



UNIVERSITÀ POLITECNICA DELLE MARCHE
FACOLTÀ DI INGEGNERIA
Corso di Laurea Magistrale in Ingegneria Elettronica

Studio, implementazione e validazione sperimentale di un volante sensorizzato per il monitoraggio del guidatore

**Study, design and experimental evaluation of an
instrumented steering wheel for driver's monitoring**

Tesi di Laurea di:
DENIS DI LEO

Relatore:
Prof.ssa **SUSANNA SPINSANTE**

Correlatore:
Dott. Ing. **ALESSANDRO TERENCE**

Anno Accademico 2023/2024

Abstract

L'obiettivo del presente lavoro di tesi è lo sviluppo, l'implementazione e la validazione sperimentale di un prototipo di volante da guida smart in grado di monitorare lo stato del guidatore, al fine di abilitare funzionalità evolute per la gestione di situazioni di potenziale rischio o di ridotto comfort. Allo scopo, sono state utilizzate due tipologie di dispositivi ad ampio uso commerciale, quali sensori di forza di tipo resistivo (Force Sensing Resistor - FSR) e un sensore di risposta galvanica della pelle (Galvanic Skin Response - GSR). Entrambe le tipologie di sensori sono state innestate positivamente sul volante utilizzato in un simulatore di guida, con il quale si è arrivati ad ottenere dei dati significativi e conformi agli obiettivi prefissati. Nella elaborazione sviluppo della presente tesi è data particolare enfasi alle criticità emerse durante lo sviluppo del prototipo.

Nel presente lavoro viene inizialmente proposta una descrizione - seppure non esaustiva - dello stato dell'arte che ha dato spunto al progetto, fornendo una panoramica delle grandezze fisiche oggetto di analisi all'interno dell'abitacolo di un'autovettura, e delle tipologie di sensori considerati. Tale analisi ha dato modo anche di operare una selezione delle grandezze fisiche utili agli scopi del progetto, e di individuare i primi vincoli del sistema oggetto di studio. Segue un capitolo relativo alla descrizione dei sensori utilizzati, con un'analisi relativa alla struttura di questi ultimi e ad eventuali limiti. In particolare, per i sensori di forza è stata effettuata un'operazione di caratterizzazione.

Viene poi presentata un'analisi condotta per la scelta della piattaforma da adoperare, in relazione ai vincoli dati dalla fattibilità di realizzazione di un prototipo, quali consumi di corrente e ingombro dell'apparato da installare. A seguito delle numerose opzioni valutate, si è deciso per l'uso di una scheda ESP32 S3, in grado di gestire agevolmente le grandezze in gioco con minore occupazione di volume, per la presenza di convertitori analogico/digitali integrati e capace di una maggiore flessibilità software.

Una volta implementato il prototipo e condotte le prove sperimentali, viene riportata una descrizione delle grandezze fisiche in gioco e della loro correlazione con lo stato psicofisico del guidatore, in relazione all'acquisizione dei dati di forza e di risposta galvanica. In particolare, i segnali GSR sono stati confrontati con quelli acquisiti mediante un dispositivo indossabile Empatica E4, assunto come riferimento, ai fini della verifica della bontà del segnale rilevato dai sensori integrati sul volante.

In conclusione, vi è la presentazione dei dati raccolti ed elaborati allo scopo di ottenere informazioni attendibili in merito alla determinazione dello stato psicofisico del guidatore. Particolare enfasi è data al segnale GSR raccolto o, meglio, ad un parametro da esso ricavabile - la frequenza di ripetizione dei picchi di risposta galvanica - con la cui variazione è possibile rilevare cambiamenti dello stato del guidatore. In aggiunta è stato sviluppato anche un indice riferito al tasso di attivazione dei sensori FSR, in grado di descrivere l'effettivo uso di questi ultimi e quindi fornire un'informazione sul posizionamento delle mani del guidatore durante la prova, allo scopo di verificarne la correttezza o valutarne la variabilità, che potrebbe denotare una guida non attenta.

Abstract

The objective of this thesis is the development, implementation, and experimental validation of a prototype smart driving wheel capable of monitoring the driver's state, in order to enable advanced functionalities for the management of situations of potential risk or reduced comfort. For this purpose, two types of commercially widespread devices were used, namely Force Sensing Resistors (FSR) and a Galvanic Skin Response (GSR) sensor. Both sensors were positively engaged on the steering wheel used in a driving simulator. In the development of this thesis, particular emphasis is placed on the critical points that emerged during the development of the prototype.

In this paper, we initially propose a description - albeit not exhaustive - of the state of the art that gave rise to the project, providing an overview of the physical quantities being analysed inside the passenger cabin of a car, and the types of sensors considered. This analysis also made it possible to perform a selection of the physical quantities useful for the purposes of the project, and to identify the initial constraints of the system under study. This is followed by a chapter on the description of the sensors used, with an analysis of their structure and possible limitations. In particular, a characterisation operation was carried out for the force sensors.

An analysis is then presented for the choice of platform to be used, in relation to the constraints given by the feasibility of building a prototype, such as low power consumption and limited size of the equipment to be installed. Following the numerous options evaluated, it was decided to use an ESP32 S3 board, capable of easily managing the quantities of interest with less volume occupation, due to the presence of integrated analogue/digital converters and capable of greater software flexibility.

Once the prototype has been implemented and the experimental tests conducted, a description of the physical quantities involved and their correlation with the driver's psychophysical state, in relation to the acquisition of force and galvanic response data, is given. In particular, the GSR signals were compared with those acquired by means of an Empatica E4 wearable device, taken as a reference, in order to verify the goodness of the signal detected by the sensors integrated on the steering wheel.

In conclusion, data collected and processed in order to obtain reliable information regarding the determination of the driver's psychophysical state is presented. Particular emphasis is placed on the collected GSR signal or, rather, on a parameter that can be derived from it - the repetition frequency of galvanic response peaks - the variation of which denotes changes in the driver's state. In addition, an index referring to the FSR sensor activation rate was also developed, which can describe the actual use of FSR sensors and thus provide information on the positioning of the driver's hands during the test, in order to verify its correctness or assess its variability, which could denote a careless driving behavior.

Indice

Elenco delle figure	iii
Elenco delle tabelle	vi
1 Stato dell'Arte	1
1.1 Introduzione	2
1.2 Tecniche proposte	5
1.3 Tipologie di sensori adoperati	5
1.4 Conclusioni	7
2 Sensore FSR	9
2.1 Sensore FSR 402	9
2.2 Caratterizzazione di FSR 402	12
2.2.1 Moku:Go	12
2.2.2 VOLTCRAFT VC-655 BT	13
2.3 Setup di misura	14
2.4 Curve di caratterizzazione	16
3 Schede di Sviluppo testate	27
3.1 Introduzione	27
3.2 Raspberry Pi 4	28
3.2.1 Analogue Digital Converter: ADS 1115	29
3.2.2 Lettura sensori	30
3.3 Raspberry Pi Pico	34
3.4 ESP32 S3	38
3.5 Consumi di corrente	41
4 Acquisizione dati FSR	43
4.1 Prototipo sviluppato	43
4.2 Tool sviluppato	46
4.2.1 Acquisizione Dati	46
4.2.2 Elaborazione Dati	47
4.3 Dati raccolti	49

5	Sensore GSR	54
5.1	Galvanic Skin Response	54
5.2	Sensore GSR	58
5.2.1	Piattaforma Hardware	60
5.2.2	Accorgimenti per le misure	60
6	Acquisizione dati GSR	62
6.1	Tool sviluppato	62
6.1.1	Calibrazione	63
6.1.2	Acquisizione GSR grezza	63
6.1.3	Acquisizione GSR mediata	65
6.2	Dati raccolti	67
6.2.1	Filtraggio	69
6.2.2	Acquisizione GSR mediata e filtraggio	70
6.3	Empatica E4	72
6.3.1	Confronto misure raccolte	73
7	Elaborazione Dati Raccolti	76
7.1	Ricostruzione segnale GSR	76
7.1.1	Frequenza dei picchi di risposta galvanica	81
7.1.2	Dati soggetto 2	82
7.1.3	Dati soggetto 3	83
7.1.4	Dati soggetto 4	85
7.2	Confronto FSR – Accelerometro	87
7.3	Frequenza d'uso dei sensori FSR	90
8	Conclusioni	93
8.1	Prototipo sviluppato	93
8.2	Criticità	95
8.3	Sviluppi futuri	96
	Bibliografia	98
	Sitografia	100

Elenco delle figure

2.1	Tipologia di sensore FSR usato nello studio.	10
2.2	Curva caratteristica resistenza/forza di un sensore FSR.	10
2.3	Esplosivo FSR 402.	11
2.4	Strumento Moku:Go.	13
2.5	VOLTCRAFT VC-655 BT.	14
2.6	Particolare FSR 402.	15
2.7	Particolare FSR 402.	15
2.8	Visione d'insieme Moku:Go + FSR 402.	16
2.9	Curva caratteristica FSR.	17
2.10	Caratterizzazione FSR.	18
2.11	Distribuzione misure 3 k Ω - 25 g.	19
2.12	Distribuzione misure 3 k Ω - 1000 g.	19
2.13	Partitore $R_M - R_{FSR}$	19
2.14	Distribuzione misure 100 k Ω - 25 g.	20
2.15	Distribuzione misure 100 k Ω - 1000 g.	21
2.16	Caratterizzazione FSR1.	21
2.17	Caratterizzazione FSR2.	22
2.18	Caratterizzazione FSR3.	22
2.19	Caratterizzazione FSR4.	23
2.20	Caratterizzazione FSR5.	23
2.21	Caratterizzazione FSR6.	24
2.22	Caratterizzazione FSR7.	24
2.23	Caratterizzazione FSR8.	25
2.24	Caratterizzazione FSR9.	25
2.25	Caratterizzazione FSR10.	26
3.1	Raspberry Pi 4.	28
3.2	ADC ADS 1115.	29
3.3	Schema a blocchi ADC ADS 1115.	30
3.4	Istruzione verifica P^2C	31
3.5	FSR non attivati.	32
3.6	FSR3 – FSR7 – FSR8 – FSR9 attivati.	32
3.7	Lettura sensori FSR Raspberry Pi 4.	33
3.8	Panoramica sistema.	34

3.9	Particolare consumo corrente.	34
3.10	Raspberry Pi Pico.	35
3.11	Registro di configurazione.	36
3.12	Bit 14:12 Registro di Configurazione.	37
3.13	Modifica della libreria.	37
3.14	Consumi corrente Raspberry Pi Pico.	38
3.15	ESP32 S3.	38
3.16	Caratteristica ADC ESP32.	39
3.17	Consumi corrente ESP32 S3.	40
3.18	Sistema ESP32 S3 + FSR.	40
4.1	Prototipo volante.	44
4.2	Retro prototipo volante.	44
4.3	Particolare ESP32 S3 e partitore resistivo.	45
4.4	Schema elettrico partitore resistivo.	45
4.5	Scrittura valori all'interno di file.	47
4.6	Flowchart lettura FSR.	47
4.7	Apertura file dati.	48
4.8	Flowchart utilizzo libreria pyCirclize.	48
4.9	Libreria pyCirclize.	49
4.10	Volante non utilizzato.	50
4.11	Valori tensione media riferiti a Fig. 4.10.	50
4.12	Volante con guida su rettilineo.	51
4.13	Valori tensione media riferiti a Fig. 4.12.	51
4.14	Volante con guida su curve.	52
4.15	Valori tensione media riferiti a Fig. 4.14.	52
4.16	Volante con guida su percorso misto.	53
4.17	Valori tensione media riferiti a Fig. 4.16.	53
5.1	Sistema Nervoso Autonomo.	55
5.2	Tipologie di ghiandole sudoripare.	56
5.3	Attività SCL e SCR di un segnale GSR.	58
5.4	Schema elettrico del circuito di condizionamento.	59
5.5	Sensore GSR Seed Studio.	60
6.1	Calcolo resistenza e conduttanza della pelle.	63
6.2	Acquisizione GSR.	64
6.3	Lettura dati GSR.	65
6.4	Acquisizione GSR "no glitch".	66
6.5	Lettura dati GSR con eliminazione glitch.	66
6.6	Lettura dati GSR con eliminazione glitch.	66
6.7	GSR filtrato (rosso) e non filtrato (blu).	68
6.8	Filtro passa-basso implementato in MatLab.	70
6.9	Particolare: segnale filtrato (rosso) vs non filtrato (blu).	70
6.10	GSR filtrato (rosso) e non filtrato (blu) con controllo spike.	71

6.11	Segnale GSR con presa da due mani.	71
6.12	Specifiche del dispositivo Empatica E4.	73
6.13	GSR Empatica vs GSR Grove.	74
6.14	GSR Empatica.	75
6.15	GSR Prototipo.	75
7.1	Ricostruzione segnale GSR.	77
7.2	Segnale GSR acquisito.	77
7.3	Segnale GSR elaborabile.	78
7.4	Segnale GSR filtrato.	79
7.5	Estrazione componente fasica e tonica.	79
7.6	Picchi GSR.	80
7.7	Segnale GSR Phasic.	80
7.8	Segnale GSR acquisito.	82
7.9	Segnale GSR elaborabile.	82
7.10	Segnale GSR filtrato.	82
7.11	Picchi GSR.	83
7.12	Segnale GSR Phasic.	83
7.13	Segnale GSR acquisito.	83
7.14	Segnale GSR elaborabile.	84
7.15	Segnale GSR filtrato.	84
7.16	Segnale GSR Empatica E4.	84
7.17	Picchi GSR.	85
7.18	Segnale GSR Phasic.	85
7.19	Segnale GSR acquisito.	85
7.20	Segnale GSR elaborabile.	86
7.21	Segnale GSR filtrato.	86
7.22	Segnale GSR Empatica E4.	86
7.23	Picchi GSR.	87
7.24	Segnale GSR Phasic.	87
7.25	Confronto segnale del giroscopio vs FSR soggetto n. 1.	89
7.26	Confronto segnale del giroscopio vs FSR soggetto n. 2.	89
7.27	Confronto segnale del giroscopio vs FSR soggetto n. 4.	90
8.1	Setup sperimentale.	95
8.2	Particolare prototipo sviluppato.	95

Elenco delle tabelle

1.1	Acronimi utilizzati.	2
1.2	Lavori analizzati.	4
1.3	Sensoristica utilizzata.	6
3.1	Connessione pin ADDR.	31
3.2	Consumi di corrente.	41
7.1	Frequenza picchi risposta galvanica.	81
7.2	Tasso di attivazione sensori FSR (%).	92

Capitolo 1

Stato dell'Arte

Nel seguente capitolo viene descritto lo stato dell'arte in merito ai sistemi di rilevazione dello stato psicofisico del guidatore di un autoveicolo, con particolare riguardo alla grandezze fisiche oggetto di analisi e della tipologia di sensori adottati. Lo scopo prefissato nel nostro caso è quello della determinazione di taluni stati di stress che potrebbero portare il guidatore in condizione di pericolo.

Durante tutto lo svolgimento del lavoro si darà luogo ad un uso importante di acronimi: ai fini di alleggerire e rendere maggiormente comprensibile il tutto, si riporta in Tab. 1 l'elenco dei principali acronimi che saranno usati e del relativo significato.

Tabella 1.1: Acronimi utilizzati.

Acronimo	Significato	Descrizione
EDA	ElectroDermal Activity	variazione della conduttanza della pelle
GSR	Galvanic Skin Response	risposta galvanica della pelle
EMG	ElectroMyoGraph	attività elettrica a livello muscolare
FSU	Force Sensing Unit	unità di sensing della variazione di una forza
ECG	Electro CardioGram	rappresentazione attività del cuore
PPG	PhotoPlethysmoGraphy	variazione del volume di sangue
IMU	Inertial Measurement Unit	misura accelerazione e velocità angolare di un corpo
MTCNN	Multi Task Convolutional Neural network	Tipologia di rete neurale convoluzionale
ML	Machine Learning	Apprendimento automatico

1.1 Introduzione

Tra le soluzioni più studiate, allo scopo di verificare se un guidatore sia nelle condizioni di poter condurre un veicolo, vi è l'analisi di biosegnali quali ad esempio, ECG, PPG, GSR, così come l'analisi del volto, al fine di identificare gesti sinonimi di stanchezza, quali l'aumento della frequenza di chiusura delle palpebre, degli sbadigli o dell'inclinazione della testa. Le informazioni riportate nel seguente capitolo non hanno assolutamente la presunzione di illustrare la totalità dello stato attuale della ricerca in questo ambito, bensì lo scopo di dare una panoramica, per quanto di ampio respiro, che sia in grado di incardinare e contestualizzare il prototipo che sarà

successivamente presentato in questa tesi.

Viene di riportata una breve sintesi dei lavori oggetto di analisi, con una breve descrizione dei punti di forza e di debolezza di questi ultimi in Tab. 1.1.

Tabella 1.2: Lavori analizzati.

Riferimento	Sensori/Metodi di rilevazione	Note
[1]	ECG* + PPG + oximetry + IMU	sistema valido ma necessita di ulteriori verifiche, economico
[2]	ECG*	sistema poco affidabile
[3]	MTCNN + GSR	sistema valido ed economico
[4]	EDA* + EMG* + FSU*	sistema valido e versatile sensori prodotti in loco
[5]	ECG** + EDA** + ML**	sistema valido ma necessita di ulteriori verifiche
[6]	GPS + temperature + CNN	sistema valido ma necessita di ulteriori verifiche, i sensori non cooperano fra loro
[7]	ECG + PPG + pressure	sistema valido, vetusto
[8]	PPG** + temperature**	sistema valido ma limitato, economico
[9]	CNN	sistema valido, economico, necessita di ulteriori sensori
[10]	R + C + L	sistema valido, economico ma limitato, necessita di ulteriori sensori
[11]	PPG	sistema valido, meglio se associato ad altri sensori
[12]	PPG	sistema valido ma necessita di ulteriori verifiche
[13]	GSR + ML	sistema valido
[14]	FSU	sistema valido ma necessita di ulteriori verifiche
[15]	panoramica dei sistemi utilizzati	
* prodotto in loco dagli autori		
** gli autori non hanno dato altre informazioni		

1.2 Tecniche proposte

Tutti i metodi proposti si propongono innanzitutto di non essere invasivi. Le tecniche presenti in letteratura suggeriscono l'uso di diverse tipologie di sensori e quantità misurate, come PPG, GSR, videocamere, radar, ECG, temperatura del corpo con e senza uso di videocamere a infrarosso, e spesso ricorrono all'utilizzo di differenti tipologie di reti neurali, CNN in primis. Si ottengono i risultati migliori quando si combinano contemporaneamente più tecniche, in particolare quelle sensoristiche in senso classico con quelle di *Machine Learning* [3] [6] [10] [14] e [16].

Le misure ricavate da ECG sono problematiche in quanto necessitano di sistemi di misura costosi o invasivi [1] e [5], che risultano scomodi da usare durante la guida. In alcuni lavori, per ovviare a ciò, si è proceduto con delle realizzazioni “home made” di difficile replicazione [2] e [4], in cui sono proposte nuove tipologie di sensori, alcuni anche realizzati con materiali innovativi. Le misure di conduttanza della pelle sono state eseguite sia tramite sistemi auto-prodotti [4] che mediante sensori commerciali [3] e [14], permettendo di ottenere dei buoni risultati. Una nota a favore di questi ultimi sensori è data dall'invasività minore di tali dispositivi rispetto al rilevamento di altri parametri fisiologici, come [3] [16] e [8].

In alcuni lavori i sistemi progettati risultano eterogenei, ovvero non in grado di amalgamare le varie informazioni acquisite fra di loro [7] [9]. Alcuni sistemi invece sono costituiti da solo un sensore, presentando quindi delle limitazioni [10] [11] [12] [13] e [15], aspetto confermato anche in [16] che fornisce una panoramica sull'argomento.

1.3 Tipologie di sensori adoperati

La Tab. 1.3 riassume le informazioni sulle tipologie di sensori adoperati in letteratura, ove disponibili. Purtroppo, nei vari casi analizzati, vi sono diversi sensori che non sono di uso commerciale, ovvero, sono stati realizzati in modo autonomo per le finalità della ricerca, e ciò rende complicata una eventuale replicazione. Viene anche riportato parallelamente il dettaglio relativo allo strumento di misura (laddove usato) che,

dotato di caratteristiche metrologiche superiori, è stato preso come riferimento per la verifica della correttezza dei dati raccolti dai sensori.

Tabella 1.3: Sensoristica utilizzata.

Riferimento	Sensori usati	Riferimento Misure
[1]	ECG* different materials + (PPG + oximetry) MAX30102 + IMU MPU-6050	BIOPAC MP36
[2]	ECG* different materials	ECG AgCl Explorer Kit
[3]	GSR V2.0 – Pi Camera v2	X
[4]	*	X
[5]	Biosignalplux **	X
[6]	LED + SiMPs + electrodes**	BITalino EDA
[7]	LM393 + DHT11 + MQ3 + NEO-6M + pulse**	X
[8]	*	X
[9]	**	X
[10]	X	X
[11]	RLC circuit	X
[12]	LED + SiMPs	X
[13]	LED + SiMPs	iMDK C101A3
[14]	Empatica E4	ProComp Infiniti
[15]	FSR-408	Spirit-10-Mark II
[16]	panoramica dei sistemi utilizzati	
* prodotto in loco dagli autori		
** gli autori non hanno dato altre informazioni		

1.4 Conclusioni

In conclusione [16], possiamo affermare che la misura di taluni parametri fisiologici quali ECG, EEG e frequenza respiratoria è presente nella larga parte dei lavori fino ad oggi realizzati. Questi parametri presentano la caratteristica di essere precisi e di associare in maniera univoca uno stato del guidatore ad una determinata variazione del segnale preso in osservazione, ma anche quella di essere piuttosto invasivi da misurare, in quanto le zone dalle quali il segnale viene prelevato sono in posizioni difficilmente accessibili durante la guida, quali testa, braccia, e mani, rendendone quindi difficoltosa l'acquisizione.

I metodi non invasivi, quali ad esempio quelli che fanno uso di telecamere ad infrarosso o termiche, permettono comunque di rilevare pochi parametri fisiologici, la temperatura corporea ad esempio, che sono a loro volta fortemente suscettibili all'ambiente esterno, in maniera più marcata rispetto ai metodi di natura invasiva. Sono inoltre presenti numerosi lavori in cui ai sensori usati in modo classico sono affiancati modelli di *machine* e *deep learning*, che coadiuvano nel risultato finale [14] e [5]. Un problema di questi approcci è dato dall'onere computazionale che risulta essere importante e dalla mancanza in letteratura di dataset specifici per il caso applicativo del monitoraggio del guidatore.

L'analisi degli studi che hanno utilizzato EEG, ECG, GSR e la telecamera a infrarossi mostra che i risultati dipendono, pur in misura diversa, dalla tipologia di sensore fisico adoperato. Ad esempio, vi sono alcuni sensori che sono estremamente sensibili alle vibrazioni della guida [2] (strada, cambio di marcia), rendendo necessaria una pesante rielaborazione del segnale. È riconosciuto ad esempio che la frequenza respiratoria è un buon metodo per la rilevazione dello stato di veglia del conducente [16]. Un ulteriore valido metodo per tale scopo è quello della rilevazione della forza applicata dal guidatore al volante [15]. Tale tecnica risulta efficace ma è doveroso segnalare come il sensore sia suscettibile a spostamenti delle mani.

Per quanto concerne i segnali EEG che ECG, nonostante siano molto vantagio-

si rispetto alla GSR, FSU e alle termocamere nel fornire dati validati come utili, presentano diverse limitazioni. I sensori EEG ed ECG sono usati in approcci di misurazione a contatto che richiedono elettrodi sul soggetto. Una termocamera (usata ad esempio per la rilevazione della temperatura del viso), invece, offre un approccio non intrusivo e monitora il soggetto a distanza, subendo comunque notevoli influenze dal contesto ambientale. I segnali EEG ed ECG mostrano una certa resistenza alle condizioni ambientali e sono notoriamente più accurati [16]. Tale caratteristica è parzialmente verificabile anche attraverso misure GSR [14] e di forza [15]. Una problematica importante delle misure effettuate via elettrodo è il continuo bisogno di avere una connessione pelle - elettrodo, poichè in mancanza di ciò si ottiene un pesante deterioramento del segnale.

Inoltre, le misure eseguite durante simulazioni vengono effettuate in ambienti che, per quanto standardizzati, presentano comunque delle notevoli differenze rispetto agli stimoli che potrebbero provenire dall' ambiente reale, in cui il soggetto risponderebbe in modo emotivamente diverso, ad esempio perchè vengono ignorati diversi scenari di guida quali pioggia, nebbia o ghiaccio e ciò può influire sull'efficacia degli approcci proposti.

In conclusione, è doveroso segnalare che nella totalità dei lavori proposti nel seguente capitolo, l'operatività dei dispositivi è stata verificata mediante simulatori di guida o in luoghi comunque controllati, per ovvi motivi di sicurezza, ovvero in condizioni che, per quanto simili a quelle reali e realistiche, non permettono di ottenere i medesimi risultati ottenibili su strada a lungo termine. Essi rappresentano comunque un'ottima validazione dei metodi proposti, per ottenere risultati concreti e confrontabili.

Capitolo 2

Sensore FSR

Nel seguente capitolo viene descritto il primo dei due sensori adoperati per lo sviluppo del prototipo, in particolare quello di forza. Nel testo che segue è riportata una descrizione delle caratteristiche fisiche ed elettroniche del sensore, inoltre verrà presentata l'attività di caratterizzazione del sensore che ne ha preceduto l'utilizzo.

