

**Studio e programmazione di un sistema
di acquisizione e controllo per gallerie del vento**

Candidati:

Giuseppe Borrelli

Matricola 0000793301

Giovanni Luciani

Matricola 0000806460

Relatore:

Ch.mo Prof. Gabriele Bellani

Correlatore:

Prof. Lucia Mascotelli

Ai nostri genitori

Sommario/Abstract

Il seguente elaborato si colloca all'interno dello studio del moto turbolento, in cui è specializzato il laboratorio CICLoPE (Center for International Cooperation in Long Pipe Experiments).

Il principale obiettivo è stato quello di creare un programma in grado di misurare la pressione e di controllarne l'acquisizione tramite l'utilizzo di una particolare strumentazione che consentisse di ottenere risultati affetti da incertezze di entità sempre minore. Questa necessità è dovuta al forte legame che ha la caduta di pressione, lungo un condotto, con le grandezze caratteristiche della turbolenza, quali velocità ed attrito, per esempio.

Gli strumenti utilizzati finora presentavano errori variabili per ogni punto lungo il pipe, riconducendo il tutto ad un'alta inesattezza nella misurazione della pressione. E' stato dunque necessario sviluppare un programma da adattare ad un nuovo sistema di acquisizione, che comprendesse anche la movimentazione di un connettore circolare (Scanivalve). In questo modo è possibile ottenere dei valori di pressione viziati da un'imprecisione minore ma, soprattutto, costante.

Una volta implementato lo schema base del programma sul software LabVIEW sono state individuate diverse ipotesi per effettuare l'acquisizione e per imprimere la rotazione alla Scanivalve.

Sono state, quindi, specificate le modalità che potrebbero essere utilizzate al centro CICLoPE per testare il programma in maniera sperimentale.

Altre modalità sono state descritte per casi più generali, in modo da poter adattare lo schema in base alle necessità ed alla strumentazione del singolo caso.

The following essay finds its application in the studies of the turbulent flow's motion, in which the CICLoPE's laboratory is specialized.

The main purpose has been to create a program that could control and measure the pressure using an instrumentation with smaller uncertainty than what is currently used in CICLoPE.. This need is due to the fact that, in turbulent pipe flows, there is a strong relationship between the pressure drop along the pipe and the typical scaling quantities of turbulence, such as friction velocities.

The tools used so far had variable errors from point to point along the pipe, having as a consequence a relevant inaccuracy on the measurement of the pressure. It has been necessary to develop an application which could adapt to the new acquisition's system, that included the movimentation of the round connector (Scanivalve), as well. In this way it is possible, ideally, to obtain the pressure's values affected from a minor and, most important, position-independent inaccuracy.

Once the implementation of the basic scheme has been done on the LabVIEW software, different solutions have been hypotized to proceed with the acquisition and to generate the rotation of the Scanivalve.

Then, one solution has been specified for the future developement at the CICLoPE's center, to experimentally validate the program.

Other options are presented as well, for more general scenarios.

Premessa

Il seguente elaborato è frutto di un progetto di tesi triennale sviluppato in collaborazione da G. Luciani e G. Borrelli.

Sebbene il percorso sia stato ideato e supervisionato da entrambi, i capitoli presentano un contributo maggiore dall'una o l'altra parte.

Per quanto riguarda la stesura dell'elaborato:

- *Introduzione*: redatto da G. Borrelli, rivisto da G. Luciani;
- Capitolo 1, *Moto Turbolento in un Condotto*: redatto da G. Borrelli, rivisto da G. Luciani;
- Capitolo 2, *Acquisizione della pressione*: redatto da G. Luciani, rivisto da G. Borrelli;
- Capitolo 3, *Analisi del sistema di acquisizione del CICLOPE*: redatto da G. Luciani, rivisto da G. Borrelli;
- Capitolo 4, *Codice LabVIEW per l'acquisizione dei dati*: redatto da G. Borrelli, rivisto da G. Luciani;
- Capitolo 5, *Sviluppi futuri e conclusioni*: redatto da G. Luciani, rivisto da G. Borrelli.

Allo stesso modo il programma di LabVIEW presenta una struttura concepita assieme. La supervisione, una prima analisi del codice e l'idea del progetto sono state effettuate in cooperazione.

La creazione dei diversi compiti è stata suddivisa come segue:

- Implementati da G. Borrelli:
 - Acquisizione del segnale simulato, grafico e coefficienti di calibrazione;
 - Scrittura della storia temporale su file *excel*;
 - Prima *case structure* per la costruzione dell'array colonna;
 - Condizione di uscita dal ciclo ed indicatore di operatività del sistema di acquisizione.
- Implementati da G. Luciani:
 - Seconda *case structure* per la costruzione dell'array completo;
 - Percorso di *storing* dell'array completo finalizzato al *post processing*;
 - *FPGA target* per l'acquisizione del segnale reale e per il *triggering*;
 - Implementazione nel VI principale dei target FPGA;

Infine, per gli script di Matlab:

- Funzione "*port_meas_corr*" di G. Luciani;
- Funzione "*mean_velocity*" di G. Borrelli.

Indice

Sommario/Abstract	v
Premessa	vii
Introduzione	1
1 Moto Turbolento in un Condotto	5
1.1 Principali caratteristiche del moto turbolento	5
1.2 Moto all'interno di un condotto	7
1.2.1 Legame tra pressioni e stress di taglio alla parete	8
1.2.2 Caratteristiche dello stress di taglio a parete	9
1.2.3 <i>Law of the Wall</i>	11
1.2.4 <i>Log Law</i> e costante di von Karman	12
2 Acquisizione della pressione	15
2.1 Sensori di pressione	15
2.1.1 Principio di funzionamento	16
2.2 Trasduttori di pressione	17
2.2.1 Trasduttori di pressione capacitivi	18
2.3 Sistema di acquisizione dati	20
2.3.1 Trattamento e acquisizione dei dati	20
2.3.2 Conversione dei dati	22
3 Analisi del sistema di acquisizione del CICLoPE	23
3.1 Sistema di misurazione attuale del centro CICLoPE	25
3.1.1 DTC Initium	25

3.1.2	Problemi relativi alla misura	27
3.2	Nuovo sistema di acquisizione della pressione	28
3.2.1	La <i>Scanivalve</i>	28
3.2.2	MKS Baratron Pressure Transducer	29
3.2.3	Funzionamento del nuovo sistema di misurazione della pressione	30
4	Codice LabVIEW per l'acquisizione dei dati	35
4.1	Schema generale del programma	35
4.1.1	Block Diagram	36
4.1.2	Front Panel	39
4.2	Acquisizione e triggering con NI-DAQmx	40
4.3	Acquisizione e triggering con FPGA	43
4.4	<i>Post-processing</i> dei dati	46
4.5	Complessità computazionale del programma	47
4.5.1	Tempi di esecuzione e <i>buffering</i>	47
4.5.2	Occupazione di memoria	49
5	Sviluppi futuri e conclusioni	51
5.1	Sviluppi futuri	51
5.2	Conclusioni	51
	Bibliografia	55
	Ringraziamenti	56
	Elenco delle figure	57
	Appendice A	63
	Appendice B	72

Introduzione

Il laboratorio CICLoPE è un centro di ricerca specializzato nello studio dei flussi turbolenti ad alti numeri di Reynolds. Questa attività è di fondamentale importanza, oltre che per il puro scopo di ricerca, anche per i diversi riscontri che si hanno nell'analisi di fenomeni naturali e nei vari ambiti ingegneristici, quali, per esempio, quello aeronautico, in cui la condizione di flusso turbolento è spesso ricorrente.

Il sito nel quale è collocato il laboratorio (Predappio) permette di ottenere condizioni stabili, e quindi ideali, di temperatura, umidità ed isolamento acustico.

Per comprendere l'importanza del laboratorio è necessario avere chiara la natura del moto turbolento. Infatti, tra i vari aspetti che lo caratterizzano, uno dei più importanti è che si sviluppi in presenza di elevati numeri di Reynolds, essendo quest'ultimo valore adimensionale il rapporto tra le forze d'inerzia e quelle viscosive che agiscono su una particella in moto all'interno di un determinato fluido.

Data la definizione di questo numero:

$$Re = \frac{UL}{\nu}$$

Si può notare come siano possibili diverse opzioni affinché il rapporto sia alto. In primo luogo si può accelerare il flusso, aumentandone la velocità U , tenendo però conto degli effetti di comprimibilità.

Un'altra opzione è quella di lavorare sulla densità ρ , termine che entra in gioco nella definizione di ν (*viscosità cinematica*): $\nu = \mu/\rho$ con μ che indica la *viscosità dinamica*. Quindi, aumentando ρ , per esempio pressurizzando il flusso, si avrebbe un aumento del Re .

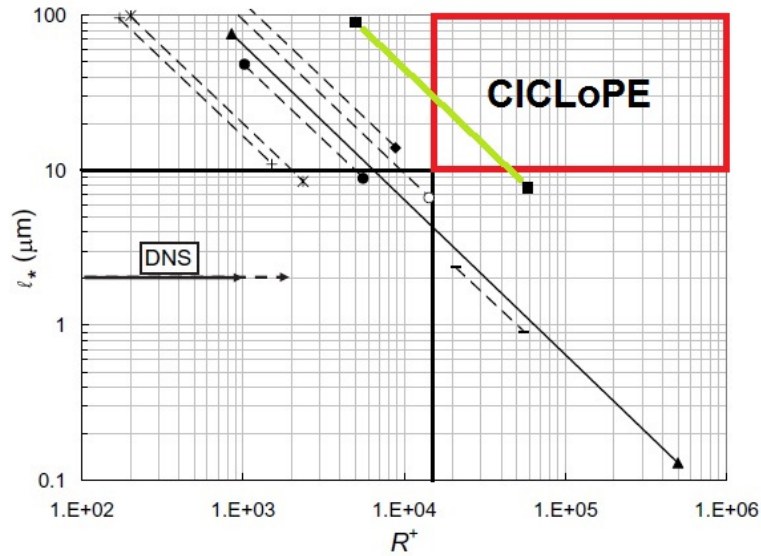


Figura 1: Grafico che mostra la risoluzione spaziale (in μm) in funzione del *friction Reynolds number* (R^+) (il significato di questa grandezza sarà discusso nel paragrafo 1.2.2) nei diversi *wind tunnels*. Evidenziata l'area di interesse del CICLoPE. Immagine del sito del laboratorio CICLoPE

Infine si può incrementare la dimensione L , cioè la lunghezza di riferimento: nel caso specifico il diametro del *pipe*. Questa è l'opzione utilizzata nel *long pipe* del CICLoPE. Infatti, si aumenta D per ottenere una buona risoluzione, la quale consenta misurazioni accurate, in modo tale da non richiedere correzioni *a posteriori*. A tal fine, la scelta di un diametro il più grande possibile permette di raggiungere elevati valori di Re .

Il successo di questi studi è quindi dovuto all'utilizzo di un *Long Pipe*, una galleria del vento a ciclo chiuso (*closed loop*), composta da un condotto di 111.5 m di lunghezza (su uno sviluppo totale di 120 m) ed avente un diametro di circa 0.9 m.

Da come verrà reso chiaro nei capitoli seguenti, la scelta di far sviluppare il flusso in un condotto è dovuta al legame diretto che hanno le tensioni di taglio con le cadute di pressione lungo il *pipe* e quindi alla capacità di ottenere informazioni in maniera precisa e diretta riguardo al comportamento del flusso turbolento nella galleria.

Al fine di raggiungere tale scopo sono stati necessari i seguenti step:

Studio del moto turbolento all'interno di un condotto

Questa prima parte, più teorica, ha permesso di far luce sulle principali grandezze tipiche del moto turbolento che sono influenzate da una più o meno precisa misurazione della pressione lungo il *pipe*. Tramite lo studio statistico delle leggi della fluidodinamica è stato possibile ricavare il legame tra le cadute di pressione lungo il condotto e le azioni di stress di taglio alla parete, per poi passare alla formulazione della *Law of the wall* e della *Log Law*.

Studio delle problematiche nella misurazione della pressione e relative incertezze

E' stato poi necessario comprendere quali fossero i problemi legati al sistema di misurazione finora utilizzato al CICLoPE. Si è osservato come lo strumento presentasse un'incertezza variabile sulla misura della pressione in ogni punto della galleria, facendone derivare un'area di imprecisione molto ampia.

Analisi preliminare di un sistema alternativo per misurare la caduta di pressione

Per ovviare al problema sono stati introdotti due nuovi strumenti: un trasduttore di pressione differenziale ed un meccanismo di movimentazione *multiplexer* circolare (*Scanivalve*). Grazie alla rotazione di quest'ultimo è stato possibile collegare diversi punti della galleria con l'unico trasduttore, ottenendo delle misure in cui l'errore rimanesse indipendente dalla porta valutata.

Codice di acquisizione dati e di movimentazione della *Scanivalve*

Al fine di acquisire le misure di pressione lungo il *pipe* è stato implementato un programma, con il software LabVIEW, che potesse ottenere questi valori dalla nuova strumentazione.

E' stato necessario implementare anche il sistema di movimentazione per la *Scanivalve*. Questo, tramite un output di voltaggio, permette di far ruotare il meccanismo una volta acquisito un numero di campioni che viene reputato sufficiente per ogni singolo punto della galleria,

Capitolo 1

Moto Turbolento in un Condotto

Sin dai tempi dell'esperimento eseguito da O. Reynolds (1883) lo studio del comportamento di un flusso lungo un *pipe* (condotto) ha avuto un ruolo fondamentale per l'analisi riguardanti lo sviluppo del moto turbolento.

E' infatti semplice, grazie a questa particolare architettura, misurare la caduta di pressione presente in un *pipe* in presenza di un flusso turbolento completamente sviluppato e, conseguentemente, le caratteristiche di velocità ed attrito che lo caratterizzano.

Questo capitolo ha come obiettivi quello di introdurre i concetti fondamentali per comprendere questi legami e di chiarire l'importanza che riveste un sistema di acquisizione e controllo della pressione per gallerie del vento.

1.1 Principali caratteristiche del moto turbolento

Il *moto turbolento* può essere osservato in natura in numerosi fenomeni quotidiani, quali l'uscita del fumo di una sigaretta o la distribuzione dell'acqua che scorre in un corso d'acqua una volta che impatta una nave, per esempio. Le principali caratteristiche di questo moto sono quelle di essere non stazionario, irregolare e caotico, rendendo impossibile la predizione della posizione che avrà ogni particella del fluido in un preciso istante. Infatti, dal confronto con il più regolare *moto laminare*, quello turbolento presenta maggiori capacità di trasporto e quindi di *diffusività* del fluido, presentando un alto contenuto *vorticoso*, il quale è la principale caratteristica di questo tipo di moto. Ciò significa aumentare la capacità di scambio di quantità di moto e

di energia tra le particelle, scambio che si ha a livello molecolare per quanto riguarda il caso laminare (i fenomeni avvengono su scale con diversi ordini di grandezza). Per questo motivo e da come si può notare dai risultati sperimentali presenti in figura 1.1 gli studi possono essere condotti unicamente in maniera statistica.

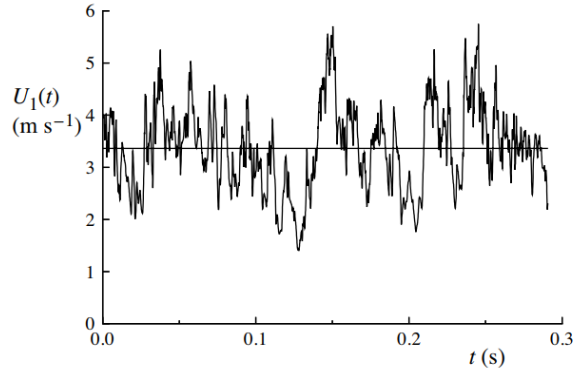


Figura 1.1: Storia temporale di una componente assiale di velocità di un getto turbolento. Immagine dell'esperimento di Tong e Warhaft (1995) presa da 'Turbulent Flows', di Stephen B. Pope (2000)

Lo studio statistico di una grandezza (quale la velocità, nel caso analizzato) si basa sulla decomposizione di Reynolds:

$$U(t) = \bar{U} + u'(t) \quad (1.1)$$

in cui $u'(t)$ indica l'oscillazione che si ha al tempo t e \bar{U} indica la grandezza media data dall'equazione:

$$\bar{U} = \frac{1}{T} \int_0^T U(t) dt \quad (1.2)$$

Gli intervalli di tempo T in cui vengono valutati i valori medi sono sufficientemente lunghi in modo tale da consentire un'analisi statisticamente stazionaria del fenomeno turbolento.

Ogni flusso è, inoltre, descritto da un parametro adimensionale che prende il nome di *numero di Reynolds* ed è definito dalla formula

$$Re = \frac{UL}{\nu} \quad (1.3)$$

in cui U ed L sono le grandezze di scala caratteristiche e ν è la viscosità cinematica. Nel caso di un condotto, per valori di $Re < 2300$ il moto è lami-

nare mentre per Re superiori a 4000 si ha moto turbolento (questo numero è indicativo in base ai dettagli dell'esperimento). Nella zona compresa tra i due range si ha il fenomeno della transizione.

