



UNIVERSITÀ POLITECNICA DELLE MARCHE

FACOLTÀ DI INGEGNERIA

*Corso di Laurea in Ingegneria Biomedica*

# **Studio di modalità di elaborazione di biosegnali con piattaforma Arduino per la misura dello stress**

**Study of biosignal processing methods with Arduino  
platform for stress measurement**

*Tesi di laurea di:*

VALENTINA MAGNO

*Relatore:*

Prof.ssa SPINSANTE SUSANNA

*Correlatore*

Prof.ssa CECCHI STEFANIA

---

*Anno Accademico 2019-2020*

# Abstract

Saper riconoscere uno stato di stress è il primo passo verso l'elaborazione di metodi per la riduzione dello stesso al fine di ottenere un miglioramento del benessere della persona. Oltre all'aspetto psicologico e alla valutazione dello stress percepito, è di grande importanza poter definire lo stress da un punto di vista oggettivo tramite parametri fisiologici regolati dal Sistema Nervoso Autonomo.

L'obiettivo della tesi è in primis quello di acquisire frequenza cardiaca e risposta galvanica della pelle mediante un fotopletismografo e un dispositivo sensibile alla resistenza cutanea, e poi di stabilire il grado di correlazione tra i due parametri, sia in tempo reale tramite piattaforma Arduino sia successivamente all'acquisizione tramite MATLAB.

Si vuole inoltre mettere a confronto gli indicatori statistici utilizzati per il calcolo della correlazione, valutandone il trend per acquisizioni precedentemente effettuate.

# Indice

<b>INDICE .....</b>	<b>3</b>
<b>INTRODUZIONE .....</b>	<b>4</b>
<b>CAPITOLO 1: STATO DELL'ARTE .....</b>	<b>5</b>
1.1 IL LEGAME TRA SEGNALI FISIOLGICI, EMOZIONI E STRESS .....	7
1.2 METODI E STRUMENTI PER L'ANALISI DELLO STRESS BASATI SU HEART RATE E RISPOSTA GALVANICA DELLA PELLE .....	7
<b>CAPITOLO 2: STRUMENTAZIONE UTILIZZATA.....</b>	<b>9</b>
2.1 SENSORE PER L'ACQUISIZIONE DELLA GSR .....	9
2.2 FOTOPLETISMOGRAFIA .....	10
2.2.1 PRINCIPIO DI FUNZIONAMENTO .....	10
2.2.2 SENSORE .....	11
<b>CAPITOLO 3: METODO DI ELABORAZIONE DEI SEGNALI.....</b>	<b>13</b>
3.1 DEFINIZIONE DI CROSS-CORRELAZIONE.....	13
3.2 CALCOLO DELLA CROSS-CORRELAZIONE IN MATLAB .....	14
3.3 INDICE DI CORRELAZIONE DI PEARSON.....	15
3.4 SCRIPT ARDUINO .....	16
3.4.1 DICHIARAZIONE VARIABILI .....	17
3.4.2 FUNZIONE VOID SETUP .....	17
3.4.3 FUNZIONE VOID LOOP .....	18
<b>CAPITOLO 4: ACQUISIZIONI E ANALISI.....</b>	<b>20</b>
4.1 PROBLEMATICHE NELL'ACQUISIZIONE .....	20
4.1.1 ANNULLAMENTO DELLA DEVIAZIONE STANDARD .....	20
4.1.2 COMPLESSITÀ COMPUTAZIONALE .....	22
4.2 CONFRONTO TRA CROSS-CORRELAZIONE E INDICE DI CORRELAZIONE DI PEARSON .....	23
<b>CONCLUSIONI .....</b>	<b>27</b>
<b>BIBLIOGRAFIA.....</b>	<b>28</b>

# Introduzione

Le emozioni influenzano sia lo stato fisiologico sia quello psicologico e giocano un ruolo fondamentale nella vita di tutti i giorni. Le emozioni positive contribuiscono a migliorare l'efficienza nel lavoro e il benessere generale della persona, mentre quelle negative possono causare problemi di salute.<sup>[1]</sup>

Lo stress viene definito come la reazione a situazioni ambientali avverse che sfidano le abilità di adattamento di un individuo. Sebbene lo stress svolga una funzione fondamentale nell'affrontare ostacoli o problemi nel breve termine – come ad esempio nel caso di reazione di attacco o fuga – un'esposizione prolungata ad agenti stressanti ha effetti deleteri sull'organismo.<sup>[2]</sup> Numerosi sono gli studi che collegano lo stress a varie patologie quali ipertensione, malattie cardiache<sup>[3]</sup>, disturbi di ansia e depressione<sup>[4][5]</sup>, disturbi del sonno<sup>[6]</sup>. Inoltre lo stress comporta un peggioramento del benessere dell'individuo – basti pensare all'impatto sulla carriera scolastica o accademica degli studenti<sup>[7][8]</sup>, alla diminuzione di efficienza in ambito lavorativo<sup>[9]</sup> o agli effetti sulla guida.<sup>[10]</sup>

Per poter valutare il livello di stress nella vita quotidiana si ricorre allo studio di segnali fisiologici, soprattutto dei biosegnali acquisibili tramite dispositivi indossabili e non invasivi, come smartphones e smartwatches, nonché a basso costo.<sup>[11]</sup>

Nel primo Capitolo della presente tesi verrà presentato lo stato dell'arte, sia per quanto riguarda i meccanismi dietro al legame tra segnali fisiologici ed emozioni, sia per quanto riguarda i metodi e gli strumenti per l'analisi dello stress, in particolare quelli basati su frequenza cardiaca (HR) e risposta galvanica della pelle (GSR). Verrà in seguito descritta la strumentazione fornita dall'UNIVPM per l'acquisizione di suddetti parametri fisiologici (Capitolo 2). Per poter ottenere dati più accurati, nel Capitolo 3 si propone di individuare la correlazione non solo tra emozioni e biosegnali ma anche le correlazioni che intercorrono tra i biosegnali stessi. A tale scopo si introdurranno diversi tipi di indici statistici per la misura della correlazione e verranno proposti uno script MATLAB e uno sketch Arduino per il loro calcolo. Infine nel Capitolo 4 verranno risolte le problematiche

di implementazione dello sketch Arduino e verrà effettuata un'analisi comparativa dei risultati ottenuti dagli indicatori di correlazione utilizzati nel capitolo precedente.

# Capitolo 1

## Stato dell'arte

### 1.1 Il legame tra segnali fisiologici, emozioni e stress

I metodi di riconoscimento delle emozioni possono essere suddivisi in due principali categorie. La prima comprende lo studio di segnali facilmente osservabili, come espressioni facciali, linguaggio del corpo e caratteristiche del parlato. Tuttavia, si tratta di parametri poco affidabili, in quanto possono essere, in una certa misura, facilmente controllabili dal soggetto, e quindi non del tutto attendibili. Per questo motivo si tende a ricorrere alla seconda categoria di metodi, legata alla misura ed elaborazione di parametri fisiologici.<sup>[1]</sup>

Gli stimoli emotigeni – in particolari gli agenti stressanti – attivano l'asse simpatoadreno-midollare (SAM) e l'asse ipotalamo-ipofisi-surrene (HPA). Nel primo, il sistema nervoso autonomo simpatico stimola la produzione di adrenalina determinando, tra gli altri effetti, l'aumento della frequenza cardiaca, della pressione sanguigna e del volume sistolico del cuore, la dilatazione dei bronchi e dei vasi sanguigni che irrorano l'apparato muscolo-scheletrico e l'attivazione delle ghiandole sudoripare. Il secondo, invece, stimola la produzione di cortisolo e dunque la gluconeogenesi.<sup>[12]</sup> Pertanto, andando a misurare come variano certi segnali fisiologici, è possibile risalire al livello di stress del soggetto in esame.