2.1 Sensore FSR 402

Il sensore FSR 402 [web15] fa parte della famiglia dei sensori passivi di forza a film di polimero spesso, ovvero *Polimer Thick Film*, che presentano un valore di resistenza inversamente proporzionale alla forza applicata. In commercio sono disponibili diversi modelli sempre della stessa famiglia, che variano sostanzialmente unicamente per l'area geometrica sensibile entro la quale è possibile rilevare una variazione di forza. Per gli scopi del presente lavoro si è deciso di adoperare un sensore comunemente detto "a bottone" come riportato nella Fig. 2.1.



Figura 2.1: Tipologia di sensore FSR usato nello studio.

Quando non soggetto a nessuna forza, il sensore presenta una resistenza dell'ordine del centinaio di $M\Omega$, la quale si riduce drasticamente a valori dell'ordine di qualche $k\Omega$ quando sottoposto ad una forza significativa e tende a una resistenza prossima allo zero quando la forza applicata è vicina al limite del sensore. La curva di risposta è mostrata in Fig. 2.2. Il range di misura è compreso fra 10 g e 10 kg.

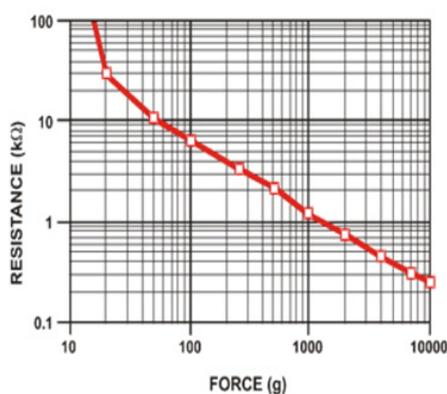


Figura 2.2: Curva caratteristica resistenza/forza di un sensore FSR.

Il sensore è costituito da diversi strati sottili e flessibili, come mostrato in Fig. 2.3. Quando viene premuto, una parte maggiore degli elementi di carbonio, che è un buon isolante, viene messa a contatto con le tracce conduttive riducendo così la resistenza complessiva. In uno stato di assenza di forza, viene misurata la resistenza offerta unicamente dallo strato di carbonio; viceversa, in caso di forza applicata, viene ad aumentare la superficie di contatto fra la faccia isolante al carbonio e quella conduttiva,

per effetto della compressione e quindi dell'avvicinamento delle due facce, portando ad una netta diminuzione della resistenza complessiva. Dalle modalità operative si deduce come, stimolando il sensore, anche a parità di forza si ottengano risultati che non sono mai i medesimi, in quanto la superficie sensibile può essere oggetto di forza in punti che differiscono fra loro, cambiando quindi il numero di piste conduttive a contatto con lo strato di carbonio. Si tenga in considerazione che in generale questa tipologia di sensore ha tempi di risposta non immediati, e ciò deriva dalla elasticità del guscio esterno che impiega un tempo non istantaneo a ritornare nello stato di riposo e rendere possibile una nuova misura.

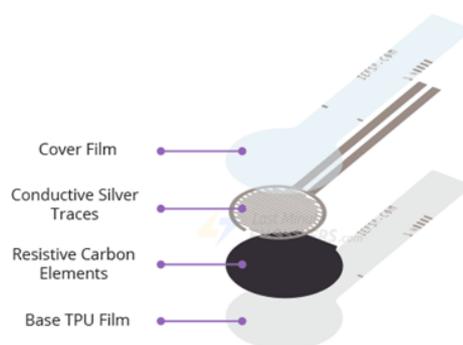


Figura 2.3: Esploso FSR 402.

Come si è brevemente accennato in precedenza, nella Fig. 2.2, in cui è rappresentato un andamento resistenza/forza, vi è una chiara indicazione di un comportamento non lineare del sensore. Infatti, la caratteristica tende a seguire un andamento esponenziale per pesi con masse inferiori ai 30 g, mentre per pesi maggiori essa tende a saturazione portando ad un crollo della sensibilità per valori di peso superiori a 500 g. Tali valori di peso sono puramente indicativi.

Nel presente lavoro si fa riferimento alla massa come indicazione della variazione che il sensore subisce. In realtà, quest'ultimo è sensibile ad una forza, ovvero:

$$F = m \cdot a \quad (2.1)$$

in cui $a = 9.81 \text{ m/s}^2$ (ovvero g), m è la massa (in kg), e F è la forza espressa in $\text{kg} \cdot \text{m/s}^2$ ovvero N.

L'accelerazione a , che si omette lungo tutto il documento, sulla falsariga di quanto fatto anche nel datasheet dagli stessi costruttori, è implicitamente presente in tutto il sistema in quanto tutte le misure sono effettuate sulla Terra, rendendo pertanto le misurazioni di un peso o, meglio, di una massa, in realtà delle misurazioni di una forza. Anche la parola "peso" è stata adoperata come di uso comune, ovvero, al posto della parola "massa", infatti correttamente dovremmo definire "una massa di 20 g" e non "un peso di 20 g".

2.2 Caratterizzazione di FSR 402

Il prototipo sviluppato è imperniato sull'uso di 10 sensori FSR opportunamente disposti e connessi lungo la corona esterna di un volante. Trattando 10 sensori, malgrado siano della stessa tipologia e dello stesso costruttore, si può affermare che certamente non sono uguali, rendendo doverosa quindi una operazione di caratterizzazione di tutti i sensori oggetto di sviluppo. Di seguito è descritta l'operazione svolta, con i dispositivi hardware utilizzati e i risultati ottenuti variando anche la sensibilità del sensore in funzione di una resistenza esterna.

2.2.1 Moku:Go

Moku:Go, mostrato in Fig. 2.4, è una piattaforma hardware portatile dotata di una FPGA Zynq 7020, che offre fino a 11 strumenti da laboratorio differenti e alimentatori programmabili opzionali con capacità di elaborazione in tempo reale. È possibile anche, attraverso la modalità "multistrumento", utilizzare due dispositivi di misura contemporaneamente per creare un banco di prova personalizzato.

Tra i vari strumenti a bordo possiamo trovare un generatore di forme d'onda, un oscilloscopio, un analizzatore di spettro ed un generatore di forme d'onda arbitrario. Per modelli di *Moku* più avanzati sono previsti anche strumenti ottici e potenzialità di elaborazione maggiori.



Figura 2.4: Strumento Moku:Go.

E' possibile programmare *Moku:Go* attraverso il proprio applicativo dotato di un'interfaccia utente intuitiva (UI) disponibile sia per Windows che macOS, scaricabile presso [web1], o anche attraverso software esterni quali *MatLab*, *LabView* o attraverso codice *Python*. La scelta nel nostro caso è ricaduta nel programmare il *Moku:Go* attraverso il proprio software proprietario, per poi passare i dati raccolti a *MatLab*, con cui elaborarli.

Sono riportati brevemente di seguito i parametri salienti della piattaforma, la quale è dotata di 2 ingressi digitali/analogici a 12 bit, 125 MSa/s con una bandwidth di 30 MHz e 16 canali digitali a 125 MSa/s con una output bandwidth di 20 MHz. Inoltre, sono presenti alcuni pin di alimentazione programmabili dei quali si è fatto ampio uso durante le sessioni di misura per alimentare i sensori.

2.2.2 VOLTcraft VC-655 BT

Ai fini della misura della resistenza a vuoto del sensore, vista la mancanza di un misuratore di resistenza tra gli strumenti supportati dal *Moku:Go*, si è deciso di utilizzare il *VOLTcraft VC-655 BT* mostrato in Fig. 2.5.



Figura 2.5: VOLTcraft VC-655 BT.

La resistenza del FSR a vuoto, ovvero senza nessuna forza applicata, come anticipato in precedenza, presenta valori estremamente elevati. Si è tentata inizialmente una misura mediante tester, modello *Protek 506*, raggiungendo il fondo scala pari a $20\text{ M}\Omega$ senza ottenere alcuna misurazione utile. In seguito, con il *VoltCraft* dotato di un range di misura più ampio e notevolmente più preciso, si è eseguita nuovamente la misura raggiungendo anche in questo caso il fondo scala pari a $50\text{ M}\Omega$. Si è allora passati ad una misura indiretta per quanto approssimativa, applicando una tensione di 10 V ai capi del sensore e misurando conseguentemente una corrente di circa $0,015\text{ }\mu\text{A}$, ottenendo pertanto una resistenza di circa $660\text{ M}\Omega$.

2.3 Setup di misura

Si è passati poi alla fase esecutiva della caratterizzazione. In Fig. 2.8 è mostrata una panoramica del setup di misurazione con particolare riguardo al sensore e alla possibilità di modificarne la sensibilità, mediante l'uso di resistenze esterne.

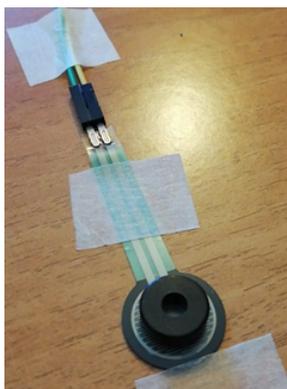


Figura 2.6: Particolare FSR 402.

Il sensore è in grado di percepire variazioni di forza anche se appoggiato lungo una superficie rigida, prestando attenzione ad applicare il peso solamente sulla parte sensibile, usando uno spessore, ad esempio come mostrato in Fig. 2.7. Bisogna evitare di applicare il peso lungo la ghiera esterna porterebbe il peso a “scaricarsi” lungo il guscio esterno di materiale plastico e non sulla parte sensibile, condizionando pertanto la misura.

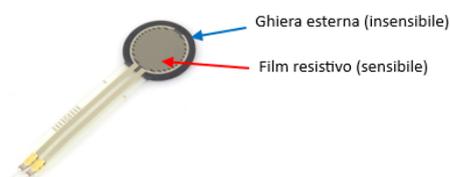


Figura 2.7: Particolare FSR 402.

Non è stato possibile misurare pesi inferiori a 20 g in quanto vi è resistenza meccanica da parte del materiale plastico di copertura del sensore che impedisce al film resistivo sottostante di percepire variazioni. Misure di forze di valore inferiore sono comunque complesse da fare, in quanto vi è una forte componente di rumore presente che permane nonostante l’attivazione di filtraggi all’interno del *Moku:Go*.

Il rumore in questione purtroppo deriva dall’uso dei cavi, sia a *morsetto* che a *jumper*, oltre che dalla *breadboard* e dai lunghi terminali del sensore. Inoltre, si deve conside-

rare la criticità rappresentata dalla scarsa superficie del film resistivo, di circa 1 cm^2 . La superficie ridotta comporta maggiore propensione ad oscillazioni dei pesi ed una maggiore sensibilità a vibrazioni sul piano di lavoro.



Figura 2.8: Visione d'insieme Moku:Go + FSR 402.

Il *Moku:Go* è in grado di campionare fino a 125 MSa/s e, nel caso considerato, visto il segnale in DC, si è optato per una finestra di 1 ms con una frequenza di campionamento pari a 144 kSa/s consigliata dallo stesso applicativo *Moku:Go*. I dati raccolti dal *Moku:Go* sono disponibili in pacchetti da 1024 elementi che contengono quindi 1024 valori del misurando, che vengono di volta in volta aggiornati con il progredire delle misure. Il vettore generato è vincolato ad una dimensione di 1024 campioni e non è possibile modificarlo.

2.4 Curve di caratterizzazione

L'obiettivo delle misure in oggetto è la replicazione delle curve di caratterizzazione di Fig. 2.9, riportate sul datasheet del sensore, per ognuno dei 10 FSR. Per quanto concerne le resistenze esterne, sono state usate le medesime riportate nel *technical note*, $100 \text{ k}\Omega$, $47 \text{ k}\Omega$, $30 \text{ k}\Omega$, $10 \text{ k}\Omega$ e $3 \text{ k}\Omega$. Sono stati presi in maniera empirica come pesi

di riferimento 25 g, 75 g, 125 g, 500 g e 1000 g allo scopo di ottenere un andamento apprezzabile della curva.

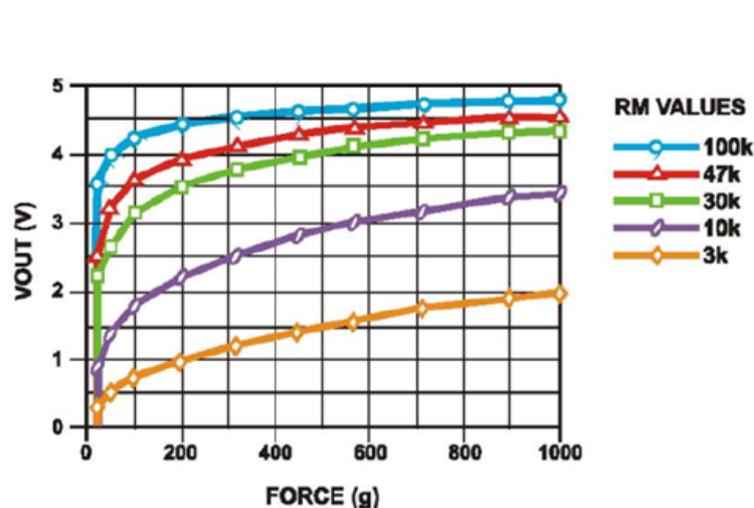


Figura 2.9: Curva caratteristica FSR.

A valle della sessione di misure effettuate, si è potuto constatare che la maggiore variabilità delle misure è presente con i valori di pesi più bassi, 25 g, ed in particolare usando le resistenze di valori inferiori quali quelle da 3 k Ω e 10 k Ω . Tutte le resistenze presentano una tolleranza pari al 5%.

In Fig. 2.10 si riporta il *flowchart* del codice sviluppato per la caratterizzazione dei sensori: essenzialmente si compone di un'acquisizione dei dati raccolti dal *Moku:Go* e di un calcolo di alcuni parametri statistici quali media, varianza e deviazione standard. I dati raccolti si compongono delle misure effettuate al variare di tutti i pesi e delle resistenze elencate in precedenza.

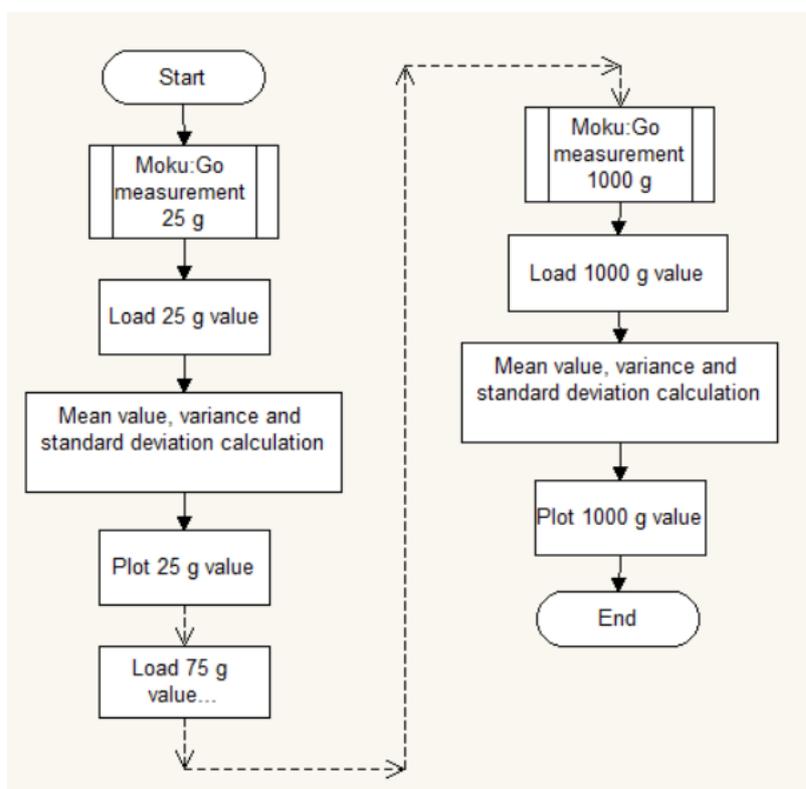


Figura 2.10: Caratterizzazione FSR.

Assumendo l'uso di una resistenza da $3\text{ k}\Omega$, è possibile notare un notevole allargamento della distribuzione gaussiana delle misure con bassi valori di resistenza, segnalando quindi un notevole aumento della variabilità delle misure della tensione rispetto al caso di misure a $100\text{ k}\Omega$ che sono notevolmente più stabili. Infatti, la gaussiana assume un andamento molto più stretto e pronunciato. Sull'asse delle ascisse è stato preso come riferimento un delta di 0.25 V per tutte le simulazioni.

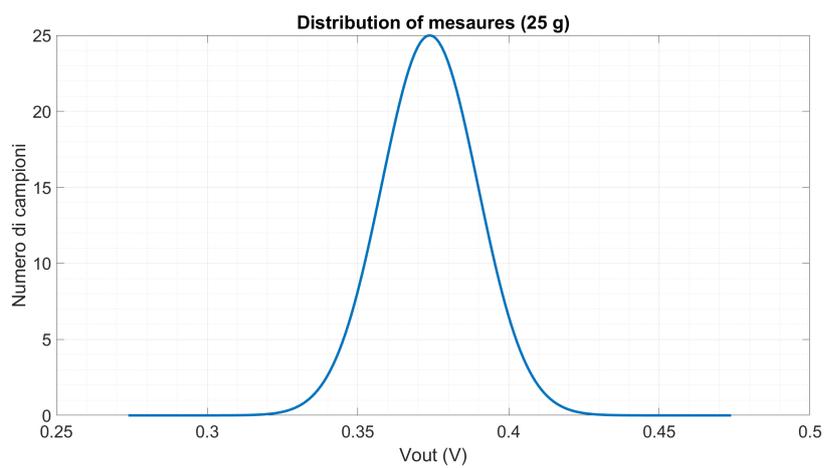


Figura 2.11: Distribuzione misure 3 k Ω - 25 g.

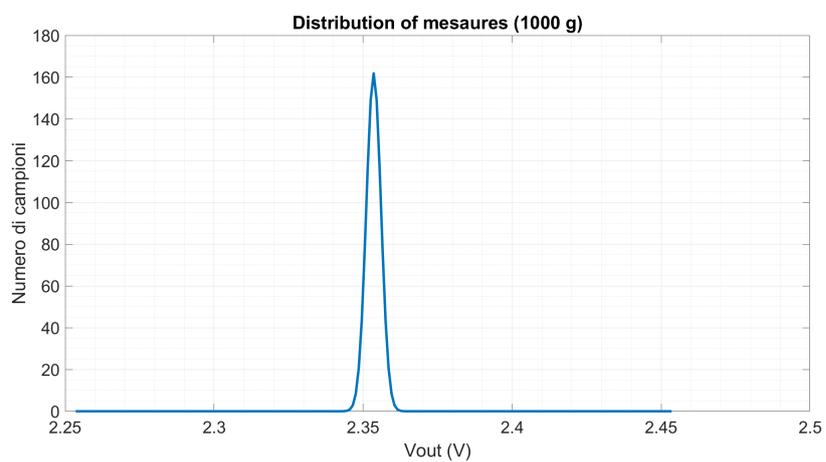


Figura 2.12: Distribuzione misure 3 k Ω - 1000 g.



Figura 2.13: Partitore R_M - R_{FSR} .

In Fig. 2.13 è riportato il partitore costituito dalla serie della resistenza R_M che di volta in volta assume un valore diverso ($100\text{ k}\Omega$, $47\text{ k}\Omega$, $30\text{ k}\Omega$, $10\text{ k}\Omega$, $3\text{ k}\Omega$) e dal valore della resistenza del sensore FSR ovviamente variabile in funzione della forza applicata.

A parità di forza applicata al sensore, abbiamo un degrado delle misurazioni della tensione lungo il partitore con valori di R_M bassi (caso $3\text{ k}\Omega$) rispetto al valore di R_M pari a $100\text{ k}\Omega$. Ciò deriva dalla caduta di tensione che avviene sulla resistenza R_M in quanto, con valori di resistenza inferiori, la caduta di tensione tende ad essere a sua volta minore e quindi più suscettibile a disturbi.

Inoltre, si deve considerare che con valori di peso superiori a 500 g inizia una saturazione del sensore, portando quindi ad una insensibilità e una conseguente robustezza delle misure a interferenze esterne.

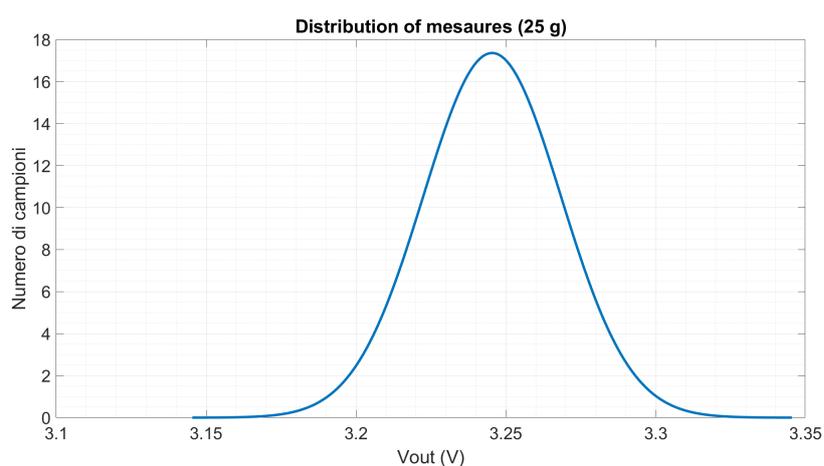


Figura 2.14: Distribuzione misure $100\text{ k}\Omega$ - 25 g .

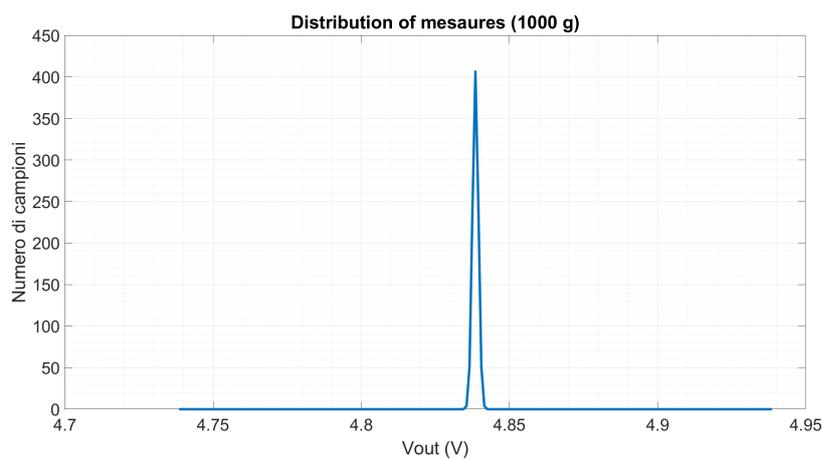


Figura 2.15: Distribuzione misure 100 k Ω - 1000 g.

I grafici che sono riportati di seguito rappresentano le caratterizzazioni dei singoli sensori. Tutti i sensori ripercorrono lo stesso andamento, come era prevedibile, ovviamente tutte le curve di caratterizzazione sono uniche e relative ai singoli sensori, in quanto, essendo un prodotto di natura industriale essi presentano delle ineliminabili differenze.

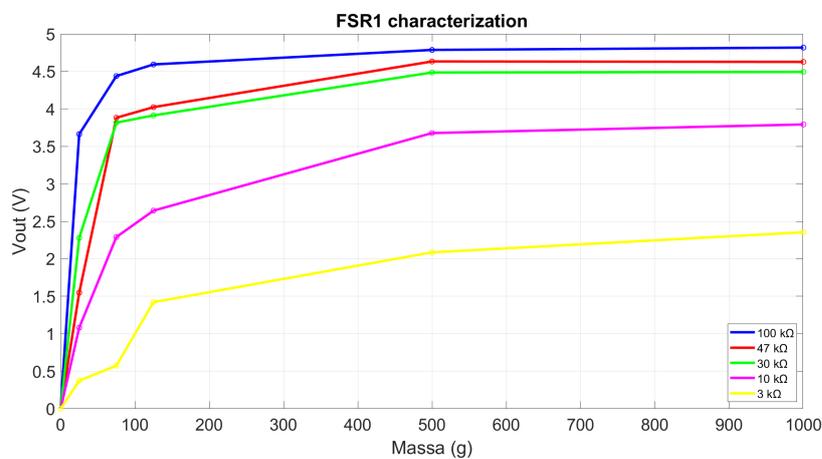


Figura 2.16: Caratterizzazione FSR1.

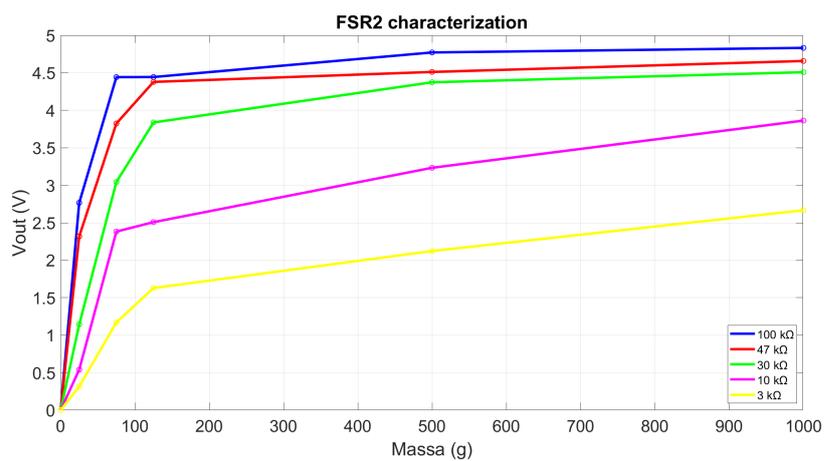


Figura 2.17: Caratterizzazione FSR2.

