



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DIPARTIMENTO DI INGEGNERIA INDUSTRIALE

- Sede di Forlì -

**CORSO DI LAUREA IN INGEGNERIA
AEROSPAZIALE**

Classe L9

**Sviluppo di un sistema di misura
della temperatura istantanea per il
controllo attivo di un flusso
turbolento tramite getto pulsato a
parete.**

Tesi di Laurea in Fluidodinamica ING-IND/06

Relatore

Prof. Gabriele Bellani

Presentata da

Maretti Matilde

Correlatore

Lorenzo Magnani

Dicembre 2025

Anno Accademico 2024/2025

Abstract

Il controllo termico nei flussi aerodinamici è un tema di crescente interesse per le sue applicazioni industriali e scientifiche, che spaziano dal raffreddamento dei motori alla gestione termica dei sistemi elettronici.

Tra le tecniche più promettenti si collocano i getti pulsati in configurazione *Jet in Cross-Flow* (JICF), in grado di perturbare attivamente lo strato limite e potenziare lo scambio termico convettivo. Affinchè tale approccio risulti efficace, è però necessario monitorare con precisione parametri chiave come pressione, frequenza di pulsazione, duty cycle e, soprattutto, la temperatura del getto.

Il presente lavoro di tesi ha avuto come obbiettivo lo sviluppo di un sistema di misura compatto e affidabile per il monitoraggio della temperatura del getto pulsato. L'attività ha previsto la scelta di un tipo di sensore di temperatura, la calibrazione, l'implementazione di un sistema di acquisizione stand-alone basato su Arduino e la progettazione di un'interfaccia per l'inserimento del sensore nel canale in pressione.

Le verifiche sperimentali hanno confermato l'efficacia del sistema, dimostrando la possibilità di disporre di uno strumento semplice, economico e replicabile per misure in tempo reale, utile a valutare l'effetto del JIFC.

Ringraziamenti

C'è un paradosso che ho sempre trovato particolarmente affascinante: quello della nave di Teseo. La storia racconta che la grande nave in legno su cui navigò l'eroe fosse stata custodita con cura dagli abitanti di Atene. Ogni volta che una trave si consumava o una vela si logorava, gli Ateniesi la sostituivano con pazienza, proteggendola dal trascorrere degli anni. Passò così tanto tempo che un giorno, pur mantenendo la stessa forma e lo stesso nome, la nave non conservava più alcun pezzo dell'originale. Fu allora che qualcuno si domandò se quella fosse ancora la nave di Teseo o se, ormai cambiata in ogni sua parte, fosse diventata un'altra imbarcazione.

Forse, alla fine, la risposta conta meno di quanto sembri. Perché il valore di quella nave non stava nelle sue tavole o nei suoi chiodi, ma nelle persone che l'hanno amata, custodita e tramandata nel tempo.

Così è stato anche per me. Nonostante i cambiamenti e le inversioni di rotta, ciò che davvero ha dato forma al mio percorso sono state le persone che mi hanno accompagnata e sostenuta: i miei genitori, mio fratello, mia sorella e tutta la mia famiglia; il mio ragazzo; gli amici di sempre e quelli incontrati sul campo da pallavolo, alle manifestazioni o tra i banchi dell'università.

E, infine, chi mi ha guidata in quest'ultimo viaggio al laboratorio CICLoPE: Lorenzo Magnani e il professor Gabriele Bellani.

A tutti loro va il mio grazie più sincero.

Indice

Elenco delle figure	x
Elenco delle tabelle	xii
Nomenclatura	xiv
1 Introduzione	1
2 Teoria di base	3
2.1 Fondamenti di fluidodinamica	3
2.2 Scambio termico e convezione	5
2.3 Getti pulsati e controllo attivo del flusso	6
3 Contesto Sperimentale	7
3.1 Laboratorio CICLoPE	7
3.2 Il setup	8
3.3 Scelta del sensore di temperatura	11
3.3.1 Termocoppie	12
3.3.2 Termoresistenze	13
3.3.3 Cold-wire thermometry	13
3.3.4 Termistori	14
3.4 Elemento di connessione	17
3.5 Arduino, circuito e codice	18
4 Calibrazione del termistore	20
4.1 Modelli di interpolazione per i termistori NTC	21
4.2 Calibrazione contemporanea di tre termistori	23
4.3 Risultati della calibrazione	24
5 Risultati sperimentali e discussione	26
5.1 Effetti di frequenza e duty cycle della valvola	27

5.2	Influenza dell'orientazione del sensore	29
5.3	Analisi dell'intrusività tramite anemometria a filo caldo	31
6	Conclusioni	33
A	Codice MATLAB per calcolare i coefficienti di calibrazione	35
B	Codice Arduino per l'acquisizione delle temperature	39
	Bibliografia	45

Elenco delle figure

2.1	Strato limite laminare, transitorio e turbolento.	4
3.1	Struttura del Long Pipe del laboratorio CICLoPE	7
3.2	Schema dell'impianto sperimentale.	9
3.3	Setup sperimentale per la misura della temperatura: (a) vista dell'impianto senza sensore, (b) dettaglio dei componenti per la misura della temperatura.	10
3.4	Particolari del setup: (a) Giunto a T con sensore alloggiato, (b) sistema di acquisizione ed elaborazione ddei dati.	11
3.5	Schema circuitale di una termocoppia.	12
3.6	Esempio di termoresistenza in platino a 4 fili.	13
3.7	Struttura cold-wire.	14
3.8	Andamento di vari sensori termoresistivi metallici e semiconduttori, da <i>Handbook of Modern Sensors</i> , Fraden [2].	15
3.9	Esempi di termistori.	15
3.10	Termistore utilizzato nel presente setup.	16
3.11	Componente di connessione del termistore al giunto a T.	17
3.12	Termistore alloggiato nel componente di connessione.	17
3.13	Rappresentazione dello schema elettrico del partitore di tensione.	18
3.14	Rappresentazione delle connessioni ad Arduino.	19
4.1	Curva resistenza-temperatura dei modelli Beta, Fraden e Steinhart-Hart.	22
4.2	Bagno termostatico utilizzato per la calibrazione.	23
4.3	Curva resistenza-temperatura dei tre sensori calibrati.	24
5.1	Disposizioni del termistore.	27
5.2	Confronto dei segnali di temperatura a diverse frequenze e $DC = 75\%$: (a) disposizione laterale, (b) disposizione frontale.	28
5.3	Confronto dei segnali di temperatura a diverse frequenze e $DC = 50\%$: (a) disposizione laterale, (b) disposizione frontale.	28

5.4	Confronto dei segnali di temperatura a diverse frequenze e $DC = 25\%$: (a) disposizione laterale, (b) disposizione frontale.	29
5.5	Posizionamento dell'hot-wire in uscita dal condotto di 4 mm per l'analisi di intrusività del sensore.	31
5.6	Confronto velocità in uscita dal condotto nelle tre disposizioni del termistore, segnali grezzi e mediati.	31
5.7	Confronto degli spettri di velocità nelle tre disposizioni del termistore.	32

Elenco delle tabelle

3.1	Caratteristiche principali del Long Pipe.	8
3.2	Caratteristiche principali del setup sperimentale.	9
3.3	Caratteristiche principali dei termistori NTC.	16
4.1	Caratteristiche principali del bagno termostatico utilizzato per la calibrazione.	23
4.2	Coefficienti di calibrazione Steinhart–Hart dei termistori NTC. . . .	24
5.1	Parametri sperimentali per le misurazioni di temperatura.	26
5.2	Parametri per le prove di valutazione dell'intrusività mediante hot–wire.	27
5.3	Valori medi di temperatura per ciascuna combinazione di frequenza, duty cycle e configurazione del sensore.	30
5.4	Velocità media misurata con Hot-Wire all'uscita del condotto	32

Nomenclatura

JICF	Jet In Cross-Flow
NTC	Negative Temperature Coefficient
PTC	Positive Temperature Coefficient
CFD	Computational Fluid Dynamics
Re	Numero di Reynolds
Nu	Numero di Nusselt
Pr	Numero di Prandtl
CICLoPE	Center for International Cooperation in LOng-Pipe Experiments
PCB	Printed Circuit Board
DAQ	Data Acquisition System
ADC	Analog to Digital Converter
RTC	Real Time Clock
NTP	Network Time Protocol

Capitolo 1

Introduzione

Il controllo dello scambio termico nei flussi aerodinamici rappresenta una sfida di grande attualità, con ricadute dirette in ambito industriale, energetico e aerospaziale. In numerosi settori applicativi, dal raffreddamento dei motori a combustione interna alla gestione termica dei sistemi elettronici, la capacità di ottimizzare lo scambio convettivo tra fluido e parete costituisce un fattore cruciale per migliorare l'efficienza e l'affidabilità dei dispositivi. La ricerca scientifica ha dimostrato come l'impiego di tecniche di controllo attivo possa incrementare significativamente l'efficacia dei processi convettivi, superando i limiti delle tradizionali soluzioni passive.

Tra le strategie più promettenti figura l'utilizzo dei getti pulsati in configurazione *Jet in Cross-Flow* (JICF), in cui un getto viene iniettato in modo perpendicolare al flusso principale. Questa configurazione permette di perturbare attivamente lo strato limite, generando strutture vorticosi che incrementano il trasporto di calore. Tuttavia, per valutarne l'efficacia è necessario un accurato monitoraggio dei parametri di controllo quali pressione, frequenza di pulsazione, duty cycle e temperatura del getto, quest'ultima particolarmente critica, in quanto direttamente legata all'effetto termico del sistema.

In tale contesto si inserisce il presente lavoro di tesi, volto allo sviluppo di un sistema di misura puntuale della temperatura del getto pulsato. Tra le diverse tecnologie disponibili, l'attenzione è stata rivolta al termistore con *negative temperature coefficient* (NTC), un sensore caratterizzato da dimensioni ridotte, che limitano l'interferenza con il flusso, tempi di risposta rapidi, buona stabilità, accuratezza e costo contenuto. Tali caratteristiche lo rendono particolarmente adatto alle condizioni operative previste e costituiscono il presupposto per la realizzazione di un sistema di acquisizione affidabile e facilmente implementabile.

Il percorso di tesi si è articolato in più fasi. In primo luogo, sono stati calibrati contemporaneamente tre termistori mediante un bagno termostatico, utilizzando un sistema di acquisizione e di un codice realizzati appositamente. L'elaborazione dei dati ha permesso di ricavare le curve caratteristiche dei sensori, indispensabili per convertire la resistenza elettrica in temperatura.

Successivamente, l'attenzione si è concentrata sull'integrazione del sensore nel setup sperimentale, con la realizzazione della catena di acquisizione della misura all'interno del canale in pressione. Questo ha richiesto sia la scrittura del codice di acquisizione finale, sia la progettazione di un componente di interfaccia meccanico-pneumatica, in grado di alloggiare il termistore, intercambiabile tra i tre calibrati, e connetterlo al sistema senza alterare il profilo del flusso.

Completata la realizzazione del sistema di misura, sono stati eseguiti i primi test acquisendo la temperatura del getto pulsato a diverse frequenze, duty cycle e posizioni del sensore, così da osservare e confrontare il comportamento nei vari casi.

L'obiettivo raggiunto è la realizzazione di un sistema compatto, sensibile e rapido, con il quale è stato possibile ottenere misure affidabili della temperatura del getto pulsato in configurazione JICF, permettendo di valutare gli effetti dell'attuazione della valvola.

Capitolo 2

Teoria di base

2.1 Fondamenti di fluidodinamica

La fluidodinamica è la branca della meccanica dei fluidi che studia il comportamento di liquidi e gas in movimento, nonché gli effetti che il loro moto esercita sui corpi circostanti o immersi. Essa rappresenta una disciplina fondamentale dell'ingegneria, in quanto fornisce gli strumenti teorico-matematici per descrivere e prevedere il comportamento di un fluido in diverse condizioni, dal moto del sangue nei vasi sanguigni fino all'aerodinamica di un velivolo.

Le leggi fondamentali che governano il moto dei fluidi sono espresse dalle *equazioni di bilancio*, o *leggi di conservazione*, che derivano direttamente dai principi della meccanica classica e in forma compatta, possono essere espresse come segue:

- **Equazione di continuità (conservazione della massa):**

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{V}) = 0 \quad (2.1)$$

- **Equazione di bilancio della quantità di moto:**

$$\rho \left(\frac{\partial \mathbf{V}}{\partial t} + \mathbf{V} \cdot \nabla \mathbf{V} \right) = -\nabla p + \rho \mathbf{f} + \nabla \cdot \boldsymbol{\tau} \quad (2.2)$$

- **Equazione dell'energia :**

$$\rho \left(\frac{\partial e}{\partial t} + \mathbf{V} \cdot \nabla e \right) = -p \nabla \cdot \mathbf{V} + \Phi - \nabla \cdot \mathbf{q} \quad (2.3)$$

Queste relazioni, applicate a un fluido newtoniano, costituiscono il sistema di *equazioni di Navier–Stokes*, che rappresenta il fondamento della descrizione matematica della fluidodinamica. Si tratta di un sistema accoppiato di equazioni differenziali alle derivate parziali, non lineari e fortemente interdipendenti. La loro forma generale non ammette una soluzione analitica chiusa e, nella maggior parte dei casi, è necessario ricorrere a metodi numerici implementati in *fluidodinamica computazionale* (CFD). Tuttavia, in particolari condizioni fisiche o geometriche, esse possono essere semplificate, consentendo lo sviluppo di soluzioni analitiche.

