



# ATTICUS

*Ambient-intelligent Tele-monitoring System*

Codice progetto: ARS01\_00860

Deliverable D4.4

## Tecniche di campionamento di segnali biologici



UNIVERSITÀ  
DEGLI STUDI  
DEL MOLISE



Consorzio Italia



## Data: Agosto 2019

Leader OR	UNISANNIO
Obiettivo realizzativo (OR)	OR4 – Realizzazione del dispositivo S-WARE: acquisizione ed elaborazione segnali
Attività	4.4 – Studio dei metodi di campionamento di segnali biologici
Descrizione dell'attività <i>(come da capitolato)</i>	Saranno analizzate tecniche di campionamento avanzate per segnali biologici, con particolare riferimento a quelli che hanno una larghezza di banda più elevata e che quindi richiederebbero frequenze di campionamento più elevate, come la frequenza di respirazione e l'ECG. L'utilizzo di tali tecniche consentirebbe di ridurre la frequenza di campionamento dei segnali, mantenendo inalterata la ricostruzione della forma d'onda, con una conseguente riduzione del consumo di potenza del dispositivo.
Partner coinvolti nell'attività	<ul style="list-style-type: none"><li>● REGMOL</li></ul>
Stato del documento	Final draft
Confidenzialità	Ristretta

## Indice del documento

1	Introduzione.....	6
2	Compressione dei segnali fisiologici .....	7
3	Metodi di compressione hardware.....	9
4	Metodi di compressione software .....	11
4.1	Metodi diretti .....	11
4.2	Metodi a estrazione di parametri .....	12
4.3	Metodi basati su trasformate .....	13
4.4	Metodi ibridi .....	15
5	Metodi basati su campionamento compresso.....	17
5.1	Matrici di sensing .....	19
5.1.1	Matrice di sensing non-uniforme .....	20
5.1.2	Matrice di sensing Gaussiana .....	20
5.1.3	Matrici di sensing Bernoulli e Rademacher .....	21
5.1.4	Matrice di sensing Deterministic Binary Block Diagonal (DBBD).....	21
5.2	Matrici dizionario .....	22
5.2.1	Discrete Cosine Transform .....	22
5.2.2	Continuous Wavelet Transform .....	23
5.2.3	Discrete Wavelet Transform.....	25
5.3	Algoritmi di ricostruzione.....	25
5.3.1	Basis Pursuit .....	26
5.3.2	Orthogonal Matching Pursuit.....	26
5.3.3	Minimizzazione della norma $\ell_{21}$ .....	27
5.3.4	MMV Orthogonal Matching Pursuit.....	28
5.3.5	ReMBO .....	29
6	Conclusioni.....	31
	Riferimenti bibliografici .....	32

## Indice delle tabelle

Tabella 1. Confronto tra le famiglie di dizionari wavelet.

24



## Indice delle figure

Figura 1. Architettura del convertitore A/D level-crossing proposto in [1].	9
Figura 2. Schema del campionatore (a) e del codificatore di entropia (b) del metodo di compressione [3].	12
Figura 3. Schema a blocchi del metodo [4].	13
Figura 4. Schema a blocchi del metodo [5].	14
Figura 5. Schema a blocchi del metodo [7].	16
Figura 6. Metodo CS implementato in hardware: a) schema a blocchi, b) tracciato dei segnali.	20
Figura 7. Schema a blocchi dell'algoritmo di Mallat, per l'implementazione della DWT.	25



## 1 Introduzione

Questo documento riporta lo stato dell'arte delle tecnologie per il campionamento di segnali fisiologici che consentono una riduzione del flusso di campioni acquisiti.

Sebbene queste tecniche possono essere applicate (con le opportune modifiche) per diversi segnali fisiologici, la ricerca si è principalmente rivolta verso l'Elettrocardiogramma (ECG), che presenta un maggiore flusso dati.



## 2 Compressione dei segnali fisiologici

I sistemi di misura indossabili si stanno diffondendo come dispositivi personali per misurare i parametri fisiologici, come la frequenza cardiaca o respiratoria.

Negli ultimi anni, questi sistemi stanno divenendo più complessi a causa dell'integrazione di molti sensori e front-end elettronici, che consentono di osservare e trasmettere segnali più veloci, come l'ECG e l'onda di respirazione. Inoltre, essi vengono integrati in sistemi Internet of Things (IoT), in cui un numero elevato di nodi deve essere connesso e gestito.

In particolare, in questo campo, sono stati proposti i sistemi Internet-of-Medical-Things (IoMT), per monitorare e gestire le applicazioni in ambito *healthcare*.

Una delle principali sfide nella realizzazione di nodi IoMT è rispettare i requisiti di consumi energetici e di dimensione dei dispositivi indossabili, garantendo contemporaneamente la memorizzazione e la trasmissione di un elevato flusso di dati e l'integrità dei segnali acquisiti, in conformità con gli standard clinici.

Questa sfida diventa ancora più impegnativa per il fatto che i nodi IoMT cominciano ad includere non soltanto trasmissione a livello personale (Personal Area Network - PAN), ma anche interfacce Wireless Wide Area Network interfaces, come LoRa, Sigfox or Narrowband-IoT (NB-IoT), che sono caratterizzate da consumi di potenza più alti e minori *data rate*.

La compressione dei segnali fisiologici è stata proposta come un mezzo per contemporaneamente: (i) ridurre il *data rate* che i nodi IoMT devono trasmettere e (ii) ridurre il consumo di energia del nodo, per il fatto che nella maggior parte dei casi la trasmissione dei dati è l'attività più dispendiosa in termini energetici.

Come già accennato prima, tra i diversi segnali fisiologici, la ricerca si è concentrata particolarmente nella compressione dell'ECG.

Una prima classificazione dei contributi scientifici in questo ambito può essere fatta distinguendo metodi di compressione hardware e metodi di compressione software.

I metodi appartenenti a ciascuna di queste classi saranno descritti rispettivamente nei paragrafi 3 e 4.



## 3 Metodi di compressione hardware

I metodi in questa categoria sfruttano la ridondanza dell'andamento temporale del segnale ECG per realizzare architetture di conversione analogico-numerica ottimizzate.

Un metodo che ricade in questa categoria è proposto in [1], dove gli autori hanno realizzato un Convertitore A/D level-crossing, per acquisire in modo efficiente segnali ECG.

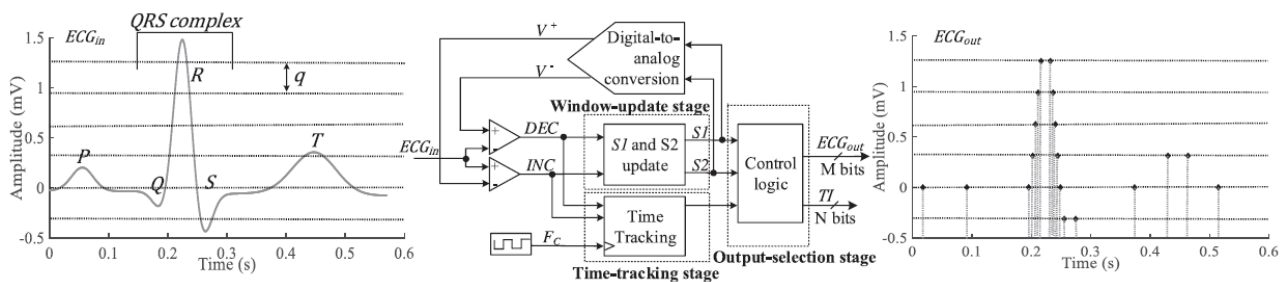


Figura 1. Architettura del convertitore A/D level-crossing proposto in [1].

Il metodo è basato sull'osservazione che, in accordo con il teorema di Shannon, per ricostruire adeguatamente un segnale a banda limitata, la frequenza di campionamento uniforme deve essere maggiore del doppio della massima frequenza del segnale. Quindi nel caso di campionamento uniforme alla frequenza di Nyquist, in corrispondenza di lente variazioni di ampiezza del segnale viene acquisito un numero ridondante di campioni. Al contrario, un convertitore A/D *level-crossing* è inattivo in intervalli in cui ci sono variazioni lente, e questo evita l'acquisizione di dati non necessari.

