

POLITECNICO DI TORINO

Dipartimento di Ingegneria Meccanica e Aerospaziale

Corso di Laurea Magistrale in Ingegneria Aerospaziale

Tesi di Laurea Magistrale



**Politecnico
di Torino**

Controllo della scia di corpi tozzi mediante Reinforcement Learning

Relatore:

prof. Iuso Gaetano

ing. Cafiero Gioacchino

Candidato:

Amico Enrico s270313

Anno Accademico 2020-2021

Sommario

Il seguente lavoro di tesi rappresenta l'applicazione in ambito sperimentale ed ad alto numero di Reynolds della tecnica del Deep Reinforcement Learning per il controllo attivo della scia di un veicolo da trasporto commerciale.

Si dimostra che l'agente sfruttando una rete neurale è in grado di individuare una strategia con cui ridurre la resistenza aerodinamica del modello mediante l'attuazione di getti posti lungo gli spigoli della base del modello.

La metodologia è stata testata per diversi numeri di Reynolds e si è visto che, nonostante si raggiungano condizioni di forcing differenti, il forcing è tale da introdurre una simmetrizzazione della scia con conseguente riduzione del coefficiente di resistenza aerodinamica pari al $\sim 9\%$ per il numero di Reynolds più basso e $\sim 8\%$ per quello più alto.

Questa implementazione è risultata particolarmente vantaggiosa anche dal punto di vista del tempo fisico necessario per eseguire l'addestramento, infatti ognuno dei due casi analizzati ha mostrato una sufficiente convergenza per 26 *episodi*, equivalenti ad appena 3 *ore* e 15 *minuti* di misura.

In letteratura, al momento della scrittura, non sono presenti applicazioni a valori simili del numero di Reynolds e dai risultati ottenuti questa metodologia risulta particolarmente vantaggiosa e versatile per problemi di controllo e ottimizzazione in ambito sperimentale. Questo apre la porta alla possibilità di implementazione della metodologia in problemi di maggiore interesse comune e pratico, spostando l'attenzione da formulazioni puramente numeriche ad applicazioni nel quotidiano.

Abstract

The following thesis shows application of Deep Reinforcement Learning to an experiment at high Reynolds for the active control of the wake behind a commercial transport vehicle. It is demonstrated that the agent, using an artificial neural network, learns a strategy to reduce the aerodynamic drag of the model through the implementation of air jets placed along the edges of the model base.

The method has been tested at different Reynolds numbers and, in spite of the different forcing conditions, the wake attains a more symmetric state with consequent reduction of the drag coefficient of $\sim 9\%$ and $\sim 8\%$ for the lowest and highest Reynolds number, respectively.

This implementation was also particularly advantageous from the physical time required to perform the training. In fact each of the two analysed cases showed convergence within 26 episodes, corresponding to about 3 hours and 15 minutes.

At the time of writing, these values of the Reynolds number for the application of DRL are still unexplored and the results obtained show that this methodology is particularly advantageous and versatile for control and optimization problems formulated on experimental problems. This opens the path to the implementation of the methodology in problems of greater common and practical interest, shifting the focus away from purely numerical formulations to day to day applications.

Indice

Elenco delle tabelle	III
Elenco delle figure	IV
1 Introduzione	1
2 Reinforcement Learning	5
2.1 Concetti base del Reinforcement Learning	5
2.1.1 Processo decisionale di Markov	8
2.1.2 Policies and Value Functions	9
2.2 Rete Neurale	11
2.3 DRL - Deep Reinforcement Learning	14
2.3.1 Deep Q-Learning	15
2.3.2 Policy Gradient Methods	16
2.3.3 Tensorforce Agent	17
3 Set up sperimentale	18
3.1 Sistema di attuazione getti	20
3.2 Sistema di misura	21
4 Implementazione modello	22
4.1 Ambiente	22
4.1.1 Implementazione ambiente	24
4.1.2 Implementazione funzione Execute	28
4.1.3 Implementazione funzione Reset	28
4.2 Algoritmo di Reinforcement Learning	30
5 Risultati	31
5.1 Caso 1: attuazione simmetrica - $Re_{\infty} \sim 0.9 \cdot 10^5$	32
5.2 Caso 2: attuazione asimmetrica - $Re_{\infty} \sim 0.9 \cdot 10^5$	33
5.2.1 Bilancio energetico	35
5.2.2 Performance dell'agente	36
5.2.3 Analisi spettrale del segnale di pressione	40

5.2.4	Ricerca moti organizzati	49
5.3	Caso 3: attuazione asimmetrica - $Re_\infty \sim 1.1 \cdot 10^5$	52
5.3.1	Performance dell'agente	52
5.3.2	Analisi spettrale del segnale di pressione	54
5.3.3	Ricerca moti organizzati	59
5.4	Confronto tra Caso 2 e Caso 3	64
	Conclusioni e sviluppi futuri	66
	Ringraziamenti	69

Elenco delle tabelle

5.1	Tensioni di forcing - caso 2	40
5.2	Velocità di forcing - caso 2	40
5.3	Frequenze di forcing - caso 2	40
5.4	Posizione microfoni analisi spettrale	41
5.5	Tabella riepilogo eventi Unforced e Forced - Caso 2	50
5.6	Tensioni di forcing - caso 3	54
5.7	Velocità di forcing - caso 3	54
5.8	Frequenze di forcing - caso 3	54
5.9	Tabella riepilogo eventi Unforced e Forced - Caso 2	62
5.10	Tabella riepilogativa forcing superiore	64
5.11	Tabella riepilogativa forcing inferiore	64
5.12	Tabella riepilogativa forcing laterale	64

Elenco delle figure

1.1	Visualizzazioni attorno al corpo di Ahmed	3
2.1	Schema Reinforcement Learning riadattato da Sutton and Barto	7
2.2	Schema Value Function	10
2.4	Schema Algoritmo Backpropagation	14
3.1	Schema del modello	18
3.2	Sistema di attuazione e controllo getto	19
3.3	Schema cilindro attuatore	20
4.1	Schema concettuale RL applicato al problema	22
4.2	Schema di interfaccia tra Python e LabView	24
4.3	Schema algoritmo di acquisizione tramite il modulo National Instrument	25
4.4	Output acquisizione modulo National Instrument	25
4.5	Schema algoritmo di acquisizione tramite la Smart-Zoc	26
4.6	Schema algoritmo attuazione getti	27
4.7	Schema algoritmo funzione execute	28
5.1	Training caso 1	33
5.2	Training caso 2	34
5.3	Training - Actions bottom jet	34
5.4	Training - Actions lateral jet	35
5.5	Training - Actions up jet	35
5.6	$\frac{\Delta C_D}{C_\mu}$ vs Episode	37
5.7	Valutazione episodi - Caso 2	37
5.8	Cp base distribution - eventi 1:8 - Caso 2	38
5.9	Cp base distribution - eventi 8:16 - Caso 2	39
5.10	Schema posizione microfoni analisi spettrale	41
5.11	PSD - <i>Power spectral density</i> - caso 2	42
5.12	Mode energies - Caso 2	44
5.13	Energia cumulativa Pod modes	44
5.14	Campo medio fluttuazioni di pressioni - Caso 2	45
5.15	Pod Modes 1:2 - Caso 2	47
5.16	Pod Modes 3:4 - Caso 2	48
5.17	Varianza locale - Caso 2	49

5.18	Evento - Caso 2	50
5.19	Evento Up - Caso 2	51
5.20	Evento Down - Caso 2	51
5.21	Training caso 3	52
5.22	$\frac{\Delta C_D}{C_\mu}$ vs Episode	53
5.23	Valutazione episodi - Caso 3	53
5.24	Cp base distribution - eventi 1:8 - Caso 3	55
5.25	Cp base distribution - eventi 8:16 - Caso 3	56
5.26	PSD - <i>Power spectral density</i> - caso 3	57
5.27	Mode energies - Caso 3	58
5.28	Campo medio fluttuazioni di pressioni - Caso 3	58
5.29	Pod Modes 1:2 - Caso 3	60
5.30	Pod Modes 3:4 - Caso 3	61
5.31	Varianza locale - Caso 3	62
5.32	Evento - Caso 3	62
5.33	Evento Up - Caso 3	63
5.34	Evento Down - Caso 3	63

Capitolo 1

Introduzione

La resistenza aerodinamica è uno dei termini più rilevanti nel consumo di combustibile per un veicolo. La resistenza aerodinamica è proporzionale al quadrato della velocità, questo implica una richiesta di potenza da parte del motore proporzionale al cubo della velocità. Queste osservazioni portano ad evidenziare come il ruolo della resistenza aerodinamica è più importante durante la percorrenza di strade ad alta velocità (i.e. autostrade) rispetto all'ambiente cittadino (si stima rispettivamente 50% e 20% del consumo complessivo di combustibile). [2]

Il ruolo della aerodinamica va ad amplificarsi nel caso di veicoli da trasporto merce: in primis per la loro forma. Infatti questi veicoli sono pensati per ottimizzare il volume trasportabile e quindi si presentano con una classica geometria a parallelepipedo che risulta influenzare pesantemente l'aerodinamica del veicolo. Se a questo si va ad aggiungere la condizione operativa, cioè sono mezzi che per la maggior parte della loro vita percorrono strade ad alta velocità, si mette in luce come la resistenza aerodinamica impatti pesantemente nel bilancio di combustibile utilizzato.

Perciò anche una riduzione della resistenza aerodinamica di pochi punti percentuali è importante in quanto permette di ridurre pesantemente il consumo di combustibile. Da uno studio condotto nel Regno Unito e negli Stati Uniti si è visto che il carburante rappresenta il 20 – 30% dei costi operativi [3], quindi riducendo la resistenza aerodinamica si ha una notevole riduzione dei costi operativi. Allo stesso tempo si riduce pesantemente l'impatto sull'inquinamento. Queste considerazioni trovano particolare interesse anche a fronte delle continue restrizioni introdotte dalle normative per la riduzione dell'inquinamento e dalle continue richieste di riduzione dei consumi da parte del mondo automotive al fine di poter ridurre il numero di celle di batteria nelle auto elettriche.

Da queste considerazioni scaturisce la necessità di ricercare una condizione di ottimizzazione aerodinamica al fine di ottenere una riduzione della resistenza. Le tecniche di riduzione si dividono in tecniche passive (esempi di applicazioni di tecniche passive per la riduzione della resistenza si possono trovare in [4] e [5]) e tecniche

attive. Per la prima tipologia non è richiesto l'utilizzo di energia, che invece è necessaria per le tecniche attive. Le tecniche passive, seppur molto promettenti dal punto di vista dei risultati, hanno delle forte limitazioni legate a problemi di sicurezza derivanti dall'aggiunta di appendici, alla modifica della carrozzeria e sono progettate per una sola condizione di funzionamento con conseguente capacità di lavorare nel punto di massima efficienza solo per la condizione di progetto. Nel caso di controllo passivo su veicoli in letteratura si trovano diverse soluzioni come l'utilizzo di vortex generators [6], di flaps [7], di un deflettore mobile automatico [8] e modifiche locali del corpo [9].

Invece le tecniche attive risultano più promettenti in ottica pratico-implementativa perché non sono invasive dal punto di vista geometrico, quindi di più facile implementazione, e ad una minore affidabilità contrappongono una elevata flessibilità che permette un corretto funzionamento non solo nella condizione di progetto.

I veicoli da trasporto merce rientrano nella categoria dei corpi tozzi, cioè corpi dove la resistenza è costituita essenzialmente da una resistenza di forma. Questi si contraddistinguono per la presenza di una scia a bassa pressione che introduce una base drag, principale responsabile della resistenza del veicolo. La maggior parte delle tecniche di riduzione della resistenza aerodinamica si concentrano sulla parte posteriore del corpo, cercando di aumentare il recupero di pressione. Il motivo per cui ci si concentra nella parte posteriore del corpo è legato alla maggior semplicità di implementazione rispetto ad altre zone del veicolo e non nella migliore performance. Infatti [Minelli et al.](#) hanno dimostrato che controllando lo strato limite nel punto in cui questo si genera, nel caso del veicolo questo auspicabilmente coincide con il montante anteriore, si ottengono delle migliori prestazioni ma, come ribadito in precedenza, questo risulterebbe di maggiore complessità implementativa.

La forma di questi corpi porta ad avere una intensa separazione, una dinamica della scia particolarmente complessa con dinamiche che interessano grandi range di scale spaziali. Peculiarità del campo di moto che si viene a generare è quella di avere una rottura della simmetria della scia con conseguente sviluppo di asimmetria, pur considerando la presenza di una geometria simmetrica.

Negli ultimi anni si è cercato di investigare su quelle che sono le cause e i meccanismi principali all'origine di questo campo di moto. Ad esempio [Evstafyeva et al.](#) hanno cercato di riprodurre numericamente i regimi che caratterizzano la scia prodotta dal corpo di Ahmed: una geometria semplificata caratterizzata da una parte anteriore curva per evitare la separazione del flusso, un corpo dritto con sezione trasversale rettangolare e una parte posteriore piatta, che può presentare o meno un angolo inclinato. Questo corpo spesso è utilizzato per studiare il campo di moto di un veicolo [12]. Dai risultati riportati da [Evstafyeva et al.](#) si osserva come a basso numero di Reynolds la scia si presenta in uno stato simmetrico e stabile caratterizzato da due regioni di ricircolo allineate orizzontalmente una accanto all'altra. Sul piano verticale invece è presente una asimmetria, infatti si evidenzia la presenza di una bolla nella parte superiore della base che crea uno shear layer.

Questo però pur interagendo con il flusso del sottoscocca non è sufficiente per creare un ricircolo anche nella parte inferiore.

Aumentando ulteriormente il numero di Reynolds si osserva la formazione di uno stato di simmetria instabile che ha una durata molto limitata, dopodiché si porta in uno stato asimmetrico stabile.

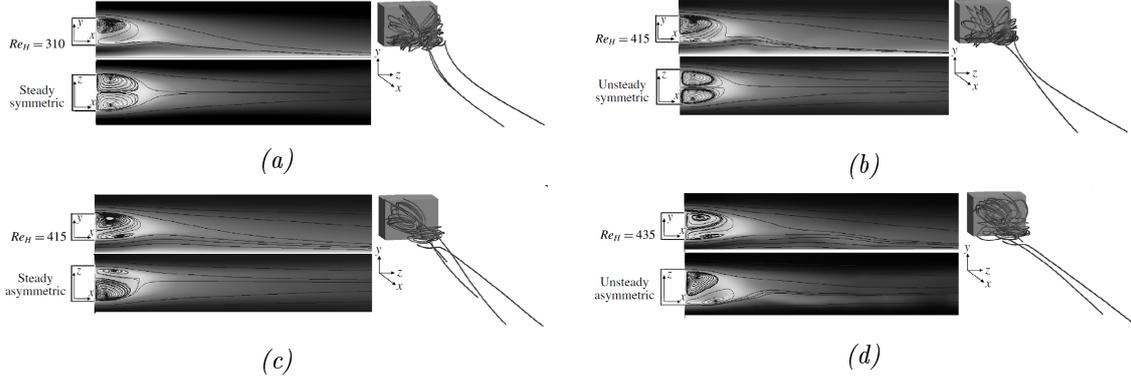


Figura 1.1: Visualizzazioni attorno al corpo di Ahmed a $Re_H = 310$ (a) e (b), $Re_H = 415$ (c) e $Re_H = 435$ (d). Le figure a destra mostrano le streamline proiettate su di un piano in una vista superiore e laterale, mentre le figure a sinistra mostrano le streamline nella scia 3D. Fonte: [Evstafyeva et al.](#)

In questa condizione si formano due regioni di ricircolo allineate verticalmente. Queste osservazioni hanno portato [Haffner et al.](#) a supporre che la rottura della simmetria si ha quando il Reynolds è sufficiente da creare un flusso nel sottoscocca tale da far interagire lo shear layer superiore ed inferiore così da creare ricircolo nella base. Quindi la rottura della simmetria si ha inizialmente per valori di Reynolds bassi ([Grandemange et al.](#)), ancora in condizioni di laminarità. Tale condizione persiste anche ad alti numeri di Reynolds e quindi anche quando si raggiunge una condizione turbolenta. La presenza di fluttuazioni nel flusso turbolento sono responsabili del passaggio della scia da uno stato asimmetrico ad un altro con scale lente e casuali, si parla di bi-stabilità ([Grandemange et al.](#)) o multi-stabilità ([Rigas et al.](#)).

Anche ([Barros et al.](#)) hanno evidenziato l'importanza cruciale del sottoscocca sulla asimmetria trasversale della scia. Aumentando ulteriormente il numero di Reynolds si osserva uno stato della scia instabile e asimmetrica, con le bolle di ricircolo fisse nella loro posizione sul piano verticale (figura 1.1d). Questa asimmetria quindi si manifesta attraverso inversioni laterali casuali della scia in modo da ripristinare statisticamente la condizione di simmetria (si parla di comportamento bi-stabile).

In letteratura si trovano diversi lavori in cui si è cercato di correggere l'asimmetria laterale (i.e. [Li et al.](#) hanno forzato la scia su entrambi i lati, con getti

pulsati a bassa frequenza, in base della posizione laterale della scia) ma si è visto una riduzione pari al il 2% della pressure drag. Mentre si è osservato che l'utilizzo di tecniche passive determina un maggior impatto sulla riduzione della resistenza. [Haffner et al.](#) evidenziano come tutte le metodologie utilizzate per controllare il meccanismo della asimmetria si basano sulla prevenzione del feedback tra gli shear layers opposti, che è all'origine della bi-modalità.

Altri studi ([Barros et al.](#)) hanno dimostrato che l'applicazione simultanea del controllo attivo del flusso e dell'effetto Coanda è estremamente vantaggiosa per la riduzione della resistenza aerodinamica. Inoltre gli autori hanno evidenziato come una attuazione ad alta frequenza è necessaria per avere riduzioni di resistenza. Al contrario si è evidenziato come un'attuazione a banda larga può portare ad un aumento della drag.

[Cerutti et al.](#) hanno effettuato un'indagine sperimentale al fine di ridurre la resistenza di un modello di un veicolo *square-back* mediante l'utilizzo di getti continui posti nei 4 lati della superficie posteriore del modello. Una prima indagine ha mostrato che l'attuazione mediante getti continui indipendenti, quindi attuati singolarmente, introduce benefici esclusivamente con l'attuazione del getto inferiore ed in particolare permette una riduzione della drag di $\sim 4\%$. Risultati migliori, riduzione della resistenza di $\sim 12.7\%$, sono stati ottenuti combinando attuazione di getti inferiori e laterali.

L'obiettivo di questo lavoro di tesi è quello di controllare la scia generata da un corpo tozzo attraverso tecniche di intelligenza artificiale al fine di ridurre la resistenza e analizzare i risultati ottenuti per valutare come evolve la scia nella configurazione di ottimo, così da capire i meccanismi che portano alla riduzione della resistenza.

[Grandemange et al.](#) hanno condotto un'indagine sperimentale al fine di cercare di comprendere come il campo risulta influenzato dalla geometria e dalla ground clearance attraverso lo studio della bi-stabilità della scia. In particolare sono stati investigati due modelli con aspect ratio pari a 0.74 e 1.34, definito come $\frac{H}{W}$ dove H è l'altezza del modello e W è la larghezza del modello, per diversi valori di ground clearance, definita come $\frac{h}{W}$ dove h è la distanza dal piano inferiore. Questo studio ha dimostrato che nella condizione di $\frac{h}{W} > 0.12$ a seconda dell'aspect ratio si ha una condizione di bi-stabilità nella direzione verticale o laterale.

Un aspetto finora trascurato è quello relativo al consumo energetico per il controllo attivo, infatti questo richiede una spesa energetica per l'attivazione del getto. Quindi l'ottimizzazione aerodinamica non avrà come fine ultimo quello di ridurre la resistenza ma anche quello di cercare il punto d'ottimo nella condizione per cui la riduzione di resistenza è energeticamente conveniente, questo concetto sarà ripreso nel seguito.

Da quanto è noto in letteratura al momento della scrittura non ci sono applicazioni di tecniche AI, ed in particolare di *DRL*, per la riduzione della resistenza per valori di Reynolds alti, come nello studio in esame, e per applicazioni sperimentali.

Capitolo 2

Reinforcement Learning

Per *Machine learning* si intende una famiglia di metodi che permettono di compiere delle operazioni in modo automatico [Bishop]. E' possibile dividere questa famiglia di algoritmi in 3 categorie:

- *Unsupervised learning*: l'addestramento viene fatto su dati non classificati. Una applicazione tipica è il *pattern recognition*;
- *Supervised learning*: l'addestramento viene fatto su dati classificati. Una applicazione tipica è *classification* o *regression*;
- *Reinforcement learning*: l'addestramento viene fatto mediante un agente che interagendo con un ambiente è in grado di raggiungere un determinato obiettivo.

2.1 Concetti base del Reinforcement Learning

Il *Reinforcement Learning* (RL), poco frequente in italiano con *apprendimento per rinforzo*, rientra quindi negli algoritmi di machine learning. Quando si parla di RL si può pensare ad un uomo che impara a fare una nuova azione: inizialmente non riuscirà a compierla e farà dei tentativi durante i quali imparerà qualcosa, poi arriverà a compiere l'azione.

Risulta quindi di fondamentale importanza l'interazione con l'ambiente poichè è proprio questa a permettere di acquisire l'esperienza necessaria per compiere l'azione. Un altro fattore importante che permette all'uomo di imparare è avere ben chiaro quale è l'obiettivo finale, in questo modo potrà migliorare da tentativo a tentativo.

Di seguito si introducono alcuni concetti fondamentali per meglio comprendere quanto verrà esposto in seguito:

- *Policy*: definisce la modalità con cui l'agente si comporta in un determinato istante e quindi come questo seleziona un'azione. Si può vedere questa come

una mappatura tra stati dell'ambiente e le azioni da intraprendere quando si è in quel determinato stato. La politica è un elemento estremamente importante per un agente. Può essere deterministica, in questo caso dipende esclusivamente dallo stato, oppure stocastica, definita da una probabilità che una data azione a sia scelta per un determinato stato s . La policy può assumere forma più o meno semplice a seconda del caso: può essere una semplice funzione, una tabella oppure una rete neurale.

- *Reward*: come detto in precedenza rappresenta la ricompensa che l'agente riceve come conseguenza di un'azione. L'agente opera in modo tale da massimizzare questo obiettivo.
- *Modello*: un sistema in grado di rappresentare il comportamento di un dato ambiente. Quindi si può usare un modello per stabilire quale azione seguire
- *Value function*: tiene conto del contributo complessivo dei reward, quindi permette di eseguire una massimizzazione che tiene conto della storia dei reward. Infatti se il reward fornisce una ricompensa nel breve periodo, la value function fornisce l'informazione su ciò che è meglio fare nel lungo periodo. Infatti è possibile che uno stato generi una ricompensa bassa, ma questo può dar vita ad una combinazione di stati che complessivamente permette di ottenere un alto reward.

Il framework è quello su cui si basa il Reinforcement Learning: si sfruttano le interazioni di un agente con l'ambiente per far stabilire all'agente l'azione da intraprendere al fine di raggiungere l'obiettivo desiderato, nel caso in questione l'agente dovrà trovare una condizione di ottimo, andando a massimizzare una ricompensa. La ricompensa non è univoca ma varia a seconda del problema, inoltre lo stesso problema può avere definizioni diverse di ricompensa e la scelta di una piuttosto che dell'altra potrebbe rivelarsi determinante ai fini del raggiungimento di un migliore risultato.

L'ambiente costituisce il sistema con cui l'agente va ad interagire (i.e. per questo lavoro l'ambiente è il modello o in senso più generico la galleria), quindi l'ambiente è costituito da uno stato s che ne descrive la condizione in cui si trova. E' importante evidenziare come nel RL si usi il *trial-and-error experience* e quindi all'agente non è richiesto di conoscere alla perfezione tutto l'ambiente ma è richiesto che sappia interagire con esso in modo da ottenere informazioni e in seguito compiere delle azioni.[[Francois-Lavet et al.](#)].

Nell'interazione agente-ambiente l'agente può ottenere informazioni circa lo stato in cui si trova l'ambiente, può agire sull'ambiente attraverso un'azione a che produrrà una modifica di s in un istante successivo. In generale le azioni possono essere qualsiasi decisione che si vuole apprendere e gli stati possono essere qualsiasi cosa che si può conoscere e che potrebbe essere utile per prenderle.

Quando l'agente esegue un'azione riceve una ricompensa (*reward*). L'obiettivo dell'agente è quello di definire una politica che permette di stabilire quale azione eseguire per un determinato stato per ricevere la miglior ricompensa possibile. Come riportato in [Sutton and Barto](#) la regola generale che si segue è che tutto quello che non può essere cambiato arbitrariamente dall'agente è da considerarsi esterno ad esso e quindi parte dell'ambiente. Inoltre non è corretto assumere che generalmente l'agente non conosce nulla dell'ambiente, in alcuni casi l'agente può sapere tutto su come funziona un ambiente ma comunque il problema di apprendimento per rinforzo rimane complesso (esempio è il cubo di Rubik).

A fronte di queste considerazioni quindi un qualsiasi problema di RL può essere descritto mediante 3 segnali: uno che permette di rappresentare la base su cui intraprendere le azioni e cioè lo stato, uno per rappresentare le azioni intraprese e quindi le azioni ed un ultimo per definire lo scopo dell'agente, la ricompensa.

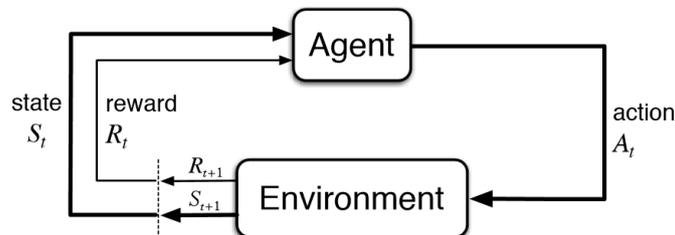


Figura 2.1: Schema Reinforcement Learning riadattato da [Sutton and Barto](#)

In figura 2.1 viene riportato uno schema di funzionamento dell'algoritmo di RL. Da qui è semplice evidenziare quanto detto in precedenza: questo algoritmo è tale da garantire che l'agente possa imparare in autonomia, diversamente dagli algoritmi di machine learning supervisionato dove l'addestramento avviene su un dataset di dati etichettati (tipico problema di classificazione). L'acquisizione dei dati per gli algoritmi di RL può avvenire offline, quindi i dati vengono utilizzati per l'addestramento, oppure online e quindi la raccolta dei dati avviene secondo il comportamento dell'agente. La possibilità di ottenere dati online è molto importante in quanto permette di ridurre sensibilmente la quantità di dati necessari per l'apprendimento. Inoltre questa metodologia permette di ottenere esperienza mediante una strategia *exploration - exploitation*.

