



UNIVERSITÀ DEGLI STUDI DI CAGLIARI  
FACOLTÀ DI INGEGNERIA E ARCHITETTURA

## APPROSSIMAZIONE SPARSA E MATCHING PURSUIT

Corso di Laurea triennale in Ingegneria Elettrica, Elettronica  
ed Informatica

Candidato:  
SILVIA MUGGIANO

Relatore:  
Prof. GIUSEPPE RODRIGUEZ

ANNO ACCADEMICO 2019-2020

*Uno speciale ringraziamento al Professore Giuseppe Rodriguez,  
mio relatore e docente,  
per l'aiuto, la disponibilità e la competenza.  
E un grazie alle dottoresse Patricia Diaz De Alba e Federica Pes,  
per l'assistenza nella programmazione.*

# Indice

INTRODUZIONE.....	4
CAPITOLO 1 PROBLEMA AI MINIMI QUADRATI.....	5
1.1 Metodo delle equazioni normali .....	6
1.2 Risoluzione mediante fattorizzazione QR .....	7
1.2.1 <i>Fattorizzazione QR</i> .....	9
CAPITOLO 2 L'ANALISI DI FOURIER .....	10
2.1 Serie di Fourier .....	10
2.2 Trasformata di Fourier .....	13
2.2.1 <i>Trasformata di Fourier discreta</i> .....	15
2.2.2 <i>Trasformata di Fourier a tempo breve</i> .....	17
CAPITOLO 3 LA TRASFORMATA WAVELET.....	21
3.1 La Trasformata Wavelet Continua.....	21
3.2 La Trasformata Wavelet discreta .....	25
3.3 Analisi Multirisoluzione .....	27
3.4 Funzione di Scaling .....	27
3.5 Wavelet .....	29
3.5.1 <i>Wavelet di Haar</i> .....	30
3.5.2 <i>Famiglie di Wavelet</i> .....	32
CAPITOLO 4 PROBLEMI DI APPROSSIMAZIONE SPARSA.....	34
4.1 Dizionari Ridondanti e Sparsità.....	35
4.2 Algoritmi <i>greedy</i> .....	35
4.3 Prove numeriche .....	36
4.3.1.1 Segnale Gaussiano.....	36
4.3.1.2 Segnale a gradino .....	38
4.3.1.3 Analisi consumo di elettricità.....	41
LISTATI MATLAB .....	44
BIBLIOGRAFIA.....	47

# Introduzione

Il problema dell'approssimazione di funzioni continue su un intervallo nasce quando si vogliono valutare numericamente funzioni trigonometriche o trascendenti. Poiché tale valutazione deve essere ridotta a un numero finito di operazioni aritmetiche, si deve approssimare la funzione per mezzo di polinomi o funzioni razionali.

Da un punto di vista anche piuttosto intuitivo, è facile immaginare cosa sia un segnale; cosa però più complicata è trasformare questo concetto astratto in termini di formule. I segnali sono quantità misurabili che trasportano informazioni relative ad un fenomeno fisico che può essere di varia natura: elettrica, ottica, acustica.

Matematicamente possono essere descritti tramite una funzione, composta da almeno una variabile indipendente, come il tempo o lo spazio, legata ad una variabile dipendente, ossia il valore misurato della grandezza fisica.

Generalmente, lo strumento utilizzato per l'analisi dei segnali è la *trasformata di Fourier*, usato soprattutto per segnali periodici, con cui è possibile scomporre il segnale in funzioni di base sinusoidali a diverse frequenze.

Il problema principale della trasformata di Fourier è dato dal fatto che seni e coseni sono perfettamente localizzati in frequenza e non nel tempo.

La *trasformata di Fourier a tempo breve* risolve tale problema. Essa utilizza una funzione finestra che viene traslata lungo l'asse temporale, ed è in grado di analizzare il segnale in piccole sezioni, così da riuscire a mantenere alcune informazioni sia in frequenza che nel tempo. Tuttavia, la trasformata di Fourier a tempo breve possiede delle limitazioni, ed una volta scelte le dimensioni della finestra, queste rimangono fissate.

Le *wavelet* sono funzioni oscillanti con valore medio nullo e durata limitata. Esse costituiscono degli strumenti matematici adatti alla rappresentazione di segnali, per l'analisi tempo-frequenza, grazie all'uso dell'*analisi multirisoluzione*.

Alcuni segnali non possono essere rappresentati in modo efficiente attraverso una base ortonormale, come gli impulsi e le sinusoidi. Per affrontare questo tipo di problema, negli ultimi 15 anni alcuni ricercatori hanno studiato una rappresentazione ridondante del segnale, chiamata *approssimazione sparsa del segnale*.

La domanda algoritmica principale è la seguente:

*Per una data classe di dizionari, come si può progettare un algoritmo veloce che calcoli in modo dimostrabile una rappresentazione sparsa, quasi ottimale, di un segnale di ingresso arbitrario?*

Sfortunatamente è abbastanza difficile rispondere. Bisogna introdurre una procedura che sceglie le più adatte forme d'onda in grado di scomporre in modo ottimale la struttura del segnale. Al momento, per risolvere questo tipo di problemi son stati presentati nuovi risultati mediante l'utilizzo di algoritmi avari (greedy), il *matching pursuit (MP)* e l'*orthogonal matching pursuit (OMP)*.

In conclusione, grazie a questo lavoro di studio è stato possibile analizzare gli algoritmi studiati e confrontare i risultati ottenuti.

# Capitolo 1

## Problema ai minimi quadrati

I problemi ai minimi quadrati nascono dal fatto che, frequentemente, i sistemi lineari abbiano un numero di equazioni differenti dal numero di incognite, del tipo

$$Ax = b,$$

con  $A$  matrice  $m \times n$ ,  $\mathbf{b} \in \mathbb{R}^m$  e  $\mathbf{x} \in \mathbb{R}^n$ .

Nel caso in cui  $A$  sia di rango pieno, possiamo distinguere due casi:

- a)  $m > n$ , vi sono più equazioni che incognite, si dice sovradeterminato. Ossia, il problema potrebbe non avere soluzione.
- b)  $m < n$ , vi sono più incognite che equazioni, si dice sottodeterminato. Il problema potrebbe avere infinite soluzioni.

Tutti questi problemi sono mal posti<sup>1</sup>, e non hanno una soluzione in senso classico, è necessario quindi riformularlo in un problema ben posto.

Se esaminiamo il primo caso a) si sta studiando un fenomeno che dipende da un certo numero di variabili, ma il numero di equazioni lineari a disposizione è superiore al numero di incognite. Nel caso ideale di dati esatti, significa che  $m - n$  equazioni sono linearmente dipendenti dalle altre. Quando i dati presentano errori, potrebbe capitare che nessuna delle equazioni sia verificata esattamente e, sarebbe preferibile utilizzarne un numero sovrabbondante per cercare di ripulire la soluzione. [1]

Se tutte le equazioni non possono essere verificate contemporaneamente, è ragionevole richiedere che lo scarto quadratico medio, o varianza, tra il primo e il secondo membro sia minimo.

Se  $A$  è rettangolare, il sistema non può essere espresso come  $Ax=b$ , si passa così ad un problema ai minimi quadrati (*least squares problem*). La condizione di varianza minima si può esprimere nella forma

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2 \tag{1.1}$$

Se il minimo risulta essere zero, questo significa che il sistema originario ammette una soluzione in senso classico.

---

<sup>1</sup> Un problema si dice ben posto se ha una e una sola soluzione che dipende con continuità dai dati, mal posto in caso contrario.

## 1.1 Metodo delle equazioni normali

La tecnica classica per risolvere un problema ai minimi quadrati è quella che opera risolvendo il sistema normale associato al problema.

Se la funzione  $f(x) = b^T x$ , con  $x, b \in \mathbb{R}^n$  possiamo facilmente verificare che

$$\nabla(b^T x) = b$$

si può dimostrare che se  $A$  è una matrice  $n \times n$  allora,

$$\nabla(x^T Ax) = Ax + A^T x.$$

Segue che  $A$  è simmetrica

$$\nabla(x^T Ax) = 2Ax.$$

Dal momento che vogliamo minimizzare una quantità non negativa, possiamo considerare il quadrato della norma del residuo

$$\|Ax - b\|^2 = (Ax - b)^T (Ax - b) = x^T A^T Ax - 2x^T A^T b + b^T b.$$

Imponiamo l'annullarsi del gradiente

$$\frac{1}{2} \nabla(\|Ax - b\|^2) = A^T Ax - A^T b = 0$$

giungendo al sistema normale

$$A^T Ax = A^T b. \tag{1.2}$$

Se  $m \geq n$ , e il rango di  $A$  è pieno, allora la matrice  $A^T A$  è invertibile, e la soluzione del sistema lineare è unica

$$x_{LS} = (A^T A)^{-1} A^T b.$$

Se  $A$  ha rango inferiore a  $n$ , la matrice  $A^T A$  risulta singolare. In questo caso, tra gli infiniti vettori che soddisfano il sistema, di solito si assume come soluzione  $x_{LS}$  quella dotata di minima norma euclidea (soluzione normale).

La matrice

$$A^+ = (A^T A)^{-1} A^T$$

è detta *pseudo-inversa* di A, o inversa di Moore-Penrose. Essa è un'inversa sinistra di A, ma non un'inversa destra. Nel caso a rango pieno la matrice  $A^T A$  è simmetrica definita positiva e per risolvere il sistema normale (1.2) si può fare ricorso alla fattorizzazione di Cholesky.

Una volta calcolata la fattorizzazione  $A^T A = R^T R$  con R matrice  $n \times n$  triangolare superiore, il vettore  $x_{LS}$  può essere calcolato risolvendo i due sistemi triangolari

$$\begin{cases} R^T y = A^T b \\ Rx = y \end{cases}$$

Il costo computazionale di tale metodo dipende dal calcolo del prodotto  $A^T A$  e della fattorizzazione di Cholesky, ed è pari a  $O\left(\frac{1}{2}n^2\left(m + \frac{1}{3}n\right)\right)$  moltiplicazioni.

## 1.2 Risoluzione mediante fattorizzazione QR

Tale risoluzione fornisce un metodo alternativo, stabile ed efficiente per calcolare la soluzione del problema (1.1). Nell'ipotesi che  $m \geq n$  consideriamo, anche in questo caso, il quadrato della norma del residuo, sostituendo al suo interno la fattorizzazione

$$A = QR$$

otteniamo

$$\|Ax - b\|^2 = \|QR - b\|^2 = \|Q(Rx - Q^T b)\|^2 = \|Rx - c\|^2,$$

ponendo  $c = Q^T b$  e ricordando che una matrice ortogonale non modifica la norma-2 di un vettore<sup>2</sup>.

---

<sup>2</sup> O anche detta norma euclidea, è una norma che denota la lunghezza (o modulo) del vettore. Dato un vettore  $x = (x_1, x_2, \dots, x_n)$ , la norma euclidea di x si indica con  $\|x\|_2$  ed è definita come

La matrice  $R$  è della forma

$$R = \begin{pmatrix} R_1 \\ 0 \end{pmatrix}$$

con  $R_1$  triangolare superiore, quadrata e, se  $A$  rango pieno, non singolare. Partizionando in modo coerente anche il vettore  $c$

$$c = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}, \quad \text{con } c_1 \in \mathbb{R}^n, c_2 \in \mathbb{R}^{m-n}.$$

Sostituendo otteniamo,

$$\|Ax = b\|^2 = \|Rx - c\|^2 = \left\| \begin{pmatrix} R_1 x \\ 0 \end{pmatrix} - \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \right\|^2 = \|R_1 x - c_1\|^2 + \|c_2\|^2$$

se  $\det(R_1) \neq 0$ , il sistema

$$R_1 x = c_1$$

ammette una sola soluzione, per la quale si ha  $\|R_1 x - c_1\|^2 = 0$ . In corrispondenza a tale soluzione

$$\min_{x \in \mathbb{R}^n} \|Ax = b\|^2 = \|c_2\|^2.$$

perciò l'eventuale annullarsi del vettore  $c_2$  significa che  $x$  è la soluzione classica del sistema iniziale  $Ax = b$ . In caso contrario, essa è la soluzione nel senso dei minimi quadrati, e la norma di  $c_2$  fornisce la misura del residuo. Rispetto al caso lineare, questo metodo ha il vantaggio di operare direttamente sulla matrice  $A$ , il cui condizionamento è pari alla radice quadrata di quello di  $A^T A$ . La complessità computazionale coincide con quella della fattorizzazione QR.

---

$$\|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$$



### 1.2.1 Fattorizzazione $QR$

Ogni matrice  $A$   $m \times n$  è fattorizzabile nella forma:

$$A = QR$$

in cui  $Q$  è una matrice di dimensione  $m \times n$  ortogonale, mentre  $R$  è triangolare superiore con le stesse dimensioni di  $A$ .

Realizzando una fattorizzazione di questo tipo, la matrice  $A$  può essere sostituita dal prodotto  $QR$ :

$$Ax = b \rightarrow QRx = b.$$

Poiché  $Q^T Q = Q Q^T = I$ , possiamo riscriverla come:

$$Q^T QRx = Q^T b$$

e perciò

$$Rx = Q^T b = c$$

Quando  $A$  è una matrice quadratica non singolare, la fattorizzazione  $QR$  può essere utilizzata per la risoluzione del sistema lineare  $Ax = b$ . Sostituendo alla matrice  $A$  la sua fattorizzazione  $QR$ , infatti, si ottengono i due sistemi:

$$\begin{cases} Qc = b \\ Rx = c \end{cases}$$

È possibile dimostrare peraltro che, se  $A = QR$ , allora:

- $\|A\|_2 = \|R\|_2$
- $k_2(A) = k_2(R)$

Ciò significa che il sistema iniziale è stato trasformato in un sistema equivalente, triangolare superiore, tale per cui il numero di condizionamento sia rimasto immutato.

