



**Università degli Studi di Padova**

---

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Corso di Laurea in Bioingegneria

TESI DI LAUREA MAGISTRALE

**Messa a Punto di una Valvola Automatizzata di  
Resistenza Periferica in un Duplicatore di Impulsi  
Cardiovascolare**

Laureanda

**Eleonora Manzoni**

Matricola 1179548

Relatore

**Prof.ssa Francesca Maria Susin**

Dipartimento di Ingegneria Civile, Edile ed Ambientale

Correlatori

**Dott. Mirco Rampazzo**

**Dott. Luigi Di Micco**

---

Anno Accademico 2018-2019

Padova, 16 Dicembre 2019



*A Ser Birra, mio mentore e fonte d'ispirazione*



# Indice

<b>Sommario</b>	<b>vii</b>
<b>1 Introduzione</b>	<b>1</b>
1.1 Laboratorio di Fluidodinamica Cardiovascolare <i>HeR Lab</i> . . .	1
1.2 Fluidodinamica della circolazione sistemica . . . . .	2
1.3 <i>Pulse Duplicator</i> . . . . .	6
1.3.1 Componenti <i>hardware</i> . . . . .	7
1.3.2 Componenti <i>software</i> . . . . .	17
<b>2 Resistenza periferica automatizzata</b>	<b>21</b>
2.1 Componenti per l'automazione . . . . .	21
2.1.1 Arduino . . . . .	22
2.1.2 H-bridge . . . . .	24
2.1.3 <i>Software</i> NI LabVIEW Interface per il Toolkit Arduino	28
2.1.4 Motoriduttore in corrente continua . . . . .	29
2.1.5 Valvola manuale . . . . .	29
2.1.6 Accoppiamenti metallici . . . . .	31
2.1.7 Sistema di condizionamento . . . . .	33
2.2 Assemblaggio . . . . .	36
<b>3 Algoritmo <i>Extremum Seeking Control</i></b>	<b>39</b>
3.1 Inquadramento storico . . . . .	39
3.2 Introduzione all'algoritmo . . . . .	39
3.3 Algoritmo <i>ESC</i> per una mappa statica . . . . .	42
<b>4 Algoritmo <i>ESC Sliding Mode</i></b>	<b>45</b>
<b>5 Implementazione dell'algoritmo</b>	<b>49</b>
5.1 <i>Computer-Aided Control Systems Design</i> - CACSD . . . . .	49
5.2 Caratterizzazione preliminare . . . . .	50
5.3 Applicazione della tecnica in simulazione . . . . .	61
5.4 Implementazione della tecnica . . . . .	63
<b>6 Conclusioni e sviluppi futuri</b>	<b>75</b>

<b>A</b>	<b>Caratterizzazione degli esperimenti</b>	<b>77</b>
<b>B</b>	<b>Simulazione del controllo</b>	<b>83</b>
	<b>Bibliografia</b>	<b>91</b>

# Sommario

In questa tesi magistrale si affronta il problema dell'implementazione di un algoritmo di controllo della pressione aortica nel duplicatore di impulsi cardiovascolare (*Pulse Duplicator*) del *Healing Research Lab* di Padova attraverso l'impiego di una valvola motorizzata di resistenza periferica.

Il *Pulse Duplicator* è un circuito meccanico-idraulico chiuso per lo studio della fluidodinamica nella circolazione sistemica umana a seguito dell'impianto di dispositivi protesici. Il duplicatore di impulsi è un sistema modulare che può essere adattato a diverse tipologie di esperimenti.

Al fine di poter assegnare desiderate specifiche per l'algoritmo di controllo della pressione aortica e garantirne adeguate prestazioni nell'ampio intervallo di condizioni operative in cui funziona il *Pulse Duplicator* (ad esempio, differenti valori di frequenze cardiache, diversi valori di cardiac output, etc.) nonché di assicurare una flessibilità di impiego del controllore al variare di molteplici configurazioni di esperimenti (ad esempio, differenti dispositivi protesici, diversa viscosità del fluido impiegato, etc.), si è optato per l'adozione di un algoritmo di natura model-free (non basato su modello). In particolare, in funzione delle specifiche desiderate, si è impostato il problema di controllo come un problema di ottimizzazione e lo si è affrontato utilizzando una variante di algoritmo di controllo di tipo model-free quale l'*Extremum Seeking* (ESC) nota come *ESC Sliding Mode*.

Per quanto concerne la progettazione, la taratura e la verifica delle prestazioni dell'algoritmo *ESC Sliding Mode*, data la delicatezza e la difficoltà di operare fin da subito sul *Pulse Duplicator* (in termini di allestimento degli esperimenti, di esecuzione delle prove, etc.), si è preferito preliminarmente giovare delle opportunità offerte da un ambiente di simulazione Matlab/Simulink relativo al *Pulse Duplicator* in assegnate condizioni qualitative nominali. Infatti, è una pratica comune, nella progettazione di algoritmi controllo fare uso di strumenti di *Computer-Aided Control Systems Design* (CACSD). Questi strumenti consentono di simulare le principali caratteristiche del sistema per una prima valutazione della strategia di controllo. Una volta verificata in simulazione l'efficacia dell'algoritmo *ESC Sliding Mode* esso è stato successivamente implementato e testato nel *Pulse Duplicator* reale attraverso l'utilizzo di hardware (Arduino, Ponte H e motoriduttore)

e software dedicato (LabVIEW e NI LabVIEW Interface per il Toolkit Arduino).

La tesi è composta da 6 capitoli, rappresentativi delle quattro fasi principali che hanno portato all'implementazione fisica del controllo: l'introduzione all'ambiente sperimentale e all'apparato stesso su cui si è operato, la sintesi e la messa a punto di una nuova valvola periferica controllata via software, la scelta dell'algoritmo di controllo più adatto al sistema, e la simulazione e l'implementazione di quest'ultimo sulla nuova valvola.

In particolare, il capitolo 1 introduce il Laboratorio *Healing Research Lab*, fornisce i rudimenti di fluidodinamica della circolazione sistemica necessari alla corretta interpretazione dello scritto, e descrive il banco di prova *Pulse Duplicator* sia dal punto di vista dell'hardware, sia dal punto di vista del software che lo governa.

Il capitolo 2 spiega nel dettaglio i componenti hardware e software necessari all'automatizzazione della valvola di resistenza periferica. Si mostrano, in particolare, i sistemi di filtraggio e condizionamento del segnale utile, essenziali per il corretto funzionamento dell'algoritmo di controllo.

Nei capitoli 3 e 4 si introduce l'algoritmo di controllo utilizzato, giustificandone la scelta e il principio di funzionamento.

Nel capitolo 5 si caratterizza il sistema per l'applicazione della tecnica di controllo scelta, si illustrano modalità e risultati di simulazioni numeriche, e si implementa il controllo sul banco di prova.

Nel capitolo 6 si traggono le conclusioni e si sottolineano i possibili sviluppi futuri.

In ultimo, si inseriscono le appendici A e B, in cui si mostrano nel dettaglio i risultati della caratterizzazione del sistema e dell'applicazione dell'algoritmo in simulazione, rispettivamente.



# Capitolo 1

## Introduzione

### 1.1 Laboratorio di Fluidodinamica Cardiovascolare *HeR Lab*

La fluidodinamica cardiovascolare è la scienza che sfrutta le conoscenze della meccanica dei fluidi per affrontare temi riguardanti il sistema cardiovascolare umano.

Nell'ambito della ricerca scientifica del sistema cardiocircolatorio, si inserisce il laboratorio per studi di fluidodinamica cardiovascolare *Healing Research Lab*, afferente al dipartimento ICEA dell'Università di Padova.

L'obiettivo principale di questo laboratorio è di unire conoscenze mediche, cliniche e ingegneristiche al fine di esplorare le caratteristiche tecniche di dispositivi protesici cardiovascolari e di valutarne gli effetti sull'organismo.

A questo scopo, è stato sviluppato il *Pulse Duplicator*, un circuito meccanico-idraulico chiuso per lo studio della fluidodinamica nella circolazione sistemica umana a seguito dell'impianto di dispositivi protesici. Questo sistema, essendo stato costruito secondo le esigenze del laboratorio stesso, sempre attenendosi alle prescrizioni delle norme, risulta essere uno strumento modulare, ovvero in grado di adattarsi largamente ai componenti da testare o agli esperimenti da eseguire.

Il dispositivo, per sua stessa natura, è in continua evoluzione, infatti ogni suo elemento può essere variato, sostituito, o modificato sia dal punto di vista geometrico, sia funzionale.

Tramite le simulazioni al *Pulse Duplicator* si vogliono individuare soluzioni tecnologiche per un miglior trattamento di patologie cardiache, comprendendo quali siano gli effetti dell'inserimento di elementi protesici all'interno della circolazione sistemica, attraverso la valutazione delle relative performance emodinamiche.

La realizzazione di un circuito idraulico chiuso che simuli la circolazione sistemica è essenziale per monitorare come variazioni di parametri emodina-

mici, ad esempio gittata cardiaca o pressione sanguigna, possano modificare le prestazioni idrodinamiche delle protesi.

## 1.2 Fluidodinamica della circolazione sistemica

L'apparato cardiocircolatorio è l'insieme degli organi che permettono la circolazione del sangue nell'organismo, al fine di rifornire le cellule di nutrienti e ossigeno, e di consentire l'eliminazione dell'anidride carbonica e di altri prodotti di scarto.

Esso ha, inoltre, il compito di veicolare importanti componenti del sistema immunitario (i globuli bianchi) e altre molecole che svolgono diverse funzioni nell'organismo (come gli ormoni) i quali, proprio mediante la fitta rete che compone questo apparato, riescono a spostarsi da un organo all'altro e da un tessuto all'altro a seconda delle necessità dell'organismo [4].

Nell'apparato circolatorio scorre il sangue, un liquido opaco di color rosso. Esso è costituito da una parte liquida, il plasma (circa il 55% del volume ematico totale), e da una parte corpuscolata che, a sua volta, consta di globuli rossi (eritrociti), globuli bianchi (leucociti), e piastrine. Il color rosso del sangue è dovuto alla presenza di un pigmento, l'emoglobina, all'interno dei globuli rossi.

La densità sanguigna è pari a 1050–1060 kg/m<sup>3</sup>, (circa 1.05 volte quella dell'acqua), il suo peso specifico è di circa 10198 N/m<sup>3</sup>, mentre la sua viscosità dinamica è 4·10<sup>-3</sup> kg/m·s, 4 volte quella dell'acqua (10<sup>-3</sup> kg/m·s).

L'apparato cardiocircolatorio è un sistema chiuso di vasi in cui il sangue circola sotto la spinta del cuore. Esso è formato da due componenti principali, riportati in figura 1.1: il circolo sistemico, che distribuisce l'ossigeno, i nutrienti e le sostanze utili a tutti i tessuti del corpo, e il circolo polmonare, che ha lo scopo di ossigenare il sangue ed eliminare le sostanze di scarto. Ognuna di queste componenti è formata da arterie, capillari e vene.

Il cuore, una pompa muscolare, è diviso in quattro camere (figura 1.2): due atri e due ventricoli, dove la coppia di atrio e ventricolo sulla sinistra del corpo corrisponde al cuore sinistro, analogamente la coppia a destra costituisce il cuore destro; il cuore destro e quello sinistro sono separati da un setto che fa sì che essi non entrino mai in comunicazione. Il cuore destro e quello sinistro lavorano in sincronia, perciò ogni contrazione cardiaca spinge simultaneamente il sangue sia nella circolazione polmonare sia nella sistemica.

Per la natura stessa delle due circolazioni, nel cuore destro scorre solo sangue privo di ossigeno, in quanto esso riceve il sangue deossigenato dalla circolazione sistemica e lo spinge verso il circuito polmonare, mentre nel cuore sinistro fluisce solo sangue ossigenato, in quanto esso riceve sangue ricco di ossigeno dai polmoni e lo spinge nel circuito sistemico.

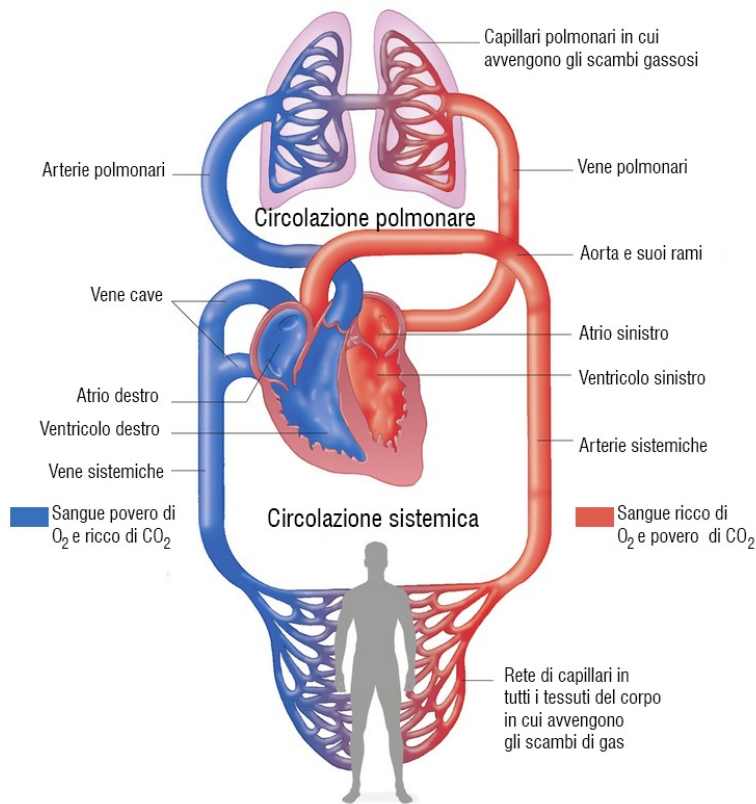


Figura 1.1: Schema dell'apparato cardiocircolatorio

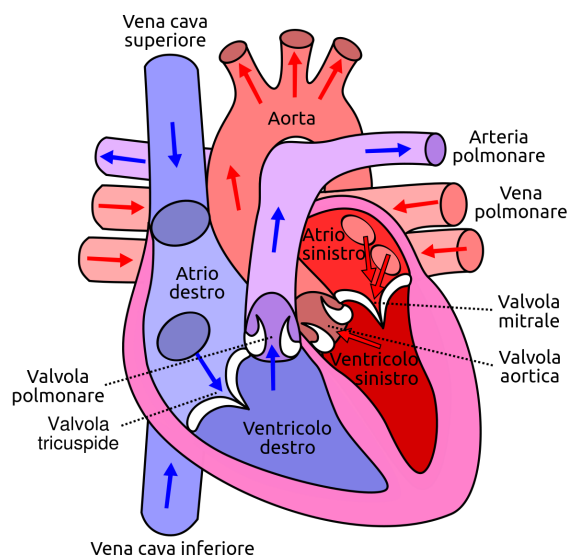


Figura 1.2: Schema del cuore umano

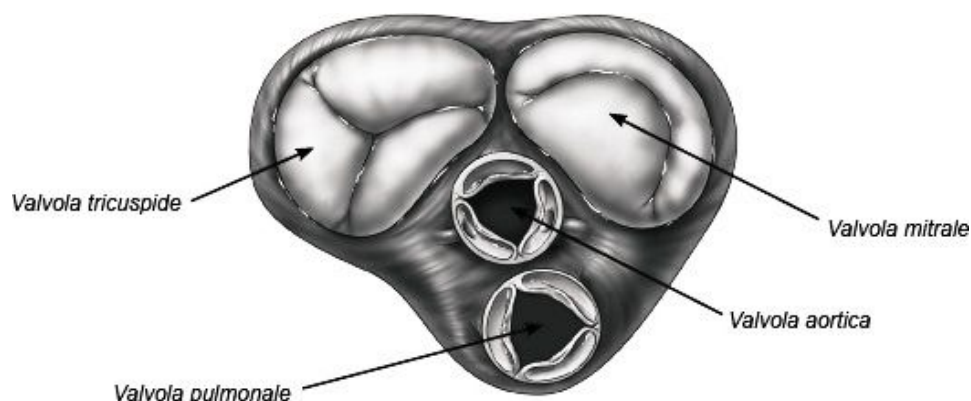


Figura 1.3: Schema delle valvole cardiache

Uno dei vantaggi di una circolazione separata è che la distribuzione di sangue nei tessuti risulta più rapida ed efficiente.

Le arterie, vasi dalla parete spessa formata da tessuto elastico e fibre muscolari, ricevono il sangue dal cuore. Esse si ramificano via via in vasi sanguigni più sottili, fino ad arrivare a dare vita alle arteriole attraverso cui il sangue viene convogliato all'interno di capillari, tubicini dalla parete sottile e permeabile da parte di nutrienti, gas e prodotti di scarto. I capillari, infine, convogliano il sangue verso piccoli vasi, le venule, che convergono a formare le vene che riportano il sangue verso il cuore.

Nel caso della circolazione sistemica, il sangue proveniente dai polmoni, ricco di ossigeno e di nutrienti, viene pompato dal ventricolo sinistro all'interno dell'aorta, che lo distribuisce al resto del corpo. Una volta che il sangue si è arricchito di anidride carbonica e prodotti di scarto viene riversato dalla vena cava all'atrio destro.

Per facilitare il movimento del sangue attraverso il cuore, in esso sono presenti quattro valvole, mostrate in figura 1.3: due valvole atrio-ventricolari, poste fra atri e ventricoli, che impediscono il reflusso del sangue nell'atrio quando il ventricolo si contrae, e due valvole semilunari (la valvola polmonare e quella aortica), posizionate fra i ventricoli e le arterie maggiori, che prevengono il reflusso di sangue nei ventricoli quando questi si rilassano.

Sinteticamente, la circolazione polmonare ha inizio dal ventricolo destro, tramite le arterie polmonari, e termina nell'atrio sinistro, tramite le vene polmonari. Nella circolazione sistemica il sangue parte dal ventricolo sinistro, passa per la valvola aortica, sfocia nella radice aortica, poi nell'arco aortico e via via si ramifica; il suo circuito si conclude nell'atrio destro, tramite le vene cave. Il muscolo cardiaco si contrae ritmicamente in funzione della quantità di ossigeno richiesta dai tessuti. Le fasi di contrazione e rilassamento interessano entrambi i lati del cuore e si alternano in modo che

il flusso del sangue si mantenga sempre unidirezionale. L'intera sequenza costituisce il ciclo cardiaco, che ha una durata media di circa 0.9 secondi e comprende una fase di sistole, cioè di contrazione, e una di diastole, cioè di rilassamento.

Durante la diastole il muscolo cardiaco è rilassato, le valvole semilunari sono chiuse, mentre quelle atrio-ventricolari sono aperte e il sangue entra spontaneamente negli atri e nei ventricoli. Nel corso della sistole atriale gli atri si contraggono e, svuotandosi, spingono con forza il sangue nei ventricoli che sono ancora in diastole.

Durante la sistole ventricolare, i ventricoli si contraggono e la pressione al loro interno diventa più alta di quella atriale, così che le valvole atrio-ventricolari si chiudano. La pressione nei ventricoli cresce rapidamente fino a provocare l'apertura delle valvole semilunari, permettendo al sangue di fluire nell'aorta e nelle arterie polmonari, rispettivamente, nel cuore sinistro e in quello destro. In questa fase gli atri sono in diastole, quindi essendo rilassati, si riempiono di sangue.

Al termine della sistole ventricolare inizia un nuovo ciclo cardiaco: i ventricoli entrano in diastole e le valvole semilunari si chiudono a causa dell'alta pressione presente nell'aorta e nelle arterie polmonari. Dato che anche gli atri sono ancora in diastole, la pressione interna al cuore è modesta, e per questo il sangue proveniente dalla circolazione sistemica e da quella polmonare entra liberamente in esso.

L'attività del cuore sinistro, durante due cicli cardiaci successivi, è riassunta nella figura 1.4: si evidenziano le curve pressorie di atrio, ventricolo e della radice aortica, unitamente alla variazione volumetrica all'interno del ventricolo sinistro. Per completezza si riportano l'elettrocardiogramma (riproduzione grafica dell'attività elettrica del cuore durante il suo funzionamento, registrata a livello della superficie corporea) e il fonocardiogramma (registrazione acustica dei toni e dei rumori cardiaci).

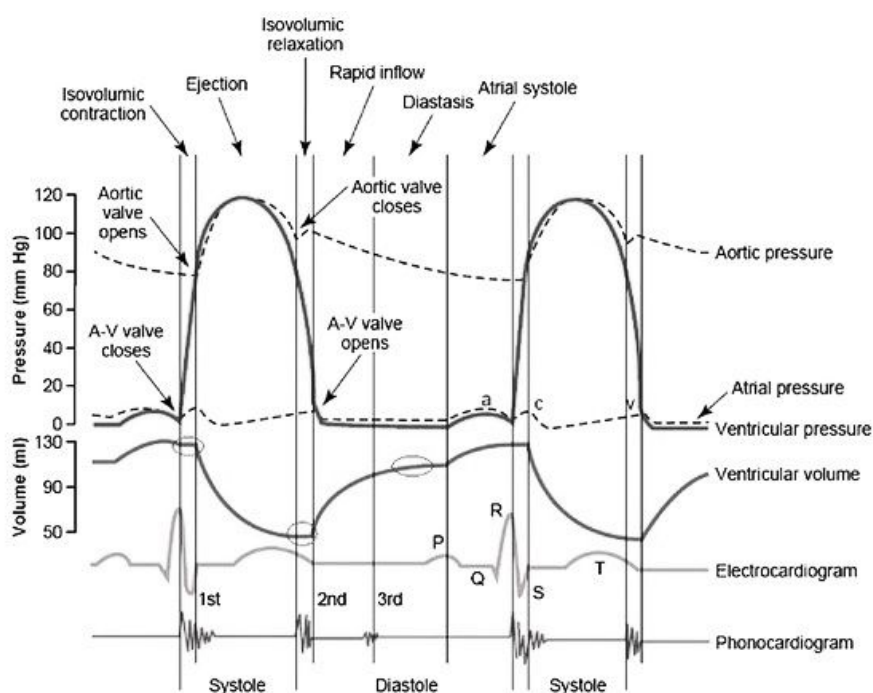
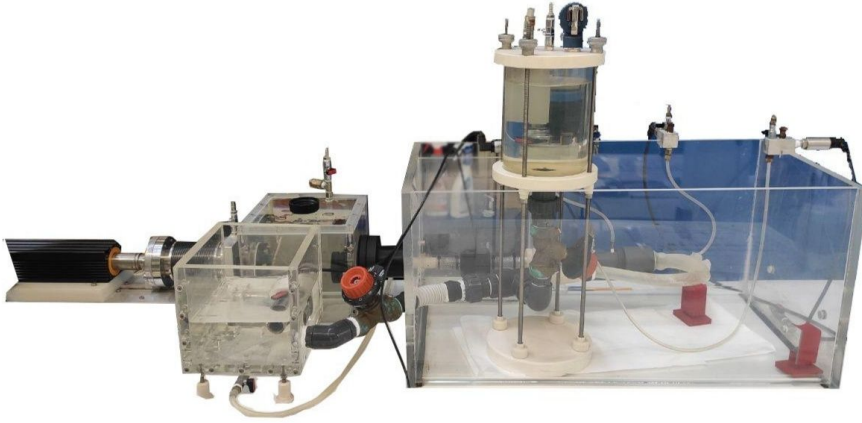
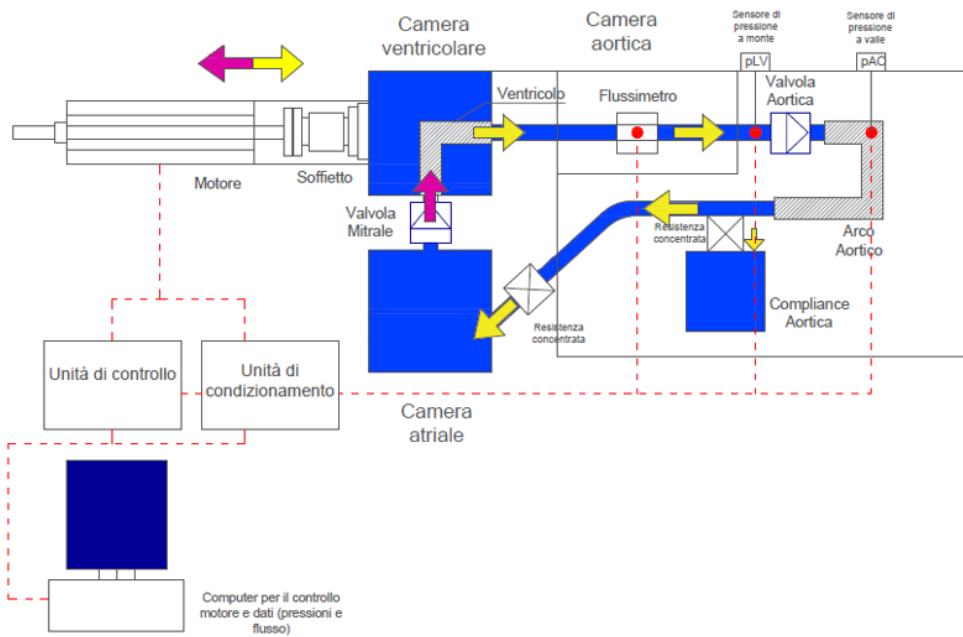


Figura 1.4: Attività del cuore sinistro

### 1.3 *Pulse Duplicator*

La funzione principale di un *Pulse Duplicator* (letteralmente “duplicatore di impulsi”) è quella di riprodurre la circolazione sistemica e il comportamento dinamico del cuore. Il dispositivo è costituito da un circuito meccanico-idraulico a flusso pulsatile, con condotti in pressione in regime di moto vario, che ha lo scopo primario di replicare le condizioni fisiologiche di pressione e portata del sistema circolatorio umano.

Esso viene generalmente utilizzato per sviluppare e testare dispositivi per assistenza ventricolare, valvole cardiache protesiche o dispositivi di altra natura, prima di studi in vivo su animali. Può essere, dunque, un utile strumento per valutare e validare le prestazioni di componenti protesici in condizioni sia fisiologiche, sia patologiche relative a diversi gradi di severità. Per semplicità esplicativa, si riportano separatamente le descrizioni dettagliate dei componenti hardware e software che permettono il funzionamento del *Pulse Duplicator*.

1.3.1 Componenti *hardware*Figura 1.5: *Pulse Duplicator* presente nel laboratorio *HeR*Figura 1.6: Schema a blocchi del *Pulse Duplicator*

Nel *Pulse Duplicator*, il cuore è simulato da un motore lineare e da un soffietto, mentre l'intero percorso del sangue è replicato da condotti rigidi e non, raccordi, elementi complianti e resistenze localizzate.

Il sistema presente nel laboratorio *HeR*, visibile in figura 1.5, simula uni-

camente la circolazione sistemica, ovvero il cuore sinistro, gli atri, l'aorta e tutti i vasi periferici che raggiungono i distretti corporei.

Lo schema funzionale del *Pulse Duplicator*, è riportato in figura 1.6, dove le frecce indicano la direzione del fluido. Le condizioni di lavoro sono prese in carico dal motore elettromagnetico lineare, gestito via software dall'operatore, mentre le unità di controllo e di condizionamento permettono la manipolazione dei segnali acquisiti dai sensori presenti nel sistema.

La curva di posizione impostata dall'utente viene trasferita al fluido tramite il soffietto, il quale agisce da pompa muovendosi linearmente avanzando ed arretrando. Il getto in uscita al gruppo motore-soffietto si dirige nella camera ventricolare e, passando attraverso un flussimetro, arriva alla valvola aortica. A monte e a valle della valvola aortica si trovano i sensori di pressione. Il flusso prosegue nell'aorta, riprodotta fedelmente in silicone, e si dirige nella camera compliante, la quale ha lo scopo di simulare l'elasticità dei vasi di maggior calibro. Successivamente, il flusso incontra una valvola che rappresenta in modo localizzato le resistenze offerte dai vasi periferici. Il flusso si dirige verso la camera atriale, un serbatoio a superficie libera che rappresenta entrambi gli atri, e si reca nuovamente nella camera ventricolare attraverso la valvola mitralica. In questo modo si conclude il ciclo.

Nel seguito si descrivono nel dettaglio le componenti *hardware* del *Pulse Duplicator* rispettando l'ordine con cui il fluido le incontra nel ciclo [1], [2].

**Motore lineare elettromagnetico e soffietto** Il fluido è messo in movimento attraverso il complesso motore-soffietto (figura 1.7), che si comporta come una pompa a due direzioni, riproducendo il comportamento pulsatile del cuore sinistro.

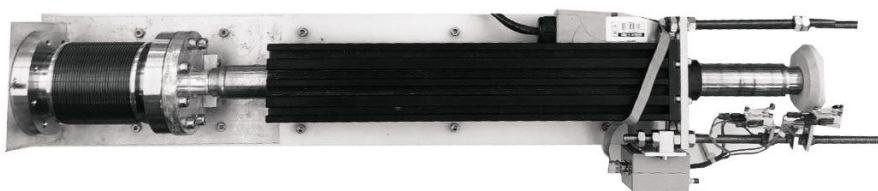


Figura 1.7: Motore lineare e soffietto

Il movimento avviene secondo una legge imposta dal calcolatore al motore (*Parker PRA3810S*), la quale riproduce le reali variazioni volumetriche del flusso sanguigno durante un ciclo cardiaco completo (sistole e diastole). Lo spostamento in avanzamento rappresenta fisiologicamente il periodo di eiezione ventricolare, quindi la fase di sistole, mentre lo spostamento retrogrado è rappresentativo della fase di dia-



stole.

