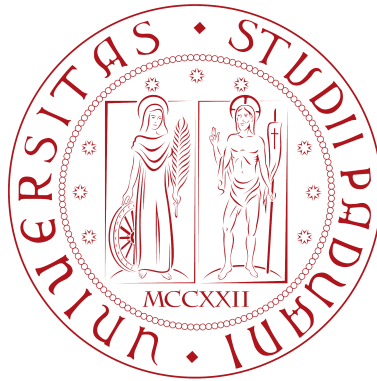




Università degli studi di Padova
Dipartimento di Ingegneria Industriale
Laurea magistrale in Ingegneria Meccanica

MIGLIORAMENTO DEL COMFORT DEL VEICOLO UTILIZZANDO UN APPROCCIO IA

Tesi di Laurea Magistrale in Ingegneria Meccanica

Relatore: Prof. Matteo Massaro
Correlatore: Prof. Davide Tavernini
Correlatore: Prof. Umberto Montanaro

Laureando: MARCO ADAMI

Padova
13 Dicembre 2024

*L'universo ha un inizio, ma non una fine. È infinito.
E poi le stelle, anch'esse hanno un inizio
ma periranno per via della loro stessa energia: si estingueranno.
Coloro che possiedono il dono dell'intelligenza sono in verità i più sciocchi;
è piuttosto evidente, basta osservare la storia.
I pesci che vivono in mare non conoscono la terraferma.
Se ne fossero consapevoli si estinguerebbero.
Che l'uomo superi la velocità della luce
è più ridicolo dei pesci sulla terraferma.
In un certo senso, questo può essere considerato
l'ultimo avvertimento di Dio a coloro che resistono:
l'epigrafe dei tenaci.*

RINGRAZIAMENTI

Questa tesi rappresenta la degna conclusione di un percorso ricco di sforzi e soddisfazioni, perciò è doveroso ringraziare in queste poche righe chiunque abbia contribuito ad arricchire tale percorso, anche se probabilmente qualunque siano le mie parole sicuramente non basterebbero ad esprimere tutta la mia gratitudine. Per cominciare, la prima persona che desidero ringraziare è colui che ha portato a termine tutto questo: me stesso. Complimenti Marco, la tua costanza e determinazione sono state più forti di tutti gli ostacoli che ti sono apparsi davanti, ma in fondo tu l'hai sempre saputo: l'unico limite è quello che decidi tu stesso. Tuttavia nessun eroe, nessun genio, nessun protagonista di qualsiasi storia porta a conclusione la sua impresa senza avere al suo fianco delle persone fidate che la rendano possibile. E la prima persona che è sempre stata al mio fianco è proprio la prima che mi ha visto al mondo: mia mamma, o meglio, la Ale. Partendo dalla lontana prima elementare, lei è sempre stata pazientemente al mio fianco, ascoltando incessantemente le mie lezioni per essere pronto al meglio per qualsiasi esame o verifica. Probabilmente riuscirebbe a diventare ingegnere senza alcun problema ad oggi. Lei, assieme a tutta la mia famiglia: Walter, Sara, Melissa, nonno Giuseppe e la nonna Teresa mi sono stati accanto, senza mettermi alcuna pressione sul futuro e confortandomi sul presente. Se sono qui oggi è soprattutto grazie a voi, per cui grazie mille. Parlando di famiglia poi è doveroso citare la mia seconda: i miei amici. Credo sia quasi impossibile trovare persone come noi che hanno condiviso quasi tutta la vita insieme, ridendo assieme nei momenti felici e facendosi forza in quelli tristi. Grazie Alberto, Mattia, Francesco, i due Nicolò, Alessandro, Riccardo, Davide, Luca, Antonio assieme a Camilla, Alice, Elisa, Chiara, Nicole e tutta la compagnia siete complici di questo mio successo e dei prossimi che arriveranno. Grazie di cuore a tutti. Poi vorrei ringraziare una delle persone più importanti nella mia vita: Letizia. Mi hai aiutato a crescere dal punto di vista personale e professionale, con i tuoi utili consigli e (alcuni) rimproveri, dandomi conforto e sostegno lungo questo difficile percorso, aiutata anche da una fantastica famiglia. Grazie a tutti voi. Passando al lato universitario, porgo i miei più sentiti ringraziamenti al professor Massaro: una figura che mi ha accompagnato lungo il mio percorso universitario con la sua competenza, professionalità e soprattutto alla sua disponibilità verso i suoi studenti, diventando nel tempo un mentore professionale. Ringrazio poi tutti i ragazzi di QuartodiLitro, con i quali ho condiviso 3 anni incredibili, conditi da infinite soddisfazioni e giri in moto! Grazie ragazzi. Questo lavoro è anche stato possibile grazie alla University of Surrey, in particolare i professori Tavernini e Montanaro che mi hanno accolto nei loro uffici, fornendomi tutto ciò di cui avessi bisogno per lavorare al meglio, creando uno splendido rapporto durante la mia permanenza all'estero. Parlando di Surrey è impossibile

non citare chi ha reso infinitamente piacevole la mia esperienza e supportato nel mio lavoro: grazie Pietro, Carmine, Eugenio, Daniele e Mario per la vostra disponibilità ed amicizia. Non sono mai stato un grande scrittore, me la cavo meglio con i numeri che con le parole, però spero che queste semplici righe abbiano fatto comprendere in minima parte la stima e riconoscenza che ripongo nei confronti di ognuno di voi. Grazie Mille.

ABSTRACT

Negli ultimi anni, l'industria automobilistica ha subito una rapida evoluzione, con un crescente interesse per i veicoli elettrici (EV) come risposta alle sfide ambientali. In questo contesto, il miglioramento del comfort di guida e della sicurezza del veicolo è diventato un fattore cruciale per l'adozione diffusa degli EV. In particolare, le vibrazioni longitudinali del veicolo, spesso trascurate rispetto a quelle verticali, possono avere un impatto significativo sul comfort di guida. In letteratura problemi di questo tipo vengono gestiti tipicamente tramite controllori di coppia di tipo NMPC, il quale costituirà il punto di partenza dell'attività descritta. Questo lavoro propone un controllore di coppia basato sul Reinforcement Learning (RL), volto a ridurre le vibrazioni longitudinali. Il controllore proposto, che utilizza l'algoritmo Deep Deterministic Policy Gradient (DDPG), è stato implementato su un modello di veicolo precedentemente validato con architettura dei motori In-wheel (IWM). L'agente è stato addestrato su diversi scenari stradali e testato in simulazioni che includevano ostacoli come scalini e dossi, oltre a profili stradali reali campionati sperimentalmente. Sono presentate diverse alternative di controllori con complessità crescente al fine di mostrare l'evoluzione durante la fase di progettazione, modificando scenari utilizzati in addestramento e input dell'agente. Si è arrivati infine ad introdurre un modello di previsione anche per il controllore RL, al fine di incrementarne le performance e permettere un confronto equo con la controparte NMPC. I risultati mostrano che il controller RL proposto offre prestazioni paragonabili a quelle di un Nonlinear Model Predictive Controller all'avanguardia in termini di riduzione delle vibrazioni, mantenendo al contempo un carico computazionale inferiore, rendendolo adatto per applicazioni in tempo reale nei veicoli stradali. Le performance sono valutate tramite l'utilizzo di alcuni Key Performance Index (KPI), tra i quali il valore efficace (o RMS) dell'accelerazione longitudinale della massa sospesa $RMS_{\ddot{x}_b}$, su una serie di prove quali: singolo gradino, dosso e profilo stradale reale. Sono stati inoltre condotti test di robustezza per valutare la risposta del controller a diverse condizioni stradali e parametri del veicolo non utilizzati durante l'addestramento, confermando la generalizzazione e robustezza dell'azione di controllo dell'approccio proposto. Infine, il controller è stato implementato su un dispositivo dSPACE MicroAutoBox III per testarne i tempi di calcolo, mostrando un tempo di risposta $< 1\text{ms}$ e garantendo performance simili a quanto visto durante la simulazione, risultando quindi adatto ad applicazioni su veicoli reali.

CONTENTS

1	Introduzione	1
2	Simulation Environment	5
2.1	Vehicle Model	6
2.2	MF-Swift	7
3	Sistemi di Controllo	11
3.1	Non Linear Model Predictive Control	11
3.1.1	Modello di inviluppo	12
3.1.2	Modello di Previsione	14
3.1.3	Problema di controllo ottimo non lineare	18
3.2	Reinforcement Learning	20
3.2.1	Deep Deterministic Policy Gradient (DDPG)	23
3.3	Controllore Proposto	24
3.3.1	Spazio degli Stati s	24
3.3.2	Spazio delle Azioni a	26
3.3.3	Reward Function	26
3.3.4	Architettura del Controllore	27
3.3.5	Introduzione della Preview	28
4	Training e Test	31
4.1	Scenari di Addestramento	31
4.2	Setup e Criticità Ricontrate	35
4.2.1	Interferenza Catastrofica	35
4.2.2	Hardware utilizzato	36
4.2.3	Ricavare il Best Agent	37
4.3	Training degli agenti	40
4.3.1	Training preliminare (scenario singolo e senza preview)	40
4.3.2	Training con preview e scenario singolo	42
4.4	Addestramento Finale e confronto con gli agenti precedenti	44
5	Risultati e Test di Robustezza	51
5.1	Test su controllore senza preview	51
5.2	Test su controllore con preview	53
5.2.1	Test di robustezza	57
5.2.2	Analisi in Frequenza	63
6	Implementazione su Veicolo Reale	67
6.1	Test su sistema dSPACE	67
7	Conclusioni	71

1 INTRODUZIONE

Negli ultimi anni, l'industria automobilistica ha assistito a una rapida evoluzione, caratterizzata da un crescente interesse per i veicoli elettrici (EV), come risposta alle sfide ambientali e all'aumento dei costi dei combustibili fossili. Tra i numerosi fattori che influenzano l'accettazione e la diffusione di questi veicoli, i miglioramenti nella sicurezza e nel comfort di guida si distinguono come considerazioni fondamentali per gli utenti finali. La dinamica verticale gioca un ruolo centrale nel garantire il comfort e la sicurezza dei passeggeri. Questa si concentra sulla risposta del veicolo alle forze che agiscono lungo l'asse verticale del veicolo, dovute principalmente alle irregolarità della strada. Un'efficace gestione della dinamica verticale è essenziale per ridurre le vibrazioni indesiderate che vengono trasmesse al corpo del veicolo e, di conseguenza, ai passeggeri. Studi come quelli di [1] dimostrano come le sospensioni attive e semi-attive siano in grado di mitigare efficacemente queste vibrazioni attraverso un controllo avanzato. Questi sistemi, infatti, regolano costantemente la rigidità e lo smorzamento delle sospensioni in risposta alle condizioni stradali, migliorando notevolmente il comfort di guida. Nonostante il grande interesse sulla dinamica verticale, le vibrazioni longitudinali si sono dimostrate di pari intensità e importanza, specialmente nei veicoli elettrici [2] [3].

Il controllo della coppia motore nei EV si pone come una delle migliori strategie a tale scopo, ottimizzando il comfort di guida tramite una precisa gestione del veicolo [4]. La configurazione con motori "In-Wheel" (IWM) ha dimostrato di essere l'ideale per un problema di torque vectoring grazie alla mancanza di un albero di trasmissione, che facilita la modulazione precisa della coppia alle ruote, migliorando al contempo il layout interno del veicolo e riducendo il peso complessivo del sistema di trasmissione. Tuttavia, questa configurazione aumenta il rapporto massa sospesa/massa non sospesa, influenzando negativamente il comfort di guida, solitamente espresso utilizzando degli indicatori di performance basati sull'accelerazione del veicolo, come evidenziato in lavori recenti [5] [6], [7]. Diversi esempi di controllori che utilizzano questa configurazione sono presenti in letteratura. Ad esempio, i lavori di Walz [8], [9] propongono un controllore MPC (Model Predictive Control), accoppiato con un FFC (Feedforward Controller), utilizzato per mitigare le vibrazioni in scenari a bassa velocità (0.7 m/s). In [10], un nuovo controllore NMPC (Nonlinear Model Predictive Control) è stato impiegato in tre diversi scenari (singolo gradino, dosso e profilo stradale) a diverse velocità del veicolo, mostrando che una calibrazione bilanciata del controllore ha ridotto le vibrazioni longitudinali fino al 50%. Un limite di questi tipi di controllori tuttavia è l'elevato sforzo computazionale richiesto, che li rende difficili da implementare in tempo reale [11]. Inoltre tali controllori richiedono modelli interni estremamente

1. INTRODUZIONE

accurati, affrontando spesso decisioni di trade-off tra costo computazionale e precisione richiesta. La descrizione della dinamica del veicolo spesso richiede modelli matematici che ne approssimano il comportamento, trascurando aspetti ritenuti poco interessanti allo scopo. Tali dinamiche non modellate (o per esempio scenari non considerati come diversi profili stradali) compromettono la robustezza del controllore, obbligando quindi a ritornare alla fase di design dello stesso [12]. L'introduzione del Reinforcement Learning (RL) ha aperto nuove prospettive per lo sviluppo di controllori nel campo della dinamica veicolare [13]. In [14], un controllore ABS basato su RL è stato confrontato con un algoritmo Bosch HVE, mentre in [15] è stato sviluppato un controllore per veicoli autonomi, addestrato per scenari di cambio corsia e inseguimento di veicoli, considerando la distribuzione del jerk come fattore di comfort. In [16], un agente DDPG (Deep Deterministic Policy Gradient) è stato utilizzato per un controllo delle sospensioni attive, addestrato utilizzando profili stradali reali ottenuti sperimentalmente. In [17], è stato utilizzato un approccio basato sul Reinforcement Learning (RL) per ridurre il jerk nei veicoli elettrici (EV) causato dalla coppia motrice trasmessa alle ruote, dimostrando la stabilità del sistema su un'ampia gamma di veicoli. Questo lavoro evidenzia la capacità dell'approccio RL di gestire situazioni dinamiche complesse, come il jerk, con un livello di precisione e controllo che migliora significativamente il comfort di guida. In [18], gli autori hanno proposto un controllore di torque vectoring utilizzando un agente TD3-DDPG. Wei *et al.* hanno ottenuto ottime prestazioni per manovre di cambio corsia (DLC) e sterzata improvvisa su superfici a bassa aderenza ($\mu = 0.3 - 0.4$), attestando la robustezza e la compatibilità di questo approccio per applicazioni in tempo reale, superando le prestazioni dei controllori basati su modelli predittivi (MPC). Questi tipi di controllori sono anche adatti a problemi di path-tracking, come dimostrato da Wasala *et al.* in [19]. In questo articolo, un agente DDPG è stato addestrato su un singolo tracciato e poi testato su diversi circuiti, a velocità variabili e utilizzando due tipi di veicoli, per valutare la robustezza del metodo. Inoltre, è stata mostrata una prima implementazione su un veicolo reale. Un argomento simile è stato presentato in [20], in cui è stato implementato un problema di drifting autonomo lungo diversi tracciati utilizzando un agente SAC (Soft Actor-Critic), testato su quattro diversi veicoli. Infine, in [21], un agente DDPG è stato implementato per il controllo della velocità di un veicolo autonomo su strade sconnesse, con l'obiettivo di ridurre l'accelerazione verticale. L'output del controllore è stato valutato anche considerando il jerk longitudinale, per garantire il comfort del conducente. Guardando allo stato dell'arte dei controllori, si può osservare una lacuna in letteratura riguardante l'uso del RL per l'attenuazione delle vibrazioni longitudinali mediante la correzione della coppia. Tra i pochi esempi presenti è opportuno citare il lavoro di Hou [22], che presenta un controllore finalizzato alla riduzione delle oscillazioni nella fase post-frenata attraverso la decompressione del sistema frenante. In questo

contesto, il presente elaborato introdurrà un nuovo controllore basato sul RL, che mira a superare le prestazioni dei noti NMPC nell'attenuazione delle oscillazioni longitudinali, garantendo al contempo un minore sforzo computazionale, rendendolo idoneo per applicazioni Real time su veicoli stradali. I seguenti capitoli sono strutturati come segue:

- **Capitolo 2:** Viene presentato l'ambiente di simulazione utilizzato per l'implementazione del controllore proposto, con una descrizione dettagliata del modello di veicolo adottato.
- **Capitolo 3:** Vengono descritti i due controllori principali utilizzati: il Nonlinear Model Predictive Control (NMPC) e il Reinforcement Learning (RL). Il capitolo introduce il NMPC come riferimento di partenza, per poi presentare il nuovo approccio RL, evidenziando le differenze principali.
- **Capitolo 4:** Dettaglia la fase di addestramento dell'agente RL, illustrando i vari scenari utilizzati e come la strategia di controllo si evolve in relazione alla complessità e varietà degli scenari affrontati durante l'addestramento.
- **Capitolo 5:** Discute i risultati ottenuti con il controllore RL ottimizzato su una serie di scenari, confrontando le prestazioni con il controllore NMPC e con un sistema passivo, senza azione di controllo attiva.
- **Capitolo 6:** Descrive l'implementazione del controllore su una centralina reale (MicroAutoBox III), verificando i tempi di esecuzione computazionale in un contesto applicativo reale.
- **Capitolo 7:** Presenta le conclusioni generali del lavoro, con considerazioni sui possibili sviluppi futuri del controllore.

2 SIMULATION ENVIRONMENT

L'ambiente di simulazione utilizzato per lo sviluppo del controllore proposto è simile a quello sviluppato precedentemente in [10] il quale pone le basi per la presente attività. Il modello è stato suddiviso in una serie di blocchi funzionali, descritti di seguito e rappresentati nello schema in Figura 2.1:

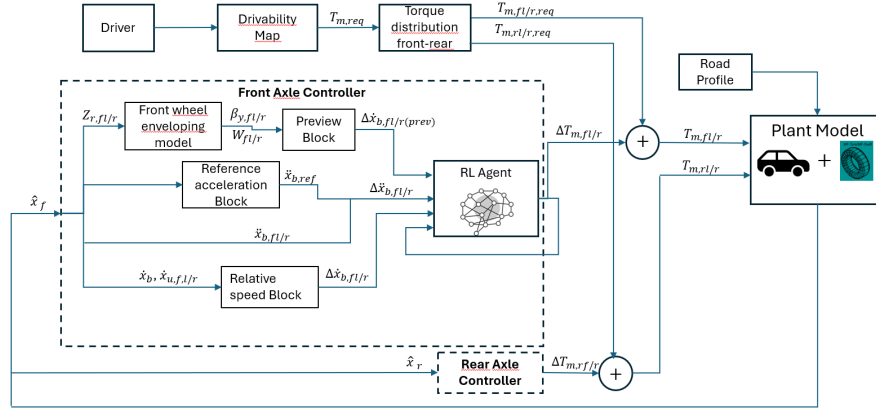


Figure 2.1: Schematizzazione dell'ambiente di simulazione utilizzato: si possono evidenziare i singoli blocchi di controllo, il modello di veicolo (o plant) e l'input di coppia richiesto dal pilota

1. Modello del conducente, che nello scenario analizzato di dinamica puramente longitudinale e profilo stradale simmetrico, definisce la coppia desiderata attraverso la posizione del pedale dell'acceleratore *APP*.
2. Mappa di guida, che fornisce il valore corrispondente della coppia totale richiesta $T_{m,req}$ in base alla posizione del pedale e alla velocità attuale del veicolo.
3. Distribuzione della coppia anteriore-posteriore, che definisce la coppia corrispondente per ogni motore in base a strategie di gestione energetica. Essendo lo studio di un caso di moto puramente longitudinale, la distribuzione della coppia viene considerata uguale per entrambi i lati di ogni asse.
4. Controllori di coppia per l'asse anteriore e posteriore, che costituiscono il nucleo di questo studio. Prendono come input lo stato attuale del veicolo e forniscono come output la correzione della coppia corrispondente $\Delta T_{m,f/r}$ che verrà aggiunta alla coppia richiesta su ciascun asse $T_{m,f/r,req}$.

2. SIMULATION ENVIRONMENT

5. Modello del veicolo, utilizzato per testare le azioni dei due controllori (descritto in dettaglio nel paragrafo successivo).

2.1 Vehicle Model

Il modello di veicolo utilizzato (implementato interamente in Matlab-Simulink) è un EV SUV utilizzato nel progetto europeo HIGHSCAPE, con i parametri principali elencati nella Tabella 2.1.

Parametro	Simbolo	Valore
Massa totale del veicolo	m_{tot}	2853 kg
Massa dei motori in-wheel	$m_{u,f/r}$	65 kg
Massa sospesa	m_b	2593 kg
Momento d'inerzia del corpo vettura	$I_{b,y}$	2200 kg*m ²
Semipasso anteriore	a	1.472 m
Semipasso posteriore	b	1.455 m
Altezza del baricentro	h_{cg}	0.631 m
Raggio ruota	$R_{f/r}$	0.372 m
Coefficiente di resistenza aerodinamica	C_d	0.28
Area frontale	A_{car}	2.65 m ²

Table 2.1: Parametri principali del veicolo

Tale modello considera sia la dinamica verticale che quella longitudinale dell'auto, tenendo conto della non linearità delle sospensioni e degli effetti dei silent block che separano il mozzo ruota dal corpo principale (modellato come un sistema molla-smorzatore), nonché degli effetti dello pneumatico, utilizzando il modello MF-Swift. Dai sottosistemi relativi al sistema sospensivo e degli pneumatici si ottengono le rispettive forze, che andranno poi ad influenzare la dinamica delle masse sospese/non-sospese attraverso le seguenti equazioni:

$$\ddot{x}_b = \frac{-F_{c,x,f} - F_{k,x,f} - F_{c,x,r} - F_{k,x,r} - F_{drag}}{m_b} \quad (2.1)$$

$$\ddot{x}_{u,f/r} = \frac{F_{c,x,f/r} + F_{k,x,f/r} - F_{x,f/r}}{m_{u,f/r}} \quad (2.2)$$

$$\ddot{z}_b = \frac{-F_{c,z,f} - F_{k,z,f} - F_{c,z,r} - F_{k,z,r}}{m_b} \quad (2.3)$$

$$\ddot{z}_{u,f/r} = \frac{F_{c,z,f/r} + F_{k,z,f/r} + F_{z,f/r}}{m_{u,f/r}} \quad (2.4)$$

$$\ddot{\theta} = \frac{(F_{c,z,r} + F_{k,z,r}) * b - (F_{c,z,f} + F_{k,z,f}) * a - (F_{c,x,f} + F_{k,x,f} + F_{c,x,r} + F_{k,x,r}) * h_{cg}}{I_{b,y}} \quad (2.5)$$

Dove il pedice u indica la massa non sospesa, mentre il pedice b fa riferimento alla massa sospesa del veicolo. $F_{c,x,f/r}$, $F_{k,x,f/r}$ sono i termini di smorzamento e rigidità dei silent blocks in direzione longitudinale, $F_{c,z,f/r}$, $F_{k,z,f/r}$ i termini di smorzamento e rigidità della sospensione in direzione verticale, $F_{x,f/r}$, $F_{z,f/r}$ le forze dello pneumatico in direzione orizzontale e verticale, ricavate dal modello MF-Swift mentre F_{drag} forza di drag data dalla resistenza aerodinamica del mezzo. E' da notare che tali termini si riferiscono solamente all'asse di appartenenza (anteriore o posteriore) ma va considerata la suddivisione in lato sinistro e destro del veicolo.

Il modello considera anche la dinamica dei motori elettrici, nella gestione della coppia richiesta, attraverso una funzione di trasferimento:

$$H(s) = \frac{1}{0.0056995s + 1} \quad (2.6)$$

Di seguito viene mostrato il modello simulink utilizzato per descrivere la dinamica dell'intero veicolo:

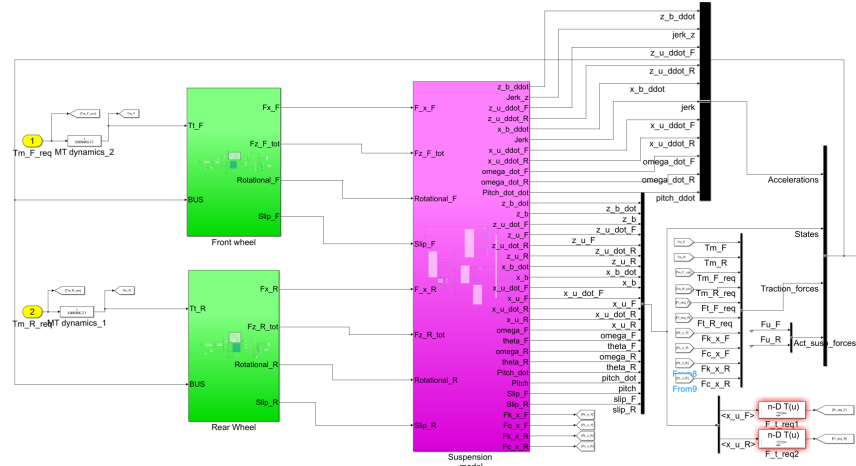


Figure 2.2: Modello del veicolo utilizzato: si evidenziano i blocchi MF-Swift per le ruote anteriori e posteriori (in verde), e blocco utilizzato per computare le forze sospensive (in viola), ricavando da quest'ultime le rispettive accelerazioni, velocità e posizioni

2.2 MF-Swift

Per una modellazione accurata del contatto pneumatico-strada su diversi profili stradali, il modello del veicolo è integrato con il modello di pneumatico MF-Swift: un modello semi-empirico validato sperimentalmente[23].