Lo schema dell'architettura proposta in [1] è riportato in Figura 1, insieme con un esempio di campionamento. È possibile osservare, infatti, nel grafico a destra, come i campioni vengano prelevati in modo più fitto lì dove le variazioni di ampiezza del segnale sono più significative, mentre nei punti in cui la pendenza è più lenta vengono acquisiti meno campioni.

Il vantaggio delle soluzioni di compressione hardware è che forniscono la compressione del segnale senza carico computazionale aggiuntivo per il microcontrollore del nodo di acquisizione. Di contro, essi fanno riferimento a specifici circuiti hardware che devono essere integrati all'interno del nodo

di acquisizione e che possono compromettere i requisiti di consumi di energia. Per quest'ultima ragione, spesso una soluzione completamente digitale viene preferita.



## 4 Metodi di compressione software

I numerosi contributi in questa categoria possono essere ulteriormente classificati in quattro tipi: metodi diretti, metodo ad estrazione di parametri, metodi basati su trasformata e metodi basati su campionamento compresso. Ciascun tipo sarà descritto nelle prossime sottosezioni.

### 4.1 Metodi diretti

I metodi diretti eseguono la compressione rimuovendo la ridondanza direttamente nel dominio del tempo, sfruttando, come nel caso dei metodi hardware, la diversa larghezza di banda delle diverse porzioni del segnale.

Un metodo appartenente a questa categoria, descritto in [2], propone un protocollo per la compressione di dati ECG, in cui il segnale è codificato mediante caratteri ASCII. In particolare, vengono prelevate 8 coppie di dati (tempo, ampiezza). Viene quindi estratto il segno dagli 8 valori di ampiezza, che viene codificato in un byte. Quindi i valori di ampiezza vengono moltiplicati per un opportuno fattore di scala, in modo che il massimo valore non ecceda 9, e troncati eliminando la parte decimale. Quindi, gli otto valori vengono raggruppati in a due a due in quattro interi moltiplicando il primo dei due per 10 e poi sommando. La versione compressa risulta quindi data da un byte contenente i bit segno, quattro byte, che contengono i campioni raggruppati e un byte contenente il fattore di scala.

In [3], gli autori hanno proposto uno schema di compressione dinamico per applicazioni di ECG wireless in tempo reale a basso consumo di potenza. Esso consiste in un campionatore digitale *integrate-and-fire* che consente di rappresentare il segnale ECG come un treno di impulsi. Un codificatore di entropia senza perdita si occupa quindi di codificare i timestamp e i segni del treno di impulsi in uno stream binario.

Lo schema del campionatore è mostrato in Figura 2a. Esso è costituito da un neurone semplificato e consiste di un integratore e una coppia di comparatori, che realizzano una codifica Sigma-Delta asincrona. Il treno di impulsi in uscita dal campionatore avrà quindi una frequenza di impulsi tanto maggiore, quanto maggiore sarà la pendenza del segnale (in valore assoluto). Il valore degli impulsi

(1 o -1) terrà conto del valore crescente o decrescente della pendenza. Il codificatore di entropia (Figura 2b) mantiene inalterati i valori degli impulsi, mentre codifica attraverso una codifica di Huffman gli intervalli tra i timestamp.

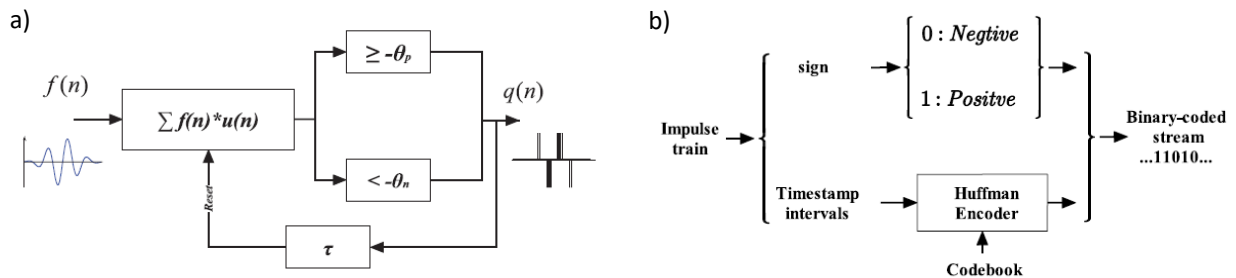


Figura 2. Schema del campionatore (a) e del codificatore di entropia (b) del metodo di compressione [3].

## 4.2 Metodi a estrazione di parametri

I metodi a estrazione di parametri sono basati sull'estrazione di alcune caratteristiche del segnale ECG come l'onda P, l'onda T e il complesso QRS. Queste caratteristiche vengono poi codificate ed inviate al posto dell'intero segnale.

Un metodo che rientra in questa categoria è presentato in [4]. Esso è composto da una fase di preelaborazione, in cui il segnale è diviso in battiti e ciascun battito viene segmentato per trovare le sezioni P, T e QRS. Ciascuna sezione, viene quindi filtrata con un filtro differente e viene rimossa la *baseline*. In una successiva fase di codifica, vengono eseguite due operazioni in parallelo: (i) il segnale a valle della pre-elaborazione viene confrontato con gli elementi di un codebook e viene restituito l'elemento che presenta la migliore corrispondenza con il segnale; (ii) vengono stimati i coefficienti di una *Long-Time Prediction*. Questi coefficienti vengono utilizzati per filtrare il segnale ottenuto dalla corrispondenza con il codebook. Successivamente una fase di *Short-Time Prediction* riceve il residuo dato dalla differenza tra il segnale stimato dalla fase di codifica e quello a valle della preelaborazione e produce il segnale compresso. Uno schema di questo metodo è riportato in Figura 3.

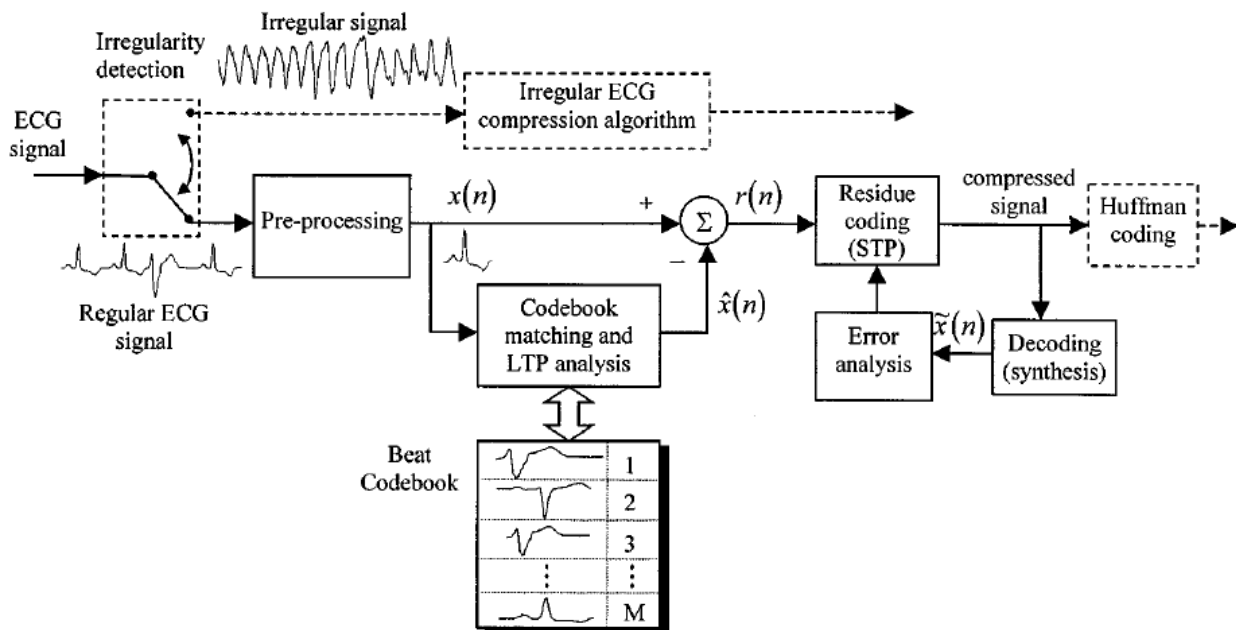


Figura 3. Schema a blocchi del metodo [4].