Il complesso è connesso all'unità di controllo tramite due cavi, uno per l'alimentazione, e uno per il trasferimento dei dati. L'intera struttura è ancorata ad una base di Teflon rinforzata con barre di alluminio per evitare movimenti del pistone fuori asse. L'attuatore può compiere un'escursione massimale di 140mm, consentendo di spostare volumi fluidi ampiamente superiori alle richieste sperimentali.

Le principali caratteristiche tecniche del motore sono: forza di picco 1860N, raggiunto per la durata di 1s; corrente di picco 14.1A, raggiunta per la durata di 1s; forza erogata in modalità continua 255.0N; corrente di tenuta, per tenere fermo l'attuatore, 1.94A; accelerazione di picco  $3.91m/s^2$ ; velocità massima 2.6m/s; Tali dati sono relativi a condizioni operative con una temperatura ambiente di 25°C, senza carico utile.



Figura 1.8: Soffietto COMVAT - Bellows

Il soffietto (figura 1.8), prodotto dalla *COMVAT - Bellows*, ha un diametro interno di 65mm e ha una escursione utile di 50mm, che corrisponde al movimento di un volume di fluido di 200ml per battito.

L'accoppiamento motore-soffietto è realizzato tramite un giunto metallico flangiato circolare, con diametro pari a quello del soffietto.

L'ultimo elemento costitutivo di questo complesso è rappresentato da tre sensori di posizione a relè ancorati alla struttura metallica di supporto. Si tratta di tre *microswitches Cherry D45X*, con il compito di fornire segnali all'unità di controllo qualora il terminale dell'attuatore li attivi esercitando una pressione sul pulsante. I sensori rappresentano gli estremi del campo di movimento del motore elettrico, costituendo così delle protezioni. Se attivati, infatti, comunicano all'unità di controllo di interrompere l'alimentazione del driver, andando così a bloccare istantaneamente il movimento del motore per evitare possibili danni alla struttura stessa.

**Unità di condizionamento** Il dispositivo *LS0609*, realizzato dalla *Lab-*

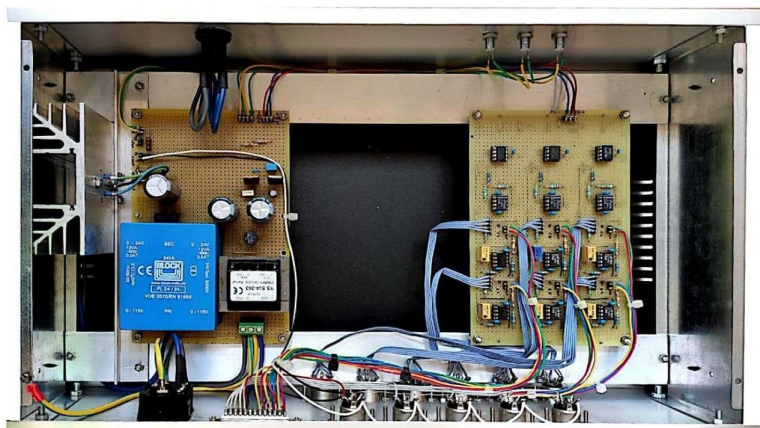


Figura 1.9: Unità di condizionamento

*Trek S.r.l.*, acquisisce, amplifica, filtra e condiziona i segnali provenienti dai sensori dislocati lungo il circuito idraulico, così da renderli adatti alla successiva elaborazione, effettuata al computer. Per il tipo di sensori ora in utilizzo, l'unità di condizionamento è inutilizzata, ma sempre disponibile secondo necessità per il condizionamento di nuovi segnali.

Il dispositivo, mostrato in figura 1.9, è costituito da due componenti elettronici:

- La scheda di amplificazione contiene una serie di generatori di tensione per l'alimentazione dei sensori di pressione, e dei canali di amplificazione necessari per il condizionamento dei segnali provenienti dai connettori (In0, ..., In5). Un collegamento diretto senza preamplificazione è previsto invece per i sensori *general purpose*, collegabili ai connettori (In6, ..., In10); per questi ingressi è previsto un solo filtraggio per alimentare le componenti spurie dei segnali analogici acquisiti.
- Il circuito di alimentazione dell'unità di condizionamento è composto da due sezioni, la prima fornisce i +24V necessari al funzionamento dei *sensori general purpose*, la seconda fornisce invece la tensione di  $\pm 15V$  necessaria all'alimentazione della scheda di amplificazione.

**Unità di controllo** L'unità di controllo *LT0510* sviluppata da *LabTrek S.r.l.*, permette di gestire il motore, di ricevere i segnali dall'unità di condizionamento e di comunicare bidirezionalmente con il PC. Il dispositivo, mostrato in figura 1.10 comprende tre componenti elettronici principali:

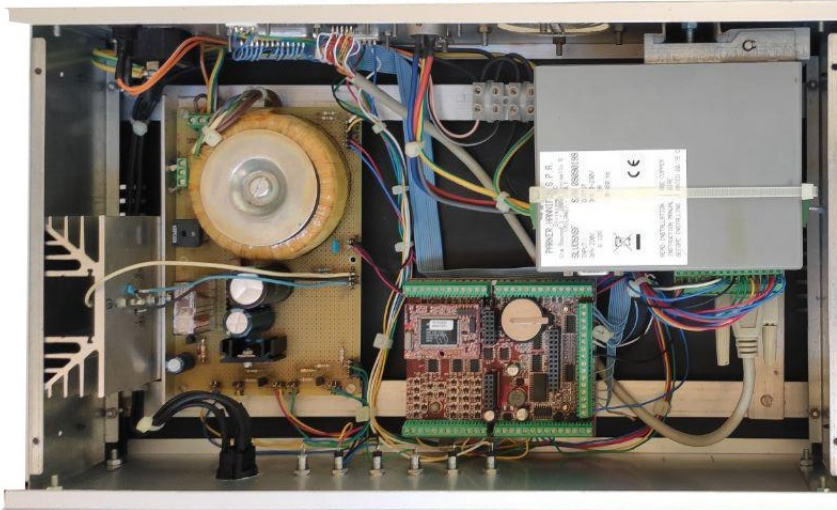


Figura 1.10: Unità di controllo

- Il drive del motore elettrico è un convertitore analogico-digitale e permette il movimento del motore lineare tramite la variazione delle tensioni a esso inviate. Il drive riceve una tensione variabile rappresentativa della legge di spostamento da imporre al pistone, e la converte in una sequenza di impulsi finalizzati alla movimentazione dell'attuatore.
- Il computer su scheda, *Rabbit BL2120*, ha il ruolo di mediatore tra il computer e il drive del motore. Comunica con il pc, attraverso la seriale, tramite un programma *LabVIEW*, e genera segnali in tensione nell'intervallo 0-10V da inviare al drive. La programmazione della scheda Rabbit avviene tramite il caricamento via seriale di programmi sviluppati in *Dynamic C*.
- Il circuito di alimentazione preleva tensione dalla rete elettrica per poter alimentare il drive del motore e il computer su scheda.

**Camera ventricolare** Essa è direttamente collegata al soffiato e alla valvola mitralica, che ne rappresentano gli ingressi, e a un'uscita, costituita dalla camera aortica. Le variazioni di volume, sono trasmesse, in fase sistolica, alla sezione aortica, oggetto principale dello studio. La camera ventricolare è realizzata con pannelli di Plexiglass trasparenti di spessore 15mm, che garantisce una buona resistenza alle sollecitazioni meccaniche indotte, ed è di forma cubica, con lato di 230mm (figura 1.11).

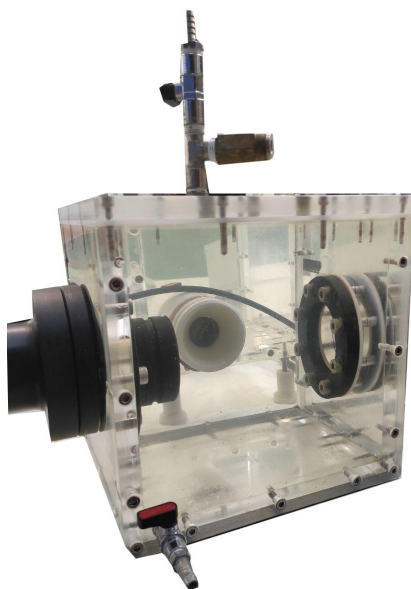


Figura 1.11: Camera ventricolare

**Camera aortica** Essa comprende la valvola aortica, che può essere facilmente sostituita per permettere l'utilizzo di valvole di diverse tecnologie e dimensioni, e la riproduzione fedele per elasticità e parametri geometrici di un arco aortico fisiologico in silicone, mostrato in figura 1.13. A monte della valvola aortica si trovano i sensori di portata e di pressione (*PCB Piezotronics*) rappresentativi dello stato del ventricolo sinistro, mentre a valle della valvola si registra la pressione aortica. La struttura esterna della camera aortica, mostrata in figura 1.12, è assimilabile ad una vasca, realizzata da componenti in Plexiglass. Essa è un parallelepipedo rettangolo con le facce maggiori di spessore pari a 10mm e con dimensioni dei lati  $1.1 \times 0.7m$ . Le lastre corrispondenti alle facce minori, sono anch'esse in Plexiglass e hanno uno spessore di 15mm e dimensione 360x400mm. Queste sono fissate alla vasca per incollaggio tramite una opportuna colla bi-componente.

**Camera compliant** Questa camera, mostrata in figura 1.14, è costituita da un cilindro chiuso, contenente parzialmente il fluido della simulazione e parzialmente aria. Il compito principale di questo componente è simulare l'elasticità dei vasi di maggior calibro. Le arterie di diametro maggiore, infatti, subiscono delle deformazioni radiali non trascurabili per effetto della pressione ventricolare. Questo comportamento causa, nei vasi periferici, un flusso pressoché costante, nettamente differente da quello generato dal moto pulsatile del cuore. La compliance, infat-

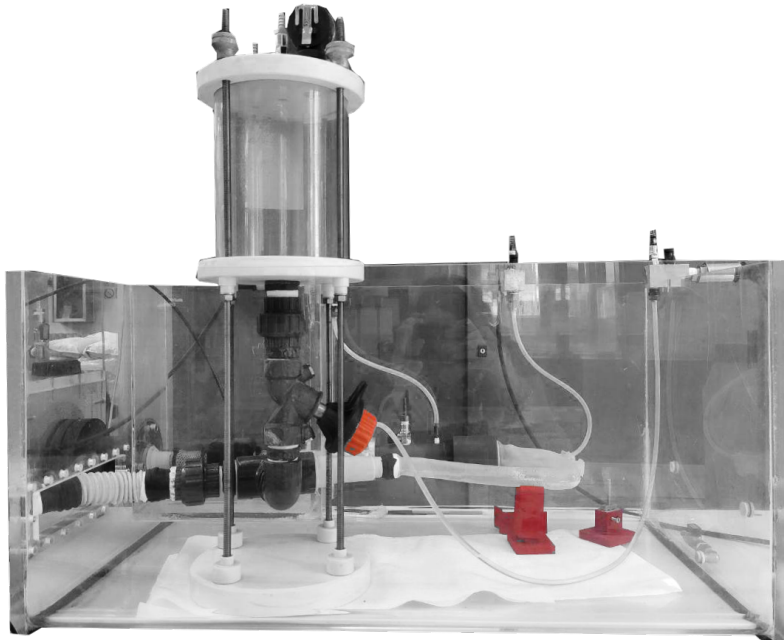


Figura 1.12: Camera aortica



Figura 1.13: Arco aortico

ti, è in grado di smorzare l'onda di pressione prodotta dal moto del pistone generando, a valle, un'onda più regolare dalle limitate caratteristiche di impulsività. Durante la fase di sistole, l'ingresso di fluido nella camera compliante determina un aumento del livello di liquido al suo interno, e una conseguente compressione del volume d'aria. In fase di diastole, il flusso entrante la camera si interrompe e si genera esclusivamente un moto in uscita provocato dalla spinta del gas che causa un abbassamento del livello liquido. La cassa d'aria sfrutta, quindi, la comprimibilità del gas al suo interno per assorbire l'onda di pressione e il volume fluido durante la fase di eiezione, smorzandone il picco pressorio e restituendo una portata con andamento più regolare durante la fase di diastole.

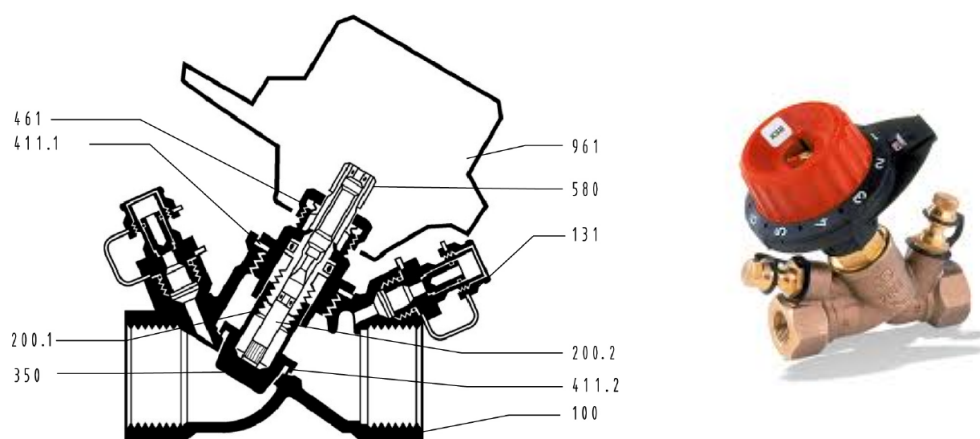


Figura 1.14: Camera compliante

Il serbatoio di compliance è realizzato da un cilindro in Plexiglass. La tenuta della superficie inferiore e di quella laterale è assicurata tramite incollaggio. La tenuta della parte superiore è garantita da un'apposita guarnizione in silicone dello spessore di 3mm e viti passanti. Sul piattello superiore sono praticati dei fori filettati necessari al posizionamento di un rubinetto di sfato, di una valvola di sicurezza e di un regolatore di pressione, collegato ad un manometro, necessario per rifornire di aria la compliance, tramite l'utilizzo di un compressore.

**Resistenza localizzata** Per ottenere il trend pressorio fisiologico a valle dell'arco aortico, si utilizza un rubinetto il cui orifizio può essere

variato manualmente. Questa valvola rappresenta, sotto forma di restringimento localizzato, la resistenza offerta dai vasi periferici, dovuta all'elevato numero di diramazioni che il sangue incontra nel suo percorso (figura 1.15). La valvola di regolazione a tenuta morbida utilizzata è la *BOA-Control SAR* della ditta *KSB* con diametro nominale di 25mm. La valvola opera a una pressione massima di 16bar e in un range di temperatura tra  $-25^{\circ}\text{C}$  e  $+150^{\circ}\text{C}$ . Il grado di apertura della valvola può essere scelto manualmente tramite un indicatore con 40 posizioni di controllo e rotazioni di  $1/10$  con possibilità di aumentare o diminuire il valore corrente. Il corpo in bronzo è idraulicamente ottimizzato per portate elevate.



Part No.	Description	Material	Note
100	Body	Bronze	-
131	Pressure measurement connection branch	Brass	-
200.1	Valve stem	Brass	-
200.2	Memo stem	Cu Zn 40 Pb3	Dezincification-free
350	Valve disc	Cu Zn 36 Pb2 AS	-
411.1	Joint ring	EPDM	-
411.2	Joint ring	EPDM	Metal gasket, for DN $\frac{3}{8}$ "
461	Gland packing	Brass	-
580	Cap	Leaded red brass	-
961	Handwheel	Polyamide 6-6 with 30 % glass fibre	-

Figura 1.15: Resistenza localizzata *BOA-Control SAR*

**Serbatoio atriale** Questa vasca a superficie libera, che rappresenta entrambi gli atri (figura 1.16), si collega nuovamente nella camera ventricolare attraverso la valvola mitralica, chiudendo così il ciclo fluido. Le dimensioni del serbatoio (5 lastre identiche di Plexiglass di dimensione 310x400mm di spessore 10mm accoppiate per incollaggio) sono tali da smorzare la pulsatilità del moto, in quanto l'ingresso o l'uscita di fluido in esso non modifica la quota della superficie libera. Nella fase di arretramento del pistone viene risucchiato del fluido: questo mec-

canismo permette che la pressione nel ventricolo sia inferiore rispetto alla pressione determinata dal livello del serbatoio atriale, innescando, così, la portata in accumulo al soffietto attraverso la valvola mitralica.

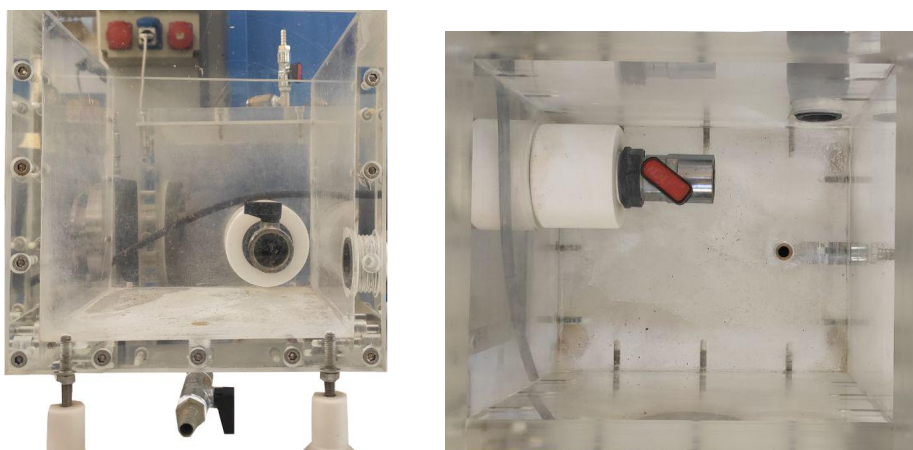


Figura 1.16: Camera atriale

**Sensori di pressione** I sensori di pressione sono di tipo piezoelettrico, e in particolare dei Serie 1502 prodotti da *Piezotronics*. Sono strumenti precisi e caratterizzati da elevata stabilità nelle misure, per questo adatti a situazioni di misura che richiedono marcata ripetitività.

I sensori *Piezotronics*, sottoposti ad una sollecitazione pressoria di tipo meccanico, rilevano tramite un apposito circuito interno, una tensione nell'intervallo 0-10Vdc (impedenza d'uscita  $<1000\Omega$ ) equivalente ad una capacità di fondo scala di 571.17mmHg. Dai datasheet si vede che la loro sensibilità è pari a 9.67mV/mmHg.

Gli *output* dei sensori di pressione vengono inviati al computer senza amplificazione, poiché essi risultano già preamplificati e compensati.

**Flussimetro** Tra la camera ventricolare e la valvola aortica è situata una vasca stagna. Essa ospita la sonda del flussimetro ultrasonico, uno strumento capace di misurare la portata di un fluido in movimento. Il dispositivo scelto è il *Transonic Systems Inc.* che si compone di due parti: la sonda *ME19PXN Inline*, inserita nell'apposita vasca stagna poiché esso non può entrare direttamente in contatto con liquidi, ed il modulo di controllo *Tubing Module TS410*. Il modulo fornisce una tensione di uscita di  $\pm 5V$ , con resistenza d'uscita da  $500\Omega$ , è anche predisposto per ospitare una seconda sonda di misura ed è stato calibrato per funzionare in modo ottimale nel range di temperature tra i  $20^{\circ}C$  e i  $37^{\circ}C$ .





Figura 1.17: Modulo di controllo e sonda del Transonic System Inc.

Il *pulse duplicator* in questione permette di cambiare agevolmente tipologia e dimensione di valvola aortica e mitrale, in questa tesi si sono utilizzate *On-X bileaflet mechanical heart valve* di diametro 23mm in posizione aortica, e la *Solin Allcarbon bileaflet mechanical heart valve* di diametro 27mm in posizione mitralica, la prima delle quali è mostrata in figura 1.18.



Figura 1.18: Valvola aortica protesica *On-X bileaflet mechanical heart valve*

La *On-X Aortic Valve* è una valvola cardiaca di nuova generazione costituita da un unico materiale e con design innovativo. Questa valvola cardiaca è l'unica valvola meccanica approvata FDA e CE che ha dimostrato di essere clinicamente affidabile anche con la somministrazione di un ridotto quantitativo di anticoagulanti. Le linee guida dell'*American Heart Association* asseriscono che un ridotto quantitativo di anticoagulanti è ragionevole per pazienti con valvola meccanica *On-X Aortic Valve*. In un trial clinico randomizzato, pazienti con la valvola aortica On-X a cui è stata somministrata una ridotta dose di anticoagulanti, hanno dimostrato di avere più della metà episodi emorragici in meno, senza rilevare un aumento di rischio di ictus. Le *On-X aortic and mitral prosthetic valves* sono state impiantate con successo in più di 260 000 pazienti nel mondo [5].

### 1.3.2 Componenti *software*

Le parti *software* necessarie per movimentare, controllare e rendere di fatto operativo il complesso sistema sono state implementate in LabVIEW [2], [1].

LabVIEW è un *software* per la progettazione di sistemi e lo sviluppo di applicazioni di test, misura e controllo con accesso rapido all'hardware e ai risultati. Esso offre un approccio di programmazione grafico che permette di visualizzare ogni aspetto dell'applicazione, inclusa la configurazione *hardware*, i dati di misura e il *debug*. La visualizzazione semplifica l'integrazione dell'*hardware* di misura di qualsiasi fornitore e permette la rappresentazione logica sui diagrammi, lo sviluppo di algoritmi di analisi dati e la progettazione di interfacce utente personalizzate.

Come in ogni programma LabVIEW è possibile individuare il pannello frontale e lo schema a blocchi. Nel primo sono collocati tutti gli elementi fisici di interesse per l'utente: pulsanti, barre di controllo, controlli numerici, indicatori e grafici. Il secondo, invece, rappresenta il codice di programmazione vero e proprio, in quanto contiene i terminali relativi ai componenti presenti nel pannello frontale, le funzioni usate e le strutture logiche, collegate tra loro per formare uno schema a blocchi. Il programma riceve i dati dalla scheda *Rabbit*, li elabora e permette all'operatore di organizzare, visualizzare e salvare i dati. Consente, inoltre, di modificare alcuni fondamentali parametri dell'esperimento di interesse tra i quali il volume sistolico e la frequenza cardiaca. È prevista anche la possibilità di analizzare e salvare gli andamenti di pressione rilevati dai sensori presenti nel circuito idraulico.

**Pannello Frontale** Il pannello frontale è l'interfaccia grafica su cui agire per l'utilizzo del programma, e ha la struttura grafica osservabile in figura 1.19.

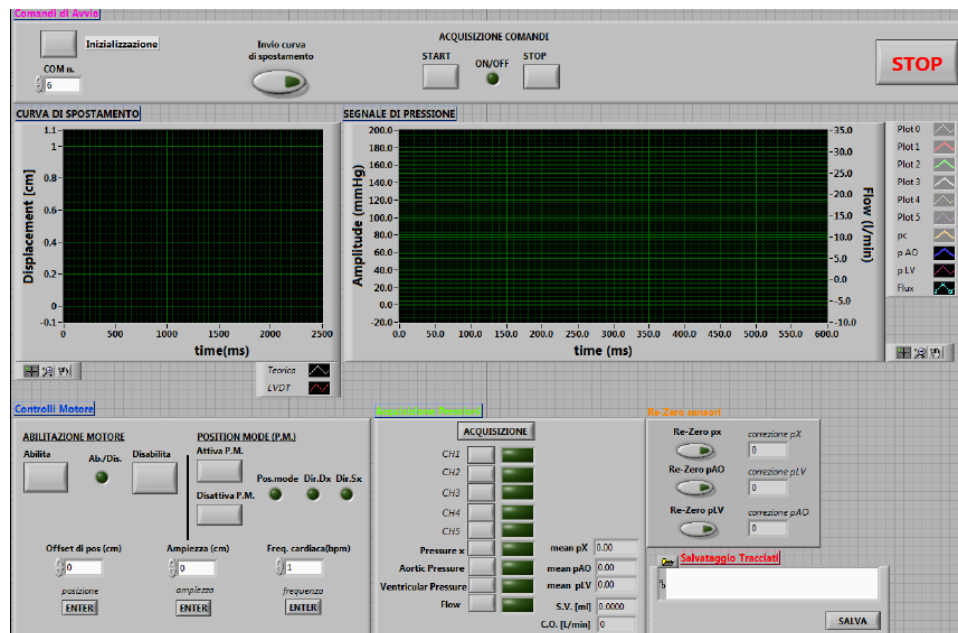


Figura 1.19: Pannello frontale del programma LabVIEW

Le azioni dell'operatore sui comandi disponibili vengono trasmesse al calcolatore sulla scheda dell'unità di controllo dove il programma *C* le interpreta per generare le rispettive procedure. I led presenti hanno il compito esclusivo di fornire segnali visivi all'utente in merito alle azioni svolte dai pulsanti e alle corrispondenti risposte elaborate dall'unità di controllo. I controlli numerici, invece, sono utilizzati per la selezione dei valori come offset e ampiezze da trasmettere. Le funzioni svolte [1] dai componenti sono, nell'ordine:

Inizializzazione: permette di resettare il sistema e di predisporre il dispositivo in modo che possa ricevere correttamente i comandi;

COM n: è il controllo numerico digitale necessario per selezionare la porta del calcolatore attraverso cui viene effettuata la trasmissione seriale dei dati;

Invio curva: è utilizzato per caricare il file contenente i campioni della curva di spostamento da far seguire al motore;

START e STOP: *START* attiva la fase di acquisizione del computer su scheda, mentre *STOP* interrompe la fase di acquisizione. Questi comandi attivano o disattivano il corrispondente led *ON/OFF*;

Abilita e Disabilita: Invia al computer i comandi che abilitano o disabilitano il motore elettrico. Conseguentemente provocano l'accensione o lo spegnimento del led *Ab./Dis.*;

Attiva e Disattiva P.M.: intervengono sulla motilità del motore, modificando anche la direzione di spostamento dello stesso. I tre led corrispondenti danno indicazioni riguardo alla modalità attivata e alla direzione in cui si sta muovendo l'attuatore;

Offset di posizione [cm]: permette di inserire la posizione di partenza dell'attuatore;

Ampiezza [cm]: serve per selezionare l'ampiezza del movimento del motore;

Freq. Cardiaca [bpm]: serve per selezionare la frequenza del ciclo;

Lettura: se premuto, consente di interrogare il sensore collegato al canale analogico selezionato tramite il controllore numerico Ingresso Analogico. La misura effettuata viene visualizzata in [V] e in [mmHg] nelle caselle di testo vicine;

Acquisisci pressione: consente di visualizzare gli andamenti di pressione nella relativa finestra grafica. È possibile visualizzare da uno a sei segnali contemporaneamente per un numero di periodi a piacere, attivando i tasti corrispondenti e digitando il numero di cicli nell'apposita casella di testo.

**Schema a Blocchi** L'architettura principale dello schema a blocchi è stata progettata come una sequenza di due strutture collegate, dette *frame*, eseguite in serie durante il funzionamento del programma. La prima

struttura ha il compito di inizializzare le variabili, predisporre il file per il salvataggio dei dati e fissare i parametri per la trasmissione via seriale. Tutte le istruzioni eseguite nella prima struttura vengono trasmesse al frame successivo.

Nel secondo *frame* della sequenza sono presenti tre cicli indipendenti: il primo gestisce la trasmissione dei comandi che l'operatore vuole inviare al computer; il secondo si occupa della ricezione delle risposte inviate dall'unità di controllo al computer, mentre il terzo è adibito alle operazioni di salvataggio dei dati.

## Capitolo 2

# Resistenza periferica automatizzata

Per poter effettuare agevolmente delle simulazioni in cui il valore della resistenza periferica cambi rapidamente, si è reso necessario automatizzare meccanicamente la valvola. Si considerano ora i passi necessari per la messa a punto di una resistenza periferica automatizzata, si vuole, cioè, motorizzare la valvola così da poterla controllare automaticamente tramite il software LabVIEW. Verranno poi illustrati gli algoritmi di controllo automatico capaci di individuare il valore ottimo di apertura della valvola, tale da garantire gli effetti desiderati sul sistema.

### 2.1 Componenti per l'automazione

Come mostrato in figura 2.1, i componenti *hardware* e *software* necessari per la messa a punto dell'automazione sono:

- Una scheda elettronica dotata di microcontrollore Arduino
- Un modulo di pilotaggio (H-bridge) per consentire il cambio del verso di rotazione del motore, e quindi l'apertura e la chiusura della valvola, con un sistema di protezione in caso di sovratensioni
- Un modulo *software* NI LabVIEW Interface per il Toolkit Arduino e un programma LabVIEW per controllare la valvola
- Un motoriduttore elettrico in corrente continua
- Una nuova valvola manuale a cui poter accoppiare il motore
- Un accoppiamento metallico in grado di trasferire il moto del motore alla valvola, consentendole di aprirsi e chiudersi

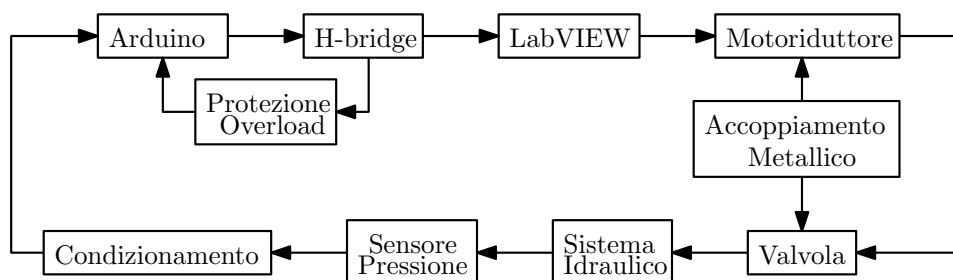


Figura 2.1: Diagramma a blocchi dei componenti per l'automazione

- Un sistema di condizionamento tale da rendere possibile la lettura del segnale di pressione aortica

### 2.1.1 Arduino

Arduino è una piattaforma *hardware* composta da una serie di schede elettroniche dotate di un microcontrollore. La piattaforma è stata ideata e sviluppata a partire dal 2003 da alcuni membri dell'*Interaction Design Institute* di Ivrea come strumento per la prototipazione rapida e per scopi hobbistici, didattici e professionali.