Un concetto fondamentale per comprendere la dinamica dei flussi in prossimità delle superfici solide è quello di *strato limite*: si tratta della regione di fluido adiacente a una parete in cui gli effetti viscosi sono predominanti. All'interno dello strato limite si concentrano intensi gradienti di velocità e, a seconda del numero di Reynolds, esso può essere laminare, transitorio o turbolento.

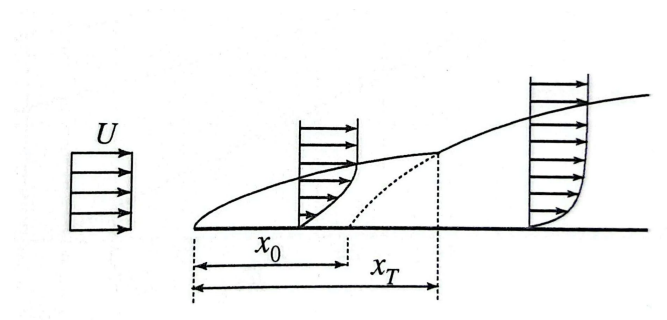


Figura 2.1: Strato limite laminare, transitorio e turbolento.

In modo analogo, quando il fluido è soggetto a scambi termici con la parete, si forma uno *strato limite termico*, in cui la temperatura varia gradualmente dalla parete verso il flusso indisturbato. Lo spessore di questo strato dipende dalle proprietà del fluido e dal numero di Prandtl, un numero adimensionale che esprime il rapporto tra la viscosità cinematica (ν) e la diffusività termica (α).

$$Pr = \frac{\nu}{\alpha} \quad (2.4)$$

Un ulteriore elemento chiave è la *turbolenza*, un fenomeno complesso e onnipresente nei flussi, caratterizzato da moti tridimensionali caotici, elevati gradienti di velocità e un marcato mescolamento. Nonostante la sua apparente irregolarità, la turbolenza presenta proprietà ben definite, tra cui un'elevata efficienza nello scambio di quantità di moto, calore e massa, dovuta alle fluttuazioni distribuite su molteplici scale spaziali e temporali. Nei regimi turbolenti anche piccole variazioni delle condizioni

iniziali possono condurre a comportamenti significativamente diversi, rendendo necessario il ricorso a modelli statistici o a simulazioni numeriche avanzate.

2.2 Scambio termico e convezione

La *temperatura* è una grandezza scalare intensiva che descrive lo stato termico di un sistema. Essa è correlata al livello medio di energia cinetica delle particelle che lo compongono e rappresenta il parametro che determina la direzione del trasferimento spontaneo di energia termica tra due corpi: il calore fluisce sempre dal corpo a temperatura più elevata verso quello a temperatura inferiore.

Il *calore*, invece, è una forma di energia in transito che si trasferisce tra sistemi o porzioni di un sistema in presenza di una differenza di temperatura. Si tratta di una grandezza estensiva, dipendente dalla massa del corpo e dalla sua capacità termica. Il trasferimento di calore può avvenire attraverso tre meccanismi principali: *conduzione*, *convezione* e *irraggiamento*.

Nel contesto del presente lavoro, il meccanismo di scambio termico predominante è la *convezione forzata*, dove il trasferimento di calore è associato al moto macroscopico di un fluido indotto da una sorgente esterna.

Il fenomeno convettivo viene generalmente descritto tramite il numero di Nusselt (Nu), parametro adimensionale che rappresenta il rapporto tra il flusso termico convettivo e quello conduttivo:

$$Nu = \frac{h d}{k} \quad (2.5)$$

dove d è la dimensione caratteristica del sistema, h è il coefficiente di scambio termico convettivo e k la conducibilità termica del fluido.

Nel caso di convezione forzata, Nu dipende principalmente dai numeri adimensionali di Reynolds (Re) e di Prandtl (Pr), secondo relazioni empiriche del tipo:

$$Nu = C Re^n Pr^m \quad (2.6)$$

in cui C , n e m sono coefficienti determinati sperimentalmente e dipendenti dalle condizioni di flusso e dalla geometria del sistema.

La misura accurata della temperatura del flusso, oggetto del presente studio, risulta quindi fondamentale per valutare l'efficienza dello scambio termico e per caratterizzare i fenomeni convettivi che si instaurano a seguito dell'inserimento del getto pulsato.

2.3 Getti pulsati e controllo attivo del flusso

Il controllo del flusso rappresenta una strategia efficace per migliorare il trasferimento di calore nei flussi turbolenti. Le tecniche disponibili si distinguono in *controllo passivo* e *controllo attivo*.

Il controllo passivo si basa su soluzioni semplici e robuste, prive di consumo energetico, che una volta impostate non si adattano alle diverse condizioni operative del sistema. Al contrario, il controllo attivo richiede un apporto energetico e l'impiego di sistemi più complessi, poichè modula l'intensità e la modalità dell'intervento in funzione dello stato istantaneo del sistema. Proprio grazie a questa modulazione dinamica, consente un'azione mirata e adattabile alle condizioni dell'impianto.

Esempi di tecniche attive sono l'*aspirazione* e il *soffiaggio* dello strato limite sulle ali dei velivoli, che permettono di ritardarne la *separazione*; in modo analogo, nel settore automobilistico sistemi come l'ABS modulano dinamicamente la frenata per garantire il controllo del veicolo in condizioni variabili.

Il caso analizzato nel presente elaborato riguarda un getto pulsato in configurazione JICF, in cui una massa di fluido viene espulsa da un ugello perpendicolare alla parete e al flusso principale. Questa configurazione rappresenta una tecnica attiva capace di agire sulla struttura e sull'organizzazione della turbolenza su larga scala, migliorando l'efficienza convettiva del flusso. La pulsazione del getto, generata dalla movimentazione di una valvola a specifiche frequenze e duty cycle, modula nel tempo l'intensità dell'emissione e modifica l'interazione tra getto e strato limite, intensificando il mescolamento con il flusso principale e producendo, di conseguenza, un aumento dello scambio termico e del trasporto di quantità di moto.

Per analizzare sperimentalmente l'effetto della pulsazione sullo scambio termico, è necessario misurare la temperatura a cui il getto viene immesso nel flusso principale. Data la natura non stazionaria del fenomeno, il sensore deve essere caratterizzato da elevata sensibilità, tempi di risposta rapidi e dimensioni ridotte, in modo da minimizzare l'intrusione nel flusso. A tal fine vengono impiegati termistori NTC, sensori resistivi la cui resistenza diminuisce con l'aumentare della temperatura. Le misure acquisite consentono di correlare la risposta termica del flusso ai parametri di attuazione (frequenza, duty cycle e velocità del getto) valutando così l'efficacia del controllo.

Capitolo 3

Contesto Sperimentale

3.1 Laboratorio CICLoPE

Le attività sperimentali sono state svolte presso il laboratorio CICLoPE (Center for International Cooperation in Long Pipe Experiments), situato a Predappio (FC).

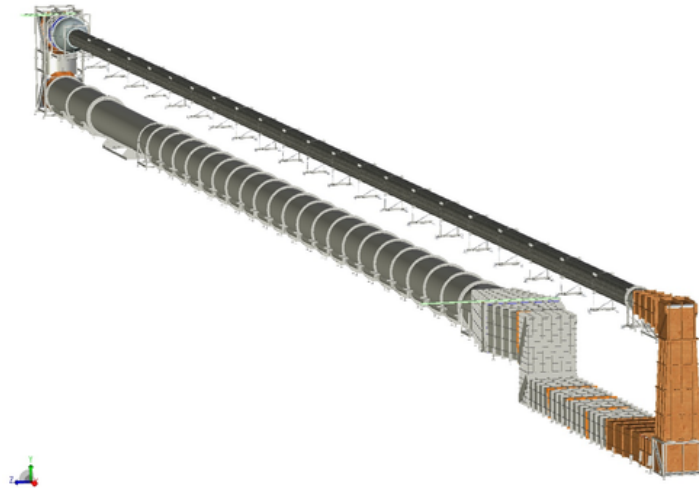


Figura 3.1: Struttura del Long Pipe del laboratorio CICLoPE

Il laboratorio si trova all'interno di un complesso di gallerie sotterranee, che permette di avere condizioni ambientali estremamente stabili e isolate. Il cuore della struttura è il Long Pipe, una galleria del vento ad anello chiuso che opera con aria a pressione atmosferica. La scelta di una configurazione chiusa consente di controllare con precisione le caratteristiche del flusso (velocità, temperatura e umidità) garantendo un'elevata ripetibilità delle misure. Il sistema è in grado di raggiungere velocità fino a 60 m/s, mantenendo variazioni di temperatura inferiori a ± 0.1 °C. Le dimensioni del condotto permettono di esplorare un ampio intervallo di numeri di Reynolds,

mantenendo i fenomeni turbolenti sufficientemente grandi da poter essere analizzati tramite le tecniche di misura convenzionali, come le sonde a filo caldo (Hot-wire), senza necessità di correzioni successive.

Tabella 3.1: Caratteristiche principali del Long Pipe.

Parametro	Valore
Lunghezza totale	120 m
Lunghezza del condotto	111.5 m
Diametro	$0.9 \text{ m} \pm 0.1 \text{ mm}$
Rugosità superficiale interna	$< 0.21 \mu\text{m}$
Velocità massima	60 m/s
Potenza della ventola	380 kW
Variazione di temperatura	$< \pm 0.1^\circ\text{C}$
Lunghezza della <i>test section</i>	1.5 m

Il laboratorio CICLoPE rappresenta dunque una struttura unica a livello internazionale per lo studio sperimentale della turbolenza ad alti numeri di Reynolds in condotti. Tale caratteristica è particolarmente utile per indagare fenomeni complessi come l'interazione tra getto pulsato e flusso, in cui si generano strutture vorticosi che influenzano lo scambio di calore.

3.2 Il setup

Il setup sperimentale utilizzato nel presente lavoro è stato realizzato da Magnani [1] durante la sua tesi di laurea magistrale presso il laboratorio CICLoPE.

Secondo lo scopo del presente elaborato, tale impianto è stato migliorato aggiungendo un sensore di temperatura, un termistore NTC, così da fornire una maggiore caratterizzazione del getto.

Il sistema opera con aria in pressione, tipicamente compresa tra 1.5 e 3 bar, e a una temperatura di circa 20 °C. Il flusso scorre all'interno di un condotto di diametro pari a 4 mm. Il sensore di temperatura è stato immerso in direzione perpendicolare al moto del fluido tramite un raccordo a T e un elemento di connessione progettato appositamente. Tale configurazione consente di ridurre al minimo l'intrusione del sensore nel flusso, preservandone le caratteristiche locali.

Tabella 3.2: Caratteristiche principali del setup sperimentale.