Per esplorazione (*exploration*) si intende l'ottenimento di informazioni sull'ambiente; lo sfruttamento (*exploitation*) riguarda la massimizzazione del reward futuro; fissata la conoscenza fino a quel momento acquisita. Quindi nelle prime fasi dell'addestramento la conoscenza presente è bassa, nulla se si considera esattamente l'istante iniziale. Inizialmente sarà necessario che l'agente acquisisca conoscenza dall'ambiente, facendo quindi un compromesso tra imparare dall'ambiente oppure

perseguire la strategia più promettente. Infatti un approccio superficiale suggerirebbe che per massimizzare il reward l'agente dovrebbe sfruttare la conoscenza sulle azioni intraprese in precedenza e selezionare quelle che hanno portato una maggiore ricompensa. Ovviamente, però, è necessario scoprire quali sono le azioni che offrono la ricompensa maggiore. Ecco che si mette in risalto la necessità di esplorare. Quindi un approccio avido, dove si utilizza una politica che seleziona le azioni con più alta ricompensa futura, può non essere la scelta migliore. L'approccio tipicamente utilizzato prende il nome di $\epsilon - greedy$ e prevede un comportamento tipicamente avido ma con la selezione di una azione random tra tutte le azioni.

2.1.1 Processo decisionale di Markov

Per comprendere la tecnica del Reinforcement Learning dal punto di vista matematico è necessario introdurre il processo decisionale di Markov *MDP*. Questi processi rappresentano un formalismo matematico utilizzato per modellare problemi di decisione sequenziale e permettono di ottenere una rappresentazione dell'ambiente astratto o concreto dove un agente si può spostare ed operare.

Come detto in precedenza l'agente interagisce con l'ambiente andando ad imporre una azione, come conseguenza l'ambiente si porta in un nuovo stato e l'agente riceve una ricompensa. Questo processo viene ripetuto per diversi timestep ($t = 0, 1, \dots, T$, dal momento che si considera un tempo discretizzato), per ognuno dei quali l'ambiente si porterà in uno stato $s \in S_t$ (S_t rappresenta lo spazio degli stati). L'insieme degli stati in cui si porta l'agente si indica con traiettoria, questa può essere modellata mediante un processo markoviano che è una rappresentazione di possibili eventi.

Il processo markoviano matematicamente si esprime attraverso una coppia stato-probabilità (s, P) , dove P rappresenta, nell'accezione più semplice, una matrice di probabilità di transizione di stato:

$$P_{ss'}^a = \mathbb{P}[S_{t+1} = s' | S_t = s, A_t = a] \quad (2.1.1)$$

Il processo markoviano ha la proprietà di Markov e cioè è privo di memoria, questo significa che la probabilità di avere un dato stato s_{t+1} dipende esclusivamente dallo stato s_t . Questo significa che in un processo markoviano la time-history permette di raccogliere informazioni, ma che lo stato futuro è indipendente da questa.

Relativamente ai processi Markoviani è bene puntualizzare che non tutti i problemi di tipo *RL* seguono le regole definite da questi, infatti quando il problema è caratterizzato da un elevato grado di imprevedibilità la probabilità di transizione di stato non può essere mappata in modo preciso.

Il concetto di traiettoria può essere generalizzato aggiungendo a quanto prima detto le azioni e le ricompense, quindi:

$$s_0, a_0, R_0, s_1, a_1, R_1, s_2, a_2, R_2, \dots \quad (2.1.2)$$

Per un processo più complicato si introduce il concetto di *Markov reward process (RDP)* a cui è possibile associare un vettore (s, A, p, R, γ) , dove R è la funzione di ricompensa mentre γ è il *discount rate*. Inoltre è possibile definire il *total discounted reward* per un dato time-step t come:

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \quad (2.1.3)$$

Quando non si a che fare con problemi episodici (problemi che hanno una naturale conclusione come vittoria o sconfitta in una partita) questo concetto sostituisce quello classico di ricompensa e quindi l'agente opera in modo tale da selezionare le azioni in modo che il *total discounted reward* venga massimizzato.

Il *discount rate* riveste un ruolo importante nell'apprendimento, infatti se $\gamma = 0$ l'agente risulta preoccuparsi esclusivamente di massimizzare le ricompense immediate (si parla di agente miope). Con γ che tende ad 1 invece l'agente acquisisce un senso di lungimiranza, questo generalmente prolunga i tempi di convergenza con dei risultati dal punto di vista dell'ottimo più affidabili.

2.1.2 Policies and Value Functions

In precedenza si è introdotto il concetto di *value function* e si è detto che questa permette di stimare quanto sia buono eseguire una data azione in un dato stato, dove per quanto è buono deve essere inteso in termini di reward. Come già illustrato la ricompensa dipende dalle azioni intraprese e quindi la *value function* è definita a partire da particolari modi di agire perciò dipende dalla politica intrapresa.

Per una data politica e per un dato stato esiste una condizione di coerenza tra il valore s dello stato e il valore del possibile stato futuro:

$$v_{\pi}(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a) [r + \gamma v_{\pi}(s')] = \mathbb{E}_{\pi}[G_t|S_t = s] \quad (2.1.4)$$

questa è anche definita equazione di Bellman [1].

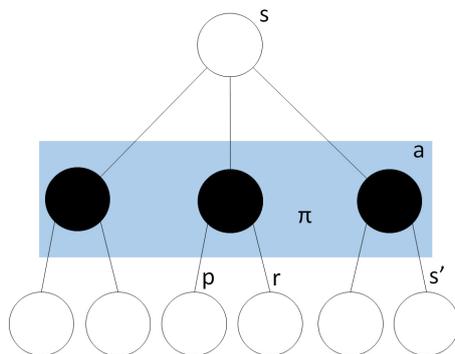


Figura 2.2: Schema Value Function

In figura 2.2 è riportato uno schema per la funzione v_π . Considerando che l'agente si trovi nello stato iniziale s , questo può intraprendere, basandosi su una data politica π , una data azione a . Di seguito si genera un nuovo stato s' , che dipende dalla dinamica p , e si avrà un reward r .

Questo diagramma permette di comprendere meglio l'equazione di Bellman 2.1.4 che è alla base di molte tecniche che permettono di calcolare, approssimare ed imparare v_π . Quindi 2.1.4 definisce che il valore dello stato iniziale deve essere uguale al valore scontato dello stato successivo con l'aggiunta della ricompensa ricevuta durante la traiettoria seguita.

Riassumendo quindi si può dire che la politica è una regola stocastica con cui l'agente seleziona le azioni in funzione dello stato in cui l'ambiente si trova, questo ha come scopo massimizzare il reward che riceve nel tempo. Mentre le *value function* per un dato stato, o comunque per una coppia stato-azione, forniscono il rendimento atteso. Una politica si dice ottimale rispetto ad un'altra quando la *value function* che si ottiene per essa risulta essere maggiore.

In analogia a $v(s)$ è possibile definire la *action-value function* che fornisce il valore che scaturisce dal prendere una azione a per uno stato s sotto una politica π e viene indicata come $q_\pi(s, a)$. Anche per quest'ultima funzione è possibile individuarne una ottimale (in letteratura le funzioni ottimali vengono definite con il pedice $*$).

L'equazione di Bellman, ottima per q_* , si presenta nella forma:

$$\begin{aligned} q_*(s) &= \max_a \mathbb{E}[R_{t+1} + \gamma \max_{a'} q_*(S_{t+1}, a) | S_t = s, A_t = a] \\ &= \sum_{s', r} p(s', r | s, a) [r + \gamma \max_{a'} q_*(s', a)] \end{aligned} \quad (2.1.5)$$

Se si ha a disposizione un ambiente completamente definito allora è possibile ricorrere alla programmazione dinamica *Dynamic Programming* che in casi complessi può risultare particolarmente onerosa dal punto di vista computazionale. Per *DP*

si intende un insieme di metodi matematici e algoritmi attraverso cui è possibile affrontare un problema di ottimizzazione. Il framework di questi algoritmi si basa sulla scomposizione del problemi in sotto-problemi di minore complessità e sull'assunzione che la soluzione ottima del problema dipende dalla soluzione ottimale dei sotto-problemi.

Diversamente, se l'ambiente non è completamente noto, si fa ricorso agli algoritmi di *Reinforcement Learning*. Nei metodi *DP* la conoscenza dell'ambiente è necessaria in quanto la stima del valore di uno stato viene eseguita partendo dalla stima dei valori assunti negli stati successivi, sfruttando la proprietà detta *bootstrapping*.

2.2 Rete Neurale

Le reti neurali sono dei modelli bio-ispirati. Sono costituiti da unità funzionali detti neuroni. Le reti possono essere più o meno complesse, si parla di multi-layer (figura 2.3b), dove per layer si intende un insieme di neuroni. I neuroni sono connessi tra loro attraverso collegamenti pesati. In input al neurone dello strato successivo arriva il segnale ottenuto sommando ogni input del neurone ad esso collegato ognuno moltiplicato per il relativo peso ed eventualmente si aggiunge una costante (detta bias). In una seconda fase al valore ottenuto si applica la funzione di attivazione. (figura 2.3a). Per una miglior comprensione si fa riferimento allo schema in figura 2.3b dove:

- i : i-esimo neuro del layer;
- j : j-esimo layer della rete;
- x_i : input i-esimo della rete neurale;
- $*_0$: bias unit
- $a_i^{(j)}$: valore assunto dall'i-esima unità del j-esimo layer;
- g : funzione di attivazione;
- $\Theta_{i,k}^{(j)}$: connessione pesata tra il neurone i-esimo del layer j+1 ed il neurone k-esimo del layer j;

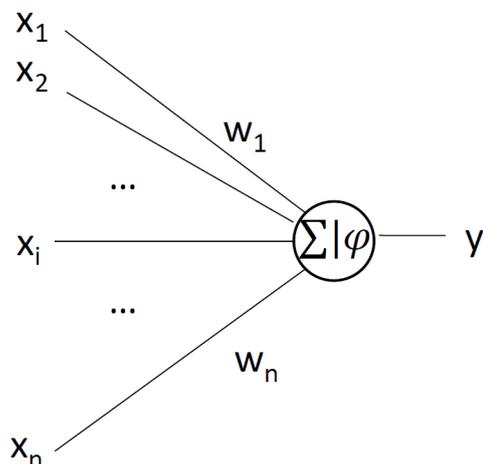
Quindi in 2.3b viene rappresenta una rete a due strati interni, con rispettivamente 3 ed 1 neurone ciascuna. Come descritto in precedenza esiste una relazione matematica con cui è possibile definire il valore che ciascun neurone di un layer successivo all'input assume:

$$a_1^{(2)} = g(\Theta_{10}^{(1)}x_0 + \Theta_{11}^{(1)}x_1 + \Theta_{12}^{(1)}x_2 + \Theta_{13}^{(1)}x_3) \quad (2.2.1)$$

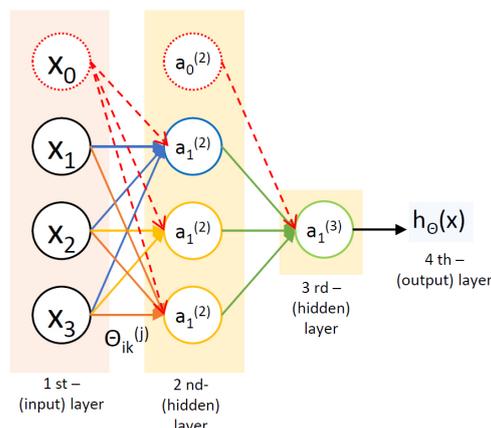
$$a_2^{(2)} = g(\Theta_{20}^{(1)}x_0 + \Theta_{21}^{(1)}x_1 + \Theta_{22}^{(1)}x_2 + \Theta_{23}^{(1)}x_3) \quad (2.2.2)$$

$$a_3^{(2)} = g(\Theta_{30}^{(1)}x_0 + \Theta_{31}^{(1)}x_1 + \Theta_{32}^{(1)}x_2 + \Theta_{33}^{(1)}x_3) \quad (2.2.3)$$

$$h_{\Theta}(x) = a_1^{(3)} = g(\Theta_{10}^{(2)}a_0^{(2)} + \Theta_{11}^{(2)}a_1^{(2)} + \Theta_{12}^{(2)}a_2^{(2)} + \Theta_{13}^{(2)}a_3^{(2)}) \quad (2.2.4)$$



(a) Schema funzionamento singolo neurone



(b) Schema rete multi-layer

Prima di arrivare al neurone però, come detto, la somma pesata viene fornita ad una funzione di attivazione; la necessità di introdurre una funzione di attivazione trova la sua origine nel dover approssimare con la rete una qualsiasi funzione e quindi è necessario introdurre un termine di non linearità che senza funzione di attivazione non sarebbe presente. Una rete neurale senza una funzione di attivazione sarebbe essenzialmente un regressore lineare.

Quella appena descritta è una architettura del tipo feed-forward: c'è una totale assenza di cicli ed ogni neurone di un layer è connesso con tutti gli input dei neuroni al livello successivo.

L'addestramento di una rete neurale (training) consiste nel determinare il peso tra le connessioni dei vari neuroni. L'algoritmo alla base prende il nome di backpropagation. L'utilizzo di questo algoritmo rende necessario garantire la differenziabilità al primo ordine della funzione di attivazione, in quanto per essere applicato bisogna conoscere la derivata prima dell'errore di previsione prima di aggiornare i pesi del modello. Per meglio comprendere il ruolo della funzione di attivazione si fa un confronto tra diverse funzioni di attivazione:

- *step function*: in questo caso il neurone viene attivato o disattivato a seconda che l'input sia maggiore o minore di una data soglia

$$f(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases} \quad (2.2.5)$$

Il gradiente di questa funzione è nullo e quindi limita l'algoritmo di backpropagation: il calcolo del gradiente serve per aggiornare i pesi durante il processo di backpropagation, essendo nullo il gradiente allora i pesi non si aggiornano.

- *funzione lineare*: in questo caso il neurone viene attivato proporzionalmente all'ingresso

$$f(x) = \alpha \cdot x \quad (2.2.6)$$

Diversamente dal caso *step function* il gradiente non è nullo ma è una costante e quindi indipendente da x . Quindi con la funzione di attivazione lineare ci sarebbe un aggiornamento dei pesi durante l'applicazione dell'algoritmo di backpropagation ma il fattore di aggiornamento sarebbe il medesimo, ne segue che la rete non si addestrerebbe in modo adeguato nel caso di problemi complessi.

- *sigmoide*: in questo caso la funzione è non lineare e può essere interpretata come un filtro in quanto trasforma un qualsiasi input in un output compreso tra 0 ed 1.

$$f(x) = \frac{1}{1 + e^{(-x)}} \quad (2.2.7)$$

In primis è importante notare come la funzione sigmoide risulta essere differenziabile e la sua derivata continua. Inoltre la non linearità in questo caso permette di concludere che la presenza di più neuroni con funzione di attivazione sigmoide allora l'output potrà essere non lineare, ne segue la possibilità di modellare problemi molto complessi.

Qui sono stati riportati solo 3 esempi di funzione di attivazione, ne esistono molti altri, con lo scopo di comprenderne il ruolo e l'importanza di introdurre la non linearità ma di avere una funzione comunque derivabile.

In breve l'algoritmo di backpropagation si può riassumere seguendo la figura 2.4: inizialmente viene preso un dato di input e questo procede dal layer di input fino a quello di output, passando per gli hidden-layer, ottenendo il dato in output. Il dato di output viene confrontato con il valore desiderato, si suppone per semplicità un problema supervisionato, in modo tale da definire l'errore. Noto l'errore si ritorna verso gli hidden-layer in modo tale da correggere i pesi ed ottenere un minor errore in futuro.

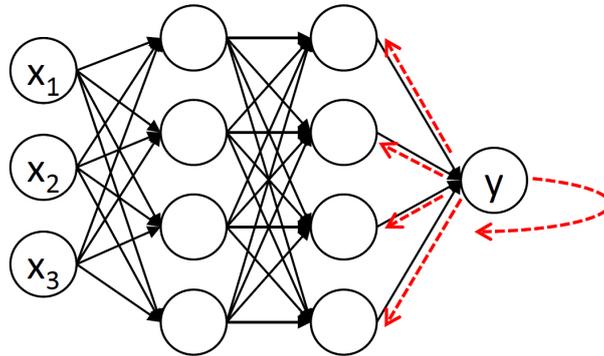


Figura 2.4: Schema Algoritmo Backpropagation

Oltre all'architettura *Feed Forward Neural Network* esiste l'architettura *Recurrent Neural Network*: in questa esistono dei cicli all'interno della rete neurale e quindi è possibile che i valori di uscita di un livello superiore vengano utilizzati come ingresso per un layer di livello inferiore, introducendo quindi una retroazione. Queste reti si prestano particolarmente bene per problemi in cui esiste una forte dipendenza temporale come ad esempio: *Speech-Recognition*, *Natural Language Processing* e *Time Series Forecasting*.

Un'altra caratteristica delle reti finora illustrate è quella di avere i neuroni in configurazione *fully connected*: ogni neurone è connesso ad ogni neurone dello strato precedente ed ogni connessione è caratterizzata da un suo peso. In questo modo si riesce ad avere un modello generico con assenza di supposizioni di dati. Un esempio di livelli con connessioni *non fully connected* è il *convolutional layer*: la connessione è tra un neurone e pochi neuroni dello strato precedente e qui viene utilizzato lo stesso set di pesi per ogni neurone. Questa connessione è tipicamente utilizzata per i dati in cui ci sono immagini dove quindi si hanno dati con caratteristiche spaziali. Questo permette di giustificare la presenza di poche connessioni tra neuroni e di pesi tutti uguali, infatti le connessioni locali trovano riscontro con l'estrazione dai dati di caratteristiche spaziali locali, mentre i pesi tutti uguali rappresenta la probabilità che questa caratteristica locale si trovi in una posizione qualsiasi dell'immagine. La presenza di un numero minore di connessioni fa intuire anche come sia necessaria una minor potenza computazionale.

2.3 DRL - Deep Reinforcement Learning

La *value function* può essere rappresentata in vario modo come ad esempio una tabella. Nel caso in cui lo spazio dello stato e quello delle azioni è molto grande non è possibile questa scelta e risulta necessario introdurre un'approssimazione per

la *value function* attraverso delle funzioni i cui parametri variano in base al reward che segue un determinato stato o stato-azione. Nel *DRL* la funzione di *function approximator* viene svolta da una rete neurale profonda.

2.3.1 Deep Q-Learning

Q-Learning

Questa tecnica si basa sul metodo a differenza temporale e ha come obiettivo principale quello di trovare una approssimazione ottimale, per ogni $s \in S$ ed ogni $a \in A$, della funzione $q(s, a)$ indipendentemente dalla politica π che si sta seguendo, infatti Q-learning si basa su di un metodo off-policy che permette di separare la politica di comportamento da quella di apprendimento [Jang et al.].

La $Q(s, a)$ viene aggiornata sulla base dell'equazione di Bellman ottima (2.1.5): i valori della *action-value function* vengono aggiornati in conseguenza del reward assunto al passo successivo e il massimo reward cumulativo atteso per il passo seguente. L'equazione per Q-value è la seguente:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \overbrace{[r + \gamma \max_{a'} Q(s', a') - Q(s, a)]}^{\text{Bellman equation}} \quad (2.3.1)$$

Al tempo t si considera di avere uno stato s , di compiere una data azione a seguendo una data politica e di conseguenza ci si porta in uno stato s' ed un reward r . Sfruttando l'equazione 2.3.1 si aggiorna la *action-value function*. I termini in 2.3.1 sono:

- $\max_{a'} Q(s', a')$: massimo reward dallo stato s' , mentre $a' \in A_t$
- α (*learning rate*): è un valore compreso tra 0 ed 1. Questo è un parametro che rappresenta quanto pesantemente vengono variati i valori attuali della *action-value function*: un valore prossimo a 0 indica un aggiornamento minimo, questo può comportare un lungo processo di allenamento che potrebbe bloccarsi, mentre un valore di 1 indica un aggiornamento pesante, può comportare l'apprendimento di una serie di pesi non ottimale troppo velocemente o un processo di allenamento instabile.

Deep Q Network

Come detto in precedenza la presenza di un numero di stati e di azioni possibili elevato rende l'algoritmo *Q-Learning* molto oneroso dal punto di vista computazionale e la rappresentazione della *action-value function* in forma matriciale molto complessa. Sorge quindi la necessità di avere un modo di rappresentare in forma compatta la matrice, si introduce la *value function approximator*: $\hat{q}(s, a; \theta) \approx q_\pi(s, a)$, dove θ sono i parametri di approssimazione.

È possibile utilizzare una qualsiasi funzione di approssimazione, quella solitamente utilizzata è la rete neurale.

2.3.2 Policy Gradient Methods

Il problema di *RL* può essere risolto mediante il metodo di *policy gradient methods*, in questo caso la politica è modellata direttamente dalla rete neurale. Esistono altre metodologie come ad esempio la *Q-learning* ma queste risultano avere proprietà meno buone in termini di convergenza e di stabilità e risulta più adatta per i problemi relativi al controllo continuo.

Nei metodi *PG*, essendo la politica modellata direttamente mediante la rete neurale, il problema RL si trasferisce nell'individuazione dei pesi della rete neurale che possono essere rappresentati da Θ e la politica con π_{Θ} .

$$R_{max} = \max \mathbb{E} \left[\sum_{t=0}^H R(s_t) | \pi_{\Theta} \right] \quad (2.3.2)$$

Quest'ultima equazione può essere risolta mediante il metodo *gradient descent*, in particolare seguendo metodi sofisticati come *gradient batch descent*.

Ad ogni passo temporale si ottiene una distribuzione di probabilità dell'azione che permette di individuare quale azione è quella ottimale. Durante l'addestramento si va ad intraprendere l'azione che viene indicata da questa distribuzione, significato della presenza di esplorazione nell'algoritmo di apprendimento. In modalità di predizione invece viene selezionata l'azione con maggiore probabilità e quindi si annulla l'esplorazione. Con DRL si intende l'utilizzo di una rete neurale profonda (Deep Neural Network) per valutare la value function o la policy nel RL. Esistono diversi algoritmi di DRL, nel seguente lavoro si considera di utilizzare il PPO (Proximal Policy Optimization). Come si intuisce dal nome, questo algoritmo appartiene alla famiglia dei policy gradient methods dove la politica è direttamente modellata dalla rete neurale. Seguendo il lavoro di [Rabault et al.](#) viene scelto PPO in quanto risulta più semplice dal punto di vista matematico, più veloce e richiede uno scarso metaparameter tuning, cioè un numero minore di parametri che risultano rilevanti al fine del comportamento dell'agente.

In letteratura è possibile trovare diverse procedure da seguire al fine di ottenere un miglior comportamento dell'agente. Ad esempio in [Mnih et al.](#) viene utilizzato una *replay memory buffer*, memoria FIFO (*First In First Out*) di dimensione fissa, che permette di memorizzare i dati campionati, in questo modo nella fase di training viene utilizzato un sottoinsieme casuale del buffer immagazzinato. In questo modo si esegue l'addestramento su un set di dati statisticamente indipendenti e ciò risulta essere vantaggioso ai fini della convergenza ed evita di rimanere bloccati in un minimo locale. Questo è abbastanza frequente nei problemi sperimentali dove abbiamo l'assenza di correlazione temporale, data la limitazione nella frequenza di sampling.

Inoltre bisogna considerare il problema dell'adattamento dell'agente ai dati (overfitting) e quello relativo alla possibilità che si verifichino delle condizioni fortuite particolarmente vantaggiose a cui la rete si adatta. Questo è possibile prevenirlo mediante il *gradient clipping*. Questa tecnica introduce una soglia predeterminata al gradiente, quindi quando le norme sui gradienti superano il valore prefissato vengono riadattate. Nel seguente lavoro vengono utilizzate la libreria Tensorforce [26] e Tensorflow. Seguendo le linee guida della libreria Tensorforce si implementa il metodo PPO con una rete neurale costituita da 2 strati densi fully connected.

2.3.3 Tensorforce Agent

Nel seguente lavoro di tesi come agente è stato selezionato *Tensorforce Agent* definito all'interno della libreria [Tensorforce](#). Come riportato nella libreria questo è un agente base ma è altamente configurabile che permette di rappresentare una vasta classe di agenti per il *DRL*. Essi agiscono con una politica parametrizzata da una rete neurale e sfruttano moduli di memoria per eseguire gli aggiornamenti. In particolare, questi agenti sono formulati in modo tale da implementare politiche diverse per migliorare la stima dei pesi della rete. In [2.3.3](#) si riporta la class estratta direttamente dalla libreria. Per una migliore comprensione dell'agente si fa riferimento a [Tensorforce](#).

Listing 2.1: class Tensorforce Agente, fonte: [Tensorforce](#)

```
// Agente Tensorforce
class tensorforce.agents.TensorforceAgent(states, actions, update,
    optimizer, objective, reward_estimation, max_episode_timesteps=None,
    policy='auto', memory=None, baseline=None, baseline_optimizer=None,
    baseline_objective=None, l2_regularization=0.0,
    entropy_regularization=0.0,
    state_preprocessing='linear_normalization', exploration=0.0,
    variable_noise=0.0, parallel_interactions=1, config=None,
    saver=None, summarizer=None, tracking=None, recorder=None, **kwargs)
```

I parametri utilizzati per questo lavoro di tesi sono quelli base suggeriti nella libreria [26].