2. SIMULATION ENVIRONMENT

Esso risulta adatto ad applicazioni veicolistiche inerenti al comfort di guida grazie alla sua ottima approssimazione della dinamica dello pneumatico in-plane (direzione longitudinale e verticale) ed out-of-plane (laterale) in casi di puro slip o slip combinato, con fenomeni di eccitazione ad alta frequenza. Il modello di inviluppo utilizzato è particolarmente adatto nel trattare ostacoli o irregolarità stradali, avvicinandosi ai dati sperimentali, come mostrato in Figura 2.3.

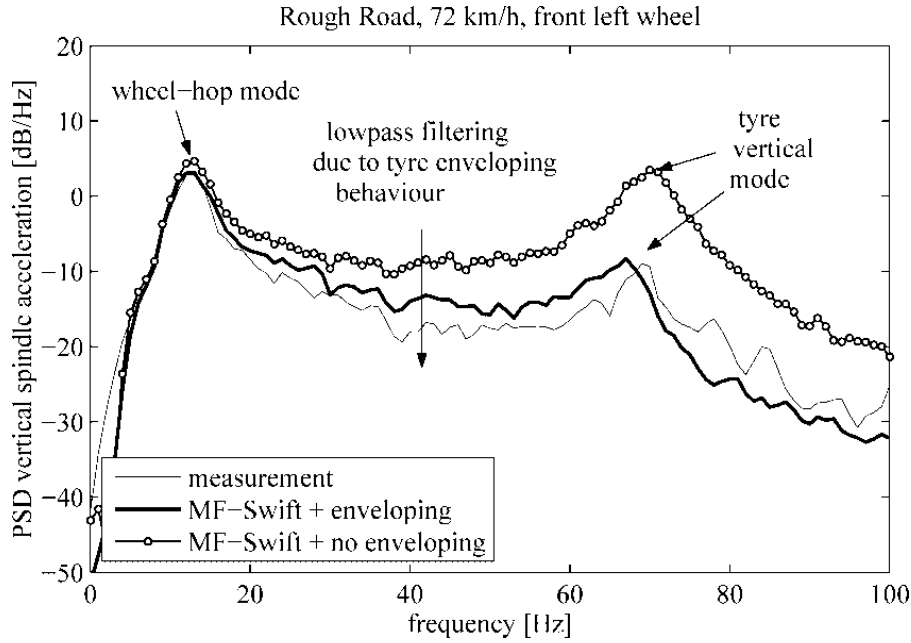


Figure 2.3: PSD dell'accelerazione verticale del mozzo ruota, su una strada ISO D, si nota come il modello di inviluppo influenzi l'accuratezza dei risultati. Tratto da [23]

MF-Swift è composto da quattro elementi principali:

- **Magic Formula**, per calcolare con alta precisione le forze di slittamento e i momenti in regime stazionario. Tale modello, ampiamente usato permette di mantenere un errore rispetto alle misure sperimentali al di sotto del 5 % per misure di puro slip longitudinale/laterale e al di sotto del 8 % nel caso di slip combinati [24].
- **Modello di slip del punto di contatto**, che tiene conto del tempo di risposta del pneumatico, dipendente dal carico e dall'angolo di sideslip. Tipicamente viene modellato come un'equazione differenziale del primo ordine, introducendo la lunghezza di rilassamento:

$$\sigma_c \dot{\alpha}' + V_x \alpha' = -V_{syc} - V_x \left(\frac{M_z}{c_\psi} \right) \quad (2.7)$$

Dove:

- σ_c è la lunghezza di rilassamento del contatto,
 - V_{syc} è la velocità di slittamento del contatto,
 - M_z è il momento autoallineamento,
 - c_ψ è la rigidezza torsionale della carcassa.
- **Modello ad anello rigido**, utilizzato per coprire una gamma di frequenza di circa 60-100 Hz, nel quale la deformazione della carcassa può essere trascurata e modellata come un corpo rigido, sospesa elasticamente rispetto al cerchione;
 - **Modello di inviluppo**, che calcola il profilo stradale effettivo per ostacoli tridimensionali arbitrari utilizzando file di camme ellittiche. Permettono di ottenere il profilo effettivo della strada calcolando:
 - la posizione verticale tramite una media delle coordinate z lungo tutta la larghezza dello pneumatico, per entrambe le file di camme;
 - l'inclinazione del profilo stradale calcolando l'angolo medio lungo la larghezza dello pneumatico, per la fila anteriore e posteriore di camme;
 - la campanatura della strada tramite una media lungo la lunghezza dello pneumatico, dell'angolo trasversale.

Uno schema del modello Swift è presentato in Figura 2.4

2. SIMULATION ENVIRONMENT

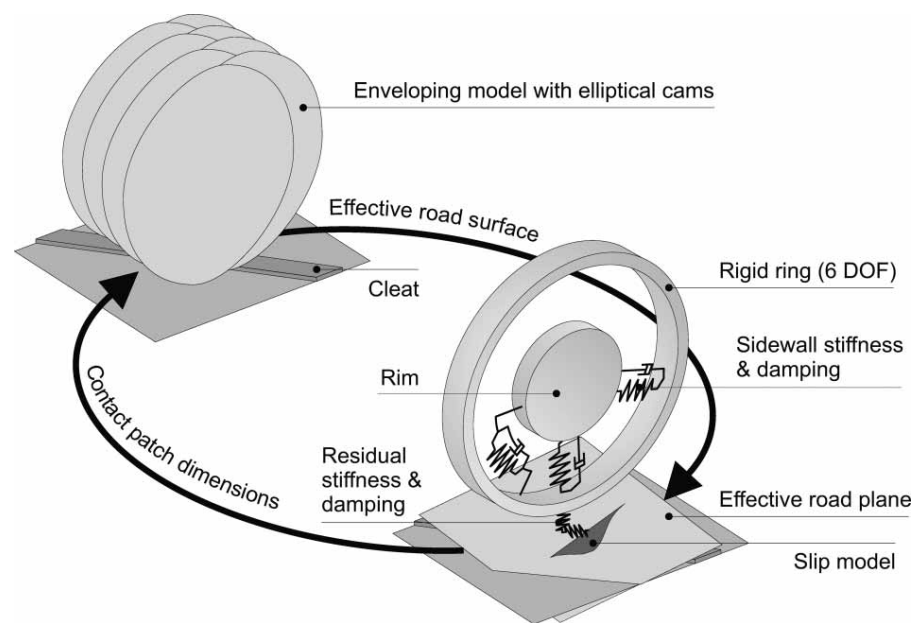


Figure 2.4: Schematizzazione del modello MF-Swift, con il modello di inviluppo. Tratto da [23]

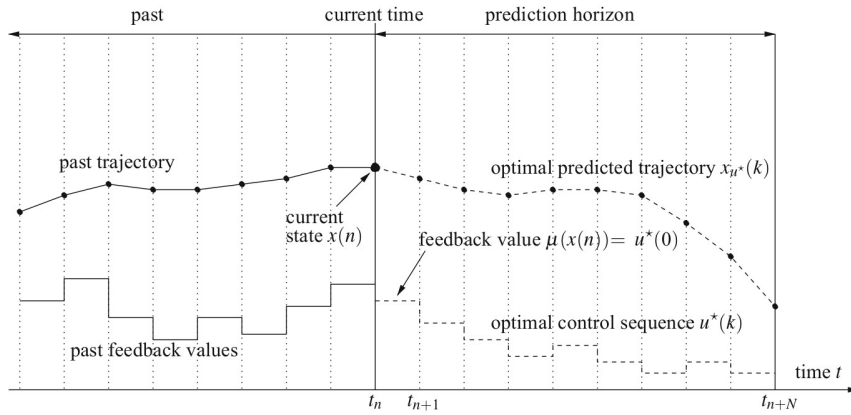
3 SISTEMI DI CONTROLLO

3.1 Non Linear Model Predictive Control

Il Non Linear Model Predictive Control (NMPC) è una tecnica di controllo avanzata che prevede il comportamento futuro di un sistema dinamico non lineare e ottimizza l'azione di controllo su un certo orizzonte temporale. NMPC lavora prevedendo l'evoluzione degli stati di un sistema e risolvendo un problema di ottimizzazione online, soddisfacendo al contempo delle equazioni di vincolo (possono essere limiti fisici o meno, un esempio è la coppia motrice nel caso in esame). Si distingue dal Linear MPC per il modello interno utilizzato per ricavare la previsione degli stati futuri, il quale presenta non linearità dovute al comportamento dello pneumatico (slip e forze applicate) e del sistema sospensivo, nonché vincoli presentati dalla coppia massima applicabile dai motori. Si consideri un sistema nel quale si vuole misurare ad un intervallo discreto di tempo $t_n, n = 1, 2 \dots N$ lo stato $x(n)$ e controllarlo tramite un'azione del controllore $u(n)$. Il controllore, per ogni istante temporale, fornisce un'azione di controllo, aggiornando lo stato secondo la legge:

$$x^+ = f(x, u) \quad (3.1)$$

Si viene così a creare una sequenza di azioni di controllo $u(0), u(1), \dots, u(N-1)$ con N lunghezza dell'orizzonte di previsione. Di conseguenza si ricava una traiettoria ottimale dello stato x . L'idea alla base del metodo prevede di utilizzare come output di controllo solamente il primo termine della citata sequenza di azioni u , aggiornando lo stato all'istante temporale successivo e ripetendo la procedura (si parla quindi di recessione dell'orizzonte di controllo). Una schematizzazione è fornita nella Figura 3.1. L'azione di controllo ottimale viene scelta basandosi su una *funzione costo* da minimizzare, solitamente espressa come funzione quadratica. Riassumendo: ad ogni istante temporale n viene calcolata una sequenza di azioni ottimali basandosi sulla predizione del comportamento del sistema su un orizzonte temporale finito $k = 0, \dots, N-1$ utilizzando poi il primo elemento di questa sequenza e passando all'intervallo successivo. Importante specificare la differenza di terminologia utilizzata per indicare l'azione del controllore nel prevedere il comportamento futuro del sistema (*predizione*) ed invece i set di dati derivanti da sistemi esterni che vengono passati al controllore per avere un riferimento reale del profilo stradale dinanzi al veicolo (*previsione*). Nel caso in esame, tale architettura è stata utilizzata come punto di partenza per le successive implementazioni tramite algoritmi di Reinforcement Learning, fornendo un confronto diretto tra le due tipologie. Nello specifico il controllore utilizzato è presentato nel lavoro di Vidal *et al.* [10], applicato ad ogni gruppo ruota del veicolo e spiegato in sintesi


 Figure 3.1: Illustrazione del NMPC al tempo t_n , tratta da [25]

nelle sezioni seguenti: suddivise in: modello di involucro dello pneumatico, modello interno di previsione, problema di controllo ottimo non-lineare.

3.1.1 Modello di involucro

Il modello di involucro utilizzato dal controllore per stimare il profilo stradale risulta simile a quello implementato nel modello MF-Swift del veicolo, tuttavia semplificato per poter garantire contemporaneamente una buona accuratezza ed un costo computazionale contenuto. In particolare il modello proposto è stato presentato da *Schemitz* in [26]. Il profilo stradale viene definito tramite due grandezze: w e β_y , ricavate tramite una coppia di camme ellittiche per ciascuna ruota del veicolo. Questo set di camme si muove solidalmente alla ruota in direzione longitudinale ad una distanza predefinita l_s , avendo contemporaneamente un movimento verticale indipendente. Le due grandezze w e β_y , funzione della coordinata longitudinale x_u (esprese rispetto il punto P_0 , allineato con il centro ruota) sono ricavate tramite le seguenti formule:

$$w(x_u) = \frac{Z_{e,fc} + Z_{e,rc}}{2} - b_c$$

$$\tan \beta_y(x_u) = \frac{Z_{e,fc} - Z_{e,rc}}{l_s}$$

Dove $Z_{e,fc}$ e $Z_{e,rc}$ sono le coordinate verticali rispettivamente della camma posteriore ed anteriore. La geometria delle camme ellittiche è data dalla seguente equazione:

$$\left(\frac{x_{e,fc}}{a_c} \right)^c + \left(\frac{z_{e,fc}}{b_c} \right)^c = 1$$

dove:

- a_c è il semiasse orizzontale,

3.1. NON LINEAR MODEL PREDICTIVE CONTROL

- b_c è il semiasse verticale,
- c parametro di forma dell'ellisse, che ne controlla la curvatura,
- $x_{e,fc}, z_{e,fc}$ è il sistema di assi relativo alla camma anteriore.

Data quindi una coordinata longitudinale x_u , il modello calcola $Z_{e,fc}$, $Z_{e,rc}$ come il massimo della somma tra l'elevazione della strada z_r e le distanze ricavate per tutto il range di posizioni possibili della camma in questione d_{fc}, d_{rc} .

$$Z_{e,fc} = \max_{x_{e,fc} \in [-a_c, a_c]} (z_r(x_u, x_{e,fc}) + d_{fc}(x_{e,fc})) \quad (3.2)$$

$$\begin{aligned} Z_{e,rc} = \max & (z_r(x_u, x_{e,rc}) + d_{rc}(x_{e,rc}) \\ & x_{e,rc} \in [-a_c, a_c] \end{aligned} \quad (3.3)$$

Infine il profilo stradale viene introdotto nel NMPC come un input esterno.

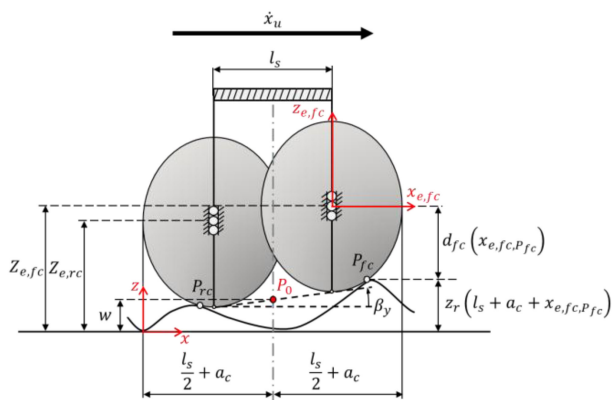


Figure 3.2: Schema del modello di involuppo a due camme ellittiche, tratta da [10]

Il modello interno tuttavia richiede informazioni sull'altezza e sulla pendenza della strada lungo tutto il tempo di previsione $t_p = N_p t_s$, che è lo stesso o più breve dell'orizzonte di predizione $t_h = N_h t_s$, dove N_p è il numero di punti di previsione, N_h numero di punti di predizione e t_s è l'intervallo di tempo. Poiché in questa specifica applicazione t_h è breve, la velocità del veicolo può essere considerata costante lungo t_h , e la posizione della ruota è data da:

$$x_{u,k} = \dot{x}_{u,0} t_{s,k} \quad (3.4)$$

dove l'indice k indica il passo temporale. Se $t_h > t_p$, il profilo stradale stimato per t_h viene mantenuto costante per $t > t_p$. Nel caso in esame il controllore, a causa delle esigenze computazionali per l'implementazione in

real time, utilizza un orizzonte di previsione e predizione uguale e pari a $t_h = N_h * t_s = 2 * 15 = 30ms$, avendo quindi due step di azioni di controllo, lungo un orizzonte di $30ms$.

3.1.2 Modello di Previsione

Il modello di previsione, noto anche come modello interno, viene integrato nella struttura del NMPC ed è riferito a ciascun gruppo ruota del veicolo(corner). Ogni modello di previsione include:

- le dinamiche del pneumatico e della ruota;
- le dinamiche delle masse non sospese longitudinali e verticali;
- le dinamiche verticali della massa sospesa (tutte relative solo al gruppo ruota considerato);
- le dinamiche longitudinali della massa sospesa dell'intero veicolo.

Di conseguenza, dal punto di vista meccanico delle dinamiche delle inerzie coinvolte, il sistema è caratterizzato da cinque gradi di libertà, ai quali si aggiungono ulteriori equazioni differenziali, ad esempio, per descrivere le dinamiche delle macchine elettriche e il comportamento di rilassamento dei pneumatici. Il modello di previsione trascura le dinamiche longitudinali e verticali degli altri corner, tuttavia la loro azione di coppia viene inserita all'interno delle equazioni come input esterno al fine di rispettare l'accelerazione totale richiesta alla massa sospesa. Le dinamiche ad alta frequenza dovute al profilo stradale sono invece mantenute in riferimento al singolo gruppo ruota (IWM). L'analisi delle dinamiche di un singolo corner all'interno del modello di previsione risulta vantaggioso in termini di efficienza computazionale, data la presenza di un numero di stati ed azioni di controllo ridotto, comparato con un controllore centralizzato per tutto il veicolo, migliorandone le performance per l'implementazione in real time. La massa non sospesa include una parte centrale non rotante, corrispondente ai componenti delle sospensioni, alla pinza freno e allo statore del motore, e una parte rotante esterna, corrispondente al cerchione della ruota, al disco freno e al rotore del motore. La rigididezza verticale e gli elementi di smorzamento non lineari tra le masse sospese e non sospese emulano la molla della sospensione e l'ammortizzatore, mentre il sistema molla-smorzatore longitudinale, con rigidità $k_{x,f}$ e coefficiente di smorzamento $c_{x,f}$, riproducono il comportamento delle boccole della sospensione per il gruppo ruota considerato. Le dinamiche dello pneumatico sono modellate attraverso due coppie di sistemi molla-smorzatore, la prima radiale, con rigidità $k_{r,f}$ e coefficiente di smorzamento $c_{r,f}$, e una seconda tangenziale, con rigidità $k_{t,f}$ e coefficiente di smorzamento $c_{t,f}$. Gli elementi della struttura del pneumatico, che trasmettono le loro forze direttamente alla parte non rotante della massa non sospesa, possono essere considerati

in contatto con la strada tramite un sistema di rulli privo di attrito. Infine la forza longitudinale dovuta allo slip dello pneumatico viene considerata applicata indipendentemente sulla massa rotante e non rotante. Le dinamiche

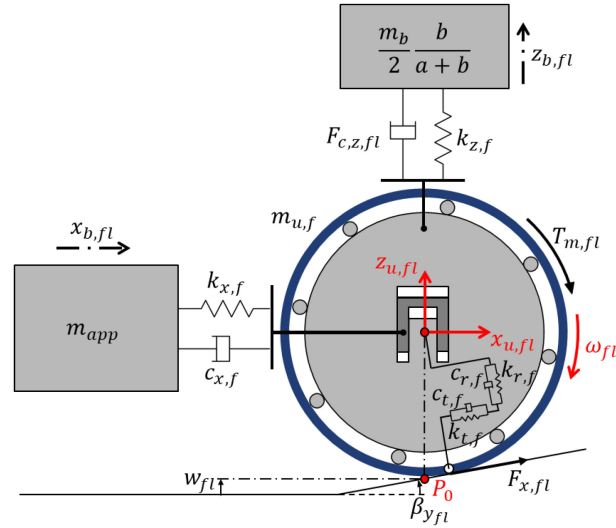


Figure 3.3: Rappresentazione del modello interno utilizzato dal NMPC riferito al motore anteriore sinistro, tratto da [10]

del modello interno sono descritte dalle seguenti equazioni, riportate per l'angolo anteriore sinistro (lo stesso approccio è utilizzato per gli altri angoli del veicolo) e fanno riferimento alla Figura 3.3. L'equazione che esprime la dinamica del motore elettrico, nell'erogazione della coppia è data da:

$$\dot{T}_{m,fl} = \frac{[T_{m,rq,cr,fl} - T_{m,fl}]}{\tau} \quad (3.5)$$

dove $T_{m,fl}$ è la coppia elettromagnetica effettiva del motore e τ è la rispettiva costante di tempo.

L'equazione di equilibrio verticale della massa sospesa è espressa da:

$$\ddot{z}_{b,fl} = [-F_{k,z,fl} - F_{c,z,fl}] \frac{2}{m_b} \frac{a+b}{b} \quad (3.6)$$

dove m_b è la massa totale sospesa, e $F_{k,z,fl}$ e $F_{c,z,fl}$ sono le forze verticali della sospensione. $F_{k,z,fl}$ è data da:

$$F_{k,z,fl} = k_{z,f}[z_{b,fl} - z_{u,fl}] \quad (3.7)$$

dove $k_{z,f}$ è la rigidità verticale della sospensione, e $z_{u,fl}$ è lo spostamento verticale della massa non sospesa. Nell'implementazione specifica, $F_{c,z}$ è calcolata con una funzione non lineare continua che approssima il comportamento

3. SISTEMI DI CONTROLLO

passivo dell'ammortizzatore:

$$F_{c,z,fl} = a_{1,f} + b_{1,f} \tan^{-1}(c_{1,f}[\dot{z}_{b,fl} - \dot{z}_{u,fl}] + d_{1,f}) + b_{2,f} \tan^{-1}(c_{2,f}[\dot{z}_{b,fl} - \dot{z}_{u,fl}] + d_{2,f}) \quad (3.8)$$

dove $a_{1,f}$, $b_{1/2,f}$, $c_{1/2,f}$, e $d_{1/2,f}$ sono coefficienti costanti.

Passando ora all'equilibrio verticale della massa non sospesa, essa è espressa nella forma:

$$\ddot{z}_{u,fl} = \frac{[F_{c,z,fl} + F_{k,z,fl} + F_{r,z,fl} - F_{t,z,fl} + F_{x,z,fl}]}{m_{u,f}} \quad (3.9)$$

dove $m_{u,f}$ è la massa non sospesa. $F_{r,z,fl}$ e $F_{t,z,fl}$ sono le componenti verticali delle forze radiali e tangenziali, $F_{r,fl}$ e $F_{t,fl}$, associate alla struttura del pneumatico.

$$F_{r,fl} = k_{r,f}[w_{fl} - z_{u,fl}] \cos \beta_{y,fl} + c_{r,f}[\dot{w}_{fl} - \dot{z}_{u,fl}] \cos \beta_{y,fl} \quad (3.10)$$

$$F_{t,fl} = k_{t,f}[w_{fl} - z_{u,fl}] \sin \beta_{y,fl} + c_{t,f}[\dot{w}_{fl} - \dot{z}_{u,fl}] \sin \beta_{y,fl} \quad (3.11)$$

sotto l'assunzione di trascurare la derivata temporale di $\beta_{y,fl}$. $F_{x,z,fl}$ è la componente verticale della forza di trazione/frenata $F_{x,fl}$, calcolata tramite la Magic Formula (versione 5.2) per condizioni di angolo di slittamento nullo. L'equazione del bilancio delle forze longitudinali della massa sospesa è data da:

$$\ddot{x}_{b,fl} = [-F_{k,x,fl} - F_{c,x,fl} + \frac{T_{m,rq,fr}}{R_{lad,f}} + \frac{T_{m,rq,rl}}{R_{lad,r}} + \frac{T_{m,rq,rr}}{R_{lad,r}} - F_{drag} - \frac{F_{roll,f}}{2} - F_{roll,r}] \frac{1}{m_{app}} \quad (3.12)$$

dove:

- $R_{lad,f/r}$ è il raggio delle ruote a carico,
- $T_{m,rq,fr}$, $T_{m,rq,rl}$, e $T_{m,rq,rr}$ sono le coppie del motore per le rispettive ruote,
- F_{drag} è la forza di resistenza aerodinamica,
- $F_{roll,f}$ e $F_{roll,r}$ sono le forze di resistenza al rotolamento sugli assi anteriore e posteriore,
- m_{app} è la massa apparente, considerando le inerzie delle ruote e dei componenti associati (ad esempio la pinza freno).