1.2 Moto all'interno di un condotto

Al contrario di molti flussi laminari, quelli turbolenti sono spesso vincolati dalla presenza di superfici solide quali quelle di velivoli o navi, nel caso di flussi esterni, oppure da condotti o canali, nel caso di flussi interni.

Si definiscono δ il raggio interno del *pipe*, y la coordinata verticale, lungo la quale si hanno le variazioni di velocità media, x quella assiale, lungo la quale si sviluppa il flusso medio, ed r la coordinata radiale, utile per descrivere la distanza alla quale si è dalla *centerline* (si veda figura 1.2).

Le tre coordinate della velocità sono espresse dalle componenti (U, V, W) , con (u', v', w') le rispettive oscillazioni.

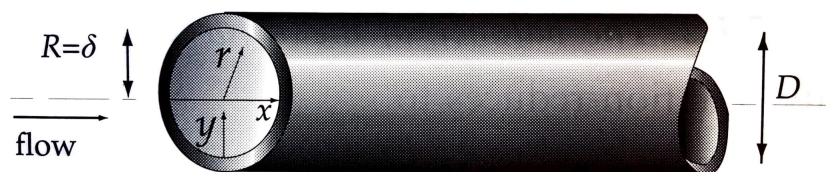


Figura 1.2: Schema di un *pipe*. Immagine presa da 'Turbulent Flows', di Stephen B. Pope (2000)

Analizzando lo sviluppo di un flusso si può notare come questo, a partire da una velocità uniforme con la quale entra nel condotto, venga ad assumere un profilo diverso nella sezione e durante la sua evoluzione assiale. Ciò è dovuto principalmente alla presenza dello strato limite ed ai fenomeni di attrito alla parete, con effetto sempre maggiore mentre ci si allontana dalla zona centrale in cui scorre il fluido (*centerline*).

Come si può vedere in figura 1.3 si ha una prima parte che viene definita *regione di entrata* ed una parte, più a valle, di *flusso completamente sviluppato*, in cui il profilo di velocità assume la tipica forma parabolica di *Poiseuille* ed in cui la velocità, statisticamente, non varia più con la coordinata x . Per arrivare a quest'ultima condizione bisognerà soddisfare l'ipotesi che L , la lunghezza in direzione assiale del condotto, sia tale da soddisfare la relazione $L/\delta \gg 1$.

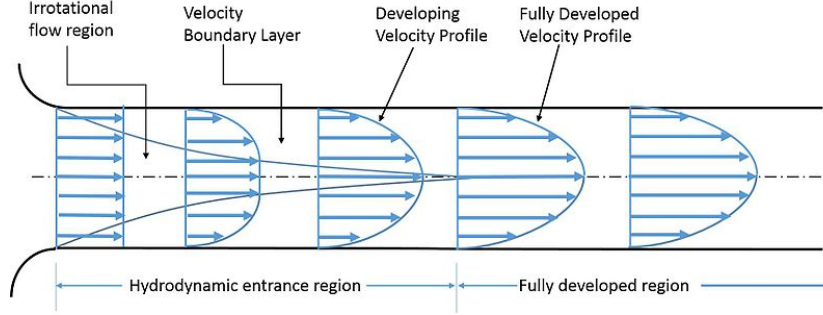


Figura 1.3: Andamento del profilo medio di velocità di un flusso lungo un condotto. Evidenziate la regione di ingresso e la regione di flusso completamente sviluppato

1.2.1 Legame tra pressioni e stress di taglio alla parete

Dallo studio dell'attrito nella zona in cui il flusso è completamente sviluppato ci si può ricondurre alla relazione che lega la misurazione della pressione e la componente di taglio a parete (τ_w).

Lo stress di taglio totale $\tau(y)$ viene definito come:

$$\tau = \rho\nu \frac{d\bar{U}}{dy} - \rho(\bar{u}'\bar{v}') \quad (1.4)$$

In cui il primo termine dell'equazione indica lo *stress viscoso* ed il secondo lo *stress di Reynolds* (componente che si annulla a parete, data la condizione al contorno per cui $U(x, t) = 0$).

Dall'equazione in direzione laterale della quantità di moto (media):

$$0 = -\frac{d}{dy} \bar{v}'^2 - \frac{1}{\rho} \frac{\delta \bar{p}}{\delta y} \quad (1.5)$$

si può ricavare, integrando ed applicando la condizione al contorno $\bar{v}'^2_{y=0}=0$, la seguente equazione:

$$\bar{v}'^2 + \frac{\bar{p}}{\rho} = \frac{p_w}{\rho} \quad (1.6)$$

in cui p_w rappresenta la pressione media che si ha alla parete.

Poichè la componente \bar{v}' delle oscillazioni è indipendente rispetto alla coordinata x , si può dedurre che il gradiente medio di pressione assiale è uniforme attraverso il fluido e quindi che:

$$\frac{\delta \bar{p}}{\delta x} = \frac{dp_w}{dx} \quad (1.7)$$

Infine, avendo l'equazione in direzione assiale della quantità di moto (media):

$$0 = \nu \frac{d^2 \bar{U}}{dy^2} - \frac{d}{dy} \bar{u}' \bar{v}' - \frac{1}{\rho} \frac{\delta \bar{p}}{\delta x} \quad (1.8)$$

utilizzando le equazioni (1.4) e (1.7) si ottiene:

$$\frac{d\tau}{dy} = \frac{dp_w}{dx} \quad (1.9)$$

L'equazione (1.9) rappresenta a tutti gli effetti un bilancio di forze in cui il gradiente degli stress di taglio del flusso attraverso la sezione equilibra il gradiente degli stress normali lungo la direzione assiale.

Data la dipendenza di $\tau(y)$ dalla sua posizione radiale, la componente in corrispondenza di $y = 0$ può essere scritta come $\tau_w = \tau(0)$, ed integrando l'equazione (1.9) tra 0 e δ , la si può esplicitare in funzione del valore dell'azione di taglio alla parete (*wall shear stress*), ottenendo il legame che quest'ultimo ha con la caduta di pressione lungo il *pipe*:

$$-\frac{dp_w}{dx} = \frac{\tau_w}{\delta} \quad (1.10)$$

L'equazione così ottenuta risulta indipendente dalle proprietà del fluido (cioè ρ e ν) e dallo stato di moto del fluido (laminare o turbolento) ed inoltre permette di definire, una volta determinata τ_w , i coefficienti di attrito (*skin friction coefficient*) adimensionalizzati rispetto a \bar{U} ed U_0 (velocità alla *centerline*):

$$c_f = \frac{\tau_w}{\frac{1}{2} \rho U_0^2}, \quad C_f = \frac{\tau_w}{\frac{1}{2} \rho \bar{U}^2}$$

1.2.2 Caratteristiche dello stress di taglio a parete

Come già introdotto nel paragrafo precedente, per la condizione al contorno alla parete, lo stress di taglio è dato dall'equazione:

$$\tau_w = \rho \nu \left(\frac{d\bar{U}}{dy} \right)_{y=0} \quad (1.11)$$

Per cui, vicino alla parete, i parametri τ_w e ν giocano un ruolo importante. Da queste due grandezze, assieme alla densità ρ vengono definite delle *scale viscose*, utilizzate per descrivere velocità e scala di lunghezza di un flusso turbolento nelle vicinanze della parete:

- *friction velocity*

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}} \quad (1.12)$$

- *viscous lengthscale*

$$\delta_\nu = \nu \sqrt{\frac{\rho}{\tau_w}} = \frac{\nu}{u_\tau} \quad (1.13)$$

E' possibile introdurre una coordinata spaziale per misurare la distanza dalla parete in unità viscosi:

$$y^+ = \frac{y}{\delta_\nu} = \frac{u_\tau y}{\nu} \quad (1.14)$$

La quale è interpretabile come un numero di Reynolds locale¹, tramite il quale è possibile determinare il peso del contributo viscoso e turbolento nel caso in esame, come mostra la figura 1.4. Si può notare come per valori di $y^+ > 50$ il contributo viscoso sia praticamente nullo, mentre assume valori elevati a $y^+ < 12$.

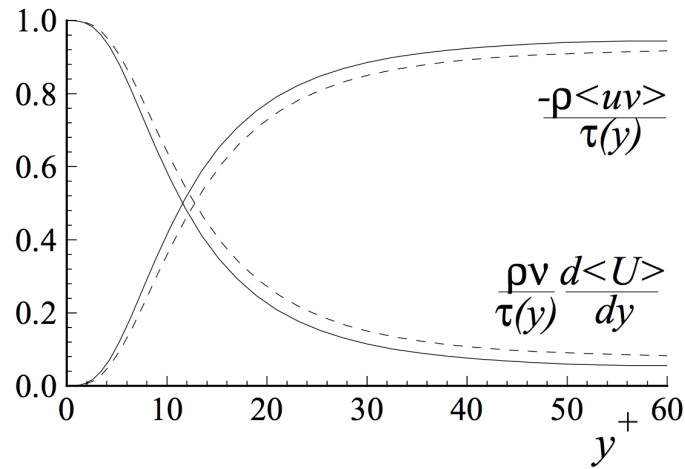


Figura 1.4: Contributi di stress viscoso e di Reynolds rapportati allo stress di taglio totale. Linea tratteggiata, $Re=5600$; linea continua, $Re=13750$. DNS data di Kim *et al.* (1987). Immagine da 'Turbulent flows', di Stephen B. Pope (2000).

I valori assunti da y^+ permettono di dividere il flusso che scorre vicino alla parete in varie regioni, alcune di queste sono:

¹Un'altra interpretazione, più intuitiva, è quella di pensare ad y^+ come al diametro del vortice più piccolo che si può formare all'interno dello strato limite turbolento.

- ($y^+ < 50$) *Viscous wall region*: nella quale si ha un contributo importante della viscosità molecolare sulla τ ;
- ($y^+ > 50$) *Outer layer*: in cui l'effetto della viscosità è trascurabile;
- ($y^+ < 5$) *Viscous sublayer*: in cui l'effetto del Reynolds è trascurabile.

Dall'equazione (1.12) può essere infine ricavato il *friction Reynolds number*, cioè il numero di Reynolds definito dalla velocità vicino a parete, quindi nella regione dell'*inner layer* (la quale verrà descritta più avanti):

$$Re_\tau = R^+ = \frac{u_\tau \delta}{\nu} = \frac{\delta}{\delta_\nu} \quad (1.15)$$

1.2.3 Law of the Wall

L'analisi dimensionale del flusso completamente sviluppato, mostra come questo sia definito dalle quattro grandezze ρ , ν , δ e dp_w/dx (o alternativamente u_τ), le quali, tramite il teorema II (conosciuto anche come teorema di Buckingham), si riducono a due sole grandezze adimensionali y/δ e Re_τ , capaci di descrivere il profilo di velocità medio.

Poichè diverse grandezze caratteristiche del moto turbolento sono espresse in funzione del gradiente di velocità medio $d\bar{U}/dy$, l'equazione alla quale si arriva dallo studio dimensionale è:

$$\frac{d\bar{U}}{dy} = \frac{u_\tau}{y} \Phi \left(\frac{y}{\delta_\nu}, \frac{y}{\delta} \right) \quad (1.16)$$

in cui Φ è una funzione universale adimensionale.

I due parametri adimensionali presentano la coordinata y rapportata a δ , quindi relativa all'*outer layer*, ed a δ_ν , relativo alla scala viscosa (che contiene l'informazione di Re_τ).

Un'ulteriore semplificazione della legge trovata si può avere considerando valori di $y \ll \delta$ e Re alti (ipotesi di Prandtl, 1925). Queste condizioni sono soddisfatte qualora ci si trovi nell'*inner layer* e permettono di descrivere il profilo delle velocità medie in funzione della sola scala viscosa, per cui, l'equazione (1.16) diventa unicamente funzione di y/δ_ν :

$$\frac{d\bar{U}}{dy} = \frac{u_\tau}{y} \Phi_1 \left(\frac{y}{\delta_\nu} \right) \quad (1.17)$$

Dalla definizione di y^+ ed essendo $u^+ = \bar{U}/u_\tau$, l'equazione (1.17) può essere riscritta come:

$$\frac{du^+}{dy^+} = \frac{1}{y^+} \Phi_1(y^+) \quad (1.18)$$

Il cui integrale definisce la legge alla parete (*law of the wall*):

$$u^+ = f_w(y^+) \quad (1.19)$$

L'importanza di questa legge risiede nel fatto che, per l'ipotesi di Prandtl, u^+ dipende unicamente da y^+ per il fluido nelle immediate vicinanze della parete. Per valori di Re lontani dalla transizione sono presenti molti risultati sperimentali che verificano l'universalità della legge.

1.2.4 Log Law e costante di von Karman

Approfondendo lo studio dell'*inner layer* ($y/\delta < 0.1$) ad elevati numeri di Reynolds, si può individuare una regione esterna corrispondente a grandi valori di y^+ , che prende il nome di *overlap region*, cioè una zona in cui *inner* ed *outer layer* si vengono a sovrapporre. Come già visto in figura 1.4, ad alti valori di y^+ corrispondono effetti trascurabili di viscosità, e quindi, nell'equazione (1.17), Φ_1 assume il valore di κ^{-1} (costante di *von Karman*), restituendo, dall'equazione (1.18) il seguente gradiente di velocità medie:

$$\frac{du^+}{dy^+} = \frac{1}{\kappa y^+} \quad (1.20)$$

che, una volta integrato, permette di ottenere la *Log Law* (nota anche come legge logaritmica alla parete di *von Karman*):

$$u^+ = \frac{1}{\kappa} \ln y^+ + B, \quad \text{per } \frac{y}{\delta} \ll 1 \text{ e } y^+ \gg 1 \quad (1.21)$$

in cui B è una costante (in letteratura si hanno solitamente i valori di $B = 5.2$ e $\kappa = 0.41$).

Un aspetto importante della *Log Law* può essere evidenziato dal grafico in figura 1.5. Infatti, per diversi valori del numero di Reynolds, i dati misurati tendono a ricondursi ad un'unica curva, a conferma della *law of the wall*, in maniera più specifica seguendo l'andamento della *Log Law* (per $y^+ > 30$)².

I dati si discostano dalla teoria nella regione in cui $5 < y^+ < 30$, definita

²Nuove teorie stanno mettendo alla luce come il valore di κ non risulti effettivamente universale, (cfr. H.M. Nagib e K.A. Chauhan, *Variations of von Kármán coefficient in canonical flows*, 2008)

buffer layer, zona di transizione tra le parti del fluido in cui predomina il termine viscoso o quello turbolento, in cui il legame è lineare e non più logaritmico ($y^+ = u^+$).

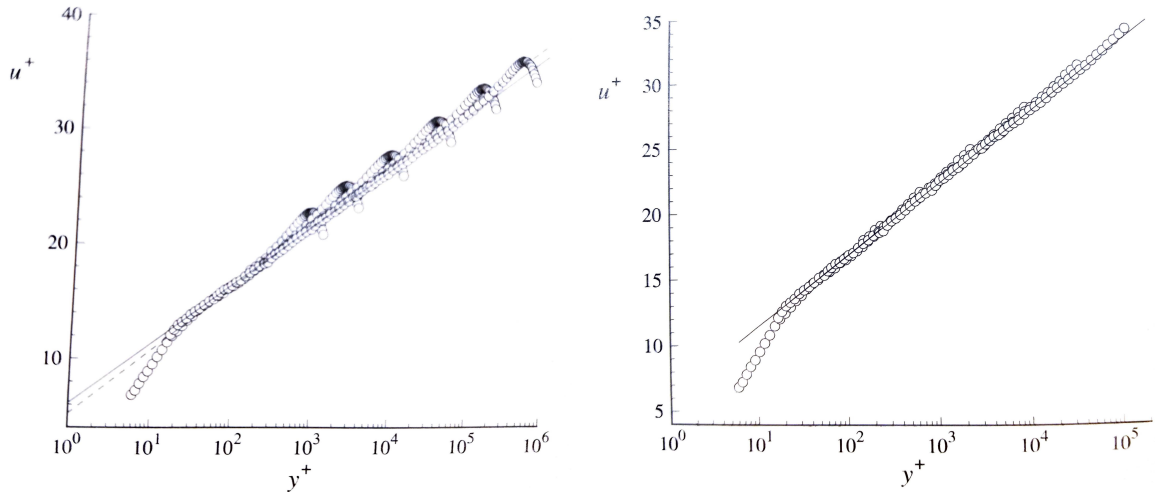


Figura 1.5: Profili di velocità medie per flusso completamente sviluppato all'interno di un condotto. A sinistra si ha il caso generale a destra il caso per $y/R < 0.1$. Sono evidenziati i dati per diversi numeri di Reynolds. Dati sperimentali di Zagarola e Smits (1997). Immagine da 'Turbulent flows', di Stephen B. Pope (2000).

