



UNIVERSITÀ DEGLI STUDI DI PADOVA

FACOLTÀ DI INGEGNERIA

Corso di Laurea in Ingegneria Biomedica

**IMPLEMENTAZIONE DI UN ALGORITMO PER IL
RECUPERO DI CAMPIONI MANCANTI IN UN
SEGNALE SPARSO**

Laureando

Federica Nardelli

Relatore

Prof. Giada Giorgi

ANNO ACCADEMICO 2015/2016

Alla mia famiglia...

Indice

1	Introduzione	1
2	Metodo del gradiente	3
2.1	Applicazione del metodo per funzione ad una variabile	3
2.1.1	Algoritmo	4
2.1.2	Condizioni di arresto	5
2.1.3	Esempio	5
2.1.4	Implementazione metodo del gradiente	6
3	Ricostruzione di segnali sparsi	8
3.1	Trasformata discreta di Fourier (DFT)	8
3.1.1	Algoritmi FFT(Fast Fourier Transform)	10
3.2	Norma l_p	10
3.2.1	Uso della norma-uno	10
3.2.2	Implementazione norma l_1	11
3.3	Mean Absolute Error (MAE)	13
3.3.1	Implementazione MAE	13
3.4	Algoritmo di ricostruzione	14
3.4.1	Algoritmo	14
3.4.2	Condizioni di arresto	15
3.4.3	Implementazione algoritmo di ricostruzione	15
4	Esempi e applicazioni	19
4.1	Campionamento coerente	19
4.1.1	Esempio campionamento coerente e risultati	20
4.2	Campionamento non coerente	21
4.2.1	Esempio campionamento non coerente e risultati	21
4.3	Finestratura	23
4.3.1	Esempio con finestra di Hanning e risultati	25
4.4	Segnali reali	28
4.4.1	Esempio con segnale biomedico e risultati	28

5 Conclusioni	30
Bibliography	30

Elenco delle figure

2.1	Applicazione metodo del gradiente	6
3.1	Rappresentazione grafica del segnale (3.10) nel dominio del tempo e della trasformazione.	12
4.1	Ricostruzione per il campionamento coerente	20
4.2	MAE relativo alla ricostruzione del segnale (campionamento coerente)	21
4.3	Ricostruzione per il campionamento non coerente	22
4.4	MAE per il campionamento non coerente	23
4.5	Esempio	24
4.6	Spettro della finestra di Hanning.	25
4.7	Ricostruzione per il campionamento coerente	26
4.8	Confronto MAE tra segnale $x'(n)$ (4.1) con e senza la finestra di Hanning	27
4.9	Errore residuo di ricostruzione tra il segnale originale e quello ricostruito privato della finestra di Hanning, valutato su una porzione centrale.	27
4.10	Ricostruzione per il campionamento coerente	29

Sommario

In questa tesi verrà analizzato un algoritmo per la ricostruzione di campioni mancanti o non disponibili in un segnale con rappresentazione sparsa nel dominio della frequenza mediante una tecnica che rientra nell'ambito del Compressive Sensing (CS), e la sua applicazione a diversi esempi numerici. Il CS si basa su due assunzioni: la possibilità di rendere sparso il segnale attraverso una qualche trasformazione, e l'alto grado di incoerenza tra questa trasformazione e il metodo con cui i dati vengono raccolti. Nella procedura per il recupero dei dati si deve far fronte ad un problema di ottimizzazione convessa, risolvibile mediante algoritmi di ricerca del minimo della funzione norma l_1 . La tecnica presentata è stata implementata e applicata a diversi segnali. Inizialmente si considererà un segnale campionato in modo coerente; a seguire il caso di un campionamento non coerente, sottoposto poi ad un'operazione di finestratura (*smoothing windows*). Infine si presenterà brevemente la ricostruzione del segnale di un accelerometro rilevato da acquisizioni in ambito biomedico, senza discutere nel dettaglio la trattazione dei segnali reali biomedici.

Capitolo 1

Introduzione

In molte applicazioni di elaborazione dei segnali risulta fondamentale la fase di compressione dei dati acquisiti. L'acquisizione viene solitamente effettuata rispettando le condizioni imposte dal teorema di Nyquist-Shannon, in modo da evitare il fenomeno dell'*aliasing* e le distorsioni che ne conseguono in fase di ricostruzione. Secondo il teorema del campionamento di Shannon è necessario che i campioni siano distanziati fra loro in maniera opportuna, con un passo di campionamento che risulta essere l'inverso della frequenza di Nyquist.

Recentemente è stato introdotto un nuovo paradigma di acquisizione/ricostruzione che va oltre i limiti imposti dallo schema classico del campionamento secondo Shannon: questo viene chiamato Compressed Sensing o Compressive Sampling (CS). Il CS sfrutta determinate proprietà dei segnali naturali per comprimere in maniera alternativa: invece di acquisire un set completo di campioni e poi scartarne la maggior parte in fase di compressione, si effettua direttamente un'acquisizione con molti meno campioni e si sfruttano determinati algoritmi per ricostruire il segnale senza perdita di informazione. Con questo metodo, quindi, è possibile ricostruire un segnale mediante un sottocampionamento, riducendo al minimo l'errore che deriva dal fatto di avere un numero insufficiente di campioni. Ciò è reso possibile sfruttando alcune proprietà matematiche dei segnali che devono essere verificate:

- Sparsità del segnale da analizzare: se si considera un segnale discreto limitato nel tempo, si dice che esso è sparso in un arbitrario dominio di trasformazione se esso contiene un numero molto minore di campioni (coefficienti) diversi da zero in questo dominio. I coefficienti di trasformazione del segnale rimanenti si presume siano uguali a zero o approssimabili a zero senza commettere errori significativi.

- Incoerenza del campionamento: l'approccio del CS richiede che l'acquisizione venga fatta seguendo l'incoerenza del segnale fra il dominio di acquisizione la trasformata di sparsificazione. Si dimostra che, se viene rispettata l'incoerenza fra dominio di acquisizione e di concentrazione, è possibile acquisire secondo un criterio prestabilito, quindi indipendentemente dall'andamento del segnale considerato.

Vi sono diversi approcci per ricostruire questo tipo di segnali. La tesi tratterà la ricostruzione di campioni mancanti di segnali sparsi mediante un algoritmo basato sul gradiente a passo costante; questo algoritmo utilizzerà una misura di concentrazione del segnale (sparsità) come funzione di minimizzazione nella ricostruzione. Verranno commentati brevemente i concetti matematici su cui si basa l'algoritmo; successivamente saranno presentati diversi esempi di applicazione di ricostruzione dei segnali e ne verranno analizzati i risultati.

Capitolo 2

Metodo del gradiente

Il metodo del gradiente è un algoritmo di ricerca dei punti di massimo e minimo di una funzione f , almeno di classe C^1 . Tale metodo si basa sullo studio del gradiente di f in una successione di punti del dominio di questa. Il gradiente ∇f permette di individuare la direzione di massima variazione della funzione f in ogni suo punto; mentre nei punti di massimo e di minimo della funzione o, più in generale, nei punti critici, si ha che ∇f si annulla. Si noti che il metodo del gradiente non garantisce di trovare un massimo o un minimo assoluto per f , ma solamente un punto critico di f , per cui il gradiente si annulla. Tale problema non si pone nel caso della ricerca del minimo di una funzione convessa, poichè se un punto è di minimo relativo per questa, allora il punto è anche di minimo assoluto per la funzione.

2.1 Applicazione del metodo per funzione ad una variabile

Data una funzione $f(x)$ differenziabile, si vuole cercare il minimo di tale funzione:

$$x^* = \operatorname{argmin}_{x \in \mathbf{R}} f(x) \quad (2.1)$$

Per semplicità il metodo viene presentato per funzioni ad una variabile. Per un problema di ricerca del minimo della funzione è possibile procedere in questo modo:

1. scegliere un punto di partenza x_0 ;
2. calcolare $\nabla f(x_0)$ che, nel caso di funzione ad una variabile, vale $\frac{df(x)}{dx}$;

3. muoversi lungo la direzione opposta, o direzione di *antigradiente*, di un opportuno passo μ ;
4. se nel nuovo punto il gradiente non si annulla, iterare il procedimento dei punti 1-3 partendo da questo.

2.1.1 Algoritmo

Una volta fissato x_0 , è possibile approssimare la funzione $f(x)$ nell'intorno del punto x_0 con il suo sviluppo di Taylor troncato al termine di primo grado:

$$\tilde{f}(x) = f(x_0) + \left. \frac{\partial f(x)}{\partial x} \right|_{x=x_0} \cdot (x - x_0) \quad (2.2)$$

La funzione approssimante $\tilde{f}(x)$ rappresenta l'equazione della retta passante per il punto $(x_0, f(x_0))$ ed avente pendenza $\left. \frac{\partial f(x)}{\partial x} \right|_{x=x_0}$.

Dalla teoria sullo studio di funzioni, si utilizza il seguente teorema:

Teorema 1 [Monotonia di una funzione in un intervallo]

Siano I intervallo di \mathbb{R} e $f : I \rightarrow \mathbb{R}$ funzione derivabile. Allora f si dice monotona crescente (risp. decrescente) se e solo se $f'(x) \geq 0$ (risp. $f'(x) \leq 0$) per ogni $x \in I$.

