

POLITECNICO DI TORINO

Laurea Magistrale in Ingegneria Meccanica

Tesi di Laurea Magistrale



**Politecnico
di Torino**

Sviluppo di un controllore Model Predictive Control per un sistema Energy Efficient Adaptive Cruise Control

Relatori:

Aldo Sorniotti

Luca DiMauro

Fabio Alberti

Candidato:

Luca Toson

Marzo 2025

Indice

Indice delle figure.....	5
Indice delle tabelle	7
Sommario	9
1. Introduzione	11
1.1 Livelli di guida autonoma	11
1.2 Sistemi ADAS	12
1.3 Dispositivi utilizzati	13
1.4 Adaptive Cruise Control.....	15
1.4.1 Tecniche di controllo	17
2. Energy Efficient Adaptive Cruise Control.....	19
2.1 Model Predictive Control	20
2.2 Formulazione del problema di ottimizzazione	20
2.2.1 Funzione di costo	20
2.2.2 Receding Control Horizon	21
2.2.3 Vincoli	22
2.3 Tuning an MPC controller.....	23
2.4 Implicit and explicit MPC	24
2.5 Optimization algorithms, solvers and toolboxes for MPC	24
2.6 Vantaggi e svantaggi dell'MPC.....	24
2.7 Plant model.....	25
2.7.1 Longitudinal dynamics.....	25
2.7.2 Powertrain model	28
2.8 Internal Model	31
2.8.1 CasADi e acados	31
2.8.2 Optimal Control Problem	32
3. Modello Simulink	60
4. Algoritmo di ottimizzazione dei pesi	69
5. Test e risultati	73
5.1 Analisi di sensitività sul ciclo WLTP	74
5.2 Scenari di test	82
5.2.1 Risultati sul primo scenario.....	84
5.2.2 Risultati sul secondo scenario	89
5.2.3 Risultati sul terzo scenario	92

6. Conclusione.....	96
Ringraziamenti	98
Bibliografia.....	99

Indice delle figure

Figura 1.1: Livelli di automazione SAE	12
Figura 1.2: Panoramica dei sensori	13
Figura 1.3: Radar anticollisione	14
Figura 1.4: Immagine 3D fornita dal lidar	15
Figura 1.5: Scenario con due veicoli	16
Figura 1.6: Componenti di un sistema ACC	16
Figura 1.7: Controller a un livello	17
Figura 1.8: Controller gerarchico	18
Figura 2.1: MPC control loop	20
Figura 2.2: Control and prediction horizon in MPC	22
Figura 2.3: Componenti della forza gravitazionale agenti su un veicolo che percorre una pendenza	27
Figura 2.4: Mappa caratteristica di coppia e potenza per un motore elettrico [27]	28
Figura 2.5: Gearbox efficiency map	30
Figura 2.6: Electric motor efficiency map	31
Figura 2.7: A sinistra la mappa di efficienza del motore elettrico ottenuta con i dati sperimentali; a destra la mappa di efficienza del motore elettrico ottenuta tramite la rete neurale	47
Figura 2.8: A sinistra la mappa di efficienza del motore elettrico ottenuta con i dati sperimentali; a destra la mappa di efficienza del motore elettrico ottenuta tramite la funzione polinomiale	48
Figura 3.1: Panoramica sul modello Simulink	60
Figura 3.2: Blocco EACC contenente il controllore	61
Figura 3.3: Blocco in cui vengono assegnati i vincoli (Constraints)	62
Figura 3.4: Valori di riferimento per i termini caratterizzanti la funzione di costo	64
Figura 3.5: Caratterizzazione del blocco Plant	65
Figura 3.6: Equazioni che caratterizzano il Vehicle Plant	66
Figura 3.7: Preceding Vehicle block	66
Figura 3.8: Energy consumption block	68
Figura 5.1: Performance indicators relativi alla variazione del termine di potenza del motore elettrico	76
Figura 5.2: Performance indicators relativi alla variazione del termine inerente l'efficienza globale	78
Figura 5.3: Performance indicators relativi alla variazione del termine inerente le perdite di potenza lungo la powertrain	79
Figura 5.4: Performance indicators relativi alla variazione del termine inerente l'energia spesa per unità di distanza	81
Figura 5.5: Primo scenario di test	82
Figura 5.6: Secondo scenario di test	83
Figura 5.7: Ciclo guida WLTP	84
Figura 5.8: Profili di velocità nel test di speed tracking e nel caso di priorità data all'efficientamento	86
Figura 5.9: Andamenti della distanza relativa nel test di speed tracking e nel caso di priorità data all'efficientamento	87