La piattaforma fisica di Arduino si basa su un circuito stampato che integra un microcontrollore con dei pin connessi alle porte I/O, un regolatore di tensione e, quando necessario, un'interfaccia USB che permette la comunicazione con il computer utilizzato per programmare. A questo *hardware* viene affiancato un ambiente di sviluppo integrato (IDE) multipiattaforma. I programmi in Arduino vengono chiamati *sketch* e sono scritti in un linguaggio di programmazione chiamato *Wiring*.

Una scheda Arduino tipica consiste in un microcontrollore a 8-bit AVR prodotto dalla Atmel, con l'aggiunta di componenti complementari per facilitare l'incorporazione in altri circuiti. Nelle schede vengono usati i chip della serie megaAVR - nello specifico i modelli ATmega8, ATmega168, ATmega328, ATmega1280 e ATmega2560. Molte schede Arduino includono un regolatore lineare di tensione a 5 Volt e un oscillatore a cristallo a 16MHz.

Per implementare il comportamento interattivo, Arduino è fornita di funzionalità di *input/output* (I/O). Tramite le funzionalità di *input*, la scheda riceve segnali raccolti da sensori esterni. Il comportamento della scheda è gestito dal microcontrollore in base ai valori provenienti dai sensori e alle operazioni determinate dal programma in esecuzione in quel momento sulla scheda. L'interazione con l'esterno avviene attraverso attuatori pilotati dal programma, che forniscono le istruzioni per mezzo dei canali di *output* in dotazione. Per l'implementazione della resistenza motorizzata si è scelto Arduino Uno [19], per via dell'ottimo rapporto tra le prestazioni necessarie



Figura 2.2: Scheda Arduino Uno

e il suo costo.

Arduino Uno, mostrato in figura 2.2 e 2.3, è una scheda con microcontrollore basata su ATmega328P. Ha 14 pin I/O digitali, 6 dei quali possono essere usati come uscite PWM, 6 ingressi analogici, un oscillatore a cristallo a 16MHz, una connessione USB, un jack di alimentazione, 13 led e un bottone di reset. La scheda contiene tutto il necessario per supportare il microcontrollore, si connette semplicemente al computer tramite un cavo USB e si alimenta con un adattatore AC-DC o una batteria. Il voltaggio raccomandato in input è 7-12V, la corrente continua per i pin I/O è 20mA e per il pin a 3.3V è di 50mA. La memoria flash è da 32KB, di cui 0.5KB sono usati dal *bootloader*<sup>1</sup>, una SRAM<sup>2</sup> da 2KB, una EPROM<sup>3</sup> da 1KB e una velocità di clock di 16MHz.

La scheda Arduino Uno può essere alimentata tramite connessione USB o con una fonte di alimentazione esterna. L'alimentazione esterna deve essere connessa inserendo una presa a centro positivo da 2.1mm nel jack della scheda.

---

<sup>1</sup>Programma che, durante la fase di avvio, carica il kernel (software che fornisce ai processi in esecuzione un accesso sicuro e controllato all'hardware) del sistema operativo dalla memoria secondaria a quella primaria, permettendone l'esecuzione da parte del processore e il conseguente avvio del sistema.

<sup>2</sup>Static Random Access Memory, memoria volatile che non necessita di frequenza di aggiornamento, ovvero consente di mantenere informazioni per un tempo teoricamente infinito.

<sup>3</sup>Erasable Programmable Read Only Memory, è una memoria di sola lettura programmabile e cancellabile tramite raggi ultravioletti.

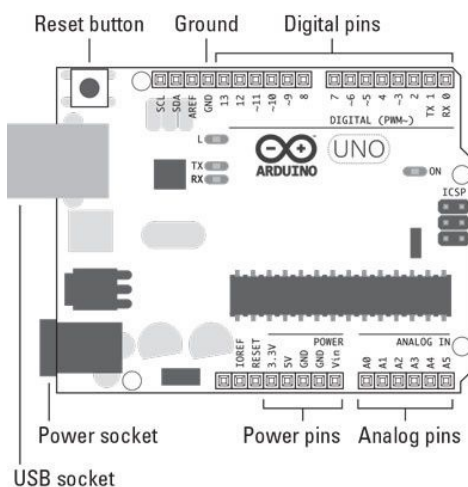


Figura 2.3: Schema di Arduino Uno

Ognuno dei 14 pin digitali di Arduino Uno può essere usato come ingresso o uscita. I pin operano a 5V, e ognuno può fornire o ricevere 20mA. Un massimo di 40mA è il valore che non si deve eccedere per evitare danni permanenti al microcontrollore. Sono presenti anche 6 ingressi analogici, ognuno dei quali prevede una risoluzione a 10-bit (1024 valori differenti). Di default essi misurano da terra a 5V e hanno una impedenza di ingresso da 100M $\Omega$ .

### 2.1.2 H-bridge

Per pilotare il motore elettrico in corrente continua si è utilizzato il *L298N Dual H-Bridge Motor Controller* [20], un controllore di piccole dimensioni ed estremamente compatto ( $H2.7cm \times W4.3cm \times D4.3cm$ ) e dal peso di soli 28g, mostrato in figura 2.4.

Si tratta di una scheda driver con integrato L298 della *STMicroelectronics*, al suo interno si trovano due ponti H, i quali supportano un elevato voltaggio (teoricamente sino a 46V) ed elevate correnti (2A per ponte). Ciascun ponte può essere disabilitato o abilitato tramite il relativo piedino PWM di *enable* per comandare un motore passo passo o due motori DC direttamente da Arduino o altri microcontrollori.

Per la sua alimentazione sono richiesti 5Vdc per la logica e una tensione compresa tra 5 e 35V per l'alimentazione del motore, così da garantire una corrente in uscita di 2A per ogni canale. Nel caso specifico l'alimentazione a 5V è garantita dal collegamento con la scheda Arduino, mentre l'alimentazione del motore è fornita da un alimentatore esterno da 12V e da un output massimo di 5A.





Figura 2.4: L298N Dual H-Bridge Motor Controller

Per controllare il verso di rotazione del motore<sup>4</sup>, è necessario invertire la direzione della corrente attraverso il motore tramite il ponte H. Questo componente comprende quattro elementi di switch, con il motore posto al centro in modo da formare una configurazione a H. Attivando due particolari switch contemporaneamente si può modificare la direzione della corrente elettrica, e quindi il verso del motore, come mostrato nelle figure 2.5 e 2.6.

L'integrato L298N presente nell'H-bridge, mostrato in figura 2.7, mette a disposizione un pin di sensing, ovvero un piedino a cui collegare una resistenza di sensing per avere informazione sulla corrente attraverso il carico (nell'applicazione specifica, il carico è rappresentato dal motoriduttore). Questo pin può essere sfruttato a livello di software per individuare i picchi di corrente assorbita dal motore quando esso è sottoposto a uno sforzo eccessivo, o overload (quando, ad esempio, il motore fornisce un comando di apertura alla valvola già completamente aperta). In questa applicazione si è individuata una soglia di tensione di 1V, ragionevolmente associata allo sforzo del motore, avendo scelto una resistenza di sensing da  $10\Omega$ <sup>5</sup>. Collegando il pin di sensing alla resistenza di sensing, e questa ad un ingresso analogico di Arduino, qualora il sistema restituisca un valore di tensione di sensing, correlato alla corrente attraverso il carico, maggiore o uguale a 1V, il programma LabVIEW arresta il motore.

Data la natura induttiva del carico, si è reso necessario filtrare i picchi di tensione generati dal cambio di verso di rotazione del motore; il risultato,

---

<sup>4</sup>Il verso di rotazione del motore implica l'apertura o la chiusura della valvola di resistenza periferica nel *pulse duplicator*.

<sup>5</sup>La calibrazione delle componenti è stata effettuata misurando tramite tester la corrente normalmente assorbita dal motoriduttore (circa 100mA), e volendo una tensione di soglia sotto sforzo dell'ordine di 1V, la resistenza è stata dimensionata a  $10\Omega$ .

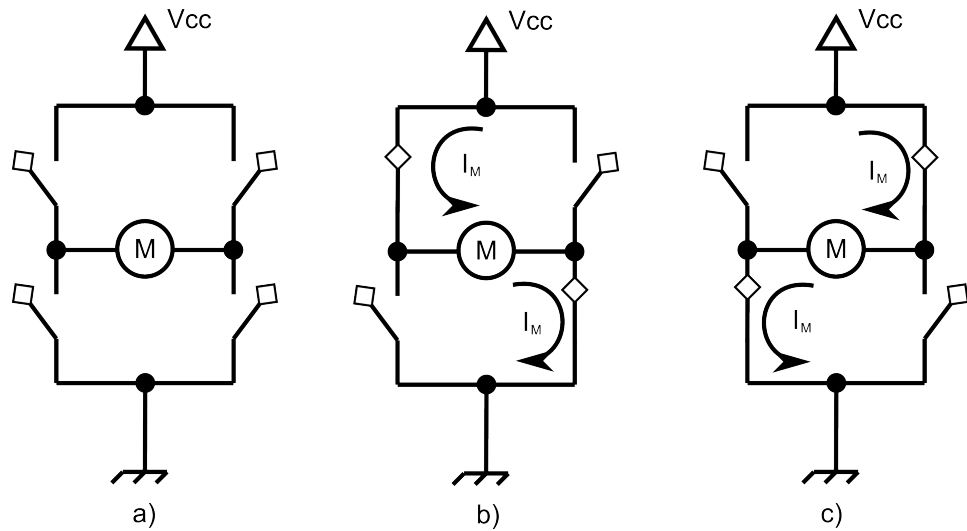


Figura 2.5: Schema di funzionamento del ponte H

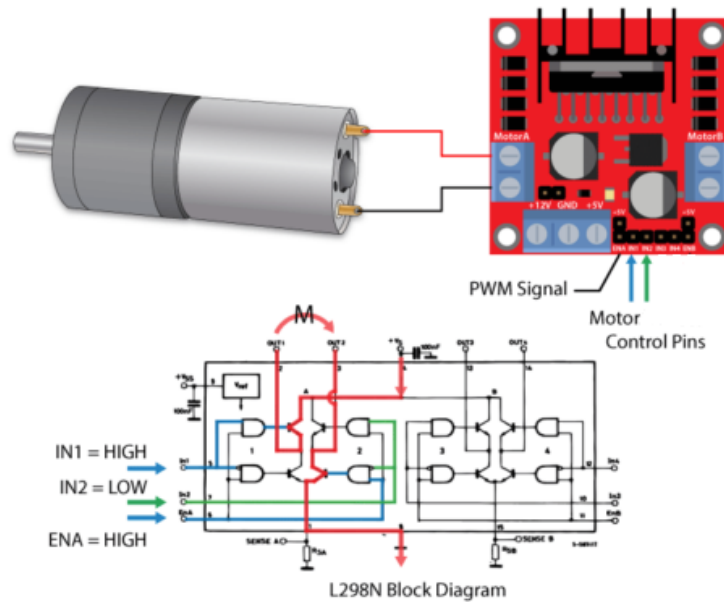


Figura 2.6: Gestione del motore tramite ponte H [21]

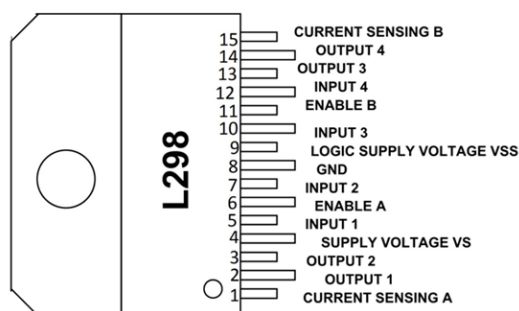


Figura 2.7: Schema del pin-out dell'integrato L298N

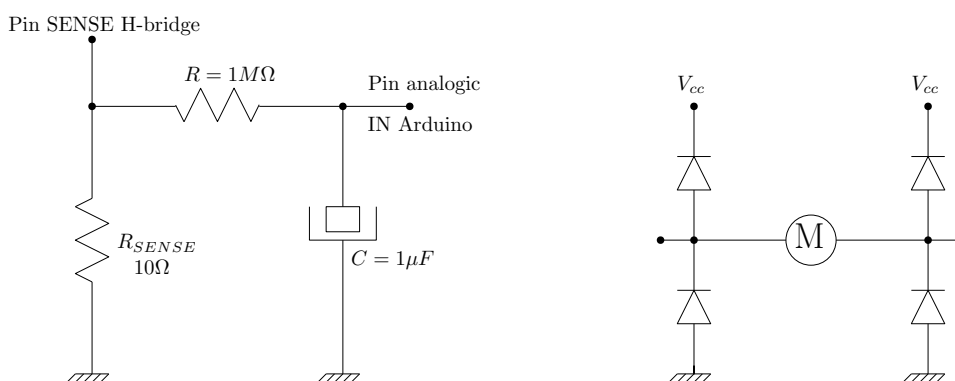
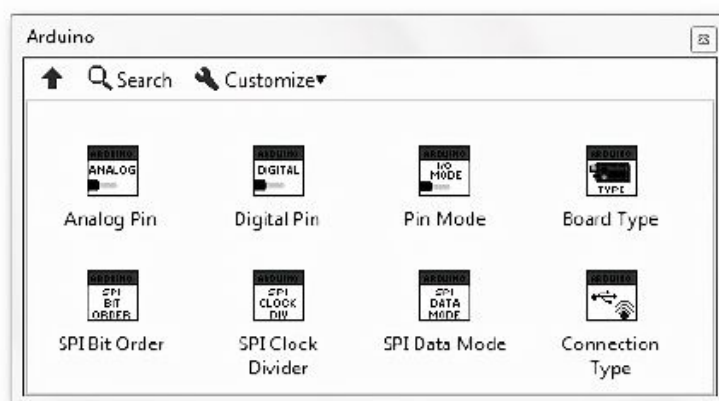
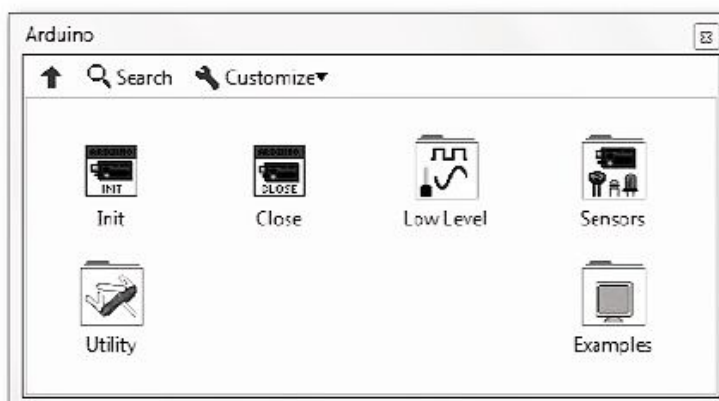


Figura 2.8: A sinistra, schema di filtraggio passabasso, a destra schema di limitazione delle sovratensioni tramite diodi di ricircolo

mostrato in figura 2.8, è ottenuto adottando un filtro passabasso passivo RC, con resistenza da  $1M\Omega$  e  $1/4W$ , e condensatore elettrolitico da  $1\mu F$ <sup>6</sup>. Sulla base di quanto indicato dai datasheet del componente L298N, si è anche provveduto ad inserire dei diodi di ricircolo ai capi del motore al fine di limitare le sovratensioni ai capi del motore, come mostrato in figura 2.8.

<sup>6</sup>Nei condensatori elettrolitici non è presente un materiale dielettrico, in quanto l'isolamento è dovuto alla formazione e mantenimento di un sottilissimo strato di ossido metallico sulla superficie di una armatura a contatto con una soluzione chimica umida. Vista la esiguità fisica del dielettrico, essi non possono sopportare tensioni elevate. A differenza dei condensatori comuni, la sottigliezza dello strato di ossido consente di ottenere, a parità di dimensioni, capacità molto più elevate.

(a) *Front panel*(b) *Block panel*Figura 2.9: Interfaccia utente del *NI LabVIEW Interface for Arduino Toolkit*

### 2.1.3 *Software* NI LabVIEW Interface per il Toolkit Arduino

La *LabVIEW Interface* per Arduino fornisce una pratica interfaccia per la piattaforma; permette di creare interfacce utente grafiche per qualsiasi componente che sia compatibile con il microcontrollore. Il *Toolkit* è *open source* e consente agli utenti di creare driver personalizzati per i sensori di interesse. Si può utilizzare Arduino come device di acquisizione per misurare, controllare e monitorare diversi applicativi e sensori. In figura 2.9 si mostrano le interfacce del *NI LabVIEW Interface for Arduino Toolkit* sia per il *front panel*, sia per il *block panel*. Sul sito *National Instruments*, si può installare il *LINX MakerHub* per scaricare l'interfaccia desiderata.

### 2.1.4 Motoriduttore in corrente continua

Per poter automatizzare il movimento della valvola si è scelto un motoriduttore elettrico, con motore in corrente continua da 12V ad alta coppia mostrato in figura 2.10 e 2.11. Il motoriduttore contiene una vite senza fine che lo rende autobloccante, ovvero in assenza di corrente elettrica, l'albero di uscita del motore rimane fermo.

L'albero di uscita del riduttore è disposto verticalmente e, essendo in acciaio ad alta durezza, consente buona tenuta e resistenza agli urti, garantendo una buona durata di vita del motore.

Il motoriduttore consente una rotazione dell'albero a velocità costante di 2rpm, una coppia di carico di 10kg·cm, una coppia di stallo di 28.2kg·cm, una corrente di stallo di 1.14A, una corrente di *no-load* di 0.22A e rapporto di riduzione di 1:400.

Il collegamento elettrico del motore si è effettuato stagnando i due fili in input al motore, GND e +12V tramite saldatura.

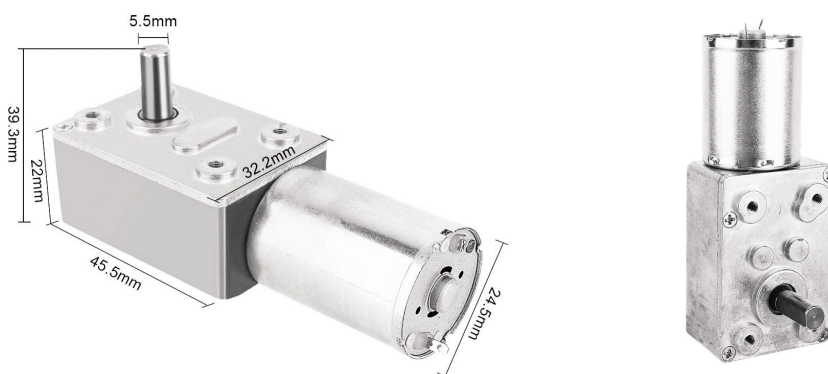


Figura 2.10: Motoriduttore DC da 12V

### 2.1.5 Valvola manuale

Per controllare tramite software il livello di apertura della valvola, si è reso necessario l'acquisto di una nuova valvola manuale da modificare opportunamente e sostituire a quella già presente; si è scelto un modello a saracinesca realizzato in ottone con manopola di chiusura a vitone, attacco FF con diametro nominale di 1" (DN25) e PN12, mostrata in figura 2.12. Per poter accoppiare meccanicamente lo stelo della valvola al motoriduttore, si è reso necessario eliminare il volantino manuale.

Il prodotto ha dimensioni  $H11cm \times W5cm \times D4.7cm$  e un peso di 266g.

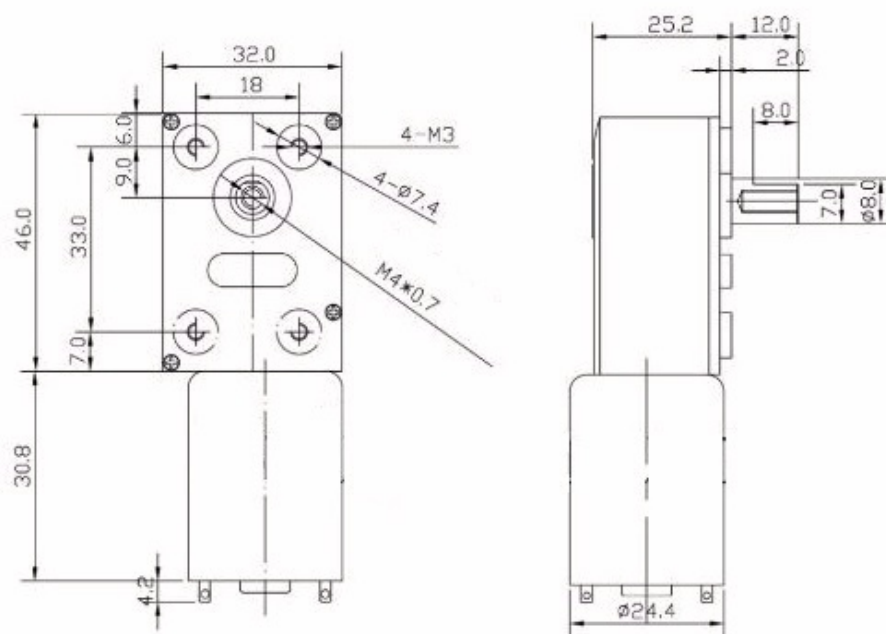


Figura 2.11: Disegno CAD del motoriduttore DC da 12V



Figura 2.12: Valvola manuale a saracinesca

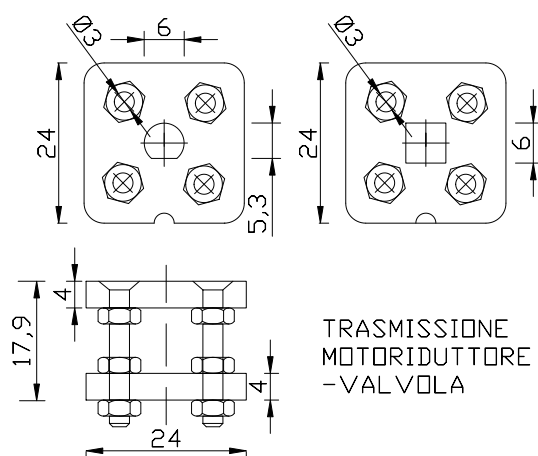


Figura 2.13: Disegno CAD del sistema di trasmissione motoriduttore-valvola

### 2.1.6 Accoppiamenti metallici

I componenti metallici per accoppiare il motoriduttore allo stelo della valvola manuale sono di due tipi: il complesso per la trasmissione del moto dall'albero del motoriduttore allo stelo della valvola, e il complesso di supporto con il compito di solidarizzare la valvola al corpo motore.

La trasmissione, mostrata in figura 2.13 e 2.14, è costituita da una torretta composta da una piastrina in acciaio di 4mm di spessore con impronta a chiavetta, tale da accogliere l'albero di acciaio del motoriduttore, e da una piastrina in ottone di 4mm di spessore con impronta quadra, tale da accogliere lo stelo in ottone della valvola. Il collegamento solidale di queste due piastrine è garantito da 4 viti in acciaio inox M3<sup>7</sup>, bloccate con dadi M3, in acciaio inox dal lato del motore, e in ottone dal lato della valvola.

La struttura di supporto mostrata in figura 2.16 e 2.15, invece, è formata da tre piastrine di alluminio anticorrosione<sup>8</sup> di cui la superiore, vincolata al motore, di 8mm di spessore, e dal lato della valvola due di 6mm di spessore ciascuna. I distanziali tra le piastrine sono barre filettate M5, bloccate con dadi, dadi autobloccanti e rondelle dal corrispettivi diametro nominale.

<sup>7</sup>Secondo lo standard ISO, riconosciuto dall'Organizzazione Internazionale per la Normazione a partire dal 1947, la lettera M indica una filettatura di tipo metrico, mentre il numero che segue indica il suo diametro nominale in millimetri.

<sup>8</sup>L'alluminio anticorrosione è contraddistinto da buone caratteristiche meccaniche e da un'ottima resistenza alla corrosione, presenta buona formabilità, lavorabilità, truciolabilità e saldabilità.



Figura 2.14: Sistema di trasmissione motoriduttore-valvola



Figura 2.15: Sistema di accoppiamento motoriduttore-valvola



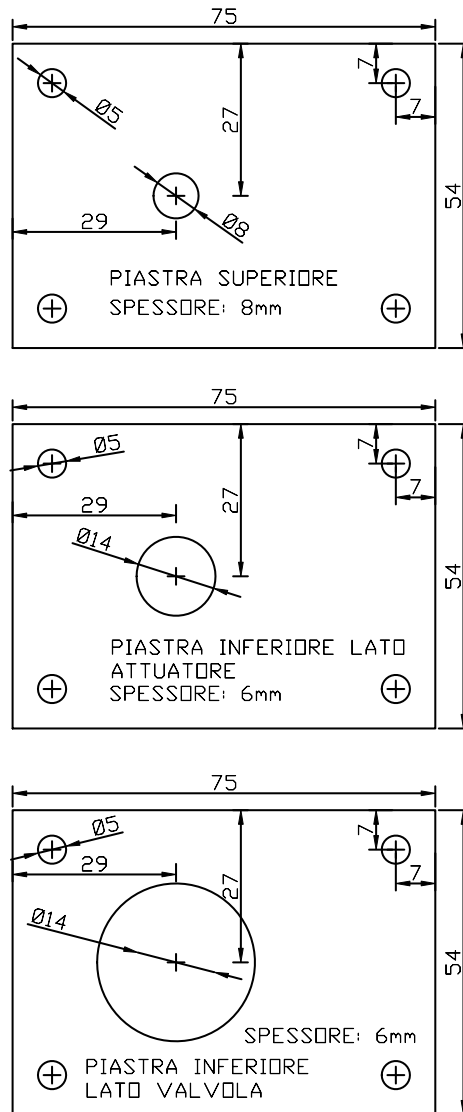


Figura 2.16: Disegno CAD della struttura di accoppiamento motoriduttore-valvola

### 2.1.7 Sistema di condizionamento

Avendo riscontrato la presenza di disturbi, rilevabili tramite un programma LabVIEW, di entità tale da compromettere la convergenza dell'algoritmo, si è reso necessario introdurre un sistema di condizionamento del segnale in uscita dal sensore di pressione aortica. Si ipotizza che una possibile causa della presenza dei disturbi sia l'alta impedenza ( $100M\Omega$ ) dell'ingresso ana-

logico della scheda Arduino. Si è ritenuto, pertanto, di introdurre il blocco di condizionamento del segnale con l'obiettivo di disaccoppiare la sorgente (sensore di pressione), dall'ingresso analogico di Arduino.

Le necessità da soddisfare per una corretta acquisizione del segnale di pressione sono:

- Filtrare il segnale tramite un passa-basso del primo ordine, con frequenza di taglio di circa 100 volte la fondamentale del segnale utile. Dato che essa è di circa 1Hz (60bpm), si è ritenuto che una frequenza di taglio di 100Hz preservasse le caratteristiche del segnale, attenuando con efficacia i disturbi ad alta frequenza.
- Avere un guadagno unitario in condizioni non invertenti per non attenuare o modificare il segnale di pressione letto.
- Avere una resistenza in ingresso sufficientemente elevata tale da garantire di introdurre un errore di al più 1% sul segnale utile. Si è ritenuto che un'impedenza d'ingresso del modulo di condizionamento del segnale di circa 100k $\Omega$  fosse tal da garantire un margine d'errore accettabile vista l'impedenza d'uscita del sensore di pressione, circa 1k $\Omega$ .
- Avere una impedenza d'uscita al sistema di condizionamento trascurabile rispetto all'impedenza di ingresso di Arduino (100M $\Omega$ ), tale da rendere il segnale letto dalla scheda assimilabile ad un generatore ideale.

Per soddisfare ognuna delle necessità e disaccoppiare il sensore e Arduino, si è deciso di utilizzare un amplificatore differenziale con ingresso invertente a GND, in particolare l'amplificatore  $\mu$ A741, il cui schema è mostrato in figura 2.17.

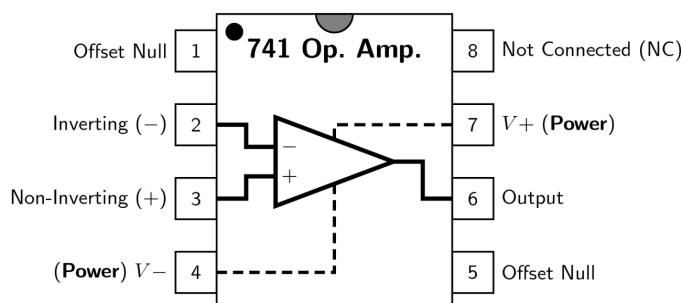


Figura 2.17: Schema dell'amplificatore  $\mu$ A741

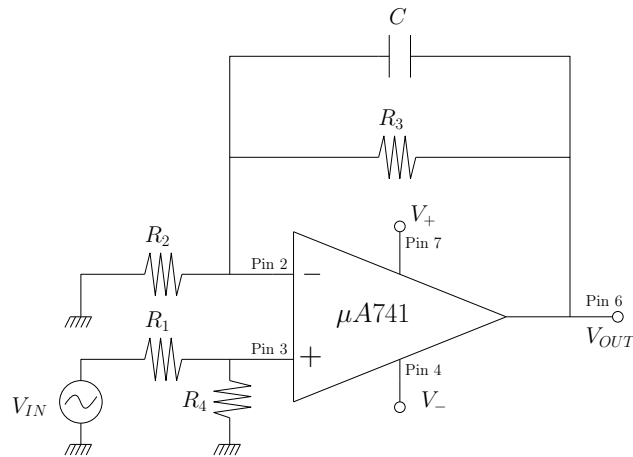


Figura 2.18: Schema del condizionamento

Il sistema di condizionamento e il suo schema sono mostrati in figura 2.19 e 2.18.