Le tipologie di biosegnali e i metodi di acquisizione studiati sono diversi. Tra i più comuni si annoverano i parametri legati all'apparato cardiocircolatorio (frequenza cardiaca, variabilità della frequenza cardiaca, gittata cardiaca, pressione arteriosa), quelli legati all'apparato respiratorio (frequenza respiratoria, capacità funzionale residua, tempo di inspirazione e di espirazione), la temperatura della pelle e la risposta galvanica della pelle. Inoltre possono essere analizzati i tracciati EEG, ECG ed EMG.<sup>[1]</sup>

## 1.2 Metodi e strumenti per l'analisi dello stress basati su HR e GSR

L'obiettivo degli studi sul rilevamento dello stress è, in generale, quello di fornire dati inquadrati nel contesto dell'attività quotidiana. È dunque fondamentale adoperare dispositivi non invasivi e indossabili, quali smartphones, smartwatches e smart bands. Per questo motivo i segnali fisiologici più significativi sono la frequenza cardiaca (HR) e la risposta galvanica della pelle (GSR), dal momento che è possibile acquisirli attraverso dispositivi da polso.<sup>[11]</sup>

Diversi sono gli studi dell'ultimo decennio incentrati sull'analisi dello stress basati su HR e GSR. Ad esempio gli autori della pubblicazione [11] hanno sviluppato un sistema basato su HR, GSR, temperatura della pelle e dati forniti da un accelerometro per il riconoscimento di diversi livelli di stress (baseline, carico cognitivo, stress e fase di recupero) attraverso degli algoritmi di machine learning. Lo scopo dello studio è di dimostrare la validità di certi metodi di riduzione dello stress analizzando sia i segnali fisiologici acquisiti sia le testimonianze dei soggetti esaminati circa lo stress percepito.

Un esempio sul metodo di acquisizione dei segnali viene fornito dalla pubblicazione [13]. La durata delle acquisizioni è stata di 1-1.5 ore, durante le quali i partecipanti sono stati fatti sedere, per simulare al meglio uno scenario di vita quotidiana. Ai soggetti sono stati somministrati diversi agenti stressanti: semplici quesiti matematici da risolvere, tracce audio di diversa natura da ascoltare in cuffia, un test sull'effetto Stroop. Gli autori suggeriscono la possibilità di utilizzare anche simulazioni di situazioni stressanti mediante dispositivi per la realtà aumentata. Dallo studio è emerso che, analizzando un solo parametro, la sola GSR fornisce maggiori informazioni rispetto al solo HR, e che combinando i due segnali si ha un miglioramento dell'accuratezza dei risultati del 2,5%.

Alcuni studi si concentrano sulla strumentazione utilizzata, come in [14], in cui lo scopo è di progettare un solo dispositivo indossabile che acquisisca contemporaneamente la GSR e la variabilità della frequenza cardiaca. Altri invece si focalizzano sui parametri statistici che permettono di correlare tra loro i vari segnali<sup>[10][5]</sup> o sullo sviluppo di algoritmi di riconoscimento<sup>[11][15]</sup>.

La presente tesi considera, da un lato, l'acquisizione contemporanea di due differenti segnali, ovvero l'HR (mediante sensore fotopleletismografico) e la GSR (mediante sensori

da indossare sulle dita), e dall'altro l'elaborazione di un indice che correli i due segnali e ne evidenzi, dunque, le variazioni concordi o discordi.



# Capitolo 2

## Strumentazione utilizzata

### 2.1 Sensore per l'acquisizione della GSR

Il Sistema Nervoso Centrale elabora gli stimoli emotigeni attraverso il sistema limbico e invia dei segnali al Sistema Nervoso Autonomo, il quale rilascia dei neurotrasmettitori, tra cui adrenalina e noradrenalina, che vanno ad attivare o disattivare le ghiandole sudoripare.<sup>[16]</sup> Tali ghiandole rilasciano sudore, un liquido ipotonico ed elettrolitico composto principalmente da acqua e ioni ( $\text{Na}^+$ ,  $\text{K}^+$ ,  $\text{Cl}^-$ ). La variazione di concentrazione di ioni facilita il passaggio di corrente e fornisce un certo valore di conduttanza cutanea, ovvero di GSR.

Per misurare la GSR si utilizzano due elettrodi da posizionare in una delle aree più sensibili agli stimoli emotigeni – palmi delle mani, dita, piante dei piedi. Si applica poi una corrente a bassa intensità tra i due elettrodi e si osservano le sue variazioni. È dunque possibile calcolare la resistenza cutanea e, calcolandone il reciproco, la conduttanza cutanea, generalmente espressa in micro-Siemens ( $\mu\text{S}$ ).<sup>[17]</sup>

In questo studio è stato utilizzato il Grove – GSR Sensor<sup>[18]</sup> (Figura 2.1), con il quale è possibile misurare la GSR attraverso due elettrodi in Nickel posizionati sulle dita indice e medio del soggetto. Gli elettrodi sono collegati con due cavi di diverso colore alla scheda elettronica Grove – GSR; il cavo rosso indica l'alimentazione del dispositivo (pari a circa 5V, fornita dalla board Arduino UNO), mentre quello nero indica la messa a terra. Si forma così un circuito chiuso, di cui è possibile calcolare l'impedenza facendo il rapporto tra la differenza di potenziale tra gli elettrodi e la corrente che circola in essi attraverso la pelle.

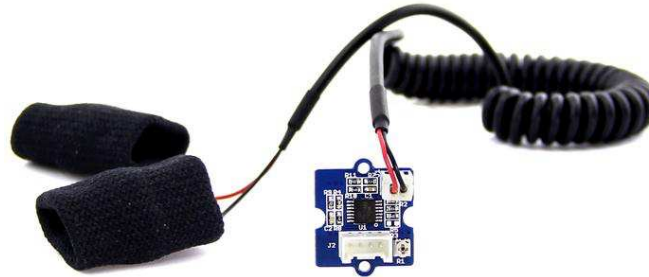


Figura 2.1 Grove – GSR Sensor

Tre fili collegano la scheda del sensore alla scheda Arduino Uno, e rappresentano alimentazione, terra e segnale. Quest'ultimo viene portato in ingresso ad Arduino e da esso convertito da analogico a digitale mediante un ADC a 10 bit, fornendo così valori numerici nell'intervallo 0-1023. Il valore di resistenza effettivo in Ohm si ottiene dai valori numerici forniti dall'ADC per mezzo della seguente formula

$$\text{Human Resistance} = \frac{(1024 + 2 \cdot \text{Serial\_Port\_Reading}) \cdot 10000}{512 - \text{Serial\_Port\_Reading}}$$

dove `Serial_Port_Reading` è appunto un numero compreso tra 0 e 1023, fornito dall'ADC.

Attraverso un cablaggio di tipo USB collegato a un PC, i dati vengono stampati a video nel monitor seriale dell'IDE di Arduino.

## 2.2 Fotopletismografia

### 2.2.1 Principio di funzionamento

La fotopletismografia (PPG) è una tecnica non invasiva a basso costo utilizzata per monitorare la frequenza cardiaca e basata sulla misurazione del volume dell'arteria mediante un LED che direziona la luce attraverso i tessuti biologici e un fotodiode che cattura la luce riflessa.

Quando la luce emessa dal trasmettitore penetra nel corpo, parte di essa viene assorbita, mentre una parte viene riflessa. A ogni ciclo cardiaco viene generata un'onda

pressoria che viene smorzata progressivamente dalle capacità elastiche dei vasi sanguigni e dalla viscosità del sangue. Tale onda provoca una variazione di volume all'interno dell'arteria che comporta variazioni dell'intensità della luce riflessa.<sup>[19]</sup> La frequenza con la quale viene rilevata l'onda pressoria viene considerata essere la frequenza cardiaca del soggetto in esame.

