

UNIVERSITÀ DEGLI STUDI DI PADOVA

FACOLTÀ DI INGEGNERIA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Corso di Laurea Magistrale in Bioingegneria

**COMPRESSIONE DEI SEGNALI BIOMEDICI
PER APPLICAZIONI IN TELEMEDICINA E
IN SENSORI INDOSSABILI**

Laureanda

Alice Gardin

Alice Gardin

Relatore

Prof. Giovanni Sparacino

Giovanni Sparacino



Anno Accademico 2019-2020

A Emma

ABSTRACT

Le nuove tecnologie possono contribuire a migliorare la sanità in termini di efficacia, efficienza e conformità alle diverse esigenze dei pazienti.

La raccolta dei dati può essere eseguita occasionalmente in ambiente ospedaliero tramite strumentazioni biomediche o in maniera continuativa tramite dispositivi.

Considerata l'ingente quantità di dati generati, si rende necessaria una loro compressione al fine di facilitarne la trasmissione e l'archiviazione, mantenendone invariata la qualità.

In particolare, la tesi tratta lo studio di alcuni algoritmi implementabili in dispositivi indossabili per la compressione di segnali elettrocardiografici.

SOMMARIO

1. IL PROBLEMA DELLA RAPPRESENTAZIONE EFFICIENTE DEI SEGNALI BIOMEDICI

1.1 La tecnologia digitale in sanità.....	1
1.2 La telemedicina.....	3
1.3 Il Fascicolo Sanitario Elettronico.....	7
1.4 I dispositivi elettromedicali.....	9
1.5 Le tendenze attuali e le sfide future della sanità digitale.....	15
1.6 Il problema della rappresentazione efficiente dei dati sanitari	17
1.7 Lo scopo della tesi.....	18

2. IL CASO DI STUDIO DEL SEGNALE ELETTROCARDIOGRAFICO

2.1 L'origine elettrofisiologica del segnale ECG.....	19
2.2 Le caratteristiche delle misurazioni del segnale ECG.....	21
2.3 Le caratteristiche del segnale ECG.....	24
2.4 L'ingombro del segnale ECG a fini trasmissione e archiviazione.....	26

3. LA COMPRESSIONE DEL SEGNALE ELETTROCARDIOGRAFICO

3.1 Aspetti introduttivi.....	27
3.2 Le tecniche considerate.....	29
3.3 I database utilizzati.....	31
3.4 Le metriche di valutazione.....	32

4. IL METODO TURNING POINT

4.1 La descrizione dell'algoritmo.....	33
4.2 L'implementazione in MatLab.....	35
4.3 I risultati dell'implementazione.....	37

5. IL METODO COMPRESSED SENSING	
5.1 La descrizione dell'algoritmo.....	45
5.2 L'implementazione in MatLab.....	47
5.3 I risultati dell'implementazione.....	49
6. IL METODO FAN	
6.1 La descrizione dell'algoritmo.....	57
6.2 L'implementazione in MatLab.....	60
6.3 I risultati dell'implementazione.....	62
7. DISCUSSIONE E VALUTAZIONE	
7.1 Valutazione conclusiva e comparazione degli algoritmi.....	71
7.2 Conclusioni.....	72
APPENDICE.....	73
BIBLIOGRAFIA.....	79

RINGRAZIAMENTI

Ringrazio il prof. Sparacino per il prezioso contributo e sostegno durante la stesura di questa tesi.

Ringrazio la mia famiglia per il costante e indispensabile supporto.

Ringrazio gli ingegneri e i medici incontrati nel mio percorso che mi hanno aiutata a superare me stessa.

Ringrazio i miei compagni di studio e i miei amici per tutto il resto.

1. IL PROBLEMA DELLA RAPPRESENTAZIONE EFFICIENTE DEI SEGNALI BIOMEDICI

1.1 La tecnologia digitale in sanità

Si stima che in Unione Europea siano 120 milioni le persone che necessitano un'attenta assistenza medica a causa di una qualche forma di invalidità [1]. Questo dato è da considerarsi in crescita per il progressivo invecchiamento della popolazione che sarà composta nel 2060 per il 30% di ultrasessantacinquenni e per il 12% da ultraottatenni [2]. Si rende quindi necessario l'adeguamento dell'offerta dell'assistenza sanitaria a questo importante aumento di richiesta.

Come si vedrà nei prossimi paragrafi (1.2-1.4), le nuove tecnologie possono contribuire a migliorare la sanità in termini di efficacia, efficienza e conformità alle diverse esigenze dei pazienti. In particolare, possono risultare utili nella gestione di emergenze e urgenze, nell'approccio continuativo e multidisciplinare a patologie croniche e multimorbilità, nello sviluppo delle prassi di prevenzione e nella riorganizzazione efficiente della rete dei servizi con l'obiettivo di potenziarne la distribuzione.

La digitalizzazione permette inoltre una maggior collaborazione tra medici e pazienti, creando nuove opportunità di miglioramento dei servizi. Si pensi ad esempio alla possibilità di far giungere direttamente a domicilio le prestazioni di uno specialista, permettendo una gestione autonoma dei problemi sotto il controllo del medico, al quale sono forniti dati, che per quantità e qualità risultano utili per una precisa valutazione.

La raccolta dei dati può essere eseguita occasionalmente in ambiente ospedaliero tramite strumentazioni biomediche o in maniera continuativa tramite dispositivi, fornendo al medico informazioni in tempo reale sui diversi aspetti dello stato di salute degli assistiti. I dati raccolti comprendono testi con dati anagrafici e storia clinica del paziente, immagini (risonanze magnetiche, radiografie, ecografie, tomografie computerizzate e a emissione di positroni, ultrasuoni, immagini riguardanti l'evoluzione di un particolare sintomo o lo stato di salute generale del paziente), video (endoscopie, ecografie, videoconferenze di consulto, riabilitazione o terapia) e registrazioni di parametri vitali come attività cardiaca, attività cerebrale, pressione arteriosa, temperatura corporea, variazioni di volume dei gas intratoracici, livello di idratazione, valori emodinamici.

Considerata l'ingente quantità di dati generati, si rende necessaria una loro elaborazione efficiente al fine di facilitarne la trasmissione e l'archiviazione, mantenendone invariata la qualità.

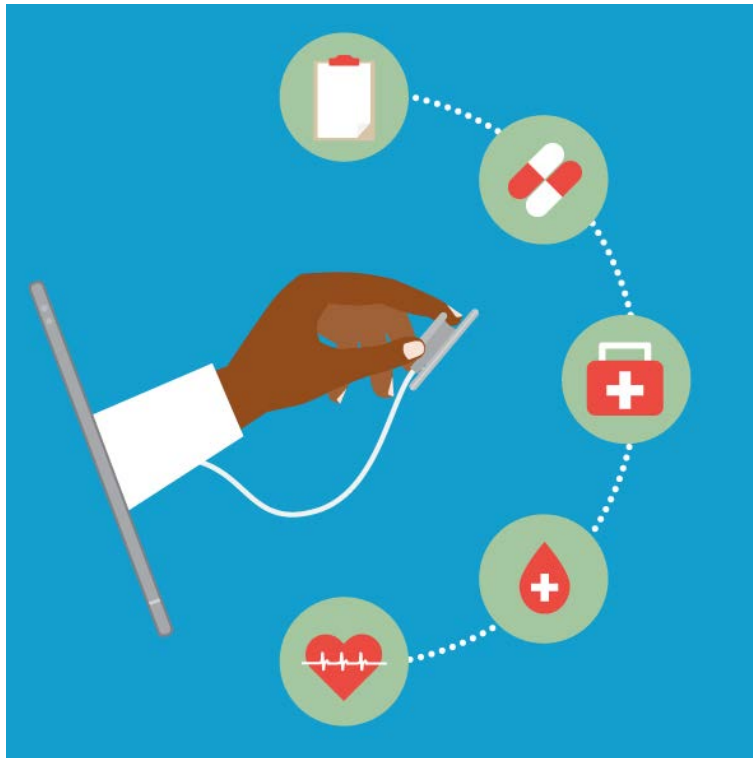


Figura 1.2 immagine tratta da <https://executiveeducation.hms.harvard.edu/>

1.2 La telemedicina

La telemedicina è l'insieme dei servizi sanitari offerti grazie alle tecnologie dell'informazione e della comunicazione al fine di rendere la medicina proattiva, preventiva, personalizzata e basata sulle evidenze con l'obiettivo di migliorare la qualità della vita.

È possibile condividere qualsiasi tipo di informazione (testi, immagini, segnali) con il personale medico, per il quale risulta vantaggioso avere una visione esaustiva dello stato di salute dei pazienti ai fini di diagnosi e trattamenti efficaci e efficienti.

I dati possono essere raccolti e successivamente inviati al medico (es. MRI e RX) per avere un parere sulla diagnosi e sulla gestione ottimale di un problema oppure possono essere trasmessi attraverso diversi devices in tempo reale (es. ecg e televisite) coinvolgendo nello scambio delle informazioni nello stesso momento paziente e medico.[3]

Affinché il servizio risulti funzionale per tutti, sono fondamentali sia la concezione che la facilità di utilizzo delle tecnologie e delle applicazioni mobili. Sono inoltre oggetto di continuo studio approcci all'analisi e alla gestione di ingenti volumi di dati. [2]

Recenti studi hanno indagato l'opportunità di delegare ad algoritmi intelligenti la valutazione dei dati e l'elaborazione di programmi di cura, cercando di testarne la realizzabilità nel campo dell'assistenza agli anziani. Se i dati in ingresso, registrati con il fine di rappresentare nella maniera più completa la situazione del paziente, spaziano dalle informazioni anagrafiche a quelle legate allo stato psicofisico, dalle abitudini al contesto sociale, dal corredo genetico alla fisiologia, dall'educazione alla situazione socioeconomica, la terapia fornita in uscita deve essere accurata e di facile comprensione anche per il paziente direttamente coinvolto nella sua pratica. Per comprendere se e come sfruttare gli algoritmi, è necessario analizzare il disegno e l'esecuzione dei programmi di assistenza, che, essendo entrambe attività svolte da uomini, possono essere limitate per efficienza e riproducibilità. Una volta automatizzato l'intero processo, si evidenzia una carenza di accuratezza causata dalla difficile raccolta di dati in alta definizione in ambiente non ospedaliero che seguano linee guida standard, dal limitato dataset utilizzato per la fase di training e basato sull'esperienza relativamente soggettività dei medici e l'associazione di un dato sintomo a una cura, impossibile da stabilire in maniera univoca. È ragionevole, quindi, che gli algoritmi debbano essere un supporto per il lavoro del medico e che la telemedicina non sia un'alternativa alle prestazioni sanitarie tradizionali, ma rappresenti piuttosto una loro integrazione con il fine di potenziarne l'appropriatezza.[4]

La telemedicina si pone diverse finalità:

- monitoraggio: trasmissione dei parametri vitali del paziente per un periodo di tempo variabile ad una postazione di monitoraggio per l'analisi dei dati;
- diagnosi: come precedentemente specificato, la telemedicina non rende possibile eseguire l'intero processo di diagnosi unicamente attraverso i suoi mezzi, ma può essere un supporto, che consente di approfondire la conoscenza dello stato di salute del paziente;
- prevenzione secondaria: servizi dedicati a persone già affette da patologie (es. cardiovascolari o diabete) che necessitano di costante monitoraggio dei parametri

vitali (es. attività cardiaca o tasso di glicemia nel sangue) per ridurre il rischio di complicazioni;

- cura: servizi dedicati a pazienti con diagnosi certa e finalizzati ad operare scelte terapeutiche e a valutare l'andamento delle prognosi. Ne sono un esempio la teledialisi, alcune tipologie di chemioterapia e di cure palliative, interventi chirurgici a distanza;
- riabilitazione: interventi riabilitativi erogati presso il domicilio del paziente o in strutture assistenziali e rivolti a bambini, pazienti fragili o con patologie croniche, anziani, disabili.

È possibile, inoltre, classificare la telemedicina a seconda del servizio offerto e dell'ambito di applicazione, come si vede in Figura 1.2.b.

TELEMEDICINA					
CLASSIFICAZIONE		AMBITO	PAZIENTI	RELAZIONE	
TELEMEDICINA SPECIALISTICA	TELE VISITA	sanitario	Può essere rivolta a patologie acute, croniche, a situazioni di post-acuzie	Presenza attiva del Paziente	B2C B2B2C
	TELE CONSULTO			Assenza del Paziente	B2B
	TELE COOPERAZIONE SANITARIA			Presenza del Paziente, <i>in tempo reale</i>	B2B2C
TELE SALUTE		sanitario	E' prevalentemente rivolta a patologie croniche	Presenza attiva del Paziente	B2C B2B2C
TELE ASSISTENZA		socio-assistenziale	Può essere rivolta ad anziani e fragili e diversamente abili		

* B2B: individua la relazione tra medici

B2B2C: individua la relazione tra un medico e un paziente mediata attraverso un operatore sanitario

B2C: individua la relazione tra medico e paziente

Figura 1.2.b. Tabella di classificazione della Telemedicina (tratta da [5])

La telemedicina specialistica offre servizi all'interno di una specifica disciplina della medicina e può essere declinata in diverse modalità:

- televisita: servizio sanitario che permette l'interazione a distanza tra medico e paziente (e eventuale personale sanitario) e consente la prescrizione di farmaci o cure. Il collegamento deve permettere una comunicazione sia audio che video e può avvenire in tempo reale o differito;

- teleconsulto: servizio che consente indicazione di diagnosi e/o di una terapia, senza la presenza del paziente. Si tratta di una consulenza a distanza tra medici con specifica formazione e competenza;
- telecooperazione sanitaria: atto di assistenza fornito da medico o operatore sanitario a colleghi impegnati in un atto sanitario (es. soccorso d'urgenza).

La telesalute riguarda i sistemi e i servizi che collegano medici e pazienti (di solito cronici) con finalità di diagnosi, supporto, gestione e monitoraggio.

Il medico di medicina generale (con l'eventuale collaborazione di un medico specialista) interpreta a distanza i dati registrati e trasmessi dal paziente, da un operatore sanitario o da devices in maniera autonomizzata. Grazie alla Telesalute anche il paziente ha un ruolo attivo nella gestione della terapia e nel miglioramento del servizio.

La teleassistenza è una prestazione socioassistenziale a domicilio rivolto a persone anziane o fragili. È erogato da un centro servizi, che gestisce allarmi, eventuali servizi di emergenza e chiamate di supporto. Avendo una finalità prettamente sociale, collabora con i servizi sanitari per garantire una continuità assistenziale.[5]



Figura 1.3 immagine tratta da <https://www.labdoc.it/>

1.3 Il Fascicolo Sanitario Elettronico

Il Fascicolo Sanitario Elettronico (FSE) è lo strumento attraverso cui si tracciano e si consultano dati e documenti sanitari di eventi clinici in formato digitale. Tutte le informazioni riguardanti lo stato di salute del paziente (anamnesi, terapie, esami di laboratorio, parametri vitali) sono inserite dal Medico di Medicina Generale e da altri Specialisti e messe a disposizione di qualunque personale sanitario, a cui il paziente abbia consentito l'accesso. Nel FSE ogni evento riguardante la storia sanitaria di un paziente è tracciato e codificato al fine di agevolare l'assistenza (soprattutto nelle situazioni di emergenza) e fornire una base informativa consistente, integrando le informazioni fornite da diverse figure professionali, scelte liberamente dal paziente. Rappresenta, quindi, una struttura unica a servizio delle comunicazioni tra medici e tra paziente e medico. Questo strumento può incentivare la rivisitazione dei processi clinici quali le attività di prevenzione, diagnosi, cura e riabilitazione, lo studio e la ricerca scientifica in ambito medico, la programmazione sanitaria, le attività di verifica della qualità delle cure e dell'assistenza sanitaria.

Il FSE contiene la storia clinica del paziente attraverso dati e documenti ed è costituito dall'obbligatorio 'nucleo minimo' e da altri documenti ritenuti facoltativi a causa anche dei limiti tecnologici di alcune aziende ospedaliere.

Il nucleo minimo del FSE contiene i dati identificativi e amministrativi dell'assistito, i referti, i verbali pronto soccorso, lettere di dimissione, *profilo sanitario sintetico (PSS)*, il dossier farmaceutico e l'eventuale consenso alla donazione degli organi e tessuti.

Il *Profilo Sanitario Sintetico (PSS)*, detto anche 'Patient Summary', è particolarmente importante in quanto ricopre la funzione di carta d'identità sanitaria. È redatto e aggiornato dal Medico di Medicina Generale per garantire un'assistenza continuativa e per migliorare la qualità della cura soprattutto in caso di emergenza o mobilità. Contiene i dati identificativi di paziente e medico curante e tutte le informazioni cliniche che descrivono lo stato di salute come diagnosi, allergie, le terapie farmacologiche per eventuali patologie croniche.

I dati di tipo integrativo comprendono prescrizioni (specialistiche, farmaceutiche, ecc.), prenotazioni (specialistiche, di ricovero, ecc.), cartelle cliniche, bilanci di salute, scheda, programma e cartella clinico-assistenziale dell'assistenza domiciliare, piani diagnostico-terapeutici, scheda multidimensionale di valutazione dell'assistenza residenziale e semiresidenziale, erogazione farmaci, vaccinazioni, prestazioni di assistenza specialistica, prestazioni di emergenza urgenza (118 e pronto soccorso), prestazioni di assistenza ospedaliera in regime di ricovero, certificati medici, *taccuino personale dell'assistito*, relazioni relative alle prestazioni erogate dal servizio di continuità assistenziale, autocertificazioni, partecipazione a sperimentazioni cliniche, esenzioni, prestazioni di assistenza protesica, dati a supporto delle attività di telemonitoraggio e dati a supporto delle attività di gestione integrata dei percorsi diagnostico-terapeutici.

Il *taccuino personale dell'assistito* è uno spazio gestito dal paziente secondo modalità di accesso standard. Vi si possono aggiungere informazioni importanti non presenti nella storia clinica, dati e documenti riguardanti i percorsi di cura svolti al di fuori dal Servizio Sanitario Nazionale, assicurando una visione esaustiva dello stato di salute.[6]

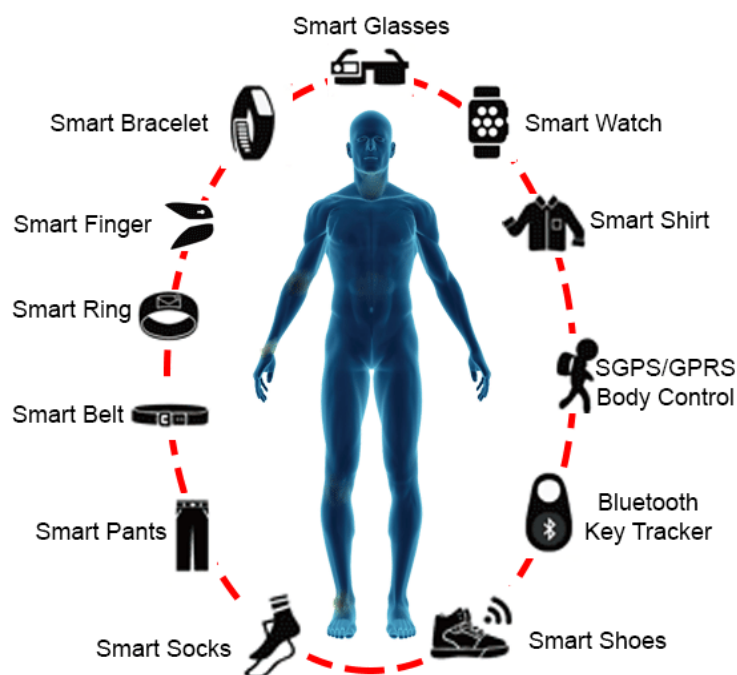


Figura 1.4 immagine tratta da <https://www.tecomilano.it/>

1.4 I dispositivi elettromedicali

La creazione dei servizi di Telemedicina ha richiesto lo sviluppo di sistemi di telemonitoraggio, che utilizzano sensori applicati al corpo del paziente e collegati tramite connessione wireless a reti WSN (wireless sensor network) per la trasmissione dei segnali biomedicali.

Nell'applicazione della tecnologia WSN alla telemedicina si possono distinguere BSN (Body Sensor Networks) e WBAN (Wireless Body Area Network), che consistono in un insieme di sensori medici situati all'interno, sulla superficie o all'esterno del corpo. Con i WBANs vengono registrati i parametri vitali per la diagnosi o la valutazione di malattie croniche come:

- le malattie cardiovascolari che sono valutate con sensori ECG posizionati sul torace al fine di monitorare l'attività cardiaca;
- il diabete che è valutato con l'ausilio di un sensore CGM inserito sottopelle, che misura le variazioni del livello di glucosio nel tempo;

- l'epilessia monitorata con sensori EEG posizionati sul cuoio capelluto, che registrano l'attività elettrica del cervello;
- i disturbi muscolo-scheletrici che sono esaminati con sensori EMG inseriti sulla superficie del corpo in corrispondenza del muscolo del quale si vuole monitorare l'attività;
- le malattie respiratorie sono monitorate con sensori SO₂ da dito che viene misurano la saturazione di ossigeno nel sangue.

I WBANs possono essere utilizzati per fornire trattamenti terapeutici tramite attuatori che agiscono attraverso la stimolazione elettrica o le iniezioni biochimiche. Esempi tipici sono i pacemaker che erogano impulsi elettrici attraverso elettrodi per regolare la conduzione elettrica del cuore, impianti cocleari che catturano il suono con un microfono e lo convertono in un segnale elettrico per stimolare direttamente il nervo uditivo, stimolatori neuromuscolari che inviano impulsi elettrici ai muscoli, imitando l'azione del sistema nervoso centrale e la pompa utilizzata per somministrare l'insulina in base al livello di glucosio nel sangue.

La struttura di un WBAN (Figura 1.4.a) è composta da tre diversi livelli: il primo livello, intra WBAN, ha un raggio di trasmissione tipico di circa un metro o più e permette la comunicazione tra i nodi del sensore e un nodo coordinatore. I nodi del sensore possono essere sensori che misurano dati fisiologici o attuatori che, in seguito a informazioni di feedback provenienti dai sensori, conducono una stimolazione elettrica o biochimica. Per quanto riguarda il secondo livello, extra WBAN, la comunicazione avviene tra il nodo coordinatore e uno o più punti di accesso (AP), che possono far parte dell'infrastruttura o essere collocati in una modalità ad hoc. Dopo aver raccolto, eventualmente elaborato e compresso i dati da diversi nodi dei sensori, il nodo coordinatore li trasmette all'AP. Il terzo livello, beyond-WBAN, interessa il collegamento via internet o rete cellulare tra l'AP e il server medico e consente che i dati vengano archiviati in database e/o utilizzati dal personale medico per diagnosi o semplice monitoraggio. [7]

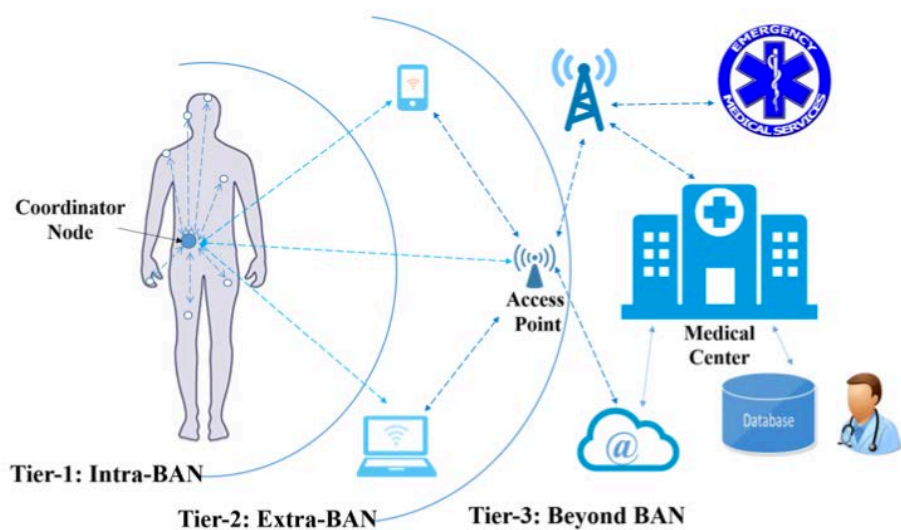


Figura 1.4.a Rappresentazione di WBAN con diversi livelli di comunicazione (tratta da [7])

WearIoT, ECG WATCH e AMICO sono alcuni recenti progetti di WBAN.