Un possibile proseguimento del seguente lavoro potrebbe essere proprio quello di analizzare il comportamento dell'agente al variare dei parametri riportati in [2.3.3](#) piuttosto che variare la tipologia di agente.

Capitolo 3

Set up sperimentale

La prova viene eseguita utilizzando una galleria del vento a circuito aperto dove il flusso viene accelerato mediante due eliche poste a monte della camera di prova. Subito a valle delle due grandi eliche viene posto un diffusore rapido e la camera di ristagno. La camera di prova ha una sezione rettangolare (altezza pari a 0.9 m e larghezza pari a 1.2 m) ed una lunghezza di 6.5m, il flusso viene convogliato mediante un convergente. La test section presenta un angolo di divergenza pari a 1% così da considerare la crescita dello strato limite legato alla presenza delle pareti.

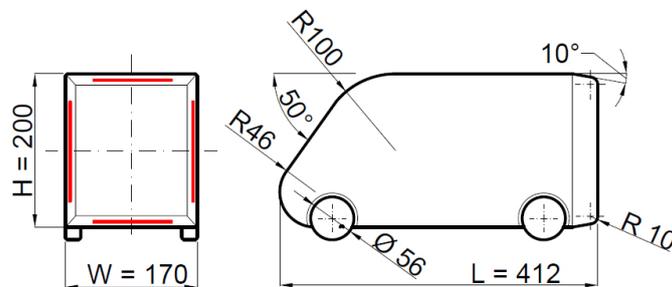


Figura 3.1: Schema del modello, la linea rossa continua rappresenta gli slot di iniezione del sistema di controllo attivo del flusso - Fonte : [20]

Il modello utilizzato è lo stesso utilizzato per studi precedenti (Cerutti et al.), un modello in scala 1 : 10 di un square-back road vehicle. Il modello è caratterizzato da una sezione con altezza $H = 0.2m$ ed una larghezza $W = 0.17m$. Inoltre la parte posteriore presenta un angolo di inclinazione di 10° nella direzione della scia (come indicato in figura 3.1) questo permette di enfatizzare l'effetto del controllo attivo come illustrato in Barros et al..

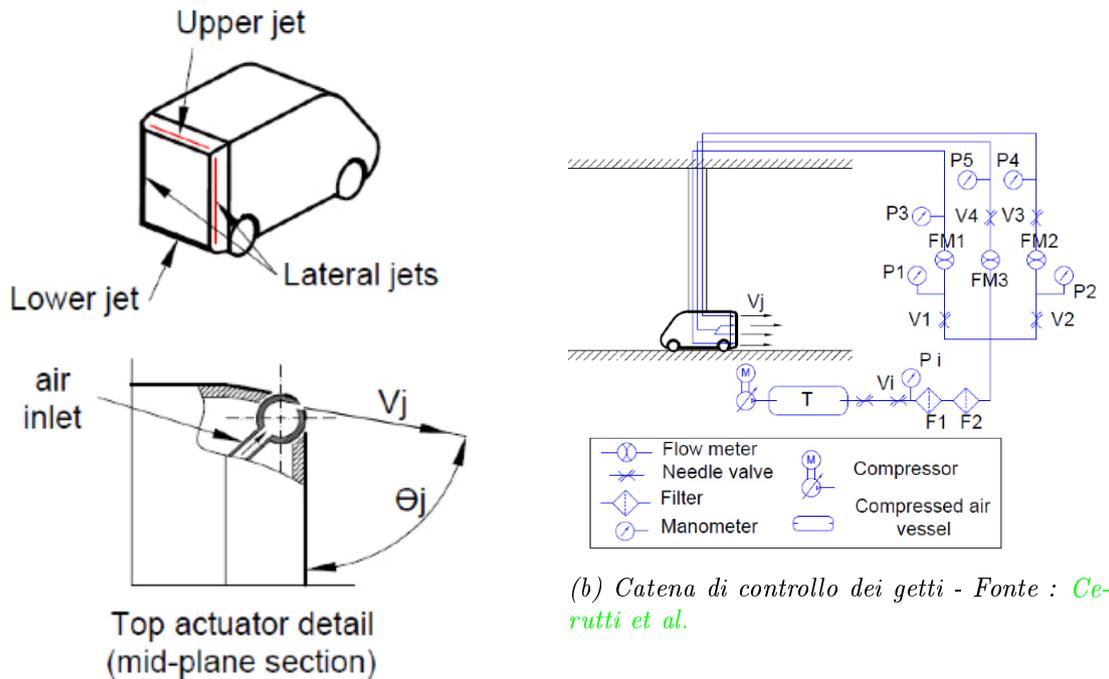


Figura 3.2: Sistema di attuazione e controllo getto

La sperimentazioni in galleria del vento per l'ambito automotive presenta, principalmente, due grandi problematiche: le ruote fisse e non in rotazione (effetto importante ai fini della resistenza in quanto si stima che circa il 25% della resistenza sia imputabile ad esse) e il secondo effetto è la presenza di una velocità relativa tra la corrente e la parete inferiore della galleria, che simula l'effetto del manto stradale. Quest'ultimo effetto non è presente in realtà e questa differenza introduce uno strato limite con conseguente diversa distribuzione di pressione e sforzi di attrito nel sottoscocca con conseguenti ripercussioni sulla portanza e in modo quasi trascurabile sulla resistenza [Bearman et al.].

Per risolvere il primo problema si utilizzano dei sistemi in grado di mettere in rotazione le ruote mentre per il secondo si procede utilizzando un sistema di aspirazione a monte (a volte si utilizzano anche dei sistemi distribuiti) del modello così da aspirare lo strato limite. Un'altra soluzione consiste nell'utilizzare un moving belt, cioè un sistema simile ad un tapis roulant: una superficie posta in rotazione ad una velocità pari a quella della corrente indisturbata.

Nell'esperimento in esame non si considerano entrambe delle soluzioni in quanto il sistema moving belt non è disponibile, mentre sebbene il sistema di aspirazione sia utilizzabile si è visto mediante misurazioni che l'effetto che si vuole correggere è trascurabile.

Nella parte superiore del modello è posto un supporto con carenatura aerodinamica che si estende fino alla parte superiore della galleria (figura 3.2b), questo ha una sezione cava e al suo interno passano le linee pneumatiche per le misure di pressione, per l'attuazione dei getti, il cavo ethernet di connessione con la Smart-Zoc e la connessione con la cella di carico. La carenatura aerodinamica viene introdotta per ridurre al minimo i problemi legati all'interferenza, infatti è realizzato con una sezione alare con $t/c = 0.2$ e $c/W = 0.07$.

L'area frontale del modello è pari al 3.1% della sezione trasversale della galleria del vento, questo valore si porta al 4.5% considerando il puntone.

3.1 Sistema di attuazione getti

Come riportato in figura 3.2a nella parte posteriore del modello sono presenti 4 slot dove sono posizionati gli attuatori: in totale sono presenti quattro attuatori, uno per ogni lato. Il singolo attuttore è costituito da un cilindro con una fessura rettangolare realizzata sulla sua lunghezza (figura 3.3) con una larghezza di 1mm . La scelta di questa geometria e configurazione permette di orientare il getto (come rappresentato in figura 3.2a). Sebbene ci sia questa possibilità per lo studio in oggetto viene mantenuto un angolo fisso pari a 65° . Gli slot rappresentati in rosso in figura 3.1 hanno una lunghezza rispettivamente di 104mm , nel caso di slot superiore ed inferiore, e di 132mm nel caso di slot laterale. In figura 3.2b è riportato il sistema di alimentazione dei getti, la regolazione della portata avviene mediante le valvole (V) e i flussimetri (FM).

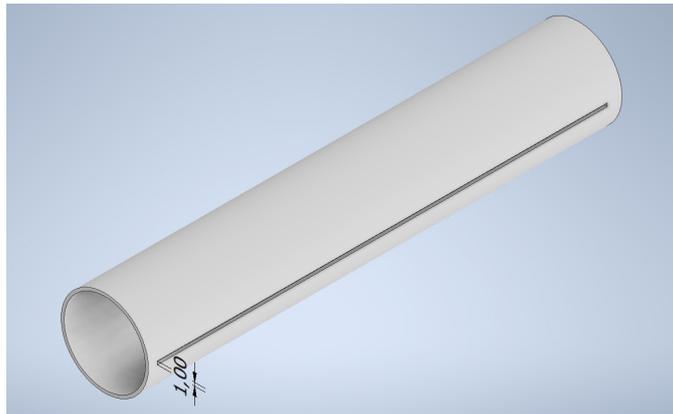


Figura 3.3: Schema cilindro attuttore

In [20] è stato verificato che la velocità del getto è uniforme per almeno il 95% della lunghezza della fessura.

3.2 Sistema di misura

Il modello è strumentato in modo tale da acquisire informazioni sulla pressione statica mediante trasduttori di pressione multi-input (per la misura di pressione statica) e capsule microfoniche per la misura delle fluttuazioni di pressione. In particolare i segnali relativi alle pressioni statiche vengono acquisiti mediante una Scanivalve 33 connessa ad una Smart Zoc 100, sistema che permette di acquisire simultaneamente fino a 64 canali. E' possibile scegliere diversi valori di frequenza di campionamento, nel caso in esame è stato scelto un valore pari a $f = 100Hz$. Il trasduttore ha un fondo scala di $\pm 2.5 kPa$ ed una accuratezza di $0.15\%FS$. I 64 canali sono connessi mediante delle connessioni pneumatiche (tubi con diametro interno pari a $\Phi = 1mm$) alle pressure taps. Queste sono distribuite in questo modo: 31 pressure taps nella parte posteriore del modello e 33 pressure taps distribuite tra superficie laterale e frontale del modello.

Per quanto riguarda l'acquisizione della pressione fluttuante si fa ricorso a 16 capsule microfoniche (capacitive) aventi un diametro esterno di $9.8 mm$ ed un elemento sensibile con diametro a $1 mm$. Le 16 capsule sono così distribuite: 12 microfoni sono posti sulla faccia posteriore e i restanti 4 sono stati posti sulle quattro facce laterali ad una distanza di $10 mm$ dal bordo di uscita. Tali microfoni hanno una risposta piatta nel range di frequenze $0.005 - 13 kHz$ ed una sensibilità di $-60 \pm 3 dB$. Come riportato in [Cerutti et al.](#) tutti i sensori sono stati calibrati usando una *Brüel & Kjær probe*, seguendo quanto riportato in [Sardu et al.](#). Sia per la misurazione che per la calibrazione è stata utilizzata una configurazione di tipo *pinhole* e il segnale elettrico è stato filtrato per eliminare i contributi spuri, come suggerito da [Cerutti et al.](#)

Come sopra descritto la misura della drag viene eseguita mediante una cella di carico, in particolare una cella *Dacell UU-K002* con un fondo scala $F.S. = 2 Kg_f$ e una precisione pari a $0.002\%FS$. Il segnale della cella di carico viene campionato utilizzando uno chassis NI-cDAQ con un modulo convertitore *A/D NI-9215* dedicato. Il segnale elettrico della cella di carico viene convertito in resistenza attraverso una mappatura di calibrazione.

La misura della velocità a monte viene eseguita mediante il valore del rapporto di pressioni nel convergente. Affinché questo sia possibile è stata effettuata una calibrazione mediante l'introduzione di un tubo di Pitot, che finita la calibrazione è stato rimosso, all'ingresso della test section. La calibrazione ha permesso di ottenere una relazione in grado di convertire la misura del salto di pressione nel convergente in una misura della velocità a monte.

Capitolo 4

Implementazione modello

4.1 Ambiente

Nel capitolo 2 si è introdotto il concetto di ambiente. Per il *RL* questo aspetto è estremamente importante e rappresenta una delle maggiori difficoltà per l'implementazione della metodologia, in quanto gli altri aspetti grazie a librerie preesistenti sono facilmente implementabili e non richiedono una conoscenza specifica dell'argomento.

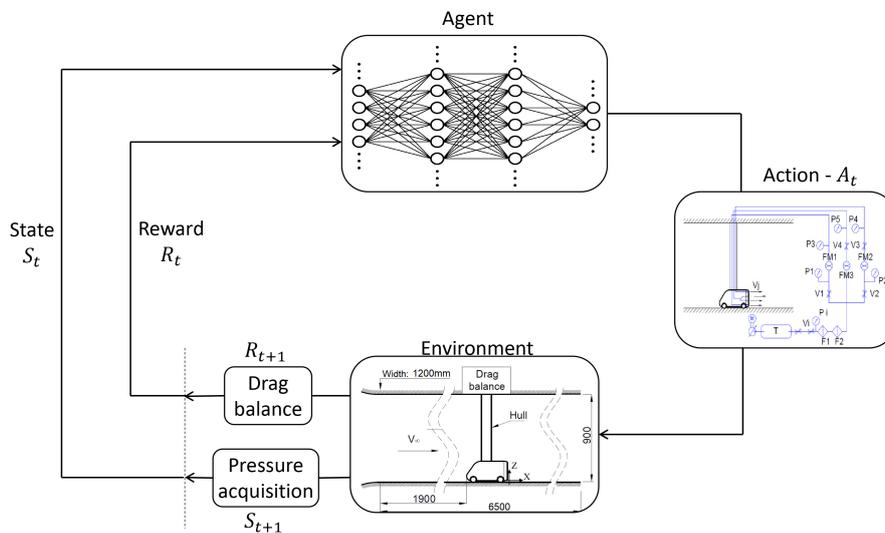


Figura 4.1: Schema concettuale RL applicato al problema - Lo stato è individuato dalle misure di pressione, quindi le prese di pressione localizzate sul modello, l'azione è determinata dal getto invece il reward dalla misura della cella di carico.

Per il lavoro in esame dall'ambiente ci si aspetta:

- *ottenimento dei dati*: i dati che vengono rilevati dai sensori disposti sul modello e all'interno della galleria (questi saranno meglio descritti in seguito) permettono di descrivere lo stato in cui si trova il sistema. In particolare lo stato andrà a fornire una descrizione della distribuzione delle pressioni intorno al modello e della velocità della corrente che interagisce con il modello. Inoltre per ottenere l'informazione relativa al reward è necessario interrogare la cella di carico che fornirà una misura della resistenza generata dal modello.
- *controllo dei getti*: l'agente deve poter controllare e quindi attuare i getti posti nella base posteriore del modello (anche di questi si discuterà in seguito).

Seguendo quanto illustrato in [Tensorforce](#) si implementa l'ambiente in *Python 3.6* andando a definire le seguenti funzioni (di seguito verranno illustrate in modo descrittivo):

- *Reset*: questa funzione ha il compito di resettare l'ambiente e quindi dovrà portare i getti in una configurazione di frequenza nulla e tensione pari a quella minima corrispondente alla configurazione unforced. Ottenuta questa condizione la funzione attende un tempo sufficientemente lungo, così da garantire che si arrivi verso la stazionarietà del flusso, dopo di che si effettua l'acquisizione dei dati al fine di ottenere l'informazione sullo stato e sulla resistenza aerodinamica, quest'ultima sarà il punto di partenza per la rete.
- *Execute*: questa funzione rappresenta il cuore dell'algoritmo. Infatti riceve in input l'azione (per il lavoro seguente per azione si intende la frequenza e la tensione di funzionamento dei getti posti nella base del modello) e aziona il getto in questa condizione. Dopodiché si effettua un'acquisizione per un tempo sufficientemente lungo così da avere una convergenza statistica delle misure di pressione. Tale tempo di campionamento è scelto come trade off tra la convergenza e il tempo di misura e training della rete. Con questa informazione è possibile andare a determinare lo stato e calcolare il coefficiente di resistenza da cui deriverà il reward.

Inoltre è particolarmente importante riportare la definizione di *reward* che si è deciso di assegnare per il problema in esame. Si è definito:

$$R = -(Cd - Cd_0) \quad (4.1.1)$$

dove Cd_0 è il valore del coefficiente di resistenza nella condizione naturale, senza forcing. E' importante osservare che questa definizione prescinde da considerazioni energetiche e si pone come unico obiettivo quello di ridurre il coefficiente di resistenza. Infatti per come definito in eq. 4.1.1 si ha che massimizzare il reward implica una riduzione di Cd .

Un possibile sviluppo di questo lavoro potrebbe riguardare l'addestramento di un

agente variando la definizione del reward, inserendo dei riferimenti energetici. In questo modo si può garantire di cercare non la condizione di minor resistenza ma quella che lo fa con miglior efficienza e questo per quanto riportato nel capitolo 1 è particolarmente importante.

4.1.1 Implementazione ambiente

La scelta di voler utilizzare una tecnica di AI per la ricerca di una condizione d'otimo richiede la necessità di definire un'interfaccia tra il modello e il codice. Questo costituisce un importante sfida in quanto è richiesta la comunicazione tra diversi linguaggi di programmazione. Infatti mediante **LabView** avviene l'acquisizione dei dati e il pilotaggio dei getti mentre **Python** è stato utilizzato per il post-processing e per la definizione di agente, ambiente e algoritmo di RL.

Per interfacciare i due diversi linguaggi di programmazione si è pensato di esportare dal progetto LabView una libreria DLL in modo da richiamare questa da Python. Di seguito si riporta uno schema di interfaccia:

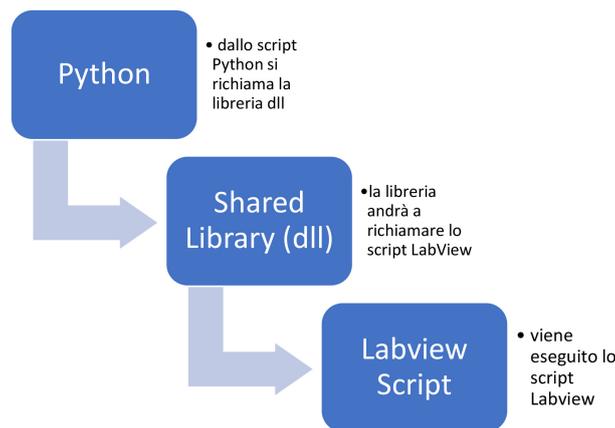


Figura 4.2: Schema di interfaccia tra Python e Lab View

DLL è acronimo che indica dynamic-link library cioè una libreria dinamica che durante l'esecuzione viene caricata in modo dinamico e quindi non è connessa staticamente ad un file eseguibile durante l'esecuzione.

Per interagire con il modello e la galleria del vento è stato necessario generare 3 diverse librerie dll: la prima libreria si occupa di comunicare con il modulo di acquisizione National Instrument, la seconda libreria permette la comunicazione con le elettrovalvole che comandano i getti mentre la terza libreria garantisce la comunicazione con la Smart-Zoc.

Comunicazione con sistema di acquisizione dati

Al modulo National Instrument sono connesse le 23 capsule microfoniche, la cella di carico e il salto di pressione della contrazione. L'interfaccia con il computer avviene mediante connessione USB. Come riportato in figura 4.3 tramite python viene chiamata la libreria DLL dove si richiama la funzione riportata in 4.1.

Listing 4.1: DLL library per acquisizione modulo National Instrument

```
// libreria dll per acquisizione da LabView
void Read3(int32_t Start, double SamplingFrequency, int32_t Samples,
char saveOnFile[], int32_t SaveMics2)
```

Delle variabili presenti nella funzione è importante evidenziare:

- *SamplingFrequency* : rappresenta la frequenza di campionamento
- *Samples* : il numero di campioni
- *saveOnFile* : il nome del file *.tdms dove viene salvato l'output. Uno schema dell'output viene riportato in 4.4

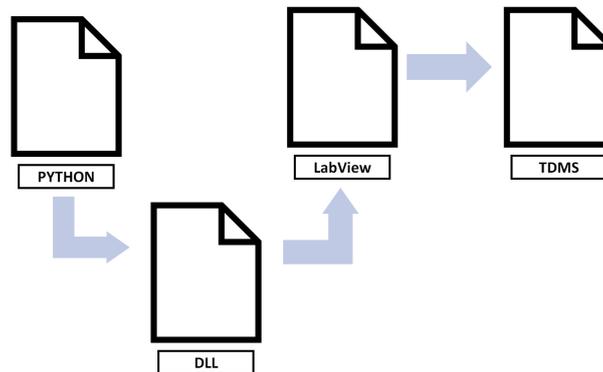


Figura 4.3: Schema algoritmo di acquisizione tramite il modulo National Instrument

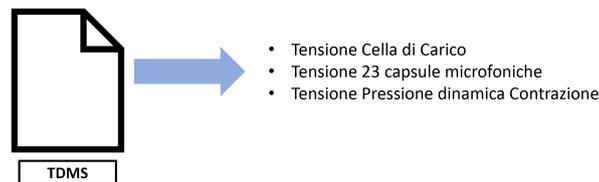


Figura 4.4: Output acquisizione modulo National Instrument

Comunicazione con Smart Zoc

Come riportato in 3.2 è presente una Scanivalve che permette di ottenere simultaneamente 64 segnali relativi alle pressioni statiche che sono distribuite su tutto il modello. La Scanivalve è connessa con una Smart-Zoc che permette una comunicazione con il Pc mediante cavo ethernet mediante protocollo TCP/IP.

Listing 4.2: DLL library per acquisizione da Smart-Zoc

```
// libreria dll per acquisizione da LabView
void mis_Pressure(char saveOnFile[], int32_t Samples, int32_t
    SamplingFrequency)
```

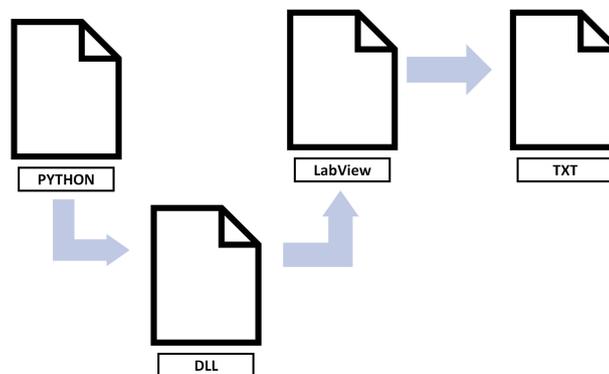


Figura 4.5: Schema algoritmo di acquisizione tramite la Smart-Zoc

Comunicazione per il controllo dei getti

Diversamente dai casi precedenti il controllo dei getti risulta essere più complicato nell'implementazione poiché per riprodurre una attuazione del getto oscillante è necessario utilizzare un segnale di riferimento che mediante il codice LabView viene creato, campionato e poi viene inviato il segnale campionato all'attuatore. Quindi questo significa che lo script di controllo del getto deve essere costantemente in esecuzione durante la prova. Per superare questo problema si è pensato di utilizzare uno script python che scrive su di un file csv le variabili relative all'attuazione, queste saranno lette ciclicamente da un ulteriore file python che sulla falsa riga di quanto visto in precedenza mediante una libreria DLL (4.3) attua i getti.

Listing 4.3: DLL library per attuazione getti

```
// libreria dll per acquisizione da LabView
```

```
void jet_control(char path_DLL_file,int32_t sampleMode,int32_t
    signalType,int32_t Stop,int32_t DC,double frequency,double
    VminV,double VmaxV)
```

Dove:

- *signalType*: definisce la tipologia di onda che verrà campionata e che quindi verrà riprodotta per l'attuazione del getto. Per il lavoro in esame si mantiene un segnale sinusoidale ma in un lavoro futuro si potrebbe pensare anche di introdurre questo come parametro di ottimizzazione.
- *DC*: definisce il duty cycle. In questo lavoro questo valore rimane fissato a 50% ma in un lavoro futuro si potrebbe pensare di estendere l'ottimizzazione anche a questo parametro.
- *VmaxV*: definisce la tensione massima di attuazione del getto.
- *frequency*: definisce la frequenza dell'onda campionata e quindi della frequenza del getto.

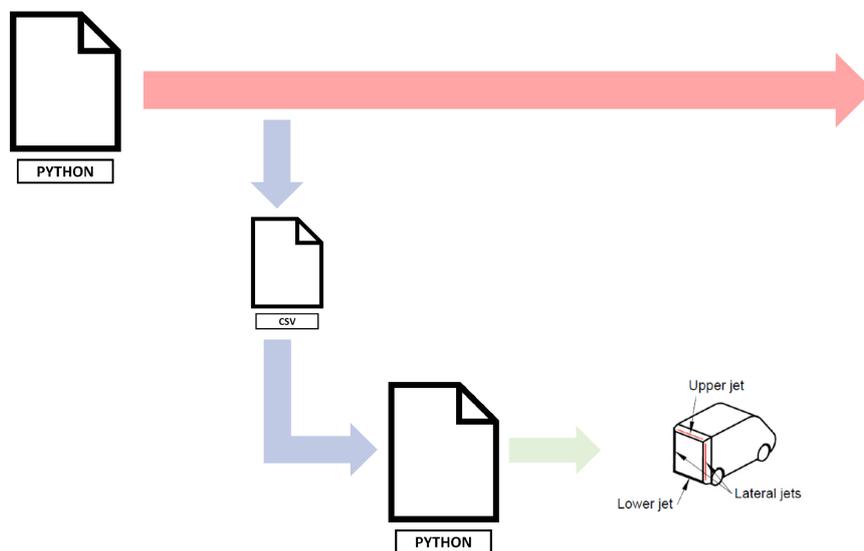


Figura 4.6: Schema algoritmo attuazione getti - In rosso viene rappresentato il processo principale ed in blu il processo secondario che ciclicamente legge il file csv e richiama la libreria DLL con i nuovi parametri di attuazione del getto

4.1.2 Implementazione funzione Execute

Di seguito si estende il funzionamento della funzione `Execute` precedentemente introdotta che si è detto essere fulcro fondamentale per l'ambiente.