I LED più utilizzati sono quelli rossi e verdi. La luce a maggiore lunghezza d'onda – in particolare nell'infrarosso – penetra meglio nei tessuti; tuttavia, la luce a infrarosso è più sensibile agli artefatti da movimento rispetto a quella nel visibile, che viene dunque preferita nella maggior parte delle applicazioni.<sup>[20]</sup> Inoltre la luce verde viene assorbita di più dall'emoglobina, risultando in una più apprezzabile variazione di luce riflessa e in un miglior rapporto segnale-rumore.<sup>[19]</sup>

### 2.2.2 Sensore

Il sensore utilizzato in questo studio è il Grove – Finger-clip Heart Rate Sensor<sup>[21]</sup>, mostrato in Figura 2.2. Si tratta di un trasduttore ottico a processo CMOS ad alte prestazioni e bassa potenza. Presenta un LED verde e DSP integrato per il rilevamento della frequenza cardiaca.

Il sensore è dotato di un guscio in plastica a protezione della piccola e delicata scheda elettronica. Il guscio è trasparente nella parte superiore e opaco in quella inferiore, nella quale un piccolo foro centrale consente il passaggio della luce LED.

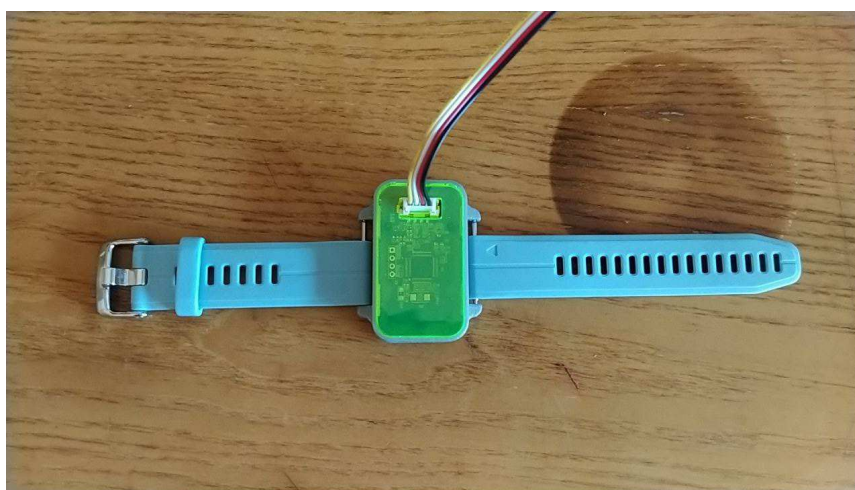


Figura 2.2 Grove – Finger-clip Heart Rate Sensor

Inoltre il produttore fornisce due nastri in velcro di diversa lunghezza per garantire l'indossabilità del sensore – uno, più corto, per mettere il sensore a contatto con un dito e l'altro, più lungo, per consentire il posizionamento all'altezza del polso. Dal momento che l'onda pressoria risulta essere minima alla periferia corporea, le acquisizioni sono state effettuate sul polso per migliorare la qualità dei dati ottenuti.

Il Grove – Finger-clip Heart Rate Sensor ha integrato un microcontrollore STM32 per assicurare l'alta velocità di elaborazione richiesta dall'algoritmo per l'acquisizione della frequenza cardiaca. Il microcontrollore può essere riprogrammato mediante interfaccia SWD.

Il sensore comunica con la scheda Arduino tramite I<sup>2</sup>C. Il bus I<sup>2</sup>C è composto da un master e uno slave che comunicano tra di loro mediante due linee seriali dette SDA (Serial Data) per i dati e SCL (Serial Clock) per il clock, che ne permette la sincronizzazione.

# Capitolo 3

## Metodo di elaborazione dei segnali

### 3.1 Definizione di cross-correlazione

In teoria dei segnali per cross-correlazione, o correlazione incrociata, si intende la misura della similitudine di due segnali come funzione di una traslazione temporale applicata a uno di essi.

Date due sequenze di valori discreti, entrambe di lunghezza  $N$ , la cross-correlazione è definita come

$$R_{xz}[k] = \frac{1}{N} \sum_{i=0}^{N-1} x_i z_{i+k} \quad \text{con } k = 0, \pm 1, \pm 2, \dots \quad (1)$$

dove il secondo segnale,  $z$ , viene gradualmente traslato nel tempo a sinistra.

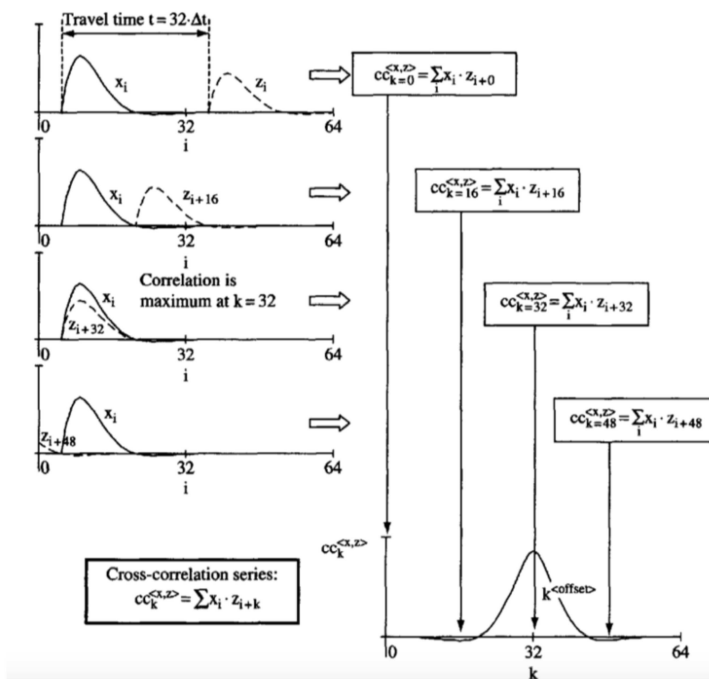


Figura 3.1 Cross-correlazione. Il segnale  $z$  trasla a sinistra all'aumentare di  $k$ . Il picco della somma di cross-correlazione si ha per  $k=32$ , quando i due segnali sono sovrapposti

Ad ogni traslazione nel tempo  $k \cdot \Delta t$ , i valori corrispondenti nei due array vengono moltiplicati tra loro e viene poi effettuata la sommatoria di tutti i prodotti così ottenuti. In Figura 3.1 tale procedimento viene mostrato a livello grafico.

La cross-correlazione – così come l'autocorrelazione, ovvero la correlazione incrociata di un segnale con sé stesso – trova applicazione in svariati ambiti. Può essere utilizzata per determinare lo sfasamento temporale tra due segnali, per ricostruire un segnale rumoroso oppure, come nel caso di seguito analizzato, per individuare somiglianze tra due segnali.<sup>[22]</sup>

## 3.2 Calcolo della cross-correlazione in MATLAB

Al fine di individuare una procedura adatta al calcolo di cross-correlazione, una prima analisi è stata effettuata su segnali GSR e HR precedentemente acquisiti con un differente dispositivo.

Sono stati forniti dei file in formato .csv contenenti delle tabelle con i valori di GSR, temperatura, frequenza cardiaca e IBI (Inter-Beat Interval, ovvero l'intervallo di tempo espresso in millisecondi tra un battito e l'altro), acquisiti tutti durante la medesima sessione, in maniera sincronizzata. Tali file sono stati importati in MATLAB sotto forma di vettori per poter essere elaborati.

