Analisi con Fotopletismografo

Ostanello - Marcato - Da Re - Di Giammarino - Ravagnan

Maggio 2022

Contents

1 Introduction

Al giorno d'oggi, le **malattie cardiovascolari** rappresentano ancora oggi un grosso problema. Nel nostro Paese, in particolare, sono la **principale causa di morte**: all'anno muoiono più di 230 mila persone all'anno tra ischemie, infarti, malattie del cuore e cerebrovascolari, circa il 35,8% di tutti i decessi (32,5% nei maschi e 38,8% nelle femmine). Quindi, per evitare che sia troppo tardi nella gestione della malattia, è necessario fare prevenzione. Un esempio di prevenzione che si prende in esame è la fotopletismografia (PPG). E' un biosensore cui il suo metodo di misurazione del battito cardiaco, è basato sul campionamento della luce modulata dai vasi sanguigni, che si espandono e si contraggono seguendo i battiti. Questo sensore è integrato nel MITCH (Multipurpose InerTial Chamaleon) che, abbinato ad un accelerometro, gli algoritmi filtrano anche gli artefatti da movimento che interferiscono con le misurazioni attribuibili al battito del cuore effettuate attraverso la Ppg. La Ppg è, anche, particolarmente adatta per applicazioni come i bracciali e gli orologi per sportivi i quali, risultando comodi ed indossabili, non interferiscono al loro normale svolgimento della loro attività sportiva.

L'obbiettivo dell'analisi è quello di arrivare a misurare in tempo reale i battiti per minuto, individuati tramite il PPG. Idealmente, si vorrebbe costruire anche un grafico che mostra la variabilità del battito per predirre eventuali problemi che possono sorgere in caso un battito particolarmente irregolare.

FONTI:
`https://www.degasperis.it/malattie-cardiovascolari-prima-causa-di-morte-in-italia.html`
`https://www.elettronicanews.it/un-biosensore-per-la-misurazione-del-battuto-cardiaco/`

2 Methods

Per cominciare l'analisi ed il calcolo dei battiti del cuore al minuto è stata necessaria una raccolta dati utilizzando il sensore foto pletismografo ed il MITCH.

Per raccogliere i dati sono state eseguite alcune misurazioni di circa 1 minuto l'una per avere un periodo di tempo adatto allo scopo dell'analisi.

Per iniziare la raccolta dati è stato posizionato il dito sopra il sensore in modo da permettere alla luce emessa da esso di "passare" attraverso il dito.

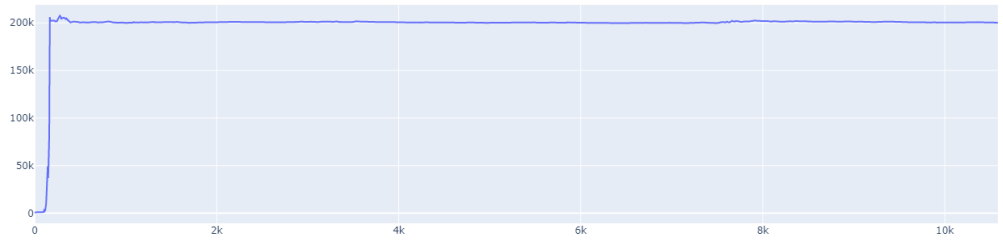
In questo modo infatti in sensore captava la luce di ritorno (rossa e infrarossa) e permetteva di capire l'ossigenazione del sangue in base ad essa.

	Timestamp	Acc.X	Acc.Y	Acc.Z	Gyro.X	Gyro.Y	Gyro.Z	Magn.X	Magn.Y	Magn.Z	...	P.9	P.10	P.11	P.12	P.13	P.14	P.15	PPG_RED.	PPG_IR.
0	1644926150016	-28.304	0.000	991.616	3.36	0.00	0.00	439.5	-196.5	-693.0	...	0	0	0	0	0	0	0	0	954
1	1644926150026	-23.912	4.880	1007.720	5.39	-6.09	4.41	-99.0	33.0	-766.5	...	0	0	0	0	0	0	0	968	929
2	1644926150036	-19.520	3.904	1002.352	13.72	-22.54	2.31	-100.5	40.5	-763.5	...	0	0	0	0	0	0	0	961	931
3	1644926150046	-14.152	-6.832	1016.504	4.34	-7.98	1.19	-102.0	42.0	-763.5	...	0	0	0	0	0	0	0	960	929
4	1644926150056	-5.368	-6.344	1018.456	0.00	-1.12	0.07	-100.5	34.5	-760.5	...	0	0	0	0	0	0	0	951	922
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
515	1644926175190	-16.592	-5.368	1017.480	0.00	-0.98	0.07	-82.5	37.5	-768.0	...	0	0	0	0	0	0	0	151431	178957
516	1644926175200	-10.736	2.440	1012.112	0.00	-0.91	0.00	-79.5	42.0	-772.5	...	0	0	0	0	0	0	0	151431	178926
517	1644926175210	-16.104	0.976	1012.600	0.00	-0.98	0.07	-79.5	37.5	-768.0	...	0	0	0	0	0	0	0	151479	178913
518	1644926175220	-22.448	0.000	1010.648	-0.14	-0.91	0.00	-78.0	37.5	-768.0	...	0	0	0	0	0	0	0	151509	178928
519	1644926175230	-21.472	8.784	1007.720	-0.21	-1.05	0.00	-79.5	31.5	-765.0	...	0	0	0	0	0	0	0	151552	178947