### 4.3 Metodi basati su trasformate

Nei metodi basati su trasformate, il segnale ECG viene proiettato in un altro dominio mediante una trasformazione ortogonale. Quindi, i coefficienti del segnale nel dominio trasformato vengono opportunamente codificati. Alcune trasformate largamente utilizzate sono la Trasformata Discreta di Fourier (Discrete Fourier transform – DFT), la Trasformata Discreta Coseno (Discrete Cosine Transform - DCT), la Trasformata di Walsh e la Trasformata Discreta Wavelet (Discrete Wavelet Transform - DWT).

Tra le diverse trasformate che sono state proposte, la trasformata di Fourier e la trasformata wavelet sono largamente utilizzate. La trasformata di Fourier, però, non è capace di fornire informazioni sull'occorrenza temporale delle diverse componenti armoniche del segnale. Per questo motivo, non è adatta allo studio di segnali non-stazionari. Le limitazioni della trasformata di Fourier possono essere superate attraverso l'utilizzo di una finestra mobile, dando quindi luogo alla Short-Time Fourier Transform (STFT). In questo caso, però, la scelta della lunghezza della finestra rimane un problema. Nella trasformata wavelet la lunghezza della finestra è variabile in funzione della

frequenza, con finestre più lunghe utilizzate per frequenze più basse e viceversa. Nella trasformata wavelet, un ruolo fondamentale è giocato dalla scelta della frequenza di campionamento e della mother wavelet. Una cattiva scelta di questi parametri produrrebbe informazione fuorviante. Per superare questo problema, in [5] viene utilizzata la trasformata di Stockwell, che combina la risoluzione dipendente dalla frequenza (tipica della trasformata wavelet) con una informazione locale sulla fase assolutamente riferita.

La trasformata di Stockwell è definita come il prodotto della Trasformata di Fourier del segnale di ingresso  $x(t)$  per una finestra Gaussiana:

$$S(\tau, f) = \frac{|f|}{\sqrt{2\pi}} \int_{-\infty}^{\infty} x(t) e^{-(\tau-t)^2 f^2 / 2} e^{-i2\pi f t} dt \quad (1)$$

In [5], la compressione del segnale ECG viene eseguita attraverso i passi mostrati in Figura 4. Inizialmente un blocco di preprocessing rimuove offset e guadagno dal segnale e applica un filtro di Savitsky-Golay per rimuovere il rumore ad alta frequenza dovuto ad artefatti, come contrazioni muscolari, interferenza dell'alimentazione elettrica e movimento degli elettrodi. Il successivo blocco si occupa di eseguire la trasformata di Stockwell. Quindi, i coefficienti della trasformata calcolati vengono quantizzati, convertiti in interi e codificati.

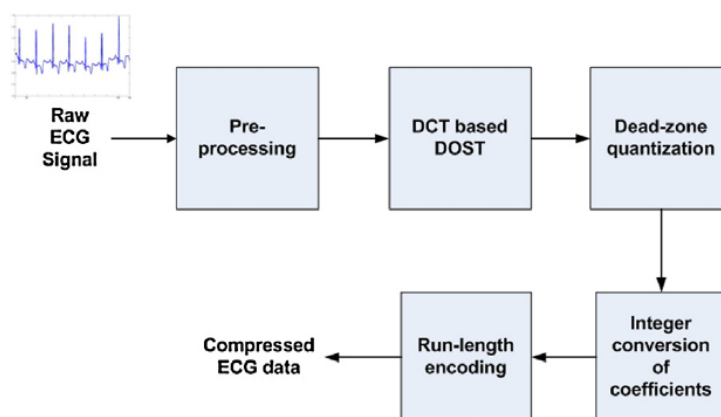


Figura 4. Schema a blocchi del metodo [5].

I metodi basati su trasformata hanno guadagnato una notevole attenzione nella letteratura grazie alla loro capacità di rappresentare accuratamente il segnale ECG anche a rapporti di compressione sostenuti. La maggior parte di essi, però, non è applicabile ad implementazioni in tempo reale su nodi di acquisizione con risorse limitate, a causa della complessità del calcolo delle trasformate e ai requisiti elevati in termini di memoria di acquisizione.

## 4.4 Metodi ibridi

I metodi ibridi combinano metodi di diverse categorie. Metodi appartenenti a questa categoria sono stati proposti in [6] e [7]. Il metodo in [6] in una prima fase rileva il battito e divide ciascun battito in blocchi “semplici” e “complessi”, sulla base della loro deviazione standard. Quindi sono utilizzate compressioni specifiche per ciascuna delle due categorie.

Uno dei principali problemi degli algoritmi di compressione del segnale ECG è che molto spesso i battiti anomali, a causa della loro morfologia complessa, non vengono correttamente ricostruiti, specialmente a valori alti del rapporto di compressione. Il metodo [7] affronta questo problema attraverso un metodo di classificazione binario basato su Support Vector Machine, che consente di distinguere tra battiti normali e anormali. Quindi, vengono utilizzati due diversi cammini di compressione per le due categorie (si veda Figura 5). Per i battiti normali, che hanno un andamento più regolare, si può utilizzare un metodo di compressione più spinto, che produca un maggiore rapporto di compressione, mentre per i battiti anormali, in cui è maggiore la necessità di preservarne l’andamento, si utilizza un metodo più semplice, che dà un rapporto di compressione minore, ma che garantisce una corretta ricostruzione dell’andamento del segnale, anche quando questo si discosta dall’andamento medio degli altri battiti.

In particolare, il segnale dei battiti riconosciuti come normali viene scomposto in un contributo medio, che viene compresso attraverso la DWT e un residuo, che viene scomposto mediante PCA (Principal Component Analysis) in autovettori e componenti principali. Il contributo medio, gli autovettori e le componenti principali vengono quindi quantizzati con dei passi di quantizzazione ottenuti mediante l’applicazione di un modello feed-forward multilayer perceptron neural network (FFMLPNN).