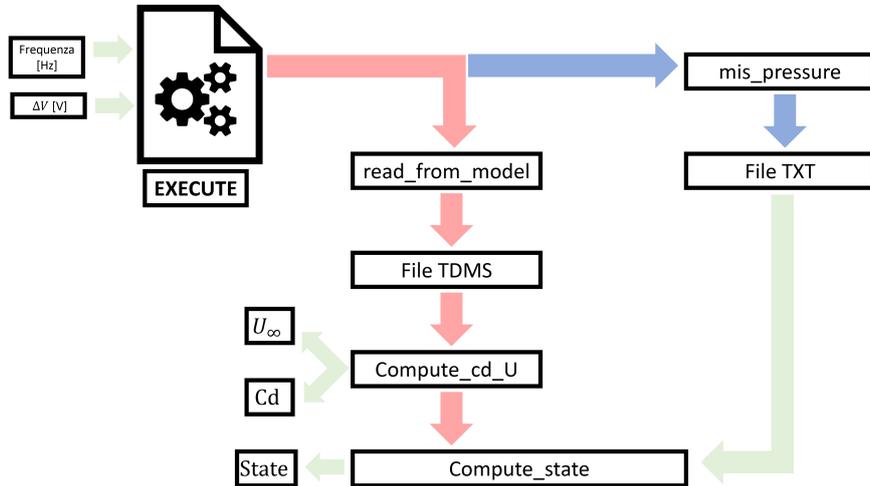


Figura 4.7: Schema algoritmo funzione `execute` - In rosso viene rappresentato il processo principale ed in blu il processo secondario che viene utilizzato per acquisire simultaneamente il segnale del modulo NI e della Smart-Zoc

Come rappresentato nello schema 4.7 è necessario considerare l'acquisizione simultanea del segnale derivante dal modulo di acquisizione National Instrument e dalla Smart-Zoc. Per questo motivo si è scelto di eseguire un calcolo multi-processo sfruttando la libreria **multiprocessing** disponibile in Python. In questo modo all'interno della funzione `execute`:

Nell'algoritmo 1 viene indicato un tempo di attesa: questo viene introdotto in modo tale da garantire che dopo l'attivazione del getto ci sia un tempo sufficiente affinché il flusso possa stabilizzarsi. Questo parametro viene fissato all'interno della classe ambiente e viene imposto pari a $t = 15s$.

4.1.3 Implementazione funzione Reset

Lo scopo della funzione `Reset`, come in parte si è detto in precedenza, ha il compito di ripristinare la condizione di partenza all'interno della galleria e quindi si occupa di interrompere l'attuazione del getto e di determinare lo stato iniziale.

Anche in questo caso nell'algoritmo 2 è presente un tempo di pausa necessario per far assestare il flusso, questo viene imposto per un tempo pari a $t = 60s$.

Algorithm 1 funzione *Execute*

Input:

$$\vec{f} = [f_1, f_2, f_3], \Delta\vec{V} = [\Delta V_1, \Delta V_2, \Delta V_3]$$

Output:

State, C_D , reward

- 1: **procedure** WRITE *.CSV FILE
 - 2: *create a structured python list*
 - 3: *write list in a csv file named as event's number*
 - 4: wait for a fixed time
 - 5: **procedure** START ACQUIRE FROM SMART-ZOC AND NI MODULE -
SIMULTANEOUSLY() \rightarrow file *.tdms and *.txt
 - 6: *call python function*
 - 7: compute flow speed, C_D and State from file created
 - 8: compute reward
-

Algorithm 2 funzione *Reset*

Input:

$$\vec{f} = [0,0,0], \Delta\vec{V} = [0,0,0]$$

Output:

State, C_{D_0}

- 1: **procedure** WRITE *.CSV FILE
 - 2: *create a structured python list*
 - 3: *write list in a csv file named as event's number*
 - 4: wait for a fixed time
 - 5: **procedure** START ACQUIRE FROM SMART-ZOC AND NI MODULE -
SIMULTANEOUSLY() \rightarrow file *.tdms and *.txt
 - 6: *call python function*
 - 7: compute flow speed, C_D and State from file created
 - 8: compute reward
-

4.2 Algoritmo di Reinforcement Learning

In questo paragrafo si vuole riportare l'algoritmo di RL utilizzato per il Training dell'agente. Questo segue quanto suggerito in [Tensorforce](#) e viene riportato in seguito per facilitare al lettore la comprensione dell'intero lavoro:

Listing 4.4: Algoritmo di training agente

```
// Algoritmo di training per N_episodes
# Train for N episodes
for _ in range(N):
    states = environment.reset()
    terminal = False
    while not terminal:
        actions = agent.act(states=states)
        states, terminal, reward =
            environment.execute(actions=actions)
        agent.observe(terminal=terminal, reward=reward)
```

Capitolo 5

Risultati

In questa sezione vengono analizzati i risultati dell'addestramento. Nel capitolo 3 si è introdotto il set-up sperimentale dove si è descritto il sistema di attuazione dei getti. In particolare si è visto che il sistema è in grado di riprodurre un getto costante o un getto che riproduce un segnale periodico (non solo sinusoidale ma anche un'onda quadra o triangolare) con una specifica frequenza, duty cycle e ampiezza. Inoltre il sistema è tale da rendere indipendente l'attuazione superiore da quella inferiore ed entrambe indipendenti dall'attuazione laterale.

Questo significa che il sistema è tale per cui si è in grado di attuare asimmetricamente sul piano verticale mentre sul piano orizzontale è possibile esclusivamente una attuazione simmetrica.

Nel seguente lavoro si è proceduto ad addestrare prima un agente in grado di generare una attuazione simmetrica, concetto che verrà ripreso in 5.1, con onda sinusoidale: questo significa che ognuno dei tre sistemi di attuazione segue un'unica onda sinusoidale con una fissata frequenza, ampiezza e duty cycle. Questo primo caso viene utilizzato per validare l'implementazione dell'algoritmo e il funzionamento della metodologia.

Questi due parametri (si esclude il duty cycle) sono esattamente quelli che la rete dovrà andare a definire al fine di ridurre la resistenza aerodinamica del modello.

In una seconda fase si è addestrato un agente in grado di attuare indipendentemente i tre sistemi di attuazione. Quindi in quest'ultimo caso la rete dovrà andare ad individuare sei parametri (una frequenza e un'ampiezza per ognuno dei tre sistemi di attuazione).

Dai risultati di Cerutti et al. si evidenzia come le alte velocità di attuazione risultano essere le più efficaci al fine di ridurre la resistenza. Questo risultato è emerso anche per altre geometrie come mostrato in Barros et al.. Inoltre Cerutti et al. hanno dimostrato che l'attuazione combinata dei getti laterali ed inferiore introduce una riduzione della resistenza aerodinamica due volte maggiori rispetto alle configurazioni indipendenti più favorevoli. Questo è stato giustificato facendo riferimento all'elevata tridimensionalità della scia che suggerisce una maggiore efficacia

agendo simultaneamente sui tre bordi rispetto alla sola parte inferiore. Per verificare il corretto funzionamento della metodologia RL si può considerare di addestrare un agente in grado di attuare indipendentemente i tre sistemi di attuazione con un'onda quadra. La rete in questo caso andrà a trovare tre parametri (frequenza, duty cycle ed ampiezza) per ognuno dei tre sistemi. Infatti se il sistema si porta in condizioni simili a quelle trovate in letteratura, si può affermare che la tecnica metodologicamente funziona in modo corretto. Tuttavia questa analisi non viene eseguita in questo lavoro, ma rappresenta un suo possibile sviluppo futuro. Per di più vengono eseguiti addestramenti a diversi valori di condizioni della corrente a monte: questo viene fatto ricominciando l'addestramento ma, in futuro, si potrebbe pensare anche di continuare un addestramento eseguito per una data condizione di monte in una diversa configurazione. Questa analisi permetterebbe di capire se l'agente è in grado di raggiungere una condizione di riduzione della resistenza aerodinamica anche all'interno intervallo compreso tra le due velocità di addestramento.

5.1 Caso 1: attuazione simmetrica - $Re_\infty \sim 0.9 \cdot 10^5$

Il primo caso che si considera riguarda una attuazione con tensioni simmetriche. Questa condizione di simmetria però non si traduce in una simmetria delle velocità del getto.

In figura 5.1a e 5.1b vengono riportati gli andamenti del reward, definito come $Cd_0 - Cd$, e della riduzione percentuale della resistenza aerodinamica al variare del numero di episodi.

Come descritto in 4.1 ogni episodio è costituito da 15 eventi, le curve rappresentate in figura riportano i valori di reward e della riduzione di resistenza aerodinamica media, massima e minima all'interno dell'episodio. Dalle curve si evidenzia come l'agente con il passare degli episodi impari, infatti il valore medio (ma analogo discorso potrebbe essere fatto per quanto riguarda il valore massimo e minimo) del reward aumenta.

Mediamente l'agente arriva ad una riduzione percentuale della drag di $\sim 8\%$ con valori massimi che arrivano a $\sim 10\%$, quindi questi valori sono quelli che ci si aspetterebbe di trovare durante la fase di training. Questi andamenti sembrano suggerire che realmente l'agente apprende il modo in cui sfruttare il sistema di attuazione, ricordando che si sta analizzando il caso dell'attuazione simmetrica, al fine di raggiungere una riduzione della resistenza aerodinamica.

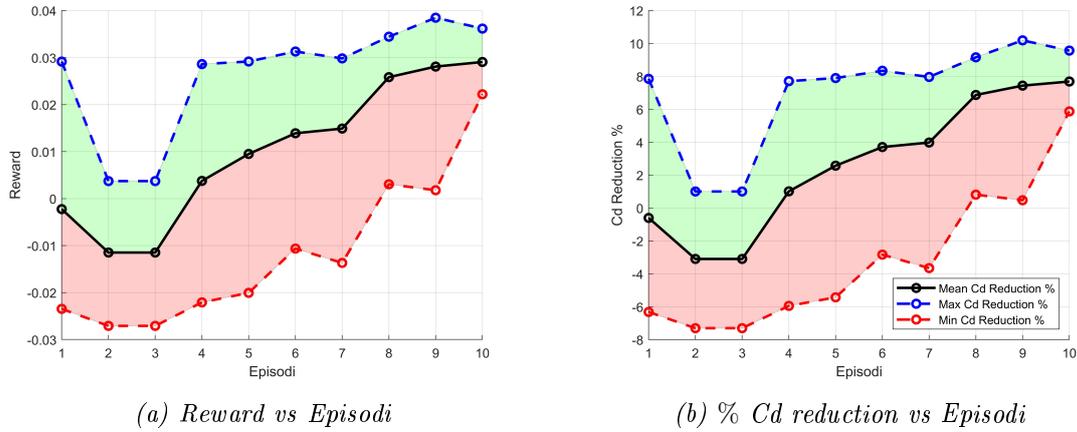


Figura 5.1: Training caso 1

5.2 Caso 2: attuazione asimmetrica - $Re_\infty \sim 0.9 \cdot 10^5$

In figura 5.2a e 5.2b vengono riportati gli andamenti del reward e della riduzione percentuale del coefficiente di resistenza al variare del numero di episodi. Si evidenzia, come già esposto in 5.1, come l'agente nella fase di training si pone nella condizione di imparare a ridurre la resistenza aerodinamica.

E' bene sempre ricordare che ciò che l'algoritmo realmente realizza è la massimizzazione del reward, ma questa condizione è stata formulata in modo tale che all'aumento del reward corrisponda una riduzione del Cd, come indicato in sezione 4.1. Da figura 5.2b emerge come mediamente si ottiene $\sim 8\%$ di riduzione percentuale di drag, con valori massimi che sfiorano $\sim 10\%$.

L'agente in questione è stato addestrato per 26 episodi, corrispondenti ad un tempo fisico di circa 3h e 25 minuti, e si osserva che il sistema si avvicina ad una condizione di convergenza.

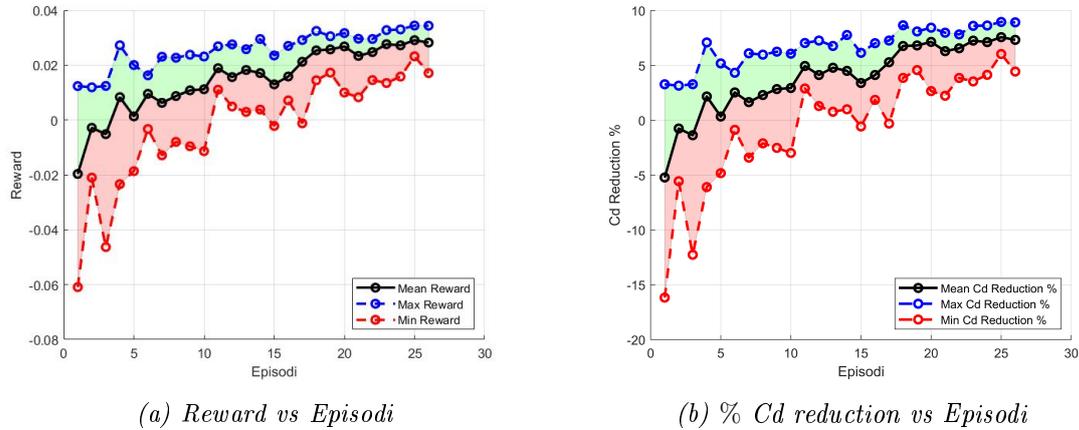
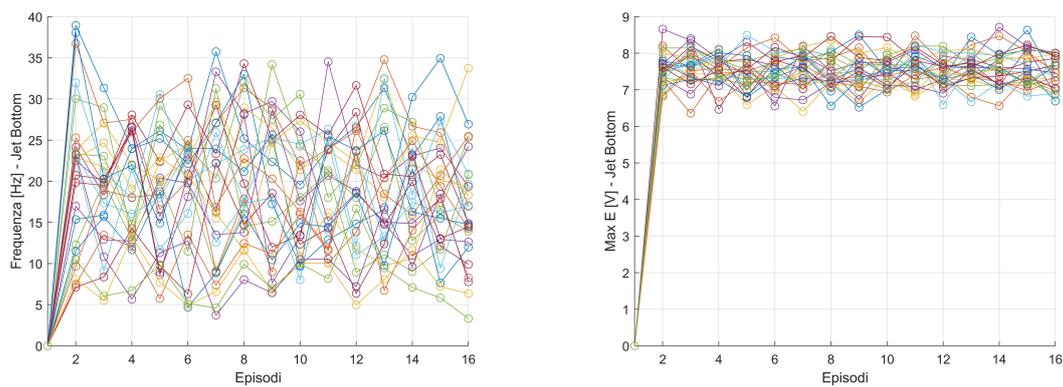


Figura 5.2: Training caso 2

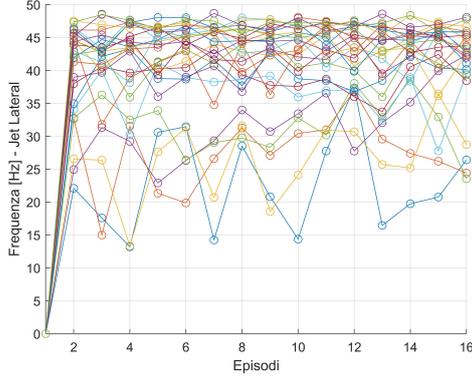
Nelle figure 5.3a - 5.5b vengono riportati gli andamenti della variazione dei parametri di addestramento, quindi frequenza e tensione massima di attuazione, della rete neurale. Dalle figure si mette in evidenza l'esplorazione dell'agente, infatti questo continua a variare i parametri all'interno del range ammissibile fissato dall'utente, tenendo in conto delle limitazioni strutturali del sistema: $f_{max} = 50 \text{ Hz}$, $tensione\ max = 9.5 \text{ V}$.



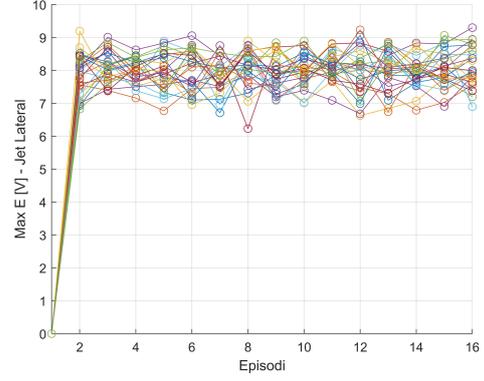
(a) Variazione della frequenza di attuazione del bottom jet nei diversi episodi

(b) Variazione della tensione di attuazione del bottom jet nei diversi episodi

Figura 5.3: Training - Actions bottom jet

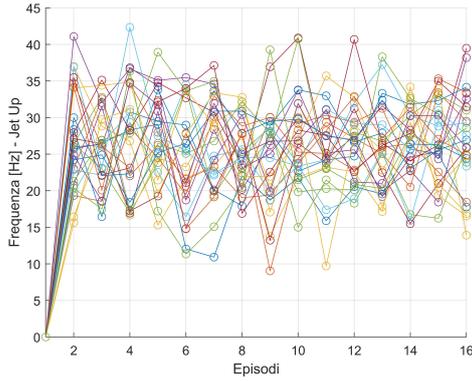


(a) Variazione della frequenza di attuazione del lateral jet nei diversi episodi

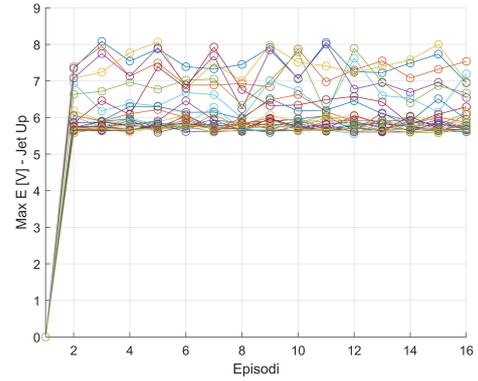


(b) Variazione della tensione di attuazione del lateral jet nei diversi episodi

Figura 5.4: Training - Actions lateral jet



(a) Variazione della frequenza di attuazione del up jet nei diversi episodi



(b) Variazione della tensione di attuazione del up jet nei diversi episodi

Figura 5.5: Training - Actions up jet

5.2.1 Bilancio energetico

Nel paragrafo 1 si è introdotto il concetto di bilancio energetico, infatti un controllo in problemi reali si pone come obiettivo non solo quello di ridurre la resistenza aerodinamica ma anche di garantire che questo avvenga in modo efficiente: il vantaggio deve essere maggiore della spesa. È importante evidenziare che la definizione di efficienza non può essere ottenuta banalmente, come suggerito da Choi et al.. Ad esempio gli autori di Cerutti et al. definiscono l'efficienza come:

$$\zeta = \frac{P_s}{P_c} \quad (5.2.1)$$

basandosi sull'idea di tenere in conto tutte le possibili perdite che si riscontrano realmente nell'implementazione (dove P_s rappresenta l'energia salvata, mentre P_c quella spesa). Seguendo il ragionamento di [Choi et al.](#) la valutazione dell'energia spesa richiederebbe la conoscenza della perdita di carico all'interno del sistema pneumatico e l'energia associata all'attuazione:

$$P_c = \sum p_{losses} \cdot Q + \sum_{i=1}^{N_{jets}} 0.5\rho V_{i,j}^3 \cdot A_i \quad (5.2.2)$$

dove Q rappresenta la portata volumetrica totale, A_i indica la cross-section del i -esimo getto e $V_{i,j}$ rappresenta la velocità di immissione dell' i -esimo getto. Analogamente è necessario definire la potenza salvata P_s :

$$P_s = (D - D^c) \cdot V_\infty \quad (5.2.3)$$

In questo modo si può concludere che $\zeta > 1$ definisce una condizione efficiente dal punto di vista energetico.

Un altro possibile approccio è quello suggerito da [Englar](#) che ha analizzato la sensibilità del controllo sulla scia attraverso il rapporto tra la variazione di resistenza e il coefficiente di quantità di moto del getto C_μ :

$$\frac{\Delta C_D}{C_\mu} = \frac{D - D^c}{0.5 \sum_{i=1}^{N_{jets}} \rho A_i V_{i,j}^2} \quad (5.2.4)$$

L'equazione [5.2.4](#) può assumere sia valori positivi che valori negativi: nel primo caso significa che c'è una riduzione della resistenza. Valori di $\frac{\Delta C_D}{C_\mu} > 1$ corrispondono a configurazioni efficienti dal punto di vista energetico. Dalla [figura 5.6](#) mostra un valore di $\frac{\Delta C_D}{C_\mu}$ particolarmente basso e durante l'addestramento non si ha un andamento crescente. Questo si giustifica ricordando che il reward è stato fissato senza considerare il lato energetico, quindi senza perseguire una condizione di efficienza. Un possibile sviluppo di questo lavoro di tesi potrebbe essere proprio quello di andare ad addestrare nuovamente la rete imponendo come reward la condizione di efficienza energetica.

5.2.2 Performance dell'agente

Conclusa la fase di training è possibile eseguire la fase di valutazione, cioè la fase in cui si va a sfruttare la rete senza esplorazione ma con un approccio puramente deterministico sfruttando i risultati raccolti durante la fase di training. Come mostrato in [figura 5.7a](#) e [5.7b](#) l'agente riesce ad imparare in un numero limitato di episodi come ridurre la resistenza aerodinamica.

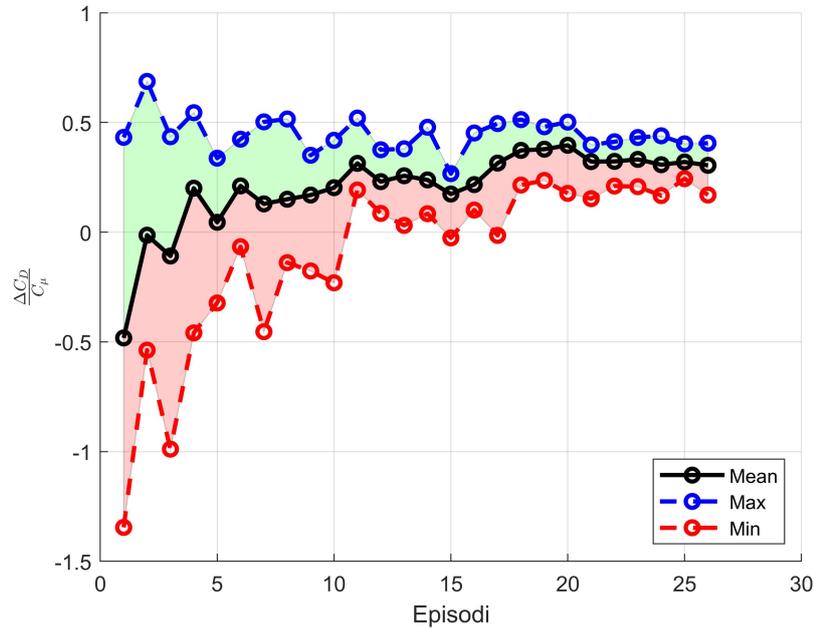
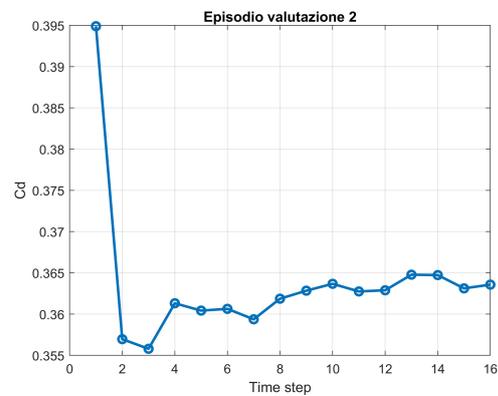
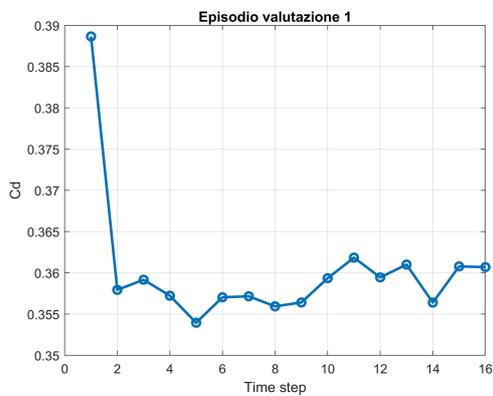


Figura 5.6: $\frac{\Delta C_D}{C_\mu}$ vs Episode



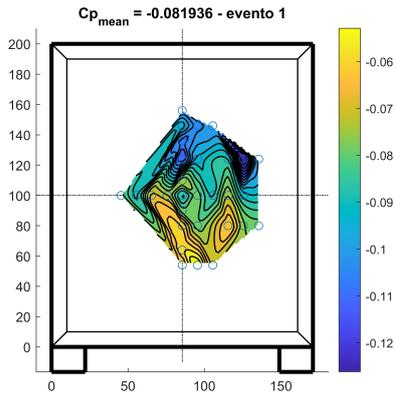
(a) Cd time history in evaluation - episode 1

(b) Cd time history in evaluation - episode 2

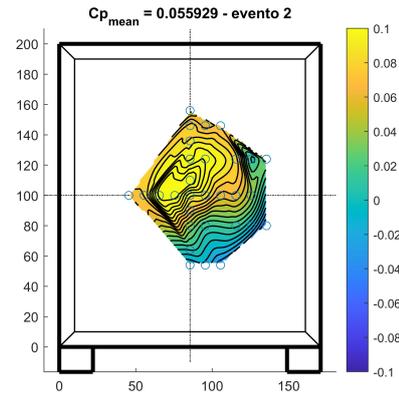
Figura 5.7: Valutazione episodi - Caso 2

Di seguito si riportano le distribuzioni di C_p base del modello in modo tale da poter osservare cosa succede dal punto di vista fluidodinamico nel momento in cui l'agente aziona i getti, ma anche per cercare di analizzare sotto quale idea evolve la politica dell'agente.

