



UNIVERSITÀ
DEGLI STUDI
DI PADOVA



DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN BIOINGEGNERIA

***Algoritmi Bayesiani per il filtraggio di segnali di heart
rate variability acquisiti da sensori indossabili in
individui affetti da sclerosi multipla o sclerosi
amiotrofica laterale***

Relatore: Prof. Facchinetti Andrea

Laureando: Leoni Luca

Correlatore: Prof. Cappon Giacomo

Correlatore: Dott. Cossu Luca

15 DICEMBRE 2022 - ANNO ACCADEMICO 2021 – 2022

Abstract

L'obiettivo del presente lavoro di tesi è lo sviluppo di un algoritmo di filtraggio del rumore di misura in segnali heart rate (HR) misurati in individui affetti da sclerosi amiotrofica laterale (SLA) o sclerosi multipla (SM) all'interno del progetto europeo BRAINTEASER. I dati vengono raccolti per mezzo di un sensore wearable (Garmin vivoactive4). Una volta acquisiti, i tracciati HR vengono processati da una pipeline di processing, sviluppata in python, che permette di visualizzare il tracciato giornaliero, segmentarlo in finestre, e filtrarlo dal rumore di misura mediante un algoritmo di smoothing Bayesiano adattativo che riesce a stimare, nella finestra stessa, il rapporto segnale-rumore. Come ultimo step le finestre vengono tra loro riconciliate per ottenere nuovamente il tracciato giornaliero intero. La capacità dell'algoritmo, che è quella di produrre un filtraggio migliore rispetto alle tecniche di letteratura, è stata validata confrontando l'errore di quantificazione nell'estrazione di features di HR rispetto ad un segnale gold standard di riferimento.

Indice

INDICE	- 4 -
INDICE DELLE FIGURE	- 6 -
INDICE DELLE TABELLE.....	- 8 -
CAPITOLO 1 - INTRODUZIONE.....	- 9 -
1.1 BRAINTEASER	- 9 -
1.2 SENSORI INDOSSABILI.....	- 11 -
1.2.1 Garmin Vivoactive® 4	- 11 -
1.3 SMOOTHING DI SEGNALI BIOMEDICI: STATO DELL'ARTE.....	- 12 -
1.3.1 Filtraggio di segnale Heart Rate	- 12 -
1.3.2 FLIRT	- 13 -
1.4 OBIETTIVI E FINALITÀ DEL LAVORO	- 13 -
CAPITOLO 2 – SMOOTHING BAYESSIANO	- 15 -
2.1 SMOOTHING BAYESIANO: CENNI DI TEORIA.....	- 15 -
2.1.1 Stima Bayesiana	- 15 -
2.1.2 Random walk e Random walk integrato	- 19 -
2.1.3 criterio di consistenza	- 21 -
2.2 ANALISI PRELIMINARE.....	- 23 -
2.2.1 Python	- 23 -
2.2.2 analisi preliminare	- 24 -
2.3 APPLICAZIONE DI SMOOTHING BAYESIANO IN AMBIENTE SIMULATO.....	- 26 -
2.3.1 Ground truth, rumore, segnale rumoroso.....	- 27 -
2.3.2 Sviluppo del codice per lo smoothing bayesiano.....	- 30 -
2.3.3 Gestione dei missing samples	- 33 -
2.3.4 Innesco	- 36 -
CAPITOLO 3 - PIPELINE PER IL FILTRAGGIO DI DATI HEART RATE	- 39 -
3.1 DIVISIONE DEI DATI HEART RATE IN FINESTRE.....	- 39 -
3.1.1 Procedura di suddivisione in finestre	- 40 -
3.2 RICONCILIAZIONE DELLE FINESTRE	- 45 -
3.2.1 Kernel Gaussiano	- 46 -
3.2.2 Risultati della riconciliazione	- 48 -
3.3 RESIDUI.....	- 50 -
CAPITOLO 4 - RISULTATI	- 52 -

4.1	POLAR H10: GOLD STANDARD DI RIFERIMENTO	- 52 -
4.1.1	Confronto Garmin-Polar	- 52 -
4.2	RISULTATI	- 55 -
CAPITOLO 5 - CONCLUSIONI		- 66 -
5.1	STEPS FUTURI	- 67 -
BIBLIOGRAFIA		- 68 -
APPENDICE		- 71 -
	<i>Ap.1 - file JSON</i>	- 71 -
	<i>Ap2 - pipeline</i>	- 76 -
	<i>Ap3 - funzioni implementate</i>	- 78 -

Indice delle Figure

Figura 1: logo del progetto europeo BRAINTEASER [3].....	- 10 -
Figura 2: Garmin vivoactive 4, smartwatch utilizzato per il progetto BRAINTEASER [11].	- 11 -
Figura 3: Steps implementati dal pacchetto FLIRT [17], i dati provenienti da sensori indossabili vengono elaborati da una pipeline in grado di processarli e generare delle features da utilizzare in algoritmi di machine learning.	- 13 -
Figura 4: in alto è riportato un processo di rumore bianco, al centro un modello di random walk simulato e in basso un modello di random walk integrato simulato. Tutti i processi sono composti da mille campioni.....	- 21 -
Figura 5: In alto vengono riportati da sx a dx i segnali di Hr, Steps e Stress, in basso nello stesso ordine quelli di SpO2 e Respiration. In tutti i grafici vengono riportati dei marcatori di colore diverso dal tracciato i punti in cui iniziano dei campioni mancanti.	- 25 -
Figura 6: tre esempi di segnale HR, con marcatori rossi in presenza di campioni mancanti. Nel primo e secondo riquadro si apprezza la diversa variabilità tra sonno e veglia, mentre in basso si vede una zona consistente di campioni mancanti e una presunta attività fisica.....	- 26 -
Figura 7: Ground truth composto da cento campioni.....	- 27 -
Figura 8: processo di rumore bianco con media nulla e varianza unitaria	- 28 -
Figura 9: ampiezza di ground truth e rumore a confronto.....	- 29 -
Figura 10: ground truth e segnale rumoroso a confronto.....	- 29 -
Figura 11: passaggio implementativo per la creazione della matrice G.....	- 31 -
Figura 12: ground truth, segnale rumoroso, segnale filtrato con relativo intervallo di confidenza a confronto. Corretto funzionamento del filtraggio bayesiano.....	- 32 -
Figura 13: primo grafico mostra il segnale filtrato, segnale rumoroso e ground truth a confronto [Figura 12], secondo grafico mette in relazione l'errore di misura con i residui assoluti, il terzo grafico mostra i residui normalizzati rispetto l'intervallo +-1	- 33 -
Figura 14: ground truth in confronto al segnale rumoroso defezionato di 40 campioni mancanti in due zone distinte	- 34 -
Figura 15: confronto tra ground truth, segnale rumoroso in ingresso allo stimatore e segnale filtrato con relativo intervallo di confidenza che aumenta in corrispondenza di campioni mancanti.....	- 35 -
Figura 16: : primo grafico mostra il segnale filtrato, segnale rumoroso e ground truth a confronto [Figura 15], secondo grafico mette in relazione l'errore di misura con i residui assoluti, il terzo grafico mostra i residui normalizzati rispetto l'intervallo +-1. Si noti la mancanza di residui in corrispondenza di campioni mancanti -	- 35 -
Figura 17: si riporta il segnale rumoroso e ground truth a confronto traslati indietro nel tempo rispetto a Figura 10, ciò verrà utilizzato per testare l'innesco	- 36 -
Figura 18: confronto tra ground truth, segnale rumoroso in ingresso allo stimatore e segnale filtrato con relativo intervallo di confidenza che aumenta in corrispondenza dell'innesco	- 37 -

Figura 19: confronto tra ground truth, segnale rumoroso in ingresso allo stimatore e segnale filtrato con relativo intervallo di confidenza. Questo è il risultato finale dell'operazione svolta per la gestione dell'innesco ...	- 38 -
Figura 20: primo grafico mostra il segnale filtrato, segnale rumoroso e ground truth a confronto [Figura 19], secondo grafico mette in relazione l'errore di misura con i residui assoluti, il terzo grafico mostra i residui normalizzati rispetto l'intervallo +-	- 38 -
Figura 21: Segnale Heart Rate complessivo da filtrare e marker per segnalare l'inizio di una zona di campioni mancanti.	- 41 -
Figura 22: in figura sono visibili i campioni della finestra da filtrare prelevati dall'intero tracciato	- 42 -
Figura 23: confronto tra segnale rumoroso in ingresso allo stimatore nella prima finestra e segnale filtrato con relativo intervallo di confidenza.....	- 43 -
Figura 24: residui assoluti e normalizzati relativi alla finestra mostrata in Figura 23 rispettivamente visionabili nel primo e nel secondo grafico	- 43 -
Figura 25: filtraggio della seconda finestra; è visibile il segnale grezzo, il segnale filtrato con relativo intervallo di confidenza. In questa finestra sono presenti due zone di campioni mancanti dove l'intervallo di confidenza aumenta.....	- 44 -
Figura 26: filtraggio dell'ultima finestra composta da 91 campioni, vengono riportati il segnale grezzo, quello filtrato con relativo intervallo di confidenza.....	- 45 -
Figura 27: in alto il kernel gaussiano, in basso porzioni di kernel usate come pesi per la finestra precedente e quella successiva.	- 47 -
Figura 28: segnale grezzo e segnale filtrato dopo aver riconciliato le finestre tra loro nei punti con marker nero..	- 48 -
Figura 29: ingrandimento della Figura 28 nel quale si apprezza l'andamento regolare nel punto di riconciliazione tra due finestre (marker nero).....	- 49 -
Figura 30: esempio di riconciliazione senza kernel, è visibile un punto spigoloso (marker) dovuto alla semplice concatenazione delle finestre	- 49 -
Figura 31: in alto segnale grezzo e segnale filtrato riconciliato, al centro l'andamento a gradino del σ^2 , in basso i residui normalizzati.....	- 51 -
Figura 32: confronto tra segnali di Heart Rate prelevati da Polar H10 e Garmin vivoactive4.....	- 53 -
Figura 33: dettaglio di figura Figura 32 nel quale è possibile riconoscere un ritardo del segnale garmin rispetto a quello polar nei fronti di salita e discesa	- 54 -
Figura 34: segnali Polar e Garmin dopo essere stati tagliati nel punto iniziale e finale per renderli uniformi ed aver eliminato gli zeri nel tracciato Polar	- 55 -
Figura 35: confronto tra segnale Polar grezzo (polar_raw) e Garmin filtrato con filtro bayesiano (garmin_bayes)	- 61 -
Figura 36: heat map rappresentante gli errori relativi delle features giornaliere tra i confronti: garmin_bayes-garmin_media, garmin_bayes-polar_raw e garmin_media-polar_raw	- 62 -
Figura 37: heat map rappresentante gli errori relativi delle features in frequenza tra i confronti: garmin_bayes-garmin_media, garmin_bayes-polar_raw e garmin_media-polar_raw	- 62 -
Figura 38: heat map rappresentante gli errori relativi delle features geometriche tra i confronti: garmin_bayes-garmin_media, garmin_bayes-polar_raw e garmin_media-polar_raw	- 63 -

Indice delle Tabele

Tabella 1: features rappresentanti l'andamento giornaliero dei segnali	- 56 -
Tabella 2: features di carattere geometrico relative a intervalli battito-battito	- 59 -
Tabella 3: fetures in frequenza relative agli intervalli battito battito	- 60 -
Tabella 4: variabili di input alla pipeline	- 77 -
Tabella 5: variabili di output dalla pipeline	- 77 -
Tabella 6: funzioni implementate durante tutto il lavoro di tesi con le relative variabili di input e di output .	- 80 -

Capitolo 1

Introduzione

1.1 Brainteaser

BRAINTEASER [1] (BRinging Artificial INTelligence home for a better cAre of amyotrophic lateral sclerosis and multiple SclERosis) [*Figura 1*] è un progetto europeo quadriennale [2] finanziato dall'iniziativa Horizon 2020, con l'obiettivo di utilizzare big data relativi allo stato di salute, alle abitudini personali ed alle variabili ambientali, al fine di supportare il processo di cura di pazienti affetti da: sclerosi multipla (SM) e sclerosi amiotrofica laterale (SLA). In particolare, tali obiettivi sono perseguiti con lo sviluppo di nuovi algoritmi di intelligenza artificiale (AI) atti ad assistere il personale clinico durante le fasi di monitoraggio della progressione delle patologie e del trattamento delle rispettive complicanze. Come brevemente introdotto lo sviluppo di tali algoritmi sarà possibile grazie a grandi moli di dati sia retrospettivi che prospettivi raccolti da sensori indossabili smartwatch quali: Heart Rate (HR), passi, stress, SpO2, respirazione. I sensori indossabili sono diventati negli ultimi anni una tecnologia molto diffusa, soprattutto nell'uso sportivo e di monitoraggio della salute. Il mercato sta avendo una crescita esponenziale, difatti si vede incrementare la vendita dai 100 milioni nel 2016 ai 373 milioni nel 2020 [3]. Per questo motivo, questi sensori hanno suscitato un grande interesse nell'ambito sanitario e della ricerca, con lo scopo di monitorare i parametri vitali dei pazienti in modo continuo e, come in questo progetto, sfruttare i dati raccolti per studiare e migliorare il trattamento di diverse patologie, in particolar modo quelle croniche e legate all'età [4].



Figura 1: logo del progetto europeo BRAINTEASER [5]

I segnali raccolti per mezzo dei sensori indossabili sono segnali grezzi affetti da rumore di misura che potrebbe andare a intaccare le informazioni estratte dai dati. All'interno del progetto la necessità risulta essere proprio l'estrazione di informazioni utili a modelli di intelligenza artificiale di predizione. Quindi la pulizia dei dati grezzi per riuscire ad utilizzare al meglio le informazioni che forniscono diventa fondamentale. Questa porzione di lavoro rientra nel task 5.1 del progetto al quale partecipa anche UNIPD [6]. Nello specifico il progetto è diviso in work packages e a loro volta in task, ai quali collaborano i vari partecipanti. Essi hanno lo scopo di dividere l'intero lavoro tra i membri, con obiettivi più piccoli e scadenze più corte dell'intero progetto. Unendo i risultati dei vari work packages si otterrà il risultato finale e quindi il raggiungimento degli obiettivi di BRAINTEASER. Come brevemente accennato in precedenza il task 5.1 è quello relativo alla pipeline di preprocessing dei segnali provenienti da uno smartwatch commerciale e features extraction da questi ultimi[7]. Proprio in questa sezione specifica si posiziona questo lavoro di tesi [paragrafo 1.4]. Nelle prossime sezioni verranno approfonditi il sensore indossabile commerciale utilizzato [paragrafo 1.2] e lo stato dell'arte su algoritmi di filtraggio [paragrafo 1.3].

1.2 *Sensori indossabili*

1.2.1 *Garmin Vivoactive® 4*

Il sensore indossabile adottato per BRAINTEASER è il Garmin vivoactive® 4 [Figura 2], orologio da polso con funzionalità avanzate (smartwatch), in grado di monitorare i parametri vitali di chi lo indossa. Il costo attualmente dell'orologio è di €299,99.



Figura 2: Garmin vivoactive 4, smartwatch utilizzato per il progetto BRAINTEASER [8].

Le dimensioni del prodotto sono le seguenti: 45.1 x 45.1 x 12.8 mm, lo schermo ha un diametro di 33 mm e il peso del prodotto è 50,5 g. La durata della batteria può variare dalle sei ore (GPS con musica) agli otto giorni (funzionalità smartwatch) in base al livello di utilizzo [9]. Per gli scopi del progetto le funzionalità di interesse permetteranno una durata della batteria di qualche giorno, fattore determinante per non sovraccaricare il paziente con la necessità di ricaricare il dispositivo troppo frequentemente. Di seguito vengono riportati i segnali monitorabili dall'orologio [7], evidenziando quelle di interesse per il progetto: Steps, Calorie, Minuti di intensità, Piani saliti, Heart Rate (HR), Intervallo battuto-battito (BBI), Accelerometro, Stress, Lettura del pulsiossimetro, Monitoraggio della respirazione, Percentuale della batteria, Body Battery, Animazioni di allenamento sullo schermo, Monitoraggio ottico della frequenza cardiaca subacquea.

Molte altre caratteristiche, suddivise anche per sport, sono consultabili nella sezione delle specifiche sul sito online del Garmin [9].

1.3 Smoothing di segnali biomedici: stato dell'arte

Come primo passo in questo lavoro di tesi, si è fatta una ricerca in letteratura per capire lo stato dell'arte sui metodi che si andranno ad utilizzare, per capire se fossero stati fatti lavori simili e quali risultati fornivano questi ultimi.

In questa fase, si è rivolta la ricerca su due aspetti distinti: il primo aspetto è stato il filtraggio di segnali di battito cardiaco con smoothing bayesiano o altre tecniche; mentre il secondo è rivolto all'implementazione di smoothing bayesiano per capirne le procedure con cui questa tecnica viene messa in atto.

1.3.1 Filtraggio di segnale Heart Rate

Per quanto riguarda il primo aspetto, in letteratura non sono presenti molti lavori. Infatti, spesso i segnali di interesse per il filtraggio sono l'elettrocardiogramma [10] e il fotopletismogramma (PPG). In particolare, quest'ultimo segnale, che ultimamente viene molto investigato, è reso interessante dalla sua facile registrazione a basso costo e per lunghi periodi. Sono stati trovati diversi lavori nei quali il PPG viene filtrato per poi ricavare il tracciato heart rate [11], ed è stato anche sperimentato per rilevare malattie cardiache [12]. È emerso che in generale molti lavori di letteratura eseguono dei filtri nel dominio del tempo, della frequenza e in tempo-frequenza (*wavelet*) su questi segnali [13], ma non si sono trovati lavori che esplorino metodi di stima bayesiana.

Per quanto riguarda la ricerca sulla stima bayesiana, si sono individuati alcuni lavori eseguiti su segnali di monitoraggio del glucosio in continua (CGM). Ad esempio, l'articolo di De Nicolao et al. "Nonparametric input estimation in physiological systems: Problems, methods, and case studies" [14] spiega l'utilizzo della stima bayesiana e il suo corretto funzionamento nei problemi di mal-condizionamento, campionamento non uniforme e valutazione degli intervalli di confidenza. Inoltre, sullo stesso segnale sono stati fatti lavori più recenti che utilizzano lo stimatore bayesiano e il filtraggio di Kalman per poter filtrare il segnale in tempo reale [15] [16].

1.3.2 FLIRT

È stata effettuata anche una ricerca di implementazioni di queste tecniche, e il risultato principale è il pacchetto “FLIRT: A feature generation toolkit for wearable data” [17]. FLIRT è un pacchetto Python open-source [18]. Questo pacchetto implementa *data processing* e *features generation*. La *Figura 3: Steps implementati dal pacchetto FLIRT* [17] mostra la struttura e la pipeline in cui il pacchetto viene inserito. È evidente la somiglianza tra l’obiettivo di questo lavoro e le finalità del task 5.1 di BRAINTEASER. Per questo motivo, si è investigato il pacchetto per verificarne la compatibilità con lo scopo. Questa analisi ha evidenziato problemi nella compatibilità tra segnali in input richiesti dal progetto e quindi non si è proceduto in questa direzione.

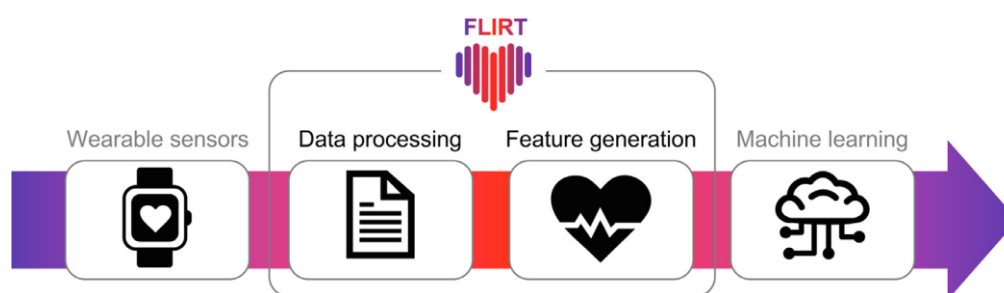


Figura 3: Steps implementati dal pacchetto FLIRT [17], i dati provenienti da sensori indossabili vengono elaborati da una pipeline in grado di processarli e generare delle features da utilizzare in algoritmi di machine learning.

1.4 Obiettivi e finalità del lavoro

L’obiettivo principale di questa tesi è quello di implementare, passando per vari steps, una pipeline in grado di ottenere un preprocessing dei segnali vitali (in particolare il segnale di battito cardiaco) che utilizzi una tecnica avanzata per eliminare il rumore di misura di segnali vitali provenienti da sensori indossabili in questo caso dal Garmin vivoactive4. Nello specifico si è deciso di utilizzare uno stimatore bayesiano in grado di ottenere un tracciato filtrato dei segnali [paragrafo 2.1.1].

La tecnica dello smoothing bayesiano verrà implementata in Python in un formato tale da permetterne l’utilizzo anche al di fuori di BRAINTEASER, rilasciando codice riutilizzabile in futuro anche su segnali diversi da quello di Heart Rate.

La tesi è stata svolta come accennato precedentemente in vari passaggi principali che sono: un'analisi preliminare dei dati [*paragrafo 2.2*] nella quale si sono visionati e valutati tutti i segnali provenienti dal sensore indossabile, scrittura di codice in linguaggio Python che implementasse lo smoothing bayesiano in ambiente simulato [*paragrafo 2.3*], successivamente ci si è focalizzato sulla implementazione della pipeline per il filtraggio del segnale Heart Rate [*capitolo 3*] ed infine vengono riportati i risultati con le valutazioni del lavoro[*capitolo 4*].

Capitolo 2

Smoothing Bayesiano

2.1 *Smoothing Bayesiano: cenni di teoria*

2.1.1 *Stima Bayesiana*

La stima bayesiana è la tecnica scelta in questa tesi per le sue caratteristiche implementative avanzate. Difatti, essa utilizza:

- inferenze statistiche dei segnali trattando questi ultimi come se fossero completamente aleatori;
- la conoscenza a priori dei segnali, la quale, anche se non troppo precisa, permette allo stimatore di ottenere ottimi risultati;
- un approccio *self-tunable*: lo stimatore è in grado di adattarsi autonomamente e non con controlli esterni, ai cambi delle caratteristiche dei tracciati, che avvengono nel corso dell'intera giornata. Infatti, trattando segnali di parametri vitali giornalieri si ha a che fare con segnali tempo-varianti, e non con la tipologia dei tempo-stazionari.

Lo stimatore con approccio bayesiano è una tecnica che si basa sull'utilizzo di conoscenze del legame tra la misura e la grandezza da stimare, e le informazioni probabilistiche a priori di quest'ultima (prior), che permettono allo stimatore di convergere più efficacemente al segnale da stimare.