Il dimensionamento dei componenti deriva dai vincoli e dalle necessità descritte: l'impedenza di ingresso desiderata ha reso necessaria la scelta di resistenze da  $100k\Omega$ , e la frequenza di taglio utile ha reso necessario l'utilizzo di un condensatore da  $100nF$ . Inoltre, dato che il segnale generato dal sensore di pressione varia in un range  $0-2V_p$ , si è reso necessario alimentare l'amplificatore operazionale con un'alimentazione bilanciata  $\pm 12V$ , tale da garantire l'assenza di effetti distorsivi del segnale.

Facendo riferimento a figura 2.18,  $V_{IN}$  è il segnale di pressione,  $V_+$  è l'alimentazione a  $+12V$ ,  $V_-$  è l'alimentazione a  $-12V$  e  $V_{OUT}$  è il segnale in uscita da leggere tramite Arduino. Ricordando che  $R_1 = R_2 = 100k\Omega$  e  $R_3 = R_4 = 100k\Omega$ , allora:

$$\begin{aligned} V_{out} &= V_{IN} \frac{R_4}{R_1 + R_4} \frac{R_2 + R_3}{R_2} \\ &= V_{IN} \end{aligned} \quad (2.1)$$

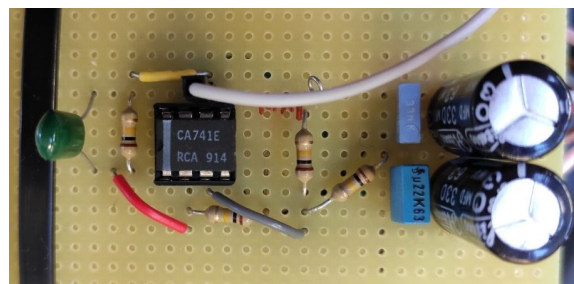


Figura 2.19: Condizionamento del segnale di pressione

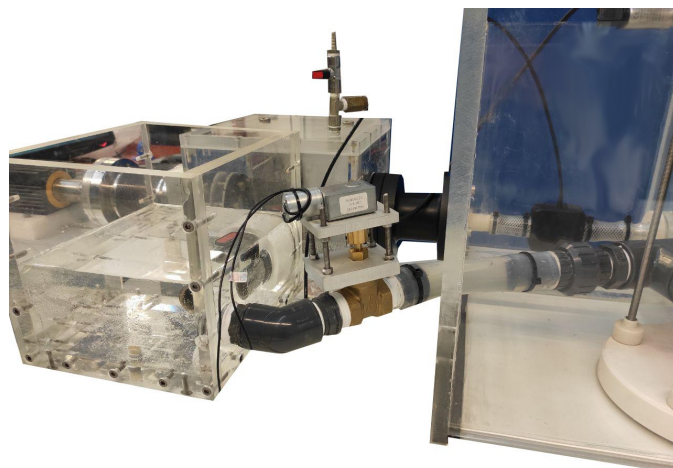


Figura 2.20: Complesso motoriduttore-valvola inserito nel *pulse duplicator*

## 2.2 Assemblaggio

Il complesso motoriduttore-valvola, completo di accoppiamenti metallici è stato inserito nel *pulse duplicator*, come mostrato in figura 2.20.

L'assemblaggio dei componenti necessari ad automatizzare la valvola si può dividere in elettrico e meccanico.

Per quanto riguarda gli accoppiamenti elettrici, in figura 2.21, si mostra lo schema di collegamento tra alimentazione, ponte H, Arduino e motoriduttore. Arduino viene configurato con tre uscite digitali: una PWM per abilitare il ponte H a muovere il motore, e due uscite per gestire la rotazione nelle due direzioni, e quindi l'apertura e la chiusura della valvola. Esso è collegato anche all'alimentazione tramite i due terminali: +12V e GND.

Il ponte H, oltre ai collegamenti digitali con Arduino, è collegato ai due fili del motoriduttore. Esso è collegato anche all'alimentazione tramite i due terminali: +12V e GND.

Per il corretto funzionamento del complesso, è necessario alimentare Arduino, H-bridge e circuito di sensing tramite una fonte di alimentazione comune, indicati in figura 2.21 con tre batterie, in modo da garantire che ogni elemento abbia lo stesso terminale di GND di riferimento. Si collega, inoltre, la  $V_{IN}$  del sistema di controllo con il sensore di pressione aortica e la  $V_{OUT}$  con un ingresso analogico di Arduino

Gli accoppiamenti metallici, invece, sono ottenuti assemblando la torretta di trasmissione contemporaneamente all'albero del motoriduttore e allo stelo della valvola, e il sistema di supporto in modo solidale alla valvola, mostrato in figura 2.22.

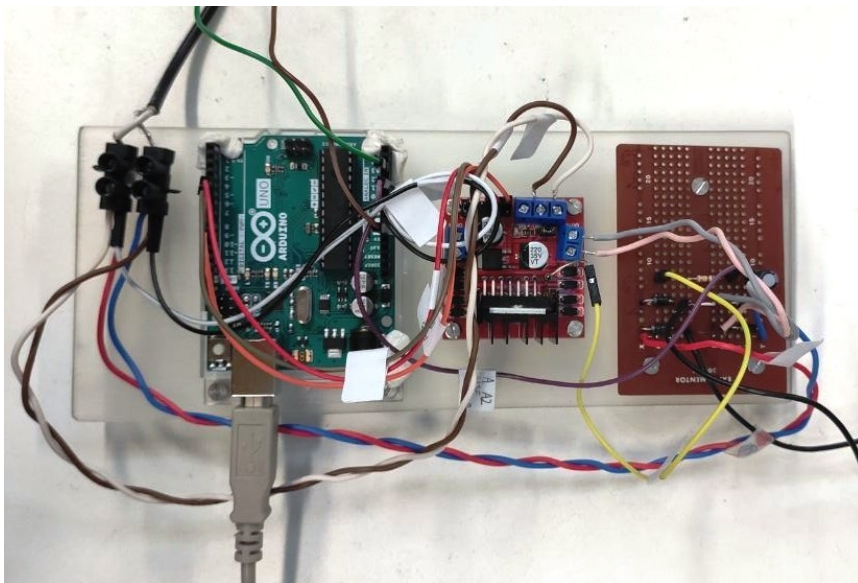
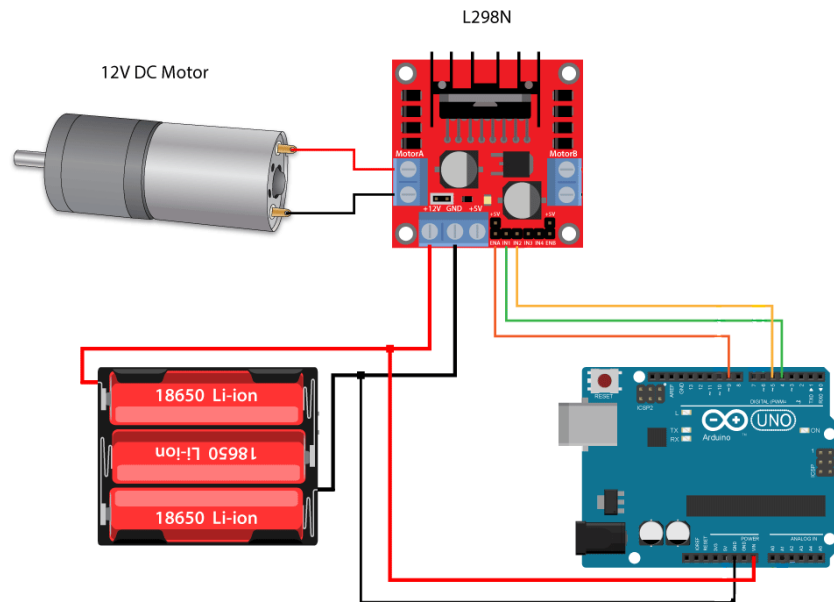


Figura 2.21: Collegamenti elettrici tra motore, ponte H, Arduino e alimentazione [21]

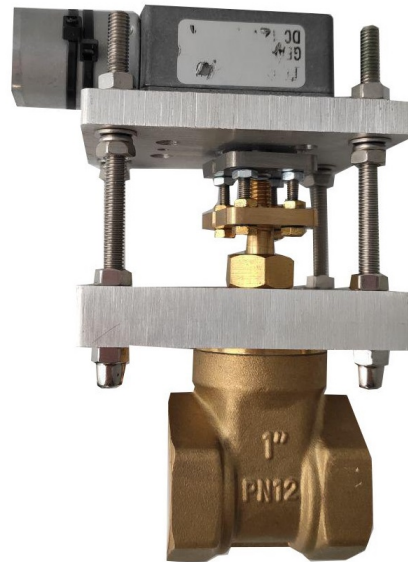
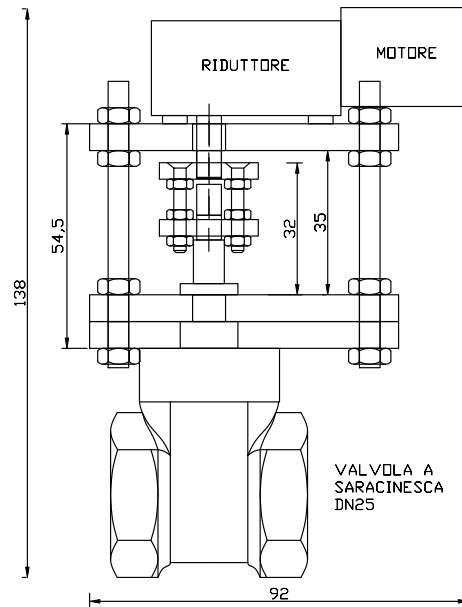


Figura 2.22: Sistema di trasmissione motoriduttore-valvola

## Capitolo 3

# Algoritmo *Extremum Seeking Control*

### 3.1 Inquadramento storico

L'origine di *ESC* può essere riportata al 1922, quando Leblanc propose un meccanismo di controllo in grado di mantenere il massimo trasferimento di potenza desiderato, usando una soluzione che essenzialmente può definirsi *extremum seeking*. Nei suoi scritti [10], Leblanc non fornisce alcuna analisi o valutazione pratica del suo algoritmo di controllo. Il primo articolo in letteratura che descrive dettagliatamente e fornisce una valutazione delle performance dell'algoritmo, viene pubblicato nel 1951 [11] e tratta il problema di ottimizzazione di un motore a combustione interna.

In seguito, negli anni '50 e '60 l'*ESC* diviene un popolare argomento di ricerca, così come altre forme di controllo adattativo. Nel 2000 la comunità del controllo automatico ha rinnovato il suo interesse verso l'algoritmo grazie ai risultati presentati da Wang e Krstic [12]. Nel loro lavoro Wang e Krstic hanno dimostrato che il classico metodo *ESC* basato su un segnale perturbante possiede una soluzione stabile in un intorno dell'ottimo in stato stazionario, quando applicato a una vasta classe di sistemi dinamici non lineari.

Il primo metodo per l'*extremum seeking* multiparametrico è stato presentato da Rotea [13] e Walsh [14] nel 2000.

### 3.2 Introduzione all'algoritmo

*Extremum Seeking Control* o *ESC* è una forma di controllo adattativo, dove la caratteristica *input-output* in stato stazionario è ottimizzata senza richiedere alcuna specifica conoscenza riguardo la caratteristica stessa, se non la sua esistenza e la presenza di un punto di estremo, sia esso di minimo o di massimo.

Per via del fatto che *Extremum Seeking* è *model-free*, esso risulta essere robusto ed efficiente in differenti domini di applicazione [7].

Paradossalmente, benché sia un algoritmo *model-free*, *ESC* è una tecnica di ottimizzazione per la ricerca di un punto di estremo basata sul gradiente e viene utilizzata per determinare e mantenere il valore di massimo o di minimo di una funzione. L'incertezza della mappa ingresso-uscita rende necessario, quindi, usare qualche sorta di controllo adattativo per individuare il *set point* che estremizza (massimizza o minimizza) l'*output*.

L'*output* del sistema potrebbe essere influenzato anche da parametri  $p$ , allora l'algoritmo di *Extremum Seeking* individua  $\vartheta = \vartheta^*(p)$ , punto di minimo o massimo, e lo mantiene nonostante le (lente) variazioni di  $p$ . Questa strategia di controllo, infatti, è locale e adattativa, e non necessita delle equazioni che modellano il sistema, ma si adatta al variare dei parametri nel tempo, ovvero compensa i loro cambiamenti senza modellarli o misurarli. In particolare, la condizione iniziale per  $\vartheta$  non deve essere necessariamente vicina al valore  $\vartheta^*$  desiderato.

Esistono due principali approcci all'*Extremum Seeking* [7]. Essi prevedono di:

- Usare un segnale perturbante continuo per esplorare, in stato stazionario, la mappa ingresso-uscita, al fine di ottenere un'approssimazione del gradiente.
- Usare una sequenza ripetuta di *input* costanti, i quali permettono di raggiungere l'obiettivo tramite metodi di ottimizzazione numerica.

In questa tesi si segue l'approccio più classico di una perturbazione sinusoidale continua, la quale permette un veloce adattamento che non necessita della stabilizzazione del sistema antecedente all'ottimizzazione.

In figura 3.1 si mostra l'*output* dell'algoritmo *ESC* quando il punto operativo è maggiore, uguale o minore del punto di estremo. Dato che il prodotto di due segnali in fase fornisce un segnale a media positiva, e che questo prodotto risulta una media negativa per due segnali fuori fase, si può sfruttare tale caratteristica per individuare il punto operativo tramite lo studio del gradiente [15].

Si considera un sistema con ingresso  $\vartheta$  e uscita  $f(\vartheta)$ , come, ad esempio, quello di figura 3.1 con una ben definita caratteristica di ingresso-uscita in stato stazionario; dato un *input* costante  $\vartheta$  nel dominio di interesse, l'*output* si porta a un valore costante  $f(\vartheta)$ .

Si suppone che esista un unico valore ottimo del controllo  $\vartheta^*$ , corrispondente all'estremo della funzione obiettivo (supposta con un unico massimo o minimo), e, anche se questa cambiasse nel tempo, la procedura di controllo deve permettere di individuare il valore  $\vartheta^*$  che la estremizzi.



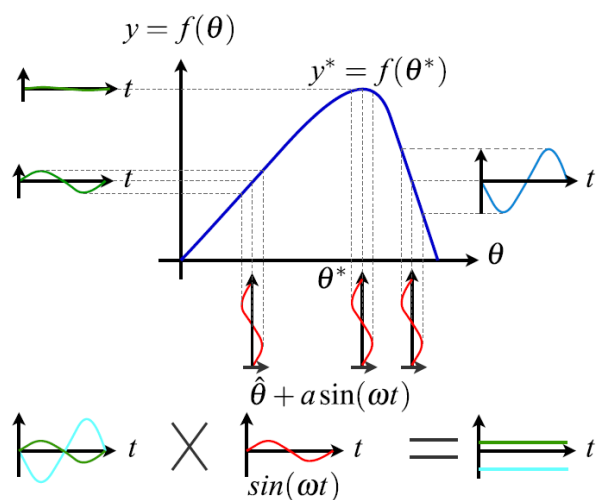


Figura 3.1: Operazioni dell'algoritmo ESC per una mappa statica non lineare [15]

Si considera ora lo schema a blocchi dell'algoritmo, rappresentato in figura 3.2.

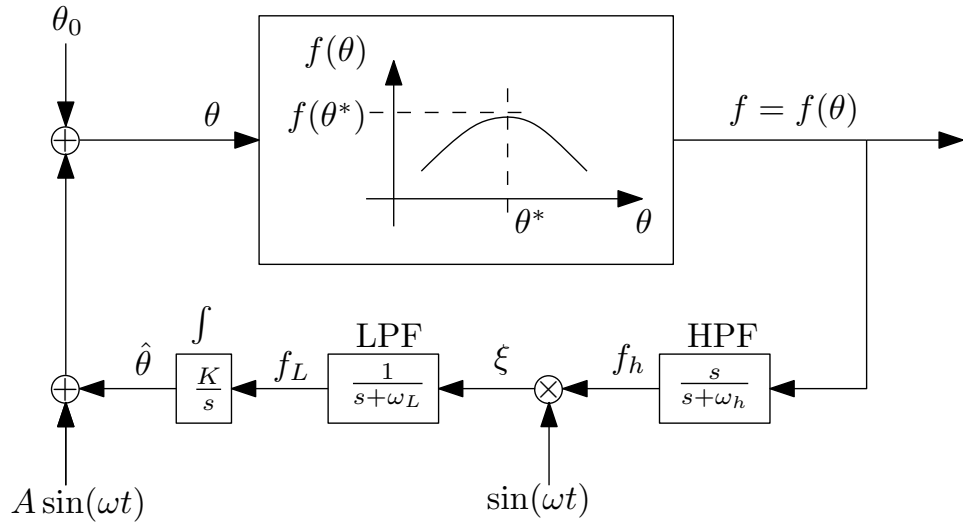
Si suppone di avere una stima del controllo ottimo,  $\hat{\vartheta}$ , e di sommarle una perturbazione sinusoidale  $A \sin(\omega t)$ .

Il ruolo della perturbazione continua è quello di testare quale sia il gradiente della funzione obiettivo in quel particolare valore di  $\hat{\vartheta}$  e, quindi, di regolare l'ingresso di conseguenza.

Successivamente, si misura l'effetto della perturbazione sinusoidale sull'uscita: questa potrà essere in fase con la perturbazione in ingresso, allora  $\hat{\vartheta}$  dovrà essere aumentato per avvicinarsi al suo valore ottimo, o potrà essere in controfase e quindi  $\hat{\vartheta}$  dovrà essere diminuito.

Dato che la perturbazione sinusoidale in uscita non ha media nulla, per eliminarne la componente continua si filtra il segnale con un filtro passa-alto. Moltiplicando il segnale in uscita al filtro passa-alto per una perturbazione sinusoidale con la stessa pulsazione dell'onda in ingresso, si ottiene un segnale demodulato, ovvero prettamente positivo se il gradiente della funzione era risultato positivo, negativo altrimenti. Il segnale viene poi integrato.

Essendo l'ESC, un algoritmo basato sul gradiente, quando  $\hat{\vartheta}$  si trova in prossimità del punto di estremo, il contributo della perturbazione sinusoidale risulta molto ridotto e qualitativamente la stima varia molto lentamente. Più  $\hat{\vartheta}$  si discosta dall'estremo, maggiore sarà il gradiente della funzione, e maggiore sarà la correzione della stima di  $\vartheta$  [6].

Figura 3.2: Schema a blocchi dell'algoritmo *Extremum Seeking Control*

### 3.3 Algoritmo *ESC* per una mappa statica

Il classico algoritmo *ESC* è rappresentato in figura 3.2. In questo classico diagramma i parametri da progettare sono [7]:

- Il guadagno  $A$ , che determina l'ampiezza del segnale perturbante.
- La pulsazione del segnale perturbante  $\omega$ .
- La frequenza di *cut-off* del filtro passa-alto  $\omega_h$ .
- La frequenza di *cut-off* del filtro passa-basso  $\omega_L$ .
- Il guadagno  $K$ , che scala il segnale dell'integratore, il quale determina  $\hat{\vartheta}$ .

Si considera  $f(\vartheta^*)$  della forma [6]

$$f(\vartheta) = f^* + \frac{f''}{2}(\vartheta - \vartheta^*)^2 \quad (3.1)$$

dove  $f'' > 0$ . Ogni funzione  $f(\vartheta)$   $C^2$  può essere approssimata localmente dall'equazione (3.1).

L'assunzione  $f'' > 0$  non riduce la generalità del metodo, infatti se  $f'' < 0$ , basterà utilizzare un guadagno  $K$ , in figura 3.2, negativo.

L'obiettivo dell'algoritmo è di minimizzare  $\vartheta - \vartheta^*$ , così che l'*output*  $f(\vartheta)$  sia portato al suo valore ottimo  $f^* = f(\vartheta^*)$ .

Il segnale perturbante,  $A \sin(\omega t)$ , introdotto nel sistema, aiuta a dare una misura dell'informazione portata dal gradiente della mappa  $f(\vartheta)$ .

Si fornisce ora l'analisi e la spiegazione dello schema in figura 3.2.

Si sottolinea che  $\hat{\vartheta}$  denota la stima dell'input ottimo incognito  $\vartheta^*$ . Si pone

$$\tilde{\vartheta} = \vartheta^* - \hat{\vartheta}$$

errore di stima. Allora,

$$\vartheta - \vartheta^* = A \sin(\omega t) - \tilde{\vartheta}$$

e se lo si sostituisce nell'equazione (3.1), porge

$$f(\vartheta) = f^* + \frac{f''}{2}(\tilde{\vartheta} - A \sin(\omega t))^2. \quad (3.2)$$

Espandendo questa espressione e applicando l'identità trigonometrica  $2\sin^2(\omega t) = 1 - \cos(2\omega t)$  si ottiene

$$f(\vartheta) = f^* + \frac{f''}{2}\tilde{\vartheta}^2 - A f'' \tilde{\vartheta} \sin(\omega t) + \frac{A^2 f''}{2} \sin^2(\omega t) \quad (3.3)$$

$$f(\vartheta) = f^* + \frac{A^2 f''}{4} + \frac{f''}{2}\tilde{\vartheta}^2 - A f'' \tilde{\vartheta} \sin(\omega t) + \frac{A^2 f''}{4} \cos(2\omega t) \quad (3.4)$$

Il filtro passa-alto o *washout*

$$\frac{s}{s + \omega_h}$$

applicato all'*output* permette di filtrare la componente  $f^*$  e priva, quindi, il segnale della sua componente media:

$$\frac{s}{s + \omega_h} f(\vartheta) \approx \frac{f''}{2}\tilde{\vartheta}^2 - A f'' \tilde{\vartheta} \sin(\omega t) + \frac{A^2 f''}{4} \cos(2\omega t). \quad (3.5)$$

Successivamente il segnale viene demodulato tramite la moltiplicazione per  $\sin(\omega t)$ , che estrae la componente utile del segnale:

$$\xi \approx \frac{f''}{2}\tilde{\vartheta}^2 \sin(\omega t) - A f'' \tilde{\vartheta} \sin^2(\omega t) + \frac{A^2 f''}{4} \cos(2\omega t) \sin(\omega t). \quad (3.6)$$

Si può notare la crucialità del secondo termine e, in particolare, la componente continua o DC in  $\sin^2(\omega t)$ . Applicando nuovamente  $2\sin^2(\omega t) = 1 - \cos(2\omega t)$ , così come l'identità

$$2\cos(2\omega t) \sin(\omega t) = \sin(3\omega t) - \sin(\omega t)$$

si ottiene

$$\xi \approx -\frac{A f''}{2} \tilde{\vartheta} + \frac{A f''}{2} \tilde{\vartheta} \cos(2\omega t) + \frac{A^2 f''}{8} (\sin(\omega t) - \sin(3\omega t)) + \frac{f''}{2} \tilde{\vartheta}^2 \sin(\omega t). \quad (3.7)$$

Notando che  $\vartheta^*$  è costante e  $f(\vartheta^*) = 0$  perché punto di estremo, allora

$$\dot{\tilde{\vartheta}} = -\hat{\vartheta}$$

si ottiene

$$\tilde{\vartheta} \approx \frac{K}{s} \left[ -\frac{Af''}{2}\tilde{\vartheta} + \frac{Af''}{2}\tilde{\vartheta}\cos(2\omega t) + \frac{A^2f''}{8}(\sin(\omega t) - \sin(3\omega t)) + \frac{f''}{2}\tilde{\vartheta}^2\sin(\omega t) \right]. \quad (3.8)$$

Si può trascurare l'ultimo termine della (3.8) perché esso è quadratico in  $\tilde{\vartheta}$  e l'analisi di interesse è solo locale, allora:

$$\tilde{\vartheta} \approx \frac{K}{s} \left[ -\frac{Af''}{2}\tilde{\vartheta} + \frac{Af''}{2}\tilde{\vartheta}\cos(2\omega t) + \frac{A^2f''}{8}(\sin(\omega t) - \sin(3\omega t)) \right]. \quad (3.9)$$

Gli ultimi due termini della (3.9) sono i segnali in alta frequenza e, una volta filtrati con un passa-basso e un integratore, vengono in gran parte attenuati. Essi vengono, perciò, trascurati dall'analisi, ottenendo:

$$\tilde{\vartheta} \approx \frac{K}{s} \left[ -\frac{Af''}{2}\tilde{\vartheta} \right] \quad (3.10)$$

$$\dot{\tilde{\vartheta}} \approx -\frac{KAf''}{2}\tilde{\vartheta}. \quad (3.11)$$

Dato che  $Kf'' > 0$  il sistema è stabile. Si può concludere, quindi, che  $\tilde{\vartheta} \rightarrow 0$  o, nei termini del problema originario,  $\hat{\vartheta}(t)$  converge a meno della piccola distanza  $\vartheta^*$ .

Risulta importante notare che le approssimazioni finora considerate sono verificate solamente quando  $\omega$  è sufficientemente maggiore (qualitativamente) di  $K, A, h$  e  $f''$ .

## Capitolo 4

# Algoritmo *ESC Sliding Mode*

L'algoritmo ESC considera il problema di tracciare un ingresso  $x$ , il quale ottimizza una funzione  $y(x)$ , generalmente incognita e tempo-variante,

$$x^* = \operatorname{argmax}_{x \in \mathbb{R}} y(x) \quad (4.1)$$

ed equivalentemente

$$x^* = \operatorname{argmin}_{x \in \mathbb{R}} y(x) \quad (4.2)$$

dove  $x^*$  è l'argomento che massimizza o minimizza la funzione  $y(x)$ . Si assume che esista un unico punto di massimo o minimo in  $y(x)$ , cioè che la funzione sia unimodale.

Si considera ora una variante all'algoritmo che non necessita di segnali perturbatori esterni, tratta da [16] e [17].

Lo Sliding Mode è una tecnica di controllo non lineare, che altera le dinamiche di un sistema non lineare tramite l'applicazione di un segnale di controllo discontinuo, tale da spingere le traiettorie del sistema ad una superficie, detta di Sliding.

Uno dei principali vantaggi dell'ESC Sliding Mode è che non necessita di sensori del gradiente, o differenziatori, i quali tendono ad amplificare il rumore, soffrono di instabilità ad alte frequenze e tipicamente agiscono in una banda sicura limitata, vincolando le prestazioni dinamiche del sensore.

Il comportamento dell'algoritmo è indicato in figura 4.1, dove in particolare:

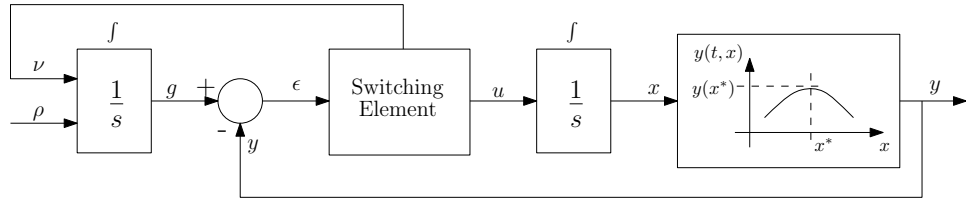
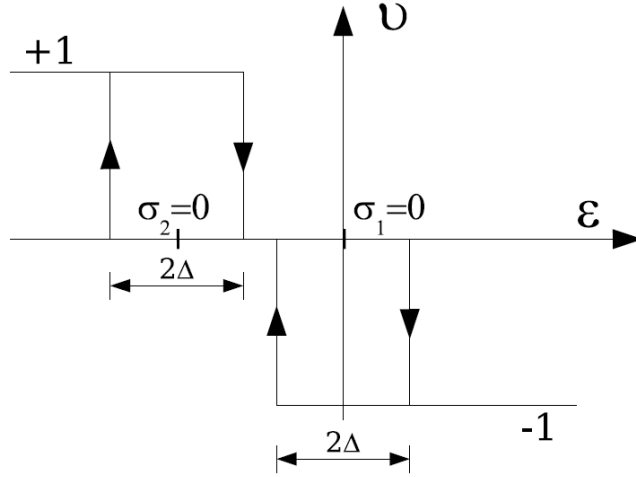


Figura 4.1: Diagramma a blocchi dell'algoritmo ESC Sliding Mode

Figura 4.2: Funzione dello Switching Element  $\nu(\sigma_1\sigma_2)$  [17]

$$u = \frac{dx}{dt}, \quad (4.3)$$

$$u = U_0 \text{sign}(\sigma_1\sigma_2), \quad (4.4)$$

$$\sigma_1 = \epsilon, \quad (4.5)$$

$$\sigma_2 = \epsilon + \delta, \quad (4.6)$$

$$\epsilon = g - y, \quad (4.7)$$

$$\frac{dg}{dt} = \rho + M\nu(\sigma_1, \sigma_2), \quad (4.8)$$

dove  $U_0, \delta, \rho, M$  sono costanti positive e  $\nu$  è funzione di  $\sigma_1$  e  $\sigma_2$ , mostrata in figura 4.2. Si considera la seguente disuguaglianza

$$M > U_0 \left| \frac{df}{dx} \right| + \rho, \quad (4.9)$$

allora, assumendo lo stato iniziale  $(\sigma_1 - \Delta)(\sigma_2 + \Delta) > 0$ , la variazione di  $\sigma_1$  e  $\sigma_2$  è

$$\dot{\sigma}_1 = \dot{\sigma}_2 = \frac{df}{dx} U_0 + \rho + M\nu. \quad (4.10)$$

Grazie alla (4.10),  $g$  raggiunge valori vicini a  $y$  dove  $(\sigma_1 - \Delta)(\sigma_2 + \Delta) < 0$  e  $\nu = 0$ . Il tempo necessario a raggiungere questo stato è facilmente ridotto incrementando  $M$ . Una volta che  $\nu = 0$ , la variazione di  $\sigma_1$  e  $\sigma_2$  è

$$\dot{\sigma}_1 = \dot{\sigma}_2 = \rho - \frac{df}{dx} U_0 \text{sign}(\sigma_1 \sigma_2), \quad (4.11)$$

e contemporaneamente vale

$$\left| \frac{df}{dx} \right| U_0 > \rho. \quad (4.12)$$

In questo stato ( $\nu = 0$ ), a dipendenza del segno di  $df/dx$ ,  $\sigma_1$  o  $\sigma_2$  tenderanno a zero, poiché in questo stato essi hanno segno opposto. Ogni cambiamento del segno di  $\sigma_1$  o  $\sigma_2$ , inoltre, cambierà il segno della rispettiva derivata, e questo genera nel sistema uno Sliding Mode. Idealmente lo switch di  $u$  avviene a frequenza infinita, e il valore di  $\sigma_1$  o  $\sigma_2$  è zero. Se  $\nu = 0$ , il riferimento  $g$  cresce monotonicamente con tasso  $\rho$ , e dunque  $y$  segue il riferimento in crescita

$$\sigma_1 = g - y \approx 0, \quad (4.13)$$

$$\sigma_2 = g - y + \delta \approx 0. \quad (4.14)$$

Quindi l'output  $y$  aumenta fino ad essere vicino al massimo, dove non regge più la (4.12).