Parametro	Valore
Tipo di flusso	Getto d'aria compressa
Pressione di alimentazione	1.5–3 bar
Temperatura aria in ingresso	$\approx 20\text{ }^{\circ}\text{C}$
Diametro condotto	4 mm
Sensore di temperatura	Termistore NTC
Orientazione del sensore	Perpendicolare al flusso
Montaggio	Raccordo a T con supporto dedicato
Finalità	Caratterizzazione termica del getto

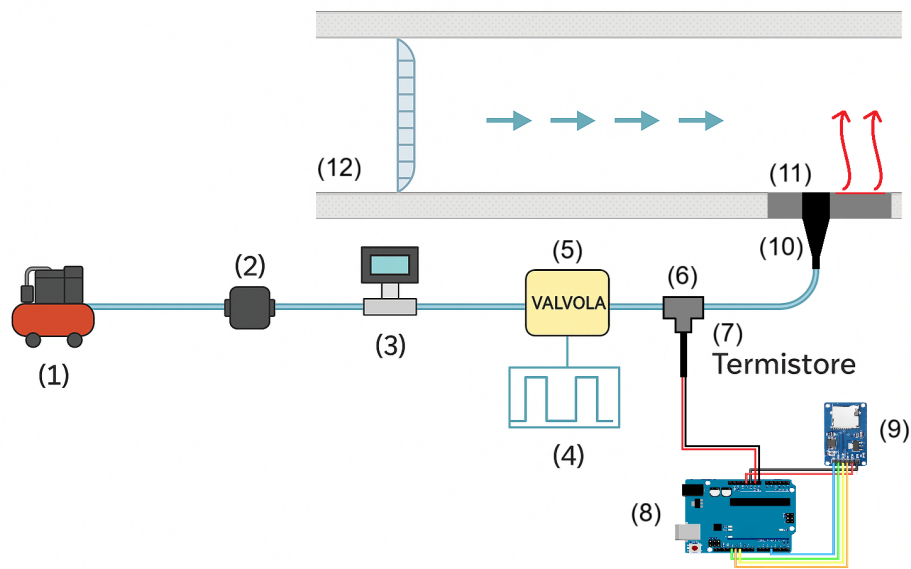
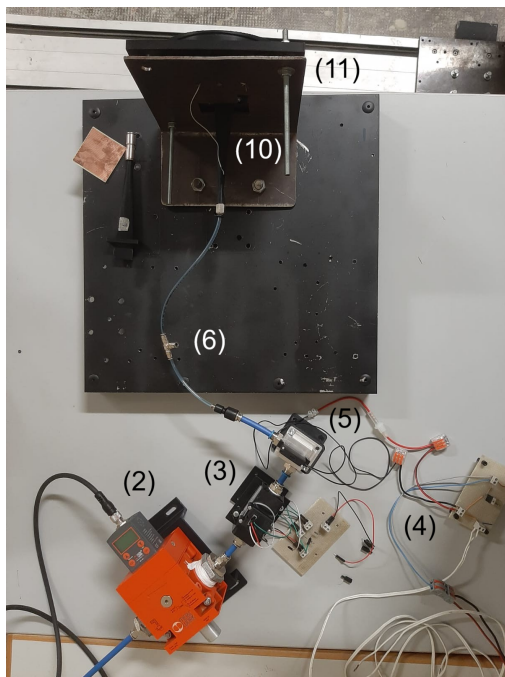


Figura 3.2: Schema dell'impianto sperimentale.

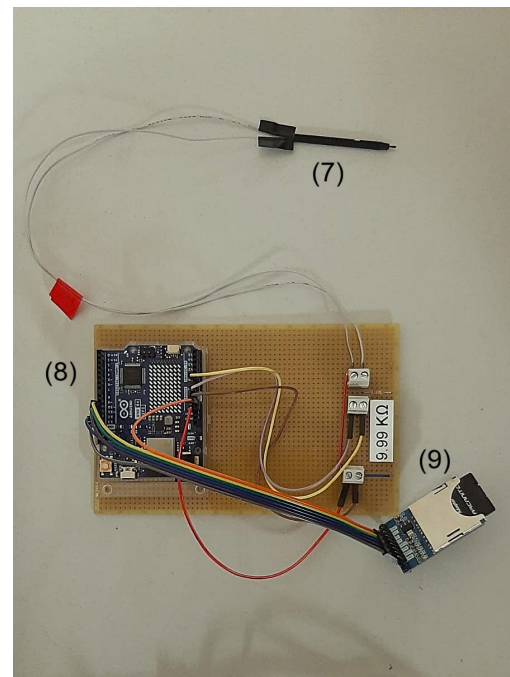
L'impianto è stato realizzato con i seguenti componenti:

1. Aria compressa;
2. Regolatore di pressione;
3. Sensore di portata massica *Honeywell HAFUHT0200L4AXT*;
4. Circuito elettronico di controllo;
5. Valvola solenoide *SMC SX12F-BH*;
6. Giunto a T;

7. Interfaccia meccanica per l'inserimento del sensore e termistore NTC;
8. Scheda *Arduino Uno R4 WiFi*;
9. Scheda SD per la memorizzazione dei dati;
10. Getto scanalato (slotted jet);
11. Innesto stampato in 3D che ospita *Printed Circuit Board* (PCB)
12. Long Pipe del laboratorio CICLoPE;
13. *CompactDAQ NI cDAQ-9189* con diversi moduli;
14. Sistema *Dantec StreamLine 90N10*;
15. Computer con *NI LabVIEW*.



(a)



(b)

Figura 3.3: Setup sperimentale per la misura della temperatura: (a) vista dell'impianto senza sensore, (b) dettaglio dei componenti per la misura della temperatura.

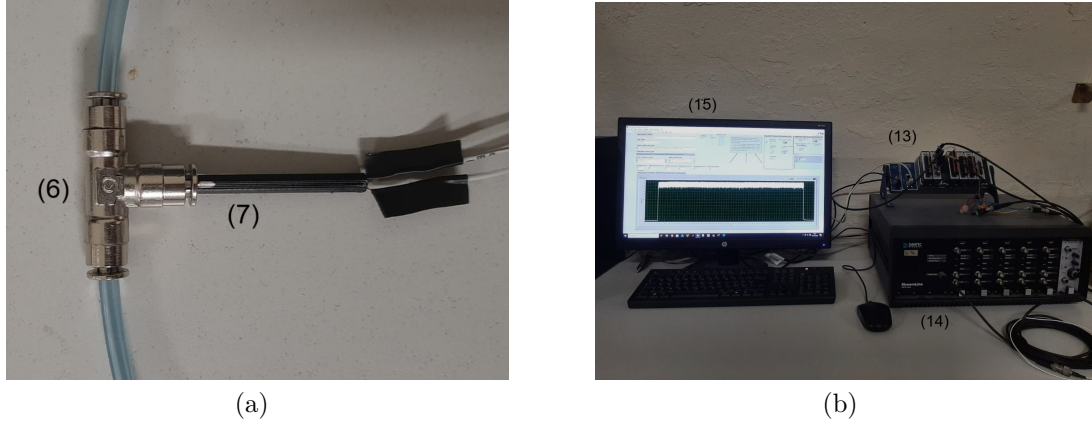


Figura 3.4: Particolari del setup: (a) Giunto a T con sensore alloggiato, (b) sistema di acquisizione ed elaborazione dei dati.

3.3 Scelta del sensore di temperatura

La misura della temperatura del flusso all'interno dell'impianto sperimentale rappresenta un requisito fondamentale per caratterizzare gli effetti del getto pulsato. Per questo motivo, tra le diverse tecnologie di misura della temperatura è stato necessario individuare un sensore sufficientemente preciso, rapido e compatibile con le dimensioni ridotte del canale.

Le tecniche di misura della temperatura possono essere distinte in intrusive e non intrusive. Nel caso della presente applicazione, la misura richiede necessariamente un sensore immerso nel flusso, e pertanto la selezione deve tenere conto anche delle conseguenze di tale posizionamento. Quando un sensore intrusivo è posto in un fluido in moto, si genera uno scambio termico tra il sensore e il flusso, comportando la misura di una temperatura leggermente superiore rispetto alla temperatura indisturbata. Tuttavia, per velocità inferiori a Mach 0.22 l'errore rimane inferiore all'1%, e nel caso in oggetto tale contributo può considerarsi trascurabile. Allo stesso modo, fenomeni quali scambio termico per irraggiamento, conduzione verso strutture solide e dissipazione viscosa risultano secondari rispetto alla convezione forzata.

Un aspetto invece rilevante riguarda l'inerzia termica del sensore, in quanto determina il tempo necessario affinché la misura raggiunga la temperatura reale del flusso. Di conseguenza, dimensioni ridotte e bassa massa termica costituiscono requisiti fondamentali.

Sulla base delle precedenti valutazioni sono state analizzate tre famiglie di sensori: *termocoppie*, sensori termoresistivi metallici (*termoresistenze* e *cold-wire*) e *termistori*. La seguente trattazione riassume le principali caratteristiche di queste tecnologie e motiva la scelta finale adottata nel presente lavoro di tesi.

3.3.1 Termocoppie

Le termocoppie sono sensori di temperatura largamente utilizzati in contesti industriali e sperimentali. Il loro principio di funzionamento si basa sull'*effetto Seebeck*: una differenza di potenziale elettrico si genera tra due conduttori metallici diversi quando le giunzioni sono soggette a temperature differenti. Le termocoppie sono quindi sensori relativi, poiché misurano la differenza di temperatura tra la giunzione calda e quella di riferimento, rendendo necessaria la compensazione della giunzione fredda per ottenere misure affidabili.

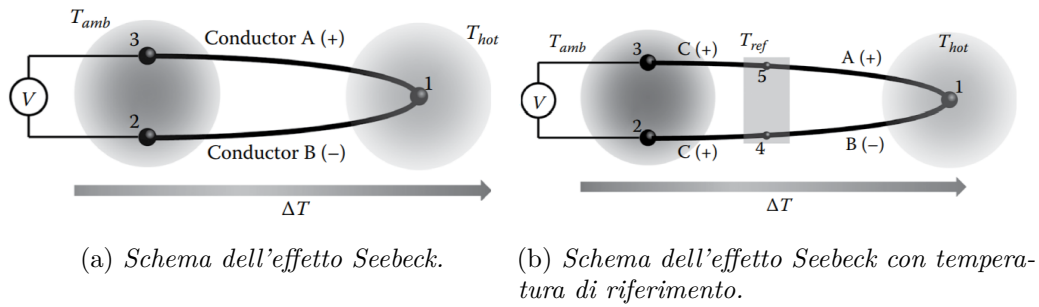


Figura 3.5: Schema circuitale di una termocoppia.

I vantaggi principali delle termocoppie risiedono nell'ampio intervallo operativo (da circa -270°C fino a oltre 2000°C), nella robustezza meccanica, nella rapidità di risposta e nella vasta disponibilità commerciale. Tuttavia, presentano una sensibilità relativamente bassa, una marcata esposizione al rumore elettrico e una non linearità che richiede calibrazione o curve di conversione dedicate. L'accuratezza tipica, pari a circa $\pm 0.5^{\circ}\text{C}$, rappresenta un limite significativo nel ristretto intervallo termico del setup ($20\text{--}30^{\circ}\text{C}$), in cui tali incertezze risultano troppo elevate rispetto alla precisione richiesta. Inoltre, la necessità di configurazioni geometriche molto compatte ha ristretto notevolmente le opzioni disponibili, con costi sensibilmente maggiori per i modelli adatti al canale sperimentale. Per questi motivi, le termocoppie sono state escluse.

3.3.2 Termoresistenze

Le termoresistenze, o *Resistance Temperature Detector* (RTD), sono sensori termoresistivi realizzati con metalli, tipicamente il platino. Il loro principio di funzionamento si basa sulla variazione della resistività elettrica del materiale con la temperatura: nei metalli, infatti, l'aumento della temperatura comporta un incremento della resistività. La misura si ottiene monitorando la variazione di resistenza dell'elemento sensibile e convertendola direttamente in temperatura, a differenza delle termocoppie che rilevano un gradiente termico tra due giunzioni.

Le RTD offrono un'elevata stabilità e buona linearità della relazione resistenza-temperatura, con accuratizie tipiche dell'ordine di $\pm 0.1^\circ\text{C}$. Sono disponibili in configurazioni a 2, 3 o 4 fili, che permettono di compensare la resistenza parassita dei collegamenti e migliorare la precisione della misura.



Figura 3.6: Esempio di termoresistenza in platino a 4 fili.

Nonostante i vantaggi in termini di precisione, le RTD presentano tempi di risposta più elevati, dovuti alla massa termica maggiore. Inoltre, i modelli sufficientemente sensibili e rapidi risultano costosi e offrono geometrie non compatibili con il canale di misura. Per tali ragioni, anche le RTD sono state escluse.

3.3.3 Cold-wire thermometry

Le tecniche di *cold-wire thermometry* impiegano un filo metallico estremamente sottile ($1\text{--}5\text{ }\mu\text{m}$), alimentato a corrente costante e mantenuto a una temperatura lievemente superiore di quella del fluido. Anche in questo caso, la misura si basa sulla variazione della resistenza elettrica del filo, che risponde quasi istantaneamente alle fluttuazioni della temperatura del flusso grazie alla bassa inerzia termica. Questa tecnologia garantisce un'elevata sensibilità alle variazioni rapide di temperatura.

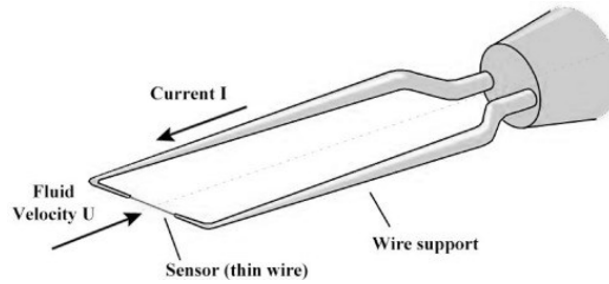


Figura 3.7: Struttura cold-wire.

Per quanto tali qualità rendano questa tecnica adatta all'uso in flussi turbolenti, l'utilizzo di un cold-wire non è risultato compatibile con l'impianto oggetto di questo studio. La procedura di calibrazioni è complessa e richiede condizioni di riferimento molto stabili; inoltre, il sensore è estremamente fragile e l'inserimento all'interno del canale ne comprometterebbe facilmente l'integrità. Per queste ragioni, si è esclusa la cold-wire thermometry come tecnica adatta a questo lavoro.