In questa tesi ci si è posti nella condizione di modello di misure lineare nel caso gaussiano. Questa scelta è stata fatta per mancanza di riferimenti sul modello e sul tipo di rumore additivo non correlato con il dato. Per le condizioni descritte la formulazione matematica risulta essere:

$$\mathbf{y} = \mathbf{G} \cdot \mathbf{u} + \mathbf{v} \quad (2. 1)$$

dove \mathbf{y} è il vettore delle misure, \mathbf{G} = matrice di trasferimento, \mathbf{u} il vettore da stimare ed infine \mathbf{v} il rumore additivo.

Dato \mathbf{y} , il vettore che stima al meglio \mathbf{u} è il valor atteso a posteriori $\mathbf{E}[\mathbf{u}|\mathbf{y}]$, che sarà quindi considerato la stima di \mathbf{u} ($\hat{\mathbf{u}}$), per cui:

$$\hat{\mathbf{u}} = \mathbf{E}[\mathbf{u}|\mathbf{y}] \quad (2. 2)$$

Inoltre, $\hat{\mathbf{u}}$ è il vettore che minimizza il valor atteso della norma quadratica del vettore errore di stima ($\tilde{\mathbf{u}}$) calcolato come:

$$\tilde{\mathbf{u}} = \mathbf{u} - \hat{\mathbf{u}} \quad (2. 3)$$

Per questo motivo, lo stimatore è anche detto a minima varianza d'errore. Quest'ultima può essere calcolata come traccia della matrice di covarianza di $\tilde{\mathbf{u}}$ ($\Sigma\tilde{\mathbf{u}}$).

$$\mathbf{E}[|\tilde{\mathbf{u}}|^2] = \mathbf{E}[|\mathbf{u} - \hat{\mathbf{u}}|^2] = \mathbf{trace}(\Sigma\tilde{\mathbf{u}}) \quad (2. 4)$$

Le formulazioni che permettono di calcolare la stima di \mathbf{u} ($\hat{\mathbf{u}}$) e la matrice di covarianza ($\Sigma\tilde{\mathbf{u}}$) nelle condizioni sopra descritte sono le seguenti:

$$\hat{\mathbf{u}} = \boldsymbol{\mu}_u + (\mathbf{G}^T \cdot \Sigma_v^{-1} \cdot \mathbf{G} + \Sigma_u^{-1})^{-1} \cdot \mathbf{G}^T \cdot \Sigma_v^{-1} \cdot (\mathbf{y} - \boldsymbol{\mu}_y) \quad (2. 5)$$

$$\Sigma \hat{u} = (\mathbf{G}^T \cdot \Sigma_v^{-1} \cdot \mathbf{G} + \Sigma_u^{-1})^{-1} \quad (2.6)$$

Nel caso di valore atteso a priori nullo la (2.5) diventa:

$$\hat{u} = (\mathbf{G}^T \cdot \Sigma_v^{-1} \cdot \mathbf{G} + \Sigma_u^{-1})^{-1} \cdot \mathbf{G}^T \cdot \Sigma_v^{-1} \cdot \mathbf{y} \quad (2.7)$$

Dalla (2.5) si può apprezzare come opera lo stimatore bayesiano scomponendo i due elementi dell'equazione:

- In caso di informazioni a priori certa e dati del tutto inaffidabili, Σ_u tende a zero e Σ_v tende a infinito, perciò la stima di u diventa il valor atteso a priori perché i dati non aggiungono informazioni.

$$\hat{u} = \mu_u \quad (2.8)$$

- In caso di informazioni a priori del tutto incerta e dati molto affidabili, Σ_u tende a infinito e Σ_v tende a zero, perciò la stima di u diventa la stima ai minimi quadrati.

$$\hat{u} = (\mathbf{G}^T \cdot \Sigma_v^{-1} \cdot \mathbf{G})^{-1} \cdot \mathbf{G}^T \cdot \Sigma_v^{-1} \cdot \mathbf{y} \quad (2.9)$$

- In tutti i casi intermedi lo stimatore opera facendo una mediazione ottima tra il dato e la sua stima a priori.

Nel contesto bayesiano in cui si sta operando i vettori \mathbf{u} e \mathbf{v} sono aleatori, considerati tra loro non correlati e con valore atteso a priori nullo. Inoltre, si dispone delle loro matrici di covarianza che rispettivamente sono Σ_u e Σ_v . Di seguito vengono riportate le loro formulazioni.

$$\Sigma_u = \lambda^2 \cdot (\mathbf{F}^T \cdot \mathbf{F})^{-1} \quad (2.10)$$

$$\Sigma_v = \sigma^2 \cdot B$$

(2. 11)

Gli elementi della matrice di covarianza di u (2. 10) λ^2 e F vengono descritti accuratamente nel paragrafo 2.1.2 con λ^2 che risulta essere, nel caso specifico, una varianza di un processo di rumore bianco e F una matrice che può assumere diverse forme in base a quante volte si intende integrare il modello di rumore bianco.

La matrice di covarianza di v (2. 11) contiene i due elementi σ^2 e B , che nel caso di rumore bianco (situazione ipotizzata in questa tesi) risultano essere rispettivamente la varianza del rumore e una matrice identità, difatti non c'è correlazione tra gli elementi del rumore e quindi la matrice Σ_v avrà elementi diversi da zero solamente sulla diagonale principale.

Sostituendo la (2. 10) e la (2. 11) nella (2. 7) si ottiene la seguente formulazione utilizzata per calcolare la stima di u (\hat{u}) nel contesto bayesiano.

$$\hat{u} = (G^T \cdot B^{-1} \cdot G + \gamma^0 \cdot F^T \cdot F)^{-1} \cdot G^T \cdot B^{-1} \cdot y$$

(2. 12)

$$\text{con } \gamma^0 = \frac{\sigma^2}{\lambda^2}$$

(2. 13)

Dove λ^2 e σ^2 sono rispettivamente le varianze del modello di rumore bianco in input al random walk e la varianza del rumore di misura additivo.

Grazie a questo approccio, inoltre, è possibile particolarizzare anche la matrice di covarianza dell'errore di stima ($cov(\tilde{u})$). Partendo dalla (2. 6) si ottiene:

$$cov(\tilde{u}) = \sigma^2 \cdot (G^T \cdot B^{-1} \cdot G + \gamma^0 \cdot F^T \cdot F)^{-1}$$

(2. 14)

La matrice di covarianza di u (2. 14) è una matrice quadrata di dimensione pari alla lunghezza del vettore \hat{u} sulla cui matrice diagonale principale si trovano le varianze dell'errore di stima.

Grazie a questa informazione è possibile sviluppare gli intervalli di confidenza (I.C.) intorno alla stima di u calcolati come:

$$I.C. = \hat{u}_i \pm \sqrt{\text{cov}(\tilde{u})_{ii}} \quad (2.15)$$

2.1.2 Random walk e Random walk integrato

Poiché non si dispone di un preciso modello da poter utilizzare come informazione a priori, si è deciso di sviluppare il modello di rumore bianco semplice (random walk) oppure pluri-integrato (random walk pluri-integrato). Questo modello fornisce solo un aspetto qualitativo del segnale e cioè la sua regolarità, ma non aggiunge informazioni in ambito quantitativo. Quante più volte esso sarà integrato, tanto più esso risulterà regolare.

La formulazione generale del random walk è la seguente:

$$\mathbf{u}_k = \mathbf{u}_{k-1} + \mathbf{w}_k \quad (2.16)$$

Con $\mathbf{w}_k \in \mathbf{WN}(\mathbf{0}, \lambda^2)$.

Questa formulazione mostra che il valore allo step k vale quanto il valore dello step precedente ($k - 1$) con aggiunta una incertezza (\mathbf{w}_k), valore randomico prelevato da un processo di rumore bianco con media nulla e varianza λ^2 .

La formulazione (2.16) può essere scritta in forma matriciale come mostrato nella (2.17)

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & \dots & \dots & 0 \\ 0 & -1 & \dots & \dots & 0 \\ 0 & 0 & \dots & 1 & 0 \\ 0 & \dots & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \end{bmatrix}$$

$$\mathbf{F} \cdot \mathbf{u} = \mathbf{w}$$

(2.17)

Considerando $\mathbf{F} = \Delta^m$, se $m = 1$ allora la matrice \mathbf{F} , che è una matrice triangolare inferiore di Toeplitz, risulta essere uguale a Δ ed assume la seguente forma:

$$\Delta = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & \dots & \dots & 0 \\ 0 & -1 & \dots & \dots & 0 \\ 0 & 0 & \dots & 1 & 0 \\ 0 & \dots & 0 & -1 & 1 \end{bmatrix}$$

(2. 18)

Se $m > 1$ allora il random walk risulterà pluri-integrato.

Per esempio, con $m = 2$ si otterrà il modello di random walk integrato, per il quale la formulazione risulta essere:

$$\mathbf{u}_k - 2 \cdot \mathbf{u}_{k-1} + \mathbf{u}_{k-2} = +\mathbf{w}_k$$

(2. 19)

e la matrice \mathbf{F} diverrà:

$$\mathbf{F} = \Delta^2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & \dots & \dots & 0 \\ -1 & 2 & \dots & \dots & 0 \\ 0 & -1 & \dots & 1 & 0 \\ 0 & \dots & \dots & 2 & 1 \end{bmatrix}$$

(2. 20)

Da ciò è possibile ottenere la formulazione della matrice di covarianza di questo processo, la quale dopo alcuni passaggi algebrici risulta essere l'equazione (2. 10): $\Sigma_{\mathbf{u}} = \lambda^2 \cdot (\mathbf{F}^T \cdot \mathbf{F})^{-1}$.

In *Figura 4* è possibile visionare come la regolarità dei processi di rumore bianco, random walk e random walk integrato cambi in base all'ordine di integrazione utilizzato. In questo lavoro questo valore è settato a due, quindi si utilizza un modello di random walk integrato.

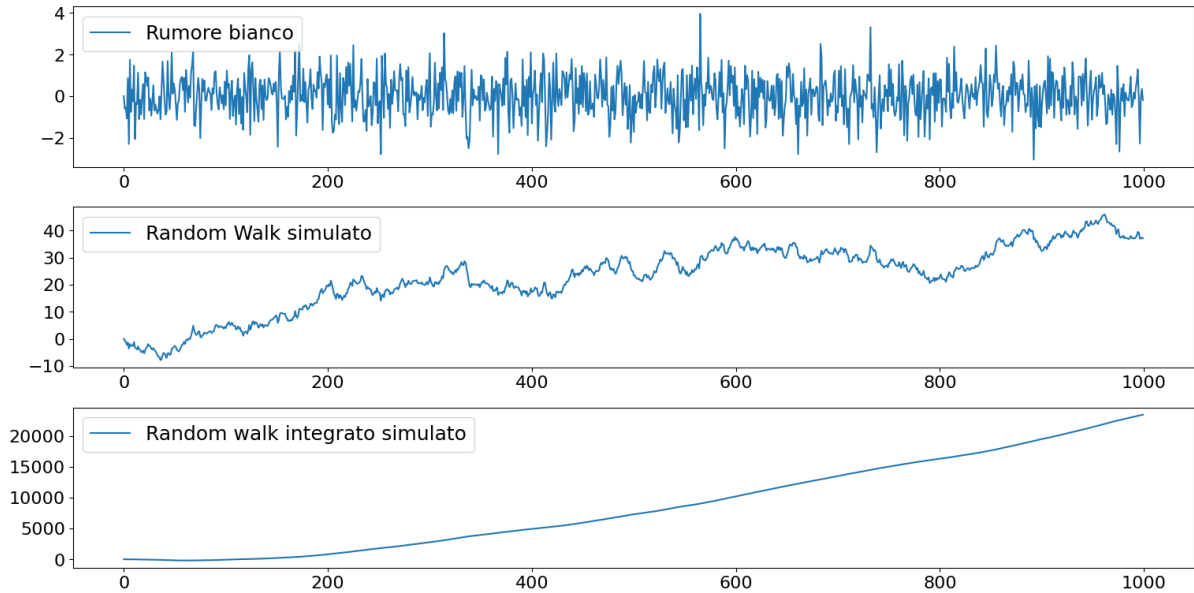


Figura 4: in alto è riportato un processo di rumore bianco, al centro un modello di random walk simulato e in basso un modello di random walk integrato simulato. Tutti i processi sono composti da mille campioni.

2.1.3 criterio di consistenza

La stima di u dipende da gamma ottimo (γ^0) per cui $\hat{u}(\gamma^0)$. Questo parametro come descritto precedentemente nella formulazione (2. 13) risulta essere un rapporto di varianze. È rilevante fornire una metodologia per poter calcolare il gamma ottimo, grazie al quale poi verranno forniti facilmente gli intervalli di confidenza.

La procedura prevede di definire la somma dei quadrati dei residui pesati (**WRSS**), la somma dei quadrati della stima pesata (**WESS**) e una funzione q che dipende anch'essa da γ^0 detta gradi di libertà equivalenti, definite come segue:

$$WRSS(\gamma^0) = (\mathbf{y} - \mathbf{G} \cdot \hat{\mathbf{u}})^T \cdot \mathbf{B}^{-1} \cdot (\mathbf{y} - \mathbf{G} \cdot \hat{\mathbf{u}}) \tag{2. 21}$$

$$WESS(\gamma^0) = \| \mathbf{F} \cdot \hat{\mathbf{u}} \|^2 = \hat{\mathbf{u}}^T \cdot \mathbf{F}^T \cdot \mathbf{F} \cdot \hat{\mathbf{u}} \tag{2. 22}$$

$$q(\gamma^0) = \text{trace} \left(\mathbf{B}^{-\frac{1}{2}} \cdot \mathbf{G} \cdot (\mathbf{G}^T \cdot \mathbf{B}^{-1} \cdot \mathbf{G} + \gamma^0 \cdot \mathbf{F}^T \cdot \mathbf{F})^{-1} \cdot \mathbf{G}^T \cdot \mathbf{B}^{-\frac{1}{2}} \right) \quad (2.23)$$

La matrice $\mathbf{B}^{-\frac{1}{2}}$ viene definita come segue:

$$\mathbf{B}^{-\frac{1}{2}} = \begin{bmatrix} \frac{1}{\sqrt{b_{11}}} & \frac{1}{\sqrt{b_{12}}} & \dots & \dots & \frac{1}{\sqrt{b_{1n}}} \\ \frac{1}{\sqrt{b_{21}}} & \frac{1}{\sqrt{b_{22}}} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \frac{1}{\sqrt{b_{n1}}} & \dots & \dots & \dots & \frac{1}{\sqrt{b_{nn}}} \end{bmatrix} \quad (2.24)$$

Inoltre, vengono definite due uguaglianze che permettono di ricavare i criteri di consistenza per la stima del gamma ottimo e sono le seguenti:

$$E[WRSS] = \sigma^2 \cdot \{n - q(\gamma^0)\} \quad (2.25)$$

$$E[WESS] = \frac{\sigma^2}{\gamma^0} \cdot q(\gamma^0) \quad (2.26)$$

Poiché WRSS e WESS in ambito bayesiano sono variabili aleatorie si può calcolare il loro valore atteso (2.25), (2.26). Questo nel caso reale non è possibile, in quanto si dispone solamente di una realizzazione; quindi, si accetta il valore di WRSS e WESS come loro valor medio. Di conseguenza grazie a queste equazioni scaturiscono tre criteri di consistenza descritti dalle seguenti formulazioni (2.27), (2.28), (2.29).

$$WESS(\gamma) = \sigma^2 \cdot \frac{q(\gamma)}{\gamma} \quad (2.27)$$

$$WRSS(\gamma) = \sigma^2 \cdot \{n - q(\gamma)\} \quad (2.28)$$

$$\frac{WRSS(\gamma)}{n - q(\gamma)} = \gamma \cdot \frac{WESS(\gamma)}{q(\gamma)} \quad (2.29)$$

Il terzo criterio di consistenza (2.29) a differenza degli altri ha il vantaggio di stimare il gamma ottimo senza essere a conoscenza della varianza dell'errore additivo (σ^2); caso compatibile con il presente lavoro e dunque è il criterio scelto.

Una volta raggiunto il gamma ottimo è possibile calcolare anche la σ^2 con la seguente formula:

$$\sigma^2 = \frac{WRSS(\gamma)}{n - q(\gamma)} \quad (2.30)$$

2.2 *Analisi preliminare*

2.2.1 *Python*

In questo lavoro l'unico linguaggio di programmazione utilizzato è Python [18]. Questa condizione è l'unico vincolo imposto da BRAINTEASER, in quanto l'infrastruttura del progetto in cui la pipeline verrà inserita, utilizza questo linguaggio. Python ha attualmente numerosi vantaggi rispetto ad altri linguaggi di programmazione e il suo utilizzo sta crescendo in modo esponenziale. Tra i più importanti ci sono il fatto che è opensource e multiplatforma, inoltre è possibile consultare Python Package Index (PYPI) dal quale si possono scaricare numerose librerie per il libero utilizzo [19]. Infine, grazie alla sua estrema versatilità è un linguaggio di programmazione molto sfruttato nell'ambito dei big data: analisi di dati, intelligenza artificiale e machine learning [20].

La versione di Python utilizzata per questa tesi è la 3.10.8

2.2.2 *analisi preliminare*

Come primo steps implementativo in questo lavoro di tesi è stata fatta un'analisi preliminare dei dati. Essi sono stati raccolti per mezzo di Garmin vivoactive 4 [8] indossati dai soggetti da monitorare. Il dato viene esportato dal sensore wearable e poi collezionato dall'infrastruttura del progetto, da qui prelevato e utilizzato sotto forma di file *JSON* (JavaScript Object Notation)[21], formato di testo per la strutturazione di dati, molto diffuso per la sua facile interpretazione[22]. In appendice [paragrafo 5.1] è possibile consultare un esempio di file *JSON* utilizzato in questo lavoro.

In questa fase ci si è posto il problema di importare e leggere un file *JSON* all'interno dell'ambiente Python, con l'obiettivo di visionare il tipo di dati, e cercare di capire alcuni aspetti dei tracciati che saranno trattati in seguito. Tra questi aspetti l'attenzione è stata posta in particolare: nel riconoscere trend o pattern, nel capire l'entità dei campioni mancanti, la frequenza con cui questi ricorrono all'interno dei tracciati e riconoscere empiricamente un rumore additivo il più bianco possibile.

I tracciati forniti dal sensore presi in considerazione sono i seguenti: HR (heart rate), Steps, Stress, SpO2 e Respiration.

In *Figura 5* è possibile vedere un tracciato d'esempio dei segnali sopra indicati. L'immagine contiene sulla prima riga tre grafici (HR, Steps, Stress) che hanno in asse x riportate le ore della giornata, questo per permettere di capire a colpo d'occhio se il tracciato è completo oppure il sensore non è stato indossato per periodi all'interno del giorno. I due grafici sulla seconda riga (SpO2 e Respiration) sono dati prelevati dal sensore solamente durante la fase di sonno che solitamente avviene a cavallo tra due giornate e vengono riportati all'interno del file json con un offset negativo. Quest'ultimo indica di quanti secondi prima della mezzanotte va traslato l'asse dei tempi di questi due grafici. Difatti in figura è possibile vedere che questi segnali hanno nell'asse delle ascisse valori che vanno da negativi a positivi passando per lo 0 che risulta essere l'ora 00:00.

Inoltre, è stata sviluppata una funzione denominata `findHole.py` con l'obiettivo di capire se ci sono valori mancanti nel tracciato non direttamente visibili ad occhio in quanto di piccola entità. La funzione ha come ingressi: il tempo di campionamento, il vettore dei tempi e il vettore dei valori e restituisce in uscita il tempo in cui inizia una sezione di dati mancanti, lo stesso tempo trasformato in ore, il valore dell'asse y in quel tempo e il numero di campioni

mancanti. Grazie a queste informazioni è stato possibile disegnare sui cinque tracciati i marcatori di diverso colore [Figura 5] che indicano il punto di inizio degli hole nei tracciati.

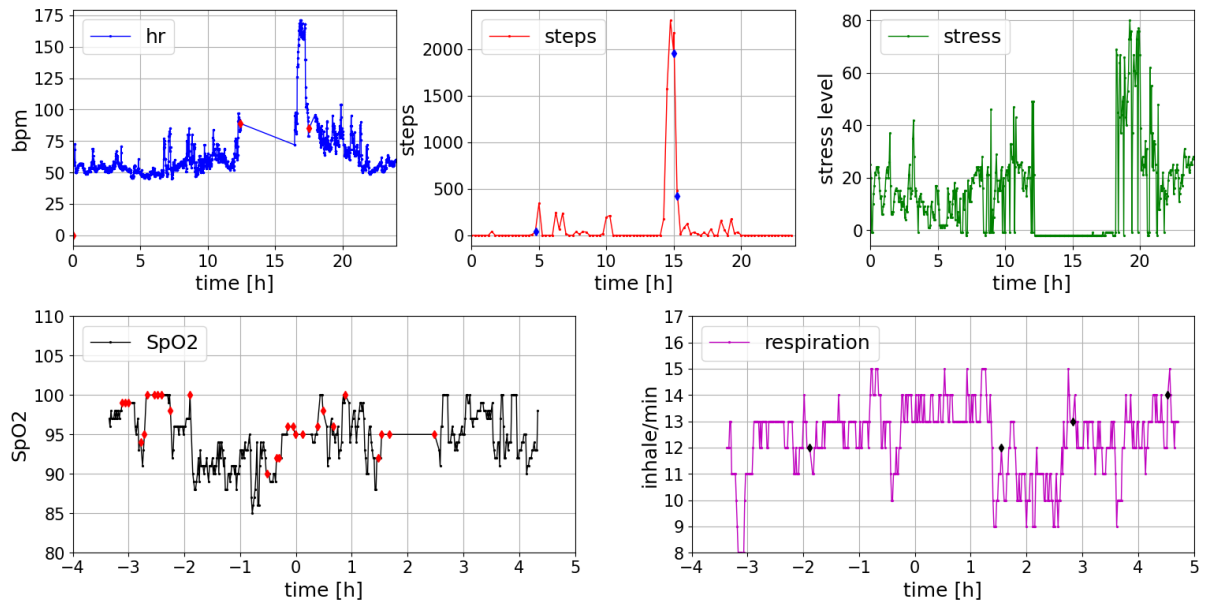


Figura 5: In alto vengono riportati da sx a dx i segnali di Hr, Steps e Stress, in basso nello stesso ordine quelli di SpO2 e Respiration. In tutti i grafici vengono riportati dei marcatori di colore diverso dal tracciato i punti in cui iniziano dei campioni mancanti.

Dopo un’analisi sui segnali di cinquanta giornate di diversi soggetti, è emerso che il tracciato con meno anomalie, più dati presenti ed empiricamente più consono alla stima bayesiana è quello di battito cardiaco (HR). Per questo motivo si è deciso di concentrarsi solamente su questo. Di seguito vengono riportare alcune tracce rappresentative del segnale.