Si ricava, quindi, che:

- se f è *crescente*, al crescere delle ascisse crescono le corrispondenti ordinate, cioè

$$\begin{aligned} x > x_0 &\Rightarrow f(x) > f(x_0) \\ (\text{viceversa, } x < x_0 &\Rightarrow f(x) < f(x_0)); \end{aligned}$$

- se f è *decrescente*, al crescere delle ascisse decrescono le corrispondenti ordinate, cioè

$$\begin{aligned} x > x_0 &\Rightarrow f(x) < f(x_0) \\ (\text{viceversa, } x < x_0 &\Rightarrow f(x) > f(x_0)); \end{aligned}$$

Poiché l'obiettivo è trovare il punto di minimo x^* , analizzando la derivata prima calcolata in x_0 è possibile ricavare un criterio per individuare, procedendo

in modo iterativo, il punto di minimo (o quantomeno una sua approssimazione accettabile).

Una volta stabilito in quale insieme di valori procedere con la ricerca, risulta necessario fissare un criterio per cercare il punto da considerare nell'iterazione successiva. Si fissa in modo (quasi) arbitrario un parametro μ , definito *step size*; il punto successivo x_1 viene quindi calcolato come:

$$x_1 = x_0 - \mu \cdot \left. \frac{\partial f(x)}{\partial x} \right|_{x=x_0} \quad (2.3)$$

La ricerca riparte considerando il punto x_1 come nuovo punto iniziale.

Dopo aver fissato il nuovo punto di partenza, si deve verificare di non essere in una situazione di divergenza; ciò accade quando, ad esempio, vengono scelti valori troppo elevati per il parametro μ . Si calcola quindi il valore della funzione in x_1 , ovvero $f(x_1)$, e si controlla se la condizione corretta $f(x_1) < f(x_0)$ è effettivamente verificata; in caso contrario, l'algoritmo si trova in una fase divergente o oscillatoria.

2.1.2 Condizioni di arresto

L'algoritmo trova il minimo della funzione f quando la derivata diventa nulla. Tuttavia, tale condizione è quasi impossibile da raggiungere; pertanto, risulta necessario accettare un'approssimazione del minimo che si ottiene, fermando l'algoritmo di ricerca quando la derivata è circa pari a zero. Si definiscono quindi le seguenti condizioni di stop:

- se $-\epsilon_a < \left. \frac{\partial f(x)}{\partial x} \right|_{x=x_i} < \epsilon_a$: è stato raggiunto un punto critico;
- se $|x_i - x_{i-1}| < \epsilon_a$: si ha convergenza ad un valore di x ;
- se $i > I_{max}$, : il numero i di iterazioni ha raggiunto il numero massimo fissato I_{max} .

2.1.3 Esempio

Ad esempio, considerando la funzione convessa $f(x) = x^2 - 2x$, si ricava in 8 iterazioni il punto di minimo di f pari a 1.0046, che, come si nota, è in buon accordo con il valore di minimo teorico ricavato mediante studio di funzione¹,

¹ $f'(x) = 2x - 2$; $f'(x) = 0 \implies x = 1$.

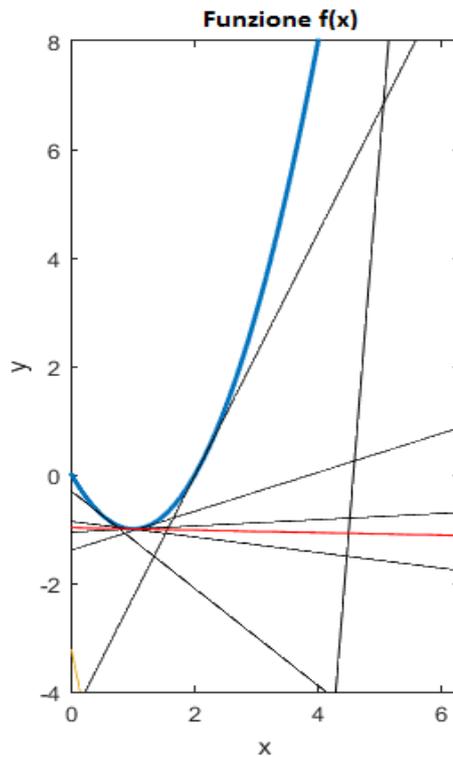


Figura 2.1: Applicazione metodo del gradiente

ponendo i seguenti parametri:

- punto iniziale: $x_0 = 8$,
- *step size*: $\mu = 0.7$,
- approssimazione risultato: $\sigma = 0.02$.

Le rette raffigurate in figura (2.1) rappresentano le derivate dei punti considerati ad ogni iterazione, ovvero le rette tangenti alla funzione in ciascun punto. Nella sezione seguente è presentato l'algoritmo utilizzato.

2.1.4 Implementazione metodo del gradiente

Listing 2.1: Metodo del gradiente

```

1 % Metodo del gradiente
2 % Ricerca del punto di minimo di una funzione convessa
3
4 function [minimo, iter] = gradient_descend(x0)

```

```

5
6 % Parametri e inizializzazioni delle variabili
7 x = sym('x');           % Variabile simbolica
8 f = x^2-2*x;            % Funzione considerata
9 fx = diff(f,x);        % Derivata di f
10 m = subs(fx,x0);       % Calcolo derivata nel punto x0
11 y0 = subs(f,x0);       % f(x0)
12 r = m*(x-x0) + y0;     % Retta tangente a f in x0
13
14 %Definisco le variabili e i parametri
15 minimo = 0;
16 Imax = 100;
17 iter = 0;
18 mou = 0.7;
19 epsilon = 0.01;
20 sigma = 0.02;
21 x1 = x0;
22 x2 = x1;
23 while(iter < Imax)     % Condizione di stop
24     minimo = x2;
25     if(abs(m) < epsilon) % Condizione di stop
26         minimo = x2;
27         break;
28     end
29     x2 = x1 - mou*subs(fx,x1);
30     y1 = subs(f,x1);
31     y2 = subs(f,x2);
32     iter = iter+1;
33     if(y2 - y1 > 0)
34         disp('L'algoritmo e' in fase divergnete o
35             oscillatoria: step size troppo grande');
36         minimo = NaN;
37         break;
38     end
39     if((abs(x2 - x1)) < sigma) % Condizione di stop
40         disp('Convergenza ad un valore di x');
41         minimo = x2;
42         break;
43     end
44     m = subs(fx,x2);
45     x1 = x2;
46 end
47 end

```

Capitolo 3

Ricostruzione di segnali sparsi

In questo capitolo verranno inizialmente forniti alcuni cenni sulla trasformata discreta di Fourier, sulle misure di concentrazione di un segnale sparso e verrà proposto un metodo di confronto dell'accuratezza dei risultati previsti mediante il *mean absolute error*(MAE). Successivamente si presenterà l'algoritmo di ricostruzione di un segnale sparso, basato su tali applicazioni.

3.1 Trasformata discreta di Fourier (DFT)

I principali metodi di analisi dei segnali possono essere riassunti nei concetti di "analisi nel dominio del tempo" e "analisi nel dominio della frequenza". Risulta importante osservare che questi due modi di affrontare un problema sono tra loro intercambiabili, nel senso che, sotto opportune condizioni, nessuna informazione viene persa nel passare da un dominio all'altro. Il vantaggio che deriva dall'introduzione dei due domini è la possibilità di cambiare la prospettiva con la quale si osserva un dato fenomeno. In questo modo un problema che appare di difficile soluzione in un dominio può risultare molto più semplice nell'altro. Lo strumento matematico che consente di trasferire lo studio dei segnali e dei sistemi dal dominio del tempo al dominio della frequenza è la trasformata di Fourier. La trasformata di Fourier $X(f)$ di una funzione continua nel tempo $x(t)$ è data dalla relazione:

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft} dt \quad (3.1)$$

il risultato è una funzione $X(f)$ a valori complessi nella variabile reale f . Nel caso di un segnale campionato nel tempo $x(nT_s)$, dove $T_s = 1/F_s$ è l'intervallo di campionamento, supponendo che F_s rispetti il teorema di Nyquist, ovvero $F_s > 2f_0$ (f_0 è la frequenza fondamentale del segnale), non è più possibile

utilizzare l'integrale di Fourier perchè la variabile su cui si integra non è più continua. Inoltre, nella realtà non è possibile acquisire nè memorizzare un numero infinito di campioni. Per questo motivo si utilizza la trasformata discreta di Fourier DFT (*Discret Fourier Transform*) definita nel seguente modo:

$$X(kF) = \sum_{n=0}^{N-1} T_s x(nT_s) e^{-j2\pi k T_s n F} \quad (3.2)$$

dove N è il numero di campioni raccolti, con passo di campionamento costante e frequenza $F_s = 1/T_s$, e F è l'intervallo di campionamento in frequenza:

$$F = \frac{F_s}{N} = \frac{1}{T_s N}. \quad (3.3)$$

La selezione degli N campioni viene rappresentata da una funzione finestra $w(nT_s)$ definita nel seguente modo:

$$w(nT_s) = \begin{cases} 1 & 0 < n < N - 1 \\ 0 & \text{altrove} \end{cases} \quad (3.4)$$