3.1. NON LINEAR MODEL PREDICTIVE CONTROL

L'accelerazione longitudinale si riferisce all'intera massa sospesa, ma il pedice $'fl'$ indica che essa è espressa dal punto di vista dell'angolo specifico, il quale è l'unico che trasmette la forza di trazione tramite il sistema molla-smorzatore longitudinale (infatti le restanti ruote trasmettono la propria forza direttamente alla massa sospesa).

Le forze longitudinali della sospensione sono descritte da:

$$F_{k,x,fl} = k_{x,f}(x_{b,fl} - x_{u,fl}) \quad (3.13)$$

$$F_{c,x,fl} = c_{x,f}(\dot{x}_{b,fl} - \dot{x}_{u,fl}) \quad (3.14)$$

La forza di resistenza aerodinamica è descritta da:

$$F_{drag} = \frac{1}{2}\rho_{air}C_dA_{car}\dot{x}_b^2 \quad (3.15)$$

dove: ρ_{air} è la densità dell'aria, C_d è il coefficiente di resistenza aerodinamica, A_{car} è l'area frontale del veicolo.

Le forze di resistenza al rotolamento per gli assi anteriore e posteriore sono espresse come:

$$F_{roll,f} = f_{roll}m_{tot}\frac{b}{a+b}g \quad (3.16)$$

$$F_{roll,r} = f_{roll}m_{tot}\frac{a}{a+b}g \quad (3.17)$$

dove f_{roll} è il coefficiente di resistenza al rotolamento:

$$f_{roll} = f_0 + f_2\dot{x}_b^2 \quad (3.18)$$

con f_0 e f_2 come coefficienti costanti.

Il bilancio delle forze longitudinali della massa non sospesa è dato da:

$$\ddot{x}_{u,fl} = \frac{F_{c,x,fl} + F_{k,x,fl} - F_{r,x,fl} - F_{t,x,fl} + F_{x,x,fl}}{m_{u,f}} \quad (3.19)$$

dove $F_{r,x,fl}$ e $F_{t,x,fl}$ sono le componenti longitudinali delle forze radiali e tangenziali $F_{r,fl}$ e $F_{t,fl}$.

L'equazione di bilancio del momento della ruota è:

$$\dot{\omega}_{fl} = \frac{T_{m,fl} - F_{x,fl}R_{lad,f} - M_{y,fl}}{I_{y,f}} \quad (3.20)$$

dove $\dot{\omega}_{fl}$ è l'accelerazione angolare della ruota e $M_{y,fl}$ è il momento di resistenza al rotolamento.

Le dinamiche di rilassamento longitudinale del pneumatico sono descritte da:

$$\dot{\sigma}_{d,fl} = [\sigma_{th,fl} - \sigma_{d,fl}]\frac{\dot{x}_{u,fl}}{L_{rel,f}} \quad (3.21)$$

3. SISTEMI DI CONTROLLO

dove $L_{rel,f}$ è la lunghezza di rilassamento e $\sigma_{th,fl}$ è il rapporto di slittamento teorico, calcolato come:

$$\sigma_{th,fl} = 1 - \frac{\dot{x}_{u,fl}}{\omega_{fl} R_f} \quad (3.22)$$

Il modello interno appena descritto viene riorganizzato come un sistema di equazioni differenziali del primo ordine:

$$\dot{x}_{fl}(t) = f(x_{fl}(t), u_{fl}(t), p_{fl}(t)) \quad (3.23)$$

dove:

- f indica una funzione non lineare;
- x_{fl} è il vettore degli stati;
- $u_{fl} = \Delta T_{m,fl}$ è l'azione di controllo, cioè la correzione della coppia del motore imposta dal controllore;
- p_{fl} è la matrice dei parametri.

Il vettore degli stati x_{fl} è definito come:

$$x_{fl} = [\dot{z}_{b,fl}, z_{b,fl}, \dot{z}_{u,fl}, z_{u,fl}, \dot{x}_{b,fl}, x_{b,fl}, \dot{x}_{u,fl}, x_{u,fl}, \omega_{fl}, \sigma_{d,fl}, T_{m,fl}]^T \quad (3.24)$$

La matrice dei parametri p_{fl} include il profilo stradale effettivo previsto e le richieste di coppia non corrette:

$$p_{fl} = [w_{fl}, \beta_{y,fl}, T_{m,rq,fl}, T_{m,rq,fr}, T_{m,rq,rl}, T_{m,rq,rr}]^T \quad (3.25)$$

3.1.3 Problema di controllo ottimo non lineare

Come descritto nell'introduzione al NMPC, il controllore necessita di una funzione costo al fine di determinare l'azione migliore basandosi sulla sua predizione. Data l'architettura del sistema, riferito al singolo gruppo ruota, la soluzione al problema di controllo ottimo è data dalla *minimizzazione di J*:

$$\min_U J = \frac{1}{2} \|\mathbf{z}_{N_h} - \mathbf{z}_{ref,N_h}\|_{Q_x}^2 + \frac{1}{2} \sum_{k=0}^{N_h-1} (\|\mathbf{z}_k - \mathbf{z}_{ref,k}\|_{Q_x}^2 + \|\mathbf{u}_k\|_R^2) \quad (3.26)$$

con:

$$\begin{aligned} \mathbf{x}_0 &= \mathbf{x}_{in} \\ \mathbf{x}_{k+1} &= f_d(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p}_k) \\ \mathbf{z}_k &= g_d(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p}_k) \\ \mathbf{u}^- &\leq \mathbf{u}_k \leq \mathbf{u}^+ \end{aligned}$$

dove $U = [u_0, u_1, \dots, u_{N_h}]^T$ rappresenta la sequenza di azioni di controllo lungo l'orizzonte di predizione. La funzione di costo è composta da due

3.1. NON LINEAR MODEL PREDICTIVE CONTROL

termini: il primo termine mira a minimizzare l'errore alla fine dell'orizzonte di predizione, mentre il secondo termine ottimizza l'azione di controllo durante tutto l'orizzonte. I vettori z e z_{ref} rappresentano, rispettivamente, gli output predetti del sistema ed il loro valore di riferimento (o target). Q_x e R sono le matrici diagonali dei pesi relativi ai singoli termini. Il vettore di stato iniziale è x_{in} , e le funzioni f_d e g_d rappresentano rispettivamente la dinamica del sistema discretizzata e l'uscita del sistema. L'ingresso di controllo u è limitato da u^- e u^+ (coppie massime applicabili dal motore, mostrate in seguito). Nella specifica implementazione, z include l'accelerazione longitudinale del corpo del veicolo e la velocità longitudinale relativa tra la massa non sospesa e quella sospesa dell'angolo considerato, che contribuisce alle oscillazioni longitudinali:

$$z_k = [\ddot{x}_{b,k}, \dot{x}_{b,k} - \dot{x}_{u,k}]^T \quad (3.27)$$

mentre il riferimento $z_{ref,k}$ è definito come:

$$z_{ref,k} = [\ddot{x}_{b,ref}, 0]^T \quad (3.28)$$

dove $\ddot{x}_{b,ref}$ è l'accelerazione di riferimento:

$$\ddot{x}_{b,ref} = \left(\frac{T_{m,rq,fl} + T_{m,rq,fr}}{R_{lad,f}} + \frac{T_{m,rq,rl} + T_{m,rq,rr}}{R_{lad,r}} - f_{roll}m_{tot}g - F_{drag} \right) \frac{1}{m_{tot} + \frac{2I_{y,r}}{R_r^2} + \frac{2I_{y,f}}{R_f^2}} \quad (3.29)$$

In questa equazione, i valori di coppia del motore divisi per il raggio della ruota rappresentano le forze di trazione, da cui vengono sottratte la forza di resistenza al rotolamento delle gomme e la resistenza aerodinamica. Questi valori sono poi divisi per la massa apparente totale del veicolo.

3.2 Reinforcement Learning

Prima di addentrarci nel Reinforcement Learning, è opportuno fornire una visione generale di cosa si intende con Machine Learning. Il Machine Learning (ML) è un'area dell'intelligenza artificiale dedicata allo studio di programmi informatici in grado di risolvere problemi che richiedono un'apprendimento (e quindi, di intelligenza) tramite dei dati. Vi sono 3 macro aree del machine learning:

- **Supervised Learning (SL)**, il quale ha come obbiettivo l'imparare da dati etichettati. In questo approccio, un operatore umano decide quali dati raccogliere e come etichettarli. L'obiettivo dell'SL è riuscire a generalizzare. Un esempio classico è il riconoscimento di cifre scritte a mano: un essere umano raccoglie immagini contenenti cifre scritte a mano, le etichetta e addestra un modello per riconoscere e classificare correttamente le cifre presenti nelle immagini. Il modello addestrato dovrebbe quindi essere in grado di generalizzare, ossia classificare correttamente le cifre in nuove immagini non viste in precedenza.
- **Unsupervised Learning (UL)**, il quale si basa invece su dati non etichettati. Sebbene non sia più necessario etichettare i dati, le metodologie utilizzate per raccogliere e organizzare i dati devono comunque essere progettate da un umano. L'obiettivo principale dell'UL è comprimere l'informazione, riducendo la complessità e scoprendo modelli sottostanti. Un esempio classico è la segmentazione della clientela: un essere umano raccoglie dati sui clienti e addestra un modello per raggruppare i clienti in cluster. Questi gruppi rappresentano una forma di compressione dell'informazione, evidenziando relazioni nascoste nei dati.
- **Reinforcement Learning (RL)**, il quale è basato su un processo di apprendimento tramite tentativi ed errori. In questo approccio, non vi è bisogno né di etichettare i dati né di un intervento umano nella loro raccolta. L'obiettivo principale dell'RL è imparare a compiere azioni efficaci. Un esempio tipico è un agente che gioca a Pong: l'agente interagisce ripetutamente con un simulatore di Pong, apprendendo quali azioni compiere in base ai risultati ottenuti. Alla fine dell'addestramento, l'agente dovrebbe essere in grado di agire in modo da vincere le partite.

Il *Reinforcement Learning (RL)* è una forma di apprendimento automatico in cui un agente impara a prendere decisioni interagendo con un ambiente. L'obiettivo dell'agente è massimizzare la ricompensa cumulativa nel tempo attraverso una sequenza di azioni. Con ambiente si intende tutto ciò con cui l'agente interagisce (ad esempio il modello di veicolo nel nostro caso) mentre con agente si intende la parte di codice incaricata di prendere le decisioni (comunemente indicata come controllore).

Nel contesto del RL, l'agente interagisce con l'ambiente in istanti temporali discreti. Ad ogni passo temporale, l'agente osserva lo stato corrente s_t , seleziona un'azione a_t , riceve una ricompensa r_t , e transita in un nuovo stato s_{t+1} . Le ricompense future vengono "scontate" di un fattore γ , diminuendone di fatto il peso nella reward totale. Il processo decisionale attuato dal controllore viene svolto attraverso un MPD (Markov Decision Process).

La funzione di valore d'azione $Q^\pi(s, a)$ valuta la qualità di una coppia stato-azione e si esprime come:

$$Q^\pi(s, a) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_{t+1} \mid s_0 = s, a_0 = a \right] \quad (3.30)$$

Dove il termine nella sommatoria esprime la reward totale stimata seguendo la policy π . Tali Q values (solitamente espressi in una matrice) esprimono quanto è "buono" trovarsi in un determinato stato e prendere una determinata decisione. La policy ottimale viene ricavata come quella che massimizza il valore d'azione:

$$\pi^*(s) = \arg \max_a Q^\pi(s, a) \quad (3.31)$$

Nei metodi denominati *off-policy*, la matrice Q viene aggiornata indipendentemente dalla politica utilizzata per indirizzare l'agente a compiere l'azione a nello stato corrente s . L'agente aggiorna il valore Q associato alla coppia stato-azione (s, a) utilizzando il valore ottimale Q dello stato successivo, in accordo con l'equazione di ottimalità di Bellman, come mostrato di seguito:

$$Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha \left[r + \gamma \max_{a' \in A} Q(s', a') \right] \quad (3.32)$$

Con α tasso di apprendimento, che assume valori tra 0 e 1.

Pertanto, la selezione dell'azione a nello stato corrente s e la selezione dell'azione a' nello stato successivo s' non derivano dalla stessa politica. Questo approccio differisce dall'approccio *on-policy*, dove la selezione dell'azione nello stato successivo s' segue la stessa politica utilizzata per selezionare l'azione a nello stato corrente s [13]. L'obiettivo principale è quindi quello di apprendere una policy $\pi(s)$, che determina la miglior azione da compiere in ogni stato per massimizzare la ricompensa cumulativa attesa [27] [28]. Spostandosi verso problemi di applicazione nel mondo reale, risulta immediato comprendere come sia impossibile avere una matrice Q-value che discretizzi tutto lo spazio delle azioni e degli stati per modelli complessi, ad esempio per un veicolo. Pertanto viene introdotto il Deep Reinforcement Learning, il quale approssima i Q-value tramite una Deep Neural Network (DNN) [29]. Inoltre l'utilizzo di una rete neurale permette all'agente di esplorare tutte le possibili combinazioni state-action, a vantaggio della generalizzazione del controllore (con controllore generale si intende che tale agente ottiene performance buone anche su scenari o test non utilizzati in fase di training). Gli algoritmi di RL si suddividono in 5 macro-categorie:

3. SISTEMI DI CONTROLLO

- **Value-based algorithms:** Questi algoritmi si concentrano sull'apprendimento delle funzioni di valore, il cui obiettivo è valutare le policy d'azione. Esempi includono algoritmi come *Q-learning*, *double Q-learning* e *SARSA*, che imparano a stimare il valore di una coppia stato-azione. Sono particolarmente utili quando si ha a che fare con ambienti con azioni discrete.
- **Policy-based algorithms:** Gli algoritmi basati sulla policy, come *REINFORCE*, ottimizzano direttamente una policy per massimizzare la ricompensa attesa. Invece di stimare una funzione di valore, questi metodi cercano di apprendere una policy ottimale (in questo caso stocastica). Sono meno efficienti in termini di campioni (richiedono una grande quantità di dati), ma possono lavorare meglio in ambienti con azioni continue.
- **Actor-critic algorithms:** Questi algoritmi combinano l'apprendimento di una policy e di una funzione di valore. Sono costituiti da due reti neurali: un *actor* che prende decisioni e un *critic* che valuta quanto siano buone tali decisioni. Gli algoritmi actor-critic combinano metodi value-based e policy-based, sfruttano il meglio di entrambi i mondi. Utilizzano l'efficienza dei metodi policy-based per esplorare meglio spazi di azione continui, e l'efficacia dei metodi value-based per stabilizzare e migliorare il processo di apprendimento. Esempi includono *A3C*, *A2C*, *DDPG* e *TD3*.
- **Derivative-free algorithms:** Questi algoritmi non utilizzano la retropropagazione per l'ottimizzazione della rete neurale. Al contrario, si basano su strategie evolutive o algoritmi genetici, che possono funzionare bene con funzioni discontinue o multi-modali. Tuttavia, sono meno efficienti in termini di campioni e richiedono più tempo per ottenere buone prestazioni.
- **Model-based algorithms:** Questi algoritmi cercano di apprendere un modello dell'ambiente, come la funzione di transizione o di ricompensa, per migliorare il processo decisionale. Utilizzano tecniche di apprendimento profondo per modellare la dinamica dell'ambiente, il che consente all'agente di apprendere più rapidamente e con maggiore stabilità. Tuttavia, richiedono un costo computazionale più elevato e sono suscettibili agli errori del modello.

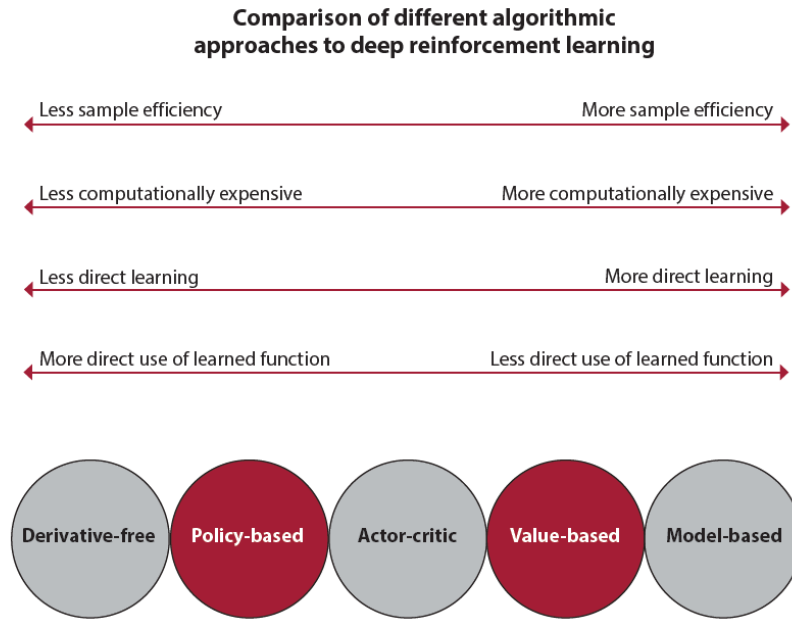


Figure 3.4: Classificazione dei principali algoritmi di RL. Tratta da [28]

3.2.1 Deep Deterministic Policy Gradient (DDPG)

Un algoritmo di spicco *off-policy* che estende i metodi di policy gradient tradizionali è il *Deep Deterministic Policy Gradient (DDPG)*, introdotto da Lillicrap [30]. Il suo framework integra il *deep learning* per approssimare sia la policy che la funzione di valore. Nel DDPG, due reti neurali principali vengono utilizzate:

- **Actor network:** suggerisce le azioni basandosi sullo stato corrente, seguendo la policy $\pi(s)$.
- **Critic network:** valuta queste azioni utilizzando i Q-values.

Il DDPG stabilizza l'apprendimento attraverso l'utilizzo del *replay buffer*, dove le esperienze passate vengono memorizzate e campionate casualmente per migliorare l'efficienza dell'addestramento (tale porzione di dati viene definita minibatch). I benefici di tale minibatch sono:

1. L'utilizzo di dati random permette di avere una minor correlazione rispetto all'apprendimento su episodi consecutivi;
2. L'experience replay migliora l'efficienza dei dati, aggiornando i pesi della policy;

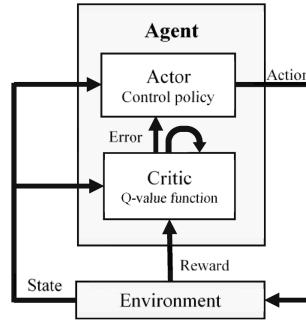


Figure 3.5: Schema di un controllore DDPG, si notano le due reti neurali (Actor e Critic). Tratta da [13]

3. Tali dati permettono di evitare problematiche di divergenza o oscillazione nei parametri durante l'addestramento, migliorandone la stabilità.

Tale architettura è stata utilizzata nel controllore sviluppato per l'attività.

3.3 Controllore Proposto

Obiettivo finale dell'attività è sviluppare un controllore per ciascun corner del veicolo (ciascun gruppo ruota) che moduli la coppia richiesta dal conducente, al fine di ridurre i picchi di accelerazione/decelerazione della massa sospesa causati da irregolarità stradali o ostacoli. Il problema di torque correction quindi ha come output la minimizzazione delle vibrazioni in direzione longitudinale.

3.3.1 Spazio degli Stati s

Il vettore degli stati s , che è l'input della rete neurale, contiene le variabili utilizzate dall'agente per calcolare l'azione di controllo:

$$s = [\Delta \ddot{x}_{b,fl/r}, \ddot{x}_{b,fl/r}, \int \Delta \ddot{x}_{b,fl/r}, T_{m,f/r}] \quad (3.33)$$

Le variabili sono definite come segue: $\Delta \ddot{x}_{b,fl/r}$ è la differenza tra l'accelerazione della carrozzeria su un corner specifico del veicolo e l'accelerazione di riferimento:

$$\Delta \ddot{x}_{b,fl/r} = \ddot{x}_{b,fl/r} - \ddot{x}_{b,ref} \quad (3.34)$$

L'accelerazione della massa sospesa $\ddot{x}_{b,fl/r}$ è calcolata come:

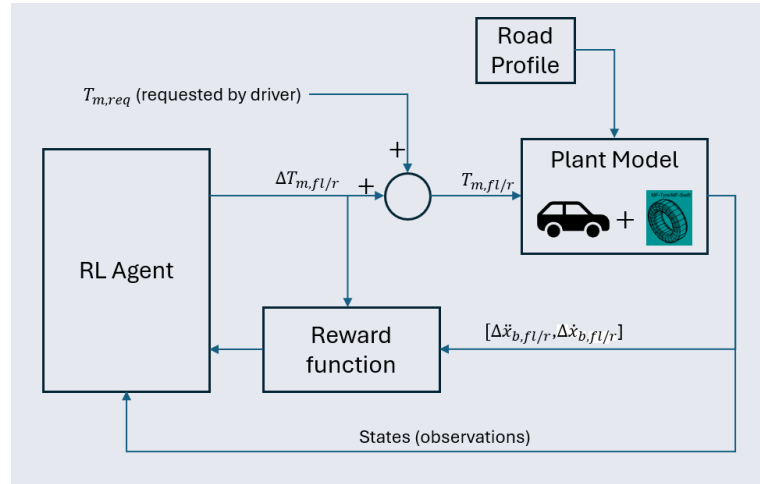


Figure 3.6: Modello del controllore proposto, comprensivo dei 3 segnali che costituiscono la reward function (in questo schema il controllore si riferisce all'asse anteriore, analogo schema si ha per i due all'asse posteriore)

$$\ddot{x}_{b,fl/r} = \frac{-k_x(x_b - x_{u,fl/r}) - c_x(\dot{x}_b - \dot{x}_{u,fl/r}) - F_{Drag}}{m_{total} - m_{u,fl/r}} + \frac{F_{t,r,req} + F_{t,fr/l,req}}{m_{total} - m_{u,fl/r}} \quad (3.35)$$

Dove $F_{t,r,req}$ è la forza totale di trazione sull'asse opposto, $F_{t,fr/l,req}$ è la forza di trazione nell'angolo opposto, e m_{total} è la massa totale del veicolo, tenendo conto dell'inerzia del motore in-wheel (IWM). L'accelerazione di riferimento della massa sospesa è data da:

$$\ddot{x}_{b,ref} = \frac{F_{t,r,req} + F_{t,fr,req} - F_{Drag} - f \cdot m_{total} \cdot g}{m_{total}} \quad (3.36)$$

L'utilizzo di due termini apparentemente identici (errore dell'accelerazione e valore assoluto) risulta di cruciale importanza nei casi in cui il pilota richieda una certa coppia (e quindi, accelerazione) ai singoli corner. Infatti per le prime serie di test si è considerata una richiesta di coppia nulla da parte del conducente (e di conseguenza, i due termini tendevano entrambi a 0 in condizioni stazionarie). Dopodichè si è svolto un secondo test sul profilo stradale al fine di mostrare la robustezza che deriva dall'utilizzare entrambi i termini (Capitolo 5), permettendo al controllore di capire in quale situazione esso si trovi ed evitando strategie di controllo che "tagliano" la coppia nonostante il conducente voglia accelerare. Ai primi due termini è stato aggiunto l'integrale dell'errore, dato che si è notato aver un effetto benefico nell'indirizzare il

3. SISTEMI DI CONTROLLO

controllore verso una strategia corretta. L'ultimo termine è il feedback della coppia motrice (post dinamica del motore, spiegata nel Capitolo 2) $T_{m,fl/r}$.