Come è possibile vedere dai tracciati riportati in *Figura 6*, alcune giornate non presentano campioni mancanti (tracciato al centro), mentre in altri casi ci sono delle zone in cui questi sono presenti anche in modo molto consistente (tracciato in basso). Inoltre, è visibile l’innalzamento del battito per un periodo di tempo confinato legato probabilmente all’attività fisica svolta (tracciato in basso). Un ulteriore commento emerso, è che la variabilità del segnale nei tracciati di HR varia decisamente nell’arco della giornata si nota una differenza importante, per esempio, tra veglia e riposo in (tracciato in alto). Per questo motivo si è deciso

di segmentare il tracciato così da rendere il più uniforme possibile la varianza del modello di rumore di misura.

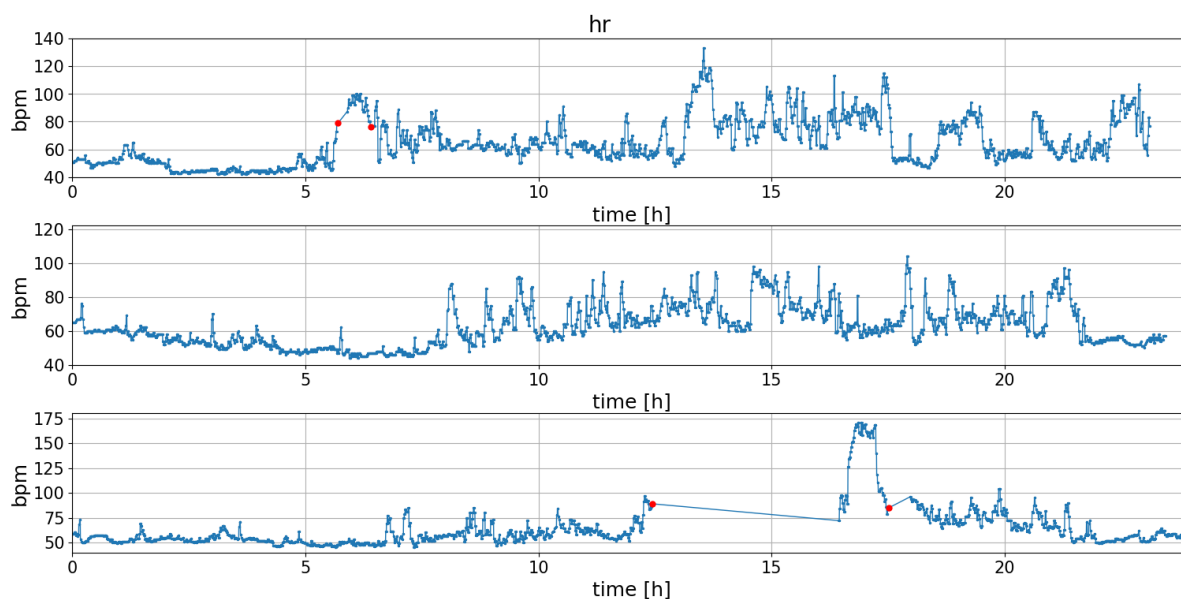


Figura 6: tre esempi di segnale HR, con marcatori rossi in presenza di campioni mancanti. Nel primo e secondo riquadro si apprezza la diversa variabilità tra sonno e veglia, mentre in basso si vede una zona consistente di campioni mancanti e una presunta attività fisica.

2.3 Applicazione di smoothing Bayesiano in ambiente simulato

Dopo aver svolto attentamente un'analisi preliminare dei dati [paragrafo 2.2] si è implementato il codice in ambiente Python che eseguisse lo smoothing bayesiano. Per poterlo testare e verificarne il corretto funzionamento, si è eseguito il codice in ambiente controllato. Questo permette di verificare se lo stimatore approssima correttamente il segnale, e di conseguenza ne fa un corretto denoising, che è l'obiettivo del lavoro.

2.3.1 Ground truth, rumore, segnale rumoroso

Sono stati generati tre segnali simulati su cui svolgere i test del codice sviluppato. In seguito, vengono descritti brevemente:

- *Ground truth*: è la somma di due distribuzioni gaussiane con stessa standard deviation, ma con due medie differenti, che infine, per aumentarne l'ampiezza, è stata moltiplicata per 40. I campioni che compongono questo segnale sono cento [Figura 7].
 - $\mathcal{N}(3,1)$ distribuzione normale con media = 3 e $std = 1$
 - $\mathcal{N}(7,1)$ distribuzione normale con media = 7 e $std = 1$

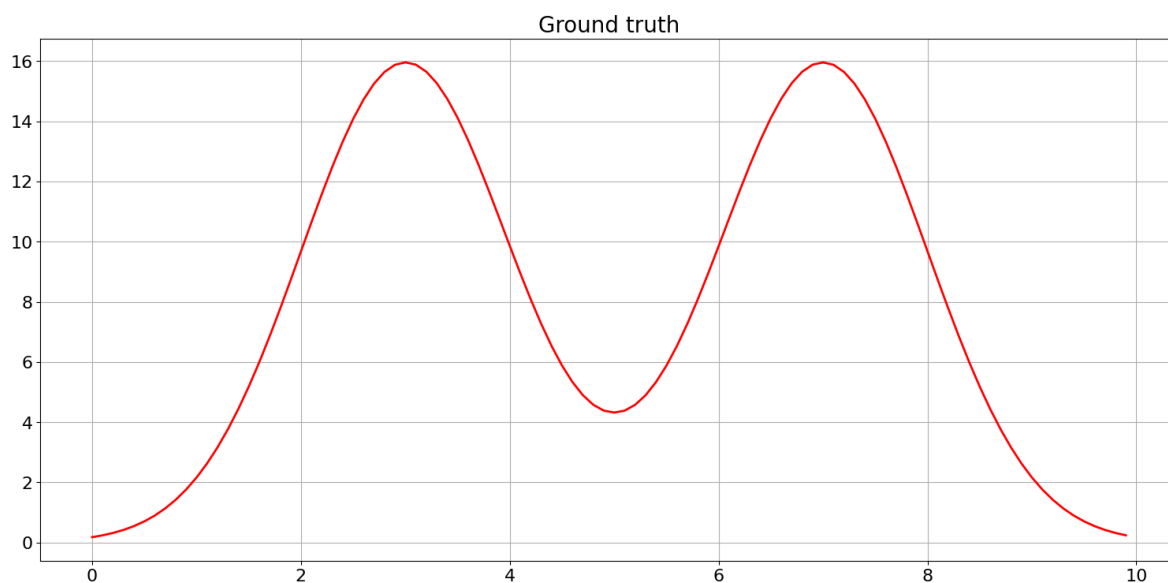


Figura 7: Ground truth composto da cento campioni

- *Noise*: processo di rumore bianco simulato, cioè valori presi casualmente da una distribuzione normale con media nulla e varianza uguale a uno $\mathcal{N}(0,1)$. Anche in questo caso il vettore è composto da cento campioni [Figura 8].

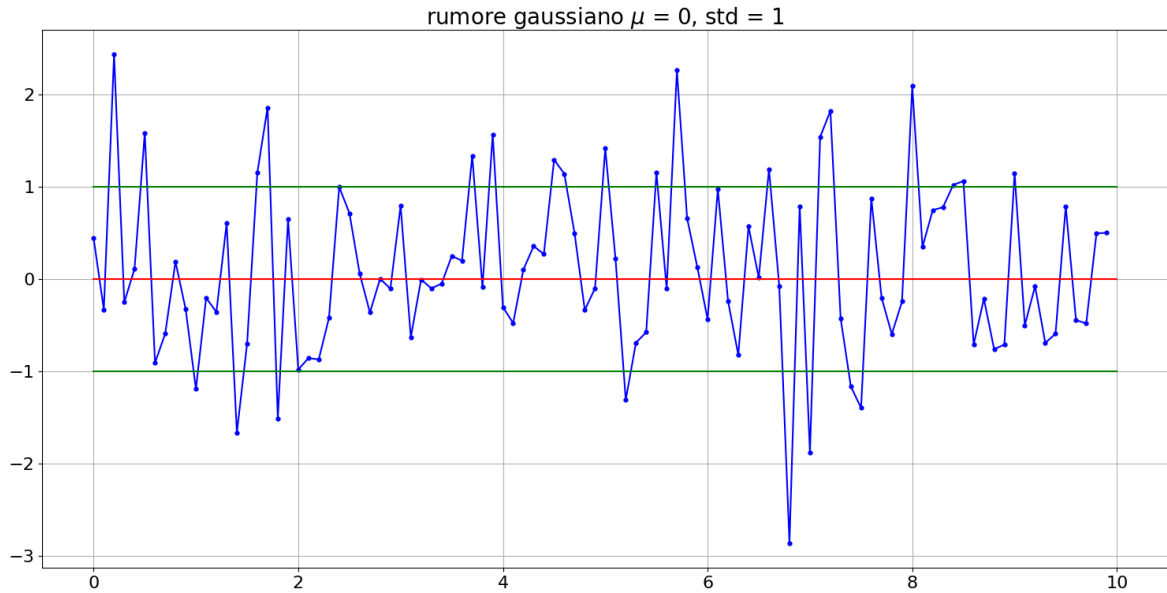


Figura 8: processo di rumore bianco con media nulla e varianza unitaria

Infine, i due segnali sopra presentati vengono sommati tra loro e originano il tracciato che verrà utilizzato come input al filtro bayesiano [Figura 10].

In Figura 9 è possibile apprezzare l'ampiezza della ground truth rispetto a quella del rumore. Per costruzione quest'ultima ha andamento variabile, con picchi e nadir, con zone ad alte e basse frequenze. Inoltre, permette di avere un rapporto segnale rumore (SNR) variabile: zone in cui il segnale assume valori minori del rumore (inizio e fine del tracciato) e zone dove il segnale ha ampiezza molto maggiore del rumore (in prossimità dei due picchi).

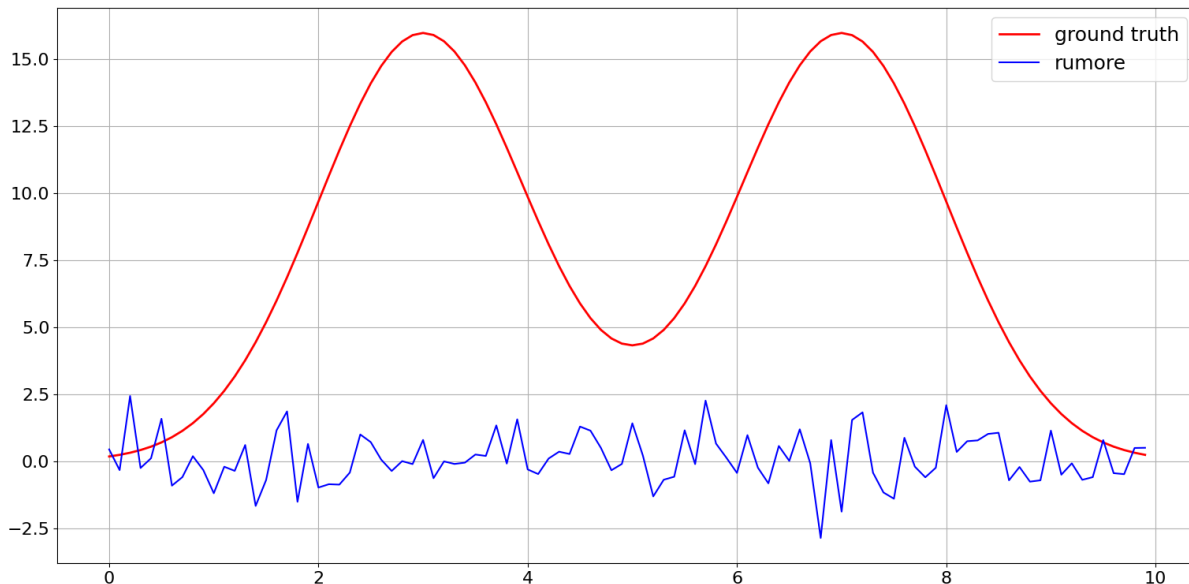


Figura 9: ampiezza di ground truth e rumore a confronto

In *Figura 10* viene mostrato il segnale rumoroso risultante (ground truth + rumore) in riferimento alla ground truth. Quest’ultimo è il segnale di riferimento che si vuole ottenere in uscita allo stimatore in quanto si vuole eliminare il rumore additivo, quando come input si fornisce il segnale rumoroso.

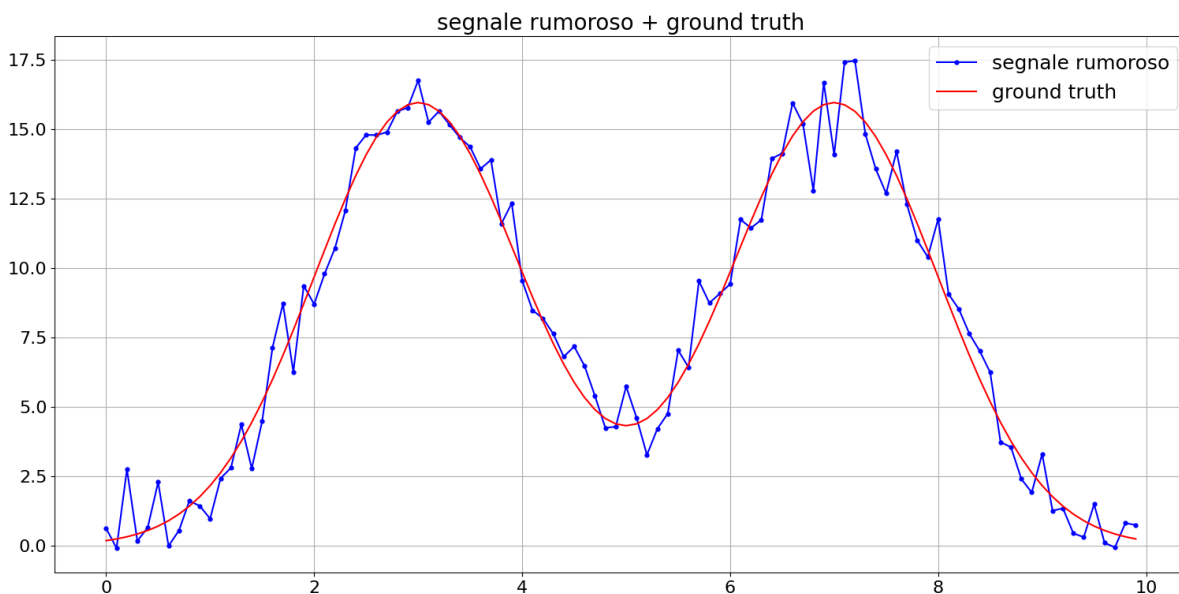


Figura 10: ground truth e segnale rumoroso a confronto

2.3.2 Sviluppo del codice per lo smoothing bayesiano

La funzione implementata in ambiente Python è quella che permette di eseguire tutta la teoria della stima bayesiana descritta nel paragrafo 2.1. Essa riceve in ingresso cinque variabili:

- **ts**: vettore contenente i tempi del segnale da filtrare (griglia reale);
- **ys**: vettore contenente il segnale da filtrare;
- **B**: matrice utilizzata per la formula (2. 12) in questo lavoro risulta essere sempre una matrice identità di dimensione pari al vettore ys e ts;
- **tv**: vettore dei tempi della griglia virtuale, che può in ambito generale essere più fitto del vettore ts, contenere i valori di ts, ma anche non contenerli;
- **m**: ordine della integrazione del rumore bianco in quanto si utilizza un modello a priori di rumore bianco integrato.

Come *output* la funzione fornisce sei variabili:

- **uhat** [\hat{u}]: vettore contenente il segnale filtrato derivante dalla formula (2. 12);
- **res**: vettore dei residui per definizione differenza tra il segnale da filtrare e segnale filtrato;
- **gamma** [γ^0]: gamma ottimo rappresentante il rapporto tra varianze come mostra la (2. 13);
- **itergamma**: numero di iterazioni compiute per giungere al gamma ottimo;
- **cov_u** [$cov(\hat{u})$]: matrice di covarianza dell'errore di stima (2. 14);
- **sigma2** [σ^2]: varianza dell'errore di misura calcolato come nella (2. 30).

Segue una breve descrizione dei passi svolti dalla funzione. Come primo step essa crea una matrice G che inizialmente è identità con dimensioni pari a tv. In seguito, vengono cancellate delle righe per farla corrispondere alle dimensioni di ts. Le righe eliminate sono quelle dove nella griglia virtuale si ha il dato, ma in quella reale no. La matrice G risultante è la matrice di trasferimento della formula (2. 1) che permette di adeguare le dimensioni di u a quelle di y ed è definita quasi Toeplitz. Nella figura sottostante [Figura 11] si può vedere come viene eseguito questo passaggio.

$$\begin{array}{cccccccc}
 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 \hline
 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 \hline
 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
 \hline
 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 \hline
 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 \hline
 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
 \end{array}$$

↓

$$\begin{array}{cccccccc}
 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
 \end{array}$$

Figura 11: passaggio implementativo per la creazione della matrice G

In seguito, si genera la matrice F come descritto nel *paragrafo 2.1.2* in base al valore di m fornito alla funzione. Come ultimo passaggio si entra in un ciclo implementato per far convergere la stima al gamma ottimo utilizzando il metodo di discrepanza e il criterio di consistenza descritto sopra [*paragrafo 2.1.3*]. La formula utilizzata per il calcolo della stima di u (2. 31) è una semplificazione della (2. 12) che permette di ridurre il costo computazionale dell'intero metodo in quanto si devono invertire matrici più piccole rispetto alle originali, in particolare con dimensione pari alla griglia reale e non virtuale.

$$\hat{u} = (F^T \cdot F)^{-1} \cdot G^T \cdot (G \cdot (F^T \cdot F)^{-1} \cdot G^T + B \cdot \gamma^o)^{-1} \cdot y \quad (2. 31)$$

Giunti al gamma ottimo, come ultimo passaggio grazie alle formule (2. 14), (2. 30), si calcola $cov(\hat{u})$ e σ^2 .

Finite di implementare la funzione sopra descritta, quest'ultima viene testata sui dati simulati descritti precedentemente. Di seguito vengono riportati i grafici per mostrare il corretto funzionamento della stessa. In questo caso come griglia virtuale dei tempi si è creato un vettore di mille campioni equidistanti mentre il tracciato di segnale rumoroso è composto da cento campioni [*paragrafo 2.3.1*], questo per verificare anche il corretto funzionamento di tv e ts con lunghezza differenti. In *Figura 12* vengono riportati: Ground truth, Segnale rumoroso, Segnale filtrato, Intervallo di confidenza (I.C.).

È importante segnalare che il tracciato filtrato è praticamente sovrapposto alla *ground truth* e anche quando si discosta leggermente come per esempio nei picchi e in coda al tracciato l'intervallo di confidenza racchiude sempre il segnale vero. Inoltre, il *denoising* avviene correttamente in ogni regione del tracciato anche quando l'energia del rumore sovrasta quella del segnale.

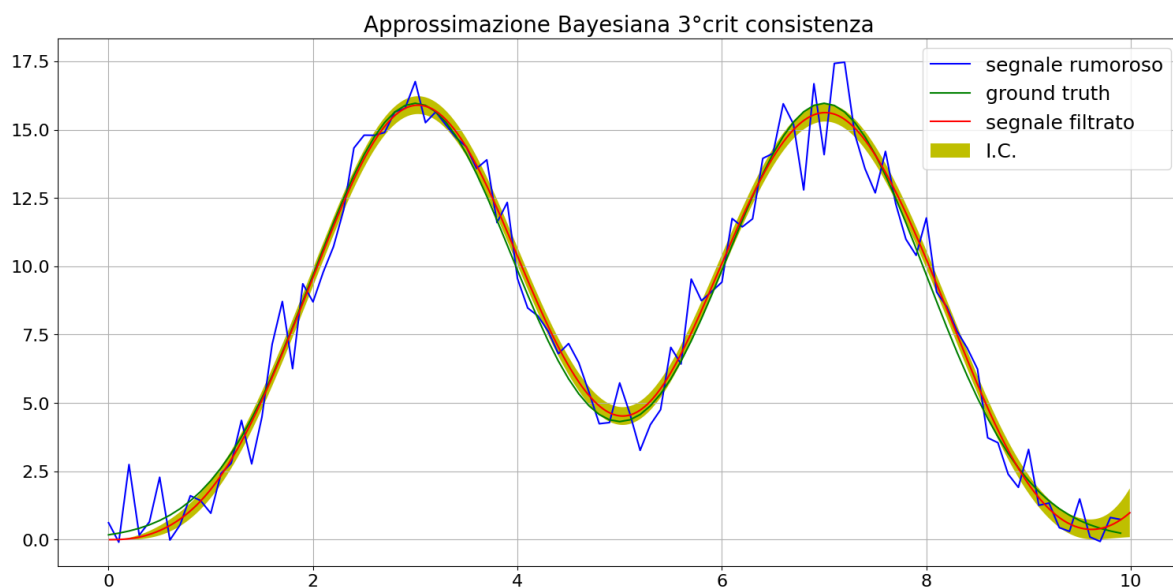


Figura 12: *ground truth, segnale rumoroso, segnale filtrato con relativo intervallo di confidenza a confronto. Corretto funzionamento del filtraggio bayesiano*

In *Figura 13* sono riportati i residui assoluti e i residui normalizzati.

- **Residui assoluti**

$$\mathbf{res} = \mathbf{y}_s - \mathbf{G} \cdot \hat{\mathbf{u}}$$

(2. 32)

Essi sono paragonati all'errore di misura, infatti se la stima fosse perfetta quindi riproducesse esattamente la *ground truth* la differenza tra segnale rumoroso e stima sarebbe esattamente il rumore di misura. In questo caso si vede come i residui e l'errore siano molto simili. Questo permette di concludere che è stato eseguito un ottimo *denoising*.

- **Residui normalizzati**

$$\text{res}_n = \frac{\text{res}}{\sigma} = \frac{y_s - G \cdot \hat{u}}{\sigma}$$

(2.33)

Con σ che è la radice quadrata del sigma2 in uscita alla funzione descritta nel paragrafo 2.3.2 ***Sviluppo del codice per lo smoothing bayesiano.***

In questo caso i residui vengono controllati essere limitati nella banda ± 1 . Nella situazione specifica poiché la varianza del rumore di misura era stata impostata a uno, ci si aspetta che il sigma2 fornito dalla funzione rispecchi questo valore. In particolare, il valore risulta essere 1.024, perfettamente paragonabile a quello impostato con un errore relativo percentuale di 2.45%.