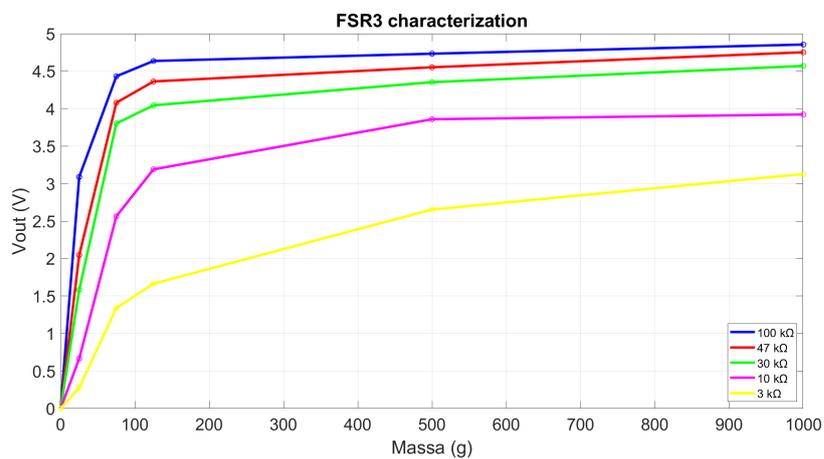


Figura 2.18: Caratterizzazione FSR3.

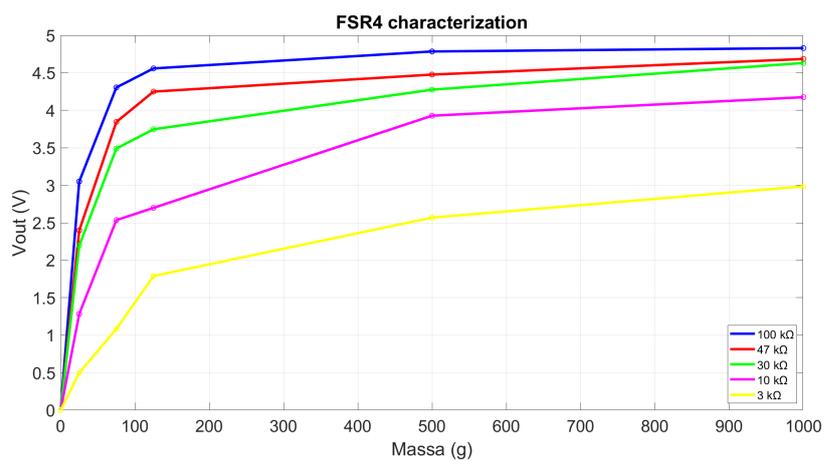


Figura 2.19: Caratterizzazione FSR4.

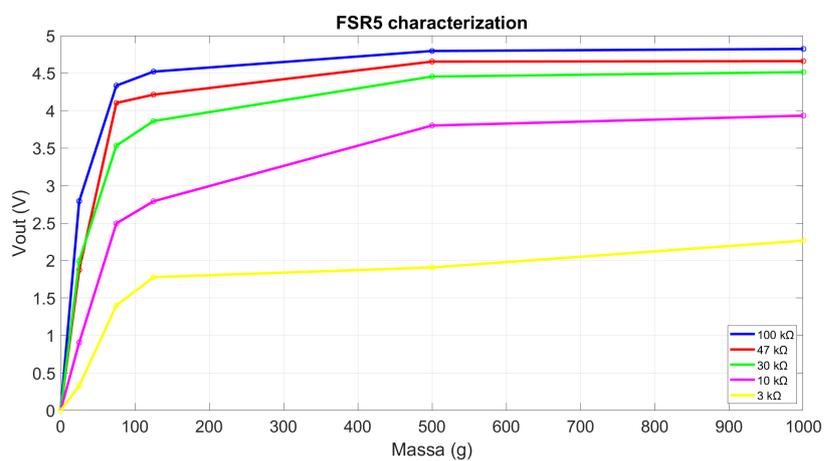


Figura 2.20: Caratterizzazione FSR5.

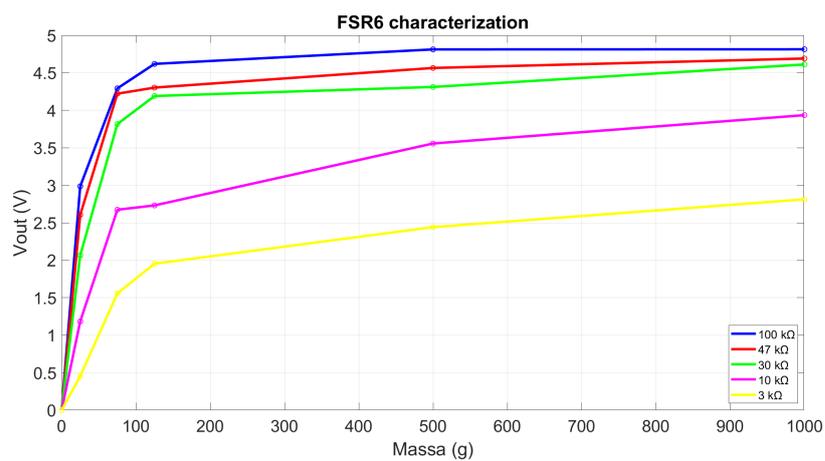


Figura 2.21: Caratterizzazione FSR6.

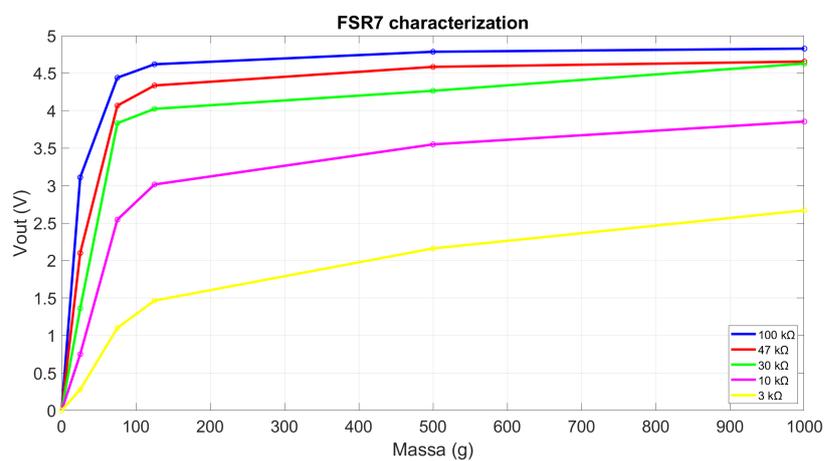


Figura 2.22: Caratterizzazione FSR7.

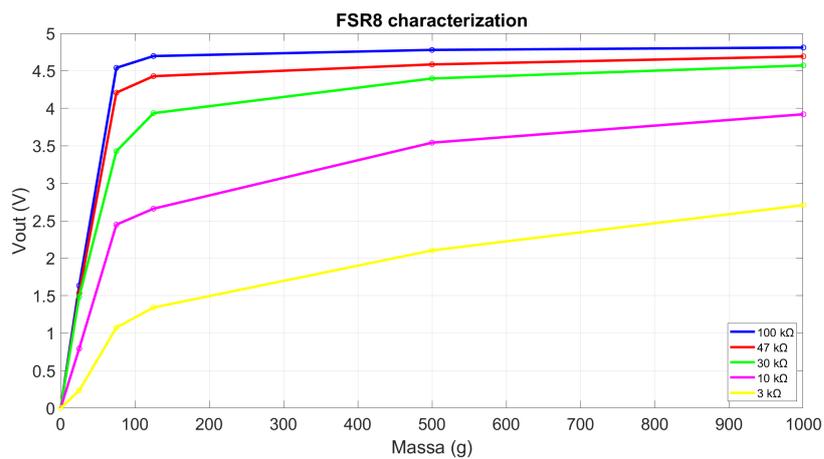


Figura 2.23: Caratterizzazione FSR8.

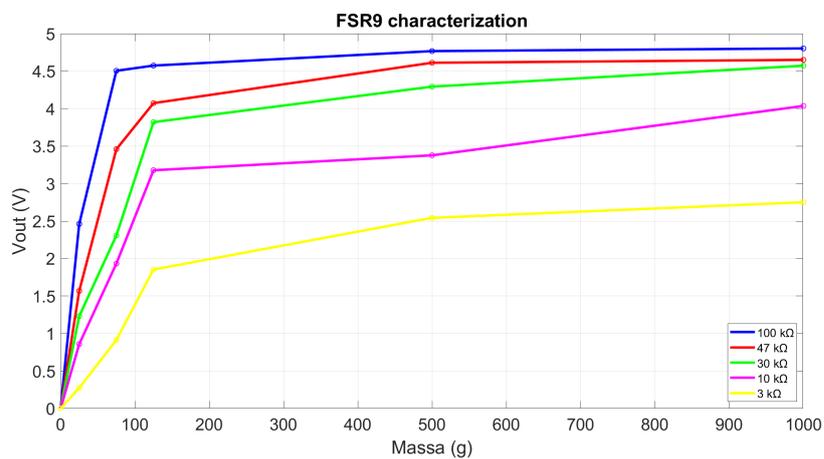


Figura 2.24: Caratterizzazione FSR9.

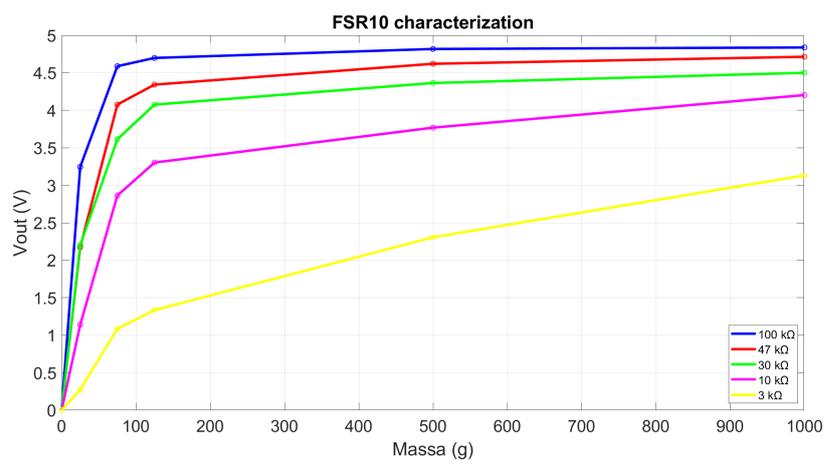


Figura 2.25: Caratterizzazione FSR10.

Capitolo 3

Schede di Sviluppo testate

Nel capitolo che segue sono riportate le motivazioni che hanno portato alla scelta della piattaforma da utilizzare nel prototipo. Sono inoltre presenti dei richiami alle piattaforme hardware ed ai sistemi software utilizzati.

3.1 Introduzione

Allo scopo di verificare la fattibilità di realizzazione del prototipo è stato affrontato lo studio di diverse piattaforme hardware di largo uso presenti in commercio, valutandone le prestazioni, i consumi di corrente, finanche l'occupazione di spazio e la necessità di utilizzo di sistemi esterni quali ad esempio un *Analogic Digital Converter* (ADC).

La prima fase di studio si è concentrata sulla possibilità di usare un sistema Raspberry Pi, il quale, oltre ad avere ingombri importati, necessita anche di sistemi esterni quali ADC e di diverse periferiche quali schermo e tastiera. E' seguito poi lo studio della possibilità di realizzare un sistema attraverso una Raspberry Pico, in questo caso non si è riusciti nell'intento a causa di criticità date dalla mancanza di librerie adeguate e da una poca flessibilità software di quelle presenti, come di seguito riportato. Infine si è cambiata completamente tipologia di piattaforma per passare alla famiglia *ESP32* (*Espressif 32*), in particolare la piattaforma ESP32 S3, che presenta un otti-

mo compromesso di consumi di corrente, ingombri e adattabilità di programmazione software.

3.2 Raspberry Pi 4



Figura 3.1: Raspberry Pi 4.

Raspberry Pi 4 [web2] è la quarta generazione del computer single-board della serie *Raspberry Pi*. Si discosta dalle precedenti versioni per una maggiore capacità di calcolo e memoria, oltre che all'aggiornamento hardware di alcuni elementi quali *GPU*, *RAM*, *CPU*, e periferiche quali *USB* e *HDMI* e degli standard di trasmissione quali *Bluetooth* o *Wi-Fi*. Sono disponibili 40 *GPIO* (General Purpose Input Output) che sono programmabili dall'utente, e compatibili con i principali formati di comunicazione quali *SPI*, *I²C* e *UART*. Il costo della scheda è pari a circa 70€.

La Raspberry Pi differisce dalle altre piattaforme in quanto non è concepita prevalentemente per la *programmazione embedded* quanto l'uso anche come computer, infatti, attraverso la Raspberry Pi è possibile svolgere funzioni quale navigare in internet o scrivere file *Latex* come in un qualsiasi altro PC che esulano dal concetto di microcontrollore in senso stretto.

A bordo della scheda è necessario installare un sistema operativo *Raspbian* o *Pi OS*, basato sulla distribuzione *Debian (Linux)*, vi è inoltre anche la possibilità di installare anche altri sistemi operativi in versione ridotta. La procedura guidata per il setup e

l'installazione del sistema operativo è disponibile [web3]. Le possibilità di sviluppo e di implementazione della scheda sono vaste ed eterogenee fra di loro.

Da segnalare è la necessità dell'uso di apparati esterni quale schermo mouse e tastiera. Nulla ovviamente vieta di programmare la scheda e usarla altresì come un microcontrollore nel senso stretto del termine, lasciando da parte un eventuale uso come computer, con ovvie limitazioni computazionali date dalla natura stessa della piattaforma. Tale procedimento esulerebbe dalla natura originaria della scheda. È possibile programmare la Raspberry Pi in differenti linguaggi come *C*, *C++* o *Python*, il quale è la scelta su cui è ricaduto il seguente lavoro. Per scrivere un codice è possibile usare uno dei compilatori consigliati e già installati a bordo quale Geany [web4].

3.2.1 Analogue Digital Converter: ADS 1115



Figura 3.2: ADC ADS 1115.

Una caratteristica di tutte le schede della famiglia Raspberry è la mancanza di ingressi analogici, ovvero, l'assenza a bordo di un ADC. Tutti gli ingressi sulla scheda sono di tipo digitale, quindi non sono in grado di leggere segnali che variano discretamente. Per ovviare a ciò si rende pertanto necessario l'uso di un convertitore analogico digitale esterno alla scheda.

Si è optato come convertitore analogico digitale l'*ADS 1115* [web16], che offre una precisione di 16 bit a 860 campioni/s. Il collegamento con la Raspberry Pi avviene tramite il protocollo di comunicazione *I²C*. Il convertitore può essere configurato con

4 canali di ingresso *single-ended*, o come due canali differenziali. Eventualmente è disponibile anche un amplificatore del segnale se necessario.

In Fig. 3.3 è riportato uno schema di principio del convertitore. Sul lato sinistro sono presenti gli ingressi analogici che vengono scelti uno alla volta dal *MUX* (Multiplexer), al fine abbattere i costi in quanto con questa tecnica è necessario avere soltanto un ADC. Segue un eventuale passaggio (PGA) di amplificazione del segnale. A valle vi è la fase di effettiva conversione analogico digitale del tipo *Sigma-Delta* a 16 bit. Sul lato destro sono riportati invece gli ingressi del protocollo *I²C*, *SDA*, *SCL* e *ADDR*. È inoltre disponibile anche un ulteriore pin di alert come segnalamento di superamento di soglia.

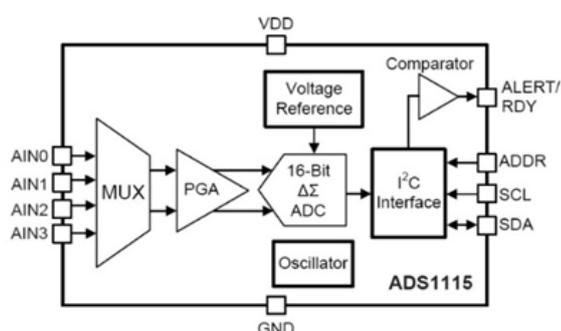


Figura 3.3: Schema a blocchi ADC ADS 1115.

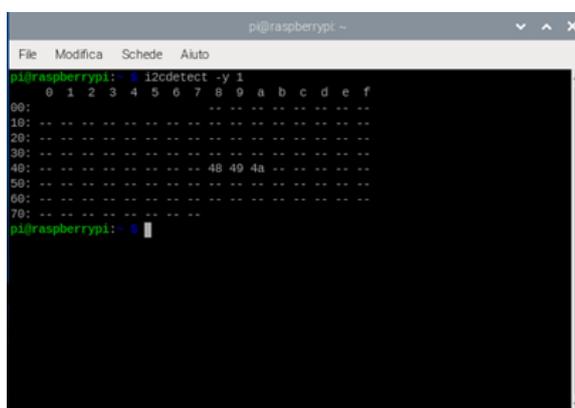
Ogni *ADS 1115* è in grado di gestire la conversione analogico digitale al massimo di 4 sensori, nel caso in essere, avendo a che fare con 10 FSR, è necessario usare almeno 3 *ADS* i quali vengono gestiti singolarmente attraverso l'*I²C* mediante l'opportuno collegamento fisico del pin *ADDR* il quale imposta un indirizzo diverso in riferimento al valore impostato, come riportato in Tab. 3.1. Ad un indirizzo viene associato un unico convertitore ADC, è possibile associare fino ad un massimo di quattro indirizzi.

3.2.2 Lettura sensori

Per la verifica del corretto collegamento dei convertitori ADC al rispettivo indirizzo è stata effettuata una prova via terminale attraverso il comando `i2cdetect -y 1`. È possibile vedere dalla Fig. 3.4 infatti i 3 indirizzi configurati.

Address	ADDR Pin Connection
0x48	GND
0x49	VDD
0x4A	SDA
0x4B	SCL

Tabella 3.1: Connessione pin ADDR.



```

pi@raspberrypi ~
File Modifica Schede Aiuto
pi@raspberrypi:~$ i2cdetect -y 1
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  -- 48 49 4a --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
pi@raspberrypi:~$

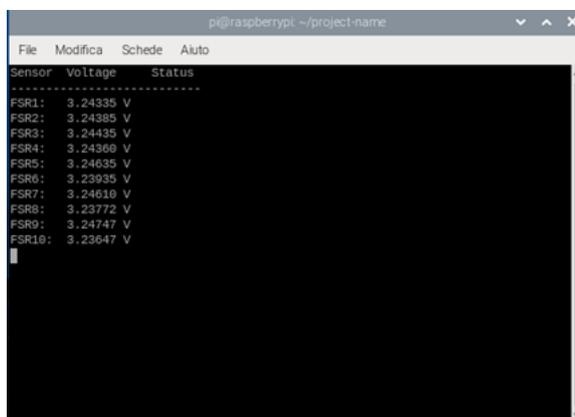
```

Figura 3.4: Istruzione verifica I^2C .

Il controllo e la gestione del convertitore sono possibili in modo facilitato attraverso l'impiego dell'opportuna libreria disponibile in [web5]. Si tenga presente che tale libreria in realtà era nata per un uso su piattaforme differenti dalla Raspberry Pi, infatti, per poterla utilizzare su quest'ultima è necessario utilizzare il terminale in un ambiente virtuale lanciando direttamente il comando di esecuzione e non è possibile richiamarla da un editor come *Geany*

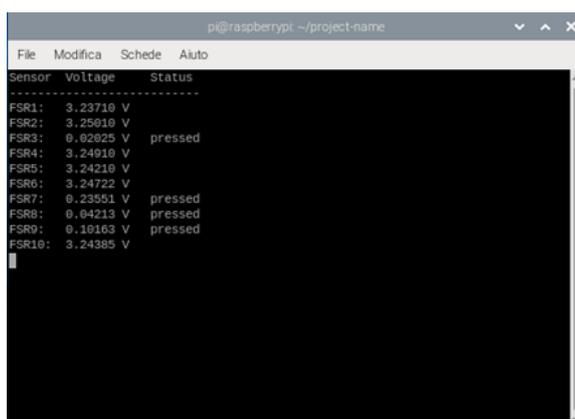
Di seguito sono riportati dei risultati preliminari del software e quindi del prototipo sviluppato in cui sono raffigurati in modo grafico gli FSR che vengono attivati. Quando non sono soggetti a forza hanno indicativamente tutti la medesima tensione pari a circa 3.24 V. Nel codice è stata impostata infatti una soglia di innesco, risultato di vari test empirici, che indica che effettivamente sta agendo una forza sul sensore solo al di sotto di un valore pari a 3.0 V. Infatti, per valori di tensione inferiori possiamo

affermare con certezza che sul sensore sta operando una forza come ad esempio una mano.



```
pi@raspberrypi: ~/project-name
File Modifica Schede Aiuto
-----
Sensor Voltage Status
FSR1: 3.24335 V
FSR2: 3.24385 V
FSR3: 3.24435 V
FSR4: 3.24368 V
FSR5: 3.24635 V
FSR6: 3.23935 V
FSR7: 3.24618 V
FSR8: 3.23772 V
FSR9: 3.24747 V
FSR10: 3.23647 V
```

Figura 3.5: FSR non attivati.



```
pi@raspberrypi: ~/project-name
File Modifica Schede Aiuto
-----
Sensor Voltage Status
FSR1: 3.23718 V
FSR2: 3.25918 V
FSR3: 0.02025 V pressed
FSR4: 3.24918 V
FSR5: 3.24218 V
FSR6: 3.24722 V
FSR7: 0.23551 V pressed
FSR8: 0.04213 V pressed
FSR9: 0.10163 V pressed
FSR10: 3.24385 V
```

Figura 3.6: FSR3 – FSR7 – FSR8 – FSR9 attivati.

In Fig. 3.7 viene riportato uno schema di sintesi del programma sviluppato. Si compone di una iniziale gestione delle librerie, a cui segue una lettura ed un controllo a soglia del valore a cui succede un'attivazione o meno del sensore.

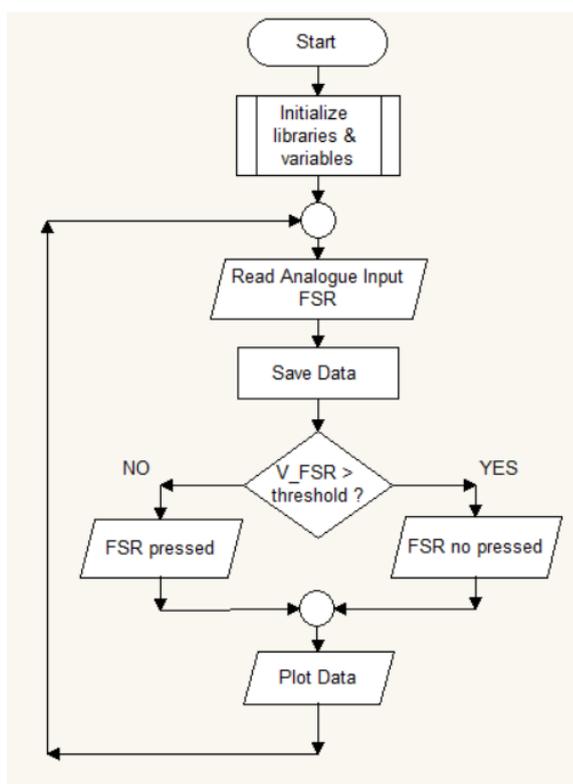


Figura 3.7: Lettura sensori FSR Raspberry Pi 4.

Di seguito viene riportata una panoramica del sistema, sono state usate delle resistenze da $100\text{ k}\Omega$, sulla falsariga di quanto riportato in precedenza per avere maggiore stabilità e affidabilità nei valori. Allo scopo di verificare la fattibilità di un sistema *stand alone*, sono stati misurati i consumi di corrente della Raspberry Pi 4 pari a circa 600 mA , valore di corrente non idoneo ad un sistema dove è richiesto un basso consumo energetico. infine si aggiunge la criticità data dalla Raspberry Pi 4 di non disporre di nessun di sistema *watch-dog* integrato il quale lo rende dipendente dalla presenza di un operatore in caso di crash.

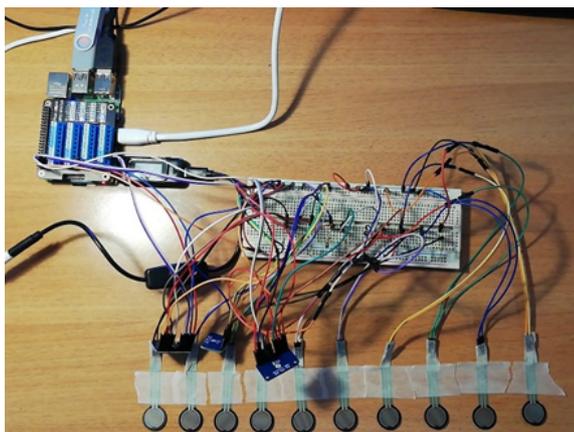


Figura 3.8: Panoramica sistema.