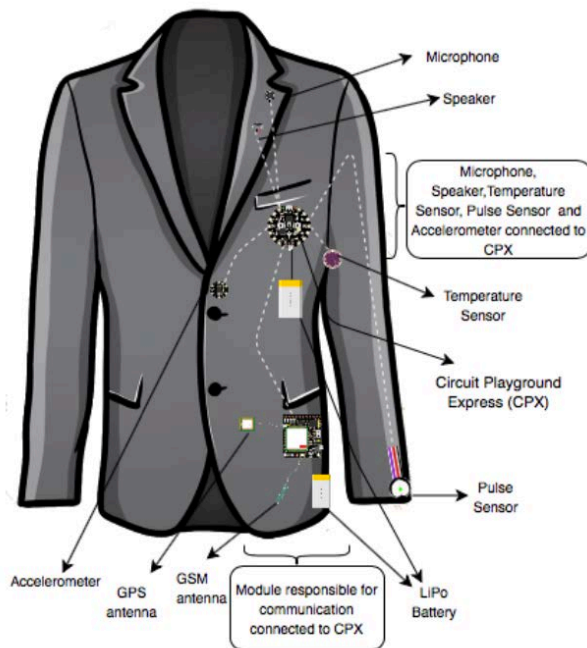


Figura 1.4.b WearIoT (tratta da [8])

Le batterie del sistema hanno un'autonomia fino a tre giorni.

WearIoT [8] è un sistema di monitoraggio costituito da sensori e implementato su una giacca (Figura 1.4.b).

Permette di acquisire temperatura corporea e frequenza cardiaca, rilevare cadute grazie ad un sensore GPS e inviare allarmi.

È stata creata una giacca con solo tecnologia indossabile rimovibile e comoda, in cui i sensori non causino disagio all'utente.



Figura 1.4.c ECG WATCH (tratta da [9])

L'ECG WATCH [9] è un dispositivo a basso costo, indossabile, wireless e non intrusivo che esegue su richiesta una registrazione ECG a una derivazione in soli 10 secondi in ogni momento e in ogni luogo, rilevando anomalie sporadiche come la fibrillazione atriale (Figura 1.4.c). Può registrare qualsiasi derivazione e condividere i risultati con i medici semplicemente toccando un pulsante su un software per smartphone o PC.

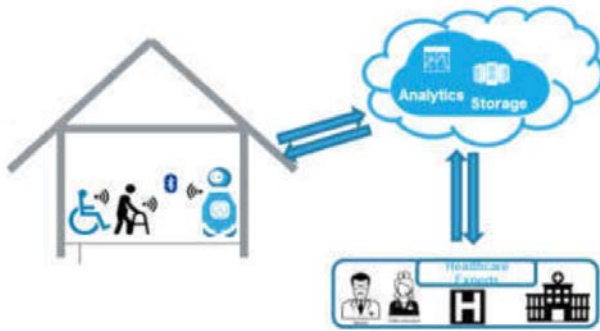


Figura 1.4.d Schema del progetto AMICO (tratta da [10])

AMICO (Medical Assistance in Contextual awareness) [10] è un sistema composto dall'ambiente domestico, dalla persona e da un robot (Figura 1.4.d). La casa e il paziente sono dotati di sensori e interfacciano una piattaforma di servizi di

telemedicina. Il robot opera come mediatore tra la persona e l'ambiente. Il robot sarà dotato di sensori per rilevare la direzione dei suoni, il riconoscimento dei comandi vocali e una videocamera per il riconoscimento facciale e la valutazione dello stato psicologico del paziente.

I nodi di una WBAN richiedono un range di velocità di trasmissione che va dai 10 Kbps ai 10 Mbps: si spazia, infatti, dai bps richiesti per la trasmissione di valori da un sensore di temperatura ai Mbps richiesti per quella del segnale trasmesso da un sensore EMG (si veda Figura 1.4.e). La rete dovrebbe garantire l'affidabilità della comunicazione, che è quantificata nel bit-error-rate (BER), che dipende dal particolare device medicale e assume valori compresi tra 10^{-10} e 10^{-3} , come evidenziato in Figura 1.4.f.

Sensor nodes	Target data rate	Target BER
EEG	86.4 Kbps	$< 10^{-10}$
EMG	1.536 Mbps	$< 10^{-10}$
SPO2	32 bps	$< 10^{-10}$
Temperature, glucose level, pH	20 bps	$< 10^{-10}$
Cochlear implant	~ 200 Kbps	$< 10^{-10}$
Deep brain stimulation	128 – 320 Kbps	$< 10^{-3}$
Accelerometer, blood pressure	< 10 Kbps	$< 10^{-10}$
Endoscope capsule	1 Mbps	$< 10^{-10}$

Figura 1.4.e Velocità di trasmissione e BER di alcuni dispositivi (tratta da [7])

È importante che i WBAN siano a basso consumo di energia per l'influenza che può avere sulla durata della batteria dei dispositivi medicali, la cui sostituzione può essere onerosa o richiedere operazioni chirurgiche. Non è da trascurare il fatto che un eccessivo consumo di energia possa provocare un aumento della temperatura nel tessuto circostante il sensore, provocando danni termici ad organi sensibili. Per questo per i sensori WBAN il tasso di assorbimento specifico (SAR) di energia deve essere ridotto al minimo, in particolare è fissato a 2 W / Kg su 10 g di tessuto nei paesi dell'Unione Europea e in Giappone e 1,6 W / Kg su 1 g di tessuto negli Stati Uniti e in Australia.

Sebbene non soddisfino l'affidabilità, la velocità di trasmissione e il SAR richiesti da alcuni dispositivi medicali, i sistemi di comunicazione più usati per la costruzione di WBAN sono Bluetooth, che consente una trasmissione fino a 2 Mbps, e Zigbee (versione a basso costo e a bassa energia della tradizione rete wireless), che si assesta tra i 20 e 250 Kbps.

Le tecnologie basate su radiofrequenze sono una soluzione assodata per connettere WBANs, ma i rischi di interferenza, sovraffollamento, errore e eccessivo consumo di energia incentivano l'uso di tecnologie alternative o complementari come le comunicazioni ad ultrasuoni e le comunicazioni wireless ottiche.

I trasduttori piezoelettrici potrebbero essere un supporto per le trasmissioni di dispositivi all'interno del corpo, in quanto la propagazione degli ultrasuoni è poco attenuata dai tessuti e raggiunge frequenze maggiori di 20 KHz.

Le comunicazioni ottiche wireless (OWC) nella banda infrarossa, visibile o ultravioletta dello spettro sono un'ulteriore alternativa o un complemento alle tecnologie RF per i WBAN grazie alla loro elevata immunità alle interferenze esterne e alla loro sicurezza intrinseca dovuta al confinamento negli spazi interni. Inoltre, la potenza di trasmissione nei sistemi OWC non è limitata dalle normative SAR come per le tecnologie RF. Si è stimato che un WBAN realizzato con ray tracing consenta trasmissioni tra i 10 e gli 890 Kbps con basso BER e un consumo di energia di 17 mW per la velocità minima.

L'insieme di tutti i WBAN forma una rete strutturata dinamica denominata Internet of Things (IoT) [11], in grado di auto configurarsi in base a protocolli di comunicazione standard e interoperanti. Si può considerare IoT come un'infrastruttura flessibile, complessa e dinamica che connette chiunque, qualunque cosa, in ogni momento, ovunque per qualsiasi servizio.

Se Internet connette le persone alla rete, IoT permette il collegamento tra dispositivi eterogeni (D2D) e tra uomo e dispositivo (H2D) fungendo da supporto a varie applicazioni adibite ad esempio all'identificazione, alla localizzazione, al tracciamento, al monitoraggio e al controllo.

Si prevede uno sviluppo di IoT anche grazie all'introduzione della tecnologia 5G, caratterizzata da alta velocità, capacità e scalabilità. Le funzionalità chiave di questa nuova wireless sono bassa latenza pari o inferiore a 1 ms, velocità di trasmissione dei dati nel range 10 ÷ 20 Gbps, alta densità di connessioni di cui una parte massiva di tipo D2D e alta mobilità fino ad un massimo di 500 Km/h.

Le due principali sfide da affrontare sono l'intensificazione delle connessioni dovuta all'utilizzo di una grande quantità di devices (nel 2020 si stimano 10^6 connessioni per Km^2) e l'alto consumo di energia richiesto dalle applicazioni dell'IoT. È da tenere in considerazione che l'energia richiesta per la trasmissione dei dati supera quella utilizzata dalle funzioni di elaborazione, rilevamento, comunicazione e monitoraggio e che alcuni devices richiedono una batteria di durata almeno 10 anni (si pensi alla batteria di un pacemaker).[7]

1.5 Le tendenze attuali e le sfide future della sanità digitale

Oltre ai suddetti progressi, ci sono numerose sfide e problemi di ricerca aperti dall'utilizzo della telemedicina e, in generale, della sanità digitale.

Una rete sanitaria è composta da miliardi di dispositivi e può avere successo solo se è in grado di fornire connettività a tutti i dispositivi ad essa connessi con le loro specifiche capacità di rilevamento di diverse tipologie di informazioni. Si ricorda che qualsiasi rete di comunicazione disponibile può essere utilizzata dai dispositivi IoT come Bluetooth, Wi-Fi, LTE e 5G.

Le sfide da affrontare sono molte:

- garantire la connettività ad un'ampia gamma di dispositivi distribuiti nella rete o a dispositivi in alta mobilità (ad es. in un'ambulanza ad alta velocità durante il trasporto di pazienti);
- permettere l'interoperabilità tra due o più reti o dispositivi diversi, ossia l'interconnessione per lo scambio di dati. Essa svolge nel campo della telemedicina un ruolo importante in quanto permette la connettività tra diversi dispositivi IoT di vari settori (ad es. chirurgia a distanza e monitoraggio ECG) utilizzando tecnologie di comunicazione diverse, con il fine di sopperire alla mancanza di standard universali. Per superare il problema dell'interoperabilità è necessario operare a livello di applicazione, dispositivo, comunicazione e rete;
- fornire una fonte continua di energia necessaria per mantenere in funzione i dispositivi IoT utilizzati per la telemedicina, il che rappresenta una grande sfida sia in termini di costi che di durata della batteria. Per affrontare questi problemi, i dispositivi devono essere a basso costo e a basso consumo energetico. Pertanto, sono necessarie innovazioni nel dominio della microelettronica per diminuire la spesa e le dimensioni dei dispositivi e progressi nei software e nella comunicazione wireless che consentano ai dispositivi di comunicare tra loro con un minore consumo di energia;
- tutelare la sicurezza, la cui importanza deriva dal fatto che la telemedicina si basa sulla connessione di diversi dispositivi, limitati sia nell'elaborazione che nella durata della batteria, e risulta difficile implementarvi anche protocolli e algoritmi di sicurezza complessi. Questo porta al rischio di numerosi attacchi e minacce per

sicurezza e privacy. Per progettare una telemedicina di successo, è necessario tenere conto della necessità di fornire una comunicazione consapevole sulla privacy all'utente, di garantire l'integrità e l'autenticità dei dati con una comunicazione semplice e sicura tra i dispositivi e un centro applicativo basato su cloud e di valutare dettagliatamente i rischi per la privacy. [11]

1.6 Il problema della rappresentazione efficiente dei dati sanitari e lo scopo della tesi

I dati hanno un ruolo centrale nella Sanità Digitale ed è necessario trattarli sfruttando al meglio i mezzi tecnologici a disposizione.

La maggior parte dei dispositivi utilizzati per la raccolta dei dati hanno risorse limitate, in particolare sono generalmente muniti di piccole memorie RAM e flash e di batterie durata ridotta.

Queste limitazioni creano la necessità di ridurre le dimensioni dei dati al fine di minimizzare i cicli di CPU delle elaborazioni, lo spazio destinato all'archiviazione e la larghezza di banda utilizzata la trasmissione. Ridurre il traffico di rete è essenziale, infatti, per evitare la saturazione e far sì che molti dispositivi funzionino in modo cooperativo all'interno dello stesso hub. [12]

I limiti energetici sono un vincolo critico perché quando l'energia della batteria è esaurita, il monitoraggio dello stato viene interrotto fino a quando la batteria non viene caricata o sostituita. L'eventuale dotazione di batterie di grande capacità sui dispositivi porta ad un aumento di dimensioni e peso ed un minor confort per i pazienti. L'alternativa è una raccolta sporadica di energia per fornire alimentazione continua e le fonti più promettenti possono essere sia ambientali (come energia solare, energia termoelettrica, radiofrequenza (RF) e possono essere influenzate dal tempo, dalla posizione e dalle condizioni meteorologiche) sia corporee, le quali, a differenza delle prime, sono prevedibili e controllabili perché sono distribuite in maniera esplicita. L'idea è stata concepita perché l'energia umana, che proviene dal consumo di cibo, può essere trasformata in energia chimica, termoelettrica e cinetica, sfruttabili nei dispositivi indossabili, considerato che l'energia umana può fornire 35-100 volte più energia di una batteria. [13]

In questo contesto, la progettazione di algoritmi per la compressione dei segnali permette che siano archiviati nello spazio di memoria limitato dei dispositivi indossabili e efficacemente trasmessi.

1.7 Lo scopo della tesi

La tesi ha come scopo l'analisi di alcuni algoritmi di compressione utilizzabili nell'ambito delle tecnologie IoT, selezionati non solo per la qualità dei risultati, ma anche per l'utilizzo efficiente dell'energia. Solo recentemente, infatti, con lo sviluppo delle nuove tecnologie indossabili si è reso necessario studiare software dalla bassa complessità computazionale che richiedano ridotte risorse per essere implementati. Dall'analisi sono esclusi metodi (ad esempio quelli basati sulle trasformate) con i quali si ottengono ottimi risultati con importante dispendio energetico.

Gli algoritmi sono testati e valutati in base all'entità della compressione e alla qualità della ricostruzione del segnale originale.

In particolare, è oggetto di studio della tesi la compressione di segnali elettrocardiografici eseguita con gli algoritmi Turning Point (4), Compressed Sensing (5) e Fan (6) sui database SHAREE, European ST-T e St Petersburg INCART 12-lead Arrhythmia pubblicati online con fini di ricerca e di educazione (3.3).

2. IL CASO DI STUDIO DEL SEGNALE ELETTROCARDIOGRAFICO

2.1 L'origine elettrofisiologica del segnale elettrocardiografico

Di seguito l'attenzione è rivolta ai segnali elettrocardiografici in quanto sono di particolare interesse data dall'incidenza di malattie cardiache e del sistema circolatorio, che in Europa risultano essere la principale causa di mortalità con più di 3.9 milioni di decessi all'anno (45% del totale), di cui il 40% sono uomini e il 49% donne. [14]

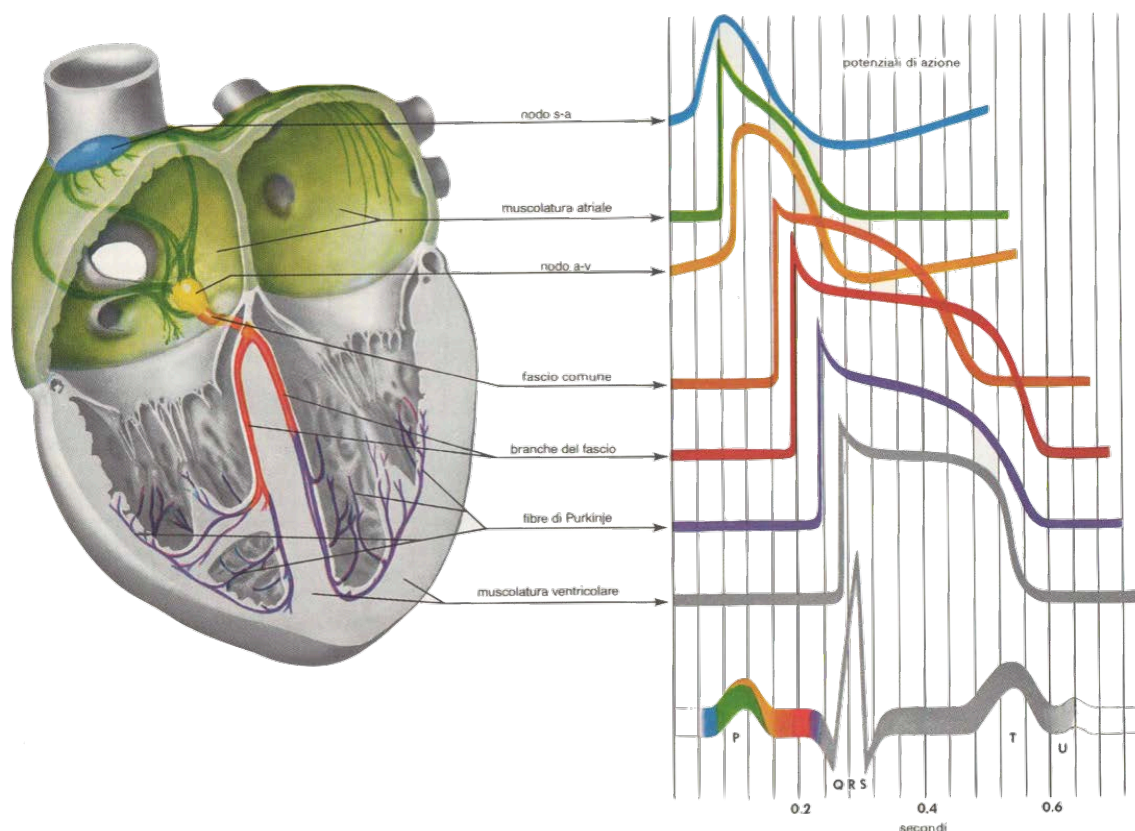


Figura 2.1.a Forme d'onda di potenziali d'azione in differenti zone del cuore e il loro contributo al segnale ECG (tratta da Collezione Ciba, vol.1 Frank Netter)

Il sistema di conduzione del cuore è mostrato in Figura 2.1.a. Il nodo senoatriale (s-a) funge da pacemaker primario e in condizioni normali genera impulsi che stimolano la contrazione degli atri. Questo nodo si trova nella parete superiore dell'atrio destro, vicino all'apertura della vena cava superiore. Altri elementi del sistema di conduzione

comprendono il nodo atrioventricolare (a-v), situato tra gli atri e i ventricoli, nel setto atriale inferiore adiacente all'anello della valvola mitrale e il fascio di His, che si divide in un ramo destro e sinistro, ramificato a sua volta in fascio anteriore e posteriore. Le fibre di Purkinje sono le componenti finali del sistema di conduzione, che si intrecciano con le fibre muscolari e i muscoli papillari. Il loro compito è quello di condurre i fronti d'onda direttamente ai due ventricoli in modo che si contraggano contemporaneamente. Le fibre di Purkinje hanno un'automaticità intrinseca che genera circa 30 bpm (battiti al minuto). Anche le cellule del nodo a-v si depolarizzano spontaneamente ma a un ritmo più elevato (circa 40-50 bpm). In condizioni fisiologiche l'automaticità di questi pacemaker è soppressa dall'attività del nodo s-a.

Se l'attività elettrica appare tardi sulla registrazione dell'ECG del previsto significa che un potenziale d'azione ha avuto origine in uno dei pacemaker inferiori, mentre una comparsa prematura indica la presenza di un battito ectopico prematuro, che di solito è seguito da una pausa compensativa.

L'attività del nodo s-a è regolata principalmente dal sistema nervoso autonomo. La normale frequenza cardiaca a riposo è di circa 60-70 bpm. Il ritmo cardiaco è normalmente regolare ad eccezione di lievi variazioni con la respirazione, specialmente nei giovani.

Utilizzando la terminologia associata ai dispositivi elettrici, il sistema di conduzione del cuore può essere descritto come un pacemaker (nodo s-a), una resistenza che agisce anche come un fusibile (nodo a-v) e due cavi elettrici isolati (rami del fascio dei His). Il termine "resistore" per il nodo a-v è appropriato perché rallenta la depolarizzazione: la velocità di conduzione attraverso il nodo a-v è più lenta (-0,5 m/s) rispetto ad altre parti del sistema di conduzione (4 m/s), consentendo il trasferimento di sangue dagli atri ai ventricoli e garantendo che la sequenza della contrazione ventricolare segua la contrazione atriale. Il nodo a-v può essere comparato anche ad un fusibile grazie alla presenza del fascio di Wenckebach, che, in alcune situazioni, come nella fibrillazione atriale, fornisce protezione contro la propagazione della fibrillazione atriale ai ventricoli, ma può fallire in presenza di percorsi di conduzione accessori che bypassano il nodo a-v. Un sistema di conduzione correttamente funzionante garantisce un'adeguata frequenza cardiaca e contrazioni sequenziali degli atri e dei ventricoli. La disfunzione elettrica cardiaca può essere causata da un danno o da un funzionamento improprio di uno qualsiasi dei

componenti del sistema di conduzione separatamente o in combinazione ad altri problemi. Altre cause di aritmie cardiache possono essere una generazione di stimoli patologici (ad esempio battiti ectopici prematuri) o anelli conduttivi patologici. [15]

2.2 Le caratteristiche delle misurazioni del segnale ECG

Dall'inizio del 1900, grazie al contributo di Einthoven, l'eccitazione cardiaca può essere rappresentata come un vettore. Si disegna un triangolo equidistante con due vertici a due spalle e uno all'ombelico (che rappresenta la gamba sinistra) e il vettore cardiaco rappresenta la diffusione dell'eccitazione cardiaca all'interno del triangolo. La differenza potenziale misurata tra due vertici del triangolo (noti come derivazioni dell'arto) rispetto alla gamba destra, è proporzionale alla proiezione del vettore su ciascun lato del triangolo (Figura 2.2.a).

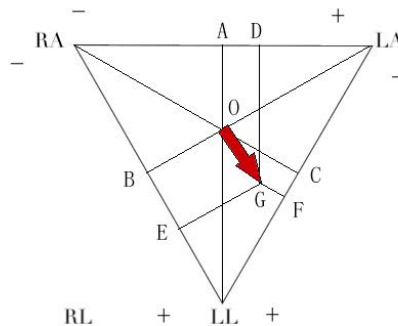


Figura 2.2.a Triangolo di Einthoven (tratta da [13])

Le derivazioni bipolari dell'arto (I, II, III) mirano a calcolare il vettore di depolarizzazione media del cuore nel piano frontale, detto asse elettrico del cuore, che è compreso tra $+30^\circ$ e -110° nel piano frontale. La morfologia dell'ECG registrata dipende dall'orientamento del cuore. Le derivazioni unipolari degli arti aumentati di Goldberger (aVR, aVL e aVF) vengono utilizzate per determinare l'orientamento del cuore (Figura 2.2.b)

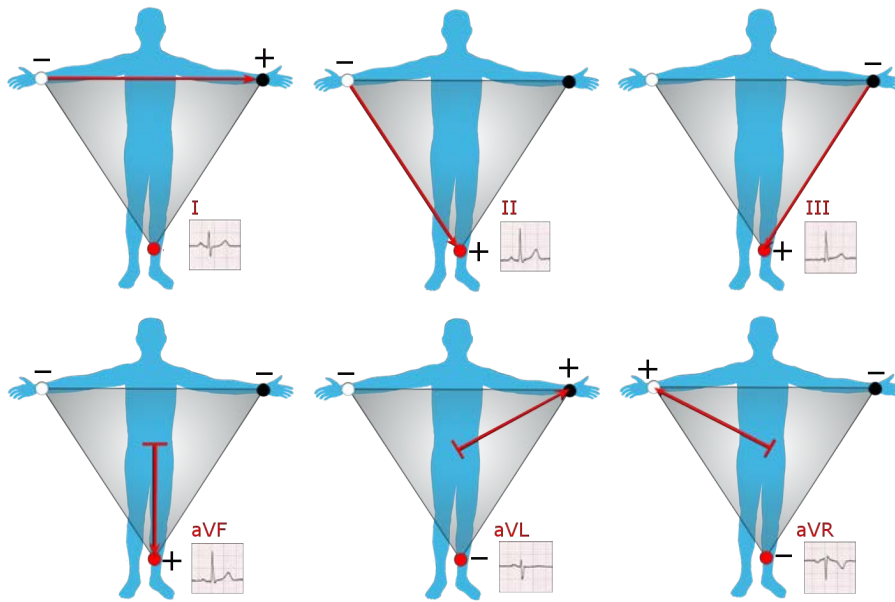


Figura 2.2.b Derivazioni bipolari e derivazioni unipolari aumentate (tratta da [15])

Le derivazioni precordiali unipolari di Wilson V1 e V2 sono posizionate sopra la parete anteriore del ventricolo destro e sono indicate come derivazioni ventricolari destre. Quando il cuore è normalmente orientato lungo l'asse lungo, le derivazioni V5 e V6 sono, invece, posizionate sopra la parete laterale del ventricolo sinistro e sono note come derivazioni ventricolari sinistre. La zona di transizione tra i ventricoli sinistro e destro (setto interventricolare) si trova a livello delle derivazioni V3 e V4 (Figura 2.2.c).