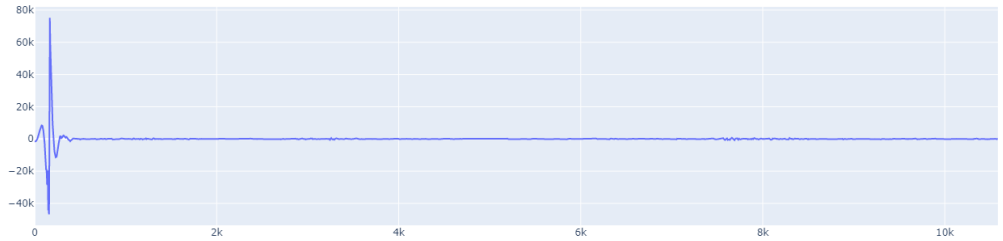
Dai dati forniti è stata poi utilizzata la luce rossa per il calcolo dei battiti al minuto, questa scelta è stata presa data la maggior qualità del segnale generato attraverso essa.

3 Algorithm

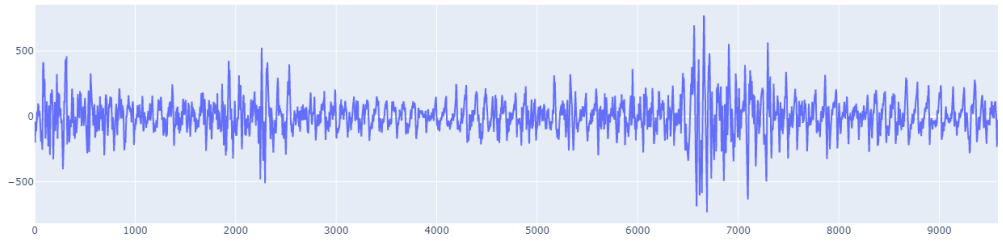
Si parte dall'analisi del segnale IR(Infra Rosso) non filtrato per analizzarne il comportamento.



Si nota un grosso cambiamento nella parte in cui il dito viene appoggiato al sensore. Per togliere tale cambiamento viene usata la serie di Fourier (`https://it.wikipedia.org/wiki/Serie_di_Fourier`) che ci dice che il segnale è una rappresentazione periodica mediante la combinazione lineare di funzioni sinusoidali e il filtro butterworth (`https://it.wikipedia.org/wiki/Filtro_Butterworth`) con un highpass, ossia togliendo la parte di segnale più bassa, a frequenza 0.7Hz controllata sui 3 punti vicini, con frequenza di base 100Hz, che è la frequenza di campionamento.



Analizzando ora il segnale "appiattito" cerchiamo di individuare la fascia superiore del segnale da "eliminare" per ottenere un battito cardiaco più comprensibile, viene pertanto preso in considerazione un intervallo che non coprende i primi 1000 campioni.



Su questo viene applicato un filtro butterworth lowpass, che esclude la parte di segnale più alta, di frequenza 3.5Hz sui 3 punti vicini a 100Hz. Siccome applicare i due filtri in separata sede sarebbe

un peso per il processore vengono uniti in un unico filtro butterworth bandpass, ossia che esclude gli intervalli minori di una soglia min e gli intervalli maggiori di una certa soglia max, con intervallo [0.7, 3.5]Hz con 3 vicini a 100Hz. Per calcolare i picchi viene usata una funzione a cui viene passato il segnale, un minimo e una distanza, questa funzione poi analizza il segnale, "esclude" la parte di segnale sotto il minimo e trova i picchi presenti che siano distanti almeno quanto la distanza data. Per calcolare i battiti al minuto viene poi utilizzata la seguente formula:

$$\frac{frequenzadicampionamento}{media(picchiposizione(i+1) - picchiposizione(i))} * 60$$