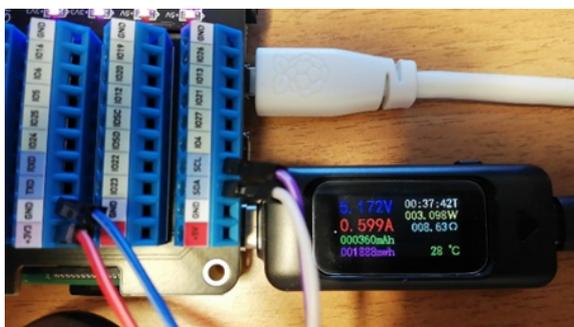


Figura 3.9: Particolare consumo corrente.

3.3 Raspberry Pi Pico

Raspberry Pi Pico differisce dalla precedente piattaforma utilizzata in quanto classificata come scheda microcontrollore, che per loro natura ha ottimizzazioni in termini di consumi, e delle conseguenti limitazioni per quanto riguarda la capacità computazionale. Si anticipa che con la Raspberry Pi Pico purtroppo non si è riusciti a creare un sistema in grado di leggere correttamente i sensori, mentre parallelamente si è potuto progettare ciò attraverso la ESP32 S3, grazie ai suoi consumi inferiori e convertitori analogici digitali già integrati.

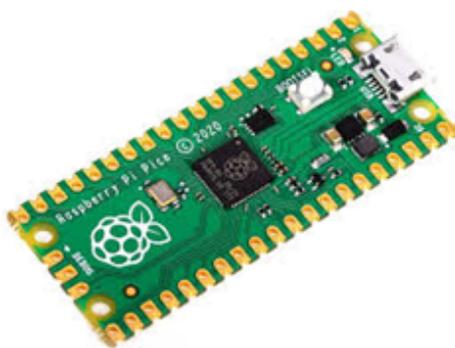


Figura 3.10: Raspberry Pi Pico.

La Raspberry Pi Pico è il primo microcontrollore della famiglia *Raspberry Pi*, è basata su un chip *RP2040* e differisce infatti dalla canonica famiglia Raspberry Pi in quanto orientato per la prima volta ad applicazioni di natura *embedded*, spostandosi quindi dalla caratteristica familiare del microcomputer. La peculiarità della Raspberry Pi Pico, rispetto ai simili della stessa famiglia è quella di essere più semplice in termini di utilizzo e di non necessitare di un sistema operativo. Ovviamente ha prestazioni e supporto alla memoria molto più limitati rispetto a quelle, ad esempio, di una Raspberry Pi 4.

Avendo un taglio più basso permette di ottenere un funzionamento real-time, indicato in applicazioni orientate ad un risparmio energetico e ad un consumo in termini di risorse di calcolo non eccessivo. Infatti, rispetto alla scheda precedentemente utilizzata, Raspberry Pi 4, le potenze di calcolo in questo caso sono nettamente inferiori a favore dei consumi di corrente di un ordine di grandezza inferiore, come riportato in Tab. 3.2.

Raspberry Pi Pico nasce con l'idea di supportare nativamente i linguaggi *C*, *C++* e anche *MicroPython*, una implementazione di *Python* per microcontrollori. Per ottenere il massimo delle prestazioni dal Raspberry Pi Pico in termini di latenza, ad esempio, gli sviluppatori consigliano il *C*. È possibile comunque usare il *MicroPython* anche su altri microcontrollori come ad esempio la famiglia *ESP32*. Si consideri che i pin del *Pico* sono a 3.3 V, e inoltre sono disponibili 26 pin di Input/Output digitali e 3 ingressi analogici rendendo quindi ancora necessario l'uso di un convertitore ana-

logico digitale come un *ADS 1115* visto nel caso precedente. Sono disponibili come protocolli di comunicazione *I²C*, *UART* e *SPI*, oltre che vari pin per la gestione *PWM*. I costi di acquisto della scheda sono di circa 15€, inferiore rispetto alla Raspberry Pi 4.

Per poter programmare la scheda, si è optato per il *MicroPython* sulla falsa riga di quanto già fatto con la Raspberry Pi 4. L'azienda produttrice consiglia di programmare la scheda via il software *Thonny* [web6]. Si possono usare altri applicativi per programmare la scheda quali l'IDE di *Arduino* ma vi è il rischio di non avere a disposizione tutte le librerie e di problemi di incompatibilità data dalla natura della *Raspberry* di nascere in ambiente *Python*. Dopo avere scaricato l'applicativo è sufficiente ripercorrere i passi [web7].

La medesima libreria che è stata utilizzata in precedenza con la Raspberry Pi 4, ha presentato problemi importanti di incompatibilità, nonostante l'appartenenza alla medesima famiglia. Per questo motivo si è optato per una nuova libreria che è risultata essere a sua volta incompleta. Come si riporta di seguito sono state fatte numerose operazioni di debug e di aggiunte alla libreria in quanto, quest'ultima era predisposta solamente per la lettura di un FSR, comportando quindi la non possibilità di leggere più sensori e gestirli attraverso più di un ADC.

Il cuore della libreria è dato dalla gestione del registro di Fig. 3.11, in particolar modo riferito ai bit 14:12, attraverso i quali avviene la gestione in maniera multiplexata degli ingressi dell'ADC, come riportato in Fig. 3.12. Sono state inoltre effettuate delle modifiche minori in sede di gestione via *I²C*.

Figure 36. Config Register

15	14	13	12	11	10	9	8
OS	MUX[2:0]			PGA[2:0]			MODE
R/W-1h	R/W-0h			R/W-2h			R/W-1h
7	6	5	4	3	2	1	0
DR[2:0]		COMP_MODE	COMP_POL	COMP_LAT	COMP_QUEUE[1:0]		
R/W-4h		R/W-0h	R/W-0h	R/W-0h	R/W-3h		

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Figura 3.11: Registro di configurazione.

14:12	MUX[2:0]	R/W	0h	Input multiplexer configuration (ADS1115 only) These bits configure the input multiplexer. These bits serve no function on the ADS1113 and ADS1114. 000 : AIN _p = AIN0 and AIN _n = AIN1 (default) 001 : AIN _p = AIN0 and AIN _n = AIN3 010 : AIN _p = AIN1 and AIN _n = AIN3 011 : AIN _p = AIN2 and AIN _n = AIN3 100 : AIN _p = AIN0 and AIN _n = GND 101 : AIN _p = AIN1 and AIN _n = GND 110 : AIN _p = AIN2 and AIN _n = GND 111 : AIN _p = AIN3 and AIN _n = GND
-------	----------	-----	----	---

Figura 3.12: Bit 14:12 Registro di Configurazione.

Come riportato, la criticità era data dalle operazioni di *shift* dei registri interni alla scheda a 16 bit, ai quali non era possibile associare nuovi bit e quindi un nuovo ingresso di lettura dell'ADC, come si riporta nella figura che segue, per ovviare alle rigidità si è modificata la libreria attraverso delle nuove variabili (x). È stato possibile quindi leggere i 10 FSR attraverso i tre convertitori ADC 1115, ma l'ordine di lettura non veniva associata correttamente all'input del MUX impostato, a titolo esplicativo, l'FSR1 viene letto al posto del FSR3, il quale a sua volta è letto al posto del FSR2 e così via, presentando una sorta di casualità nella lettura.

Nonostante diversi tentativi è stato possibile leggere i sensori in modo anche corretto in termini di valori, con un range di lettura fino a 3.3 V, ma non è possibile associarli correttamente all'FSR effettivamente oggetto di misurazione.

```
def read_value(self, x):
    self.i2c.writeto(self.address, bytearray([0]))
    result = self.i2c.readfrom(self.address, 2)
    config = self.read_config()
    config &= ~(7 << 12) & ~(7 << 9)
    config |= (x << 12) | (1 << 9) | (1 << 15)
    config = [int(config >> i & 0xff) for i in (8, 0)]
    self.i2c.writeto(self.address, bytearray([1] + config))
    return result[0] << 8 | result[1]
```

Figura 3.13: Modifica della libreria.

A scopo di completezza si è comunque proceduto ad una misurazione dei consumi di corrente della Raspberry Pi Pico, attestandosi a circa 26.5 mA. Il modulo *ADS 1115* invece presenta dei consumi trascurabili pari a circa 300 μ A.



Figura 3.14: Consumi corrente Raspberry Pi Pico.

3.4 ESP32 S3

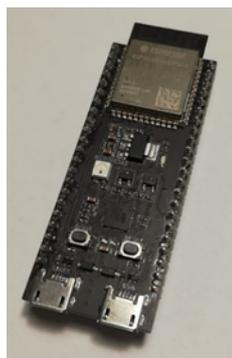


Figura 3.15: ESP32 S3.

A seguito dell'impossibilità di completare il progetto attraverso la Raspberry Pi Pico, si è passati ad un ulteriore piattaforma quale la ESP32 S3, che rappresenta uno degli ultimi dispositivi che è stato lanciato e che va ad arricchire la ricca famiglia degli *ESP32*. È dotata di un *MCU* dual-core da 240 MHz con possibilità di espansione della memoria. Inoltre, l'ESP32 S3 dispone di un ricco set di periferiche I/O, con ben 44 GPIO programmabili e supporta tutte le periferiche comunemente utilizzate, come *SPI*, *I2S*, *I2C*, *PWM*, *RMT*, *ADC* e *UART*, è dotata di un core a bassi consumi che supporta diverse modalità di funzionamento. La scheda presenta un costo di acquisto di circa 10€.

Come anticipato in precedenza per poter programmare la ESP32 S3 è possibile adoperare come linguaggi di programmazione sia il *C*, *C++* o anche il *Python*. Per comodità di utilizzo e data la presenza di librerie idonee si è programmato via l'editor di *Thonny* [web3] utilizzando il *Python*.

La piattaforma ESP32 S3, nonostante presenti il vantaggio di avere già a bordo dei convertitori analogici digitali (a 10 bit) che non necessitano di librerie aggiuntive, presenta un problema legato al campo di misura inadeguato, il quale è compreso fra 0.1 V - 3.1 V, rendendo impossibile la misura di valori superiori a 3.1 V, in particolare 3.3 V, che corrisponde alla tensione ai capi di un sensore FSR quando non soggetto a nessuna forza.

Come riportato nella Fig. 3.16 dagli stessi costruttori, anche in seguito a operazioni di calibrazione software e hardware che linearizzano per quanto possibile la caratteristica di comportamento del convertitore, comunque permane la criticità del clipping dei valori prossimi a 3.3 V.

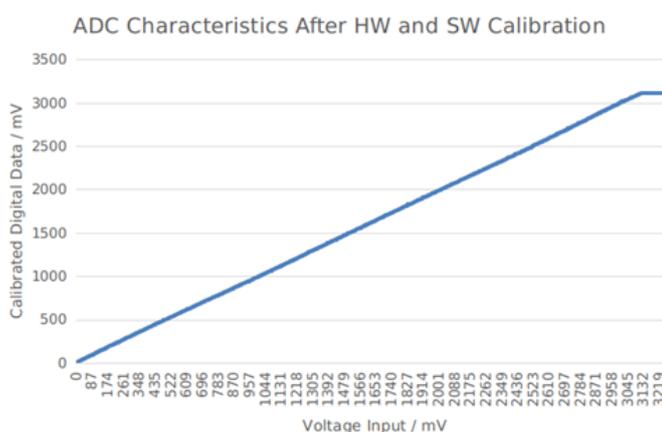


Figura 3.16: Caratteristica ADC ESP32.

Il costruttore della scheda, per ovviare a tale inconveniente consiglia di costruire un partitore resistivo esterno al sistema e procedere quindi ad una misura di tensione inferiore al massimo consentito che si rapporta con quella incognita in maniera indiretta. Tale possibilità è da escludere nel caso in essere in quanto stiamo trattando resistenze oltre che variabili quali l'FSR anche con valori estremamente elevati, (caso

senza forze agenti). Un'altra strada consigliata ma che risulta scomoda e non agevole è della riscrittura dei registri interni mediante una procedura di *Range Extension*, la quale, pena una non linearità e un offset, aumenta il campo di misura del convertitore ma non permettendo comunque di raggiungere i 3.3 V.

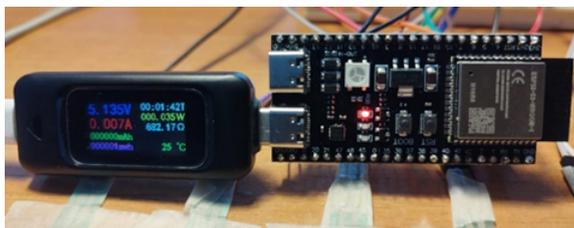


Figura 3.17: Consumi corrente ESP32 S3.

È riportata di seguito una panoramica del sistema sviluppato attraverso l'ESP32 S3, si noti l'assenza di convertitori analogici digitali esterni. Per dare una continuità ai vari elementi del progetto sviluppati, anche in questo caso si sono adottate delle resistenze da 100 k Ω . Si nota anche una maggiore compattezza dello stesso rispetto al precedente sviluppato con la Raspberry Pi 4.

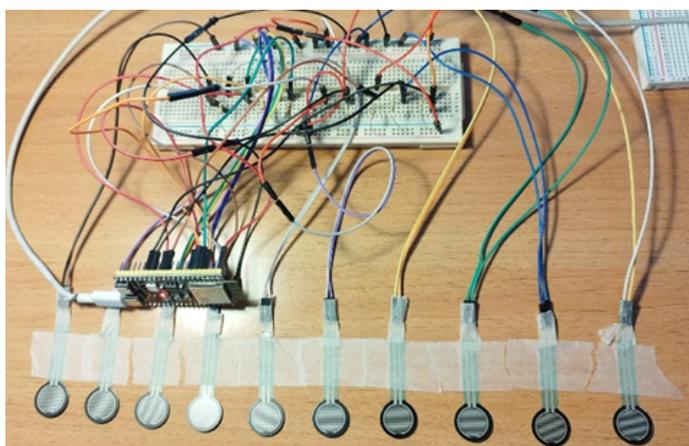


Figura 3.18: Sistema ESP32 S3 + FSR.

3.5 Consumi di corrente

In sede di analisi, sono stati presi in considerazione solamente i consumi dello stato attivo del dispositivo, non valutando lo stato di *deep sleep* in quanto, il nostro sistema è progettato per essere sempre in stato di lettura dei sensori, quando ovviamente sono presenti i maggiori consumi.

Piattaforma utilizzata	Consumo Corrente (mA)
RaspBerry Pi 4	599
ESP32 S3	8
RaspBerry Pi Pico	26.5

Tabella 3.2: Consumi di corrente.

Lo stato attivo ovviamente presenta dei consumi maggiori ma permette di svolgere la lettura dei sensori, e quella di una eventuale elaborazione dati che sarebbero ovviamente impossibili in stato di *sleep*. Si deve inoltre tenere presente che lo stato attivo del dispositivo, per quanto oneroso, non presenta grossi problemi nel caso in cui si voglia considerare il sistema come una *black box* alimentato attraverso l'impianto elettrico del veicolo durante la fase attiva, eventualmente, sarebbe possibile impostare uno stato di *deep sleep* da attivarsi al momento dello spegnimento del veicolo ai fini del mantenimento della carica della batteria di quest'ultimo.

Indipendentemente dalla criticità mostrata dalla Raspberry Pi Pico con l'inaffidabilità delle librerie ad oggi utilizzate, quindi della non possibilità d'uso, la scelta molto probabilmente sarebbe comunque ricaduta sull'uso di una ESP32 S3. In primis dalla caratteristica di avere già integrati al proprio interno dei moduli ADC, permettendo quindi di evitare l'uso di dispositivi esterni quali *ADC 1115* con minore possibilità di guasti, non da meno, vi è la questione di minore ingombro, meno cavi e quindi minore necessità di spazio, in ottica di una configurazione *stand alone*.

In secundis vi è la questione relativa ai consumi come riportato in Tab. 3.2, ove è facilmente desumibile come la ESP32 S3 sia la piattaforma che garantisce consumi

inferiori nello stato attivo, tale consumo di corrente sicuramente può essere oggetto di migliorie scegliendo frequenze di *clock* inferiori e disattivando moduli non necessari quali il *Wi-Fi* ad esempio.

Capitolo 4

Acquisizione dati FSR

Lo scopo del presente capitolo è la descrizione dell'acquisizione e della successiva elaborazione grafica dei dati raccolti dai sensori FSR tramite la piattaforma ESP32 S3 attraverso lo sviluppo di un prototipo con relativo tool software.

4.1 Prototipo sviluppato

Viene riportato un'immagine del prototipo sviluppato vedi Fig. 4.1 e Fig. 4.2, si notino in particolare i sensori montati sulla corona esterna del volante e la piattaforma ESP32 S3 e la scheda con i relativi partitori resistivi installati all'interno del volante dove viene di norma alloggiato l'airbag. Tale assetto è sviluppato anche ai fini della riduzione dell'ingombro durante una guida e di una possibile fattibilità del prototipo come *black box*.



Figura 4.1: Prototipo volante.



Figura 4.2: Retro prototipo volante.

I 10 sensori FSR sono stati posizionati in maniera equa sulla corona del volante ad una distanza di circa 12 cm l'uno dall'altro (lunghezza della corona esterna di 117 cm) in posizioni tali da essere toccati con le mani durante la guida. Si può anticipare che, dalle prove effettuate, i sensori più soggetti a forza siano il n. 3, 4, 7 e 8, in quanto sono delle più classiche posizioni di guida. Altri sensori come ad esempio i n.1, 10, 5 e 6 situati nella parte superiore e inferiore del volante vengono ovviamente soggetti ad una interazione minore, eccetto in caso di una curva, ad esempio. Le figure riportate vanno lette in funzione dello stato del sensore che segue una logica inversa, infatti, si ricorda che presenta una tensione pari a 3.3 V quando non soggetto a forza. Il valore

di tensione invece viene a modificarsi scendendo al di sotto dei 3.3 V in funzione della forza esercitata quando avviene una interazione con quel settore di volante.

Come ulteriore metodo di confronto sono state calcolate le medie del valore di tensione del singolo FSR, che equivale ad uno specifico settore del volante. L'allontanamento dal valore medio di 3.3 V indica una attività notevole del sensore ovvero, che in quel settore vi è stato un notevole utilizzo del volante.



Figura 4.3: Particolare ESP32 S3 e partitore resistivo.

Sempre sulla falsa riga del precedente capitolo, è stata eliminata la parte puramente prototipale della *breadboard* ed è stata iniziata una finalizzazione attraverso l'impiego di una *scheda millefori* con relative saldature per i partitori resistivi composti dalle 10 resistenze da 100 k Ω . Ciò ha permesso di eliminare falsi contatti oltre che a migliorare la stabilità meccanica del circuito. In questo caso, non è stato necessario l'uso di ADC esterni in quanto già implementati all'interno della ESP32 S3.

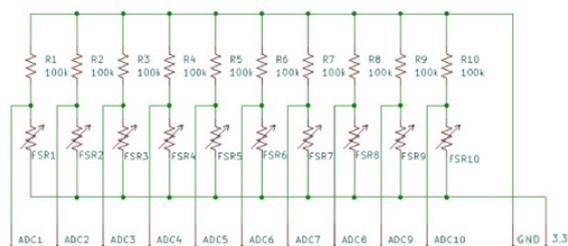


Figura 4.4: Schema elettrico partitore resistivo.

I sensori nonostante siano posizionati su una superficie concava, per quanto minima, come quella della corona del volante, non presentano problemi di letture o valori diversi rispetto a quelli precedentemente visti su una superficie piana. Anche in questo caso, la non totale rigidità del materiale del volante non reca problemi alla lettura. Durante delle simulazioni del prototipo, inoltre, si è potuto constatare come la soglia di innesco dei sensori lungo il volante ai fini del riconoscimento di un contatto pari a 3.0 V è veritiera, in quanto tale valore ci permette di discriminare in maniera affidabile il comportamento del sensore quando è effettivamente premuto scartando tutte le possibili false letture. Sono state effettuate inoltre diverse prove empiriche del volante azionando volutamente i sensori in differente ordine e modo e controllati via terminale verificando che vi fosse accordanza fra i due.

4.2 Tool sviluppato

Sono stati sviluppati due programmi diversi, il primo lavora a bordo della ESP32 S3 ed è quello che effettivamente acquisisce i dati raccogliendo le misurazioni dei sensori con una frequenza di campionamento di 4 Hz ovvero ogni 0.25 s, e salvandole all'interno di un file tipo .CSV nella memoria flash interna. I dati raccolti su file, quindi, vengono analizzati ed elaborati via MatLab.

4.2.1 Acquisizione Dati

Il primo programma sviluppato è simile a quello considerato nel capitolo che precede come in Fig. 3.7, ed è costituito semplicemente da una fase di lettura degli ingressi analogici della ESP32 S3, unica differenza rispetto ai precedenti è la scrittura e salvataggio dei dati raccolti all'interno di un file mediante la formattazione dei dati entro una stringa per la generazione del file tipo .CSV.

```
f.write(str("{:2.3f}".format(FSR1_value)) + ',' + str("{:2.3f}".format(FSR2_value)) + ',' + \
str("{:2.3f}".format(FSR3_value)) + ',' + str("{:2.3f}".format(FSR4_value)) + ',' + \
str("{:2.3f}".format(FSR5_value)) + ',' + str("{:2.3f}".format(FSR6_value)) + ',' + \
str("{:2.3f}".format(FSR7_value)) + ',' + str("{:2.3f}".format(FSR8_value)) + ',' + \
str("{:2.3f}".format(FSR9_value)) + ',' + str("{:2.3f}".format(FSR10_value)) + '\n')
```

Figura 4.5: Scrittura valori all'interno di file.

Viene riportato in Fig. 4.8 uno schema di massima del programma in cui è possibile evincere una classica struttura ciclica di lettura e salvataggio dei dati raccolti.

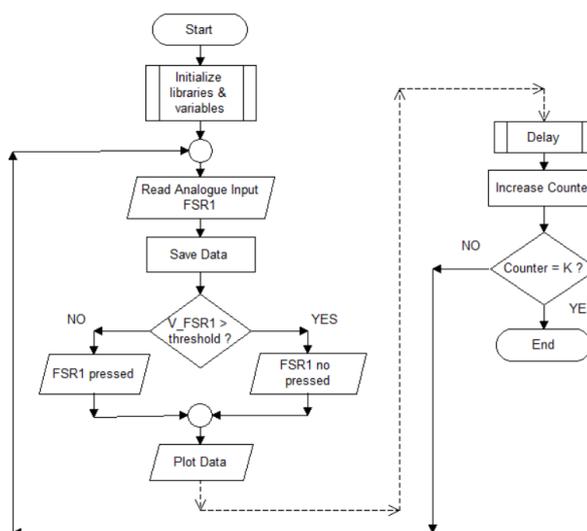


Figura 4.6: Flowchart lettura FSR.

4.2.2 Elaborazione Dati

Per quanto riguarda invece il secondo programma che gestisce unicamente l'elaborazione grafica dei valori ottenuti ed il calcolo della loro media, è formato da una prima parte di creazione di 10 vettori (uno per ogni FSR) di N elementi in funzione del numero delle letture da effettuare. Tale fase è essenziale in quanto la libreria non è in grado di lavorare in tempo reale ma solamente con dati già acquisiti e in memoria. Segue una fase di apertura del file contenente le letture per estrapolare i singoli dati di ogni sensore.

```

with open('lettture.csv') as f:
    c=0
    for line in f:
        y1[c] = float(line.strip(',')[0:5])
        c += 1

```

Figura 4.7: Apertura file dati.

Per la gestione della rappresentazione grafica dei dati del volante si è utilizzata la libreria *pyCirclize* la quale in realtà non è nata per il disegno di figure geometriche ma nasce dal mondo delle Scienze in generale per la rappresentazione di dati del tipo genetici e genomici in formato di circonferenze di differenti tipologie. Per maggiori informazioni si rimanda al link su *GitHub* [web8] dove è possibile prendere visione della trattazione dell'intera libreria e di come poterla installare.

In Fig. 4.8 è riportato un breve schema logico del funzionamento del programma relativo alla rielaborazione grafica dei dati raccolti.

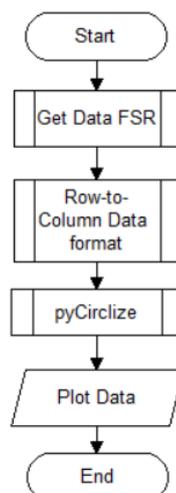


Figura 4.8: Flowchart utilizzo libreria *pyCirclize*.

I dati estrapolati dal file vengono inseriti in vettori i quali, opportunamente ordinati, vengono dati alla libreria che procede alla rappresentazione grafica all'interno di una circonferenza precedentemente suddivisa in 10 settori al fine di delineare o, meglio, simulare, in maniera grafica i sensori FSR con le loro variazioni di tensione.

```

for sector in circos.sectors:

    sector.text(f"{sector.name}", r=110, size=15)

    x = np.arange(sector.start, sector.end)

    # Plot points
    track = sector.add_track((75, 100), r_pad_ratio=0.1)
    track.axis()
    track.scatter(x, yy[c])

    # Plot line
    line_track = sector.add_track((45, 70), r_pad_ratio=0.1)
    line_track.axis()
    line_track.line(x, yy[c])

    c += 1

```

Figura 4.9: Libreria pyCirclize.

