolo "stretto".

Inoltre, da ciò che si è potuto osservare dai risultati sperimentali, il CBAR non permette di condurre il risultato della ricostruzione verso dei valori desiderati, perciò il declipping risultante deficiata in particolare di precisione in ampiezza. Anche per sopperire a questa mancanza, si è passati a un problema regolarizzato.

Nel nostro caso, il problema ai minimi quadrati definito per il CBAR diviene

$$\min_{\mathbf{x}_c} \|D_2 S_c^T \mathbf{x}_c - D_2 S_r^T \mathbf{x}_r\|^2 + \lambda_0^2 \|S_c^+ \mathbf{x}_c - \mathbf{t}_0\|^2 + \lambda_1^2 \|S_c^- \mathbf{x}_c - \mathbf{t}_1\|^2 ,$$

dove con S_c^+ si intende una matrice ottenuta eliminando dalla matrice identità $I \in \mathbb{R}^{C \times C}$ le righe corrispondenti ai campioni clippati negativamente. Si avrà quindi che $S_c^+ \in \mathbb{R}^{C_p \times C}$, dove C_p è il numero di campioni clippati positivamente (ovvero maggiori o uguali a $+\tau$). In modo analogo, S_c^- è ottenuta dalla matrice identità $I \in \mathbb{R}^{C \times C}$ rimuovendo le righe corrispondenti ai campioni clippati positivamente. Di conseguenza $S_c^- \in \mathbb{R}^{C_n \times C}$, con C_n numero di campioni clippati negativamente (minori o uguali a $-\tau$). [10]

I due vettori \mathbf{t}_0 e \mathbf{t}_1 sono detti vettori *target* e permettono di definire i valori desiderati attraverso la regolarizzazione.

Ponendo i due parametri di regolarizzazione $\lambda_0 = \lambda_1 = \lambda$, si ha

$$\|D_2 S_c^T \mathbf{x}_c - D_2 S_r^T \mathbf{x}_r\|^2 + \lambda^2 \left\| \begin{bmatrix} S_c^+ \\ S_c^- \end{bmatrix} \mathbf{x}_c - \begin{bmatrix} \mathbf{t}_0 \\ \mathbf{t}_1 \end{bmatrix} \right\|^2$$

e ponendo

$$\begin{bmatrix} S_c^+ \\ S_c^- \end{bmatrix} = L_m \quad e \quad \begin{bmatrix} \mathbf{t}_0 \\ \mathbf{t}_1 \end{bmatrix} = t$$

otteniamo infine

$$\min_{x_c} \|D_2 S_c^T \mathbf{x}_c - D_2 S_r^T \mathbf{x}_r\|^2 + \lambda \|L_m \mathbf{x}_c - t\|^2. \quad (3.3)$$

Ora è necessario determinare i valori da attribuire al parametro di regolarizzazione λ e ai vettori target \mathbf{t}_0 e \mathbf{t}_1 . Come già detto nel paragrafo 2.4, λ deve essere sufficientemente grande da avere un sistema stabile ma allo stesso tempo sufficientemente piccolo da avere un sistema il più simile possibile a quello iniziale e di conseguenza una soluzione adatta al problema originale. Valori tipici variano tra 10^{-9} e 10^{-2} .

Per quanto riguarda i vettori target, i valori minimi assegnabili sono pari a $+\tau$ per \mathbf{t}_0 e a $-\tau$ per \mathbf{t}_1 . È chiaro che questi valori sono troppo estremi e di conseguenza non è appropriato utilizzarli. È più che giustificato pensare di porre $\mathbf{t}_0 = (\tau + \alpha) \cdot \mathbf{1}$ e $\mathbf{t}_1 = (-\tau - \alpha) \cdot \mathbf{1}$, dove α rappresenta un incremento legato alla quantità di segnale danneggiato. Risultati soddisfacenti si sono ottenuti ponendo $\alpha = \tau \cdot \frac{C\%}{100}$, dove $C\%$ indica la percentuale di campioni distorti.

È da notare che, affinché i valori del vettore t abbiano la posizione corretta per la regolarizzazione, è necessario moltiplicare t per L_m .

Giunti a questo punto, come già detto nel paragrafo 2.4, è possibile utilizzare il metodo delle equazioni normali per poi risolvere tramite la fattorizzazione di Cholesky, ma si è preferito utilizzare il metodo della

GSVD.

Applicando la GSVD al problema (3.2), otteniamo che $A = D_2 S_c^T = U \Sigma X^{-1}$ e $L_m = V M X^{-1}$. Sostituendo nuovamente in (3.2), il problema diviene

$$\|U \Sigma X^{-1} \mathbf{x}_c - D_2 S_r^T \mathbf{x}_r\|^2 + \lambda^2 \|V M X^{-1} \mathbf{x}_c - t\|^2 .$$

Poniamo $X^{-1} \mathbf{x}_c = z$

$$\|U \Sigma z - D_2 S_r^T \mathbf{x}_r\|^2 + \lambda^2 \|V M z - t\|^2$$

e moltiplichiamo per U nella parte sinistra e per V nella parte destra

$$\|\Sigma z - U^T D_2 S_r^T \mathbf{x}_r\|^2 + \lambda^2 \|M z - V^T t\|^2 .$$

Ponendo $U^T D_2 S_r^T \mathbf{x}_r = c$ e $V^T t = d$ e facendo banali calcoli matriciali, si arriva a

$$(\Sigma^T \Sigma + \lambda^2 M^T M) z = \Sigma^T c + \lambda^2 M^T d$$

e quindi a

$$z = (\Sigma^T \Sigma + \lambda^2 M^T M)^{-1} (\Sigma^T c + \lambda^2 M^T d) ,$$

che in termini di singolo elemento della matrice z equivale a

$$z_i = \frac{\sigma_i c_i + \lambda^2 \mu_i d_i}{\sigma_i^2 + \lambda^2 \mu_i^2} .$$

Ricordando che $z = X^{-1} \mathbf{x}_c$, allora per ricavare la soluzione x_c è sufficiente moltiplicare z per X :

$$\mathbf{x}_c = X \cdot z = X \cdot X^{-1} \mathbf{x}_c = \sum_{i=1}^n x_i z_i .$$

Sostituendo il vettore \mathbf{x}_c all'interno dell'equazione (3.1), si ottiene il segnale declippato.