# Capitolo 2

## L'ANALISI DI FOURIER

In analisi matematica, l'analisi di Fourier, nota anche come analisi armonica, è una branca di ricerca che ha preso avvio dalle ricerche di Jean Baptiste Joseph Fourier che, nei primi anni dell'Ottocento, riuscì a dimostrare matematicamente come una qualunque funzione periodica poteva essere scomposta in una somma di infinite "opportune" funzioni o componenti sinusoidali (seno e coseno) dette *armoniche*. Da tale constatazione nasce dunque l'idea di scomporre funzioni complicate in una serie di funzioni, nota come serie di Fourier, rendendone l'analisi più semplice e vantaggiosa. Dal concetto matematico di serie di Fourier discende anche la nozione di trasformata di Fourier ed il relativo concetto associato di dominio della frequenza. [2]

### 2.1 Serie di Fourier

Se consideriamo una funzione  $F$  integrabile in un intervallo compreso tra  $[-L, L]$ , ad essa può essere associata, in modo univoco, la sua serie di Fourier

$$f(x) = a_0 + \sum_{n=1}^{\infty} \left( a_n \cos n \frac{\pi}{L} x + b_n \sin n \frac{\pi}{L} x \right),$$

dove, per l'ortogonalità delle funzioni di base 1,  $\cos n \frac{\pi}{L} x$ ,  $\sin n \frac{\pi}{L} x$ ,

$$a_0 = \frac{1}{2L} \int_{-L}^L f(x) dx,$$

$$a_n = \frac{1}{L} \int_{-L}^L f(x) \cos n \frac{\pi}{L} x dx,$$

$$b_n = \frac{1}{L} \int_{-L}^L f(x) \sin n \frac{\pi}{L} x dx.$$

La serie fornisce una estensione periodica della funzione  $F$  su tutta la retta reale, con periodo  $T=2L$ . Vi sono, a seconda degli ambienti, diverse formulazioni delle serie di Fourier. Sia  $F$  una funzione periodica con periodo fondamentale  $2L$  i cui valori sono compresi tra  $[-L, L]$ . La sua serie di Fourier è

$$f(x) = a_0 + \sum_{n=1}^{\infty} (a_k \cos n\omega x + b_k \sin n\omega x)$$

con  $\omega = \frac{\pi}{L}$ . Possiamo sostituire, per semplicità, tale forma con la seguente

$$a_0 + \sum_{k=1}^{\infty} c_k \cos(k\omega x + \delta_k). \quad (2.1)$$

essa prende il nome di forma armonica. È sufficiente determinare i coefficienti  $c_k$  e  $\delta_k$  in modo che, per ogni  $k = 1, 2, \dots$ , risulti

$$\begin{aligned} a_k \cos k\omega x + b_k \sin k\omega x &= c_k \cos(k\omega x + \delta_k) \\ &= c_k \cos k\omega x \cos \delta_k - c_k \sin k\omega x \sin \delta_k, \end{aligned}$$

ricaviamo  $c_k$  e  $\delta_k$  dal sistema

$$\begin{cases} c_k \cos \delta_k = a_k, \\ c_k \sin \delta_k = -b_k, \end{cases}$$

da cui segue

$$\begin{cases} c_k = \sqrt{a_k^2 + b_k^2}, \\ \delta_k = \arctan\left(-\frac{b_k}{a_k}\right), \end{cases} \quad (2.2)$$

se  $a_k \neq 0$ , invece per  $a_k = 0$ , abbiamo

$$\begin{cases} c_k = |b_k|, \\ \delta_k = \frac{\pi}{2} \text{ se } b_k < 0 \text{ e } \delta_k = \frac{3}{2}\pi \text{ se } b_k > 0. \end{cases}$$

Se  $f$  è periodica con periodo fondamentale  $2L$ , per forma armonica della  $f$  si intende la (2.1) con i coefficienti  $c_k$  e gli angoli  $\delta_k$  determinati nella (2.2). Un altro modo per rappresentare le serie di Fourier più adatto nell'analisi dei segnali digitali è la forma complessa. Per la sua introduzione è essenziale ricordiamo la formula di Eulero:

$$e^{i\phi} = \cos\phi + i \sin\phi, \quad \forall \phi \in \mathbb{R},$$

dalla quale ne consegue immediatamente che

$$\cos = \frac{e^{i\phi} + e^{-i\phi}}{2} \quad \text{e} \quad \sin\phi = \frac{e^{i\phi} - e^{-i\phi}}{2i}$$

L'utilizzo di queste rappresentazioni ci permette di passare facilmente dalla forma trigonometrica a quella complessa. Sostituendola alla serie:

$$a_0 + \sum_{k=1}^{\infty} (a_k \cos k\omega x + b_k \sin k\omega x)$$

essa assume la forma

$$a_0 + \sum_{k=1}^{\infty} \left( a_k \frac{e^{ik\omega x} + e^{-ik\omega x}}{2} + b_k \frac{e^{ik\omega x} - e^{-ik\omega x}}{2i} \right) =$$

$$a_0 + \sum_{k=1}^{\infty} \left( \frac{1}{2} (a_k - ib_k) e^{ik\omega x} + \frac{1}{2} (a_k + ib_k) e^{-ik\omega x} \right),$$

ponendo

$$c_0 = a_0, \quad c_k = \frac{1}{2}(a_k - ib_k), \quad c_{-k} = \frac{1}{2}(a_k + ib_k), \quad k = 1, 2, \dots, \quad (2.3)$$

otteniamo, in forma più compatta,

$$S_f(x) = \sum_{k=-\infty}^{\infty} c_k e^{ik\omega x}$$

Possiamo osservare che, se si conosce la forma complessa si può facilmente passare alla forma trigonometrica, difatti, prendendo nuovamente la formula di Eulero

$$e^{ik\omega x} = \cos \omega x + i \sin \omega x, \quad \text{per } k = \pm 1, \pm 2, \dots,$$

$$e^{ik\omega x} = 1, \quad \text{per } k = 0$$

Partendo dalla funzione  $F$  anche i coefficienti  $c_k$  possono essere espressi

$$c_0 = \frac{1}{2L} \int_{-L}^L f(x) dx$$

e, per  $k = 1, 2, \dots$ ,

$$c_k = \frac{1}{2L} \int_{-L}^L f(x) e^{-ik\omega x} dx.$$

Quest'ultima rappresenta la forma complessa della serie di Fourier di una funzione di periodo  $T=2L$ . È utile osservare che l'eventuale simmetria (parità-disparità) della  $f$  fornisce importanti informazioni anche sui coefficienti della forma complessa della serie di Fourier. La (2.3) mostra che, se la  $f$  è pari tutti i coefficienti  $c_k$  sono reali, dato che gli  $a_k$  sono reali e  $b_k$  i nulli, se la  $f$  è dispari  $c_0=0$  e tutti i  $c_k$  sono immaginari puri, dato che gli  $a_k$  sono nulli e i sono  $b_k$  reali.

## 2.2 Trasformata di Fourier

Lo sviluppo di una funzione mediante le serie di Fourier è possibile per le funzioni periodiche. In questo caso, le frequenze necessarie sono gli interi, oppure i multipli di  $\frac{2\pi}{T}$  per le funzioni aventi periodo  $T$ .

Per le funzioni non periodiche, invece, tutte le frequenze sono ammissibili e la serie deve essere sostituita da un integrale. I coefficienti  $a_k$  e  $b_k$  della forma trigonometrica e i  $c_k$  della forma complessa, diventano i valori della trasformata di Fourier  $\mathcal{F}\{f\}$  calcolata in  $k$ , la quale è definita sull'intera retta reale, che dà una misura della densità di ciascuna frequenza  $k \in \mathbb{R}$  nella funzione  $f$ .

Per capire meglio il passaggio da una funzione periodica ad una funzione definita su tutto  $\mathbb{R}$  dobbiamo considerare i coefficienti e far tendere all'infinito l'ampiezza del periodo.

Sia  $f(x), x \in \mathbb{R}$ , integrabile su tutto  $\mathbb{R}$

$$\int_{-\infty}^{\infty} |f(x)| dx < \infty$$

e sviluppabile in serie di Fourier su ogni intervallo  $[-L, L]$ , indichiamo con  $f_T(x)$  la sua estensione periodica di periodo  $T$  su un intervallo  $[-\frac{T}{2}, \frac{T}{2}]$  e studiamo al tendere di  $T$  all'infinito, i suoi coefficienti di Fourier. Si ha che

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

Per tale espressione periodica si possono usare sia la forma trigonometrica che quella complessa. Utilizzando la forma complessa, i coefficienti di Fourier di  $f_T$ , di periodo  $T$ , sono:

$$c_k = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f_T(x) e^{-ik\omega x} dx ,$$

dove  $\omega = \frac{2\pi}{T}$ . Si può dunque scrivere

$$f_T(x) = \sum_{k=-\infty}^{\infty} c_k e^{-ik\omega x} = S_f(x) = \sum_{k=-\infty}^{\infty} \left[ \frac{1}{T} \int_{-\infty}^{\infty} f_T(x) e^{ik\frac{2\pi}{T}x} dx \right] e^{ik\frac{2\pi}{T}x} \Delta k$$

dove  $\Delta k = 1$ , rappresenta la variazione dell'indice  $k$ . Per far tendere  $T \rightarrow \infty$  si effettua un cambio di variabile  $s = \frac{2\pi}{T}k$  da cui otteniamo  $\Delta k = \frac{T}{2\pi} \Delta s$ . Il cambio di variabile conduce a

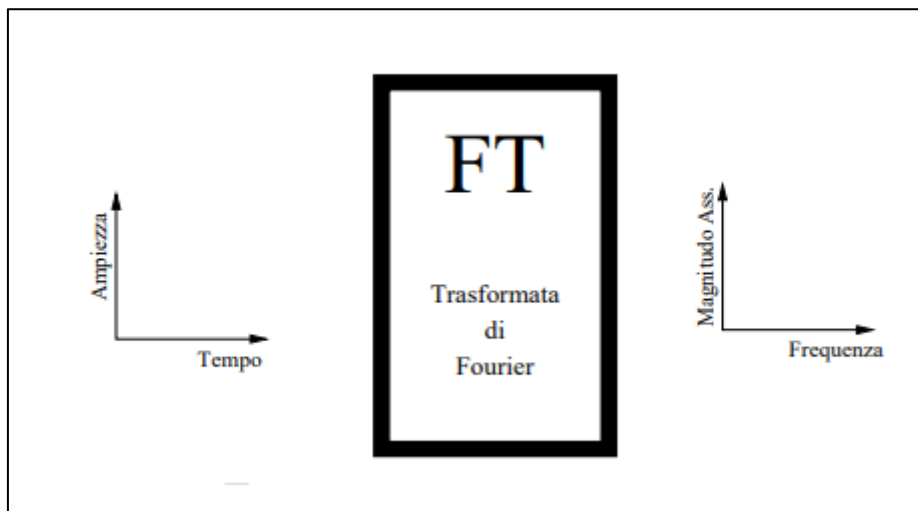
$$f_T(x) = \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} \left[ \int_{-\frac{T}{2}}^{\frac{T}{2}} f_T(x) e^{-isx} dx \right] e^{isx} \Delta s$$

e, passando al limite, la somma precedente diventa un'integrale rispetto a  $s$ ,

$$f_T(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \left[ \int_{-\infty}^{\infty} f(x) e^{-isx} dx \right] e^{isx} ds$$

Questa identità evidenzia che l'integrale interno trasforma una  $f(x)$  in una funzione di  $s$  e poi, mediante l'integrale, restituisce la  $f(x)$ . Ciò significa che una funzione definita in un dominio spaziale viene trasformata in un'altra, definita nel dominio delle frequenze, e che questa, mediante una trasformazione inversa, viene ricondotta nel dominio spaziale. Indicando con  $F(k)$  l'integrale interno alle parentesi quadre, otteniamo

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{ikx} F(k) dk, \text{ dove } F(k) = \int_{-\infty}^{\infty} e^{-ikx} f(x) dx$$



**Figura 1:** La trasformata di Fourier.

La trasformata di Fourier, definita su tutto l'asse reale, è

$$\mathcal{F}\{f\} = F(k) = \int_{-\infty}^{\infty} e^{-ikx} f(x) dx. \quad (2.4)$$

definita nello spazio delle frequenze. Per trasformata inversa di Fourier di F si intende la funzione

$$\mathcal{F}^{-1}\{F\} = f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-ikx} F(k) dk,$$

definita nello spazio ordinario.

### 2.2.1 Trasformata di Fourier discreta

La trasformata di Fourier a tempo discreto (DFT) permette l'analisi in frequenza dei segnali a tempo discreto. I segnali a tempo discreto costituiscono un modello per i segnali digitali. Possiamo definire la DFT come:

$$X_n = \sum_{m=0}^{N-1} x_m e^{-i2\pi \frac{m}{N} n} \text{ con } m = 0, 1, 2, \dots, N-1$$

dove  $i$  è l'unità immaginaria e  $e^{\frac{2\pi i}{N}}$  è una radice N-esima primitiva dell'unità, dal momento che  $\left(e^{\frac{2\pi i}{N}}\right)^N = 1$ , e la fase è la più piccola possibile. Essa costituisce un'approssimazione<sup>3</sup> della sequenza di campioni della trasformata  $X(f) = \mathcal{F}\{x(t)\}$  calcolata per  $f = \frac{n}{NT_c}$  e divisa per  $T_c$ :

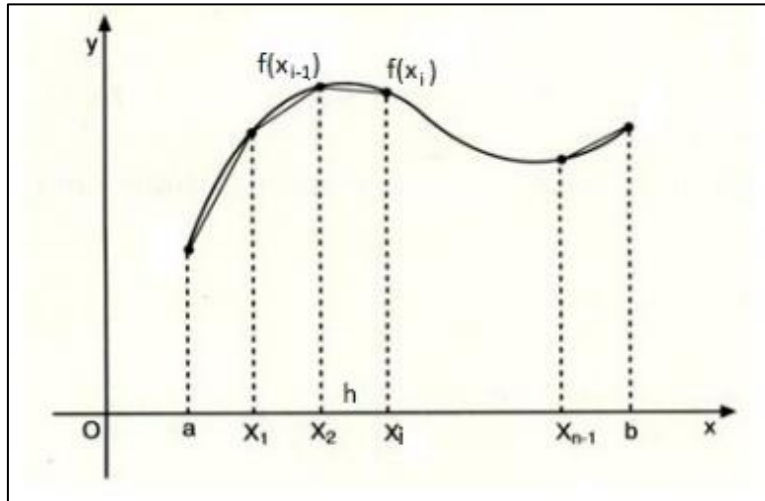
---

<sup>3</sup> Una prima fonte di approssimazione deriva dall'operazione di finestrazione legata all'uso di un numero finito di campioni, operando quindi su  $x(\omega) = x(t) \omega(t_c)$  anziché su  $x(t)$ . Scriviamo l'espressione di  $X_\omega(f) = \mathcal{F}\{x_\omega(t)\}$  per  $f = \frac{n}{NT_c}$ :

$$X_\omega\left(f = \frac{n}{NT_c}\right) = \int_0^{(N-1)T_c} x(t) e^{-i2\pi \frac{n}{NT_c} t} dt \simeq \sum_{m=0}^{N-1} x_m \cdot \int_0^{(N-1)T_c} \text{sinc}\left(\frac{t - mT_c}{T_c}\right) e^{-i2\pi \frac{n}{NT_c} t} dt$$

$$X_n \approx \frac{1}{T_c} X \left( f = \frac{n}{NT_c} \right)$$

Possiamo utilizzare l'espressione della DFT per approssimare l'integrale che ci permette di calcolare i coefficienti  $c_k$  della serie di Fourier, per farlo ci avvaliamo della *regola dei trapezi composta*.