Per prima cosa sono stati disegnati i grafici dei segnali in funzione del tempo in un'unica figura (mostrata in Figura 3.2), per poter effettuare una prima analisi qualitativa sui vari andamenti e osservare possibili similitudini.

Sono stati poi presi in esame i segnali di principale interesse in questo studio, ovvero frequenza cardiaca e GSR. Per poter effettuare la cross-correlazione, è stato necessario ricampionare il vettore HR affinché avesse la stessa frequenza del vettore EDA. Poi entrambi i vettori sono stati normalizzati – per rendere i segnali più facilmente comparabili senza perdere informazioni importanti – ed è stata sottratta da essi la media – per rendere i dati stazionari.

Infine è stata applicata la funzione  $[r, \text{lags}] = \text{xcorr}(x, y)$  che, dati in input due segnali, restituisce il vettore di cross-correlazione  $r$ , di lunghezza pari a  $2 \cdot M - 1$  (dove  $M$  è la lunghezza dei vettori forniti in input), e il vettore  $\text{lags}$  degli intervalli di spostamento del segnale che viene traslato nel tempo - come da definizione di cross-correlazione. Andando

a graficare  $r$  in funzione di lags, è possibile determinare, tramite i picchi del grafico, in quali istanti i due segnali sono più o meno simili tra loro.<sup>[23]</sup>

Nelle Figure 3.3 e 3.4 sono mostrati rispettivamente il codice in MATLAB e l'andamento della funzione di cross-correlazione.

```

1  %IMPORTARE E PLOTTARE TEMPERATURA, HR,EDA E IBI
2  %definizione dei vettori tempo
3  - FC_t = 4;
4  - smpl_t = (1:1:length(TEMP));
5  - time_t = smpl_t/FC_t;
6
7  - FC_hr = 1;
8  - smpl_hr = (1:1:length(HR));
9  - time_hr = smpl_hr/FC_hr;
10
11 - FC_eda = 4;
12 - smpl_eda = (1:1:length(EDA));
13 - time_eda = smpl_eda/FC_eda;
14
15 - HR_new = HR(1:1:660);
16 - smpl_hr_new = (1:1:length(HR_new));
17 - time_hr_new = smpl_hr_new/FC_hr;
18
19 %plot di TEMP, HR, EDA e IBI
20 - figure
21 - subplot(4,1,1);
22 - plot(time_t, TEMP);
23 - xlim([0 722]);
24 - xlabel('Tempo');
25 - ylabel('Temperatura (°C)');
26
27 - subplot(4,1,2);
28 - plot(time_hr, HR);
29 - xlim([0 722]);
30 - xlabel('Tempo');
31 - ylabel('Heart Rate');
32
33 - subplot(4,1,3);
34 - plot(time_eda, EDA);
35 - xlim([0 722]);
36 - xlabel('Tempo');
37 - ylabel('EDA (µS)');
38 - |
39 - subplot(4,1,4);
40 - plot(time_hr_new, IBI(:,2));
41 - xlim([0 722]);
42 - xlabel('Tempo');
43 - ylabel('IBI (s)');
44
45
46
47 %ricampionamento HR
48 - HR_ric = resample(HR, FC_eda, FC_hr);
49
50 %normalize
51 - HR_norm = abs(HR_ric)/max(abs(HR_ric));
52 - EDA_norm = abs(EDA)/max(abs(EDA));
53
54 %detrend
55 - HR_det = detrend(HR_norm);
56 - EDA_det = detrend(EDA_norm);
57
58 %correlazione
59 - [r, lags] = xcorr(HR_det, EDA_det);
60 - figure
61 - plot(lags, r);
62 - title('Cross-correlation HR-EDA');
63 - xlabel('Time lag');
64 - ylabel('Correlation');
65 - max(lags);
66 - max(r)
67 - min(r)

```

Figura 3.2 Codice per il plot di temperatura, HR, GSR e IBI

Figura 3.3 Codice per il calcolo della cross-correlazione

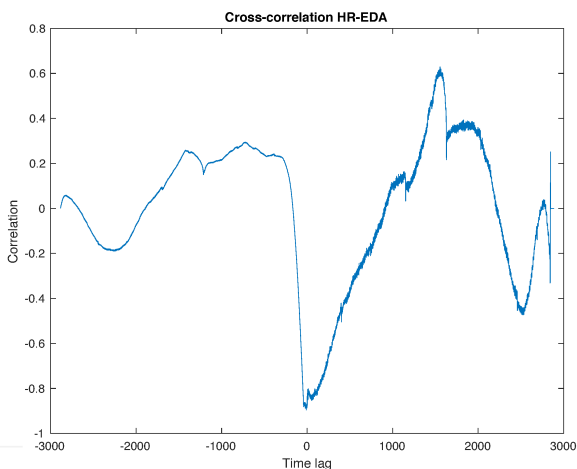


Figura 3.4 Andamento della funzione di cross-correlazione

### 3.3 Indice di correlazione di Pearson

Il coefficiente di correlazione di Pearson è un indice che esprime l'eventuale relazione di linearità tra due variabili statistiche. Esso è definito come rapporto tra la covarianza tra le due variabili per il prodotto tra le rispettive deviazioni standard

$$r_{XY} = \frac{cov(X, Y)}{\sigma_{XY}\sigma_{XY}} \quad (2)$$

La covarianza viene definita come

$$cov(X, Y) = \frac{1}{n} \sum_{i=0}^n (X_i - X_0)(Y_i - Y_0) \quad (3)$$

dove  $X_0$  e  $Y_0$  indicano il valor medio delle due variabili. La covarianza misura come le due variabili si discostano dai rispettivi valori medi. Una covarianza positiva implica che le variazioni di X e Y rispetto a  $X_0$  e  $Y_0$  hanno lo stesso segno, mentre una covarianza negativa implica che le variabili fluttuano intorno ai loro valori medi in modo discorde.

La deviazione standard è definita come la radice quadrata della varianza, ovvero della media aritmetica dei quadrati degli scarti dei valori dalla loro media aritmetica.

L'indice di correlazione di Pearson assume un valore compreso tra -1 e +1. -1 corrisponde alla perfetta correlazione lineare negativa, 0 alla totale assenza di correlazione lineare e +1 alla perfetta correlazione lineare positiva.

Non esiste una regola universale per stabilire la solidità della correlazione a partire dal coefficiente di Pearson; la sua interpretazione varia a seconda dell'ambito nel quale uno specifico studio viene effettuato. Tuttavia, generalmente, si parla di una correlazione significativa per valori dell'indice pari o superiori in valore assoluto a 0.7.<sup>[24]</sup>

### 3.4 Script Arduino

Lo sketch originale permetteva di leggere i dati in uscita dai due sensori descritti nel capitolo precedente e stamparli a video sul monitor seriale dell'IDE di Arduino. Tramite l'applicazione CoolTerm essi venivano salvati in un file .txt, così da permetterne in un secondo momento l'importazione in MATLAB e l'esecuzione del programma descritto nel paragrafo 3.2.

L'obiettivo del seguente sketch è quello di ottimizzare la procedura calcolando in tempo reale a bordo della scheda Arduino UNO il grado di correlazione tramite l'indice di correlazione di Pearson, e restituendolo in uscita.



### 3.4.1 Dichiarazione variabili

Se originariamente i dati acquisiti venivano semplicemente stampati a video, adesso essi vengono salvati in due array, `array_ppg` e `array_gsr`. La dimensione dei due array viene stabilita dalla variabile `nSamples` che indica il numero dei campioni che si desidera analizzare; tale variabile è definita come “const int”, ovvero un tipo di dato a 16 bit che rimane costante durante tutta l’esecuzione.