Dal punto di vista pratico, la regolarizzazione è stata eseguita utilizzando le routine `tikhonov` e `cgsvd` presenti all'interno del pacchetto Matlab *Regularization Tools*, scritto da Per Christian Hansen e disponibile gratuitamente presso il suo sito Web.[9]

Implementazione in pseudo codice

Algoritmo 2: Regularized Blind Amplitude Reconstruction

Dati: \mathbf{x} , τ , L , λ

Risultato: $\mathbf{x}_{declipped}$

- 1 **for** $i : 1$ **to** L **do**
- 2 **if** $|\mathbf{x}[i]| < \tau$ **then**
- 3 $\mathbf{x}_r[i] = \mathbf{x}[i]$;
- 4 **else**
- 5 $\mathbf{x}_c[i] = \mathbf{x}[i]$;
- 6 $S_r = S_c = I \in \mathbb{R}^{L \times L}$;
- 7 Rimuovi da S_r le righe corrispondenti ai campioni clippati;
- 8 Rimuovi da S_c le righe corrispondenti ai campioni intatti;
- 9 $S_c^+ = S_c^- = I \in \mathbb{R}^{C \times C}$;
- 10 Rimuovi da S_c^+ le righe corrispondenti ai campioni clippati **negativamente**;
- 11 Rimuovi da S_c^- le righe corrispondenti ai campioni clippati **positivamente**;
- 12 $A = D_2 \cdot S_c^T$;
- 13 $L_m = [S_c^+; S_c^-]^T$;
- 14 $\mathbf{b} = -D_2 S_r^T \mathbf{x}_r$;
- 15 Calcola GSVD;
- 16 Risolvi Tikhonov tramite GSVD per ottenere $\mathbf{x}_c(\text{ricostruito})$;
- 17 $\mathbf{x}_{declipped} = S_r^T \mathbf{x}_r + S_c^T \mathbf{x}_c(\text{ricostruito})$;

3.2.1 Risultati sperimentali RBAR

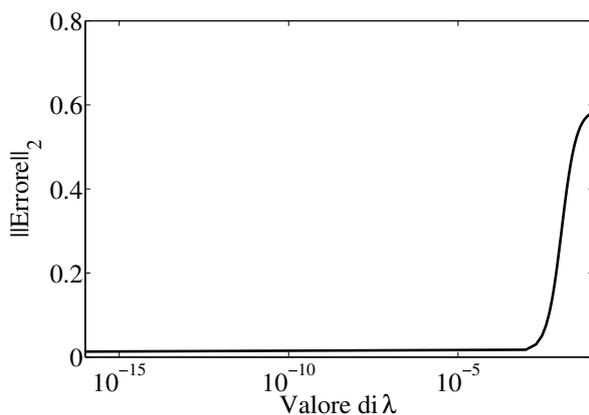
Per quanto riguarda le prove effettuate sull'algoritmo RBAR, a differenza di quanto fatto per il metodo precedente, si è deciso di fissare tre valori percentuali di campioni clippati (25%, 50% e 75%). In questo modo si è potuto bloccare uno dei gradi di libertà mantenendo comunque una certa significatività dei risultati.

I test hanno riguardato principalmente l'influenza del valore del parametro di regolarizzazione λ sulla correttezza della soluzione. Le prove eseguite

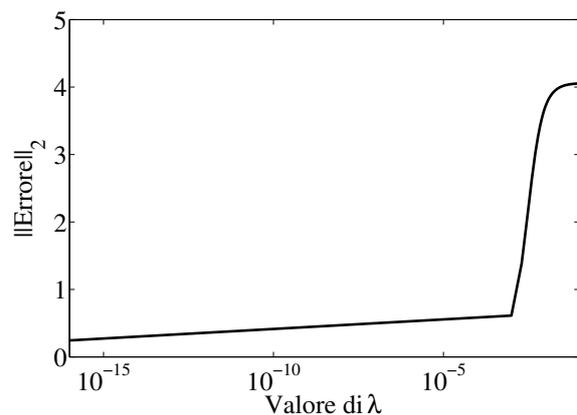
non hanno rivelato una diretta influenza di λ sui tempi di calcolo dell'algoritmo RBAR. Gli unici parametri che hanno influenzato i tempi di calcolo sono le percentuali di segnale distorto e la frequenza di campionamento dei segnali, come si è visto anche per il CBAR.

Declipping di una sinusoide con RBAR

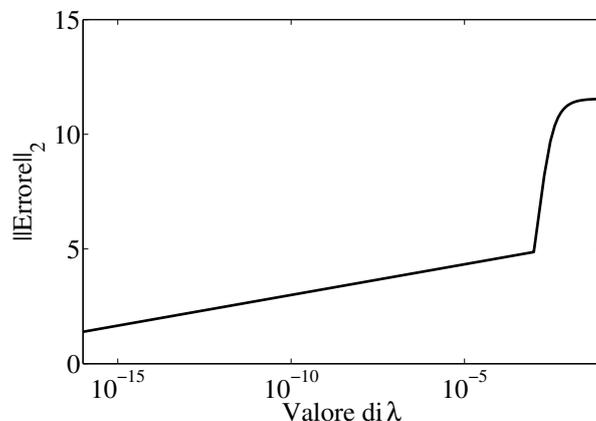
Alle tre percentuali di campioni clippati di 25%, 50% e 75% corrispondono, in questo caso, valori di τ rispettivamente pari a 0.92, 0.71 e 0.38.