3.3.2 Spazio delle Azioni a

Lo spazio delle azioni è definito come la variazione della coppia motrice richiesta:

$$a = [\Delta T_{m,fl/r}] \quad (3.37)$$

Con i vincoli imposti dalla scheda tecnica del motore elettrico $T_{m,max/min} = \pm 1500 Nm$:

$$T_{m,min} \leq \Delta T_{m,fl/r} \leq T_{m,max} \quad (3.38)$$

3.3.3 Reward Function

La ricompensa finale di ciascun episodio R è la somma di tre termini:

$$R = \sum_{i=1}^{N_{episode}} (R_{acc}(i) + R_{speed}(i) + R_{contr.action}(i)) \quad (3.39)$$

Dove i è il passo di discretizzazione e $N_{episode}$ è il numero totale di passi nell'episodio. I singoli termini sono definiti come: Penalizzazione dell'accelerazione della massa sospesa:

$$R_{acc}(i) = -k_{acc} \cdot |\Delta \ddot{x}_{b,fl/r}| \quad (3.40)$$

Penalizzazione della velocità relativa tra massa sospesa e massa non sospesa:

$$R_{speed}(i) = -k_{speed} \cdot |\Delta \dot{x}_{b,fl/r}| \quad (3.41)$$

$\Delta \dot{x}_{b,fl/r}$ è la differenza tra la velocità della massa sospesa e quella della massa non sospesa:

$$\Delta \dot{x}_{b,fl/r} = \dot{x}_b - \dot{x}_{u,fl/r} \quad (3.42)$$

Infine vi è la penalizzazione dell'azione di controllo per evitare strategie di controllo eccessivamente oscillanti:

$$R_{contr.action}(i) = -k_{ca} \cdot |\Delta T_{m,fl/r}| \quad (3.43)$$

Da notare la differenza di notazione tra il termine dell'azione di controllo mostrata nella 3.33 e 3.43: nella prima viene fornito come segnale in feedback la coppia totale esercitata dai motori (quindi composta da coppia richiesta dal guidatore e output del controllore) mentre nella seconda viene penalizzato solamente il termine di correzione esercitato dall'agente. I pesi k sono stati ottimizzati tramite vari cicli di addestramento (procedura descritta nel 4), con i seguenti valori finali:

Peso	Valore	
k_{acc}	100	$\left[\frac{s^2}{m}\right]$
k_{speed}	1000	$\left[\frac{s}{m}\right]$
k_{ca}	0.001	$\left[\frac{1}{Nm}\right]$

Table 3.1: Pesi della reward function utilizzati

3.3.4 Architettura del Controllore

Il controllore è implementato utilizzando il toolbox di Reinforcement Learning di Matlab 2023b [31] durante la fase di addestramento, con l'algoritmo *DDPG actor-critic*. L'architettura dell'actor include due hidden layer *ReLU* da 40 neuroni ciascuno e un output layer *tanh* con 1 neurone. Il critic ha due hidden layer *ReLU* da 40 neuroni per lo state path, e due hidden layer *ReLU* da 10 e 40 neuroni per l'action path. Le due porzioni della rete critic sono incaricate di processare rispettivamente lo spazio degli stati e delle azioni. La dimensione è stata scelta dopo un test di sensibilità, svolto impostando una rete rispettivamente con una dimensione ridotta del 50% ed una seconda con una dimensione aumentata del medesimo valore. Dalla fase di training è emerso che la rete a dimensione minore mostrava un training più rapido ma i risultati ottenuti non sono stati giudicati soddisfacenti. Viceversa la rete con dimensione maggiore ha mostrato un addestramento più lungo, ottenendo risultati simili al controllore "standard", confermando di fatto la scelta della dimensione corretta. Dopo le prime fasi di training si è notato che il controllore aveva difficoltà nel mantenere l'output di coppia pari a zero, bensì risultava in una piccola coppia (nell'ordine della decina di Nm) che provocava un leggero offset dell'accelerazione $\ddot{x}_{b,fl/r}$. Questo si è scoperto essere dovuto all'output layer costituito da una funzione *tanh*, la quale presenta problemi nell'intorno dello zero (a causa della forma di tale funzione) che ovviamente vengono amplificati una volta scalato il segnale (ricordiamo infatti che tale output layer produce un valore tra -1 ed +1, il quale viene moltiplicato per il massimo della coppia). Il problema è stato risolto utilizzando un secondo blocco posizionato all'uscita del blocco RL, costituito da un saturation block tra 0.1-1 (quindi in maniera speculare, qualsiasi valore tra -0.1, 0.1 viene imposto pari a 0) seguito da una lookup table che riporta il corrispettivo di coppia all'uscita. Si utilizza una lookup table per fare in modo che la coppia parta da un valore di 0 Nm per un output di 0.1 per poi salire linearmente (evito quindi una discontinuità passando a valori poco superiori a 0.1 che avrei nel caso utilizzassi solamente un saturation block). Uno schema semplificato è mostrato in Figura 3.7. Infine il tempo di campionamento dell'agente è stato impostato come nel controllore NMPC sviluppato in [10], con $t_s = 15$ ms.

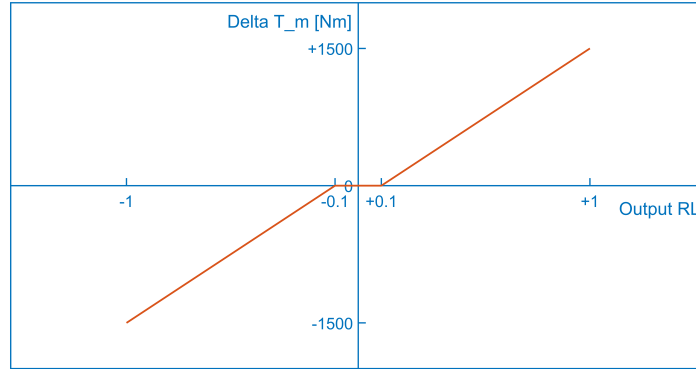


Figure 3.7: Rappresentazione dello Scaling block utilizzato per migliorare le prestazioni del controllore nell'intorno dello zero

3.3.5 Introduzione della Preview

Per un confronto equo con un controllore NMPC (per massimizzare le prestazioni), è stato aggiunto un blocco di preview per creare un agente capace di agire in modo predittivo al fine di smorzare le irregolarità della strada. Questo blocco di preview utilizza lo stesso modello di inviluppo del NMPC presentato in [10], ma è stato integrato con un modello di previsione interno simile a quello descritto nel paragrafo precedente. Questo perché l'agente non può apprendere direttamente dall'anteprima della strada, cioè da $W_{fl/r}, \beta_{y,fl/r}$; al contrario, richiede un modello di previsione interno per calcolare gli effetti di quel profilo stradale in termini di velocità relativa $\Delta \dot{x}_{b,fl/r(prev)}$ e fornire tale input come una delle observation necessarie al controllore per comprendere l'azione corretta da eseguire. Altri input del blocco di preview includono la coppia richiesta dal conducente per ciascun corner e la coppia totale richiesta, riassunti di seguito:

$$OnlineData = [\Delta \dot{x}_{b,fl/r(prev)}, T_{m,tot,req}, T_{m,r,req}] \quad (3.44)$$

L'orizzonte di preview è stato impostato come quello dell'NMPC "Real time", con 2 step di preview su un orizzonte di previsione pari a 30 ms.

Data la difficoltà incrementale del progetto, è stato realizzato dapprima un *controllore privo del blocco preview*, al fine di ottimizzare le fasi di addestramento e tuning dei pesi della reward function. Una volta ricavato il controllore corretto si è passati poi ad integrare il blocco preview, permettendo quindi un confronto alla pari con il menzionato NMPC. Il controllore una volta ultimato viene presentato in Figura 3.9. Le fasi di addestramento dei due controllori e relativi risultati sono mostrati nei 4 e 5.

3.3. CONTROLLORE PROPOSTO

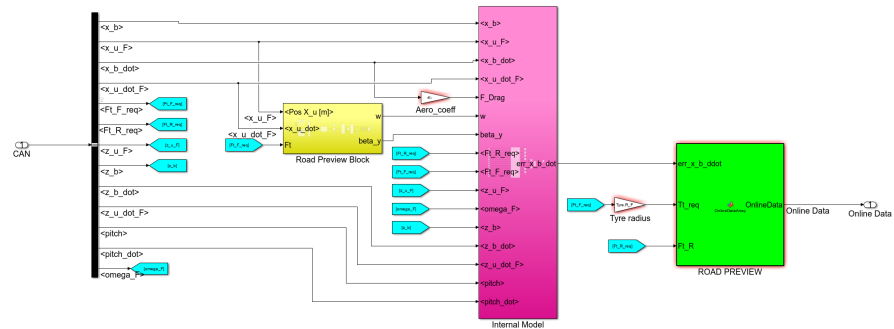


Figure 3.8: Preview subsystem responsabile della generazione della previsione stradale, accompagnato dall'internal model per restituire tale input come $\Delta \dot{x}_{b, fl/r}(prev)$

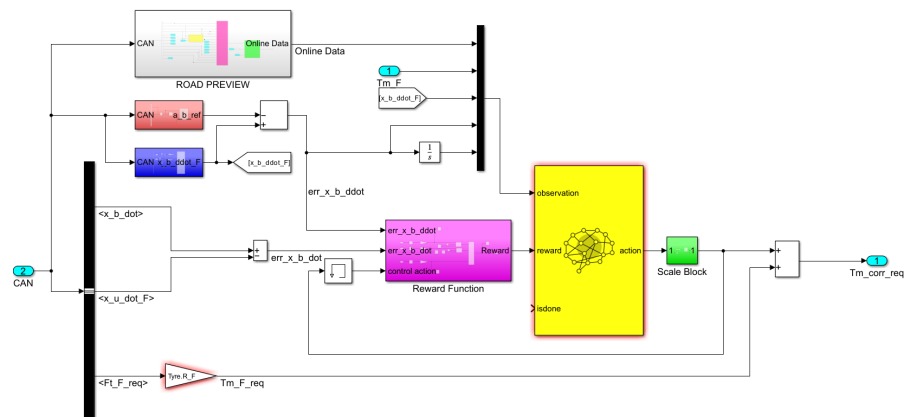


Figure 3.9: Modello Simulink del controllore completo, si possono notare i blocchi che generano: l'accelerazione di riferimento $\ddot{x}_{b,ref}$ (rosso), l'accelerazione della massa sospesa $\ddot{x}_{b,fl/r}$ (blu), la preview (grigio), la reward function (rosa), lo scaling block (verde) ed infine il controllore RL (giallo). La correzione viene infine sommata alla coppia richiesta dal conducente nel blocco somma posizionato prima dell'uscita

4 TRAINING E TEST

In questo capitolo viene descritta la procedura di addestramento utilizzata per il controllore proposto, con successivo test per estrapolare il miglior agente tra quelli ottenuti. Le prime due sezioni descrivono la fase di addestramento, concentrandosi su scenari utilizzati e criticità riscontrate, mentre le ultime due descrivono l'evoluzione dell'addestramento sino a giungere a quello finale (mostrando come si comportano i diversi agenti ottenuti).

4.1 Scenari di Addestramento

La scelta degli scenari sul quale addestrare il controllore è una decisione critica per l'ottenimento di un agente efficace. L'obiettivo è quello di ottenere un controllore che sia in grado di superare efficacemente diversi ostacoli mostrando quindi una buona generalizzazione dei diversi casi riscontrabili. I 3 scenari utilizzati in questo studio sono: a) strada con un profilo a singolo gradino, b) un dosso singolo, c) un profilo stradale reale campionato sperimentalmente. Si sono scelte queste 3 prove poichè rappresentano la quasi totalità delle casistiche che un veicolo può essere chiamato a dover superare. Inoltre questi 3 scenari sono stati affrontati dal controllore sviluppato in [10] permettendo quindi un diretto collegamento con il lavoro precedente. Partendo con il primo scenario, il profilo stradale è stato realizzato con una funzione a gradino, con l'altezza dell'ostacolo che viene fatta variare tra: $10mm \leq h_{step} \leq 30mm$

Nel secondo caso il profilo del dosso è stato modellato come definito in [32]:

$$z_r = \frac{h_b}{2} \left[1 + \cos \left(\frac{\pi x_p}{b_b} \right) \right] \quad (4.1)$$

dove h_b rappresenta l'altezza del dosso, b_b è la sua semi-larghezza e x_p la coordinata longitudinale. Si sono utilizzate dimensioni caratteristiche reperite da cataloghi online e riassunte nella tabella 4.1.

	Step 1	Step 2	Step 3	Bump 1	Bump 2	Bump 3
h (mm)	10	20	30	55	40	30
b (mm)	-	-	-	450	300	250

Table 4.1: Altezza e base degli step e bump utilizzati nel training

4. TRAINING E TEST

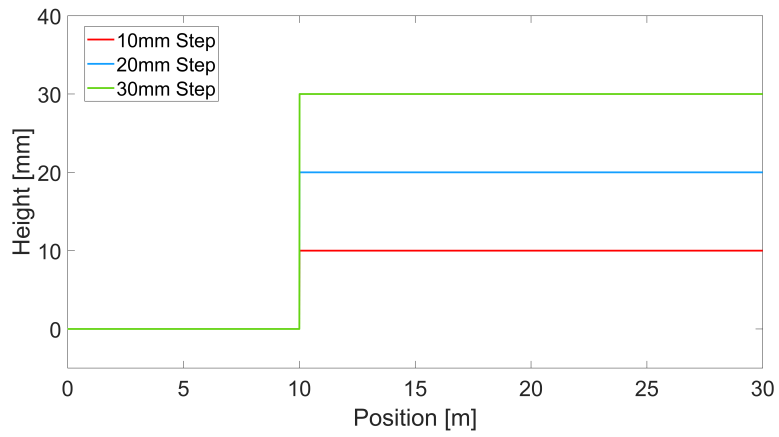


Figure 4.1: Scenario a gradino, con le tre prove che sono state utilizzate nel training

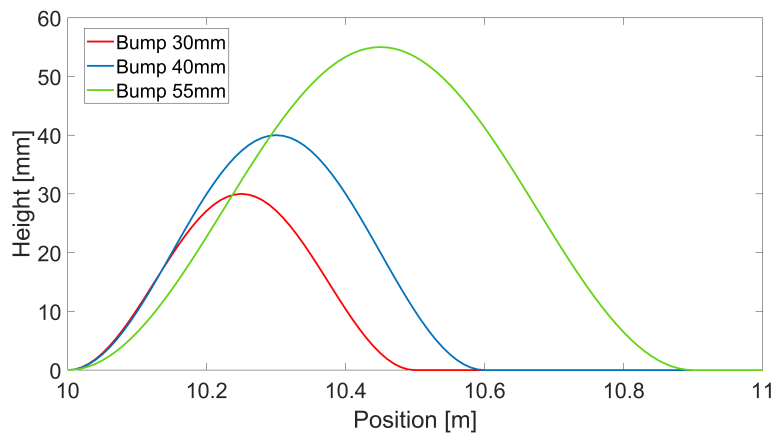


Figure 4.2: Scenario con dosso, sono mostrate le tre prove utilizzate nell'addestramento

La terza tipologia di prove è costituita da un profilo stradale ricavato tramite una misura sperimentale, su una porzione di strada reale. Il profilo è stato classificato come ISO-C. La normativa ISO-8608-2016 prevede infatti la classificazione dei profili stradali in base al loro grado di rugosità, valutata tramite la PSD (Power Spectral Density) della loro elevazione. La forma generale della PSD è data dalla formula

$$G_d(n) = G_d(n_0) * \left(\frac{n}{n_0} \right)^{-w} \quad (4.2)$$

con $n_0 = 0.1 \text{ cicli/m}$ è la frequenza spaziale di riferimento, w è un esponente che nel caso di PSD a velocità costante è posto pari a 2 (il vantaggio di ciò è poter ottenere il profilo stradale semplicemente integrando del rumore bianco). A seconda del valore di PSD le strade sono suddivise in una serie di bande, indicate dalle lettere A-H (dalla meno frastagliata alla più dissestata), come mostrato in Figura 4.5. Il valore della PSD mi permette di risalire all'elevazione media del profilo (RMS) di ogni ottava individuata dalla normativa tramite la seguente formula:

$$RMS_{road} = \sqrt{G_d(n_0)} \quad (4.3)$$

Nella simulazione descritta in questo lavoro, si considera la strada come isotropa, pertanto il profilo visto dalle due metà del veicolo è il medesimo, semplificando la trattazione dato che non si rende necessario l'utilizzo di funzioni di cross-correlazione. Il campione di strada utilizzato per il training è stato ottenuto suddividendo il profilo di partenza in una serie di sotto-sezioni di lunghezza predefinita, pertanto è da notare che tali profili presi singolarmente possono avere PSD differenti con il profilo unificato di tipo C di partenza (a beneficio di una maggior generalizzazione del controllore).

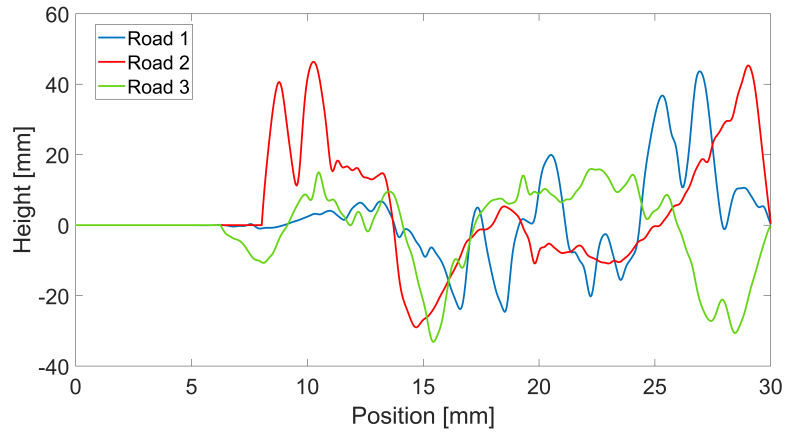
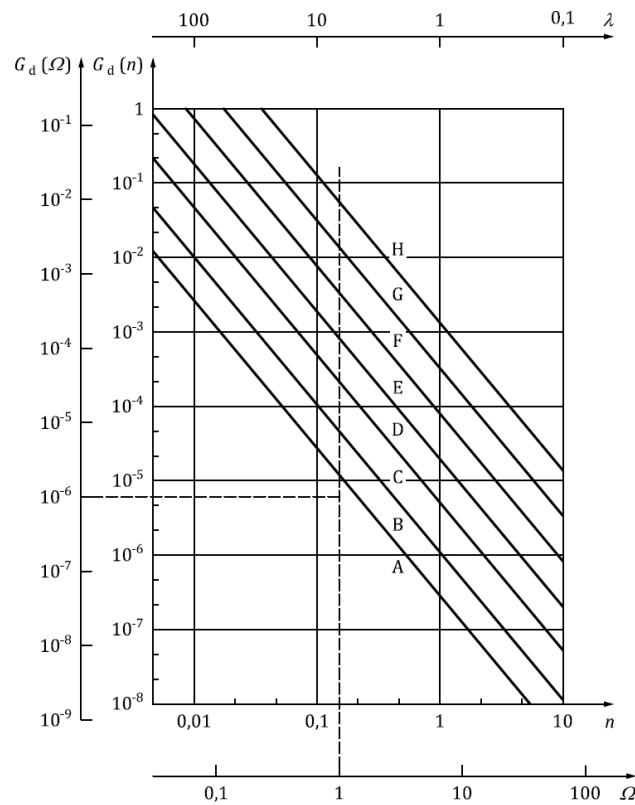


Figure 4.3: porzioni di strada utilizzate nel training

4. TRAINING E TEST



Key

λ wavelength, m n spatial frequency, cycles/m
 $G_d(n)$, $G_d(\Omega)$ displacement power spectral density, m^3 Ω angular spatial frequency, rad/m

Figure 4.4: Classificazione di profili stradali in base alla loro PSD. Si notano le diverse ottave disposte in ordine crescente di rugosità. Tratta dalla normativa [33]

Road class	Degree of roughness			
	Lower limit	Geometric mean	Upper limit	Geometric mean
	Spatial frequency units, n			
		$G_d(n_0)^a$		$G_v(n)$
		10^{-6} m^3		10^{-6} m
A	—	16	32	6,3
B	32	64	128	25,3
C	128	256	512	101,1
D	512	1 024	2 048	404,3
E	2 048	4 094	8 192	1 617
F	8 192	16 384	32 768	6 468
G	32 768	65 536	131 072	25 873
H	131 072	262 144	—	103 490

Figure 4.5: Suddivisione dei profili stradali in funzione della loro PSD. Essendo delle bande sono individuati i limiti superiori ed inferiori di ciascuna categoria, nonché il valore $G_d(n_0)$ per la frequenza di riferimento

4.2 Setup e Criticità Riscontrate

La fase di addestramento del controllore è stata una delle parti più critiche del progetto. Dato il grado di innovazione di tali algoritmi non è descritta una procedura univoca in letteratura, pertanto è stato seguito un approccio trial and error per determinare il corretto setup dei pesi relativi ai singoli termini della Reward Function.

Una delle criticità presenti in questa fase è stata l'evitare il fenomeno della *Interferenza Catastrofica* (o *Catastrophic Forgetting*), spiegato nella sezione successiva. I restanti paragrafi descrivono il setup hardware utilizzato per l'addestramento e lo script utilizzato per valutare il miglior agente tra quelli ottenuti.

4.2.1 Interferenza Catastrofica

Il catastrophic forgetting (o interferenza catastrofica) è un problema comune nelle reti neurali e nei modelli di apprendimento automatico, in particolare nel contesto dell'apprendimento sequenziale o continuo. Si verifica quando un modello, addestrato su un nuovo compito o su nuovi dati, dimentica completamente quanto appreso in compiti o dati precedenti. Questo problema è particolarmente pronunciato quando i dati del nuovo compito non sono correlati a quelli visti in precedenza, come per esempio lo scenario randomizzato utilizzato nel training finale del controllore proposto. L'addestramento di una rete neurale prevede solitamente che i dati siano *identicamente ed indipendentemente distribuiti (i.i.d.)* tuttavia questo collide con l'architettura utilizzata dal Reinforcement Learning, dove vi è una non-stazionarietà della distribuzione dei dati, ovvero quando essa cambia nel tempo. Nel reinforcement learning (RL) i dati provengono da osservazioni sequenziali e correlate nel tempo. Inoltre, durante il processo di apprendimento, la policy dell'agente evolve, generando ulteriori cambiamenti nella distribuzione dei dati, che porta a non stazionarietà [34].

Nella Figura 4.6 è presentato uno schema di come si presenta tale fenomeno, visualizzabile durante la fase di addestramento dalla forma che assume il grafico Ricompensa-Tempo. Risulta doveroso far notare che l'agente deve poter esplorare un ampio set di azioni possibili, specialmente durante le prime fasi di addestramento, per cui eventuali fluttuazioni della ricompensa del singolo episodio sono la normalità se avvengono nei primi episodi di addestramento (solitamente si può impostare una soglia del 25% del numero totale di episodi). Il fenomeno in questione mostra un marcato distaccamento dall'andamento del grafico fino all'episodio precedente e solitamente l'agente non riesce a "recuperare" la policy corretta, portando a scartare l'addestramento. Il metodo utilizzato per mitigare tale fenomeno è un tuning adeguato del replay buffer, citato nel Capitolo 3. Esso rappresenta contemporaneamente la soluzione ed il problema stesso, essendo che ripropone all'agente un campione randomizzato

4. TRAINING E TEST

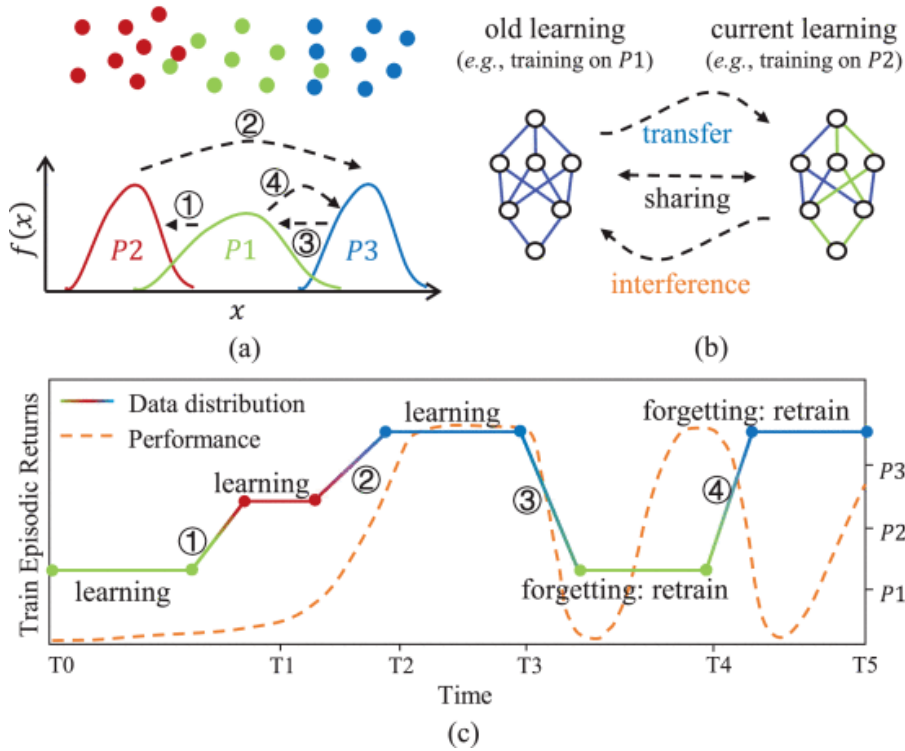


Figure 4.6: Rappresentazione del Catastrophic Forgetting, evidenziato dal grafico (c) nel quale l'agente passa dallo stato 3 allo stato 4, ritornando di fatto ad utilizzare la policy P1 e ripetendo il processo di apprendimento. Tratto da [34]

di dati da episodi precedenti (quindi la non stazionarietà è estremamente marcata) che può aiutare a ridurre l'effetto dell'interferenza tra esperienze nuove e vecchie, rallentando il degrado delle prestazioni su vecchi compiti. Le prestazioni dell'agente dipendono molto da come viene configurato il buffer (ad esempio, la dimensione del buffer, la frequenza con cui vengono aggiornate le esperienze). Se la dimensione del buffer è troppo piccola, potrebbe non contenere una rappresentazione adeguata di tutte le esperienze rilevanti; se è troppo grande, potrebbe memorizzare esperienze ormai irrilevanti o ridondanti.