4.3 Dati raccolti

Seguono adesso una serie di immagini esplicative del tool sviluppato. Il volante è rappresentato come una circonferenza i cui settori sono la rappresentazione di un singolo FSR. Il movimento lungo la circonferenza rappresenta l'andamento temporale sempre del singolo sensore, i 10 settori sono eterogenei fra di loro e lavorano in modo distinto; infatti, se il sensore non è soggetto a forze sul grafico vi è una rappresentazione di una tensione continua di 3.3 V. Il raggio della circonferenza invece ha rappresentazione della tensione misurata, purtroppo per limiti della libreria utilizzata non è possibile riportare le unità di misura.

Di seguito sono rappresentati in termini grafici e numerici sotto forma di media delle simulazioni di differenti tipi di guida effettuate sul volante. Una prova è relativa al volante senza essere soggetto a nessuna interazione, infatti, dal grafico in Fig. 4.10 si possono desumere tensioni sempre pari a 3.3 V, come anche le medie. Ogni settore è anche identificato con una colorazione diversa, in questo caso si nota che tutti i settori sono colorati sul perimetro esterno del cerchio indicando che nessun sensore è stato "toccato", o meglio oggetto di una forza.

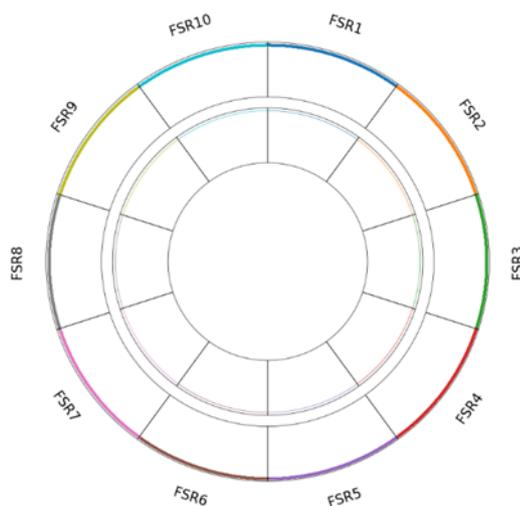


Figura 4.10: Volante non utilizzato.

```

Mean value FSR1: 3.300 V
Mean value FSR2: 3.300 V
Mean value FSR3: 3.300 V
Mean value FSR4: 3.300 V
Mean value FSR5: 3.300 V
Mean value FSR6: 3.300 V
Mean value FSR7: 3.300 V
Mean value FSR8: 3.300 V
Mean value FSR9: 3.300 V
Mean value FSR10: 3.300 V
  
```

Figura 4.11: Valori tensione media riferiti a Fig. 4.10.

Segue adesso i risultati ottenuti con uno stile di guida tipico di un lungo rettilineo, come in Fig. 4.12, ovvero con le mani sul volante nelle posizioni centrali, quindi sensori 2, 3, 4, 7, 8 e 9, modificando ovviamente le posizioni assunte durante la guida. I sensori 1, 10, 5 e 6 invece sono sempre a tensione costante in quanto non soggetti a nessun contatto durante una guida in assenza di cambiamenti di traiettoria del veicolo, come può essere una guida lungo un tratto di strada dritto.

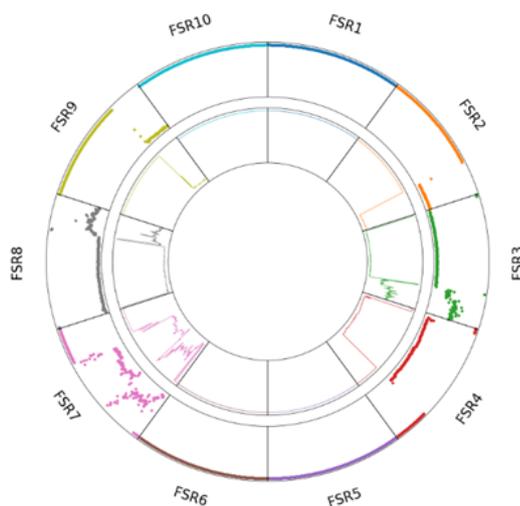


Figura 4.12: Volante con guida su rettilineo.

Mean value FSR1:	3.300 V
Mean value FSR2:	2.482 V
Mean value FSR3:	0.432 V
Mean value FSR4:	1.074 V
Mean value FSR5:	3.300 V
Mean value FSR6:	3.300 V
Mean value FSR7:	1.753 V
Mean value FSR8:	0.344 V
Mean value FSR9:	2.477 V
Mean value FSR10:	3.300 V

Figura 4.13: Valori tensione media riferiti a Fig. 4.12.

In questo ulteriore step, come in Fig. 4.14 sono state svolte una simulazione di guida estremizzata con molte curve e assenza di rettilinei, infatti, si desume un forte uso dei sensori 1, 2, 5, 6, 9 e 10, lasciando inalterati i sensori 3, 4, 7 e 8.

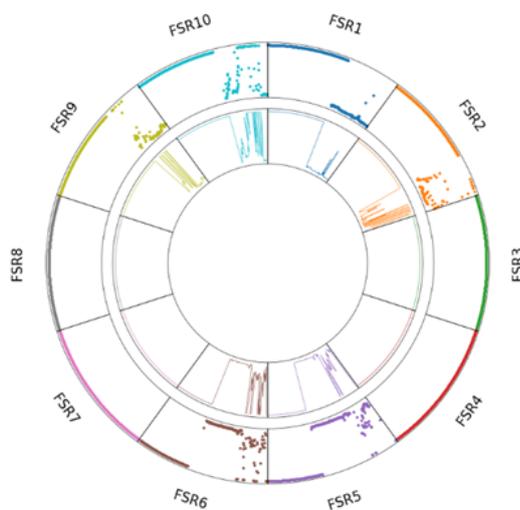


Figura 4.14: Volante con guida su curve.

Mean value FSR1:	2.037 V
Mean value FSR2:	2.497 V
Mean value FSR3:	3.300 V
Mean value FSR4:	3.300 V
Mean value FSR5:	1.629 V
Mean value FSR6:	1.763 V
Mean value FSR7:	3.300 V
Mean value FSR8:	3.300 V
Mean value FSR9:	2.456 V
Mean value FSR10:	2.574 V

Figura 4.15: Valori tensione media riferiti a Fig. 4.14.

Come ultimo stile di guida è stato improntato quello del classico stile di guida misto, come in Fig. 4.17, con un fittizio percorso di curve e rettilinei.

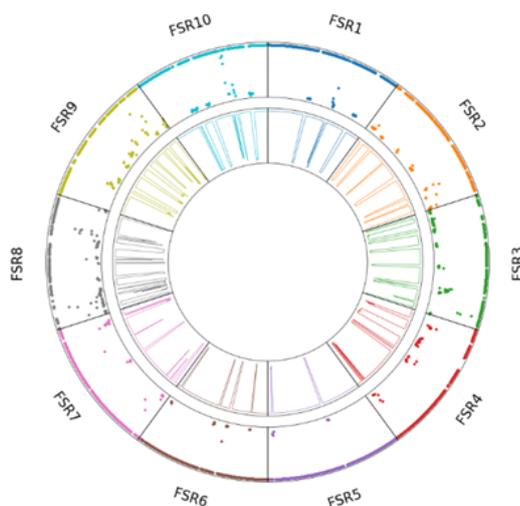


Figura 4.16: Volante con guida su percorso misto.

Mean value FSR1:	3.015 V
Mean value FSR2:	2.760 V
Mean value FSR3:	2.514 V
Mean value FSR4:	2.716 V
Mean value FSR5:	3.186 V
Mean value FSR6:	3.112 V
Mean value FSR7:	3.027 V
Mean value FSR8:	2.284 V
Mean value FSR9:	2.600 V
Mean value FSR10:	2.829 V

Figura 4.17: Valori tensione media riferiti a Fig. 4.16.

I grafici ottenuti hanno carattere indicativo e variano molto in funzione dello stile di guida del singolo, è lecito pensare ad esempio che nel caso in cui si voglia affrontare una curva, si è propensi a prendere il volante per la parte superiore attivando i corrispettivi sensori, e non impugnando la parte centrale, come in Fig. 4.12.

Partendo dal presupposto che sono state fatte delle prove con il prototipo senza potere avere un effettivo riscontro “pratico” su una guida reale o simulata, in questa fase si sono usati dei video trovati in rete di situazioni di guida, per permettere di avere comunque un riferimento durante le prove.

Capitolo 5

Sensore GSR

Viene adesso ad essere trattato il secondo sensore implementato nel prototipo, ovvero relativo alla risposta galvanica della pelle. E' riportata di seguito in primis una contestualizzazione delle grandezze fisiche in gioco, segue una trattazione relativa al sensore utilizzato. Tale sensore, rispetto al precedente trattato, è in grado di fornirci in maniera diretta dati importanti relativi a stati di stress del guidatore.

5.1 Galvanic Skin Response

Nel presente capitolo viene descritto lo studio e la strutturazione del prototipo relativo alla misura della Galvanic Skin Response (GSR) che rappresenta la variazione della conduttanza elettrica della pelle. L'unità di misura è espressa in μS , che varia in funzione della secrezione di sudore da parte delle ghiandole sudoripare eccrine, la quale a sua volta deriva da una risposta di un soggetto ad uno stimolo esterno o interno di tipo emotivo come ad esempio lo stress. Prima di entrare nel merito del prototipo, è doveroso fare una breve contestualizzazione delle grandezze fisiche in gioco e della loro origine fisiologica.

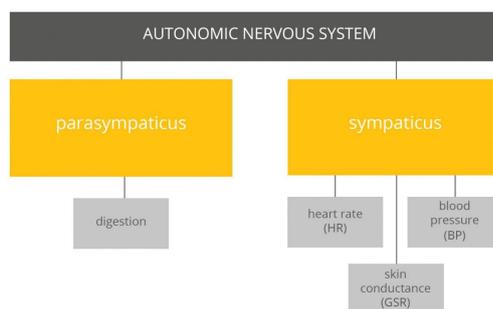


Figura 5.1: Sistema Nervoso Autonomo.

La sudorazione del corpo umano è regolata dal Sistema Nervoso Autonomo (ANS), il quale viene diviso in due componenti, il sistema simpatico, e quello parasimpatico che entrano in azione in momenti diversi e che, pur agendo sugli stessi organi, hanno effetti diametralmente opposti. Vengono controllate funzioni come la dilatazione delle pupille, la produzione della saliva, del muco e del sudore, la frequenza cardiaca, la contrazione dei muscoli dei bronchi, i movimenti di stomaco e intestino o l'accumulo di glicogeno nel fegato. Il sistema simpatico interviene nelle situazioni di emergenza quali fuga o stress, mentre il sistema parasimpatico agisce nei momenti di rilassamento. In particolare, se il ramo simpatico (SNS) del sistema nervoso autonomo è altamente attivato, l'attività delle ghiandole sudoripare aumenta di conseguenza, incrementando a sua volta la conduttanza cutanea, e viceversa. In questo modo, la conduttanza cutanea può essere una misura delle risposte del Sistema Nervoso Simpatico umano in termini generali [web9].

In sintesi per i nostri scopi possiamo affermare che ogni volta che il nostro corpo percepisce uno stimolo che potrebbe modificare il nostro stato di riposo, il SNS attiva in modo autonomo e non comandabile una risposta fisiologica che comprende un'accelerazione della frequenza cardiaca, una dilatazione della pupilla e un aumento dell'attività delle ghiandole sudoripare. Questa risposta autonoma è innescata da qualsiasi tipo di reazione emotiva, come sorpresa, paura o rabbia, o quando siamo sotto stress [web10].



Figura 5.2: Tipologie di ghiandole sudoripare.

Il fenomeno della sudorazione che si va a rilevare con i sensori GSR, dipende dalle ghiandole sudoripare che sono di due tipi: le ghiandole sudoripare eccrine, presenti su tutta la superficie cutanea e particolarmente concentrate nelle ascelle, sulla fronte, palmo delle mani e pianta dei piedi, e quelle apocrine, presenti solo in alcune zone: ascelle, areole mammarie, regione inguinale e perineale. Le ghiandole eccrine impediscono un eccessivo aumento della temperatura corporea, per le ghiandole apocrine, invece, l'obiettivo è di stimolare, con l'odore, l'interesse sessuale. Quando una persona è soggetta ad un'interazione che sia positiva che negativa di tipo emotivo (ad esempio stress) il sistema nervoso simpatico si attiva; quindi, l'attività delle ghiandole sudoripare eccrine aumenta proporzionalmente e indipendentemente dalle volontà dello stesso, e di conseguenza anche la conduttanza cutanea. Quindi, ricapitolando, ci sono due diversi tipi di sudore:

- durante la sudorazione termica indotta dalle temperature elevate o dall'attività fisica, solo le ghiandole sudoripare eccrine sono attive;
- durante le situazioni di stress emotivo sono attive sia le ghiandole eccrine che apocrine.

Nel caso attuale, non vi è fatta alcuna differenza dalla tipologia di sudorazione attiva in quanto l'obiettivo posto nel seguente lavoro è la ricerca e l'analisi di un eventuale stato di stress di un individuo alla guida di un autoveicolo. È doveroso analizzare la condizione di stress e renderla distinta dalla condizione, ad esempio, di guida ad alte temperature che potrebbe avvenire ad esempio durante il periodo estivo, che potrebbe

portare ad una sudorazione fino a venire ad essere scambiata con uno stato di stress.

La sudorazione causa un aumento di conduttività in quanto il sudore è prevalentemente costituito da acqua e sali minerali. Tale tecnica di misura per quanto semplice permette di mettere in correlazione la condizione psichica interna ed esterna con quella esterna di un individuo. Volendo adesso innestarci più all'aspetto ingegneristico, l'attività di un segnale GSR è costituita da una componente tonica ed una fasica, queste due tipi di attività elettrodermica, sono analizzabili in termini di variazione di resistenza elettrica e presentano le seguenti caratteristiche:

- l'attività tonica (SCL), che esprime il valore assoluto della resistenza elettrica cutanea, costituisce un indice dello stato generale di attivazione del sistema nervoso dell'organismo.
- l'attività fasica (SCR), cioè le rapide risposte provocate da stimoli prettamente emozionali, sensoriali o ideativi, in questo segnale sono presenti numerose fluttuazioni rapide.

Di seguito è riportata una rappresentazione grafica di un segnale GSR. È frequente l'uso come acronimo di EDA (ElectroDermal Activity) in letteratura equivalente al già noto GSR.

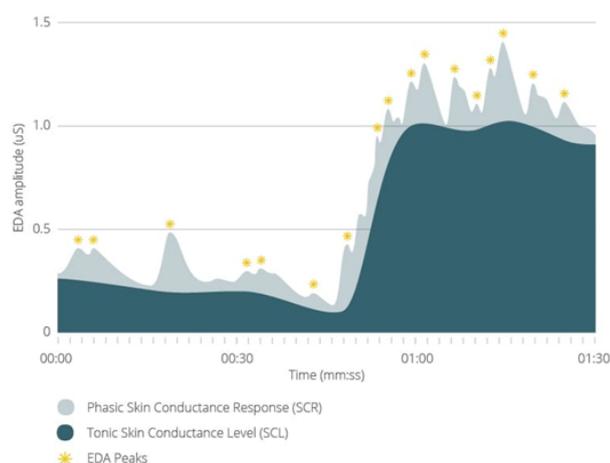


Figura 5.3: Attività SCL e SCR di un segnale GSR.

5.2 Sensore GSR

Il sensore utilizzato nell'ambito delle misurazioni di conduttanza della pelle è il Grove GSR sensors [web11] della *Seeed Studio*. Si fa presente fin da ora che il sensore non misura la conduttività ma la resistenza del corpo umano attraverso la *legge di Ohm* iniettando una corrente nota e misurando la tensione in conseguenza della resistenza della pelle.

Il sensore è costituito da due elettrodi che permettono di stabilire il contatto elettrico tra il circuito e la pelle. Nel caso in essere, questi ultimi sono costituiti in nichel in quanto garantisce economicità, biocompatibilità e riduce lo sviluppo di potenziali di polarizzazione che potrebbero falsificare le misure. I due elettrodi vengono applicati sul dito indice e medio della mano ed è consigliato di usare la mano non dominante in sede di misure in quanto con pelle meno inspessita. Gli elettrodi iniettano una corrente che, in funzione di una resistenza diversa data dalla sudorazione corrisponde una differente tensione ai capi degli elettrodi.

Di seguito in Fig. 5.4 è riportato lo schema elettrico del circuito di condizionamento del sensore. Sono visibili due amplificatori operazionali in configurazione inseguitore

o, meglio, separatore di impedenza tra il circuito a monte ed a valle. Segue un ulteriore amplificatore operazionale in configurazione differenziale il quale fornisce in uscita l'amplificazione delle differenze delle tensioni in ingresso. Sono presenti vari condensatori per il disaccoppiamento di eventuali correnti in alternata e di eventuali disturbi elettromagnetici. È presente anche un potenziometro da 200 k Ω per variare la sensibilità del circuito.

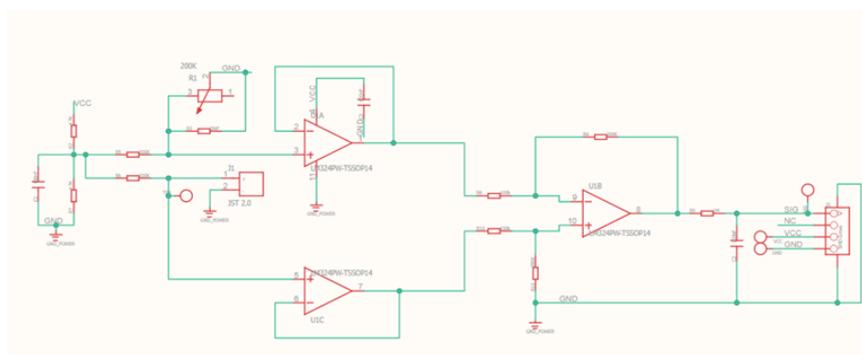


Figura 5.4: Schema elettrico del circuito di condizionamento.

Al fine di semplificare il sistema nel complesso e renderlo anche ipoteticamente utilizzabile durante una guida il contatto degli elettrodi è effettuato con delle fasce elastiche già presenti all'interno del kit fornito dalla casa costruttrice evitando l'uso di gel e similari che avrebbero garantito un contatto totale ma allo stesso tempo poca praticità d'uso.

Nella Fig. 5.5 è presente il sensore nel complessivo, l'elemento sensibile è in realtà l'elettrodo a forma di bottone all'interno degli elastici. La scheda elettronica interposta fra gli elettrodi e il microcontrollore è sostanzialmente il circuito di condizionamento riportato nella Fig. 5.4.

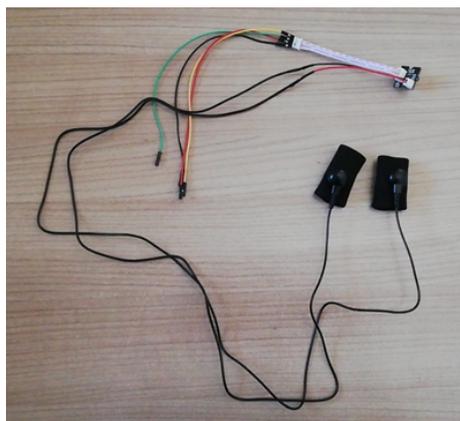


Figura 5.5: Sensore GSR Seed Studio.

5.2.1 Piattaforma Hardware

Al fine di poter lavorare adeguatamente con il sensore, la casa costruttrice consiglia come piattaforme hardware l'Arduino UNO e la Raspberry Pi di cui fornisce anche delle librerie software [web11]. Allo scopo di ottenere, come largamente discusso nei precedenti documenti un sistema *black box* con riduzione degli spazi e con un'ovvia continuità progettuale rispetto a quanto già fatto si è provveduto ad utilizzare invece la piattaforma ESP32 S3.

5.2.2 Accorgimenti per le misure

Per la raccolta dei dati, come in precedenza segnalato, è doveroso precisare che dovrebbero avvenire in ambiente climatizzato e con umidità costante. La misura di conduttività in quanto funzione della quantità di sudore sulla pelle, ovviamente dipende in maniera significativa da temperatura e in particolare umidità. In ambiente con alte temperature si è propensi a sudare per questioni relative alle termoregolazione, portando ad una falsificazione delle misure, ipotesi peggiorativa ulteriore che può essere data anche da un ambiente di guida umido [web12]. Sono da evitare anche ambienti con correnti d'aria in quanto il corpo umano risponderebbe in modo discontinuo a causa della termoregolazione. Pertanto, le misure sono state eseguite in

ambiente climatizzato e con temperatura comunque confortevole e tale da impedire che avvenisse una sudorazione in grado di falsificare le misure.

Capitolo 6

Acquisizione dati GSR

Nel capitolo che segue viene descritto lo sviluppo del tool attuato allo scopo di acquisire dei dati dal sensore di conduttanza. Vengono altresì descritti i motivi di talune scelte e delle soluzioni tecniche adottate allo scopo di ottenere dei dati tangibili.

6.1 Tool sviluppato

Allo scopo di raccogliere dei dati utili ai nostri scopi è stato necessario costruire tre distinti codici in *Python*, con i primi due che nonostante una similitudine hanno una diversa elaborazioni dei dati, e l'ultimo con obiettivo la portabilità dei dati da *Python* a *MatLab*. Entrambi i codici in questione svolgono la lettura attraverso un ingresso analogico del microcontrollore ESP32 S3 nella medesima maniera di quanto discusso nei precedenti capitoli. Per potere ricavare la resistenza della pelle (Ω), i costruttori del sensore offrono la possibilità di calcolarla in loco mediante l'uso di una formula [web11]:

$$HumanResistance = \frac{(maxADCvalue + 2 \cdot SerialPortReading) \cdot 10000}{Serialcalibration - SerialPortReading} (\Omega) \quad (6.1)$$

Dove, sostituendo opportunatamente i valori si ottiene:

$$HumanResistance = \frac{(4095 + 2 \cdot GSRvalue) \cdot 10000}{2047 - GSRvalue} (\Omega) \quad (6.2)$$

avendo indicato con *GSRvalue* il valore fornito in uscita dall'ADC e indicato con *SerialPortReading* nella equazione (6.1).

Ovviamente, è possibile ricavare anche la conduttanza della pelle (S) attraverso il calcolo dell'inverso della resistenza della pelle. A titolo puramente informativo, è lecito aspettarsi valori di resistenza della pelle nell'ordine del centinaio di k Ω e della conduttanza della pelle nell'ordine di una decina di μ S. Per maggiore chiarezza, di seguito è riportato un frammento di codice dove vengono sviluppati il calcolo della resistenza e della conduttanza della pelle.

```
num = (4095 + 2*gsr_average)*10000 # numerator of the formula
den = 2047-gsr_average # denominator of the formula

human_resistance = num/den # datasheet resistance formula
human_resistance = round(human_resistance)

conductance = 1/(human_resistance) # conductance
conductance = conductance*1000000 # uS
```

Figura 6.1: Calcolo resistenza e conduttanza della pelle.

6.1.1 Calibrazione

Prima di essere messo in funzione, il sensore necessita di una calibrazione. Come consigliato dagli stessi costruttori [web11], tale procedura è molto semplice e consiste nel regolare il trimmer in maniera tale che, con gli elettrodi a vuoto, si ottenga in uscita (ovvero il valore attribuito alla variabile *SerialPortReading* in eq. (6.1)) l'esatta metà del range di funzionamento dell'ADC in uso (0-4095 nel caso in questione - 12 bit), ovvero 2047. In tale maniera si potrà sfruttare efficacemente tutta la dinamica disponibile.

6.1.2 Acquisizione GSR grezza

Il codice è formato da una lettura dei valori del convertitore ADC in quanto il dispositivo è in grado di fornirci unicamente un valore di tensione grezzo letto dal convertitore, che è a sua volta legato al valore di conduttanza come riportato in for-

mula (6.1). In Fig. 6.2 è riportato uno schema di funzionamento del tool sviluppato.

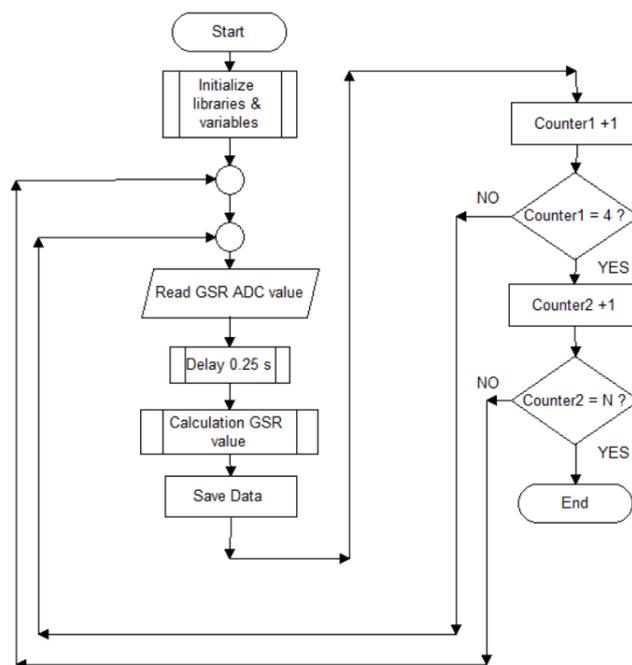


Figura 6.2: Acquisizione GSR.