La DFT esprime un segnale in una combinazione lineare di funzioni seno e coseno. Come si può notare dagli indici della sommatoria, essa permette di calcolare lo spettro su un numero di campioni finito, quindi con una osservazione del segnale della durata di:

$$T_w = \frac{N}{F_s} = N \cdot T_s \quad (3.5)$$

Poiché le trasformazioni discrete di Fourier (diretta e inversa) coinvolgono solo campioni (sia nel dominio del tempo che della frequenza), queste vengono definite in forma normalizzata rispetto a variabili indipendenti di tipo adimensionale: pertanto la variabile tempo diventa l'indice n , mentre la variabile frequenza corrisponde all'indice k . Si ottiene che:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j \frac{2\pi}{N} kn} \quad (3.6)$$

Si osservi che, a differenza di quanto avviene per la trasformata continua, il segnale tempo-discreto $x(n)$ e ciascun coefficiente $X(k)$ vengono espressi mediante le stesse unità di misura. Ciò è dovuto al fatto che, mentre nel caso continuo la trasformata di Fourier rappresenta una densità spettrale, in questo caso i coefficienti $X(k)$ rappresentano direttamente i valori delle singole componenti spettrali.

3.1.1 Algoritmi FFT(Fast Fourier Transform)

La DFT è uno strumento molto utile per determinare lo spettro in frequenza di un segnale, ma l'onere computazionale richiesto per implementare questa tecnica può essere elevato. Infatti, entrambi gli indici n e k , in riferimento alla formula (3.6), devono variare tra 0 e N per produrre tutti i risultati desiderati; pertanto sono necessarie N^2 operazioni di moltiplicazione e addizione. Per questo motivo, si utilizza un algoritmo alternativo per la valutazione della DFT, come la trasformata rapida di Fourier (*Fast Fourier Transform, FFT*) che, sfruttando le proprietà della DFT, ne consentono una valutazione più rapida. Tale algoritmo è caratterizzato da un carico computazionale estremamente ridotto ($N \log(N)$). In questo caso, però, il numero di campioni N è vincolato ad essere una potenza di due.

3.2 Norma l_p

Per un segnale $x(n)$ tempo-limitato, il cui segnale trasformato equivale a $X(k) = T[x(n)]$, la norma l_2 , o *norma Euclidea*, è definita come la radice quadrata della sua energia totale:

$$l_2 = M_2[T[x(n)]] = \frac{1}{N} \sqrt{\left(\sum_k |X(k)|^2 \right)}, \quad (3.7)$$

normalizzata rispetto al numero totale di campioni N .

Generalizzando la norma Euclidea normalizzata si ottiene la norma l_p , sempre normalizzata rispetto ad N :

$$l_p = M_p[T[x(n)]] = \frac{1}{N} \left(\sum_k |X(k)|^p \right)^{1/p} \quad (3.8)$$

con $1 \leq p \leq \infty$.

3.2.1 Uso della norma-uno

Per $p = 1$ può essere introdotta la seguente norma l_1 , normalizzata rispetto al numero di campioni N :

$$l_1 = M_1[T[x(n)]] = \frac{1}{N} \sum_k |X(k)| = \frac{1}{N} \|X(k)\|_1 \quad (3.9)$$

La norma-uno è la metrica più comunemente usata negli algoritmi di *Compressing Sensing* (CS) per misurare la sparsità del segnale, assieme alla pseudonorma l_0 che tuttavia non viene discussa in questa tesi.

3.2.2 Implementazione norma l_1

Si consideri, ad esempio, un segnale discreto

$$x(n) = \sin(20\pi n/N) \quad (3.10)$$

per $n = 0, 1, \dots, N - 1$, e $N = 256$ indica il numero di campioni del segnale.

Le rappresentazioni in figura (3.1) indicano rispettivamente il segnale $x(n)$ nel dominio del tempo e il segnale trasformato $X(k)$. L'algoritmo per il calcolo della norma l_1 del segnale $x(n)$ fornisce il risultato $Mp = 1.0000$.

Listing 3.1: Norma-uno l_1

```

1 % Calcolo metrica per un segnale x(n) con trasformata di
  Fourier X(k)
2
3 function [Mp] = metrica(X)
4 N = 256;
5 sommatoria = 0;
6 for(k=1:1:N)
7     somma_kesima = abs(X(k));
8     sommatoria = sommatoria + somma_kesima;
9 end
10 metrica = sommatoria / N;
11 end

```

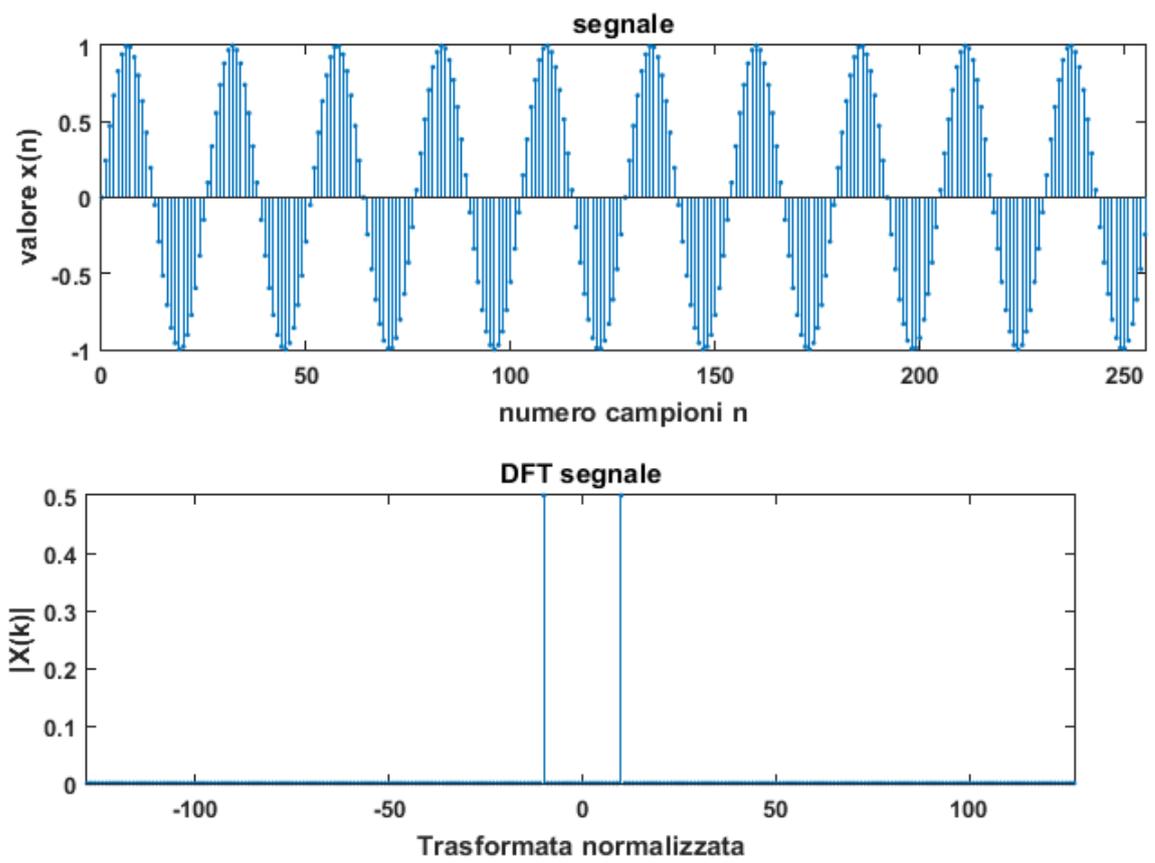


Figura 3.1: Rappresentazione grafica del segnale (3.10) nel dominio del tempo e della trasformazione.

3.3 Mean Absolute Error (MAE)

In statistica, l'errore medio assoluto (MAE) è una grandezza utilizzata per misurare quanto le previsioni sono vicine agli eventuali risultati. Il MAE è dato da:

$$MAE = \frac{1}{n} \sum_{i=1}^n |f_i - y_i| = \frac{1}{n} \sum_{i=1}^n |e_i|. \quad (3.11)$$

Come suggerisce il nome, l'errore medio assoluto è una media degli errori assoluti $|e_i| = |f_i - y_i|$, in cui f_i è la previsione e y_i il valore vero. In particolare, il MAE corrisponde alla norma l_1 dello scostamento tra segnale di partenza e segnale ricostruito.

Nella ricostruzione di un segnale, il valore dell'errore dipende dalla scelta dei parametri dell'algoritmo di ricostruzione, dal numero di iterazioni necessario e dal MAE relativo a ciascuna iterazione, che può essere definito come

$$MAE(k) = \frac{1}{N} \sum_n |x(n) - y^{(k)}(n)| \quad (3.12)$$

dove $x(n)$ rappresenta un segnale tempo-discreto e $y^{(k)}(n)$ il segnale ricostruito, valutato alla k -esima iterazione. A seguire si presenta l'algoritmo per il calcolo del MAE.