Figura 5.10: Andamenti dell'energia consumata nel test di speed tracking e nel caso di priorità data all'efficientamento	88
Figura 5.11: Profili di coppia nel test di speed tracking e nel caso di priorità data all'efficientamento.....	88
Figura 5.12: Profili di velocità nel test di speed tracking e nel caso di priorità data all'efficientamento.....	90
Figura 5.13: Andamenti dell'energia consumata nel test di speed tracking e nel caso di priorità data all'efficientamento	91
Figura 5.14: Andamenti della distanza relativa nel test di speed tracking e nel caso di priorità data all'efficientamento	91
Figura 5.15: Profili di coppia nel test di speed tracking e nel caso di priorità data all'efficientamento.....	92
Figura 5.16: Profili di velocità nel test di speed tracking e nel caso di priorità data all'efficientamento.....	94
Figura 5.17: Andamenti dell'energia consumata nel test di speed tracking e nel caso di priorità data all'efficientamento	94
Figura 5.18: Andamenti della distanza relativa nel test di speed tracking e nel caso di priorità data all'efficientamento	95
Figura 5.19: Profili di coppia nel test di speed tracking e nel caso di priorità data all'efficientamento.....	95

Indice delle tabelle

Tabella 2.1: Parametri impostati nell'MPC	24
Tabella 2.2: Variabili che definiscono l'equazione per la massa equivalente	26
Tabella 2.3: Variabili che definiscono la forza di resistenza al rotolamento	26
Tabella 2.4: Variabili che definiscono la forza di resistenza aerodinamica.....	27
Tabella 2.5: Impostazioni del solver.....	32
Tabella 2.6: Definizione degli stati, variabile di controllo, parametri esterni e variabili non lineari.....	33
Tabella 2.7: Variabili simboliche CasADi	34
Tabella 2.8: Struttura della rete neurale.....	37
Tabella 2.9: Variabili di normalizzazione inerenti la potenza del motore elettrico e la perdita di potenza	40
Tabella 2.10: Dati relativi al prediction horizon.....	40
Tabella 2.11: Pesi lineari per le slack variables	43
Tabella 2.12: Termini quadratici per le slack variables	44
Tabella 2.13: Porte di ingresso dell'S-function	50
Tabella 2.14: Porte di uscita dell'S-function	50
Tabella 2.15: Valori di riferimento delle variabili che compongono la funzione di costo	52
Tabella 2.16: Vincoli sugli stati, sulla variabile di controllo e sulle variabili che compongono la funzione di costo	53
Tabella 4.1: Variabili necessarie alla definizione dei performance indicators	71
Tabella 5.1: Pesi caratterizzanti la funzione di costo dell'algoritmo di ottimizzazione nel caso di Speed Tracking.....	73
Tabella 5.2: Pesi caratterizzanti la funzione di costo dell'algoritmo di ottimizzazione nel caso in cui la priorità venga data alle variabili di efficientamento.....	74
Tabella 5.3: Pesi ottimizzati nel caso di priorità data ai termini di efficientamento	75
Tabella 5.4: Analisi di tendenza dei performance indicators nel caso di variazione del termine di potenza del motore elettrico	75
Tabella 5.5: Analisi di tendenza dei performance indicators nel caso di variazione del termine di efficienza globale della powertrain	77
Tabella 5.6: Analisi di tendenza dei performance indicators nel caso di variazione del termine inerente le perdite di potenza nella powertrain	78
Tabella 5.7: Analisi di tendenza dei performance indicators nel caso di variazione del termine inerente l'energia consumata per unità di distanza	80
Tabella 5.8: Pesi dei termini costituenti la funzione di costo al fine di eseguire il test sul tracking della velocità nel primo scenario.....	84
Tabella 5.9: Performance indicators on the Tracking Speed test.....	85
Tabella 5.10: Pesi dei termini costituenti la funzione di costo al fine di eseguire il test sul primo scenario dando priorità ai termini di efficientamento energetico	85
Tabella 5.11: Performance indicators relativi al test sul primo scenario. Sono riportate le variazioni percentuali relative al confronto tra il caso in cui si dà priorità all'inseguimento della velocità e il caso in cui viene data priorità ai termini di efficientamento.....	85
Tabella 5.12: Performance indicators on the Tracking Speed test.....	89
Tabella 5.13: Performance indicators relativi al test sul secondo scenario. Sono riportate le variazioni percentuali relative al confronto tra il caso in cui si dà priorità all'inseguimento della velocità e il caso in cui viene data priorità ai termini di efficientamento.....	89