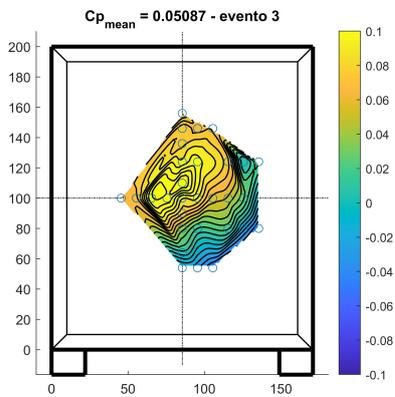
Nelle figure 5.8 e 5.9 si osserva che nella condizione unforced (figura 5.8a) il C_p medio risulta essere negativo mentre quando l'agente comincia a forzare tramite



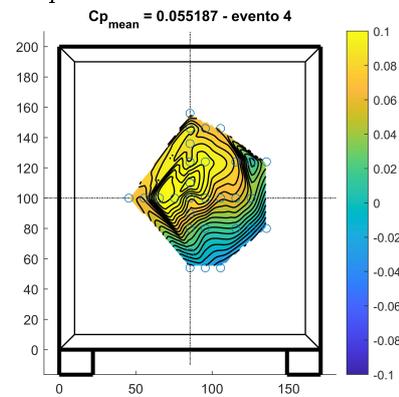
(a) Cp base distribution - unforced



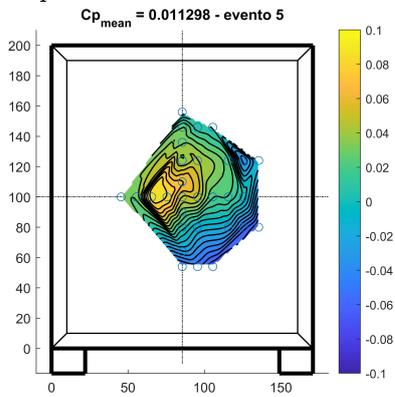
(b) Cp base distribution - time-step 1



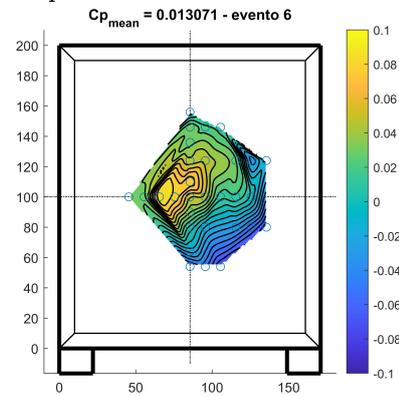
(c) Cp base distribution - time-step 2



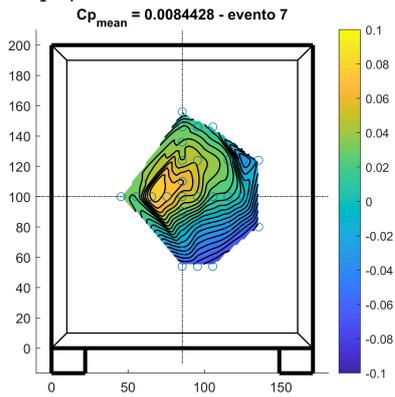
(d) Cp base distribution - time-step 3



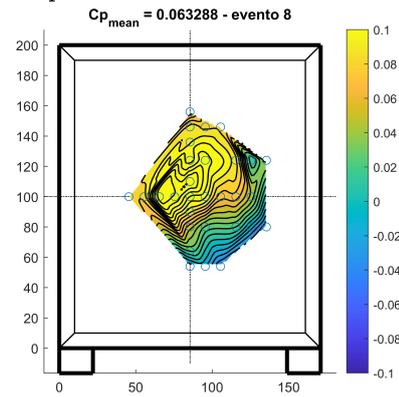
(e) Cp base distribution - time-step 4



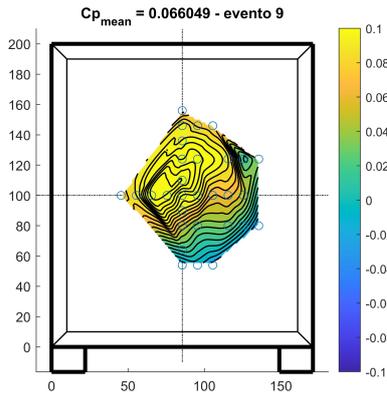
(f) Cp base distribution - time-step 5



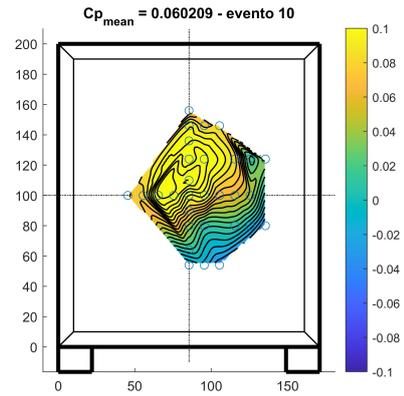
(g) Cp base distribution - time-step 6



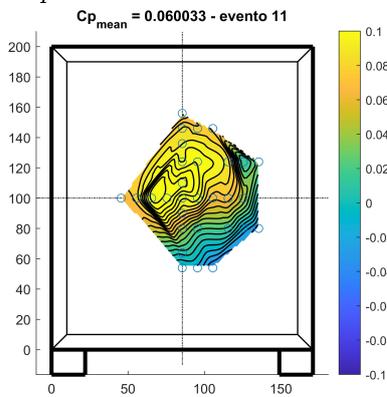
(h) Cp base distribution - time-step 7



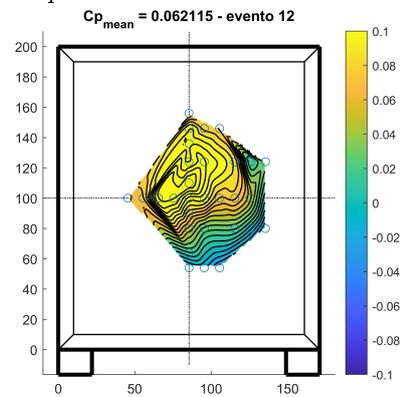
(a) Cp base distribution - time-step 8



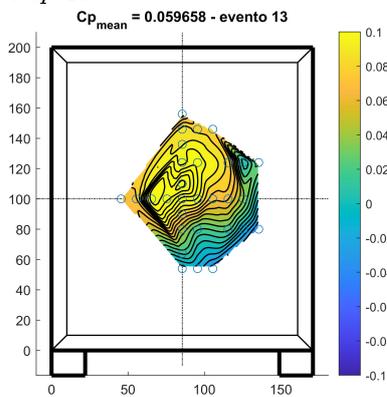
(b) Cp base distribution - time-step 9



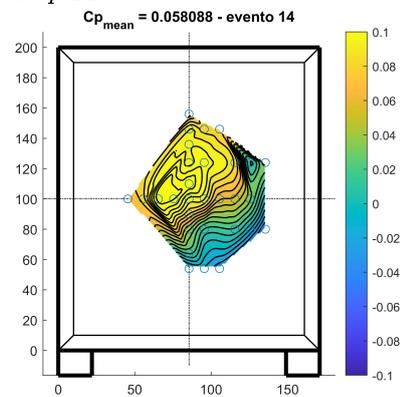
(c) Cp base distribution - time-step 10



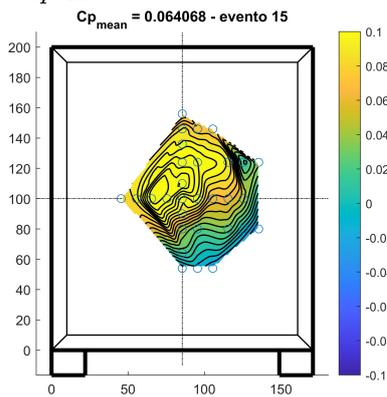
(d) Cp base distribution - time-step 11



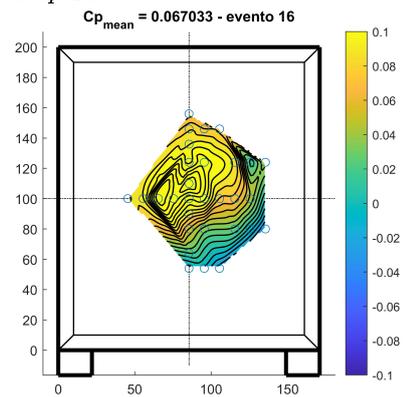
(e) Cp base distribution - time-step 12



(f) Cp base distribution - time-step 13



(g) Cp base distribution - time-step 14



(h) Cp base distribution - time-step 15

l'attuazione dei getti il C_p medio risulta positivo. Quindi nella condizione unforced si ha una azione di suction che genera una forte resistenza aerodinamica. Il forcing permette di ottenere una riduzione della resistenza aerodinamica (come mostrato in figura 5.7b) agendo su questa riduzione di pressione.

Dal punto di vista della distribuzione di pressione è possibile distinguere chiaramente il caso forzato dal caso unforced, infatti nel caso unforced (figura 5.8a) si distingue una regione di sovrappressione (C_p positivo) nella parte inferiore mentre una zona di depressione (C_p negativo) nella parte superiore.

Quindi già dalla distribuzione è possibile intuire una struttura asimmetrica della scia nella condizione naturale, risultato atteso data la condizione al contorno non simmetrica per la presenza del pavimento. Nelle altre condizioni, che sono delle condizioni di forcing, la distribuzione appare molto simile in tutti i casi con la presenza di una forte regione di sovrappressione nella zona centrale della base. In realtà si evidenzia uno spostamento della zona di sovrappressione verso il lato destro della base, questo si potrebbe giustificare con la presenza di un angolo di imbardata al quale la scia risulta essere molto sensibile [32].

Nelle tabelle 5.1, 5.2 e 5.3 vengono riportate le condizioni di forcing ottenute dall'agente. E' importante evidenziare come l'agente si porti in una condizione di forcing diversa per ogni time-step, ma i valori ottenuti sono molto simili tra loro: le differenze sono apprezzabili a partire dalla terza cifra decimale.

E_{maxUp} [V]	$E_{maxBottom}$ [V]	$E_{maxLateral}$ [V]
8.50	8.74	9.83

Tabella 5.1: Tensioni di forcing - caso 2

U_{Up} [m/s]	U_{Bottom} [m/s]	$U_{Lateral}$ [m/s]
26.45	26.46	15.78

Tabella 5.2: Velocità di forcing - caso 2

$FreqUp$ [Hz]	$FreqBottom$ [Hz]	$FreqLateral$ [Hz]
14.80	42.79	48.81

Tabella 5.3: Frequenze di forcing - caso 2

5.2.3 Analisi spettrale del segnale di pressione

Le capsule microfoniche poste nella parte posteriore del modello registrano le fluttuazioni di pressione. È possibile utilizzare questo segnale per effettuare un'analisi

nel dominio delle frequenze per ottenere informazioni sulle caratteristiche di instabilità della scia. Delle capsule presenti se ne prendono in esame 3: una in posizione top, l'altra in posizione bottom e un'ultima in posizione laterale come riportato nello schema in figura 5.10 e in tabella 5.4.

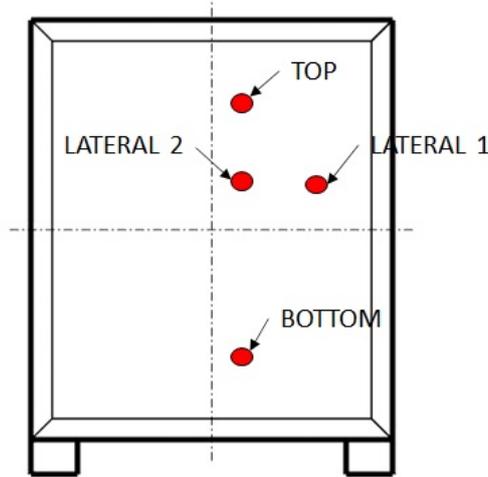


Figura 5.10: Schema posizione microfoni analisi spettrale - Posizione dei microfoni: tabella 5.4

Microfono	y [mm]	z [mm]
Top	106.16	135.19
Lateral 1	135.19	110.5
Lateral 2	95.19	110.5
Bottom	106.16	35.7

Tabella 5.4: Posizione microfoni analisi spettrale

In figura 5.10 viene riportato la PSD (*Power spectral density*) con in ascisse la frequenza analizzata $St = \frac{f \cdot W}{U_\infty}$ per la condizione forzata e per la condizione naturale.

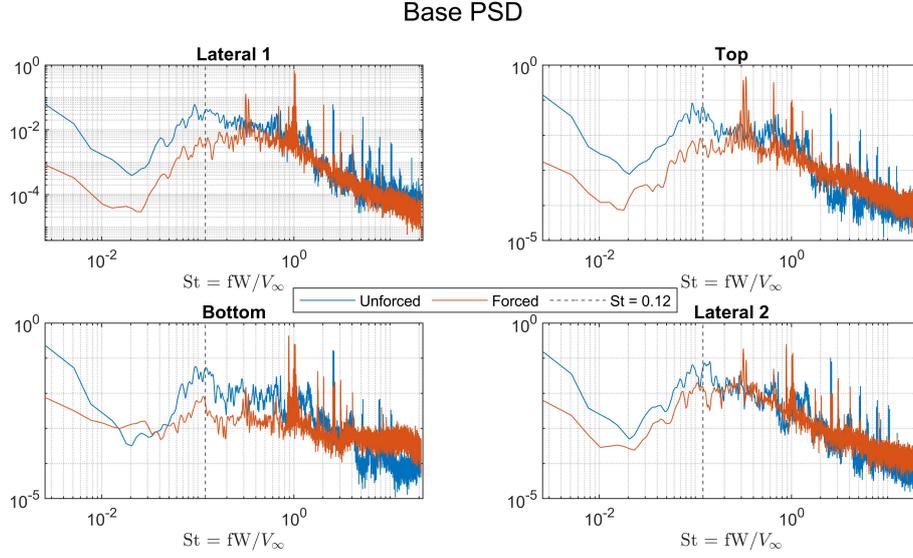


Figura 5.11: PSD - Power spectral density - caso 2

E' importante considerare che la geometria del modello è tale per cui si ha uno shear layer tridimensionale. Infatti da ogni spigolo della base nasce uno shear layer e quindi questi interagiscono tra di loro in modo da formare una struttura tridimensionale e complessa. Questo rende particolarmente ardua l'interpretazione dei risultati che derivano dall'analisi spettrale.

Da figura 5.11 si osserva che nella condizione naturale si ha un picco in corrispondenza di $St \approx 0.12$ che coincide con il valore che si trova in letteratura (Cerutti et al., Grandemange et al.). Sempre nel caso naturale si osserva la presenza di un ulteriori picchi per St maggiori che emergono chiaramente nella PSD del microfono in posizione laterale (posizione *lateral 1* tabella 5.4).

Facendo la stessa analisi per un microfono in posizione più vicina al piano di simmetria trasversale del modello (posizione *lateral 2* tabella 5.4) si evidenzia la riduzione di questo picco. Questa osservazione suggerisce che la presenza di tale picco si potrebbe spiegare come conseguenza della traccia shear layer: infatti il microfono in più vicina allo spigolo del modello risente maggiormente della traccia dello shear layer.

Questa osservazione è avvalorata dal fatto che i picchi presenti nello spettro del microfono *lateral 1* tendono ad allargarsi in termini di frequenza nel caso di *lateral 2*. La PSD in figura 5.11 è stata eseguita sul segnale di fluttuazione di pressione normalizzato con la p'_{rms} .

Confrontando gli spettri nel caso naturale e nel caso forzato si osserva che nella condizione di forcing emergono valori di energia spettrale minori per St bassi rispetto al caso forzato. Per il microfono in posizione *bottom* si evidenzia una grande

quantità di energia associata ad alti St nel caso forzato.

Su questo risultato si possono fare diverse considerazioni: sicuramente la presenza di un sistema di attuazione introduce all'interno del sistema energia e quindi è lecito aspettarsi una maggiore energia associata alla condizione di forcing, risultato che emergerebbe se non si facesse l'adimensionalizzazione. Inoltre la presenza del forcing introduce una rottura della strutture presenti nella condizione naturale, di conseguenza si ha la formazione di strutture di più piccola scala e quindi a frequenza più elevata. Questa condizione è quella auspicabile dal punto di vista della riduzione della resistenza aerodinamica, infatti implica una più veloce dissipazione della turbolenza.

Naturalmente nella condizione forzata si evidenzia la presenza di picchi attribuibili al forcing: infatti nelle figure 5.11 si osservano dei picchi per valori di St equivalenti alle frequenze di forcing e multipli di queste (tabella 5.1 e 5.3).

Analisi POD

Per meglio comprendere il ruolo dell'agente si effettua una analisi *POD* sfruttando i dati raccolti dalle capsule microfoniche posizionate nella base del modello (come mostrato nel paragrafo 3) e facendo un confronto tra la condizione forced e quella unforced. L'analisi POD (Proper Orthogonal Decomposition) si pone come obiettivo quello di determinare i modi con maggior energia, così da confrontarle con la letteratura, e di valutare l'effetto del forcing sulla struttura della scia. La tecnica POD, come spiegato in Berkooz et al., decompone il segnale temporale in modi e coefficienti modali. Dato un generico segnale temporale $p'(Y, Z, t)$ questo viene decomposto in una somma di k modi ognuno dei quali è costituito da un coefficiente temporale $a_k(t)$ ed una autofunzione $\varphi_k(Y, Z)$ nel seguente modo:

$$p'(y, z, t) = \sum_k a_k(t) \varphi_k(Y, Z) \quad (5.2.5)$$

Ogni autofunzione $\varphi_k(Y, Z)$ deve soddisfare la condizione:

$$\int \langle p'(Y, Z, t) \cdot p'(Y, Z, t)^T \rangle_t \cdot \varphi_j(Y, Z) dydz = \lambda_j \cdot \varphi_j(Y, Z) \quad (5.2.6)$$

dove j varia da 1 al numero totale delle capsule microfoniche: 12.

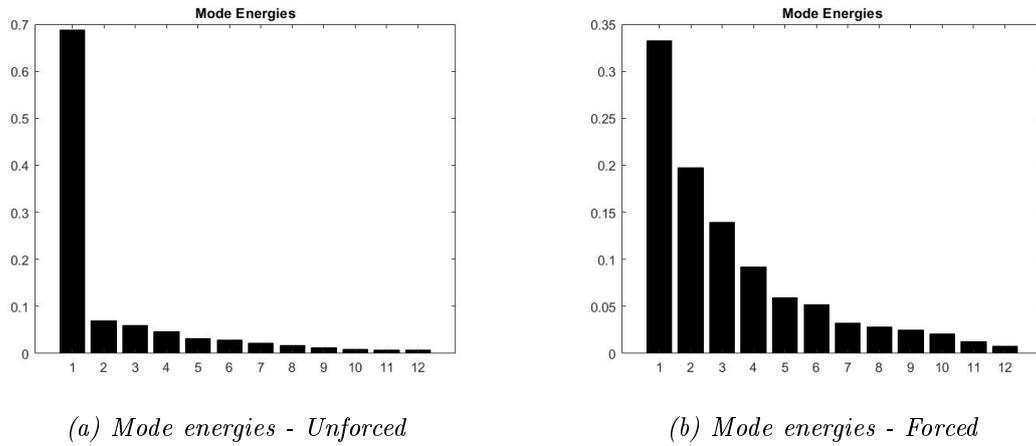


Figura 5.12: Mode energies - Caso 2

Nelle figure in 5.27 appare evidente la differenza tra i due casi. Nel caso unforced (5.12a) il primo modo è sede di circa il 70% dell'energia complessiva, mentre nel caso forced (5.12b) l'energia ha una più equa suddivisione tra i modi.

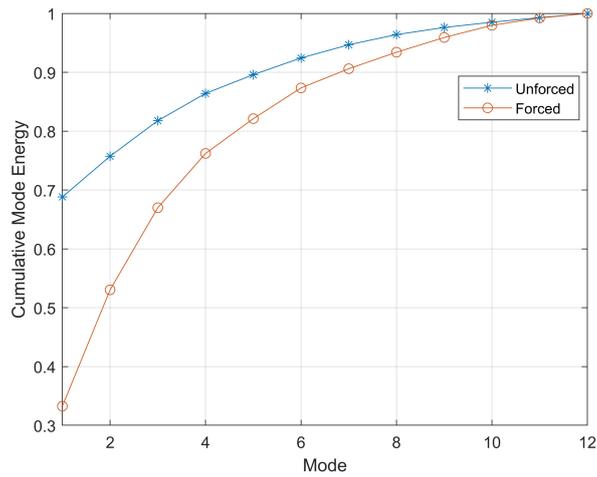
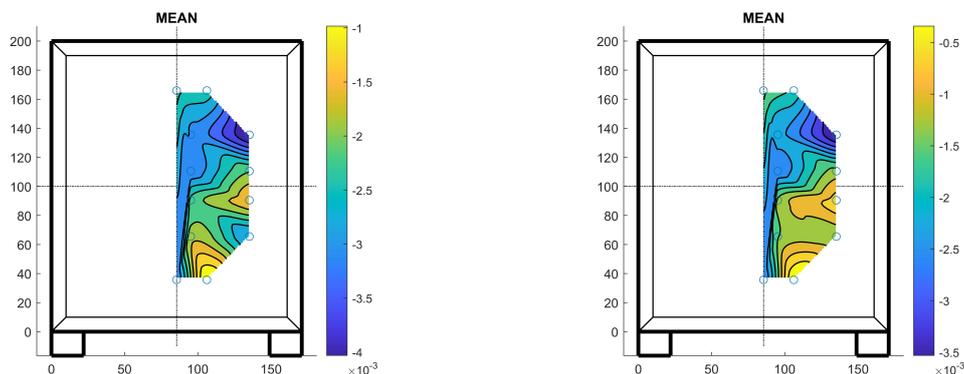


Figura 5.13: Energia cumulativa Pod modes



(a) Campo medio fluttuazioni di pressioni - Unforced (b) Campo medio fluttuazioni di pressioni - Forced

Figura 5.14: Campo medio fluttuazioni di pressioni - Caso 2

Analizzando i primi quattro modi Pod (figura 5.15 e 5.16) è possibile evidenziare differenze sostanziali tra le due condizioni. Per il modo 1 in condizione naturale (figura 5.15a) si nota una condizione asimmetrica che ben si concilia con quanto visto nella distribuzione del C_p . Come mostrato in precedenza questo è anche il modo a cui è associata la maggior quantità di energia, infatti gli altri modi possono essere considerati trascurabili. Il modo 1 sembra suggerire una condizione di shedding sopra-sotto. Diversamente nel caso forzato, qui per il modo 1 (figura 5.15b) appare evidente come il forcing vada a generare una simmetrizzazione della struttura fluidodinamica. Analizzando i risultati dell'analisi spettrale in figura 5.15c, caso non forzato, si osserva una serie di picchi nell'intorno di $f \sim 5.6Hz$ che corrisponde a $St \sim 0.12$, risultato concorde con quanto riportato nell'analisi in sezione 5.2.3. Diversamente nel caso forzato (figura 5.15d): qui si evidenziano dei picchi, con un contributo energetico circa uguale tra loro, in corrispondenza delle frequenze di forcing (tabella 5.3).

Il modo 2 sembra essere il duale del modo 1 in entrambe le condizioni. La condizione di forcing asimmetrico e la prevalenza di una direzione sull'altra permettono di giustificare l'assenza di un secondo modo con energia uguale al primo.

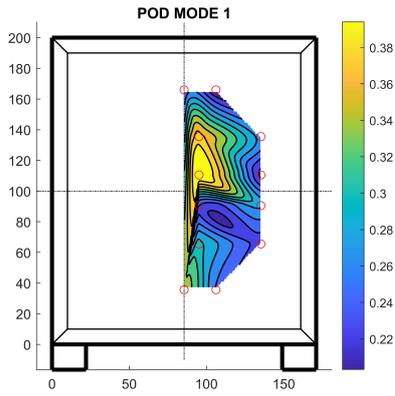
Analizzando i risultati dell'analisi spettrale dei coefficienti temporali per il modo 2 si osserva che per la condizione naturale (figura 5.15g) non si riesce ad identificare un vero e proprio picco: per giustificare questo comportamento, che come si vedrà è più accentuato nel modo 3, servirebbero della analisi più approfondite ma una possibile spiegazione potrebbe trovarsi andando a considerare che al modo 2 in condizione naturale è associato un basso livello energetico e quindi è equiparabile al rumore e lo spettro del rumore è piatto.

Mentre per la condizione forzata, rappresentata in figura 5.15h, si evidenzia una riduzione dell'energia associata alla frequenza del forcing superiore ($\sim 14.8Hz$), questa si porta circa allo stesso livello energetico di un suo multiplo: $\sim 30.8Hz$.

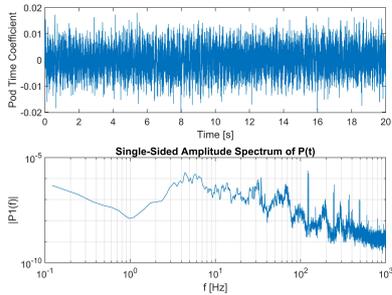
Diversamente si ha un aumento dell'energia associata ad una frequenza multipla della frequenza del forcing inferiore ($\sim 83Hz$). Secondo l'autore questo è dovuto alla velocità con cui avviene il forcing: infatti nella il forcing superiore avviene a velocità inferiore rispetto a quello inferiore.

Dal modo 2 in poi ha senso discuterne solo per il caso forzato, infatti da figura 5.12a si evidenzia chiaramente come già il modo 2 abbia una energia associata $< 10\%$, mentre nel caso forzato solo dal quinto modo in poi si ha una tale energia associata. Il modo 3 risulta, a parti invertite, analogo al modo 1. Infatti quanto rappresentato in 5.16a si ritrova in 5.15b ed analogamente tra 5.15a e 5.16b.

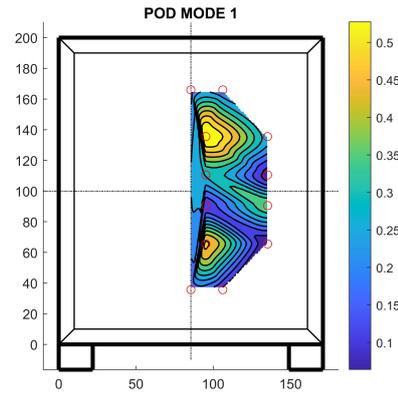
Analoghe osservazioni per l'andamento dei coefficienti temporali si possono fare per il modo 3 (figura 5.16c e 5.16d), dove però le frequenze associate al forcing sono associate circa alla stessa energia. Nel modo 4 invece, come si evidenzia da figura 5.16g, si evidenzia nuovamente il picco in prossimità della frequenza per cui $St \sim 0.12$ mentre nel caso forzato (figura 5.16g) la maggior energia è associata alla frequenza $\sim 30.5Hz$ che si può identificare come multiplo del forcing superiore ($f = 14.8Hz$).