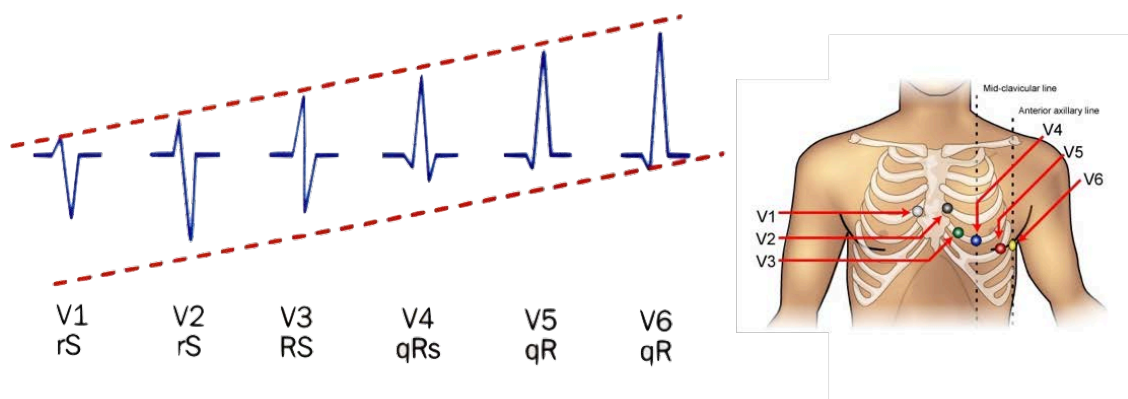


Figura 2.2.c Derivazioni unipolari e rispettive forme d'onda QRS (tratta da Collezione Ciba, vol.1 Frank Netter)

Se la posizione del cuore è verticale la deflessione netta del complesso QRS in aVF è positiva e assomiglia a quella in V6 e risulta negativa in aVL come in V1 (Figura 2.2.d-a). Al contrario, quando l'orientamento del cuore è orizzontale, il complesso QRS in aVF assomiglia a quello in V1 (deflessione negativa), mentre in aVL assomiglia alla deflessione in V6 (Figura 2.2.d-b). [15]

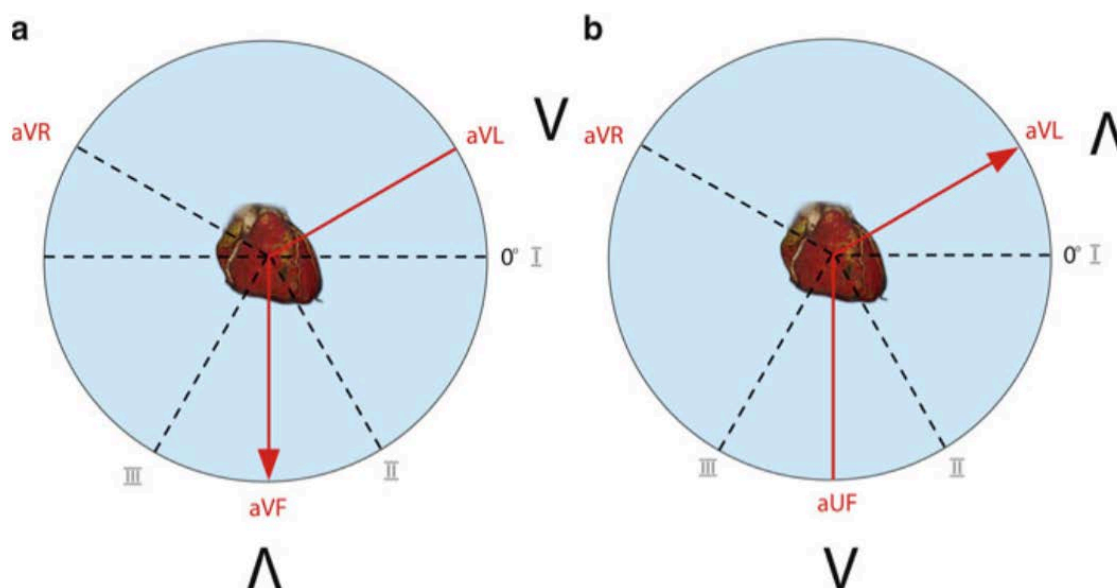


Figura 2.2.d L'orientamento del cuore nella cavità toracica (a) posizione verticale (b) posizione orizzontale (tratta da [15])

2.3 Le caratteristiche del segnale ECG

Il segnale elettrocardiografico è pseudoperiodico con banda di frequenza 1-100 Hz e ampiezza dell'ordine dei millivolt. (Figura 2.3.a)

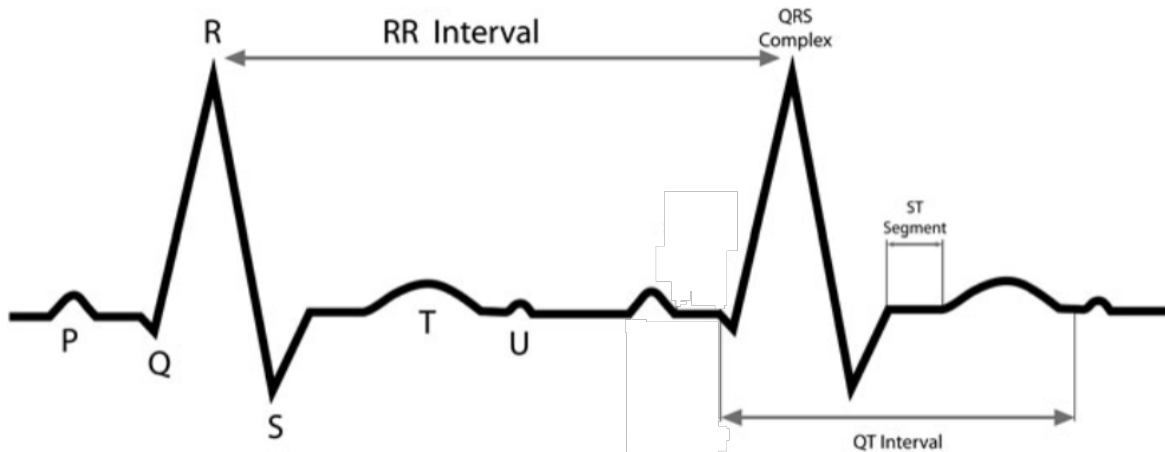


Figura 2.3.a Segnale ECG (tratta da [15])

L'onda P è la prima deflessione dell'ECG e deriva dalla depolarizzazione degli atri. È generalmente positiva in molte derivazioni, eccetto in aVR in cui è sempre negativa. Nelle derivazioni III e V1 può presentare una deflessione negativa o bifasica, che può apparire anche in aVL. La durata normale dell'onda P è minore di 0.12 secondi e l'ampiezza non deve superare 0.25 mV nelle derivazioni degli arti e 0.15 mV in quelle precordiali. Un'onda P dentellata e che supera i valori indicati è considerata anormale. Ciò può derivare dall'allargamento e dall'ipertrofia degli atri o dalla depolarizzazione atriale generata in posizioni diverse dal nodo SA.

Il segmento PQ corrisponde agli impulsi elettrici trasmessi attraverso il nodo s-a, il fascio di His e i suoi rami e le fibre di Purkinje ed è solitamente isoelettrico. Esprime il tempo trascorso dalla depolarizzazione atriale all'insorgenza della depolarizzazione ventricolare e dura in condizioni normali $0.12 \div 0.20$ sec.

Il complesso QRS è il più grande gruppo di onde sull'ECG e rappresenta la depolarizzazione ventricolare: la prima deflessione verso il basso è l'onda Q, seguita dalla

deflessione verso l'alto chiamata onda R e successivamente dall'onda S (deflessione verso il basso). L'onda Q non è un componente costante del complesso QRS (ventricolare). Durante la conduzione fisiologica attraverso i rami del fascio di His, il ventricolo sinistro e destro vengono depolarizzati simultaneamente e si contraggono lungo la direzione del tratto di efflusso ventricolare. La depolarizzazione ventricolare si propaga dalla parte mediale del setto verso il basso attraverso l'apice del cuore e ulteriormente lungo la parete libera dei ventricoli verso la base del cuore. La durata normale del complesso QRS è minore di 0.12 se l'ampiezza di solito varia tra 1.5 e 2.0 mV.

Il segmento ST è il tratto di ECG tra la fine del complesso QRS e l'inizio dell'onda T e in condizioni normali è isoelettrico.

L'onda T rappresenta la ripolarizzazione ventricolare, cioè il ripristino del potenziale di membrana a riposo. Può essere seguita (in circa un quarto della popolazione) da un'onda U, che di solito ha la stessa polarità. L'onda T dovrebbe essere positiva nella maggior parte delle derivazioni ad eccezione di aVR e talvolta in V1, in cui può essere orizzontale o negativa.

L'intervallo QT corrisponde alla durata del potenziale d'azione ventricolare e alla ripolarizzazione. Viene misurato dall'inizio del complesso QRS alla fine dell'onda T e normalmente ha una durata massima di 0.44 s. La lunghezza dell'intervallo QT è direttamente influenzata dalla frequenza cardiaca (più lenta è la frequenza cardiaca, maggiore è l'intervallo). Per eliminare questa influenza, un QT corretto (QT_c) dovrebbe essere calcolato usando la formula di Bazett ($QT_c = QT / RR^{0.5}$) o la formula di Fridericia ($QT_c = QT / RR^{0.33}$). Lo svantaggio è che queste formule sopravvalutano l'intervallo QT quando la frequenza cardiaca è veloce e la sottovalutano quando la frequenza cardiaca è lenta.

Infine, l'intervallo TP è il periodo per il quale gli atri e i ventricoli sono in diastole e l'intervallo RR rappresenta un ciclo cardiaco e viene utilizzato per calcolare la frequenza cardiaca. [15]

2.4 L'ingombro del segnale ECG a fini trasmissione e archiviazione

Sensori ECG	Dimensioni dati per minuto	Dimensioni dati per 24 ore	Dimensioni dati per 30 giorni
Frequenza 125 Hz	30 KB	43.2 MB	1.296 GB
Frequenza 200 Hz	50 KB	72 MB	2.160 GB
Frequenza 500 Hz	110 KB	150.4 MB	4.752 GB
Frequenza 1000 Hz	180 KB	259.2 MB	7.776 GB

Figura 2.4.a Stime delle dimensioni dei dati prodotti da sensori ECG a diverse frequenze

I dati presenti in Figura 2.4.a [16] sono frutto di rilevazioni di laboratorio e indicano le stime delle dimensioni che possono assumere i segnali ECG a seconda della frequenza di campionamento. Ne risulta che un'intera giornata di monitoraggio può portare ad una produzione di dati di dimensione che varia da diverse decine ad alcune centinaia di MB e in 30 giorni le dimensioni diventano dell'ordine dei GB.

Si sottolinea che l'attenzione è rivolta ai soli segnali ECG, ma anche altri segnali possono essere oggetto di monitoraggio, aumentando le dimensioni dei dati raccolti.



Figura 3 immagine tratta da <https://www.softplaceweb.com/>

3. LA COMPRESSIONE DEL SEGNALE ELETTROCARDIOGRAFICO

3.1 Aspetti introduttivi

La compressione è un'elaborazione che permette di ridurre il numero di bit necessari per rappresentare le informazioni contenute nei dati.

Esistono due tipi di compressione (Figura 3.1.a): la compressione lossy, che non permette il recupero i dati originali dal file compresso in quanto vengono eliminati in maniera permanente i dati ridondanti, e la compressione lossless, che può includere una fase di decompressione per riportare i dati allo stato originale.

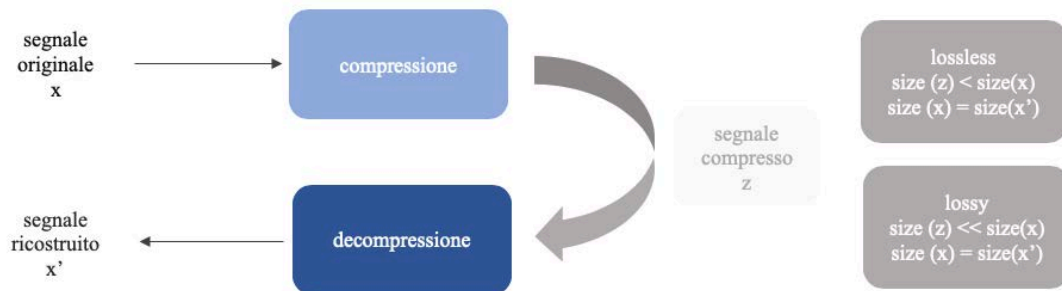


Figura 3.1.a Differenze tra compressione lossy e lossless

La maggior parte delle tecniche di compressione utilizzate nei dispositivi IoT è di tipologia lossy, ha un'implementazione hardware con approccio statistico e ha come obiettivo primario l'ottimizzazione delle risorse energetiche per la trasmissione e l'elaborazione dei dati. [12]

Per quanto riguarda i segnali fisiologici di interesse, come preannunciato, la tesi focalizza sui segnali ECG, importanti per la diagnosi di problemi cardiaci e soggetti ad essere monitorati tramite le tecnologie IoT. È necessario che, nonostante il ridimensionamento del segnale, gli algoritmi rilevino i complessi QRS, dividendo le serie temporali in segmenti, ognuno contenente un battito cardiaco (cosa resa possibile dalla pseudoperiodicità del segnale). Una combinazione di tecniche come la sogliatura, le reti neurali, la trasformazione wavelet e i filtri corrispondenti viene utilizzata per individuare i segmenti, utilizzati per stimare l'impulso e comprimere le tracce dell'ECG.

Per una corretta individuazione, è, inoltre, importante potenziare le caratteristiche del complesso QRS tramite il thresholding dell'ampiezza, i metodi basati sulle derivate di primo e secondo grado, la morfologia matematica e i filtri. Tutto ciò consente una valutazione approfondita delle caratteristiche dei diversi segmenti al fine di individuare eventuali patologie.

Dopo la compressione, al fine di procedere alla visualizzazione e eventuale elaborazione dei segnali, è necessaria la ricostruzione del segnale originale. [17]

Secondo il teorema di campionamento di Shannon, solo un campionamento con frequenza pari almeno al doppio della larghezza di banda del segnale permette di ripristinare l'originale, ma è evidente come l'acquisizione, la trasmissione e l'archiviazione rimangono piuttosto onerose. Inoltre, è da considerare che lo schema tradizionale di compressione prevede che l'intero segnale sia prima campionato e poi elaborato con algoritmi complessi, che inevitabilmente scartano campioni acquisiti.

Nell'ultimo ventennio, ha prevalso l'idea che i dati comprimibili possano essere campionati e ricostruiti accuratamente con frequenza di campionamento inferiore alla frequenza di Nyquist.

Di seguito sono proposte alcune metodologie di compressione utilizzate negli ultimi anni nell'ambito della ricerca sugli algoritmi di elaborazione dei dati per i dispositivi IoT indossabili, che richiedono attenzione sia per la qualità della compressione e della ricostruzione che per il consumo di energia richiesta dal software. [18]

3.2 Le tecniche considerate

Gli algoritmi proposti in letteratura per la compressione di segnali ECG possono essere classificati in tre categorie principali.

1. L'elaborazione del dominio del tempo consente la compressione scartando alcuni dei campioni di segnale e applicando un'approssimazione lineare come nel caso degli algoritmi AZTEC [19] e CORTES [20] oppure approssimando le serie temporali originali attraverso segmenti lineari a tratti, di cui vengono inviati solo i due punti finali (algoritmo Fan [21]).
2. La codifica basata su trasformate sfrutta la Fast Fourier Transform [22], la trasformata discreta del coseno (DCT) [23] e la trasformazione discreta wavelet (DWT) [24]. La logica è quella di rappresentare il segnale in un altro dominio e selezionare un numero di coefficienti di trasformazione da inviare al posto dei campioni originali. La quantità di compressione dipende dal numero di coefficienti selezionati, mentre l'accuratezza della rappresentazione dipende dalla quantità e dalle caratteristiche dei coefficienti conservati. Sebbene gli schemi appartenenti a questa classe abbiano buone capacità di compressione, la loro complessità computazionale è spesso troppo elevata per i dispositivi indossabili.[17]
3. Le tecniche parametriche che utilizzano reti neurali [25], quantizzazione vettoriale [26], compressive sensing [18] e pattern matching [27] per ottenere dalle serie temporali conoscenze da usare per modellare la morfologia del segnale. Gli autocodificatori [28] sono stati proposti per approssimare modelli di biosegnali e hanno dimostrato di fornire eccellenti prestazioni di compressione e costi di calcolo molto inferiori rispetto agli altri algoritmi.

Alcuni metodi di compressione si basano sul grado di correlazione tra segmenti successivi (modelli) detta 'intersegment correlation' (es. quantizzazione vettoriale e autocodificatori), mentre altri sulla 'intra-segment correlation', ossia sulla similarità all'interno dello stesso segmento (es. PCA, LTC, trasformate coseno e wavelet). [17]

La tesi è focalizzata sui metodi basati sulla ‘intrasegment correlation’, valutati più adatti ad essere implementati su dispositivi indossabili. In particolare, nella tesi si concentra sui i metodi Turning Point [34], Compressed Sensing [18] e Fan [21].

3.3 I database utilizzati

Per la valutazione degli algoritmi sono utilizzati 3 database dalla piattaforma Physionet, che ha come fine la diffusione e il libero scambio di segnali come supporto agli studi e alla ricerca in campo biomedicale. [29]

Il database SHAREE (Smart Health for Assessing the Risk of Events via ECG) include registrazioni nominali elettrocardiografiche 24 ore su 24 di 139 pazienti ipertesi reclutati presso il Centro di ipertensione dell'Ospedale Universitario di Napoli Federico II.

I pazienti hanno età pari o superiore a 55 anni, in particolare sono 49 femmine e 90 maschi, di età compresa tra 7 e 72 anni. Le singole registrazioni hanno una durata di circa 24 ore ciascuna e contengono tre segnali ECG ciascuno campionato a 128 campioni al secondo con precisione a 8 bit. [30]

L'European ST-T Database è destinato a essere utilizzato per la valutazione di algoritmi per l'analisi delle variazioni delle onde ST e T. Questo database è composto da 90 estratti di registrazioni ECG di 79 soggetti, di cui 70 sono uomini di età compresa tra 30 e 84 anni e 8 sono donne di età compresa tra 55 e 71 anni. Ogni record ha una durata di due ore e contiene due segnali, ciascuno campionato a 250 Hz con una risoluzione di 12 bit su un intervallo di input nominale di 20 millivolt. I valori del campione sono stati riscaldati dopo la digitalizzazione con riferimento ai segnali di calibrazione nelle registrazioni analogiche originali, al fine di ottenere una scala uniforme di 200 unità ADC per millivolt per tutti i segnali. Ciascuno dei file di segnale è di dimensioni 5.400.000 byte. [31]

Il database St Petersburg INCART 12-lead Arrhythmia è composto da 75 registrazioni estratte da 32 record Holter. Ogni record dura 30 minuti e contiene 12 derivazioni standard, ciascuna campionata a 257 Hz con guadagni che variano da 250 a 1100 unità di conversione da analogico a digitale per millivolt. I documenti originali sono stati raccolti da 17 uomini e 15 donne, di età compresa tra 18 e 80 anni, sottoposti a test per la malattia coronarica. [32]

Nell'analisi degli algoritmi, sono considerati i dati registrati dalla derivazione V5 perché è l'unica unica presente in tutti i database.

3.4 Le metriche di valutazione

Per la valutazione della compressione vengono utilizzati i seguenti parametri.

Il Percentage Root mean square Difference (PRD) viene comunemente utilizzato per valutare le prestazioni dell' algoritmo di ricostruzione. Rappresenta le differenze tra l'ECG originale e il segnale ricostruito e il valore PRD più piccolo indica le migliori prestazioni di ricostruzione. PRD è definito come in (1):

$$PRD = 100 \frac{\|\hat{x} - x\|_2}{\|x\|_2} \quad (1)$$

dove \hat{x} è il segnale ricostruito, x il segnale ECG originale e $\|\cdot\|_2$ indica la norma quadratica.

È ampiamente riconosciuto che un segnale ricostruito può essere accettato solo se il PRD corrispondente $\leq 9\%$.

Le prestazioni di ricostruzione sono altamente correlate al Compression Rate (CR) che è definito come in (2):

$$CR = 100 \frac{M}{N} \quad (2)$$

dove M è la dimensione (bits) del segnale compresso e N è la dimensione (bits) del segnale originale.

Infine, il Quality Score (QS) (3) quantifica le prestazioni globali dell'algoritmo di compressione rappresentando il rapporto tra CR e PRD. [33]

$$QS = \frac{CR}{PRD} \quad (3)$$

4. IL METODO TURNING POINT

4.1 La descrizione dell'algoritmo

L'algoritmo Turning Point (TP) [34] si basa sul fatto che la frequenza di campionamento dei segnali ECG può essere maggiore della larghezza di banda del segnale. Ad esempio, un elettrocardiogramma può avere una larghezza di banda di 50 Hz ed essere campionato a 200 Hz per migliorare la visualizzazione delle ampiezze e le pendenze dei complessi QRS soggette a cambiamenti più repentini rispetto al resto del segnale. TP fornisce la possibilità di ridurre la frequenza di campionamento salvando selettivamente picchi e valli o punti di cambio pendenza del segnale.

L'algoritmo prende in considerazione tre punti dati alla volta (X_0 , X_1 e X_2). Memorizza il primo campione come punto di riferimento X_0 e mantiene soltanto uno dei campioni successivi X_1 o X_2 , a seconda di quale sia quello che rappresenti il cambio di pendenza del segnale originale, come si può vedere in Figura 4.1.a.

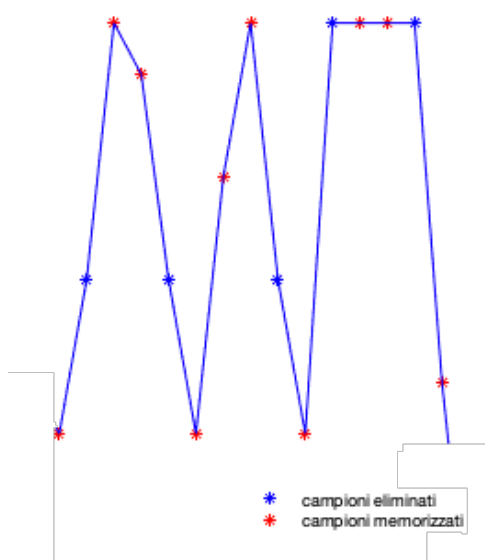


Figura 4.1.a Rappresentazione dell'algoritmo Turning Point

I passaggi di Turning Point sono:

- acquisizione del segnale ECG;
- selezione dei tre campioni X_0 , X_1 e X_2 e memorizzazione del campione X_0 ;

- c) verifica della condizione $(X_1 - X_0) * (X_2 - X_1) < 0$;
- d) se la condizione è corretta, memorizzazione del campione X_1 , altrimenti del campione X_2 ;
- e) ripetere i punti b), c), d)
- f) ricostruzione il segnale.