**Figura 2:** Regola dei trapezi composta

In su la seconda eguaglianza utilizza l'interpolazione  $x(t) = \sum_{m=-\infty}^{\infty} x_m \cdot \text{sinc} \left( \frac{t-mT_c}{T_c} \right)$  fornita dal teorema di campionamento, e introduce una seconda approssimazione legata all'intervallo *finito* di variazione per  $m$ . l'ultimo integrale a sua volta è anch'esso un'approssimazione della trasformata calcolata in *sinc*, che quando valutata per  $f = \frac{n}{NT_c}$ , fornisce

$$X_\omega \left( f = \frac{n}{NT_c} \right) = T_c \sum_{m=0}^{N-1} x_m e^{-i2\pi \frac{m}{N} n}$$

per valori di  $|n| \leq \frac{N}{2}$ . È inoltre possibile verificare che  $X_\omega \left( \frac{n}{NT_c} \right)$  è periodica in  $n$  con periodo  $N$ , quindi i valori assunti per  $n = \frac{N}{2} + 1, \frac{N}{2} + 2, \dots$ , sono uguali a quelli per  $n = -\frac{N}{2} + 1, -\frac{N}{2} + 2, \dots$ ,



Consideriamo una funzione nell'intervallo di integrazione  $[a, b]$ , e dividiamo tale intervallo in  $N + 1$  punti che delimitano gli  $N$  sottointervalli, tutti di eguale ampiezza. Ricordiamo che i coefficienti della serie di Fourier possiamo calcolarli come

$$c_k = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(x) e^{-\frac{i2\pi mx}{T}} dx,$$

con  $T=2L$  e  $\omega=\frac{2\pi}{T}$ .

L'integrale è definito nell'intervallo di ampiezza  $T$ , che suddividiamo in  $N$  sottointervalli ciascuno di ampiezza  $h = \Delta x = \frac{T}{N}$ . Gli  $N+1$  punti compresi tra  $[-\frac{T}{2}, \frac{T}{2}]$ . Ponendo  $g(x) = f(x)e^{-i2\pi mx/T}$  approssimiamo l'integrale come

$$\int_{-\frac{T}{2}}^{\frac{T}{2}} g(x) dx = \frac{\Delta x}{2} \left[ g\left(-\frac{T}{2}\right) + 2 \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}-1} g(x_n) + g\left(\frac{T}{2}\right) \right],$$

poiché è una funzione periodica

$$g\left(-\frac{T}{2}\right) = g\left(\frac{T}{2}\right),$$

quindi possiamo approssimare i coefficienti

$$c_k = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} g(x) dx = \frac{1}{T} \frac{\Delta x}{2} \left[ 2 \sum_{n=-\frac{N}{2}}^{\frac{N}{2}-1} g(x_n) \right] = \frac{1}{N} \sum_{n=-\frac{N}{2}}^{\frac{N}{2}-1} f(x_n) e^{-\frac{i2\pi mn}{N}}$$

Quest'ultima coincide con la definizione di trasformata discreta data all'inizio. L'importanza della DFT è dovuta non solo al fatto che è necessario, anche per l'elaborazione dei segnali continui, che questi vengano prima campionati e discretizzati, ma anche perché per calcolarla è possibile implementare un algoritmo ottimizzato, noto come *trasformata di Fourier veloce* (FFT). Grazie ad esso si abbassa drasticamente la complessità computazionale; infatti per la trasformata discreta sono richieste  $O(n^2)$  operazioni, mentre con la trasformata veloce otteniamo lo stesso risultato in  $O(n \log_2(n))$  operazioni.

### 2.2.2 Trasformata di Fourier a tempo breve

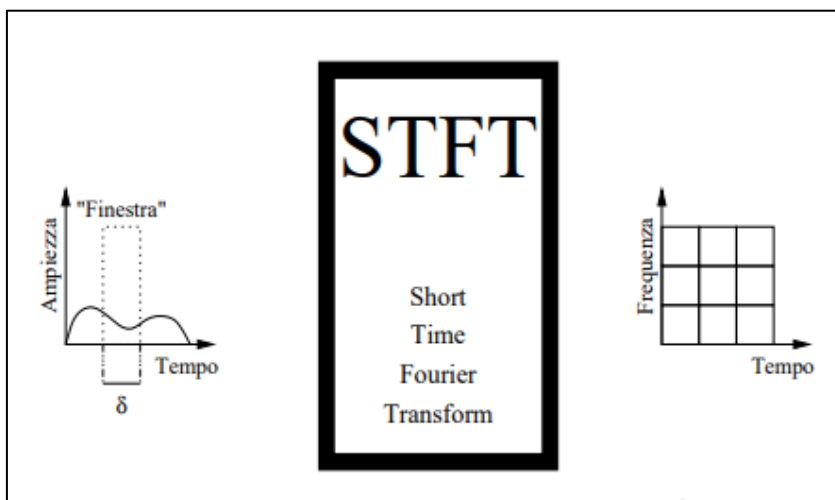
La *trasformata di Fourier* definita nella (2.4) è ideale per studiare segnali stazionari, in cui, cioè, le proprietà sono statisticamente invarianti nel tempo. La maggior parte

dei segnali e processi però sono non-stazionari, ossia essi evolvono nel tempo.<sup>4</sup> Un modo semplice per analizzare segnali non-stazionari è quello di utilizzare una rappresentazione che sia locale sia nel dominio del tempo che in quello delle frequenze. Il problema è che la trasformata di Fourier non permette la localizzazione temporale, una via alternativa per ottenere una trasformata tempo-frequenza consiste nel “tagliare” il segnale in parti e applicare un’analisi di Fourier su ciascuna di esse. Poiché le funzioni ottenute da una segmentazione di questo tipo non sono periodiche, viene introdotto il concetto di *windowing*; invece di localizzare il segnale mediante una funzione rettangolare, si usa una funzione finestra  $g(t)$ . Quindi prima si individua la porzione di segnale da elaborare e poi si effettua la trasformata di Fourier. Tale procedura viene detta *Short-time Fourier Transform* o *windowed Fourier transform* o ancora, *trasformata di Gabor*<sup>5</sup>.

Possiamo definire la *trasformata di Fourier a tempo breve (STFT)* come:

$$STFT[x(t)] = X(\tau, \omega) = \int_{-\infty}^{\infty} x(t)g(t - \tau)e^{-i\omega t} dt \quad (2.5)$$

Essa utilizza una versione traslata e modulata di  $g(t)$ , detta finestra elementare.

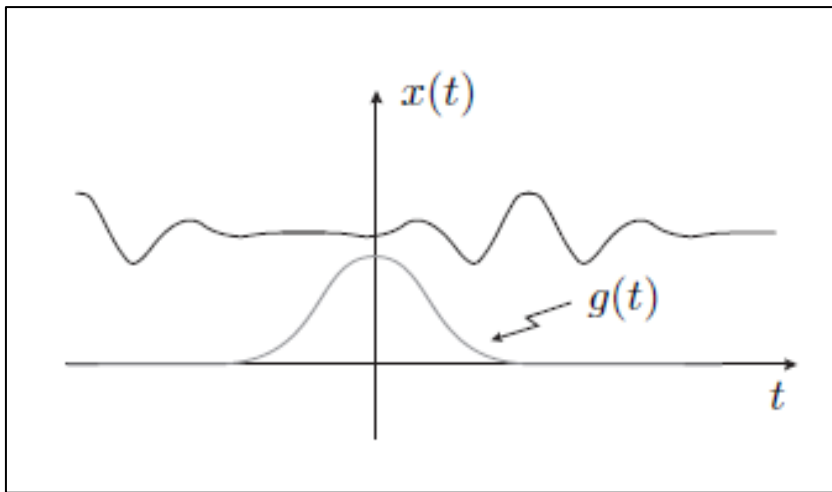


**Figura 3:** *Short Time Fourier Transform*

<sup>4</sup> Un esempio: i segnali vocali e la musica

<sup>5</sup> Il filtro di Gabor è un filtro lineare la cui risposta all’impulso è definita da una funzione armonica moltiplicata per una funzione Gaussiana. La trasformata di Fourier della risposta all’impulso di un filtro di Gabor, risulta essere la convoluzione fra la trasformata di Fourier della funzione armonica e la trasformata di Fourier della funzione Gaussiana. I filtri di Gabor sono strettamente legati alle wavelet di Gabor, e possono essere creati per infinite combinazioni di dilatazioni e rotazioni. L’elaborazione di un generico segnale avviene costruendo un banco di filtri di Gabor di differente scalatura e orientazione.

Generalmente è sconsigliato utilizzare una finestra rettangolare, poiché determina delle discontinuità “artificiali”, che non sono presenti nel segnale originale. Spesso si sceglie una finestra gaussiana del tipo  $g(t) = \alpha e^{-\beta t^2}$ .

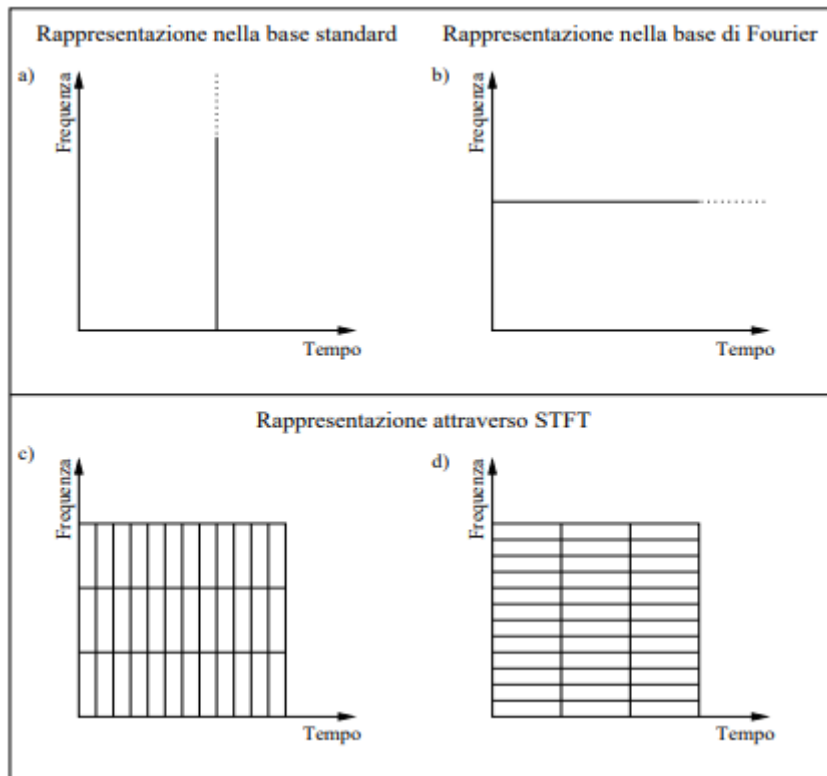


**Figura 4:** Finestra Gaussiana

Si dimostra che la trasformazione definita nella (2.5) è invertibile e la formula di ricostruzione può essere scritta nella forma:

$$x(t) = \frac{1}{2\pi} \iint_{-\infty}^{\infty} x(\tau, \omega) e^{-i\omega t} d\tau d\omega$$

Il problema principale di questa soluzione è dato dal fatto che l’ampiezza della finestra, una volta stabilita, non può essere modificata e rimane sempre la stessa sia per analizzare fenomeni alle basse frequenze che per analizzare fenomeni alle alte frequenze, i cui comportamenti sono invece molto diversi. Inoltre, la moltiplicazione nel dominio del tempo tra la funzione e la finestra equivale a fare una convoluzione tra i loro spettri nel dominio della frequenza, quindi le informazioni spettrali del segnale vengono alterate.



**Figura 5:** Le “scatole” di Heisenberg: a) la rappresentazione classica di un segnale in funzione del tempo fornisce una incertezza nulla sull’asse dei tempi ed una incertezza infinita sull’asse delle frequenze; b) la rappresentazione di un segnale attraverso l’analisi di Fourier garantisce una incertezza nulla sull’asse delle frequenze ed una incertezza infinita sull’asse dei tempi; c) e d) la STFT consente una rappresentazione del segnale sia in tempo che in frequenza tale che a basse incertezze sull’asse dei tempi corrispondono alte incertezze sull’asse delle frequenze, e viceversa.

# Capitolo 3

## La trasformata Wavelet

Nell'utilizzo della serie di Fourier per approssimare un segnale periodico, il sistema trigonometrico ha, insieme a tanti pregi, anche alcuni difetti.

1. I polinomi trigonometrici sono funzioni continue e rettangolari, perciò una serie di Fourier approssima male una funzione discontinua.
2. L'approssimazione con polinomi trigonometrici ha una natura globale: supponiamo di voler approssimare una funzione  $f \in L^2[-\pi, \pi]$  che ha un grafico abbastanza "semplice e liscio" in  $[-\pi, 0]$  e piuttosto frastagliato in  $[0, \pi]$ . Affinché la somma parziale di Fourier segua bene il grafico di  $f$  in  $[0, \pi]$  dovremmo aumentare il numero di termini, i cui coefficienti sono stati calcolati in base al comportamento di  $f$  su tutto  $[-\pi, \pi]$ ; ciò significa che l'unico modo per migliorare i dettagli di approssimazione in  $[0, \pi]$  è quello di aumentare i dettagli in tutto  $[-\pi, \pi]$ .