In questo terzo stato, l'output  $y$  non cresce più monotonicamente con  $\rho$ , e le (4.13) e (4.14) non valgono più. In questo caso, però,  $\nu$  mantiene  $\sigma_1$  e  $\sigma_2$  nell'intervallo  $\sigma_1 < \Delta$  e  $\sigma_2 > -\Delta$ ; è dimostrato che l'input  $x$  si mantiene vicino al punto di massimo  $x^*$ .

In [18], è dimostrato che sotto queste condizioni  $\sigma_1$  o  $\sigma_2$  raggiungono  $+\Delta$  o  $-\Delta$  durante  $t_1$  e  $t_2$ , definiti da

$$t_1 = \frac{\Delta}{\rho - U_0 \frac{df}{dx}}, \quad (4.15)$$

$$t_2 = \frac{\Delta}{\rho + U_0 \frac{df}{dx}}. \quad (4.16)$$

Se assumiamo che  $\rho \gg U_0 |df/dx|$  in prossimità di  $x^*$ , allora il periodo temporale è descritto da

$$t_{12} = t_1 + t_2 = \frac{2\Delta}{\rho}. \quad (4.17)$$

Se si approssima la (4.11) da

$$\dot{\sigma}_1 = \dot{\sigma}_2 = \rho, \quad (4.18)$$

allora  $\sigma_1$  (o  $\sigma_2$ ) raggiungerà  $+\Delta$  (o  $-\Delta$ ) al tempo  $t_{12}$ , quando  $\nu$  sarà nuovamente differente da zero e  $\sigma_1$  ritornerà a  $+\Delta$  in un tempo prossimo a zero, dato che  $M$  è molto grande.

Quindi, se si considera che, quando  $x$  è prossimo a  $x^*$  esiste uno Sliding Mode la cui frequenza è

$$f_{sm} = \frac{\rho}{2\Delta}, \quad (4.19)$$

e il massimo errore tra  $x$  e  $x^*$  è

$$e_{sm} = \frac{\Delta}{2\rho} U_0. \quad (4.20)$$

I parametri necessari per il set up dell' algoritmo ESC Sliding Mode sono  $\Delta, \delta, M, U_0$  e  $\rho$ , secondo le seguenti relazioni:

$$\Delta < \delta, \quad (4.21)$$

$$M \gg U_0 \quad (4.22)$$

$$M \gg \rho, \quad (4.23)$$

in altre parole,  $U_0$  deve essere scelto sufficientemente grande da garantire la (4.12) e sufficientemente piccolo da garantire la (4.14).



## Capitolo 5

# Implementazione dell'algoritmo

### 5.1 *Computer-Aided Control Systems Design - CACSD*

La modellistica matematica mira a descrivere, in termini matematici, i molteplici aspetti del mondo reale insieme alla loro dinamica; essa è ampiamente consolidata in svariati settori, come ad esempio quello industriale, ambientale, e biologico. I modelli matematici offrono nuove possibilità per dominare la crescente complessità delle tecnologie e permettono di accelerare i cicli di innovazione, esplorando rapidamente nuove soluzioni.

Una ragione del crescente successo della modellistica matematica è da ascrivere al repentino sviluppo del calcolo scientifico, che consente di tradurre un modello matematico in algoritmi che possono essere risolti da calcolatori elettronici di potenza sempre più elevata.

Un problema di controllo si può definire come la necessità di ottenere un comportamento desiderato all'uscita agendo sull'ingresso. Se l'ingresso viene elaborato da un dispositivo automatico si parla di controllo automatico. L'idea centrale e fondamentale per la soluzione dei problemi di controllo è la retroazione, o *feedback*.

Nel controllo a retroazione o a catena chiusa, il valore misurato all'uscita del sistema viene comparato con il riferimento o *set point*. La differenza tra *set point* e *output* viene utilizzata per calcolare il valore da dare in *input* al sistema. L'utilizzo di un controllo a catena chiusa permette di mitigare sia gli effetti di disturbi esterni, tramite la reiezione degli stessi, sia quelli dovuti alla discrepanza tra modello e realtà determinando, quindi, la robustezza del controllo.

Per quanto concerne la progettazione, la taratura e la verifica delle prestazioni dell'algoritmo *ESC Sliding Mode*, data la delicatezza e la difficoltà

di operare fin da subito sul *Pulse Duplicator* (in termini di allestimento degli esperimenti, di esecuzione delle prove, etc.), si è preferito preliminarmente giovare delle opportunità offerte da un ambiente di simulazione Matlab<sup>1</sup>/Simulink<sup>2</sup> relativo al *Pulse Duplicator* in assegnate condizioni qualitative nominali. Infatti, è una pratica comune, nella progettazione di algoritmi controllo fare uso di strumenti di *Computer-Aided Control Systems Design* (CACSD). Questi strumenti consentono di simulare le principali caratteristiche del sistema per una prima valutazione della strategia di controllo. Una volta verificata in simulazione l'efficacia dell'algoritmo *ESC Sliding Mode* esso è stato successivamente implementato e testato nel *Pulse Duplicator* reale attraverso l'utilizzo di hardware (Arduino, Ponte H e motoriduttore) e software dedicato (LabVIEW<sup>3</sup> e NI LabVIEW Interface per il Toolkit Arduino).

## 5.2 Caratterizzazione preliminare

Si vuole ora caratterizzare il sistema per poter applicare in simulazione ed implementare la teoria dell'algoritmo Extremum Seeking Control Sliding Mode per il controllo automatico della resistenza periferica<sup>4</sup> presente nel *pulse duplicator* (cfr sezione 1.3.1).

Preliminarmente, si sono acquisiti i dati relativi alla pressione a valle della valvola aortica (o pressione aortica) secondo diverse condizioni operative di *cardiac output*<sup>5</sup> e di *heart rate*<sup>6</sup> riportate nella croce sperimentale di figura 5.1. I parametri della croce sperimentali non sono valori arbitrari, ma sono condizioni fisiologiche verosimili che permettono di conferire attendibilità alla simulazione; in direzione verticale verso il basso sono presenti gli esperimenti con frequenza cardiaca fissata e *cardiac output* decrescente, mentre muovendosi in direzione orizzontale, da sinistra a destra, sono presenti gli esperimenti a *cardiac output* fissato e a frequenza cardiaca crescente.

L'acquisizione dei dati si è basata sul programma *LabVIEW* che governa il *pulse duplicator*: per ogni ciclo cardiaco acquisito si è registrato, oltre alla

---

<sup>1</sup>Matlab è un ambiente per il calcolo numerico con un proprio linguaggio di programmazione che permette di manipolare agevolmente dati, matrici, visualizzare grafici, implementare funzioni e algoritmi. Esso è strettamente interfacciato con Simulink

<sup>2</sup>Simulink consente la modellizzazione, l'analisi e la simulazione di sistemi dinamici, attraverso un'interfaccia grafica che permette la realizzazione di schemi a blocchi

<sup>3</sup>LabVIEW è il software per la progettazione di sistemi pensato appositamente per lo sviluppo di applicazioni di test, misura e controllo con accesso rapido all'hardware e ai risultati

<sup>4</sup>D'ora in avanti ci si riferisce, per semplicità, alla resistenza periferica con il termine equivalente di valvola.

<sup>5</sup>Volume di sangue eiettato dal cuore nell'unità di tempo, rappresenta la portata cardiaca.

<sup>6</sup>Numero di cicli cardiaci nell'unità di tempo.

pressione aortica, il livello di apertura della resistenza periferica<sup>7</sup>, così da poter associare il valore di resistenza e la pressione aortica in una mappa. Ognuno dei 5 esperimenti ha visto acqua come fluido presente nel *pulse duplicator*, come valvola aortica la *On-X bileaflet mechanical heart valve* di diametro 23mm, e in posizione mitralica la *Solin Allcarbon bileaflet mechanical heart valve* di diametro 27mm.

Nel seguito, si mostrano nel dettaglio esclusivamente i risultati relativi al

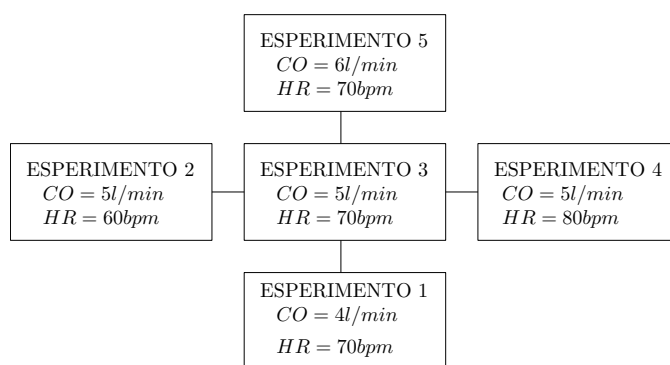


Figura 5.1: Croce sperimentale relativa ai 5 esperimenti riprodotti al *pulse duplicator*

primo esperimento, in quanto gli andamenti delle curve sperimentali si ripetono qualitativamente, e sono riportati nell'appendice A.

Una volta conclusa la raccolta dati, tramite il *software* Matlab, si può graficare l'andamento della pressione aortica nelle condizioni operative dell'esperimento.

In figura 5.2 si vede qualitativamente come la pressione media in ogni ciclo cardiaco vada via via aumentando; questo comportamento è atteso ed è dovuto alle modalità in cui si è svolto l'esperimento: dopo ogni ciclo cardiaco registrato, si è aumentato il valore di resistenza periferica localizzata creando, così, un restringimento sempre maggiore che, di conseguenza, ha aumentato la pressione aortica media.

Per ogni ciclo di ogni esperimento si è calcolata la media aritmetica dei valori di pressione aortica. Questi valori, al variare del grado di apertura della valvola, sono stati approssimati tramite un polinomio di grado 4.

Si considerano  $n$  punti  $(\vartheta_i, y_i)$  con i  $\vartheta_i$  distinti e ordinati che rappresentano il grado di apertura della resistenza periferica, da totalmente aperta a totalmente chiusa, e gli  $y_i$  valori di pressione aortica media in quel punto. Si assume che il segnale approssimante incognito  $u(\vartheta)$  abbia una descrizione

<sup>7</sup>L'apertura della valvola è governata da un programma *LabVIEW* a sè che mappa l'intero range di chiusura tramite impulsi di tempo finito in ingresso al motore.

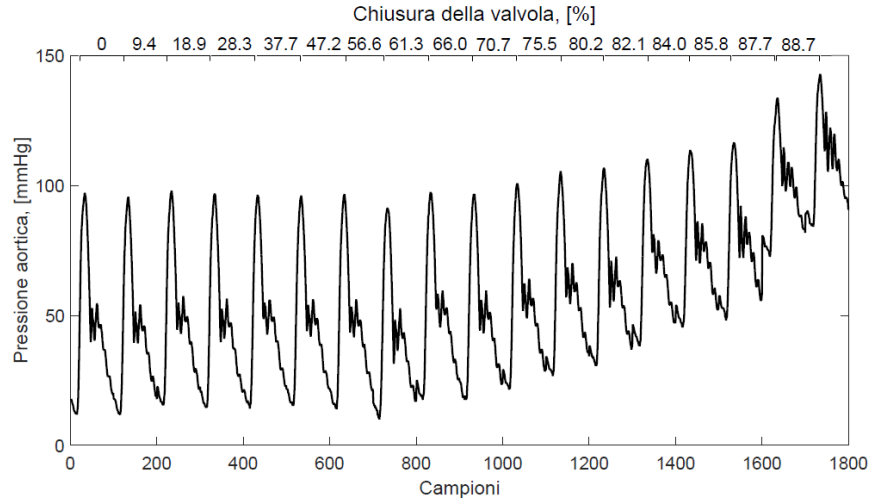


Figura 5.2: Campioni relativi alla pressione aortica in condizioni di cardiac output 4l/min e frequenza cardiaca di 70bpm

parametrica  $u(\vartheta) = u(\vartheta, a)$ , e che in particolare sia un polinomio di grado  $m - 1$  con  $m \ll n$ . Allora:

$$u(\vartheta) = u(\vartheta, a) = a_1\vartheta^{m-1} + a_2\vartheta^{m-2} + \dots + a_{m-1}\vartheta + a_m. \quad (5.1)$$

Tramite gli algoritmi *polival* e *polifit* in MATLAB, si sono stimati i parametri  $a_i$  e il grado del polinomio che meglio approssimassero i punti sperimentali. Il risultato dello *smoothing* è riportato in figura 5.3.

Per poter applicare l'algoritmo di controllo è necessario introdurre una funzione di costo quadratica che lega la pressione aortica media al grado di apertura della resistenza periferica. Una possibile funzione costo è riportata nell'equazione (5.2), dove 100mmHg rappresenta la pressione aortica media desiderata nell'arco del battito cardiaco, come prescritto dalla normativa BS EN ISO5840 1:2015:

$$J(\vartheta) = (u(\vartheta, a) - 100)^2. \quad (5.2)$$

L'andamento di questa funzione costo è riportato in figura 5.4. L'obiettivo della simulazione è minimizzare la funzione (5.2) in modo da individuare automaticamente quale sia il valore di apertura della resistenza che garantisce una pressione aortica media pari a quella desiderata.

Si considerano ora tutti e 5 gli esperimenti della croce sperimentale. Si riproducono le figure equivalenti alle 5.3 e 5.4, sovrapponendo le curve relative ai diversi esperimenti.

Si nota come ognuna delle condizioni sperimentali, riportate nella figura 5.5 abbia una specifica curva approssimante e uno specifico valore di apertura

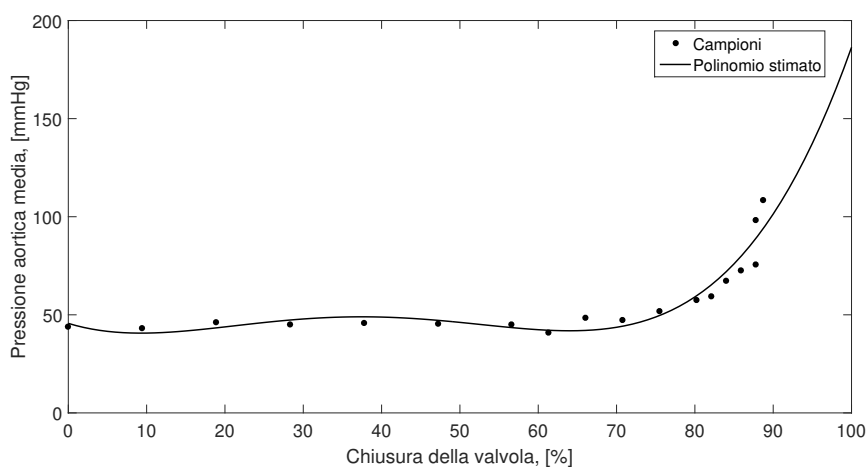


Figura 5.3: Approssimazione polinomiale dei dati tramite un polinomio di grado 4, per prove sperimentali con cardiac output 4l/min e frequenza cardiaca di 70bpm

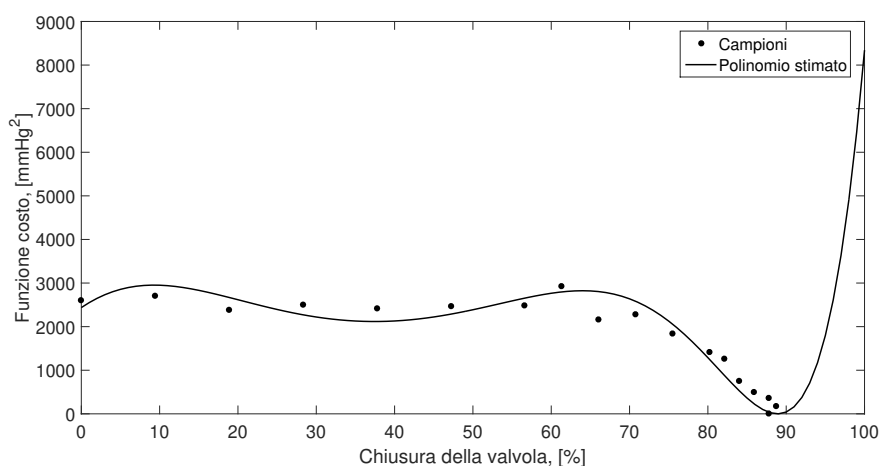


Figura 5.4: Funzione costo della pressione aortica media per diversi livelli di apertura della resistenza, per prove sperimentali con cardiac output 4l/min e frequenza cardiaca di 70bpm

della resistenza che garantisca la pressione aortica media nel ciclo pari a 95mmHg, ovvero l'obiettivo del controllo.

I polinomi approssimanti (figura 5.5) sono rappresentati in una griglia virtuale<sup>8</sup> di valori di apertura della resistenza per permettere una visualizza-

<sup>8</sup>Si decide di disaccoppiare la griglia temporale di campionamento di dimensione  $n$ , da

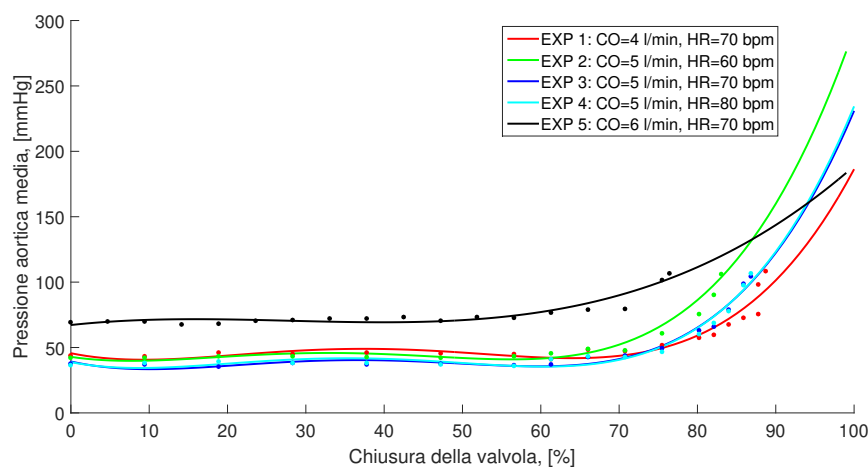


Figura 5.5: Stima tramite approssimazione con polinomi di grado 4 della pressione aortica media per diversi livelli di chiusura della resistenza nei 5 esperimenti

zione più chiara degli andamenti stimati delle curve. In ogni caso, le griglie virtuali utilizzate assumono valori fino al 100% di chiusura, supplendo tramite simulazione alla limitatezza e alla fragilità dei componenti fisici, che non permetterebbero un tale livello di chiusura. Soprattutto per valori di restringimento più spinti, le acquisizioni sperimentali della pressione sarebbero state estremamente rischiose nei riguardi dell'integrità degli elementi più delicati del circuito, come ad esempio l'arco aortico in silicone. Questo spiega il fatto che i punti sperimentalmente acquisiti non siano simmetricamente distribuiti rispetto al minimo della funzione costo (figura 5.13).

Le figure 5.6 e 5.7, mostrano le curve sovrapposte delle pressioni aortiche medie degli esperimenti con valori di cardiac output e frequenza cardiaca rispettivamente in comune. Dall'analisi di questi grafici si nota come, a cardiac output imposto fissato a 5l/min (figura 5.6), la variazione di frequenza cardiaca non alteri sensibilmente la pressione aortica media per un fissato grado di apertura della valvola, mentre la variazione di cardiac output a frequenza cardiaca fissata (figura 5.7) risulta più rilevante. Questo comportamento è confermato anche nei grafici dell'andamento delle pressioni minime e massime di ogni ciclo cardiaco per i vari livelli di chiusura della valvola per ogni esperimento, riportate nelle figure 5.8, 5.9, 5.10 e 5.11. Osservando figura 5.7, si vede come, a parità di frequenza cardiaca, nell'esperimento con cardiac output maggiore (6l/min), la pressione aortica media ha una crescita rapida già a partire dal 50% di chiusura della valvola.

Il grafico tridimensionale di figura 5.12 riporta le condizioni ottime di apertura della valvola per ogni esperimento, ovvero le condizioni di apertura una griglia virtuale di dimensione  $N \gg n$  su cui si vuole approssimare il segnale.

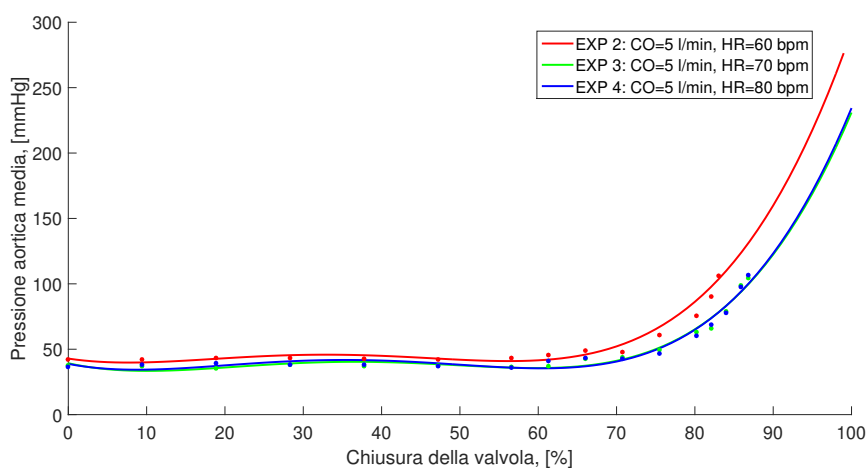


Figura 5.6: Stima tramite approssimazione con polinomi di grado 4 della pressione aortica media per diversi livelli di chiusura della resistenza negli esperimenti a cardiac output di 5l/min

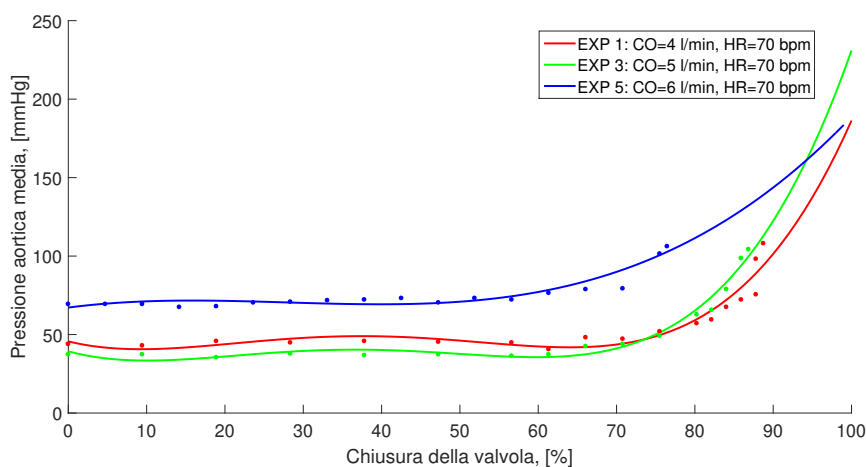


Figura 5.7: Stima tramite approssimazione con polinomi di grado 4 della pressione aortica media per diversi livelli di chiusura della resistenza negli esperimenti a frequenza cardiaca di 70bpm

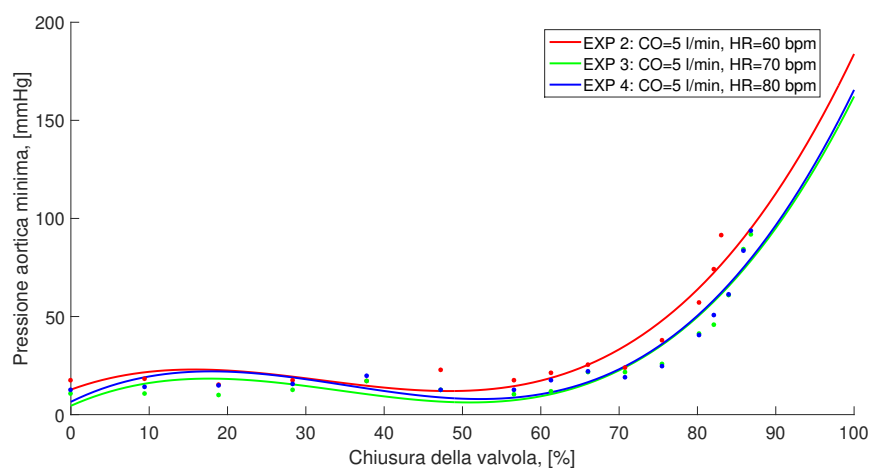


Figura 5.8: Stima tramite approssimazione con polinomi di grado 3 della pressione aortica minima per diversi livelli di chiusura della resistenza negli esperimenti a cardiac output di 5l/min

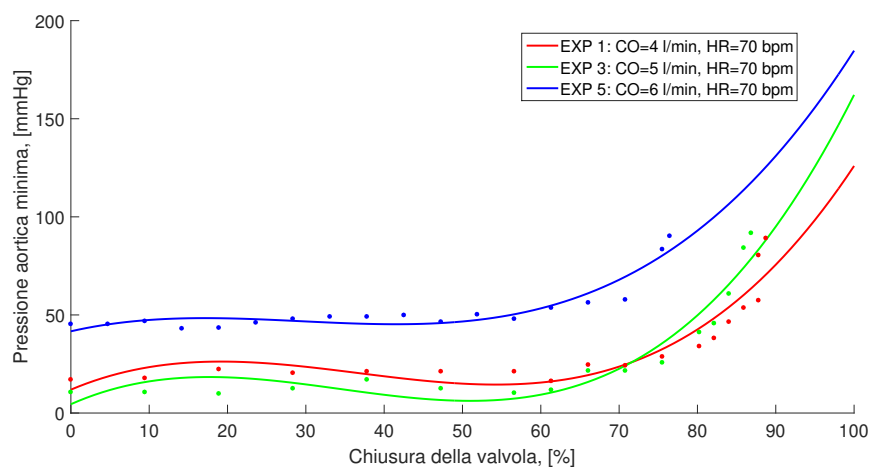


Figura 5.9: Stima tramite approssimazione con polinomi di grado 3 della pressione aortica minima per diversi livelli di chiusura della resistenza negli esperimenti a frequenza cardiaca di 70bpm



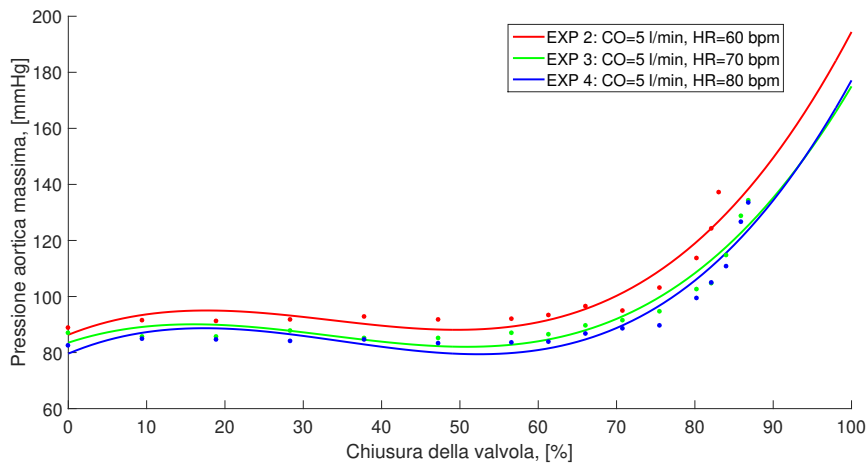


Figura 5.10: Stima tramite approssimazione con polinomi di grado 3 della pressione aortica massima per diversi livelli di chiusura della resistenza negli esperimenti a cardiac output di 5l/min

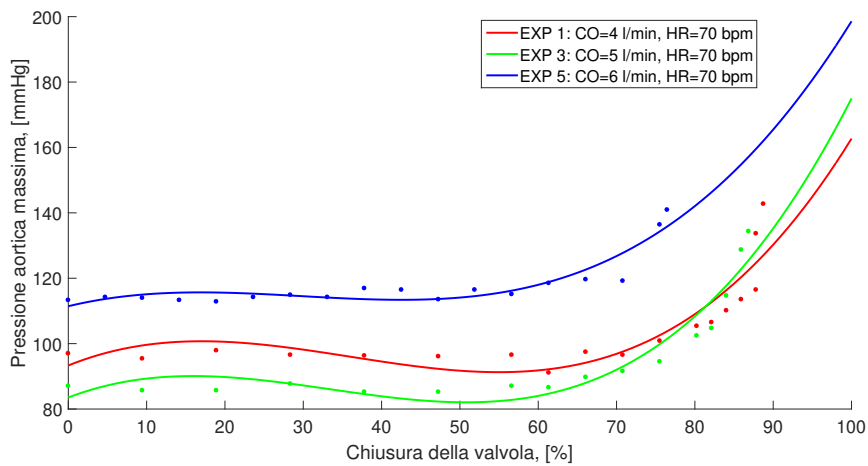


Figura 5.11: Stima tramite approssimazione con polinomi di grado 3 della pressione aortica massima per diversi livelli di chiusura della resistenza negli esperimenti a frequenza cardiaca di 70bpm

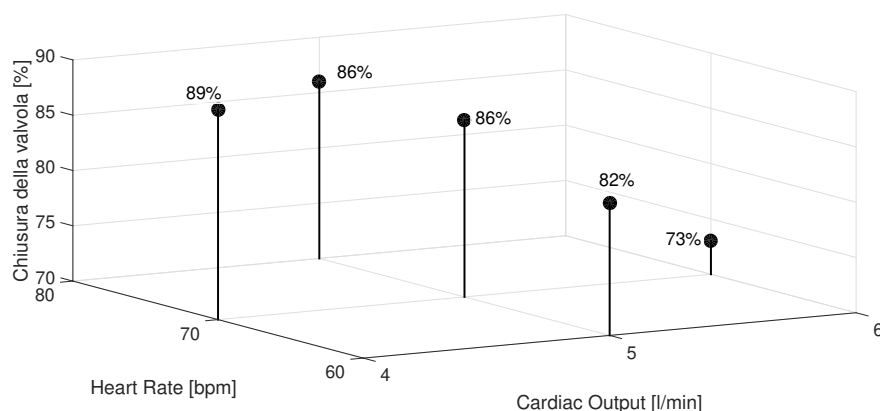


Figura 5.12: Distribuzione delle aperture ottime della valvola per i 5 esperimenti, rispetto a cardiac output e frequenza cardiaca

della resistenza che minimizzano la funzione costo del relativo esperimento; si nota come il ragionamento appena fatto guardando le curve di pressione aortica media, minima e massima, trovi riscontro in questo grafico: gli esperimenti a valore di *cardiac output* comune (5 l/min), presentano un valore ottimo di apertura della valvola simile tra loro con variabilità del 4%, indipendentemente dal valore di frequenza cardiaca, mentre a parità di battiti cardiaci, la variabilità del valore ottimo è del 16%.