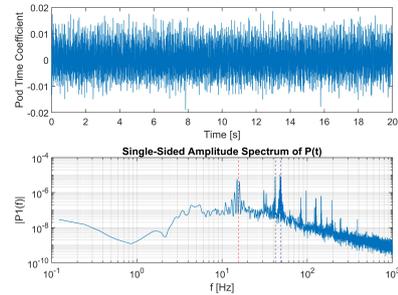
(a) Pod Mode 1 - Unforced
POD MODE 1



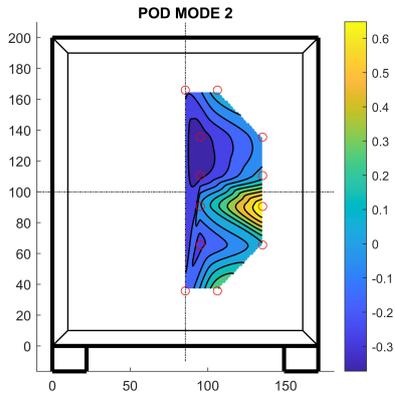
(c) Pod Mode 1 - Time Coefficient - Unforced



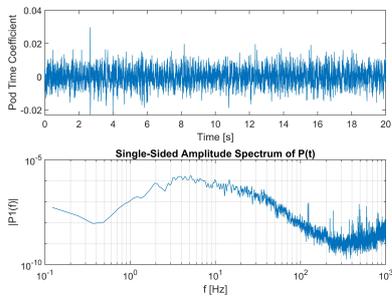
(b) Pod Mode 1 - Forced
POD MODE 1



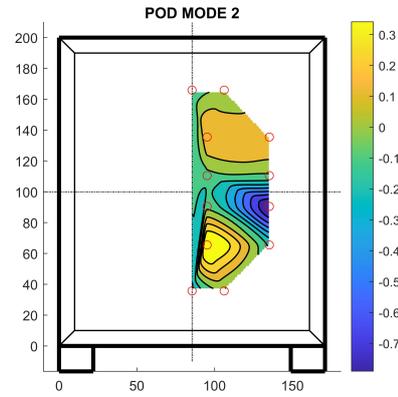
(d) Pod Mode 1 - Time Coefficient - Forced



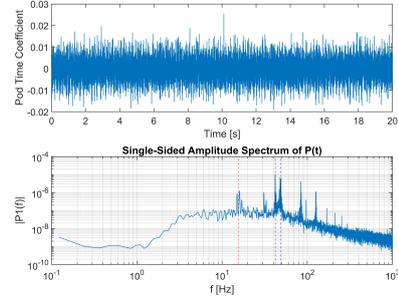
(e) Pod Mode 2 - Unforced
POD MODE 2



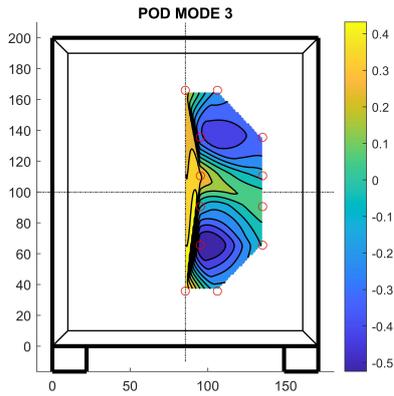
(g) Pod Mode 2 - Time Coefficient - Unforced



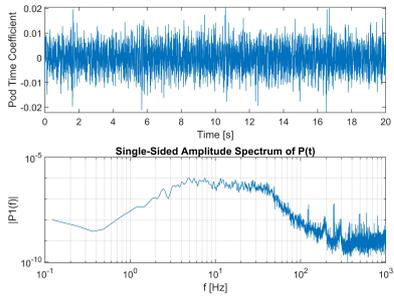
(f) Pod Mode 2 - Forced
POD MODE 2



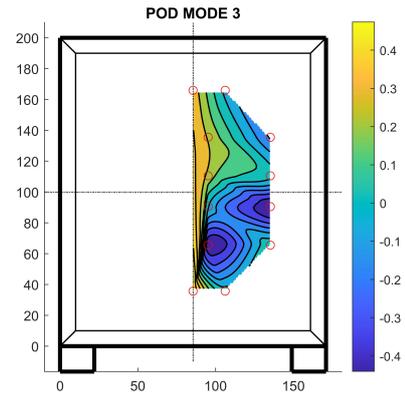
(h) Pod Mode 2 - Time Coefficient - Forced



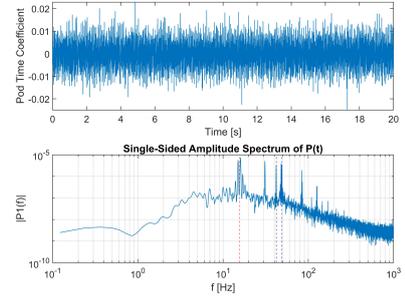
(a) Pod Mode 3 - Unforced
POD MODE 3



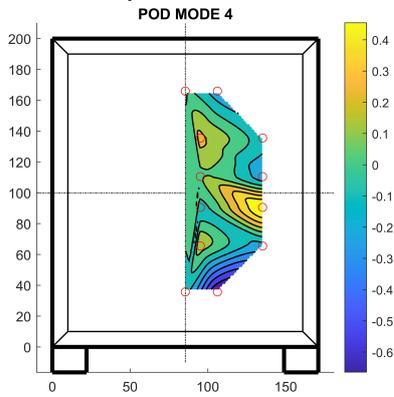
(c) Pod Mode 3 - Time Coefficient - Unforced



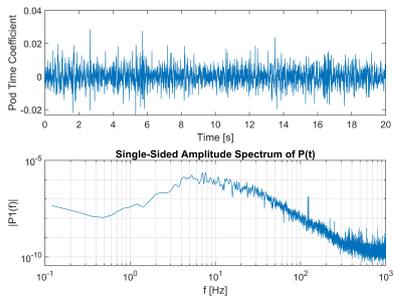
(b) Pod Mode 3 - Forced
POD MODE 3



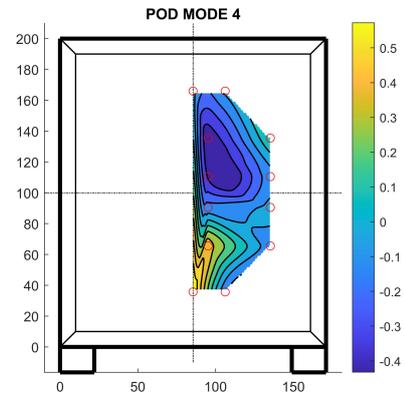
(d) Pod Mode 3 - Time Coefficient - Forced



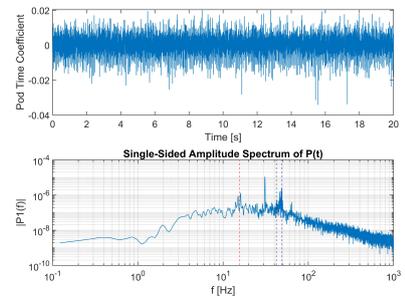
(e) Pod Mode 4 - Unforced
POD MODE 4



(g) Pod Mode 4 - Time Coefficient - Unforced



(f) Pod Mode 4 - Forced
POD MODE 4



(h) Pod Mode 4 - Time Coefficient - Forced

5.2.4 Ricerca moti organizzati

Un'ulteriore analisi che permette di comprendere meglio cosa implica la condizione di forcing è quella che sfrutta la tecnica VITA (*Variable-Interval Time-Averaging*) mediante la definizione della varianza locale (equazione 5.2.7), seguendo il lavoro di [Blackwelder and Kaplan](#).

Questa analisi è stata introdotta per studiare un flusso di parete e prendendo come grandezza di riferimento la velocità. Invece in questo lavoro si sfrutta tale analisi per un segnale di pressione fluttuante e viene utilizzato per studiare la scia. Queste differenze fanno sì che sebbene la metodologia utilizzata segue esattamente il lavoro di [Blackwelder and Kaplan](#), l'interpretazione dei risultati è diversa e quindi risulta di maggiore complessità.

Definito un intervallo temporale T (non necessariamente l'intervallo deve essere temporale, infatti in questo studio sarà un intervallo di campioni) si definisce la varianza locale come:

$$var_{local}(x_i, t, T) = \widehat{u}^2(x_i, t, T) - [\widehat{u}'(x_i, t, T)]^2 \quad (5.2.7)$$

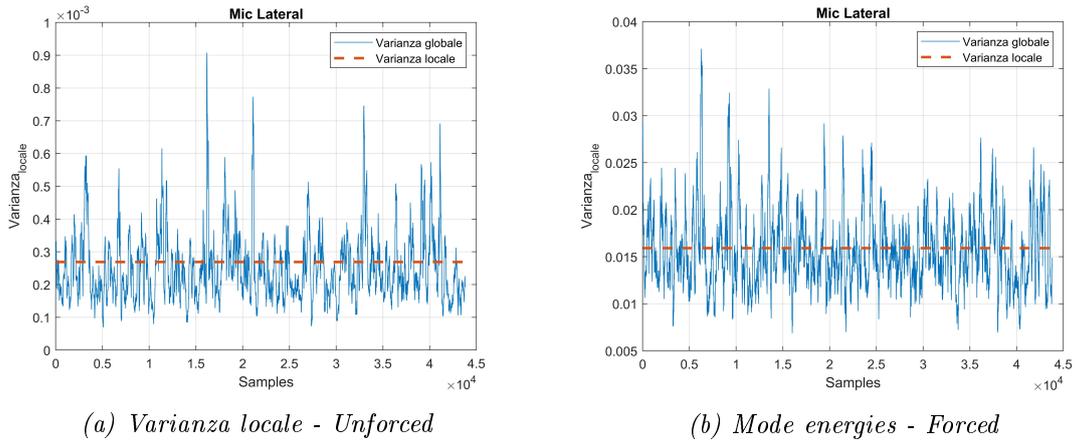


Figura 5.17: Varianza locale - Caso 2

Nota la varianza globale:

$$var_{glob} = \bar{u}^2 = \sum_{i=0}^T \frac{[x(i) - \bar{x}]^2}{T} \quad (5.2.8)$$

è possibile definire il numero degli eventi, dove per evento si definisce il numero di occorrenze per cui il valore della varianza locale supera di k volte, con k fissato pari a 1, il valore della varianza globale (figura 5.17a e 5.17b). Individuati gli eventi si può costruire la matrice degli eventi: matrice dove ogni riga rappresenta un evento ed in particolare per il dato evento E , corrispondente al valore x_i nei

campioni, viene estrapolata una finestra di campioni che va da $x_i - \Delta$ e $x_i + \Delta$, con Δ scelto arbitrariamente. E' possibile dividere gli eventi individuati in ulteriori due categorie: eventi dove la pressione registra un incremento rispetto al campione precedente ed eventi dove la pressione registra un decremento rispetto al campione precedente.

	Unforced	Forced
Eventi	15202	17929
Eventi Up	7599	9136
Eventi Down	7545	8777
Grandi Eventi	150	247

Tabella 5.5: Tabella riepilogo eventi Unforced e Forced - Caso 2

Si introduce il concetto di *Grande evento* che indica un evento in cui la condizione per cui la varianza locale supera la varianza globale si estende per un intervallo sufficientemente ampio.

Dalle figure 5.17a e 5.17b è possibile notare la forte differenza in termini di ampiezza della varianza locale tra la condizione naturale e quella di forcing. Questo risultato è coerente con i risultati precedentemente osservati con l'analisi spettrali per le fluttuazioni di pressione, dove .

I risultati riportati in tabella 5.5 sono in linea con quanto prima riportato, infatti nella condizione di forcing ci si aspetta di avere un numero di *grandi eventi* maggiore.

Nelle figure 5.18, 5.19 e 5.20 si riportano gli eventi medi condizionati sia per la condizione di forcing che per la condizione naturale.

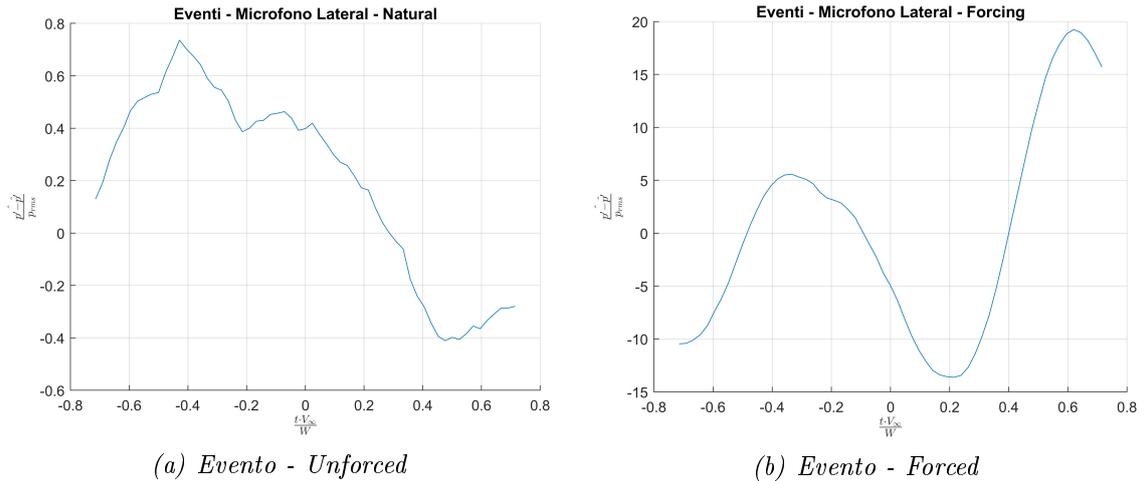


Figura 5.18: Evento - Caso 2

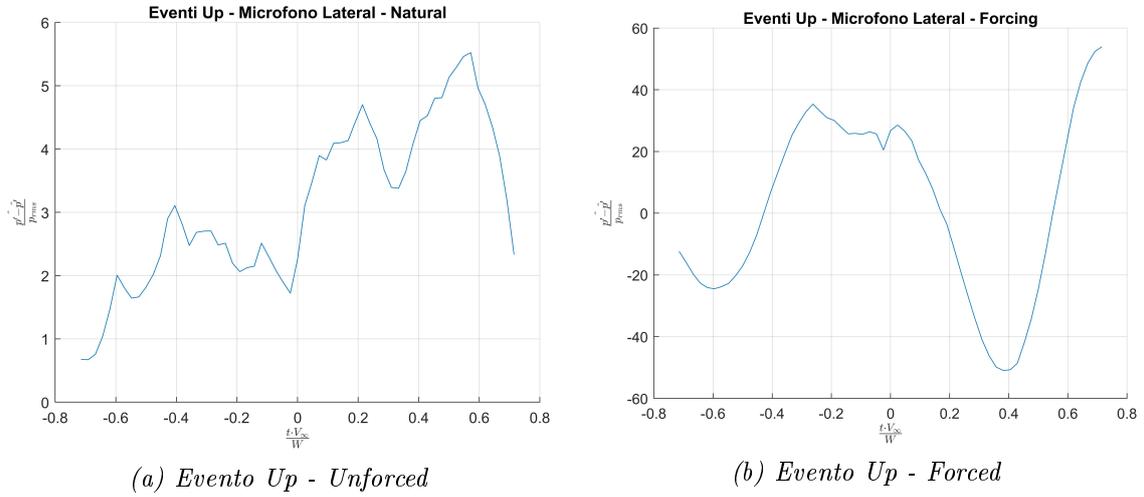


Figura 5.19: *Evento Up - Caso 2*

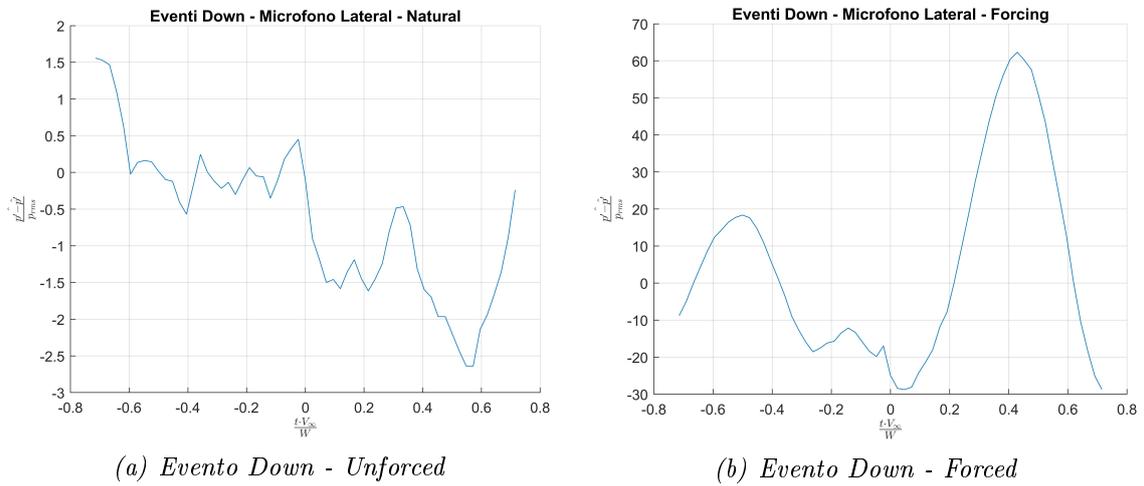


Figura 5.20: *Evento Down - Caso 2*

5.3 Caso 3: attuazione asimmetrica - $Re_\infty \sim 1.1 \cdot 10^5$

Analogamente a quanto osservato per il caso 2 (sezione 5.2) le figure 5.21a e 5.21b mostrano la fase di training e da queste si può osservare la capacità dell'agente di imparare e cioè di apprendere come aumentare il reward al progredire con i reward. Ma diversamente dal caso 2 emerge una riduzione percentuale minore, infatti mediamente si ha una riduzione del $\sim 5\%$, con picchi a $\sim 6\%$. Anche in questo caso la rete neurale è stata addestrata per 26 episodi, anche in questo caso corrispondenti ad un tempo fisico di circa 3 h e 15 minuti.

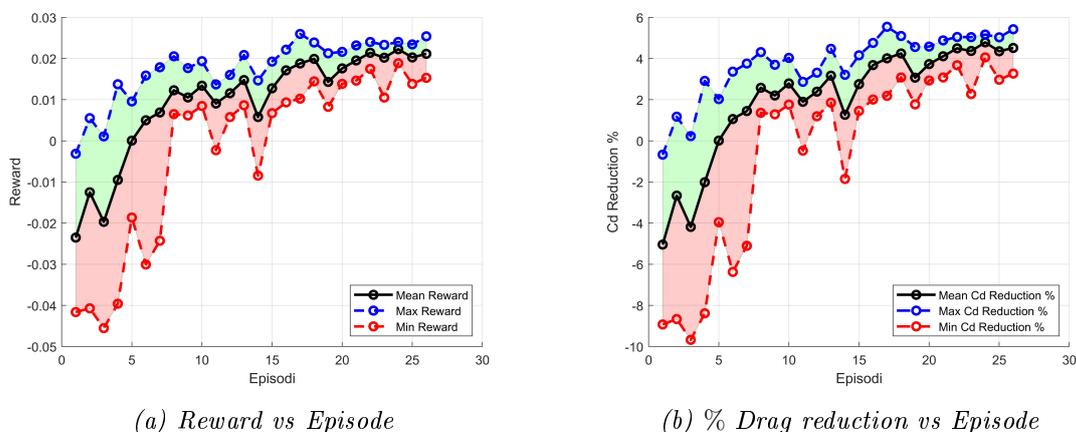


Figura 5.21: Training caso 3

Con riferimento a quanto descritto in sezione 5.2.1 in figura 5.22 viene rappresentato l'andamento del $\frac{\Delta C_D}{C_\mu}$ durante la fase di training. Diversamente dal caso precedente (sezione 5.2.1) si evidenziano dei valori di $\frac{\Delta C_D}{C_\mu}$ prossimi a 2, quindi l'agente si porta in una condizione di maggior efficienza. Ovviamente il raggiungimento di questa condizione è una conseguenza fortuita, in quanto, come più volte ribadito, l'addestramento avviene in modo tale da massimizzare il reward definito come $-(Cd - Cd_0)$ quindi senza valutazioni dal punto di vista energetico.

5.3.1 Performance dell'agente

In figura 5.23 si riporta l'andamento del Cd durante la fase di valutazione sfruttando la rete precedentemente addestrata. Si evidenzia come attivando il forcing, a partire dall'episodio 2, l'agente è in grado di ridurre il coefficiente di resistenza e quindi la resistenza.

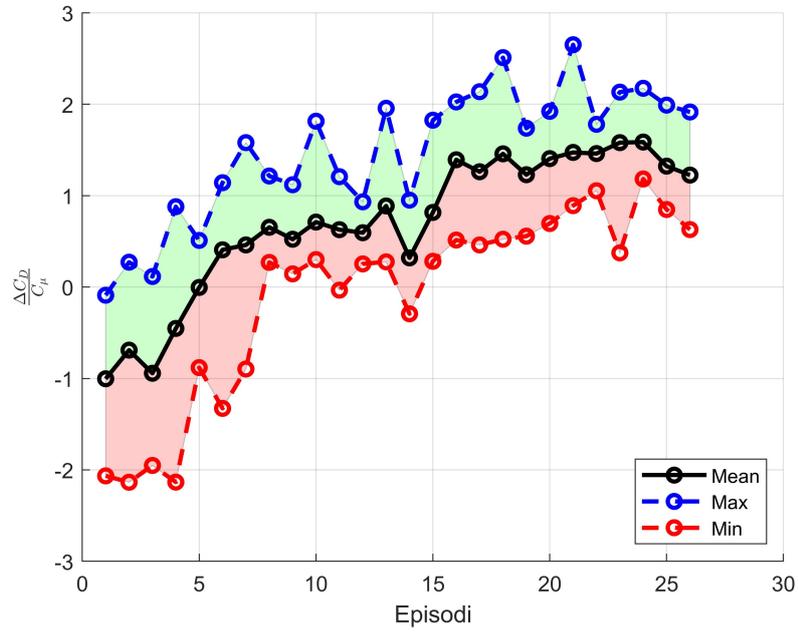


Figura 5.22: $\frac{\Delta C_D}{C_\mu}$ vs Episode

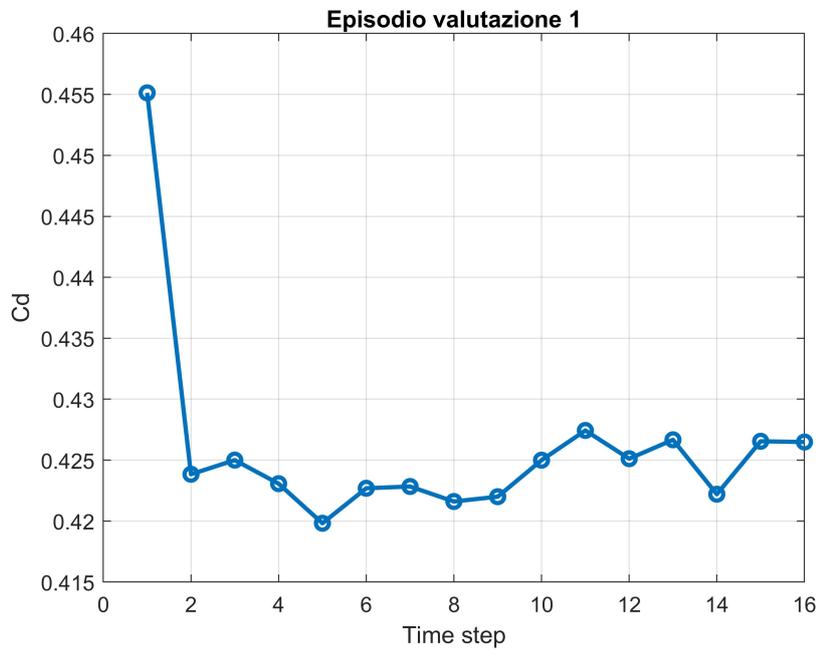


Figura 5.23: Valutazione episodi - Caso 3

Di seguito (figure 5.24 e 5.25) si riportano le distribuzioni del C_p nella base del modello durante la fase di valutazione dell'agente.

Dalle figure 5.24 e 5.25 si evince come nella condizione unforced si ha una forte regione di C_p negativo e quindi una forte azione di suction che permette di giustificare il maggiore valore di Cd rispetto al caso precedente. Infatti si osserva la presenza di $Cd_0 = 0.455$ contro $Cd_0 = 0.39$ nel caso precedente. La condizione di forcing introduce un aumento del C_p medio, quindi il forcing riduce l'azione di suction, ma comunque il C_p resta negativo, diversamente dal caso precedente in cui si è osservato come durante il forcing si passava ad un valore di C_p positivo. Le distribuzioni mostrano chiaramente la differenza tra il caso naturale e quello forzato: nel caso naturale (figura 5.24a) si osserva una zona di sovrappressione (C_p positivo) nella parte inferiore ed una zona di depressione (C_p negativo) nella parte superiore, comportamento molto simile a quanto visto nel caso a Reynolds più basso (figura 5.8a).

Dalle considerazioni fatte quindi emerge come la condizione naturale è caratterizzata da una condizione asimmetrica, comportamento simile si era osservato nel caso a Reynolds più basso. Le altre distribuzioni riportate, che rappresentano le condizioni di forcing, appaiono molto simili tra loro: presentano una zona di sovrappressione nella zona centrale della base. Diversamente dal caso precedente però non si ha uno sbilanciamento da un lato della regione di sovrappressione.