Per ovviare a questi problemi, in particolare al secondo, si possono utilizzare altri tipi di sistemi ortonormali completi di  $L^2[-\pi, \pi]$ , che hanno la proprietà di permettere un'analisi su scale diverse in parti diverse dell'intervallo. Sistemi di questo tipo sono per l'appunto le *wavelets*.

Il concetto di *wavelet (ondina)* fu introdotto per la prima volta dal geofisico francese J. Morlet attorno al 1975. Insieme al fisico francese A. Grossmann; Morlet mise a punto, agli inizi degli anni Ottanta, la *trasformata wavelet continua*. La trasformata wavelet è un potente strumento per l'analisi e l'elaborazione dei segnali e risulta estremamente efficiente in diversi campi di applicazione, come la compressione, il *denoising* di segnali e, in generale, quando si ha a che fare con segnali non stazionari, come le immagini. Queste infatti sono caratterizzate da andamenti di lungo periodo alle basse frequenze (sfondi, background), e andamenti di breve periodo alle alte frequenze (discontinuità, bordi). Essa viene spesso paragonata alla *trasformata di Fourier*, dove i segnali sono rappresentati come somma di armoniche. La differenza principale è che le wavelet sono localizzate sia nel tempo sia nella frequenza mentre la trasformata di Fourier standard è localizzata solo in frequenza. Come abbiamo detto precedentemente, la *trasformata di Fourier a tempo breve (STFT)* è localizzata in tempo e in frequenza, ma ci sono dei problemi di risoluzione e le wavelet spesso offrono una migliore rappresentazione del segnale grazie all'uso dell'analisi multirisoluzione. [3] [4]

### 3.1 La Trasformata Wavelet Continua

La *STFT* risulta essere non adeguata all'elaborazione di segnali naturali, dato che non è possibile localizzare con la stessa precisione, fenomeni che evolvono rapidamente e fenomeni che invece evolvono lentamente. Per ottenere un'analisi a risoluzione variabile occorre far in modo che all'aumentare della frequenza, la banda aumenti in modo proporzionale. Questo si può ottenere costruendo un insieme di funzioni base mediante traslazione e cambiamento di scala di un'unica funzione  $\psi(t)$  detta *mother wavelet*:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right)$$

A partire da essa è possibile definire la *trasformata wavelet continua* (CWT)

$$\begin{aligned} CWT[x(t)] = CWT[a, b] &= \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t) \psi\left(\frac{t-b}{a}\right) dt \\ &= \langle x(t), \psi_{a,b}(t) \rangle \end{aligned} \tag{3.1}$$

Si dimostra che la trasformata che ci permette di ricostruire il segnale originale è

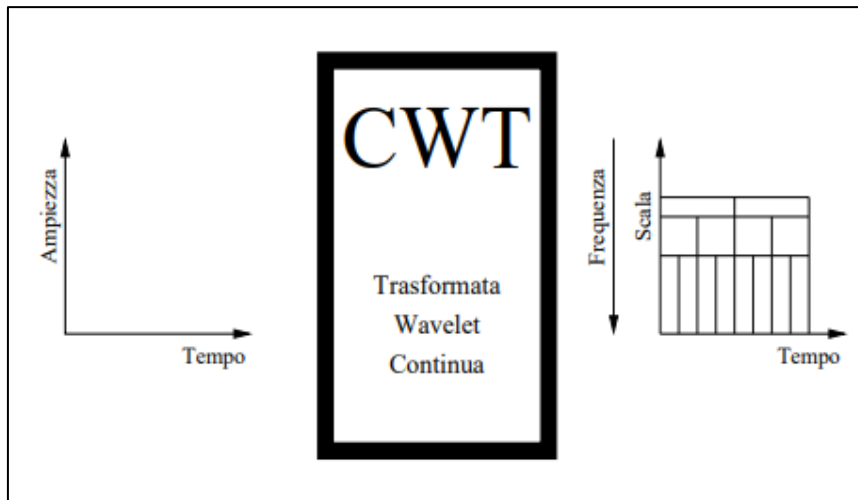
$$x(t) = \frac{1}{C_\psi} \iint_{-\infty}^{\infty} \frac{1}{a^2} CWT[a, b] \psi_{a,b}(t) da db$$

dove la costante  $C_\psi$  deve essere una quantità finita:

$$C_\psi = \int_0^{+\infty} \frac{|\Psi(f)|^2}{f} df < \infty$$

con  $\Psi(f) = \mathfrak{F}(\psi(t))$ .

La Wavelet  $\psi(t)$  deve quindi avere derivata continua, perché non avere discontinuità garantisce che la sua trasformata decada in modo sufficientemente veloce e  $C_\psi$  risulta finito. I parametri  $a$  e  $b$  sono detti rispettivamente parametro di traslazione e parametri di dilatazione. La traslazione è relativa alla posizione della finestra, ed essa viene spostata attraverso il segnale mentre, il parametro di dilatazione o scala utilizzato nell'analisi wavelet è simile alla scala usata nelle mappe: alte scale corrispondono ad una visione globale non dettagliata e scale basse corrispondono a una vista dettagliata (del segnale). In termini di frequenza, le frequenze basse (alte scale) corrispondono ad una informazione globale del segnale, che di solito si estende per l'intero segnale, mentre le alte frequenze (basse scale) corrispondono ad informazioni più dettagliate.



**Figura 6:** Analisi e risoluzioni variabili tramite CWT.

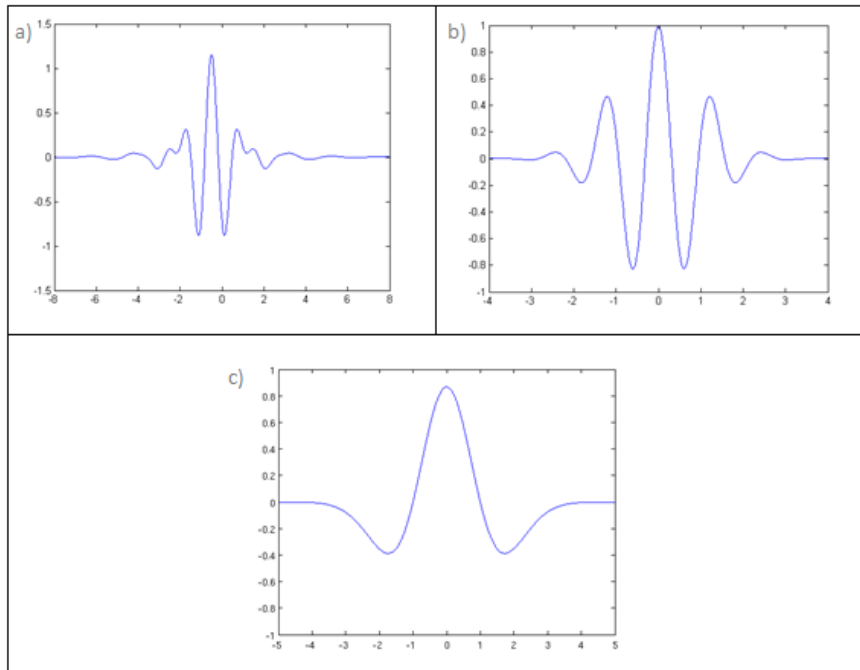
L'unica condizione da imporre per garantire che la costante  $C_\psi$  non diverga è  $\Psi(0) = 0$ , che equivale a dire che il valor medio della *Mother wavelet* deve essere nullo dato che:

$$\Psi(0) = \int_{-\infty}^{\infty} \psi(t) dt$$

Nella maggior parte delle situazioni è utile richiedere che  $\psi(t)$  sia continua e che abbia un numero alto  $M$  di momenti nulli, cioè per ogni intero  $m < M$ .

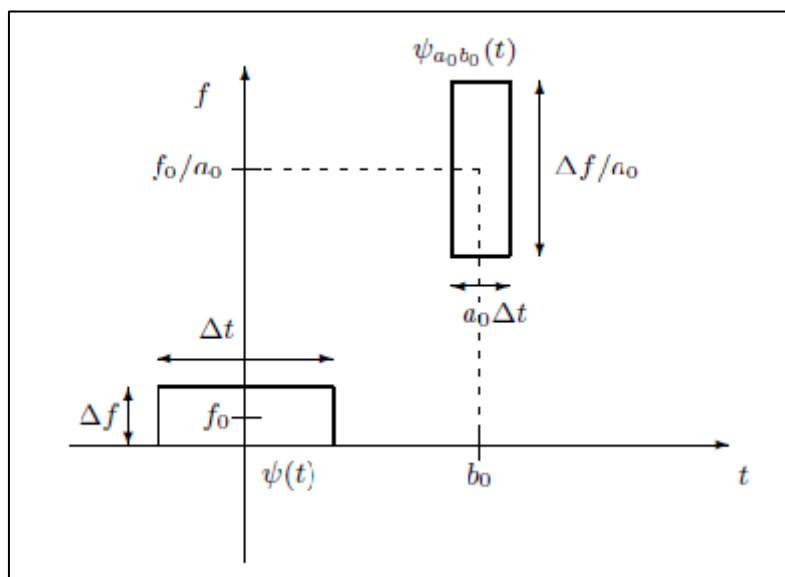
$$\int_{-\infty}^{\infty} t^m \psi(t) dt = 0$$

Questo significa che la *Mother wavelet* deve essere non nulla e avere una media nulla. In figura possiamo vedere alcuni esempi di *Mother wavelet*:



**Figura 7:** a) Meyer b) Morlet c) Mexican hat.

Dunque, la CWT si calcola intuitivamente traslando una wavelet sul segnale di analisi e calcolando la correlazione tra i due. Queste caratteristiche sono necessarie per la perfetta ricostruzione del segnale a partire dalla sua trasformata, tenendo presente che la funzione wavelet si comporta come un filtro passa-banda.



**Figura 8:** Finestra di tipo wavelet. All'aumentare della frequenza (scale basse) la risoluzione nel tempo aumenta, mentre la risoluzione in frequenza diminuisce.



Inoltre, ricordiamo che la CWT non fornisce un'analisi in tempo-frequenza ma in tempo-scala, dalla definizione di  $\psi_{a,b}(t)$  si nota che a scale basse la wavelet è una versione compressa della wavelet madre, e che corrisponde alle alte frequenze, ugualmente le scale alte rappresentano segnali che cambiano poco nel tempo, e le wavelet usate per le approssimazioni sono versioni dilatate della wavelet madre.

### 3.2 La Trasformata Wavelet discreta

La *Trasformata Wavelet discreta (DWT)* è una rappresentazione in scala temporale di un segnale digitale che viene ottenuta tramite opportune tecniche di filtraggio digitale. I filtri di diverse frequenze di taglio sono utilizzati per analizzare il segnale a diverse scale, viene poi fatto passare attraverso una serie di filtri passa-alto per analizzare alte frequenze, e attraverso dei filtri passa-basso per analizzare basse frequenze. Tuttavia, pur avendo grandi vantaggi la *trasformata wavelet discreta*, comporta una serie di problematiche; la valutazione dell'integrale di analisi risulta essere un calcolo troppo oneroso, inoltre la trasformazione produce valori su scale continue e traslazioni continue; la studiosa belga Daubechies ha dimostrato che con una scelta opportuna e minimale di parametri di traslazione e di scala, è possibile ricostruire perfettamente il segnale, il che comporta un certo grado di ridondanza<sup>6</sup> presente nella CWT. [5] [6]

Con un adeguata scelta dei parametri  $a$  e  $b$ , la trasformata perde la ridondanza e il calcolo diventa molto più efficace. La *DWT* mantiene abbastanza informazioni sul segnale in modo tale che sia possibile ricostruirlo in maniera perfetta, a partire da alcuni coefficienti analoghi ai coefficienti presenti nella serie di Fourier.

Prendendo nuovamente in considerazione la (3.1), la trasformata wavelet discreta consiste nel prendere i parametri  $a$  e  $b$  nella forma:

$$a \rightarrow 2^{-s} \qquad b \rightarrow k2^{-s}$$

per  $k, s \in \mathbb{Z}$ .

La trasformata viene dunque effettuata su una scala logaritmica; scegliendo la base 2 si parla di campionamento *diadico*<sup>7</sup>. Con questo tipo di campionamento, la (3.1) diventa

$$\begin{aligned} CWTf[2^{-s}, k2^{-s}] &= 2^{s/2} \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \psi\left(\frac{t - k2^{-s}}{2^{-s}}\right) dt \\ &= 2^{2/s} \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \psi(2^{-s}t - k) dt \end{aligned}$$

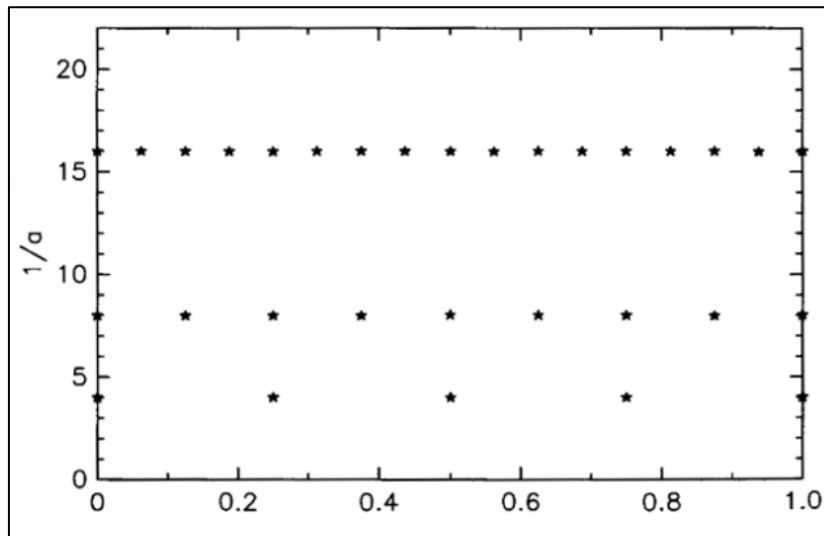
---

<sup>6</sup> Ridondanza; il fatto che molti dei valori prodotti dalla trasformata non sono in realtà necessari alla ricostruzione del segnale.

<sup>7</sup> Relativo a una coppia (o diade), è un'operazione mediante la quale una coppia di elementi, in un insieme, si fa corrispondere un elemento del medesimo insieme.