3.3.4 Termistori

I termistori, analogamente alle RTD, appartengono alla categoria dei sensori termoresistivi, ma a differenza di queste ultime sono realizzati con materiali semiconduttori anziché metallici. La tipologia di termistori più diffusa è quella degli NTC, prodotti con ossidi semiconduttori nei quali l'aumento della temperatura favorisce la generazione di portatori di carica, con conseguente riduzione della resistività. La misura di temperatura viene quindi ottenuta alimentando il termistore e determinando il valore della resistenza elettrica, funzione diretta della temperatura del materiale.

Oltre agli NTC esiste la categoria di termistori PTC (Positive Temperature Coefficient), per i quali la resistenza aumenta all'aumentare della temperatura. Questi ultimi trovano impiego prevalentemente come dispositivi di protezione o come elementi di commutazione, più che come sensori per la misura accurata della temperatura.

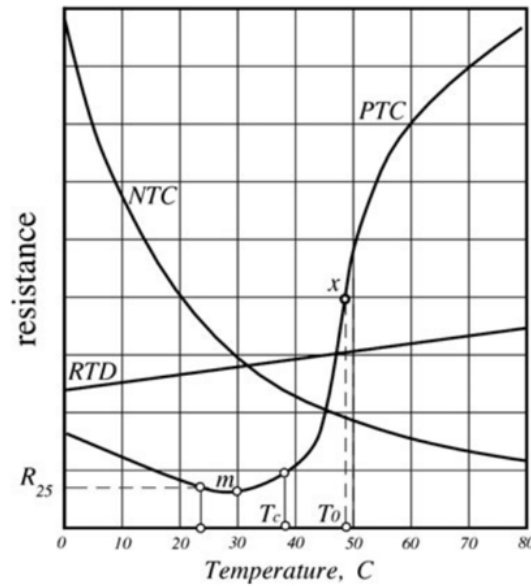


Figura 3.8: Andamento di vari sensori termoresistivi metallici e semiconduttori, da *Handbook of Modern Sensors*, Fraden [2].

La relazione tra resistenza e temperatura in un termistore è fortemente non lineare e dipende da parametri specifici dell'elemento, quali dimensioni, geometria e composizione del materiale. Per questo motivo è necessaria una calibrazione individuale, che può essere condotta mediante modelli matematici di diversa complessità. Nonostante tale esigenza, i termistori presentano caratteristiche particolarmente vantaggiose: la sensibilità è molto elevata e superiore a quella delle RTD, mentre le ridotte dimensioni garantiscono una bassa massa termica, traducendosi in tempi di risposta estremamente rapidi, caratteristica che li rende idonei a misure in condizioni variabili o transitorie. A ciò si aggiungono una buona stabilità e ripetibilità, se opportunamente calibrati, un costo contenuto e un'ampia disponibilità di configurazioni geometriche, che ne agevolano l'impiego anche in spazi ridotti.



Figura 3.9: Esempi di termistori.

Gli svantaggi principali riguardano la mancanza di standardizzazione delle curve di taratura e la marcata non linearità della relazione resistenza-temperatura. Un'ulteriore limitazione è rappresentata dal campo operativo, generalmente compreso tra $-50\text{ }^{\circ}\text{C}$ e $150\text{ }^{\circ}\text{C}$, al di fuori del quale le prestazioni decadono sensibilmente. Tale intervallo risulta comunque pienamente compatibile con le condizioni previste nel setup sperimentale.

Alla luce di queste considerazioni, i termistori NTC si sono rivelati la soluzione più adatta a questo progetto, grazie alla combinazione di elevata sensibilità, rapidità di risposta, dimensioni ridotte e adeguata affidabilità nelle condizioni operative previste.

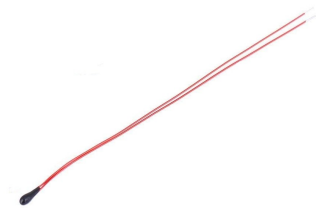


Figura 3.10: Termistore utilizzato nel presente setup.

Tabella 3.3: Caratteristiche principali dei termistori NTC.

Principio di funzionamento	Variazione della resistenza di un semiconduttore con coefficiente di temperatura negativo.
Tipo di misura	Assoluta
Materiali	Ossidi metallici semiconduttori (Mn, Ni, Co, Cu, Fe)
Range tipico di temperatura	Da $-50\text{ }^{\circ}\text{C}$ a $150\text{ }^{\circ}\text{C}$
Sensibilità	Molto alta, ma non lineare
Tempo di risposta	Molto rapido ($< 1\text{ s}$)
Stabilità nel tempo	Media (dipendente da materiali e condizioni d'esercizio)
Costo	Basso-medio
Robustezza	Buona
Linearità	Non lineare (richiede calibrazione)
Alimentazione richiesta	Sì (misura resistiva)
Applicazioni tipiche	Misura e controllo della temperatura, applicazioni a basso costo e risposta rapida

3.4 Elemento di connessione

La realizzazione di questo componente si è resa necessaria per consentire l'inserimento del termistore all'interno del canale in pressione tramite il giunto a T, garantendone al tempo stesso il fissaggio stabile per calzamento. Sono state analizzate diverse soluzioni e geometrie con l'obiettivo di assicurare una tenuta meccanica efficace, evitare il contatto tra i poli del termistore e permettere una rapida intercambiabilità tra i tre sensori calibrati.

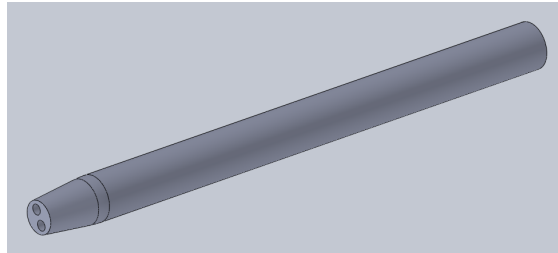


Figura 3.11: Componente di connessione del termistore al giunto a T.

La configurazione finale scelta è quella di un componente progettato ad hoc, in grado di ospitare il sensore su una delle estremità, così da poter essere agevolmente inserito nel giunto, mentre i cavi di collegamento vengono alloggiati in due canali separati all'interno del corpo del pezzo. Il modello è stato progettato in SolidWorks e successivamente stampato in 3D presso il laboratorio CICLoPE.



Figura 3.12: Termistore alloggiato nel componente di connessione.

3.5 Arduino, circuito e codice

Arduino è un dispositivo basato su un microcontrollore, progettato per leggere input e generare output sotto forma di segnali elettrici. L'informazione viene trasmessa attraverso livelli di tensione: i pin della scheda Arduino, quando configurati come ingressi digitali, restituiscono lo stato Low (0) per tensioni prossime a 0 V e High (1) per valori vicini a 5V.

Un segnale digitale può quindi assumere solo questi valori discreti, mentre un segnale analogico, come quello prodotto da un termistore, può variare in modo continuo all'interno di un certo intervallo (es. 1.04 V, 2.34 V, 3.78 V). Di conseguenza il segnale proveniente dal termistore deve essere convertito in digitale, ciò è possibile tramite l'ADC (Analog to Digital Converter) integrato nella scheda.

Nel caso dell'Arduino UNO R4 WiFi, l'ADC opera con un risoluzione di 14 bit, restituendo valori compresi tra 0 e 16383 ($2^{14}-1$). Ciò significa che l'intervallo di tensione viene suddiviso in 16384 livelli, con una risoluzione di circa $\frac{5V}{16384} = 0.305mV$, molto più fine rispetto ai 10 bit dei modelli Arduino precedenti. Inoltre, per garantire una maggiore stabilità del segnale si è scelto di alimentare il termistore a 3.3 V, tensione regolata internamente da Arduino e quindi più stabile rispetto ai 5 V forniti dalla porta USB. Questa scelta comporta una risoluzione effettiva ancora più elevata, pari a $\frac{3.3V}{16384} = 0.201mV$.

Per poter misurare la resistenza del termistore, che non è acquisibile dal microcontrollore, il sensore è stato inserito in un *partitore di tensione* insieme a una resistenza di pull-down. In questo modo, con una tensione di alimentazione del sensore di 3.3 V, la variazione di resistenza del termistore si traduce in una variazione di tensione direttamente acquisibile dall'ADC.

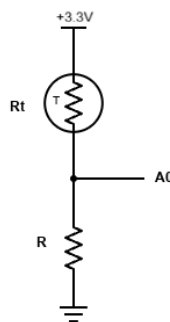


Figura 3.13: Rappresentazione dello schema elettrico del partitore di tensione.

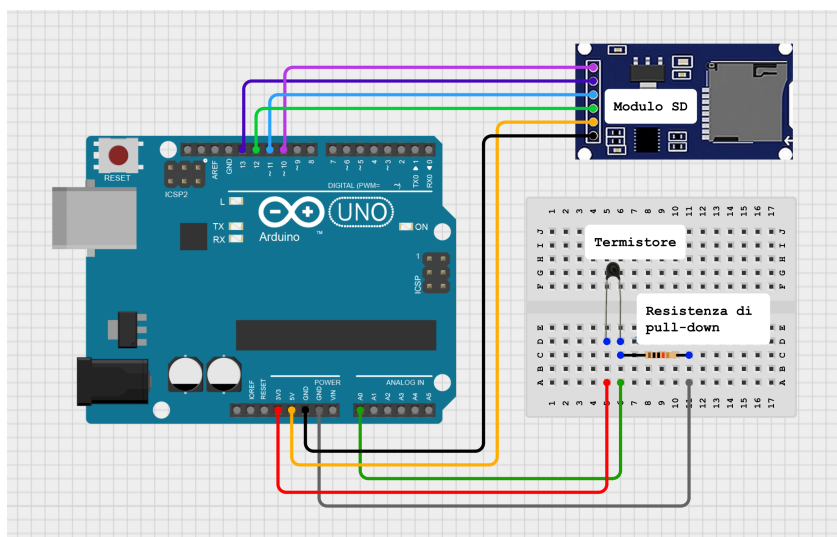


Figura 3.14: Rappresentazione delle connessioni ad Arduino.

L'acquisizione della temperatura è stata possibile tramite un codice dedicato che, a partire dalla misura della tensione ai capi del termistori, calcola il corrispondente valore di resistenza tramite la legge del partitore di tensione. La resistenza così ottenuta viene convertita in temperatura attraverso i modelli di interpolazione del sensore, utilizzando i coefficienti ottenuti dalla calibrazione. Il codice include anche funzionalità avanzate: la scheda si connette tramite WiFi a un server NTP (Network Time Protocol), sincronizza l'RTC (Real Time Clock) interno e assegna a ogni campione un timestep preciso, ottenuto combinando il tempo assoluto fornito dall'RTC con la componente di millisecondi restituita da `millis()`. Per garantire affidabilità e continuità, soprattutto perchè la fase di acquisizione è significativamente più veloce della scrittura su SD, è stato implementato un sistema a triplo buffer circolare in RAM. In questo schema:

- il buffer A è utilizzato per l'acquisizione in tempo reale;
- il buffer B contiene i dati appena acquisiti e che vengono scritti su SD;
- il buffer C rimane libero e pronto a essere utilizzato non appena uno dei primi due si riempie.

Questo approccio permette di evitare blocchi o perdite di dati eccessive durante le operazioni più lente di scrittura, garantendo una registrazione continua. L'acquisizione procede a una frequenza teorica di 200 Hz, impostata nel codice tramite un `delay(5)`, e prosegue fino allo scadere del tempo prefissato; al termine, i buffer residui vengono svuotati e il file viene chiuso in modo sicuro. Il risultato è un sistema di acquisizione robusto, affidabile e adatto alla lettura delle temperature misurate dal sensore.

Capitolo 4

Calibrazione del termistore

La calibrazione del termistore è una fase fondamentale per garantire l'affidabilità delle misure di temperatura, poichè permette di determinare con precisione la relazione che lega la resistenza elettrica del sensore alla temperatura. In un termistore NTC, la resistenza decresce in modo non lineare all'aumentare della temperatura, tale comportamento non può essere descritto accuratamente da un'unica funzione, motivo per cui si ricorre a modelli empirici di interpolazione.

Durante la calibrazione vengono misurate le resistenze del termistore a diverse temperature note, mantenute stabili tramite l'uso di un bagno termostatico. I dati sperimentali T_i, R_i vengono quindi utilizzati per determinare i parametri del modello prescelto, il quale consente di interpolare la curva $R = f(T)$ e, in seguito, di ricavare la temperatura corrispondente a un qualsiasi resistenza misurata durante l'esperimento.

L'accuratezza complessiva della misura dipende fortemente dal modello utilizzato per l'interpolazione. Modelli semplici offrono implementazioni rapide e computazionalmente leggere, ma con un errore più elevato; viceversa, modelli più complessi garantiscono precisione superiore, al prezzo di una maggiore difficoltà di calibrazione e di un incremento nel numero di punti sperimentali richiesti.