(a) Norma-2 dell'errore nel declipping di una sinusoide col 25% di campioni clippati al variare di λ .



(b) Norma-2 dell'errore nel declipping di una sinusoide col 50% di campioni clippati al variare di λ .

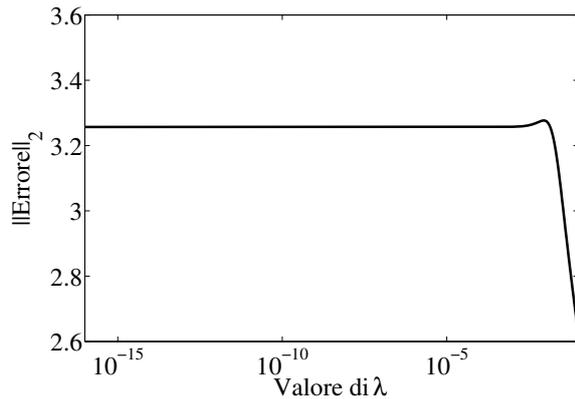


(c) Norma-2 dell'errore nel declipping di una sinusoide col 75% di campioni clippati al variare di λ .

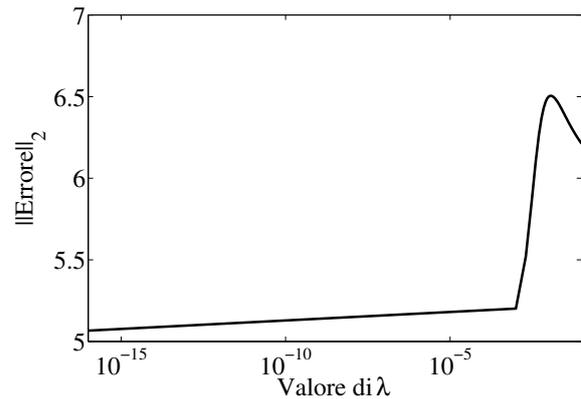
Figura 3.18: Variazione dell'errore al variare del parametro di regolarizzazione λ per una sinusoide.

Per tutte e tre le percentuali valutate, come si evince dalla Figura 3.18, si ottengono risultati migliori (in termini di errore) utilizzando valori molto piccoli del parametro di regolarizzazione λ , quindi con un problema regolarizzato di Tikhonov molto vicino al problema ai minimi quadrati originale

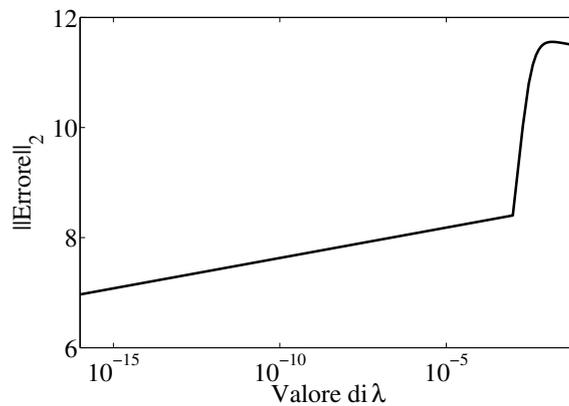
del CBAR. Questo era piuttosto prevedibile, perché si era già osservato precedentemente che l'algoritmo CBAR produce risultati molto buoni con segnali puramente sinusoidali. Perciò una regolarizzazione, anche minima come in questo caso, migliora notevolmente l'errore. Chiaramente se invece il problema viene sovra regolarizzato utilizzando un valore di λ maggiore, l'errore aumenta a causa del termine di regolarizzazione che assume un peso troppo alto nel problema.



(a) Norma-2 dell'errore nel declipping di un segnale triangolare col 25% di campioni clippati al variare di λ .



(b) Norma-2 dell'errore nel declipping di un segnale triangolare col 50% di campioni clippati al variare di λ .



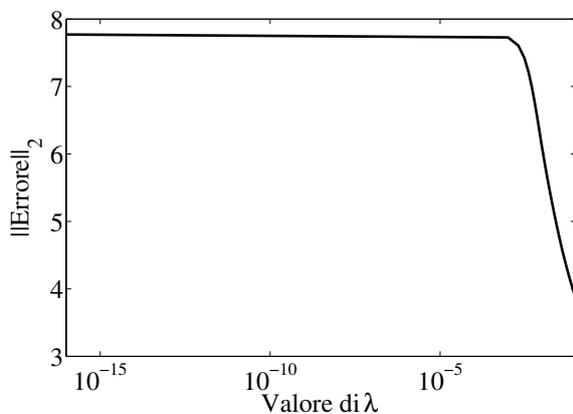
(c) Norma-2 dell'errore nel declipping di un segnale triangolare col 75% di campioni clippati al variare di λ .

Figura 3.19: Variazione dell'errore al variare del parametro di regolarizzazione λ per un segnale triangolare.

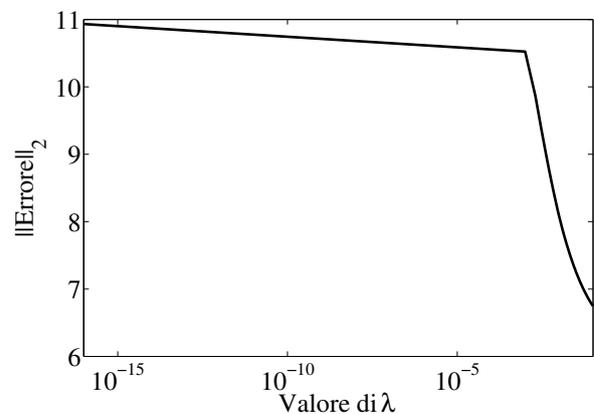
Declipping di un'onda triangolare

L'analisi della variazione della norma-2 dell'errore in funzione del parametro λ per un segnale triangolare ha mostrato un comportamento interessante. Con il segnale distorto al 25% (Figura 3.19a) l'errore minore si