Sempre dalla figura 1.5 si può notare l'allontanarsi dalla *Log Law* qualora ci si sposti verso le zone del fluido più interne del *pipe* (*ceterline*), quindi per $y/\delta > 0.1$, in cui la funzione Φ è definita unicamente dal rapporto y/δ ($\Phi = \Phi_0$) ed è trascurabile il termine dovuto a ν .

Vale, in questo caso, la *velocity-defect law* di von Karman (*Log Law* rivalutata), ottenuta integrando la (1.18) tra y e δ , con Φ_0 al posto di Φ_1 :

$$\frac{U_0 - \bar{U}}{u_\tau} = F_D \left(\frac{y}{\delta} \right) \quad (1.22)$$

Nella quale la differenza tra U_0 ed \bar{U} rappresenta il 'difetto' di velocità. F_D , al contrario della *law of the wall*, non ha valore universale, differendo, quindi, in base al tipo di flusso che si sta analizzando.

L'uguaglianza tra Φ_0 e Φ_1 è possibile unicamente, come già detto all'inizio del paragrafo, all'interno della regione di sovrapposizione, in cui le due funzioni non possono che essere uguali alla costante κ .

Capitolo 2

Acquisizione della pressione

Il processo più utilizzato al fine di acquisire la pressione e, quindi, conoscerne la caduta lungo il *pipe* è la catena di misurazione, composta da quattro elementi fondamentali:

1. Sensori di pressione
2. Trasduttori di pressione
3. Sistemi di acquisizione dati
4. Sistemi di analisi di dati

2.1 Sensori di pressione

Nella valutazione delle pressioni, l'utilizzo dei sensori è dovuto alla peculiarità che questi hanno di cambiare la loro caratteristica una volta in contatto con la quantità da valutare. Essi possono essere suddivisi in base alle loro caratteristiche. Le più importanti sono:

- Risoluzione dello strumento in termini di valutazione spaziale e temporale;
- Precisione dello strumento in base al luogo di applicazione, come il sensore reagisce e quanto la misura conseguente sia simile alla realtà;
- Capacità del sensore di non influenzare attraverso il suo posizionamento il flusso circostante;
- Robustezza del sensore: quanto la reazione del sensore sia affidabile nonostante un ambiente di lavoro non ideale od un uso incorretto;
- Costo del sensore.

2.1.1 Principio di funzionamento

La creazione di un sensore è legata strettamente alla fisica dei fluidi in movimento. La pressione in un punto interno del flusso è definita come la media degli stress normali che agiscono su tre superfici perpendicolari fra di loro, chiamata *pressione statica*. Se il fluido viene rallentato sino a velocità nulla attraverso un processo isoentropico si raggiunge un valore di pressione massimo, chiamato *pressione totale*. La pressione ottenuta a velocità nulla viene detta *pressione al punto di ristagno*, mentre la differenza fra pressione totale e pressione statica definisce la pressione dinamica (come conseguenza della validità dell'equazione di Bernoulli):

$$p_{dyn} = p_{tot} - p_s \quad (2.1)$$

in cui

$$p_{dyn} = \frac{1}{2}\rho V^2 \quad (2.2)$$

I sensori di pressione più utilizzati al fine di misurare la pressione statica del flusso indisturbato sono chiamati *wall tapings*. Sono ottenuti realizzando delle cavità lungo il condotto e ricondotti ad una camera in cui la pressione è la medesima del flusso indisturbato. Un cambiamento di pressione nel flusso indisturbato comporta perciò un cambiamento di pressione all'interno della camera, influenzando dunque la lettura del sensore.

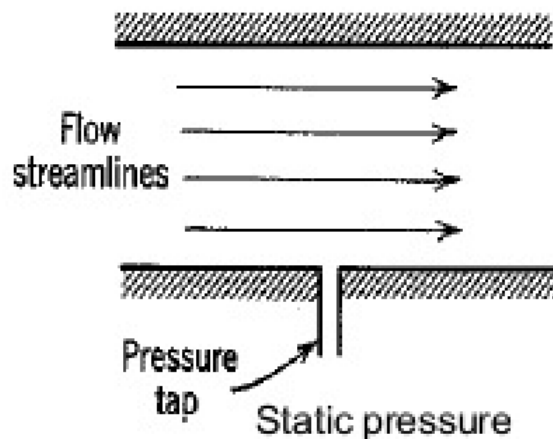


Figura 2.1: Sensore wall tapping

Il cambio di pressione all'interno della cavità del sensore può essere affetto da errori, che ne caratterizzano la sua robustezza ed accuratezza.

In particolar modo si hanno due tipologie di errore:

- Errori randomici, cioè dovuti alle incertezze date dalla ripetizione di misurazioni sperimentali, dunque strettamente connesse alla condizione dell'ambiente nell'istante di tempo in cui viene effettuata la rilevazione;
- Errori sistematici o di bias, dipendenti strettamente dal tipo di sensore quindi dalla sua orientazione, dalla sua forma e dalla posizione nella sezione.

Per ridurre al minimo gli errori si limita la simulazione in galleria del vento a numeri di *Reynolds* bassi: il flusso sarebbe infatti troppo complesso all'intorno della cavità, amplificandone possibili imperfezioni. Dall'altro lato, a questi valori di *Re* la viscosità non risulta più trascurabile, rendendo la valutazione dell'errore più complicata.

2.2 Trasduttori di pressione

Il trasduttore di pressione è l'elemento che, una volta collegato al sensore, ne trasforma la variazione della grandezza analizzata in un segnale di output, a cui corrisponde uno specifico valore di pressione.

Si hanno diversi tipi di trasduttori in base al tipo di valutazione della pressione che si vuole avere:

1. Trasduttore di pressione assoluta: effettua una misurazione in relazione alla pressione del vuoto. E' caratterizzato da un'accuratezza minore, dell'ordine dello 0.1%;
2. Trasduttore di pressione di gauge: effettua una misurazione relativa ad una data pressione atmosferica in un dato punto;
3. Trasduttore di pressione differenziali: effettua una misurazione tra due valori di pressione presi come input.

I trasduttori sono, inoltre, definiti da alcuni parametri caratteristici, dipendenti dal tipo di applicazione, quali:

- Range di funzionamento: campo di valori in input tali da garantirne un funzionamento ideale;
- Legge di trasferimento input/output: legame matematico fra le grandezze in entrata e quelle in uscita;

- Segnale di output: se analogico o digitale;
- Accuratezza in percentuale: errore massimo nella grandezza in uscita;
- Tempo di risposta in frequenza.

2.2.1 Trasduttori di pressione capacitivi

La tipologia di trasduttore di pressione utilizzata al centro CICLoPE è quella dei trasduttori capacitivi. Questi si basano sul funzionamento di un condensatore e perciò sulla valutazione di come l'azione di una pressione possa variarne la sua capacità. Il condensatore, alimentato da corrente alternata, viene idealmente schematizzato a facce parallele, le cosiddette armature, in cui la capacità è variabile secondo la legge:

$$C = \epsilon_r \frac{A}{d} \quad (2.3)$$

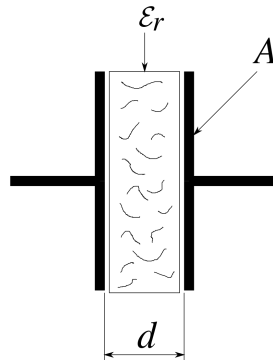


Figura 2.2: Schema di un condensatore a facce parallele.

dove:

- A è la superficie delle armature;
- ϵ è il dielettrico interposto alle armature;
- d è la distanza fra le armature.

Un reale trasduttore capacitivo di pressione è, invece, realizzato mediante una membrana che separa le due zone in cui sono posizionati e collegati gli elettrodi. Quando la pressione viene applicata alla membrana, essa si

deforma modificando anche la distanza fra gli elettrodi e pertanto anche la capacità del condensatore schematizzato. Questa variazione è così convertita in corrente AC e successivamente amplificata e demodulata come DC in modo da avere un segnale ben leggibile di output.

L'output del trasduttore è solitamente debole, deve quindi essere amplificato e riadattato alla scala di valori desiderata. Una volta amplificato il segnale si applicano anche eventuali offset ed errori di fondo scala, che devono essere misurati e limitati, in modo da avere un output nel range di valori desiderato.

Nonostante l'elevato numero di aspetti positivi di questa categoria di trasduttori, quali:

- Elevato grado di accuratezza nel calcolo della pressione;
- Legame lineare fra l'input di pressione e l'output di voltaggio;
- Sistema con elevata risposta in frequenza, fino ad almeno 5 kHz, limitata dal collegamento fra sensore e trasduttore: data la cavità del *wall tapping* come sensore, quando la pressione nel condotto varia questa subisce la stessa variazione nella cavità che sarà collegata al trasduttore tramite un piccolo tubo.
A seconda della distanza fra i due, il trasduttore riceve la pressione in input in un determinato range temporale. Riducendo così la risposta dell'intero sistema a circa 30Hz considerando una velocità del suono $a=343 \text{ m/s}$ a quota $h=0 \text{ m}$ e temperatura ambiente;
- Ridotta isteresi e dunque ridotto errore di linearità rispetto alla retta ideale.

Posseggono alcuni difetti, soprattutto legati alla deformazione plastica degli elettrodi, causata dall'elevato utilizzo. Necessitano perciò di un frequente numero di calibrazioni ottenute attraverso la presenza di un trasduttore di pressione differente.

Data una funzione lineare del tipo:

$$V_2 = k\Delta p_1 - \Delta p_0 \quad (2.4)$$

riadattandone la formula si ottiene:

$$k\Delta p_1 = V_2 - V_0 \quad (2.5)$$

dove:

- V_2 è il voltaggio in output dal secondo trasduttore;

- Δp_1 è il valore di pressione in input al primo trasduttore;
- Δp_0 è il valore di pressione iniziale;
- V_0 valore di tensione iniziale;

si ottiene, così, un'equazione che permette di calibrare il trasduttore capacitivo grazie ad un altro trasduttore che misuri il voltaggio (V) ed alla sua caratteristica lineare.

Il costo elevato ne causa la poca convenienza per l'acquisizione in più punti.

2.3 Sistema di acquisizione dati

Un sistema di acquisizione dati è un sistema in grado di rilevare e memorizzare grandezze in output, sia analogiche che digitali, dal trasduttore. È costituito da un insieme di dispositivi elettronici il cui fine è quello di visualizzare, controllare, memorizzare ed in seguito trasmettere dati. Si può schematizzare come un sistema centrale che governa controllo e frequenza per acquisizione e trasmissione dei dati, eseguendo dunque operazioni di elaborazione e comunicazione.

Un processo di acquisizione si articola in tre differenti fasi chiamate:

1. Trattamento dei dati;
2. Acquisizione dei dati;
3. Conversione dei dati.

2.3.1 Trattamento e acquisizione dei dati

L'acquisizione dei dati si basa sul passaggio A/D, ovvero ricevere in input un segnale analogico campionandolo in digitale, in modo da poterlo curare attraverso un elaboratore di dati. Utilizza, dunque, un processo di campionamento dei dati (*sampling*).

Sampling:

Tramite il processo di sampling si può impostare a quale frequenza si voglia discretizzare una data funzione continua di input.

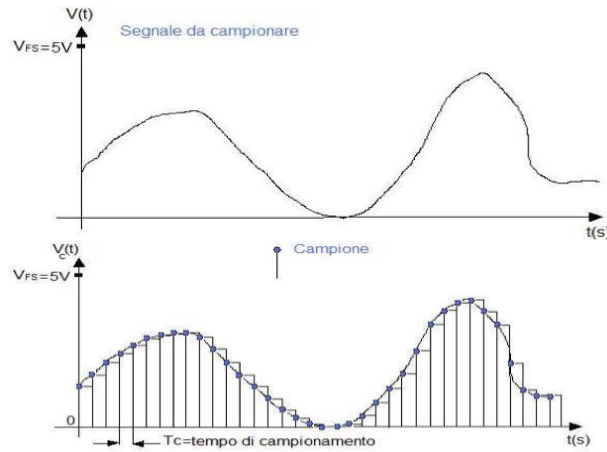


Figura 2.3: Funzione continua e suo campionamento.

Nel caso di un trasduttore di pressione si indica a che frequenza acquisire i valori di voltaggio misurati.

Si ottiene:

$$t_s = \frac{1}{f}$$

dove f è la frequenza acquisizione e t_s il tempo di sampling.

Maggiore è dunque la frequenza di acquisizione, maggiore sarà il numero di valori campionati mentre minore sarà il tempo di campionamento.

Per valutare la corretta frequenza di acquisizione si utilizza il *criterio di Nyquist*:

$$f_x = 2f_{known} \quad (2.6)$$

che enuncia come la frequenza di acquisizione debba essere almeno il doppio della frequenza del segnale preso in input dal trasduttore. Se il criterio non fosse applicato si potrebbero verificare errori di *aliasing*, con possibile confusione fra elementi a bassa ed elevata frequenza.

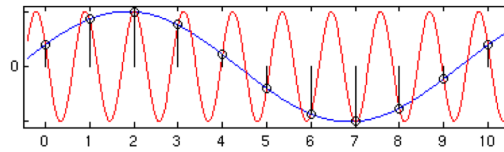


Figura 2.4: Errore di aliasing. Immagine da Talamelli A., *Laboratorio di Aerodinamica Sperimentale, appunti e dispense, anno accademico 2018/19*

Questo comporterebbe un errore nella valutazione della frequenza vera di un segnale. E' dunque necessario conoscere la frequenza che ci si aspetta dal segnale in input al sistema, oppure applicare un filtro che limiti la frequenza alla metà della frequenza che si desidera in acquisizione. E' necessario effettuare un processo di trattamento dei dati anche al fine di filtrare le frequenze registrate dal segnale a causa degli errori randomici.

Bisogna inoltre avere un numero di campioni tale da poter fornire un'analisi statistica stabile. Avendo limitato la frequenza questo comporta un tempo di acquisizione dell'intero processo maggiore.

Per effettuare l'acquisizione di più canali si usano dispositivi *multiplexer* (la *Scanivalve*, nel caso analizzato) che collegano sequenzialmente più linee di input ad una singola linea.

2.3.2 Conversione dei dati

Ottenuti i dati in voltaggio il sistema deve elaborarli e convertirli in sequenze di bit, segnali digitali che possano essere analizzati da un processore. Questo processo è direttamente collegato al numero di bit che si vuole utilizzare per valutare la funzione campionata. Un sistema a 16 bit permette un'elevata precisione nella caratterizzazione della curva ma, al contempo, un aumento esponenziale di dimensione nel segnale digitale che può causare problemi di *storing* oppure di calcolo computazionale (approfondito nella sezione dedicata del capitolo 4).

L'errore legato al fissaggio di un numero definito di bits si chiama errore di quantizzazione (*SNR - Signal-to-noise ratio*), descritto dall'equazione:

$$SNR = 6N_{bits} - 20 \log \frac{\sigma}{N_{range}}$$

con σ rappresentante la deviazione standard.

Capitolo 3

Analisi del sistema di acquisizione del CICLoPE

La valutazione della caduta di pressione è fondamentale in un condotto poiché consente di conoscere le caratteristiche di attrito e velocità del flusso turbolento.

Risulta dunque necessario limitare gli errori di misurazione in modo da ottenere valori di pressione che possano descrivere nel modo più preciso possibile le caratteristiche del flusso, andando ad influenzare la legge logaritmica della parete di *von Karman* e la valutazione delle grandezze tipiche di questo tipo di flusso.

Inoltre, ridurre gli errori sulla pressione consente di effettuare misure più precise sulla velocità nel condotto attraverso il *tubo di Pitot*, al termine del long pipe, tramite il dato di pressione statica (p_s) e di pressione totale (p_{tot}).

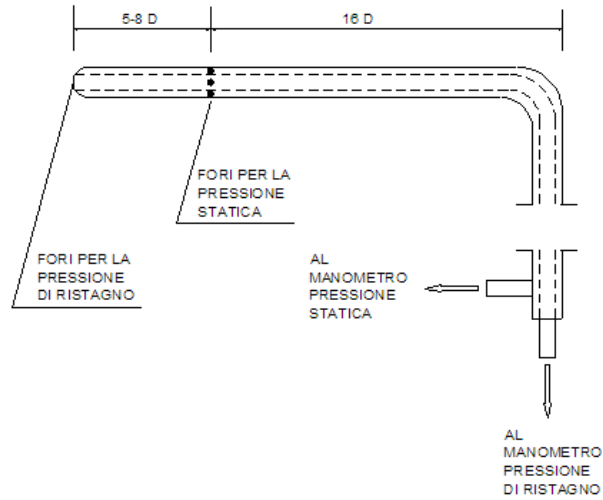


Figura 3.1: Schema del tubo di Pitot. Evidenziate le prese statiche, la presa totale ed il collegamento con il trasduttore di pressione

La pressione statica è misurata tramite un corpo emisferico, che presenta due fori posizionati sufficientemente a valle della parte frontale dello strumento. A questi due punti corrispondono valori del coefficiente di pressione (C_p) uguali a zero.