Di seguito vengono illustrati i principali modelli impiegati per la caratterizzazione dei termistori NTC.

4.1 Modelli di interpolazione per i termistori NTC

I modelli di interpolazione comunemente utilizzati per descrivere il comportamento termoresistivo di un termistore NTC sono tre: il *modello Beta*, il *modello di Fraden* e il *modello di Steinhart–Hart*. Ciascuno rappresenta un livello di approssimazione differente del legame tra temperatura e resistenza.

Modello Beta

Il modello Beta rappresenta l'approccio più semplice e largamente utilizzato in ambito industriale. Esso descrive la relazione tra resistenza e temperatura secondo l'equazione:

$$\ln R \simeq A + \frac{\beta_m}{T} \quad (4.1)$$

dove R è la resistenza del termistore, T la temperatura e β_m una costante caratteristica del materiale, fornita dal costruttore o determinata sperimentalmente a partire da due punti di calibrazione.

Sebbene semplice da implementare, questo modello introduce un errore massimo tipico di circa ± 0.7 °C nell'intervallo 0-70 °C, rivelandosi quindi inadatto per applicazioni che richiedono alta precisione.

Modello di Fraden

Un miglioramento del modello precedente è proposto da Fraden [2], che considera la dipendenza del coefficiente β_m dalla temperatura, introducendo il termine γ che ne rappresenta la pendenza locale:

$$R = R_0 e^{\beta_m [1 + \gamma(T - T_0)] \left(\frac{1}{T} - \frac{1}{T_0} \right)} \quad (4.2)$$

Questo modello garantisce una maggiore accuratezza (errore inferiore a ± 0.03 °C) ma richiede almeno tre punti di calibrazione per determinare completamente i parametri.

Modello di Steinhart–Hart

Il modello di Steinhart–Hart rappresenta l'approccio più preciso e ampiamente riconosciuto nella letteratura scientifica per la descrizione del comportamento termico dei termistori NTC. La relazione tra temperatura assoluta T (in Kelvin) e

resistenza R è data da:

$$T = \frac{1}{c_1 + c_2 \ln R + c_3 (\ln R)^3} \quad (4.3)$$

dove c_1 , c_2 e c_3 sono i coefficienti di calibrazione determinati sperimentalmente a partire da tre coppie R_i, T_i note.

Secondo [2], il modello di Steinhart–Hart è in grado di ridurre l'errore di interpolazione a circa ± 0.003 °C nell'intervallo 0–70 °C , risultando nettamente superiore agli altri modelli in termini di accuratezza. Tale livello di precisione è particolarmente rilevante quando il termistore rappresenta l'unico elemento sensibile del sistema di misura, come nel caso del setup sperimentale considerato in questo lavoro.

Nonostante il maggior onere computazionale rispetto ai modelli precedenti, la semplicità con cui può essere implementato in microcontrollori moderni, come Arduino, lo rende la scelta più equilibrata tra complessità, accuratezza e stabilità numerica.

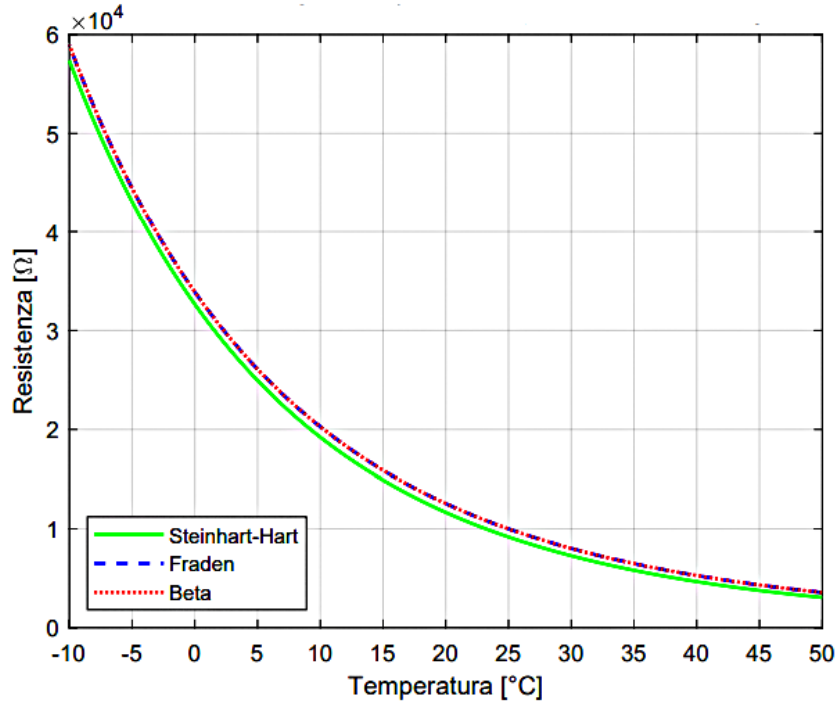


Figura 4.1: Curva resistenza-temperatura dei modelli Beta, Fraden e Steinhart-Hart.

4.2 Calibrazione contemporanea di tre termistori

La calibrazione dei tre termistori è stata effettuata mediante un bagno termostatico *VWR 1166*, uno strumento che permette di portare un fluido alla temperatura desiderata e mantenerla con elevata stabilità.



Figura 4.2: Bagno termostatico utilizzato per la calibrazione.

Tabella 4.1: Caratteristiche principali del bagno termostatico utilizzato per la calibrazione.

Parametro	Valore
Intervallo di temperatura	$-20 \div 200\text{ }^{\circ}\text{C}$
Variazione di temperatura	$\pm 0.01\text{ }^{\circ}\text{C}$
Capacità della vasca	6 L
Potenza del riscaldatore	2200 W
Apertura della vasca	$135 \times 135\text{ mm}$
Profondità della vasca	150 mm

I tre termistori sono stati immersi simultaneamente nell'acqua del bagno termostatico e fissati in una posizione stabile, in modo da limitare gli effetti dovuti al moto del fluido, alle pareti refrigerate e alla resistenza di riscaldamento. La procedura è stata ripetuta per i tre valori di temperatura scelti in accordo con il range operativo previsto nel setup sperimentale: $15\text{ }^{\circ}\text{C}$ come limite inferiore, $30\text{ }^{\circ}\text{C}$ come valore intermedio e $50\text{ }^{\circ}\text{C}$ come valore massimo. Per ciascuna temperatura sono state acquisite serie temporali della durata di 2 minuti, con una frequenza teorica di

campionamento di 200 Hz, tramite uno script Arduino sviluppato appositamente. Terminata l'acquisizione, le fasi di transitorio iniziale sono state rimosse e i segnali stabilizzati sono stati elaborati mediante uno script MATLAB dedicato. Per ogni sensore è stato calcolato il valore medio di resistenza in corrispondenza di ciascuna temperatura, e le coppie R_i, T_i sono state utilizzate per ricavare i coefficienti della relazione di Steinhart–Hart. Questo ha permesso di ottenere una calibrazione specifica per ciascun termistore, necessaria per garantire accuratezza nelle misure successive.

4.3 Risultati della calibrazione

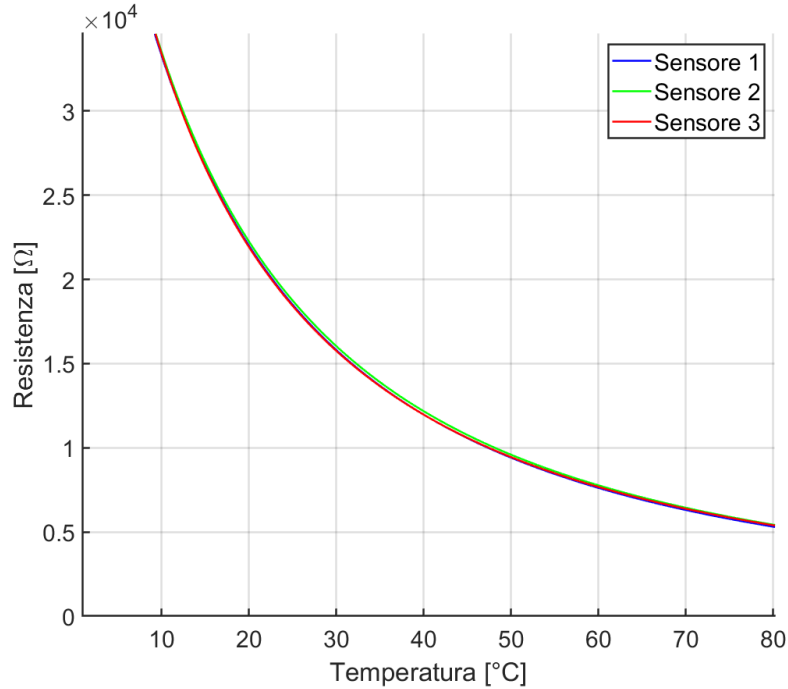


Figura 4.3: Curva resistenza-temperatura dei tre sensori calibrati.

Tabella 4.2: Coefficienti di calibrazione Steinhart–Hart dei termistori NTC.

Sensore	c_1	c_2	c_3
1	-0.0041852	0.0009805	-2.2082×10^{-6}
2	-0.0042458	0.00098654	-2.2117×10^{-6}
3	-0.0045697	0.0010393	-2.4102×10^{-6}

Un aspetto critico emerso durante la fase di calibrazione riguarda la differenza tra le misure acquisite con Arduino e quelle ottenute tramite la DAQ. Nonostante il termistore fosse alimentato alle stesse condizioni e inserito nel medesimo partitore

di tensione, i valori letti dai due sistemi risultano diversi.

La differenza osservata tra le misure è riconducibile soprattutto all'impedenza di ingresso dello strumento di acquisizione. Nel partitore formato dal termistore e dalla resistenza di pull-down, l'impedenza del nodo di misura è relativamente elevata; quindi strumenti come multimetro e DAQ, caratterizzati da un'impedenza di ingresso molto maggiore, risultano praticamente trasparenti e rilevano la tensione reale del partitore. L'ingresso analogico di Arduino, invece, durante ogni conversione deve caricare un condensatore interno: se la sorgente ha impedenza alta, questa richiesta provoca un abbassamento della tensione sul nodo, producendo una misura sistematicamente diversa rispetto a quella della DAQ.

Questo comportamento evidenzia come i valori acquisiti non dipendano solo dal sensore, ma anche del circuito con cui viene svolta la calibrazione.

Per mitigare questo effetto è sufficiente inserire un condensatore tra il nodo di misura e la massa, in questo modo la misura risulta più stabile e coerente con quella ottenuta tramite la DAQ.

Capitolo 5

Risultati sperimentali e discussione

Una volta completato lo sviluppo del sistema di misura e verificato il corretto funzionamento dell'intera catena di acquisizione, sono state condotte diverse analisi sperimentali finalizzate a valutare l'influenza dei parametri di funzionamento della valvola sul valore di temperatura misurato nel getto. In particolare, sono state eseguite prove a varie frequenze di pulsazione e duty cycle della valvola, mentre la pressione di ingresso è rimasta costante a 1.5 bar. I dati sono stati post-processati tramite script MATLAB sviluppati appositamente.

Tabella 5.1: Parametri sperimentali per le misurazioni di temperatura.

Parametro	Valore
Pressione di ingresso	1.5 bar
Durata acquisizione	60 s
Frequenza di campionamento	180–200 Hz
Frequenze di pulsazione testate	50 Hz, 150 Hz, 300 Hz
Duty cycle	25%, 50%, 75%
Sistema di acquisizione	Arduino
Orientazione sensore	Perpendicolare al flusso
Posizionamento sensore	A taglio / Frontale

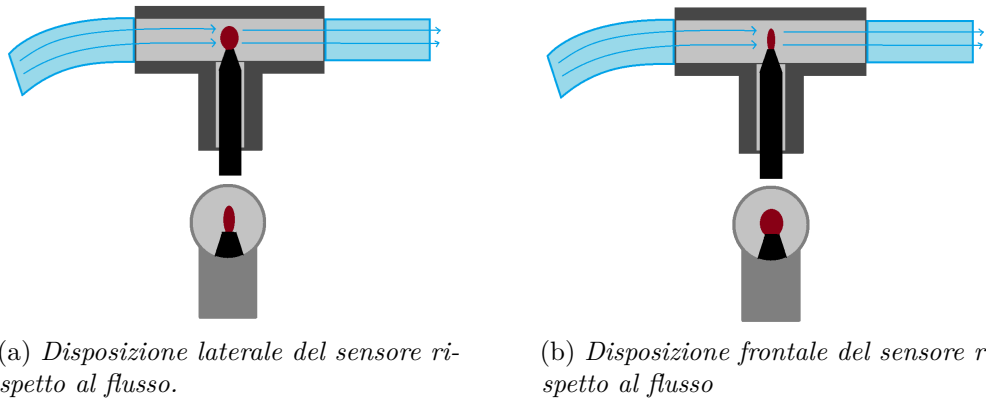


Figura 5.1: Disposizioni del termistore.