3.3.1 Implementazione MAE

Listing 3.2: mean absolute error (MAE)

```
1 % Calcolo del MAE
2
3 function [MAE] = mae(x,y)
4 e = x-y;
5 MAE = metrica(e);
6 end
```

3.4 Algoritmo di ricostruzione

Viene presentato un algoritmo per la ricostruzione di un segnale sparso con campioni mancanti posizionati arbitrariamente; questo è basato sull'uso diretto del metodo del gradiente, dove la funzione costo che si cercherà di minimizzare è stata costruita in modo da quantificare la sparsità del segnale. Verrà considerato il caso in cui l'algoritmo procederà verso la soluzione con un passo costante e riferendosi alla norma-uno (3.9).

3.4.1 Algoritmo

Si consideri un segnale tempo-discreto $x(n)$ i cui coefficienti del segnale trasformato sono $X(k) = DFT[x(n)]$. Il segnale $x(n)$ è sparso nel dominio di trasformazione: il numero di coefficienti trasformati significativamente diversi da zero è molto più piccolo del numero di campioni N del segnale originale. Si assuma che solamente un sottoinsieme di campioni del segnale nel dominio del tempo siano disponibili agli istanti corrispondenti alle posizioni tempo-discrete:

$$n_i \in \mathbb{N}_A = \{n_1, n_2, \dots, n_M\} \subset \mathbb{N} = \{0, 1, 2, \dots, N-1\}. \quad (3.13)$$

Si definisce il segnale iniziale $y^{(0)}(n)$ per $n \in \mathbb{N} = \{0, 1, \dots, N-1\}$ come:

$$y^{(0)}(n) = \begin{cases} 0 & n \in \mathbb{N}_Q \\ x(n) & n \in \mathbb{N}_A, \end{cases} \quad (3.14)$$

dove \mathbb{N}_Q è il complementare di \mathbb{N}_A rispetto a \mathbb{N} definito da (3.7) e (0) indica che è la prima iterazione dell'algoritmo. Per ogni campione mancante alle posizioni $n_i \in \mathbb{N}_Q$, i loro valori sono calcolati creando due segnali $y_1(n)$ e $y_2(n)$ in ogni iterazione successiva come:

$$y_1^{(k)}(n) = y^{(k)}(n) + \Delta \cdot \delta(n - n_i) \quad (3.15)$$

$$y_2^{(k)}(n) = y^{(k)}(n) - \Delta \cdot \delta(n - n_i), \quad (3.16)$$

dove k è il numero di iterazioni e la costante Δ indica il passo dell'algoritmo, usata per determinare se il campione del segnale considerato debba essere aumentato o diminuito. I campioni mancanti vengono quindi corretti iterativamente nel seguente modo:

$$y^{(k+1)}(n) = y^{(k)}(n) - \mu \cdot g, \quad (3.17)$$

dove μ è una costante definita come *step size*, che influenza le prestazioni dell'algoritmo (come l'errore e la velocità di convergenza) e g è un vettore

gradiente avente la stessa dimensione del segnale $x(n)$. Alle posizioni dei campioni del segnale disponibili, $n \in \mathbb{N}_A$, $g(n) = 0$, mentre nelle posizioni dei campioni mancanti i valori sono calcolati mediante

$$g(n_i) = \frac{M_p[T[y_1^{(k)}(n)]] - M_p[T[y_2^{(k)}(n)]]}{2\Delta} \quad (3.18)$$

dove M_p è definito dalla (3.8). L'espressione di $g(n_i)$ rappresenta una stima della misura di sparsità del vettore gradiente. Ripetendo il procedimento iterativamente, i valori mancanti convergeranno ai valori del segnale che producono una misura di concentrazione minima nel dominio della trasformazione.

3.4.2 Condizioni di arresto

Le condizioni di arresto sono indispensabili: oltre ad assicurare che l'algoritmo abbia un tempo di esecuzione finito, esse si rivelano particolarmente utili qualora sia difficile ottenere un risultato esatto. Molto spesso, infatti, è necessario accettare un'approssimazione del risultato che si vorrebbe ottenere. L'algoritmo in esame si arresta e fornisce la ricostruzione dei campioni mancanti quando si verifica una delle seguenti condizioni di stop:

- se $(M_{p_{ricostruito}} - M_{p_{originale}}) \leq \epsilon$: la misura di concentrazione del segnale ricostruito si discosta al più di ϵ rispetto al segnale originale;
- se $i > I_{max}$: l'indice di iterazione i raggiunge il numero massimo di iterazioni I_{max} prefissate.

3.4.3 Implementazione algoritmo di ricostruzione

Viene ripreso il segnale $x(n)$ definito da (3.10) come esempio per l'algoritmo di ricostruzione. Il parametro *soglia* rappresenta la percentuale di campioni del segnale originale resi non disponibili; il suo valore corrisponde al numero di campioni mancanti sugli N totali, espresso in percentuale. Per il calcolo delle misure di concentrazione e del MAE necessari si è fatto riferimento rispettivamente agli algoritmi (3.1) e (3.2).

Listing 3.3: Algoritmo di ricostruzione basato sul gradiente

```

1 % Algoritmo di ricostruzione basato sul metodo del
  gradiente
2
3 %% PRIMA PARTE. Creazione segnali x(n) e y(n)
4
5 % Parametri del segnale originale
6 VO = 1;
7 K = 10;
8 Phi0 = 0;
9 N = 256;
10 n = [0:N-1];
11
12 % Definisco il segnale x(n) originale e la sua
  trasformata di Fourier X
13 x = VO*sin((2*pi*K*n)/N+Phi0);
14 X = fft(x);
15
16 % Parametri del segnale y(n) con r campioni mancanti
17 % Percentuale di campioni mancanti
18 soglia = 0.2734;      %(70 campioni mancanti)
19 r = ceil(soglia*N);
20
21 % Definisco il segnale y(n) disponibile
22 % Ciclo che crea un vettore di r numeri interi
23 % compresi tra 1 e 256, scelti in modo
24 % casuale senza ripetizione
25 ind = [1:1:N];
26 M = N;
27 sequenza = zeros(1,r);
28 for(i=1:r)
29     j = floor(M*rand)+1;
30     sequenza(i) = ind(j);
31     ind(j) = ind(M);
32     M = M-1;
33 end
34 x_disponibile = x;
35 x_disponibile(sequenza) = 0;
36 y = x_disponibile;
37 Y = fft(y);
38
39
40

```

```

41 %% SECONDA PARTE. Ricostruzione segnale y(n)
42
43 % Parametri e inizializzazione delle variabili
44 Delta = 0.5;
45 mu = 1;
46 Imax = 100;
47 iter = 1;
48
49 y1 = zeros(1,N);
50 y2 = zeros(1,N);
51 Y1 = zeros(1,N);
52 Y2 = zeros(1,N);
53 g = zeros(1,N);
54 y_ricostruito = y;      % Inizializzo al valore y(n)
55 y_precedente = y;      % Tenere aggiornato il penultimo
                           risultato
56 y_successivo = y_precedente;
57 % Calcolo metrica di X
58 Mpx = metrica(X);
59 % Calcolo metrica di Y
60 Mpy = metrica(Y);
61 Mpy_successivo = Mpy;
62 % Calcolo MAE con segnale disponibile y(n)
63 MAE(1) = mae(x,y);
64
65 %Condizioni di stop
66 while(iter < Imax & (abs(Mpy_successivo - Mpx) > 0.05))
67     y_ricostruito = y_successivo;
68     % Creazione segnali y1(n) e y2(n)
69     for(i=1:N)
70         % Caso di campione mancante
71         if(y(i) == 0)
72             z = zeros(1,N);
73             z(i) = 1;
74             y1 = y_successivo + Delta*z;
75             y2 = y_successivo - Delta*z;
76             Y1 = fft(y1);
77             Y2 = fft(y2);
78             % Calcolo misure di concentrazione di y1(n)
                           e y2(n)
79             Mp1 = metrica(Y1);
80             Mp2 = metrica(Y2);
81             % Calcolo differenziale della concentrazione

```

```

82         g(i) = (Mp1 - Mp2) / (2*Delta);
83     % Caso di campione disponibile
84     else
85         y1 = y_successivo;
86         y2 = y_successivo;
87         g(i) = 0;
88     end
89 end
90 % Ricostruzione segnale
91 y_successivo = y_precedente - mu*g;
92 % Aggiornamento iterazione
93 iter = iter+1;
94
95 % Metrica del segnale ricostruito
96 Y_successivo = fft(y_successivo);
97 Mpy_successivo = metrica(Y_successivo);
98 % Calcolo mean absolute error (MAE) ad ogni
    iterazione
99 MAE(iter) = mae(x,y_successivo);
100
101 % Aggiornamento risultati
102 y_precedente = y_successivo;
103 end
104
105 % Calcolo errore di ricostruzione
106 errore = x - y_ricostruito;