Il primo dei codici, come in Fig. 6.2, si sviluppa essenzialmente attraverso l'esecuzione di N cicli che derivano dalle 4 letture al secondo (4 Hz), dato scelto al fine di ottimizzare il costo computazionale e, per la verifica dei dati ottenuti mediante confronto con il dispositivo *Empatica E4*, dispositivo che usa la medesima frequenza di campionamento di 4 Hz. Si eseguono N cicli che, date 4 letture al secondo e ripetute per 60 s, si ottengono i minuti di lettura richiesti. Segue una fase di formattazione e scrittura dei dati all'interno di un file. E' inserito un ritardo di 0.25 s allo scopo di compiere 4 letture al s. Ad ogni lettura è associato un valore di conduttanza.

```

while n<2400: # 10(point for s)*60(second)*4(minute)
    n = n + 1
    tot = 0;
    i = 0;

    for i in range (1,11): # Average the 10 measurements to remove the glitch
        GSR_value = GSR.read() # reading analog pin 1
        tot = tot + GSR_value
        sleep(0.010) # 0.010*10=0.10 s

    gsr_average = tot/10; # average value
    gsr_average = round(gsr_average) # approximate number

    voltage = gsr_average*(3.3/4095);

    num = (4095 + 2*gsr_average)*10000 # numerator of the formula
    den = 2047-gsr_average # denominator of the formula

    human_resistance = num/den # datasheet resistance formula
    human_resistance = round(human_resistance)

    conductance = 1/(human_resistance) # conductance
    conductance = conductance*1000000 # uS

```

Figura 6.3: Lettura dati GSR.

6.1.3 Acquisizione GSR mediata

La differenza data dal secondo codice visibile in Fig. 6.4 è l'implementazione all'interno di un *ciclo for* allo scopo di rimuovere segnali spuri quale un *glitch*, ad esempio. Si svolge una media di 10 campioni misurati al fine di omogeneizzare e rendere il più possibile vero il valore misurato finale e abbattere eventuali campioni falsati. Il tempo di lettura complessivo varia in funzione della prova sperimentale. Successivamente il codice prosegue in maniera completamente analoga al precedente.

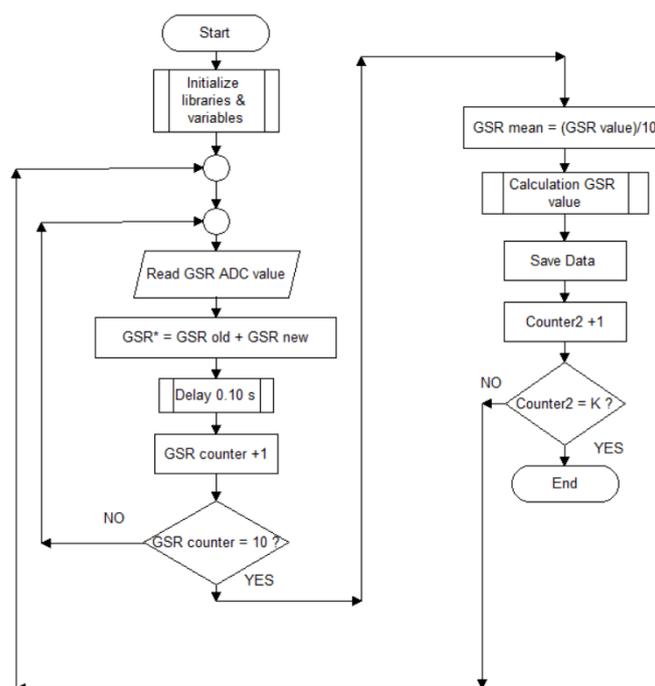


Figura 6.4: Acquisizione GSR "no glitch".

```

for i in range(1,11): # Average the 10 measurements to remove the glitch
    GSR_value = GSR.read() # reading analog pin 1
    tot = tot + GSR_value
    sleep(0.010) # 0.010*10=0.10 s

gsr_average = tot/10; # average value
gsr_average = round(gsr_average) # approximate number
  
```

Figura 6.5: Lettura dati GSR con eliminazione glitch.

Viene riportato un frammento del codice che converte i dati raccolti in un formato adeguato a *Matlab* attraverso l'istruzione *savemat*. Tale istruzione non è presente nativamente su *Python* ma è necessario installarla autonomamente [web13].

```

with open('conductance.csv') as f: # open data file
    c=0
    for line in f:
        data[c] = float(line.strip())[0:5]
        c += 1

savemat('conductance.mat', {'data': data}, oned_as='row')
  
```

Figura 6.6: Lettura dati GSR con eliminazione glitch.

La frequenza di campionamento del segnale GSR è stata scelta a 4 Hz in quanto quest'ultimo ha delle frequenze a livello fisiologico comprese fra 1 Hz e 0.2 Hz, che cambiano da individuo a individuo. Con frequenze di campionamento minori si sono ottenuti dei buoni risultati. Non è possibile aumentare la frequenza di campionamento ulteriormente in quanto viene prelevato molto rumore oltre che per questioni relative all'eccessivo onere computazionale dei codici sviluppati, in particolare durante la fase di porting *Python - Matlab*. Inoltre, è stato necessario utilizzare la medesima frequenza di 4 Hz di campionamento del dispositivo *Empatica E4* al fine di poter realizzare un confronto con il sensore adoperato nel prototipo della *Seed Groove*. Il dispositivo *Empatica E4* è dotato di prestazioni metrologiche maggiori permettendo di verificare l'affidabilità dei dati raccolti

6.2 Dati raccolti

A seguito di una serie di misure effettuate con il sensore si è constatato come effettivamente il sistema sia in grado di determinare stati di stress di un individuo. Ovviamente come stato di stress principe per le nostre simulazioni future verrà considerato quello della stanchezza a lungo termine durante la guida di un veicolo.

Nelle figure che seguono è riportato l'andamento della conduttanza della pelle, ovvero del segnale GSR, il quale si ricorda è l'inverso della resistenza della pelle. Con uno stato di stress che porta fisiologicamente ad una sudorazione maggiore, si ha una diminuzione della resistenza o aumento della conduttanza. Viceversa, con uno stato di rilassamento si ha una diminuzione nella produzione di sudore ovvero, un aumento della resistenza quindi una diminuzione della conduttanza. I grafici che seguono sono rappresentativi della conduttanza della pelle.

In Fig. 6.7 è riportato un andamento della conduttanza rilevata dal sensore da cui è desumibile come il livello scenda e salga con il passare del tempo indicando quindi una serie di stati di stress e rilassamento che possono essere dati magari da una "agitazione" provocata dal dover indossare il sensore. Vi è riportata sia la versione del

segnale come fornita dal sensore che la versione filtrata di cui verrà data spiegazione successivamente.

Tutte le figure dei vari segnali GSR misurati sono state riportate con delle fasce colorate atte ad indicare i differenti minuti, 1° in verde, 2° in rosso, 3° in azzurro e 4° in giallo, per una più chiara temporizzazione.

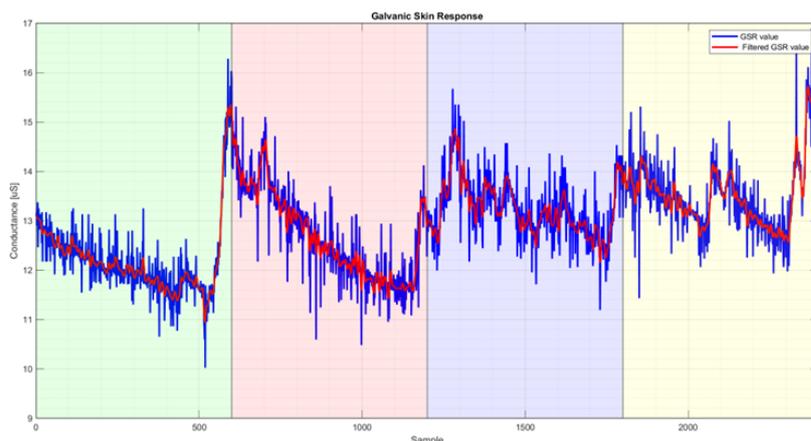


Figura 6.7: GSR filtrato (rosso) e non filtrato (blu).

Sono stati indotti anche degli stati di stress come ad esempio, mantenimento del respiro o dei movimenti bruschi della mano ai quali è seguita una diminuzione della resistenza della pelle, ovvero maggior produzione di sudore.

Si è potuto constatare inoltre come un collegamento degli elettrodi sulla pelle non effettuato a regola d'arte ovvero con un contatto mal riuscito, porti ad avere degli scarsi risultati con la presenza di forti irregolarità del segnale date dalla creazione di condizioni assimilabili a circuiti aperti.

Come anticipato, si ricorda che i tempi di reazione in termini di variazione di conduttanza in seguito ad uno stimolo di tipo emotivo variano in un arco temporale di 1-5 s. Non è possibile quindi rilevare variazione di stress o di qualsiasi stato dell'individuo in modo istantaneo come farebbero ad esempio la rilevazione di altri segnali fisiologici che hanno tempi di risposta di pochi millisecondi. Ad esempio, una reazione visiva del cervello può essere misurata con l'EEG dopo soli 50 ms dall'inizio dell'evento vi-

sivo. Si fa presente che la misura di tali parametri fisiologici ad oggi siano di difficile applicabilità, stante l'ingombro e l'invasività dei sensori necessari per tali misure.

Perciò, nonostante questo tempo di ritardo di qualche secondo nella ipotesi peggiorativa di massima della risposta emozione – sudore, la misura GSR ricade perfettamente nelle specifiche. Inoltre, non si è intenzionati a rilevare variazioni di stati emotivi istantanei come un pericolo lungo la strada dove anche 1 s di ritardo potrebbe essere eccessivo, ma bensì lungo un arco temporale di vari minuti per la determinazione di un eventuale stato di stress.

Anche lo stato di durata del segnale varia in funzione dello stato emotivo, in caso di stress improvviso quale un ostacolo lungo la strada, ad esempio, può essere oggetto di un forte picco di stress con un riassetto allo stato precedente dopo qualche minuto. Nel caso in cui si venga incontro ad uno stato di stress da stanchezza, quest'ultimo non deriva da un singolo episodio ma bensì da un arco temporale durante il quale vi è un segnale rappresentativo della conduttanza che ha una tendenza decrescente indicando quindi una diminuzione dell'attività di sudorazione.

6.2.1 Filtraggio

Il valore della conduttanza della pelle quindi, avendo dei tempi di reazione all'interno di un intervallo tra 1 s e 5 s, ovvero una banda utile di 0.2 Hz – 1 Hz, in sede di elaborazione dati è possibile applicare un filtro passa basso in grado di eliminare tutte le componenti armoniche di rumore aventi frequenza maggiore. Cause di questo rumore normalmente sono l'errore di precisione del sensore GSR, movimenti della mano che causano uno spostamento impercettibile del sensore e rumori elettrici derivanti dalla rete di alimentazione.

In fase di *plot* dei valori ottenuti in MatLab si è implementato un filtro passa basso come riportato di seguito. Si è progettato un filtro di *Butterworth* a partire dalla frequenza di lavoro del sensore pari a 4 Hz. La frequenza di taglio è stata fissata ad 1 Hz, annullando quindi, con le ovvie limitazioni, tutti i segnali con frequenze maggiori.

```

%% Filter Construction
% Define filter parameters
order = 4; % Filter order
cutoff_freq = 1; % Cutoff frequency in Hz

% Design the low-pass Butterworth filter
[b, a] = butter(order, cutoff_freq / (4 / 2)); % 4 Hz is the sampling rate of gsr signal

% Apply the filter to your GSR data
filtered_data = filtfilt(b, a, data);

```

Figura 6.8: Filtro passa-basso implementato in MatLab.

Per maggiore chiarezza è riportato un particolare di un segnale filtrato rispetto ad un segnale non filtrato. Come anche visibile nella Fig. 6.7, l'utilizzo di un filtro permette un notevole *smoothing* del segnale con l'eliminazione di larga parte dei segnali di disturbo sovrapposti. In Fig. 6.9 è riportato un particolare del funzionamento del filtro rispetto al segnale non filtrato.

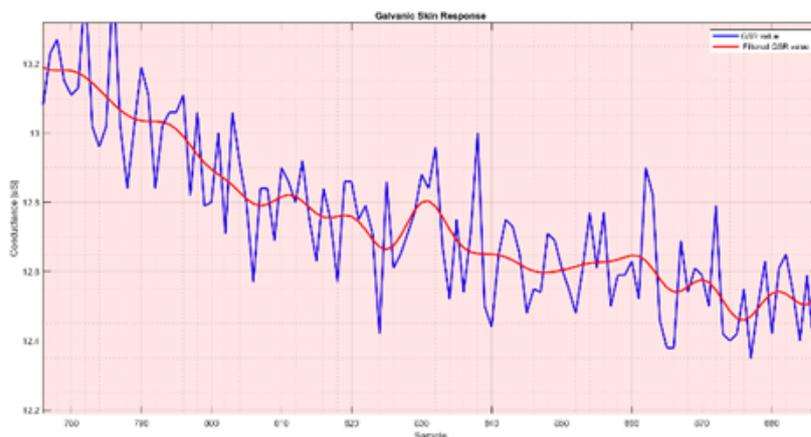


Figura 6.9: Particolare: segnale filtrato (rosso) vs non filtrato (blu).

6.2.2 Acquisizione GSR mediata e filtraggio

Al fine di ottimizzare il segnale prodotto è stato ulteriormente implementato nel codice il calcolo di una media di 10 campioni misurati precedenti (Fig. 6.10) allo scopo di omogenizzare eventuali *spike* di conduttanza. In questo caso è possibile apprezzare uno stato di agitazione iniziale a cui segue uno stato di rilassamento con un aumento

finale della conduttività a causa di un improvviso stato di stress dato che si ipotizza essere dato dall'approssimarsi della conclusione della misura.

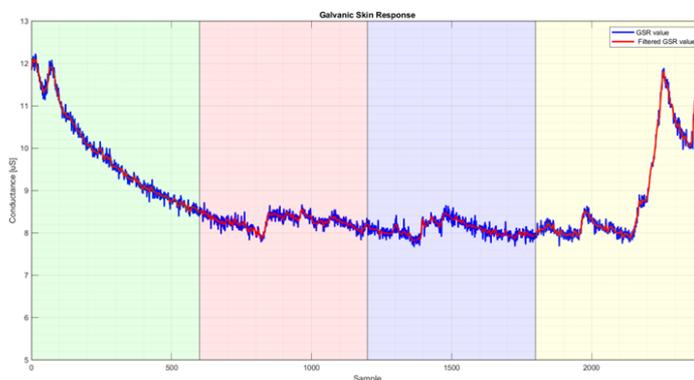


Figura 6.10: GSR filtrato (rosso) e non filtrato (blu) con controllo spike.

Sono state effettuate diverse prove su differenti soggetti e si è constatato che i valori di conduttanza misurati sono diversi in maniera significativa da individuo a individuo, a partire da differenti risposte ai vari tipi di stress ed emozioni fino ad una diversa risposta in termini di sudore dati da una medesima temperatura nel luogo della misurazione. Sono state anche effettuate diverse prove del sensore collegando i due elettrodi non più sulle dita di una sola mano ma sulle dita di due mani diverse ottenendo dei valori di conduttanza mediamente più alti dei casi riportati in precedenza. Ciò deriva ovviamente dal maggior percorso che la corrente deve effettuare per chiudere il circuito.

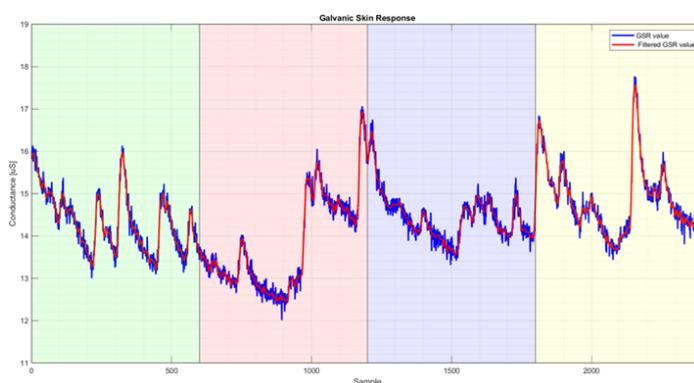


Figura 6.11: Segnale GSR con presa da due mani.

6.3 Empatica E4

L'*Empatica E4* [web14] è un dispositivo elettronico indossabile in grado di rilevare diversi dati fisiologici usato nel caso in essere come “validazione” del prototipo. Per i nostri scopi è stata presa in considerazione solamente la misura GSR. Il dispositivo è in grado di campionare con una frequenza unicamente di 4 Hz con una risoluzione di 1 Digit pari a circa 900 pS, come range di funzionamento è situato all'interno di una banda di valori 0.01 μ S – 100 μ S, gli elettrodi sono fatti in argento. Il dispositivo presenta delle caratteristiche tecniche migliori rispetto al nostro prototipo, pertanto si presta ad essere usato come riferimento.

Esteticamente si presenta come un classico smartwatch senza quadrante, vi è solo un pulsante per la marcatura di un evento durante l'acquisizione. L'istante di tempo contrassegnato può essere ricondotto temporalmente ai segnali fisiologici registrati grazie al *clock* interno. Inoltre, esternamente è presente un led il quale, in funzione della colorazione indica un diverso stato del dispositivo.

Il dispositivo è in grado di lavorare in differenti maniere, la prima è quella in locale o in registrazione, in cui i segnali vengono temporaneamente salvati nella memoria del dispositivo e successivamente mediante l'app proprietaria caricati in *cloud* su di un applicativo sempre della casa madre in cui sono disponibili tutti i dati raccolti sia per essere lavorati in secondo tempo che visualizzati in formato grafico temporale. Altra modalità di acquisizione delle misure è quella streaming, in cui i segnali vengono trasferiti in tempo reale tramite una connessione Bluetooth un dispositivo esterno per la visualizzazione, la registrazione e/o l'analisi in tempo reale, sulla falsa riga di quanto vale per il caso precedente.

Il dispositivo utilizza in maniera massiccia, come anticipato, software proprietari attraverso l'uso di applicazioni *Empatica*. Tutte le misure effettuate possono essere scaricate nel formato CSV per poi essere elaborate attraverso applicazioni di terze parti, come ad esempio MatLab. Particolarità nell'uso dei file .CSV prodotti in totale autonomia dal dispositivo è che contengono nella prima riga il tempo di inizio

dell'acquisizione, nella seconda riga è riportata la frequenza di campionamento, e dalla terza riga in poi il segnale misurato con tutti i valori acquisiti. La raccolta dei dati è avvenuta, sulla falsa riga di quanto fatto in precedenza mediante letture dalla medesima durata dell'acquisizione mediante prototipo. Contemporaneamente all'innescò delle misure dell'*Empatica E4* sono state effettuate le letture del sensore della *Groove Seed*, allo scopo di ottenere anche un adeguato confronto temporale dei dati ottenuti. Entrambe le letture sono state effettuate sul polso della mano non dominante. L'applicativo *Empatica E4* è possibile scaricarlo [web14], ove, previa registrazione, è possibile avere accesso ai dati registrati e scaricarli liberamente.



Figura 6.12: Specifiche del dispositivo Empatica E4.

6.3.1 Confronto misure raccolte

In Fig. 6.13 è riportato un grafico in cui è rappresentato l'andamento della conduttività della pelle misurata sia con il sistema prototipale sviluppato, che con l'*Empatica E4*. Si nota subito che i segnali, nonostante un diverso valore assoluto di conduttività, presentino un comportamento simile con un graduale aumento dei valori a forma di una scaletta. Dal grafico si evidenzia, dato l'aumento di conduttività della pelle, uno stato emotivo che incrementa nel tempo quale ad esempio uno stato di stress. Inoltre, è desumibile dal grafico come, l'andamento del segnale sia pressoché lo stesso, ma il valore effettivo della misura sia di un ordine di grandezza inferiore. A titolo di esempio, al primo istante di lettura con il prototipo è misurata una conduttanza pari a 1.6

μS , valore che scende a circa $0.16 \mu\text{S}$ nel caso delle misure effettuate con *Empatica E4*. Il diverso valore è dato dalla differente corrente che viene iniettata sulla pelle da parte del dispositivo. Il prototipo, infatti, presenta una tensione a vuoto degli elettrodi pari a 1.6 V , mentre l'*Empatica* che è un dispositivo certificato e valido per misurazioni, presenta una tensione alternata agli elettrodi notevolmente inferiore.

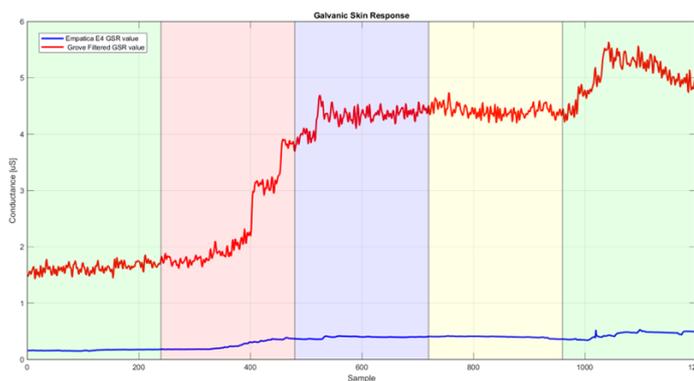


Figura 6.13: GSR Empatica vs GSR Grove.

Sono riportati di seguito dei particolari delle misure effettuate allo scopo di evidenziare ancora meglio la bontà del prototipo sviluppato. Infatti, come sotto riportato, i segnali assumono medesimo andamento, con le ovvie approssimazioni del caso. Quindi, partendo dal presupposto che l'*Empatica E4* è un dispositivo in grado di fornirci un segnale certamente più affidabile e referenziabile di quello del prototipo e considerato i risultati ottenuti, possiamo affermare che il prototipo sviluppato con il sensore della Groove Seed garantisce un buon risultato.

In Fig. 6.14 e 6.15 sono riportati i grafici normalizzati dei due sistemi di acquisizione per apprezzare maggiormente l'andamento dei due.

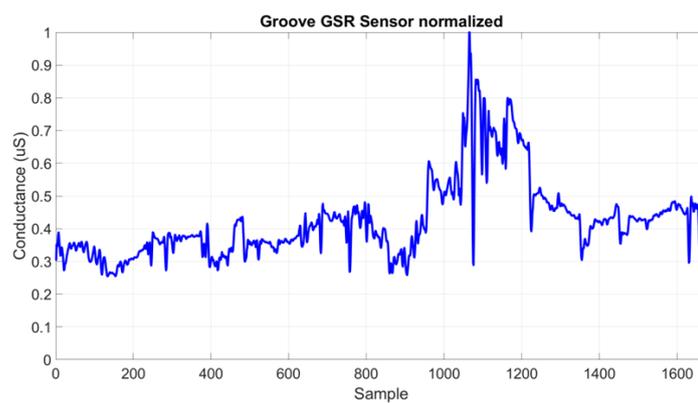


Figura 6.14: GSR Empatica.

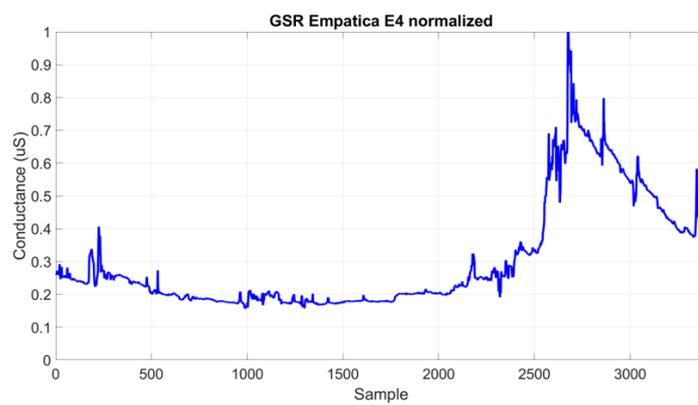


Figura 6.15: GSR Prototipo.

Capitolo 7

Elaborazione Dati Raccolti

A valle della raccolta dei dati effettuata, si passa ora all'analisi di questi ultimi allo scopo di estrapolare delle grandezze e delle informazioni utili. Sono stati presi in considerazione 10 soggetti di differente età e sesso che hanno preso parte ad una simulazione di guida svolgendo tutti sempre il medesimo percorso. Vengono riportati a titolo di esempio solamente i risultati delle elaborazioni dei segnali di 4 dei soggetti coinvolti durante la campagna di misure.

7.1 Ricostruzione segnale GSR

In un primo approccio ai segnali raccolti, è possibile osservare dalle Fig. 7.2, 7.8, 7.13 e 7.19 che il segnale GSR raw non è sempre correttamente acquisito; infatti, sono presenti diverse finestre temporali dove è assente o comunque notevolmente degradato, e ciò deriva dalla criticità già nota della mancanza di contatto tra il palmo delle mani e i due elettrodi durante tutta la fase di guida. Inoltre, il contatto anche solo parziale con gli elettrodi porta ad ottenere un segnale comunque non utile ai nostri scopi. Ciò ha reso necessario l'uso di una elaborazione in grado di considerare il segnale nelle finestre dove quest'ultimo è disponibile.

Viene proposto uno schema di massima della procedura svolta per rendere possibile l'estrazione di caratteristiche significative dal segnale raccolto attraverso una ricostruzione di quest'ultimo.