Successivamente è stata eseguita un'analisi di intrusività del sensore, volta a valutare l'effetto della sua presenza osservando la velocità dell'aria in uscita dal condotto di diametro 4 mm. Tale analisi è stata eseguita mediante anemometria a filo caldo, confrontando tre configurazioni: assenza del sensore, sensore inserito lateralmente e sensore inserito frontalmente. Anche in questo caso la pressione di ingresso è stata mantenuta pari a 1.5 bar e le acquisizioni, della durata di 60s, sono state elaborate tramite appositi script MATLAB.

Tabella 5.2: Parametri per le prove di valutazione dell'intrusività mediante hot-wire.

Parametro		Valore
Pressione di ingresso		1.5 bar
Durata acquisizione		60 s
Frequenza di campionamento		12 kHz
Pulsazione valvola		Nessuna (valvola sempre aperta)
Duty cycle		100%
Sistema di acquisizione		DAQ
Orientazione sensore temperatura		Perpendicolare al flusso
Posizionamento sensore		Laterale / Frontale / Assente

5.1 Effetti di frequenza e duty cycle della valvola

Una prima analisi è stata effettuata confrontando l'effetto della frequenza di pulsazione a diversi valori di duty cycle. Dal punto di vista teorico, si può ipotizzare che frequenze maggiori conducano a temperature medie più elevate per due motivi principali: la valvola compie cicli più rapidi e quindi tende a riscaldarsi maggiormente, e la durata ridotta della fase di apertura limita il passaggio di aria nel

condotto, riducendo la capacità del flusso di asportare calore. Tale comportamento si osserva nei grafici relativi ai casi con sensore disposto lateralmente e duty cycle del 75% e del 50%, nei quali la temperatura media misurata cresce all'aumentare della frequenza ($300 \text{ Hz} > 150 \text{ Hz} > 50 \text{ Hz}$). Invece, nei casi a disposizione frontale si osserva una inversione per i segnali a 50 Hz e 150 Hz.

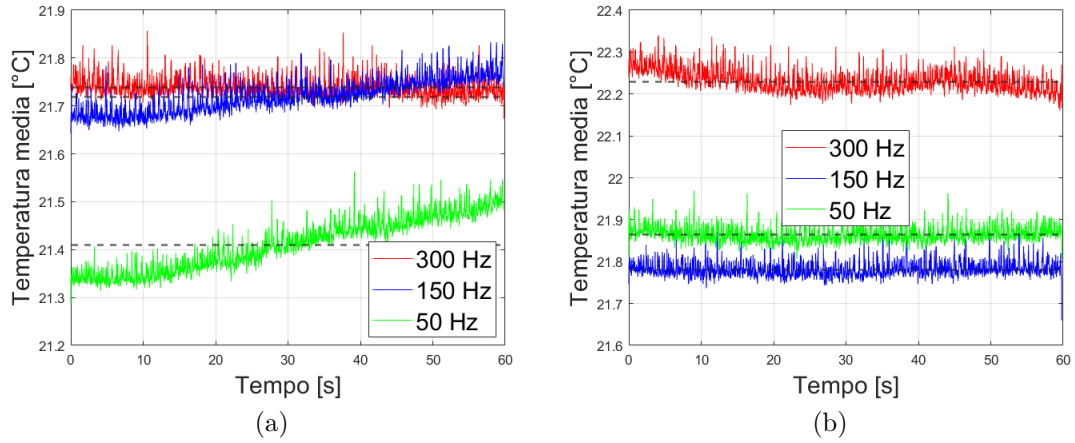


Figura 5.2: Confronto dei segnali di temperatura a diverse frequenze e $DC = 75\%$: (a) disposizione laterale, (b) disposizione frontale.

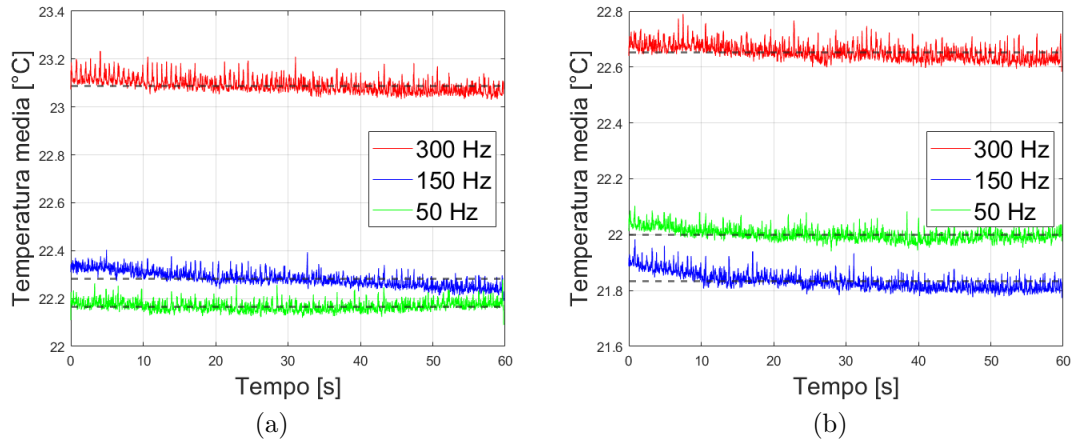


Figura 5.3: Confronto dei segnali di temperatura a diverse frequenze e $DC = 50\%$: (a) disposizione laterale, (b) disposizione frontale.

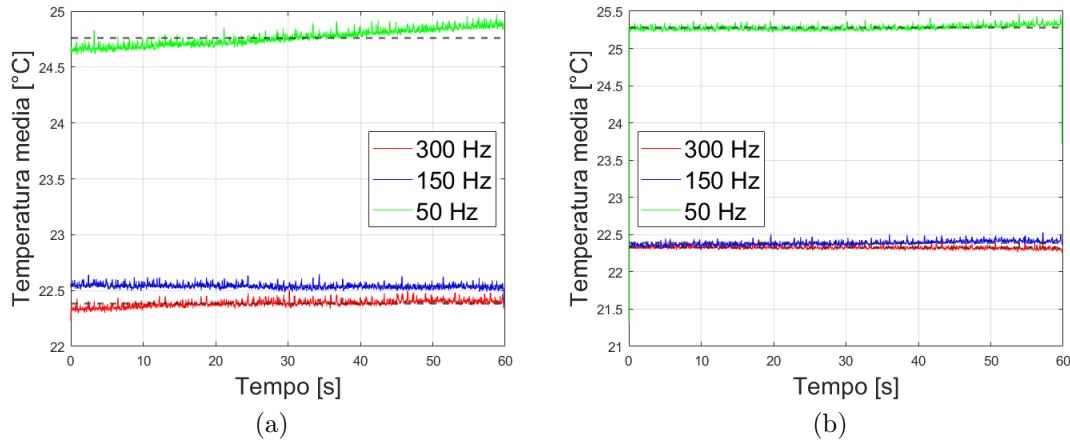


Figura 5.4: Confronto dei segnali di temperatura a diverse frequenze e $DC = 25\%$: (a) disposizione laterale, (b) disposizione frontale.

Nel caso a duty cycle 25% , invece, si osserva un andamento opposto rispetto a quello atteso, con temperatura maggiore a 50 Hz e valori inferiori per 150 Hz e 300 Hz . In questa configurazione si riscontra anche un marcato riscaldamento della valvola, e un comportamento simile si verifica anche per il posizionamento frontale del sensore.

Per interpretare tali osservazioni è utile considerare i tempi effettivi di apertura della valvola. A causa delle inerzie meccaniche ed elettriche dell'attuatore, è presente un ritardo di risposta che diventa rilevante quando il tempo di apertura imposto dal comando si riduce. Alle frequenze più basse tale ritardo non influenza in modo significativo la dinamica della valvola e la portata erogata conserva un andamento coerente con il duty cycle nominale. Al crescere della frequenza, invece, il tempo disponibile per la fase di apertura diventa paragonabile al ritardo stesso, e la valvola opera prevalentemente in regime transitorio, senza completare la corsa e raggiungere una reale condizione di chiusura. In queste condizioni il passaggio di aria, pur ridotto, risulta continuo e favorisce un maggiore raffreddamento dell'attuatore. Al contrario, alle frequenze più basse la corrente ha il tempo di salire ai valori di regime e la valvola completa quasi interamente il movimento, dissipando più energia. La combinazione di questi fattori può giustificare il maggiore riscaldamento osservato nel caso a 50 Hz con duty cycle del 25% .

5.2 Influenza dell'orientazione del sensore

E' stata compiuta anche un'analisi confrontando la temperatura misurata dal sensore in due configurazioni geometriche: orientazione frontale rispetto al flusso e

orientazione a laterale. Dal punto di vista teorico, si ipotizza che la disposizione laterale comporti una temperatura misurata leggermente superiore, poiché il sensore ostacola meno il flusso, subisce minore raffreddamento e perturba in misura inferiore il getto. Tale andamento è stato confermato per il duty cycle del 50% a tutte le frequenze testate (50, 150 e 300 Hz), per cui $T_{lato} > T_{fronte}$. Per gli altri valori di duty cycle invece non si evidenzia un chiaro andamento.

Tabella 5.3: Valori medi di temperatura per ciascuna combinazione di frequenza, duty cycle e configurazione del sensore.

Configurazione	300 Hz	150 Hz	50 Hz
Duty Cycle = 75%			
Fronte	22.23°C	21.78°C	21.86°C
Taglio	21.74°C	21.72°C	21.41°C
Duty Cycle = 50%			
Fronte	22.65°C	21.83°C	22.00°C
Taglio	23.09°C	22.28°C	22.17°C
Duty Cycle = 25%			
Fronte	22.33°C	22.39°C	25.28°C
Taglio	22.38°C	22.54°C	24.76°C

Si può quindi dedurre che, nel presente setup, l'orientazione del sensore abbia un'influenza secondaria rispetto ai parametri di movimentazione della valvola e ai fenomeni di inerzia meccanica ed elettrica associati al suo funzionamento. I risultati esposti in Tabella 5.3 mostrano una variabilità significativa, indicando che il comportamento termico del getto è fortemente determinato dall'interazione tra frequenza, duty cycle e dinamica della valvola.

5.3 Analisi dell'intrusività tramite anemometria a filo caldo

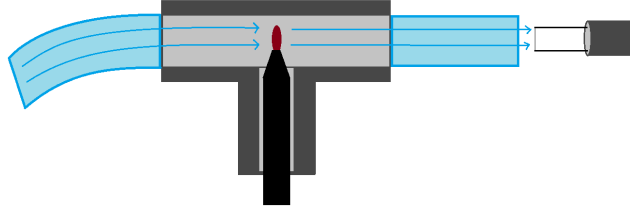


Figura 5.5: Posizionamento dell'hot-wire in uscita dal condotto di 4 mm per l'analisi di intrusività del sensore.

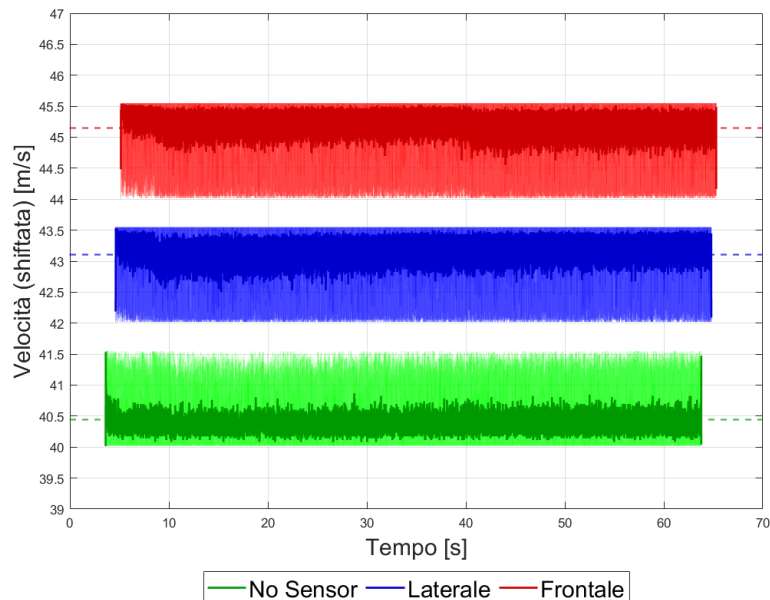


Figura 5.6: Confronto velocità in uscita dal condotto nelle tre disposizioni del termistore, segnali grezzi e mediati.

Nel confronto tra i segnali di velocità acquisiti, rappresentati in Figura 5.6 con un offset costante (0, 2, 4 m/s) per motivi di leggibilità, si nota che il posizionamento frontale produce un valore medio leggermente superiore rispetto alla disposizione totale e al caso senza sensore. Questi andamenti sono coerenti con la maggiore area esposta e l'incremento di intrusività del sensore nel flusso.