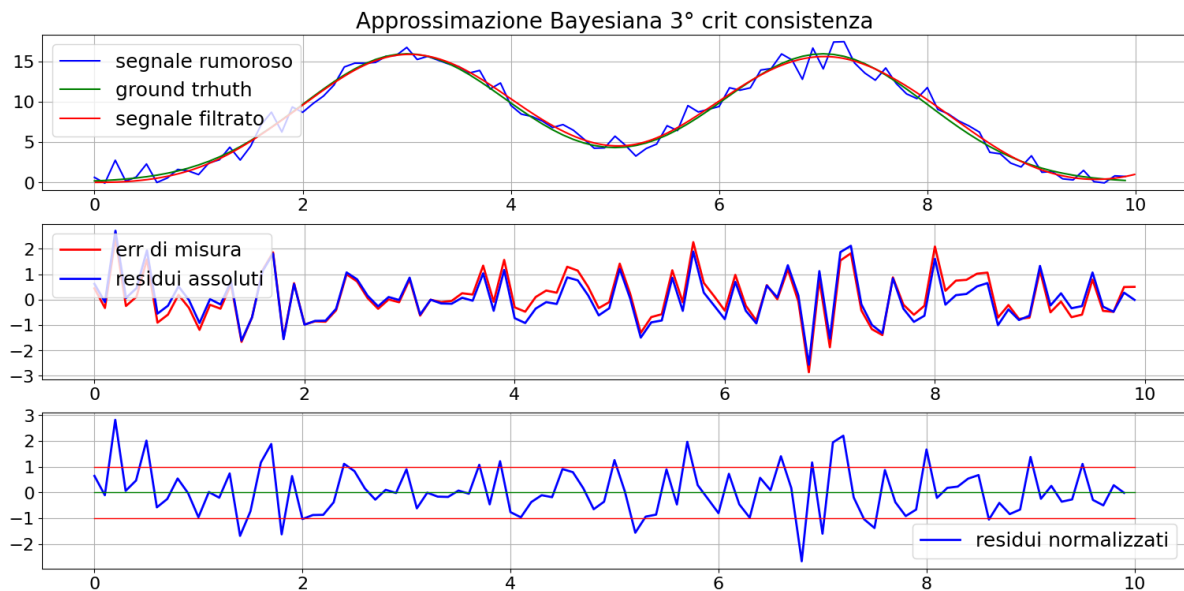


Figura 13: primo grafico mostra il segnale filtrato, segnale rumoroso e ground truth a confronto [Figura 12], secondo grafico mette in relazione l'errore di misura con i residui assoluti, il terzo grafico mostra i residui normalizzati rispetto l'intervallo ± 1

2.3.3 Gestione dei missing samples

Come visionato nel capitolo sull'analisi preliminare [paragrafo 2.2] utilizzando i dati di HR si avranno dei campioni mancanti e quindi si dovrà gestire una griglia reale non uniforme, mentre invece in uscita allo stimatore si vuole ottenere una griglia uniforme. Anche in questo

caso si è proceduto con un test in ambiente controllato simile al precedente, la cui unica differenza è che il segnale rumoroso da filtrare è stato modificato eliminando dei campioni.

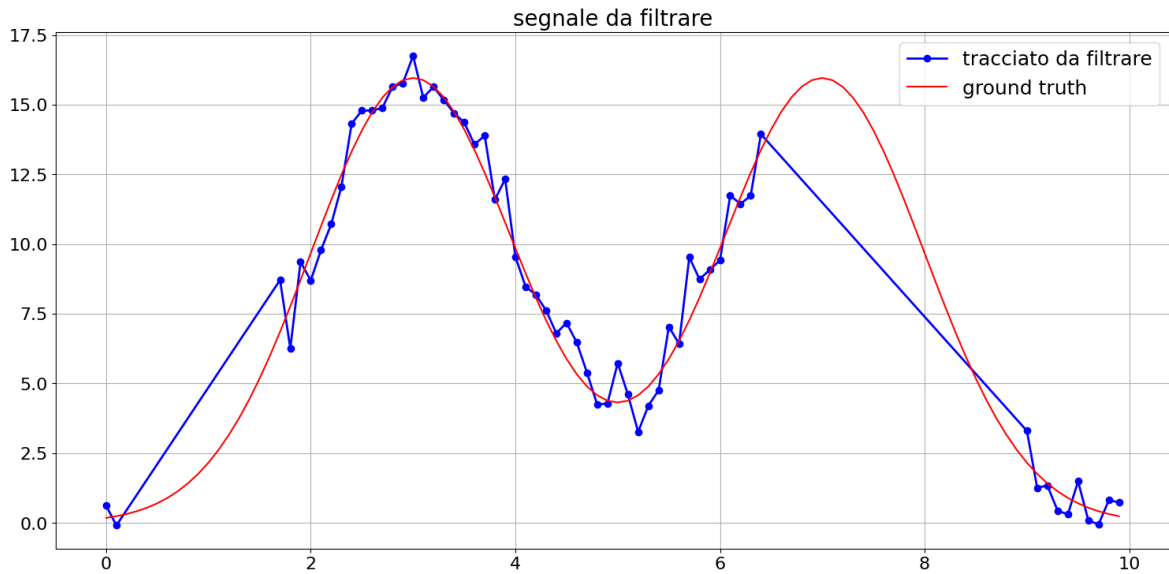


Figura 14: *ground truth in confronto al segnale rumoroso defezionato di 40 campioni mancanti in due zone distinte*

Per l'occasione è stata creata una funzione ad hoc che permette di eliminare dei campioni da un vettore inserendo come input: asse dei tempi (asse x), vettore contenente il segnale (asse y), il valore dell'asse x da cui inizia il tratto da eliminare, lunghezza del tratto da eliminare in numero di campioni. In uscita a questa procedura si ottiene un vettore con due zone di campioni mancanti il primo di 15 campioni a partire dal campione 2 e il secondo di 25 a partire dal campione 50, come mostrato in *Figura 14*. Con questa configurazione bisogna considerare che su 100 campioni di partenza ne rimangono solamente 60 quindi l'informazione in ingresso allo stimatore è molto ridimensionata rispetto a quella di partenza e in aggiunta è anche rumorosa (l'obiettivo rimane comunque il denoising). In *Figura 15* e *Figura 16* vengono riportati i risultati.

Il risultato che si ottiene dalla funzione è comunque ottimo; quindi, la presenza di missing samples non arreca particolare danno alla stima. Inoltre, è importante notare che lo stimatore percepisce la mancanza di informazione nei punti sopra descritti e quindi, essendo la stima più incerta in queste zone, aumenta l'intervallo di confidenza. Anche per questa simulazione i residui sia assoluti, confrontati all'errore di misura, sia quelli normalizzati suggeriscono un ottimo filtraggio del tracciato. Inoltre, si osserva come i residui non sono presenti nelle

porzioni di grafico corrispondente ai campioni mancanti del tracciato da filtrare, questo è dovuto al fatto che per definizione il residuo non si può calcolare se non si ha il valore del campione del tracciato originale (y_s) [formulazioni (2. 32), (2. 33)].

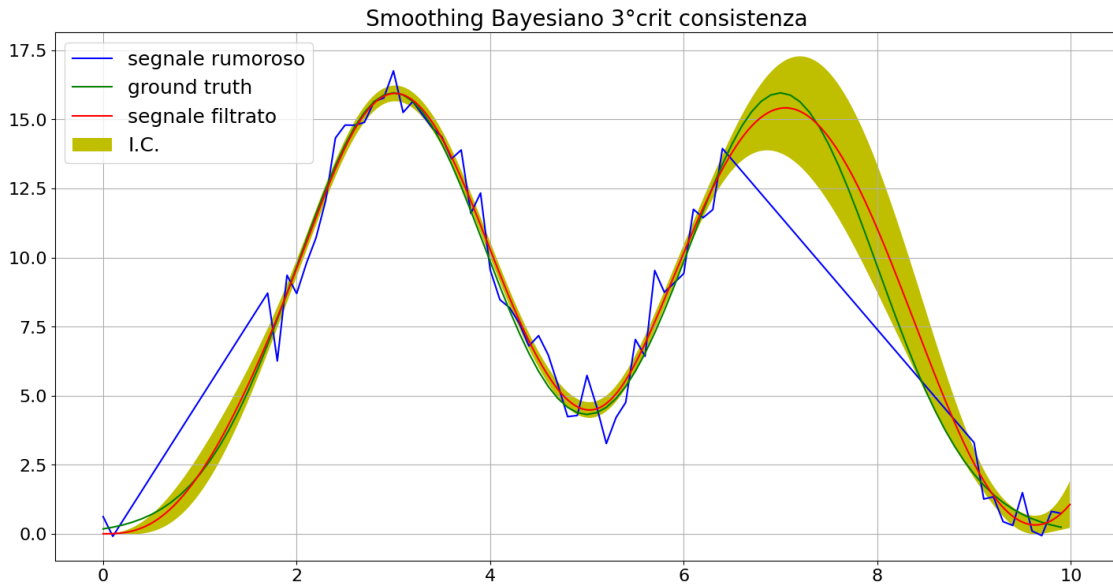


Figura 15: confronto tra ground truth, segnale rumoroso in ingresso allo stimatore e segnale filtrato con relativo intervallo di confidenza che aumenta in corrispondenza di campioni mancanti

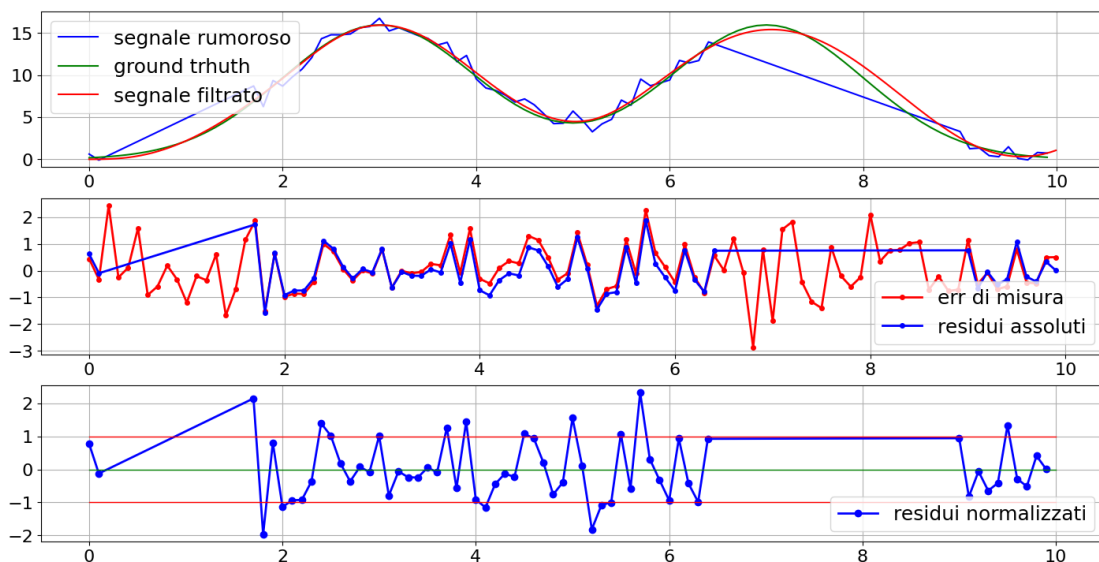


Figura 16: : primo grafico mostra il segnale filtrato, segnale rumoroso e ground truth a confronto [Figura 15], secondo grafico mette in relazione l'errore di misura con i residui assoluti, il terzo grafico mostra i residui normalizzati rispetto l'intervallo ± 1 . Si noti la mancanza di residui in corrispondenza di campioni mancanti

2.3.4 Innesco

Come ultimo lavoro di simulazione è stato testato l'innesco. Lo smoothing bayesiano per ipotesi impone che il segnale da stimare sia causale cioè con valori uguali a zero per tempi minori di zero. Considerando che la curva stimata si muoverà con una certa regolarità e non in modo repentino, occorreranno diversi campioni perché quest'ultima inizi a stimare correttamente il segnale, arrivando al primo campione di quest'ultimo con la giusta pendenza. Se il segnale da stimare fosse causale questa situazione non si presenterebbe, ma poiché HR non lo è, bisogna prendere in considerazione anche questo caso in fase di simulazione.

Come prima, si parte dalla solita traccia che viene traslata indietro nel tempo per fare in modo che il primo valore sia considerevolmente diverso da zero *Figura 17*.

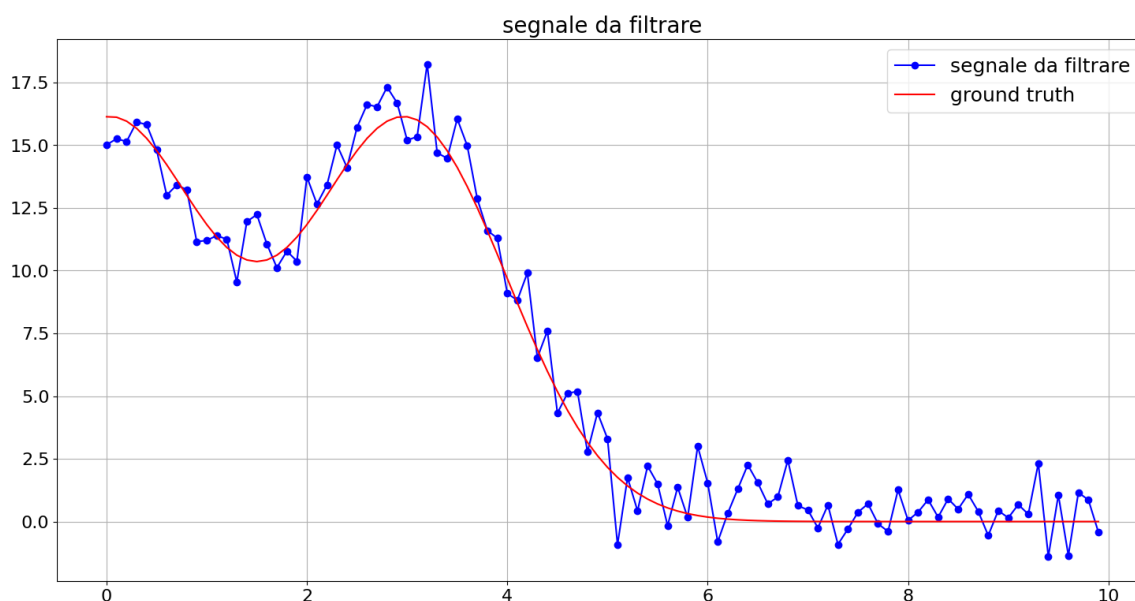


Figura 17: si riporta il segnale rumoroso e ground truth a confronto traslati indietro nel tempo rispetto a Figura 10, ciò verrà utilizzato per testare l'innesco

Per gestire il problema dell'innesco, l'accortezza utilizzata è quella di traslare il segnale in avanti in modo da allontanare il primo campione dal tempo zero. In questo test, per esempio, il tracciato è stato traslato di 10 (cioè della lunghezza del segnale stesso). Quindi si crea una griglia virtuale inerente al nuovo tracciato che parta da zero e giunga alla fine della curva con la frequenza di campionamento desiderata. In questo modo si permette alla stima di

raggiungere il primo campione con la pendenza ideale. In *Figura 18* è possibile visionare il corretto funzionamento del denoising. Come specificato nel capitolo precedente l'intervallo di confidenza nella prima sezione si ingrandisce in quanto non ci sono campioni del segnale grezzo in quella zona. Poiché questa strategia è ideata per “ingannare” lo stimatore ed è puramente artificiale, come ultimo steps è stata tagliata la parte relativa all'innesco e traslato a ritroso, nei tempi corretti, il segnale filtrato. I risultati sono riportati in *Figura 19* e *Figura 20*. Come precedentemente vengono confrontato i residui assoluti con l'errore di misura e quelli normalizzati con l'intervallo di ± 1 .

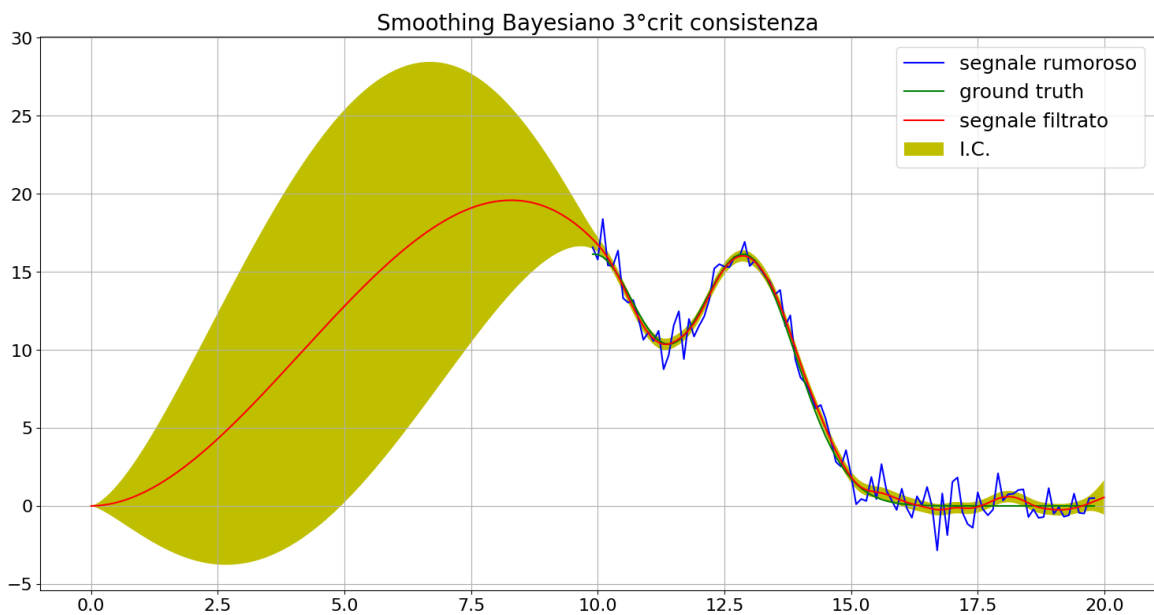


Figura 18: confronto tra ground truth, segnale rumoroso in ingresso allo stimatore e segnale filtrato con relativo intervallo di confidenza che aumenta in corrispondenza dell'innesco

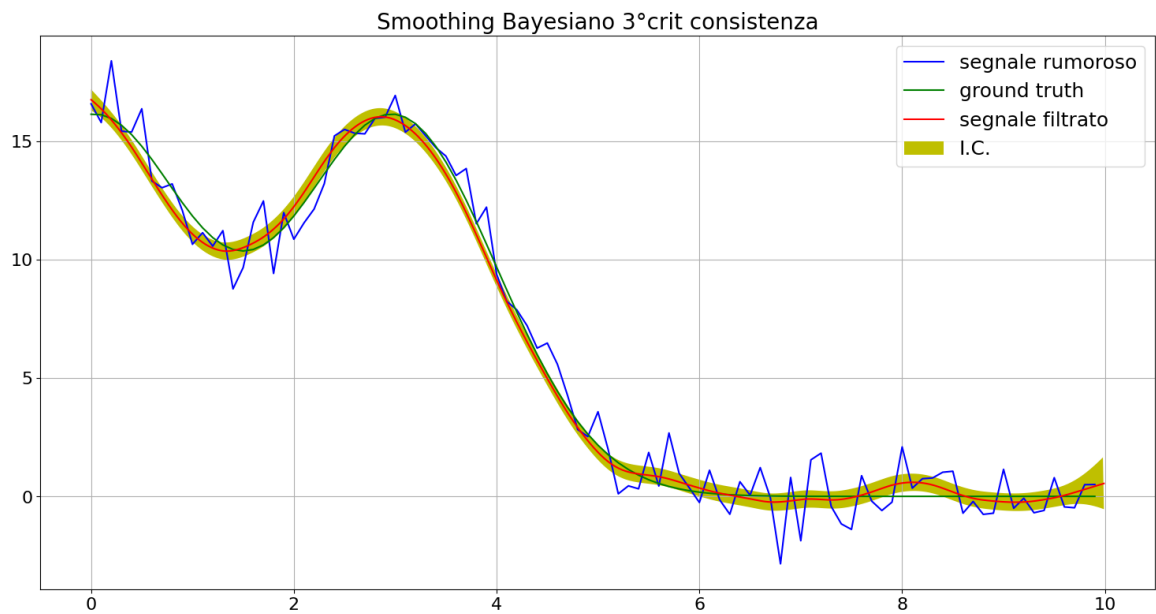


Figura 19: confronto tra ground truth, segnale rumoroso in ingresso allo stimatore e segnale filtrato con relativo intervallo di confidenza. Questo è il risultato finale dell'operazione svolta per la gestione dell'innesco

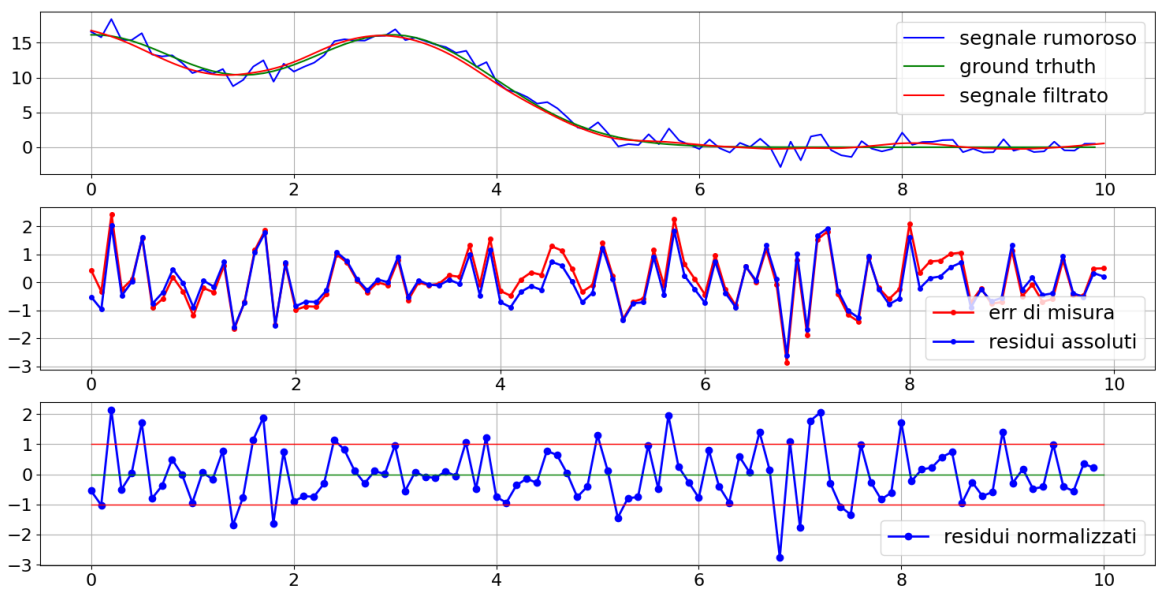


Figura 20: primo grafico mostra il segnale filtrato, segnale rumoroso e ground truth a confronto [Figura 19], secondo grafico mette in relazione l'errore di misura con i residui assoluti, il terzo grafico mostra i residui normalizzati rispetto l'intervallo +/-

Capitolo 3

Pipeline per il filtraggio di dati Heart Rate

3.1 Divisione dei dati Heart Rate in finestre

Dall'analisi preliminare dei tracciati di Heart Rate si è osservato che la variabilità del segnale intorno a un ipotetico andamento senza rumore non è costante per tutta la durata della giornata, in particolare tra ore di sonno e di veglia il segnale hanno variabilità molto diverse. Anche se si può immaginare il rumore diverso da un processo di rumore bianco, non si può fare altro che assumerlo come tale, in quanto non si hanno elementi aggiuntivi per poter fare ipotesi più specifiche e precise sul processo che regola l'errore di misura. Un'altra ipotesi che risulta forzata per l'andamento dell'intera giornata è che la varianza σ^2 del processo rumore bianco (formulazione (2. 11)(2. 11)) che si assume modellare l'errore di misura, sia costante. Anche questa ipotesi viene accettata per motivi di mancanza di informazione più dettagliata a riguardo. Quindi per poter raggiungere un corretto livello di aderenza alle ipotesi sopra specificate, si è deciso di segmentare l'intero tracciato in finestre più piccole da considerarsi a sestanti e da filtrare indipendentemente una dall'altra. Questo permette di creare uno stimatore self-tunable per quanto riguarda la stima della varianza del rumore, cioè in grado di correggere autonomamente il suo valore.