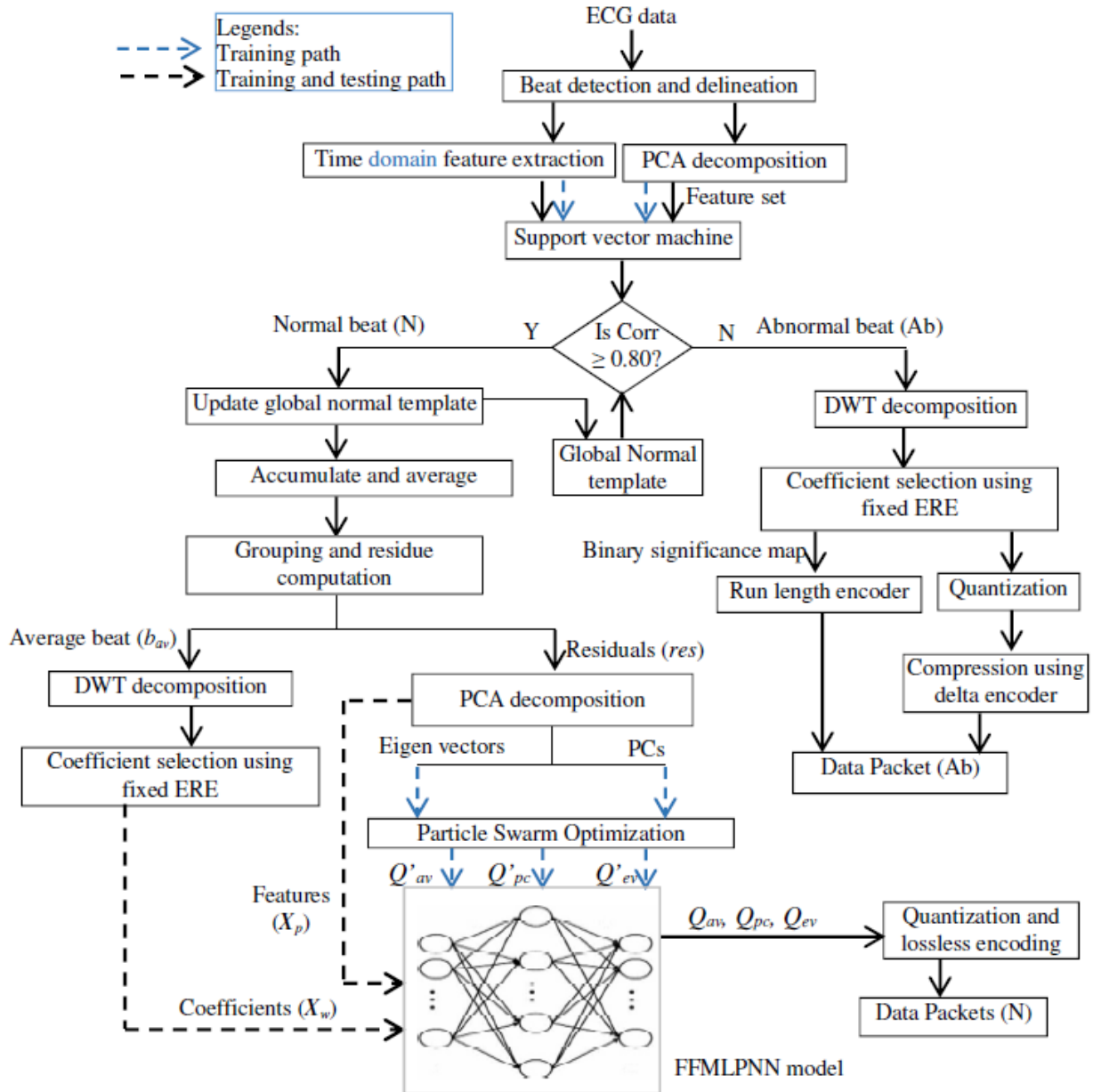


Figura 5. Schema a blocchi del metodo [7].

## 5 Metodi basati su campionamento compresso

Recentemente, sono stati proposti metodi di compressione del segnale ECG basati sul Compressive Sampling (CS). Il vantaggio di questi metodi sta nella loro capacità di ottenere prestazioni (in termini di rapporto di compressione e di accuratezza della ricostruzione) confrontabili con i metodi basati su trasformate, ma spostando l'onere computazionale dal nodo di acquisizione verso il nodo ricevitore. Poiché il nodo ricevitore è normalmente posizionato sul cloud, ha a disposizione risorse computazionali molto più elevate. Grazie a questo, i metodi basati su CS consentono di realizzare la compressione di segnali ECG anche su nodi con risorse limitate, e quindi con requisiti in termini di dimensione e consumi energetici compatibili con quelli dei dispositivi indossabili.

Per questo motivo, metodi CS rappresentano soluzioni largamente proposte la compressione (non soltanto dei segnali ECG) nei sistemi IoT [8], in cui l'acquisizione è eseguita da nodi semplici che sono connessi ad un layer di integrazione realizzato in cloud.

Il CS è basato sull'assunzione che il segnale ECG è sparso in un dominio rappresentato da una trasformazione matriciale, cioè che la sua rappresentazione in tale dominio è un vettore con pochi elementi non nulli. In particolare, se si considera un vettore  $\mathbf{x}$  di  $N$  campioni, acquisiti in una certa finestra temporale alla frequenza di Nyquist, questo può essere rappresentato in funzione dei coefficienti  $\boldsymbol{\theta}$  nel dominio trasformato, come:

$$\mathbf{x} = \boldsymbol{\Psi}\boldsymbol{\theta}, \quad (1)$$

dove  $\boldsymbol{\Psi}$  è una matrice  $N \times P$ , con  $P \approx N$  che descrive la trasformazione di dominio, spesso chiamata anche matrice dizionario (*dictionary matrix*) e  $\boldsymbol{\theta}$  è il vettore dei coefficienti di  $\mathbf{x}$  nel dominio della trasformazione.

Si assume che il vettore  $\boldsymbol{\theta}$  sia  $K$ -sparso, ovvero che esso contenga al massimo  $K$  elementi non nulli.

La compressione viene eseguita effettivamente moltiplicando i campioni del segnale ECG acquisito per una matrice  $\boldsymbol{\Phi}$  di dimensioni  $M \times N$ , con  $M < N$ , chiamata matrice di sensing (*sensing matrix*):

$$\mathbf{y} = \boldsymbol{\Phi}\mathbf{x}, \quad (2)$$

dove  $\mathbf{y}$  è il vettore di  $M$  campioni dei campioni compressi.

Per ricostruire la forma d'onda del segnale ECG sarebbe necessario invertire il sistema di equazioni in forma matriciale ottenuto combinando la (1) e la (2):

$$\mathbf{y} = \Phi\Psi\boldsymbol{\theta}. \quad (3)$$

Tale sistema è però sottodeterminato nella maggior parte dei casi, ovvero quando  $M < P$ .

Per ottenere la ricostruzione, è necessario quindi ricorrere alla proprietà di sparsità del vettore  $\boldsymbol{\theta}$ , andando a selezionare la più sparsa tra le soluzioni del sistema (3). Questo corrisponde a risolvere il seguente problema di minimizzazione:

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} \|\boldsymbol{\theta}\|_0, \quad \text{sottoposto a: } \mathbf{y} = \Phi\Psi\boldsymbol{\theta}, \quad (4)$$

dove  $\|\boldsymbol{\theta}\|_0$  rappresenta la norma  $\ell_0$  di  $\boldsymbol{\theta}$ , ovvero  $\|\boldsymbol{\theta}\|_0 = \sum_{i=1}^P \theta_i^0$  che corrisponde al conteggio degli elementi non nulli del vettore  $\boldsymbol{\theta}$ .

Questo problema è però molto complesso da risolvere perché richiede l'enumerazione completa delle possibili soluzioni.

E' stato dimostrato che il problema (4) può essere rilassato all'analogo problema di minimizzazione della norma  $\ell_1$ :

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} \|\boldsymbol{\theta}\|_1, \quad \text{sottoposto a: } \mathbf{y} = \Phi\Psi\boldsymbol{\theta}, \quad (5)$$

in cui  $\|\boldsymbol{\theta}\|_1 = \sum_{i=1}^P |\theta_i|$ .

Questo problema è invece risolvibile attraverso algoritmi di programmazione lineare o algoritmi *greedy*.

Per garantire la ricostruzione, è necessario che la matrice di sensing presenti una bassa coerenza con la matrice dizionario, dove la coerenza è definita come:

$$\mu(\Phi, \Psi) = \sqrt{N} \cdot \max_{i,j} \frac{\phi_i^T \psi_j}{\|\phi_i\|_2 \|\psi_j\|_2}, \quad (6)$$

dove  $\phi_i^T$  e  $\psi_j$  sono rispettivamente la  $i$ -ma riga di  $\Phi$  e la  $j$ -ma colonna di  $\Psi$  e  $\|\cdot\|_2$  indica la norma  $\ell_2$ , ovvero  $\|\mathbf{x}\|_2 = \sqrt{\sum_i |x_i|^2}$ .

La letteratura scientifica riguardo l'applicazione di CS per la compressione dei segnali ECG è principalmente concentrata sulla valutazione della qualità e sul miglioramento del segnale ricostruito, attraverso la proposta di specifiche matrici di sensing, matrici dizionario o algoritmi per la risoluzione del problema (5) [9].