per i che va da 0 a uno in meno della lunghezza dell'array dei picchi e moltiplicato per 60 allo scopo di trovare i battiti al minuto. Trovato l'algoritmo resta da decidere come applicarlo in maniera attiva e non su un dataframe. A tal scopo viene creata una classe di python con 5 funzioni principali oltre alla funzione di inizializzazione, che viene chiamata Buffer(). Nell'inizializzazione vengono create due liste, 'buffer' e 'beat_buffer', vuote; viene creata una variabile chiamata 'lastbeat' con valore 0 e un'altra variabile '_freq' che contiene la frequenza di campionamento, che nel nostro caso viene inizializzata di base a 100, ma è possibile modificarla quando viene chiamata. La prima funzione della classe è quella chiamata 'add_dato(dato)', che aggiunge a 'buffer' il dato e poi controlla che la dimensione di 'buffer' non superi della frequenza di campionamento *10 in modo da avere un massimo di 10 secondi di dati, e se supera toglie il primo elemento a 'buffer'. La seconda funzione è 'fiter()', la quale crea il filtro menzionato sopra, lo applica a 'buffer' e ritorna i dati filtrati. La terza 'check()' è una funzione di controllo che controlla che 'buffer' abbia almeno 2 secondi di dati in quel caso esegue la funzione 'fiter()', crea una lista dei valori assoluti dei dati filtrati, fa la media di questi dati e controlla che sia compresa tra 520 e 40, in caso contrario resetta la lista 'buffer', e infine ritorna i dati filtrati. La quarta funzione 'battito(numerose, videose, unfiltered)' richiama la funzione 'check()' e salva in una variabile 'filtrato' i dati filtrati, controllo che la lunghezza di 'buffer' se supera un quarto della lunghezza massima calcola il battito al minuto e la salva in 'beat_buffer', con un sistema di controllo che non superi la lunghezza di 30, se numerose è vero viene poi cancellato l'output sotto stampata la media di 'beat_buffer' ogni 20 misurazioni in caso sia diversa da quella precedente, a seconda se 'videose' e/o 'unfiltered' sono veri ritorna 'buffer' filtrato e/o 'buffer' non filtrato. La quinta è 'lastmesure()' che stampa la media degli ultimi 30 bmp calcolati, sotto forma di numero intero, e ritorna un array di numpy con gli utimi 30 bpm calcolati.

4 Results

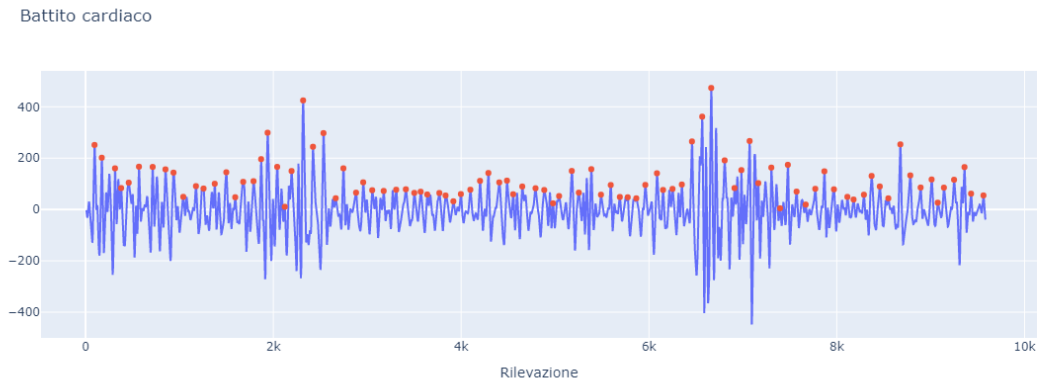
Per rappresentare i risultati si è ricreato un grafico vuoto aggiornato ogni volta che i dati del battito venivano analizzati e restituiti dalle funzioni di filtro. In particolare, è stato deciso di agire nel seguente modo:

1. Vengono prese le misurazioni del sensore
2. Viene creato un oggetto Buffer() a cui vengono aggiunte le informazioni della colonna IR tramite la funzione *'add_dato'*
3. Viene invocato uno *'Sleep'* di un centesimo di secondo per simulare l'acquisizione live dei dati
4. Richiamando la funzione *'battito'* con i dovuti parametri possiamo scegliere di:
    - Aggiornare la figura inizializzata nella cella precedente con il ciclo di inserimento e lettura dati
    - Visualizzare solo il battito cardiaco lasciando il campo *'numerose'* invariato, in quel caso mostrerà solo il valore dei battiti sotto la cella

Con i dati rilevati abbiamo notato che nella parte iniziale dell'osservazione il battito varia intorno ai 55 BPM. Andando avanti con la rilevazione tuttavia si nota che il valore si stabilizza intorno ad una soglia più alta; una volta che tutti i dati sono stati raccolti infatti abbiamo in media al minuto tra i 65 e gli 75 battiti. Curioso è il picco momentaneo che si nota a circa 10 secondi dall'avvio dell'algoritmo; in quel momento si rilevano oltre i 90 battiti. Tale dato anomalo tuttavia non è abbastanza rilevante o differente dal resto per rendere nullo lo studio. Oltre al calcolo numerico è stato prodotto anche un grafico che visualizza la distribuzione dei battiti nei dati raccolti.