Essendo la definizione di C_p :

$$C_p = \frac{p - p_\infty}{\frac{1}{2}\rho V_\infty^2} \quad (3.1)$$

Si ha che, laddove il $C_p=0$, dall'equazione (3.1), la pressione ottenuta equivarrà a quella del flusso indisturbato p_∞ .

Nel caso della pressione totale bisogna avere una sonda capace di rallentare il flusso in maniera *isoentropica*, cioè adiabatica e reversibile (come già spiegato nel paragrafo 2.1.1). La misurazione della p_{tot} si effettua nella parte frontale dello stesso corpo utilizzato per la misurazione della p_s .

I due valori ottenuti dai fori praticati sullo strumento (che lavorano, quindi come veri e propri sensori di pressione) sono poi trasformati in segnali di output dal trasduttore e consentono di ricavare, tramite le equazioni (2.1) e (2.2), la legge della velocità:

$$V = \sqrt{\frac{2(p_{tot} - p_s)}{\rho}}$$

3.1 Sistema di misurazione attuale del centro CICLoPE

In primo luogo, è stata effettuata un'analisi preliminare del sistema di acquisizione della pressione lungo il *long pipe* del centro CICLoPe.

L'attuale sistema di misurazione è composto da una serie di sensori della tipologia *wall tapping* lungo il condotto (illustrati nel capitolo 2) in modo da poter analizzare la caduta di pressione a intervalli regolari.

La misurazione della pressione viene effettuata attraverso un sistema di trasduttori contenuti all'interno del *DTC Initium*. Ogni sensore è direttamente collegato al sistema di trasduttori attraverso canali diversi: viene effettuata perciò una misura di pressione separata per ogni collegamento sensore-trasduttore.

3.1.1 DTC Initium

L'Initium DTC (dove DTC indica la tecnologia di *Digital Temperature Compensation*) è uno scanner di pressione ESP (sistema di sensori) tramite il quale è possibile acquisire dati.

Presenta un numero di ingressi che può arrivare fino a 512, cioè 64 canali di pressione per ogni scanner DTC (32 nel caso specifico del CICLoPE), con un'accuratezza dello $\pm 0.5\%$ FS (*full scale precision*) dopo il ri-azzeramento. Per ogni canale si ha una capacità di trasmissione ad una frequenza di 1200 Hz.



Figura 3.2: Initium utilizzato al ciclope. Immagine da *DTC Initium Manual*.

Il sistema dispone di un circuito analogico avanzato con tecnologia DTC che consente di mantenere un'accuratezza ideale senza bisogno di continue calibrizioni. Ciò avviene grazie l'utilizzo di una tecnica particolare di montaggio del sensore tramite la quale si riesce a misurare la resistenza agente sul ponte per effettuare correzioni in tempo reale degli errori termici. Il tutto è implementato, infine, in algoritmi integrati all'interno degli scanner. L'initium DTC può essere controllato con *triggering* da software o da hardware, in modo da renderne possibile la movimentazione secondo input inseriti dall'utente od automatizzati in base ai dati che sono stati acquisiti.

L'interfaccia di controllo si basa su un sistema di rete Ethernet, la cui connessione è gestita da:

- Un primo microprocessore, che ne valuta anche la conversione ingegneristica delle unità di misura (da pressione a digitale);
- un secondo microprocessore che si occupa, invece, dell'orientamento dei canali (con lo scanner che ne controlla l'input, tramite segnali dello scanner in linee analogiche dedicate) ed effettua il *data sampling*.

Questa particolare architettura consente un'acquisizione continua, limitata unicamente dal tempo richiesto per la stabilizzazione degli output (50 μ s). La conversione dei dati viene eseguita quindi in maniera automatica.

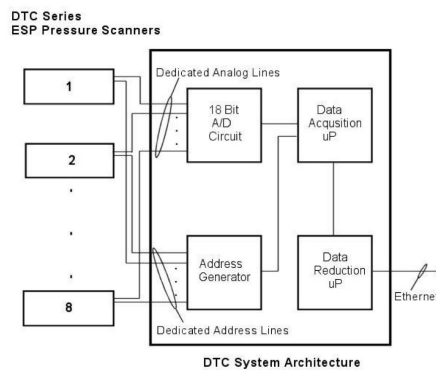


Figura 3.3: Initium Mainframe. Immagine da *DTC Initium Manual*.

3.1.2 Problemi relativi alla misura

L'uso di questa tecnologia risulta svantaggioso nonostante le caratteristiche evidenziate.

Nella valutazione dell'errore, essendo ogni sensore collegato ad un singolo scanner di pressione si avranno perdite in ognuno di questi e dunque un'acquisizione meno precisa con intervalli di errore variabili. Un errore variabile da punto a punto lungo la galleria (nel caso dell'*Initium*) conduce ad un'area di incertezza molto ampia, con conseguente incongruenza dei dati sperimentali con quelli ipotizzati dalla *Law of the wall* e dalla *Log law*, e quindi sulla misura delle quantità y^+ e, soprattutto, u^+ .

Una stima dell'errore può essere effettuata nota la linearità della caduta di pressione lungo il *pipe* per un flusso turbolento completamente sviluppato. Analizzando i dati sperimentali ottenuti dalle misure lungo lo stesso in figura 3.4 (Fiorini T., *Turbulent Pipe Flow - High Resolution Measurements in CICLoPE*, (2017)) si può verificare di quanto differisca la caduta di pressione dalla relazione supposta lineare.

Questa analisi è inoltre strettamente connessa al numero di punti analizzati andando a valutare sia un errore di bias di ogni sensore, dovuto a possibili piccole imperfezioni, che l'errore dovuto alla misura tramite sistema *DTC Initium*. Dunque a crescente numero di sensori si valutano diverse interpolazioni lineari, quantificando l'errore come:

$$p_e(x) = p_{data}(x) - p_{fit}(x) \quad (3.2)$$

Noto l'errore di pressione si può valutare la deviazione standard e analizzare quanto i risultati reali si discostano da quelli attesi.

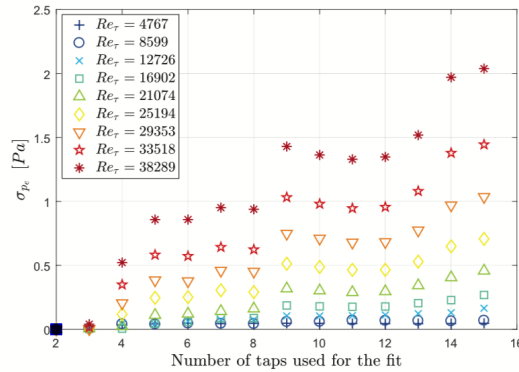


Figura 3.4: Deviazione standard dell'errore di pressione definita in funzione del numero di sensori su cui è stata effettuata l'interpolazione. Dalla tesi di dottorato di T. Fiorini *Turbulent Pipe Flow - High Resolution Measurements in CICLoPE*

3.2.2 MKS Baratron Pressure Transducer

Il trasduttore *MKS Type 120A* è un manometro capacitivo che permette di misurare pressioni nel range da 25000 *torr* fino a valori di 10^{-5} *torr*. Il segnale ottenuto è un output di corrente continua tra 0 e 10 Volt, letto direttamente da un DVM (*digital Voltmeter*).



Figura 3.6: Il trasduttore MKS utilizzato per le misurazioni

Al fine di minimizzare gli effetti dovuti all'ambiente e rendere le misurazioni più stabili, lo strumento dispone di un controllo del sensore di temperatura con valore fisso a 45 °C. Per prevenire interferenze dovute a radio frequenze o rumore elettronico (EMI - *Electromagnetic interference*), oltre all'istallazione, che deve essere eseguita lontana da sorgenti di disturbo, è presente un *RFI Shielding* (schermatura per *radio frequency interference*). Infine, per proteggere lo strumento da sovra-pressioni e contaminazione dovuta a processi chimici, viene applicata una valvola di isolamento.

L'*MKS* è diviso in tre parti principali:

- Sensore a diaframma di capacità variabile *Inconel*;
- Pre-amplificatore elettronico e circuito a ponte ad alta impedenza;
- *Housing* controllato in temperatura per riurre l'effetto dell'ambiente e schemi circuitati per fornire energia e per il condizionamento del segnale.

Il sensore *Inconel* è alloggiato, assieme al circuito a ponte ed al pre-amplificatore, all'interno dell'*housing* di alluminio.

Essendo un sensore capacitativo, una volta applicata la pressione al diaframma, la deflessione modifica la distanza tra gli elettrodi, traducendosi in un cambiamento di capacità risultante. Questa variazione viene convertita in un voltaggio AC dal ponte ad alta impedenza, il quale è eccitato da un oscillatore ad una frequenza costante ben precisa. Il segnale viene poi amplificato e successivamente de-modulato, creando un output stabile nel range tra 0 e 10 VDC (*Volts of Direct Current*), direttamente proporzionale alla tensione. Al fine di ottenere la misura di pressione, si può utilizzare la curva di calibrazione ricavata dalle informazioni fornite dal manuale, essendo il legame tra pressioni e voltaggio lineare. Utilizzando i range di tensioni (da 0 a 10 Volt) e di pressioni (da 1 a 25000 mmHg) a cui lavora l'*MKS* è immediato ricavare la legge:

$$p = 333236,7V + 133.3 \quad (3.3)$$

in cui p rappresenta la pressione e V il voltaggio.

L'*MKS* presenta due canali collegati a due condotti nei quali scorre il fluido a due diversi valori di pressione ed è utilizzato per ottenere misurazioni di tipo differenziale.

L'*MKS* è connesso con il cRIO utilizzato in laboratorio (il quale sarà approfondito nel paragrafo seguente) tramite un cablaggio che fa uso di un connettore BNC (connettore unipolare a baionetta).

3.2.3 Funzionamento del nuovo sistema di misurazione della pressione

L'utilizzo della *Scanivalve* permette di conglomerare i canali di pressione (compresi i due necessari alla misura della velocità tramite tubo di pitot) lungo il *pipe* nell'unico trasduttore MKS. Questo permette di avere un'incertezza sicuramente minore della somma delle incertezze che si hanno con lo scanner di pressione *Initium* perchè l'MKS ha un'inesattezza minore e, soprattutto, uguale per tutti i canali, riflettendosi in migliori valutazioni per lo studio del flusso turbolento.

L'MKS possiede un proprio sistema di valutazione della capacità del trasduttore, misurando direttamente la pressione. Per l'elaborazione dei dati e l'elaborazione dei segnali di output è necessario adoperare strumenti di acquisizione di segnali analogici, in particolar modo sono stati usati a livello hardware un dispositivo *Compact Rio* e a livello software *LabVIEW*, entrambi sviluppati da National Instruments.

Compact RIO 9068

Il modello di CompactRIO di cui si è fatto uso, per la ricezione dei segnali dell'MKS è il cRIO NI 9068 ad 8 slot. Questo sistema di tipo *controller embedded* (sistema di controllo integrato) è un dispositivo fisico alimentato direttamente, composto da uno chassis collegabile al computer, nel quale sono presenti prese seriali per l'input di dati, e dal collegamento di diversi moduli *C-series* di I/O che differiscono nelle funzionalità a seconda del modello installato, consentendo un'ottima personalizzazione.

Lo chassis del Compact RIO presenta, oltre ad un collegamento di alimentazione, diverse soluzioni per il collegamento al computer, sia attraverso connessione USB che tramite rete ethernet. E' presente inoltre un comando di reset.

Per il controllo dello strumento al centro CICLoPE si è usato la connessione diretta ethernet, mentre la connessione USB è stata solamente adoperata per una prima implementazione e configurazione.

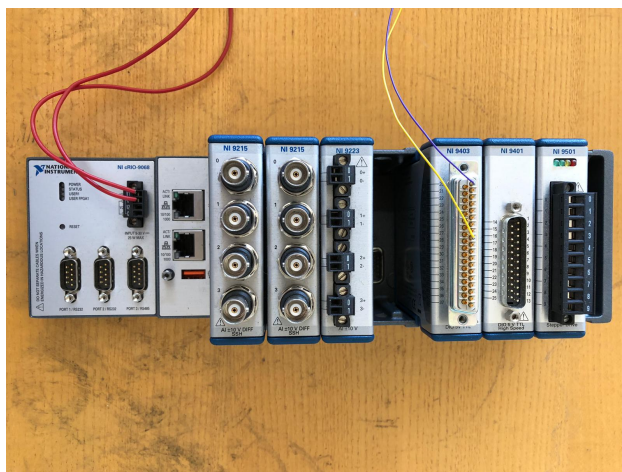


Figura 3.7: Il cRIO NI 9068 utilizzato per le misurazioni.

Per la ricezione dei segnali di Input analogico da parte dell'MKS al Compact Rio si è preferito usare un collegamento attraverso una porta seriale (RS232), collocata nello chassis, che ha consentito un collegamento diretto fra gli strumenti, mentre per i segnali di Output analogico/digitale sono stati utilizzati i moduli *C-series* collegati, usati per il comando alla *Scanivalve*. La gestione dei segnali all'interno dello strumento è svolta da due sistemi separati: un microprocessore in grado di elaborare dati e usare la CPU del computer, implementando algoritmi per il controllo, in modo da aver elevata capacità di calcolo nello svolgimento del codice e da una FPGA (*field programmable gate array*).

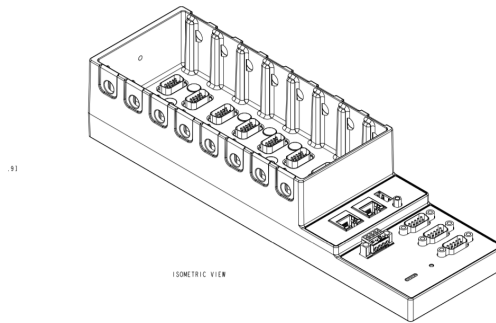


Figura 3.8: Schema di allocazione delle porte I/O. Immagine da *NI cRIO-9068 user manual* (2016).

Sistema FPGA

Il sistema FPGA è una componente fondamentale del cRIO: è rappresentabile come un circuito integrato direttamente controllato tramite un linguaggio di programmazione, che consente dunque di elaborare e modificare attraverso un linguaggio grafico l'hardware del Compact RIO. E' solitamente composta da una matrice di blocchi logici configurabili detti CLB (*Configurable logic blocks*), collegati tra di loro attraverso interconnessioni programmabili. Legata a questa struttura interna vi sono i blocchi di ingresso/uscita alla matrice denominati IOB *Input/Output Blocks*.

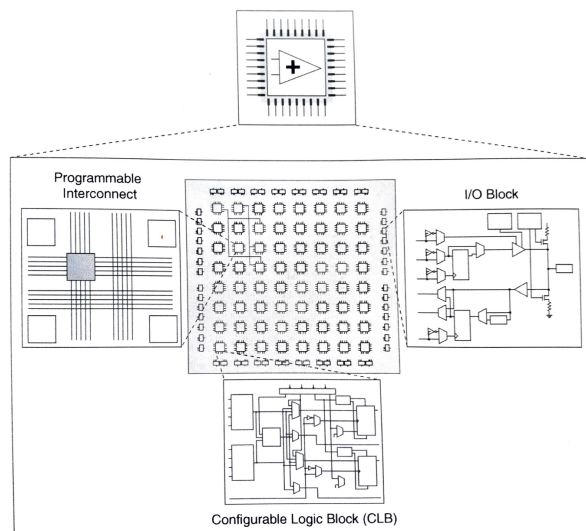


Figura 3.9: Schema funzionale FPGA da *Control and Embedded sistem*.

Il sistema funziona, perciò, tramite la creazione dei programmi logici per mezzo di CLB, comunicanti attraverso le interconnessioni con l'esterno per mezzo degli IOB.

L'FPGA è un metodo di approccio all'utilizzo dei dati che si discosta dai circuiti della CPU mantenendo, nonostante ciò, delle analogie legate a come vengono elaborati gli algoritmi, soprattutto attraverso funzioni logiche booleane e memorizzazione dei dati. Riorganizzare una FPGA risulta però più complesso, implicando la configurazione di nuovi schemi di calcolo. Questo comporta tempistiche nella creazione di porte logiche maggiore (compilazione dell'FPGA) ma un approccio più diretto al problema una volta configurata. L'uso dell'FPGA svincola dall'uso della CPU del computer e consente di avere una memoria di calcolo direttamente usata per i segnali di input, mentre la CPU è legata a fenomeni di tipo TDMA (Time Division Multiple Access) che non consentirebbero una risposta immediata ai segnali nel caso di altre attività svolte dal computer in concomitanza.