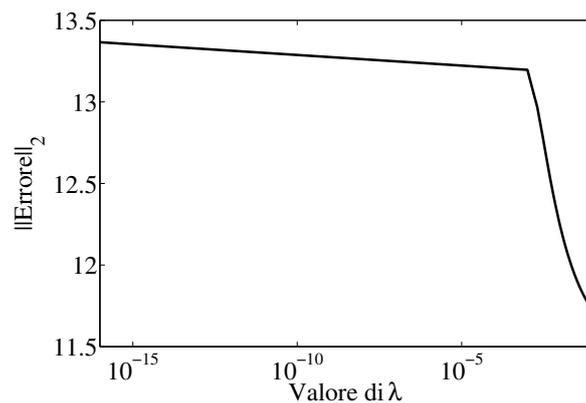
ottiene utilizzando un valore di λ abbastanza grande (0.1 o anche maggiore). Aumentando la percentuale di distorsione del segnale, il comportamento si inverte: come visibile in Figura 3.19b e 3.19c, si ottiene un errore minore con valori di λ molto piccoli (nell'ordine di 10^{-16}). Questo può essere giustificato dal fatto che il segnale triangolare presenta una punta particolarmente difficile da ricostruire, e a percentuali di distorsione basse l'algoritmo riesce ad avvicinarsi sovra regolarizzando. Con percentuali più alte invece l'errore introdotto da un parametro λ di valore alto per ricostruire la punta dell'onda triangolare è maggiore di quello ottenibile con poca regolarizzazione.



(a) Norma-2 dell'errore nel declipping di un segnale a dente di sega col 25% di campioni clippati al variare di λ .



(b) Norma-2 dell'errore nel declipping di un segnale a dente di sega col 50% di campioni clippati al variare di λ .

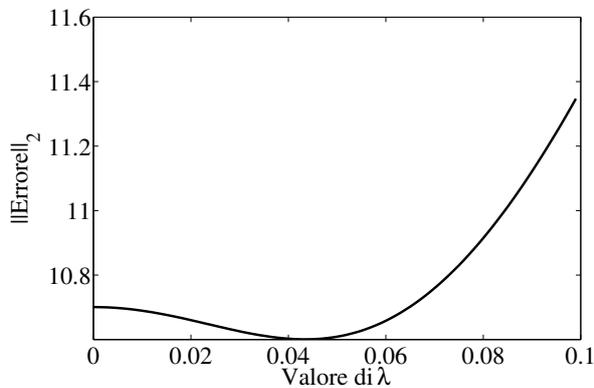


(c) Norma-2 dell'errore nel declipping di un segnale a dente di sega col 75% di campioni clippati al variare di λ .

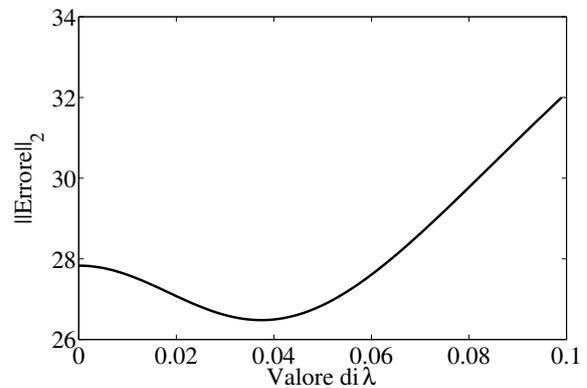
Figura 3.20: Variazione dell'errore al variare del parametro di regolarizzazione λ per un segnale a dente di sega.

Declipping di un'onda a dente di sega

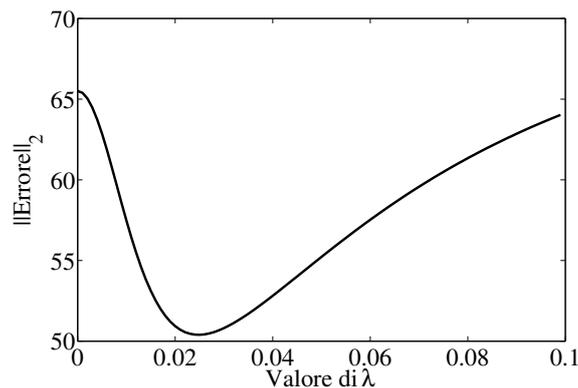
Il declipping di un'onda a dente di sega ha mostrato la necessità di una forte regolarizzazione. Il caso è molto simile a quanto visto per la Figura 3.19a; in effetti un segnale a dente di sega ha parecchie similitudini con un segnale triangolare.¹ Il motivo del valore di λ necessariamente così alto può essere ricercato, oltre che nella punta caratteristica (come già fatto per il segnale triangolare), soprattutto nella discontinuità. Si è già detto che l'algoritmo CBAR (e di conseguenza l'RBAR) non riesce a distinguere le discontinuità e quindi tenta comunque di ricostruirle. Una ricostruzione migliore della discontinuità (e quindi una diminuzione dell'errore) si può ottenere regolarizzando con un valore di λ alto.



(a) Norma-2 dell'errore nel declipping del segnale di Figura 3.1.1 col 25% di campioni clippati al variare di λ .



(b) Norma-2 dell'errore nel declipping del segnale di Figura 3.1.1 col 50% di campioni clippati al variare di λ .



(c) Norma-2 dell'errore nel declipping del segnale di Figura 3.1.1 col 75% di campioni clippati al variare di λ .

Figura 3.21: Variazione dell'errore al variare del parametro di regolarizzazione λ per il segnale (3.2).

¹Infatti su Matlab il comando per generare entrambi i segnali è lo stesso.