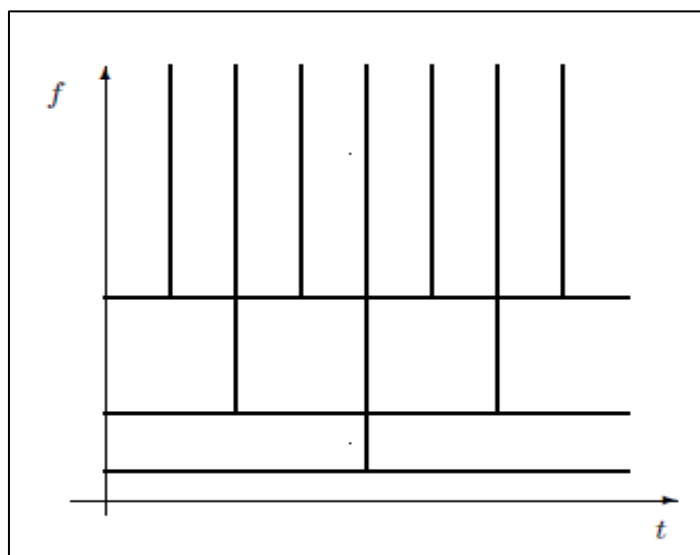
La *DWT* genera quindi un insieme sparso di valori sul piano che rappresentano la correlazione tra il segnale e la wavelet.

A frequenze basse si ha una buona risoluzione in frequenza ma bassa risoluzione nel tempo, e le wavelet catturano fenomeni di lunga durata. A frequenze alte esse hanno un supporto piccolo e la risoluzione nel tempo migliora rendendo possibile il rilevamento di dettagli, mentre la risoluzione in frequenza peggiora.



**Figura 9:** Campionamento diadico piano tempo-frequenza.

Per il principio di Heisemberg non si può avere una localizzazione veramente istantanea, ma possiamo solo sapere quali bande esistono a quali intervalli di tempo. La *DWT* si presta ad applicazioni di elaborazione dei segnali piuttosto che di analisi in tempo-frequenza.



**Figura 10:** Copertura nel piano tempo-frequenza

### 3.3 Analisi Multirisoluzione

La teoria multirisoluzione (MRA) è legata alla rappresentazione e all'analisi di segnali (o immagini) a più risoluzioni con il chiaro obiettivo di individuare le caratteristiche di interesse alla risoluzione in cui quest'ultime siano più facilmente individuabili.

Consideriamo un generico segnale  $f(t)$  che può essere rappresentata come una combinazione lineare di funzioni di espansione:

$$f(t) = \sum_k c_k \varphi_k(t)$$

dove i  $c_k$  sono i coefficienti, dati dalla relazione<sup>8</sup>:

$$c_k = \langle f(t), \varphi_k(t) \rangle = \int_{-\infty}^{\infty} f(t) \varphi_k(t) dt = \int_k^{k+1} f(t) dt$$

L'insieme delle funzioni di base  $\{\varphi_k(t)\}$  genera uno spazio di funzioni che viene definito *span chiuso* dell'insieme di espansione e viene indicato come:

$$V = \overline{\text{Span}_k\{\varphi_k(t)\}}$$

per cui, dire che la funzione  $f(t)$  appartiene allo spazio di  $V$  vuol dire che appartiene allo *span chiuso* di  $\{\varphi_k(t)\}$

### 3.4 Funzione di Scaling

Consideriamo come insieme di funzioni di espansione quello formato dalla funzione  $\varphi_k$ , e dalle sue traslazioni intere e ridimensionamenti in scala binaria, otteniamo l'insieme  $\{\varphi_{i,k}(t)\}$  dove

$$\varphi_{i,k}(t) = 2^{\frac{i}{2}} \varphi(2^i t - k)$$

Le funzioni di  $\varphi_{i,k}(t)$  sono dette *funzioni di scaling* perché al variare di  $i=0, \dots, n$ , forniscono una rappresentazione del segnale  $f(t)$  a diverse risoluzioni:

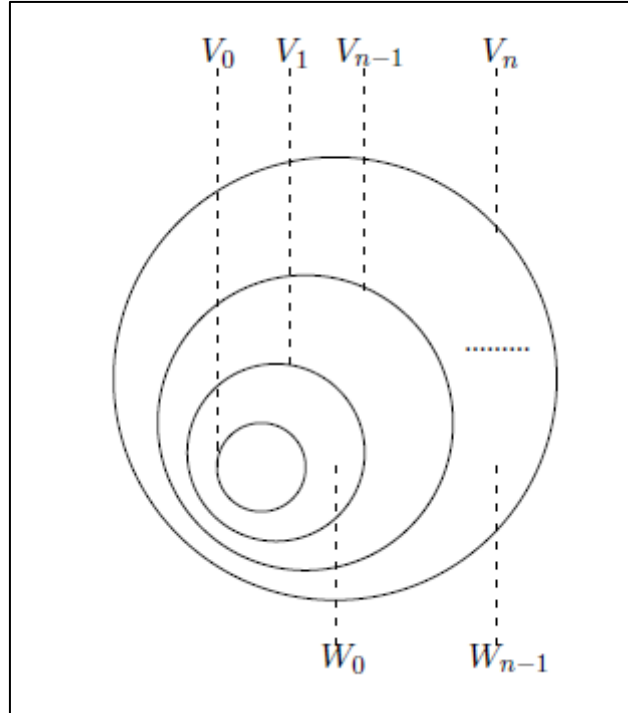
$f^0(t), f^1(t), \dots, f^n(t)$ .

Il termine  $k$  determina la posizione di  $\varphi_{i,k}(t)$  lungo l'asse dei tempi, e  $i$  determina l'ampiezza di  $\varphi_{i,k}(t)$ , e dunque come si restringe o allarga lungo l'asse temporale, il termine  $2^{\frac{i}{2}}$  è una costante di normalizzazione.

Per ogni valore di  $i$  fissato si determina uno spazio  $V_i$  e se la funzione  $f(t) \in V_i$  può essere scritta come:

$$f(t) = \sum_k c_k \varphi_{i,k}(t).$$

Aumentando il valore di  $i$  aumenta anche la misura dello spazio  $V_i$  corrispondente, perché le funzioni  $\varphi_{i,k}(t)$  vengono compresse e sono in grado di individuare maggiori variazioni e di analizzare più dettagli. Si può quindi rappresentare un maggior numero di funzioni, che verranno incluse nel sottospazio in questione.



**Figura 11:** Rappresentazione mediante spazi vettoriali

La sequenza dei segnali  $f^0(t), f^1(t), \dots, f^n(t)$ , per come è stata costituita rappresenta una gerarchia di segnali a diversa risoluzione, che però risulta essere fortemente ridondante, essendoci una relazione di inclusione tra gli spazi:

$$V_{-\infty} \subset \dots \subset V_0 \subset V_1 \subset \dots \subset V_{\infty}$$

Al limite per  $i \rightarrow \infty$  lo spazio  $V_i$  dovrebbe includere tutte le funzioni del dominio di interesse; poiché stiamo considerando funzioni reali, misurabili e integrabili al quadrato, questo vuol dire che per noi

$$V_{\infty} = \{L^2(R)\}$$

---

<sup>8</sup> Quando si considerano espansioni in serie del tipo  $f(t) = \sum_n a_n \varphi_n(t)$ , dove  $\varphi_n(t)$  rappresentano un insieme di funzioni ortonormali tra loro, allora risulta:

$$\langle f, \varphi_k \rangle = \sum_n a_n \langle \varphi_n, \varphi_k \rangle = \sum_n a_n \delta_{n-k} = a_k$$

Cioè i coefficienti di espansione sono dati proprio dal prodotto scalare del segnale con le funzioni di  $\varphi$ .

e che le serie di funzioni di espansione del sottospazio  $V_i$  può essere determinata tramite somme pesate delle funzioni di espansione a più alta risoluzione del sottospazio  $V_{i+1}$ , in questo modo:

$$\varphi_{i,k}(t) = \sum_k d(k) 2^{\frac{i+1}{2}} \varphi(2^{i+1}t - k)$$

La scelta del sottospazio di riferimento è arbitraria, per cui riferendoci allo spazio  $V_0$  possiamo considerare  $\varphi(t) = \varphi_{0,0}(t)$ , con entrambi i parametri  $i$  e  $k$  nulli, e ottenere un'espressione più semplice

$$\varphi(t) = \sum_k d(k) \sqrt{2} \varphi(2t - k)$$

dove  $d(k)$  sono chiamati *coefficienti della funzione di scaling*.

Quest'ultima equazione è fondamentale per la risoluzione multirisoluzione e viene chiamata *equazione di dilatazione*.

### 3.5 Wavelet

Uno spazio a risoluzione  $i+1$  è formato da tutte le funzioni a risoluzione inferiore a cui vengono aggiunti dei dettagli e questi dettagli sono dati da una certa funzione wavelet  $\psi(t)$ . Tale funzione, con le sue traslazioni intere e i ridimensionamenti in scala binaria, dà origine ad uno *span chiuso* che è l'insieme delle differenze esistenti tra i due sottospazi adiacenti  $V_i$  e  $V_{i+1}$ .

Questa rappresentazione può essere visibile attraverso la Fig.8.

Come abbiamo visto per la funzione di scaling, abbiamo l'insieme  $\{\psi_{i,k}(t)\}$  formato dalle wavelet

$$\{\psi_{i,k}(t)\} = 2^{\frac{i}{2}} \psi(2^i t - k)$$

che determinano lo spazio

$$W_i = \overline{\text{Span}_k \{\psi_{i,k}(t)\}}.$$

Se la funzione  $f(t)$  appartiene a tale spazio allora potrà essere rappresentata tramite le wavelet di base  $\psi_{i,k}$

$$f(t) = \sum_k c_k \psi_{i,k}(t).$$

La relazione tra i sottospazi delle scaling function e i sottospazi delle wavelet

$$V_{i+1} = V_i \oplus W_i$$

dove  $\oplus$  è il simbolo che determina la somma diretta tra gli spazi, ossia la loro unione, con tutti i termini di  $V_i$  ortogonali rispetto a quelli di  $W_i$ .  
 Uno spazio di wavelet ad una certa risoluzione è incluso nello spazio delle funzioni di scaling a risoluzione più alta per la quale qualsiasi funzione wavelet può essere espressa mediante una somma pesata di funzioni di scaling traslate e a risoluzione doppia

$$\psi(t) = \sum_k q(k) \sqrt{2} \varphi(2x - k)$$

dove  $q(k)$  sono i coefficienti della funzione wavelet e sono legati ai coefficienti della funzione di scaling  $d(k)$  dalla relazione

$$q(k) = (-1)^k d(1 - k).$$

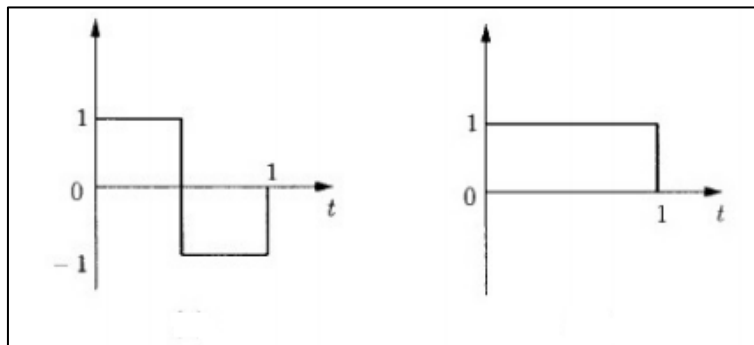
### 3.5.1 Wavelet di Haar

La sequenza Haar è stata proposta nel 1909 da Alfréd Haar. Haar utilizzò queste funzioni per dare un esempio di un sistema ortonormale per lo spazio di funzioni quadrate integrabili sull'intervallo unitario  $[0,1]$ . Essa è anche la più semplice *mother wavelet* a supporto compatto definita come:

$$\psi(t) = \begin{cases} 1, & 0 \leq t < 1/2, \\ -1, & \frac{1}{2} \leq t < 1, \\ 0 & \text{altrimenti.} \end{cases}$$

e la sua *father function*

$$\varphi(t) = \begin{cases} 1, & 0 \leq t < 1, \\ 0, & \text{altrimenti.} \end{cases}$$



**Figura 12:** Wavelet di Haar; scaling function.

In questo modo una funzione costante può essere rappresentata da un multiplo della  $\varphi(t)$ , che rappresenta il comportamento a grande scala (bassa frequenza), essa è anche chiamata *funzione di scalamento (scaling function)*.

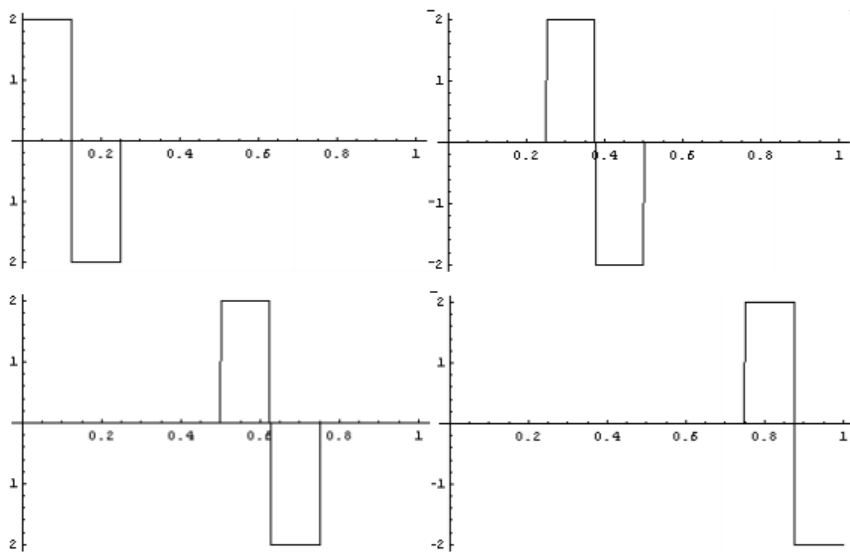
Lo svantaggio principale della wavelet è che non è continua<sup>9</sup>, quindi non differenziabile, tuttavia questa proprietà può essere un vantaggio per l'analisi dei segnali con transizioni brusche.

Per ogni coppia  $n, k$  degli interi  $\mathbf{Z}$ , la funzione di Haar  $\psi_{n,k}$  viene definita sulla retta reale  $\mathbf{R}$  della formula

$$\psi_{n,k}(t) = 2^{n/2}\psi(2^n t - k), \quad t \in \mathbf{R}.$$

Possiamo visualizzare i grafici di alcune di queste funzioni.