## 5.1 Matrici di sensing

La fase di compressione, che consiste nell'implementazione della (2) può essere realizzata sia in hardware che in software. Le implementazioni hardware hanno il vantaggio di non aggiungere alcun carico computazionale al microcontrollore del nodo sensore, che quindi può essere molto semplice. Esse, però richiedono un frontend progettato appositamente che è difficile da realizzare, tenendo in considerazione i requisiti di bassissimo rumore e di basso consumo di potenza dei frontend per biopotenziali. Le implementazioni software, invece, aggiungono un onere computazionale (seppure minore, in confronto agli altri metodi di compressione, come quelli basati su trasformate o ibridi) e sono basati su frontend di tipo general purpose. Tra le due, le implementazioni software sono sicuramente le più diffuse, grazie alla maggiore versatilità e all'applicazione con hardware anche commerciale.

In Figura 6 è riportato un esempio di un metodo CS implementato in hardware [10]. Esso realizza una matrice di sensing di tipo diagonale a blocchi, attraverso un circuito di accumulazione hardware (Figura 6a), realizzato con un sommatore e due sample & hold che funzionano in opposizione (quando uno in fase di sample, l'altro è in fase di hold e viceversa) (Figura 6b).

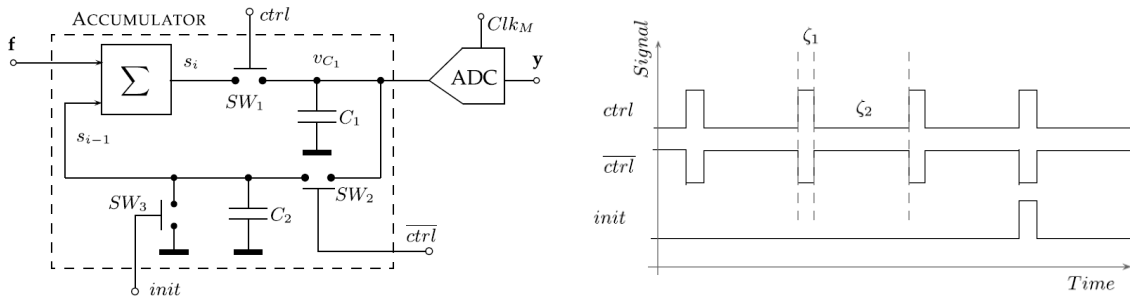


Figura 6. Metodo CS implementato in hardware: a) schema a blocchi, b) tracciato dei segnali.

Riguardo la tipologia della matrice di sensing che è utilizzata, si possono distinguere le seguenti categorie, che saranno analizzate in dettaglio nei prossimi sottoparagrafi:

- Matrice di sensing non-uniforme;
- Matrice di sensing Gaussiana;
- Matrice di sensing Bernoulli o Rademacher.

### 5.1.1 Matrice di sensing non-uniforme

Nel caso più semplice, è possibile utilizzare una matrice di sensing che implementa un campionamento non uniforme. È possibile ottenere tale matrice a partire dalla matrice identità.

Indicata con  $\mathbf{I}_N$  la matrice identità di dimensione  $N \times N$ , e con  $\Omega$  l'insieme di  $M$  indici scelti casualmente nell'intervallo  $[1, \dots, N]$ , la matrice  $\Phi$  si può ottenere selezionando le righe di  $\mathbf{I}_N$  i cui indici appartengono ad  $\Omega$ . Pertanto:

$$\Phi = [\mathbf{I}_N]_{\Omega}, \quad (7)$$

dove la notazione  $[\cdot]_{\Omega}$  esprime la selezione delle righe in accordo agli indici contenuti nell'insieme  $\Omega$ .

### 5.1.2 Matrice di sensing Gaussiana

È largamente diffuso l'utilizzo di matrici di sensing casuali. Queste si differenziano per la distribuzione con cui gli elementi della matrice sono ottenuti.

Una matrice di sensing Gaussiana è una matrice  $M \times N$  in cui ciascun elemento è estratto da una distribuzione Gaussiana a media nulla. Pertanto:

$$\Phi = \{\varphi_{ij}\}, \quad \text{with } \varphi_{ij} \sim N(0, \sigma^2), \quad (8)$$

dove  $\sigma^2$  è la varianza della distribuzione.

### 5.1.3 Matrici di sensing Bernoulli e Rademacher

Una matrice di sensing Bernoulliana è una matrice  $M \times N$  in cui ciascun elemento è estratto da una distribuzione di Bernoulli con parametro  $p = 0.5$ . Pertanto:

$$\Phi = \{\varphi_{ij}\}, \quad \text{with } \varphi_{ij} \sim B(0.5), \quad (9)$$

In accordo con la distribuzione di Bernoulli, ciascun elemento può assumere valori nell'insieme  $\{0, 1\}$ .

Uno svantaggio della matrice di Bernoulli è la presenza di numerosi 0 che riduce la quantità di informazioni acquisita.

Alternativamente, è spesso utilizzata la distribuzione di Rademacher, in cui invece ciascun elemento può assumere valori nell'insieme  $\{-1, 1\}$ :

$$\Phi = \{\varphi_{ij}\}, \quad \text{with } \varphi_{ij} \sim \text{Rademacher}. \quad (10)$$

### 5.1.4 Matrice di sensing Deterministic Binary Block Diagonal (DBBD)

Le matrici casuali hanno lo svantaggio di richiedere un discreto onere computazionale per la generazione, in tempo reale, dei campioni della matrice. In alcune matrici inoltre, gli elementi sono numeri reali. In quel caso la moltiplicazione matriciale richiede la presenza di risorse di elaborazione in virgola mobile, non sempre presenti nei processori o nei circuiti logici.

In alcune applicazioni è quindi preferibile una matrice deterministica, in cui gli elementi sono interi. In [11] è stata proposta una matrice deterministica chiamata Deterministic Binary Block Diagonal (DBBD). Questa matrice è definita utilizzando blocchi di  $m = N/M$  uni disposti sulla diagonale:

$$\Phi = \begin{bmatrix} \overbrace{1, \dots, 1}^m & \dots & 0 \\ 0 & \overbrace{1, \dots, 1}^m & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \overbrace{1, \dots, 1}^m \end{bmatrix} \quad (11)$$

In **Errore. L'origine riferimento non è stata trovata.**, la matrice è stata applicata alla compressione di segnali elettrocardiografici (ECG) ottenendo risultati migliori rispetto alle matrici casuali, con una base di tipo Discrete Cosine Transform (DCT).

## 5.2 Matrici dizionario

Le matrici dizionario sono ottenute principalmente a partire dalle seguenti trasformate, descritte nei seguenti sottoparagrafi:

- Discrete Cosine Transform;
- Continuous Wavelet Transform;
- Discrete Wavelet Transform.

### 5.2.1 Discrete Cosine Transform

La trasformata discreta coseno (Discrete Cosine Transform - DCT) è una trasformata spesso utilizzata per la compressione di segnali.

La DCT è definita come:

$$C(u) = \alpha(u) \sum_{n=0}^{N-1} x(n) \cos \left[ \frac{\pi(2n+1)u}{2N} \right], \quad (12)$$

per  $u = 0, \dots, N-1$ , dove  $\alpha(u) = \begin{cases} \sqrt{1/N}, & \text{se } u = 0 \\ \sqrt{2/N}, & \text{se } u \neq 0 \end{cases}$ .