4.2.2 Hardware utilizzato

Per la fase di addestramento si è utilizzato un portatile con le seguenti specifiche, riassunte in Tabella 4.2.

Tale Hardware ha permesso il completamento delle fasi di addestramento più brevi (numero episodi = 5000) nell'intorno delle 6 ore di simulazione, mentre per il training finale da 10000 episodi sono state necessarie all'incirca 12 ore per completarlo. Una volta concluso l'addestramento si è svolta un'iterazione

Processore	11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz 2.30 GHz
RAM installata	16,0 GB (15,7 GB utilizzabile)

Table 4.2: Specifiche del PC

tra gli agenti al fine di ricavarne il migliore. Tale fase ha richiesto lo stesso numero di ore del training corrispondente (6h-12h). Importante specificare che la versione di Matlab utilizzata (già specificata nel Capitolo 3) presenta tempi e risultati di addestramento completamente diversi da versioni precedenti o successive (ad esempio Matlab 2024a) seppur eseguite sulla stessa periferica.

4.2.3 Ricavare il Best Agent

L'ottenimento del miglior agente dopo la fase di addestramento ha richiesto un'attenzione particolare: si è infatti notato sin da subito come il solo valore della episode reward non fosse sufficiente a classificare i vari agenti ottenuti (addirittura reward simili potevano portare a strategie di controllo totalmente differenti tra un controllore ed un altro). Si è quindi integrato il primo approccio impostando un valore soglia di reward per considerare "buono" un agente (-5000) per poi passare ad una procedura iterativa che prevede la valutazione del valore efficace (o root mean square) dell'accelerazione longitudinale della massa sospesa $RMS_{\ddot{x}_b}$, classificando come migliore il controllore che ne minimizza il valore. L'iterazione è svolta mantenendo il medesimo scenario utilizzato nell'addestramento (singolo step nel caso di addestramenti su scenario singolo, per addestramenti su scenari randomizzati si è utilizzato uno scenario custom comprensivo di tutte e 3 le tipologie di ostacolo utilizzate, mostrato in Figura 4.7).

4. TRAINING E TEST

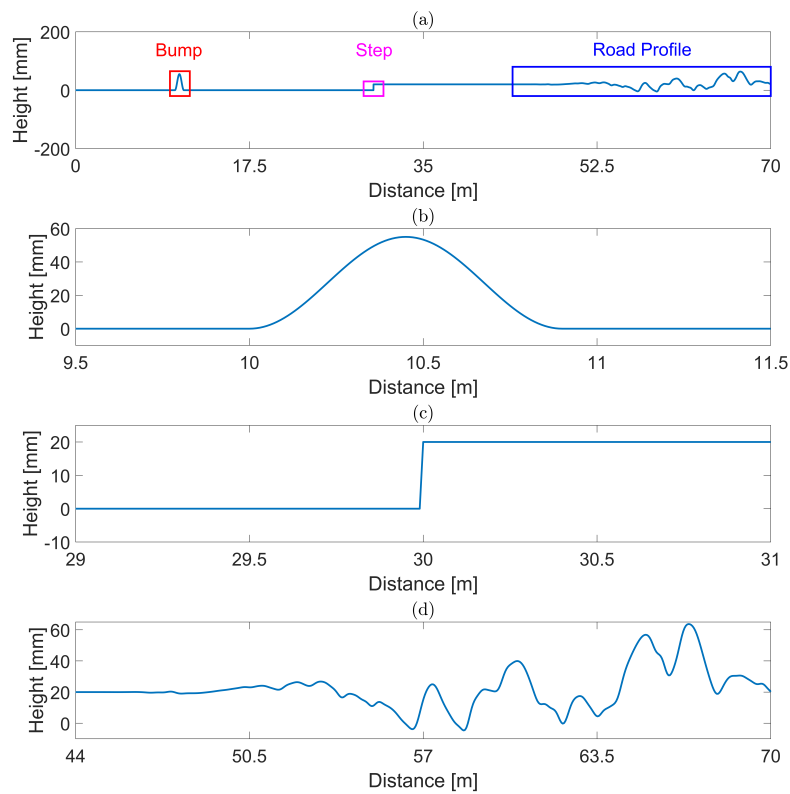


Figure 4.7: Scenario di valutazione, con il dettaglio dei singoli ostacolo. Le zone piatte tra un ostacolo ed il successivo sono necessarie per far ritornare il veicolo ad una condizione stazionaria

Di seguito è riportato una parte dello script che si occupa di tale iterazione tra gli agenti:

Listing 4.1: Script di iterazione per ricavare il miglior agente dal training

```

1
2 for index = [1:1:5000];
3 clear max
4 counter = counter + 1;
5 filename = ['Agent',num2str(index),'.mat'];
6
7
8 if isfile(['savedAgents\Agent',num2str(index),'.mat'])
9     load(['savedAgents\Agent',num2str(index),'.mat'])
10    agent = saved_agent;
11    num2str(index)
12    out = sim('Plant_Model_inwheel_motors')
13    T_SteadyState = 0.3;
14    T_1 = 0.5;
15    T_2 = 1.5;
16    position_1 = find(round(out.Plant_model_SWIFT.
17        Accelerations.x_b_ddot.time,4)==T_1);
18    position_2 = find(round(out.Plant_model_SWIFT.
19        Accelerations.x_b_ddot.time,4)==T_2);
20    Position_SteadyState = find(round(out.Plant_model_SWIFT.
21        Accelerations.x_b_ddot.time,4)==T_SteadyState);
22    x_b_ddot_ss = out.Plant_model_SWIFT.Accelerations.x_b_ddot.
23        Data(Position_SteadyState);
24    pitch_ddot_ss = out.Plant_model_SWIFT.Accelerations.pitch_
25        ddot.Data(Position_SteadyState);
26    t_analysis = T_1:Ts_sim:T_2;
27    RMS_x_b_ddot = round(sqrt(1/(T_2-T_1)*(trapz(t_analysis, (
28        abs(out.Plant_model_SWIFT.Accelerations.x_b_ddot.Data(
29            position_1:position_2) - x_b_ddot_ss).^2))),3);
30
31    if RMS_x_b_ddot < Best_RMS;
32        Best_RMS = RMS_x_b_ddot;
33        save(['Agent_best_reward_',num2str(index),'.mat'],'agent')
34    end
35 end
36 end

```

La procedura prevede l'individuazione di 3 istanti, denominati "steady state", " T_1 " (pre-ostacolo) e " T_2 " (post-ostacolo). Il primo dei tre è necessario per poter definire il valore dell'accelerazione nel caso stazionario, che andrà sottratto all'accelerazione ricavata tra i gli istanti 1 e 2. Il modello richiede circa 0.2s per potersi assestare pertanto nella pratica è stato scelto un valore che non comprendesse tale oscillazione iniziale del modello verso la condizione di equilibrio.

4.3 Training degli agenti

In questa sezione viene mostrata l'evoluzione della fase di addestramento degli agenti, iniziando da setup più semplici ed incrementando gradualmente il grado di complessità.

4.3.1 Training preliminare (scenario singolo e senza preview)

Come prima tipologia di addestramento, necessaria per definire i pesi dei termini nella reward function, è stato utilizzato un singolo scenario con un gradino da 20mm (si è deciso di mantenerlo uguale ad una delle prove mostrate nell'articolo di riferimento [10]). Il controllore non presenta il blocco responsabile della generazione preview del profilo stradale, pertanto viene utilizzato come un classico controllore in feedback. Il numero massimo di episodi di addestramento è stato impostato in base alla varietà degli scenari casuali possibili a cui l'agente è sottoposto. Per questo addestramento, essendo utilizzato solamente 1 scenario, il numero è stato fissato a 5000 episodi.

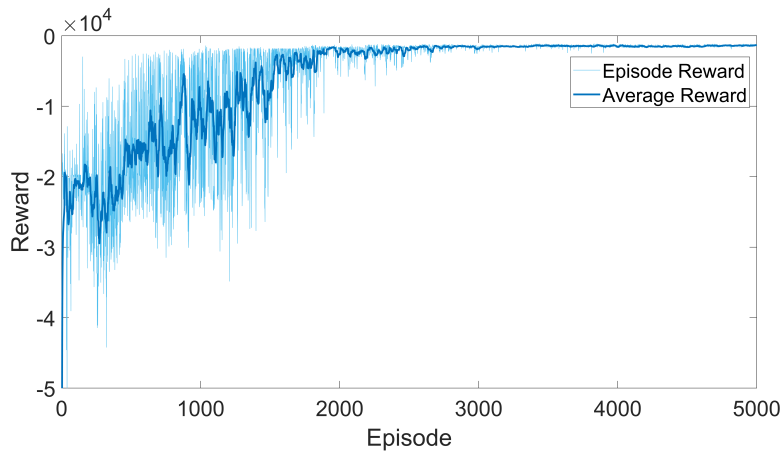


Figure 4.8: Grafico dell'addestramento, con indicate la reward media (in blu scuro) e la reward del singolo episodio (in azzurro). Si nota la fluttuazione della reward, causata dall'esplorazione del controllore durante le prime fasi.

Come si può vedere, il grafico della reward totale (Figura 4.8) raggiunge un punto di convergenza intorno all'episodio 3500. Dai risultati mostrati in Figura 4.9 si nota che il controllore non riesce a smorzare il primo picco di decelerazione (data l'azione in feedback), tuttavia mostra un ottimo andamento nella zona post-urto permettendo di migliorare la condizione rispetto al caso passivo (privo di alcuna azione di controllo).

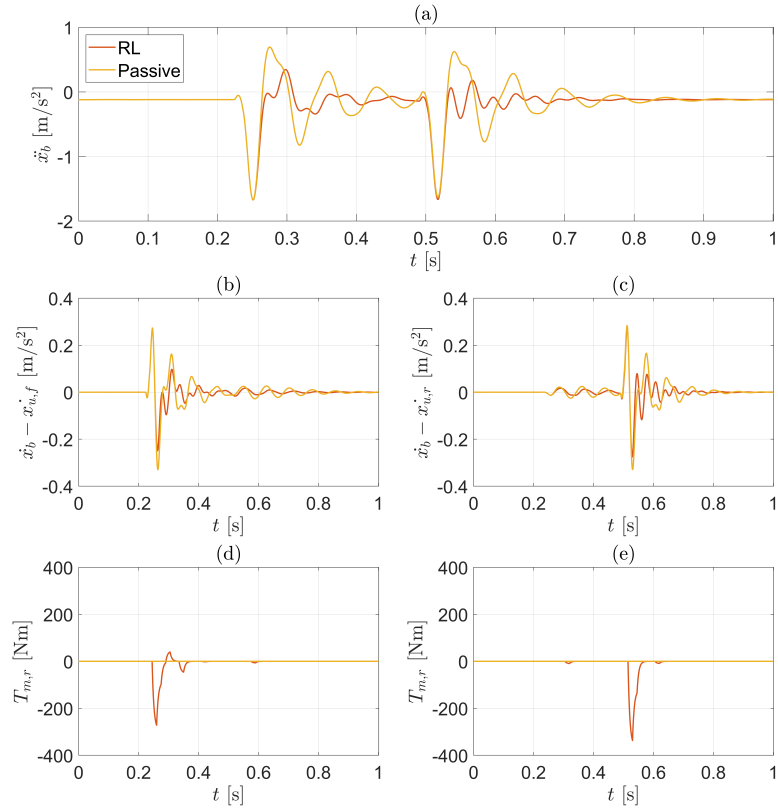


Figure 4.9: Risultati del controllore ottenuto sullo scenario step 20mm: si nota come il controllore abbia compreso l'azione corretta da svolgere, validando di fatto il setup utilizzato dei pesi nella reward function

4.3.2 Training con preview e scenario singolo

Dopo aver completato il primo addestramento con successo, ed aver ricavato il setup corretto della reward function, si è introdotto il blocco preview spiegato nel Capitolo 3. Tuttavia come descritto in precedenza, tale introduzione ha richiesto diverse iterazioni al fine di verificare la correttezza del segnale in input: motivo per il quale non si è passati direttamente al training finale bensì si è mantenuto il singolo step come unico scenario utilizzato durante l'addestramento. In Figura 4.10,4.11 sono presentati il grafico dell'addestramento ed i risultati del miglior controllore ottenuto. Come si nota dall'azione di controllo, l'agente ha compreso come utilizzare l'input derivante dal blocco di previsione. Inoltre se si stacca manualmente in Simulink il segnale, l'agente produce un'azione nulla, indice del fatto che necessita di tale segnale e non c'è il rischio che abbia memorizzato solamente l'istante di intervento corretto iterando attraverso gli episodi (ma invece l'agente ha creato una correlazione tra l'input di previsione e l'output corretto). Nonostante l'azione di controllo sia positiva in ogni istante della manovra (senza le oscillazioni che ha mostrato il controllore in feedback) il controllore riesce a smorzare adeguatamente le vibrazioni nella zona post-urto.

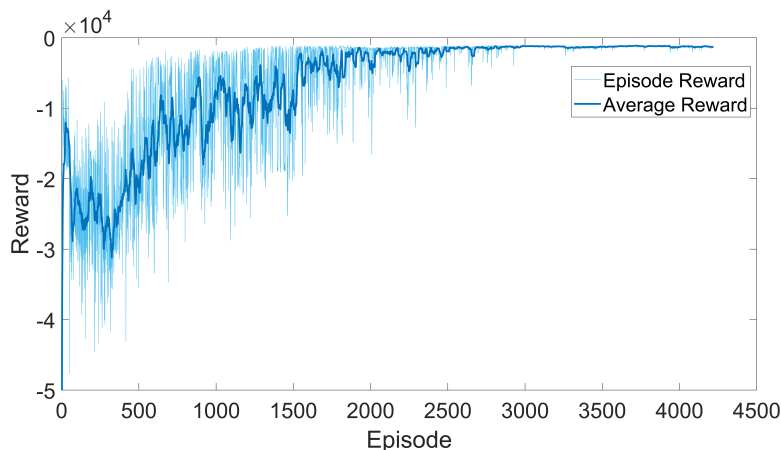


Figure 4.10: Training con preview e scenario singolo: si nota anche in questo caso la convergenza ottenuto circa all'episodio 3500

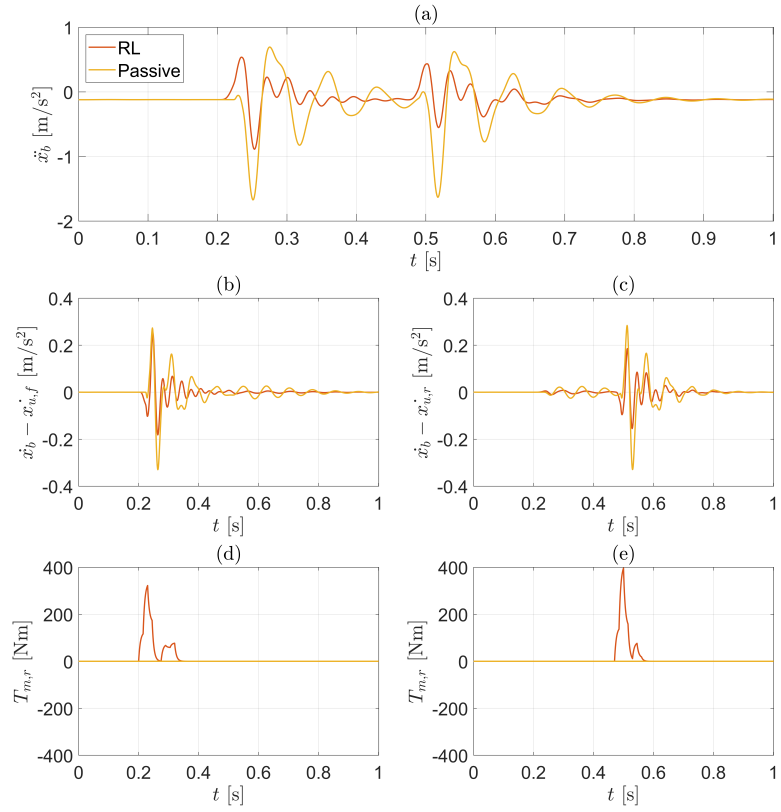


Figure 4.11: Risultati del controllore con previsione: l'azione di controllo ora permette di smorzare anche il primo picco negativo (indice che il controllore riceve un input corretto di previsione)

4.4 Addestramento Finale e confronto con gli agenti precedenti

Successivamente, una volta verificata la correttezza del segnale di previsione, si è passati a integrare nell'addestramento finale più scenari possibili, al fine di aumentare la generalizzazione del controllore rispetto al tipo di ostacolo da superare. Sono stati considerati 9 scenari possibili in totale, randomizzati all'inizio di ogni episodio: 3 profili a gradino, 3 profili di dosso e 3 profili stradali campionati. Le dimensioni dei gradini e dossi sono stati definiti precedentemente. La scelta di randomizzare gli scenari deriva dalla necessità di disaccoppiare tra loro i singoli ostacoli e relative azioni di controllo, sfruttando l'experience buffer per far memorizzare all'agente le azioni corrette sui vari ostacoli. Inoltre l'utilizzo di scenari con diversi ostacoli consecutivi all'interno, incrementava notevolmente il tempo computazionale della simulazione portando a saturazione la memoria RAM disponibile. Il numero totale di episodi è stato aumentato a 10000, al fine di consentire al controllore di raggiungere il punto di convergenza. Nella Tabella 4.3 viene riportata la distribuzione degli scenari ripetuti durante il training.

Step 10	Step 20	Step 30	Bump 55	Bump 40	Bump 30	Road 1	Road 2	Road 3
1068	1134	1165	1110	1159	1090	1083	1061	1130

Table 4.3: Distribuzione degli scenari durante il training

A causa del grande numero di agenti ottenuti in questo addestramento, si è deciso di considerare buoni solamente quelli a partire dall'episodio 4000 (dove l'average reward inizia a convergere) e che avessero contemporaneamente una reward del singolo episodio superiore a -5000 (valore arbitrario impostato all'inizio dell'addestramento) come mostrato nella Figura 4.12. L'agente migliore tra quelli ottenuti è stato selezionato iterandolo attraverso lo scenario personalizzato con tre ostacoli consecutivi differenti (dosso, gradino, profilo stradale) definito nel paragrafo iniziale del capitolo, al fine di ottenere l'agente migliore complessivo, ovvero quello capace di minimizzare l'accelerazione della massa sospesa $\ddot{x}_{b,fl/r}$. Nelle Figure 4.13, 4.14, 4.15 sono presentati i dettagli delle singole prove che compongono lo scenario personalizzato, comparando l'azione di controllo dei tre agenti trattati.

Nel primo test (bump) si nota come i due agenti "preliminari" non comprendono quale sia l'azione migliore da intraprendere, mostrando azioni di controllo eccessivamente contenute (caso del controllore in feedback, in rosso) oppure troppo invasive (caso dell'agente con preview addestrato solo sul gradino di 20mm, in blu). Si nota invece il comportamento ottimale del con-

4.4. ADDESTRAMENTO FINALE E CONFRONTO CON GLI AGENTI PRECEDENTI

trollore finale (in verde) che smorza adeguatamente le vibrazioni, indice che tale test è stato necessario includerlo tra quelli utilizzati in addestramento a causa della strategia di controllo molto diversa rispetto al caso del singolo gradino.

Nella prova a singolo gradino, tutti e 3 i controllori mostrano delle azioni correttive ottimali, con il controllore addestrato sul singolo episodio e completo di preview (in blu) che mostra un andamento leggermente migliore dell'agente finale, dovuto al fatto che è stato addestrato (quindi in senso lato, ottimizzato) solamente su questa specifica prova.

Il caso della prova sul profilo stradale mostra invece un risultato interessante: nonostante il controllore finale sia largamente il migliore dei tre, vi è da registrare una discreta performance del controllore in feedback, il quale presenta un'azione di coppia simile al controllore finale che permette di smorzare leggermente le vibrazioni. Invece, il controllore ottimizzato sullo scenario a gradino tramite l'utilizzo della preview fatica a reggere il confronto con i due controllori, mostrando delle azioni di coppia sempre positive che ne penalizzano i risultati.