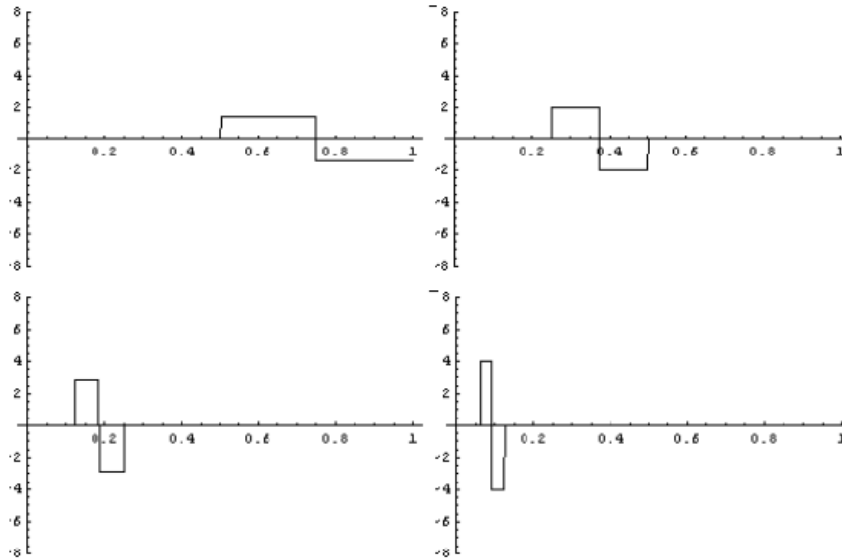
Ad esempio, per  $n=2$  e quindi per  $k=0, 1, 2, 3$  si ha:



I grafici per di  $\psi_{n,1}$  per i valori di  $n=1, 2, 3, 4$  sono:

---

<sup>9</sup> In matematica, una funzione continua è una funzione per la quale piccole variazioni del risultato comportano piccoli cambiamenti del valore.



Questa funzione è supportata sul giusto intervallo aperto  $L_{n,k} = [k2^{-n}, (k+1)2^{-n}]$ , cioè essa svanisce al di fuori di questo intervallo. Ha integrale 0 e norma 1 nello spazio di Hilbert<sup>10</sup>  $L^2(\mathbb{R})$ , notiamo che per tutte le funzioni  $\psi_{n,k}$  si ha:

$$\int_{\mathbb{R}} \psi_{n,k}(t) dt = 0, \text{ e } \|\psi_{n,k}\|_{L^2(\mathbb{R})}^2 = \int_{\mathbb{R}} \psi_{n,k}(t)^2 dt = 1$$

È facile rendersi conto che il prodotto tra due funzioni diverse tra le  $\psi_{n,k}$  o è identicamente nullo oppure è una delle due moltiplicata per una costante, perciò le funzioni di Haar sono anche a due a due ortogonali

$$\int_{\mathbb{R}} \psi_{n_1,k_1}(t) \psi_{n_2,k_2}(t) dt = \delta_{n_1,n_2} \delta_{k_1,k_2},$$

dove  $\delta_{i,j}$  rappresenta il delta di Kronecker<sup>11</sup>. Poiché tutte queste funzioni hanno media nulla, se vogliamo sperare di avere un sistema ortonormale completo è necessario aggiungere almeno una funzione che *non* abbia media nulla.

### 3.5.2 Famiglie di Wavelet

Oltre alle wavelet di Haar, ulteriori famiglie di wavelet che si sono dimostrate particolarmente utili sono le seguenti:

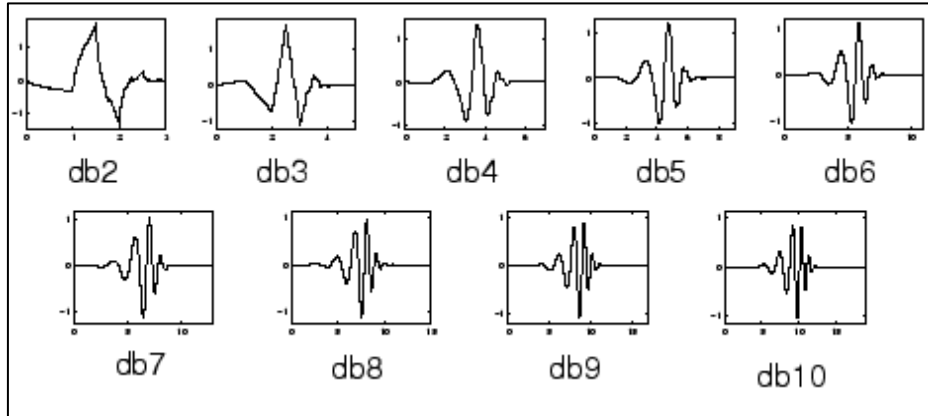
<sup>10</sup> È uno spazio vettoriale astratto che possiede la struttura di un prodotto interno che permette di misurare lunghezze e angoli.

<sup>11</sup> Il delta di Kronecker è una funzione di due indici interi, di solito solo interi. La funzione è 1 se gli indici sono uguali, e 0 altrimenti.

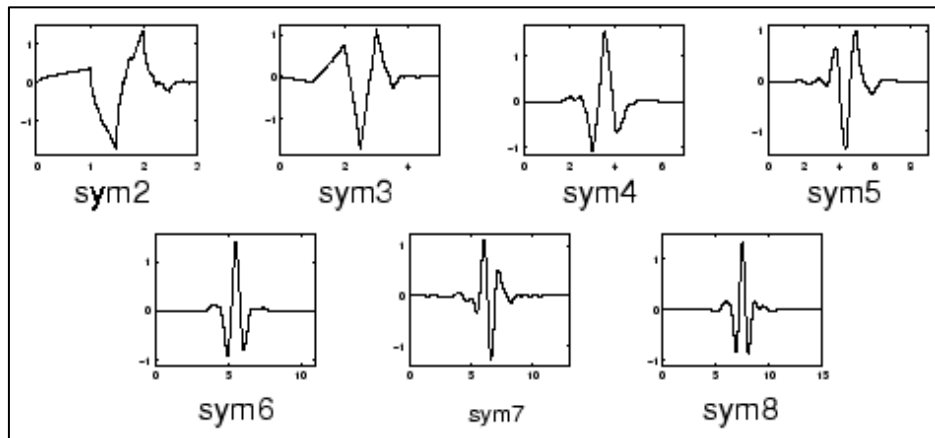


- DAUBECHIES. Ingrid Daubechies, una delle ricercatrici più brillanti nel mondo della ricerca sulle wavelet, ha inventato quelle che vengono chiamate wavelet ortonormali a supporto compatto, rendendo così praticabile l'analisi wavelet discreta.

Sono indicate come  $dbN$  dove  $N$  è l'ordine della wavelet. Sono generalmente asimmetriche e diventano sempre più regolari all'aumentare di  $N$ , ossia all'aumentare dei primi momenti nulli.



- SYMLETS. É una famiglia di wavelets che possono essere considerate delle varianti rispetto alle daubechies con maggiori proprietà di simmetria. Si indicano come  $symN$ .



# Capitolo 4

## Problemi di approssimazione sparsa

Il problema della ricerca della rappresentazione sparsa<sup>12</sup> e ridondante di un segnale richiede la risoluzione di un sistema di equazioni lineari sottodeterminato<sup>13</sup>.

Consideriamo un sistema sottodeterminato di equazioni lineari  $b = Ax$  dove  $A$  è una matrice  $m \times p$  ( $m < p$ ) e,  $b \in \mathbb{R}^m, x \in \mathbb{R}^p$ .

Il problema centrale della rappresentazione sparsa è definito come la ricerca della rappresentazione più sparsa possibile  $x$  che soddisfi  $b = Ax$ . A causa della natura di  $A$ , questo sistema ammette infinite soluzioni, e tra di esse, andranno cercate quelle con il minor numero di coefficienti non nulli.

Un metodo usuale per risolvere problemi di questo tipo è introdurre una funzione  $J(\cdot)$  per valutare la desiderabilità della soluzione  $x$ , dove i valori più piccoli sono preferiti. Si definisca il problema di ottimizzazione:

$$\min_{x \in \mathbb{R}^p} J(x) \text{ subject to } b = Ax$$

la scelta della funzione  $J(x)$  garantisce l'unicità della soluzione. Sfruttando l'ipotesi del segnale sparso, si può porre come funzione obiettivo del problema

$$\min_{x \in \mathbb{R}^p} \|x\|_0 \text{ subject to } b = Ax \quad (4.1)$$

dove  $\|x\|_0 = \#\{i: x_i \neq 0, i = 1, \dots, p\}$  è la pseudo-norma  $l_0$ , che conta il numero di componenti diversi da zero di  $x$ . Grazie all'ipotesi di sparsità questo problema può essere affrontato come un problema di ottimizzazione non lineare e NP-hard.

La sparsità di  $x$  implica che solo poche componenti ( $k \ll m < p$ ) sono diverse da zero. La motivazione alla base di una scomposizione sparsa è il desiderio di fornire una rappresentazione più semplice possibile di  $b$  come combinazione lineare con il minor numero possibile di colonne di  $A$ .

Sebbene il problema sopra esposto sia effettivamente NP-hard, la sua soluzione può essere spesso trovata utilizzando algoritmi di approssimazione.

Il *matching pursuit* (MP) è un algoritmo avido (greedy), funziona trovando gradualmente le posizioni dei non zeri in  $x$  uno alla volta.

L'idea principale è quella di trovare in ogni passaggio la colonna (atomo) in  $A$  che meglio si correla con il residuo inizializzato in  $b$ , aggiornandolo per tener conto della nuova colonna e del suo coefficiente. La principale differenza tra *matching pursuit* e *orthogonal matching pursuit* è che in quest'ultimo, durante tutte le fasi dell'algoritmo

---

<sup>12</sup> Nell'analisi numerica una matrice sparsa è una matrice in cui la maggior parte degli elementi è zero

<sup>13</sup> Un sistema di equazioni lineari viene chiamato sottodeterminato se presenta meno equazioni che incognite (a differenza di un problema sovradeterminato che presenta più equazioni che incognite).

i coefficienti diversi da zero vengono aggiornati risolvendo un problema ai minimi quadrati e, di conseguenza, il residuo risulterà ortogonale alle colonne già scelte.

## 4.1 Dizionari Ridondanti e Sparsità

Rappresentare un segnale in una base particolare implica trovare l'insieme unico di coefficienti di espansione in quella base. La capacità di fornire una rappresentazione sparsa dipende da quanto bene le caratteristiche del segnale corrispondono alle caratteristiche dei vettori di base.

I segnali del mondo reale spesso contengono delle caratteristiche che impediscono la rappresentazione sparsa in una singola base. Per questi segnali, vi è la possibilità di scegliere dei vettori da un insieme non limitato a una singola base, poiché si vuole assicurare di poter rappresentare ogni vettore nello spazio. Questo può essere fatto utilizzando un *dizionario*. Possiamo definire il dizionario come una raccolta di elementi, o vettori, con norma unitaria, chiamati atomi. Se gli atomi del dizionario coprono l'intero spazio del segnale, il dizionario viene detto *completo*, se invece, formano un insieme lineare dipendente, il dizionario è *ridondante*.

Poiché i vettori nel dizionario non sono un insieme linearmente indipendente, la rappresentazione del segnale nel dizionario non è univoca. Tuttavia, creando un dizionario ridondante, è possibile espandere il segnale in un insieme di vettori che si adattano alle caratteristiche della frequenza o della scala temporale del segnale.

## 4.2 Algoritmi *greedy*

Gli algoritmi *greedy* sono approcci iterativi che cercano di individuare gli atomi del dizionario che meglio si avvicinano alle proprietà del segnale. Come suggerisce apertamente il nome, l'algoritmo è stato progettato per essere avaro, in ogni fase seleziona la migliore approssimazione, sperando di ottenere una soluzione finale soddisfacente. Il principio infatti, è quello di partire da una soluzione parziale del problema ed estenderla iterativamente col fine di raggiungere una soluzione ottima, od eventualmente, una condizione di stop. Durante i tentativi di estensione della soluzione parziale non vengono considerate tutte le estensioni possibili, ma solamente quelle definite locali, che sono in numero molto minore, rispetto a quelle possibili. Una volta decretate le soluzioni locali, l'algoritmo procede selezionando tra queste quella che sembra essere la migliore, cioè quella più conveniente per raggiungere la soluzione ottima. Gli algoritmi *greedy* hanno dei vantaggi, quali la loro semplicità di implementazione e la velocità di esecuzione, ma d'altro canto non garantiscono sempre il buon esito della procedura; ciò si verifica a causa del fatto che i dati non vengono considerati nella loro totalità ma bensì a livello locale in maniera iterativa. Uno dei più noti algoritmi *greedy* è sicuramente il *Matching Pursuit*, che ha come obiettivo quello di decomporre qualsiasi segnale attraverso un'espansione lineare di forme d'onda appartenenti ad un dizionario ridondante, questo, servendosi di funzione di scelta alla quale viene riservato il compito di selezionare gli atomi che meglio approssimano le caratteristiche del segnale.

È possibile creare una soluzione utilizzando un termine alla volta, che seleziona ad ogni passo la colonna  $A$  creando una correlazione sempre più forte con il segnale

residuo. Viene impostato il residuo iniziale uguale al segnale in ingresso, facendo una banale approssimazione si ha:

$$r_0 = s \quad \text{e} \quad a_0 = 0$$

Allo step  $k$ , MP sceglie un altro indice  $\lambda_k$  risolvendo il problema di approssimazione

$$\lambda_k \in \arg \max_{\omega \in \Omega} \langle r_{k-1}, \varphi_\omega \rangle$$

dove  $\varphi$  fa parte del dizionario, calcola una nuova approssimazione e un nuovo residuo

$$a_k = a_{k-1} + (r_{k-1}, \varphi_{\lambda_k}) \varphi_{\lambda_k} \quad \text{e} \quad r_k = r_{k-1} - (r_{k-1}, \varphi_{\lambda_k}) \varphi_{\lambda_k}$$

Il residuo può anche essere espresso come  $r_k = s - a_k$ .