La matrice dizionario DCT è ottenuta dalla trasformata del coseno inversa, come:

$$\Psi = \begin{bmatrix} \sqrt{\frac{1}{N}} x(0) & \sqrt{\frac{2}{N}} x(0) \cos \frac{\pi}{2N} & \dots & \sqrt{\frac{2}{N}} x(0) \cos \frac{\pi(N-1)}{2N} \\ \sqrt{\frac{1}{N}} x(1) & \sqrt{\frac{2}{N}} x(1) \cos \frac{3\pi}{2N} & \dots & \sqrt{\frac{2}{N}} x(1) \cos \frac{3\pi(N-1)}{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \sqrt{\frac{1}{N}} x(N-1) & \sqrt{\frac{2}{N}} x(N-1) \cos \left[ \frac{\pi(2N-1)}{2N} \right] & \dots & \sqrt{\frac{2}{N}} x(N-1) \cos \frac{\pi(2N-1)(N-1)}{2N} \end{bmatrix}. \quad (13)$$

La matrice DCT è di dimensione  $N \times N$ .

## 5.2.2 Continuous Wavelet Transform

La trasformata wavelet continua è anch'essa spesso utilizzata come matrice dizionario. La versione discretizzata della trasformata wavelet continua è definita come:

$$W(n, a, b) = \sum_{n=0}^{N-1} x(n) \psi\left(\frac{n-b}{a}\right) \quad (14)$$

dove  $b$  ed  $a$  sono rispettivamente i fattori di traslazione e di scala e  $\psi(n)$  è la wavelet madre (mother wavelet). Per definire una matrice dizionario basata sulla trasformata wavelet continua è necessario scegliere una mother wavelet. Inoltre, è necessario scegliere un campionamento del piano tempo-scala, che determina i valori delle coppie  $(b, a)$  che saranno utilizzate per determinare gli elementi della matrice dizionario. Il numero dei valori scelti determinerà il numero di colonne della matrice  $\Psi$ .

I tipi di wavelet più utilizzati sono ottenuti a partire dalle mother wavelet Gabor e Mexican Hat, le cui espressioni sono riportate di seguito:

- Gabor:

$$\psi_{Gabor}(n) = \frac{1}{\sqrt{2\pi}} e^{-n^2} \cos(\omega n + \theta), \quad (15)$$

dove  $\omega$  e  $\theta$  sono rispettivamente la frequenza e la fase della sinusoide;

- Mexican hat:

$$\psi_{Mexican\ hat}(n) = \frac{2}{\sqrt{3}\pi^{1/4}} (1 - n^2) e^{-\frac{n^2}{2}}. \quad (16)$$

In [9] sono state confrontate diverse matrici dizionario, basate sulle seguenti tipologie di wavelet: Gabor, Cubic Spline, Cubic B-Spline, Linear Spline, Linear B-Spline, e Mexican Hat. I risultati di quest'analisi sono riportati in Tab. 1, in termini di Percentage of Root-mean-square Difference, definito come:

$$PRD = \frac{\sum_{n=0}^{N-1} (\hat{x}(n) - x(n))^2}{\sum_{n=0}^{N-1} x(n)^2} \cdot 100, \quad (17)$$

per valori del rapporto di compressione (Compression Ratio - CR) di 2 e 4.

Dalla tabella si può osservare che i valori più bassi del PRD sono stati ottenuti dalla wavelet di tipo Mexican Hat.

Tabella 1. Confronto tra le famiglie di dizionari wavelet.

Famiglia di wavelet	Numero di colonne della matrice	CR	PRD [%]
Gabor	3072	2	5.18
		4	20.46
Cubic Spline	8192	2	15.335
		4	25.79
Cubic B-Spline	5888	2	14.4
		4	27.05
Linear Spline	7168	2	8.03
		4	19.66
Linear B-Spline	3584	2	19.24
		4	44.86
Mexican Hat	2048	2	4.39
		4	8.63

La matrice dizionario, si ottiene costruendo una matrice trasformata, in cui ogni colonna è ottenuta valutando la mother wavelet opportunamente traslata e scalata, a seconda di ciascuna delle coppie  $(b, a)$ :

$$\Psi = \begin{bmatrix} W(0, (b, a)_0) & W(0, (b, a)_1) & \dots & W(0, (b, a)_{P-1}) \\ W(1, (b, a)_0) & W(1, (b, a)_1) & \dots & W(1, (b, a)_{P-1}) \\ \vdots & \vdots & \ddots & \vdots \\ W(N-1, (b, a)_0) & W(N-1, (b, a)_1) & \dots & W(N-1, (b, a)_{P-1}) \end{bmatrix}, \quad (18)$$

dove  $P$  è il numero delle coppie  $(b, a)$  e di conseguenza il numero delle colonne della matrice.

E' da notarsi che la matrice ottenuta mediante trasformata wavelet continua ha dimensione  $N \times P$ , dove  $P$  può essere diverso da  $N$ . In genere  $N$  è maggiore di  $P$ , perché la trasformata wavelet continua non è ortogonale e quindi per poter rappresentare l'intero spazio delle funzioni, è necessario un numero di elementi maggiore rispetto alla cardinalità dello spazio.

## 5.2.3 Discrete Wavelet Transform

Un'altra categoria di matrici dizionario molto utilizzate è ottenuta mediante la trasformata wavelet discreta (Discrete Wavelet Transform – DWT).

In questo caso, si considera una decomposizione DWT diadica, ottenuta a partire da famiglie di wavelet ortogonali, come Daubechies e Symlet.

La DWT è normalmente descritta dai coefficienti dei filtri QMF (Quadrature Mirror Filters)  $h(n)$  e  $g(n)$ , che sono tabellati per ciascuna trasformata. I coefficienti della trasformata si ottengono quindi applicando l'algoritmo di Mallat, il cui schema è riportato in Figura 7.

La matrice  $\Psi$  è costituita dalla matrice IDWT (Inverse Discrete Wavelet transform). E' possibile ottenere tale matrice valutando i coefficienti DWT per ciascuna colonna della matrice identità e poi trasponendo la matrice risultante.

Poiché la DWT è ortogonale, sono sufficienti N colonne per rappresentare qualsiasi segnale di lunghezza N.

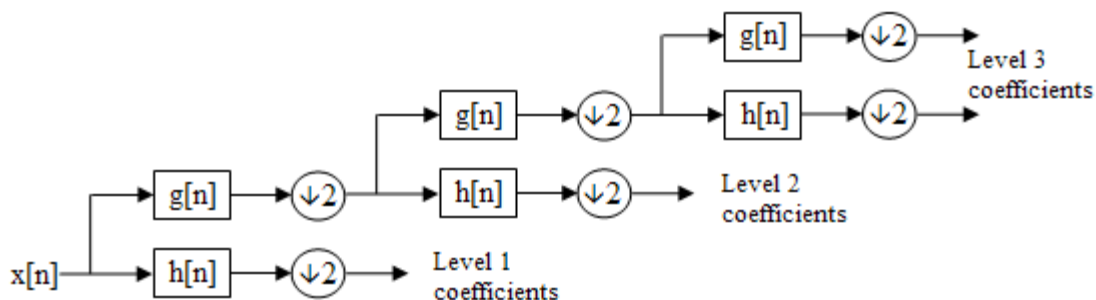


Figura 7. Schema a blocchi dell'algoritmo di Mallat, per l'implementazione della DWT.

## 5.3 Algoritmi di ricostruzione

Come accennato sopra, la ricostruzione della forma d'onda viene eseguita attraverso la risoluzione di un algoritmo di minimizzazione. Nei paragrafi seguenti saranno analizzati i principali algoritmi.

### 5.3.1 Basis Pursuit

L'algoritmo Basis Pursuit è una tecnica per decomporre un segnale in una sovrapposizione ottima degli elementi della base del dominio [12]. Basis Pursuit consente di selezionare, tra le infinite soluzioni del sistema  $\mathbf{A}\boldsymbol{\theta} = \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{\theta} = \mathbf{y}$ , quella i cui coefficienti hanno la minima norma  $\ell_1$ :

$$\hat{\boldsymbol{\theta}} = \operatorname{argmin}\|\boldsymbol{\theta}\|_1, \quad \text{subject to: } \mathbf{y} = \mathbf{A}\boldsymbol{\theta}, \quad (19)$$

Questa soluzione è calcolata con un algoritmo di programmazione lineare.