Inoltre, se si filtrasse l'intero tracciato HR, che nel caso in cui non ci siano campioni mancanti contiene quasi sei mila campioni, si avrebbe un costo computazionale troppo elevato. Quindi

anche per abbattere i tempi di processing è necessaria una suddivisione del tracciato in porzioni più brevi.

Il numero dei campioni per ogni finestra è 300. Questo valore è stato scelto in modo empirico, in quanto è stato testato come buon compromesso tra un costo computazionale e di conseguenza un tempo di processing adeguato e una informazione sufficiente passata allo stimatore per eseguire un buon filtraggio.

Il numero di campioni per ogni finestra è una variabile lasciata come parametro utente; quindi, è in realtà possibile fare diverse prove con lunghezze differenti. Questa pipeline è sviluppata con l'intento di poterla utilizzare per fare smoothing di dati con approccio bayesiano in generale, anche su altri tipi di tracciati; quindi, potrebbe essere utile variare questo valore. In ogni modo tutti i risultati presentati in questo lavoro di tesi si rifanno allo stesso tipo di tracciato e il valore in questione viene lasciato fisso.

Infine, si osserva come durante questa fase non si ha una ground truth di riferimento e quindi viene lasciato allo stimatore (come successo nelle simulazioni del capitolo 2.3) il calcolo del gamma ottimo che denota il giusto compromesso tra undersmoothing e oversmoothing.

3.1.1 Procedura di suddivisione in finestre

Dopo aver importato i dati di cui si vuole filtrare il tracciato HR, lo si visiona con i relativi campioni mancanti [Figura 21] con la stessa procedura citata nell'analisi preliminare [paragrafo 2.2].

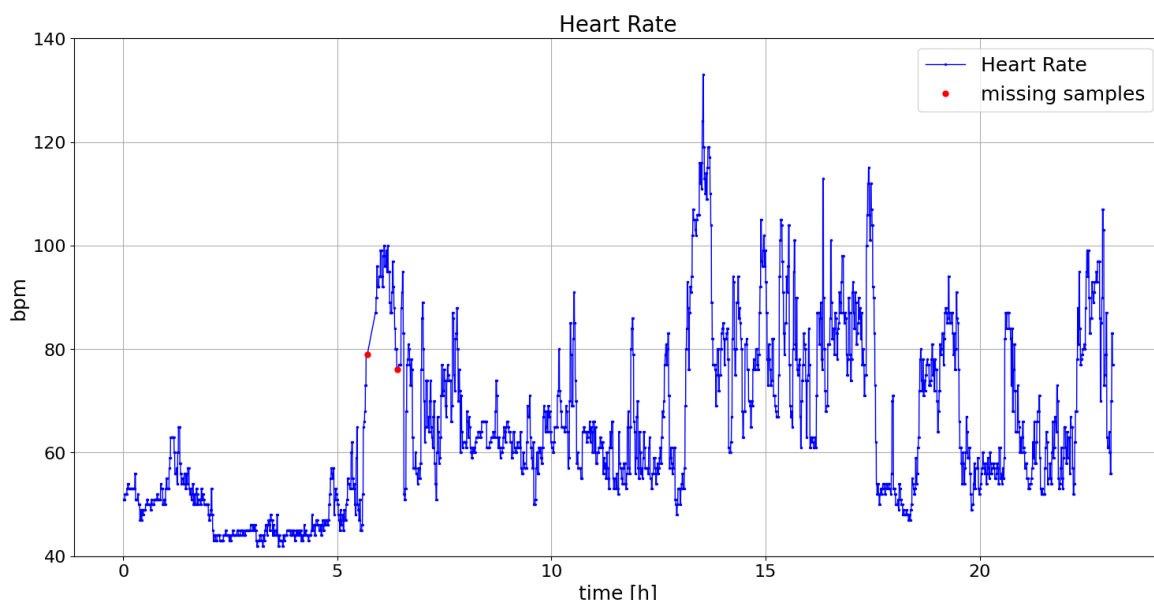


Figura 21: Segnale Heart Rate complessivo da filtrare e marker per segnalare l'inizio di una zona di campioni mancanti.

Successivamente si genera l'asse dei tempi da utilizzare come griglia virtuale. Poiché la si vuole creare con passo regolare, indipendentemente dai missing samples del tracciato originale, si crea un vettore che parta dal primo valore dei tempi fino all'ultimo con passo di 45 secondi. Infatti, benché il tempo di campionamento del Garmin sia di 15 secondi, si è operato un ulteriore campionamento che prelevasse un valore ogni tre e quindi il nuovo tempo di campionamento risulta essere 45 secondi. Se si considera che possono esserci dei campioni mancanti lungo il tracciato, un mero sotto campionamento non è sufficiente per essere sicuri che la griglia virtuale fitti con la griglia reale, quindi oltre a sotto-campionare si esegue anche un controllo che si stiano prelevando i campioni relativi ai tempi presenti in virtual grid. Questa operazione è stata fatta per motivi empirici di rispetto delle ipotesi considerate per lo stimatore bayesiano; infatti, se non si operasse questo passaggio (ulteriore sotto-campionamento) lo stimatore compirebbe un forte undersmoothing praticamente fino a fittare i dati con un intervallo di confidenza prossimo allo zero. Ciò è dovuto al fatto che ci sono tanti campioni consecutivi con il medesimo valore che vincolano lo stimatore allo stesso e non a fare un giusto compromesso con il modello di informazione a priori. Inoltre, questa operazione è giustificata anche dal fatto che c'è correlazione tra campioni consecutivi di HR [23], quindi la perdita di campioni non è una perdita consistente di informazione.

In seguito, vengono settati dei parametri utenti che sono: i campioni per ciascuna finestra, i campioni dell'innesco, e i campioni di sovrapposizione tra una finestra e quella successiva (in seguito verrà specificato l'utilizzo di queste variabili). Per i risultati riportati in questa tesi i valori rispettivamente sono: 300, 60, 10.

A questo punto, la pipeline si occupa di tagliare il tracciato in finestre e restituire per ciascuna quattro grafici sotto riportati. In *Figura 22* sono mostrati i campioni che compongono la prima finestra presa in considerazione rispetto all'intero tracciato. Ciò permette di rendersi conto anche a colpo d'occhio qual è la sua dimensione rispetto al tracciato totale. La *Figura 23* mostra il segnale grezzo e il rispettivo segnale filtrato con lo smoothing bayesiano e relativo l'intervallo di confidenza. In *Figura 24* si possono vedere: i residui assoluti (primo subplot) e i residui normalizzati rispetto alla banda di riferimento ± 1 (secondo subplot).

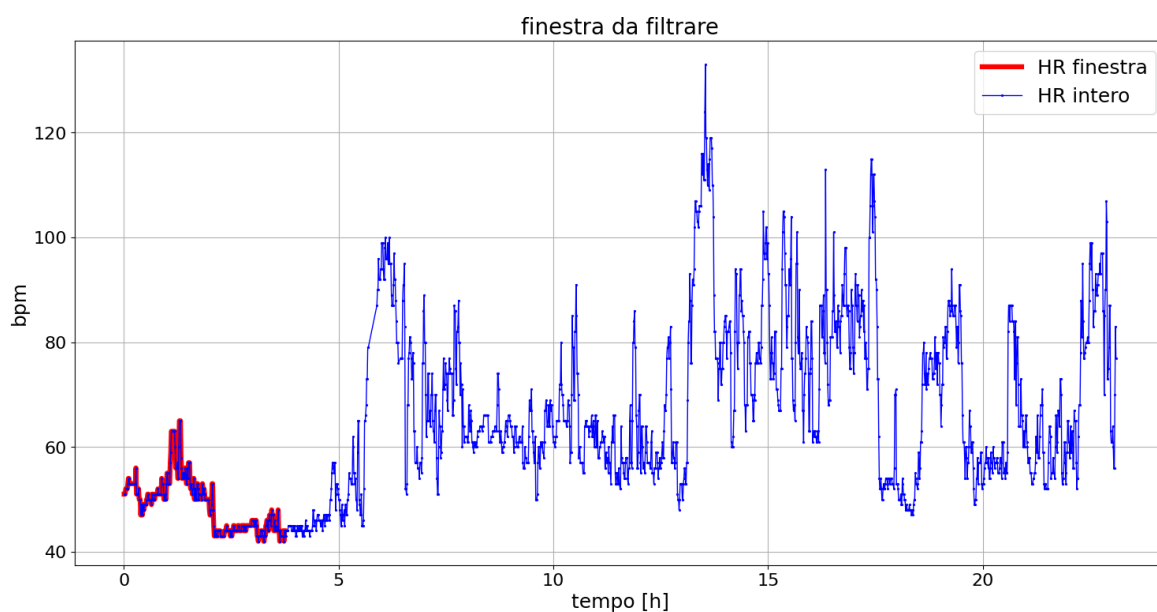


Figura 22: in figura sono visibili i campioni della finestra da filtrare prelevati dall'intero tracciato

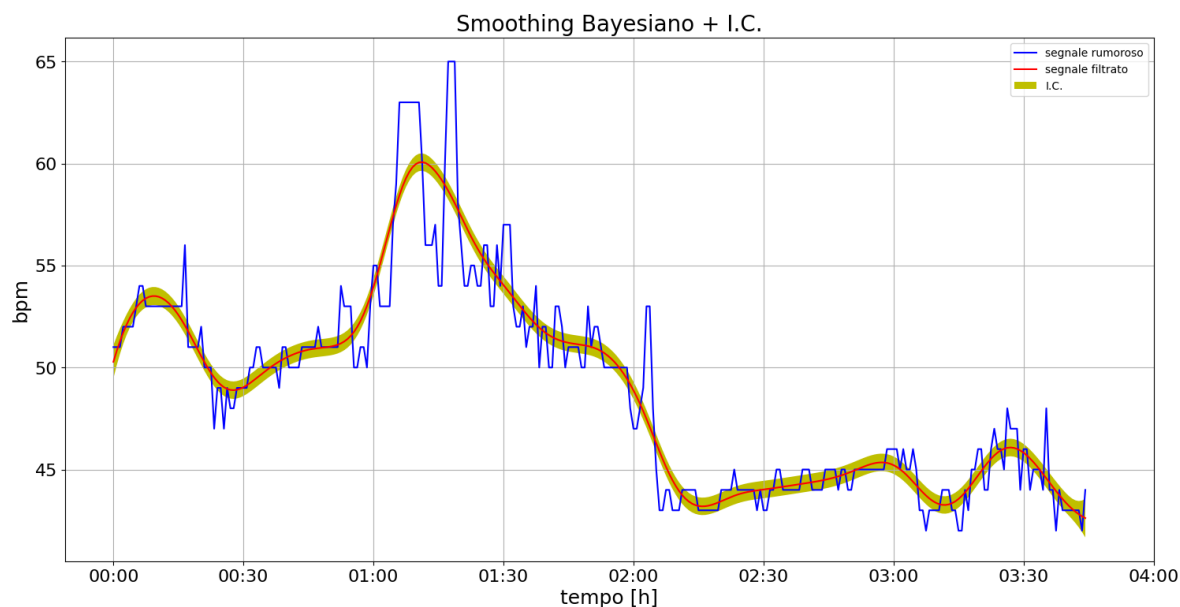


Figura 23: confronto tra segnale rumoroso in ingresso allo stimatore nella prima finestra e segnale filtrato con relativo intervallo di confidenza

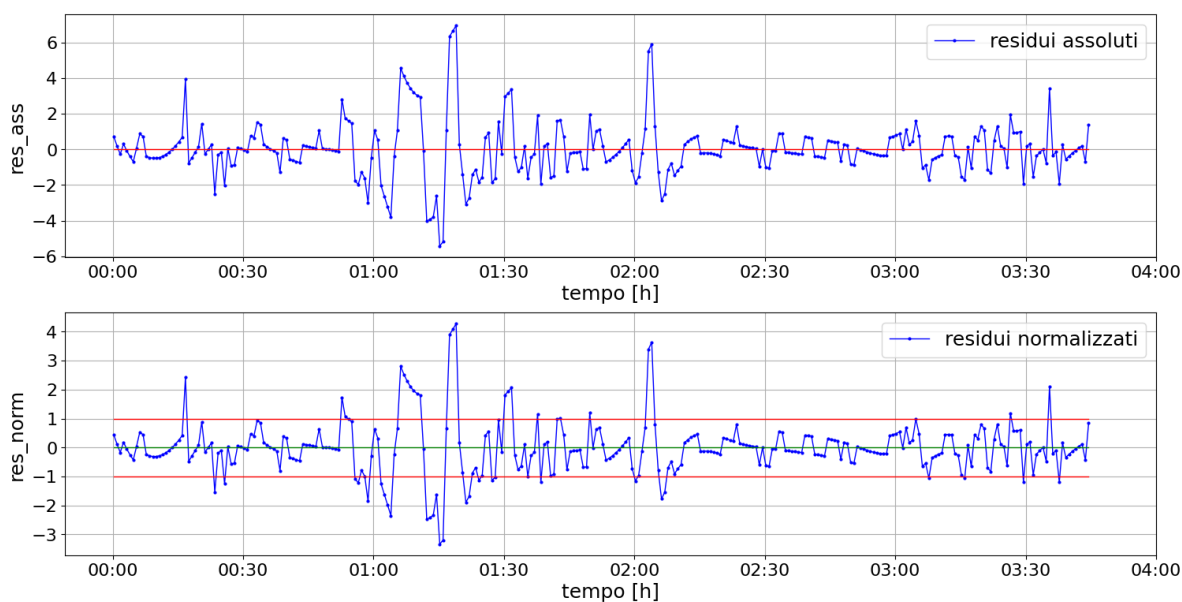


Figura 24: residui assoluti e normalizzati relativi alla finestra mostrata in Figura 23 rispettivamente visionabili nel primo e nel secondo grafico

Di seguito viene riportato il filtraggio relativo alla seconda finestra *Figura 25*. Come si può vedere, il tracciato presenta dei campioni mancanti in due zone riconoscibili grazie ai marker in *Figura 21*, ma anche per mezzo dell'intervallo di confidenza che si vede aumentare

notevolmente in quelle porzioni, in particolare la prima in quanto è piuttosto consistente. In queste sezioni la stima si muove utilizzando solo l'informazione a priori e quindi cercando di raccordare nel modo più lineare possibile i due estremi. È bene annotare che in questa finestra il numero di campioni contenuti nel segnale grezzo sono 300, ma quelli in uscita allo stimatore saranno di più, in quanto ci sono campioni anche nelle zone messe in evidenza.

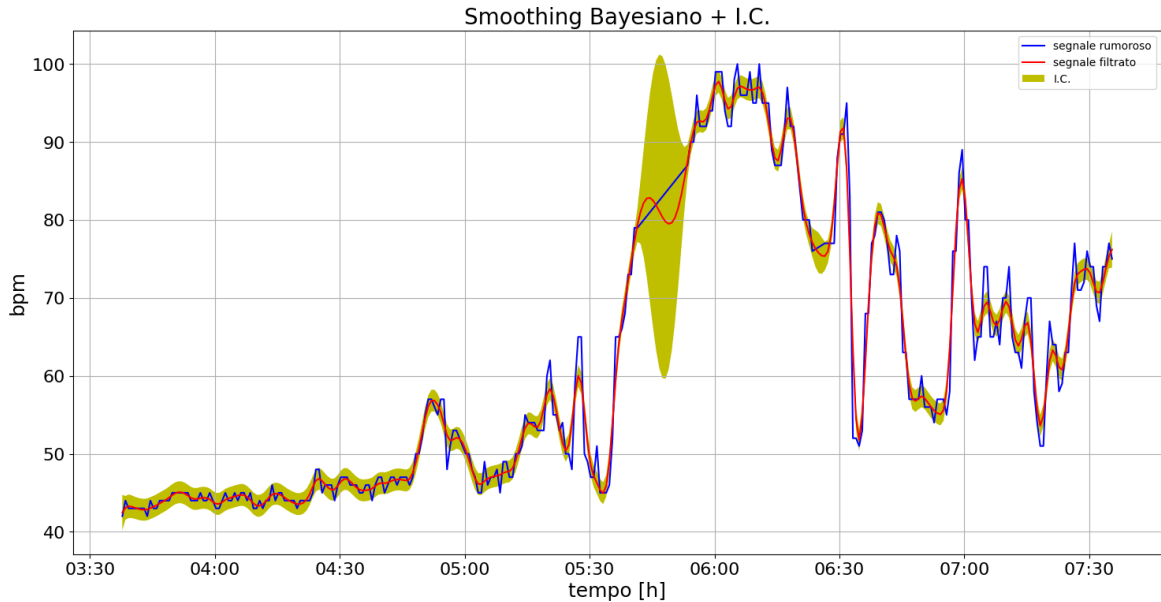


Figura 25: filtraggio della seconda finestra; è visibile il segnale grezzo, il segnale filtrato con relativo intervallo di confidenza. In questa finestra sono presenti due zone di campioni mancanti dove l'intervallo di confidenza aumenta

L'ultima finestra *Figura 26* invece è in generale l'unica eccezione alla regola dei 300 campioni per finestra. Infatti, l'ultima avrà lunghezza variabile in base ai campioni rimasti. In questo caso, per esempio, sono 91. Valore troppo basso per compiere una buona stima, ciò lo si vede dall'oversmoothing compiuto. Anche la varianza dell'errore di misura stimata risulta un valore elevato superiore a quello delle altre finestre, questo argomento verrà trattato nel paragrafo 3.3.

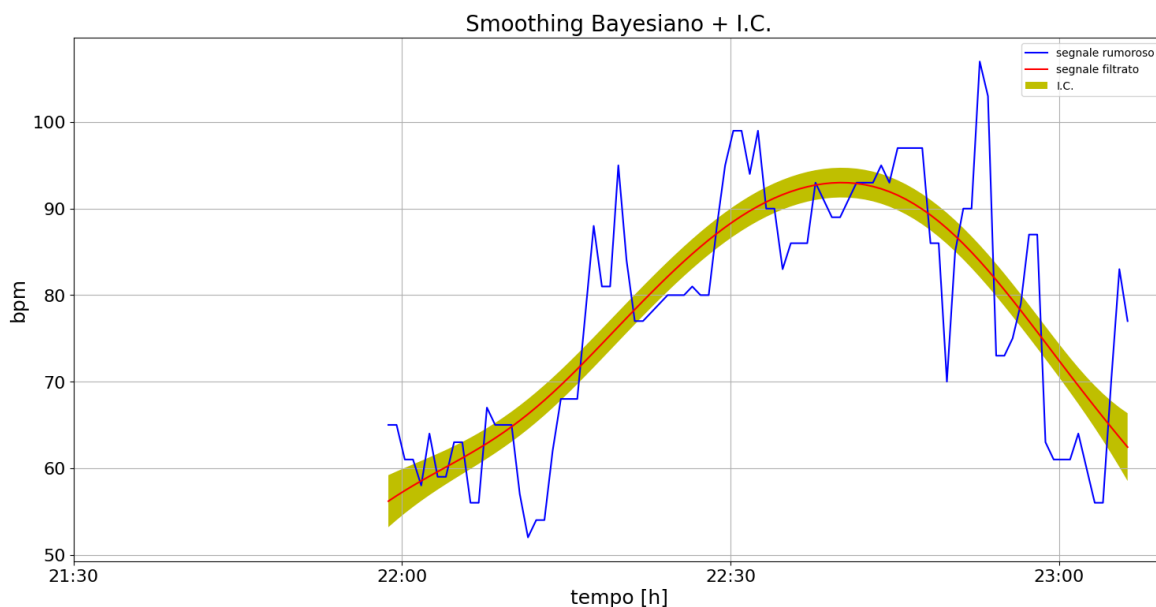


Figura 26: filtraggio dell'ultima finestra composta da 91 campioni, vengono riportati il segnale grezzo, quello filtrato con relativo intervallo di confidenza

3.2 Riconciliazione delle finestre

Conclusa la fase descritta nel paragrafo 3.1 si ottengono i vettori di stima e residui normalizzati segmentati nelle varie finestre, per questo motivo si necessita di una fase di riconciliazione dei segmenti per ottenere nuovamente i tracciati nella loro interezza. Nel caso si concatenassero semplicemente i vari array tra loro, si rischierebbe di ottenere dei tracciati spigolosi nel punto di raccordo, mentre invece l'obiettivo del denoising è anche quello di ottenere un segnale regolare; quindi, tutto il lavoro svolto verrebbe danneggiato in questi punti specifici. Inoltre, si deve considerare che questi punti angolosi sono il risultato di un'operazione artificiale, dovuta al metodo utilizzato per l'implementazione pratica, e non sarebbero presenti neanche nel segnale rumoroso. Ciò potrebbe andare a inficiare anche sull'estrazione delle features, che è il passaggio successivo al denoising, ottenendone anche alcune non rappresentative della realtà.

Quindi per superare questa problematica si è deciso di utilizzare un kernel gaussiano che permetta di raccordare in modo regolare i tracciati sopra citati.

3.2.1 Kernel Gaussiano

Per poter utilizzare un kernel per raccordare due segnali serve stabilire un numero di campioni di transizione. Questa variabile rientra tra i parametri utenti, deve essere necessariamente un intero e descrive di quanti campioni si sovrappongono due finestre consecutive. Come citato sopra questo valore per questa tesi è stato settato a 10. Ciò significa, per riportare un esempio pratico, che se la prima finestra comprende i campioni da 0 a 299 la seconda contiene quelli da 290 a 589, con i dieci campioni da 290 a 299 in comune alle due finestre e così via fino all'esaurimento del vettore. Su questi ultimi agirà il kernel, che nel caso particolare è mostrato in *Figura 27*. Esso ha origine da una densità di probabilità gaussiana, la quale è divisa per il valore del picco per normalizzarla. Ha un asse delle ascisse composto da 19 campioni in modo tale che dal primo al picco siano esattamente 10. Questi valori vengono prelevati ed utilizzati per una media ponderata secondo la seguente formulazione:

$$\hat{u}_i = \frac{x1_i \cdot w_i + x2_{n+1-i} \cdot w_{n+1-i}}{w_i + w_{n+1-i}} \quad (3.1)$$

Con $n = 10$, $1 \leq i \leq 10$.