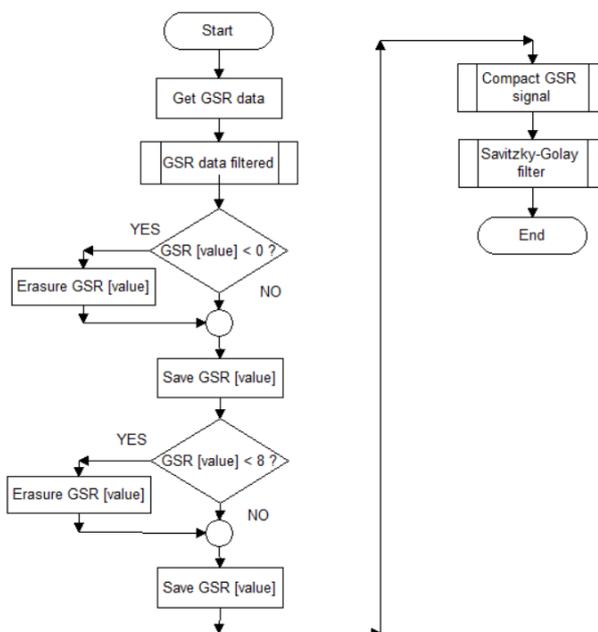


Figura 7.1: Ricostruzione segnale GSR.

In primo luogo, è stata disposta una soglia pari a $0 \mu\text{S}$, così da eliminare tutti i valori di conduttanza inferiori allo zero, che derivano sempre da cattivi contatti fra elettrodo e il palmo. In seguito, è stato disposto un blando filtraggio passa basso, allo scopo di eliminare spike di corrente: tale operazione non è strettamente necessaria in quanto tale problematica era stata superata mediante l'uso della somma di 10 campioni mediati, come riportato nei capitoli precedenti.

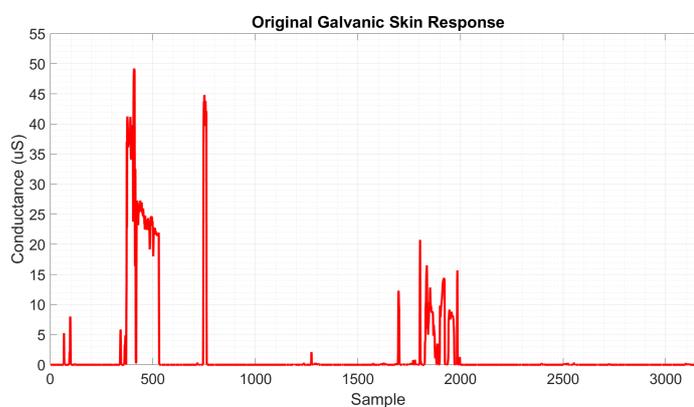


Figura 7.2: Segnale GSR acquisito.

In questa fase quindi, al fine di estrapolare un segnale, si è fissata una soglia a $8 \mu\text{S}$ per poter scartare tutti i valori di segnale GSR inferiori. In questi casi in particolare, siamo certi della presenza di una lettura che non è avvenuta nella maniera corretta e possiamo pertanto procedere a questa operazione forzata di cancellazione. Ciò deriva da una osservazione empirica effettuata sul prototipo, in quanto si è visto che, nelle misure, valori inferiori alla soglia di $8 \mu\text{S}$ derivano dal contatto con solo un elettrodo e non con la coppia, portando quindi a misure inidonee ai nostri scopi. I risultati sono riportati in Fig. 7.3, 7.9, 7.14 e 7.20, ove si è in presenza di un segnale GSR ottenuto accorpendo tutte le finestre ove quest'ultimo era disponibile, a partire dal segnale grezzo originario. La lunghezza di tale vettore è chiaramente dipendente dalla qualità delle misure effettuate a monte, infatti esso presenta lunghezze comprese fra qualche centinaio di campioni fino al migliaio.

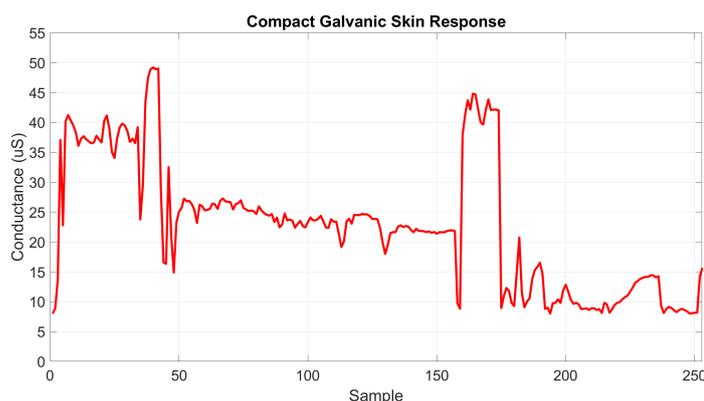


Figura 7.3: Segnale GSR elaborabile.

Segue una fase di filtraggio più importante rispetto alla precedente, mediante l'uso di un filtro *Savitzky-Golay*, in grado di levigare il segnale, come si osserva confrontando Fig. 7.3 e Fig. 7.4. Il motivo principale che ha spinto all'uso di questo filtro è la necessità di eliminare o quanto meno smussare le finestre con falsi segnali "simil dente di sega" del segnale originario, originati dal contatto non corretto palmo-elettrodo.

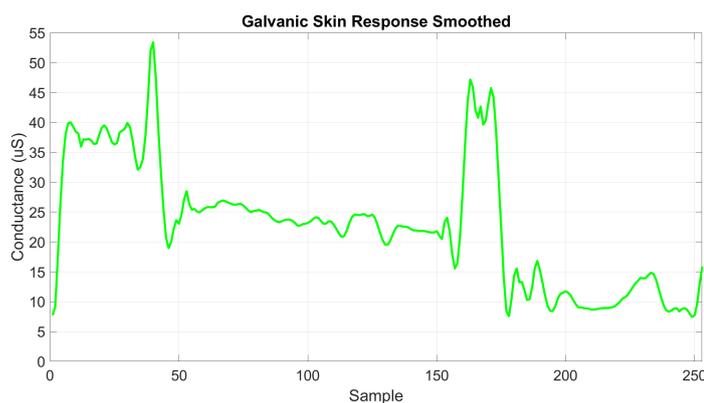


Figura 7.4: Segnale GSR filtrato.

Infine, vi è la fase di estrazione della componente fasica e tonica. Si riporta uno schema di massima delle operazioni svolte che, si ricorda, avvengono secondo una procedura nota e largamente utilizzata in letteratura.

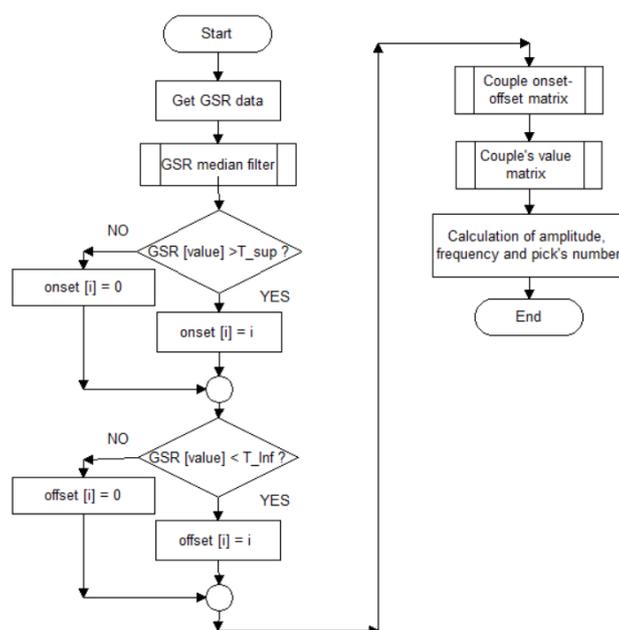


Figura 7.5: Estrazione componente fasica e tonica.

In particolare, per l'estrazione della componente tonica *SCL* si ricorre all'uso di un filtro mediano, che lavora su finestre di 3 s, ovvero 12 campioni. Il segnale *SCR* invece, è definito come la differenza fra il segnale originario e la mediana del segnale oggetto

di finestra precedente. L'uso della finestra in questa fase è stato critico a causa delle precedenti operazioni di filtraggio che hanno aggiunto ritardo al segnale.

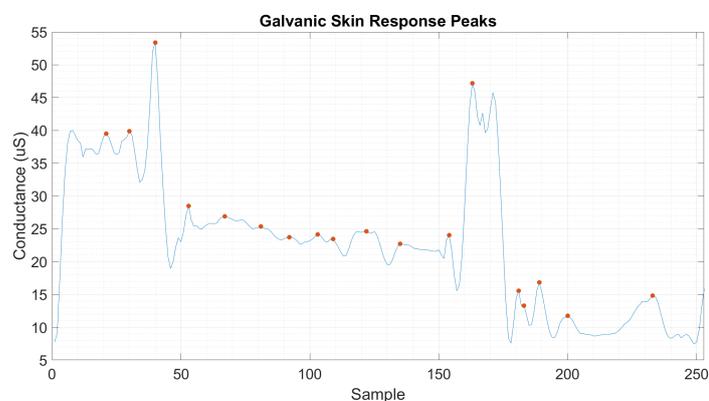


Figura 7.6: Picchi GSR.

Successivamente, per individuare i picchi all'interno del segnale *SCR*, si utilizzano due soglie a $0.05 \mu\text{S}$ e $0 \mu\text{S}$ per la creazione di due vettori detti di onset e di offset. Attraverso dei cicli che scorrono l'intero segnale, è possibile individuare nella componente *SCR* tutti i valori che superano la soglia di $0.05 \mu\text{S}$ e di salvare la posizione di tali valori nel vettore onset, idem dicasi dei valori inferiori $0 \mu\text{S}$ e della relativa posizione di tali valori nel vettore offset. Operando in questa maniera è possibile determinare una coppia onset-offset che indica un intervallo ove è possibile individuare un picco corrispondente ad una stimolazione, attraverso la ricerca di un massimo valore.

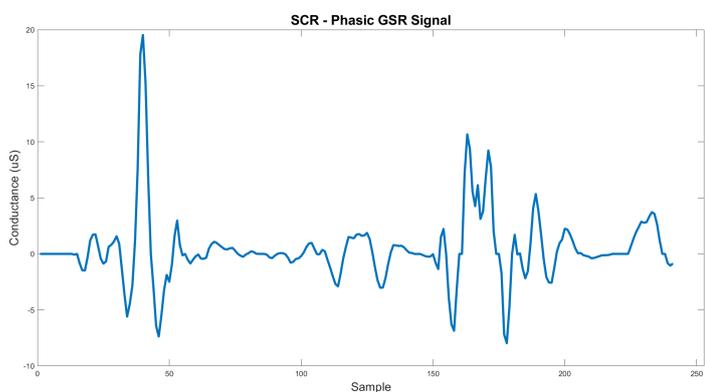


Figura 7.7: Segnale GSR Phasic.

7.1.1 Frequenza dei picchi di risposta galvanica

È stata infine calcolata la frequenza dei picchi del segnale GSR per tutti e 10 i soggetti presi in considerazione, come riportato in Tab. 7.1.1. Tali valori sono conformi al range 0.1 - 0.5 Hz preventivato. Tale grandezza è importante in quanto ci permette di individuare, in presenza di una sua variazione, il verificarsi di uno stato di stress. Si può immaginare infatti di calcolare la frequenza dei picchi di risposta galvanica per un intervallo di tempo e confrontarlo con i precedenti intervalli ed a quel punto appurare eventuali differenze e quindi variazioni nello stato del guidatore. Un aumento della frequenza sarebbe un'indicazione dell'occorrere di uno stato non ottimale.

Tabella 7.1: Frequenza picchi risposta galvanica.

Soggetti										
	n.1	n.2	n.3	n.4	n.5	n.6	n.7	n.8	n.9	n.10
f peak (Hz)	0.285	0.257	0.291	0.274	0.321	0.362	0.312	0.316	0.268	0.358

Ai fini della verifica della *catena della riferibilità*, sono riportati nelle Fig. 7.16 e 7.22 dei confronti tra il segnale GSR correttamente misurato con il dispositivo *Empatica E4* e il segnale estratto a valle di tutto il processo di elaborazione precedentemente descritto. Si evince come l'andamento sia sovrapponibile, confermando pertanto la bontà del segnale estratto.

Vengono riportati nei seguenti sotto paragrafi i dati raccolti nella campagna di misura di ulteriori 3 soggetti a scopo puramente esplicativo.

7.1.2 Dati soggetto 2

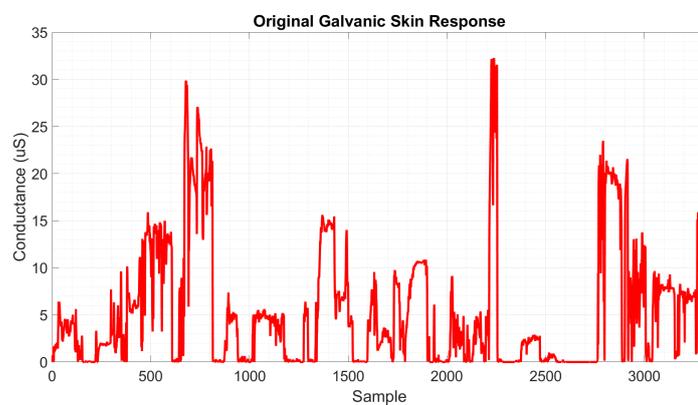


Figura 7.8: Segnale GSR acquisito.

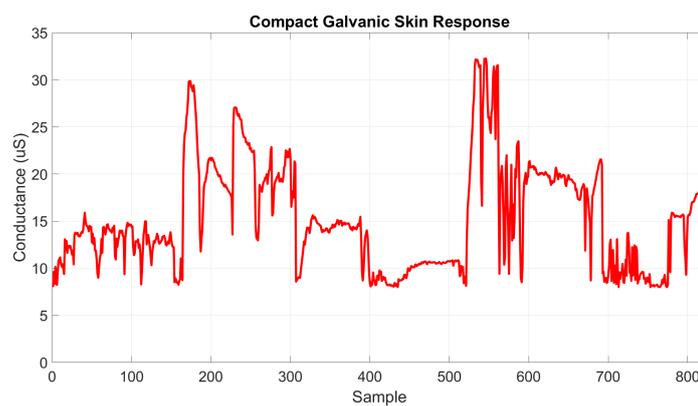


Figura 7.9: Segnale GSR elaborabile.

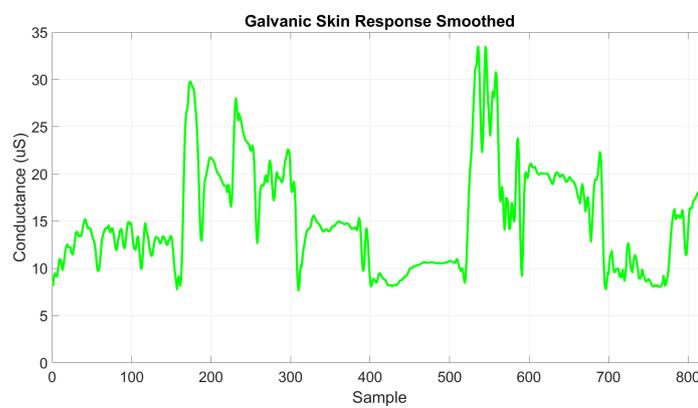


Figura 7.10: Segnale GSR filtrato.

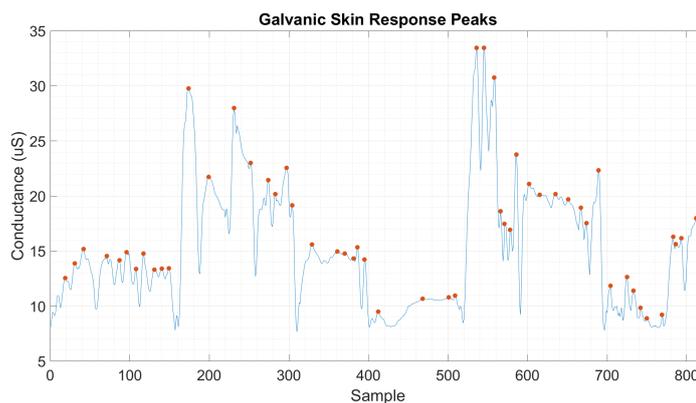


Figura 7.11: Picchi GSR.

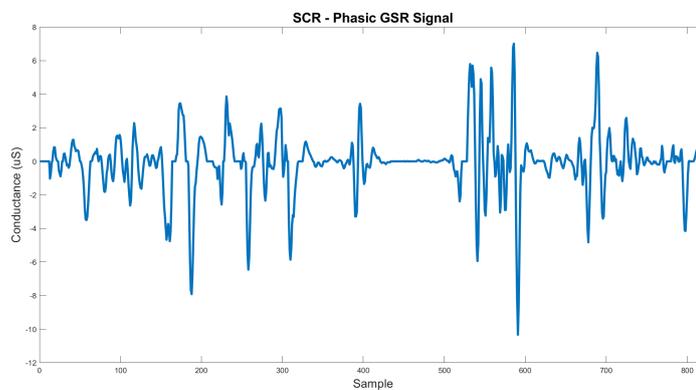


Figura 7.12: Segnale GSR Phasic.

7.1.3 Dati soggetto 3

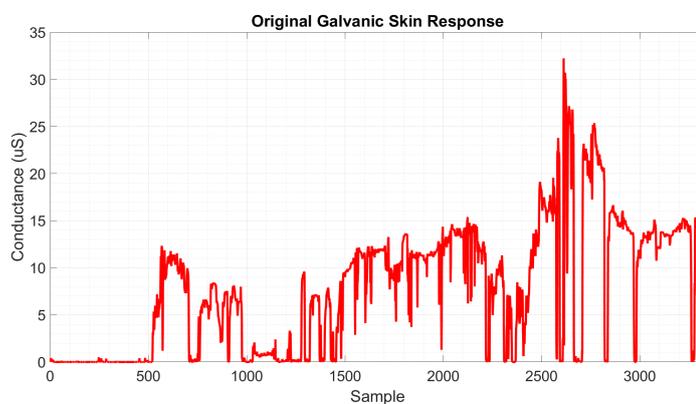


Figura 7.13: Segnale GSR acquisito.

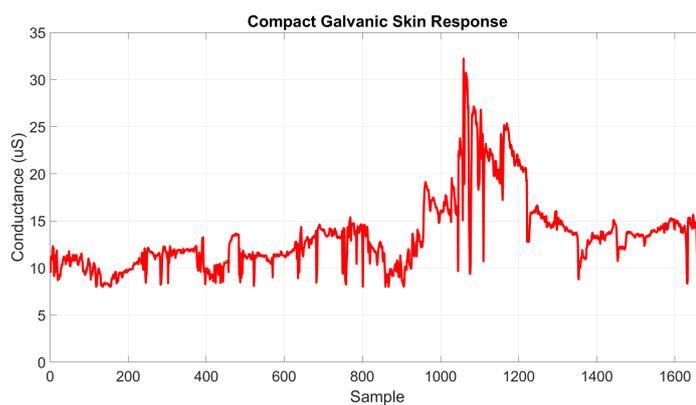


Figura 7.14: Segnale GSR elaborabile.

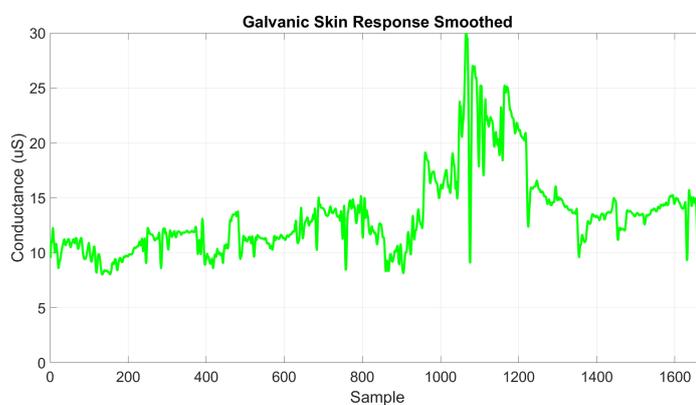


Figura 7.15: Segnale GSR filtrato.

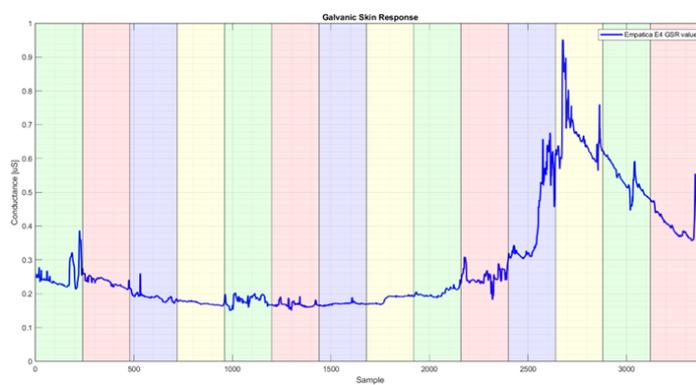


Figura 7.16: Segnale GSR Empatica E4.

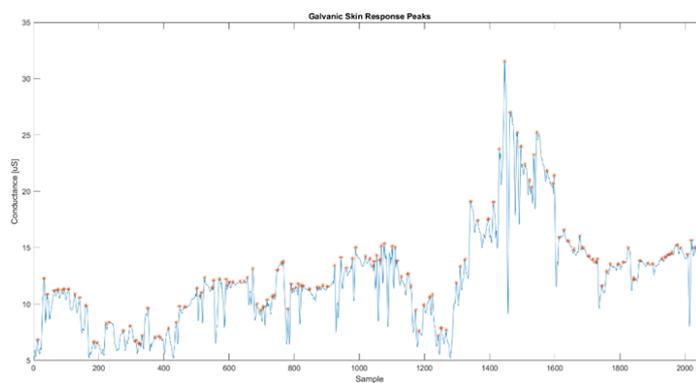


Figura 7.17: Picchi GSR.

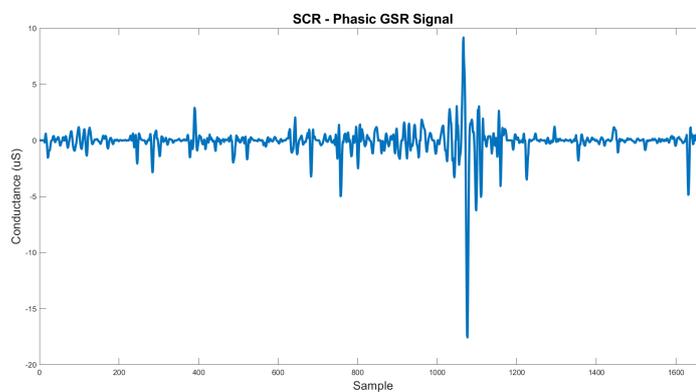


Figura 7.18: Segnale GSR Phasic.

7.1.4 Dati soggetto 4

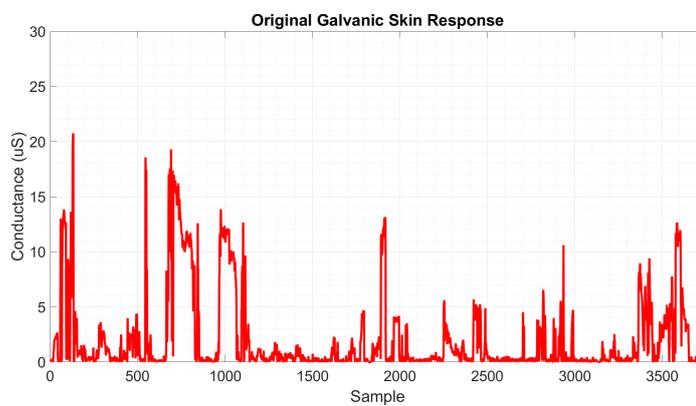


Figura 7.19: Segnale GSR acquisito.

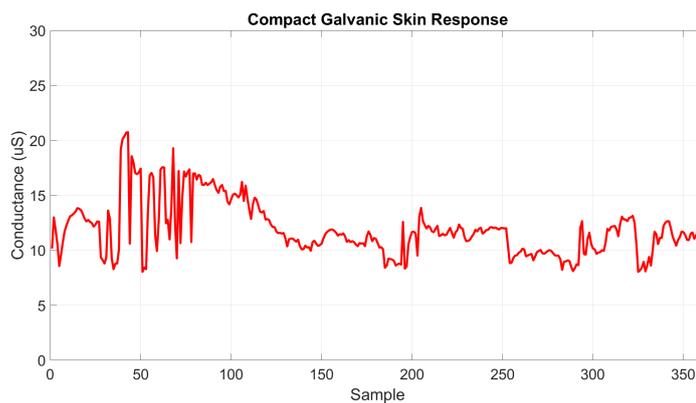


Figura 7.20: Segnale GSR elaborabile.

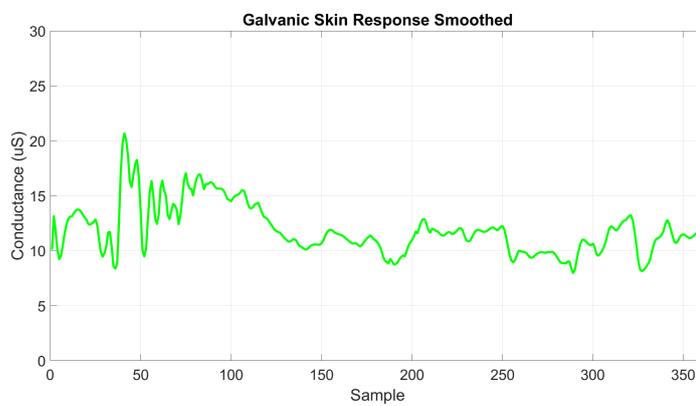


Figura 7.21: Segnale GSR filtrato.