Tabella 5.4: Velocità media misurata con Hot-Wire all'uscita del condotto

Configurazione	Velocità media [m/s]
No Sensor	40.446
Laterale	41.107
Frontale	41.145

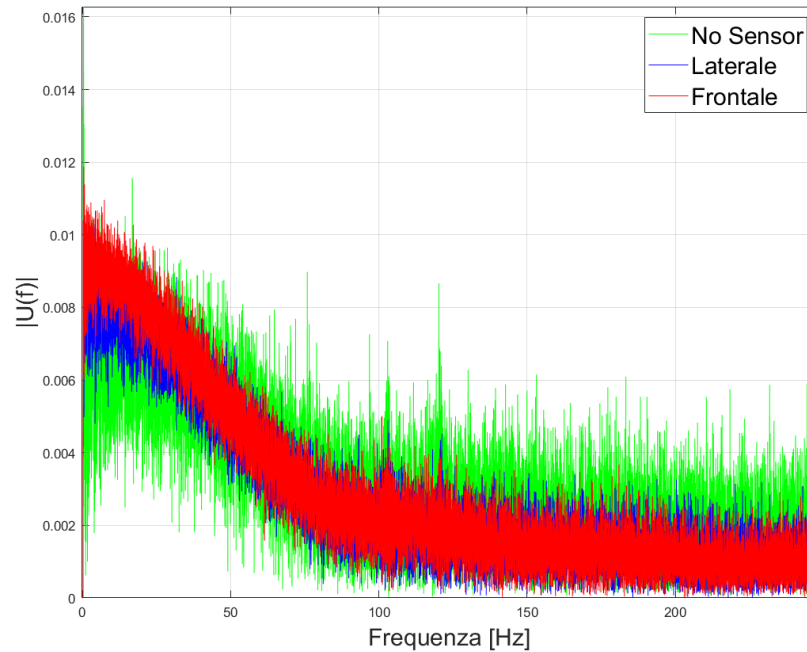


Figura 5.7: Confronto degli spettri di velocità nelle tre disposizioni del termistore.

L'andamento spettrale mostra una sovrapposizione quasi completa per le tre configurazioni, indicando che l'inserimento del sensore non modifica il contenuto energetico del flusso. Un'analisi più dettagliata dello spettro alle basse frequenze ($f < 50 \text{ Hz}$) mostra che la configurazione frontale del sensore presenta valori di $|U(f)|$ leggermente maggiori rispetto sia al caso senza sensore, sia alla disposizione laterale. Ancora una volta, tale comportamento è riconducibile alla maggiore area esposta, che genera perturbazioni più di ampia scala, associate alle basse frequenze. Per quanto il contenuto energetico nella configurazione laterale risulti inferiore rispetto al caso frontale, esso rimane comunque superiore a quello misurato in assenza del sensore. Questo indica che, per quanto contenuta, anche la disposizione laterale introduce una perturbazione del getto.

Capitolo 6

Conclusioni

L'obiettivo principale di questo lavoro di tesi era lo sviluppo di un sistema di misura puntuale della temperatura di un getto pulsato in configurazioni JICF. Tale esigenza nasce dal crescente interesse verso l'impiego di tecniche di controllo attivo dei flussi per l'incremento dello scambio termico. Di conseguenza, la possibilità di rilevare in modo affidabile e rapido le variazioni di temperatura del getto diventa un requisito essenziale per analizzare l'efficienza dell'attuazione.

Il lavoro svolto ha portato al raggiungimento di tutti gli obiettivi prefissati. In primo luogo, sono stati selezionati e calibrati tre termistori NTC, individuati come la soluzione più adatta grazie alla loro elevata sensibilità, al ridotto ingombro e alla rapidità di risposta. La calibrazione, eseguita mediante bagno termostatico, ha permesso di ottenere con elevata precisione i coefficienti della relazione di Steinhart–Hart, necessari per la conversione dei valori di resistenza in temperatura. Durante questa fase è stato evidenziato un aspetto rilevante: la calibrazione risulta strettamente dipendente dal sistema di acquisizione utilizzato.

Successivamente, i termistori sono stati integrati all'interno del canale in pressione mediante un supporto pneumatico-meccanico dedicato, progettato per ospitare in maniera intercambiabile i tre sensori calibrati. È stata inoltre sviluppata l'intera catena di acquisizione, comprensiva del codice di lettura, della gestione e salvataggio dei dati e degli strumenti di post-processing. Il sistema ottenuto è risultato compatto, ripetibile e sufficientemente robusto da operare nelle condizioni sperimentali del setup JICF.

Le prove con valvola pulsata hanno evidenziato la forte dipendenza della temperatura misurata dai parametri di funzionamento dell'attuatore, in particolare

dalla combinazione tra frequenza, duty cycle e dinamica della valvola stessa. Infine, l'analisi di intrusività tramite anemometria a filo caldo ha confermato le ipotesi, evidenziando come l'inserimento del sensore alteri il profilo di velocità nel caso frontale rispetto a quello laterale e alla configurazione senza sensore.

Il lavoro svolto apre la strada ad ulteriori approfondimenti e miglioramenti. In particolare, futuri sviluppi potrebbero includere:

- una caratterizzazione più dettagliata del comportamento elettrico del sistema di acquisizione, con l'obiettivo di comprendere meglio le differenze tra Arduino e la DAQ e di uniformare il processo di calibrazione. Tra le soluzioni da esplorare rientrano l'inserimento di un condensatore sul nodo di misura oppure l'implementazione di un circuito operativo in grado di incrementare l'impedenza di ingresso vista dal partitore;
- un'analisi più approfondita della relazione tra frequenza di pulsazione e duty cycle, per comprendere come la loro combinazione influenzi la temperatura;
- caratterizzazione sperimentale e numerica degli effetti dei ritardi in apertura e chiusura della valvola;
- inserimento di una camera di calma a temperatura controllata a valle dell'impianto, in modo da garantire che le variazioni temperatura siano attribuibili solo all'attuazione della valvola. In questo modo si eliminano i possibili effetti termici legati al compressore e alla pipeline fino alla valvola.

Appendice A

Codice MATLAB per calcolare i coefficienti di calibrazione

```
1 T1 = 15+273.15;
2 T2 = 30+273.15;
3 T3 = 50+273.15;
4 syms A B C
5 %vettore resistenza
6 R_vettore = linspace(2000, 35000, 100000);
7 idx = 1;
8
9 %%--- Misura a 15 C ---
10 lettura10 = fopen('CalibrazioneTripla_15_prova.txt');
11 andamento15 = fscanf(lettura10, '%g%g', [7, inf]);
12 tempo15 = andamento15(1,:);
13 fclose(lettura10);
14
15 %%--- Misura a 30 C ---
16 lettura30 = fopen('CalibrazioneTripla_30_prova.txt');
17 andamento30 = fscanf(lettura30, '%g%g', [7, inf]);
18 tempo30 = andamento30(1,:);
19 fclose(lettura30);
20
21 %%--- Misura a 50 C ---
22 lettura50 = fopen('CalibrazioneTripla_50_prova.txt');
23 andamento50 = fscanf(lettura50, '%g%g', [7, inf]);
24 tempo50 = andamento50(1,:);
```

```

25 fclose(lettura50);
26
27 % Array per salvare coefficienti
28 coefficienti_media = zeros(3, 3);      % colonne: A, B, C
29 coefficienti_mediamobile = zeros(3, 3);
30 coefficienti_media_cut = zeros(3, 3);
31 coefficienti_cut = zeros(3, 3);
32 colori = ['b', 'g', 'r'];
33 legenda = {};
34
35 figure
36 hold on
37 grid on
38
39 for i = 2:2:6
40
41 % Estrai i dati di ciascun sensore
42 Rth_mm_15 = andamento15(i,:);
43 Rth_15 = andamento15(i+1,:);
44 Rth_15_cut = andamento15(i+1, 2650:end);
45 media_15_Rth_mm = mean(Rth_mm_15);
46 media_15_Rth = mean(Rth_15);
47 media_15_Rth_cut = mean(Rth_15_cut);
48
49 Rth_mm_30 = andamento30(i,:);
50 Rth_30 = andamento30(i+1,:);
51 Rth_30_cut = andamento30(i+1, 2650:end);
52 media_30_Rth_mm = mean(Rth_mm_30);
53 media_30_Rth = mean(Rth_30);
54 media_30_Rth_cut = mean(Rth_30_cut);
55
56 Rth_mm_50 = andamento50(i,:);
57 Rth_50 = andamento50(i+1,:);
58 Rth_50_cut = andamento50(i+1, 2650:end);
59 media_50_Rth_mm = mean(Rth_mm_50);
60 media_50_Rth = mean(Rth_50);
61 media_50_Rth_cut = mean(Rth_50_cut);

```

```

62
63 % Coefficienti - Resistenza media
64 eq1 = A + B*log(media_15_Rth) + C*(log(media_15_Rth))^3
    == 1/T1;
65 eq2 = A + B*log(media_30_Rth) + C*(log(media_30_Rth))^3
    == 1/T2;
66 eq3 = A + B*log(media_50_Rth) + C*(log(media_50_Rth))^3
    == 1/T3;
67 sol = solve([eq1, eq2, eq3], [A, B, C]);
68 a = double(sol.A); b = double(sol.B); c = double(sol.C);
69 coefficienti_media(idx, :) = [a, b, c];
70
71 % Coefficienti - Media mobile
72 eq1mm = A + B*log(media_15_Rth_mm) + C*(log(
    media_15_Rth_mm))^3 == 1/T1;
73 eq2mm = A + B*log(media_30_Rth_mm) + C*(log(
    media_30_Rth_mm))^3 == 1/T2;
74 eq3mm = A + B*log(media_50_Rth_mm) + C*(log(
    media_50_Rth_mm))^3 == 1/T3;
75 solmm = solve([eq1mm, eq2mm, eq3mm], [A, B, C]);
76 a_RMM = double(solmm.A); b_RMM = double(solmm.B); c_RMM
    = double(solmm.C);
77 coefficienti_mediamobile(idx, :) = [a_RMM, b_RMM, c_RMM
    ];
78
79 % Coefficienti - Resistenza media tagliata
80 eq1mm = A + B*log(media_15_Rth_cut) + C*(log(
    media_15_Rth_cut))^3 == 1/T1;
81 eq2mm = A + B*log(media_30_Rth_cut) + C*(log(
    media_30_Rth_cut))^3 == 1/T2;
82 eq3mm = A + B*log(media_50_Rth_cut) + C*(log(
    media_50_Rth_cut))^3 == 1/T3;
83 solmm = solve([eq1mm, eq2mm, eq3mm], [A, B, C]);
84 a_cut = double(solmm.A); b_cut = double(solmm.B); c_cut
    = double(solmm.C);
85 coefficienti_media_cut(idx, :) = [a_cut, b_cut, c_cut];
86

```

```

87 % Curve T(R)
88 T = 1./(a + b.*log(R_vettore) + c.*(log(R_vettore)).^3)
    - 273.15;
89 T_RMM = 1./(a_RMM + b_RMM.*log(R_vettore) + c_RMM.*(log(
    R_vettore)).^3) - 273.15;
90 T_cut = 1./(a_cut + b_cut.*log(R_vettore) + c_cut.*(log(
    R_vettore)).^3) - 273.15;
91
92 % Plot di ciascun sensore sul grafico finale
93 plot(T, R_vettore, colori(idx), 'LineWidth', 1.5)
94 legenda{end+1} = ['S', num2str(idx), ' - media'];
95
96 plot(T_RMM, R_vettore, ['k', '--'], 'LineWidth', 1.5)
97 legenda{end+1} = ['S', num2str(idx), ' - media mobile'];
98
99 plot(T, R_vettore, 'y', 'LineWidth', 0.75)
100 legenda{end+1} = ['S', num2str(idx), ' - cut'];
101
102 idx = idx+1;
103 end
104 xlabel('Temperatura [ C ]', 'FontSize', 22)
105 ylabel('Resistenza [\Omega]', 'FontSize', 22)
106 title('Curve T(R)', 'FontSize', 24)
107 legend(legenda, 'FontSize', 18)
108 grid on