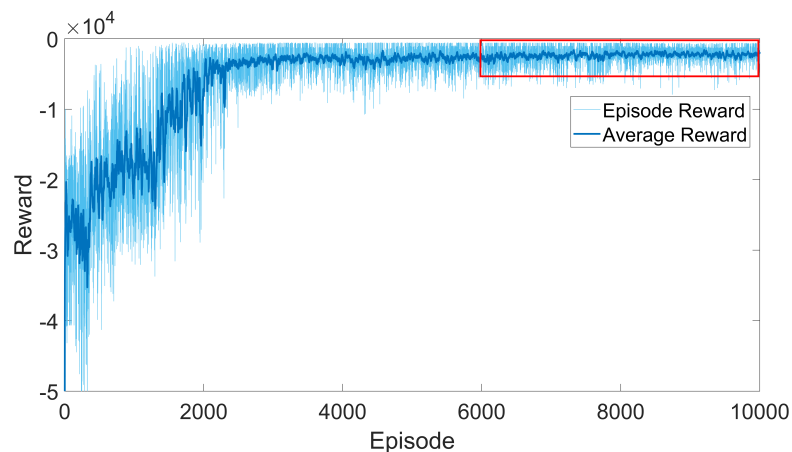


Figure 4.12: Training Finale con scenari randomizzati. In rosso sono evidenziati gli agenti considerati buoni, che verranno successivamente iterati attraverso lo scenario comprensivo dei tre ostacoli

4. TRAINING E TEST

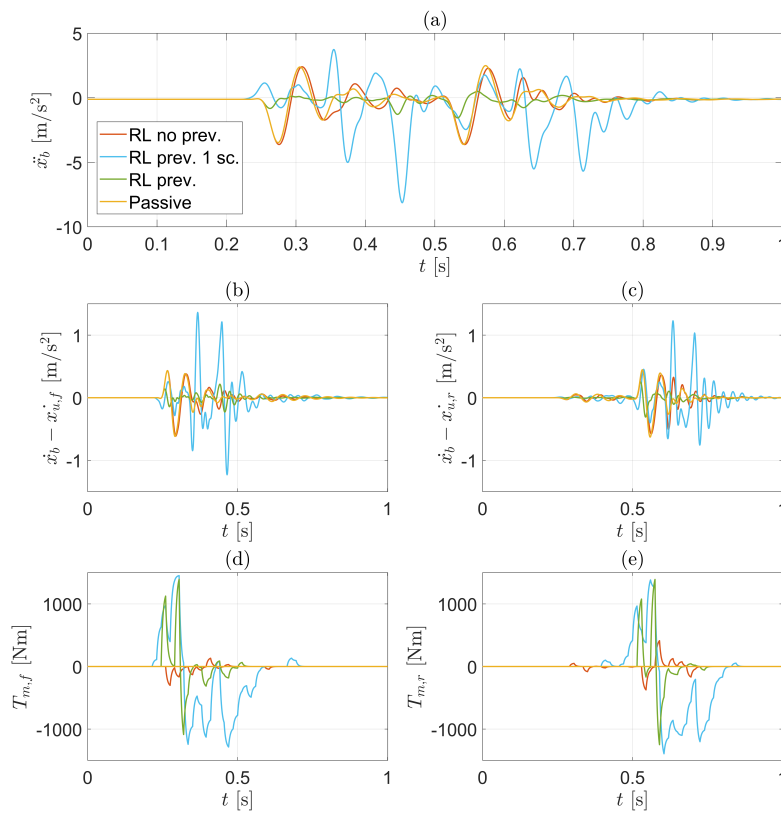


Figure 4.13: Confronto dei tre agenti sullo scenario a dosso: a differenza del controllore finale, i primi due agenti faticano a comprendere quale sia l'azione migliore da applicare a causa di un mancato addestramento su questo specifico ostacolo

4.4. ADDESTRAMENTO FINALE E CONFRONTO CON GLI AGENTI PRECEDENTI

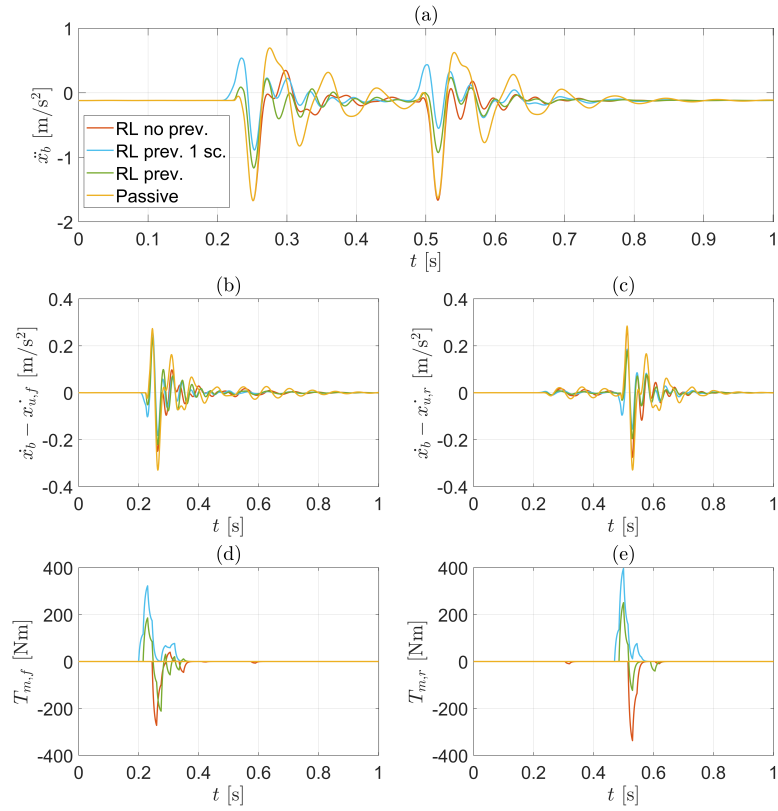


Figure 4.14: Risultati dei tre agenti sullo scenario a gradino singolo: le performance dei tre sono molto simili, con l'agente addestrato solamente su tale scenario (in blu) che mostra una vibrazione leggermente minore del controllore finale

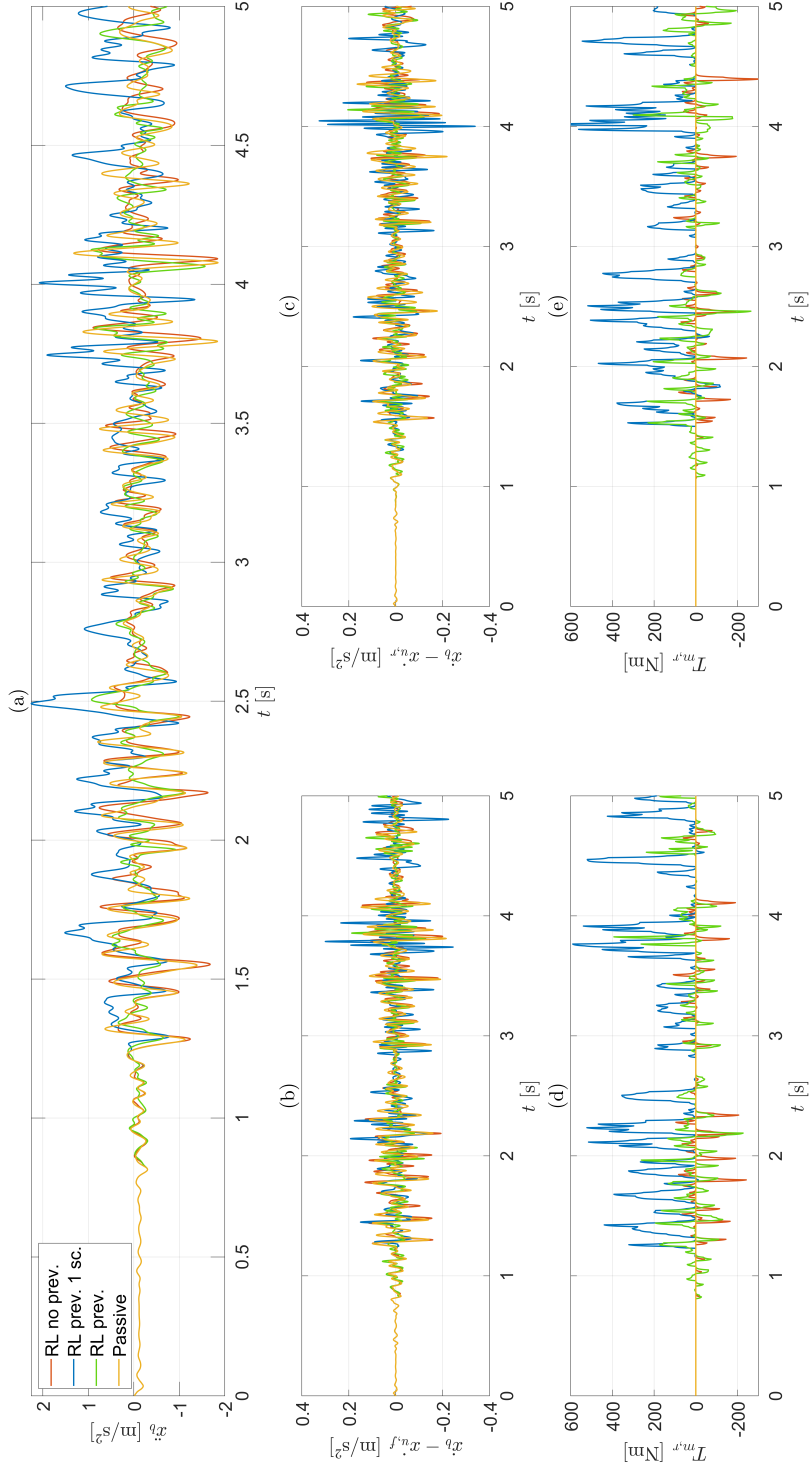


Figure 4.15: Risultati dei tre agenti sul profilo stradale reale: si nota l’ottima performance del controllore in feedback (in rosso) che risulta secondo solamente al controllore finale, mentre quello addestrato sul singolo step tramite preview fatica capire l’azione di coppia corretta da applicare

4.4. ADDESTRAMENTO FINALE E CONFRONTO CON GLI AGENTI PRECEDENTI

Table 4.4: KPIs dei 3 controllori RL addestrati

Step	$RMS_{\ddot{x}_b}$	$VDV_{\ddot{x}_b}$	$RMS_{\dot{\ddot{x}}_b}$	$\Delta\ddot{x}_{b,\max}$	$IACA_F$	$IACA_R$
RL no prev.	0.21	0.58	23.36	1.56	3.0	3.0
RL prev. 1 sc.	0.12	0.28	18.32	0.77	3.4	3.8
RL prev.	0.13	0.34	18.86	1.0500	4.6	3.4
Passivo	0.27	0.59	28.32	1.56	0	0
Bump						
RL no prev.	0.75	1.52	67.94	1.95	7.1	10.9
RL prev. 1 sc.	1.36	2.89	142.07	2.12	46.5	39.4
RL prev.	0.21	0.50	24.44	1.45	35.5	33.8
Passivo	0.70	1.47	63.61	3.5	0	0
Road						
RL no prev.	0.42	0.90	30.14	1.73	10.4	9.3
RL prev. 1 sc.	0.60	1.27	36.18	2.37	24.1	24.6
RL prev.	0.31	0.67	22.21	1.45	16.7	15.8
Passivo	0.43	0.91	34.67	1.73	0	0

5 RISULTATI E TEST DI ROBUSTEZZA

L'agente ottenuto è stato testato negli stessi scenari descritti nel precedente lavoro [10], confrontandolo con una versione "Real time" del NMPC citato nel capitolo precedente. I tre scenari includono: gradino singolo di 20mm, dosso singolo di altezza 55mm e una sezione del profilo stradale. Il confronto tra i controllori è stato eseguito valutando diversi KPI, spiegati di seguito:

- **Root Mean Square dell'Accelerazione Longitudinale**

$$\text{RMS}_{\ddot{x}_b} = \sqrt{\frac{1}{T_2 - T_1} \int_{T_1}^{T_2} (\ddot{x}_b - \ddot{x}_{b,\text{ref}})^2 dt} \quad (5.1)$$

- **Vibration Dose Value per l'Accelerazione Longitudinale**

$$\text{VDV}_{\ddot{x}_b} = \left(\int_{T_1}^{T_2} (\ddot{x}_b - \ddot{x}_{b,\text{ref}})^4 dt \right)^{\frac{1}{4}} \quad (5.2)$$

- **Root Mean Square del Jerk Longitudinale**

$$\text{RMS}_{\dddot{x}_b} = \sqrt{\frac{1}{T_2 - T_1} \int_{T_1}^{T_2} (\dddot{x}_b)^2 dt} \quad (5.3)$$

- **Deviazione Massima dell'Accelerazione Longitudinale**

$$\Delta\ddot{x}_{b,\text{max}} = \max(|\ddot{x}_b - \ddot{x}_{b,\text{ref}}|) \quad (5.4)$$

- **Integrale dell'Azione di Controllo Assoluta**

$$\text{IACA} = \frac{1}{T_2 - T_1} \int_{T_1}^{T_2} \sum_{i,j} |\Delta T_{m,ij}| dt \quad (5.5)$$

5.1 Test su controllore senza preview

Come citato nel Capitolo 4 si è testato inizialmente un controllore privo del blocco preview al fine di attestarne la correttezza dell'addestramento. Per garantire un confronto equo con il NMPC si è modificato l'orizzonte di previsione e predizione di quest'ultimo, portandolo a zero. Tuttavia data l'architettura di tale controllore, rimane comunque il termine di predizione basato sullo stato attuale. Il controllore RL invece possiede un funzionamento in feedback, come mostrato nella Figure 5.1. Di seguito sono mostrati i risultati per uno scenario con gradino singolo, confrontando l'agente RL, l'NMPC e il caso passivo (senza azione di controllo):

Come si può notare, i due controllori agiscono in istanti temporali differenti, proprio a causa di quel termine di previsione del NMPC, inoltre si

5. RISULTATI E TEST DI ROBUSTEZZA

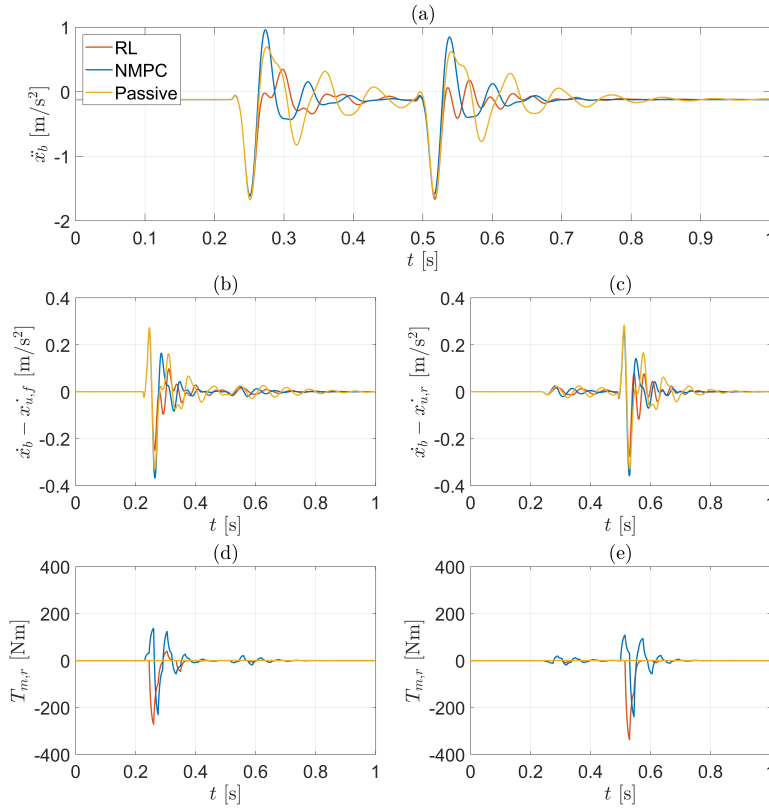


Figure 5.1: Risultati dello scenario con gradino singolo per il controllore senza preview

nota come il controllore RL non faccia nulla per smorzare il primo picco, applicando l'azione frenante solamente dopo aver subito la prima vibrazione. Complessivamente il controllore RL migliora le performance post-urto validando di fatto l'addestramento svolto e la scelta corretta dei pesi della reward function. Di seguito sono presentati i KPI:

Table 5.1: KPI per gli scenari con gradini

Step	$RMS_{\ddot{x}_b}$	$VDV_{\ddot{x}_b}$	$RMS_{\ddot{\ddot{x}}_b}$	$\Delta\ddot{x}_{b,\max}$	$IACA_F$	$IACA_R$
NMPC	0.24	0.57	28.89	1.49	5.6	5.1
RL Agent	0.21	0.58	34.19	1.56	6.5	6.3
Passive	0.27	0.59	28.32	1.56	0	0

5.2 Test su controllore con preview

Per la validazione finale si è utilizzato l'agente ricavato dall'addestramento comprendente i diversi ostacoli randomizzati. Come per il paragrafo precedente, vengono mostrati i risultati per lo scenario dello step singolo, confrontando RL Agent, NMPC e il caso passivo (senza azione di controllo):

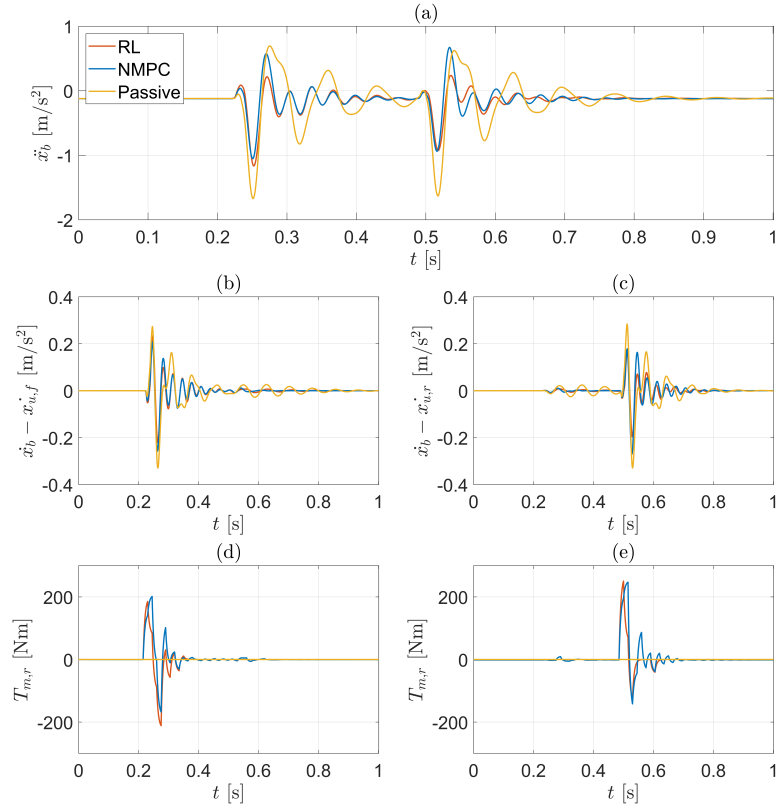


Figure 5.2: Risultati dello scenario dello step singolo. Si nota che l'RL ha smorzato maggiormente il secondo picco rispetto all'NMPC, mantenendo le stesse prestazioni nelle oscillazioni successive.

5. RISULTATI E TEST DI ROBUSTEZZA

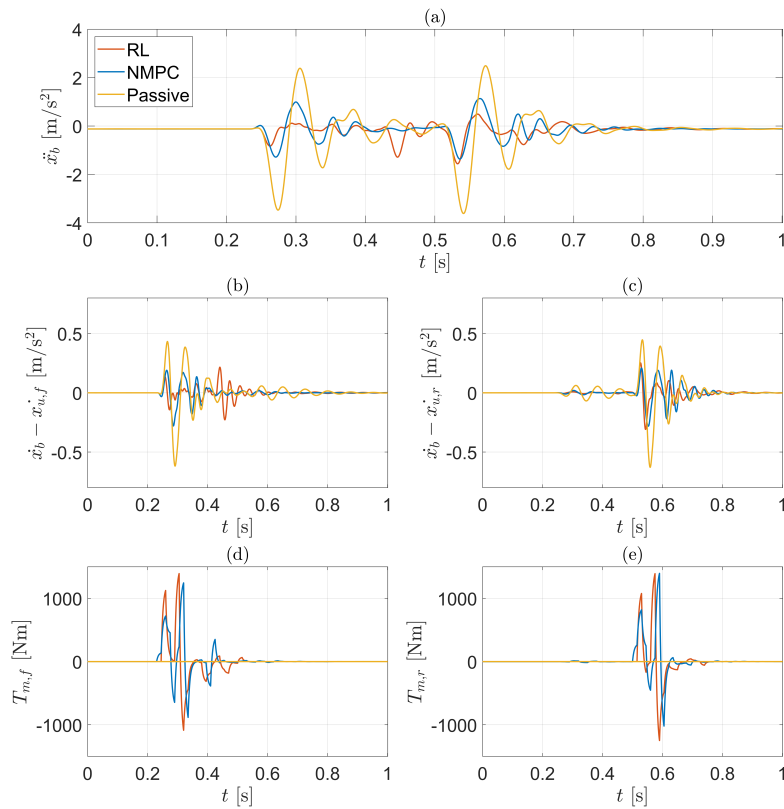


Figure 5.3: Risultati dello scenario del dosso. I grafici mostrano le migliori prestazioni dell'RL, che riesce quasi completamente a smorzare le vibrazioni.

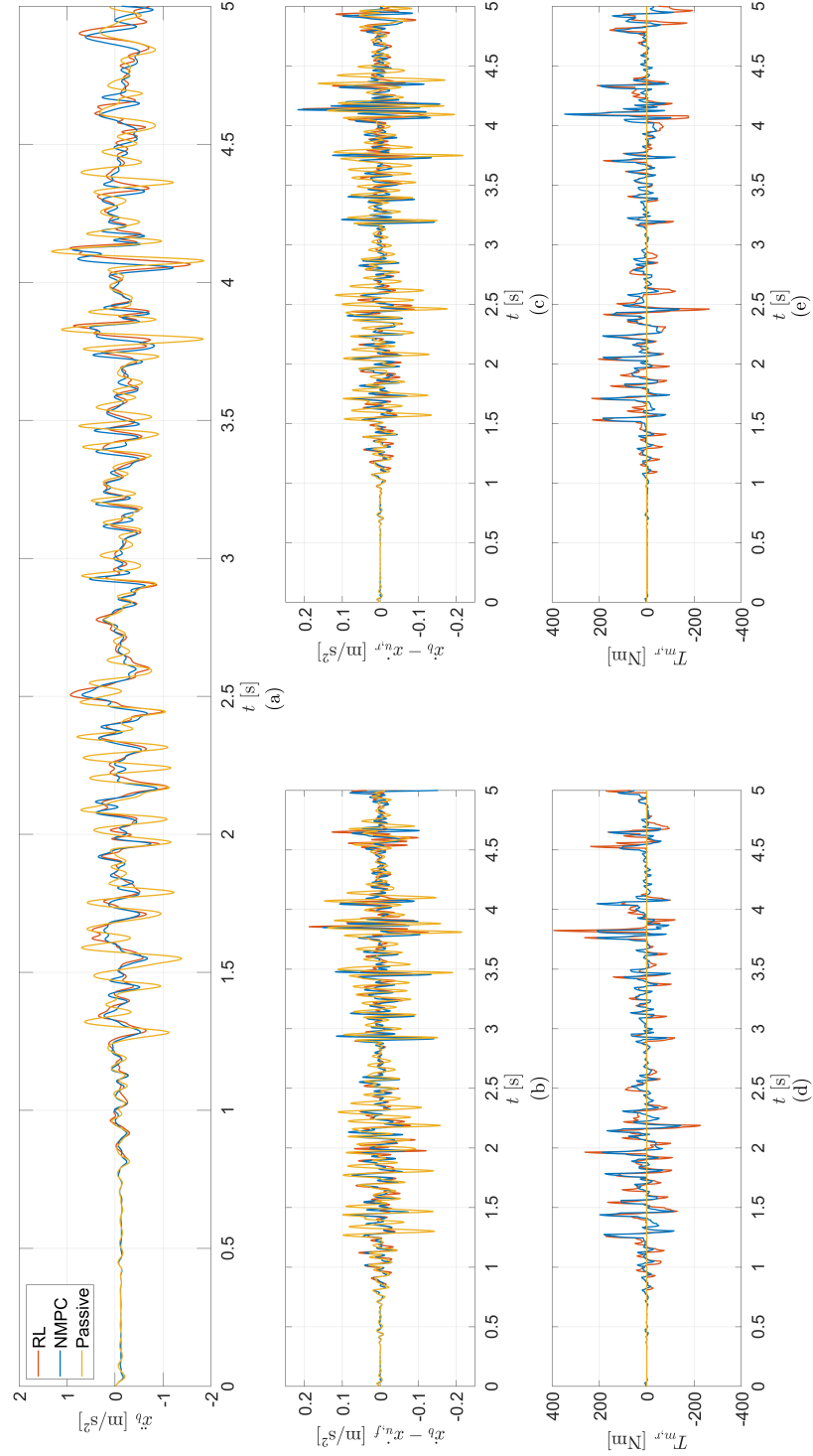


Figure 5.4: Risultati dello scenario su strada reale. I risultati mostrano un comportamento simile tra i due controllori, con un'azione di controllo più elevata applicata dall'RL.

Nel scenario dello step, sia i controllori RL che NMPC sono riusciti ad attenuare con successo il primo picco negativo dell'accelerazione, che si verifica a $t = 0.25s$, applicando una coppia positiva di 200 Nm ad entrambi gli assi. Questa azione ha ridotto l'accelerazione del primo picco del 30% rispetto al caso passivo. Per il secondo picco, situato a $t = 0.27s$, il controllore RL ha superato il NMPC, ottenendo un ulteriore smorzamento del 60%. Nel complesso, le strategie di controllo dei due controllori sono risultate simili, concentrandosi principalmente sulla minimizzazione delle prime due oscillazioni della massa sospesa per ciascun asse. I KPI riportati nella Tabella 5.2 mostrano risultati comparabili per entrambi i controllori, con una riduzione significativa del $RMS_{\ddot{x}_b}$ pari al 52%. Un aspetto notevole è che il controllore RL ha fornito un'azione di controllo più fluida, riducendo il $RMS_{\ddot{\ddot{x}}_b}$ (jerk) del 15% rispetto al NMPC. Questo miglioramento è ulteriormente supportato dalla riduzione dell'oscillazione dell'azione di controllo osservata nei subplot (d) ed (e) di Figura 5.2, particolarmente evidente nel controllore dell'asse posteriore, come quantificato dai valori più bassi di IACA. Nei risultati del test sul dosso mostrati in Figura 5.3, il controllore RL ha smorzato efficacemente le oscillazioni durante l'intero test. Le vibrazioni sull'asse anteriore sono state mitigate con successo, con il valore massimo osservato nel picco anteriore a $t = 0.26 s$ di $\ddot{x}_b = -0.83 m/s^2$. Per il primo picco si è ottenuta una riduzione del 36% e 76% rispetto ai casi NMPC e passivo, rispettivamente. Il secondo picco ha mostrato il miglioramento più significativo, con una riduzione del 90% rispetto al NMPC, compensando efficacemente l'oscillazione della massa sospesa. Sull'asse posteriore, i risultati hanno indicato una riduzione rispetto al caso passivo del 57% e 80% per il primo e il secondo picco, rispettivamente. Un risultato interessante nel subplot (a) è un piccolo picco negativo che si verifica dopo la vibrazione dell'asse anteriore e prima dell'oscillazione dell'asse posteriore (a $t = 0.45 s$). Questo fenomeno, analizzato successivamente, ha aumentato la vibrazione complessiva del 89% rispetto al NMPC. Nel complesso, i KPI riportati nella Tabella 5.2 indicano una riduzione globale di $RMS_{\ddot{x}_b}$ del 22% e 70% rispetto ai casi NMPC e passivo, rispettivamente. L'azione di controllo applicata in questo scenario ha raggiunto il limite superiore dell'EM, evidenziando la gravosità dello scenario, con un valore massimo di 1390 Nm applicato a entrambi gli assi. Nonostante l'alta oscillazione dell'azione di controllo, il $RMS_{\ddot{\ddot{x}}_b}$ è stato ridotto del 22% rispetto al NMPC. Il test finale di confronto, eseguito su un campione di strada reale, mostra un comportamento simile tra i due controllori in termini di accelerazioni. Complessivamente, il $RMS_{\ddot{x}_b}$ per il controllore RL è stato ridotto del 28% rispetto al caso passivo, anche se è 13% più alto rispetto al caso NMPC. Questa differenza è parzialmente dovuta a un'azione di controllo leggermente più aggressiva applicata dal controllore RL, con picchi di coppia di 400 Nm per l'asse anteriore rispetto ai 200 Nm raggiunti dal NMPC, e 350 Nm e 260 Nm rispettivamente per l'asse posteriore (subplot (d), (e) di Figura 5.4). Nonostante il maggiore $RMS_{\ddot{x}_b}$, il

jerk $RMS \ddot{x}_b$ è praticamente identico tra i due approcci, con entrambi i controllori che ne riducono il valore del 35% rispetto al caso passivo. Da questi test di validazione, il nuovo controllore proposto è riuscito a superare le prestazioni del precedente approccio NMPC in 2 dei 3 scenari, raggiungendo prestazioni simili nel test su strada.