Sono state inoltre definite le variabili utili nel calcolo della formula dell’indice di correlazione di Pearson: `m_ppg` e `m_gsr` (i valori medi di frequenza cardiaca e GSR), `diff_ppg` e `diff_gsr` (la differenza tra i singoli elementi dell’array e il valor medio) `s_ppg` e `s_gsr` (le deviazioni standard dei due segnali), `s` (la covarianza tra i segnali), `denom` (il denominatore della formula, costituito dal prodotto delle deviazioni standard) e `r` (il coefficiente di correlazione). Trattandosi di numeri reali non necessariamente interi, le variabili sono state definite di tipo “float”.

È poi necessario l’inclusione della libreria “Wire” che consente di comunicare con dispositivi a uscita digitale I<sup>2</sup>C come il fotopleitismografo utilizzato.

La dichiarazione delle variabili è mostrata in Figura 3.5.

```
#include <Wire.h>
const int GSR=A0;
int sensorValue=0;
const long tempo_ppg=500; //tempo di campionamento per i bpm
const long tempo_gsr=100; //tempo di campionamento per la gsr
const int nSamples=150;
int array_ppg[nSamples];
int array_gsr[nSamples];
float m_ppg, m_gsr, s_ppg, s_gsr, s, denom, r;
float diff_ppg, diff_gsr;
```

Figura 3.5 Dichiarazione variabili

### 3.4.2 Funzione void setup

Nella prima funzione, `void setup` (Figura 3.6), si determina la porta seriale della scheda dalla quale transiteranno i dati, si inizializza il processo di lettura/scrittura del I<sup>2</sup>C e si stampa a video un messaggio per l’operatore, nel quale si specifica che i dati relativi a frequenza cardiaca, GSR e indice di correlazione verranno stampati su tre colonne differenti.

```
void setup(){
  Serial.begin(9600);
  Serial.println("Battiti al minuto (bpm) \tValore della risposta galvanica misurata \tr");
  Wire.begin();
}
```

Figura 3.6  
Funzione void

### 3.4.3 Funzione void loop

Con un ciclo for, i dati acquisiti dai sensori vengono immagazzinati nei rispettivi array. La media dei campioni salvati viene calcolata tramite un ulteriore ciclo for.

Utilizzando le variabili precedentemente elencate, si procede separatamente con il calcolo delle deviazioni standard dei due segnali e del loro prodotto – così da ottenere il denominatore della formula desiderata – e della covarianza – ottenendo così il numeratore.<sup>[25]</sup> Infine vengono stampati sia i due array sia l'indice di correlazione (Figure 3.7 e 3.8).

Caricando lo sketch sulla scheda Arduino, verranno stampati un numero pari a nSamples di valori di frequenza cardiaca e GSR e il coefficiente di correlazione calcolato su tali valori. Dopo una pausa di qualche secondo verrà stampato un secondo blocco analogo al primo, e così via per l'intera durata dell'acquisizione.

```
void loop(){
  for (int i=0; i<nSamples; i++){
    Wire.requestFrom(0xA0 >> 1, 1);
    // request 1 bytes from slave device
    while(Wire.available()) { // slave may send less than requested
      unsigned char c = Wire.read(); // receive heart rate value (a byte)
      //Serial.println(c, DEC); // print heart rate value
      array_ppg[i]=c;
    }
    delay(500);
    sensorValue=analogRead(GSR);
    array_gsr[i] = sensorValue;
  }

  //Calcolare la media
  m_ppg=0; m_gsr=0;
  int a;
  for(a=0; a<nSamples; a++){
    m_ppg += array_ppg[a];
    m_gsr += array_gsr[a];
  }
  m_ppg /= nSamples;
  m_gsr /= nSamples;

  //Calcolare il denominatore
  s_ppg=0; s_gsr=0;
  for(int b=0; b<nSamples; b++){
    diff_ppg = array_ppg[b]-m_ppg;
    diff_gsr = array_gsr[b]-m_gsr;

    s_ppg += (diff_ppg)*(diff_ppg);
    s_gsr += (diff_gsr)*(diff_gsr);
  }
  denom = sqrt(s_ppg*s_gsr);
```

Figura 3.7 Prima parte della funzione void loop; riempimento degli array, calcolo della media e del denominatore

```

//correlation series
int maxdel=nSamples;
int del;
for(del=-maxdel; del<maxdel; del++){
    s=0;
    for(int cc=0; cc<nSamples; cc++){
        float d = cc+del;
        while (d<0)
            d += nSamples;
        d /= nSamples;
        s += (array_ppg[cc]-m_ppg)*(array_gsr[cc]-m_gsr);
    }
    r=s/denom;
}

for(int k=0; k<nSamples; k++){
    Serial.print(array_ppg[k]);
    Serial.print('\t');
    Serial.print('\t');
    Serial.print('\t');
    Serial.print('\t');
    Serial.print(array_gsr[k]);
    Serial.print('\t');
    Serial.print('\t');
    Serial.print('\t');
    Serial.print('\t');
    Serial.print(r);
    Serial.print('\n');
}
}

```

Figura 3.8 Seconda parte della funzione loop; calcolo della covarianza, stampa degli array e dell'indice di correlazione di Pearson

# Capitolo 4

## Acquisizioni e analisi

### 4.1 Problematiche nell'acquisizione

#### 4.1.1 Annullamento della deviazione standard

Lo sketch descritto nel precedente capitolo presenta alcune problematiche, in parte legate al tipo di studio nel quale viene utilizzato, in parte dovute a fattori computazionali.

A volte può capitare che i valori acquisiti siano tutti uguali tra di loro. Ciò accade soprattutto quando si svolgono studi a riposo – per cui i parametri non subiscono variazioni sostanziali – oppure quando le variazioni avvengono in tempi più lunghi rispetto al numero di campioni analizzati per volta. Inoltre tale problema si presenta soprattutto per quanto riguarda la frequenza cardiaca, dal momento che si tratta di un segnale piuttosto stabile e subisce variazioni più lente rispetto alla GSR.

Se i valori sono tutti uguali tra loro, allora la loro media sarà pari al valore stesso, e la differenza tra campioni e media risulta essere pari a zero. Ciò significa che sia la covarianza sia la deviazione standard si annullano e dunque non è possibile calcolare l'indice di correlazione di Pearson.

Per ovviare al problema si ricorre all'introduzione di una variabile di tipo float, chiamata *epsilon*, e la si pone a un numero piccolo maggiore di zero; nelle acquisizioni effettuate, *epsilon* è stato posto pari a 0.01, ma il valore può essere modificato in base alle esigenze. All'interno della funzione *void setup*, quando vengono calcolate *diff\_ppg* e *diff\_gsr*, si aggiungono due cicli if: se la differenza si annulla, allora la si pone pari a *epsilon*. Tale modifica allo sketch è mostrata in Figura 4.1.

```

//Calcolare il denominatore
s_ppg=0; s_gsr=0;
for(int b=0; b<nSamples; b++){
    diff_ppg = array_ppg[b]-m_ppg;
    if(diff_ppg==0){
        diff_ppg = epsilon;
    }

    diff_gsr = array_gsr[b]-m_gsr;
    if(diff_gsr==0){
        diff_gsr = epsilon;
    }

    s_ppg += (diff_ppg)*(diff_ppg);
    s_gsr += (diff_gsr)*(diff_gsr);
}
denom = sqrt(s_ppg*s_gsr);

```

Figura 4.1 Codice di correzione in caso di annullamento della deviazione standard

Per verificare l'effettiva necessità di suddetta aggiunta al codice, sono state effettuate delle acquisizioni con lo sketch descritto nel paragrafo 3.4 modificando di volta in volta il valore della variabile nSamples e si è andati a studiare in quali condizioni l'indice di correlazione risulta essere NaN (Not a Number). Inizialmente sono stati considerati 5 campioni, poi 10, 50 e, aumentando il numero a intervalli di 50 campioni, ne sono stati considerati fino a 350.