Il rapporto di compressione dell'algorithm di Turning Point è 2:1, ma, per ottenere una compressione maggiore, è possibile iterare più volte l'implementazione dell'algorithm sul segnale compresso, ponendo attenzione al valore del parametro PRD.

4.2 L'implementazione in MatLab

```

clear all
close all
clc

load data                                % a) ACQUISIZIONE SEGNALE
N = size(signal,1);                       % dimensione segnale originale
c_sig = signal;                            % valori iniziali
                                           % segnale compresso
c_N = size(c_sig,1);                       % dimensione segnale compresso
iter = 4;                                  % numero di iterazioni
k = 1;                                     % indice di iterazione
while k <= iter                            % ciclo di iterazione
    compr = [];                            % segnale compresso (variabile temporanea)
    idx = 1;                               % indice ciclo di scansione segnale
    while idx <= c_N - 2                  % ciclo di scansione segnale
        x0 = c_sig(idx,:);                % b) SELEZIONE CAMPIONI
                                           % primo campione
        compr = [compr; x0];              % b) MEMORIZZAZIONE x0 NEL SEGNALE COMPRESSO
        idx = idx + 1;                    % aggiornamento indice di scansione
        x1 = c_sig(idx,:);                % secondo campione
        idx = idx + 1;                    % aggiornamento indice di scansione
        x2 = c_sig(idx,:);                % terzo campione
        cond1 = x1(2) - x0(2);            % composizione condizione
        cond2 = x2(2) - x1(2);
        condizione = cond1 * cond2;
        if condizione < 0                 % c) VERIFICA CONDIZIONE
            compr = [compr; x1];          % d) MEMORIZZAZIONE x1 NEL SEGNALE COMPRESSO
        else
            compr = [compr; x2];          % d) MEMORIZZAZIONE x2 NEL SEGNALE COMPRESSO
        end % if (x1-x0) * (x2-x1) < 0
        idx = idx+1;                       % aggiornamento indice di scansione
    end % while i <= cN - 2
    c_sig = compr;                         % aggiornamento segnale compresso
    c_N = size(c_sig,1);                   % aggiornamento dimensione segnale compresso
    k = k + 1;                             % aggiornamento indice di iterazione
end % while k <= iter

                                           % e) RICOSTRUZIONE SEGNALE
r_signal = zeros(N,2);                     % segnale ricostruito
r_signal(:,1) = signal(:,1);
r_signal(:,2) = interp1(c_sig(:,1),c_sig(:,2),r_signal(:,1),'linear');

```

Figura 4.2.a Codice di implementazione dell'algoritmo Turning Point in MatLab

In Figura 4.2.a è riportato il codice utilizzato per implementare l'algoritmo Turning Point in MatLab.

Nella fase iniziale, si acquisisce la matrice *data*, che contiene al suo interno la matrice *signal*, composta da due colonne contenenti la numerazione dei campioni e i campioni.

Dopo aver calcolato la dimensione del segnale, si pongono le variabili del segnale compresso *c_sig* e della sua dimensione *c_N* rispettivamente uguali al segnale originale e alla sua dimensione.

Stabilito il numero delle iterazioni (ad esempio in Figura 4.2.a, *iter=4*), si definisce un ciclo *while* con cui eseguirle. Al suo interno si rappresenta temporaneamente il segnale compresso con la variabile *compr*, che viene aggiornata ad ogni iterazione e si implementa il cuore dell'algoritmo con un ulteriore ciclo *while*, con il quale vengono considerati tutti campioni del segnale.

Come precedentemente spiegato, l'algoritmo seleziona 3 campioni alla volta, memorizzandoli in sequenza nelle variabili *x0*, *x1* e *x2*.

Le variabili *cond1* e *cond2* memorizzano rispettivamente la differenza tra *x1* e *x0* e tra *x2* e *x1*, il cui prodotto definisce matematicamente la condizione che determina quale tra i campioni *x1* o *x2* deve essere inserito nel segnale compresso.

Una volta esaminato l'intero segnale, la variabile temporanea *compr* viene salvata nella variabile del segnale compresso *c_sig*, per essere utilizzata per un'ulteriore iterazione o per la ricostruzione del segnale originale, e viene aggiornato il valore della sua dimensione nella variabile *c_N*.

Quando tutte le iterazioni previste sono eseguite, si ricostruisce in *r_signal* il segnale originale a partire dal segnale compresso *c_sig* con un'interpolazione lineare.

4.3 I risultati dell'implementazione

L'algoritmo è stato implementato sui database presenti nel paragrafo 3.3 e i suoi risultati valutati con i parametri stabiliti nel paragrafo 3.4.

Per stabilirne il numero adeguato da eseguire, si sono valutate da 2 a 5 iterazioni e le rispettive medie dei parametri per ciascun database (Figure 4.3.a, 4.3.b, 4.3.c)

SHAREE			
numero iterazioni	CR _{media}	PRD _{media}	QS _{media}
2	44.4 %	1.691 %	30.328
3	29.6 %	2.496 %	14.143
4	19.7 %	3.738 %	6.260
5	13.1 %	5.045 %	3.053

Figura 4.3.a Database SHAREE: numero iterazioni vs parametri di valutazione

European ST-T			
numero iterazioni	CR _{media}	PRD _{media}	QS _{media}
2	44.4 %	4.275 %	55.655
3	29.6 %	7.0694 %	19.995
4	19.7 %	11.188 %	8.652
5	13.1 %	19.707 %	3.675

Figura 4.3.b Database European ST-T: numero iterazioni vs parametri di valutazione

St Petersburg INCART 12-lead Arrhythmia			
numero iterazioni	CR _{media}	PRD _{media}	QS _{media}
2	44.4 %	2.941 %	36.470
3	29.6 %	4.671 %	15.286
4	19.7 %	7.822 %	6.698
5	13.1 %	14.281 %	2.691

Figura 4.3.c Database St Peterburg: numero iterazioni vs parametri di valutazione

Per tutti i database, risulta che eseguendo 2 iterazioni dei campioni si ottiene un buon rapporto tra il CR (44.4%) e i diversi PRD. In particolare, sul database SHAREE si ottiene PRD medio pari a 1.691% e QS medio pari a 30.328, sul database European ST-T il PRD medio vale 4.275% e il QS medio 55.655 e infine sul database St Peterburg i valori medi di PRD e di QS sono rispettivamente 2.941% e 36.470.

Nelle pagine seguenti sono riportati alcuni grafici di casi esemplificativi dell'applicazione dell'algoritmo Turning Point ai diversi database.

In Figura 4.3.d e Figura 4.3.e si riportano due esempi di ottima ricostruzione del segnale originale dal segnale compresso.

In Figura 4.3.f la ricostruzione del segnale originale è buona, ma manca di precisione nei dettagli.

In Figura 4.3.g tra il campione 600 e il campione 1250 il segnale ricostruito riporta con imprecisione i minimi locali del segnale.

In Figura 4.3.h e in Figura 4.3.i i segnali originali e i segnali ricostruiti non corrispondono, pur mantenendo andamenti simili.

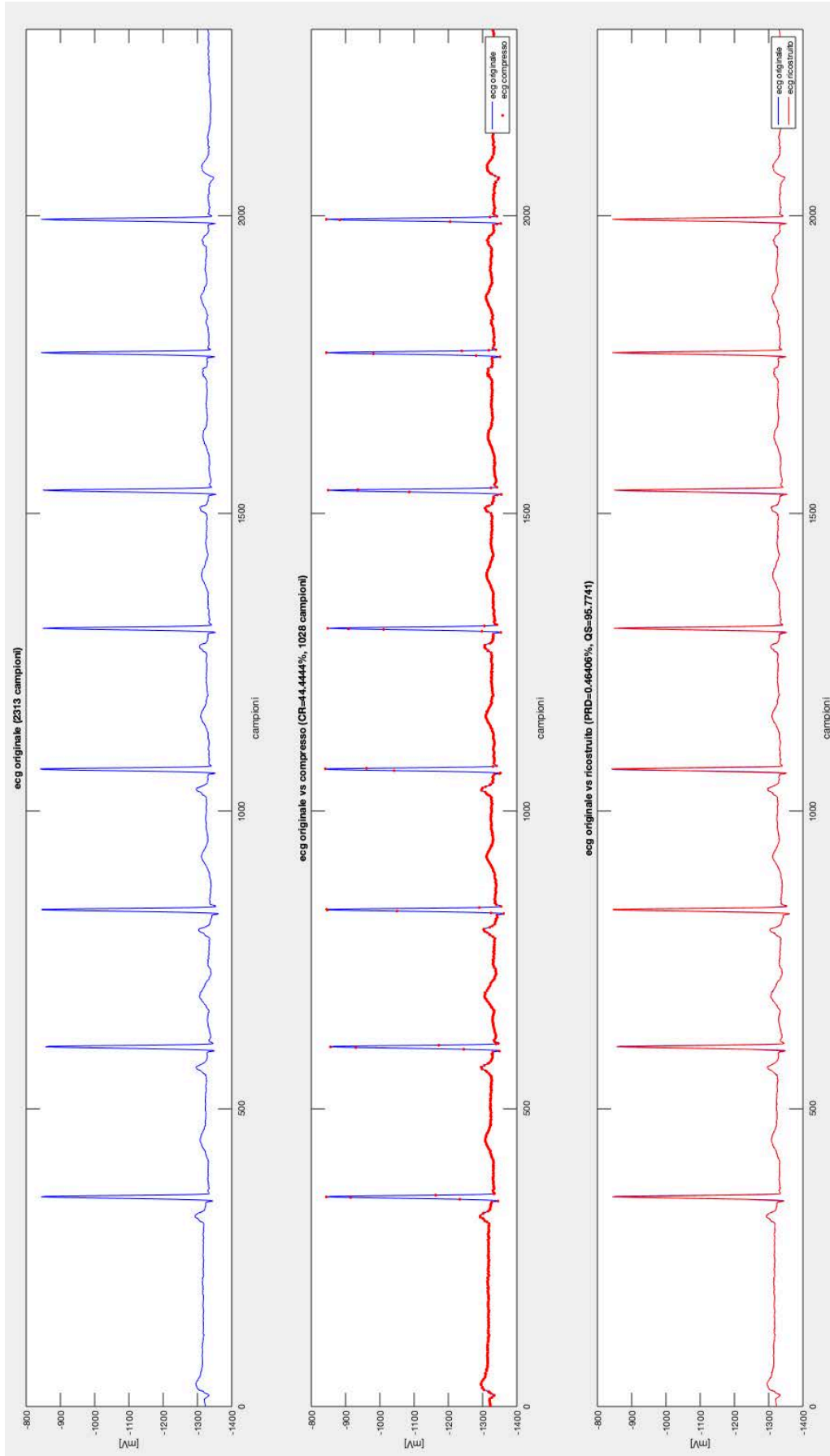


Figura 4.3.d Turning Point è applicato al segnale I31m del database St Petersburg INCART 12-lead Arrhythmia. C'è perfetta corrispondenza tra il segnale originale e quello ricostruito.

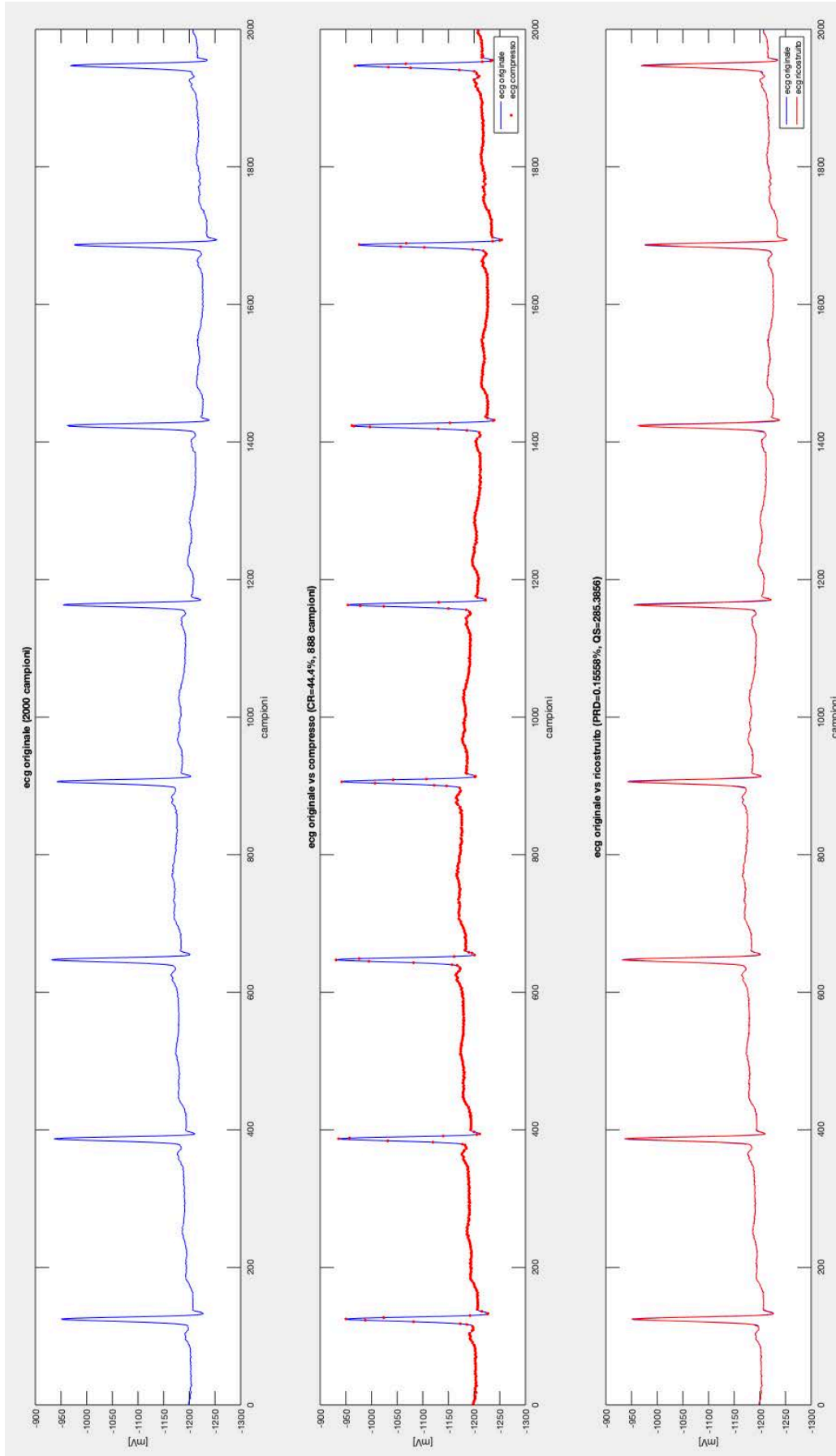


Figura 4.3.e Turning Point è applicato al segnale e404m del database European ST-T. C'è perfetta corrispondenza tra il segnale originale e quello ricostruito.

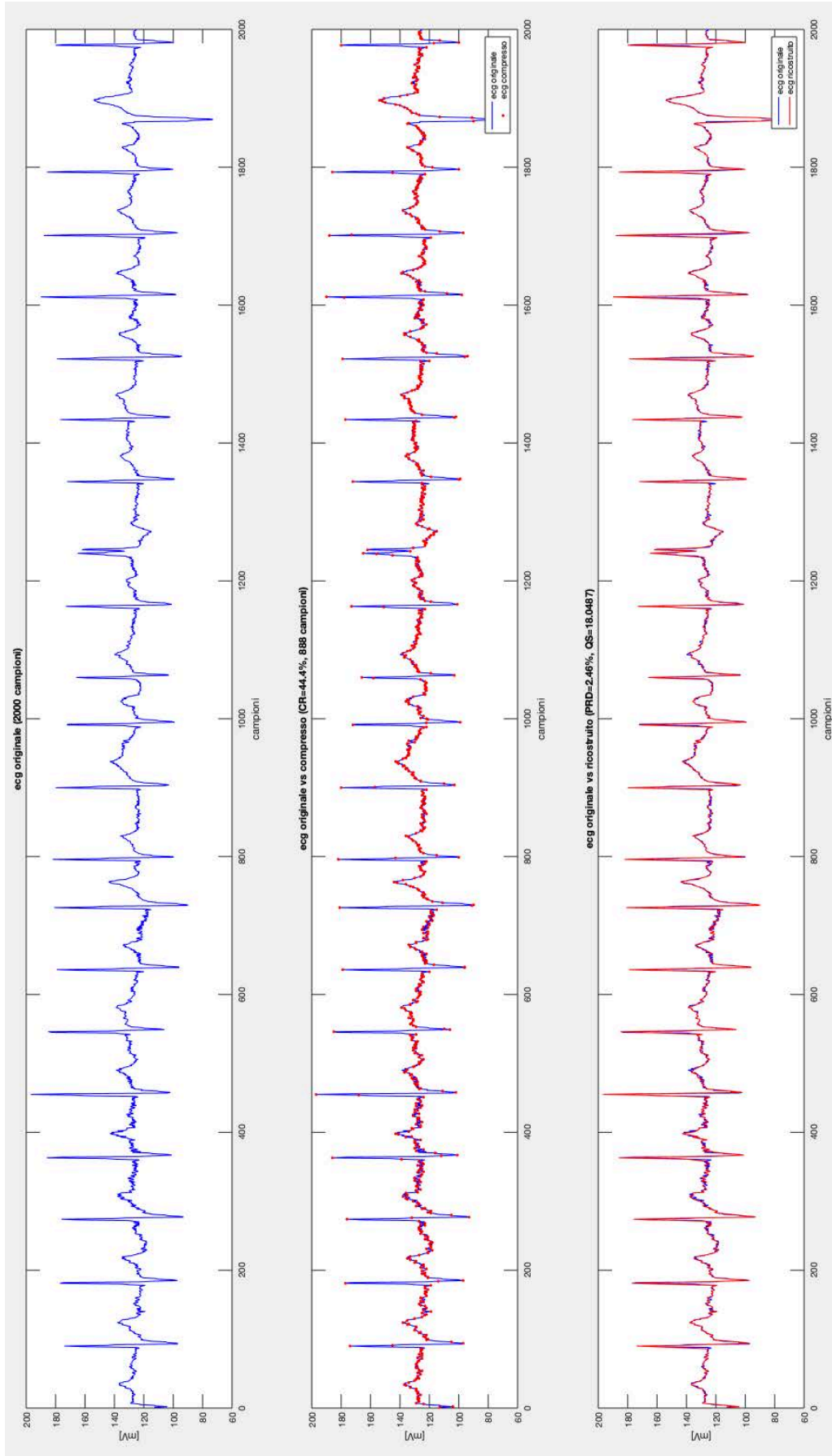


Figura 4.3.f Turning Point è applicato al segnale 02033m del database SHAREE. Il segnale ricostruito segue quello originale senza replicarne i valori esatti.

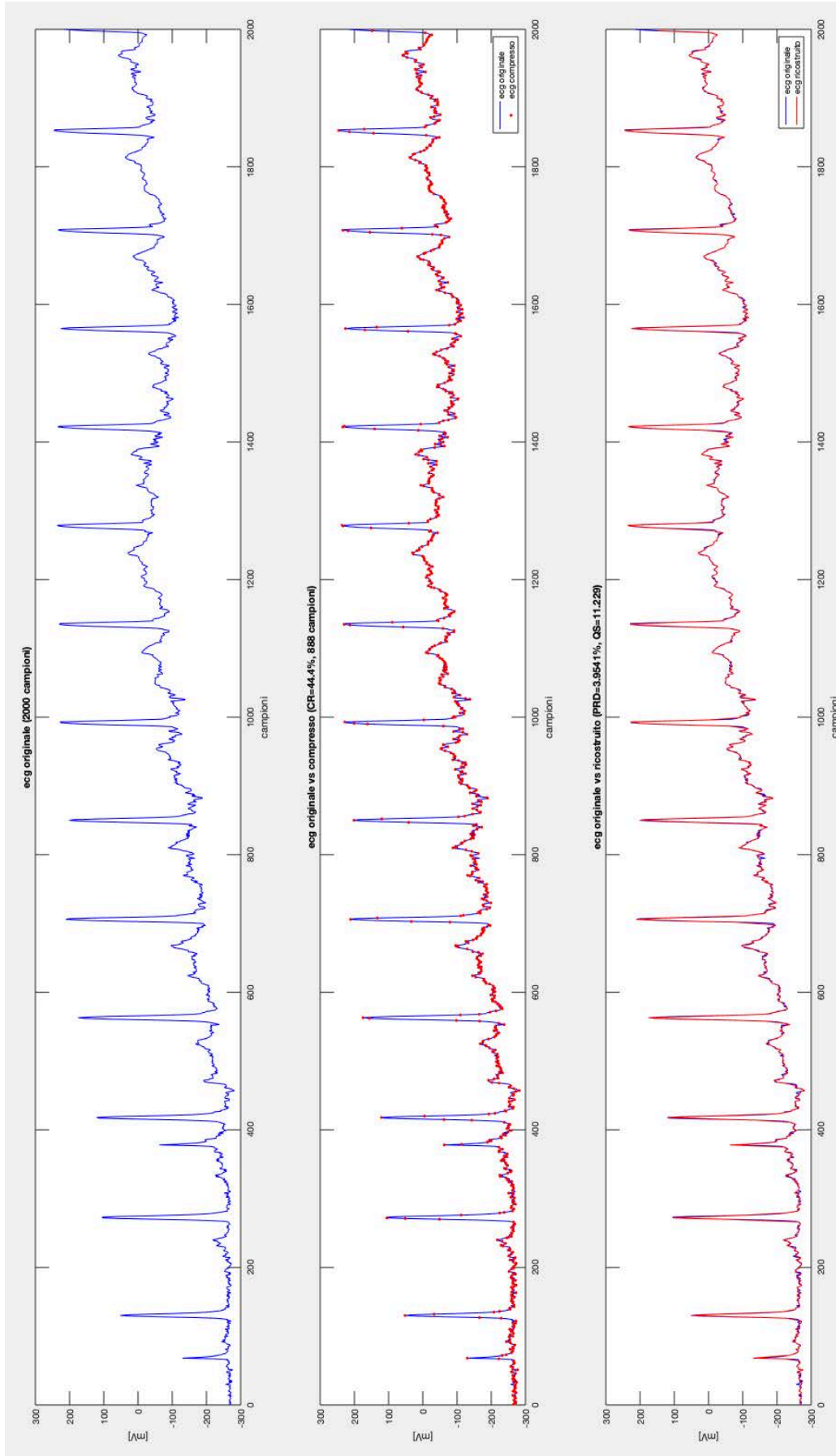


Figura 4.3.g Turning Point è applicato al segnale 02013m del database European ST-T. Il segnale ricostruito segue quello originale senza replicarne i valori esatti.