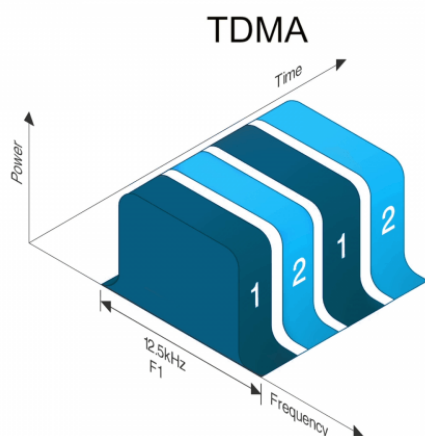


Figura 3.10: Esempio di sezionamento del lavoro della CPU di un computer. Immagine dal sito *Tait Radio Academy*.

L'uso in contemporanea di questi due sistemi rappresenta la migliore soluzione: la CPU può concentrarsi sugli algoritmi più complessi ed in relazione a cambi di contensto rapidi, mentre l'FPGA svolge le attività ripetitive e programmate. Il sistema così descritto consente un valutazione in *Real Time*, come nel caso del sistema di acquisizione sviluppato: utilizzando l'FPGA per la ricezione e l'invio di segnali analogici mentre la CPU elabora e scrive *buffer* di dati.

Software LabVIEW

Il principale software utilizzato per l'acquisizione e l'analisi dei dati ottenuti in ingresso al compact RIO, sul quale poi è stato progettato il codice di acquisizione, è LabVIEW, il quale permette di creare interfacce basate su schemi a blocchi tramite un linguaggio grafico (*Graphic Language*). I dati elaborati dal software possono essere simulati oppure acquisiti da input digitali ed analogici.

LabVIEW gestisce inoltre la programmazione dell'FPGA e la gestione della CPU. Il software usato è composto da 3 parti principali:

- LabVIEW 2014
- LabVIEW FPGA module
- LabVIEW Real Time

Queste componenti creano il software di gestione completo.

L'utilizzo del programma LabVIEW consente la gestione del codice svolto dalla CPU ed inoltre la gestione delle informazioni provenienti dall'esterno con due metodologie:

- Modulo FPGA, che gestisce la compilazione della CPU ed in cui i sistemi vengono integrati nel modulo Real Time, per sincronizzare i movimenti di dati;
- DAQ-mx (Data Acquisition), ovvero un altro tipo di canale di acquisizione di dati presente in alcuni Compact RIO che svincola dall'utilizzo di una FPGA.

I programmi LabVIEW vengono denominati VI (*Virtual Instrument*) e presentano dei diagrammi di flusso con i quali sono collegate le varie funzionalità ed i vari operatori, sotto forma di icone, eseguiti secondo l'ordine del linguaggio *dataflow* (flusso di dati).

Il codice di programmazione è gestito dall'interfaccia del *Block Diagram*, in cui vengono effettuate le connessioni logiche, e non, fra gli operatori in modo da strutturare il codice. Il diagramma di flusso che si genera rappresenta un codice sorgente in formato grafico, non visualizzato all'utente. L'interfaccia grafica di uso di LabVIEW è, invece, il *Front Panel*, dove il programmatore gestisce quali risultati mostrare a video ed in che modalità farlo.

Capitolo 4

Codice LabVIEW per l'acquisizione dei dati

Una volta resi chiari i limiti del sistema di misurazione della pressione utilizzato al CICLoPE è stato necessario sviluppare un programma che si adattasse alla nuova strumentazione, consentendo di acquisire dati dal trasduttore MKS e di gestire, allo stesso tempo, la rotazione della *Scanivalve*.

A tal fine è stato creato uno schema generale di VI (*Virtual Instrument*) su LabVIEW. Lo schema è stato poi implementato secondo due modelli di acquisizione e di generazione di output: il sistema NI-DAQmx della *National Instruments* e quello dell'FPGA.

4.1 Schema generale del programma

L'idea di partenza per sviluppare il programma è stata quella di dividerlo nelle seguenti parti:

1. Inserimento di input da parte dell'utente del numero di *samples* che si vogliono acquisire per ogni porta, specificando quante porte della *Scanivalve* saranno utilizzate e con quale frequenza di acquisizione si vogliono eseguire le misurazioni;
2. Implementazione del modello di acquisizione (NI-DAQmx e FPGA);
3. Creazione di singoli array per ogni porta e successivo raggruppamento in una tabella;
4. Implementazione del *triggering* per generare un output di voltaggio al fine di movimentare la *Scanivalve* (anche questo presenta due diversi metodi in base al modello NI-DAQmx o FPGA);

5. Aggiunta di sistemi di scrittura dati su Hard Disk per consentire il *post processing* ed avere la storia temporale;
6. Condizione di uscita dal *while loop* una volta che sono state effettuate tutte le misurazione per ogni porta oppure su comando da parte dell'utente.

Il programma è inserito all'interno di un *while loop*, come già accennato. In questo modo è possibile eseguirlo in maniera continua.

4.1.1 Block Diagram

L'interfaccia utilizzata per definire, tramite uno schema a blocchi, le operazioni che devono essere compiute dal sistema viene denominata *Block Diagram*. Il *Block Diagram* utilizza la *Function Palette* in cui sono raggruppate le varie categorie di cui si può usufruire. Per questo tipo di applicazione, come può essere visto nello schema presentato nelle appendici, si è ipotizzato un segnale simulato (DC con un rumore di fondo uniforme) il quale può essere trovato nel blocco di *Input* nell'ambiente *Express*, i cui parametri possono essere regolati dall'utente.

Il segnale in corrente (simulato) che si acquisisce può essere poi visualizzato nel *Front Panel* tramite un grafico (*Waveform graph*). Per ottenere i valori di pressione che vengono effettivamente misurati dall'MKS si ricorre all'utilizzo del *Polynomial Evaluation VI*, già presente all'interno di LabVIEW, nel quale vengono richiesti i coefficienti di calibrazione (specificati dall'equazione (3.3)) ed il valore di voltaggio da convertire in pressione.

Prima *case structure* per costruzione dell'array della singola porta

A questo punto è possibile inserire i dati acquisiti all'interno di una array 1D attraverso la prima *case structure*. Questo tipo di struttura permette di definire diversi casi (solitamente due: caso vero e caso falso) e le relative operazioni da eseguire. In corrispondenza del verificarsi una condizione specificata, il programma seleziona un caso od un altro.

Per il programma analizzato la condizione da soddisfare è quella di uguaglianza tra numero di *samples* da acquisire per ogni porta e lunghezza dell'array (caso vero). La *case structure* permette di inserire il dato di pressione acquisito nella posizione successiva dell'array, qualora l'uguaglianza non sia verificata. L'inserimento avviene con il blocco *replace array subset*, in cui viene sostituito il valore ottenuto allo zero dell'array inizializzato. Nel caso vero, si ha che l'array relativo ad una determinata porta della *Scanivalve* è

completo, di conseguenza il sistema azzerava il contatore della posizione, ricominciando ad inserire i valori di pressione in un nuovo array.

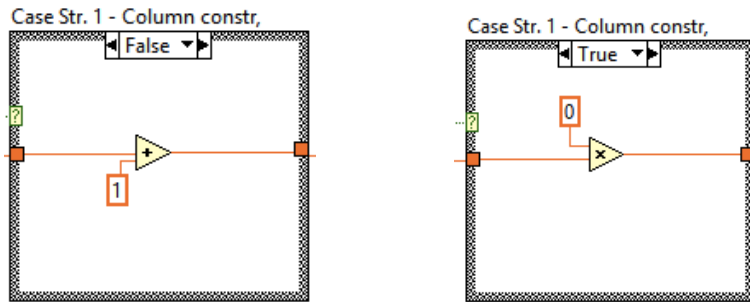


Figura 4.1: Case structure nel caso falso (a sinistra) e nel caso vero (a destra)

E' all'interno di questa struttura che si colloca, idealmente, l'operazione di triggering. Infatti, nel caso in cui la condizione sia verificata, si vuole che il programma faccia ruotare la *Scanivalve* per effettuare le misurazioni relative alla porta successiva.

Seconda *case structure* per la creazione dell'array in output

Una volta organizzati i valori per una porta è possibile utilizzare una seconda *case structure* per creare un array 2D in cui visualizzare tutte le misurazioni effettuate per le diverse porte, in un'interfaccia (*Front Panel*) che viene mostrata a video all'utente in maniera più immediata tramite un *output array* (nel VI in esame questo prende il nome di *Pressures Array*).

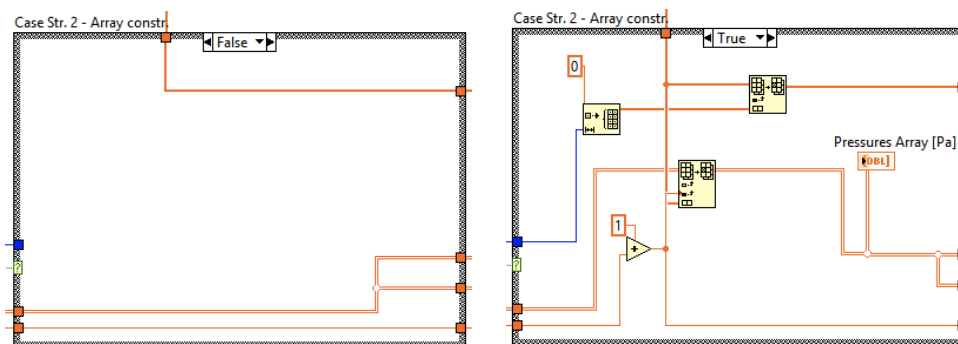


Figura 4.2: Seconda case structure nel caso falso (a sinistra) e nel caso vero (a destra)

Il funzionamento è consequenziale a quello della prima *case structure*. La condizione da verificare è di nuovo l'uguaglianza tra il numero di *sample* e la lunghezza dell'array 1D (in questo caso, però, a valle dell'incremento del contatore). Nel caso in cui la condizione sia vera, il programma inserisce nella colonna l'array 1D dei valori misurati per la porta rispettiva, anche in questo caso con un blocco *replace array subset*, in cui le dimensioni delle righe e delle colonne sono specificate dai contatori. Nel caso in cui la condizione non sia verificata la struttura viene bypassata, senza effettuare cambiamenti.

Condizione di uscita dal *while loop*

Poichè si sta trattando un *while loop* è necessario inserire la condizione di uscita, affinché tutto il programma possa essere eseguito correttamente.

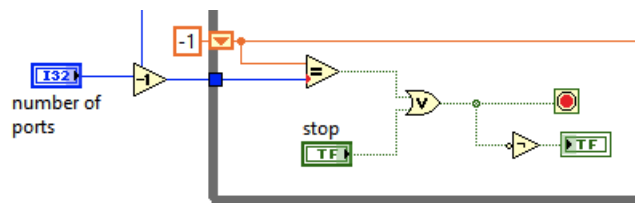


Figura 4.3: Schema logico della condizione di uscita dal *while loop*.

Ciò che si vuole è che il programma smetta di acquisire nel caso in cui siano state effettuate tutte le misurazioni per tutte le porte che sono state scelte oppure nel caso in cui sia l'utente stesso a voler interrompere il processo di acquisizione. Queste specifiche sono visualizzabili nella figura 4.3. Si può notare anche l'inserimento di un output booleano la cui funzione verrà spiegata nel paragrafo riguardante il *Front Panel*.

Output di scrittura su hard disk

Al fine di rendere possibile l'operazione di *post processing* è necessario implementare dei blocchi che permettano la scrittura dei dati acquisiti in formati fruibili da altri software (per esempio Matlab, utilizzato per l'elaborazione dati in questo caso).

L'array di pressioni può essere quindi scritto, nella forma in cui viene visualizzato sul *Front panel*, in un formato di tipo *.txt*, tramite l'utilizzo della VI di LabVIEW *write to spreadsheet file*, nella quale viene chiesto di specificare il percorso in cui salvare il file ed in cui si collega l'*output array* in uscita dal loop.

Un altro formato in cui viene salvato l'insieme di valori acquisiti può essere il formato excel (.xlsx). Salvando i dati in questo tipo di formato, tramite il blocco *write to measurement file*, è possibile avere a disposizione la storia temporale del processo, aspetto fondamentale durante le procedure sperimentali. Questo strumento richiede all'utente di indicare le varie specifiche da seguire, oltre al formato, per lo *storage* dei dati (quali percorso, sovrascrittura, simbologia..).

4.1.2 Front Panel

In relazione al *Block Diagram*, l'interfaccia che viene visualizzata dall'utente è denominata *Front Panel*. In questo ambiente è possibile gestire gli input e gli output dati al programma, senza però visualizzare le relazioni e le interazioni che si hanno per ottenere i risultati mostrati a video.

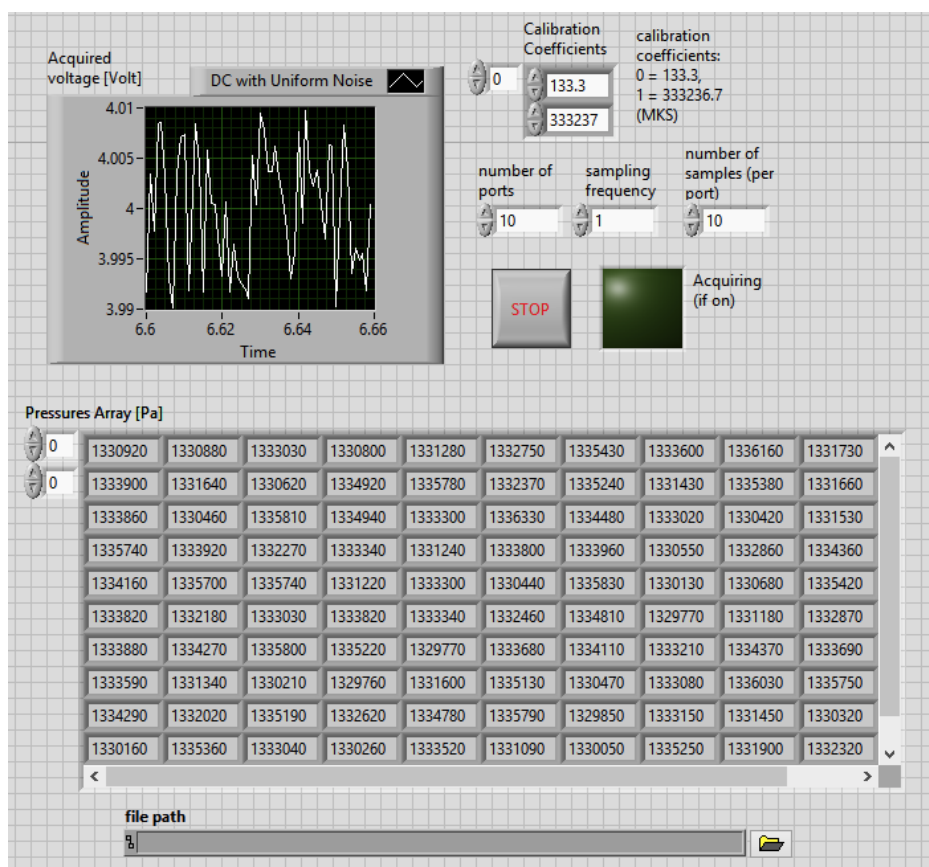


Figura 4.4: Il *Front Panel* del programma di acquisizione progettato per il laboratorio CICLoPE. L'immagine riporta dei dati ottenuti dopo una simulazione.

L'interfaccia utente del programma creato per il CICLOPE permette a chi lo utilizza di (vedere figura 4.4):

- Inserire i coefficienti di calibrazione (fissati nel caso dell'MKS), i quali permettono di ottenere i valori di pressione che vengono poi inseriti nell'array di output finale (*Calibration coefficients*);
- Specificare quanti campioni si vogliono acquisire per ogni singola porta (*number of samples - per port*) e di quante porte della *Scanivalve* si voglia acquisire la pressione (*number of ports*);
- Impostare la frequenza alla quale si vuole effettuare l'intero processo, quindi ogni quanto si voglia ottenere il valore della pressione (*sampling frequency*);
- Fermare l'acquisizione, premendo il pulsante di stop;
- Capire se il sistema sta acquisendo tramite l'interruttore booleano, che rimane acceso fino a quando non viene soddisfatta la condizione di uscita dal *while loop* del *Block Diagram*;
- Visualizzare l'andamento dell'input di corrente prima che venga trasformato in pressione tramite il grafico dell'*Acquired Voltage*;
- Visualizzare a video l'array di pressioni in output (*Pressures Array*). Nella figura 4.4 è l'array è compilato da valori di pressioni dovuti alla simulazione di un segnale di voltaggio in ingresso;
- Specificare il percorso in cui salvare il file con i valori dell'array in formato *.txt* (*file path*).

4.2 Acquisizione e triggering con NI-DAQmx

Nel campo dell'acquisizione di un segnale una delle ultime innovazioni è quella di utilizzare l'insieme hardware NI-DAQmx, il quale consente di ottenere dati tramite sensori (nel caso analizzato, questo è il trasduttore capacitivo MKS), che possono essere trasmessi direttamente sul programma LabVIEW tramite il blocco di input chiamato *DAQ Assistant*.

Il procedimento risulta molto semplice poichè non richiede passaggi attraverso i vari livelli di acquisizione (come sarà invece il caso dell'FPGA, descritto nel paragrafo successivo) ma permette di far funzionare tutto con gli strumenti disponibili all'interno di un unica VI.