In un'acquisizione di 5 minuti, analizzando 5 campioni per volta l'indice di correlazione è risultato essere quasi sempre impossibile da calcolare, con alcune brevi eccezioni, per la mancanza di variabilità nella frequenza cardiaca. Portando il numero di campioni a 10, sempre per 5 minuti, l'indice non è un numero in circa la metà delle iterazioni. Per nSamples superiore a 50, r è sempre pari a un valore numerico.

È importante specificare che le acquisizioni, essendo di prova, sono state effettuate sull'autore stesso della presente tesi in condizioni parzialmente a riposo. Sarebbe necessario valutare su un più ampio campione di popolazione gli effetti che l'attività svolta hanno sul calcolo dell'indice di correlazione – si potrebbe aver bisogno di ricorrere alla correzione nel caso di studio a riposo anche per nSamples pari a 50, oppure potrebbe rivelarsi non necessario per acquisizioni sotto sforzo per un numero di campioni inferiore. È tuttavia consigliabile aggiungere il meccanismo di correzione in ogni caso se questo non influisce sulla velocità computazionale in maniera significativa (cosa che verrà analizzata nel paragrafo seguente).

## 4.1.2 Complessità computazionale

Per complessità computazionale si intende la quantità di risorse impiegate da un algoritmo durante la sua esecuzione, in particolare in termini di tempo e di memoria.

Le considerazioni nel presente capitolo sono state effettuate sulla base delle acquisizioni di cui si è parlato nel paragrafo precedente – ovvero variando il valore di campioni analizzati – sia utilizzando lo sketch descritto nel paragrafo 3.4, sia quello modificato descritto nel paragrafo 4.1.1.

In entrambi gli sketch il numero massimo di campioni immagazzinabili è 395; fissando a tale valore nSamples, le variabili globali occupano il 99% della memoria dinamica, lasciando solo l'1% per le variabili locali. Tuttavia, caricando gli sketch sulla scheda, non si ottiene alcun risultato sul monitor seriale, neanche per tempi lunghi. Per garantire il corretto funzionamento del codice, è necessario mantenere il numero di campioni pari o inferiore a 350, riservando così il 97% della memoria dinamica alle variabili globali e il 3% a quelle locali.

Per quanto riguarda la velocità di calcolo, è stato cronometrato il tempo che intercorre tra la fine della stampa del primo blocco degli n campioni e l'inizio della stampa del secondo blocco per misurare il tempo impiegato dalla scheda per calcolare l'indice di correlazione. Da questo punto di vista, tra i due sketch non sussistono differenze sostanziali. Nella seguente tabella sono riportati i valori ottenuti.

nSamples	5	10	50	100	150	200	250	300	350
Tempo	2.09''	5.22''	23''	52''	1' 22''	1' 46''	2' 14''	2' 46''	3' 13''

Per 5 campioni sono sufficienti circa 2 secondi, mentre per 350 campioni si arriva a superare i tre minuti; pertanto non si tratta di acquisizioni in tempo reale, bensì in tempo quasi-reale.

## 4.2 Confronto tra cross-correlazione e indice di correlazione di Pearson

Nella presente tesi sono stati presi in considerazione separatamente due diversi indicatori di correlazione, ovvero il vettore di cross-correlazione e l'indice di correlazione di Pearson. Si vuole adesso mettere a confronto tali indicatori e verificare che le informazioni da essi forniti siano coerenti tra loro. Tuttavia i due indicatori non possono essere comparati in valore assoluto, trattandosi di grandezze di diversa natura – il primo è un vettore mentre il secondo è uno scalare. Pertanto si ricorre allo script MATLAB mostrato nelle Figure 4.2 e 4.3.

I due indicatori non vengono calcolati sull'intera lunghezza dei due vettori HR e GSR, precedentemente importati in un workspace, bensì su finestre contenenti un numero limitato di campioni. I valori così ottenuti vengono salvati in due array (Figura 4.2), i quali vengono poi normalizzati rispetto ai rispettivi valori massimi e plottati in una stessa figura (Figura 4.3).

```
1 %%normalizzo i valori delle due grandezze
2 HR1(:,1)=HR1(:,1)/max(HR1);
3 GSR1(:,1)=GSR1(:,1)/max(GSR1);
4
5 %% indice Pearson: lo calcolo su finestre di segnale di lunghezza 100 campioni %%
6 dim=size(HR1);
7 maxNSamples=dim(1,1); %%dimensione del vettore originale
8
9 dimFinestra=100; %%con questo valore, trovo 11 (numValoriPearson) valori di indice di Pearson e max_xcorr
10 indice=1;
11 numValoriPearson=floor(maxNSamples/dimFinestra);
12 pearson=zeros(numValoriPearson,1);
13 maxCrossCorr=zeros(numValoriPearson,1);
14 iP=1;
15
16 while ((indice < maxNSamples) && ((maxNSamples-indice)>dimFinestra))
17     mHR1 = sum(HR1(indice:(indice+dimFinestra-1),1))/dimFinestra;
18     mGSR1 = sum(GSR1(indice:(indice+dimFinestra-1),1))/dimFinestra;
19     sHR1 = 0;
20     sGSR1 = 0;
21     for i=1:dimFinestra
22         sHR1 = sHR1 + (HR1(i+indice-1,1)-mHR1)^2;
23         sGSR1 = sGSR1 + (GSR1(i+indice-1,1)-mGSR1)^2;
24     end
25     denom=sqrt(sHR1*sGSR1);
26
27     d=0;
28     maxDel=dimFinestra;
29     for del=-maxDel:maxDel
30         s=0;
31         for cc=1:dimFinestra
32             d=cc+del;
33             while(d<0)
34                 d=d+dimFinestra;
35             end
36             d=d/dimFinestra;
37             s=s+(HR1(cc+indice-1,1)-mHR1)*(GSR1(cc+indice-1,1)-mGSR1);
38         end
39         r=s/denom;
40     end
41 end
```

Figura 4.2 Codice per la definizione delle finestre di campioni e il calcolo degli array contenenti cross-correlazione e indici di Pearson

```

52 %% plotto su uno stesso grafico gli "numValoriPearson" valori di maxCrossCorr e pearson anch'essi normalizzati
53 - x = 1:1:numValoriPearson;
54 - figure
55 - hold on
56 - plot(x,maxCrossCorr/max(maxCrossCorr))
57 - plot(x,pearson/max(pearson));
58 - ylim([-1.5 1.5]);
59 - xlabel('Finestre di campioni');
60 - ylabel('Valori assumibili');
61 - hold off
62 - legend('cross-correlazione','indice di Pearson')

```

Figura 4.3 Codice per il plot degli array normalizzati

Sono state analizzate quattro coppie di vettori HR e GSR, ottenute da altrettante acquisizioni congiunte effettuate con il sistema di sensori descritto in precedenza e basato su scheda Arduino UNO, tre di lunghezza pari a 1170 campioni e una pari a 1150. Al fine di poter trattare i segnali congiuntamente, il vettore di campioni HR è stato sovra-campionato per allinearli a quello della GSR; ogni valore di HR è stato replicato 4 volte per ottenere lo stesso numero di campioni in entrambi gli array. Ogni finestra contiene 100 campioni, fissando così la lunghezza degli array degli indicatori a 11. Di seguito sono mostrati i grafici ottenuti.