In molti casi (per esempio per denoising) è necessario trovare una soluzione con un errore limitato da un parametro  $\varepsilon$ . L'ottimizzazione in (19) viene rilassata come:

$$\hat{\boldsymbol{\theta}} = \operatorname{argmin}\|\boldsymbol{\theta}\|_1, \quad \text{subject to: } \|\mathbf{y} - \mathbf{A}\boldsymbol{\theta}\|_2 \leq \varepsilon. \quad (20)$$

Questo è un problema di minimizzazione convessa che è possibile calcolare minimizzando la corrispondente funzione Lagrangiana:

$$L_1(\lambda, \mathbf{y}, \boldsymbol{\Phi}\boldsymbol{\Psi}) = \|\mathbf{y} - \mathbf{A}\boldsymbol{\theta}\|_2^2 + \lambda\|\boldsymbol{\theta}\|_1. \quad (21)$$

È possibile implementare in MATLAB l'algoritmo Basis Pursuit ricorrendo ad una libreria per la minimizzazione convessa. In particolare, la libreria CVX [13] è stata già proposta per la risoluzione di Basis Pursuit.

### 5.3.2 Orthogonal Matching Pursuit

Orthogonal Matching Pursuit (OMP) [14] è un algoritmo iterative greedy (ovvero che cerca una soluzione ottima da un punto di vista globale attraverso la scelta della soluzione più vantaggiosa a ogni passo locale). OMP seleziona ad ogni passo la colonna della matrice  $\mathbf{A}$  che è maggiormente collegata con il residuo tra la soluzione corrente  $\mathbf{A}\boldsymbol{\theta}$  ed il vettore  $\mathbf{y}$ . Quindi, l'algoritmo aggiorna il residuo proiettando il vettore di misura  $\mathbf{y}$  sullo spazio lineare rappresentato dalle colonne della matrice  $\mathbf{A}$  che sono state selezionate fino a quell'iterazione e poi ripete lo stesso procedimento per il numero specificato di iterazioni [15].

OMP offre, rispetto ad i metodi alternativi, una maggiore semplicità ed una veloce implementazione. Nella sua implementazione più efficiente, basata sul lemma dell'inversione della

matrice, OMP necessita di un numero di operazioni  $\sim \mathcal{O}((M + N)K^2)$  [16], dove  $K$  è il numero di iterazioni, che coincide con la sparsità ovvero il numero di elementi non-nulli del vettore  $\boldsymbol{\theta}$ .

L'algoritmo OMP si può riassumere nei seguenti passi:

1. Si inizializza il residuo  $\mathbf{r}_0 = \mathbf{y}$ , l'insieme degli indici di colonna  $\Lambda_0 = \emptyset$ , la matrice degli elementi selezionati  $\mathbf{A}_0$  a una matrice vuota  $[\ ]$ , e il contatore delle iterazioni  $t = 1$ .
2. Si trova l'indice  $\lambda_t$  che risolve il problema di ottimizzazione:

$$\lambda_t = \arg \max_{j=1,\dots,N} |\mathbf{a}_j^T \mathbf{r}_{t-1}|, \quad (22)$$

dove  $\mathbf{a}_j$  è la  $j$ -ma colonna di  $\mathbf{A}$ .

3. Si aggiorna l'insieme degli indici e la matrice degli elementi selezionati:  $\Lambda_t = \Lambda_{t-1} \cup \{\lambda_t\}$   
 $\mathbf{A}_t = [\mathbf{A}_{t-1} \ \mathbf{a}_{\lambda_t}]$ .
4. Si risolve il seguente problema ai minimi quadrati per trovare la nuova stima:

$$\boldsymbol{\theta}_t = \arg \min_{\boldsymbol{\theta}} \|\mathbf{y} - \mathbf{A}_t \boldsymbol{\theta}\|_2. \quad (23)$$

5. Si calcola il nuovo residuo:

$$\mathbf{r}_t = \mathbf{y} - \mathbf{A}_t \boldsymbol{\theta}_t. \quad (24)$$

6. Si incrementa  $t$  e si ritorna al passo 2 se  $t < K$ .
7. La stima  $\hat{\boldsymbol{\theta}}$  ha elementi non nulli agli indici elencati nell'insieme  $\Lambda_K$ . Il valore della stima  $\hat{\boldsymbol{\theta}}$  all'elemento  $\lambda_j$  è uguale al  $j$ -mo elemento di  $\boldsymbol{\theta}_t$ .

Nella sua formulazione originale OMP necessita di conoscere la sparsità  $K$  del segnale. In questo caso l'algoritmo avrà un numero di iterazioni pari a  $K$ . Sono state proposte versioni modificate di OMP che propongono diverse regole di arresto, basate sulla norma  $\ell_2$  o  $\ell_\infty$  del residuo [15].

### 5.3.3 Minimizzazione della norma $\ell_{21}$

Gli algoritmi descritti nei paragrafi precedenti si riferiscono al problema (4), che viene definito Single Measurement Vector (SMV), ovvero le misure sono rappresentate da un singolo vettore.

In alcuni casi è necessario risolvere una versione del problema (4) in cui le misure sono ottenute da più sensori e quindi sono rappresentate da più vettori. Un problema di questo tipo viene chiamato

Multiple Measurement Vector (MMV). In questo caso, è possibile raggruppare in una matrice  $\mathbf{Y}$  i vettori di misura  $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_L$ , come:

$$\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_L]. \quad (25)$$

Il problema (4) diventa quindi:

$$\hat{\Theta} = \operatorname{argmin} \|\Theta\|_{2,0}, \quad \text{subject to: } \mathbf{Y} = \Phi\Psi\Theta, \quad (26)$$

dove  $\|\Theta\|_{2,0}$  misura il numero di righe della matrice  $\Theta$  che contengono elementi tutti nulli. In particolare, la norma  $\|\Theta\|_{2,0}$  ha la seguente espressione:

$$\|\Theta\|_{2,0} = \sum_{i=1}^N \left\| \sum_{j=1}^L \theta_{ij}^2 \right\|_0 \quad (27)$$

Così come il problema (4), anche (26) è un problema computazionalmente molto complesso perché richiede l'enumerazione delle possibili soluzioni. In modo analogo alla sostituzione della norma  $\ell_1$  alla norma  $\ell_0$  per il problema SMV, nel caso dell'MMV è stata proposta la sostituzione della norma  $\ell_{2,0}$  con la norma  $\ell_{2,1}$ :

$$\|\Theta\|_{2,1} = \sum_{i=1}^N \sqrt{\sum_{j=1}^L \theta_{ij}^2} \quad (28)$$

Il problema (26) diventa quindi:

$$\hat{\Theta} = \operatorname{argmin} \|\Theta\|_{2,1}, \quad \text{subject to: } \mathbf{Y} = \Phi\Psi\Theta, \quad (29)$$

È possibile risolvere il problema (29) analogamente a (19) attraverso un algoritmo di programmazione lineare o una sua versione con rumore attraverso una minimizzazione convessa.

### 5.3.4 MMV Orthogonal Matching Pursuit

Come nel caso del problema SMV, è possibile utilizzare un algoritmo greedy per la risoluzione. In particolare, è stata proposta una versione di OMP per il problema MMV [17][18].

Questo algoritmo è chiamato OMPMMV o MMV Orthogonal Matching Pursuit (M-OMP). I passi dell'algoritmo sono i seguenti:

1. Si Inizializza il residuo  $\mathbf{R}_0 = \mathbf{Y}$ , il supporto  $\Lambda_0 = \emptyset$  ed il contatore delle iterazioni  $t = 1$ .
2. Si trova l'indice  $\lambda_t$  che risolve il problema di ottimizzazione:

$$\lambda_t = \arg \max_{j=1,\dots,N} \frac{\|\mathbf{z}_j\|_2}{\|\mathbf{a}_j\|_2}, \quad (30)$$

dove  $\mathbf{z}_j = \mathbf{R}_{t-1}^T \mathbf{a}_j$ .