- $x1$ contiene gli ultimi dieci campioni della finestra precedente;
- $x2$ = contiene i primi dieci campioni della finestra successiva;
- w contiene i campioni dal 10 al 19 del vettore kernel.

L'operazione corrisponde a eseguire una media tra i due elementi delle due finestre che giacciono sullo stesso punto dell'asse dei tempi. I campioni appartenenti alla prima finestra e più vicini a quest'ultima avranno peso maggiore di quelli appartenenti alla seconda e viceversa come visibile in *Figura 27*.

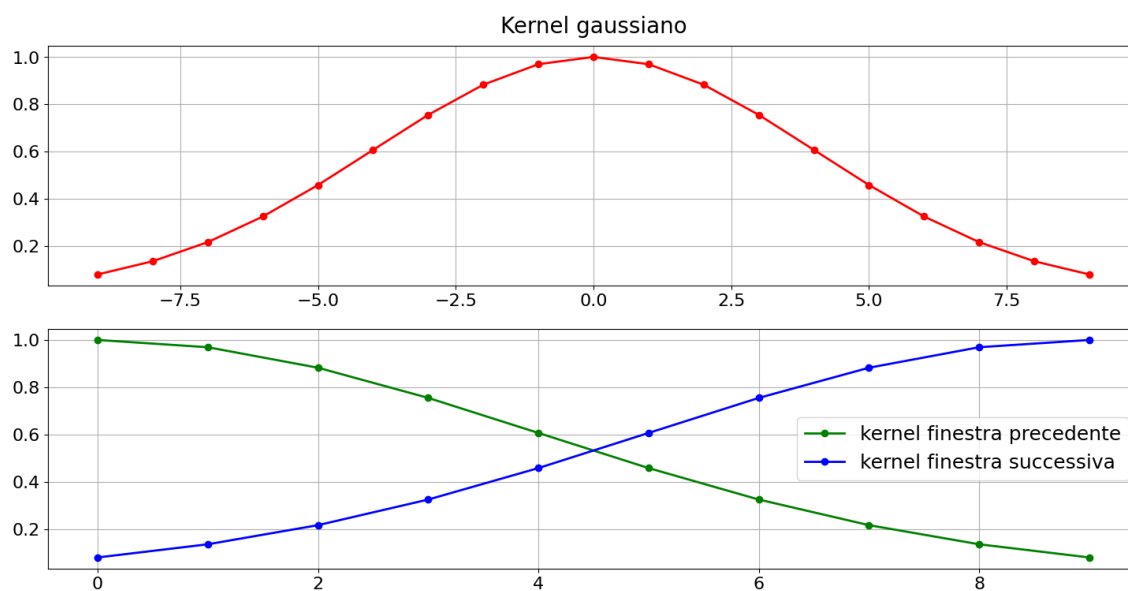


Figura 27: in alto il kernel gaussiano, in basso porzioni di kernel usate come pesi per la finestra precedente e quella successiva.

3.2.2 Risultati della riconciliazione

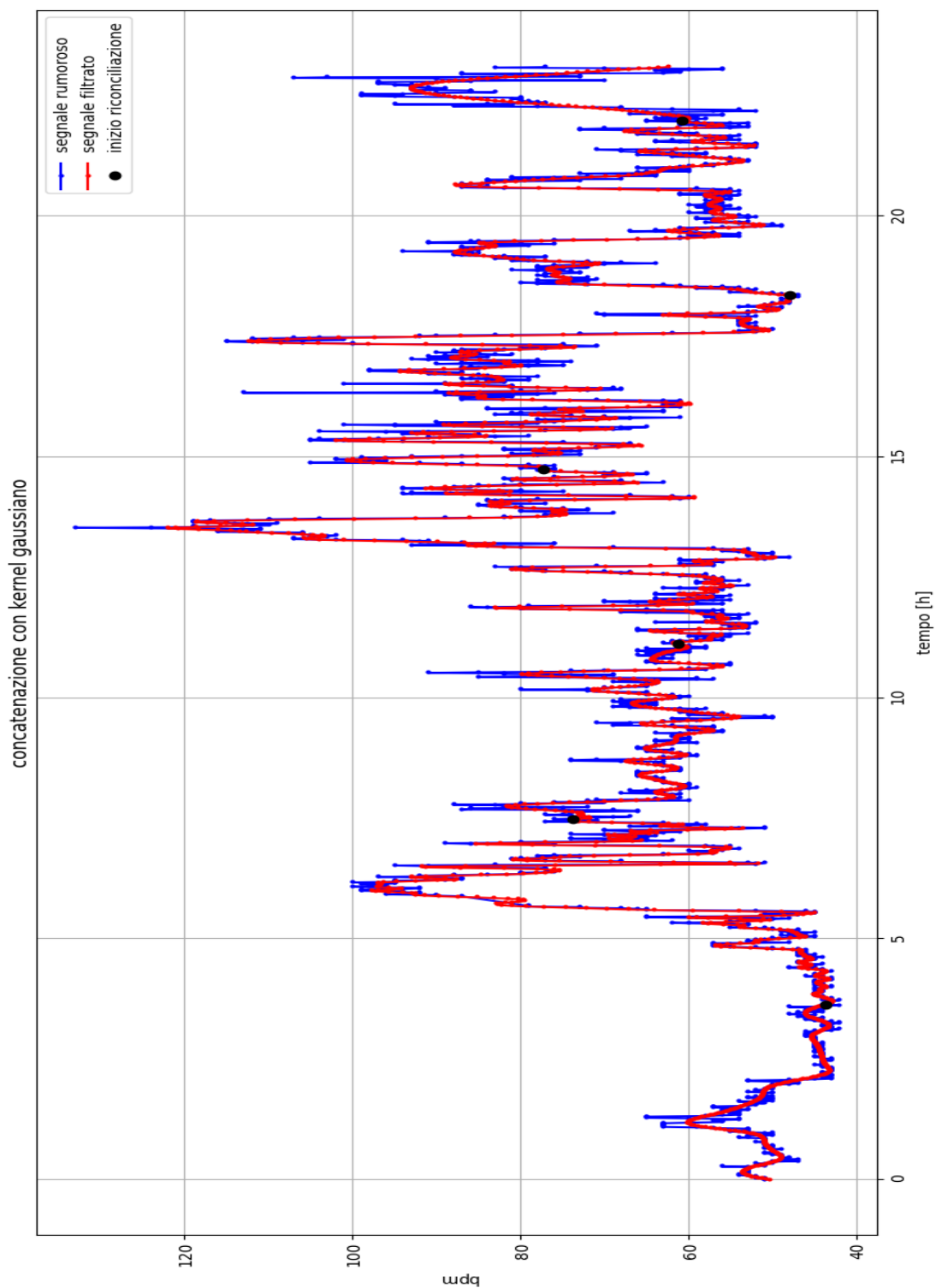


Figura 28: segnale grezzo e segnale filtrato dopo aver riconciliato le finestre tra loro nei punti con marker nero.

Una volta operata la riconciliazione tra tutte le finestre, la pipeline fornisce una immagine che mostra il segnale grezzo, quello filtrato, e anche i marker che segnalano il primo campione tra quelli impiegati nella riconciliazione che avviene come sopra enunciato [Figura 28].

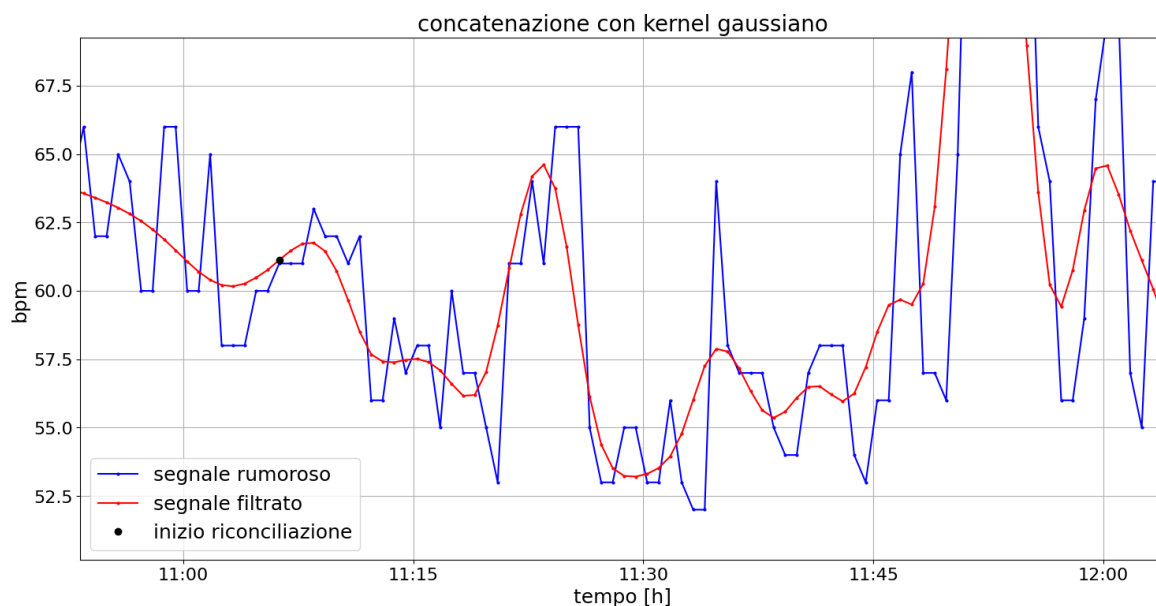


Figura 29: ingrandimento della Figura 28 nel quale si apprezza l'andamento regolare nel punto di riconciliazione tra due finestre (marker nero)

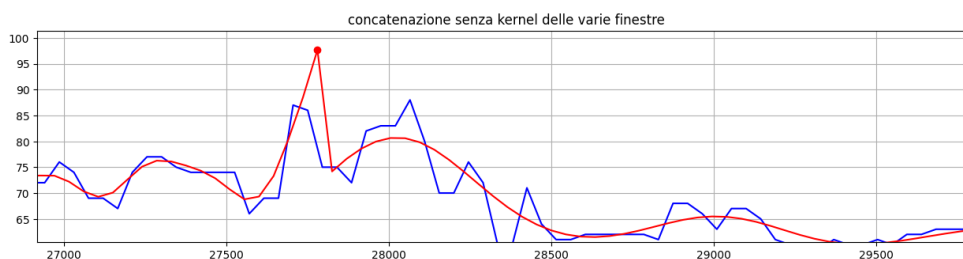


Figura 30: esempio di riconciliazione senza kernel, è visibile un punto spigoloso (marker) dovuto alla semplice concatenazione delle finestre

Inoltre, viene riportata un'immagine [Figura 29] che riporta il punto di riconciliazione per mostrare come l'utilizzo del kernel gaussiano abbia apportato una notevole miglioria dal punto di vista della regolarità del segnale filtrato. In Figura 30, invece, si può vedere la concatenazione di finestre senza l'utilizzo del kernel. In questo caso si riconosce un picco innaturale nell'andamento del segnale. Quindi in conclusione è possibile capire la miglioria che è stata apportata con questa operazione.

3.3 Residui

Infine, la pipeline si occupa di calcolare i residui assoluti e normalizzati. Anche questi vettori come quello del segnale filtrato devono essere riconciliati partendo dagli array forniti per ciascuna finestra fino a ottenere i vettori con i residui giornalieri. Anche in questo caso si è usata la stessa strategia del kernel gaussiano sopra descritto. In *Figura 31*(terzo riquadro) sono riportati i residui normalizzati in riferimento alla banda di confidenza ± 1 . È bene considerare che per calcolare i residui normalizzati si divide il valore dei residui assoluti per la varianza del rumore bianco σ^2 . Poiché quest'ultima risulta costante per ipotesi sulle singole finestre, si avrà un andamento costante a tratti sull'intera giornata (secondo riquadro). I raccordi anche in questo caso sono stati eseguiti con il kernel gaussiano come si vede dall'immagine non c'è un gradino netto tra il valore precedente e quello successivo, bensì occorrono diversi campioni (in questo caso sempre dieci) per passare da un valore a quello successivo. Questo risultato è molto importante in quanto permette di poter affermare che la pipeline è in grado di auto correggere il valore di σ^2 processando man mano i dati dell'intera giornata. Esso è un parametro self-tunable, che quindi permette di calcolare sicuramente una stima più accurata, rispetto a quella calcolata se il suo valore rimanesse costante per l'intero tracciato. Ulteriore commento è il fatto che lo stimatore interpreta correttamente le fasi della giornata con variabilità maggiore aumentando il sigma quadrato e viceversa. In particolare, nell'ultima finestra, che contiene pochi campioni, lo stimatore esegue una stima meno precisa, che si discosta molto dai dati. Ciò è riscontrabile dai residui assoluti che aumentano notevolmente in valore assoluto, mentre quelli normalizzati sono confrontabili con quelli del resto della giornata, in quanto una varianza maggiore li normalizza nel range atteso. Infine, bisogna anche annotare che questo aspetto è un punto di merito del lavoro, ma potrebbe essere ancora migliorato, poichè i residui non hanno varianza costante, come considerato nelle ipotesi, in ciascuna finestra; inoltre, presentano anche una correlazione, di conseguenza anche la seconda ipotesi decade. Ciò significa che la matrice B potrebbe essere implementata non come una matrice identità bensì contenente i parametri per un processo autoregressivo (AR), che denota quindi una correlazione nei campioni.

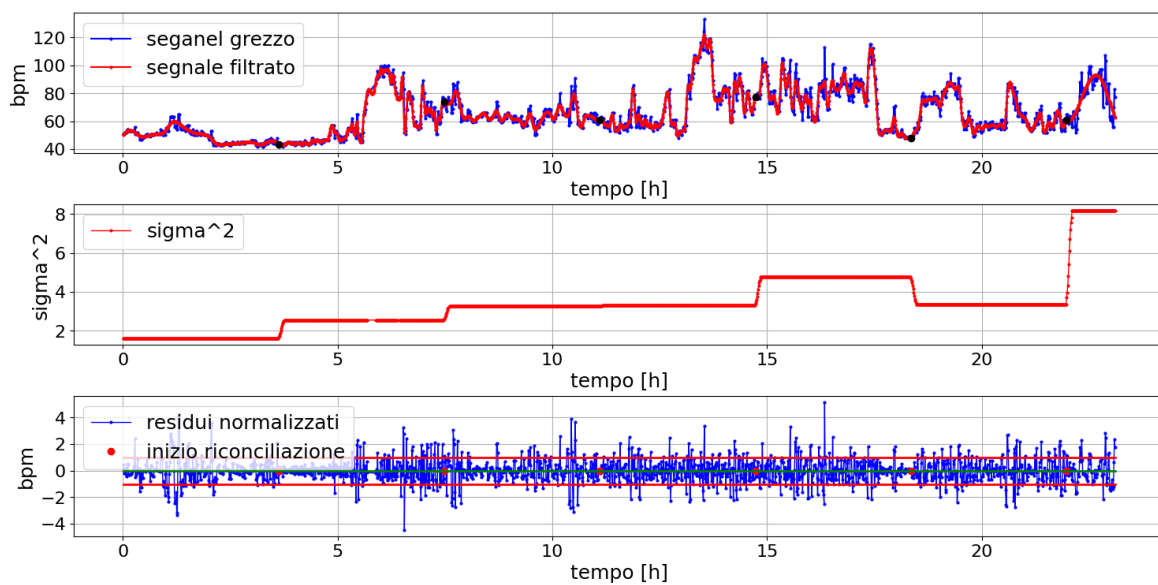


Figura 31: in alto segnale grezzo e segnale filtrato riconciliato, al centro l'andamento a gradino del σ^2 , in basso i residui normalizzati

Capitolo 4

Risultati

4.1 Polar H10: gold standard di riferimento

Per valutare le performance dello smoothing bayesiano, si è svolto un test confrontando le features estratte dallo smartwatch con quelle estraibili dal segnale di un sensore di riferimento. Si è quindi cercato in letteratura quale potesse essere il tracciato di riferimento più attendibile. Il sensore ideale è risultato essere un holter, dispositivo che permette la raccolta di 24 ore di ECG continuo, anche durante un'attività fisica [24]. L'esecuzione di una raccolta dati Holter e la successiva esportazione dei dati è una procedura complessa, e si è quindi deciso di procedere con un confronto con il **Polar H10** [25], considerato il gold standard di riferimento in questo settore[26] [27]. In aggiunta, è uno strumento presente in dipartimento e quindi si è potuto usare facilmente per questo scopo.

4.1.1 Confronto Garmin-Polar

Per porsi nelle condizioni ideali per il confronto, è stata ideata una prova in cui un soggetto: uomo sano di 30 anni ha indossato contemporaneamente i due sensori per dodici ore. Successivamente sono stati prelevati i dati sia dal Garmin sia quelli del Polar estratti in formato CSV dal sensore. In *Figura 32* è possibile vedere il grafico dei due tracciati sovrapposti. Si nota che il tracciato del Polar presenta più campioni per via del tempo di campionamento pari a 1 secondo. Inoltre, i due sensori hanno un approccio diverso nella gestione dei campioni mancanti. Il Polar imposta a zero il valore di questi ultimi, mentre il

Garmin non li riporta. Visualizzando un dettaglio mostrato in *Figura 33*, tra le ore 12:00 e le 14:00, dove sono bene visibili i fronti di salita e di discesa, si nota un discostamento tra i tracciati. Nei punti segnalati dai marker in *Figura 33* il ritardo del tracciato Garmin è quantificato in 120-150 secondi. Ritardo piuttosto consistente considerando che questo parametro varia molto più velocemente nel tempo. Questo è un problema aperto che verrà investigato in lavori futuri.

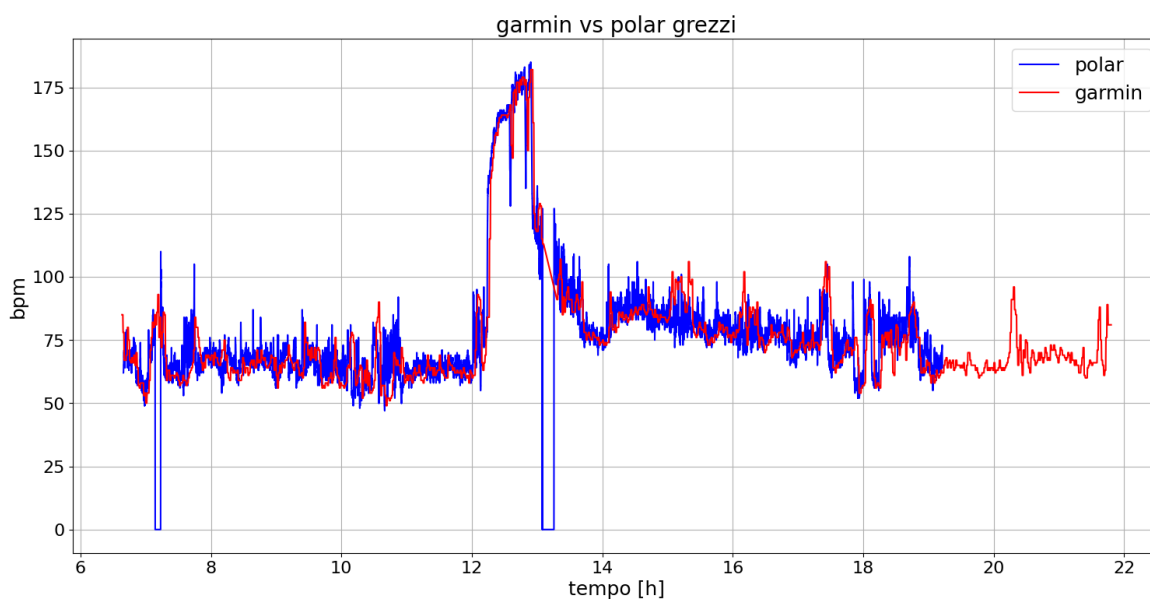


Figura 32: confronto tra segnali di Heart Rate prelevati da Polar H10 e Garmin vivoactive4

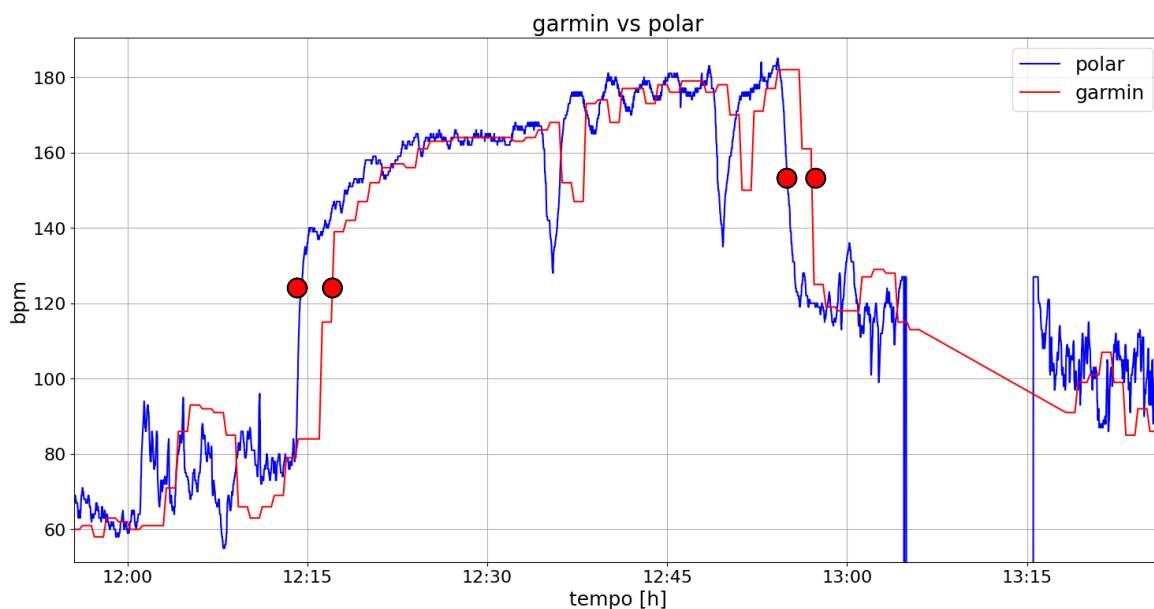


Figura 33: dettaglio di figura Figura 32 nel quale è possibile riconoscere un ritardo del segnale garmin rispetto a quello polar nei fronti di salita e discesa

Le prime due operazioni fatte sui due tracciati grezzi provenienti dal Garmin e dal Polar per poterli rendere uniformi sono state rispettivamente eliminare gli zeri dal tracciato del Polar e tagliare i segnali nel punto iniziale e finale facendo in modo che i due tracciati fossero corrispondenti sull'asse dei tempi.