Tabella 5.14: Performance indicators on the Tracking Speed test.....	92
Tabella 5.15: Performance indicators relativi al test sul terzo scenario. Sono riportate le variazioni percentuali relative al confronto tra il caso in cui si dà priorità all'inseguimento della velocità e il caso in cui viene data priorità ai termini di efficientamento.....	93

Sommario

Attualmente c'è un incremento costante dell'interesse nei confronti delle nuove tecnologie che alimenta una ricerca verso lo sviluppo di forme di integrazione sempre più sofisticate tra uomo e macchina. In particolare, i sistemi di guida assistita, come l'Adaptive Cruise Control (ACC), rappresentano soluzioni fondamentali nel panorama automobilistico moderno. Si stima che l'errore umano rappresenti la causa del 93% degli incidenti stradali, il che ha spinto negli ultimi anni le case automobilistiche ad adottare sistemi di sicurezza alla guida sempre più complessi, inclusi sistemi di mantenimento della corsia, cruise control adattativo e sistemi di parcheggio automatico. Questi sviluppi portano alla realizzazione di veicoli di nuova concezione dotati di un numero sempre maggiore e più avanzato di sistemi di sicurezza che hanno lo scopo di ridurre l'intervento umano nella guida.

I sistemi ACC permettono di alleviare lo stress del guidatore e migliorare il comfort durante la guida contribuendo a una notevole riduzione degli incidenti stradali. Inoltre, si può ottenere dei benefici riguardanti la riduzione dei consumi di carburante e di emissioni inquinanti nell'ambiente. L'implementazione di tali tecnologie, perciò, ha un impatto positivo sull'efficienza e sulla sostenibilità della mobilità contemporanea.

Il presente lavoro di tesi si propone di esplorare in dettaglio le fasi di ricerca, sviluppo e implementazione di un ACC basato su una strategia di controllo Model Predictive Control (MPC), sfruttando il framework acados per facilitare l'ottimizzazione del processo di controllo. Dopo una fase iniziale di studio e catalogazione dei sistemi ACC esistenti, il lavoro si è concentrato sulla stesura del codice del controllore nell'ambiente Matlab. Successivamente è stato fatto ricorso a Simulink per eseguire le varie simulazioni sulla base degli scenari definiti per fare i test.

Una volta progettato l'ACC standard, l'attenzione si è spostata all'implementazione di strategie per tener conto dell'efficientamento energetico andando a creare un Energy Efficient Adaptive Cruise Control (EACC) per poter sfruttare in modo ottimale la potenza del motore elettrico, aumentare l'efficienza globale della driveline e minimizzare l'energia spesa per unità di distanza. Quindi, oltre a regolare in modo autonomo la velocità del veicolo ciò permette di minimizzare il consumo energetico.

La validazione della strategia di controllo proposta su Matlab sarà testata in ambiente Simulink. In questa fase si procederà con un confronto tra le performance dell'EACC. Si andrà ad effettuare un test in cui verrà data priorità al tracking della velocità e un test in cui la priorità verrà data ai termini di efficientamento al fine di dimostrare i benefici in termini di riduzione dell'energia consumata. Questo per confermare l'efficacia dell'algoritmo di controllo sviluppato e sottolineare la possibilità di raggiungere obiettivi di sostenibilità ed efficienza nel contesto dei moderni sistemi di trasporto.

Abstract

Currently, there is a constant increase in interest in new technologies, fueling an incessant pursuit of the development of increasingly sophisticated forms of integration between humans and machines. Assisted driving systems, such as Adaptive Cruise Control (ACC), represent fundamental solutions in the modern automotive landscape. It is estimated that human error accounts for 93% of road accidents, which in recent years has driven car manufacturers to adopt increasingly complex driving safety systems, including lane-keeping systems, adaptive cruise control, and automatic parking systems. These developments have led to the creation of a new generation of vehicles equipped with a growing number of more advanced safety systems aimed at reducing human intervention in driving.