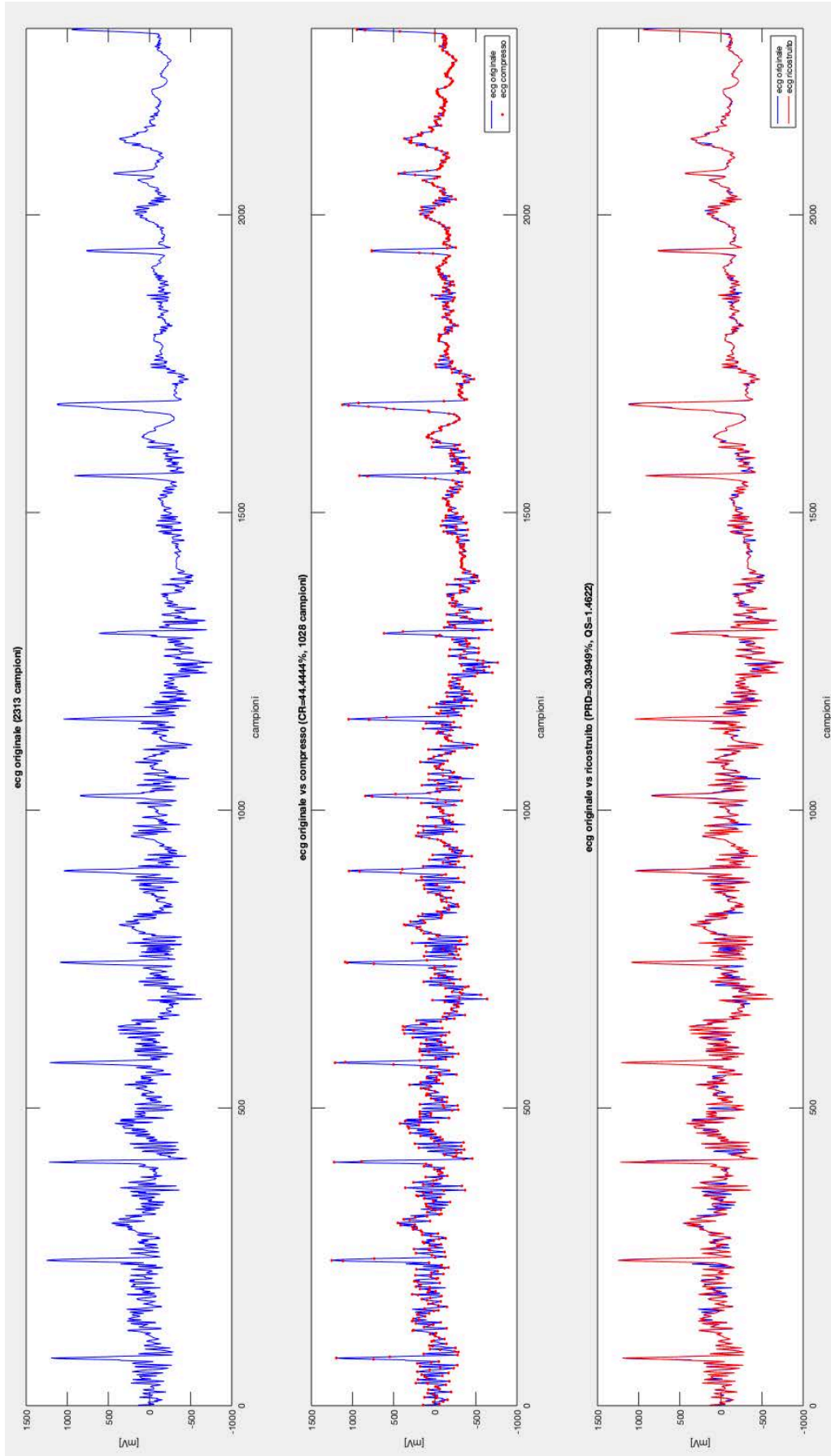


Figura 4.3.h Turning Point è applicato al segnale I08m del database St Petersburg INCART 12-lead Arrhythmia. Il segnale ricostruito non segue esattamente il tracciato originale.

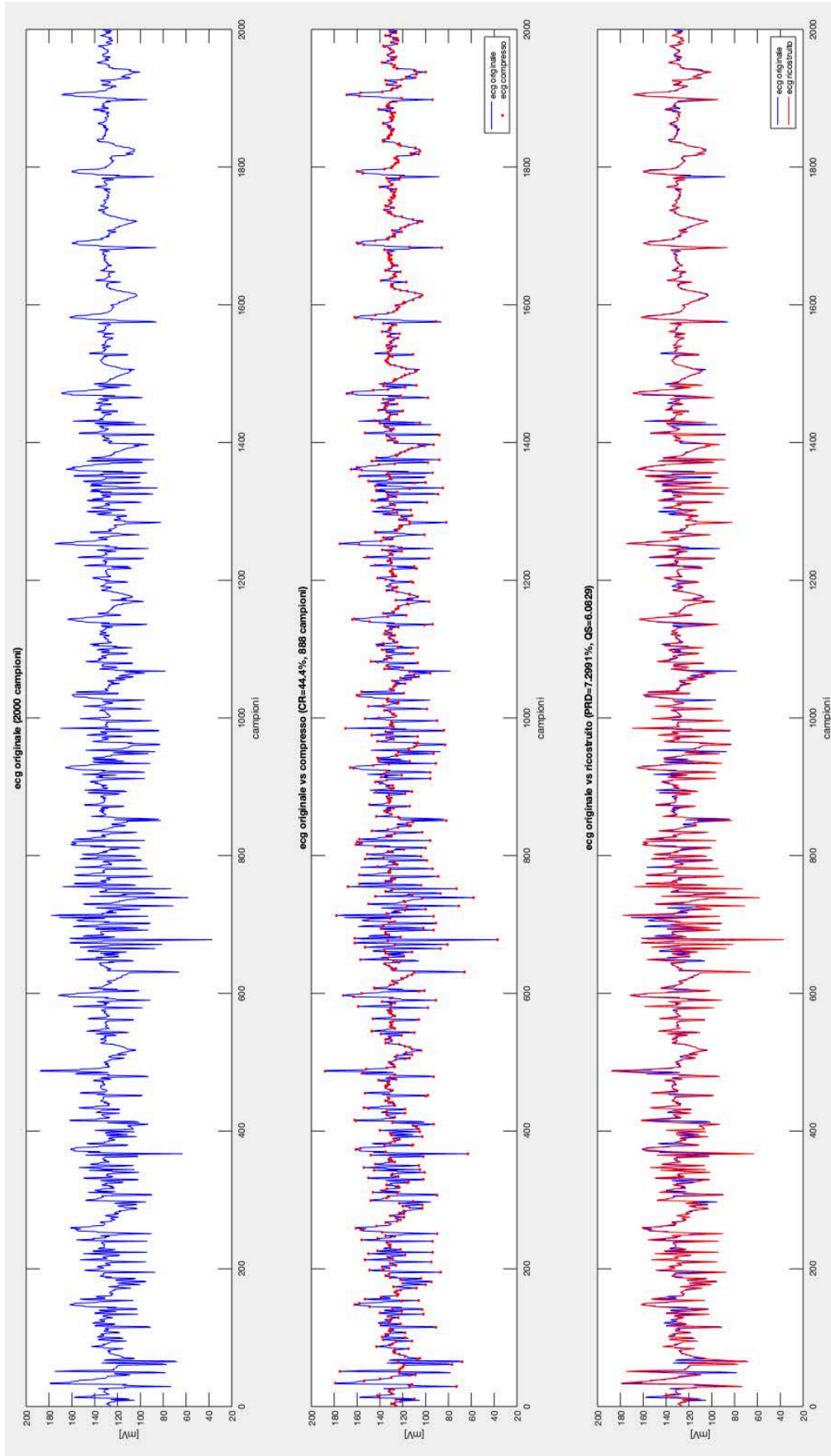


Figura 4.3.i Turning Point è applicato al segnale 02019m del database SHAREE.
 Il segnale ricostruito non segue esattamente il tracciato originale.

5. IL METODO COMPRESSED SENSING

5.1 Descrizione dell'algoritmo

Compressed Sensing [35] è un metodo che permette la compressione di un segnale unidimensionale $x \in \mathbb{R}^{N \times 1}$ attraverso la moltiplicazione per una matrice casuale $\Phi \in \mathbb{R}^{M \times N}$, con $M \ll N$ e rango pari a M

$$y = \Phi x \quad (1)$$

dove y è il segnale compresso, Φ è la matrice di rilevamento (*sensing matrix*) e x il segnale originale.

Il segnale compresso e la matrice di rilevamento vengono utilizzati per la ricostruzione del segnale originale. Il successo dell'algoritmo si basa sul fatto che il segnale x sia sparso, ossia gran parte dei suoi campioni siano zero. Quando questa ipotesi non è verificata, si può utilizzare una matrice dizionario $D \in \mathbb{R}^{N \times M}$, che permetta di esprimere il segnale originale come $x = Dz$, dove z è un segnale sparso.

Quindi (1) può essere riscritta come

$$y = \Phi D z \quad (2)$$

A questo punto, l'algoritmo CS può ricostruire z a partire da y e ΦD e in seguito il segnale originale x .

Quando è utilizzato in un sistema di telemonitoraggio, i segnali sono compressi nei sensori secondo quanto espresso in (1). La ricostruzione del segnale originale è eseguita in remoto da un computer utilizzando la formula (2), nota la matrice Φ e definita dall'utente la matrice D .

Per comprimere il segnale, si moltiplica a sinistra per una matrice $M \times N$ ($M \ll N$) sparsa e binaria.

Riguardo la ricostruzione del segnale, l'algoritmo Bound Optimization Based Sparse Block Bayesian Learning (BO_BSBL) pone come condizione di applicazione che il

segnale sia partizionato in una concatenazione di blocchi non sovrapposti e che alcuni blocchi siano diversi da zero. Tuttavia, risulta che tale suddivisione definita dall'utente non debba essere coerente con una reale partizione del segnale.

5.2 L'implementazione in MatLab

```
clear all
close all
clc

load(data) % acquisizione segnale
N = size(X,1); % dimensione segnale

M = N/2; % numero campioni segnale compresso

PHI = randi([0 1],M,N); % matrice di rilevamento
Y = PHI*X; % segnale compresso (1)

THETA = zeros(M,N); % matrice dizionario
for k = 1:M
    THETA(k,:) = dct(PHI(k,:));
end

X_tmp=BSBL_BO(THETA,Y); % ricostruzione del segnale
X_hat=(idct(X_tmp.x))'; % segnale ricostruito
```

Figura 5.1.a Codice di implementazione dell' algoritmo Compressed Sensing in MatLab

In Figura 5.2.a è riportato il codice utilizzato per implementare l'algoritmo Compressed Sensing in MatLab.

Si inizia con l'acquisizione dei dati e la determinazione del valore della variabile M , pari al numero di campioni da cui è costituito il segnale compresso.

Si prosegue con la definizione della matrice di rilevamento PHI di dimensioni $M \times N$, dove M rappresenta la dimensione del segnale compresso e N quella del segnale originale. La matrice è una matrice casuale binaria sparsa, costruita con la funzione *randi*.

Si calcola il segnale compresso Y moltiplicando a sinistra il segnale originale X per la matrice PHI .

Il dizionario per la ricostruzione di X è rappresentato dalla matrice $THETA$, che contiene nelle righe la trasformata coseno discreta delle righe della matrice PHI .

Per procedere la ricostruzione del segnale originale, $THETA$ e Y sono passati come argomenti a *BSBL_BO* (si veda Appendice) che restituisce la struttura X_tmp , all'interno della quale la variabile x contiene la derivata coseno del segnale originale.

Infine, il segnale ricostruito X_hat si ottiene con la trasformata inversa coseno della variabile x .

5.3 Risultati dell'implementazione

L'algoritmo è stato implementato sui database presenti nel paragrafo 3.3 e i suoi risultati valutati con i parametri stabiliti nel paragrafo 3.4.

Per determinare il numero di campioni del segnale compresso, si sono valutate i casi 30%, 50%, 60% e 70% dei campioni del segnale originale e le rispettive medie dei parametri per ciascun database (Figure 5.3.a, 5.3.b, 5.3.c)

SHAREE			
campioni considerati	CR _{media}	PRD _{media}	QS _{media}
30 %	30 %	3.356%	9.972
50 %	50 %	2.184 %	25.803
60 %	60 %	1.772 %	38.220
70 %	70 %	1.416 %	55.836

Figura 5.3.a Database SHAREE: percentuale di campioni vs parametri di valutazione

European ST-T			
campioni considerati	CR _{media}	PRD _{media}	QS _{media}
30 %	30 %	11.508 %	10.641
50 %	50 %	4.054 %	67.615
60 %	60 %	2.745 %	138.234
70 %	70 %	1.924 %	249.336

Figura 5.3.b Database European ST-T: percentuale di campioni vs parametri di valutazione

St Petersburg INCART 12-lead Arrhythmia			
campioni considerati	CR _{media}	PRD _{media}	QS _{media}
30 %	30 %	6.478 %	12.430
50 %	50 %	3.043 %	42.246
60 %	60 %	1.832 %	90.160
70 %	70 %	1.217 %	140.395

Figura 5.3.c Database St Peterburg: percentuale di campioni vs parametri di valutazione

Per tutti i database, risulta che considerando il 60% dei campioni si ottiene un buon rapporto tra il CR (60 %) e i diversi PRD. In particolare, sul database SHAREE si ottiene PRD medio pari a 1.772% e QS medio pari a 38.220, sul database European ST-T il PRD medio vale 2.745% e il QS medio 138.234 e infine sul database St Peterburg i valori medi di PRD e di QS sono rispettivamente 1.832% e 90.160.

Nelle pagine seguenti sono riportati alcuni grafici di casi esemplificativi dell'applicazione dell'algoritmo Compressed Sensing ai diversi database.

In Figura 5.3.d e in Figura 5.3.e si riportano due esempi di ottima ricostruzione del segnale originale dal segnale compresso.

In Figura 5.3.f e in Figura 5.3.g i segnali ricostruiti seguono gli originali, ma risultano poco precisi nei punti di massimo e minimo locali.

In Figura 5.3.h e in Figura 5.3.i i segnali originali e i segnali ricostruiti non corrispondono, pur mantenendo andamenti simili.

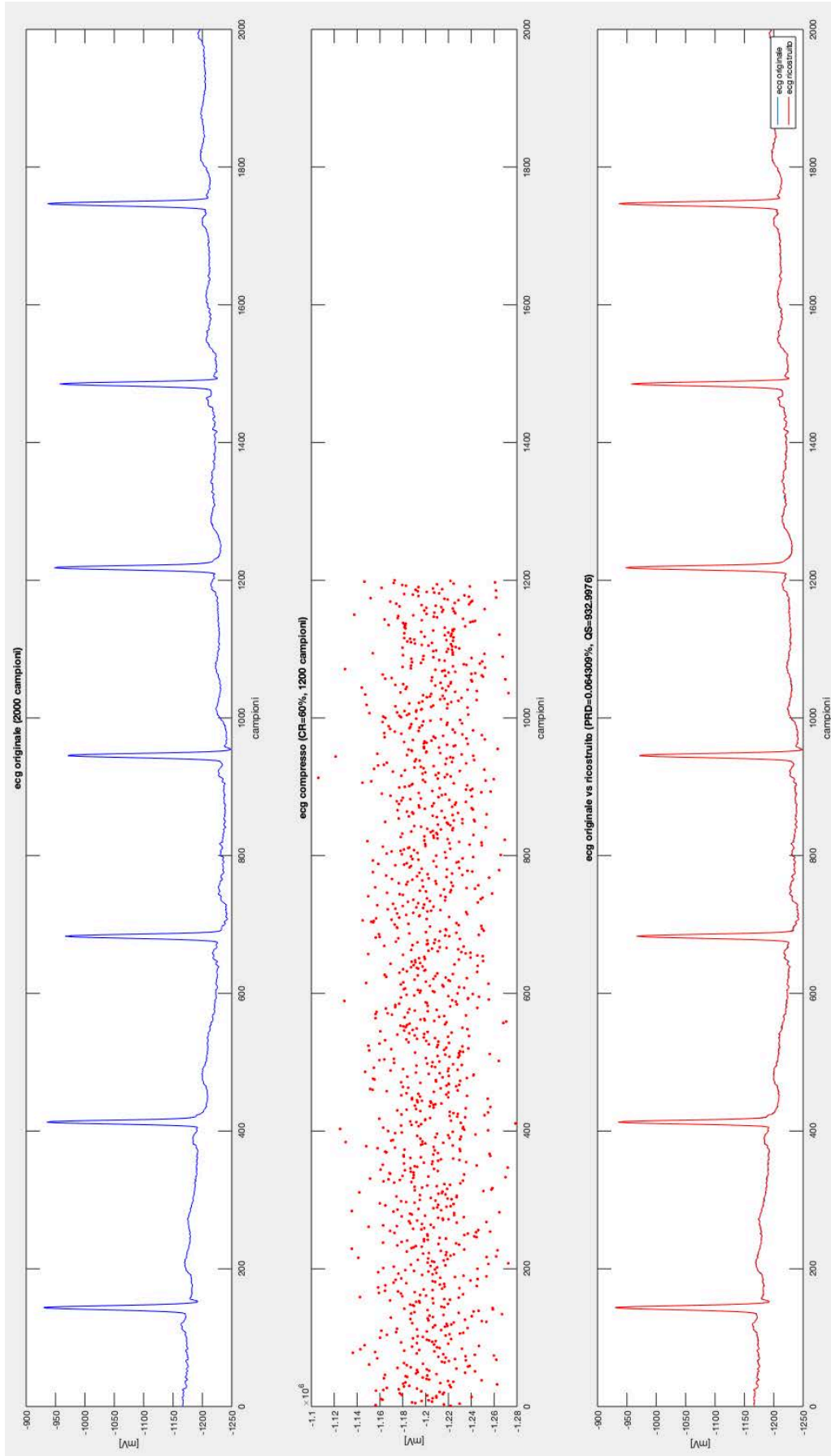


Figura 5.3.d Compressed Sensing è applicato al segnale e0615m del database European ST-T. C'è perfetta corrispondenza tra il segnale originale e quello ricostruito.

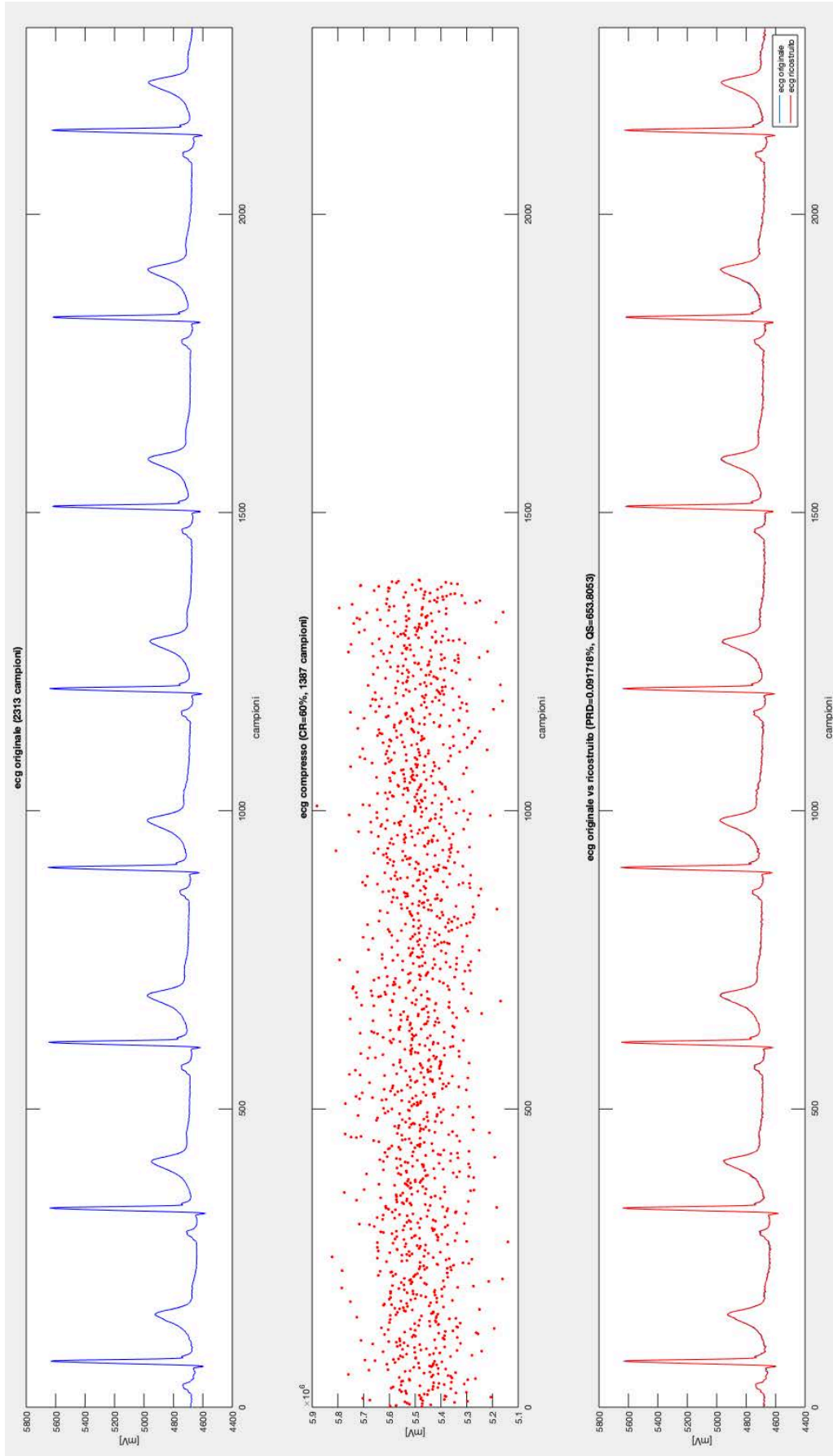


Figura 5.3.e Compressed Sensing è applicato al segnale I14m del database St Petersburg INCART 12-lead Arrhythmia. C'è perfetta corrispondenza tra il segnale originale e quello ricostruito.

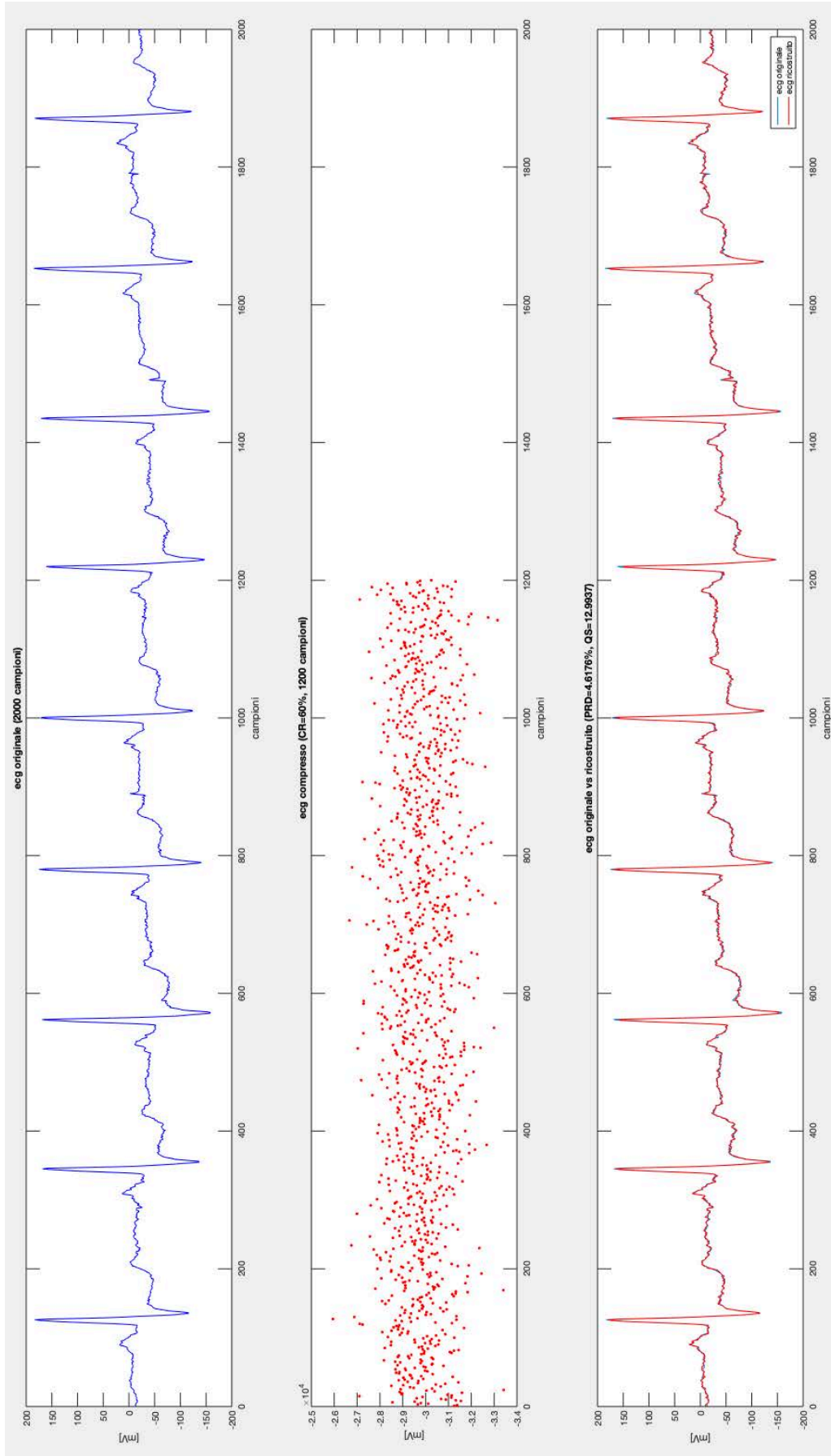


Figura 5.3.f Compressed Sensing è applicato al segnale 02063m del database SHAREE. Il segnale ricostruito segue quello originale senza replicarne i valori esatti.