Declipping di un segnale complesso

Anche l'analisi della ricostruzione del segnale complesso di Figura 3.1.1 ha mostrato risultati particolarmente interessanti. In tutti e tre i casi analizzati (le tre percentuali di segnale distorto) la ricostruzione migliore si è ottenuta con un valore del parametro di regolarizzazione λ intermedio (Figura 3.21). Per le percentuali del 25 e 50%, il valore ottimo è molto simile e si attesta a circa 0.04, mentre per il segnale distorto al 75% il declipping migliore è ottenuto con $\lambda = 0.023$.

Il segnale in considerazione è ottimo per mostrare le differenze nella ricostruzione tra i due algoritmi CBAR e RBAR, perché è sufficientemente complesso da avvicinarsi a un segnale audio ma allo stesso tempo permette di vedere il declipping in maniera diretta. In Figura 3.2.1 sono rappresentati i risultati del declipping del segnale (clippato per $\tau = 3$) con i due diversi algoritmi. Si può vedere che spesso la ricostruzione effettuata dal CBAR tende a sovrastimare l'ampiezza originale del segnale, mentre l'RBAR (in questo caso con $\lambda = 0.04$) fornisce un errore minore (Figura 3.2.1).

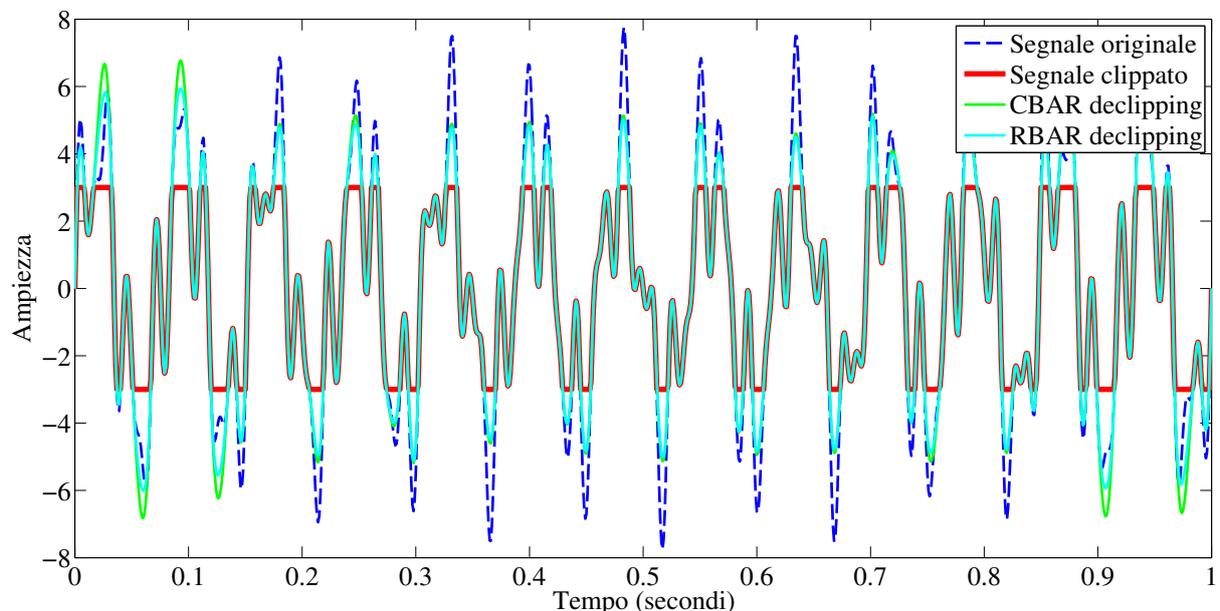


Figura 3.22: Confronto dei due metodi CBAR e RBAR nel declipping del segnale di Figura 3.1.1.