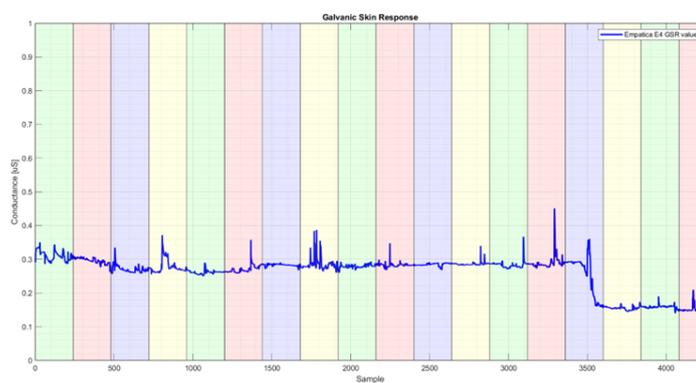


Figura 7.22: Segnale GSR Empatica E4.

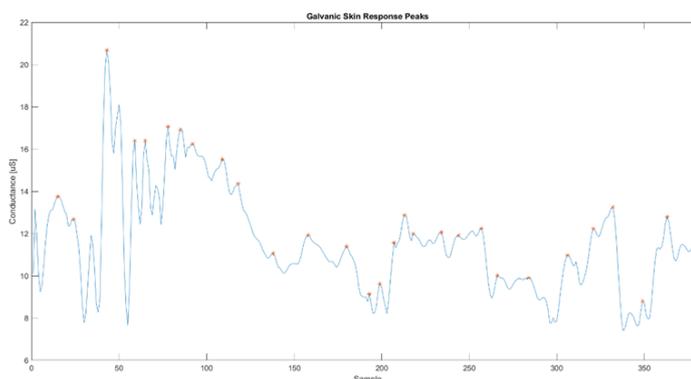


Figura 7.23: Picchi GSR.

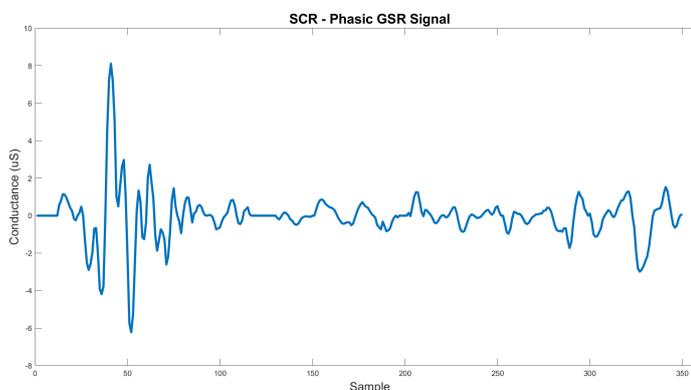


Figura 7.24: Segnale GSR Phasic.

7.2 Confronto FSR – Accelerometro

Sul medesimo prototipo è installato un accelerometro + giroscopio triassiale (la cui trattazione esula dai nostri scopi) con il quale sono state effettuate delle misure durante le simulazioni di guida, mantenendo i sensori di forza analogamente in funzione. Per quanto riguarda l'uso dei sensori di forza, essi lavorano nella medesima maniera largamente descritta in precedenza. In merito all'accelerometro + giroscopio, quest'ultimo è in grado di fornirci i valori delle accelerazioni e delle velocità angolari lungo i 3 assi di riferimento x , y e z . Ovviamente, nel caso in essere, siamo interessati unicamente alle variazioni lungo l'asse z che corrispondono a rotazioni parziali o totali

del volante, ovvero, ad esempio, in prossimità di una curva o di un ostacolo durante la simulazione di guida.

I dati raccolti sono stati opportunamente lavorati via MatLab. In questo caso non è stata necessaria un'operazione di filtraggio in quanto il segnale è sufficientemente "pulito". Tuttavia, si è reso necessario l'uso di un'operazione di *up sample* e *down sample* allo scopo di ottenere una medesima temporizzazione degli eventi fra le due tipologie di sensori, in quanto operanti con frequenze di lavoro diverse.

Il sensore accelerometrico lavora con una frequenza di 10 Hz, viceversa i sensori di forza a 4 Hz. Ciò rende necessario lavorare sui campioni dell'accelerazione procedendo ad un opportuno *up sample* ($10 \text{ sample/s} * 2 = 20 \text{ sample/s}$) a cui segue un *down sample* ($(20 \text{ sample/s}) / 5 = 4 \text{ sample/s}$). Così facendo è possibile ottenere i due segnali campionati negli stessi istanti, e quindi resi confrontabili. È necessario svolgere questa operazione in due passi in quanto MatLab non permette lo svolgimento con un'unica operazione.

Si consideri inoltre che l'accelerometro effettua sempre la lettura di eventuali variazioni lungo l'asse preso in considerazione, i sensori FSR invece, in parte per effetto della loro non continuità lungo la corona del volante, e in parte per una non buona presa della mano sul sensore, presentano comunque dei limiti e non sono in grado di fornirci una lettura di tutti i contatti della mano con il volante.

Di seguito sono riportati dei confronti fra i dati ottenuti dagli FSR e i dati ottenuti osservando le variazioni della velocità angolare sull'asse z , dai quali si è in grado di risalire a come, in occasione di curve o ostacoli nella simulazione di guida, vi sia effettivamente un cambiamento della posizione delle mani. In particolare lo si osserva dai sensori 8 e 9 posizionati nella fascia "alta" del volante. Segue adesso la rappresentazione dell'attivazione dei sensori FSR e della variazione dei segnali letti dall'accelerometro, da cui si è trovato come ci sia un importante legame fra lo stato delle due tipologie di sensori considerati.

Ad esempio, è posto in evidenza come il soggetto nella finestra presa in considerazione

stia svolgendo un percorso misto di curve e rettilinei, stante la sostenuta attività del giroscopio, confermata dall'attivazione di numerosi sensori di forza.

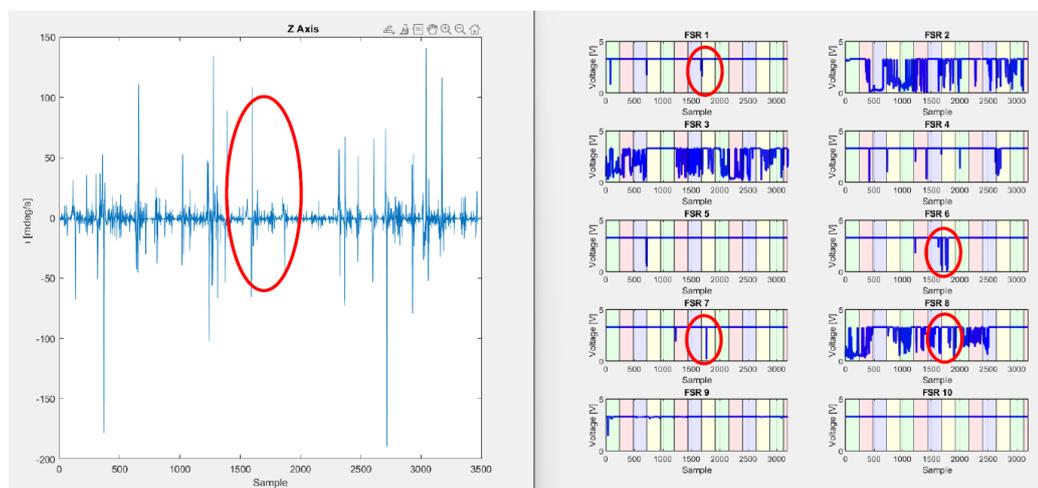


Figura 7.25: Confronto segnale del giroscopio vs FSR soggetto n. 1.

Dal caso che segue possiamo assumere di essere in una situazione di rettilineo (caso in rosso) in quanto il segnale del giroscopio non presenta grandi variazioni e i sensori di forza attivati risultano confermare tale ipotesi. Viene riportato anche il caso di un esempio di rotazione del volante (caso in verde), dove, osservando quali siano i sensori di forza attivati, siamo in grado di attribuire la causa ad una curva.

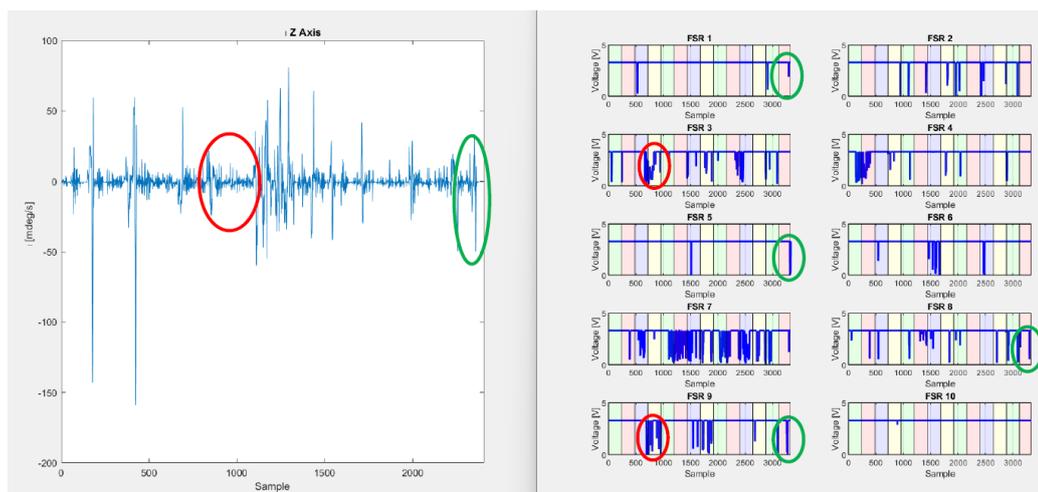


Figura 7.26: Confronto segnale del giroscopio vs FSR soggetto n. 2.

Dalla rappresentazione che segue, possiamo ipotizzare di essere in presenza di una curva od un ostacolo in quanto vengono ad essere attivati i sensori posti nella posizione “alta” del volante, oltre al verificarsi di una notevole attivazione del giroscopio.

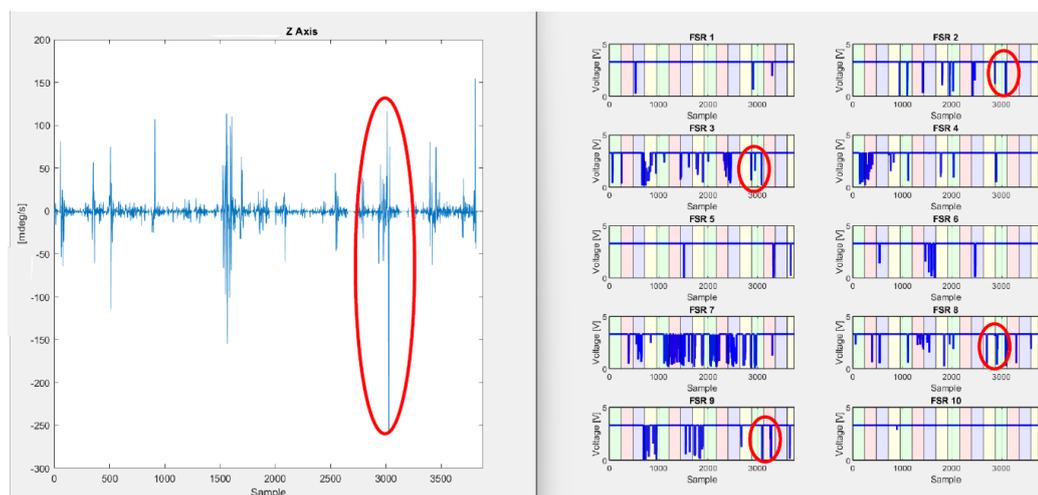


Figura 7.27: Confronto segnale del giroscopio vs FSR soggetto n. 4.

7.3 Frequenza d’uso dei sensori FSR

Viene ora ad essere trattato e calcolato un indice in grado di fornirci, fra i 10 FSR adoperati, una indicazione di quali risultino essere maggiormente attivati durante le simulazioni di guida. Viene calcolato per ognuno dei sensori una sorta di “tasso di attivazione”, il quale è espresso in percentuale ed è rappresentativo del rapporto fra il numero di attivazioni del singolo sensore e il numero totale di letture effettuate, che variano in funzione della durata della simulazione di guida. Al fine di poter considerare valida un’attivazione del sensore sono state assunte solamente le letture di valori inferiori a 3.2 V, in quanto, per via empirica e come discusso in precedenza, in questa ipotesi si è certamente in presenza di un contatto veritiero e non di un tocco involontario.

Da un’analisi dei dati raccolti è interessante notare come il tasso di attivazione sia maggiore per taluni sensori quali il n. 2 e 3 ed il n. 7 ed 8, che corrispondono alla classica posizione di guida. I sensori posti nella parte alta e bassa rispettivamente del

volante invece non presentano attivazioni degne di nota. A titolo di esempio, i sensori n. 1 e 10 risultano praticamente mai attivi, come lo sono specularmente i n. 5 e 6.

I dati riportati in Tab. 7.3 non sono da intendersi come attivazione del sensore in termini assoluti, bensì nei termini relativi, in quanto, in funzione della posizione delle mani assunta durante la guida, è possibile attivare più sensori simultaneamente. A titolo puramente esemplificativo, nel soggetto n. 1 il sensore n. 3 è stato attivo durante il 46.59% della misurazione o dicasi altresì della durata del tempo di guida. Si può inoltre affermare che data la forte attivazione dei sensori n. 3, 4, 7 ed 8, il percorso presentava diversi tratti rettilinei, e ciò deriva anche dalla osservazione dell'ovvia posizione che le mani dovrebbero assumere in caso di curve rilevanti od ostacoli che porterebbero ad un uso più importante dei sensori posti nella fascia bassa e alta del volante. Si tenga presente che anche durante l'esperienza di guida reale la durata temporale in cui si svolta (per compiere una curva) è sempre notevolmente inferiore al tempo in cui ci si trova a guidare lungo dei rettilinei. Infine, si vuole rimarcare come non vi sia un unico sensore che risulta preminente ma bensì un insieme, desumendo come nonostante la forte componente soggettiva della posizione assunta dalle mani durante la guida, si è in grado di individuare i sensori maggiormente attivati.

Tabella 7.2: Tasso di attivazione sensori FSR (%).

Soggetti										
	n.1	n.2	n.3	n.4	n.5	n.6	n.7	n.8	n.9	n.10
FSR1	0.21	1.86	0	0.34	0	0.13	0	0	0	0
FSR2	32.02	7.62	2.67	1.44	11.15	22.02	0.38	10.04	0.27	3.55
FSR3	46.59	12.87	69.27	7.23	23.88	29.55	34.03	27.37	76.60	19.17
FSR4	1.28	11.42	1.17	5.44	1.19	2.18	1.08	0.57	0.34	1.09
FSR5	0.21	0.06	0	0.42	1	0.6	0	0	0	0
FSR6	0.75	0.57	0.21	1.46	0.48	0.53	0	0	0.13	0.15
FSR7	0.09	20.65	0	17.85	1.25	17.24	0	0.11	0	4.38
FSR8	34.74	3.79	19.20	1.49	7.45	0.23	13.12	20.06	0.61	4.10
FSR9	1.09	1.05	0.75	5.07	11.58	0	0	0.74	0.82	0.15
FSR10	0	0.03	0	0.02	0.32	0.70	0	0.25	0	0

Capitolo 8

Conclusioni

L'obiettivo principe del lavoro di tesi era quello di sviluppare un prototipo in grado di rilevare stati di stress occorsi durante la guida. Viene adesso ad essere ricapitolato ed in parte esteso quanto presentato largamente durante il lavoro, esponendo le criticità riscontrate e anche fornendo degli spunti per possibili sviluppi futuri.

8.1 Prototipo sviluppato

Il prototipo sviluppato è in grado di rilevare stati di stress del guidatore, come ampiamente illustrato nel capitolo precedente. Nonostante l'utilizzo di due sensori che afferiscono a differenti grandezze fisiche, quali forza e conduttanza, è possibile studiare lo stato del guidatore in modo omogeneo. Per quanto concerne la misura della conduttanza della pelle, si è derivata una metrica molto importante come la frequenza di ripetizione dei picchi di risposta galvanica, la cui variazione assume un significato molto importante. Nello studio dei sensori di forza invece è stato sviluppato un indice definito tasso di occupazione che riesce a segnalare l'attività effettiva dei singoli sensori e quindi di uno specifico settore del volante. Nel caso in cui non risultasse rilevata alcuna attivazione dei sensori di forza, sarebbe un segno evidente che non si stanno tenendo le mani sul volante.

Come precedentemente discusso, lo stress di cui si cercano evidenze è quello risultante ad esempio da una durata medio - lunga della sessione di guida, pertanto si può

immaginare di ricavare la frequenza di ripetizione ad intervalli di tempo regolari tra i quali poi effettuare un confronto per verificarne l'andamento. Un andamento crescente di tale frequenza indicherebbe l'occorrere di uno stato psicofisico potenzialmente rischioso per chi è alla guida.

Le criticità riscontrate nella misura della conduttanza della pelle sono state superate dalla successiva elaborazione che permette di recuperare almeno in parte il segnale originario. L'uso dei sensori di forza è stato relativamente più agevole data la loro naturale robustezza a interferenze esterne. Da essi si è ricavato un parametro che ne descrive la frequenza di attivazione. Contestualizzata al caso della guida, una sua importante variazione in funzione delle finestre di lettura precedentemente definite potrebbe indicare una condizione di rilassamento o di stress del guidatore. Si immagini un incremento della frequenza d'uso dei sensori nelle parti basse del volante (n. 5 e n. 6) e una diminuzione della frequenza d'uso dei sensori nella parte alta (n. 3 e n. 8): questo potrebbe essere interpretato come chiaro sintomo di stanchezza, rappresentativo della tendenza a portare le mani verso la parte inferiore del volante.



Figura 8.1: Setup sperimentale.



Figura 8.2: Particolare prototipo sviluppato.

8.2 Criticità

Il prototipo non presenta criticità importanti dovute ai sensori in senso stretto, quanto piuttosto alla natura di questi ultimi e alla loro modalità di utilizzo. I sensori di forza,

come largamente spiegato, non sono attualmente in grado di coprire l'intera corona del volante, lasciando dei segmenti scoperti. Si consideri inoltre che i sensori sono estremamente sensibili e anche se non correttamente premuti sono in grado di fornire valori utili. La famiglia dei sensori FSR 40x comunque dispone di numerose varianti di sensori di forza, pertanto è agevolmente possibile diminuire le lacune in termini di copertura del volante usando un FSR 408.

L'uso dei sensori di conduttanza mostra la problematica relativa alla impossibilità di garantire un contatto continuativo pelle-elettrodo, portando ad ottenere un segnale deteriorato che necessita di una ricostruzione importante a valle. Anche questa problematica può essere superata andando a considerare letture che vengono effettuate su un tempo medio-lungo, rendendo quindi difficile ipotizzare la mancanza di un contatto pelle-elettrodo di durata sufficiente, da cui estrapolare dati utili ai nostri scopi. Tale ipotesi inoltre è conforme ai nostri limiti in quanto lo stress subentra in caso di affaticamento e stanchezza nel tempo, non in modo immediato. In aggiunta si deve tenere conto del problema relativo all'ossidazione degli elettrodi.

8.3 Sviluppi futuri

Un primo aspetto di possibile sviluppo futuro sicuramente sarebbe quello di implementare all'interno del prototipo ulteriori sensori, così da coadiuvare quelli già presenti e permettere una migliore discriminazione di uno stato di stress. Alcune tipologie di sensori utili possono essere chiamati in causa in virtù di quanto detto nel Capitolo 1. A titolo puramente di esempio, un sensore PPG che necessita di un contatto con la pelle continuativo, sulla falsariga di quanto affermato per il sensore GSR, troverebbe condizioni utili per essere attuato in questo prototipo in quanto tale criticità è stata ampiamente superata. Sempre dal paniere dello Stato dell'Arte, per quanto riguarda tecniche non invasive, si può suggerire l'uso di radar per la determinazione della frequenza respiratoria, o di tecniche di *machine learning* per la determinazione di stati di sonnolenza a partire dalla osservazione del volto. Si sconsiglia di adottare sensori

che siano invasivi e quindi il cui uso sia sconveniente durante la guida.

Sarebbe inoltre possibile pensare alla durata effettiva delle finestrate e a quanto effettivamente si venga a modificare la frequenza di ripetizione dei picchi di risposta galvanica e del tasso di risposta dei sensori FSR, per simulazioni di svariate ore di durata, in ambienti di guida simulata spinti quali guide notturne e in presenza traffico pesante.

Bibliografia

- [1] Branko Babusiak et al., “Design of smart steering wheel for unobtrusive health and drowsiness monitoring.” *Sensors*, vol. 21, no. 16, 2021.
- [2] Joana M. Warnecke et al., “Printed and flexible ecg electrodes attached to the steering wheel for continuous health monitoring during driving.” *Sensors*, vol. 22, no. 11, 2022.
- [3] Jaspreet Singh Bajaj et al., “System and method for driver drowsiness detection using behavioral and sensor-based physiological measures.” *Sensors*, vol. 23, no. 3, 2023.
- [4] Chuwei Ye et al., “Smart steering sleeve (s3): A non-intrusive and integrative sensing platform for driver physiological monitoring.” *Sensors*, vol. 22, no. 19, 2022.
- [5] Oluwaseyi Elizabeth Shodipe et al., “Modelling the relationship between the objective measures of car sickness.” *2023 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2023.
- [6] Andrea Amidei et al., “Unobtrusive multimodal monitoring of physiological signals for driver state analysis.” *IEEE Sensors Journal*, vol. 1, p. 1, 2020.
- [7] S. Pravinth Raja et al., “Smart steering wheel for improving driver’s safety using internet of things.” *SN Computer Science*, vol. 4, no. 277, 2023.
- [8] YouJun Choi et al., “Smart steering wheel system for driver’s emergency situation using physiological sensors and smart phone.” *2014 IEEE Internatio-*

nal Symposium on Innovations in Intelligent Systems and Applications (INISTA) Proceedings, 2014.

- [9] Laavanya Rachakonda et al., “Smart-steering: An iomt-device to monitor blood alcohol concentration using physiological signals.” *IEEE International Conference on Consumer Electronics (ICCE)*, 2020.
- [10] Safie S. I. et al., “Raspberry Pi Based Driver Drowsiness Detection System Using Convolutional Neural Network (CNN).” *2022 IEEE 18th International Colloquium on Signal Processing & Applications (CSPA)*, 2022.
- [11] Yutaro Ono et al., “Smart steering wheel with swept frequency capacitive sensing.” *IEICE transactions on electronics*, vol. 1, pp. 486–489, 2016.
- [12] Andrea Amidei et al., “ANGELS - Smart Steering Wheel for Driver Safety.” *2023 9th International Workshop on Advances in Sensors and Interfaces (IWASI)*, 2023.
- [13] —, “Validating photoplethysmography (ppg) data for driver drowsiness detection.” *2021 IEEE International Workshop on Metrology for Automotive (MetroAutomotive)*, 2021.
- [14] —, “Driver drowsiness detection: A machine learning approach on skin conductance.” *Sensors*, vol. 23, no. 8, 2023.
- [15] Rui Li et al., “A method for fatigue detection based on driver’s steering wheel grip.” *Industrial Ergonomics*, vol. 82, p. 103083, 2021.
- [16] Adil Ali Saleem et al., “A systematic review of physiological signals based driver drowsiness detection systems.” *Cognitive Neurodynamics*, vol. 17, p. 1229 1259, 2023.

Sitografia

- [web1] “Moku:Go,” URL: <https://www.liquidinstruments.com/products/hardware-platforms/mokugo/>.
- [web2] “Raspberry Pi 4,” URL: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>.
- [web3] “Installazione Sistema Operativo Raspbian,” URL: <https://www.raspberrypi.com/documentation/computers/getting-started.html>.
- [web4] “Compilatore Geany,” URL: <https://www.geany.org/>.
- [web5] “Convertitore ADC ADS 1115,” URL: <https://learn.adafruit.com/raspberry-pi-analog-to-digital-converters/ads1015-slash-ads1115>.
- [web6] “Thonny,” URL: <https://thonny.org/>.
- [web7] “Raspberry Pi Pico,” URL: <https://projects.raspberrypi.org/en/projects/getting-started-with-the-pico/3>.
- [web8] “Libreria pyCirclize,” URL: <https://github.com/moshi4/pyCirclize>.
- [web9] “GSR,” URL: https://web.uniroma1.it/lab_nsi/labnsi/tecnologie/risposta-galvanica-della-pelle.
- [web10] “GSR emotion,” URL: https://connect.tobii.com/s/article/galvanic-skin-response-gsr?language=en_US.
- [web11] “Groove GSR Sensors,” URL: https://wiki.seeedstudio.com/Grove-GSR_Sensor/.

- [web12] “Attività elettrodermica,” URL: https://it.wikipedia.org/wiki/Attivit%C3%A0_elettrodermica.
- [web13] “Savemat Instruction,” URL: <https://scipy.org/install/>.
- [web14] “Empatica E4,” URL: <https://support.empatica.com/hc/en-us/articles/206373545-Download-and-install-the-E4-manager-on-your-Windows-computer>.
- [web15] “FSR 402,” URL: <https://cdn.sparkfun.com/assets/8/a/1/2/0/2010-10-26-DataSheet-FSR402-Layout2.pdf>.
- [web16] “ADS 1115,” URL: <https://www.ti.com/lit/ds/symlink/ads1115.pdf>.