Osservando figura 5.13, si nota come le funzioni costo relative ai diversi esperimenti abbiano un grado di curvatura differente, e il valore minimo si trovi a un valore differente di apertura della resistenza. Sarà compito dell'algoritmo di controllo individuare automaticamente tali valori. Per quanto spiegato nella sezione 3.3, questa diversità di curvatura influirà inevitabilmente sulle prestazioni dell'algoritmo, in quanto esso convergerà più rapidamente in caso di curvatura importante, mentre convergerà molto più lentamente per curvature più lievi.

In ognuno dei 5 esperimenti si è anche calcolato il cardiac output effettivo, per confrontarlo con il valore costante imposto in ogni esperimento. Per ottenere il valore sperimentale in ogni ciclo cardiaco, si è calcolato l'integrale del flusso positivo (AFV- Aortic Forward Volume) e l'integrale del flusso negativo o retrogrado (ARV- Aortic Reversal Volume), un esempio dell'operazione è mostrato in figura 5.14. Indicando la frequenza cardiaca come Heart Rate (HR), il cardiac output si individua tramite la (5.3).

$$CO = HR \times AFV - HR \times |ARV| \quad (5.3)$$

In figura 5.15, si vede come, in media, il valore di cardiac output si mantenga vicino al valore nominalmente imposto in ogni prova, durante l'intero processo di regolazione della valvola; al variare della frequenza cardiaca e

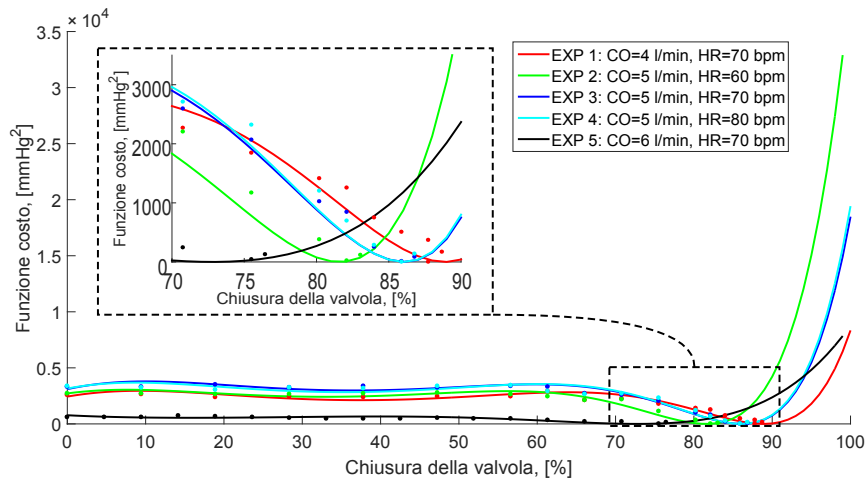


Figura 5.13: Funzione costo della pressione aortica media per diversi livelli di apertura della resistenza nei 5 esperimenti

della chiusura della valvola, il sistema riesce quindi a mantenere il cardiac output pressoché costante.

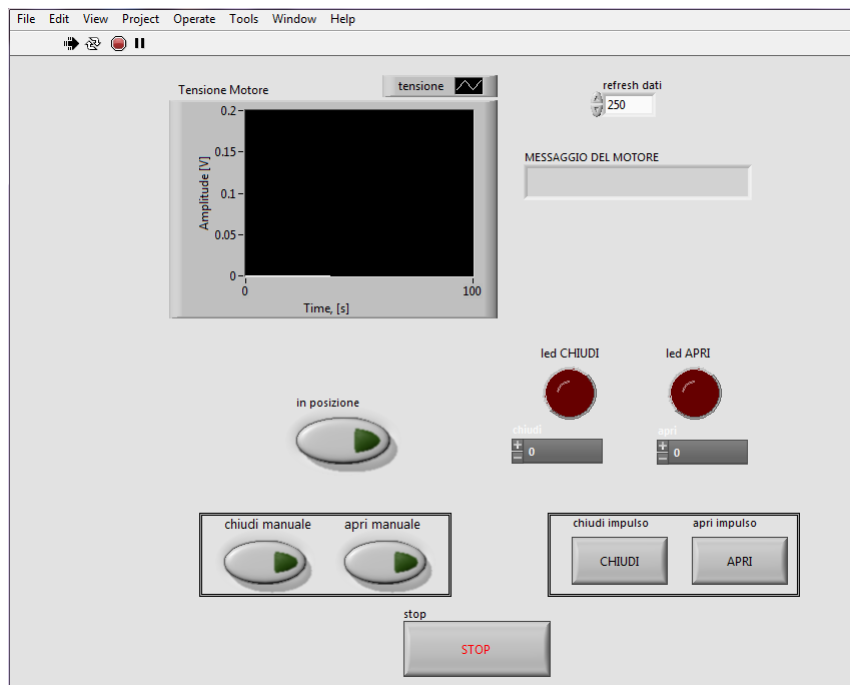


Figura 5.16: Front Panel del programma LabVIEW per gestire arbitrariamente la valvola

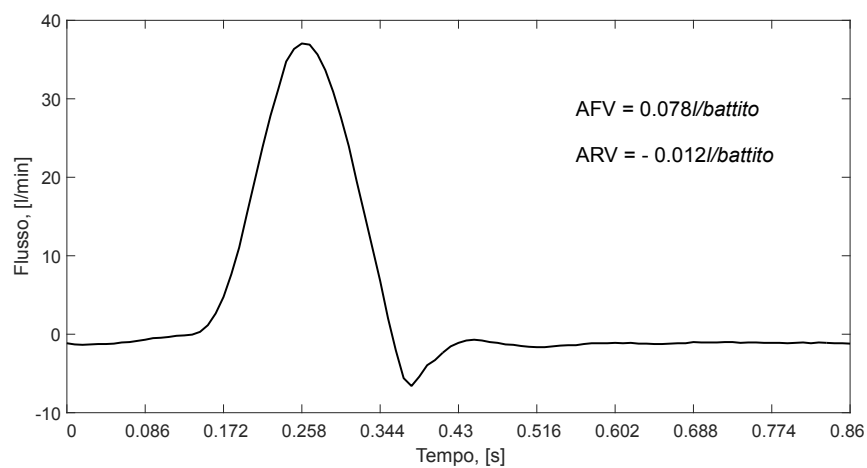


Figura 5.14: Andamento del flusso corrispondente a un battito cardiaco in un esperimento con cardiac output 5l/min e frequenza cardiaca di 70bpm

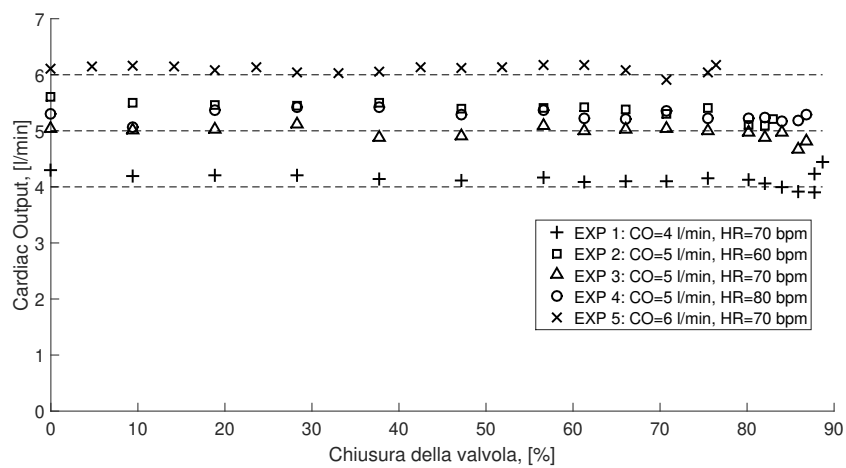


Figura 5.15: Cardiac Output calcolato per diversi livelli di apertura della resistenza nei 5 esperimenti

Per acquisire i dati relativi alle pressioni e per posizionare la valvola ai diversi gradi di apertura, si è utilizzato un programma LabVIEW, mostrato in figura 5.16. Nel Front Panel si vede il grafico riguardante il voltaggio del motore legato all'operazione di sensing (*tensione motore*), i pulsanti per aprire o chiudere la valvola per un tempo arbitrario (*apri manuale e chiudi manuale*), quelli per il movimento dell'1% della corsa totale (*apri impulso e chiudi impulso*), il pulsante *in posizione* che permette alla saracinesca di portarsi a metà corsa, partendo da completamente aperta, il *messaggio del motore* che indica se il motore è arrivato a fine corsa o se la valvola si trova a metà apertura, il *refresh dati* che indica il periodo, in ms, di aggiornamento della comunicazione con il motore, e i led indicativi del verso di rotazione del motore, se in apertura o in chiusura. Il diagramma a blocchi relativo al Front Panel di figura 5.16 è mostrato in figura 5.17.

### 5.3 Applicazione della tecnica in simulazione

Dopo aver caratterizzato il sistema, si può procedere nell'applicazione in simulazione dell'algoritmo Extremum Seeking Control Sliding Mode in ambiente Matlab/Simulink. Il diagramma Simulink usato nella simulazione è riportato in figura 5.18. Esso risulta leggermente modificato rispetto allo schema dell'algoritmo presentato nella sezione 4, in figura 4.1, al fine di facilitare tuning e implementazione dello stesso. In particolare:

- La funzione di switching permette di associare la pendenza dell'andamento del riferimento  $r$  nell'iterazione successiva, all'entità dello scostamento tra la funzione costo e il riferimento stesso.  
In particolare se il riferimento risulta minore del valore del costo più di un certo valore di tolleranza ( $\delta = 8mmHg^2$ ), la pendenza di  $r$  sarà elevata (quattro volte il valore di soglia diviso per il tempo di campionamento) per permettere al riferimento di crescere rapidamente e riportarsi maggiore della funzione costo, altrimenti il riferimento si aggiornerà con pendenza minore arbitraria.
- In uscita alla funzione di switching, la velocità del riferimento viene integrata per permettere il calcolo di  $\epsilon$ , definito come  $\epsilon = j - r$ .
- Il segno di  $\epsilon$  viene utilizzato per decidere il verso di rotazione del motore, e quindi l'apertura o la chiusura della valvola, moltiplicandolo per  $U_0$ , variazione dell'apertura della valvola nell'unità di tempo.
- La mappa apertura/pressione sfrutta le curve individuate nel processo di caratterizzazione, descritto nella sezione 5.2, per simulare il valore di pressione aortica media e massima in ogni battito, corrispondenti al grado di apertura della valvola  $U$ .

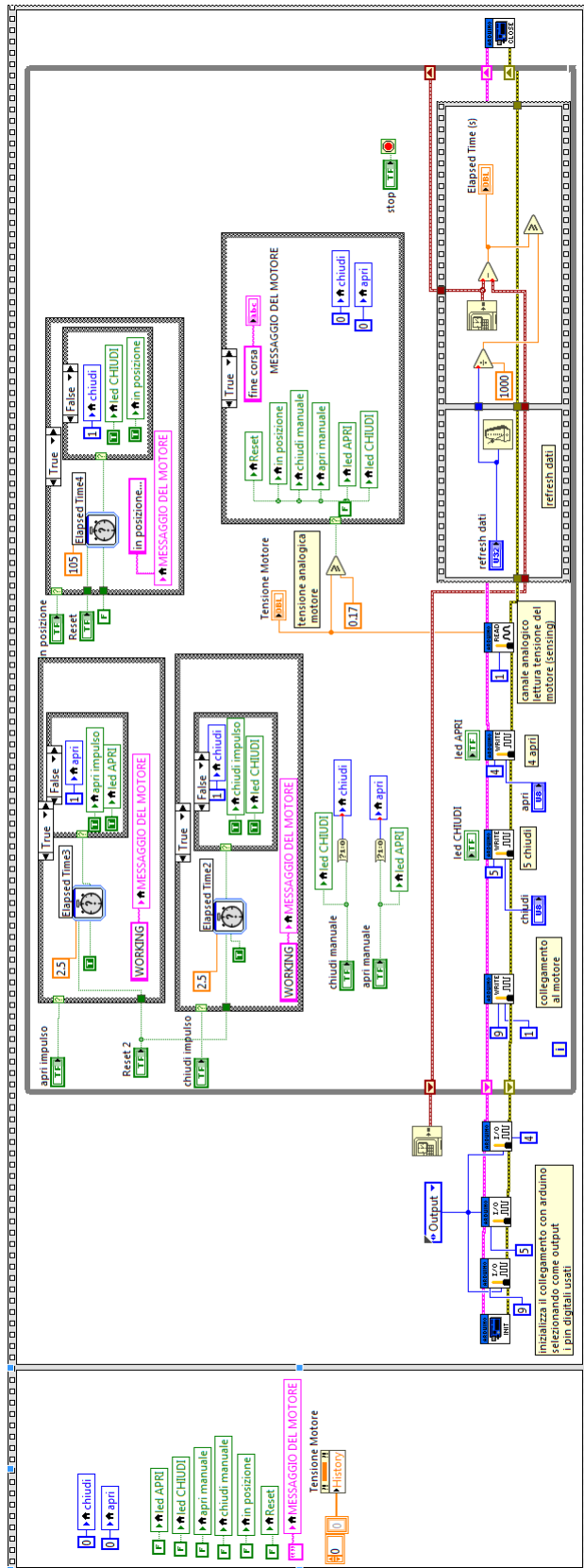


Figura 5.17: Block Diagram del programma LabVIEW per gestire arbitrariamente la valvola

- Grazie ai valori di pressione aortica si può calcolare la funzione costo, ovvero la discrepanza quadratica tra il valore di pressione media e il valore obiettivo (100mmHg, secondo normativa BS EN ISO5840 1:2015).

Si applica ora l'algoritmo al primo esperimento della croce sperimentale di figura 5.1, ovvero in condizioni di 70 pulsazioni al minuto con cardiac output di 4 litri al minuto, gli andamenti delle curve relative alle altre condizioni operative si ripetono qualitativamente, e sono riportati nell'appendice B. Gli andamenti del riferimento  $r$  e della funzione costo  $j$  sono riportati in figura 5.19. L'incrocio delle curve di riferimento e funzione costo, causa, all'iterazione seguente dell'algoritmo, il cambio nel segno di  $\epsilon$ , e quindi la variazione del verso di rotazione del motore. Il periodo di simulazione è stato scelto di 50ms per farlo coincidere con il periodo necessario nell'implementazione fisica, ovvero al tempo necessario ad Arduino per elaborare le informazioni provenienti dal programma LabVIEW e inviarle al motoriduttore. Questo comportamento è evidenziato nel particolare di figura 5.20.

In figura 5.21, si mostra l'andamento del riferimento e della funzione di costo a cui è stato sottratto il valore di tolleranza  $\delta = 8mmHg^2$ , tale da rendere evidente che quando le due curve si incrociano, il riferimento aumenta con pendenza elevata per essere nuovamente maggiore della funzione costo.

L'andamento della pressione aortica media durante la simulazione è riportato in figura 5.22. Si nota come il valore di pressione, a regime, oscilla attorno al valore che minimizza la funzione costo, ovvero i 100mmHg imposti dalla normativa BS EN ISO5840 1:2015. Questa oscillazione è causata dalla continua variazione del segno di  $\epsilon$ , e quindi del verso di rotazione del motoriduttore, derivante dal rapido e continuato incrocio delle curve di riferimento e costo. Essendo, infatti, la funzione costo da minimizzare di tipo quadratico, essa si avvicina al limite inferiore di  $0mmHg^2$ , per poi risalire, oscillando, così, attorno al suo valore minimo. Questa sua oscillazione causa inevitabilmente l'oscillazione della pressione aortica media attorno al suo valore obiettivo.

## 5.4 Implementazione della tecnica

Per poter implementare l'algoritmo Extremum Seeking Control Sliding Mode per controllare l'apertura della valvola di resistenza periferica presente nel pulse duplicator, occorre adattare l'algoritmo utilizzato in simulazione, presentato nella sezione 5.3, alle esigenze della strumentazione presente sul campo.

Al fine di una più facile visualizzazione e interazione tra il sistema fisico e l'algoritmo stesso, si è deciso di implementare il controllo in un ambiente

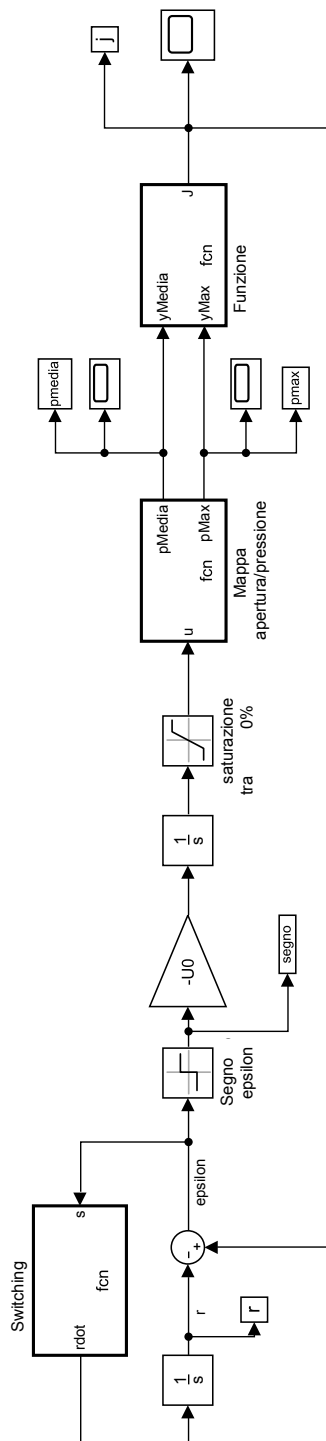


Figura 5.18: Diagramma Simulink dell'algoritmo Extremum Seeking Control Sliding Mode usato in simulazione



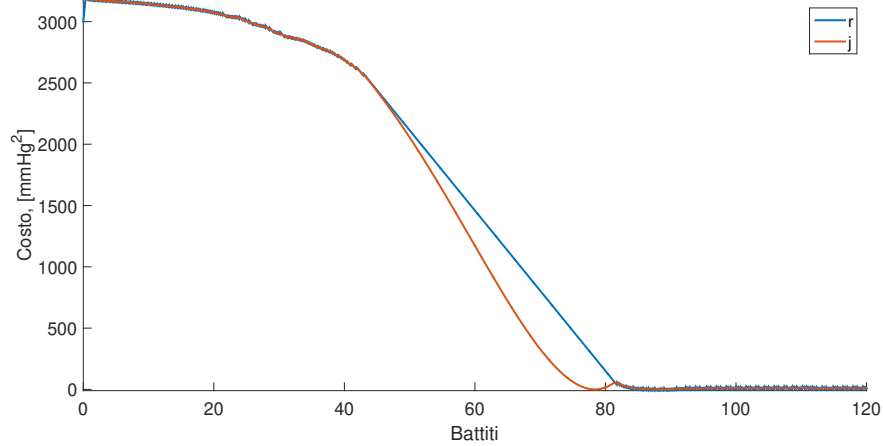


Figura 5.19: Andamento del riferimento  $r$  e della funzione costo  $j$  nel corso della simulazione a cardiac output di  $4\text{l}/\text{min}$  e frequenza cardiaca di  $70\text{bpm}$

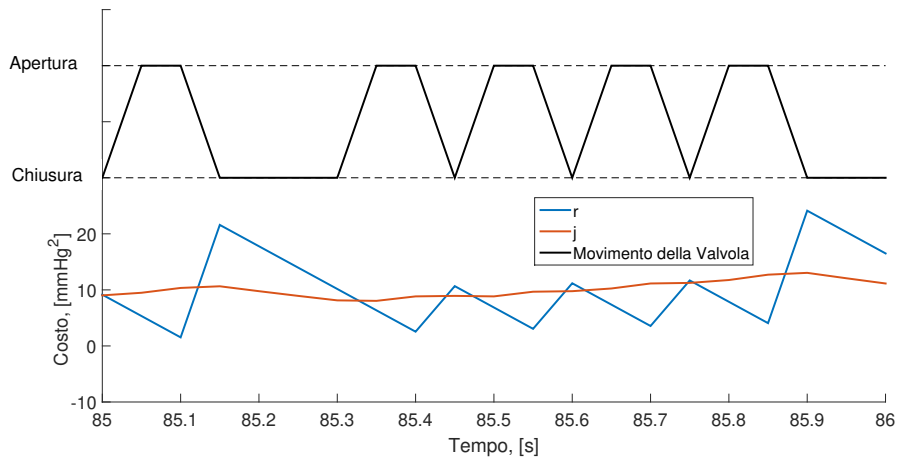


Figura 5.20: Variazione del verso di chiusura della valvola all'incrocio delle curve di riferimento e costo in simulazione a cardiac output di  $4\text{l}/\text{min}$  e frequenza cardiaca di  $70\text{bpm}$

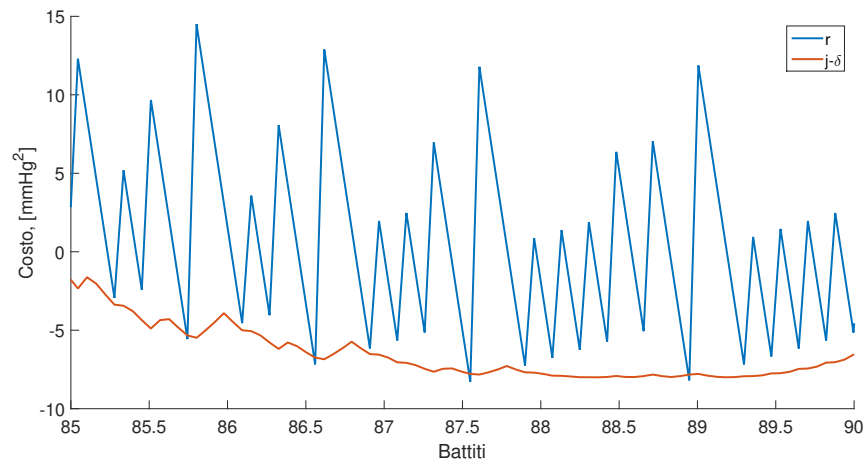


Figura 5.21: Aumento del riferimento quando esso incontra la funzione costo ridotta di un valore di tolleranza

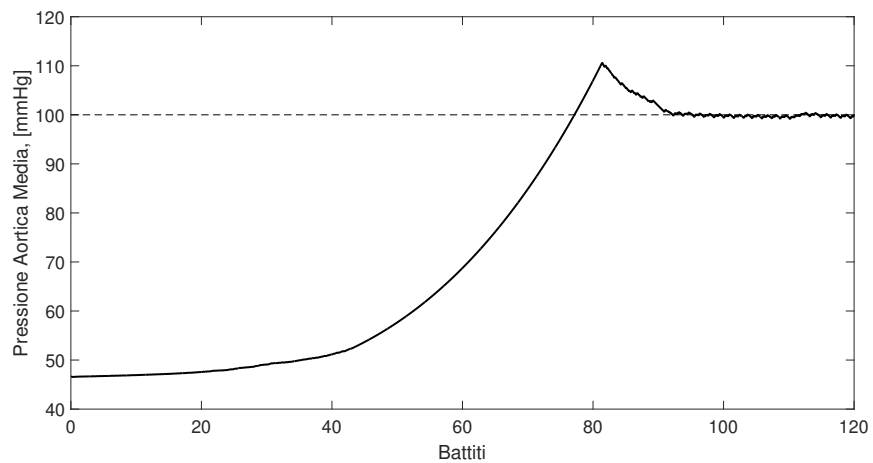


Figura 5.22: Andamento della pressione aortica media durante la simulazione a cardiac output di 4l/min e frequenza cardiaca di 70bpm

LabVIEW separato da quello utilizzato per governare il pulse duplicator. Si è reso necessario, quindi, rendere fruibili real time le misure di pressione aortica, in parallelo a quanto già reso disponibile dall'unità di controllo propria del sistema originario (*cfr* sezione 1.3.1). Tramite il nuovo sistema di condizionamento del segnale di pressione aortica, presentato nella sezione 2.1.7, si rende disponibile all'implementazione del controllo il segnale di interesse filtrato da eventuali disturbi.

Per calcolare il valore di pressione aortica media in ogni battito cardiaco, si è implementata una media mobile basata sul numero di campioni necessario, a seconda della frequenza cardiaca considerata e della frequenza di acquisizione dei campioni, per ottenere un battito cardiaco completo. Essendo una media eseguita su una finestra mobile, ogni iterazione dell'algoritmo di controllo, ovvero ogni 50ms, fornisce un valore di pressione media che rappresenta il valor medio eseguito su di un numero di campioni rappresentativi un battito completo. In questo modo, ad ogni iterazione, l'algoritmo è in grado di prendere una decisione nella movimentazione della valvola, basata su un valore pressorio con significato fisico.

Il pannello frontale del programma LabVIEW usato per implementare l'algoritmo di controllo è mostrato in figura 5.23. Esso si basa sull'algoritmo di movimentazione manuale della valvola periferica, introdotto per la caratterizzazione del sistema fluidodinamico e rappresentato in figura 5.16, e ne integra le funzionalità per permettere il controllo automatico della valvola stessa. In particolare si visualizza l'andamento della pressione aortica ad ogni iterazione del ciclo, l'andamento della media e del massimo sulla finestra mobile di campioni della lunghezza di un battito, e se ne mostrano i valori istantanei. Si mostrano, inoltre, le curve relative al riferimento, alla funzione costo e al segno di  $\epsilon$ , responsabile dell'apertura o chiusura della valvola, riassuntive dello stato di convergenza dell'algoritmo.

L'algoritmo di controllo è riassunto nel diagramma a blocchi mostrato in figura 5.24.

Si nota come la funzione di switching aggiorni la pendenza del riferimento a seconda della sua posizione rispetto alla funzione costo da minimizzare  $j$ , e di come questo valore venga integrato per consentire il calcolo di  $\epsilon$ . A seconda del segno di  $\epsilon$ , ovvero del segno della differenza tra il riferimento e la funzione costo, l'algoritmo decide se aprire o chiudere la valvola, inviando il segnale alle porte digitali di Arduino responsabili della comunicazione con il motoriduttore.