Il costo del calcolo di una soluzione con  $k_0$  elementi non nulli è dell'ordine  $O(k_0 mn)$  operazioni, notevolmente minore, tuttavia questo approccio non garantisce che la soluzione più sparsa sia effettivamente trovata. Data la *NP-hardness* del problema generale, non si può garantire che questi algoritmi trovino una soluzione in tutti i casi; infatti se la procedura si interrompe dopo un numero finito di iterazioni, l'approssimazione risultante è subottimale. Esiste una soluzione "sufficientemente" sparsa attraverso l'*orthogonal matching pursuit (OMP)*: che ha come obiettivo quello di far fronte ai problemi di convergenza garantendo un'approssimazione ottimale. L'*OMP* viene inizializzato allo stesso modo di *MP*, e, ad ogni passaggio, una colonna viene selezionata tramite:

$$\lambda_k \in \arg \max_{\omega \in \Omega} \langle r_{k-1}, \varphi_\omega \rangle$$

ma l'approssimazione viene calcolata diversamente, preso  $\Lambda_k = \{\lambda_1, \dots, \lambda_k\}$  possiamo scrivere l'approssimazione come

$$a_k \stackrel{\text{def}}{=} \arg \min_a \|s - a\|_2$$

dove  $a \in \text{span} \{\varphi_\lambda: \lambda \in \Lambda_k\}$ . Si noti che l'*orthogonal matching pursuit* non seleziona mai la stessa colonna due volte perché il residuo è ortogonale, di conseguenza deve essere uguale a zero dopo  $d$  passi.

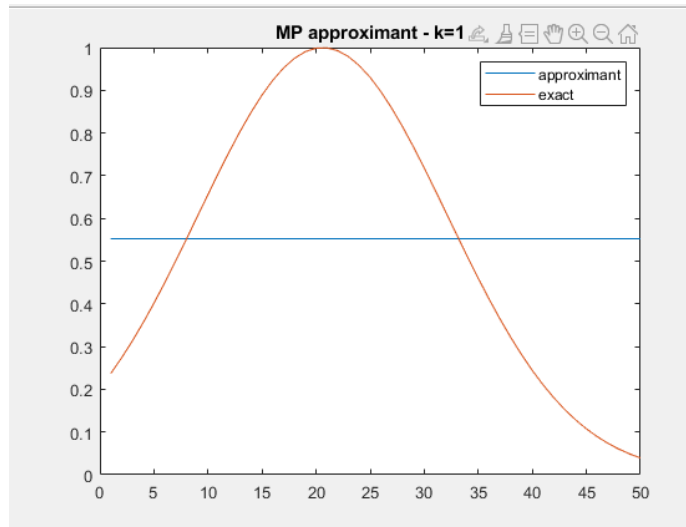
## 4.3 Prove numeriche

In questa parte forniamo un esempio esteso di come viene definito un problema di test, implementando in MATLAB gli algoritmi visti greedy precedentemente; in particolare con il *matching pursuit (MP)* e l'*orthogonal matching pursuit (OMP)*, si possono trovare i listati usati per le implementazioni nelle note di chiusura.

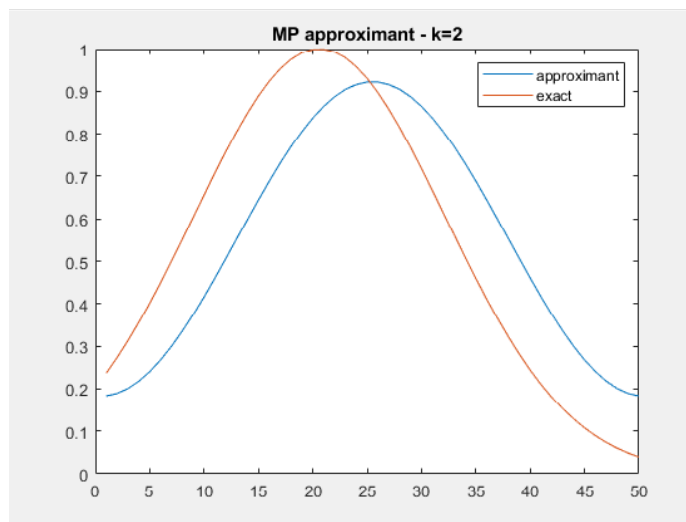
### 4.3.1.1 Segnale Gaussiano

Consideriamo un segnale di dimensione  $N=50$ , mediante l'algoritmo greedy *matching pursuit (MP)*.

Si nota che all'aumentare del numero di iterazioni  $K$  il segnale approssimante tende ad approssimare sempre meglio il segnale esatto. In particolare, si può notare che per  $K=1$  il segnale viene approssimato con una retta (Figura 13) mentre, per  $K \geq 2$  viene approssimato con una curva (Figura 14).

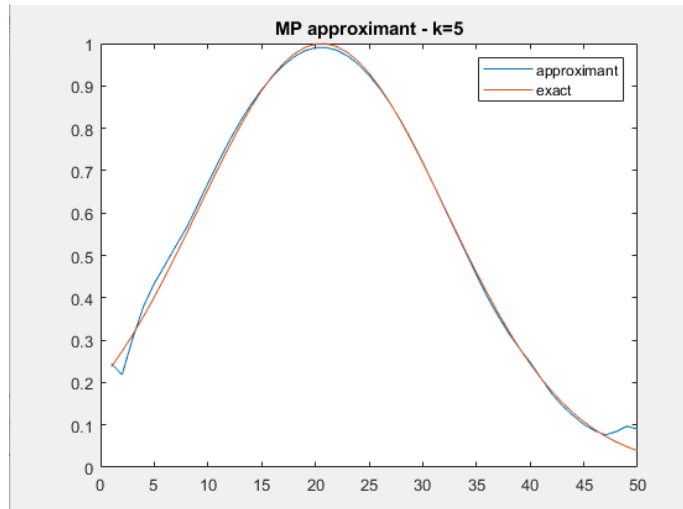


**Figura 13:** Approssimazione del segnale gaussiano tramite MP per  $K=1$

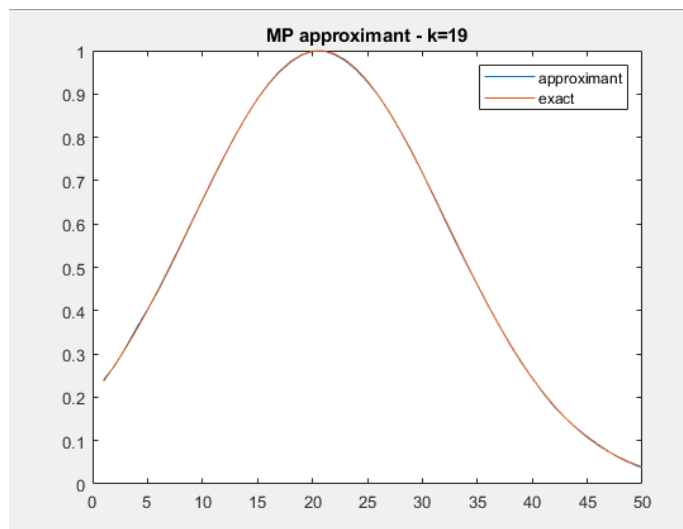


**Figura 14:** Approssimazione del segnale gaussiano tramite MP per  $K=2$

Aumentando il numero di iterazioni è facile vedere che già per un  $K=5$ , il *matching pursuit* (MP) approssima in modo ottimale il segnale gaussiano in esame (Figura 15). Per un  $K=19$ , (Figura 16) troviamo l'approssimazione *matching pursuit* (MP) che approssima meglio il segnale.



**Figura 15:** Approssimazione del segnale gaussiano tramite MP per  $K=5$



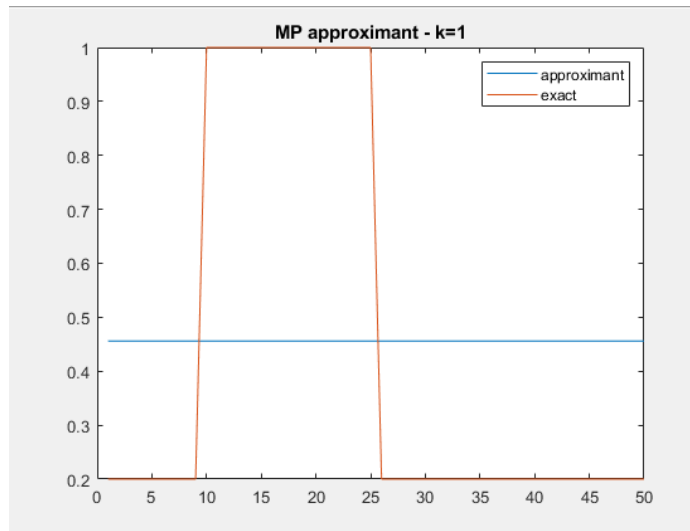
**Figura 16:** Approssimazione del segnale gaussiano tramite MP per  $K=19$

#### 4.3.1.2 Segnale a gradino

Il secondo segnale che prendiamo in considerazione mediante l'algoritmo di approssimazione greedy *matching pursuit* (MP), è il segnale a gradino, anch'esso di dimensione  $N=50$  con un  $K=1, 2, 5, 19$ .

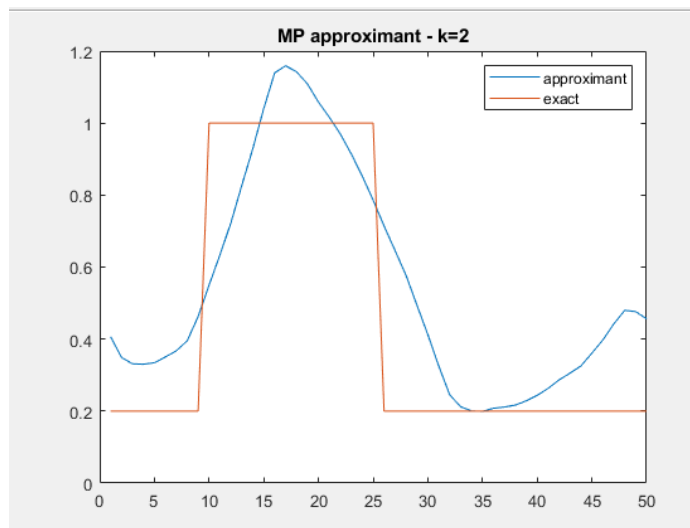
Il numero di iterazioni  $K$  è stato preso identico per tutti i segnali, per visualizzare nel dettaglio le varie differenze di approssimazione.

Notiamo, che per un  $K=1$  (Figura 17) abbiamo anche in questo caso un'approssimazione lineare come nel caso precedente (segnale gaussiano).



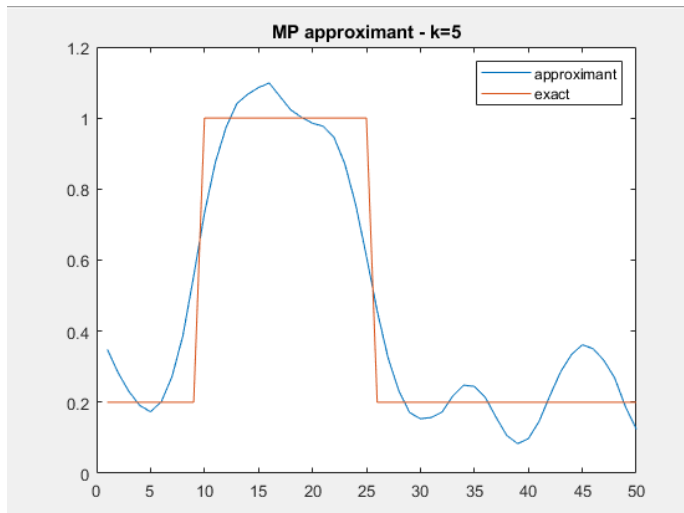
**Figura 17:** Approssimazione del segnale a gradino tramite MP per  $K=1$

Continuando con le varie iterazioni (Figura 18,19) salta subito all'occhio che l'algoritmo di approssimazione *matching pursuit* (MP) ha difficoltà nell'approssimare il segnale esatto.

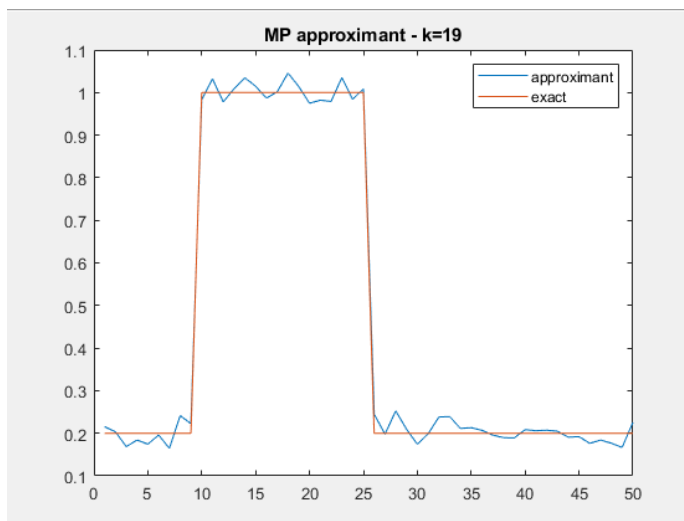


**Figura 18:** Approssimazione del segnale a gradino tramite MP per  $K=2$

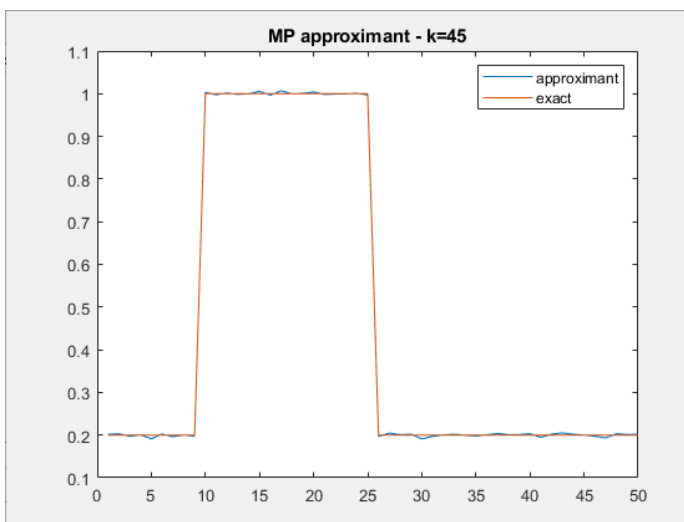
Infatti, solo per  $K=19$ , (Figura 20) vi è un'approssimazione grossolana; per avere una visibile buona approssimazione occorrono ben 45 iterazioni su un massimo di 100, visibile in (Figura 21).