```

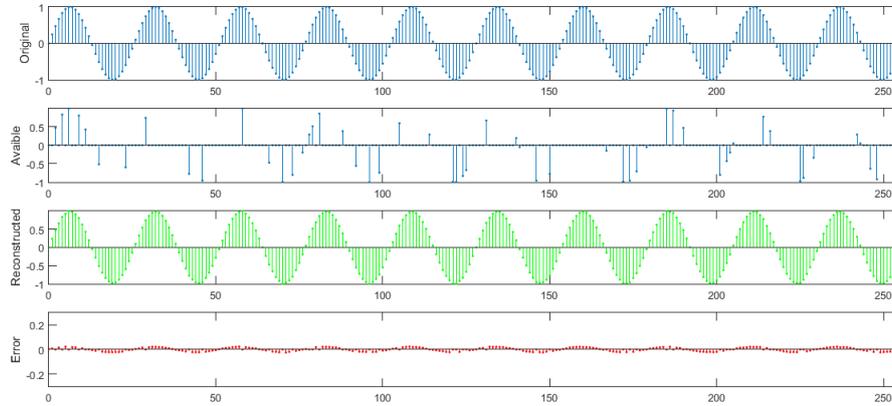
Capitolo 4

Esempi e applicazioni

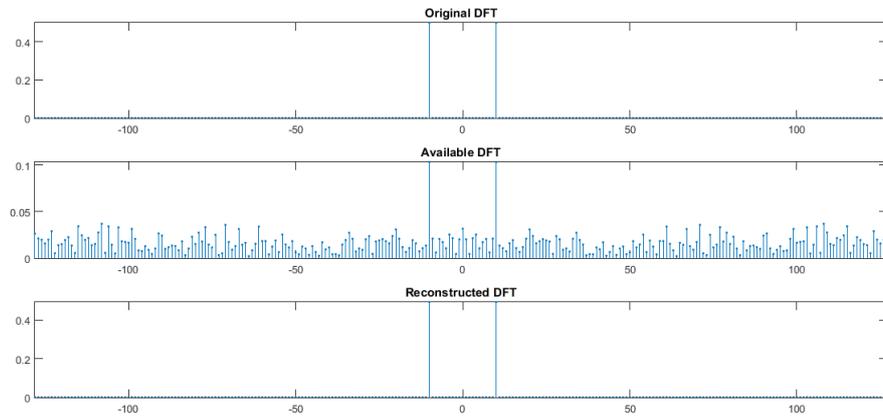
I segnali periodici sono di particolare interesse pratico. In tali casi, l'analisi armonica mediante DFT richiede particolare cautela, soprattutto per quanto riguarda la scelta della finestra di troncamento e il fatto che la frequenza di campionamento sia o meno sincronizzata con la frequenza fondamentale del segnale da analizzare. A seconda del numero di periodi contenuti nella finestra di osservazione, che può essere intero o frazionario, si hanno rispettivamente il caso del campionamento *coerente* e campionamento *non-coerente*. In questo capitolo vengono presentati alcuni esempi di applicazione dell'algoritmo implementato in (3.3). Inoltre, utilizzando l'algoritmo sono stati provati e verificati con successo gli esempi in [1] relativi al caso di un segnale sparso campionato in modo coerente e di un segnale sparso in modo approssimativo; non verranno riportati in questa tesi.

4.1 Campionamento coerente

Il primo esempio riprende il segnale $x(n)$ (3.10) considerato precedentemente. Si tratta di un caso di campionamento coerente, dove la finestra di osservazione $T_w = kT_0$ contiene $k = 10$, $k \in \mathbb{Z}$, periodi T_0 del segnale da analizzare. In questo caso ripetere la finestra di osservazione T_w indefinitamente nel tempo significa riprodurre in forma esatta la funzione periodica originaria. Il calcolo della DFT fornisce una rappresentazione corretta dello spettro, come mostrato in figura (3.1), con componenti tutte nulle tranne proprio l'unica componente armonica effettivamente presente alla frequenza $k f_w$.



(a) Ricostruzione di un segnale campionato in modo coerente



(b) DFT rispettivamente del segnale originale, disponibile e ricostruito

Figura 4.1: Ricostruzione per il campionamento coerente

4.1.1 Esempio campionamento coerente e risultati

Viene creato il segnale $y(n)$ secondo (3.14), con 200 campioni mancanti o non disponibili posizionati in modo arbitrario. Si esegue la ricostruzione del segnale mediante l'algoritmo (3.3), nel quale sono stati definiti i parametri costanti $\Delta = 0.5$, $\mu = 1$.

I risultati sono riportati in (4.1). Il segnale $y(n)$ parte da una metrica di 4.2173. L'algoritmo di ricostruzione genera il nuovo segnale basandosi sull'approccio di minimizzazione della sparsità, ottenendo una misura di concentrazione pari a 1.0693 in 51 iterazioni, con un errore residuo massimo di 0.0241. Si noti che per parametri dell'algoritmo costanti il MAE non si può migliorare, oltre un certo limite, aumentando il numero di iterazioni.

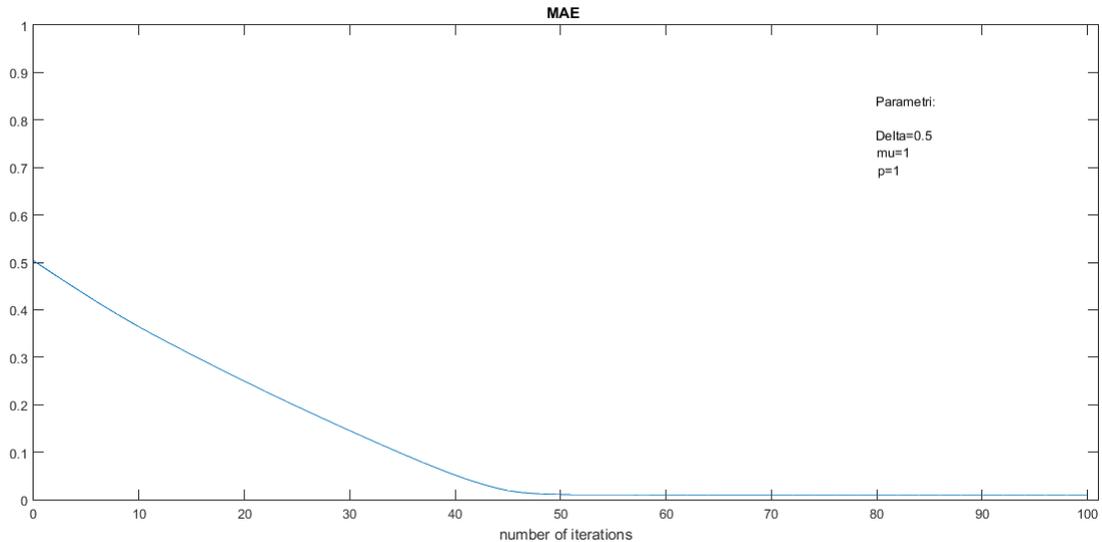


Figura 4.2: MAE relativo alla ricostruzione del segnale (campionamento coerente)

4.2 Campionamento non coerente

Quando la finestra di osservazione T_w contiene un numero frazionario di periodi k di T_0 del segnale, ad esempio $k = 10, 7$, $k \in Q$, la ripetizione nel tempo del segnale campionato e troncato non riprodurrà esattamente la funzione periodica originaria, con una conseguente distorsione nello spettro.

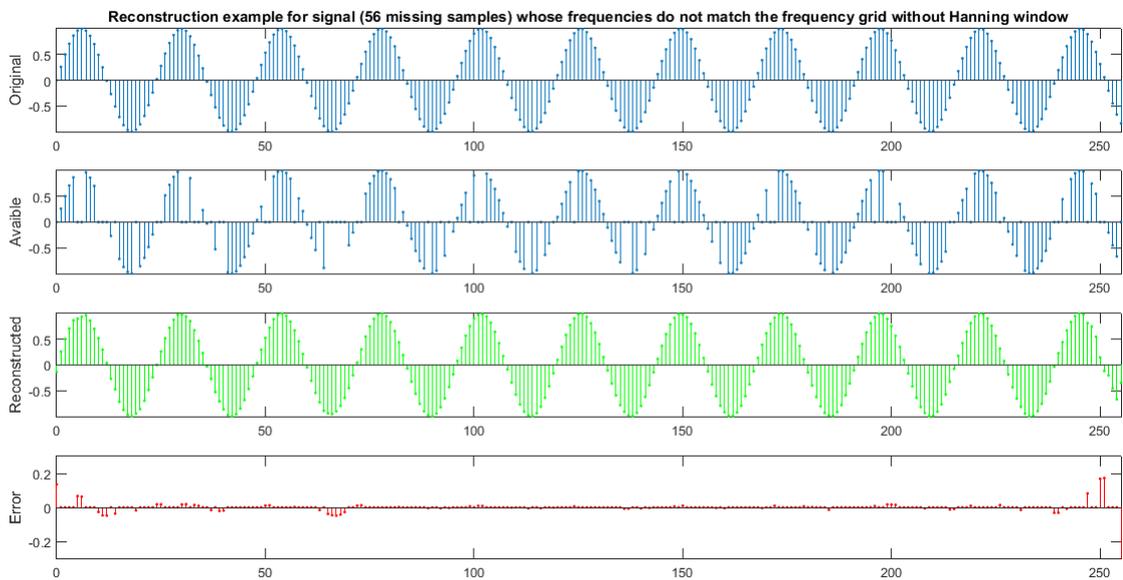
4.2.1 Esempio campionamento non coerente e risultati