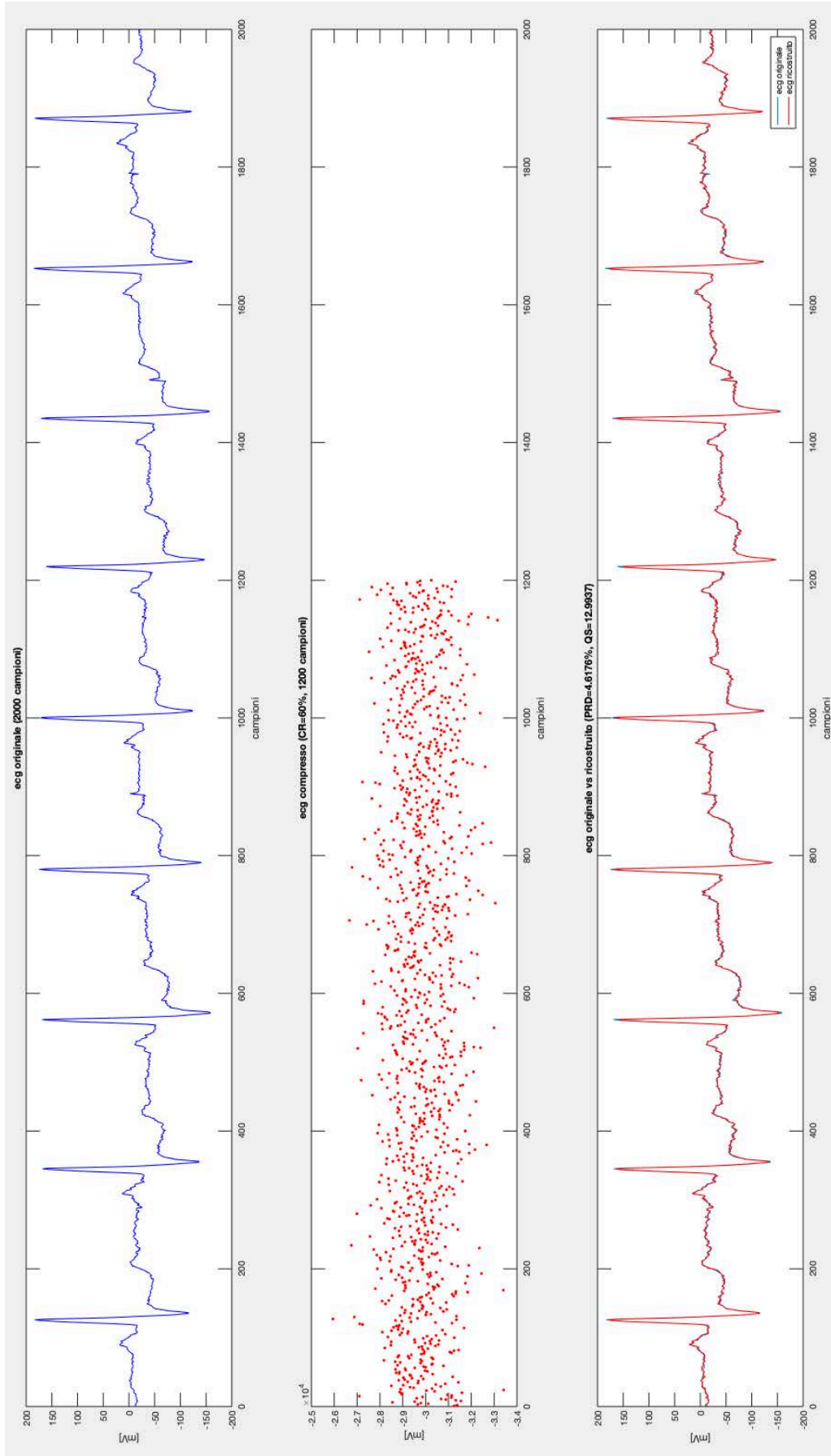


Figura 5.3.g Compressed Sensing è applicato al segnale e0404m del database European ST-T. Il segnale ricostruito segue quello originale senza replicarne i valori esatti.

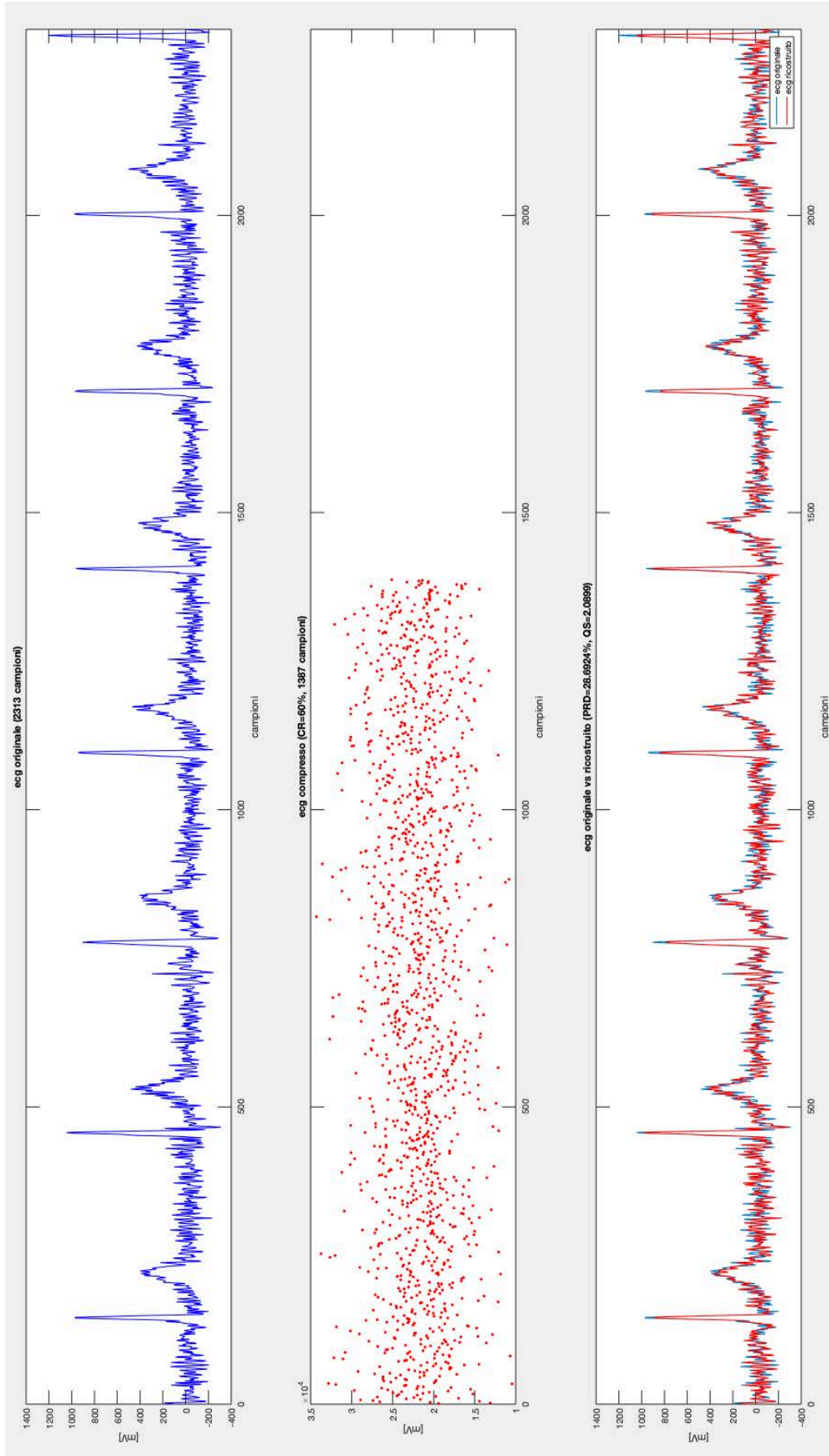


Figura 5.3.h Compressed Sensing è applicato al segnale 108m del database St Petersburg INCART 12-lead Arrhythmia. Il segnale ricostruito non segue esattamente il tracciato originale.

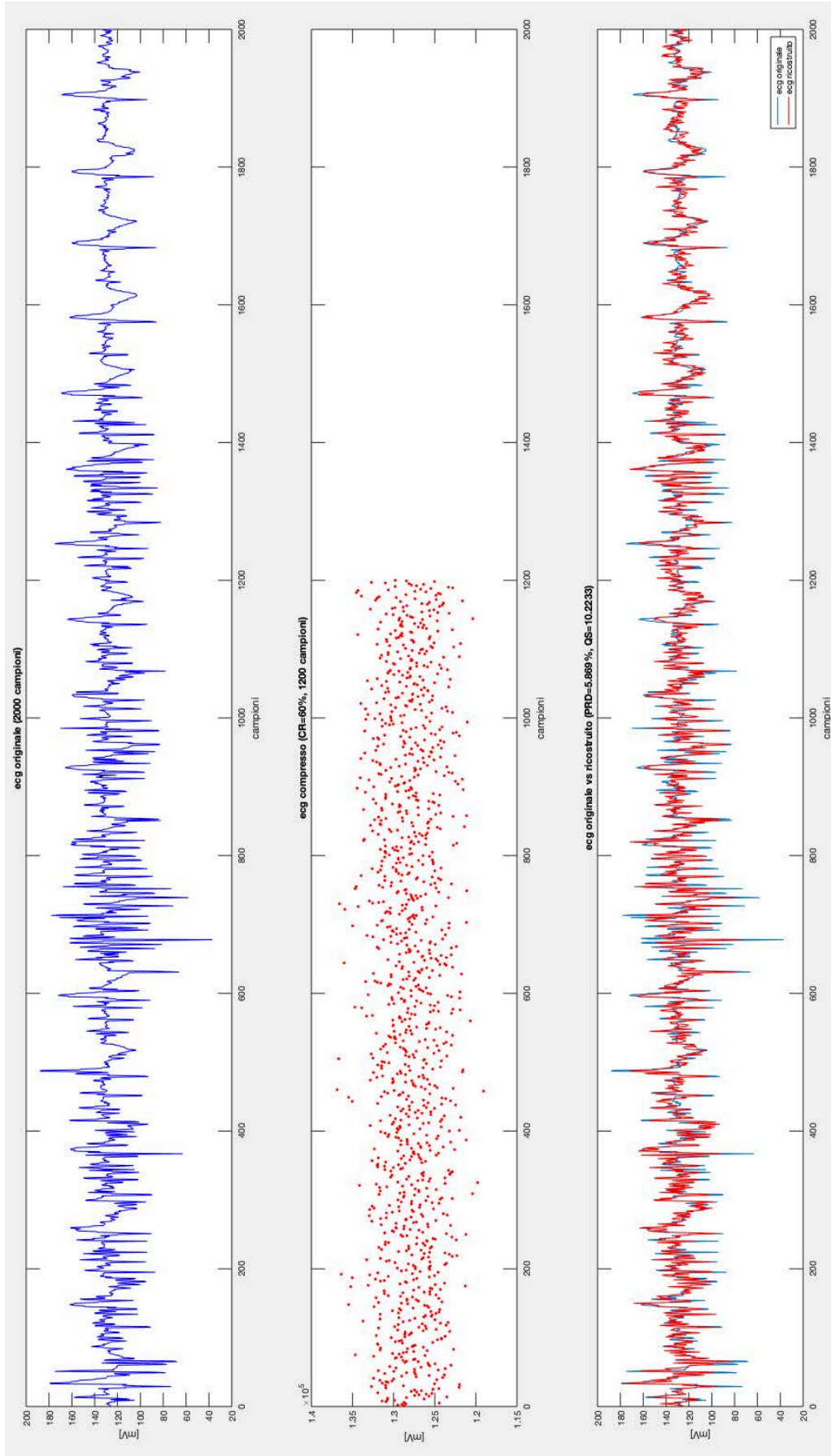


Figura 5.3.i Compressed Sensing è applicato al segnale 02013m del database SHAREE. Il segnale ricostruito non segue esattamente il tracciato originale.

6. IL METODO FAN

6.1 Descrizione dell'algoritmo

L'algoritmo Fan [21] è una tecnica di campionamento non uniforme, caratteristica che permette di ottenere una fedele corrispondenza al segnale originale.

La finalità è quella di tracciare la linea retta più lunga possibile tra il campione iniziale e il campione finale, in modo tale che l'errore nella ricostruzione dei campioni intermedi sia inferiore a ϵ , errore massimo specificato, che, per ottimizzare le prestazioni dell'algoritmo al fine di ottenere un elevato CR e una buona efficienza energetica, deve avere un valore compreso tra 0.1% e 1% del range dinamico del segnale.

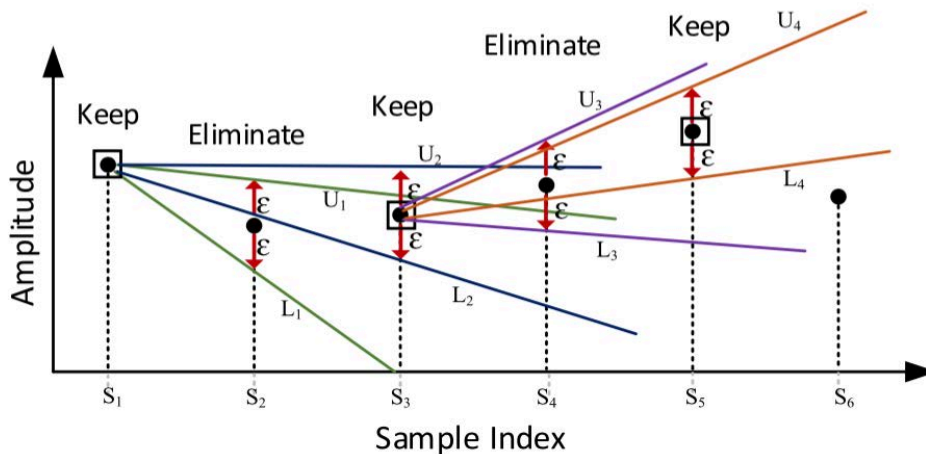


Figura 6.1.a Rappresentazione dell'algoritmo FAN (tratta da [25])

Come rappresentato in figura 6.1.a, si memorizza un primo campione, S_1 , che funge da punto di origine iniziale. Si considera, poi, il campione successivo S_2 e si determinano due diversi limiti di ampiezza $S_2 \pm \epsilon$, per i quali passano le due rette U_1, L_1 disegnate rispettivamente tra il punto di origine e il limite superiore ($S_2 + \epsilon$) e tra il punto di origine e il limite inferiore ($S_2 - \epsilon$). Se il valore del prossimo campione (S_3) è compreso tra i limiti creati dalle rette (U_1, L_1), si elimina il campione S_2 e si memorizza il campione S_3 .

Il processo continua con il calcolo dei limiti di errore superiore e inferiore del campione S_3 ($S_3 \pm \epsilon$) e la rappresentazione delle due nuove rette U_2, L_2 passanti per il punto S_1 e i punti $S_3 + \epsilon$ e $S_3 - \epsilon$.

Si confrontano, poi, queste nuove rette con le precedenti (U_1, L_1) e si mantengono quelle più convergenti, che, in Figura 6.1.a, sono U_1 e L_2 .

Questo viene fatto considerando i triangoli retti ABC e ABD in Figura 6.1.b, come indicato nelle seguenti espressioni (1),(2),(3),(4)

$$BC = AB \frac{EF}{AE} = \frac{len + 1}{len} (S_{N-1} + \varepsilon - S_1) \quad (1)$$

$$S_N UB = S_1 + BC \quad (2)$$

$$BD = AB \frac{EG}{AE} = \frac{len + 1}{len} (S_{N-1} - \varepsilon - S_1) \quad (3)$$

$$S_N LB = S_1 + BD \quad (4)$$

Se il valore di S_N non rientra nei limiti superiore e inferiore $S_N UB$ e $S_N LB$, il campione corrente S_{N-1} viene scelto come campione di origine successivo, altrimenti viene eliminato. I limiti vengono verificati ricorsivamente rispetto ai valori passati per selezionare le rette con pendenze più convergenti per il campione successivo.

6.2 L'implementazione in MatLab

```
clear all
close all
clc

load(data) % acquisizione segnale
N = size(x,1); % dimensione segnale
range = max(x) - min(x); % range dinamico segnale
perc = 0.1; % percentuale errore
e = range*perc/100; % errore epsilon
S_orig = [1,x(1)]; % origine iniziale (indice, valore)
compr_x = zeros(N,2); % matrice segnale

for s = 2:N-1
    SN_1 = [s,x(s)]; % campione corrente (indice, valore)
    len = SN_1(1)-S_orig(1); % distanza tra origine e campione corrente
    [SnLB,SnUB] = bounds(S_orig,SN_1,len,e); % limiti
    SN = [s+1,x(s+1)]; % campione successivo (indice, valore)
    if SN(2) >= SnLB && SN(2) <= SnUB % se SN è entro limiti...
        compr_x(s+1,:) = SN; %...si inserisce il campione successivo
        compr_x(s,:) = [0,0]; %...si elimina campione corrente
        len = SN(1) - S_orig(1); % distanza tra il nuovo campione e l'origine
        [L,U] = bounds(S_orig,SN,len,e); % nuovi limiti
        if L > SnLB % se il nuovo limite inferiore è più convergente del precedente...
            SnLB = L; %...viene considerato come limite inferiore
        end %if L>SnLB
        if U < SnUB % se il nuovo limite superiore è più convergente del precedente
            SnUB = U; %...viene considerato come limite superiore
        end %if U>SnUB
    else %... se SN non è entro i limiti
        compr_x(s,:) = SN_1; %...si inserisce campione corrente nel segnale compresso...
        S_orig = SN_1; %... e si considera come nuova origine
    end %if x(n+1)>=SnLB && x(n+1)<=SnUB
end % for s

sign_ricost = zeros(1,N); %#ok<PREALL> % segnale ricostruito
ti = linspace(1,N,N);
sign_ricost = interp1(compr(:,1),compr(:,2),ti,'linear');
```

Figura 6.2.a Codice di implementazione dell'algoritmo Fan in MatLab

In Figura 6.2.a è riportato il codice utilizzato per implementare l'algoritmo Fan in MatLab.

Dopo aver acquisito il segnale e averne misurato la dimensione, si definiscono il range dinamico del segnale *range*, come la differenza tra il massimo e il minimo del segnale, la percentuale di errore consentita *e*, il campione origine iniziale *S_orig*, memorizzato in un vettore contenente posizione e valore, e infine il segnale compresso *compr_x*, matrice che contiene nella prima colonna le posizioni e nella seconda i valori dei campioni.

Segue un ciclo *for* per considerare tutti campioni del segnale e al suo interno vengono memorizzati in *SN_I* il campione corrente, in *len* la distanza tra il campione corrente e l'attuale origine, in *SnLB* e *SnUB* rispettivamente il limite inferiore e superiore per la selezione dei campioni. Questi ultimi vengono calcolati tramite la funzione *bounds* (Figura 6.2.b), sfruttando le formule (1) e (3) del paragrafo 6.1.

```
function [SnLB,SnUB]=bounds(orig,sample,len,eps)
SnUB=orig(2)+(len+1)/(len)*(sample(2)+eps-orig(2));
SnLB=orig(2)+(len+1)/(len)*(sample(2)-eps-orig(2));
```

Figura 6.2.b Funzione bounds

Viene, quindi, considerato il campione successivo *SN* e se ha un valore entro i limiti *SnLB* e *SnUB*, viene memorizzato nel segnale compresso e si procede con l'eliminazione del campione *SN_I*, l'aggiornamento della distanza tra l'origine *S_orig* e il campione appena inserito *SN* e il calcolo dei nuovi limiti *U* e *L*, che sono considerati solo nel caso siano più convergenti dei precedenti (*if L > SnLB, if U < SnUB*).

Se il valore di *SN* non rientra tra i limiti (*SnLB* e *SnUB*), il campione corrente *SN_I* viene inserito nel segnale compresso e considerato come nuova origine.

Quando il ciclo *for* è concluso e si sono considerati tutti campioni del segnale, si ricostruisce in *sign_ricost* il segnale originale a partire dal segnale compresso *compr_x* con un'interpolazione lineare.

6.3 Risultati dell'implementazione

L'algoritmo è stato implementato sui database presenti nel paragrafo 3.3 e i suoi risultati valutati con i parametri stabiliti nel paragrafo 3.4.

Per determinare la percentuale di errore da considerare, si sono valutati i casi 0.1%, 0.5%, 0.7% e 1% e le rispettive medie dei parametri per ciascun database (Figure 6.3.a, 6.3.b, 6.3.c)

SHAREE			
errore percentuale	CR _{media}	PRD _{media}	QS _{media}
0.1 %	83 %	0.101 %	1.554e+03
0.5 %	81 %	0.121 %	1.413e+03
0.7 %	76 %	0.160 %	1.118 e+03
1 %	66 %	0.232 %	617.607

Figura 6.3.a Database SHAREE: percentuale di errore vs parametri di valutazione

European ST-T			
errore percentuale	CR _{media}	PRD _{media}	QS _{media}
0.1 %	87 %	0.189 %	3.922e+03
0.5 %	49 %	0.831 %	247.765
0.7 %	39 %	1.527 %	115.476
1 %	28 %	3.353 %	48.544

Figura 6.3.b Database European ST-T: percentuale di errore vs parametri di valutazione

St Petersburg INCART 12-lead Arrhythmia			
errore percentuale	CR _{media}	PRD _{media}	QS _{media}
0.1 %	71 %	0.158 %	1.684e+03
0.5 %	31 %	3.649 %	88.344
0.7 %	24 %	7.5375 %	39.498
1 %	17 %	16.614 %	15.171

Figura 6.3.c Database St Peterburg: percentuale di errore vs parametri di valutazione

Per tutti i database, risulta che tollerando un errore di 0.1% del range dinamico si ottiene un buon rapporto tra i diversi CR e PRD. In particolare, sul database SHAREE (CR=83%) si ottiene PRD medio pari a 5.298% e QS medio pari a $1.554e+03$, sul database European ST-T (CR=87%) il PRD medio vale 0.189% e il QS medio $3.9322e+03$ e infine sul database St Peterburg (CR=71%) i valori medi di PRD e di QS sono rispettivamente 0.158% e $1.684e+03$.

Nelle pagine seguenti sono riportati alcuni grafici di casi esemplificativi dell'applicazione dell'algorithm Fan ai diversi database.

In Figura 6.3.d c'è un'ottima corrispondenza tra segnale ricostruito e segnale originale.

In Figura 6.3.e il segnale ricostruito e originale sono addirittura indistinguibili.

In Figura 6.3.f e in Figura 6.3.g i segnali ricostruiti e originali hanno un'ottima corrispondenza eccetto all'estremo sinistro.

In Figura 6.3.h la ricostruzione del segnale rimane molto buona, ma meno precisa rispetto ad altri segnali del database.

In Figura 6.3.i il segnale ricostruito e il segnale ricostruito sono identici, eccetto all'estremo sinistro.