E_{maxUp} [V]	$E_{maxBottom}$ [V]	$E_{maxLateral}$ [V]
5.63	7.52	8.63

Tabella 5.6: Tensioni di forcing - caso 3

U_{Up} [m/s]	U_{Bottom} [m/s]	$U_{Lateral}$ [m/s]
0.36	16.30	11.27

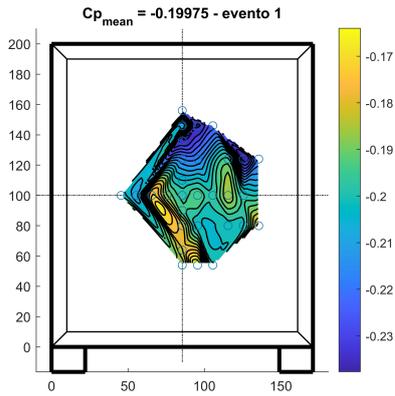
Tabella 5.7: Velocità di forcing - caso 3

$FreqUp$ [Hz]	$FreqBottom$ [Hz]	$FreqLateral$ [Hz]
24.48	7.55	47.52

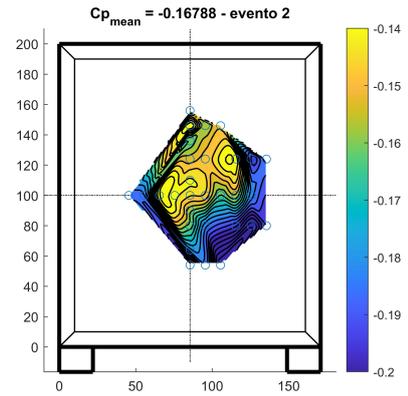
Tabella 5.8: Frequenze di forcing - caso 3

5.3.2 Analisi spettrale del segnale di pressione

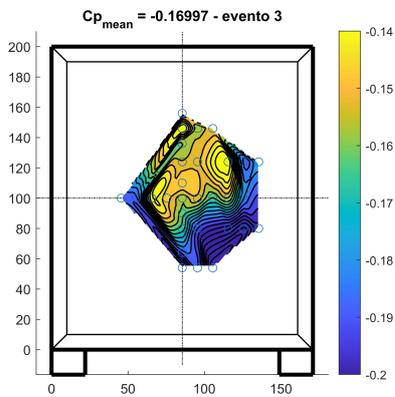
Come fatto nel caso precedente si effettua l'analisi nel dominio delle frequenze sfruttando i segnali di pressione fluttuante misurata e adimensionalizzata mediante



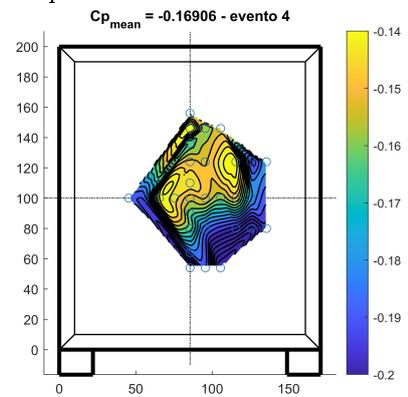
(a) Cp base distribution - unforced



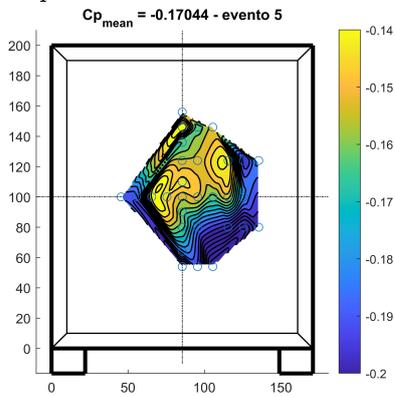
(b) Cp base distribution - time-step 1



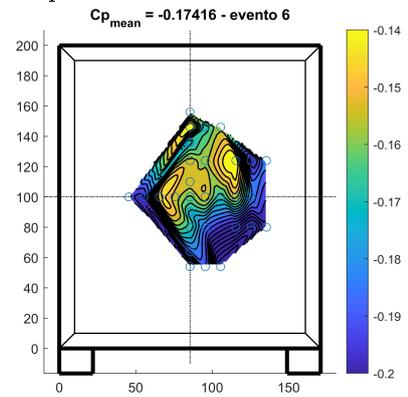
(c) Cp base distribution - time-step 2



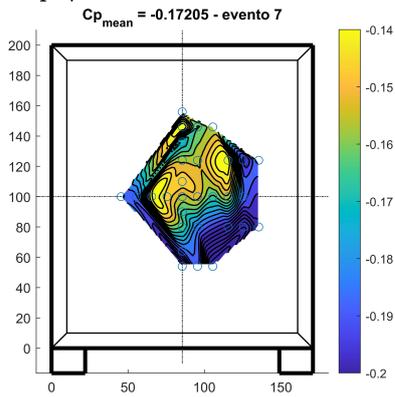
(d) Cp base distribution - time-step 3



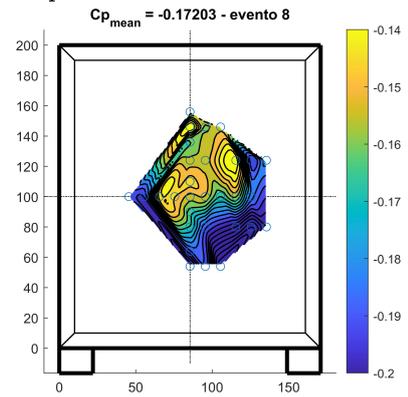
(e) Cp base distribution - time-step 4



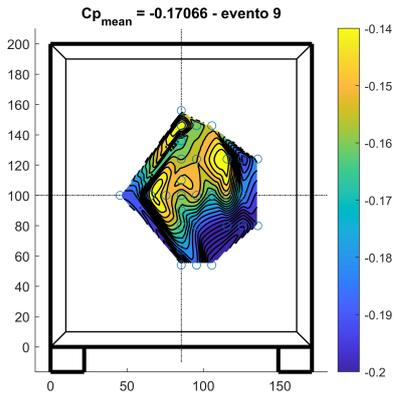
(f) Cp base distribution - time-step 5



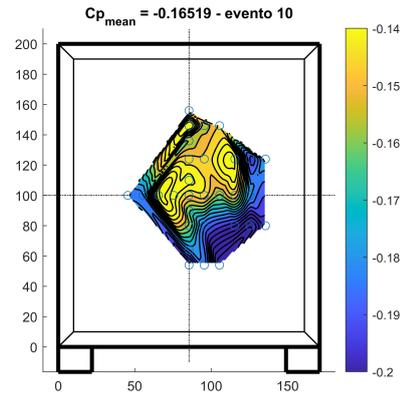
(g) Cp base distribution - time-step 6



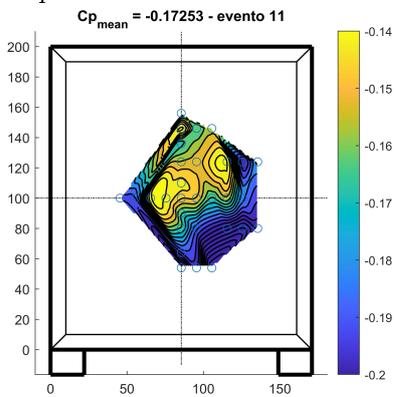
(h) Cp base distribution - time-step 7



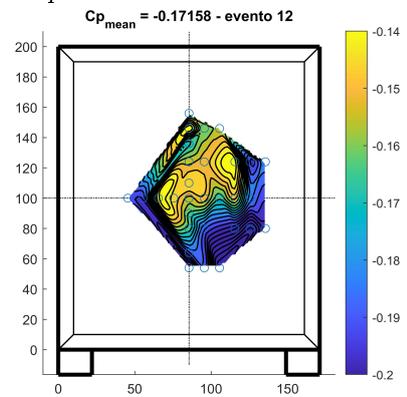
(a) C_p base distribution - time-step 8



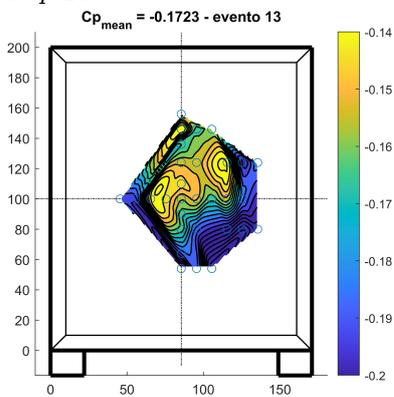
(b) C_p base distribution - time-step 9



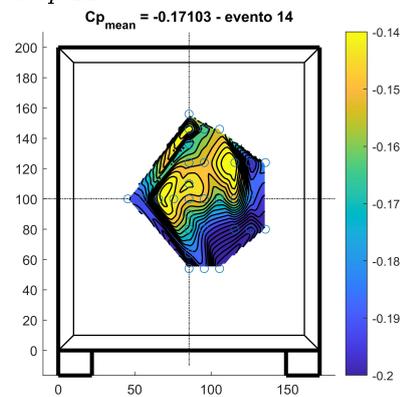
(c) C_p base distribution - time-step 10



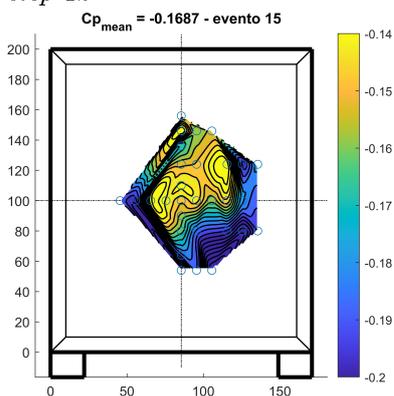
(d) C_p base distribution - time-step 11



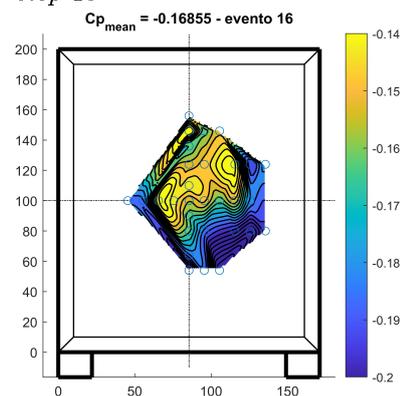
(e) C_p base distribution - time-step 12



(f) C_p base distribution - time-step 13



(g) C_p base distribution - time-step 14



(h) C_p base distribution - time-step 15

p'_{rms} , in particolare si usano i segnali derivanti dai 4 microfoni riportati in figura 5.10.

In analogia al caso 2 gli spettri nella condizione naturale mostrano un picco in corrispondenza di $St \sim 0.12$ ed un ulteriori picchi in corrispondenza di St maggiori che si è provato a giustificare in sezione 5.2.3.

Gli spettri mostrano un comportamento molto simile a quello rilevato nel caso 1. Sebbene a bassi St solo in caso del microfono in posizione *lateral 1* si osserva un valore di energia associato al caso forzato minore di quello naturale, in realtà per questo microfono per tutta la banda analizzata si evidenziano valori di energia associata minore, esclusa una stretta banda in corrispondenza della condizione di forcing.

Nello spettro nella condizione forzata si mette in evidenza la presenza di picchi attribuibili al forcing: in figura 5.26 si osservano dei picchi per valori di St equivalenti alle frequenze di forcing e multipli di queste (tabella 5.6 e 5.8).

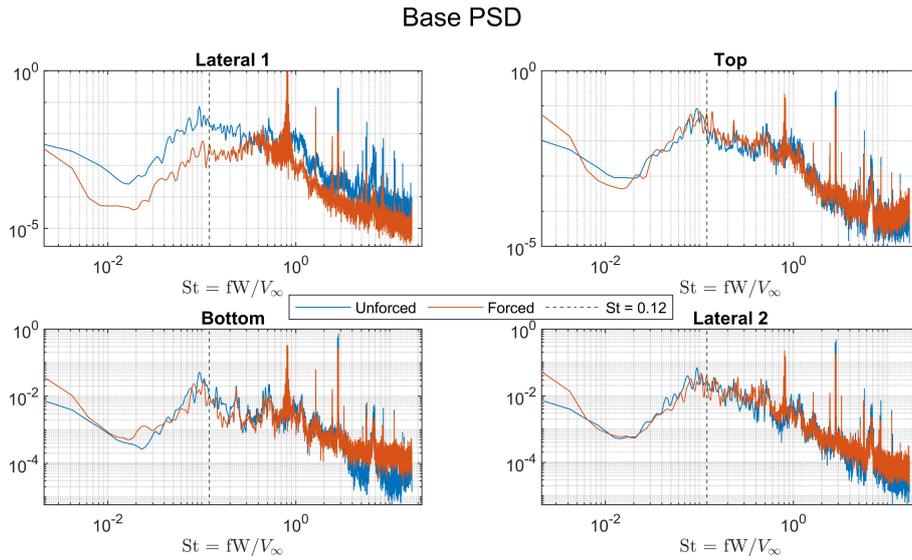


Figura 5.26: PSD - Power spectral density - caso 3

Analisi POD

Anche per questa condizione si esegue l'analisi POD come indicato in 5.2.3. Nelle figure 5.27a e 5.27b vengono riportati gli andamenti della distribuzione dell'energia tra i vari modi. Nel caso con forcing appare evidente la differenza rispetto al caso 2 (figure 5.12b): infatti qui si evidenzia come già al modo 2 è associata una percentuale di energia di $\sim 10\%$.

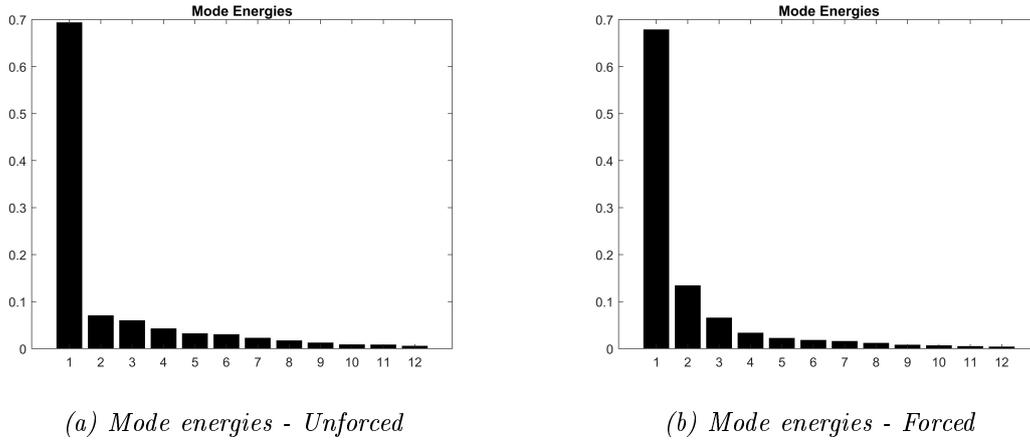


Figura 5.27: Mode energies - Caso 3

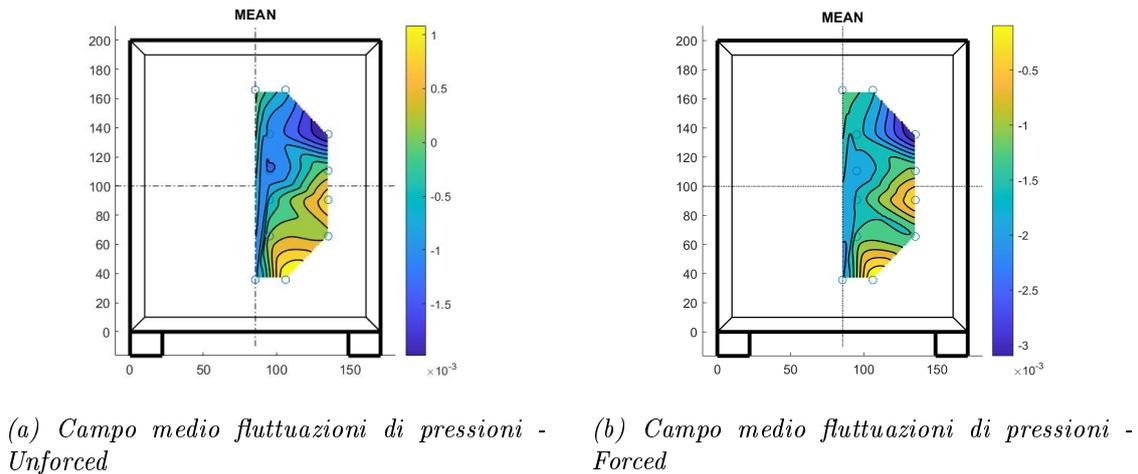


Figura 5.28: Campo medio fluttuazioni di pressioni - Caso 3

Di seguito (figure 5.29, 5.30) vengono riportati i primi 4 modi POD. Anche per questa condizione di Re , come già osservato a Re minore -sezione 5.2.3, il forcing introduce una simmetrizzazione. Infatti in figura 5.29a, condizione naturale, il primo modo sembra suggerire la presenza di uno shedding sopra-sotto, in un modo asimmetrico, diversamente in figura 5.29b, condizione forzata, si evidenzia la simmetrizzazione. Analizzando l'andamento dei coefficienti temporali e relativo spettro per il primo modo (figure 5.29c e 5.29d) si evidenzia la presenza di un picco per una frequenza $f \sim 5.6Hz$ che equivale a $St \sim 0.12$, analogamente a quanto visto nel caso 2.

Come è lecito aspettarsi la presenza del forcing introduce dei picchi in corrispondenza delle frequenze di attuazione dei getti, in particolare questo si evidenzia per

$f \sim 47.5Hz$ che dalla tabella 5.8 si osserva essere la frequenza del forcing laterale. Le energie associate alle frequenze del forcing superiore $f \sim 24Hz$ ed inferiore $f \sim 7Hz$ sono molto basse. Analizzando il modo 2 sarà possibile giustificare questo comportamento.

Il modo 2 appare come il duale del modo 1, sia nella condizione naturale che in quella di forcing (figure 5.29e e 5.29f). Come già riportato in sezione 5.2.3 la presenza di un forcing asimmetrico e la prevalenza di una direzione rispetto all'altra giustifica l'assenza di un secondo modo con energia circa uguale al primo.

Particolarmente interessante risulta l'analisi spettrale dei coefficienti temporali per il secondo modo (figure 5.29g e 5.29h). Nel caso naturale vale esattamente quanto visto in sezione 5.2.3.

Mentre nel caso forzato si evidenzia un picco rilevante in corrispondenza della frequenza del forcing inferiore, invece i picchi relativi alle altre condizioni di forcing sono molto meno intensi.

Queste considerazioni, insieme a quanto osservato per il modo 1, permettono di fare alcune osservazioni: la tabella 5.6 mostra come l'agente ha trovato una condizione di forcing tale per cui il forcing superiore è quasi spento, infatti va ricordato che la tensione minima è di $5.5V$ e il forcing applicato è di $\sim 5.6V$ e quindi nei limiti di sensibilità dell'elettrovalvola la portata introdotta da tale getto è veramente bassa, come mostrato in tabella 5.7.

Diversamente invece per quanto riguarda le altre due condizioni di forcing: da tabella 5.6 e più propriamente da tabella 5.7 si nota come la velocità di attuazione è molto maggiore. I risultati dell'analisi spettrali del modo 1 e 2 nel caso forzato quindi sembrano evidenziare che il modo 1 è particolarmente influenzato dal forcing laterale, mentre il modo 2 da quello superiore.

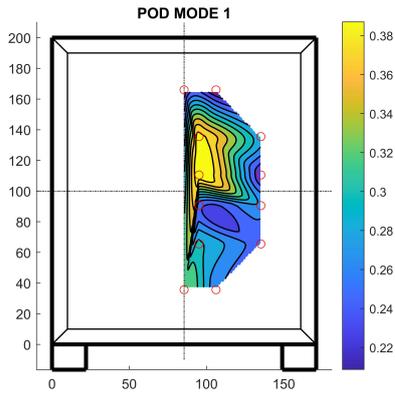
5.3.3 Ricerca moti organizzati

Anche per questo caso si esegue un'ulteriore analisi statistica sfruttando la tecnica VITA, come in sezione 5.2.4.

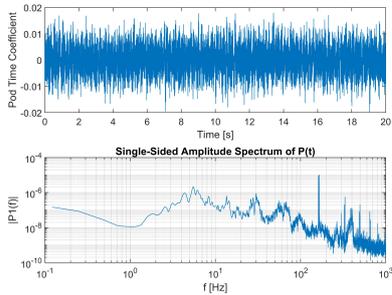
Da figure 5.31a e 5.31b si evincono le differenze tra le due condizioni come osservato per il caso 2. Anche in questo caso viene confermato il risultato atteso, cioè un valore di ampiezza della varianza locale di diversi ordini di grandezza maggiore nel caso forzato rispetto a quello naturale.

Anche i risultati riportati in tabella 5.9 mostrano quanto osservato in precedenza e cioè che il forcing fa aumentare in maniera significativa il numero di *Grandi Eventi*.

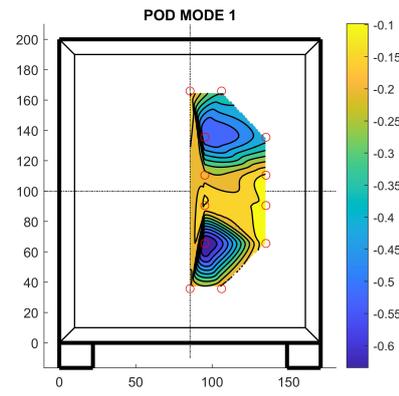
Nelle figure 5.32, 5.33 e 5.34 si riportano gli eventi medi condizionati sia per la condizione di forcing che per la condizione naturale.



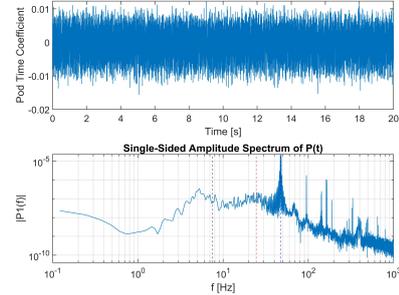
(a) Pod Mode 1 - Unforced



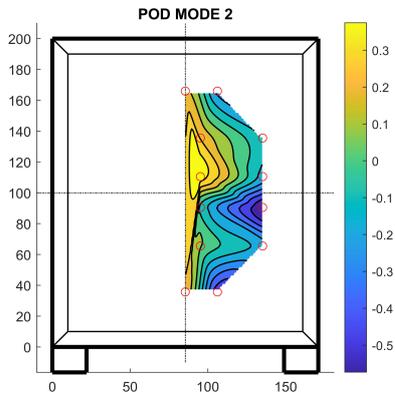
(c) Pod Mode 1 - Time Coefficient - Unforced



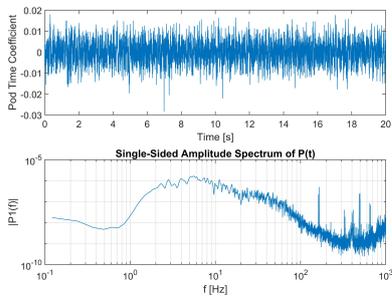
(b) Pod Mode 1 - Forced



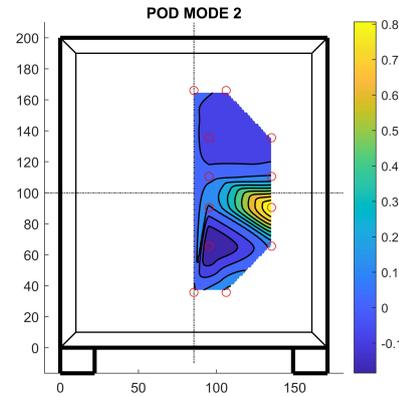
(d) Pod Mode 1 - Time Coefficient - Forced



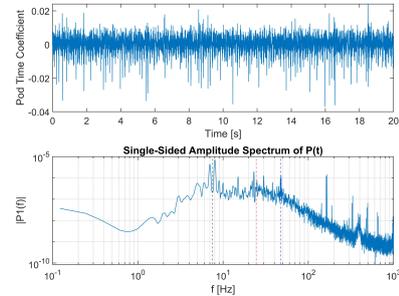
(e) Pod Mode 2 - Unforced



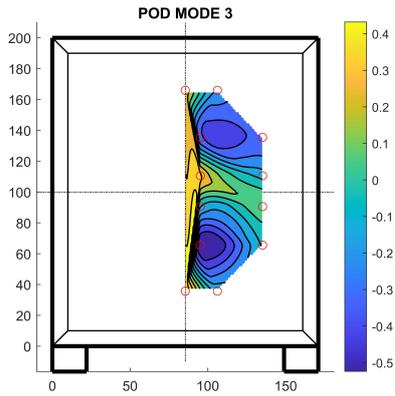
(g) Pod Mode 2 - Time Coefficient - Unforced



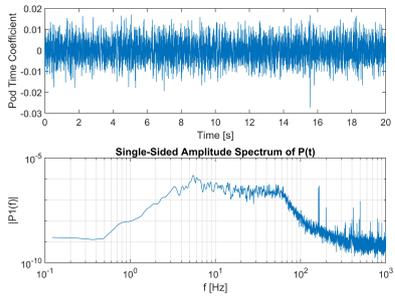
(f) Pod Mode 2 - Forced



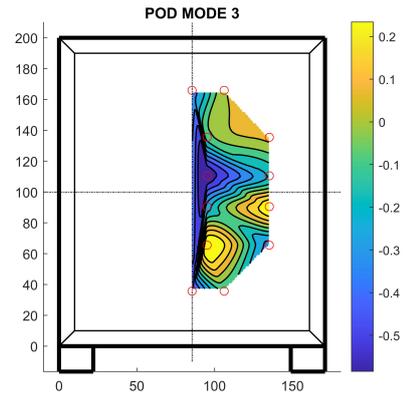
(h) Pod Mode 2 - Time Coefficient - Forced



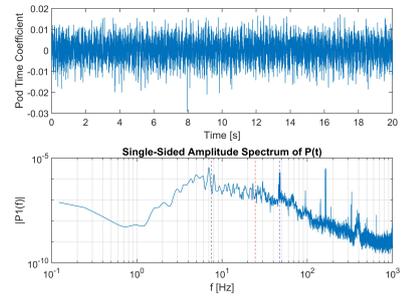
(a) Pod Mode 3 - Unforced
POD MODE 3



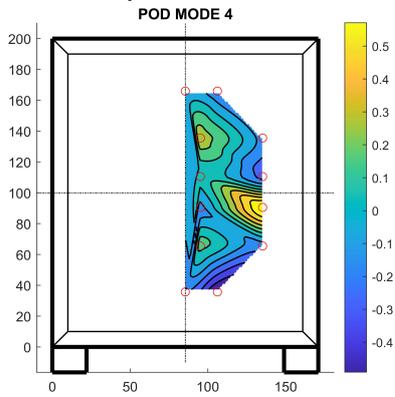
(c) Pod Mode 3 - Time Coefficient - Unforced