Il segnale $x'(n)$ considerato per l'analisi è dato da:

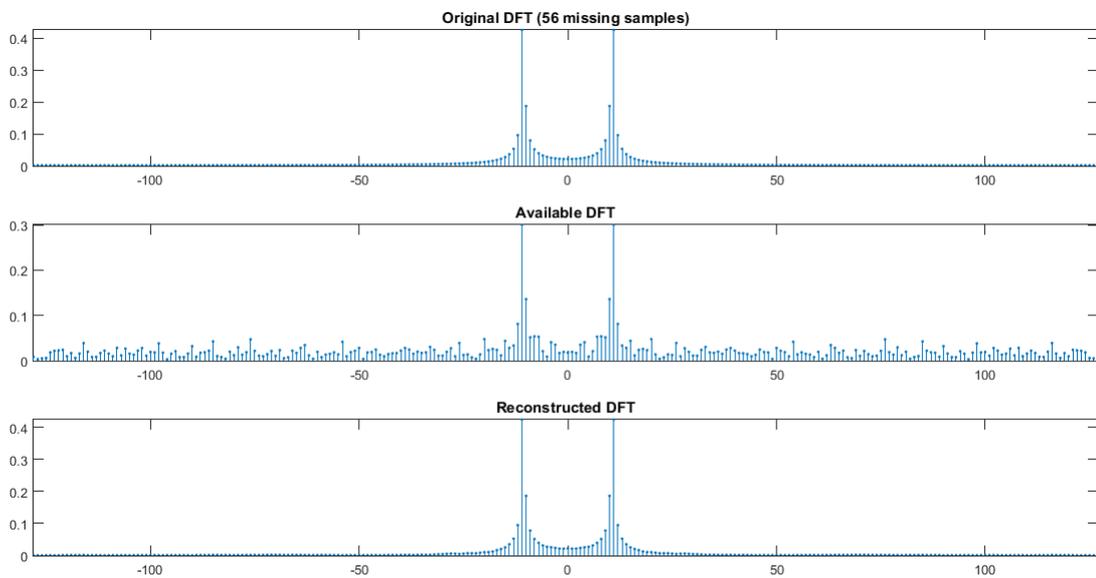
$$x'(n) = \sin(20.7\pi n/N), \quad (4.1)$$

dove $n = 0, 1, \dots, N - 1$ e $N = 256$. La non coerenza del campionamento trova riscontro nella DFT, che evidenzierà, in tal caso, componenti armoniche non presenti nello spettro del segnale periodico originario. Sebbene questo tipo di segnale non sia sparso in senso stretto nel dominio della trasformazione, applicando l'algoritmo del gradiente presentato con i parametri $\Delta = 0.5$ e $\mu = 1$ ed eliminando arbitrariamente 70 campioni dal segnale originale, si riescono comunque ad ottenere risultati soddisfacenti.

Si noti che con 20 iterazioni, è possibile ricostruire il segnale di partenza. Si ottiene, a partire da una misura di concentrazione di $y(n)$ di 5.9555, un valore finale di 3.1874, che rappresenta un ottimo risultato se paragonato



(a) Ricostruzione di un segnale campionato in modo non coerente



(b) DFT rispettivamente del segnale originale, disponibile e ricostruito

Figura 4.3: Ricostruzione per il campionamento non coerente

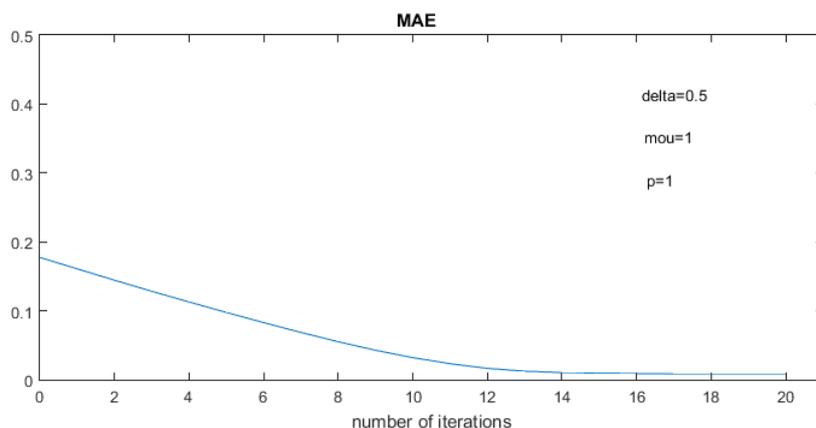


Figura 4.4: MAE per il campionamento non coerente

alla sparsità del segnale originale, equivalente a 3.0825. Inoltre, bastano 10 iterazioni per far raggiungere al MAE lo stato stazionario con l'uso dei parametri costanti, come si evidenzia in figura (4.4). L'errore residuo, tuttavia, risulta aumentare in corrispondenza dei bordi del segnale; la motivazione sta nel fatto che la finestra di osservazione non è esattamente multipla del periodo del segnale. Un modo per limitare l'inconveniente può essere quello di impiegare finestre molto ampie rispetto al periodo della fondamentale f_0 e soprattutto con transizione graduale alle estremità.

4.3 Finestratura

Come è già stato precedentemente osservato, la DFT consente il calcolo dello spettro in frequenza di un segnale a partire da un blocco di campioni dell'ingresso, chiamato finestra di osservazione (o *time record*). Questo calcolo è basato sull'ipotesi che il *record* di dati si ripeta indefinitamente nel tempo. Si è visto, inoltre, come tale considerazione diventi un aspetto critico se il segnale da analizzare è periodico. Qualora non sia attuabile la soluzione del campionamento coerente, può essere utile impiegare le funzioni di "finestratura" (*windowing*), che permettano di ridurre gli effetti del troncamento.

Le finestre, in via generale, presentano spettri abbastanza simili tra loro: è sempre presente un lobo principale centrato nell'origine di ampiezza molto maggiore rispetto al resto dello spettro e con una occupazione in frequenza limitata e nota. Inoltre, è presente una successione di lobi laterali con ampiezze decrescenti all'aumentare della distanza dal loro principale. Come mostrato in figura (4.5), queste funzioni sono caratterizzate a seconda delle proprietà del loro spettro; i parametri principali sono l'attenuazione, espres-

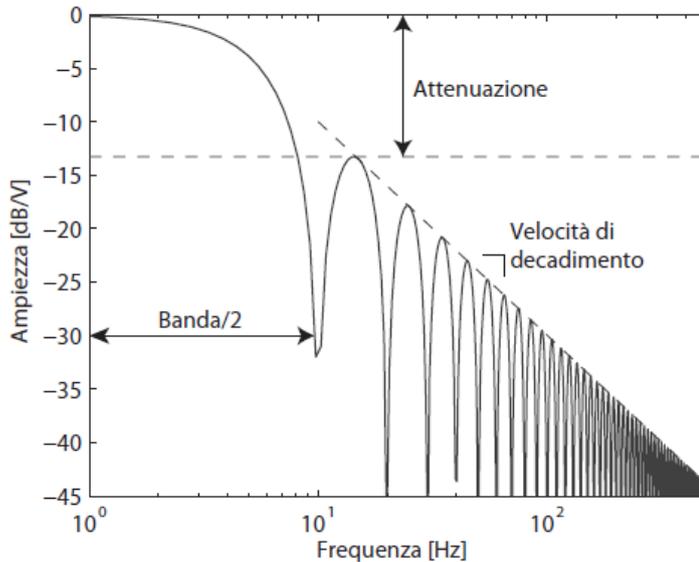


Figura 4.5: Esempio

sa generalmente in decibel, tra il lobo principale e il primo lobo laterale. Si confrontano la larghezza di banda del lobo principale e la velocità di decadimento dei lobi laterali, espressa in dB/Dec .

In un'applicazione reale, il periodo di osservazione T_w è finito, pertanto le finestre devono essere delle funzioni limitate nel tempo. Di conseguenza, i loro spettri saranno infiniti, quindi per ridurre lo *spectral leakage* è fondamentale individuare delle forme che permettano di ottenere lobi principali molto ampi e quelli laterali più attenuati. Purtroppo queste due richieste sono in contrasto tra di loro; è quindi necessaria una soluzione di compromesso, eventualmente ottimizzata in base alle caratteristiche del segnale da analizzare.

Esistono molte tipologie diverse di finestre, ognuna delle quali mette in risalto una particolare proprietà. La tesi si limiterà a considerare la finestra di Hanning e ad utilizzarla sul segnale $y(n)$ considerato nel caso del campionamento non coerente (4.2), per ottenere, mediante l'algoritmo di ricostruzione (3.3), risultati migliori sul segnale ricostruito. La finestra di Hanning ha un andamento cosinusoidale (massimo al centro e nullo agli estremi), la cui trasformata di Fourier ha un lobo unico, ma di larghezza doppia rispetto a quello della finestra rettangolare. Pertanto, la risoluzione in frequenza peggiora, ma viene ridotto il contributo delle frequenze spurie. In figura (4.6) è mostrato lo spettro della finestra di Hanning.