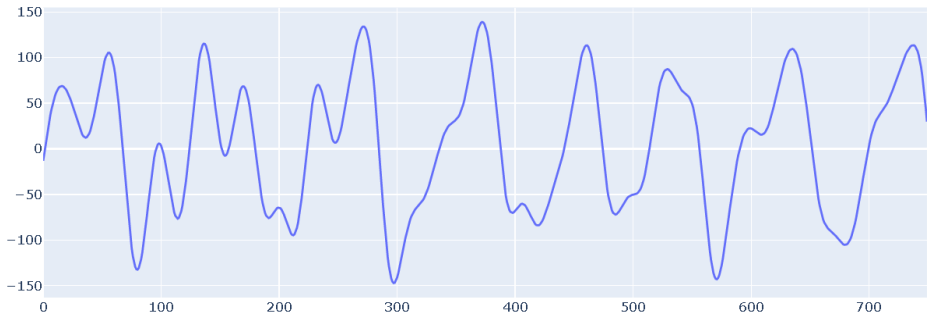
```
66 53 52 57 62 59 58 61 60 57 60 62 65 68 69 68 69 72 74 75 76 74 73 74 75 76 79
80 78 79 81 83 85 86 84 87 89 90 91 90 87 86 87 84 82 83 84 81 80 79 77 78 77 74
72 74 73 72 71 72 73 72 71 72 68 66 67 69 70 72 71 73 75 78 79 78 79 81 84 85 86
84 82 83 86 87 88 89 87 88 91 90 91 90 86 82 83 84 82 84 83 81 80 81 84 83 81 78
75 72 68 69 74 75 79 80 82 83 80 82 83 80 75 78 81 78 74 75 73 69 68 72 70 71 69
68 66 68 69 68 70 72 70 71 69 68 67 66 68 69 68 69 71 70 69 68 69 71 69 70 72 71
67 65 67 66 65 67 68 69 67 68 67 66 65 64 62 61 60 61 62 61 62 63 61 62 64 65 66
62 61 62 63 64 65 66 64 66 67 68 69 71 72 74 72 75 76 77 78 80 81 83 85 84 83 82
83 82 81 83 85 86 85 87 88 89 88 84 82 79 77 78 77 75 74 76 77 76 73 70 71 72 73
72 74 75 76 75 77 78 77 76 75 74 75 74 73 71 72 74 73 74 73 74 75 78 82 81 82 85
86 83 86 85 90 93 92 91 88 87 88 87 89 91 89 88 90 92 96 95 93 94 95 93 94 93 92
90 88 86 87 86 83 84 83 77 79 83 82 80 81 82 81 80 82 81 77 75 74 75 74 73 74 72
73 74 73 72 74 75 78 75 71 72 74 73 72 71 70 68 69 67 68 67 68 69 68 66 65
```

Media degli ultimi 30 BPM stampata ogni 0.2 secondi se diversa dalla precedente



5 Conclusion

Alla luce dei fatti, si può affermare che l'obiettivo che ci eravamo posti, ovvero quello di riuscire a visualizzare un segnale pulito e filtrato relativo al battito cardiaco,come si può notare dal grafico è stato centrato.



Per quanto l'algoritmo sia accettabile e gestisca diverse anomalie, si riscontrano due criticità in quanto , in alcune occasioni, il segnale creato all'interno del grafico compare filtrato e pulito, ma in altre occasioni risulta non del tutto filtrato. Questo genera una conseguenza poco piacevole e paradossale: quando il segnale è filtrato non è perfettamente sincronizzato con il resto del codice ma quando non si manifesta completamente pulito, sembra andare perfettamente in sincronia. La seconda anomalia, invece, è apparsa confrontando dei battiti *'normali'* con i battiti in risposta dall'algoritmo: quest'ultimi sembrerebbero essere accettabili ma,talvolta, mostrano variazioni un po' ambigue. Come si potrà immaginare, le motivazioni di tutto ciò possono essere molteplici, come ad esempio il fatto che i dati di partenza,essendo sporchi ,magari presi velocemente e, forse in maniera imprecisa, è stato più difficile del previsto riuscire a sincronizzare un segnale pulito in tempo reale senza alcun tipo di bug. Possiamo però, certamente non escludere migliorie future all'algoritmo, magari provando con altri dati e facendo ulteriori ricerche.

In conclusione; l'utilizzo dei filtri è stato un successo; vediamo infatti dalla figura sottostante la differenza del segnale prima e dopo essere stato processato.