Una volta che il dispositivo di misurazione è stato connesso tramite USB

DAQ, od anche attraverso Ethernet, al cRIO di cui si dispone, questo sarà visualizzabile nella parte che comprende i *Remote system* del MAX (Measurement & Automation Explorer) e sarà quindi configurato dal sistema.

Acquisizione con NI-DAQmx

Dopo essersi assicurati che l'entrata fisica sia effettivamente ricevuta dal computer è possibile inserire nel VI il blocco *DAQ Assistant*, che può essere trovato negli input della categoria *Express*. Nello schema generale del programma questo prenderà il posto dell'input simulato.

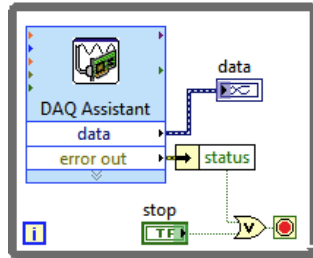


Figura 4.5: Il blocco del *DAQ Assistant*: nell'esempio viene evidenziata l'uscita dedicata ai dati acquisiti. Esempio del sito della National Instruments.

Fatto questo sarà richiesta la compilazione per configurare il tipo di *DAQ Assistant*. Sarà necessario:

- Specificare la funzionalità, che sarà quella di acquisizione (*Acquire signal*);
- Selezionare il tipo di input, se analogico o digitale;
- Definire quale input sarà inviato al cRIO, nel caso dell'MKS sarà un voltaggio;
- Selezionare il nome del dispositivo e l'ingresso al quale corrisponde.

Successivamente sarà possibile configurare le impostazioni di acquisizione quali il numero di *sample* da acquisire ed il *sample rate* ed inserire un grafico che renda possibile la visualizzazione della grandezza acquisita sul *Front Panel* tramite il comando del *Graph Indicator*.

Triggering con NI-DAQmx

L'utilità del triggering è quella di consentire l'acquisizione o la generazione di un output ogni volta che una condizione viene soddisfatta.

Nel caso del programma sviluppato si vuole che il triggering produca un segnale di voltaggio nel momento in cui il numero di misurazioni effettuate per una singola porta risultino uguali a quelle specificate dall'utente. Per questo motivo, si può pensare di collocare, in maniera ideale, l'intero processo di triggering all'interno della prima *case structure*. La produzione dell'output di voltaggio permette, passando attraverso la *Scanivalve*, di imprimerle una rotazione che faccia iniziare l'acquisizione per la porta successiva.

Sarà necessario, anche in questo caso, accertarsi di aver configurato lo strumento al quale si vuole inviare il segnale ed assicurarsi che il terminale sia adatto al canale di input/output fisico da controllare.

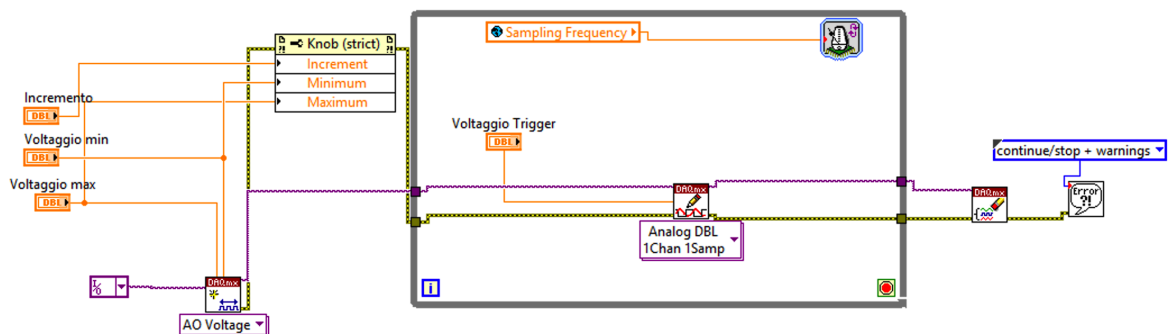


Figura 4.6: Schema ideato per il triggering tramite i blocchi NI-DAQmx (ispirato ad un esempio sul forum NI).

Nell'esempio riportato in figura 4.6 si può individuare il procedimento tramite il quale il VI implementa il triggering:

- La creazione dell'output di voltaggio per il relativo canale. In questa parte si possono definire i valori minimi e massimi di voltaggio prodotti, inserendo anche un incremento tra un valore e l'altro. L'output è generato grazie all'utilizzo del blocco *AO Voltage*, sempre facente parte degli strumenti offerti dall'hardware NI-DAQmx;

- Il passaggio dell'output generato attraverso il blocco *Analog DBL 1Chan 1Samp* con il fine di limitare il singolo output al canale indicato, effettuando un controllo sullo stato del triggering.
- L'azione di pulizia del comando di output, una volta usciti dal loop. E' presente, inoltre, un blocco che mostra a video la presenza di errori nel caso in cui questi vengano a verificarsi e, quindi, l'interruzione dell'esecuzione del programma.

4.3 Acquisizione e triggering con FPGA

Come già introdotto nel paragrafo 3.2.3, l'FPGA è un circuito miniaturizzato con la capacità di essere riprogrammabile attraverso la compilazione di un file, chiamato bitstream, e capace di effettuare le operazioni più semplici, tramite un linguaggio medio/basso. E' un sistema deterministico in quanto, una volta implementata la funzione che si vuole che questo abbia, il suo comportamento non varia a meno che non venga ricompilato.

Questa tecnologia si basa su un sistema integrato che comprende diversi strati. Infatti, una volta compilato il codice questo può essere riutilizzato all'interno di un VI Real-Time che consente di utilizzare un numero maggiore di funzionalità tramite un linguaggio di alto livello. In questa applicazione, l'FPGA ha lo scopo di rielaborare in maniera più rapida i dati ottenuti, senza però dover passare dalla CPU, quindi dalla scrittura su hardware. Il CPU consente di mostrare a video i risultati ottenuti e di inviare comandi al processore Real-Time ed all'FPGA.

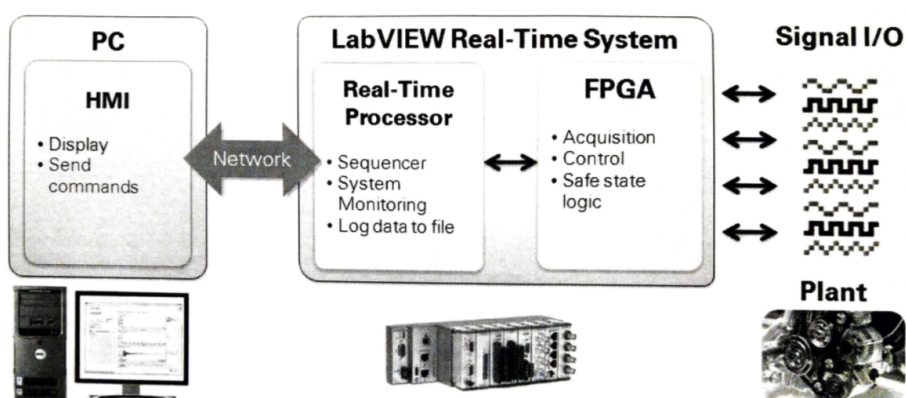


Figura 4.7: Schema di un sistema di controllo integrato. Sono evidenziate le funzioni dell'FPGA e del processore Real-Time e come queste comunicano con una CPU. Immagine da *Embedded Control and Monitoring Using LabVIEW* (2015)

Nel caso analizzato del CICLoPE, poichè si ha a disposizione un cRIO 9068 che non dispone della tecnologia NI-DAQmx, è necessario ottenere input e generare output tramite la metodologia dell'FPGA, la quale consente di ottenere gli stessi risultati di acquisizione e controllo con un procedimento leggermente più complesso.

Acquisizione con FPGA

L'acquisizione è resa possibile, tramite FPGA e LabVIEW, grazie all'utilizzo degli *C Modules* i quali sono dei moduli digitali da inserire all'interno degli slot del cRIO 9068 in modo da poter essere configurati all'interno dal sistema e, successivamente, dal computer.

Al fine di modificare un circuito FPGA per la specifica funzione, una volta iniziato un nuovo progetto sul software LabVIEW, sarà necessario compilare una VI che prende il nome di *FPGA target*, con lo stesso aspetto di una VI Real-Time ma con operazioni più limitate.

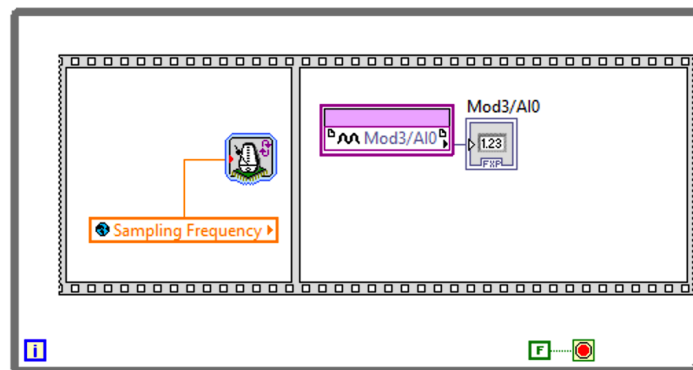


Figura 4.8: Creazione dell'FPGA target che utilizza un input analogico.

All'interno del progetto è possibile trovare il modulo al quale ci si sta riferendo e l'ingresso interessato (in base al cablaggio effettuato dall'utente), visualizzando il tipo di input che si vuole ottenere. Nel caso specifico dell'MKS, l'input ottenuto sarà un voltaggio compreso tra i valori di 0 e 10 Volt, ottenuti tramite il modulo NI 9215, a quattro ingressi, collegato direttamente al trasduttore, quindi a monte della conversione del segnale nel dato di pressione, il quale sarà implementato direttamente dal VI principale. Trascinando il blocco interessato all'interno del *Block Diagram* è possibile implementare le operazioni di basso livello che sono richieste all'FPGA per poi compilare il circuito fisico.

A questo punto è possibile inserire il *target FPGA* (che diventerà l'input del programma) all'interno dello schema generale del VI Real-Time, sostituendolo al segnale simulato, per consentire al sistema di acquisizione della pressione di effettuare le operazioni desiderate.

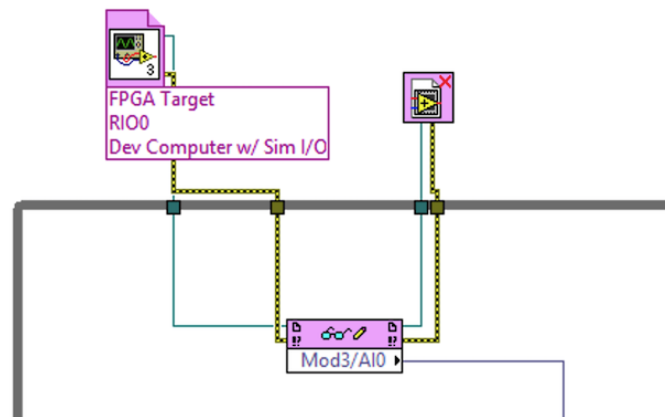


Figura 4.9: Implementazione dell'FPGA target all'interno del programma generale (Real-Time VI). Il segnale di voltaggio acquisito viene poi convertito in pressione passando attraverso la curva di calibrazione.

Triggering con FPGA

Con le stesse modalità definite dall'acquisizione è possibile generare un output verso un sistema fisico esterno.

In questo caso il modulo di interesse dovrà generare un output che consenta la movimentazione della *Scanivalve*.

I moduli di interesse, nel caso del CICLoPE, sono l'NI 9403 e l'NI 9401 (ad alta velocità), i quali producono un output digitale di voltaggio di 5 Volt.

Una volta compilata la VI relativa all'*FPGA target* è possibile richiamare il blocco all'interno della VI principale per ottenere la movimentazione voluta.

Inserendo il target all'interno della prima *case structure* è possibile cambiare la porta di acquisizione della *Scanivalve* ogni qual volta sia soddisfatta la condizione di ingresso alla struttura, cioè quando il numero di campioni acquisiti risulta uguale a quelli specificati dall'utente.

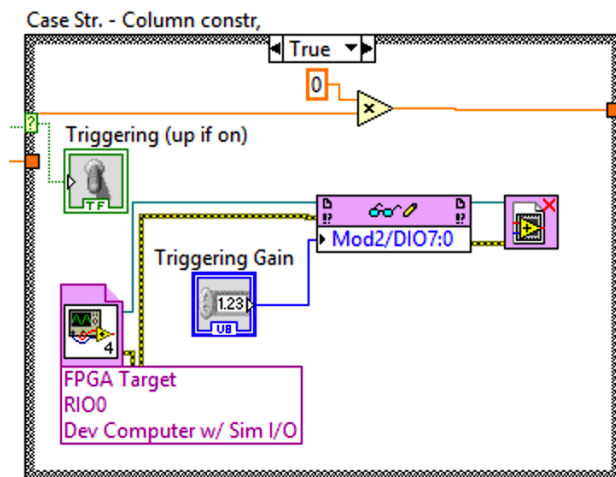


Figura 4.10: FPGA Target per il triggering, collocato all'interno della prima *case structure*. E' possibile specificare il valore di voltaggio che si vuole dare in output.

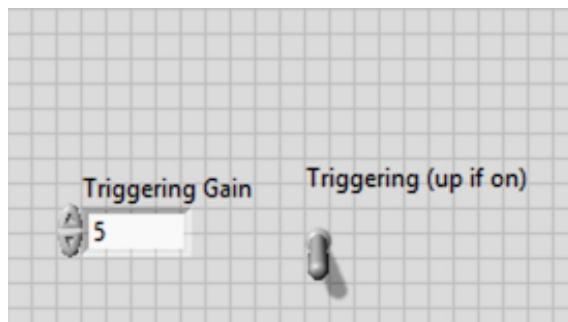


Figura 4.11: Aggiunta nel *Front Panel* dell'indicatore di *triggering* e dell'input di gain da assegnare.

4.4 *Post-processing* dei dati

Un altro aspetto importante del programma creato è la possibilità di scrittura dei dati ottenuti sotto dei formati (.txt, .xlsx) riutilizzabili da altri software per effettuare una successiva elaborazione.

Come già detto, la necessità di avere una storia temporale delle misurazioni effettuate è di fondamentale importanza nell'ambito sperimentale.

Un altro scopo della trascrizione dei dati può essere quello del *post-processing* dai dati su software come Matlab.

Dato l'array di pressioni in output si vuole conoscere il valore medio delle pressioni per ogni singola porta, in modo da poter visualizzare l'andamento di questa grandezza lungo tutto il *pipe* ed avere un'idea dell'errore di misura per ognuno di questi punti. Come già ampiamente spiegato, l'incertezza attorno alla misura, nel caso del sistema MKS-*Scanivalve*, rispetto alle misurazioni effettuate dall'Initium, dovrebbe risultare minore e costante lungo tutto il condotto.

Calcolo della media e rimozione delle ampie oscillazioni sulla misura di pressione

Si è partiti da uno script utilizzato al CICLoPE¹ per rimuovere le oscillazioni troppo ampie che si vengono a registrare durante le misurazioni dell'MKS. Si rimuovono queste oscillazioni poichè, discostandosi in maniera troppo marcata dall'andamento medio delle pressioni, non sono di interesse per l'analisi effettuata. Il modello di partenza è stato poi integrato in modo da poter ottenere, su richiesta dell'utente, il valore medio della porta della *Scanivalve* ed il vettore di pressioni, corretto dalle oscillazioni, della rispettiva porta.

Calcolo del valore medio della velocità

Successivamente, potendo accedere ai valori di pressioni medi di ogni porta, è stato creato un altro script per visualizzare il valore medio della velocità alla fine del condotto, quindi a valle delle cadute di pressione, tramite la misura delle porte della *Scanivalve* collegate alla pressione totale ed alla pressione statica del tubo di Pitot.²

4.5 Complessità computazionale del programma

4.5.1 Tempi di esecuzione e *buffering*

Il programma descritto lavora salvando i dati acquisiti, raggruppati nell'array di pressioni, direttamente sulla memoria RAM dell'elaboratore, la quale rappresenta un tipo di memoria più volatile, che funziona unicamente qualora il computer sia acceso. Infatti, nel caso in cui il sistema si interrompa in maniera inaspettata, tutti i dati raccolti fino a quell'istante sarebbero facilmente perduti.

¹Script creato da L.Mascotelli, dottoranda presso l'Università di Bologna

²Gli script sono riportati nelle appendici.

Ad esempio, considerando l'applicazione al centro CICLoPE, è stata analizzato un processo di acquisizione di 48 diversi canali di pressione, pensato per ottenere 60 misurazioni con la frequenza di 1 Hz, quindi registrando un campione di pressione ogni secondo. Nel caso in cui il processo venga portato a termine secondo questa modalità, la durata complessiva sarebbe di 48 minuti. Per altri tipi di acquisizione i tempi possono essere anche più lunghi, quindi il rischio di perdere dati in caso di imprevisti risulta ancora di maggiore rilevanza.