Table 5.2: KPIs per gli scenari di test

Step	$RMS \ddot{x}_b$	$VDV \ddot{x}_b$	$RMS \ddot{x}_b$	$\Delta \ddot{x}_{b,max}$	$IACA_F$	$IACA_R$
NMPC	0.14	0.34	22.02	0.93	4.9	5.4
RL Agent	0.13	0.34	18.86	1.0500	4.6	3.4
Passivo	0.27	0.59	28.32	1.56	0	0
Bump						
NMPC	0.27	0.56	31.25	1.26	36.3	32.5
RL Agent	0.21	0.50	24.44	1.45	35.5	33.8
Passivo	0.70	1.47	63.61	3.5	0	0
Road						
NMPC	0.27	0.55	22.92	1.09	15.7	15.9
RL Agent	0.31	0.67	22.21	1.45	16.7	15.8
Passivo	0.43	0.91	34.67	1.73	0	0

5.2.1 Test di robustezza

Per valutare la robustezza del controllore proposto, sono stati testati diversi scenari e condizioni non incluse nella fase di addestramento. Allo stesso modo, l'NMPC utilizzato per il confronto non è stato tarato su questi scenari per garantire un confronto equo. Il primo test prevede i due controllori sullo stesso profilo stradale discusso in precedenza, utilizzando una coppia richiesta dal guidatore di 2000Nm (500Nm per ogni motore nelle ruote). Ciò è stato fatto per valutare la necessità di utilizzare sia l'errore di accelerazione $\Delta \ddot{x}_{b,fl/r}$ che l'accelerazione della massa sospesa $\ddot{x}_{b,fl/r}$ nelle osservazioni dell'agente.

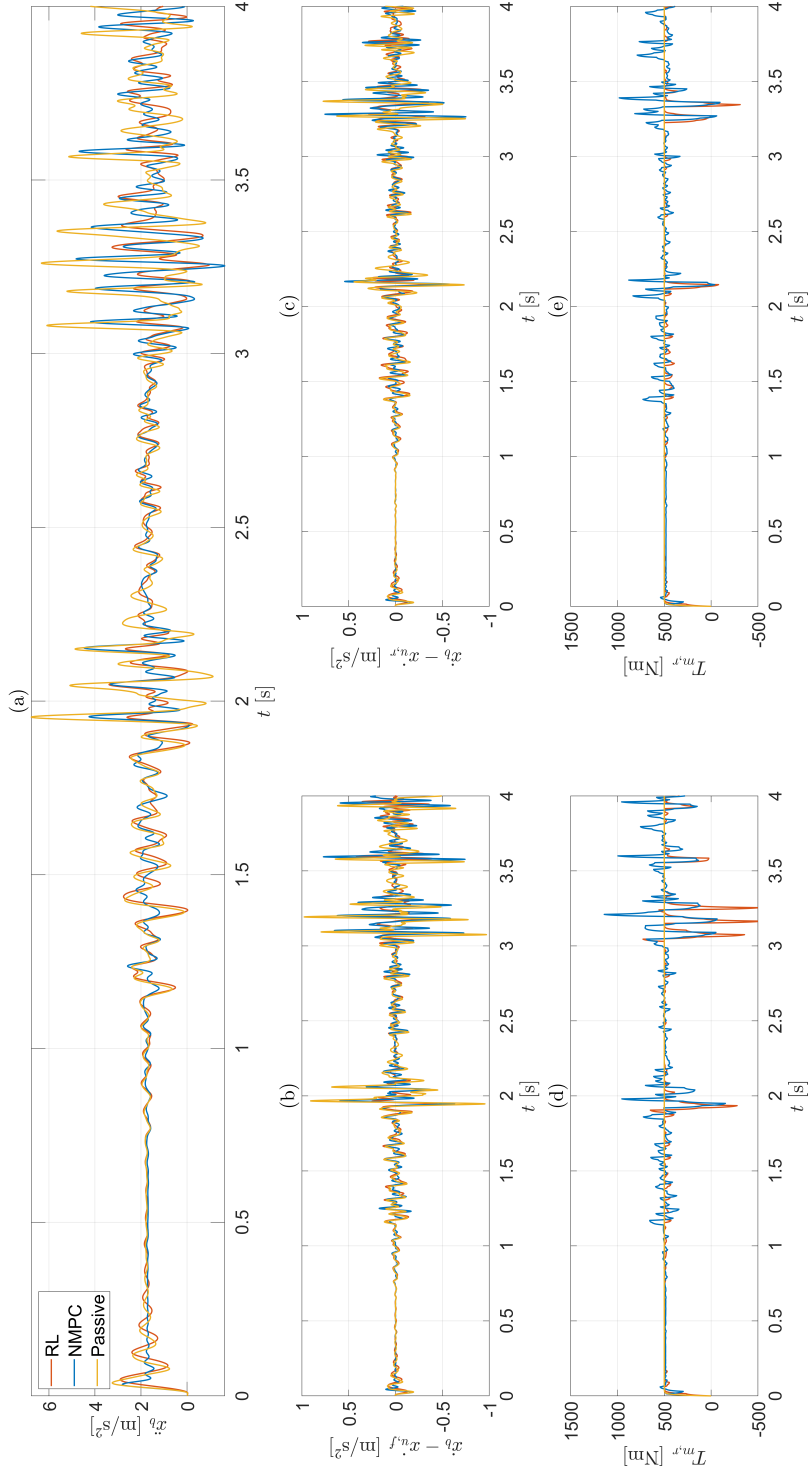


Figure 5.5: Scenario su strada reale con richiesta di coppia di 2000Nm. Si nota l'azione prevalentemente decelerante applicata dall'RL.

L'utilizzo di entrambi i termini sopra citati nelle osservazioni dell'agente evita in questo caso che il controllore interpreti l'accelerazione richiesta dal conducente come un errore da minimizzare. Se si fosse utilizzata solamente l'accelerazione $\ddot{x}_{b,fl/r}$ l'azione di controllo risultante sarebbe stata una coppia frenante, sino al completo arresto del veicolo (chiaramente da evitare). Nei risultati mostrati in Figura 5.5, si può notare che l'RL applica principalmente un'azione decelerante, mantenendo la coppia totale il più vicino possibile alla richiesta. Inoltre, l'agente è riuscito a smorzare maggiormente le vibrazioni nelle sezioni più dure della strada, confermato dal minore RMS mostrato nella Tabella 5.3.

Table 5.3: KPIs per gli scenari di test con coppia richiesta di 2000Nm

	$RMS_{\ddot{x}_b}$	$VDV_{\ddot{x}_b}$	$RMS_{\ddot{\ddot{x}}_b}$	$\Delta\ddot{x}_{b,max}$	$IACA_F$	$IACA_R$
NMPC	0.69	1.57	94.87	3.34	43.0	33.3
RL Agent	0.63	1.32	60.79	2.78	22.2	11.8
Passivo	0.99	2.27	103.34	5.08	0	0

Nei successivi test, i due controllori sono stati simulati su una serie di profili diversi, variando le dimensioni dei dossi e degli step (per il dosso è stato mantenuto lo stesso fattore di forma $\frac{h}{b}$) e utilizzando diverse sezioni del profilo stradale campionato sperimentalmente. Nelle Tabelle 5.4, 5.5, 5.6 sono presentati i valori per ogni test, confrontati con il caso passivo usando come termine di paragone $RMS_{\ddot{x}_b}$.

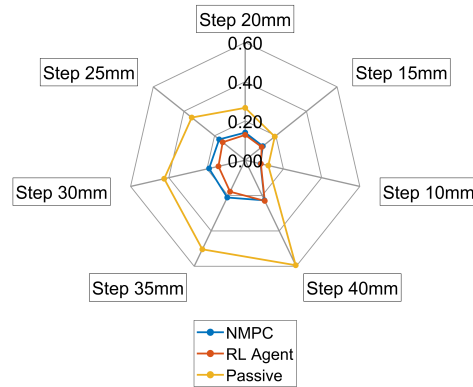


Figure 5.6: Grafico a ragnatela per il test a gradino

5. RISULTATI E TEST DI ROBUSTEZZA

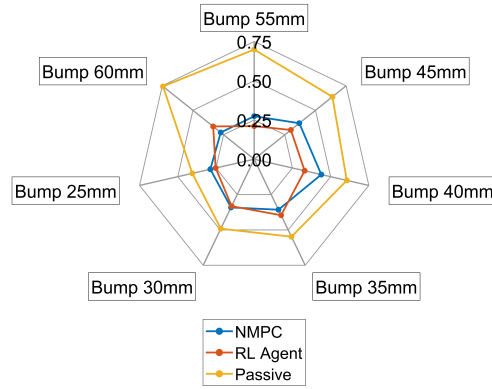


Figure 5.7: Grafico a ragnatela per il test del dosso



Figure 5.8: Grafico a ragnatela per il test su strada

Table 5.4: KPI ed errori percentuali per gli scenari a gradino

Altezza Step	10mm	15mm	20mm	25mm	30mm	35mm	40mm	Media
NMPC	0.08	0.12	0.14	0.17	0.19	0.21	0.23	
RL Agent	0.08	0.11	0.14	0.15	0.18	0.23	0.23	
Passivo	0.12	0.19	0.27	0.35	0.42	0.50	0.59	
err % RL-Passivo	32.23%	44.04%	51.69%	57.88%	66.98%	61.28%	64.68%	54.11%
err % NMPC-Passivo	47.19%	33.88%	55.42%	51.29%	66.98%	61.78%	61.28%	49.73%

Ciò che emerge da questi risultati è che l'agente RL proposto è riuscito a smorzare maggiormente le vibrazioni in due dei tre scenari considerati (gradino e dosso), ottenendo un miglioramento complessivo delle prestazioni del 9% rispetto all'NMPC, pur mantenendo prestazioni simili nel test su strada. In termini di $RMS_{\ddot{x}_b}$, rispetto al caso passivo, i risultati mostrano un miglioramento del 54% e del 46% rispettivamente nei test su strada e dosso.

Table 5.5: KPI ed errori percentuali per gli scenari del dosso

Altezza Dosso	25mm	30mm	35mm	40mm	45mm	55mm	60mm	Media
NMPC	0.29	0.34	0.36	0.37	0.44	0.27	0.27	
RL Agent	0.25	0.33	0.40	0.30	0.33	0.21	0.34	
Passivo	0.41	0.49	0.55	0.61	0.64	0.70	0.74	
err % RL-Passivo	37.78%	32.38%	27.55%	53.13%	45.54%	69.81%	54.97%	45.88%
err % NMPC-Passivo	29.14%	30.55%	34.85%	42.32%	27.56%	60.80%	63.31%	41.22%

Table 5.6: KPI ed errori percentuali per i profili stradali

Profili Stradali	Strada 1	Strada 2	Strada 3	Strada 4	Strada 5	Media
NMPC	0.27	0.31	0.33	0.39	0.29	
RL Agent	0.31	0.36	0.31	0.36	0.26	
Passivo	0.43	0.47	0.42	0.52	0.38	
err % RL-Passivo	29.72%	22.27%	26.56%	30.77%	31.51%	28.17%
err % NMPC-Passivo	38.94%	33.62%	20.10%	25.19%	23.44%	28.26%

Complessivamente, il controllore ha dimostrato una robustezza soddisfacente in tutti e tre gli scenari. Questo risultato è ulteriormente valorizzato dal ridotto sforzo computazionale richiesto dal controllore proposto. Un altro aspetto fondamentale nella valutazione della robustezza è assicurarsi che il controllore sia in grado di generalizzare efficacemente non solo in condizioni stradali diverse, ma anche su differenti veicoli. Per valutare questa capacità, è stato condotto un test di robustezza variando alcuni parametri chiave del modello veicolare, mantenendo invariato il modello interno utilizzato da entrambi i controllori. In questo modo, il controllore è rimasto "standard" e le modifiche sono state applicate solo ai parametri del veicolo. I parametri selezionati per il test sono stati: la massa sospesa m_b , la massa non sospesa m_u , la rigidità longitudinale dei silent block k_x e la rigidità radiale e tangenziale degli pneumatici k_t . Il profilo stradale utilizzato per questa analisi è stato lo *scenario di valutazione*, che ha permesso di ottenere il miglior controllore dopo la fase di addestramento. Come nei test precedenti, sia l'NMPC che l'agente RL hanno dimostrato una buona robustezza rispetto alle variazioni dei parametri. Le prestazioni dell'agente RL sono risultate leggermente inferiori rispetto all'NMPC (del 5%), ma hanno comunque ottenuto un miglioramento significativo del 45% – 50% rispetto al caso passivo. Dai dati riportati nelle Tabelle 5.7, 5.8, 5.9 e 5.10, si può osservare che, per i parametri k_t , k_x e m_u , un aumento dei valori non rappresenta un problema, mentre una loro diminuzione porta a prestazioni peggiori. Questo indica che una variazione di -20% potrebbe rappresentare il limite inferiore per questo setup di addestramento. Per quanto riguarda m_b , l'agente RL ha raggiunto prestazioni comparabili all'NMPC in caso di diminuzione della massa sospesa. Tuttavia, ha mostrato una minore

5. RISULTATI E TEST DI ROBUSTEZZA

robustezza in caso di aumento di m_b , con una riduzione delle prestazioni rispetto alla controparte NMPC (tuttavia va notato come ci sia comunque un guadagno pari al 35% rispetto al caso passivo). Migliori performance potrebbero essere ottenute introducendo parametri veicolari randomizzati durante ogni episodio di addestramento.

Table 5.7: KPI ed errori percentuali per variazioni a k_t

Modifiche al k_t	Nominale	$k_t + 15\%$	$k_t - 15\%$	$k_t + 20\%$	$k_t - 20\%$	Media
KPI_NMPC	0.20	0.18	0.25	0.18	0.26	
KPI_RL	0.19	0.20	0.26	0.17	0.36	
KPI_Passivo	0.44	0.44	0.44	0.44	0.44	
err % RL-Passivo	56.09%	54.48%	40.46%	60.23%	18.39%	45.93%
err % NMPC-Passivo	54.94%	58.39%	42.07%	59.31%	39.54%	50.85%

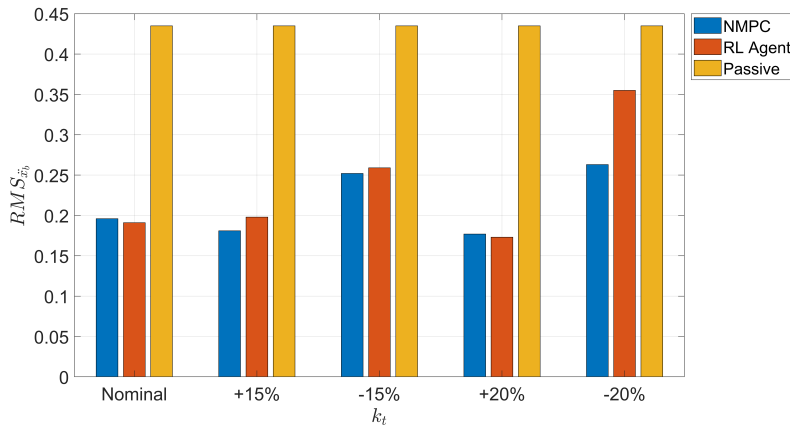
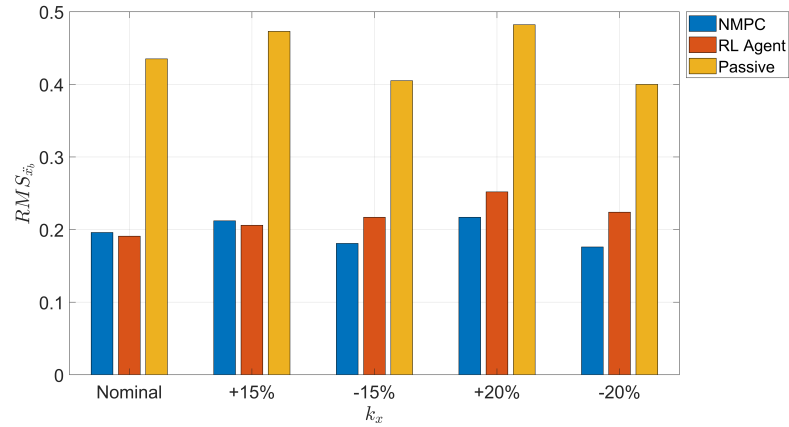


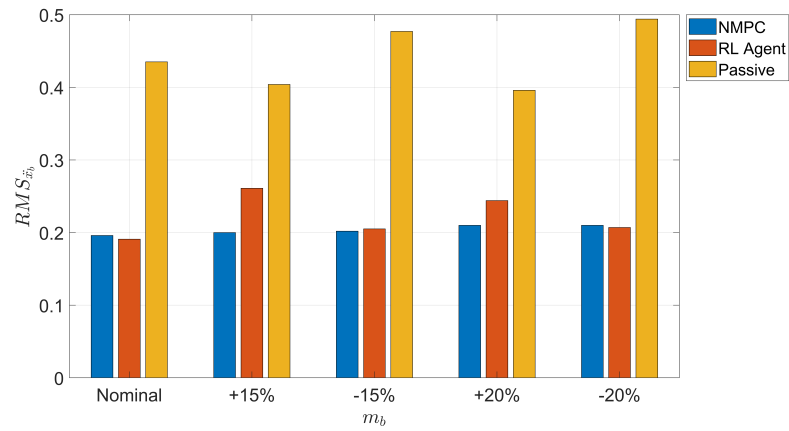
Figure 5.9: Grafico a barre per il test su k_t , si può vedere che entrambi i controllori hanno valori vicini, tranne che per il test "-20%" in cui l'RL mostra una differenza di $0.1m/s^2$, comunque accettabile.

Table 5.8: KPI ed errori percentuali per variazioni a k_x

Modifiche al k_x	Nominale	$k_x + 15\%$	$k_x - 15\%$	$k_x + 20\%$	$k_x - 20\%$	Media
KPI_NMPC	0.20	0.21	0.18	0.22	0.18	
KPI_RL	0.19	0.21	0.22	0.25	0.22	
KPI_Passivo	0.44	0.47	0.41	0.48	0.40	
err % RL-Passivo	56.09%	56.45%	46.42%	47.72%	44.00%	50.14%
err % NMPC-Passivo	54.94%	55.18%	55.31%	54.98%	56.00%	55.28%

Figure 5.10: Grafico a barre per il test su k_x Table 5.9: KPI ed errori percentuali per variazioni della massa del veicolo m_b

Modifiche al m_b	Nominale	$m_b + 15\%$	$m_b - 15\%$	$m_b + 20\%$	$m_b - 20\%$	Media
KPI_NMPC	0.20	0.20	0.20	0.21	0.21	
KPI_RL	0.19	0.26	0.21	0.24	0.21	
KPI_Passivo	0.44	0.40	0.48	0.40	0.49	
err % RL-Passivo	56.09%	35.40%	57.02%	38.38%	58.10%	49.00%
err % NMPC-Passivo	54.94%	50.50%	57.65%	46.97%	57.49%	53.51%

Figure 5.11: Grafico a barre per il test su m_b

5.2.2 Analisi in Frequenza

Dai risultati mostrati in Figura 5.3, si osserva un picco negativo nel grafico dell'accelerazione (a), tra i due assi del veicolo, oltre a vibrazioni significative nell'errore di velocità relativa all'asse anteriore (b). Per identificare la causa di questa risonanza, è stata effettuata una linearizzazione del sistema seguita da

5. RISULTATI E TEST DI ROBUSTEZZA

Table 5.10: KPI ed errori percentuali per variazioni della massa non sospesa m_u

Modifiche al m_u	Nominale	$m_u + 15\%$	$m_u - 15\%$	$m_u + 20\%$	$m_u - 20\%$	Media
KPI_NMPC	0.20	0.20	0.20	0.20	0.24	
KPI_RL	0.19	0.20	0.20	0.20	0.30	
KPI_Passivo	0.44	0.44	0.42	0.44	0.41	
err % RL-Passivo	56.09%	55.35%	50.96%	54.55%	26.04%	48.60%
err % NMPC-Passivo	54.94%	53.76%	52.16%	53.86%	41.03%	51.15%

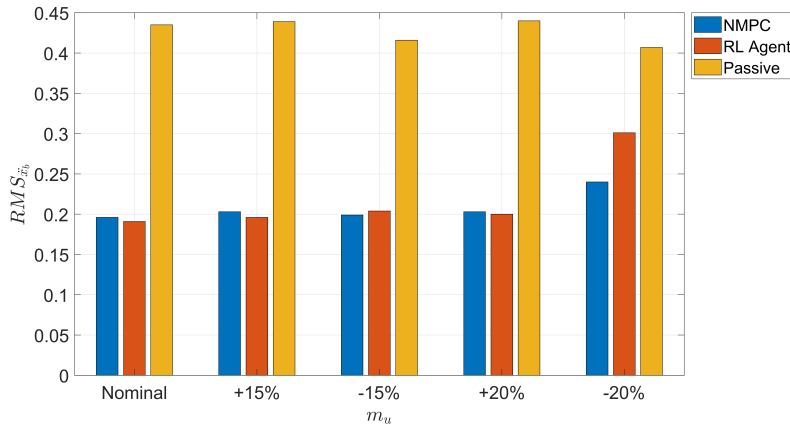


Figure 5.12: Grafico a barre per il test su m_u

un'analisi in frequenza dell'accelerazione della massa sospesa, nello scenario selezionato (Dosso altezza 55mm). La linearizzazione è stata realizzata utilizzando il toolbox *Linearization Manager* di Simulink, che consente di estrarre modelli lineari da sistemi non lineari in corrispondenza di specifici punti di funzionamento o istanti di simulazione. In questo caso, è stato utilizzato uno scenario con fondo stradale piano e la linearizzazione è stata effettuata al tempo $t = 1s$, affinché il sistema raggiungesse una condizione stazionaria. Il *Linearization Manager* offre grande flessibilità nella definizione delle coppie ingresso-uscita, nella scelta dei punti operativi e nella selezione dei tipi di analisi (ad esempio, diagrammi di Bode o mappe poli-zero). Per questa analisi, l'ingresso era la posizione della massa non sospesa $x_{u,fl/r}$, mentre l'uscita era l'accelerazione e la posizione della massa sospesa x_b . Il diagramma di Bode risultante per $\frac{x_{u,fl/r}}{x_b}$ ha evidenziato un modo longitudinale attorno a 14Hz all'interno della fascia di frequenze tipicamente associate al comfort (0 – 20Hz). Questi risultati sono in linea con ricerche precedenti, come quelle condotte da Yamada *et al.* [3], e sono illustrati in Figura 5.13.