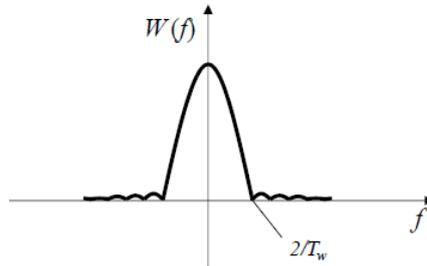


Figura 4.6: Spettro della finestra di Hanning.

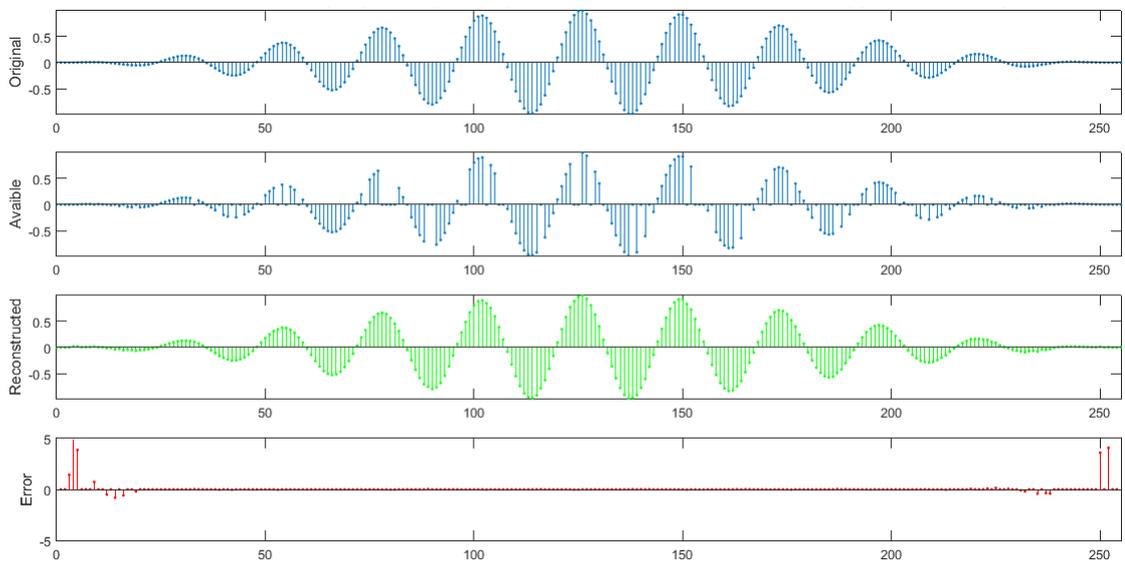
4.3.1 Esempio con finestra di Hanning e risultati

Riprendendo il segnale $x'(n)$ (4.1), si sfrutta la finestra di Hanning per ridurre i problemi presenti ai bordi, legati al troncamento del segnale. Si utilizza l'algoritmo di ricostruzione nel caso di un segnale disponibile $y'(n)$ strutturato come (3.14), con 70 campioni mancanti; i parametri dell'algoritmo sono dati ancora da $\Delta = 0.5$, $\mu = 1$.

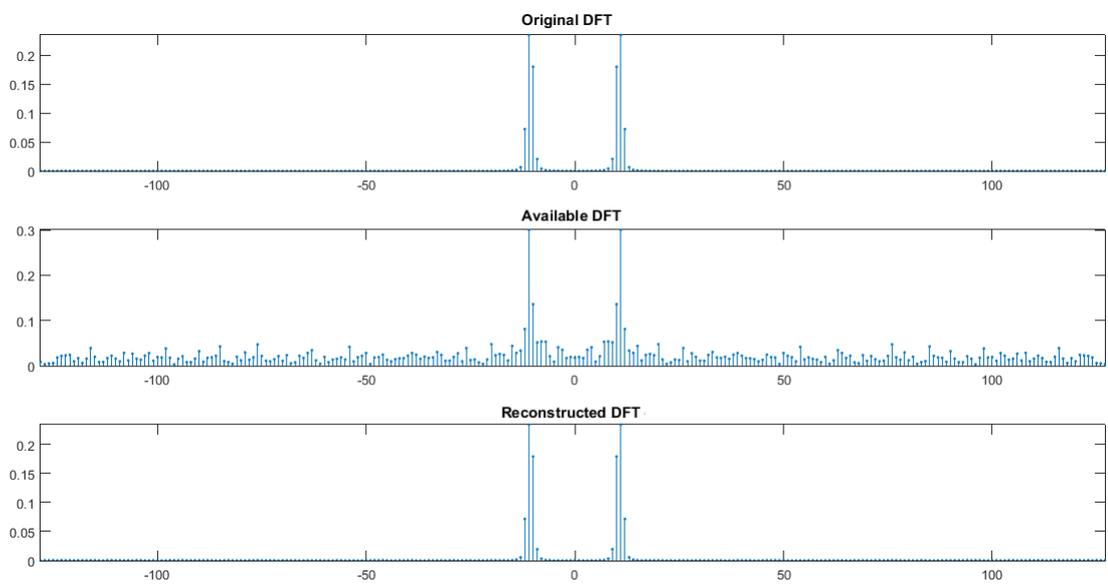
I risultati sono riportati in figura (4.7). La misura di concentrazione del segnale di partenza $x'(n)$ è pari a 1.0512, quella del segnale disponibile $y'(n)$ è 3.4498; sono necessarie 23 iterazioni per ottenere una concentrazione di 1.1011 (utilizzando come condizione di arresto la ricerca di un segnale con una misura di concentrazione al più il 5% maggiore di quella del segnale originale). Il MAE inoltre raggiunge il suo stato stazionario, con ordine di grandezza di 10^{-2} , in 8 iterazioni. In figura (4.8) vi è il confronto, sulla base del MAE, tra le prestazioni del segnale manipolato con la finestra di Hanning e il segnale nel caso del campionamento non coerente; a parità di parametri, il segnale finestrato raggiunge risultati migliori in minor tempo.

Un altro aspetto importante riguarda l'errore residuo tra il segnale originale $x'(n)$ e quello ricostruito, a cui è stata rimossa la finestra di Hanning in seguito al processo di ricostruzione. Come si evince dal risultato ottenuto in figura (4.7(a)), gli errori di ricostruzione tendono ad addensarsi ai bordi; ciò è dovuto all'azione della finestra di Hanning che, come già rilevato, produce una forte attenuazione agli estremi della finestra di osservazione. Una soluzione per limitare il problema consiste nel considerare solo una porzione centrale e verificare l'errore di ricostruzione (4.9); questo, infatti, sarà minimo grazie alla forma sinusoidale della finestra, e il risultato ottenuto nel segnale ricostruito potrà ritenersi molto buono.

Per ridurre l'errore ai bordi si potrebbe aggiungere un sezione di finestra step-by-step, in modo da inglobare più segnale possibile nella banda passante della finestra. Questo aspetto, tuttavia, non verrà trattato nella tesi.



(a) Ricostruzione di un segnale con finestra di Hanning



(b) DFT rispettivamente del segnale originale, disponibile e ricostruito

Figura 4.7: Ricostruzione per il campionamento coerente

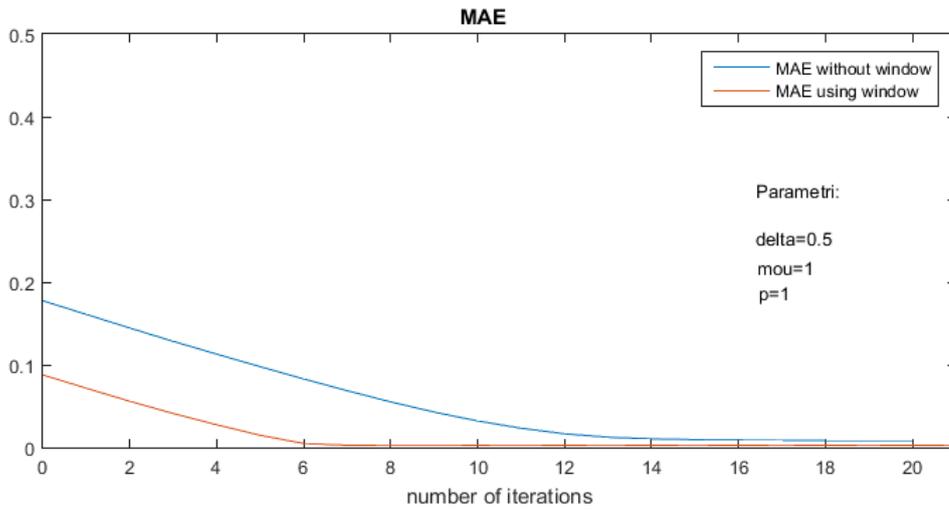


Figura 4.8: Confronto MAE tra segnale $x'(n)$ (4.1) con e senza la finestra di Hanning