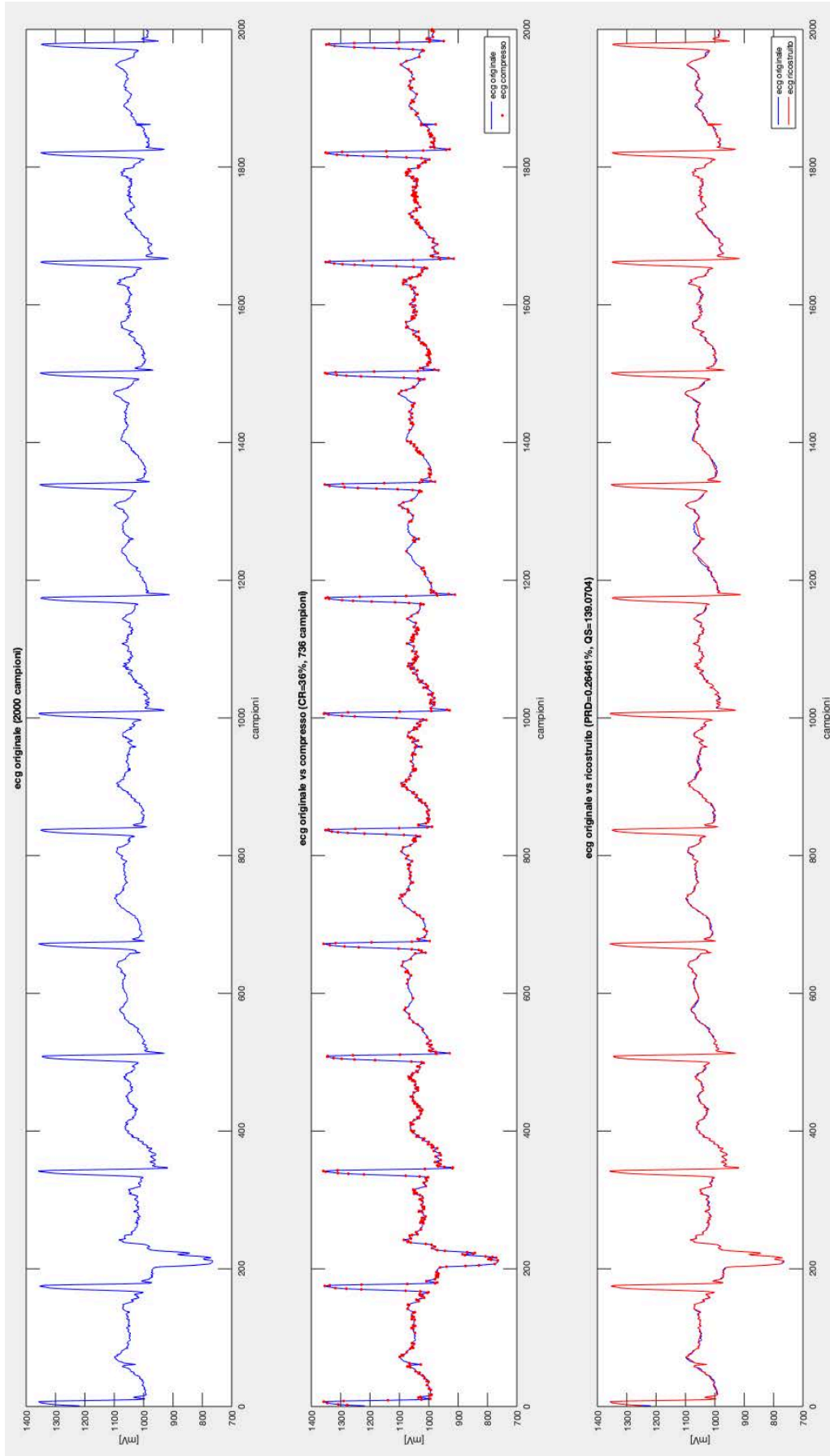


Figura 6.3.d Fan è applicato al segnale e0206m del database European ST-T. C'è perfetta corrispondenza tra il segnale originale e quello ricostruito.

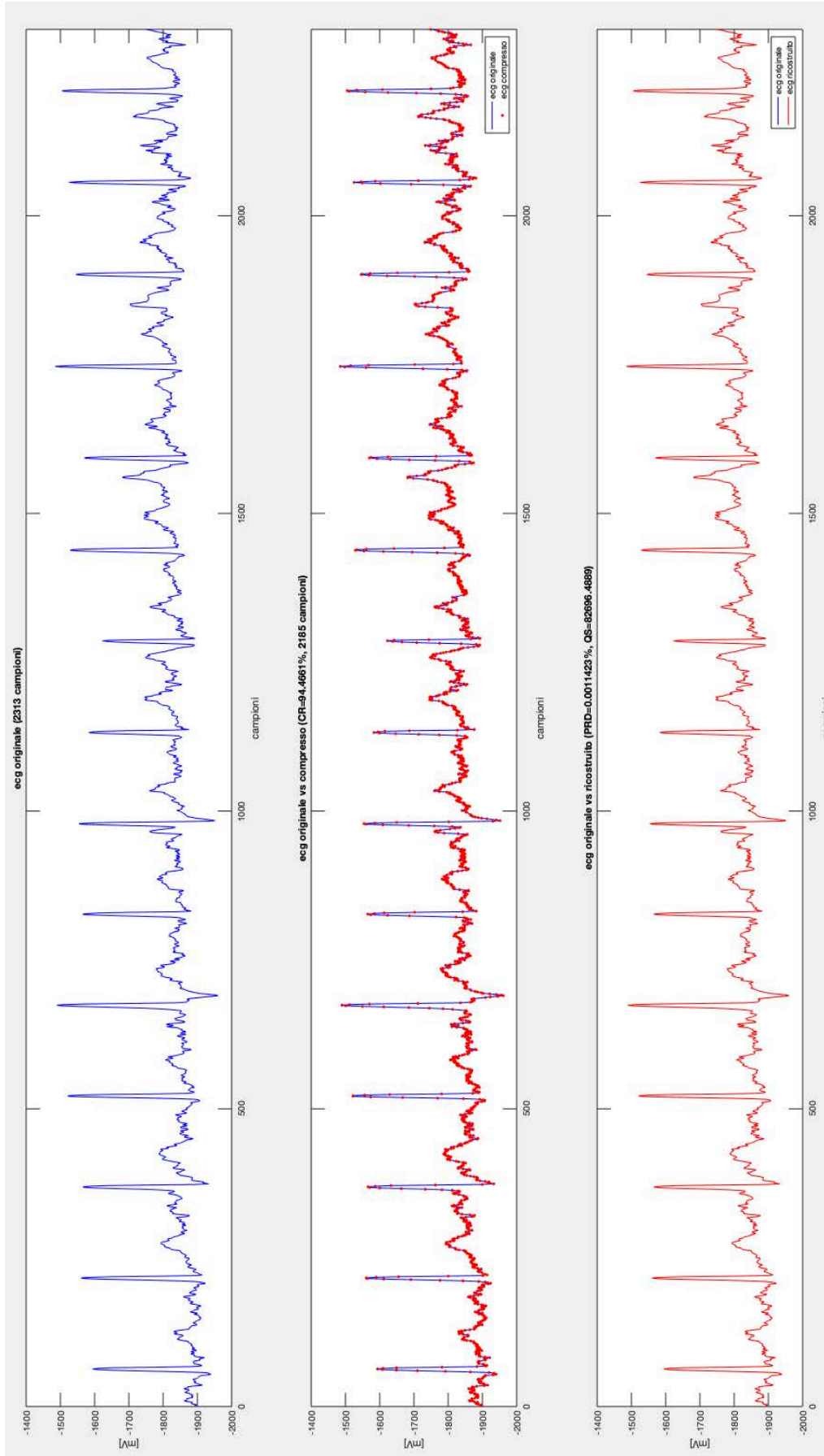


Figura 6.3.e Fan è applicato al segnale I01m del database St Petersburg INCART 12-lead Arrhythmia. C'è perfetta corrispondenza tra il segnale originale e quello ricostruito.

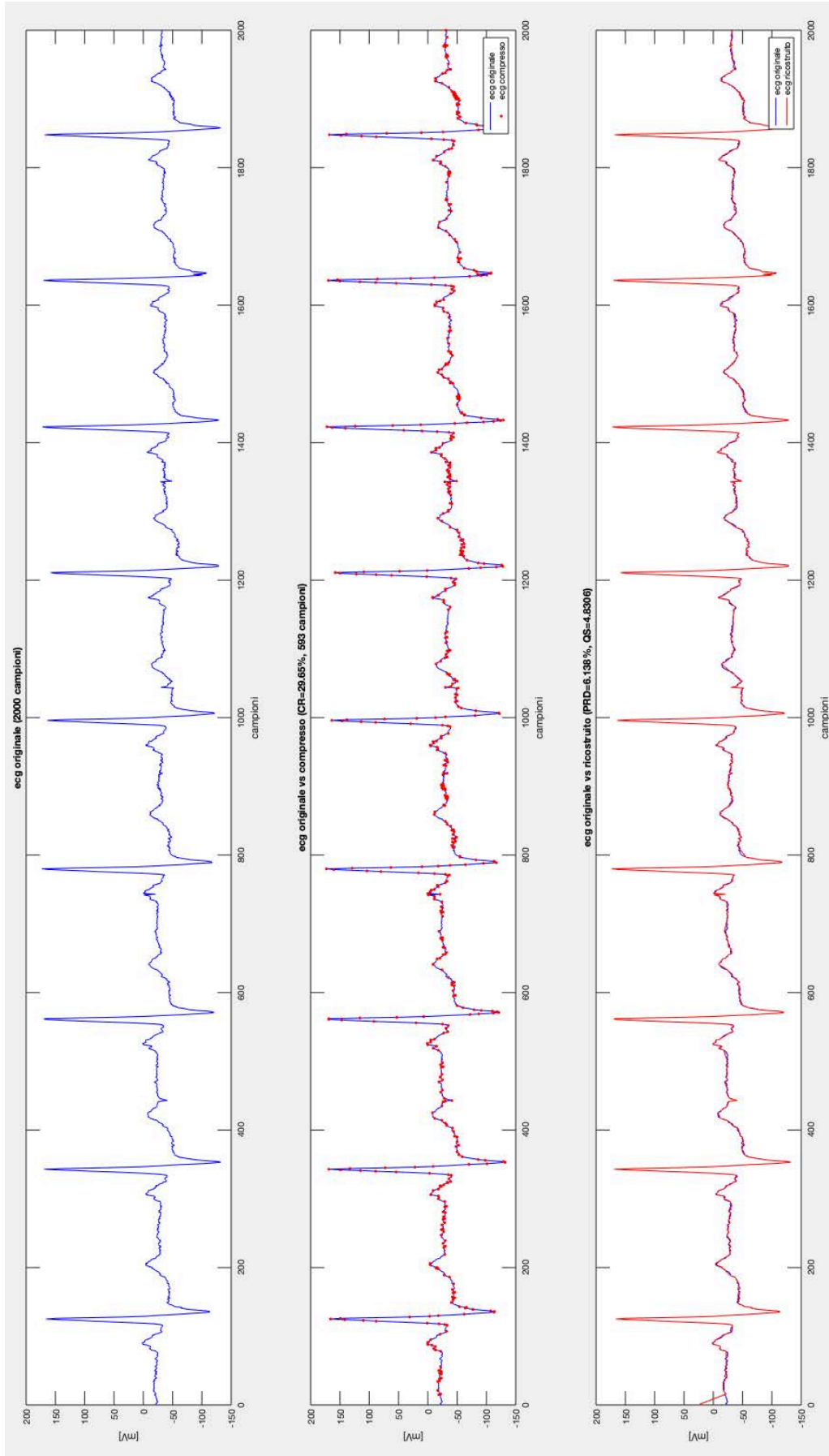


Figura 6.3. *f* Fan è applicato al segnale e0208 del database European ST-T. Il segnale ricostruito segue bene l'originale, eccetto all'estremo sinistro.

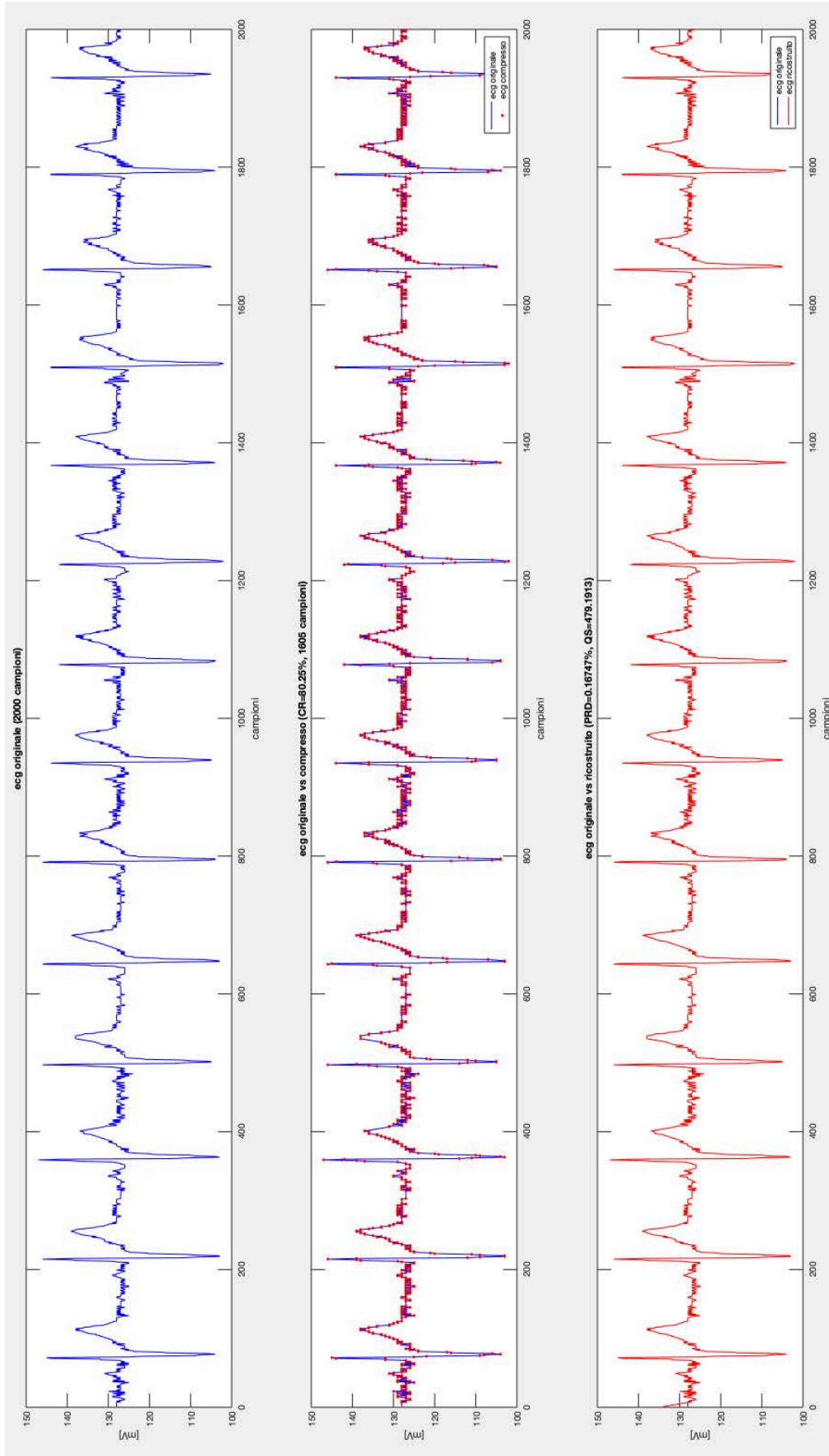


Figura 6.3.g Fan è applicato al segnale 02084m del database SHAREE. Il segnale ricostruito segue bene l'originale, eccetto all'estremo sinistro.

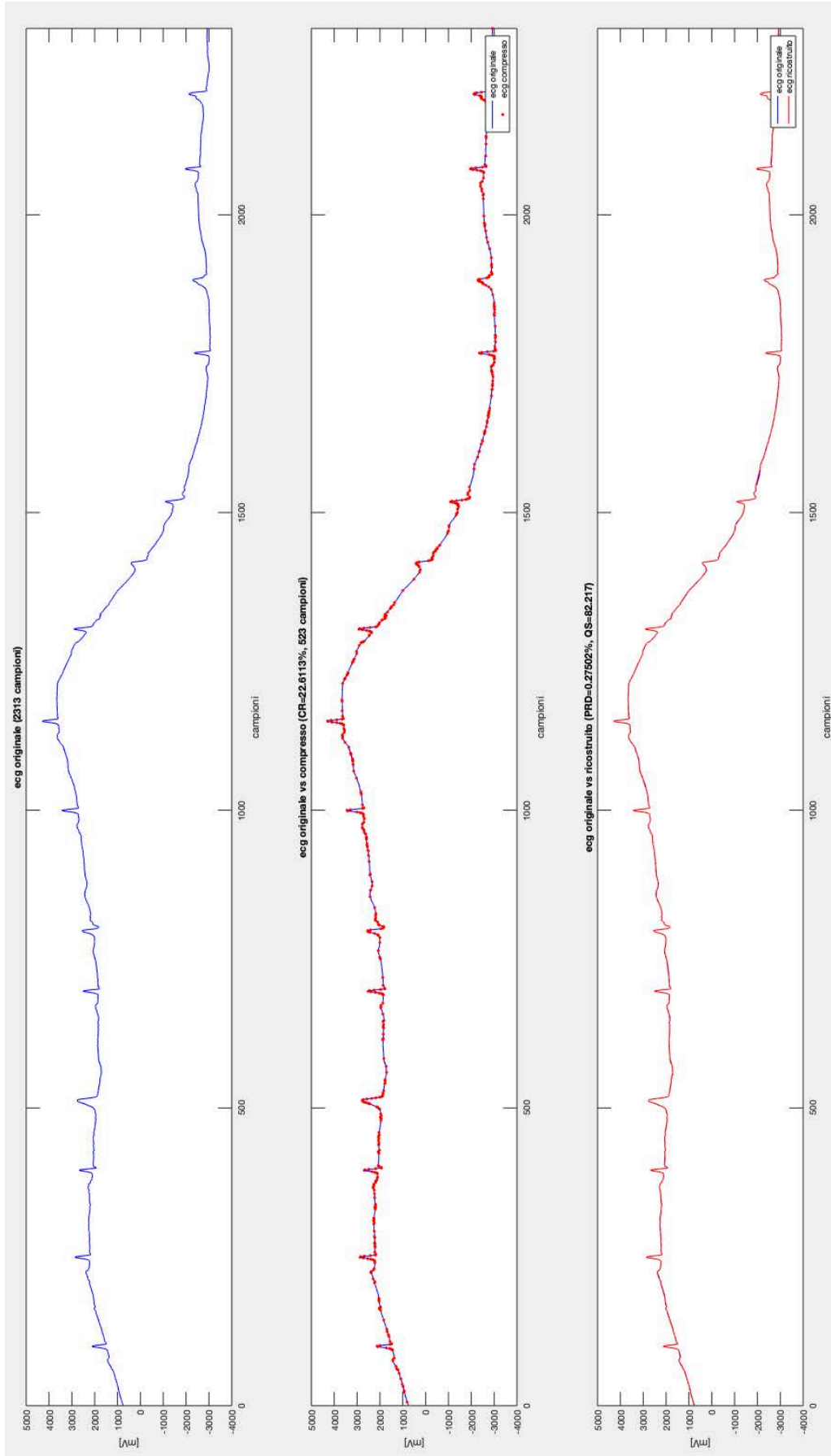


Figura 6.3.h Fan è applicato al segnale I01m del database St Petersburg INCART 12-lead Arrhythmia. Il segnale ricostruito segue l'originale in maniera meno precisa rispetto ad altri segnali del database, soprattutto all'estremo sinistro.

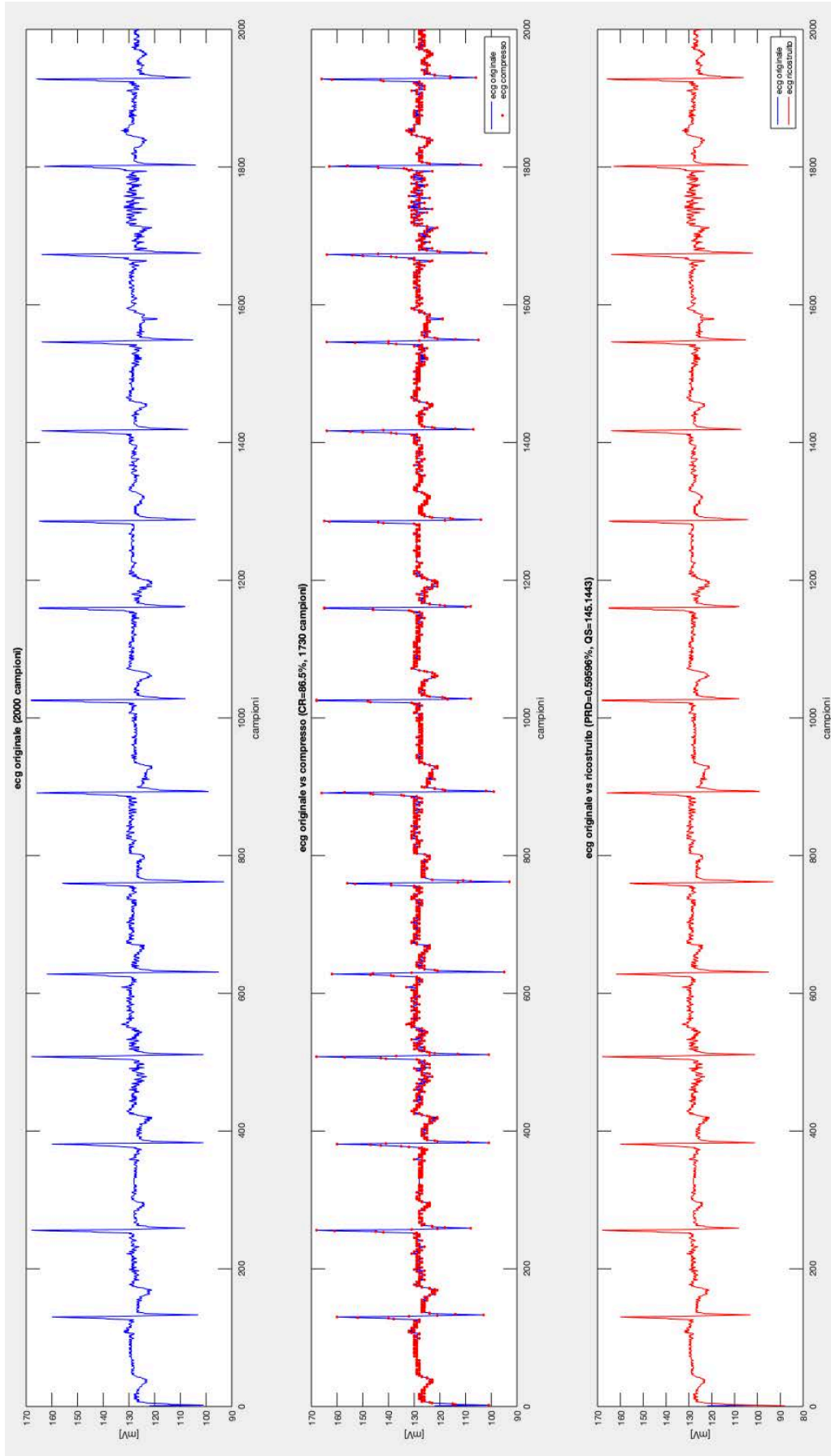


Figura 6.3.i Fan è applicato al segnale 02032m del database SHAREE. Il segnale ricostruito segue l'originale in maniera meno precisa rispetto ad altri segnali del database, soprattutto all'estremo sinistro.

7. DISCUSSIONE E VALUTAZIONE

7.1 Valutazione conclusiva e comparazione degli algoritmi

Come indicato nel paragrafo 3.2, gli algoritmi sono stati selezionati per la loro bassa complessità computazionale che li rende facilmente implementabili su dispositivi indossabili. Se da una parte i livelli di compressione ottenuti sono inferiori a quelli di tecniche parametriche più complesse [17], dall'altra le informazioni ricavate dai segnali ricostruiti risultano più appropriate.

Sulla base del confronto dei parametri si osserva che per Turning Point e Compressed Sensing i rapporti di compressione sono costanti, quindi il segnale è comprimibile solo fino a un limite definito che è indesiderabile. I parametri CR del metodo Fan dipendono dalla percentuale di errore tollerato, che dipende dal range dinamico del segnale.