L'implementazione dell'algoritmo in condizioni di frequenza cardiaca a 70bpm e cardiac output di 4l/min fornisce gli andamenti di riferimento e funzione costo mostrati in figura 5.25. Osservando il particolare di figura 5.26, si nota che, come atteso, l'incrocio di riferimento e funzione costo provoca una variazione nel segno di  $\epsilon$  all'iterazione successiva dell'algoritmo, e quindi una

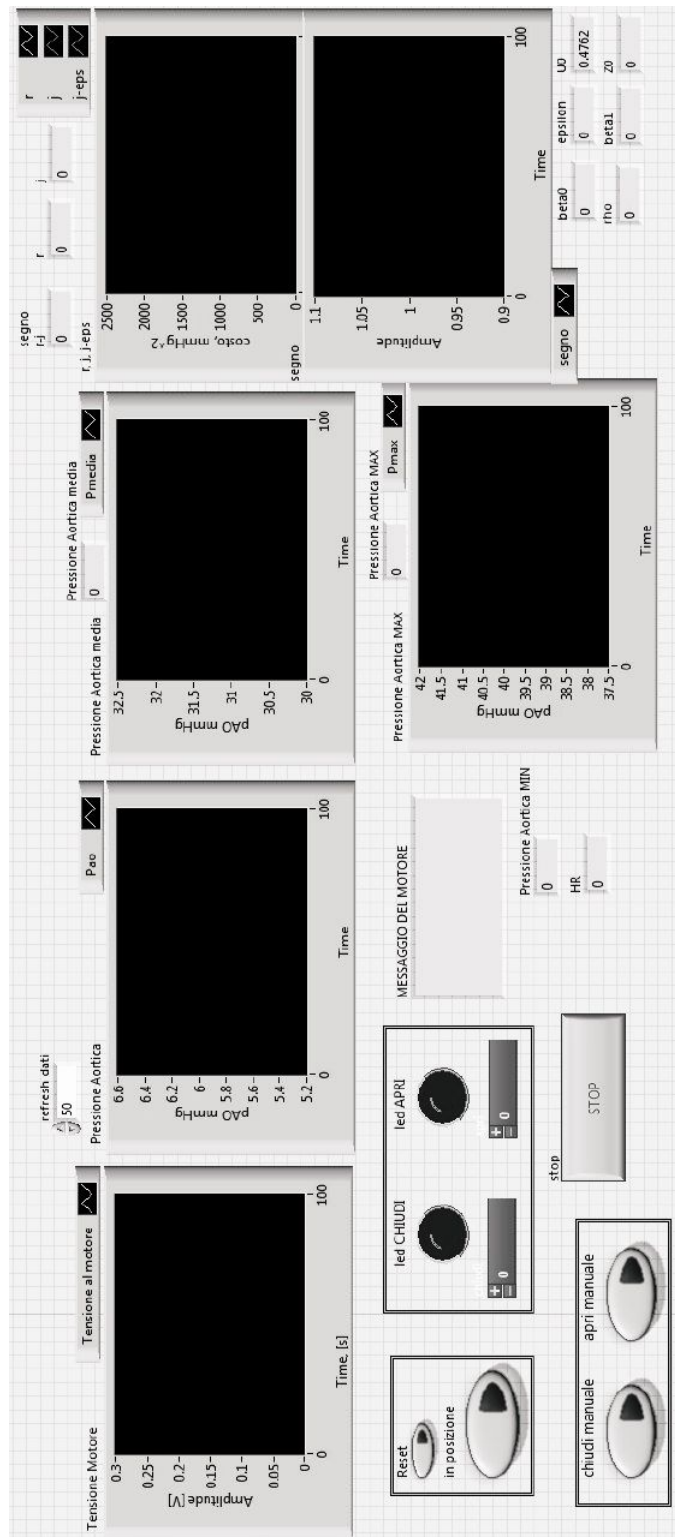


Figura 5.23: Pannello frontale del programma LabVIEW usato per implementare l'algoritmo di controllo Extremum Seeking Control Sliding Mode

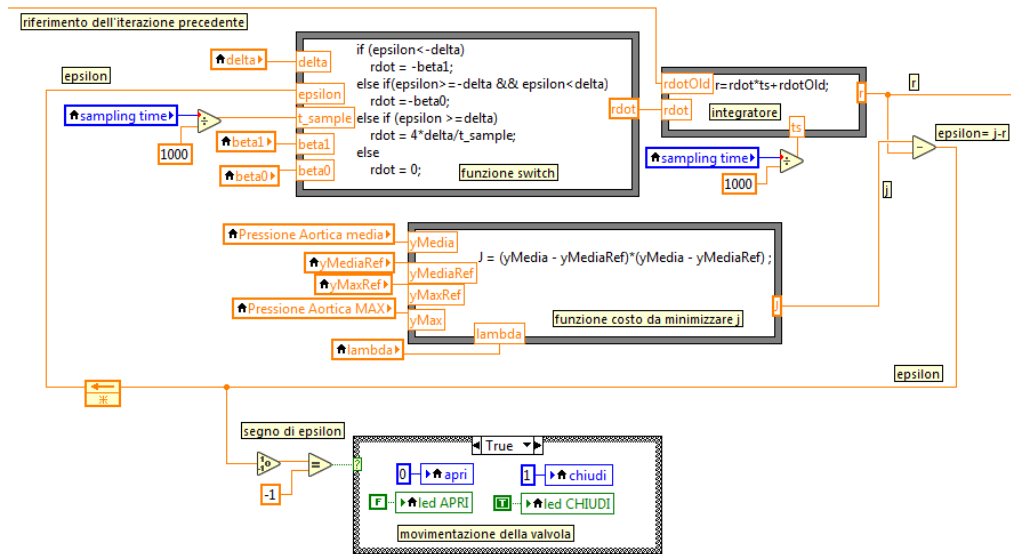


Figura 5.24: Porzione del diagramma a blocchi del programma LabVIEW responsabile dell'implementazione dell'algoritmo di controllo Extremum Seeking Control Sliding Mode

variazione nel verso di rotazione del motoriduttore.

Come descritto nella sezione 5.3, l'incontro del riferimento e della funzione costo ridotta del fattore di soglia  $\delta = 8mmHg^2$ , fa sì che il riferimento tenda ad aumentare per ritornare maggiore di  $j$ ; questo comportamento è evidenziato in figura 5.27.

Durante l'esperienza, è stata considerata una pressione aortica media di riferimento di 45mmHg, così da garantire un ampio margine di sicurezza al fine di preservare le componenti più fragili nel circuito meccanico/idraulico. L'algoritmo, quindi, vuole minimizzare la funzione di costo quadratica  $j = (\bar{P}_{AO} - 45)^2$ . L'andamento della pressione aortica media durante il controllo è riportato in figura 5.29; osservando la figura si nota come l'algoritmo, dopo essersi portato al valore di pressione aortica medio desiderato, tale da render minimo il costo, oscilla nell'intorno di quel valore ottimo aprendo e chiudendo la valvola, come mostrato in figura 5.28.

Dal confronto con la sezione sezioni 5.3, si nota come l'implementazione fisica del controllo su banco di prova sperimentale, dia origine a uno stato pressorio differente da quanto emerso dall'applicazione dello stesso algoritmo in simulazione. Durante la simulazione numerica la curva di pressione aortica media (figura 5.22), di cui è mostrato un particolare in figura 5.30, risulta stabilizzarsi attorno al valore pressorio obiettivo di 100mmHg, con una precisione maggiore dell'1%. Dall'analogo particolare di figura 5.29, ottenuto dall'implementazione del controllo su banco di prova, mostrato in

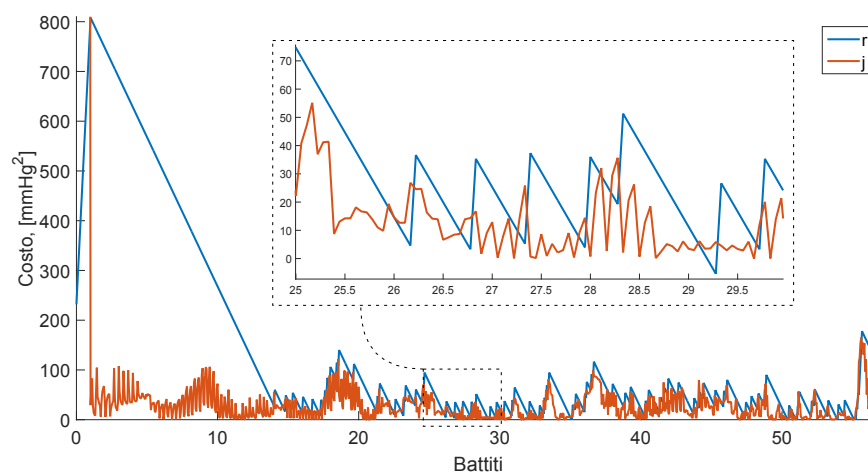


Figura 5.25: Andamento di riferimento e funzione costo durante l'implementazione dell'algoritmo di controllo in condizioni di 70 battiti al minuto e cardiac output di 4l/min

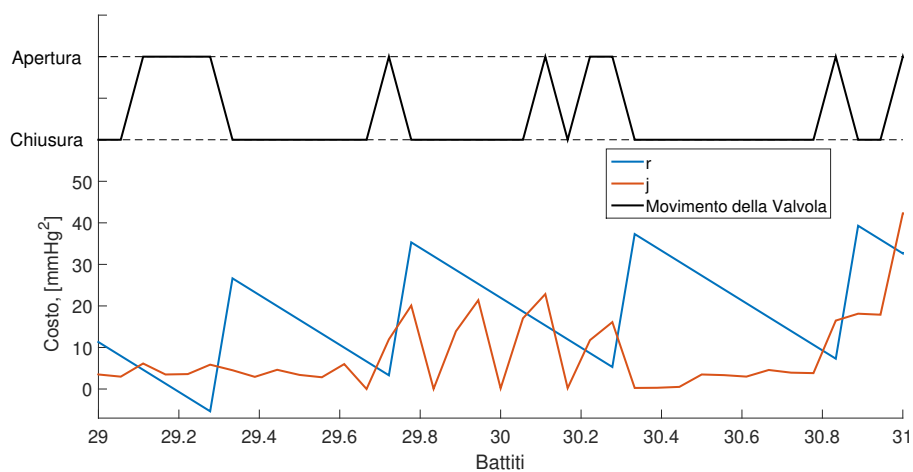


Figura 5.26: Variazione del segno di  $\epsilon$  rispetto all'andamento di  $r$  e  $j$  durante l'implementazione dell'algoritmo di controllo in condizioni di 70 battiti al minuto e cardiac output di 4l/min

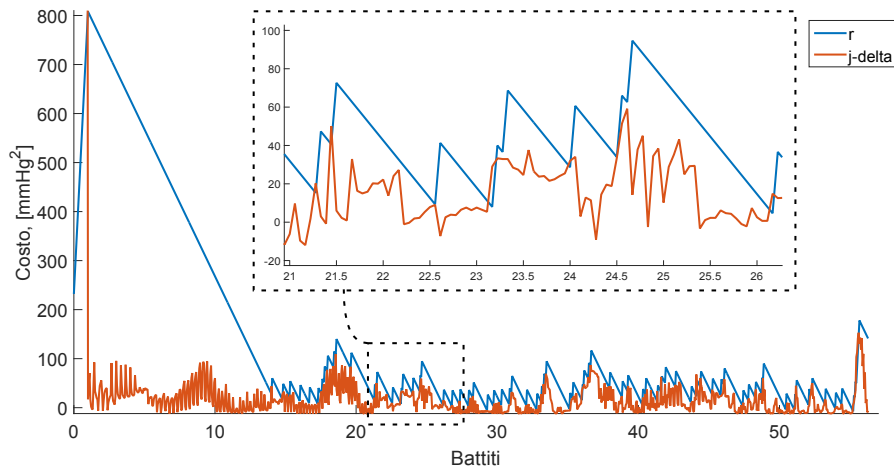


Figura 5.27: Andamento di riferimento e funzione costo ridotta del fattore di soglia  $\delta$  durante l'implementazione dell'algoritmo di controllo in condizioni di 70 battiti al minuto e cardiac output di 4l/min

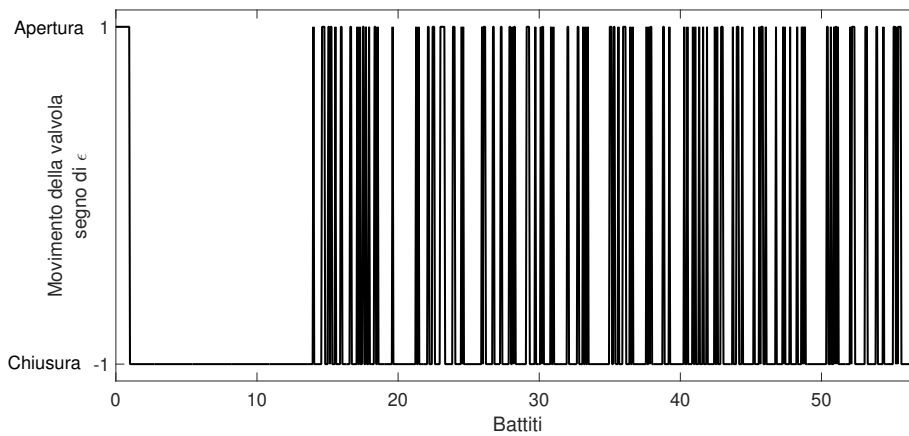


Figura 5.28: Variazione del segno di  $\epsilon$ , e quindi del movimento della valvola, durante l'implementazione dell'algoritmo di controllo in condizioni di 70 battiti al minuto e cardiac output di 4l/min

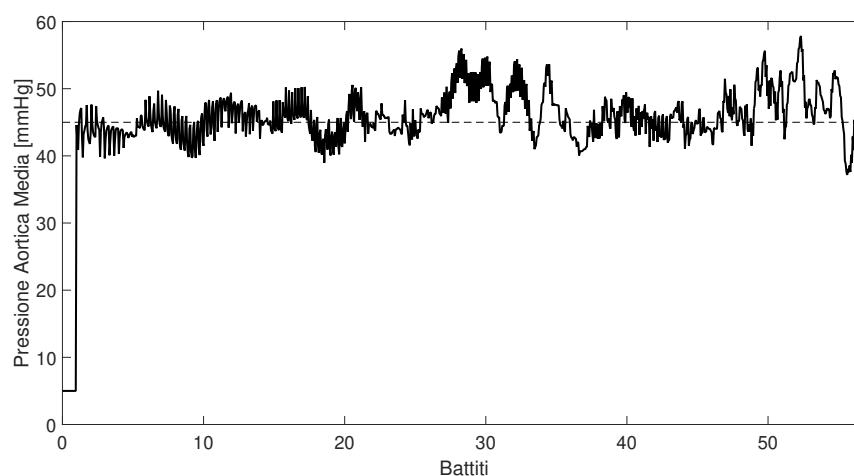


Figura 5.29: Andamento della pressione aortica media durante l'implementazione dell'algoritmo di controllo in condizioni di 70 battiti al minuto e cardiac output di 4l/min

figura 5.31, si vede che il controllo non mantiene la pressione media con altrettanta efficacia nell'intorno del valore obiettivo, in questo esperimento fissato a 45mmHg; dall'analisi della figura si vede, infatti, che l'oscillazione attorno al valore obiettivo si sviluppa in un intervallo pressorio pari al 33% del valore obiettivo stesso.

La causa di questa disparità può essere individuata nella natura dinamica del dispositivo stesso e nella presenza inevitabile di disturbi. Durante la simulazione, infatti, non si considera il fatto che al sistema risulta necessario del tempo per stabilizzarsi ad un certo valore pressorio, dopo ogni variazione del grado di apertura della valvola. Questo transitorio manifesta i suoi effetti nell'implementazione. Essendo i movimenti della valvola effettuati rapidamente rispetto al tempo necessario al sistema per terminare il transitorio ed entrare in condizione di regime, il controllo tende a oscillare maggiormente intorno al suo valore obiettivo. Una possibile soluzione potrebbe individuarsi nell'inserimento di pause nelle iterazioni dell'algoritmo tanto maggiori in durata quanto maggiore il valore di pressione aortica medio. Più lo stato pressorio del sistema è elevato, infatti, maggiore è il tempo necessario al sistema stesso per stabilizzarsi; ciò è legato al fatto che pressioni elevate sono ottenute tramite una chiusura della resistenza periferica dell'ordine del 90%, e a questi valori di chiusura, un piccolo movimento della valvola causa un'importante variazione dello stato pressorio.

In futuro, inoltre, si può particolareggiare il tuning dell'algoritmo rendendolo adatto al raggiungimento di pressioni aortiche fisiologiche, con attenzione all'integrità delle componenti più fragili del circuito.



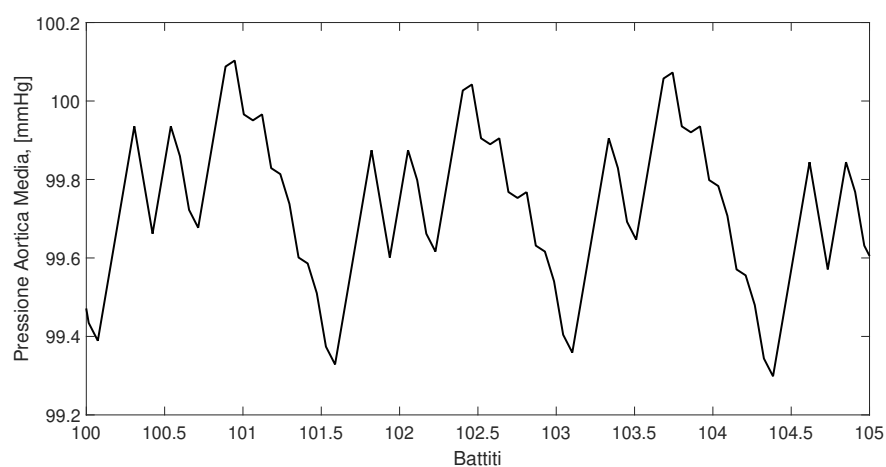


Figura 5.30: Particolare dell'andamento a regime della pressione aortica media durante la simulazione a cardiac output di 4l/min e frequenza cardiaca di 70bpm

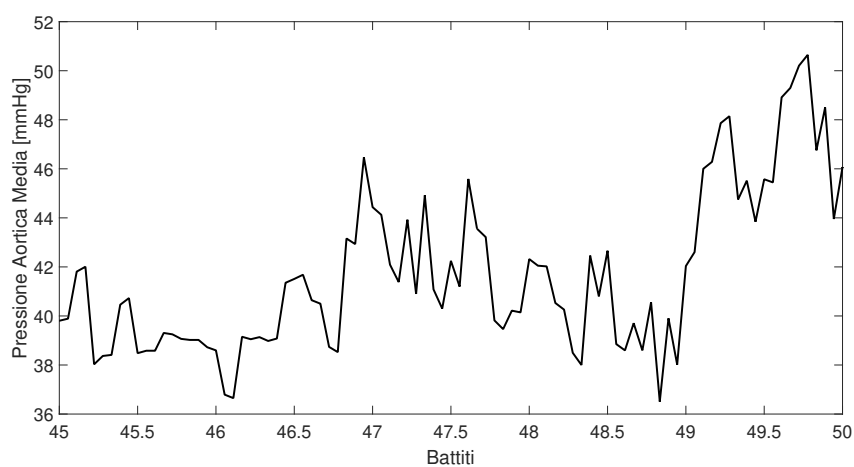


Figura 5.31: Particolare dell'andamento a regime della pressione aortica media durante l'implementazione su banco sperimentale a cardiac output di 4l/min e frequenza cardiaca di 70bpm



## Capitolo 6

# Conclusioni e sviluppi futuri

I circuiti circolatori come il *Pulse Duplicator* forniscono una realizzazione in vitro delle condizioni circolatorie presenti in vivo al fine, ad esempio, di esplorare le caratteristiche tecniche di dispositivi protesici cardiovascolari e di valutarne, attraverso esperimenti, gli effetti sull'organismo.

L'impostazione della resistenza periferica cardiovascolare (prodotta dalla transizione da vasi a diametro maggiore a quelli con diametro minore) attraverso ad esempio l'utilizzo di una valvola, è fondamentale. Spesso tale impostazione è effettuata manualmente conducendo prove ripetute che richiedono la competenza degli esperti e notevole investimento di tempo. Altresì, risulta difficile garantire la ripetibilità degli esperimenti al variare delle diverse configurazioni e condizioni operative. Per ovviare a tali inconvenienti, può essere utile l'implementazione di un dispositivo automatizzato capace di assicurare impostazioni accurate della resistenza e in grado di riprodurre con precisione e accuratezza i parametri cardiovascolari, consentendo una migliore realizzazione di condizioni sperimentali all'interno del circuito circolatorio in vitro.

A tal fine, in questa tesi è stata utilizzata una valvola motorizzata di resistenza periferica con l'obiettivo di controllare la pressione aortica. Per garantire una certa flessibilità per quanto concerne l'obiettivo di controllo, nonché di assicurare adeguate prestazioni del controllore al variare delle diverse configurazioni di esperimenti e condizioni operative, il problema è stato impostato come un problema di ottimizzazione ed è stato affrontato utilizzando un algoritmo di tipo model-free quale *ESC Sliding mode*.

Preliminarmente l'algoritmo è stato progettato e verificato sfruttando un ambiente di simulazione Matlab/Simulink che riproduce il *Pulse Duplicator* in assegnate condizioni qualitative nominali, e successivamente è stato implementato nel sistema reale.

Al fine di rendere automatica la valvola di resistenza periferica è stato necessario accessoriare il *Pulse Duplicator* con opportuni componenti hardware e software. In particolare, si sono impiegati un motoriduttore in corrente con-

tinua, un circuito di condizionamento dei segnali, una scheda Arduino, e un ponte H, mentre è stato utilizzato l'ambiente di programmazione LabVIEW per lo sviluppo dell'algoritmo di controllo.

Vale la pena sottolineare che l'allestimento della parte hardware e software ha richiesto una particolare attenzione in relazione alla peculiarità del sistema in esame e dei vincoli tecnologici presenti. Si è trattato infatti di un'operazione di upgrade rispetto alla situazione precedente laddove l'apertura della valvola di resistenza periferica era impostata manualmente di volta in volta.

Gli esperimenti reali hanno confermato le buone prestazioni dell'algoritmo di controllo di tipo *ESC Sliding Mode*. La pressione aortica media impostata come valore obiettivo, infatti, viene raggiunta e mantenuta.

Possibili sviluppi futuri includono:

- Il test delle prestazioni dell'algoritmo di controllo, con l'obiettivo di minimizzare l'errore tra riferimento di pressione aortica e misura, in diverse condizioni di esperimenti (ad esempio seguendo le indicazioni della normativa BS EN ISO5840 1:2015, imponendo una pressione aortica media obiettivo di 100mmHg) ;
- Definire ulteriori funzioni di costo (di cui ad esempio non è noto a priori il valore ottimo), considerando anche l'inclusione di vincoli e verificare le opportunità di impiego e l'efficacia dell'approccio model-free proposto.

## Appendice A

# Caratterizzazione degli esperimenti

Nel seguito si mostrano in dettaglio i risultati delle elaborazioni per la caratterizzazione del sistema per i rimanenti 4 esperimenti, descritti nel capitolo 5.2, e riassunti dalla croce sperimentale riportata in figura A.1. Nell'ordine

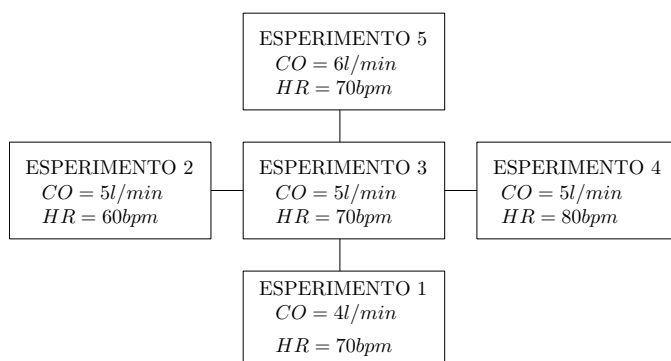


Figura A.1: Croce sperimentale relativa ai 5 esperimenti riprodotti al *pulse duplicator*

si mostrano gli andamenti delle pressioni aortiche medie, al variare del grado di chiusura della resistenza periferica, negli ultimi 4 esperimenti. Ogni polinomio utilizzato per la stima delle curve è di ordine 4. Si mostrano, inoltre, i corrispettivi andamenti delle funzioni costo quadratiche.

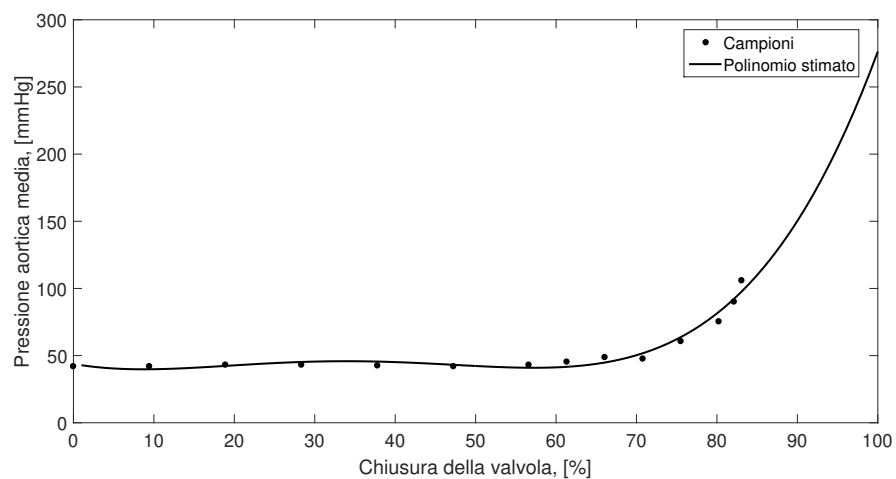


Figura A.2: Andamento della pressione aortica media per le varie percentuali di chiusura della valvola in condizioni di 60bpm e cardiac output di 5l/min

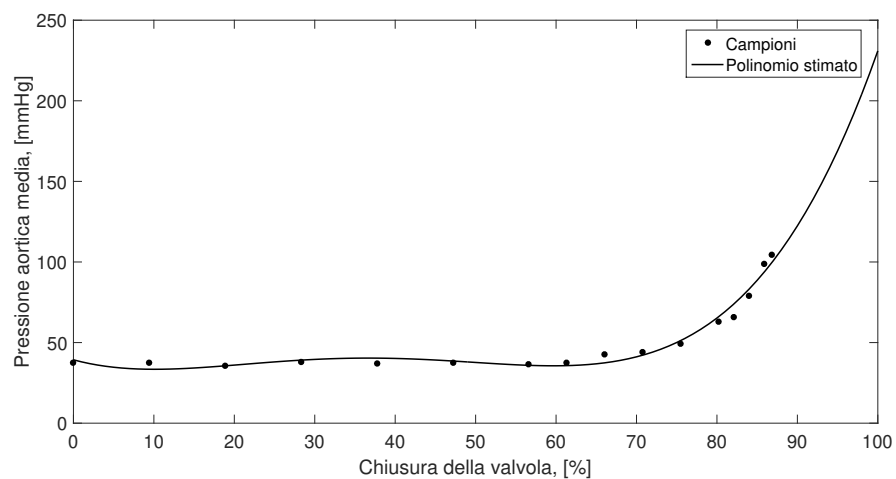


Figura A.3: Andamento della pressione aortica media per le varie percentuali di chiusura della valvola in condizioni di 70bpm e cardiac output di 5l/min

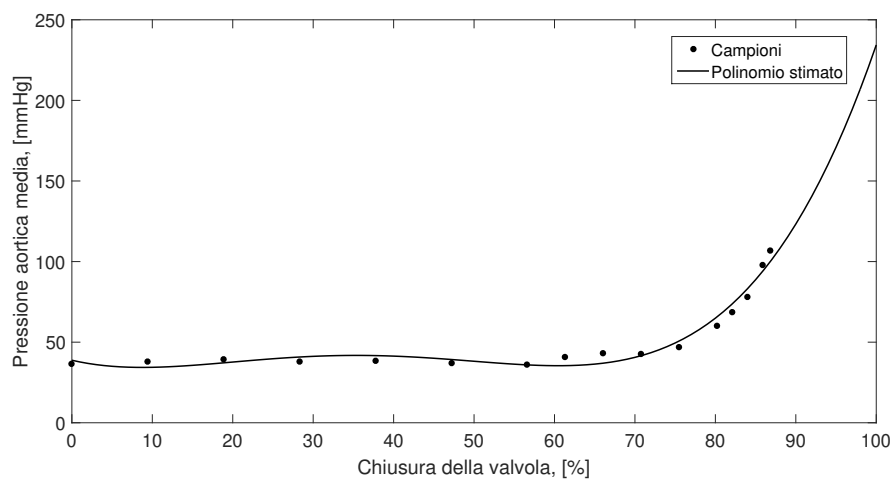


Figura A.4: Andamento della pressione aortica media per le varie percentuali di chiusura della valvola in condizioni di 80bpm e cardiac output di 5l/min

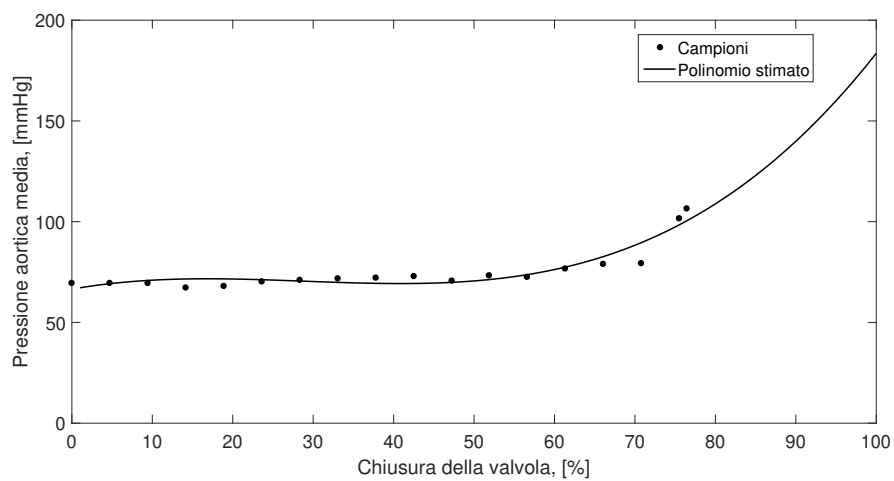


Figura A.5: Andamento della pressione aortica media per le varie percentuali di chiusura della valvola in condizioni di 70bpm e cardiac output di 6l/min