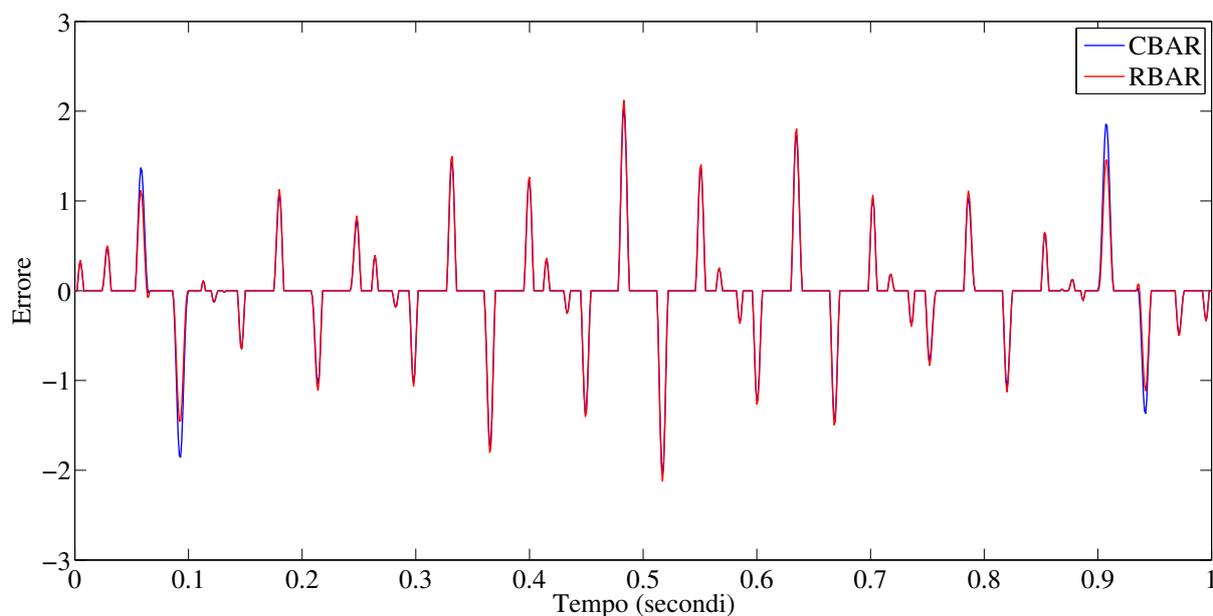


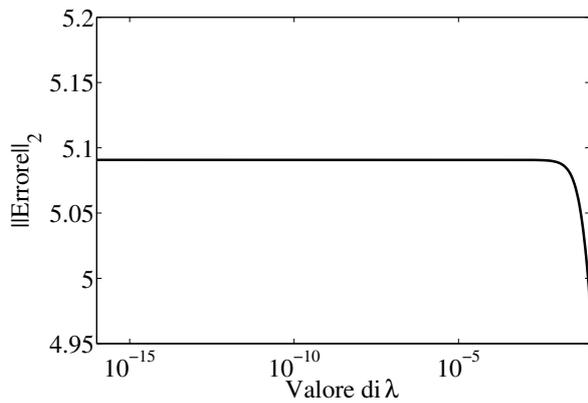
Figura 3.23: Confronto dell'errore risultante dai due metodi CBAR e RBAR nel declipping del segnale di Figura 3.1.1.

Declipping di un segnale audio musicale

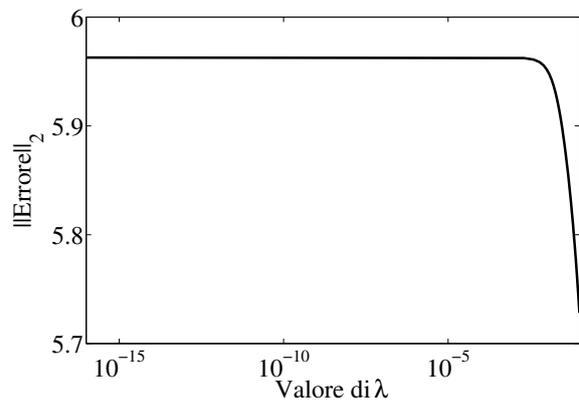
Terminiamo la serie di test con l'analisi del funzionamento nel declipping di un segnale audio musicale. Come già visto, il segnale in questione ha la durata di 1 secondo ed è stato sotto-campionato alle frequenze di 8 e 12 kHz. Anche in questo caso si sono fissate delle percentuali di campioni clippati per bloccare uno dei gradi di libertà.

I primi risultati riguardano il declipping del segnale campionato a 8 kHz. Anche in questo caso il risultato migliore in termini di errore è ottenuto con una forte regolarizzazione (Figura 3.24). Si era già detto che il limite del CBAR era l'impossibilità di regolare la direzione della soluzione. In questo modo, tramite la regolarizzazione, si è in grado di "tirare" la soluzione verso l'ampiezza originale del segnale. Poiché le informazioni sulla stima dell'ampiezza del segnale originale sono contenute nel termine di regolarizzazione, è chiara l'esigenza di dare maggior peso a quest'ultimo per avere un errore minore.

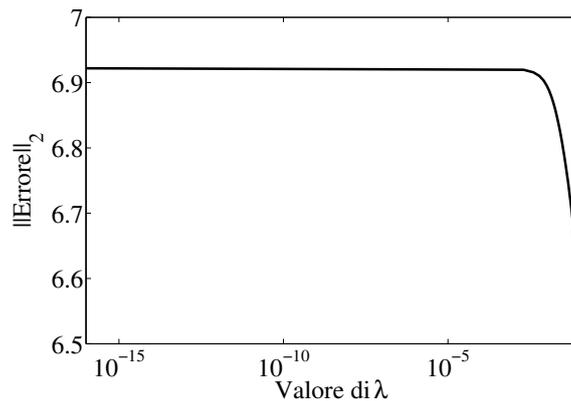
I test eseguiti al segnale campionato a 12 kHz hanno dimostrato l'assoluta impossibilità di lavorare con questo algoritmo con un calcolatore comune



(a) Norma-2 dell'errore nel declipping del segnale col 25% di campioni clippati al variare di λ .



(b) Norma-2 dell'errore nel declipping del segnale col 50% di campioni clippati al variare di λ .

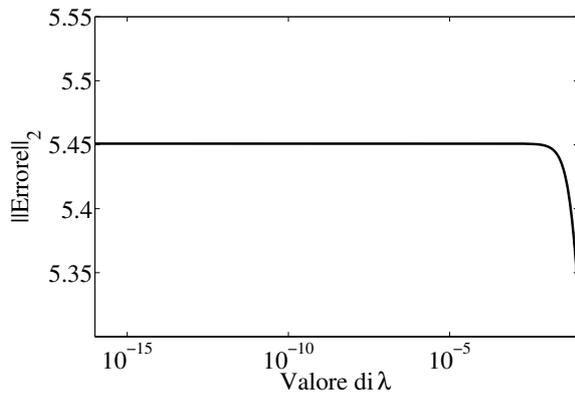


(c) Norma-2 dell'errore nel declipping del segnale col 75% di campioni clippati al variare di λ .

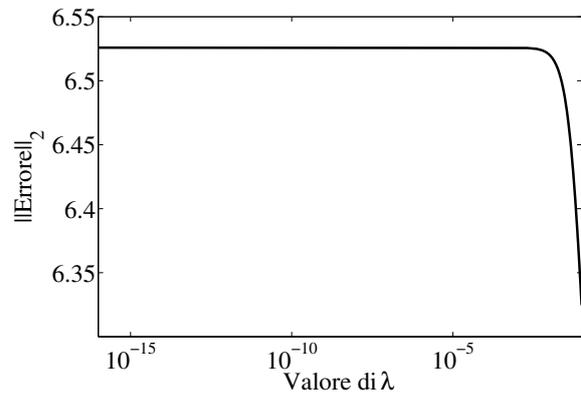
Figura 3.24: Variazione dell'errore al variare del parametro di regolarizzazione λ per il segnale campionato a 8 kHz.

senza ricorrere all'elaborazione in frames. Per questi ultimi risultati non si è potuto fare a meno di suddividere il segnale in 10 frames di 100 ms, ognuno contenente 1200 campioni e con un leggero overlap, in modo da ottenere un risultato più preciso agli estremi dei frames. Infatti le sole valutazioni degli errori introdotti dal restauro al variare di λ avrebbero richiesto un tempo variabile tra le 5 e le 7 ore, con il comune calcolatore utilizzato per questa tesi. In Figura 3.25 sono rappresentati i tre andamenti dell'errore al variare di λ . Come già visto per il caso del CBAR, l'aumento della frequenza di campionamento aumenta la norma-2 dell'errore ma l'errore risultante dalla ricostruzione è minore.