Con la tecnica Turning Point si raggiungono i migliori livelli di compressione, mentre la tecnica Fan permette una migliore ricostruzione del segnale.

Il metodo Compressed Sensing permette di ottenere risultati intermedi, ma ha un'richiesta maggiore elaborazione dei dati per la ricostruzione del segnale.

Una volta stabilito il grado di precisione del segnale che si desidera, i tre metodi permettono una buona ricostruzione.

7.2 Conclusioni

Lo sviluppo della tematica della compressione dei segnali si inserisce uno studio esaustivo dei diversi aspetti della sanità digitale.

Si osserva per quali patologie è necessario un monitoraggio continuo, che sia un utile supporto per individuare una terapia che possa migliorare la vita dei pazienti. Si individuano i dispositivi adatti alla rilevazione dei dati biologici richiesti, ponendo attenzione sia alle caratteristiche fisiche che a quelle applicative. È necessario, infatti, che i devices siano dotati di tecnologie che permettano l'elaborazione e la trasmissione dei dati e siano confortevoli per rendere il monitoraggio poco invasivo e più completo possibile. Sono poi da valutare anche le strutture di archiviazione e i metodi di visualizzazione dei segnali.

Le tecniche di compressione analizzate sono solo alcune delle più recenti implementabili su dispositivi indossabili con una complessità computazionale tale da permettere rilevazione e trasmissione in maniera continuativa.

È opportuno proseguire nella ricerca di algoritmi in grado di bilanciare la compressione con l'esigenza di una rilevazione precisa dei segnali, che possano essere un supporto significativo per le diagnosi.

APPENDICE

Nelle seguenti pagine è riportato il codice della funzione *BSBL_BO*, che fa riferimento all'articolo [35] ed è scaricabile al link

<https://sites.google.com/site/researchbyzhang/bsbl>.

```

function Result = BSBL_BO(Phi, y, blkStartLoc, LearnLambda, varargin)

% BSBL-BO: Recover block sparse signal (1D) exploiting intra-block correlation, given the block partition.
%
%       The algorithm solves the inverse problem for the block sparse
%       model with known block partition:
%       y = Phi * x + v
%
%
% ===== INPUTS =====
% Phi      : N X M known matrix
% y        : N X 1 measurement vector
% blkStartLoc : Start location of each block
% LearnLambda : (1) If LearnLambda = 1, use the lambda learning rule for very LOW SNR cases (SNR<10dB)
%               (using lambda=std(y)*1e-2 or user-input value as initial value)
%               (2) If LearnLambda = 2, use the lambda learning rule for medium noisy cases (SNR>10dB)
%               (using lambda=std(y)*1e-2 or user-input value as initial value)
%               (3) If LearnLambda = 0, do not use the lambda learning rule
%               ((using lambda=1e-14 or user-input value as initial value)
%
% [varargin values -- in most cases you can use the default values]
% 'LEARNNTYPE' : LEARNNTYPE = 0: Ignore intra-block correlation
%               LEARNNTYPE = 1: Exploit intra-block correlation
%               [ Default: LEARNNTYPE = 1 ]
% 'PRUNE_GAMMA' : threshold to prune out small gamma_i
%               (generally, 10^{-3} or 10^{-2})
% 'LAMBDA'      : user-input value for lambda
%               [ Default: LAMBDA=1e-14 when LearnLambda=0; LAMBDA=std(y)*1e-2 in noisy cases]
% 'MAX_ITERS'   : Maximum number of iterations.
%               [ Default value: MAX_ITERS = 600 ]
% 'EPSILON'     : Solution accuracy tolerance parameter
%               [ Default value: EPSILON = 1e-8 ]
% 'PRINT'       : Display flag. If = 1: show output; If = 0: suppress output
%               [ Default value: PRINT = 0 ]
%
% ===== OUTPUTS =====
% Result :
% Result.x      : the estimated block sparse signal
% Result.gamma_used : indexes of nonzero groups in the sparse signal
% Result.gamma_est : the gamma values of all the groups of the signal
% Result.B      : the final value of the B
% Result.count  : iteration times
% Result.lambda : the final value of lambda
%
% ===== Command examples =====
% < Often-used command >
% For most noisy environment (SNR > 10dB):
%
%     Result = BSBL_BO(Phi, y, blkStartLoc, 2);
%
% For very low SNR cases (SNR < 10 dB):
%
%     Result = BSBL_BO(Phi, y, blkStartLoc, 1);
%
% For noiseless cases:
%
%     Result = BSBL_BO(Phi, y, blkStartLoc, 0);
%
% To recover non-Sparse structured signals (noiseless):
%     Result = BSBL_BO(Phi,y,groupStartLoc,0,'prune_gamma',-1);
%     ('prune_gamma' can be set any non positive constant)
%
% < Full-Command Example >
%     Result = BSBL_BO(Phi, y, blkStartLoc, learnlambda, ...
%                     'LEARNNTYPE', 1,...
%                     'PRUNE_GAMMA',1e-2,...
%                     'LAMBDA',1e-3,...
%                     'MAX_ITERS', 800,...
%                     'EPSILON', 1e-8,...
%                     'PRINT',0);

% scaling...
scl = std(y);
if (scl < 0.4) | (scl > 1) %#ok<R2>
    y = y/scl*0.4;

```

Figura A1 prima parte del codice MatLab della funzione BSBL-BO

```

end

% Default Parameter Values for Any Cases
EPSILON = 1e-8; % solution accuracy tolerance
MAX_ITERS = 600; % maximum iterations
PRINT = 0; % don't show progress information
LEARNTYPE = 1; % adaptively estimate the covariance matrix B

if LearnLambda == 0
    lambda = 1e-12;
    PRUNE_GAMMA = 1e-3;
elseif LearnLambda == 2
    lambda = scl * 1e-2;
    PRUNE_GAMMA = 1e-2;
elseif LearnLambda == 1
    lambda = scl * 1e-2;
    PRUNE_GAMMA = 1e-2;
else
    error(['Unrecognized Value for Input Argument 'LearnLambda']);
end

if(mod(length(varargin),2)==1)
    error('Optional parameters should always go by pairs\n');
else
    for i=1:2:(length(varargin)-1)
        switch lower(varargin{i})
            case 'learntype'
                LEARNTYPE = varargin{i+1};
                if LEARNTYPE ~= 1 && LEARNTYPE ~= 0
                    error(['Unrecognized Value for Input Argument 'LEARNTYPE']);
                end
            case 'prune_gamma'
                PRUNE_GAMMA = varargin{i+1};
            case 'lambda'
                lambda = varargin{i+1};
            case 'epsilon'
                EPSILON = varargin{i+1};
            case 'print'
                PRINT = varargin{i+1};
            case 'max_iters'
                MAX_ITERS = varargin{i+1};
            otherwise
                error(['Unrecognized parameter: ' varargin{i} '']);
        end
    end
end

if PRINT
    fprintf('\n===== \n');
    fprintf('          Running BSBL-BO ..... \n');
    fprintf('          Information about parameters... \n');
    fprintf('===== \n');
    fprintf('PRUNE_GAMMA : %e\n', PRUNE_GAMMA);
    fprintf('lambda : %e\n', lambda);
    fprintf('LearnLambda : %d\n', LearnLambda);
    fprintf('LearnType : %d\n', LEARNTYPE);
    fprintf('EPSILON : %e\n', EPSILON);
    fprintf('MAX_ITERS : %d\n', MAX_ITERS);
end

% Initialization
[N,M] = size(Phi);
Phi0 = Phi;
blkStartLoc0 = blkStartLoc;
p = length(blkStartLoc); % block number
for k = 1 : p-1
    blkLenList(k) = blkStartLoc(k+1)-blkStartLoc(k);
end
blkLenList(p) = M - blkStartLoc(end)+1;
maxLen = max(blkLenList);
if sum(blkLenList == maxLen) == p
    equalSize = 1;
else
    equalSize = 0;
end

for k = 1 : p
    Sigma0{k} = eye(blkLenList(k));
end

gamma = ones(p,1);
keep_list = [1:p]';
usedNum = length(keep_list);
mu_x = zeros(M,1);
count = 0;

```

Figura A2 seconda parte del codice MatLab della funzione BSBL-BO

```

% Iteration
while (1)
    count = count + 1;

    %===== Prune weighys as yheir hyperparameyers go yo zero =====
    if (min(gamma) < PRUNE_GAMMA)
        index = find(gamma > PRUNE_GAMMA);
        usedNum = length(index);
        keep_list = keep_list(index);
        if isempty(keep_list)
            fprintf('\n=====');
            fprintf('x becomes zero vector. The solution may be incorrect. \n');
            fprintf('Current 'prune_gamma' = %g, and Current 'EPSILON' = %g.\n',PRUNE_GAMMA,EPSILON);
            fprintf('Try smaller values of 'prune_gamma' and 'EPSILON' or normalize 'y' to unit norm.\n');
            fprintf('=====');
            break;
        end
        blkStartLoc = blkStartLoc(index);
        blkLenList = blkLenList(index);

        % prune gamma and associated components in Sigma0
        gamma = gamma(index);
        temp = Sigma0;
        Sigma0 = [];
        for k = 1 : usedNum
            Sigma0{k} = temp{index(k)};
        end

        % construct new Phi
        temp = [];
        for k = 1 : usedNum
            temp = [temp, Phi0(:,blkStartLoc(k):blkStartLoc(k)+blkLenList(k)-1)];
        end
        Phi = temp;
        %clear temp;
    end

    %===== Compute new weights =====
    mu_oLd = mu_x;

    PhiBPhi = zeros(N);
    currentLoc = 0;
    for i = 1 : usedNum

        currentLen = size(Sigma0{i},1);
        currentLoc = currentLoc + 1;
        currentSeg = currentLoc : 1 : currentLoc + currentLen - 1;

        PhiBPhi = PhiBPhi + Phi(:, currentSeg)*Sigma0{i}*Phi(:, currentSeg)';
        currentLoc = currentSeg(end);
    end

    H = Phi' / (PhiBPhi + lambda * eye(N));
    Hy = H * y;
    HPhi = H * Phi;

    mu_x = zeros(size(Phi,2),1);
    Sigma_x = [];
    Cov_x = [];

    B = []; invB = []; B0 = zeros(maxLen); r0 = zeros(1); r1 = zeros(1);
    currentLoc = 0;
    for i = 1 : usedNum

        currentLen = size(Sigma0{i},1);
        currentLoc = currentLoc + 1;
        seg = currentLoc : 1 : currentLoc + currentLen - 1;

        mu_x(seg) = Sigma0{i} * Hy(seg); % solution
        Sigma_x{i} = Sigma0{i} - Sigma0{i} * HPhi(seg,seg) * Sigma0{i};
        Cov_x{i} = Sigma_x{i} + mu_x(seg) * mu_x(seg)';
        currentLoc = seg(end);

        %===== Learn correlation structure in blocks =====
        % do not consider correlation structure in each block
        if LEARNTYPE == 0
            B{i} = eye(currentLen);
            invB{i} = eye(currentLen);

        % constrain all the blocks have the same correlation structure
        elseif LEARNTYPE == 1
            if equalSize == 0
                if currentLen > 1
                    temp = Cov_x{i}/gamma(i);
                    r0 = r0 + mean(diag(temp));
                    r1 = r1 + mean(diag(temp,1));
                end
            end
        end
    end
end

```

Figura A3 terza parte del codice MatLab della funzione BSBL-BO

```

        elseif equalSize == 1
            B0 = B0 + Cov_x{i}/gamma(i);
        end
    end % end of learnType
end

%===== Learn correlation structure in blocks with Constraint 1 =====
% If blocks have the same size
if (equalSize == 1) & (LEARNTYPE == 1) %#ok<AND2>

    % Constrain all the blocks have the same correlation structure
    % (an effective strategy to avoid overfitting)
    b = (mean(diag(B0,1))/mean(diag(B0)));
    if abs(b) >= 0.99, b = 0.99*sign(b); end
    bs = [];
    for j = 1 : maxLen, bs(j) = (b)^(j-1); end
    B0 = toeplitz(bs);

    for i = 1 : usedNum
        B{i} = B0;
        invB{i} = inv(B{i});
    end

% if blocks have different sizes
elseif (equalSize == 0) & (LEARNTYPE == 1)
    r = r1/r0; if abs(r) >= 0.99, r = 0.99*sign(r); end

    for i = 1 : usedNum
        currentLen = size(Cov_x{i},1);

        bs = [];
        for j = 1 : currentLen, bs(j) = r^(j-1); end
        B{i} = toeplitz(bs);
        invB{i} = inv(B{i});
    end

end

end

% estimate gamma(i) and lambda
if LearnLambda == 1
    gamma_old = gamma;
    lambdaComp = 0; currentLoc = 0;
    for i = 1 : usedNum

        currentLen = size(Sigma_x{i},1);
        currentLoc = currentLoc + 1;
        currentSeg = currentLoc + currentLen - 1;

        gamma(i) = gamma_old(i)*norm( sqrtm(B{i})*Hy(currentSeg) )/sqrt(trace(HPhi(currentSeg,currentSeg)*B{i})
        lambdaComp = lambdaComp + trace(Phi(:,currentSeg)*Sigma_x{i}*Phi(:,currentSeg)');
        Sigma0{i} = B{i} * gamma(i);
        currentLoc = currentSeg(end);
    end
    lambda = norm(y - Phi * mu_x,2)^2/N + lambdaComp/N;

elseif LearnLambda == 2
    gamma_old = gamma;
    lambdaComp = 0; currentLoc = 0;
    for i = 1 : usedNum

        currentLen = size(Sigma_x{i},1);
        currentLoc = currentLoc + 1;
        currentSeg = currentLoc : 1 : currentLoc + currentLen - 1;

        gamma(i) = gamma_old(i)*norm( sqrtm(B{i})*Hy(currentSeg) )/sqrt(trace(HPhi(currentSeg,currentSeg)*B{i})
        lambdaComp = lambdaComp + trace(Sigma_x{i}*invB{i})/gamma_old(i);
        Sigma0{i} = B{i} * gamma(i);
        currentLoc = currentSeg(end);
    end
    lambda = norm(y - Phi * mu_x,2)^2/N + lambda * (length(mu_x) - lambdaComp)/N;

else % only estimate gamma(i)
    gamma_old = gamma;
    currentLoc = 0;
    for i = 1 : usedNum
        % gamma(i) = trace(invB{i} * Cov_x{i})/size(Cov_x{i},1);

```

Figura A4 quarta parte del codice MatLab della funzione BSBL-BO

```

        currentLen = size(Sigma0{i},1);
        currentLoc = currentLoc + 1;
        seg = currentLoc : 1 : currentLoc + currentLen - 1;

        gamma(i) = gamma_old(i)*norm( sqrtm(B{i})*Hy(seg) )/sqrt(trace(HPhi(seg,seg)*B{i}));

        Sigma0{i} = B{i} * gamma(i);

        currentLoc = seg(end);
    end
end

% ===== Check stopping conditions, eyc. =====
if (size(mu_x) == size(mu_old))
    dmu = max(max(abs(mu_old - mu_x)));
    if (dmu < EPSILON) break; end %#ok<*>SEPEX>
end
if (PRINT)
    disp([' iters: ', num2str(count), ...
        ' num coeffs: ', num2str(usedNum), ...
        ' min gamma: ', num2str(min(gamma)), ...
        ' gamma change: ', num2str(max(abs(gamma - gamma_old))), ...
        ' mu change: ', num2str(dmu)]);
end
if (count >= MAX_ITERS), if PRINT, fprintf('Reach max iterations. Stop\n\n'); end; break; end
end

if isempty(keep_list)
    Result.x = zeros(M,1);
    Result.gamma_used = [];
    Result.gamma_est = zeros(p,1);
    Result.B = B;
    Result.count = count;
    Result.lambdatrace = lambda;
else
    % Expand hyperparameters
    gamma_used = sort(keep_list);
    gamma_est = zeros(p,1);
    gamma_est(keep_list,1) = gamma;

    % reconstruct the original signal
    x = zeros(M,1);
    currentLoc = 0;
    for i = 1 : usedNum
        currentLen = size(Sigma0{i},1);
        currentLoc = currentLoc + 1;
        seg = currentLoc : 1 : currentLoc + currentLen - 1;

        realLocs = blkStartLoc0(keep_list(i)) : blkStartLoc0(keep_list(i))+currentLen-1;

        x( realLocs ) = mu_x( seg );
        currentLoc = seg(end);
    end

    if (scl < 0.4) | (scl > 1)
        Result.x = x * scl/0.4;
    else
        Result.x = x;
    end
    Result.gamma_used = gamma_used;
    Result.gamma_est = gamma_est;
    Result.B = B;
    Result.count = count;
    Result.lambda = lambda;
end
return;

```

Figura A5 quinta parte del codice MatLab della funzione BSBL-BO

BIBLIOGRAFIA

- [1] Progress Report on the implementation of the European Disability Strategy (2010 - 2020)
- [2] Comunicazione della Commissione Europea del 6/12/2012 sul Piano d'azione "Sanità elettronica" 2012-2020 – Una sanità innovativa per il 21° secolo
- [3] WHO Library Cataloguing-in-Publication Data, Telemedicine: opportunities and developments in Member States: report on the second global survey on eHealth 2009 (Global Observatory for eHealth Series, 2)
- [4] Ni L., Sun Y., Yang Y., He L., "Human-Engaged Health Care Services Recommendation for Aging and Longterm Care", In: *Proceedings of the 2019 IEEE 23rd International Conference on Computer Supported Cooperative Work in Design*
- [5] Linee di indirizzo nazionali sulla Telemedicina del Ministero della Salute Italiano
- [6] <https://www.fascicolosanitario.gov.it>
- [7] Haddad O., Khalighi M. A., "Enabling Communication Technologies for Medical Wireless Body-Area Networks". In: *2019 Global LIFI Congress (GLC)*
- [8] Batista M., Sebastião P., Cortesão R., "WearIoT - Wearable Iot Human Emergency System". In: *2019 European Conference on Networks and Communications (EuCNC)*
- [9] Randazzo V., Ferretti J., Pasero E., "ECG WATCH: a real time wireless wearable ECG", in *IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, 2019
- [10] Di Palma V., De Venuto D., Ricci S., Frangi A., Stuart Savoia A., Di Nocera D., Zampognaro P., Coronato A., Infantino I., Pescosolido L., De Luca N., Di Matteo A., "Medical Assistance in Contextual awareness (AMICO): a project for a better cardiopathic patients quality of care." In: *2019 IEEE 8th International Workshop on Advances in Sensors and Interfaces (IWASI)*
- [11] Ahad A., Tahir M., Yau K.A., "5G-Based Smart Healthcare Network: Architecture, Taxonom, Challenges and Future Research Direction", *IEEE Access*, Vol. 7
- [12] Hamdan S., Awaian A., Almajali S., "Compression Techniques Used in Iot: A Comparitive Study". In: *2019 2nd International Conference on new Trends in Computing Sciences (ICTCS)*
- [13] Chong Y., Ismail W., Ko K., Lee C., "Energy Harvesting For Wearable Devices: A Review", *IEEE Sensors Journal*, Volume 19, No.20, Oct.15 2019

- [14] <http://www.ehnheart.org> (European Cardiovascular Disease Statistics 2017)
- [15] Polon'ski L., Wasilewski J., "An Introduction to ECG Interpretation", in Gacek A., Pedrycz W., *ECG Signal Processing, Classification and Interpretation: A Comprehensive Framework of Computational Intelligence*, Springer, 2012, pp.1-21
- [16] Simjanoska M., Koteska B., Kostoska M., "Information System for Biosensors Data Exchange in Healthcare", in Kulakov A., Stojanov G., *ICT Innovations 2016, Advances in Intelligent Systems and Computing*, Springer, pp.220-230
- [17] Hooshmand M., Zordan D., Del Testa D., Grisan E., Rossi M., "Boosting the Battery Life of Wearables for Health Monitoring Through the Compression of Biosignals", *IEEE Internet of Things Journal*, Volume 4, Issue 5, Oct. 2017
- [18] Zhang Z., Liu X., Wei S., Gan H., Liu F., Li Y., Liu C., "Electrocardiogram Reconstruction Based on Compressed Sensing", *IEEE Access*, Vol. 7
- [19] Cox J.R., Nolle F. M., Fozzard H. A., Oliver G. C., "AZTEC, a preprocessing system for real-time ECG rhythm analysis," *IEEE Trans. Biomed. Eng.*, vol. BME-15, no. 2, pp. 128–129, Apr. 1968
- [20] Abenstein J. P., Tompkins W. J., "A new data-reduction algorithm for real-time ECG analysis," *IEEE Trans. Biomed. Eng.*, vol. BME-29, no. 1, pp. 43–48, Jan. 1982
- [21] Deepu C.J., C. Heng, Y Lian, A Hybrid Data Compression Scheme for Power Reduction in Wireless Sensors for IoT. In: *IEEE Transactions On Biomedical Circuits And Systems*, VOL. 11, no. 2, April 20
- [22] Reddy B. R. S., Murthy I. S. N., "ECG data compression using Fourier descriptors," *IEEE Trans. Biomed. Eng.*, vol. BME-33, No. 4, pp. 428–434, Apr. 1986.
- [23] Lee H., Buckley K. M., "ECG data compression using cut and align beats approach and 2-D transforms," *IEEE Trans. Biomed. Eng.*, vol. 46, No. 5, pp. 556–564, May 1999
- [24] Rajoub B. A., "An efficient coding algorithm for the compression of ECG signals using the wavelet transform," *IEEE Trans. Biomed. Eng.*, vol. 49, No. 4, pp. 355–362, Apr. 2002.
- [25] Maglaveras N., Stampkopoulos T., Diamantaras K., Pappas C., Strintzis M., "ECG pattern recognition and classification using non-linear transformations and neural networks: A review," *Int. J. Med. Inf.*, vol. 52, No. 1–3, pp. 191–208, Oct. 1998

[26] Sun C.-C., Tai S.-C., “Beat-based ECG compression using gain- shape vector quantization,” *IEEE Trans. Biomed. Eng.*, vol. 52, No. 11, pp. 1882–1888, Nov. 2005.

[27] Cardenas-Barrera J. L., Lorenzo-Ginori J. V., “Mean-shape vector quantizer for ECG signal compression,” *IEEE Trans. Biomed. Eng.*, vol. 46, No. 1, pp. 62–70, Jan. 1999.

[28] Del Testa D., Rossi M., “Lightweight lossy compression of biometric patterns via denoising autoencoders,” *IEEE Signal Process. Lett.*, vol. 22, No. 12, pp. 2304–2308, Dec. 2015

[29] <https://physionet.org>

[30] Melillo P., Izzo R., Orrico, Scala R. P., Attanasio M., Mirra M., De Luca N., Pecchia L., “Automatic prediction of cardiovascular and cerebrovascular events using Heart Rate Variability analysis”, Plos One, in press

[31] Taddei A, Distanto G, Emdin M, Pisani P, Moody GB, Zeelenberg C, Marchesi C. The European ST-T Database: standard for evaluating systems for the analysis of ST-T changes in ambulatory electrocardiography. *European Heart Journal* 13: 1164-1172 (1992). Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark RG, Mietus JE, Moody GB, Peng C-K, Stanley HE. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals (2003). *Circulation*. 101(23): e215-e220.

[32] Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark RG, Mietus JE, Moody GB, Peng C-K, Stanley HE. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals (2003). *Circulation*. 101(23): e215-e220.

[33] *El B’charri O., Latif R., Abenaou A., Dliou A., Jenkal W., “An Efficient Lossless Compression Scheme for ECG Signal”, *International Journal of Advanced Computer Science and Applications*, Vol. 7, No. 7, 2016

[34] Anjum M. S., Chakraborty M., “ECG Data Compression using Turning Point Algorithm”, *International Journal of Innovative Research in Engineering & Multidisciplinary Physical Sciences (IJIRMP)*, Vol. 2, No. 6, Dec. 2014

[35] Zhang Z., Jung T., Makeig S., Rao B. D., “Compressed Sensing of EEG for Wireless Telemonitoring with Low Energy Consumption and Inexpensive Hardware”, *IEEE Transactions On Biomedical Engineering*, vol. 60, No.1, pp. 221-224, 2013