Ciò è visibile in *Figura 34*, in particolare sono rappresentati i segnali derivanti dal Polar con campionamento 1 secondo e quello proveniente dal Garmin campionamento 15 secondi, prima di essere filtrati.

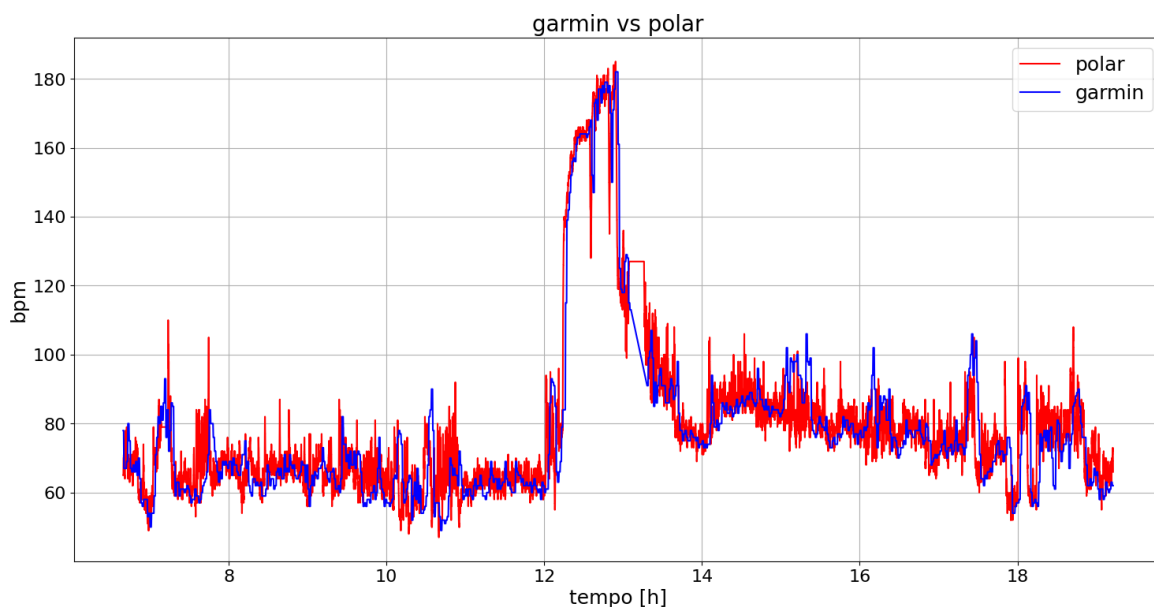


Figura 34: segnali Polar e Garmin dopo essere stati tagliati nel punto iniziale e finale per renderli uniformi ed aver eliminato gli zeri nel tracciato Polar

4.2 Risultati

Per poter giungere a risultati quantificabili, si è deciso di fare una valutazione delle features estratte dai tracciati. Esse sono rappresentative delle caratteristiche dei segnali Heart Rate, sono descritte nel dettaglio in *Tabella 1*, *Tabella 2*, *Tabella 3*. In particolare, sono state fatte delle analisi su quelle utilizzate già all'interno del progetto BRAINTEASER. I tracciati che sono stati utilizzati per l'estrazione delle features sono tre:

- **Garmin_media**: segnale Garmin filtrato con la tecnica della media mobile;
- **Garmin_bayes**: segnale Garmin filtrato con stima bayesiana con tempo di campionamento uguale a 45s;
- **Polar_raw**: segnale Polar grezzo con tempo di campionamento uguale a 1 s;

4.2.2 Features

Le *features* estratte dai tracciati come sopra descritto vengono divise per categorie in tre tabelle. *Tabella 1* contiene quelle inerenti a trend giornalieri significativi.

Features	Descrizione
lineartrend	Parametro corrispondente al trend lineare.
quadratictrend	Parametro corrispondente al trend quadratico
r2	Indica la qualità del modello quadratico.
mean	Frequenza cardiaca media dopo l'inizio dello stimolo
std	Standard Deviation dell'HR dopo l'inizio dello stimolo
maxTime	Tempo relativo al picco massimo
minTime	Tempo relativo al picco minimo

Tabella 1: features rappresentanti l'andamento giornaliero dei segnali

In *Tabella 2* si trovano tutte le feature di carattere geometrico relative agli intervalli battito-battito estratti dai sensori.

Features	Descrizione
meanNN	Media degli intervalli RR
sdNN	Standard deviation degli intervalli RR
sdaNN1/sdaNN2/sdaNN5	Standard deviation di media di intervalli di 1/2/5 min.
sdNNI1/sdNNI2/sdNNI5	La media delle standard deviation di intervalli di 1/2/5 min.

medianNN	La mediana di valori assoluti delle differenze successive tra intervalli RR.
madNN	La deviazione mediana assoluta degli intervalli RR.
hcvNN	$\text{madNN}/\text{medianNN}$.
iqrNN	Range interquartile degli intervalli RR.
pNN50	Numero di intervalli RR > 50ms, diviso totale di intervalli RR.
pNN20	Numero di intervalli RR > 20ms, diviso totale di intervalli RR.
sdsd	Standard deviation delle differenze successive tra intervalli RR.
rmssd	Radice quadrata della media delle somme delle differenze successive tra intervalli adiacenti.
cvNN	$\text{sdNN}/\text{meanNN}$.
cvsd	$\text{rmssd}/\text{meanNN}$.
tiNN	Parametro geometrico, lunghezza della linea di base per interpolazione triangolare.??
HTI	Indice triangolare HRV che misura numero totale degli intervalli RR diviso per l'altezza dell'istogramma degli intervalli RR.
sd1	È una misura della diffusione degli intervalli RR sulla perpendicolare del diagramma di Poincaré alla linea di identità. È un indice delle fluttuazioni dell'intervallo RR a breve termine. È equivalente (sebbene su un'altra scala) a <i>rmssd</i> e quindi è ridondante riportare correlazioni con entrambi (Ciccone, 2017).
sd2	Misura della diffusione degli intervalli RR sul diagramma di Poincaré lungo la linea di identità. È un indice delle

	fluttuazioni dell'intervallo RR a lungo termine.
sd1sd2	Il rapporto tra le fluttuazioni a breve ea lungo termine degli intervalli RR. $sd1/sd2$.
cvi	Indice cardiaco simpatico (Toichi, 1997). Logaritmo del prodotto della variabilità longitudinale del diagramma di Poincaré per la sua variabilità trasversale.
csi	Indice cardiaco simpatico (Toichi, 1997). Calcolato dividendo la variabilità longitudinale del diagramma di Poincaré dalla sua variabilità trasversale.
csiModified	Il CSI modificato (Jeppesen, 2014) ottenuto dividendo il quadrato della variabilità longitudinale per sua variabilità trasversale.
GI	Indice di Guzik, definito come la distanza dei punti sopra la linea di identità (LI), diviso per la distanza di tutti i punti del tracciato di Poincaré rispetto a LI eccetto quelli che lo sono situati sopra LI.
SI	Indice di pendenza (slope index SI), definito come l'angolo di fase dei punti sopra LI diviso per l'angolo di fase di tutti i punti nella trama di Poincaré tranne quelli che si trovano sopra LI.
AI	Indice di Area (Area Index AI), definito come l'area cumulativa dei settori corrispondenti ai punti che si trovano sopra LI, diviso per l'area cumulativa dei settori corrispondenti a tutti i punti della trama di Poincaré eccetto quelli che si trovano sopra LI.

PI	Indice di Porta (Porta's Index PI), definito come il numero di punti sotto LI diviso per il numero totale di punti nella trama di Poincaré eccetto quelli che si trovano sopra LI.
C1d / C1a	Rispettivamente: contributi delle decelerazioni e delle accelerazioni della frequenza cardiaca nell'HRV a breve termine (Piskorski, 2011).
SD1d / SD1a	Rispettivamente: varianza a breve termine dei contributi delle decelerazioni (prolungamenti di RR intervalli) e accelerazioni (accorciamento degli intervalli RR), (Piskorski, 2011).
C2d / C2a	Rispettivamente: contributi delle decelerazioni e delle accelerazioni della frequenza cardiaca nell'HRV a lungo termine (Piskorski, 2011).
SD2d / SD2a	Rispettivamente: varianza a lungo termine dei contributi delle decelerazioni (prolungamenti di RR intervalli) e accelerazioni (accorciamento degli intervalli RR), (Piskorski, 2011).
Cd / Ca	Contributi totali delle decelerazioni e accelerazioni della frequenza cardiaca all'HRV.
SDNNd / SDNNa	Rispettivamente: varianza totale dei contributi delle decelerazioni (prolungamenti degli intervalli RR) e accelerazioni (accorciamento degli intervalli RR).
PIP	Percentuale dei punti di flesso della serie di intervalli RR.
IALS	Inverso della lunghezza media dei segmenti di accelerazione/decelerazione.
PSS	Percentuale di segmenti corti.
PAS	Percentuale di intervalli NN nei segmenti alternati.

Tabella 2: features di carattere geometrico relative a intervalli battito-battito

In *Tabella 3* si trovano tutte le feature in frequenza relative agli intervalli battito-battito estratti dai sensori.

lf	Potenza nelle basse frequenze (0.04-0.15 Hz). Riflette un misto di attività simpatica e parasimpatica, ma nelle registrazioni a lungo termine riflette l'attività simpatica e può essere ridotta dall'antagonista beta-adrenergico propanolo
hf	Potenza nelle alte frequenze (0.15-0.40 Hz). Riflette i rapidi cambiamenti nella variabilità da battito a battito dovuti all'attività parasimpatica. A volte anche chiamata banda respiratoria perché corrisponde ai cambiamenti HRV legati al ciclo respiratorio e può essere aumentato con una respirazione lenta e profonda (circa 6 o 7 respiri per minuto) e ridotta dai farmaci anticolinergici o dal blocco vagale.
lf_hf_ratio	lf/hf. A volte utilizzato come indice per investigare l'equilibrio simpatico/vagale.
lfnu	Potenza a basse frequenze normalizzata
hfnu	Potenza ad alte frequenze normalizzata
total_power	Densità di potenza spettrale totale
vlf	Potenza nelle bassissime frequenze (0.003-0.04 Hz). Riflettono un ritmo intrinseco prodotto dal cuore che è modulato principalmente dall'attività simpatica.

Tabella 3: fetures in frequenza relative agli intervalli battito battito

Per poter confrontare i valori delle *features* estratte, sono stati calcolati gli errori relativi percentuali, in particolare sono stati eseguiti tre confronti diretti (di seguito si usa la stessa nomenclatura utilizzata precedentemente):

- **garmin_bayes – garmin_media**
- **garmin_bayes - polar_raw**

- **polar_media - polar_raw**

I tre confronti sono stati scelti perché il primo permette di verificare lo scopo della tesi cioè il confronto diretto tra le due strategie di filtraggio, il secondo e il terzo per confrontare i due filtri disponibili al gold standard di riferimento. Per il calcolo dell'errore relativo percentuale si è utilizzata la seguente formulazione (4. 1).

$$errore\ rel = \frac{feature\ 1 - feature\ 2}{feature\ 2} \cdot 100$$

(4. 1)

Dove *feature 1* corrisponde alle features relative al tracciato che si vuole confrontare ad un altro tracciato da cui derivano le *features 2*. Questo calcolo viene eseguito per ognuna delle sessanta features estratte dai segnali. Di seguito si riporta un grafico rappresentativo di un esempio di confronto eseguito in particolare quello tra il segnale garmin smooth e il polar grezzo *Figura 35*.

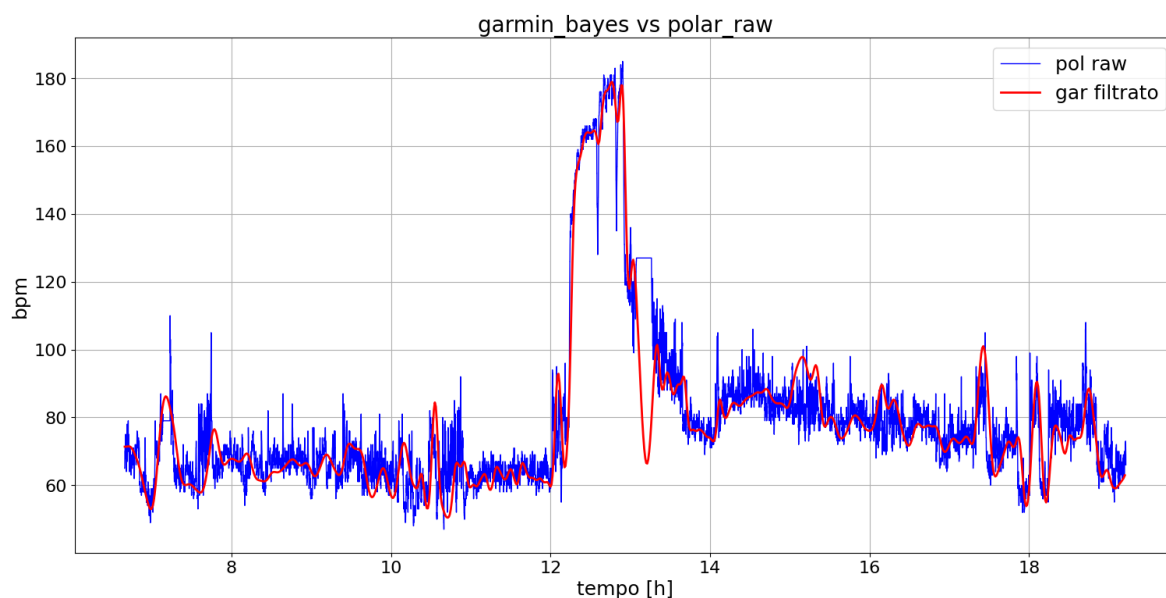


Figura 35: confronto tra segnale Polar grezzo (*polar_raw*) e Garmin filtrato con filtro bayesiano (*garmin_bayes*)

I risultati dei vari confronti cioè gli errori relativi percentuali sono riportati in tre heat map, le quali rispecchiano le tre tabelle sopra riportate *Figura 36*, *Figura 37*, *Figura 38*.

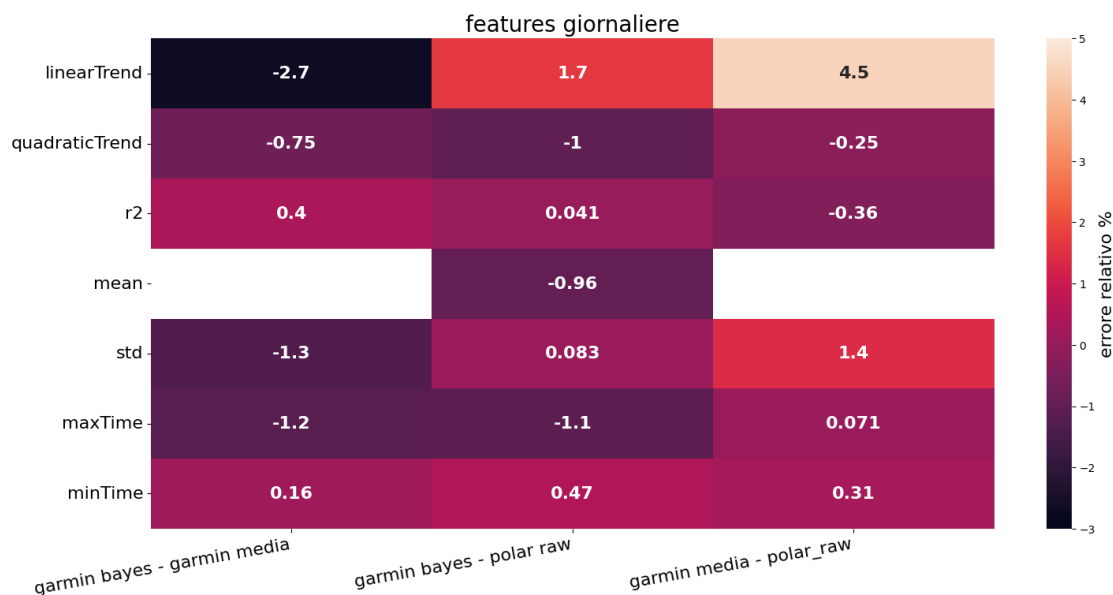


Figura 36: heat map rappresentante gli errori relativi delle features giornaliere tra i confronti: *garmin_bayes-garmin_media*, *garmin_bayes-polar_raw* e *garmin_media-polar_raw*

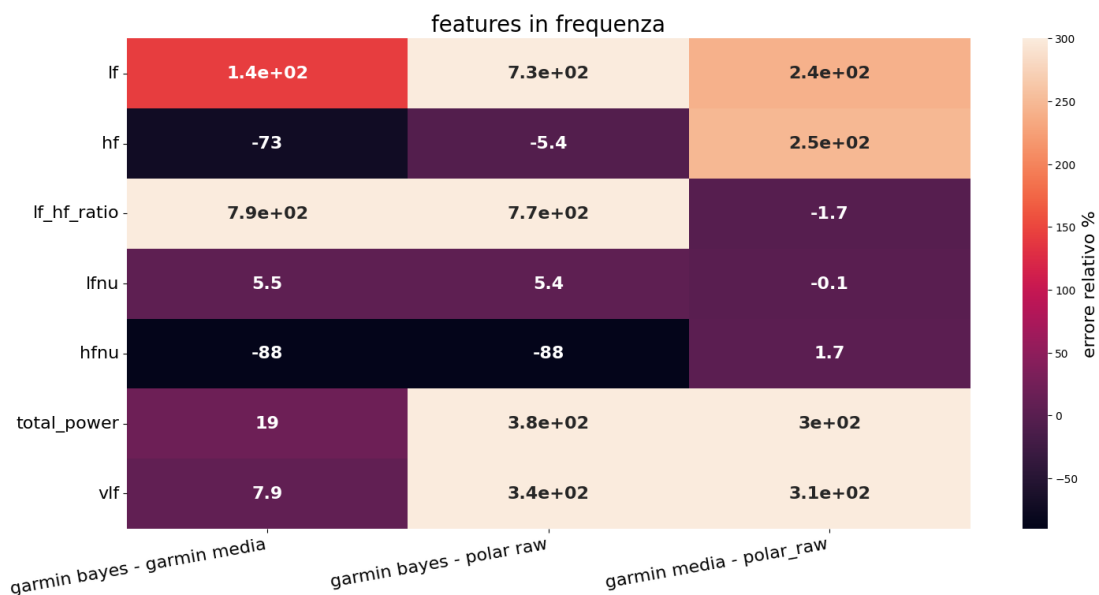


Figura 37: heat map rappresentante gli errori relativi delle features in frequenza tra i confronti: *garmin_bayes-garmin_media*, *garmin_bayes-polar_raw* e *garmin_media-polar_raw*

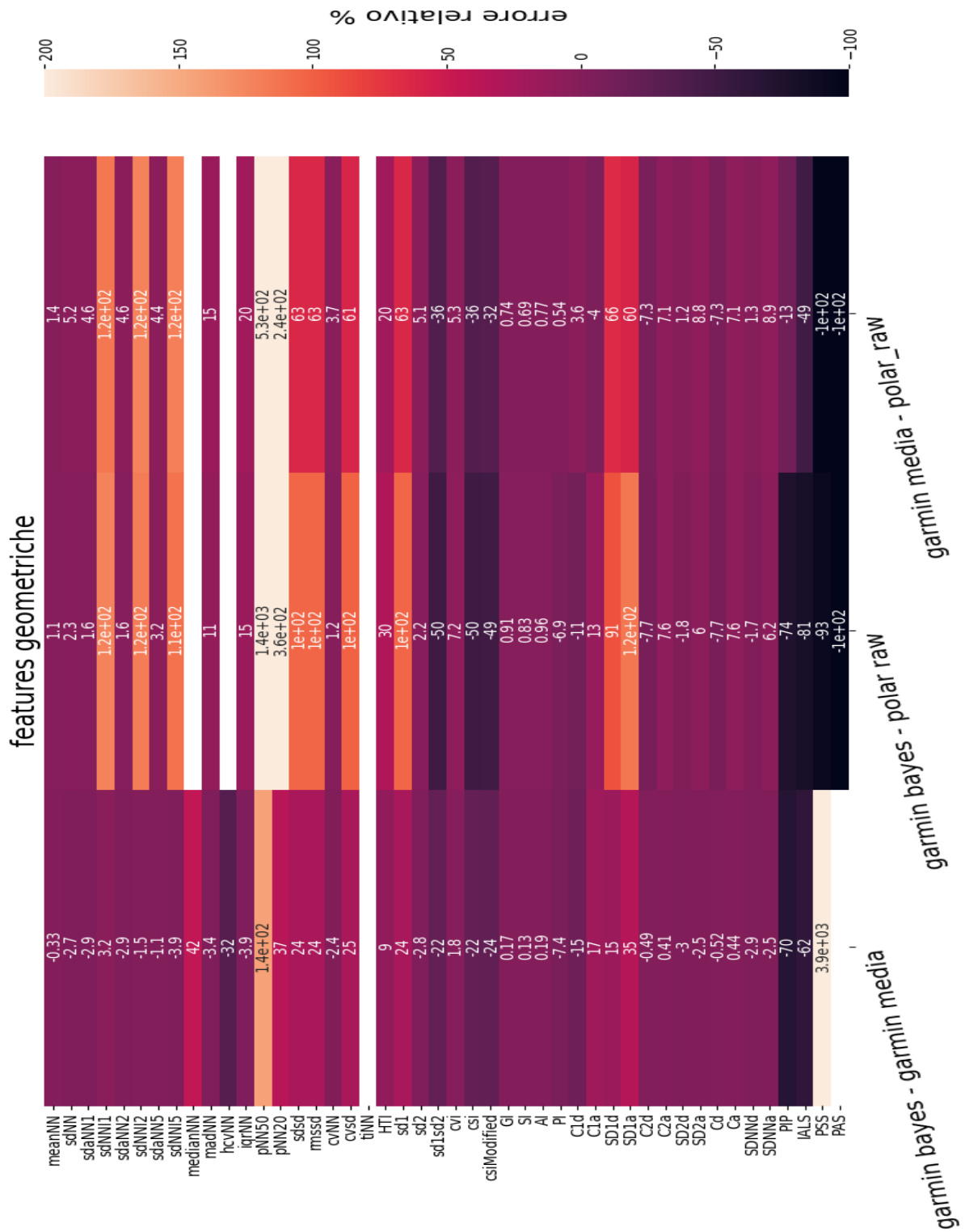


Figura 38: heatmap rappresentante gli errori relativi delle features geometriche tra i confronti: garmin_bayes-garmin_media, garmin_bayes-polar_raw e garmin_media-polar_raw

Dall'analisi dei risultati si osserva come i filtraggi nel lungo periodo [Figura 36] non modificano eccessivamente le features estratte rispetto a quelle provenienti dal gold standard, quindi gli andamenti giornalieri con le relative caratteristiche vengono preservate. In particolare, si vede come gli errori relativi percentuali siano tutti prossimi allo zero, sia nei confronti tra i due filtraggi e il gold standard, sia nel confronto diretto. Il parametro che riporta valori più alti è linearTrend, ma si vede come il filtraggio bayesiano sia più vicino al polar di quanto non lo sia il filtraggio a media mobile. Per quanto riguarda le features in frequenza [Figura 37] si hanno degli errori relativi percentuali molto maggiori con valori estremamente alti anche superiori al 100%, questo è dovuto al fatto che i filtraggi riducendo i picchi nel segnale ne modificano lo spettro. In particolare, si può osservare che tutte le feature riportano un valore maggiore nel confronto tra filtraggio bayesiano e gold standard rispetto a quello media mobile e gold standard, anche questo aspetto era prevedibile in quanto il filtraggio bayesiano compie un denoising maggiore di quello a media mobile. Nel confronto diretto si può vedere che il rapporto tra basse e alte frequenze (lf_hf_ratio) è il parametro con errore relativo maggiore pari quasi al 800%, questa situazione si giustifica dal fatto che le basse frequenze (lf) hanno un valore di 123% mentre le alte frequenze (hf) di -73%, quindi distanze molto ampie in valore assoluto ma con segni opposti e di conseguenza il loro rapporto ne risulta ampliato. Inoltre, come detto in precedenza le alte e basse frequenze con relativi segni dimostrano che il filtraggio bayesiano, paragonato alla media mobile, compie uno smorzamento nelle alte frequenze più consistente rispetto alle basse per la maggiore perdita di oscillazioni. In ultima analisi, situazione molto variegata risulta essere quella degli errori relativi percentuali relativi alle features geometriche consultabili in Figura 38. In questa analisi è bene notare che in diverse features (meanNN, sdNN, sdaNN1/2/5, Gi, Si, Ai, Pi, C1d, C1a, C2d, C2a, Cd, Ca, SD2d, SD2a, SDNNd, SDNNa) gli errori relativi percentuali dei confronti sono valori vicini tra loro e prossimi allo zero, quindi in questo caso si può dire che i tre segnali presi in considerazione non differiscono relativamente a questi aspetti. Altre features che invece presentano errori relativi alti nei confronti tra i due filtraggi e il Polar sono per esempio pNN20, pNN50. Ciò significa che per via dell'eliminazione di tante piccole oscillazioni, rispetto al tracciato di riferimento si perdono molti intervalli R-R e di conseguenza quelli maggiori di 20 o 50 ms in rapporto al totale di intervalli risultano molti di più. Infine, PIP, IALS, PSS, PAS sono i quattro parametri che differiscono maggiormente con

segno negativo nei confronti tra filtraggi sia baiyesiano che media mobile con il gold standard.