Per evitare che ciò avvenga è possibile collezionare i dati seguendo il metodo chiamato *buffering*.

Il *buffering* è sostanzialmente un processo che utilizza salvataggi periodici di dati su file che vengono sovrascritti sull'hard disk, per ovviare ai problemi che possono derivare dalle diverse velocità di trasferimento dati dei vari livelli di cui è composto il programma. Infatti, il metodo comunemente utilizzato per trasferire i dati elaborati dal *buffering* tra gli elementi di un FPGA target o anche tra il VI dell'FPGA ed il VI Real-time, è il FIFO (first-in-first-out) *buffer*. Questo sistema assicura un comportamento deterministico, cioè nel quale i valori di input ed output rimangono solitamente fissati, consentendo sia all'utente che al programma di accedere ai dati simultaneamente.

Un primo sistema di scrittura simultanea su hard disk può essere quello già implementato dal blocco *write to measurement file*.

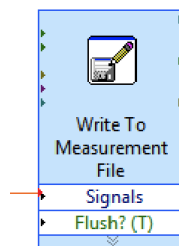


Figura 4.12: Blocco *write to measurement file*, dalla categoria *Express*.

Infatti, se collegato direttamente al segnale di pressione in uscita, anche se il processo viene abortito improvvisamente, i dati acquisiti fino a quell'istante temporale rimarranno comunque reperibili all'interno del file *excel*.

Ciò che invece non viene recuperato sono i dati ordinati all'interno dell'array di pressioni misurate, poichè vengono sovrascritti unicamente nel caso in cui siano soddisfatte le condizione di uscita dal *while loop*.

4.5.2 Occupazione di memoria

Guardando un codice, è anche necessario capire quale sia la sua complessità computazionale sul piano della memoria utilizzata, dato utile a valutarne l'efficienza, l'affidabilità e la sua capacità di eseguire operazione in maniera deterministica.

Nel caso in cui un sistema non abbia sufficiente memoria a disposizione per completare un'operazione, questa potrebbe non essere eseguita e quindi fallire. Causa di questo problema è in alcuni casi la perdita di memoria graduale. Parte della memoria utilizzata dal programma è quella caratterizzata dal pre-allocaimento, la quale va a sommarsi a quella già allocata dal programma prima che questo venga eseguito. Fin dall'inizializzazione è, quindi, possibile ottenere informazioni utili riguardo all'eseguitabilità dell'applicazione.

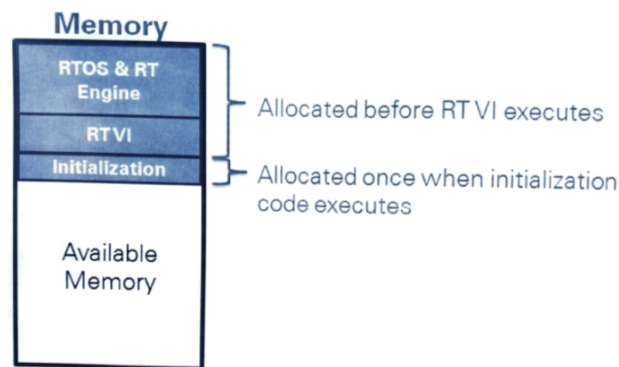


Figura 4.13: Schema di suddivisione della memoria. Immagine da *Embedded Control and Monitoring Using LabVIEW* (2015)

Oltre al pre-allocaimento è utile:

- Eliminare copie di dati non utilizzate. Ciò è possibile tramite lo strumento *show buffer allocations*, che mostra in quali punti dell'applicazione il software crea dei *buffer* per utilizzare lo spazio al fine di memorizzare i dati. Una volta individuati, è possibile modificare il VI di conseguenza;
- Eliminare dei driver non necessari o non utilizzati direttamente dal MAX;
- Utilizzare un target hardware con maggiore memoria disponibile;

- Nel caso di allocazione dinamica della memoria è possibile effettuare il *reboot* del sistema (dove supportato), capace di liberare spazio in maniera continua basandosi sull'utilizzo nel tempo di questi dati.

Nel programma sviluppato, i problemi di memoria potrebbero essere dati dalla mole in byte elaborate dall'hard disk, ma soprattutto dalla RAM. Un primo step per evitare la frammentazione della memoria ed il fenomeno del *jitter* (differenza di tempo tra i cicli del loop) è stato quello di pre-allocare gli array prima di entrare nel loop.

Capitolo 5

Sviluppi futuri e conclusioni

5.1 Sviluppi futuri

Lo sviluppo futuro del progetto risiede nell'applicazione pratica del codice al centro CICLoPE. Non avendo effettuato misurazioni dirette al *long pipe* è stato necessario applicare il codice LabVIEW in versione simulata, quindi senza una acquisizione real time delle grandezze fisiche ma gestendo dati simulati, sia in input che in output.

Nonostante il codice si basi pienamente sui sistemi adoperati, sia nei segnali di ingresso che d'uscita, con una simulazione esatta dei componenti *C-series* connessi al cRIO, si necessita in ugual modo una verifica con misurazioni reali. Da queste sarà possibile verificare l'implementazione teorica del programma e comprendere quali errori possano essere corretti tramite un'operazione di *debugging*.

5.2 Conclusioni

Lo scopo principale del progetto è stata la creazione di un sistema che consentisse misurazioni più precise lungo il *long pipe* del centro CICLoPE, al fine di poter arrivare ad una descrizione più dettagliata del flusso turbolento completamente sviluppato all'interno dello stesso. Per arrivare a ciò si è pensato di cambiare il sistema di acquisizione delle pressioni del laboratorio, in modo da limitare il più possibile gli errori di misura legati all'analisi della caduta di pressione e dunque alle tensioni tangenziali all'interno del condotto.

Si è passati dal sistema *DTC Initium* che associa ad ogni sensore uno scanner di pressione al trasduttore *MKS Baratron* integrato con un sistema multiplexer *Scanivalve* con la funzione di collegamento fra ogni sensore e il trasduttore.

Avendo il sistema *DTC Initium* un software per l'acquisizione dei dati separato, è stato necessario sviluppare un nuovo metodo che adempisse a questa funzione.

Si è adoperato uno chassis *Compact RIO 9068* con annessi moduli *C-Series*, a cui è stato collegato in input l'MKS e in output un segnale di *triggering* a movimentare la scanivalve.

Per la ricezione e l'analisi dei dati e per l'invio di segnali digitali in output si è creato un codice su software *LabVIEW* che consentisse l'acquisizione e lo storing dei dati lungo i 48 sensori di pressione del centro CICLoPE e che gestisse il movimento della scanivalve in modo da essere collegati alle diverse uscite di pressione lungo il *pipe*.

Il codice è stato analizzato in modalità 'simulazione'. Deve perciò essere testato, in modo da valutarne il funzionamento e l'effettivo miglioramento nell'acquisizione della pressione, rapportato al precedente sistema.

Bibliografia

- A-Tech Instruments Ltd, *DTC Initium Manual*
- Buresti G., *Elements of Fluid Dynamics*, London, Imperial College Press, (2012)
- *CICLoPE website*, [http : //www.ciclope.unibo.it/lab_temp.htm](http://www.ciclope.unibo.it/lab_temp.htm), (2004-2012)
- *CICLoPE website*, [http : //www.ciclope.unibo.it/long_temp.htm](http://www.ciclope.unibo.it/long_temp.htm), (2004-2012)
- Fiorini T., *Turbulent Pipe Flow - High Resolution Measurements in CICLoPE*, (2017)
- MKS Instruments Inc. *MKS Baratron® Type 120A Pressure Transducer. Instruction Manual*, (2013)
- National Instruments, *Embedded Control and Monitoring Using LabVIEW. Course Manual*, (2015)
- National Instruments, *NI cRIO-9068. User Manual*, (2016)

- Pope S. B., *Turbulent Flows*, Cambridge, Cambridge University Press, (2000), sesta edizione
- Talamelli A., *Lezioni di aerodinamica, appunti, anno accademico 2018/19*
- Talamelli A., *Laboratorio di Aerodinamica Sperimentale, appunti e dispense, anno accademico 2018/19*
- Yarin L. P. *The Pi-Theorem. Applications to Fluid Mechanics and Heat and Mass Transfer*, Israel, Springer (2012)

Ringraziamenti

Vorremmo ringraziare il professor G. Bellani e la professoressa L. Mascotelli per averci dato l'occasione di poter approfondire determinati aspetti legati all'aerodinamica in un ambiente di ricerca quale il CICLoPE e comprendere gli aspetti legati al mondo sperimentale. Li ringraziamo inoltre per la loro disponibilità e cura dedicatoci durante tutto il percorso.

Un grazie speciale a tutte le persone che in una qualche maniera ci hanno sostenuto durante questo percorso triennale.

In particolar modo, alle nostre famiglie, che ci hanno permesso di seguire questa strada, senza impedimenti.

Elenco delle figure

1	Grafico che mostra la risoluzione spaziale (in μm) in funzione del <i>friction Reynolds number</i> (R^+) (il significato di questa grandezza sarà discusso nel paragrafo 1.2.2) nei diversi <i>wind tunnels</i> . Evidenziata l'area di interesse del CICLoPE. Immagine del sito del laboratorio CICLoPE	2
1.1	Storia temporale di una componente assiale di velocità di un getto turbolento. Immagine dell'esperimento di Tong e Warhaft (1995) presa da 'Turbulent Flows', di Stephen B. Pope (2000)	6
1.2	Schema di un <i>pipe</i> . Immagine presa da 'Turbulent Flows', di Stephen B. Pope (2000)	7
1.3	Andamento del profilo medio di velocità di un flusso lungo un condotto. Evidenziate la regione di ingresso e la regione di flusso completamente sviluppato	8
1.4	Contributi di stress viscoso e di Reynolds rapportati allo stress di taglio totale. Linea tratteggiata, $Re=5600$; linea continua, $Re=13750$. DNS data di Kim <i>et al.</i> (1987). Immagine da 'Turbulent flows', di Stephen B. Pope (2000).	10

1.5	Profili di velocità medie per flusso completamente sviluppato all'interno di un condotto. A sinistra si ha il caso generale a destra il caso per $y/R < 0.1$. Sono evidenziati i dati per diversi numeri di Reynolds. Dati sperimentali di Zagarola e Smits (1997). Immagine da 'Turbulent flows', di Stephen B. Pope (2000).	13
2.1	Sensore wall tapping	16
2.2	Schema di un condensatore a facce parallele.	18
2.3	Funzione continua e suo campionamento.	21
2.4	Errore di aliasing. Immagine da Talamelli A., <i>Laboratorio di Aerodinamica Sperimentale, appunti e dispense, anno accademico 2018/19</i>	22
3.1	Schema del tubo di Pitot. Evidenziate le prese statiche, la presa totale ed il collegamento con il trasduttore di pressione	24
3.2	Initium utilizzato al ciclope. Immagine da <i>DTC Initium Manual</i>	25
3.3	Initium Mainframe. Immagine da <i>DTC Initium Manual</i>	26
3.4	Deviazione standard dell'errore di pressione definita in funzione del numero di sensori su cui è stata effettuata l'interpolazione. Dalla tesi di dottorato di T. Fiorini <i>Turbulent Pipe Flow - High Resolution Measurements in CICLoPE</i>	27
3.5	<i>Scanivalve</i> usata al centro CICLoPE. Immagine dalla rivista <i>Racercar engineering. Formula 1</i> , (2018)	28
3.6	Il trasduttore MKS utilizzato per le misurazioni	29
3.7	Il cRIO NI 9068 utilizzato per le misurazioni.	31
3.8	Schema di allocazione delle porte I/O. Immagine da <i>NI cRIO-9068 user manual</i> (2016).	32

<i>ELENCO DELLE FIGURE</i>	61
3.9 Schema funzionale FPGA da <i>Control and Embedded sistem.</i>	32
3.10 Esempio di sezionamento del lavoro della CPU di un computer. Immagine dal sito <i>Tait Radio Academy.</i>	33
4.1 Case structure nel caso falso (a sinistra) e nel caso vero (a destra)	37
4.2 Seconda case structure nel caso falso (a sinistra) e nel caso vero (a destra)	37
4.3 Schema logico della condizione di uscita dal <i>while loop.</i>	38
4.4 Il <i>Front Panel</i> del programma di acquisizione progettato per il laboratorio CICLoPE. L'immagine riporta dei dati ottenuti dopo una simulazione.	39
4.5 Il blocco del <i>DAQ Assistant:</i> nell'esempio viene evidenziata l'u- scita dedicata ai dati acquisiti. Esempio del sito della National Instruments.	41
4.6 Schema ideato per il triggering tramite i blocchi NI-DAQmx (ispi- rato ad un esempio sul forum NI).	42
4.7 Schema di un sistema di controllo integrato. Sono evidenziate le funzioni dell'FPGA e del processore Real-Time e come queste comunicano con una CPU. Immagine da <i>Embedded Control and Monitoring Using LabVIEW (2015)</i>	43
4.8 Creazione dell'FPGA target che utilizza un input analogico.	44
4.9 Implementazione dell'FPGA target all'interno del programma ge- nerale (Real-Time VI). Il segnale di voltaggio acquisito viene poi convertito in pressione passando attraverso la curva di calibrazione.	45

4.10	FPGA Target per il triggering, collocato all'interno della prima <i>case structure</i> . E' possibile specificare il valore di voltaggio che si vuole dare in output.	46
4.11	Aggiunta nel <i>Front Panel</i> dell'indicatore di <i>triggering</i> e dell'input di gain da assegnare.	46
4.12	Blocco <i>write to measurement file</i> , dalla categoria <i>Express</i>	48
4.13	Schema di suddivisione della memoria. Immagine da <i>Embedded Control and Monitoring Using LabVIEW</i> (2015)	49
5.1	Schema generale del programma LabVIEW. <i>Case structures</i> entrambe nel caso <i>true</i>	67
5.2	Schema generale del programma LabVIEW. <i>Case structures</i> entrambe nel caso <i>false</i>	68
5.3	Schema FPGA per l'acquisizione.	69
5.4	Schema FPGA per il <i>triggering</i>	70
5.5	Host VI con implementazione di acquisizione e <i>triggering</i> tramite FPGA. Nell'immagine è evidenziato il caso in cui entrambe le <i>case structures</i> siano vere.	71
5.6	<i>Front Panel</i> modificato nel VI con FPGA. Si nota l'inserimento dell'indicatore booleano nel caso in cui stia avvenendo il <i>triggering</i>	72
5.7	Schermata del progetto generale con FPGA. Evidenziati i tre <i>C-Modules</i> utilizzati per simulare quelli presenti nel cRIO del CICLoPE.	72
5.8	Script ideato da L. Mascotelli per rimuovere le ampie oscillazioni ricevute dall'MKS.	75

5.9 Script adattato dal precedente. In questo è possibile anche selezionare la porta della *Scanivalve* di interesse. 76

5.10 Script per il calcolo della velocità a valle del condotto. In questo caso, le ultime due porte sono quelle dedicate all'acquisizione di pressione statica e pressione totale. 77

Appendice A

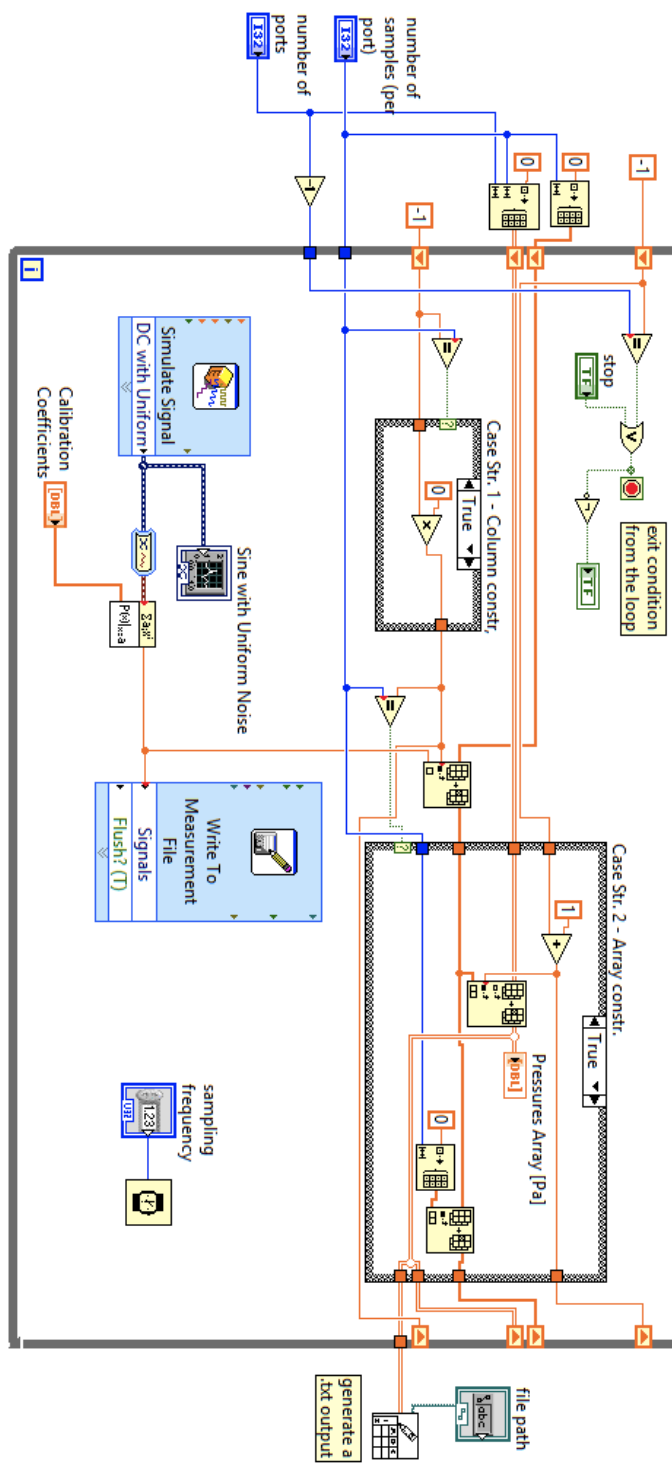


Figura 5.1: Schema generale del programma LabVIEW. *Case structures* entrambe nel caso *true*.