Nel primo caso (Figura 4.4) non è possibile individuare uno stesso andamento per i due indicatori; in entrambi si ha una prima diminuzione seguita da un aumento e un picco – nella terza finestra per l'indice di Pearson e nella quarta per la cross-correlazione – seguito da un andamento discorde, fatta eccezione per la settima, l'ottava e la nona finestra, nelle quali entrambi subiscono una diminuzione.

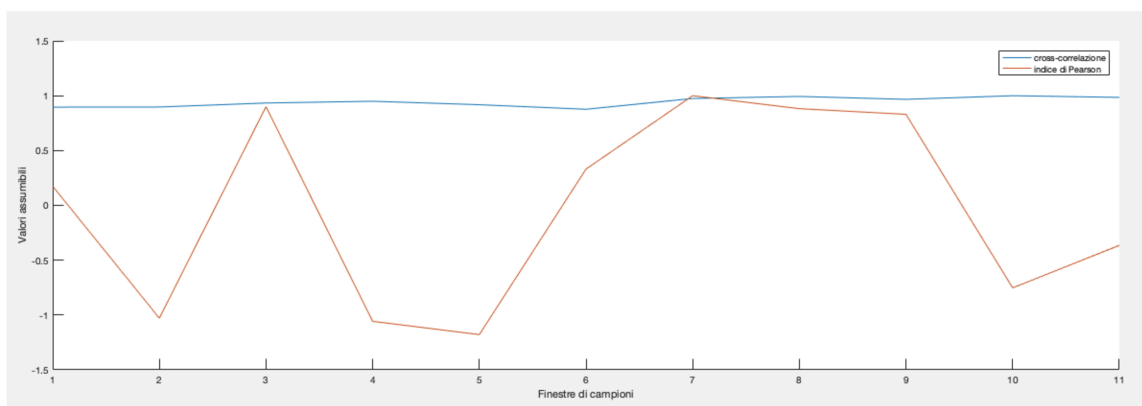


Figura 4.4 Confronto tra cross-correlazione e indice di correlazione di Pearson per la prima coppia di vettori HR e GSR



Nel secondo caso (Figura 4.5) è più visibile una somiglianza nel trend, ad eccezione nelle finestre centrali, nelle quali si ha un andamento discordante.

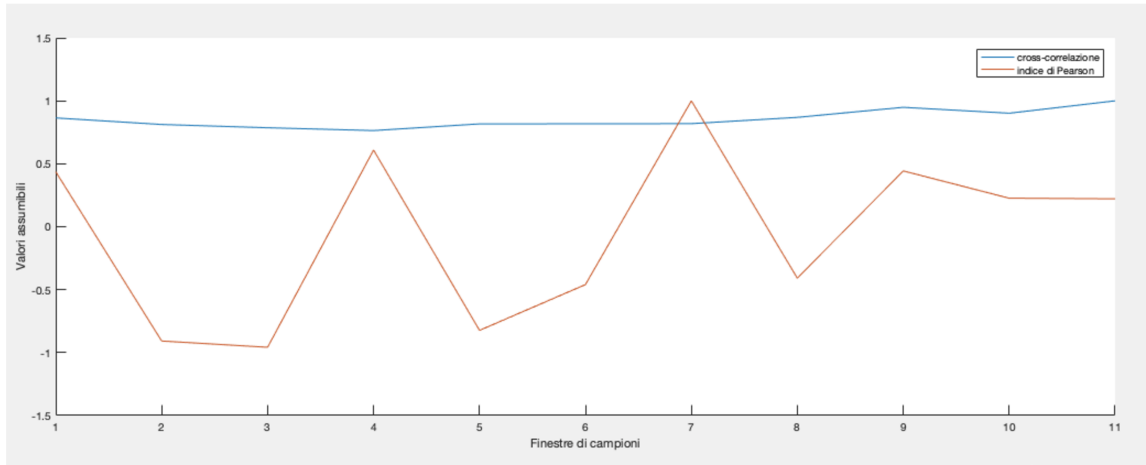


Figura 4.5 Confronto tra cross-correlazione e indice di correlazione di Pearson per la seconda coppia di vettori HR e GSR

Nel terzo caso (Figura 4.6) si individua una sovrapposizione quasi perfetta, fatta eccezione per l'ultima finestra, nella quale la cross-correlazione tende ad aumentare mentre l'indice di Pearson diminuisce.

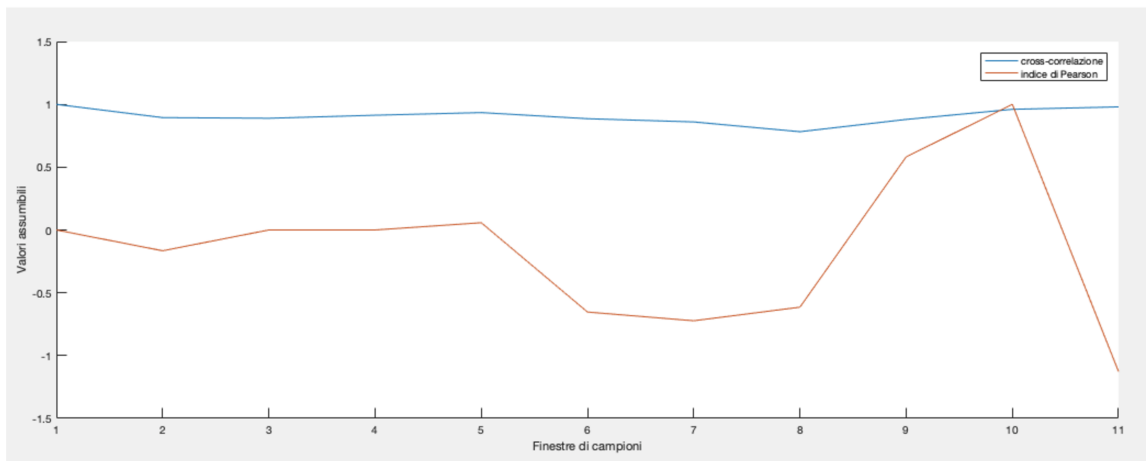


Figura 4.6 Confronto tra cross-correlazione e indice di correlazione di Pearson per la terza coppia di vettori HR e GSR

Infine nel quarto caso (Figura 4.7) i due andamenti si mantengono abbastanza simili sebbene alcuni picchi sia ritardati nella cross-correlazione rispetto all'indice di Pearson, come ad esempio avviene tra l'ottava e la nona finestra.

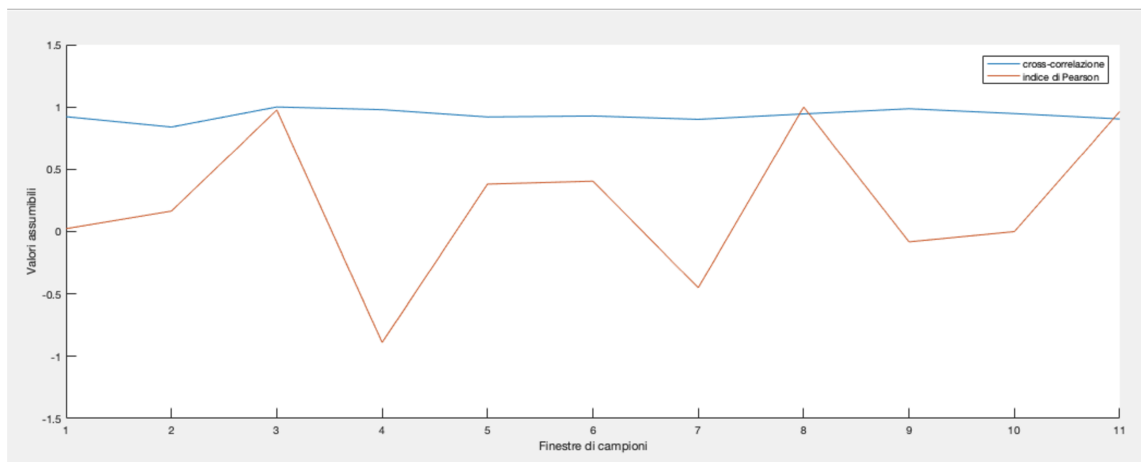


Figura 4.7 Confronto tra cross-correlazione e indice di correlazione di Pearson per la quarta coppia di vettori HR e GSR

# Conclusioni

A seguito dell'analisi svolta nella presente tesi, è possibile affermare che la piattaforma Arduino UNO si conferma essere un ottimo strumento per l'acquisizione e l'elaborazione di biosegnali, malgrado i limiti computazionali della scheda.