Capitolo 5

Conclusioni

Questo lavoro di tesi si posiziona all'interno del progetto BRAINTEASER, progetto europeo con lo scopo di aiutare pazienti affetti da sclerosi multipla e sclerosi laterale amiotrofica e i loro medici nella gestione della malattia con soluzioni centrate sull'utente. L'obiettivo è quello di raccogliere e monitorare parametri vitali tramite un sensore indossabile, integrarli con dati clinici retrospettivi ed infine utilizzare questa informazione come input ad algoritmi di intelligenza artificiale.

In particolare, in questa tesi si è sviluppato un algoritmo di filtraggio del segnale Heart Rate acquisito dal sensore Garmin Vivoactive4, basato su smoothing bayesiano, valutando i risultati ottenuti in termini di errori percentuali relativi che consentono un confronto tra le due tecniche (media mobile e smoothing bayesiano) ed il gold standard di riferimento. Tutto ciò è stato svolto in ambiente Python creando una nuova pipeline in grado di implementare questo algoritmo automaticamente.

In conclusione, dopo il lavoro e le analisi svolte in questa tesi, si può affermare che il filtraggio bayesiano implementato svolge a pieno la sua funzione di denoising. In particolare, la tecnica implementata è in grado di gestire autonomamente dei parametri, come per esempio la varianza del processo di rumore di misura bianco, che varia all'interno della giornata. Inoltre, i risultati sugli errori relativi percentuali calcolati sulle features estratte dai segnali indicano quello che ci si poteva aspettare. Paragonando il segnale filtrato con quello di riferimento (gold standard) prelevato dal Polar H10, si evince che il filtraggio elimina gran parte delle oscillazioni più piccole (alte frequenze) e mantiene inalterate le features

rappresentanti l'andamento a lungo termine nel tempo. Mentre non si riscontrano differenze ipoteticamente significative tra il segnale filtrato con smoothing bayesiano e quello filtrato con la tecnica della media mobile, quindi si può concludere che la pipeline implementata svolge a pieno le sue funzioni, ma all'interno del progetto BRAINTEASER non si riscontra una necessità di aumentare il tempo computazionale a discapito di una variazione nelle features che sarà da valutare successivamente all'implementazione di modelli di Intelligenza Artificiale che riceveranno in input le features estratte.

5.1 Steps futuri

Risultano aperte delle questioni e delle migliorie da studiare per aumentare ulteriormente la qualità del filtraggio. In prima istanza si è trattato il rumore di misura additivo alla stima come un processo di rumore bianco. È emerso che questa ipotesi non fosse rispettata in toto e quindi si potrebbe implementare il processo di rumore additivo come un processo autoregressivo pilotato da rumore bianco. Questa tecnica è già stata utilizzata su diverse tipologie di segnale, per esempio, per modellizzare il rumore di segnali EEG sulla singola sweep [28] [29] e su tracciati di CGM [30]. Inoltre, si potrebbe valutare la possibilità di avere finestre di lunghezze differenti per assicurarsi che i residui siano inerenti alle ipotesi fatte e si può considerare di unire l'ultima finestra, se molto breve, alla penultima per garantirne un miglior filtraggio. In seguito, sarebbe utile trovare dei criteri che permettano di scegliere in modo automatico i parametri utente in ingresso alla pipeline (per esempio lunghezza della finestra) e si dovrebbero adottare tecniche per ridurre il tempo computazionale. Infine, sarebbe interessante capire e giustificare il motivo del ritardo del segnale Garmin rispetto a quello Polar.

Bibliografia

- [1] “Brainteaser Home Page.” <https://brainteaser.health/> (accessed Nov. 12, 2022).
- [2] “Brainteaser milestone.” <https://brainteaser.health/milestones/> (accessed Nov. 12, 2022).
- [3] M. Dehghani, K. J. Kim, and R. M. Dangelico, “Will smartwatches last? factors contributing to intention to keep using smart wearable technology,” *Telematics and Informatics*, vol. 35, no. 2, pp. 480–490, May 2018, doi: 10.1016/j.tele.2018.01.007.
- [4] A. Sharma, M. Badea, S. Tiwari, and J. L. Marty, “Wearable Biosensors: An Alternative and Practical Approach in Healthcare and Disease Monitoring,” *Molecules*, vol. 26, no. 3, p. 748, Feb. 2021, doi: 10.3390/molecules26030748.
- [5] “Logo Brainteaser.” http://brainteaser.health/wp-content/uploads/2021/02/Brainteaser_LogoExtended.png (accessed Nov. 12, 2022).
- [6] “Brainteaser partners .” <https://brainteaser.health/partners/> (accessed Nov. 12, 2022).
- [7] M. Zanon, “Design of a pipeline for processing vital signals from wearable sensors in individuals affected by chronic diseases,” Università degli Studi di Padova, Padova, 2021.
- [8] “Garmin vivoactive® 4.” <https://www.garmin.com/it-IT/p/643382> (accessed Nov. 14, 2022).
- [9] “Garmin vivoactive® 4 specifiche.” <https://www.garmin.com/it-IT/p/643382#specs> (accessed Nov. 14, 2022).
- [10] M.-B. Hossain, J. Lazaro, Y. Noh, and K. H. Chon, “Denoising Wearable Armband ECG Data Using the Variable Frequency Complex Demodulation Technique,” in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, Jul. 2020, pp. 592–595. doi: 10.1109/EMBC44109.2020.9175665.
- [11] M. A. Motin, C. K. Karmakar, and M. Palaniswami, “Robust Heart Rate Estimation During Physical Exercise Using Photoplethysmographic Signals,” in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Jul. 2018, pp. 494–497. doi: 10.1109/EMBC.2018.8512405.
- [12] M.-T. Liao *et al.*, “Impact of recording length and other arrhythmias on atrial fibrillation detection from wrist photoplethysmogram using smartwatches,” *Sci Rep*, vol. 12, no. 1, p. 5364, Dec. 2022, doi: 10.1038/s41598-022-09181-1.

- [13] J. Tejedor, C. A. García, D. G. Márquez, R. Raya, and A. Otero, "Multiple Physiological Signals Fusion Techniques for Improving Heartbeat Detection: A Review," *Sensors*, vol. 19, no. 21, p. 4708, Oct. 2019, doi: 10.3390/s19214708.
- [14] G. de Nicolao, G. Sparacino, and C. Cobelli, "Nonparametric input estimation in physiological systems: Problems, methods, and case studies," *Automatica*, vol. 33, no. 5, pp. 851–870, May 1997, doi: 10.1016/S0005-1098(96)00254-3.
- [15] A. Facchinetti, G. Sparacino, and C. Cobelli, "An Online Self-Tunable Method to Denoise CGM Sensor Data," *IEEE Trans Biomed Eng*, vol. 57, no. 3, pp. 634–641, Mar. 2010, doi: 10.1109/TBME.2009.2033264.
- [16] A. Facchinetti, G. Sparacino, and C. Cobelli, "Online Denoising Method to Handle Intraindividual Variability of Signal-to-Noise Ratio in Continuous Glucose Monitoring," *IEEE Trans Biomed Eng*, vol. 58, no. 9, pp. 2664–2671, Sep. 2011, doi: 10.1109/TBME.2011.2161083.
- [17] S. Föll *et al.*, "FLIRT: A feature generation toolkit for wearable data," *Comput Methods Programs Biomed*, vol. 212, p. 106461, Nov. 2021, doi: 10.1016/j.cmpb.2021.106461.
- [18] "Python." <https://www.python.org/> (accessed Nov. 12, 2022).
- [19] "Python, PYPI." <https://pypi.org/> (accessed Nov. 12, 2022).
- [20] "K", "R" "Srinath," "Python – The Fastest Growing Programming Language ," *International Research Journal of Engineering and Technology (IRJET)*, vol. 04, no. 12, pp. 354–357, Dec. 2017.
- [21] S. Karppinen and M. Vihola, "Conditional particle filters with diffuse initial distributions," *Stat Comput*, vol. 31, no. 3, p. 24, May 2021, doi: 10.1007/s11222-020-09975-1.
- [22] "Json." <https://www.json.org/json-it.html> (accessed Nov. 16, 2022).
- [23] A. Staffini, T. Svensson, U. Chung, and A. K. Svensson, "Heart Rate Modeling and Prediction Using Autoregressive Models and Deep Learning," *Sensors*, vol. 22, no. 1, p. 34, Dec. 2021, doi: 10.3390/s22010034.
- [24] E. Gambi *et al.*, "Heart Rate Detection Using Microsoft Kinect: Validation and Comparison to Wearable Devices," *Sensors*, vol. 17, no. 8, p. 1776, Aug. 2017, doi: 10.3390/s17081776.
- [25] "Polar H10 ." <https://www.polar.com/it/sensors/h10-heart-rate-sensor> (accessed Nov. 11, 2022).
- [26] Polar Research and Technology, "Polar H10 Heart Rate Sensor System", Accessed: Nov. 11, 2022. [Online]. Available: <https://www.polar.com/en/img/static/whitepapers/pdf/polar-h10-heart-rate-sensor-white-paper.pdf>
- [27] Polar Research and Technology, "Polar Optical Heart Rate", Accessed: Nov. 11, 2022. [Online]. Available: <https://www.polar.com/en/img/static/whitepapers/pdf/polar-optical-heart-rate-white-paper.pdf>

-
- [28] C. D'Avanzo, S. Schiff, P. Amodio, and G. Sparacino, "A Bayesian method to estimate single-trial event-related potentials with application to the study of the P300 variability," *J Neurosci Methods*, vol. 198, no. 1, pp. 114–124, May 2011, doi: 10.1016/j.jneumeth.2011.03.010.
- [29] C. D'Avanzo, A. Goljahani, G. Pillonetto, G. de Nicolao, and G. Sparacino, "A multi-task learning approach for the extraction of single-trial evoked potentials," *Comput Methods Programs Biomed*, vol. 110, no. 2, pp. 125–136, May 2013, doi: 10.1016/j.cmpb.2012.11.001.
- [30] I. 'Siviero, "Sviluppo di algoritmi Bayesiani per il filtraggio del rumore di misura in sensori per il monitoraggio in continua della glicemia," Università degli Studi di Padova, Padova, 2021.

Appendice

Ap.1 - file JSON

Di seguito viene riportato un file JSON utilizzato in questo lavoro di tesi. Dal quale sono stati prelevati i segnali da elaborare. Nell'esempio riportato di seguito sono riportati solo pochi valori per ogni tracciato disponibile.

```
[
  {
    "base_date": 1652313600,
    "base_date_offset": 7200,
    "age": 37,
    "gender": "MALE",
    "heart_rate": {
      "min_heart_rate": 42.0000000000,
      "max_heart_rate": 133.0000000000,
      "mean_heart_rate": 65.0000000000,
      "resting_heart_rate": 44.0000000000,
      "data": [
        {
          "offset": 15,
          "value": 51
        },
        {
          "offset": 30,
          "value": 51
        },
        {
```

```
        "offset": 45,
        "value": 51
    }
]
},
"steps": {
    "data": [
        {
            "offset": 0,
            "value": 0
        },
        {
            "offset": 900,
            "value": 0
        },
        {
            "offset": 1800,
            "value": 0
        },
    ]
},
"sleep": {
    "duration": 27600.0000000000,
    "start_offset": 15000,
    "unmeasurable_sleep_duration_in_seconds": 840.0000000000,
    "deep_sleep_duration_in_seconds": 2760.0000000000,
    "light_sleep_duration_in_seconds": 13500.0000000000,
    "rem_sleep_in_seconds": 10500.0000000000,
    "awake_duration_in_seconds": 240.0000000000,
    "sleep_level_map": {
        "deep": [
```



```
{
  "offset": 660,
  "duration": 2220
},
{
  "offset": 3300,
  "duration": 540
}
],
"light": [
  {
    "offset": 0,
    "duration": 660
  },
  {
    "offset": 2880,
    "duration": 420
  }
],
"rem": [
  {
    "offset": 4140,
    "duration": 1980
  },
  {
    "offset": 9840,
    "duration": 540
  }
],
"awake": [
  {
```

```
        "offset": 8880,
        "duration": 60
    },
    {
        "offset": 20580,
        "duration": 60
    }
]
},
"spo2": [
    {
        "offset": 720,
        "value": 92
    },
    {
        "offset": 780,
        "value": 92
    },
    {
        "offset": 840,
        "value": 91
    }
],
"respiration": [
    {
        "offset": 60,
        "value": 14
    },
    {
        "offset": 120,
        "value": 13
    }
]
```

```
    },
    {
      "offset": 180,
      "value": 12
    }
  ]
},
"stress": {
  "data": [
    {
      "offset": 0,
      "value": 7
    },
    {
      "offset": 180,
      "value": 8
    },
    {
      "offset": 360,
      "value": 9
    }
  ]
},
"activity": {
  "data": [
    {
      "offset": 0,
      "met": 1.000000,
      "active_kilocalories": 0
    },
    {
```

```

    "offset": 900,
    "met": 1.000000,
    "active_kilocalories": 0
  }
]
}
}
]

```

Ap2 - pipeline

Per rendere più semplice e immediato l'uso della pipeline sono stati inseriti in *Tabella 4* tutte le variabili accettate e richieste in input dalla funzione, mentre in *Tabella 5* tutte quelle restituite come output dalla stessa.

<u>Nome variabile</u>	<u>Valore utilizzato</u>	<u>Significato della variabile</u>
file	<i>nome_file.json</i>	contenente i dati provenienti dal <i>Garmin</i>
passo	3	Utilizzato per il sotto campionamento, in particolare ogni quanti campioni prelevarne uno.
t_camp	15 (sec)	il tempo di campionamento del <i>Garmin</i> .
win_lenc	300	campioni per ogni finestra.
win_over	10	campioni di sovrapposizione delle finestre
innesco	2700 (sec)	secondi usati per l'innesco. Questo deve essere un valore che risulti un intero in termini di campioni. $2700/45 = 60$ campioni in uscita nell'innesco.

m	2	Ordine da cui deriva il grado di integrazione del processo rumore bianco
plot	True	Booleano che se True mostra i grafici della pipeline, se False riporta solo gli output.

Tabella 4: variabili di input alla pipeline

<u>Nome variabile</u>	<u>Significato della variabile</u>
uhat	Vettore contenente il segnale filtrato riferito ad <i>asse_t</i> .
res_ass	Vettore contenente i residui assoluti riferiti all'asse <i>hr_off</i> .
res_norm	Vettore contenente i residui normalizzati riferiti all'asse <i>hr_off</i> .
hr_off	asse_dei tempi per plottare entrambi i residui e σ^2 .
asse_t	Asse dei tempi per plottare vettore <i>uhat</i> .
sigma^2	Vettore contenente la σ^2 riferita all'asse <i>hr_off</i> .
kernel	Vettore contenente i pesi per riconciliazione.

Tabella 5: variabili di output dalla pipeline

Ap3 – funzioni implementate

Di seguito viene riportata una tabella con tutte le funzioni implementate durante il lavoro di tesi e tutte le variabili di input e output di ciascuna.

Nome funzione	Input	output
listToArray <i>(utilizzata per convertire una lista di due colonne in due vettori)</i>	<ul style="list-style-type: none"> • data: lista estratta dal json fornito dal garmin. Per es. <code>hr = data['heart_rate']['data']</code> 	<ul style="list-style-type: none"> • Hr_off: vettore asse dei tempi in secondi. • Hr_off_ore: vettore asse dei tempi in ore • Hr_val: vettore con valori hr da plottare con uno degli altri due.
findHole <i>(utilizzata per trovare i punti precedente ai missing samples e relativa lunghezza del buco)</i>	<ul style="list-style-type: none"> • t_camp: tempo di campionamento del segnale; • data_off: vettore asse x. • data_val: vettore dei valori relativi a data_off. 	<ul style="list-style-type: none"> • hole: valore dell'asse x in sec precedenete a uno o più campioni mancanti • hole_ore: come hole però asse_x in ore • val_hole: valore di data_val relativo al valore di hole • len_hole: numero di missing samples consecutivi.
app_baye_3crit <i>(funzione utilizzata per filtrare un segnale utilizzando lo smoothing bayesiano e il criterio di</i>	<ul style="list-style-type: none"> • x: asse dei tempi del vettore da filtrare • y: asse dei valori del vettore da filtrare 	<ul style="list-style-type: none"> • uhat: vettore contenete il segnale filtrato • res: vettore dei residui • gamma: valore calcolato

<p><i>consistenza sopra descritto)</i></p>	<ul style="list-style-type: none"> • B: matrice utilizzata come descrittiva del rumore di misura tale che $\Sigma(v) = \sigma^2 \cdot B$ • tv: vettore contenente la griglia virtuale • m: ordine di integrazione del modello di rumore bianco integrato 	<p>del gamma finale ottimo</p> <ul style="list-style-type: none"> • itergamma: numero di iterazioni necessarie per ottenere il gamma ottimo • cov_u: matrice di covarianza dell'errore di stima, utilizzata per calcolare intervallo di confidenza • sigma2: varianza dell'errore di misura.
<p>creaHole <i>(utilizzata per creare artificialmente dei buchi nel segnale e quindi poter testare successivamente lo smoothing bayesiano su segnali con griglia reale non uniforme)</i></p>	<ul style="list-style-type: none"> • x: vettore contenente asse dei tempi • y: vettore contenente valori del segnale relativi a x • x_start: valore di x da cui si vuole iniziare a eliminare i campioni • hole_len: numero di campioni da eliminare a partire da x_start 	<ul style="list-style-type: none"> • x: vettore dell'asse dei tempi con tempi mancanti • y: vettore dei valori con valori mancanti
<p>export_dict <i>(utilizzato per esportare i vettori filtrati e non e utilizzarli successivamente per la features extraction)</i></p>	<ul style="list-style-type: none"> • hr_off: asse dei tempi che diventa chiave del dizionario che si esporta • hr_val: valore del segnale relativo all'asse dei tempi, diventa il valore che assume 	<ul style="list-style-type: none"> • Non si hanno variabili di uscita ma solamente la creazione di un file compresso con il nome <i>name</i> fornito, salvato nella cartella di lavoro.

	<p>ciascuna chiave nel dizionario</p> <ul style="list-style-type: none">• name: stringa.z con cui si salva il dizionario compresso nella cartella di lavoro.	
--	---	--

Tabella 6: funzioni implementate durante tutto il lavoro di tesi con le relative variabili di input e di output

Ringraziamenti...

Innanzitutto, desidero ringraziare il prof. Facchinetti Andrea per l'opportunità di lavorare in questo progetto insieme a lui e al suo gruppo di ricerca, dandomi la possibilità di una ricca crescita professionale e personale. Inoltre, intendo ringraziare il dott. Cossu Luca e il prof. Cappon Giacomo, i quali hanno saputo sicuramente darmi sempre un supporto e dei consigli utili in tutte le fasi del lavoro e per essersi sempre offerti disponibili per i vari test.

Ringrazio tutta la mia famiglia per essermi stata di grande supporto, in particolare: mia mamma Fabiola per avermi veramente sopportato tutti i giorni e aver vissuto con me tutti gli umori 'altalenanti' di questo periodo, mio babbo Luciano per avermi aiutato nelle decisioni difficili, come dice lui "in modo democratico", giungendo insieme alla scelta che fino ad ora si è rivelata sempre la più corretta e mia sorella Lucrezia per avermi insegnato che per ottenere dei risultati bisogna metterci tanta dedizione.

Inoltre, devo ringraziare la mia fidanzata Alessia, perché con il suo animo gentile e pacato mi ha sempre dato la spinta per poter raggiungere questo traguardo e ha vissuto con me in piena sintonia, tutti i momenti di questa esperienza spronandomi sia da vicino che a distanza, non facendomi mai sentire solo in questo percorso.

Infine, ma non per importanza, ringrazio tutti i miei amici: i romagnoli perché ci conosciamo da una vita e i quali sono sempre stati capaci di farmi vivere momenti felici e spensierati; e i ragazzi di Padova, sia i compagni di corso con i quali sono state fatte sessioni di studio importanti, sia il buon vicinato di via G. Giardino per le chiacchiere e la compagnia che hanno reso questi due anni più leggeri e spensierati.