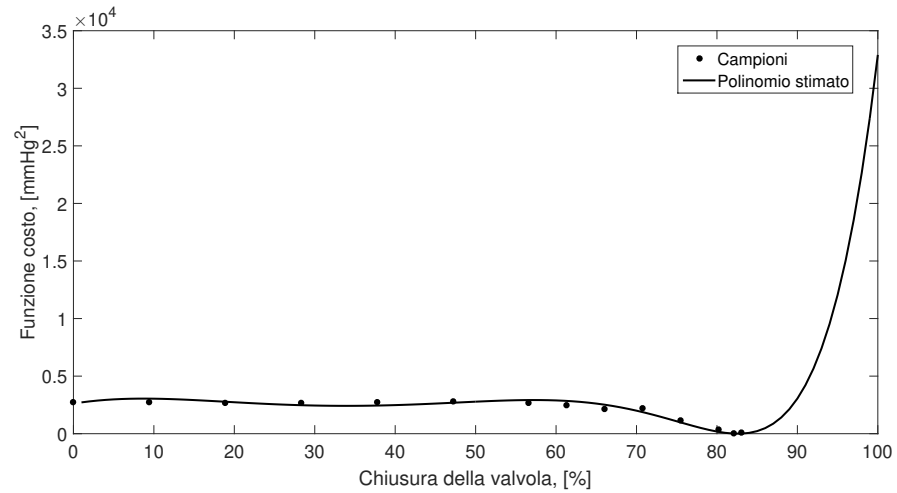


Figura A.6: Andamento della funzione costo con pressione aortica media obiettivo di 100mmHg nelle varie percentuali di chiusura della valvola in condizioni di 60bpm e cardiac output di 5l/min

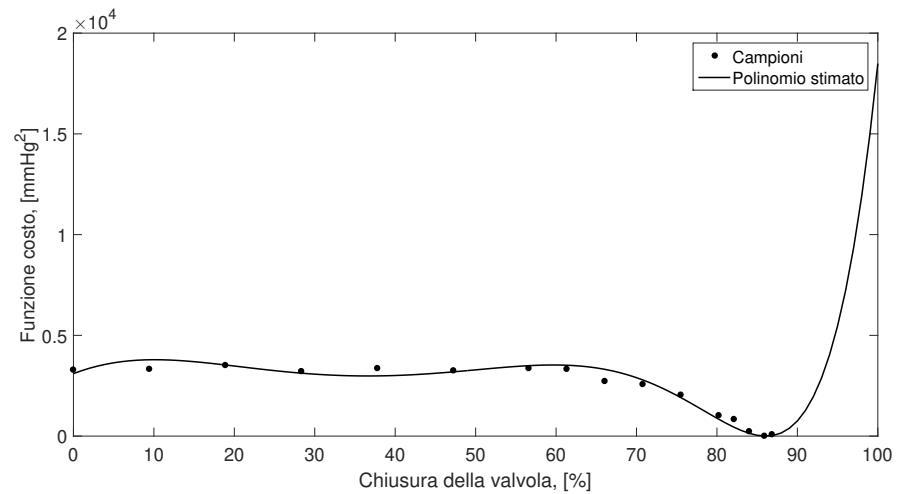


Figura A.7: Andamento della funzione costo con pressione aortica media obiettivo di 100mmHg nelle varie percentuali di chiusura della valvola in condizioni di 70bpm e cardiac output di 5l/min



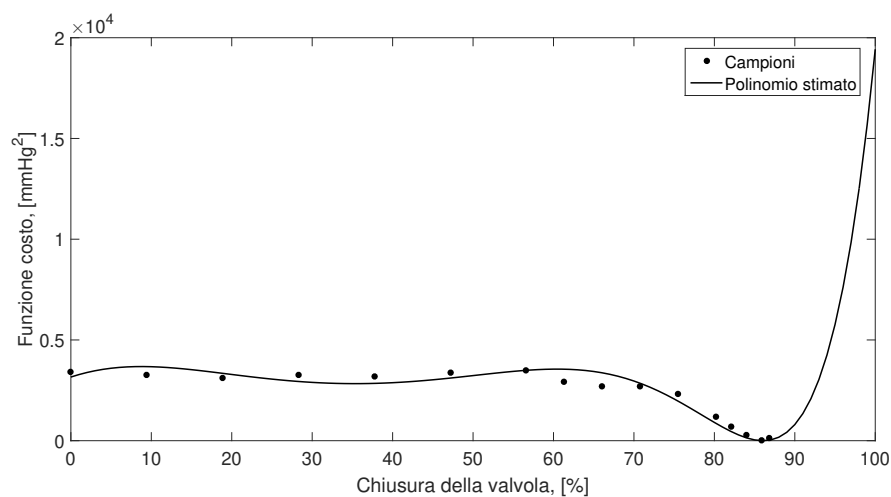


Figura A.8: Andamento della funzione costo con pressione aortica media obiettivo di 100mmHg nelle varie percentuali di chiusura della valvola in condizioni di 80bpm e cardiac output di 5l/min

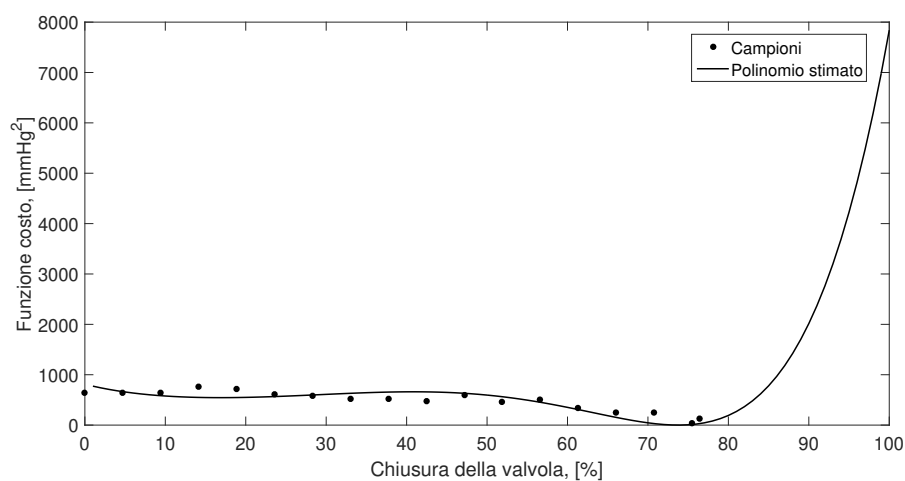


Figura A.9: Andamento della funzione costo con pressione aortica media obiettivo di 100mmHg nelle varie percentuali di chiusura della valvola in condizioni di 70bpm e cardiac output di 6l/min



## Appendice B

# Simulazione del controllo

Nel seguito si mostrano in dettaglio i risultati dell'applicazione in simulazione dell'algoritmo Extremum Seeking Control Sliding Mode per i rimanenti 4 esperimenti della croce sperimentale di figura A.1.

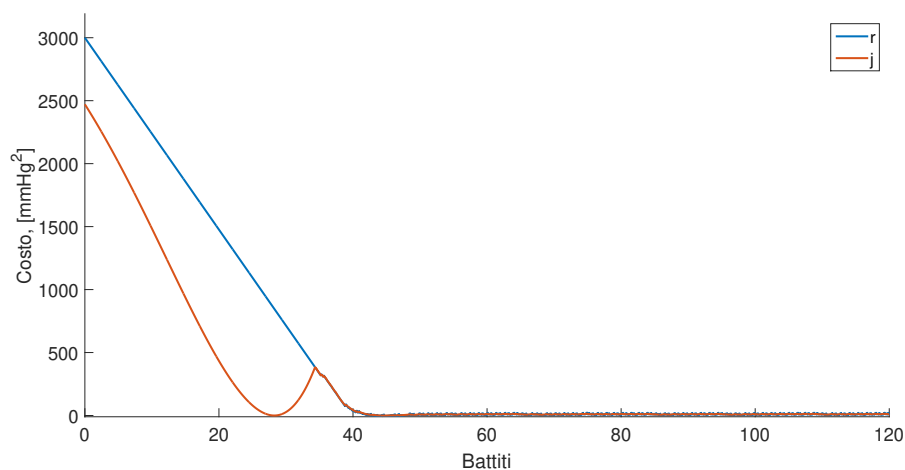


Figura B.1: Andamento di riferimento e funzione costo durante l'implementazione dell'algoritmo di controllo in condizioni di 60bpm e cardiac output di 5l/min

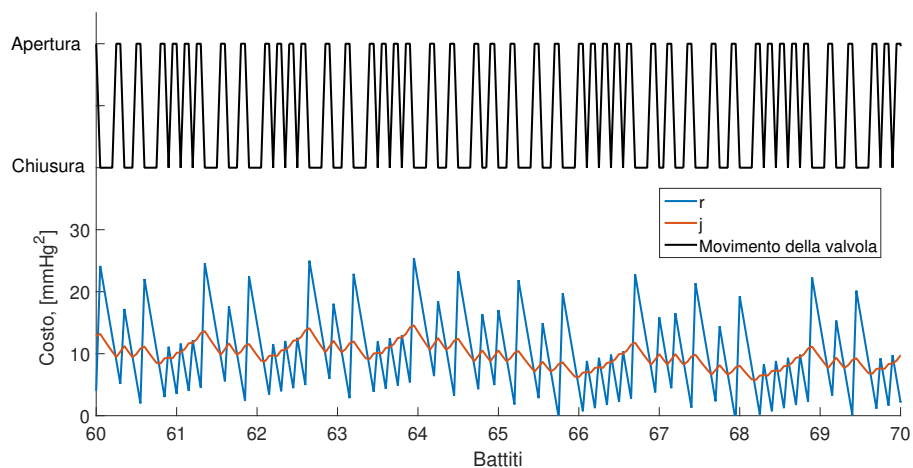


Figura B.2: Particolare dell'andamento di riferimento e funzione costo e del movimento della valvola durante l'implementazione dell'algoritmo di controllo in condizioni di 60bpm e cardiac output di 5l/min

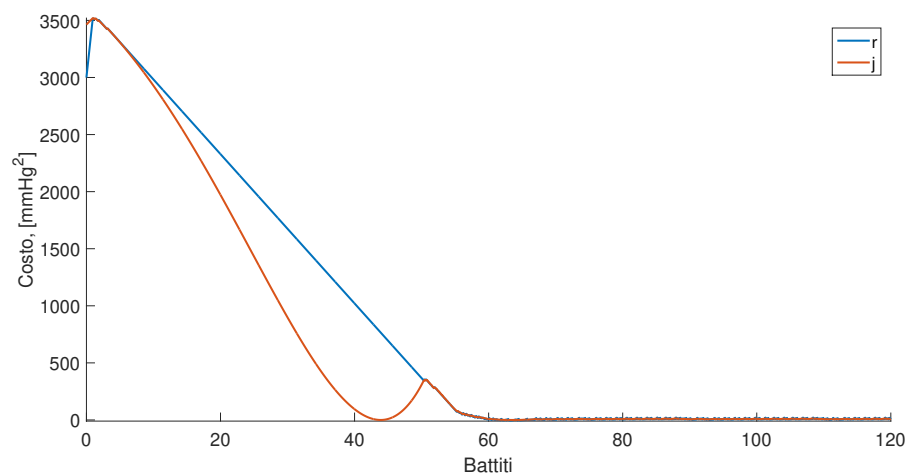


Figura B.3: Andamento di riferimento e funzione costo durante l'implementazione dell'algoritmo di controllo in condizioni di 70bpm e cardiac output di 5l/min

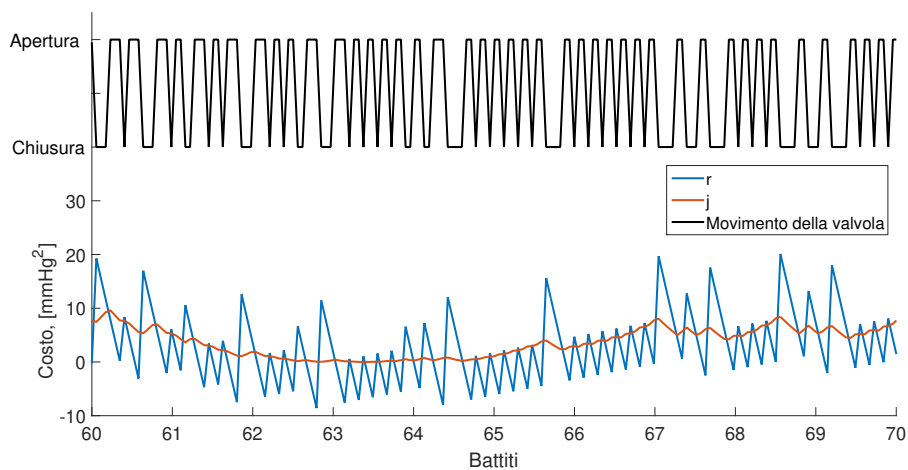


Figura B.4: Particolare dell'andamento di riferimento e funzione costo e del movimento della valvola durante l'implementazione dell'algoritmo di controllo in condizioni di 70bpm e cardiac output di 5l/min

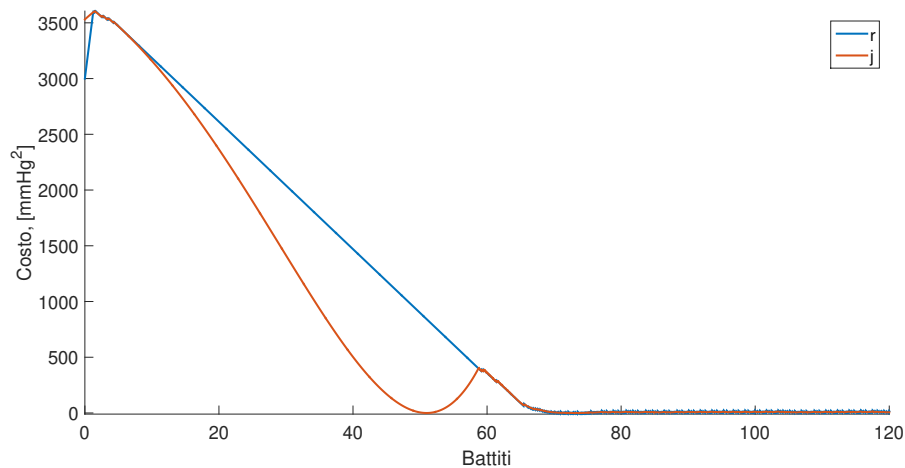


Figura B.5: Andamento di riferimento e funzione costo durante l'implementazione dell'algoritmo di controllo in condizioni di 80bpm e cardiac output di 5l/min

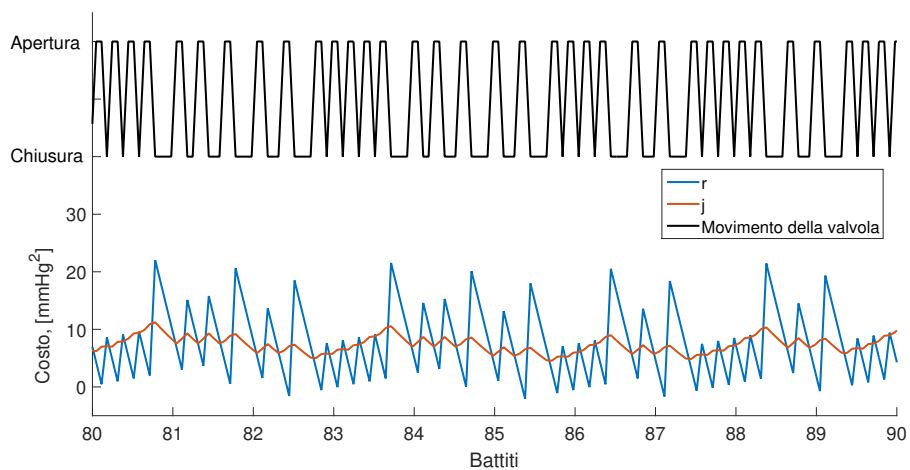


Figura B.6: Particolare dell'andamento di riferimento e funzione costo e del movimento della valvola durante l'implementazione dell'algoritmo di controllo in condizioni di 80bpm e cardiac output di 5l/min

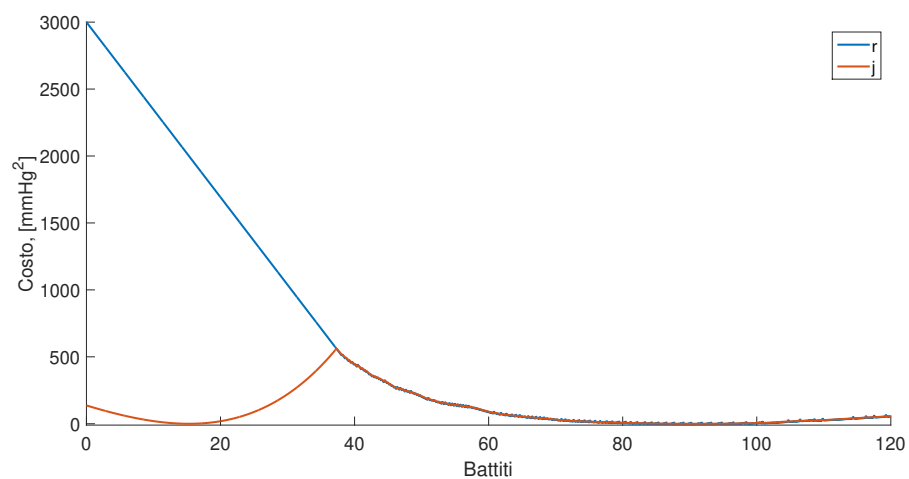


Figura B.7: Andamento di riferimento e funzione costo durante l'implementazione dell'algoritmo di controllo in condizioni di 70bpm e cardiac output di 6l/min

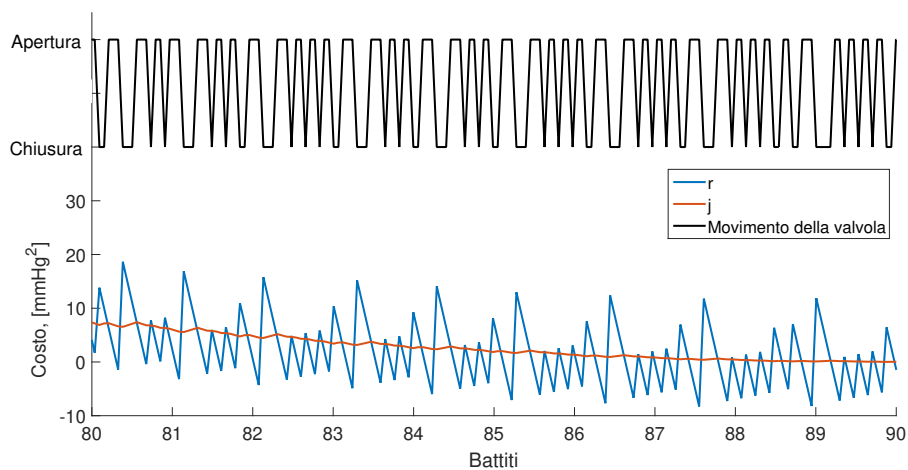


Figura B.8: Particolare dell'andamento di riferimento e funzione costo e del movimento della valvola durante l'implementazione dell'algoritmo di controllo in condizioni di 70bpm e cardiac output di 6l/min

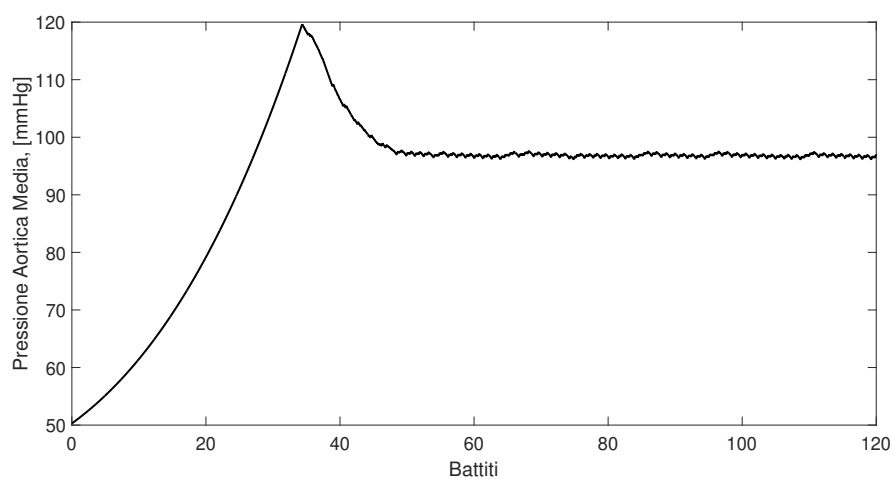


Figura B.9: Andamento della pressione aortica media durante l'implementazione dell'algoritmo di controllo in condizioni di 60 battiti al minuto e cardiac output di 5l/min

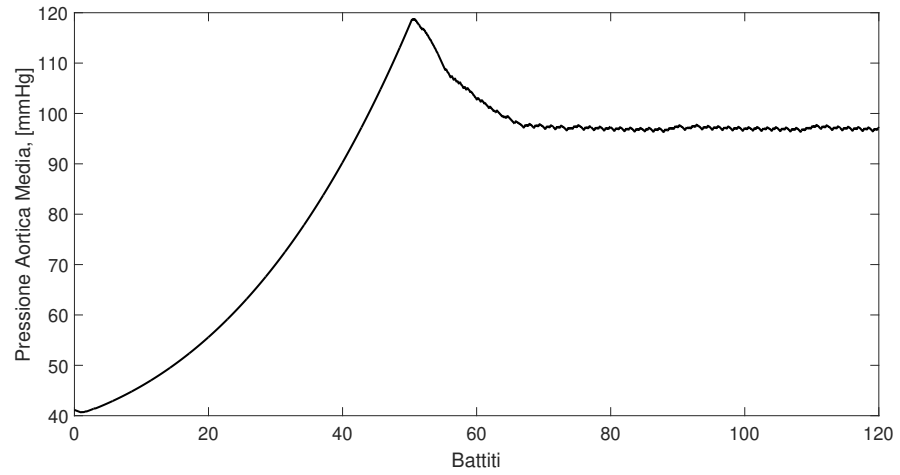


Figura B.10: Andamento della pressione aortica media durante l'implementazione dell'algoritmo di controllo in condizioni di 70 battiti al minuto e cardiac output di 5l/min

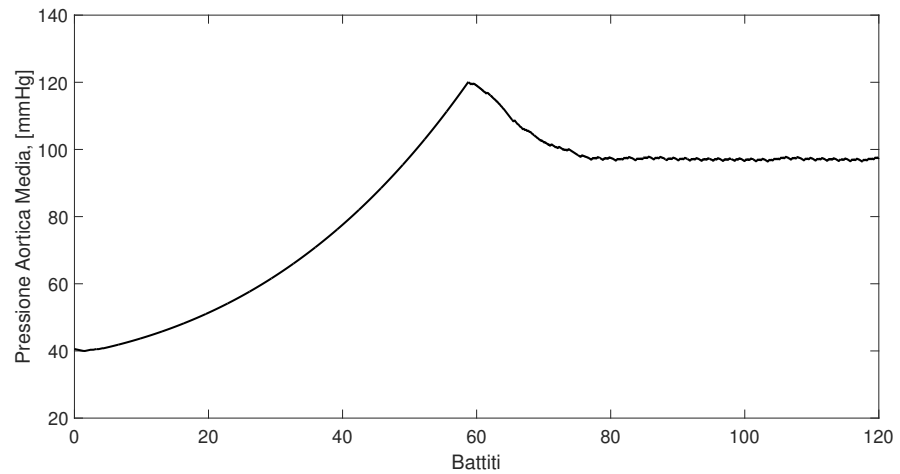


Figura B.11: Andamento della pressione aortica media durante l'implementazione dell'algoritmo di controllo in condizioni di 80 battiti al minuto e cardiac output di 5l/min



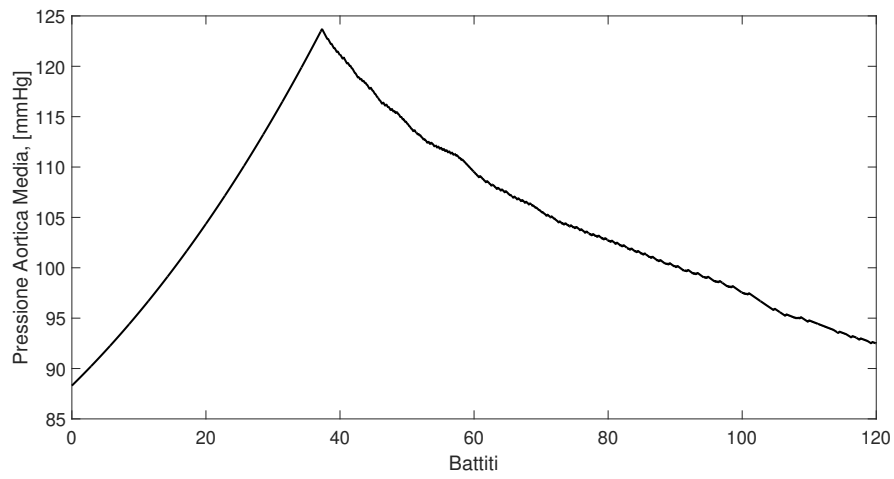


Figura B.12: Andamento della pressione aortica media durante l'implementazione dell'algoritmo di controllo in condizioni di 70 battiti al minuto e cardiac output di 6l/min



# Bibliografia

- [1] Luigi Di Micco, 2013. *Messa a punto di un circuito idraulico che simula la circolazione cardiovascolare sistemica umana*, Tesi di Laurea Magistrale.
- [2] Riccardo Toninato. *Development of a Laboratory for Cardiovascular Fluid Dynamics Studies*, PhD Thesis.
- [3] Bellio Silvia, 2014. *Messa a punto di un dispositivo Pulse Duplicator per l'analisi sperimentale di protesi cardiovascolari*, Tesi di Laurea Magistrale.
- [4] <https://www.humanitas.it/enciclopedia/anatomia/apparato-cardiocircolatorio>
- [5] <https://www.cryolife.com/products/on-x-heart-valves/>
- [6] Kartik B., Aryur and Miroslav Krisric, 2003. *Real-Time Optimization by Extremum-Seeking Control*, John Wiley & Sons. Inc.
- [7] Tan Y., Moase W.H., Manzie C., Nesic D., Mareels I.M.Y., 2010. *Extremum Seeking From 1922 to 2010*, Chinese Control Conference, 29-31 July 2010.
- [8] Moase W.H., Manzie C., Brear M.J., 2009. *Newton-Like Extremum-Seeking Part I: Theory*, Chinese Control Conference, 16-18 December 2009.
- [9] Beghi A., Lionello M., Rampazzo M., 2019. *Efficient Operation of Indirect Evaporative Data Center Cooling Systems via Newton-Like Extremum-Seeking Control*, IEEE Conference on Control Technology and Applications, 19-21 Agosto 2019, Hong Kong.
- [10] Leblanc M., 1922. *Sur l'électrification des chemins de fer au moyen de courants alternatifs de fréquence élevée*. Revue générale de l'électricité, 12(8):275–277, 1922.

- [11] Draper C.S. and Li Y.T. *Principles of optimizing control systems and an application to the internal combustion engine*. American Society of Mechanical Engineers, 1951.
- [12] Wang H.H. and Krstic M. *Extremum seeking for limit cycle minimization*. IEEE Transactions on Automatic control, 45(12):2432–2436, 2000.
- [13] Rotea M.A. *Analysis of multivariable extremum seeking algorithms*. In Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No. 00CH36334), volume 1, pages 433–437. IEEE, 2000.
- [14] Walsh G.C. *On the application of multi-parameter extremum seeking control*. In Proceedings of the 2000 American Control Conference. Volume 1, pages 411–415. IEEE, 2000.
- [15] Malek H., Manzie C., Dadras S., Chen Y., 2016. *Performance analysis of fractional order extremum seeking control*, ISA Transactions 63 (2016) 281–287.
- [16] Tinglong P., Zhicheng J., Zhenhua J., 2008. *Maximum Power Point Tracking of Wind Energy Conversion Systems Based on Sliding Mode Extremum Seeking Control*, IEEE Energy2030 Atlanta, GA USA 17-18 November, 2008.
- [17] Olalla C., Arteaga M. I., Leyva R., El Aroudi A., 2007. *Analysis and Comparison of Extremum Seeking Control Techniques*, 1-4244-0755-9/07, 2007 IEEE.
- [18] Korovin S. K. and Utkin V. I., 1972. *The use of the slip mode in problems of static optimization*, Automatic and Remote Control, pp. 50–60, 1972.
- [19] <https://store.arduino.cc/arduino-uno-rev3>
- [20] [https://win.adrirobot.it/motor\\_driver/1298/L298N\\_Dual\\_H-Bridge\\_Motor\\_Controller.htm](https://win.adrirobot.it/motor_driver/1298/L298N_Dual_H-Bridge_Motor_Controller.htm)
- [21] <https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-1298n-pwm-h-bridge/>