Non è possibile estrapolare informazioni utili circa l'effettiva rilevanza dei dati ottenuti, non essendo state effettuate acquisizioni su un campione di popolazione. Tuttavia lo sketch Arduino elaborato nel Capitolo 3 e migliorato nel Capitolo 4 si presenta come un punto di partenza per analisi future con una maggiore valenza da un punto di vista fisiologico e psicologico.

Per quanto riguarda gli indicatori di correlazione, sia la cross-correlazione sia l'indice di correlazione di Pearson sembrano essere validi strumenti. Confrontando i risultati ottenuti da un numero limitato di acquisizioni, gli indicatori, come previsto, non risultano essere in contraddizione tra di loro per quanto riguarda il trend generale.

Lo sketch Arduino e l'indice di correlazione di Pearson si prestano all'ottenimento di risultati in tempo reale o quasi-reale, caratteristica fondamentale per la progettazione di dispositivi indossabili utilizzabili nella vita quotidiana, mentre lo script MATLAB e la cross-correlazione si prestano per lo più ad analisi di laboratorio o allo svolgimento di studi per i quali è possibile elaborare i dati acquisiti in un secondo momento

# Bibliografia

- [1] Lin Shu, Jinyan Xie, Mingyue Yang, Ziyi Li, Zhenqi Li, Dan Liao, Xiangmin Xu, Xinyi Yang, *A Review of Emotion Recognition Using Physiological Signals*, Guangzhou: South China University of Technology, 2018
- [2] R.K. Nath, H. Thapliyal, A. Caban-Holt, S.P. Mohanty, *Machine Learning Based Solutions for Real-Time Stress Monitoring*, IEEE Consumer Electronics Magazine, 2020
- [3] Bohdan Myroniv, Cheng-Wei Wu, Yi Ren, Albert Christian, Ensa Bajo, Yu-Chee Tseng, *Analyzing User Emotions via Physiology Signals*, Hsinchu, Taiwa: National Chiao Tung University, 2017
- [4] Francesco Amico, Lukasz Tyszczyk, Mahnaz Arvaneh, *Using Resting State Heart Rate Variability and Skin Conductance Response to Detect Depression in Adults*, 2020
- [5] Jorge Rodriguez-Arce, Liliana Lara-Flores, Otniel Portillo-Rodriguez, Rigoberto Martinez-Mendez, *Towards an Anxiety and Stress Recognition System for Academic Environments Based on Physiological Features*, Computer Methods and Programs in Biomedicine, 2020
- [6] Chelsea da Estrela, Jennifer McGrath, Linda Booij, Jean-Philippe Gouin, *Heart Rate Variability, Sleep Quality and Depression in the Context of Chronic Stress*, Annals of Behavioral Medicine, 2020
- [7] Michaela C. Pascoe, Sarah E. Hetrick, Alexandra G. Parker, *The Impact of Stress on Students in Secondary School and Higher Education*, International Journal of Adolescence and Youth, 2020
- [8] J.W. Peer, S.B. Hillman, E. Van Hoet, *The Effects of Stress on the Lives of Emerging Adult College Students: An Exploratory Analysis*, Adultspan Journal, 2015
- [9] S. Sriramprakash, Vadana D. Prasanna, O.V. Ramana Murthy, *Stress Detection in Working People*, Procedia Computer Science, 2017
- [10] Jungyoon Kim, Jangwoon Park, Jaehyun Park, *Development of a Statistical Model to Classify Driving Stress Levels Using Galvanic Skin Responses*, Human Factors and Ergonomics in Manufacturing & Service Industries, 2020
- [11] Yekta Can, Niaz Chalabianloo, Deniz Ekiz, Javier Fernandez-Alvares, Giuseppe Riva, Cem Ersoy, *Personal Stress-Level Clustering and Decision-Level Smoothing to Enhance the Performance of Ambulatory Stress Detection with Smartwatches*, IEEE Access, 2020
- [12] Antonio Armario, Javier Labad, Roser Nadal, *Focusing Attention on Biological Markers of Acute Stressor Intensity: Empirical Evidence and Limitations*, Neuroscience & Biobehavioral Reviews, 2020
- [13] Can Bikcora, Denise Klep, Bart Monhemius, Lars van Alem, Viktor Shopov, Denisa Apostol, Reon Smits, *Real-Time Arousal Detection Based on Skin Conductance and Heart Rate Features*, Mentech Innovation, 2020
- [14] A. Affanni, *Wearable Instrument to Measure Simultaneously Cardiac and Electrodermal Activities*, IEEE International Symposium on Medical Measurements and Applications, 2016

- [15] K. Kyriakou, B. Resch, G. Sagl, A. Petutschning, C. Werner, D. Niederseer, M. Liedlgruber, F.H. Wilhelm, T. Osborne, J. Pykett, *Detecting Moments of Stress from Measurements of Wearable Physiological Sensors*, Sensors, 2019
- [16] Y. Hu, C. Converse, M. Lyons, W. Hsu, *Neural Control of Sweat Secretion: a Review*, Br J Dermatol, 2018
- [17] iMotions Biometric Research Platform, Gsr Pocket Guide, <https://imotions.com/blog/galvanic-skin-response/>, 2017
- [18] Grove – GSR Sensor, [http://wiki.seeedstudio.com/Grove-GSR\\_Sensor/](http://wiki.seeedstudio.com/Grove-GSR_Sensor/)
- [19] Toshiyo Tamura, Yuka Maeda, Masaki Sekine, Masaki Yoshida, *Wearable Photoplethysmographic Sensors – Past and Present*, Electronics, 2014
- [20] D. Castaneda, A. Esparza, M. Ghamari, C. Soltanpur, H. Nazeran, *A Review on Wearable Photoplethysmography Sensors and Their Potential Future Applications in Health Care*, Int J Biosens Bioelectron., 2018
- [21] Grove – Finger-clip Hear Rate Sensor with shell, [http://wiki.seeedstudio.com/Grove-Finger-clip\\_Heart\\_Rate\\_Sensor\\_with\\_shell/](http://wiki.seeedstudio.com/Grove-Finger-clip_Heart_Rate_Sensor_with_shell/)
- [22] J. Carlos Santamarina, Dante Fratta, *Discrete Signals and Inverse Problems: an Introduction for Engineers and Scientists*, p. 77-82, John Wiley & Sons, 2005
- [23] H.G. van Lier, M.E. Pieterse, A. Garde, *A Standardized Validity Assessment Protocol for Physiological Signals from Wearable Technology: Methodological Underpinnings and an Application to the E4 Biosensor*, Behav Res, 2020
- [24] Haldun Akoglu, *User's Guide to Correlation Coefficients*, Turkish Journal of Emergency Medicine, 2018
- [25] Paul Bourke, *Cross-correlation, Autocorrelation, 2d Pattern Identification*, 1996