3. Si aggiorna il supporto e la matrice degli elementi selezionati:  $\Lambda_t = \Lambda_{t-1} \cup \{\lambda_t\}$   $\mathbf{A}_t = [\mathbf{A}_{t-1} \ \mathbf{a}_{\lambda_t}]$ .
4. Si risolve il seguente problema ai minimi quadrati per trovare la nuova stima:

$$\Theta_t = \arg \min_{\Theta} \|\mathbf{Y} - \mathbf{A}_t \Theta\|_2. \quad (31)$$

5. Si calcola il nuovo residuo:

$$\mathbf{R}_t = \mathbf{Y} - \mathbf{A}_t \Theta_t. \quad (32)$$

6. Si incrementa  $t$  e si ritorna al passo 2 se  $t < K$ .
7. La stima  $\hat{\Theta}$  ha righe non nulle agli indici elencati nell'insieme  $\Lambda_K$ . Il valore della stima  $\hat{\Theta}$  all'elemento  $\lambda_j$  è uguale al  $j$ -mo elemento di  $\Theta_t$ .

### 5.3.5 ReMBO

L'algoritmo Reduce MMV and Boost (ReMBO), proposto in [19], consiste nel ridurre il problema MMV in uno SMV. L'algoritmo consiste nei seguenti passi:

1. Si inizializza il contatore delle iterazioni  $t = 1$ ;
2. Si sceglie un vettore causale  $\mathbf{a}$  di lunghezza  $L$ , in accordo con una distribuzione di probabilità  $\mathcal{P}$ . È necessario che la distribuzione  $\mathcal{P}$  sia assolutamente continua ovvero può essere rappresentata come un integrale su una funzione di densità integrabile;
3. Si ottiene il vettore  $\mathbf{y} = \mathbf{Y}\mathbf{a}$ ;
4. Si risolve il problema SMV  $\mathbf{y} = \mathbf{A}\Theta$  con una delle tecniche descritte nei paragrafi precedenti (es. BP o OMP). Si supponga la soluzione sia  $\hat{\Theta}$ ;

5. Si aggiorna la stima del supporto  $\hat{S} = I\{\hat{\boldsymbol{\theta}}\} = \{j \mid \theta_j \neq 0\}$ ;
6. Si costruisce  $\hat{\boldsymbol{\theta}}$  utilizzando il supporto stimato  $\hat{S}$ , come:

$$\boldsymbol{\theta}_S = \mathbf{A}_S^+ \mathbf{Y} \quad (33)$$

e

$$\boldsymbol{\theta}_j = \mathbf{0}, \quad \forall j \notin S \quad (34)$$

7. Se la dimensione del supporto  $|\hat{S}| \leq K$  e/o non si è raggiunto un valore della norma del residuo inferiore ad una certa soglia  $\epsilon$  ( $\|\mathbf{y} - \mathbf{A}\hat{\boldsymbol{\theta}}\|_2 \leq \epsilon$ ), si incrementa il contatore delle iterazioni e ripete dal passo 2.

## 6 Conclusioni

È possibile ridurre il data rate e di conseguenza il consumo di energia del nodo sensore attraverso opportune tecniche di compressione del segnale ECG. Nei sistemi IoT, come quello che si intende realizzare per il progetto ATTICUS, i sistemi di compressione basati su Compressive Sampling offrono il vantaggio di una bassa complessità computazionale sul nodo di acquisizione, mentre il carico viene spostato alla parte ricevente, dove sono disponibili risorse computazionali più elevate. Pertanto, per il progetto ATTICUS sarà valutata la possibilità di implementare tali tecniche di compressione.



## Riferimenti bibliografici

- [1] M. Tlili, M. Ben-Romdhane, A. Maalej, F. Rivet, D. Dallet, C. Rebai, "Level-crossing ADC design and evaluation methodology for normal and pathological electrocardiogram signals measurement", *Measurement*, vol. 124, no. February, pp. 413–425, 2018.
- [2] S. K. Mukhopadhyay, S. Mitra, M. Mitra, "An ECG signal compression technique using ASCII character encoding", *Measurement*, vol.45, no.6, pp. 1651–1660, 2012.
- [3] K. Luo, J. Li, J. Wu, "A dynamic compression scheme for energy-efficient real-time wireless electrocardiogram biosensors", *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 9, pp. 2160–2169, 2014.
- [4] Y. Zigel, A. Cohen, A. Katz, "ECG signal compression using analysis by synthesis coding", *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 10, pp. 1308–1316, 2000.
- [5] C. K. Jha, M. H. Kolekar, "Electrocardiogram data compression using DCT based discrete orthogonal Stockwell transform", *Biomedical Signal Processing and Control*, vol. 46, pp. 174–181, 2018.
- [6] P. Bera and R. Gupta, "Hybrid encoding algorithm for real time compressed electrocardiogram acquisition," *Measurement*, vol. 91, pp. 651–660, 2016.
- [7] P. Bera, R. Gupta, and J. Saha, "Preserving Abnormal Beat Morphology in Long-term ECG Recording: An Efficient Hybrid Compression Approach," *IEEE Trans. Instrum. Meas.*, vol. 69, N. 5, June 2019, pp. 2084-2092.
- [8] E. Balestrieri, L. De Vito, F. Lamonaca, F. Picariello, S. Rapuano, and I. Tudosa, "Research challenges in measurement for Internet of Things systems," *ACTA IMEKO*, vol. 7, no. 4, pp. 82–94, December 2018.
- [9] D. Craven, B. McGinley, L. Kilmartin, M. Glavin, and E. Jones, "Compressed Sensing for Bioelectric Signals: A Review," *IEEE J. of Biomedical and Health Informatics*, vol. 19, no. 2, pp. 529–540, March 2015.
- [10] A. Ravelomanantsoa, and H. Rabah, "Simple and Efficient Compressed Sensing Encoder for Wireless Body Area Network," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 12, pp. 2973–2982, 2014.
- [11] A. Ravelomanantsoa, H. Rabah, A. Rouane, "Compressed Sensing: A Simple Deterministic Measurement Matrix and a Fast Recovery Algorithm," in *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 12, pp. 3405-3413, Dec. 2015.
- [12] S. Chen, D. Donoho "Basis Pursuit",  
[http://redwood.psych.cornell.edu/discussion/papers/chen\\_donoho\\_BP\\_intro.pdf](http://redwood.psych.cornell.edu/discussion/papers/chen_donoho_BP_intro.pdf)
- [13] "CVX: Matlab Software for Disciplined Convex Programming", <http://cvxr.com/cvx/>.

- [14] J. A. Tropp, A. C. Gilbert, "Signal Recovery from Random Measurements Via Orthogonal Matching Pursuit", IEEE Transactions on Information Theory, vol. 53, no. 2, pp. 4655-4666, Dec. 2007.
- [15] T. T. Cai, L. Wang, "Orthogonal Matching Pursuit for Sparse Signal Recovery with Noise", IEEE Transactions on Information Theory, vol. 57, no. 7, pp. 4680-4688, July 2011.
- [16] B. L. Sturm, M. G. Christensen, "Comparison of orthogonal matching pursuit implementations," in Proc. of 20th European Signal Processing Conference, Aug. 2012, pp. 220-224.
- [17] S. F. Cotter, B. D. Rao, K. Engan, K. Kreutz-Delgado, "Sparse solutions to linear inverse problems with multiple measurement vectors," IEEE Trans. Signal Process., vol. 53, no. 7, pp. 2477-2488, Jul. 2005.
- [18] J. Chen, X. Huo, "Theoretical Results on Sparse Representations of Multiple-Measurement Vectors", IEEE Trans. Signal Process., vol. 54, no. 12, pp. 4634-4643, Dec. 2006.
- [19] M. Mishali, Y. C. Eldar, "Reduce and Boost: Recovering Arbitrary Sets of Jointly Sparse Vectors", IEEE Trans. Signal Process., vol. 56, no. 10, pp. 4692-4702, Oct. 2008.