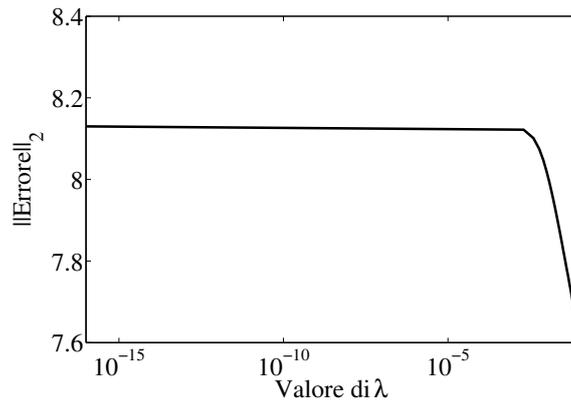
Infine, si è provato il metodo sul segnale distorto di Figura 3.14. Tale segnale era stato opportunamente clippato per $\tau = 0.3$, e con l'algoritmo



(a) Norma-2 dell'errore nel declipping del segnale col 25% di campioni clippati al variare di λ .



(b) Norma-2 dell'errore nel declipping del segnale col 50% di campioni clippati al variare di λ .



(c) Norma-2 dell'errore nel declipping del segnale col 75% di campioni clippati al variare di λ .

Figura 3.25: Variazione dell'errore al variare del parametro di regolarizzazione λ per il segnale campionato a 12 kHz.

CBAR non si era riusciti a ricostruire le ampiezze in maniera soddisfacente, restando nell'intorno della soglia di clipping. Invece, tramite l'RBAR, come è possibile osservare in Figura 3.26, si è riusciti a ripristinare maggiormente le ampiezze, nonostante non siano comunque al livello del segnale originale. In ogni caso il risultato è abbastanza soddisfacente rispetto al metodo precedente. In Figura 3.27 è rappresentato l'andamento dell'errore dovuto al declipping, ristretto ai primi 60 ms contenenti i campioni danneggiati.

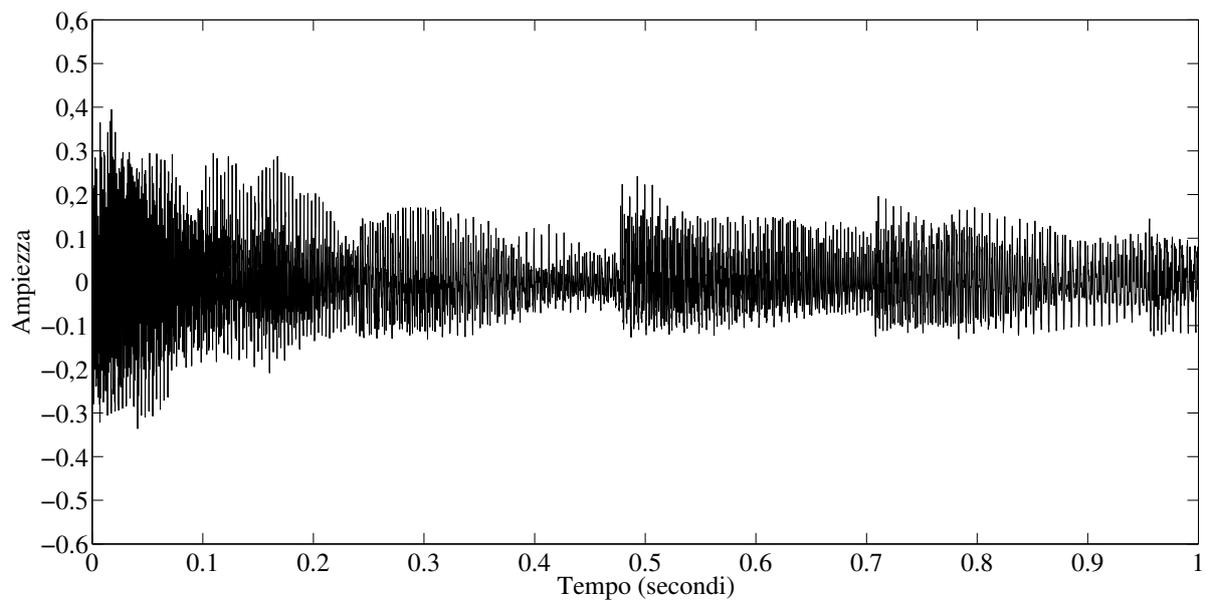


Figura 3.26: Risultato del declipping tramite CBAR del segnale di Figura 3.14.