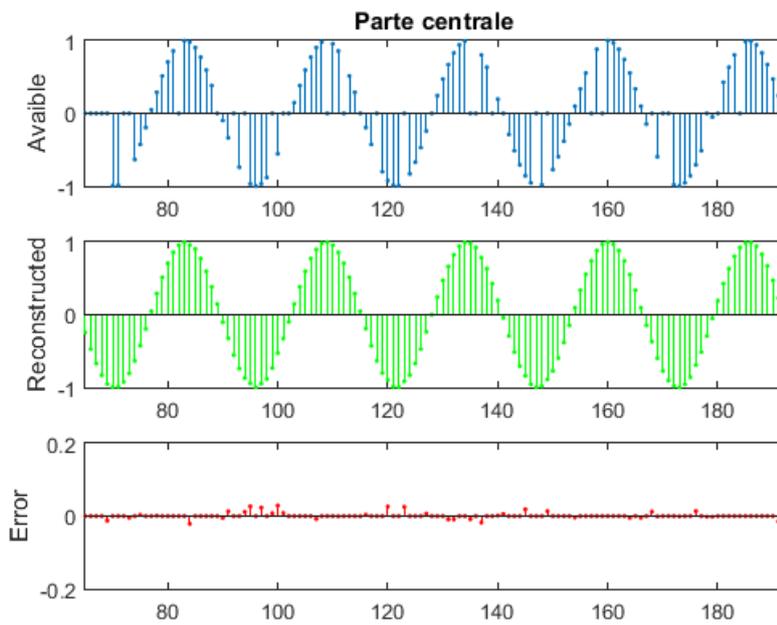


Figura 4.9: Errore residuo di ricostruzione tra il segnale originale e quello ricostruito privato della finestra di Hanning, valutato su una porzione centrale.

4.4 Segnali reali

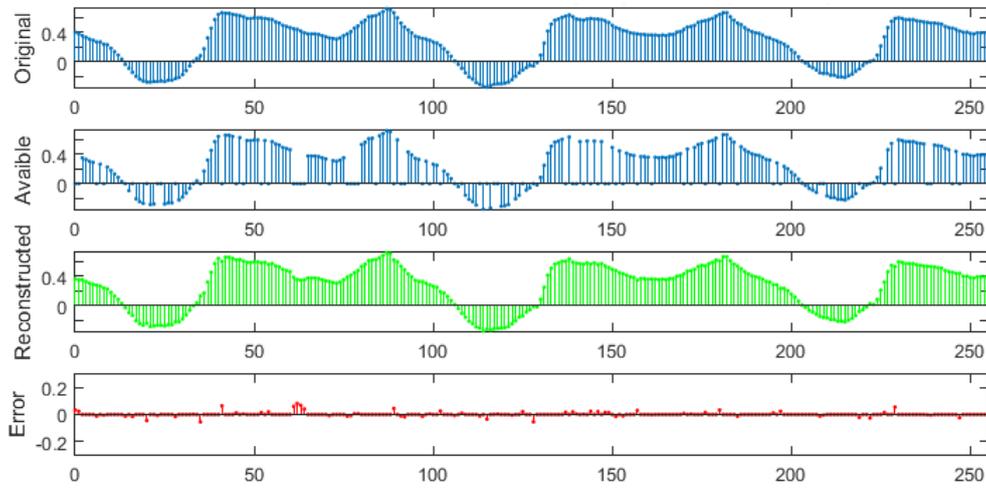
L'algoritmo di ricostruzione di segnali sparsi è usato anche per l'analisi di segnali reali, ad esempio segnali rilevati durante applicazioni biomediche.

4.4.1 Esempio con segnale biomedico e risultati

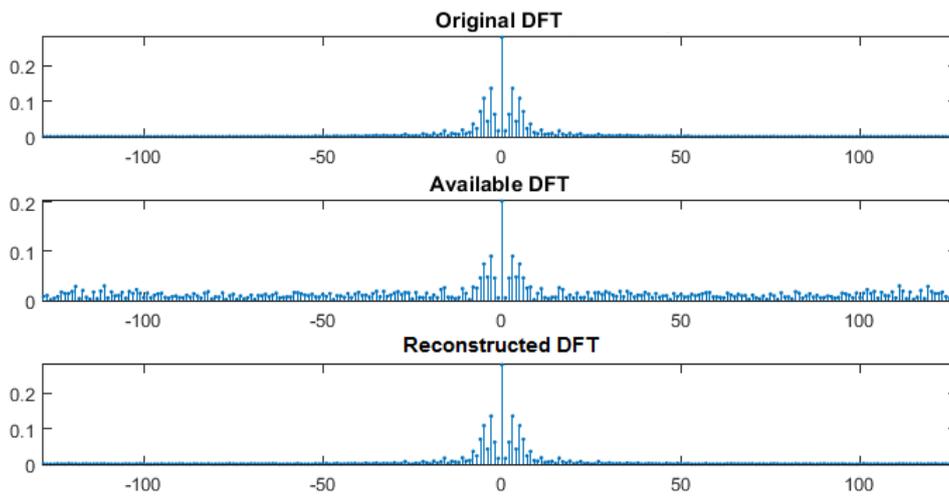
I segnali biomedici originali sono molto rumorosi e necessitano di un processo di filtraggio prima di essere utilizzati, in modo tale da rendere più agevole l'analisi sia in ambito medicale, permettendo diagnosi più accurate, che in ambito ingegneristico. Un'ulteriore difficoltà è legata alla quasi-periodicità della maggior parte dei segnali utilizzati in ambito biomedico. A causa di questo, non è possibile l'individuazione di cicli periodici semplicemente effettuando un opportuno campionamento; è necessaria una procedura preliminare che consente di individuare i punti più importanti e poi risalire al ciclo, attraverso idonee traslazioni temporali.

I segnali biomedici non saranno oggetto di questa tesi, ma verrà presentato un esempio di ricostruzione di segnale di un accelerometro. Gli accelerometri sono trasduttori che possono essere attaccati a varie parti del corpo umano del soggetto per misurare accelerazioni nelle tre dimensioni; essi sono in grado di trasdurre l'accelerazione di un corpo in un segnale elettrico.

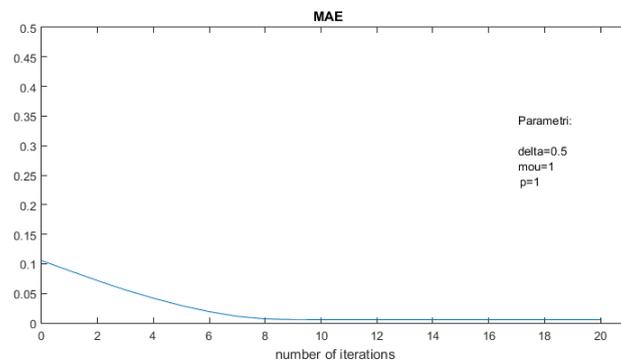
In figura (4.10) sono riportati i risultati della ricostruzione per un segnale con 70 campioni mancanti. Sono sufficienti 8 iterazioni per ottenere una buona ricostruzione del segnale che corrisponda alle richieste dell'algoritmo, con un errore di ricostruzione al più del 10%.



(a) Ricostruzione di un segnale biomedico



(b) DFT rispettivamente del segnale originale, disponibile e ricostruito



(c) MAE relativo alla ricostruzione del segnale

Figura 4.10: Ricostruzione per il campionamento coerente

Capitolo 5

Conclusioni

In questo lavoro è stato implementato e testato un nuovo metodo per la ricostruzione dei segnali ottenuti da acquisizioni con una percentuale anche significativa di campioni mancanti. L'algoritmo realizzato basato sul gradiente, con dimensione del passo costante, è stato applicato a diversi segnali; sono stati illustrati esempi numerici relativi a segnali sparsi, compresi anche quelli sparsi solo in modo approssimativo. È stato anche considerato l'esempio con un segnale biomedico reale.

La possibilità di ricostruire completamente tutto il segnale campionando un numero estremamente ridotto di informazioni si è rivelato di fondamentale importanza in diversi settori, anche riguardo temi centrali nella Bioingegneria e nelle applicazioni ospedaliere. Lo strumento teorico ottenuto permette di porre l'attenzione non più sulla banda del segnale, come nella teoria di Shannon, ma sulla proprietà di sparsità del segnale e sulle conoscenze del segnale a priori che permettono di trovare una soluzione di ottimo tra infinite possibili soluzioni.

Bibliografia

- [1] Ljubiša Stanković, Milosš Daković, Stefan Vujović, "Adaptive Variable Step Algorithm for Missing Samples Recovery in Sparse Signals", *IET Signal Processing*, September, 2013.
- [2] Ljubiša Stanković, Milosš Daković, "On the Reconstruction of Randomly Sampled Sparse Signals Using an Adaptive Gradient Algorithm", University of Montenegro, Podgorica, Apr. 26, 2015, <http://arxiv.org/abs/1412.0624>,
- [3] https://en.wikipedia.org/wiki/Mean_absolute_error.
- [4] http://www.diee.unica.it/misure/Dispense/Misure_Elettroniche_dm270/Analisi_di_segnaali_campionati.pdf.