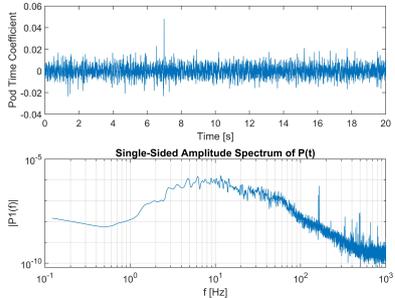
(b) Pod Mode 3 - Forced
POD MODE 3



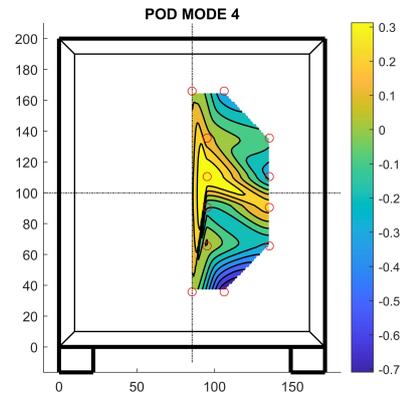
(d) Pod Mode 3 - Time Coefficient - Forced



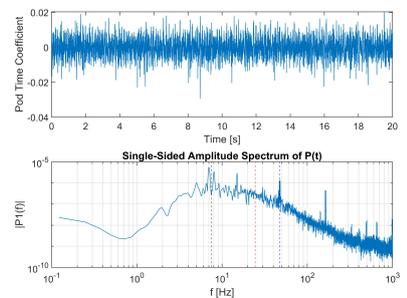
(e) Pod Mode 4 - Unforced
POD MODE 4



(g) Pod Mode 4 - Time Coefficient - Unforced



(f) Pod Mode 4 - Forced
POD MODE 4



(h) Pod Mode 4 - Time Coefficient - Forced

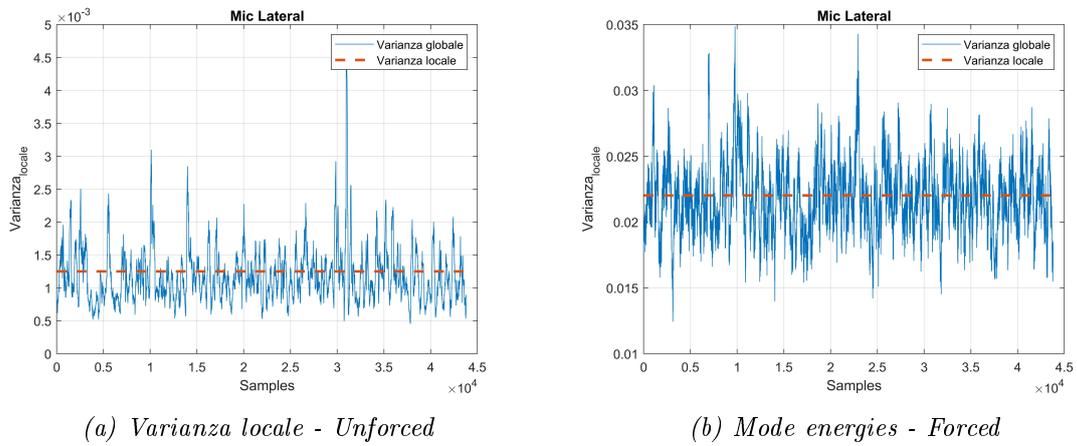


Figura 5.31: Varianza locale - Caso 3

	Unforced	Forced
Eventi	15299	20361
Eventi Up	7775	10546
Eventi Down	7498	9809
Grandi Eventi	177	507

Tabella 5.9: Tabella riepilogo eventi Unforced e Forced - Caso 2

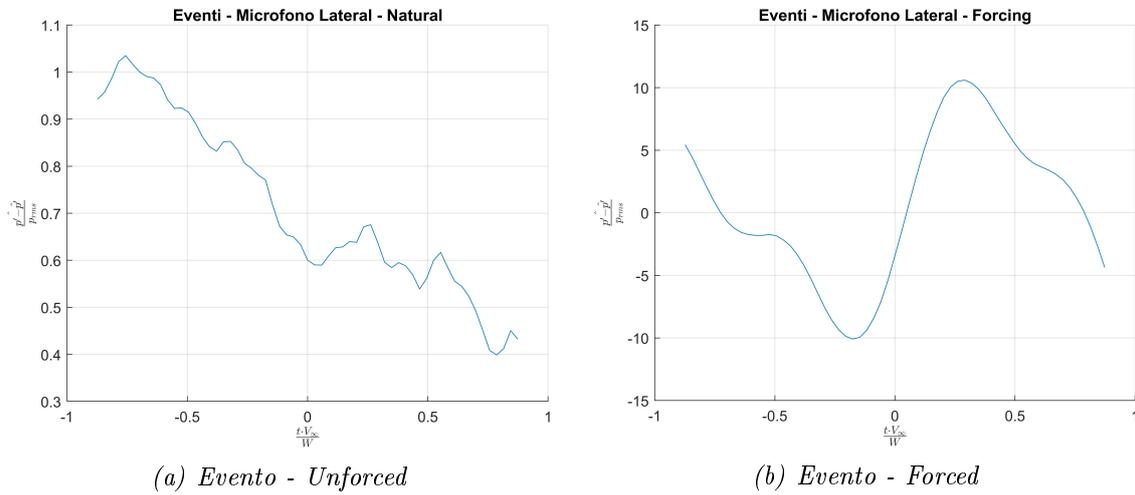


Figura 5.32: Evento - Caso 3

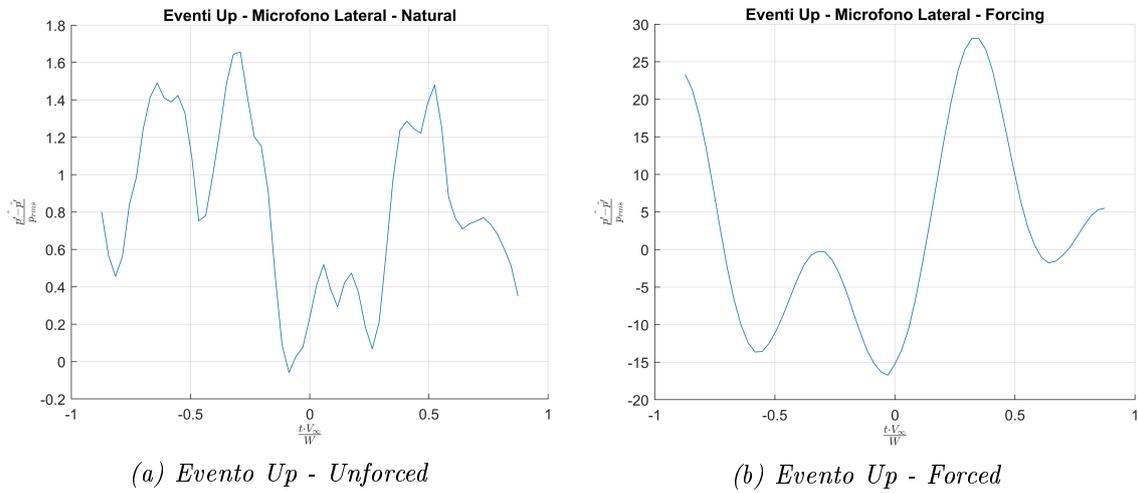


Figura 5.33: Evento Up - Caso 3

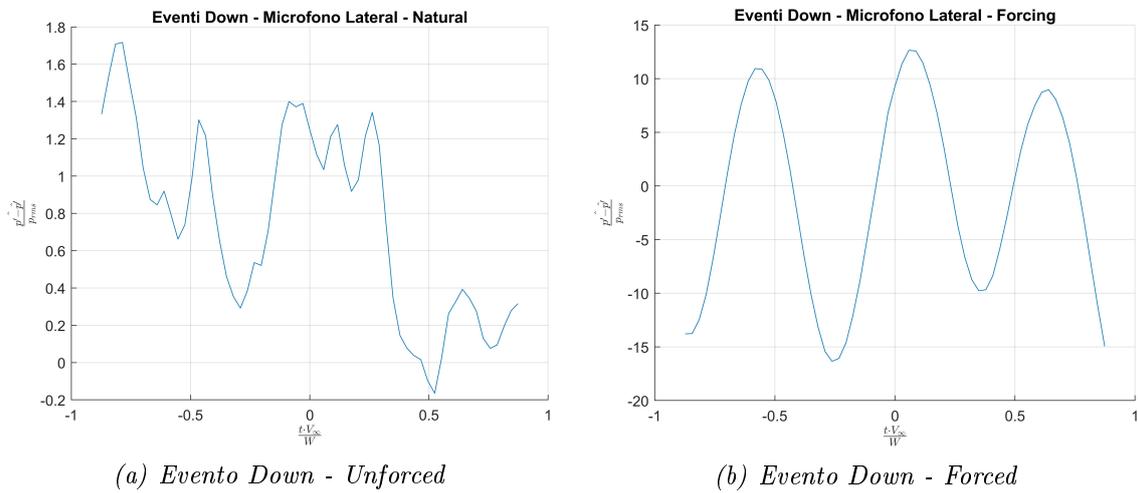


Figura 5.34: Evento Down - Caso 3

5.4 Confronto tra Caso 2 e Caso 3

Nelle sezioni 5.2 e 5.3 si sono analizzati i risultati ottenuti addestrando uno stesso agente in due diverse condizioni a monte, quindi per due diversi numeri di Reynolds. E' stato eseguito l'addestramento per lo stesso numero di episodi ed in entrambi i casi si è visto che pur mantenendo un numero relativamente basso di episodi l'agente è in grado di ridurre in modo rilevante la resistenza aerodinamica. Le due condizioni di forcing in cui si porta l'agente sono diverse tra loro, di seguito nelle tabelle 5.10, 5.11 e 5.12 si riportano valori esplicativi della condizione di forcing nei due casi:

	Caso 2	Caso 3
Frequenza [Hz]	14.8	24.48
Tensione [V]	8.5	5.63
Jet speed [m/s]	26.45	0.36

Tabella 5.10: Tabella riepilogativa forcing superiore

	Caso 2	Caso 3
Frequenza [Hz]	42.17	7.56
Tensione [V]	8.74	7.52
Jet speed [m/s]	26.45	16.3

Tabella 5.11: Tabella riepilogativa forcing inferiore

	Caso 2	Caso 3
Frequenza [Hz]	48.81	47.52
Tensione [V]	9.83	8.63
Jet speed [m/s]	15.78	11.27

Tabella 5.12: Tabella riepilogativa forcing laterale

La riduzione percentuale in termini di Cd rispetto alla condizione naturale è diversa nei due casi: nel caso 2 si raggiunge mediamente quasi il 9% mentre nel caso 3 quasi il 7%. Le tabelle 5.10, 5.11 e 5.12 mostrano chiaramente come l'agente si porti in due condizioni di forcing molto diverse: nel caso 2 c'è una simmetria relativamente alla velocità di attuazione tra forcing superiore e forcing inferiore, sebbene la frequenza del forcing inferiore è circa 3 volte quella del forcing superiore. Completamente diversa è la configurazione a cui si porta l'agente nel caso 3, qui, come già ribadito in precedenza, il getto superiore è quasi spento; la velocità

di attuazione del getto inferiore si riduce, così come si riduce la frequenza di attuazione. Il forcing laterale invece nei due casi mostra circa la stessa frequenza, con una riduzione di velocità di attuazione nel caso 2. I risultati per il caso 3 tuttavia non sono sorprendenti, infatti in letteratura si è dimostrato come la sensibilità al getto superiore è molto bassa.

Nonostante queste differenze le analisi POD condotte per entrambe i casi hanno evidenziato come la struttura fluidodinamica che viene fuori per la presenza del forcing è grossomodo la stessa, con il forcing che introduce una simmetrizzazione della naturale condizione di asimmetria.

Per avere una miglior consapevolezza di ciò che l'agente realmente apprende dal punto di vista fluidodinamico sarebbe opportuno condurre analisi di maggior dettaglio come ad esempio un'analisi PIV.

Conclusioni e sviluppi futuri

In questo lavoro di tesi si è cercato di applicare ad un problema reale la metodologia del *Reinforcement Learning*. Questa metodologia ha come obiettivo quello di costruire degli algoritmi in grado di emulare ciò che quotidianamente ogni essere umano compie: l'apprendimento per rinforzo, quindi un apprendimento basato sull'esperienza col fine di raggiungere un obiettivo; esempio classico è un bambino che impara a camminare.

Le forti potenzialità di questa metodologia hanno attratto gli interessi di molti gruppi di ricerca e grandi aziende, Google ne è un esempio, che hanno sviluppato delle librerie open-source permettendo di approcciarsi a queste tecniche con maggior facilità e maggior rapidità.

Queste famiglie di algoritmi trovano terreno fertile in ambiti dove la limitata potenza computazionale rappresenta un ostacolo spesso rilevante, arrivando spesso a costituire il collo di bottiglia del problema. Sicuramente due tra questi ambiti sono la fluidodinamica e l'ottimizzazione. Come per ogni nuova metodologia si cerca inizialmente di sviluppare e validare la stessa per problemi noti, facendo spesso uso di problemi artificiali, ed una volta validata si applica a problemi reali.

In quest'ottica possono essere letti i diversi lavori in letteratura, ad esempio il lavoro di [Rabault et al.](#) dove il *Deep Reinforcement Learning* viene utilizzato per effettuare il controllo attivo, mediante getti, su di un cilindro in modo tale da ridurre la resistenza aerodinamica. Questo è uno dei primi lavori che affianca alla fluidodinamica e controlla la metodologia del *Reinforcement Learning*, infatti l'intero problema è formulato numericamente.

L'approccio numerico mostra come principale limite il costo computazionale, l'esecuzione di una simulazione numerica su di un dominio particolarmente complesso è tale per cui il solo tempo di calcolo di un evento potrebbe essere equivalente a ore piuttosto che giorni.

Il passaggio dal caso numero al caso sperimentale da un lato elimina il limite legato alla potenza computazionale ma porta con sé una moltitudine di problematiche pratico-implementative. Tra queste sicuramente la complessità legata allo sviluppo di un modello attrezzato per la misura e tutte le problematiche che nascono nel momento in cui si vuole eseguire una data misura, giusto per citarne uno: la convergenza statistica. Aspetto cruciale inoltre è l'impossibilità nel misurare alcune

grande.

L'approccio sperimentale è proprio quello che si è cercato di seguire in questo lavoro di tesi con l'obiettivo di controllare tramite getti la scia di un modello di un veicolo da trasporto merci mediante la tecnica del *Reinforcement Learning* al fine di ridurre la resistenza aerodinamica. Oltre a dimostrare il concreto funzionamento della metodologia in problemi di questo tipo, questo problema rappresenta un caso molto interessante dal punto di vista pratico in quanto, come riportato nel capitolo 1 la riduzione della resistenza aerodinamica è un filone molto importante della ricerca attuale in campo fluidodinamico per le sue ricadute economiche e sull'ambiente.

Con questo lavoro di tesi si è riusciti a dimostrare la possibilità di applicare la metodologia del *Reinforcement Learning* anche a problemi con alto numero di Reynolds, infatti nel momento della scrittura di questo lavoro all'autore non sono note applicazioni simili in letteratura. Si è visto che si riesce a ridurre in maniera consistente la resistenza aerodinamica, in particolare si è ottenuto una riduzione percentuale del coefficiente di resistenza pari a $\sim 7\%$ per $Re \sim 1.1 \cdot 10^5$.

Questa riduzione è particolarmente rilevante considerando la tipologia di modello a cui si è applicato la metodologia, cioè un veicolo per trasporto merci: qui infatti si è visto in 1 che per via delle loro condizioni operative la resistenza aerodinamica incide anche per il $\sim 20 - 30\%$ sui costi operativi.

Un altro risultato su cui porre l'attenzione è il tempo necessario per eseguire l'addestramento. Infatti l'agente è stato addestrato in entrambi i casi (5.2 e 5.3) per 26 episodi ed ognuno dei quali contiene 16 eventi. Considerando il tempo di acquisizione e di elaborazione dei dati acquisiti, ogni evento necessita di ~ 30 secondi. Quindi in totale sono state necessarie 3 h e 25 minuti per eseguire il training. L'andamento del reward con l'avanzare degli episodi accenna il raggiungimento del plateau, quindi questo fa intendere che proseguendo con l'addestramento per un numero di episodi maggiore di 26, le performance migliorano di piccole quantità.

Questo tempo sembra essere particolarmente basso se confrontato con i tempi necessari per addestrare gli agenti in problemi di controllo su casi numerati presenti in letteratura.

Se da un lato con questo lavoro si sono date molte risposte dall'altro sono nati molti interrogativi che aprono le porte a possibili sviluppi futuri. Verificato il funzionamento della metodologia infatti sarebbe molto interessante andare a verificare se i risultati ottenuti convergono con quanto già presente in letteratura. Ad esempio permettendo all'agente di variare non solo la frequenza e la tensione di attuazione ma anche il duty cycle.

Un altro studio interessante sarebbe l'analisi del comportamento di altre tipologie di agenti tra quelli presenti in letteratura. Molto importante sarebbe verificare la capacità dello stesso agente addestrato in una data condizione di raggiungere un elevato reward anche quando utilizzato in condizioni off-design, ad esempio: in condizione di platooning, o dando un angolo di imbardata diverso da quello presente in fase di training o per una velocità nell'intervallo compreso tra due velocità a cui

si svolge il training.

Un'altra possibile indagine riguarda l'implementazione del pre-train: cioè la possibilità di utilizzare dei risultati già a disposizione per pre allenare l'agente e verificare se questa condizione influenza positivamente il tempo di convergenza del metodo.

Ringraziamenti

Vorrei approfittare di queste ultime righe per ringraziare coloro che mi hanno sostenuto in questo percorso. Ringrazio la mia famiglia, Giorgia e tutti gli amici che hanno saputo essere nel posto giusto e nel momento giusto. Infine vorrei ringraziare il Prof. Gaetano Iuso e l'Ing. Gioacchino Cafiero per aver sostenuto la mia proposta di tesi, aver creduto in questo lavoro, avermi fornito gli strumenti e le conoscenze necessarie e aver fatto accrescere in me la passione per la fluidodinamica.

Bibliografia

- [1] Richard S. Sutton and Andrew G. Barto. Reinforcement Learning: An Introduction. MIT Press, 2015.
- [2] Wolf-Heinrich Hucho and Gino Sovran. Aerodynamics of road vehicles. Annual Review of Fluid Mechanics, 25, 1993.
- [3] Brandon Schoettle, Michael Sivak, and Michael Tunnell. A survey of fuel economy and fuel usage by heavy-duty truck fleets. American Transportation Research Institute, 2016.
- [4] P. Gehlert, G. Cafiero, and J.C. Vassilicos. Effect of fractal endplates on the wingtip vortex. Proceedings of the 2018 AIAA Aerospace Sciences Meeting, (210059), 2018. doi: 10.2514/6.2018-1796.
- [5] A. Capone and G.P. Romano. Investigation on the effect of horizontal and vertical deflectors on the near-wake of a square-back car model. Journal of Wind Engineering & Industrial Aerodynamics, 185, 2019. doi: 10.1016/j.jweia.2018.12.011.
- [6] Gregory Pujals, S. Depardon, and Carlo Cossu. Drag reduction of a 3d bluff body using coherent streamwise streaks. Experiments in Fluids, 49:1085–1094, 11 2010. doi: 10.1007/s00348-010-0857-5.
- [7] Jean-François Beaudoin and J.-L. Aider. Drag and lift reduction of a 3d bluff body using flaps. Experiments in Fluids, 44:491–501, 04 2008. doi: 10.1007/s00348-007-0392-1.
- [8] Dongri Kim, Hoon Lee, Wook Yi, and Haecheon Choi. A bio-inspired device for drag reduction on a three-dimensional model vehicle. Bioinspiration & Biomimetics, 11(2):026004, mar 2016. doi: 10.1088/1748-3190/11/2/026004. URL <https://doi.org/10.1088/1748-3190/11/2/026004>.
- [9] G. Iuso. Base pressure control by passive methods. doi: 10.2514/6.1985-1620. URL <https://arc.aiaa.org/doi/abs/10.2514/6.1985-1620>.

- [10] G. Minelli, T. Dong, B. R. Noack, and S. Krajnović. Upstream actuation for bluff-body wake control driven by a genetically inspired optimization. Journal of Fluid Mechanics, 893, 2020.
- [11] O. Evstafyeva, A.S. Morgans, and L.Dalla Longa. Simulation and feedback control of the ahmed body flow exhibiting symmetry breaking behaviour. Journal of Fluid Mechanics, 917, 2017.
- [12] S.R. Ahmed, G.Ramm, and G.Faltin. Some salient features of the time-average d ground vehicle wake. SAE-Techn. Paper Series., 1984.
- [13] Y. Haffner, J. Borée, A. Spohn, and T. Castelain. Mechanics of bluff body drag reduction during transient near-wake reversals. Journal of Fluid Mechanics, 894, 2020.
- [14] M. Grandemange, M. Gohlke, and O. Cadot. Reflectional symmetry breaking of the separated flow over three-dimensional bluff bodies. Physical review. E, 83, 2012.
- [15] M. Grandemange, M. Gohlke, and O. Cadot. Bi-stability in the turbulent wake past parallelepiped bodies with various aspect ratios and wall effects. Physics of Fluids, 725, 2013.
- [16] G. Rigas, A.R. Oxlade, A.S. Morgans, and J. F. Morrison. Low-dimensional dynamics of a turbulent axisymmetric wake. Journal of Fluid Mechanics, 755, 2014.
- [17] Diogo Barros, Jacques Borée, Bernd R. Noack, Andreas Spohn, and Tony Ruiz. Bluff body drag manipulation using pulsed jets and coanda effect. Journal of Fluid Mechanics, 805, 2016.
- [18] Ruiying Li, Diogo Barros, Jacques Borée, Olivier Cadot, Bernd R. Noack, and Laurent Cordier. Feedback control of bimodal wake dynamics. Experiments in Fluids, 57(10), October 2016. ISSN 0723-4864. doi: 10.1007/s00348-016-2245-2. Publisher Copyright: © 2016, Springer-Verlag Berlin Heidelberg. Copyright: Copyright 2017 Elsevier B.V., All rights reserved.
- [19] Diogo Barros, Jacques Borée, Bernd R. Noack, and Andreas Spohn. Resonances in the forced turbulent wake past a 3d blunt body. Physics of Fluids, 2016.
- [20] Juan José Cerutti, Costantino Sardu, Gioacchino Cafiero, and Gaetano Iuso. Active flow control on a square-back road vehicle. Fluids, 2020.
- [21] Christopher M. Bishop. Pattern recognition and machine learning. Springer, 2006.

- [22] Vincent Francois-Lavet, Peter Henderson, Riashat Islam, Marc G. Belle-mare, and Joelle Pineau. An introduction to deep reinforcement learning. Foundations and Trends in Machine Learning, 11, 2018.
- [23] Beakcheol Jang, Myeonghwi Kim, Gaspard Harerimana, and Jong Wook Kim. Q-learning algorithms: A comprehensive classification and applications. IEEE Access, 7:133653–133667, 2019. doi: 10.1109/ACCESS.2019.2941229.
- [24] Jean Rabault, Miroslav Kuchta, Atle Jensen, Ulysse Reglade, and Nicolas Cerardi. Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control. Journal of Fluid Mechanics, 865, 2019.
- [25] JVolodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. 2013.
- [26] Tensorforce. <https://tensorforce.readthedocs.io/en/latest/>.
- [27] P. W. Bearman, D. De Beer, E. Hamidy, and J. K. Harvey. The effect of a moving floor on wind-tunnel simulation of road vehicles. SAE Technical Paper 880245, 1988. doi: 10.1109/ACCESS.2019.2941229.
- [28] C. Sardu, D. Lasagna, and G. Iuso. Noise filtering for wall pressure fluctuations in measurements around a cylinder with laminar and turbulent flow separation. Journal of Fluids Engineering, 2016. doi: 10.1115/1.4032034.
- [29] Juan José Cerutti, Costantino Sardu, Gioacchino Cafiero, and Gaetano Iuso. Active flow control on a square-back road vehicle. Fluids, 5(2), 2020. ISSN 2311-5521. doi: 10.3390/fluids5020055. URL <https://www.mdpi.com/2311-5521/5/2/55>.
- [30] Haecheon Choi, Woo-Pyung Jeon, and Jinsung Kim. Control of flow over a bluff body. Annual Review of Fluid Mechanics, 40(1):113–139, 2008. doi: 10.1146/annurev.fluid.39.050905.110149. URL <https://doi.org/10.1146/annurev.fluid.39.050905.110149>.
- [31] Robert Englar. Advanced aerodynamic devices to improve the performance, economics, handling, and safety of heavy vehicles. 05 2001. doi: 10.4271/2001-01-2072.
- [32] Guillaume Bonnavion and Olivier Cadot. Unstable wake dynamics of rectangular flat-backed bluff bodies with inclination and ground proximity. Journal of Fluid Mechanics, 854:196–232, 2018. doi: 10.1017/jfm.2018.630.

- [33] Juan José Cerutti, Gioacchino Cafiero, and Gaetano Iuso. Aerodynamic drag reduction by means of platooning configurations of light commercial vehicles: A flow field analysis. International Journal of Heat and Fluid Flow, 90, 2021.
- [34] M. Grandemange, M. Gohlke, and O. Cadot. Turbulent wake past a three-dimensional blunt body. part 1. global modes and bi-stability. Journal of Fluid Mechanics, 722, 2013.
- [35] G Berkooz, P Holmes, and J L Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. Annual Review of Fluid Mechanics, 25 (1):539–575, 1993. doi: 10.1146/annurev.fl.25.010193.002543. URL <https://doi.org/10.1146/annurev.fl.25.010193.002543>.
- [36] R. F. Blackwelder and R. E. Kaplan. On the wall structure of the turbulent boundary layer. Journal of Fluid Mechanics, 76:89–112, July 1976. doi: 10.1017/S0022112076003145.