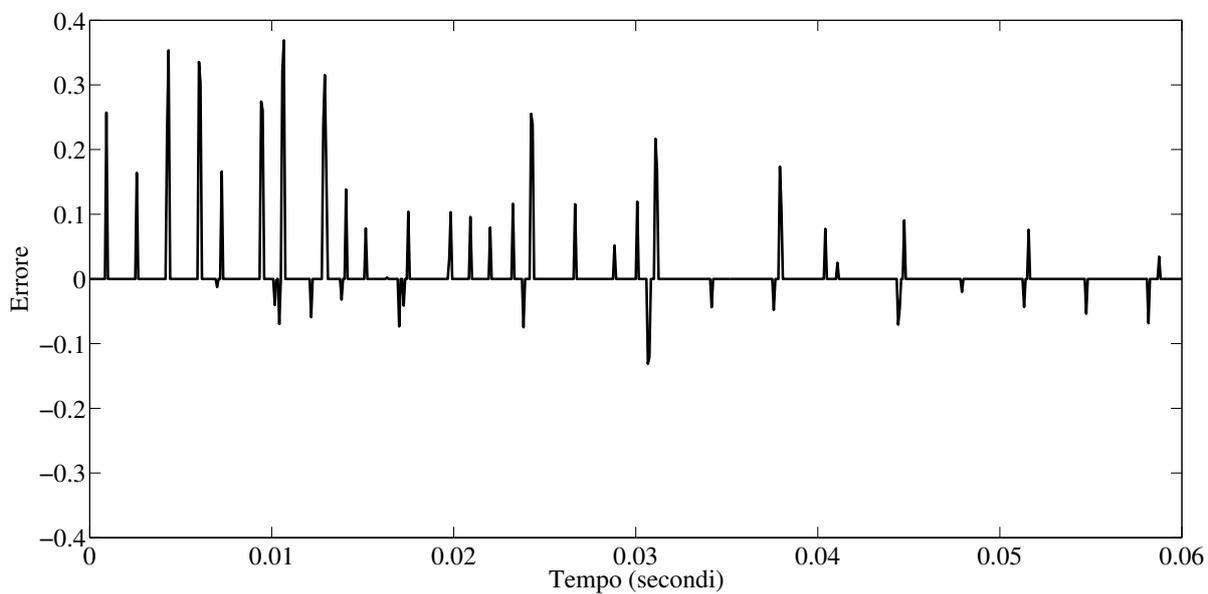


Figura 3.27: Errore risultante dal declipping tramite RBAR del segnale di Figura 3.14 ristretto ai primi 60 ms.

Capitolo 4

Conclusioni

I metodi basati su ricostruzione tramite problemi ai minimi quadrati che si sono analizzati hanno mostrato un funzionamento più che soddisfacente. Attualmente lo stato dell'arte, per quanto riguarda la precisione della ricostruzione, è rappresentato da algoritmi di *iterative hard thresholding* (in breve IHT)[11]. Tali algoritmi, nonostante riescano a ricostruire in maniera ottima la parte di segnale mancante, presentano tempi di calcolo molto lunghi. Un algoritmo come l'RBAR analizzato all'interno di questa tesi, anche se meno preciso di uno IHT, riesce a effettuare un declipping in tempi molto brevi, specialmente se l'elaborazione avviene per frames molto piccoli.

Quel che impedisce all'algoritmo RBAR di avere una precisione maggiore, almeno per le analisi effettuate in questa tesi, è la stima parecchio approssimativa dell'ampiezza del segnale originale. L'utilizzo di tecniche sviluppate appositamente porterebbe certamente a risultati migliori in termini di precisione in ampiezza.

Bibliografia

- [1] E. ANDERSON, Z. BAI, C. BISCHOF, L. S. BLACKFORD, J. DEMMEL, J. DONGARRA, J. DU CROZ, A. GREENBAUM, S. HAMMARLING, A. MCKENNEY, ET AL., *LAPACK Users' guide*, SIAM, 1999.
- [2] F. R. ÁVILA, M. P. TCHEOU, AND L. W. BISCAINHO, *Audio soft declipping based on constrained weighted least squares*, IEEE signal processing letters, 24 (2017), pp. 1348–1352.
- [3] A. BJORCK, *Numerical methods for least squares problems*, Siam, 1996.
- [4] F. DENG, C.-C. BAO, AND F. BAO, *Clipping detection of audio signals based on kernel fisher discriminant*, in Signal and Information Processing (ChinaSIP), 2013 IEEE China Summit & International Conference on, IEEE, 2013, pp. 99–103.
- [5] L. ELDÉN, *Algorithms for the regularization of ill-conditioned least squares problems*, BIT Numerical Mathematics, 17 (1977), pp. 134–145.
- [6] G. GOLUB AND W. KAHAN, *Calculating the singular values and pseudo-inverse of a matrix*, Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis, 2 (1965), pp. 205–224.
- [7] J. HADAMARD, *Lectures on Cauchy's problem in linear partial differential equations*, Courier Corporation, 2014.

- [8] P. C. HANSEN, *Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion*, vol. 4, Siam, 2005.
- [9] P. C. HANSEN, *Regularization Tools: version 4.0 for Matlab 7.3*, Numerical Algorithms, 46 (2007), pp. 189–194. Il pacchetto regtools è disponibile gratuitamente su <http://www2.compute.dtu.dk/~pcha/Regutools/>.
- [10] M. J. HARVILLA AND R. M. STERN, *Efficient audio declipping using regularized least squares*, in Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on, IEEE, 2015, pp. 221–225.
- [11] S. KITIC, L. JACQUES, N. MADHU, M. P. HOPWOOD, A. SPRIET, AND C. DE VLEESCHOUWER, *Consistent iterative hard thresholding for signal declipping*, in Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, IEEE, 2013, pp. 5939–5943.
- [12] C. LAGUNA AND A. LERCH, *An efficient algorithm for clipping detection and declipping audio*, in Audio Engineering Society Convention 141, Audio Engineering Society, 2016.
- [13] C. PAIGE, *Computing the generalized singular value decomposition*, SIAM Journal on Scientific and Statistical Computing, 7 (1986), pp. 1126–1146.
- [14] G. RODRIGUEZ AND S. SEATZU, *Introduzione alla Matematica Applicata e Computazionale*, Pitagora Editrice Bologna, 2010.
- [15] A. N. TIKHONOV, A. GONCHARSKY, V. STEPANOV, AND A. G. YAGOLA, *Numerical methods for the solution of ill-posed problems*, vol. 328, Springer Science & Business Media, 2013.