**Figura 19:** *Approssimazione del segnale a gradino tramite MP per  $K=5$*



**Figura 20:** *Approssimazione del segnale a gradino tramite MP per  $K=19$*



**Figura 21:** *Approssimazione del segnale a gradino tramite MP per  $K=45$*

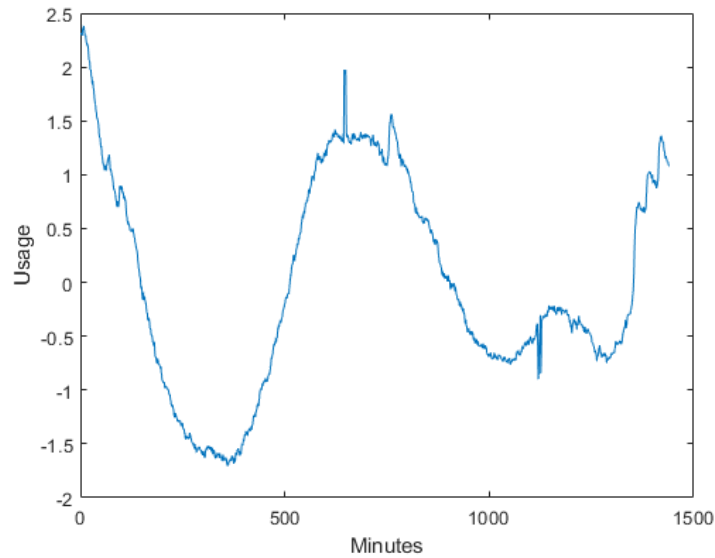


### 4.3.1.3 Analisi consumo di elettricità

Infine, implementiamo un'analisi del consumo di elettricità mediante l'algoritmo *greedy matching pursuit*.

I dati sono stati raccolti in un lasso di tempo di 24 ore, il set contiene 40 giorni di consumo elettrico, scegliamo per l'analisi il giorno 30, i dati sono centrati e scalati, quindi le unità di utilizzo effettive non sono rilevabili.

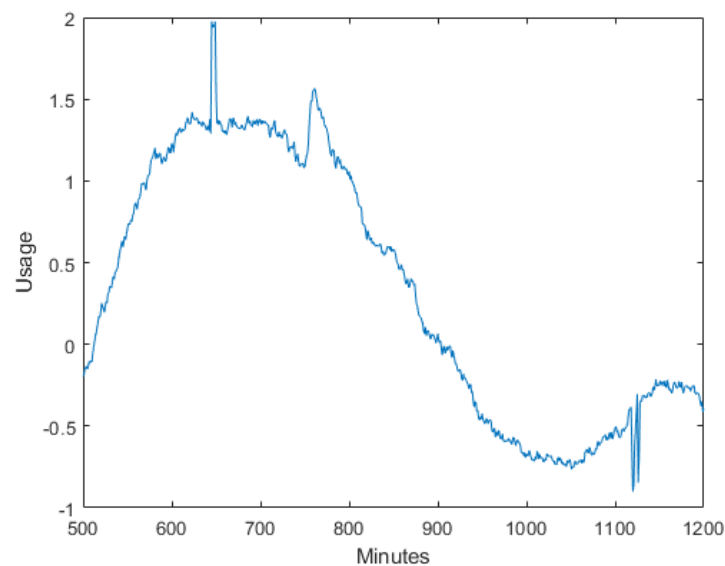
L'esempio dimostra che selezionando gli opportuni atomi da un dizionario, l'algoritmo *greedy* è spesso in grado di approssimare un vettore in modo più efficiente, rispetto a qualsiasi base singola.



**Figura 22:** Grafico analisi consumo elettrico

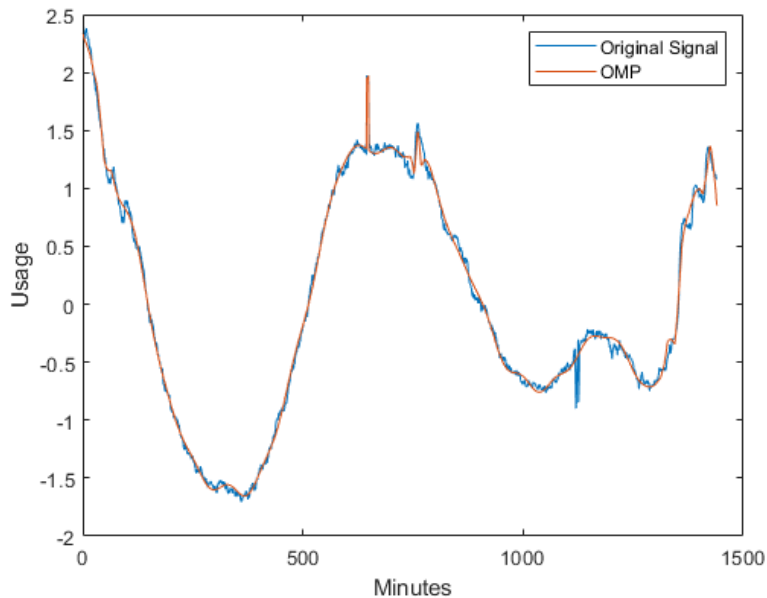
Come si può vedere (Figura 22) il segnale contiene delle oscillazioni regolari caratterizzate da bruschi aumenti e diminuzioni causate dall'utilizzo dell'energia.

Facendo uno zoom nel range di tempo compreso tra i 500 e 1200 minuti (Figura 23).



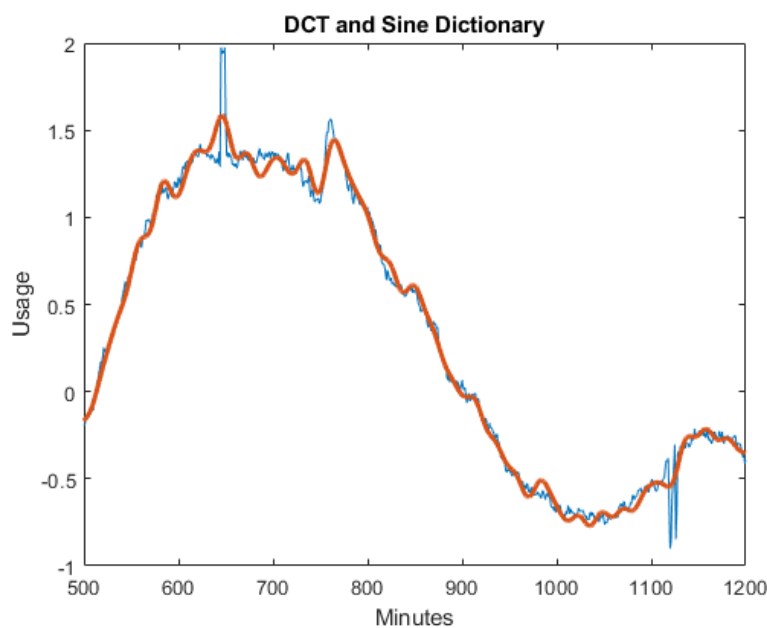
**Figura 23:** Zoom grafico dell'analisi del consumo elettrico nell'intervallo 500-1200 minuti: è facile individuare i bruschi cambiamenti del segnale che variano lentamente circa i 650, 760 e 1120 minuti.

A questo punto, costruiamo un'approssimazione del segnale utilizzando 40 vettori scelti da un dizionario *orthogonal matching pursuit (OMP)*. Questo dizionario è costituito dalla wavelet di Daubechies, dalla trasformata discreta del coseno di Fourier (DCT), da una base seno e dalla base delta di Kronecker.



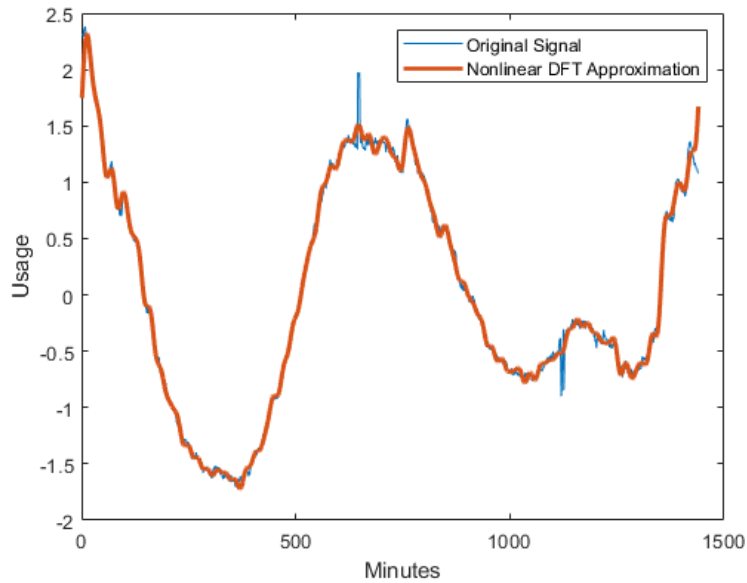
**Figura 24:** Approssimazione mediante OMP.

Possiamo vedere che con 40 coefficienti, l'*orthogonal matching pursuit (OMP)* approssima sia la parte che oscilla dolcemente, sia quella con bruschi cambiamenti.



**Figura 25:** Approssimazione OMP con il solo seno e con la sola trasformata discreta del coseno di Fourier (DCT).

Nella Figura 25, è subito visibile che l'approssimazione dei picchi verso l'alto e verso il basso non è accurato senza l'utilizzo delle wavelet. Simile al dizionario DCT-seno, l'approssimazione non lineare della *trasformata di Fourier discreta (DFT)*, si comporta apparentemente bene nel far corrispondere le oscillazioni regolari. Tuttavia, non approssima accuratamente i cambi improvvisi (Figura 26)



**Figura 26:** *Approssimazione non lineare mediante la trasformata discreta di Fourier.*

## LISTATI MATLAB

- Segnale gaussiano

```
1. N = 50; % signal dimension
2.
3. D = wmpdictionary(N); % default dictionary
4. %D = wmpdictionary(N,'lstcpt',{'sin','cos'}); %
5. %D = wmpdictionary(N,'lstcpt',{ 'haar',2}); % dictionary wavelet haar 2
6.
7. z = linspace(0,3,N);
8. s = exp(-(z-1.2).^2);
9. s = s'; % signal column
10.
11. itmax = 100; % maximum number of iterations
12. tol = 1e-2; % tolerance
13. show = 1;
14.
15. % funzioni implementate da noi
16. [amp, Rmp, imp] = MP(D,s,itmax,tol,show);
17. [aomp, Romp, iomp] = OMP(D,s,itmax,tol,show);
18.
19. nitMP = length(imp);
20. fprintf('MP - iterazioni = %d \n',nitMP)
21. nitOMP = length(iomp);
22. fprintf('OMP - iterazioni = %d \n',nitOMP)
23.
24. figure(1)
25. plot(z,[amp,s])
26. title('MP')
27.
28. figure(2)
29. plot(z,[aomp,s])
30. title('OMP')
31.
32. % funzione implementata da matlab
33. yfit1 = wmpalg('BMP',s,D);
34. yfit2= wmpalg('OMP',s,D);
```

- Segnale a gradino

```

1. N = 50; % signal dimension
2.
3. D = wmpdictionary(N); % default dictionary
4. %D = wmpdictionary(N,'lstcpt',{ 'haar',2 }); % dictionary wavelet haar 2
5.
6. z = linspace(0,3,N);
7. s = 0.2.*(z<=.5)+1.*(z>.5).*(z<1.5)+.2.*(z>=1.5);
8. s = s'; % signal (column)
9.
10. itmax = 100; % maximum number of iterations
11. tol = 1e-3; % tolerance
12. show = 1;
13.
14. % funzioni implementate da noi
15. [amp, Rmp, imp] = MP(D,s,itmax,tol,show);
16. [aomp, Romp, iomp] = OMP(D,s,itmax,tol,show);
17.
18. nitMP = length(imp);
19. fprintf('MP - iterazioni = %d \n',nitMP)
20. nitOMP = length(iomp);
21. fprintf('OMP - iterazioni = %d \n',nitOMP)
22.
23. figure(1)
24. plot(z,[amp,s])
25. title('MP')
26.
27. figure(2)
28. plot(z,[aomp,s])
29. title('OMP')
30.
31. % funzione implementata da matlab
32. yfit1 = wmpalg('BMP',s,D);
33. yfit2 = wmpalg('OMP',s,D);

```

- Matching Pursuit utilizzando i dizionari DCT, seno e Wavelet

```

1. load elec35_nor
2. x = signals(30,:);
3.
4. plot(x)
5. xlabel('Minutes')
6. ylabel('Usage')

```

```
7. xlim([500 1200])
8.
9. dictionary = {'db1',2},{'db1',3},'dct','sin','RnIdent',{'sym4',4}};
10. [mpdict,nbvect] = wmpdictionary(length(x),'lstept',dictionary);
11. [y,~,~,iopt] = wmpalg('OMP',x,mpdict);
12. plot(x)
13. hold on
14. plot(y)
15. hold off
16. xlabel('Minutes')
17. ylabel('Usage')
18. legend('Original Signal','OMP')
```

## BIBLIOGRAFIA

- [1] G. Rodriguez, Algoritmi numerici, Pitagora, 2008.
- [2] G. Rodriguez and S. Seatzu, Introduzione alla Matematica Applicata e Computazionale, 2010.
- [3] S. M. a. S. Zhong, Characterization of signals from multiscale edges, *IEEE Transactions on pattern analysis and machine intelligence*, 1992.
- [4] L. Verdoliva, la trasformata Wavelet, 2017/2018.
- [5] R. R. a. W. Kinsner, Experiments whit wavelet and oder Edge Detection Techniques, 1997.
- [6] J. Li, A Wavelet approach to edge detection, Diss. Sam Houston State University, 2003.
- [7] Z. Mallat, «Matching Pursuit with Time - Frequency Dictionaries,,» *In: IEEE Transactions on Signal Processing,,* n. 41:3397-3415., 1993,.
- [8] C. K. Chui, An Introduction to Wavelets, Elsevier, 2016.
- [9] J. A. Tropp, «Greed is Good: Algorithmic Results for,» *IEE transactions on Infotmation theory* 50.10 (2004): 2231-2242.
- [10] T. MathWork, 1994-2020. [Online].