```

Appendice B

Codice Arduino per l'acquisizione delle temperature

Il codice consente l'acquisizione della tensione ai capi del termistore e, tramite un'attenta calibrazione, il calcolo della temperatura corrispondente.

```
1 #include <SPI.h> // Libreria per comunicazione SPI
   (necessaria per SD)
2 #include <SD.h> // Libreria per gestione scheda
   SD
3 #include <RTC.h> // Libreria per RTC interno
   d e l l Arduino UNO R4
4 #include <WiFiS3.h> // Libreria per connessione WiFi
   del modulo S3
5 #include <WiFiUdp.h> // UDP per comunicazioni NTP
6 #include <NTPClient.h> // Libreria client NTP
7 #include <time.h> // Funzioni per gestire tempo
   Unix
8
9 // === WiFi e NTP ===
10 char SSID[] = "Matilde"; // Nome rete WiFi
11 char PASS[] = "pass060804"; // Password WiFi
12 WiFiUDP ntpUDP; // Oggetto UDP per protocollo NTP
13 NTPClient timeClient(ntpUDP, "pool.ntp.org", 7200, 60000); //
   Client NTP con offset +2h
14 unsigned long lastSyncTime = 0; // Ultima
   sincronizzazione eseguita
15 const unsigned long syncInterval = 5 * 60 * 1000; // Intervallo
   sincronizzazione: 5 minuti
16
17 // === SD ===
18 const int chipSelect = 10; // Pin di selezione della SD
19 File dataFile; // Oggetto file per la scrittura
20
21 // === BUFFER TRIPLIO ===
22 const int bufferSize = 3072; // Dimensione per buffer
23 const int numBuffer = 3; // Numero di buffer
24 char buffers[numBuffer][bufferSize]; // Matrice dei buffer
25 int bufferIndex = 0; // Indice di scrittura (non
   utilizzato direttamente)
26 int used[numBuffer] = {0}; // Byte usati in ogni buffer
27 int currentBuffer = 0; // Buffer attualmente in
```

```

    acquisizione
28 volatile bool writing = false; // Flag:    in corso una
    scrittura SD?
29 int writeBuffer = -1;           // Buffer da scrivere su SD
30
31 // === Termistore ===
32 int ThermistorPin = A0;         // Pin analogico del termistore
33 float V0, V0_average;          // Tensioni istantanea e mediata
34 #define V1 3.3                 // Tensione di riferimento ADC
35 float R = 9990;                // Resistenza del partitore
36 float Rth, Rth_average;        // Resistenze termistore (singola
    e mediata)
37 float logRth, T, TC;          // Variabili termiche
38
39 // === Coefficienti Steinhart-Hart ===
40 #define c1 -0.0041852
41 #define c2 0.0009805
42 #define c3 -2.2082e-06
43
44 // === Media mobile ===
45 const int N = 10;              // Numero campioni media mobile
46 float bufferMed[N];           // Buffer circolare per la media
47 int k = 0;                    // Numero di campioni acquisiti
48 int idx = 0;                  // Indice buffer media
49 float average = 0;            // Media risultante
50
51 // === Timer ===
52 #define durata 60000           // Durata totale acquisizione = 1
    minuto
53 unsigned long tempoInizio;     // Tempo inizio acquisizione
54 unsigned long tempoTrascorso;  // Tempo attuale - inizio
55
56 // === RTC + NTP sincrono ===
57 unsigned long startUnixTime = 0; // Tempo Unix d e l l inizio
    acquisizione
58
59 // Sincronizza RTC tramite server NTP
60 void sincronizzaRTCdaNTP() {
61     Serial.println("Sincronizzazione RTC da NTP...");
62     timeClient.update();        // Aggiorna client NTP
63     unsigned long unixTime = timeClient.getEpochTime(); // Ottiene
    tempo Unix
64     RTCTime rtcTime(unixTime);  // Converte in oggetto
    RTC
65     RTC.setTime(rtcTime);       // Imposta RTC hardware
66 }
67
68 // Scrive un buffer su SD (funzione bloccante, ma usata solo su
    buffer "libero")
69 void scriviBufferSuSD(int idx, int len) {
70     if (len <= 0) return;       // Nulla da scrivere
71     dataFile = SD.open("dati.txt", FILE_WRITE); // Apre file
72     if (dataFile) {
73         dataFile.write((uint8_t*)buffers[idx], len); // Scrive dati
    binari
74         dataFile.close();       // Chiude
75         Serial.print("Buffer ");
76         Serial.print(idx);
77         Serial.println(" scritto su SD");
78     } else {

```



```

79     Serial.println("Errore apertura file SD");
80 }
81 used[idx] = 0; // Reset utilizzo buffer
82 writing = false; // Segnala che non si sta
    pi scrivendo
83 writeBuffer = -1; // Nessun buffer in coda
84 }
85
86 // Salva una riga nel buffer attivo
87 void salvaRigaNelBuffer(const char* riga, int n) {
88     // Se c'è spazio nel buffer corrente
89     if (used[currentBuffer] + n < bufferSize) {
90         memcpy(buffers[currentBuffer] + used[currentBuffer], riga, n
91         );
92         used[currentBuffer] += n;
93     } else {
94         // Buffer pieno va scritto
95         if (!writing) {
96             writing = true;
97             writeBuffer = currentBuffer;
98         }
99
100         // Passa al buffer successivo
101         currentBuffer = (currentBuffer + 1) % numBuffer;
102
103         // Se pieno rischio perdita dati
104         if (used[currentBuffer] != 0) {
105             Serial.println("Tutti i buffer occupati, perdita dati
106             possibile!");
107             used[currentBuffer] = 0; // Reset forzato per evitare
108             blocco
109         }
110
111         // Salva nel nuovo buffer
112         memcpy(buffers[currentBuffer], riga, n);
113         used[currentBuffer] = n;
114     }
115 }
116
117 // Gestisce scrittura SD in modo non bloccante
118 void serviceSD() {
119     if (writing && writeBuffer >= 0) {
120         scriviBufferSuSD(writeBuffer, used[writeBuffer]);
121     }
122 }
123
124 void setup() {
125     Serial.begin(230400); // Baud rate alto per
126     debug veloce
127     analogReadResolution(14); // Risoluzione ADC a 14
128     bit
129     tempoInizio = millis(); // Segna inizio
130     acquisizione
131
132     // == WiFi ==
133     // Connessione al WiFi, attualmente disattivata
134     int status = WL_IDLE_STATUS;
135     while (status != WL_CONNECTED) {
136         Serial.print("Connessione a: ");
137         Serial.println(SSID);

```

```

132     status = WiFi.begin(SSID, PASS);
133     delay(5000);
134 }
135 Serial.println("Connesso al WiFi");
136
137 // === RTC e NTP ===
138 RTC.begin(); // Avvia RTC
139 timeClient.begin(); // Avvia NTP
140 sincronizzaRTCdaNTP(); // Sincronizza orologio
141 lastSyncTime = millis(); // Segna momento
142     sincronizzazione
143
144 RTCTime startTime; // Ottiene tempo attuale
145 RTC.getTime(startTime);
146 startUnixTime = startTime.getUnixTime(); // Salva tempo di
147     partenza
148
149 // === SD ===
150 Serial.print("Inizializzazione SD...");
151 pinMode(chipSelect, OUTPUT);
152 if (!SD.begin(chipSelect)) {
153     Serial.println("Errore inizializzazione SD");
154     while (true); // Blocco in caso di
155     errore
156 }
157 Serial.println("Fatto!");
158
159 // === Intestazione file ===
160 SD.remove("dati.txt"); // Cancella eventuale file
161     precedente
162 dataFile = SD.open("dati.txt", FILE_WRITE);
163 if (dataFile) {
164     dataFile.println("Timestamp\t Temp ( C )"); // Scrive header
165     dataFile.close();
166     Serial.println("File di scrittura aperto correttamente!");
167 } else {
168     Serial.println("Errore apertura file di scrittura");
169     while (true);
170 }
171
172 delay(5000); // Pausa prima di iniziare
173 tempoInizio = millis(); // Reset tempo inizio
174 }
175
176 void loop() {
177     tempoTrascorso = millis() - tempoInizio; // Tempo
178     da l l inizio
179
180     // Se abbiamo raggiunto la durata totale
181     if (tempoTrascorso >= durata) {
182         // Scrive eventuali buffer incompleti
183         for (int i = 0; i < numBuffer; i++) {
184             if (used[i] > 0) scriviBufferSuSD(i, used[i]);
185         }
186         Serial.println("Misurazione completata!");
187         while (true); // Stop esecuzione
188     }
189
190     // === Sincronizzazione periodica NTP ===
191     if (millis() - lastSyncTime >= syncInterval) {

```

```

187     sincronizzaRTCdaNTP();
188     RTCTime startTime;
189     RTC.getTime(startTime);
190     startUnixTime = startTime.getUnixTime();
191     lastSyncTime = millis();
192 }
193
194 // === Lettura e media mobile ===
195 int read = analogRead(ThermistorPin); // Lettura ADC
196 bufferMed[idx] = read;
197 idx = (idx + 1) % N;
198 k++;
199
200 float sum = 0;
201 for (int i = 0; i < N; i++) sum += bufferMed[i];
202 average = (k < N) ? sum / idx : sum / N; // Media mobile
203
204 // === Calcolo delle resistenze ===
205 V0_average = (V1 / 16383.00) * average; // Scala A D C Volt
206 Rth_average = R * (V1 / V0_average - 1.00);
207
208 V0 = (V1 / 16383.00) * read;
209 Rth = R * (V1 / V0 - 1.00);
210
211 // === Calcolo temperatura ===
212 logRth = log(Rth);
213 T = 1.0 / (c1 + c2 * logRth + c3 * pow(logRth, 3)); //
214 //      Steinhart-Hart
215 TC = T - 273.15; //
216 //      K e l C
217
218 // === Timestamp ===
219 unsigned long currentUnixTime = startUnixTime + tempoTrascorso
220 / 1000;
221 unsigned long millisPart = tempoTrascorso % 1000;
222 RTCTime nowTime(currentUnixTime);
223
224 // Formattazione timestamp con millisecondi
225 char timestamp[30];
226 sprintf(timestamp, "%02d:%02d:%02d.%03lu",
227         nowTime.getHour(), nowTime.getMinutes(), nowTime.
228         getSeconds(),
229         millisPart);
230
231 // === Riga da salvare ===
232 char riga[60];
233 int n = snprintf(riga, sizeof(riga), "%s\t%.2f\r\n", timestamp
234 , TC);
235
236 // === Scrive nel buffer (non blocca) ===
237 salvaRigaNelBuffer(riga, n);
238
239 // === Scrittura SD asincrona ===
240 serviceSD();
241
242 // Frequenza di campionamento      200 Hz
243 delay(5);
244 }

```

Listing B.1: Script completo Arduino

Bibliografia

- [1] L. Magnani, *Setup and validation of a pulsed jet for turbulent boundary layer active control.*, Università di Bologna, Tesi di Laurea Magistrale, Corso di Studio in Aerospace engineering, 2023.
- [2] Jacob Fraden. *Handbook of Modern Sensors: Physics, Designs, and Applications*.
4th ed., Springer, 2010.
- [3] Guido Buresti. *Elements of Fluid Dynamics*.
Imperial College Press, 2012.
- [4] Stefano Discetti and Andrea Ianiro, eds. *Experimental Aerodynamics*.
Boca Raton: CRC Press, 2017.
- [5] *Come rilevare accuratamente la temperatura utilizzando i termistori*.
<https://www.thermistor.com/thermistors/ntc-thermistors/>
- [6] *How Easy Is It to Use a Thermistor?!*
[https://projecthub.arduino.cc/Marcazzan_M/
how-easy-is-it-to-use-a-thermistor-81ae74](https://projecthub.arduino.cc/Marcazzan_M/how-easy-is-it-to-use-a-thermistor-81ae74)
- [7] L. Villafañe, G. Paniagua, *Aero-thermal analysis of shielded fine wire thermo-couple probes*.
International Journal of Thermal Sciences, Vol. 65, 2013, pp. 214–223.
- [8] *Arduino Project Hub*
<https://projecthub.arduino.cc/>
- [9] Sarah Cleve, Emmanuel Jondeau, Philippe Blanc-Benon, and Geneviève Comte-Bellot, *cold-wire constant voltage anemometry to measure temperature fluctuations and its application in a thermoacoustic system*.
Published by the American Institute of Physics, 2017

- [10] Y. Nagano and T. Tsuji, *Recent developments in hot- and cold-wire techniques for measurements in turbulent shear flows near walls*. Experimental Thermal and Fluid Science, Aug. 1994.
- [11] L. Miliani, *Arduino. La guida essenziale 2.0: La scheda, il linguaggio, le librerie di sistema, le nozioni base di elettronica, le tecniche avanzate*, Edizioni LSWR, 2^a edizione, 2022.
- [12] Tom's Hardware, *Corso base di Arduino: L'acquisizione di segnali analogici*. Disponibile online: <https://www.tomshw.it/altro/corso-base-di-arduino-lacquisizione-di-segnali-analogici>
- [13] R. Castellanos, G. Salih, M. Raiola, A. Ianaro, S. Discetti, *Heat transfer enhancement in turbulent boundary layers with a pulsed slot jet in crossflow*, International Journal of Heat and Mass Transfer, Vol. 163, 2020, Articolo n. 120511.
- [14] *Impedenza ingresso analogico Arduino*. <https://forum.arduino.cc/t/impedenza-ingresso-analogico-arduino/344257>
- [15] *Soluzioni per problematiche di impedenza su ADC di Arduino* https://www.reddit.com/r/arduino/comments/1d9mbg/can_i_increase_the_input_impedance_of_the_adc/?tl=it