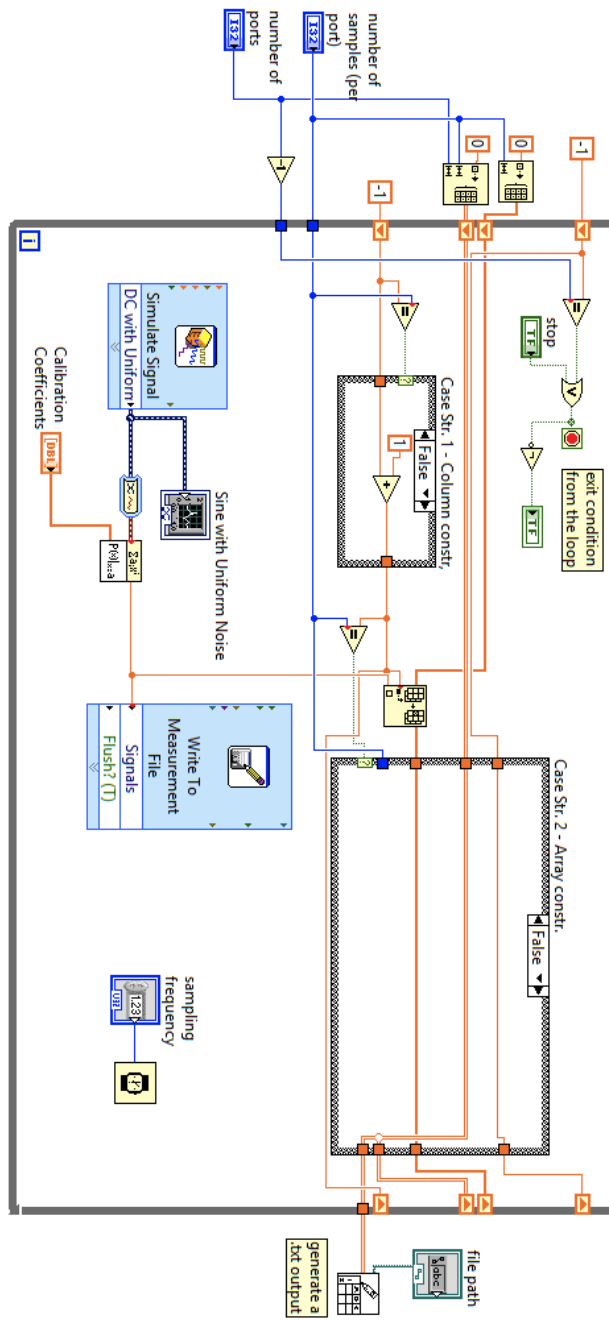


Figura 5.2: Schema generale del programma LabVIEW. *Case structures* entrambe nel caso *false*.

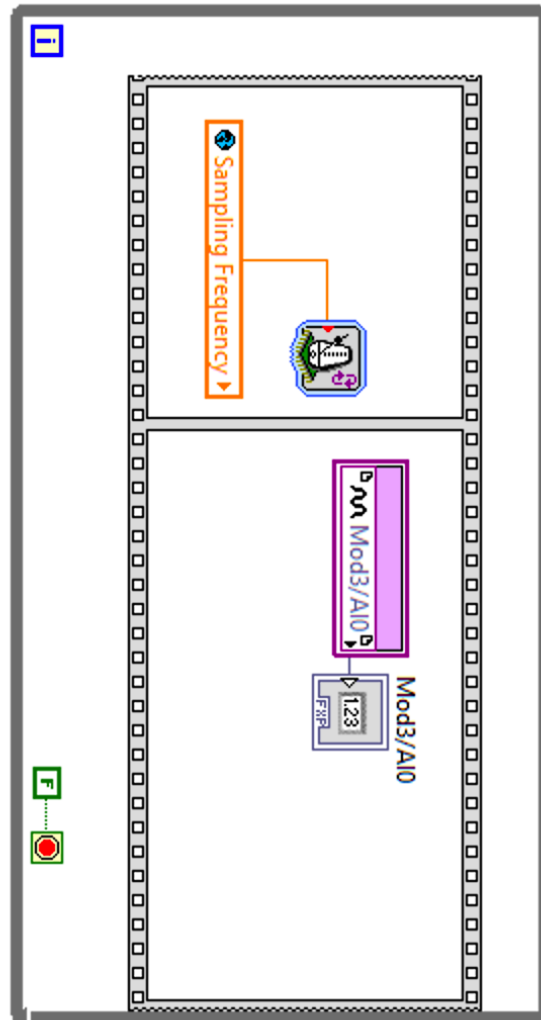


Figura 5.3: Schema FPGA per l'acquisizione.

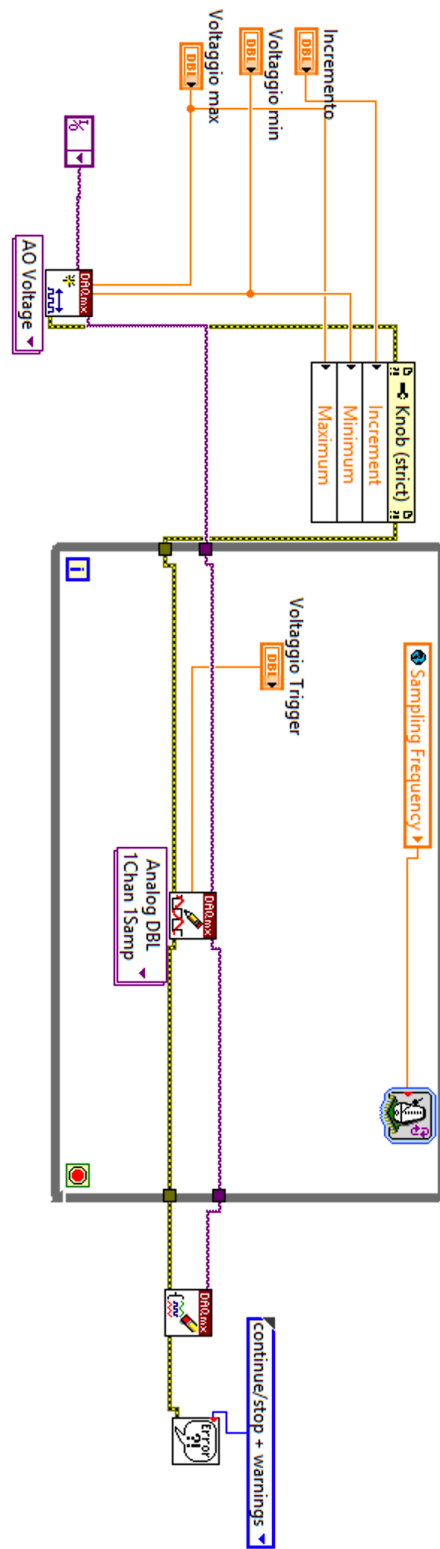


Figura 5.4: Schema FPGA per il *triggering*.

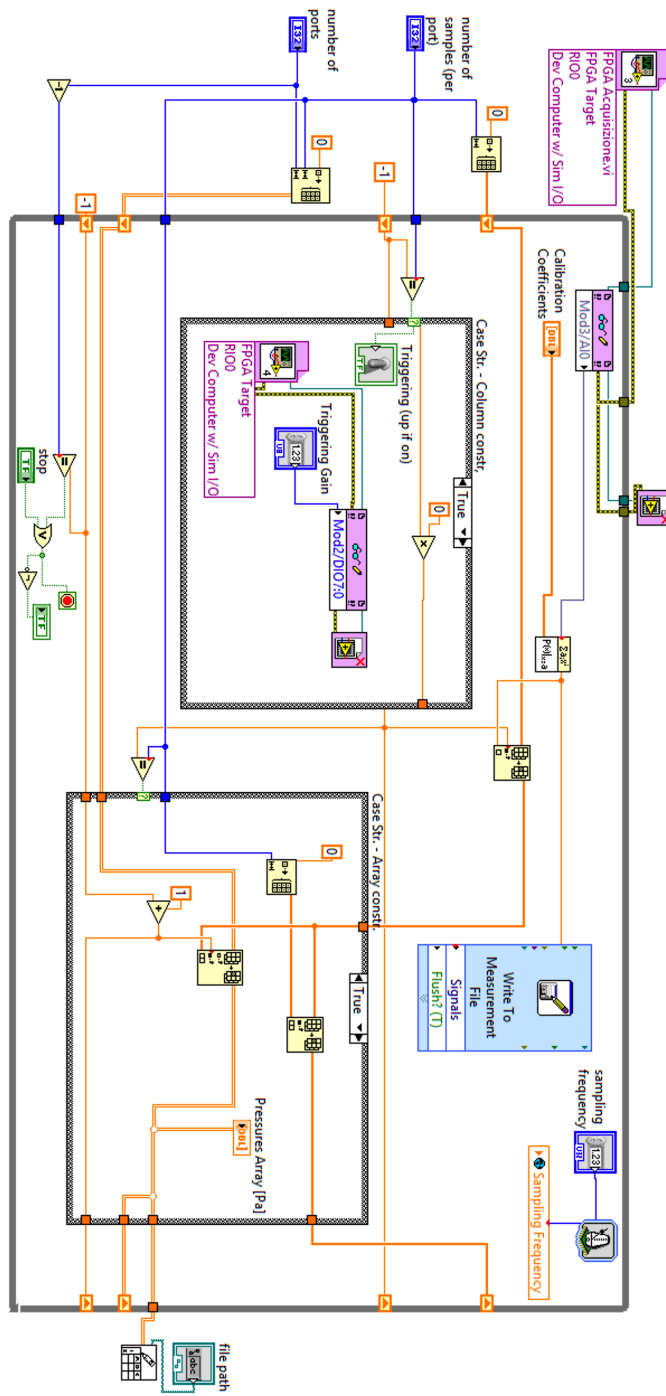


Figura 5.5: Host VI con implementazione di acquisizione e *triggering* tramite FPGA. Nell'immagine è evidenziato il caso in cui entrambe le *case structures* siano vere.

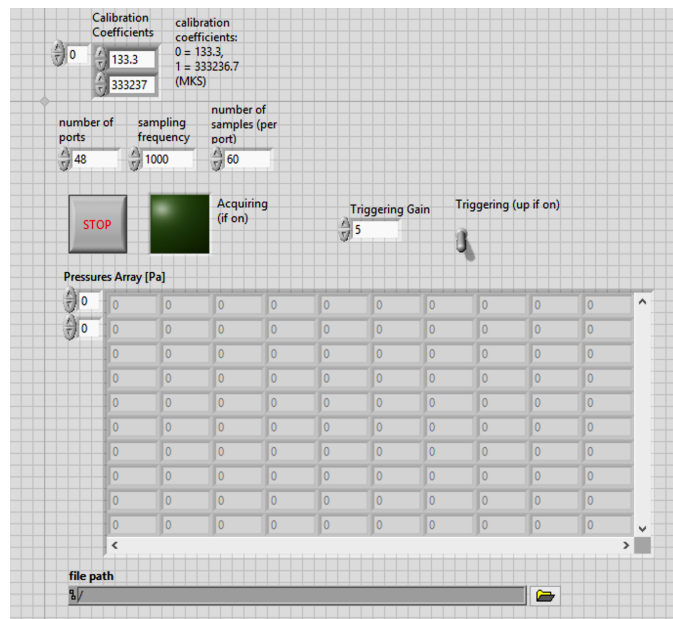


Figura 5.6: *Front Panel* modificato nel VI con FPGA. Si nota l'inserimento dell'indicatore booleano nel caso in cui stia avvenendo il *triggering*.

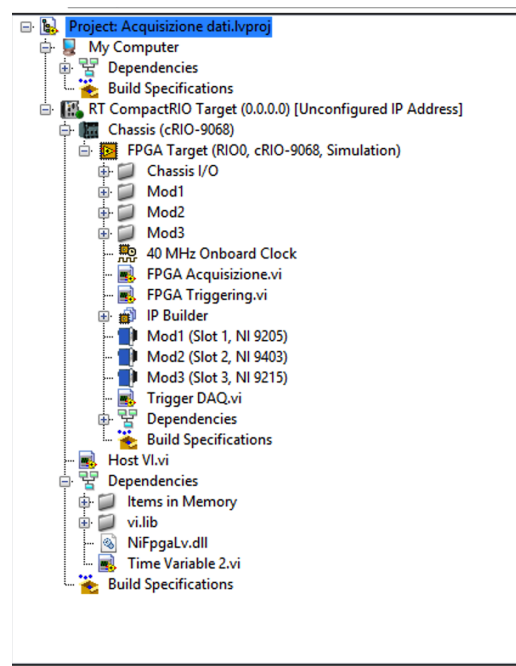


Figura 5.7: Schermata del progetto generale con FPGA. Evidenziati i tre *C-Modules* utilizzati per simulare quelli presenti nel cRIO del CICLoPE.

Appendice B


```

1
2
3
4
5
6
7
8
9
10

```

```

% Function to remove large oscillations from the mks files
function [MeanQuantity, DataMinusOffset, QuantityTimeHistory]=remove_oscillations(raw_data, offset)
    DataMinusOffset=raw_data-offset;
    Limitd=max([prctile(DataMinusOffset,25)-1.5*iqr(DataMinusOffset) min(DataMinusOffset)]);
    Limitu=min([prctile(DataMinusOffset,75)+1.5*iqr(DataMinusOffset) max(DataMinusOffset)]);
    QuantityTimeHistory=DataMinusOffset>=Limitd&&DataMinusOffset<=Limitu);
    MeanQuantity=mean(QuantityTimeHistory);
end

```

Figura 5.8: Script ideato da L. Mascotelli per rimuovere le ampie oscillazioni ricevute dall'MKS.

```

1  function [Correct_pressure,MeanQuantity,port_num]=port_meas_corr(port,data,offset)
2
3  %Fornisce il valore di media delle pressioni e pressioni per un trasduttore. Corrette dalle oscillazioni
4  %dovute a offset dell'mks e a possibili interferenze.
5
6
7  port_measure=data(:,port);
8  port_num=port;
9
10 DataMinus0ffset=data(:,port)-offset;
11 Limitd=max([prctile(DataMinus0ffset,25)-1.5*iqr(DataMinus0ffset) min(DataMinus0ffset)]);
12 Limitu=min([prctile(DataMinus0ffset,75)+1.5*iqr(DataMinus0ffset) max(DataMinus0ffset)]);
13 Correct_pressure=DataMinus0ffset(DataMinus0ffset>=Limitd&&DataMinus0ffset<=Limitu);
14 MeanQuantity=mean(Correct_pressure);
15
16 end

```

Figura 5.9: Script adattato dal precedente. In questo è possibile anche selezionare la porta della *Scanivalve* di interesse.

```
1 1
2 2
3 3
4 4
5 5
6 6
7 7
8 8
9 9
10 10
11 11
12 12
13 13
14 14
15 15
16 16
```

```
%funzione che restituisce la misura della velocità in [m/s] alla fine del condotto
%misurata tramite pitot. Penultima porta della scanivalve misura pressione
%statica, ultima porta misura la pressione totale (nell'ipotesi in cui
%queste siano due prese siano utilizzate a questo fine)
function [mean_vel,ps_mean,pt_mean]=mean_velocity(data,offset,rho)
n=size(data);
ps_mean=port_meas_corrected(n(2)-1,data,offset);
pt_mean=port_meas_corrected(n(2),data,offset);
mean_vel=sqrt((2*abs(pt_mean-ps_mean))/rho);
end
```

Figura 5.10: Script per il calcolo della velocità a valle del condotto. In questo caso, le ultime due porte sono quelle dedicate all'acquisizione di pressione statica e pressione totale.