Successivamente alla linearizzazione, è stata calcolata la densità spettrale di potenza (PSD) di $\ddot{x}_b$ e $\Delta\dot{x}_{b,fl/r}$ per l'asse anteriore nella zona post-dosso al fine di identificare lo spettro delle frequenze responsabili di tale risonanza.

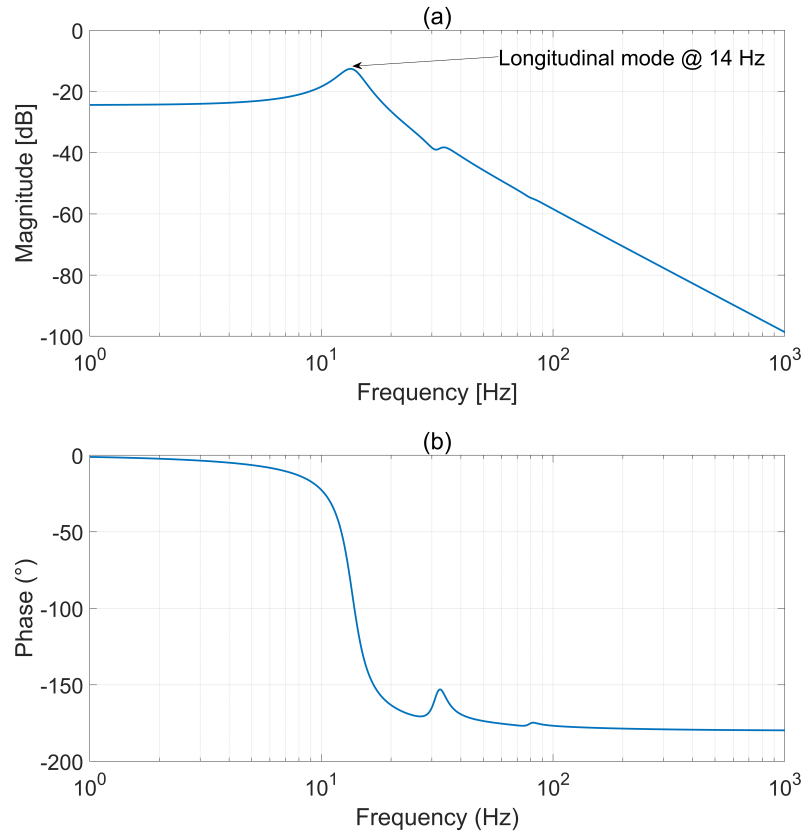


Figure 5.13: Diagramma di Bode per la risposta in frequenza di $\frac{x_{u,fl/r}}{x_b}$. Si evidenzia il modo longitudinale a 14Hz, associato al movimento relativo tra massa sospesa e non sospesa.

Come mostrato in Figura 5.14, l'agente RL è riuscito a smorzare il picco causato dal modo longitudinale a 14Hz, eccitando però un modo attorno a 27Hz responsabile delle oscillazioni nella velocità relativa (e di conseguenza nell'accelerazione). Questo risultato è comunque positivo, in quanto, come citato precedentemente, l'intervallo di frequenze tipicamente associato al comfort di guida è compreso tra 0 e 20Hz. Inoltre, il modo a 27Hz ha effetti trascurabili sull'accelerazione complessiva, confermando che ha una minima influenza sul comfort generale di guida. Complessivamente quindi, nonostante l'assenza di un termine relativo alle frequenze di eccitazione della cassa all'interno della Reward Function, il controllore ricavato è stato in grado di agire correttamente, minimizzando la fascia di frequenze associate al comfort. Il termine di risonanza evidenziato nei precedenti grafici di velocità ed accelerazione rimane dunque ininfluenza nella valutazione complessiva del comfort di guida, su quello specifico scenario.

5. RISULTATI E TEST DI ROBUSTEZZA

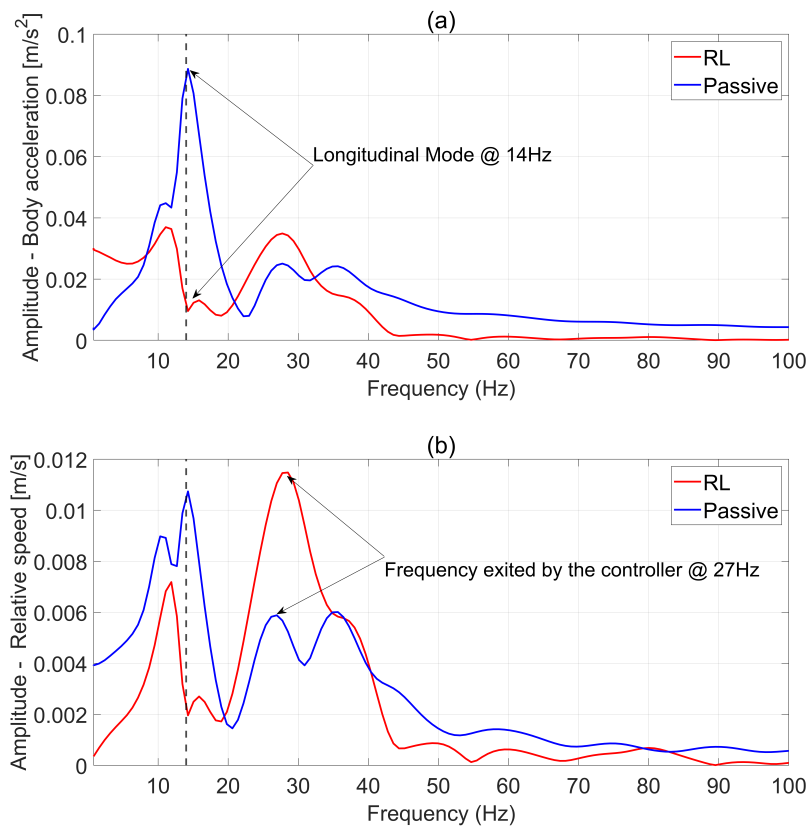


Figure 5.14: FFT dell'accelerazione longitudinale (a) e della velocità relativa misurata sull'asse anteriore (b). Nonostante l'incremento del modo attorno a 27Hz, il controllore è riuscito a smorzare il primo modo longitudinale a 14Hz, migliorando il comfort di guida. Gli effetti del secondo picco sull'accelerazione risultano trascurabili.

6 IMPLEMENTAZIONE SU VEICOLO REALE

6.1 Test su sistema dSPACE

Per la valutazione finale, il controllore è stato implementato su un sistema dSPACE MicroAutoBox III (1.4 GHz quad-core), così come la controparte NMPC, al fine di verificare lo sforzo computazionale richiesto per l'esecuzione e le differenze in termini di KPI rispetto al modello Simulink. Il dispositivo presenta attualmente lo stato dell'arte per validare controllori implementabili su centraline commerciali, seppur possedendo una capacità computazionale di gran lunga superiore alla media delle unità adottate su veicoli di serie. Il passaggio ad un sistema dSPACE ha richiesto particolare attenzione poichè si è dovuto correggere il codice in modo che potesse essere implementato in linguaggio C tramite il Code Builder di Simulink. Un esempio è stato l'integratore utilizzato nel blocco preview dell'agente (blocco in rosa mostrato nella Figura 3.8 nel Capitolo 3. Nel passaggio in C code si è dovuto sostituire l'integrazione delle equazioni del moto tramite Ode45 in un'integrazione esplicita tramite Runge-Kutta di 4° ordine, le cui equazioni sono schematizzate di seguito:

Consideriamo il problema ai valori iniziali:

$$\dot{y} = f(t, y), \quad y(t_0) = y_0 \quad (6.1)$$

dove t_0 e y_0 sono noti. Per un intervallo sufficientemente piccolo $h > 0$, possiamo definire:

$$y_{n+1} = y_n + \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4) \quad (6.2)$$

$$t_{n+1} = t_n + h \quad (6.3)$$

dove:

$$k_1 = f(t_n, y_n) \quad (6.4)$$

$$k_2 = f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right) \quad (6.5)$$

$$k_3 = f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right) \quad (6.6)$$

$$k_4 = f(t_n + h, y_n + hk_3) \quad (6.7)$$

In questo modo, $y(t_{n+1})$ viene approssimato da y_{n+1} , calcolato come una media pesata di quattro incrementi k_1 , k_2 , k_3 e k_4 . Ogni k rappresenta un'inclinazione della funzione f , calcolata in diversi punti dell'intervallo. Il

6. IMPLEMENTAZIONE SU VEICOLO REALE

tempo di discretizzazione del modello, così come il passo temporale dei controllori, è stato mantenuto lo stesso del modello ($1ms$ e $15ms$ rispettivamente).

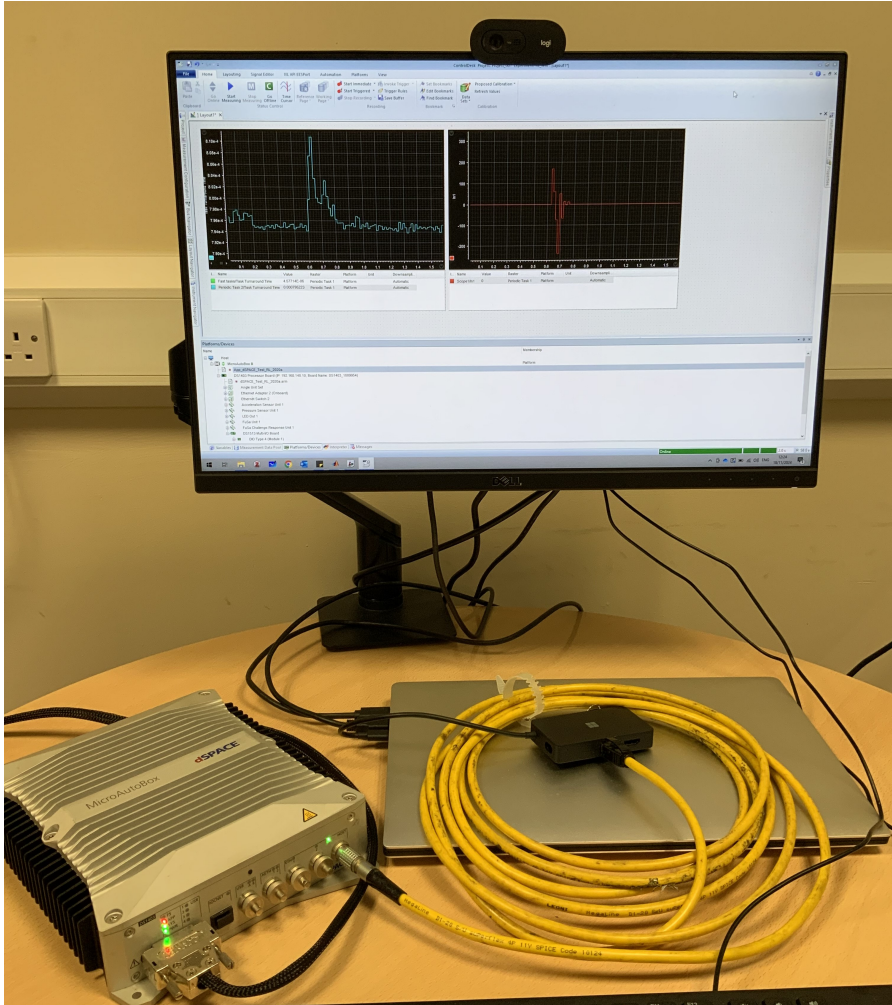


Figure 6.1: Unità dSPACE MicroAutoBox III, con l'implementazione in tempo reale del RL (si può notare l'azione di controllo sull'asse anteriore nel grafico di destra, mentre sul grafico a sinistra vi è mostrato l'andamento del turnaround time)

I risultati (Figura 6.2), ricavati su uno scenario a gradino singolo altezza $20mm$ hanno mostrato un tempo di elaborazione $< 1ms$ per l'agente RL, mentre il tempo del NMPC è stato di $\sim 6ms$, confermando la maggiore efficienza computazionale del metodo proposto.

Osservando i KPI, entrambi i metodi sono riusciti a smorzare efficacemente le vibrazioni, con una riduzione del 32% e del 42% rispettivamente rispetto al caso passivo. Va notato che risultati migliori possono essere ottenuti riducendo

il passo temporale dell'agente RL durante la fase di addestramento. Infatti, il tempo di elaborazione di $1ms$ può essere visto come il limite inferiore per il tempo di campionamento del controllore (per questo dispositivo selezionato), quindi un addestramento utilizzando lo stesso passo temporale ridurrebbe ulteriormente l'RMS dell'accelerazione longitudinale, aumentando però il tempo richiesto per la fase di addestramento.

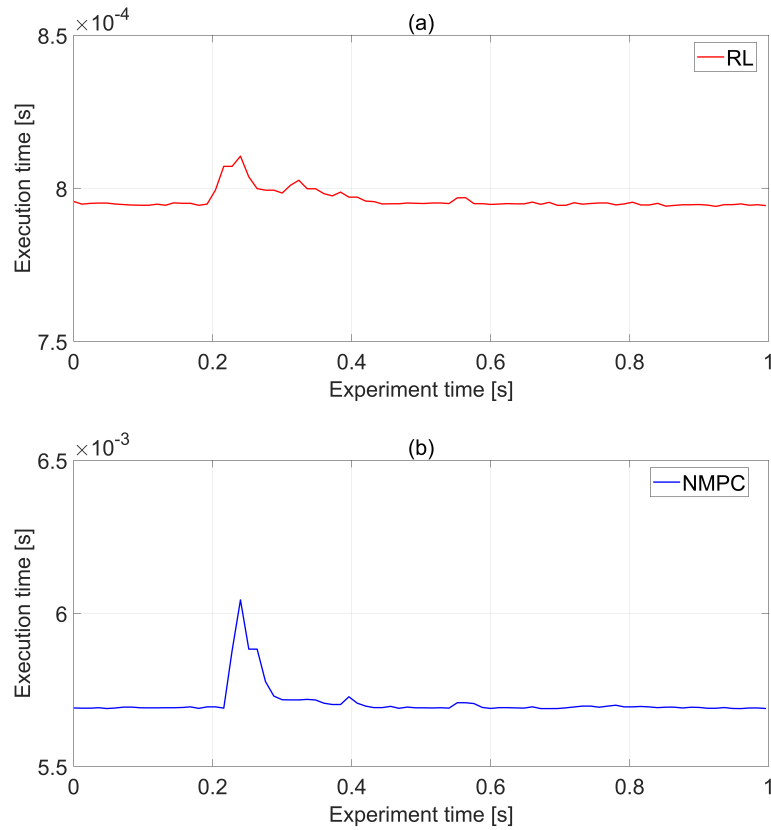


Figure 6.2: Confronto tempi computazionali dei due controllori presentati, si nota il picco di sforzo computazionale richiesto nella zona dell'ostacolo.

7 CONCLUSIONI

In questa tesi, è stato affrontato il problema dell'ottimizzazione del comfort di veicolo attraverso l'implementazione di tecniche avanzate di controllo basate sul Machine Learning, con un focus particolare sul Reinforcement Learning (RL). L'obiettivo principale è stato quello di ridurre le vibrazioni longitudinali percepite all'interno del veicolo, un fattore chiave per migliorare il comfort di guida, in particolare nei veicoli elettrici con architettura di motori In-Wheel. Dopo aver esplorato un approccio di controllo tradizionale come il Nonlinear Model Predictive Control (NMPC) utilizzato come punto di partenza per l'attività e successivo confronto, è stato sviluppato un controllore basato su RL con l'obiettivo di gestire le irregolarità stradali in tempo reale, assicurando le medesime performance in termini di smorzamento di vibrazioni ed al contempo un'ottima generalizzazione rispetto a diversi scenari o parametri di veicolo. Attraverso l'addestramento del modello in diversi scenari stradali, inclusi step e bump, e grazie all'integrazione di tecniche di preview, il controllore ha dimostrato performance e robustezza superiore rispetto agli approcci convenzionali. I test di validazione hanno confermato che il controllore RL è in grado di ridurre significativamente le accelerazioni longitudinali, migliorandone il comfort di guida. In particolare, i risultati ottenuti mostrano una riduzione oltre il 50% delle vibrazioni rispetto al caso passivo (quindi senza azione correttiva). Successivi test di robustezza hanno poi testato le capacità dell'agente di generalizzare l'azione di controllo in risposta a scenari (e parametri) non utilizzati nella fase di addestramento. I risultati ottenuti hanno mostrato un'ottimo comportamento dei due controllori, con l'agente RL che presenta un guadagno di performance di $\simeq 9\%$ su diversi scenari stradali, mentre una perdita di performance contenuta ($\simeq 5\%$) rispetto alla controparte NMPC nei test su diversi setup di parametri nel veicolo. Infine si è mostrato l'implementazione del controllore su un sistema dSPACE MicroAutoBox III come validazione finale che tale controllore può essere effettivamente impiegato su centraline di veicoli prodotti in serie, dato il tempo computazionale estremamente basso che tale approccio richiede. In conclusione, il lavoro svolto ha dimostrato l'efficacia dell'RL nel contesto del controllo veicolare, aprendo nuove prospettive per l'utilizzo di tecniche di intelligenza artificiale nel miglioramento del comfort di guida. Ulteriori sviluppi futuri potrebbero concentrarsi sull'ottimizzazione dell'addestramento del modello in scenari più complessi e sull'integrazione di altre variabili che influenzano il comfort, come le vibrazioni verticali e laterali.

BIBLIOGRAPHY

- [1] C. Poussot-Vassal, C. Spelta, O. Sename, S. Savaresi, and L. Dugard, "Survey and performance evaluation on some automotive semi-active suspension control methods: A comparative study on a single-corner model," *Annual Reviews in Control*, vol. 36, no. 1, pp. 148–160, 2012.
- [2] S. Yamada, T. Beauduin, H. Fujimoto, T. Kanou, and E. Katsuyama, "Model-based longitudinal vibration suppression control for electric vehicles with geared in-wheel motors," pp. 517–522, 2017.
- [3] S. Yamada, T. Beauduin, H. Fujimoto, T. Kanou, and E. Katsuyama, "Active model-based suppression of secondary ride for electric vehicles with in-wheel motors," *IEEE/ASME Transactions on Mechatronics*, vol. 27, pp. 5637–5646, 12 2022.
- [4] A. Scamarcio, P. Gruber, S. De Pinto, and A. Sorniotti, "Anti-jerk controllers for automotive applications: A review," *Annual Reviews in Control*, vol. 50, pp. 174–189, 2020.
- [5] M. Liu, F. Gu, and Y. Zhang, "Ride comfort optimization of in-wheel-motor electric vehicles with in-wheel vibration absorbers," *Energies*, vol. 10, 10 2017.
- [6] K. Deepak, M. A. Frikha, Y. Benômar, M. E. Baghdadi, and O. Hegazy, "In-wheel motor drive systems for electric vehicles: State of the art, challenges, and future trends," 4 2023.
- [7] S. Nie, Y. Zhuang, F. Chen, Y. Wang, and S. Liu, "A method to eliminate unsprung adverse effect of in-wheel motor-driven vehicles," *Journal of Low Frequency Noise Vibration and Active Control*, vol. 37, pp. 955–976, 12 2018.
- [8] F. Walz and S. Hohmann, "Model predictive longitudinal motion control for low velocities on known road profiles," *Vehicle System Dynamics*, vol. 58, pp. 1310–1328, 8 2020.
- [9] F. Walz and S. Hohmann, "On model-based longitudinal feed-forward motion control for low velocities on known road profiles," *Vehicle System Dynamics*, vol. 57, pp. 1126–1142, 8 2019.
- [10] V. Vidal, P. Stano, G. Tavolo, M. Dhaens, D. Tavernini, P. Gruber, and A. Sorniotti, "On pre-emptive in-wheel motor control for reducing the longitudinal acceleration oscillations caused by road irregularities," *IEEE Transactions on Vehicular Technology*, vol. 71, pp. 9322–9337, 9 2022.
- [11] G. Bai, Y. Meng, L. Liu, W. Luo, Q. Gu, and L. Liu, "Review and comparison of path tracking based on model predictive control," 10 2019.

BIBLIOGRAPHY

- [12] M. Rokonzaman, N. Mohajer, S. Nahavandi, and S. Mohamed, "Model predictive control with learned vehicle dynamics for autonomous vehicle path tracking," *IEEE Access*, vol. 9, pp. 128233–128249, 2021.
- [13] N. P. Farazi, B. Zou, T. Ahamed, and L. Barua, "Deep reinforcement learning in transportation research: A review," *Transportation research interdisciplinary perspectives*, vol. 11, p. 100425, 2021.
- [14] V. K. T. Mantripragada and R. K. Kumar, "Deep reinforcement learning-based antilock braking algorithm," *Vehicle system dynamics*, vol. 61, no. 5, pp. 1410–1431, 2023.
- [15] Y. Ye, X. Zhang, and J. Sun, "Automated vehicle's behavior decision making using deep reinforcement learning and high-fidelity simulation environment," *Transportation Research Part C: Emerging Technologies*, vol. 107, pp. 155–170, 10 2019.
- [16] Y. Du, J. Chen, C. Zhao, F. Liao, and M. Zhu, "A hierarchical framework for improving ride comfort of autonomous vehicles via deep reinforcement learning with external knowledge," *Computer-Aided Civil and Infrastructure Engineering*, vol. 38, pp. 1059–1078, 5 2023.
- [17] T. Roulet, M. Bach, M. Hauß, and M. Skull, "Ai-based methodology for the calibration of anti-jerk control on electric drives," *ATZ worldwide*, vol. 126, no. 5, pp. 16–21, 2024.
- [18] H. Wei, N. Zhang, J. Liang, Q. Ai, W. Zhao, T. Huang, and Y. Zhang, "Deep reinforcement learning based direct torque control strategy for distributed drive electric vehicles considering active safety and energy saving performance," *Energy*, vol. 238, 1 2022.
- [19] A. Wasala, D. Byrne, P. Miesbauer, J. O'Hanlon, P. Heraty, and P. Barry, "Trajectory based lateral control: A reinforcement learning case study," *Engineering Applications of Artificial Intelligence*, vol. 94, 9 2020.
- [20] P. Cai, X. Mei, L. Tai, Y. Sun, and M. Liu, "High-speed autonomous drifting with deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 5, pp. 1247–1254, 4 2020.
- [21] Y. Du, J. Chen, C. Zhao, C. Liu, F. Liao, and C. Y. Chan, "Comfortable and energy-efficient speed control of autonomous vehicles on rough pavements using deep reinforcement learning," *Transportation Research Part C: Emerging Technologies*, vol. 134, 1 2022.
- [22] X. Hou, M. Gan, J. Zhang, S. Zhao, and Y. Ji, "Vehicle ride comfort optimization in the post-braking phase using residual reinforcement learning," *Advanced Engineering Informatics*, vol. 58, 10 2023.

- [23] A. Schmeitz, I. Besselink, and S. Jansen, "Tno mf-swift," *Vehicle System Dynamics*, vol. 45, no. S1, pp. 121–137, 2007.
- [24] H. Pacejka, *Tire and vehicle dynamics*. Elsevier, 2005.
- [25] L. Grüne, J. Pannek, L. Grüne, and J. Pannek, *Nonlinear model predictive control*. Springer, 2017.
- [26] A. Schmeitz, S. Jansen, H. Pacejka, J. Davis, N. Kota, C. Liang, and G. Lodewijks, "Application of a semi-empirical dynamic tyre model for rolling over arbitrary road profiles," *International Journal of Vehicle Design*, vol. 36, no. 2-3, pp. 194–215, 2004.
- [27] M. Morales, "Grokking deep reinforcement learning," 2020.
- [28] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book, 2018.
- [29] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [30] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.
- [31] *MATLAB and Reinforcement Learning Toolbox Release 2023b*. Natick, Massachusetts, United States: The MathWorks, Inc.
- [32] M. Behzad, M. Hodaiei, and I. Alimohammadi, "Experimental and numerical investigation of the effect of a speed bump on car noise emission level," *Applied Acoustics*, vol. 68, no. 11-12, pp. 1346–1356, 2007.
- [33] International Organization for Standardization, "ISO 8608:2016 - Mechanical vibration — Road surface profiles — Reporting of measured data," 2016.
- [34] T. Zhang, X. Wang, B. Liang, and B. Yuan, "Catastrophic interference in reinforcement learning: A solution based on context division and knowledge distillation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 12, pp. 9925–9939, 2022.