ACC systems help alleviate driver stress and improve driving comfort, contributing to a significant reduction in road accidents. Furthermore, benefits can be obtained regarding the reduction of fuel consumption and polluting emissions into the environment. Therefore, the implementation of such technologies has a positive impact on the efficiency and sustainability of contemporary mobility.

This thesis aims to explore in detail the research, development, and implementation phases of an ACC based on a Model Predictive Control (MPC) strategy, leveraging the acados framework to facilitate the optimization of the control process. After an initial phase of studying and cataloging existing ACC systems, the focus of the work shifted to writing the controller code in the Matlab environment and subsequently using Simulink to perform various simulations based on predefined scenarios for testing.

Once the standard ACC was designed, attention turned to implementing strategies to account for energy efficiency by creating an Energy Efficient Adaptive Cruise Control (EACC) to optimally utilize the power of the electric motor, increase the overall efficiency of the powertrain and minimize the energy spent per unit distance. Thus, in addition to autonomously regulating the vehicle's speed, energy consumption is minimized.

The validation of the proposed control strategy will be carried out through simulations in Simulink environment. At this stage, a comparison of EACC performance will be conducted. A test prioritizing speed tracking and another prioritizing efficiency terms will be performed to demonstrate the benefits in terms of reduced energy consumption. This will confirm the effectiveness of the developed control algorithm and highlight the possibility of achieving sustainability and efficiency goals within the context of modern transportation systems.

1. Introduzione

1.1 Livelli di guida autonoma

Nel 2014 la SAE (Society of Automotive Engineers), ente di normazione nel campo dell'industria automobilistica, ha stabilito sei livelli differenti di guida autonoma [1], [2], come si vede in Figura 1.1.

Nel Livello 0 il guidatore deve occuparsi interamente della conduzione del veicolo. Sterzo, acceleratore e tutti gli altri sistemi dipendono completamente dal guidatore che deve prestare attenzione all'ambiente circostante e agli eventuali segnali di pericolo. Rientrano in questa categoria tutte le auto realizzate entro gli anni 2000 e quelle dotate di cruise control non adattativo.

Nel Livello 1 il conducente ha ancora il completo controllo del mezzo, ma vi è un sistema che provvede a comunicare all'utente, tramite segnali acustici o visivi, la presenza di eventuali pericoli o situazioni avverse. Rispetto a un livello 0, le implementazioni riguardano diversi aspetti di seguito riportati. Si ha il cruise control adattativo che permette di mantenere un'andatura a velocità costante e nel caso di rilevamento di un mezzo che precede si ha un rallentamento del veicolo; il sistema di Lane Keeping Assistance (LKA) ed Emergency Lane Keeping (ELK), che rendono il veicolo in grado di effettuare piccole correzioni di sterzo o di avvisare il conducente in modo da evitare il superamento non voluto di corsia.

Il Livello 2 è un livello di autonomia parziale. Il veicolo diventa, in determinate situazioni, in grado di gestirsi in maniera autonoma. Ad esempio, in uno scenario come lungo un tragitto in autostrada può assumere il controllo di sterzo e acceleratore, utilizzando in alcuni casi la frenata assistita e la frenata d'emergenza. Secondo la SAE, la differenza sostanziale tra un livello 1 e un livello 2 è che il guidatore può togliere allo stesso momento le mani dal volante e i piedi dai pedali. Tuttavia, ha ancora piena responsabilità del veicolo. La maggior parte dei veicoli attualmente disponibili sul mercato che vantano sistemi di guida automatica, come Tesla, Volvo, Audi, e Maserati, appartengono al livello 2 di automazione e vengono in genere definiti semi-autonomi.

Nel Livello 3 si ha un'automazione condizionale. Le automobili appartenenti a questo livello sono equipaggiate con sistemi in grado di monitorare completamente l'ambiente circostante grazie alla combinazione di sensori e intelligenza artificiale. La differenza rispetto al livello 2 è che l'intervento dell'utente è richiesto solo in situazioni di pericolo, permettendo così al conducente di dedicarsi ad altre attività quando si viaggia a basse velocità. Questa possibilità di distrarsi dalla guida è ciò che distingue sostanzialmente i veicoli di livello 3 da quelli di livello 2.

Nel Livello 4 si ha automazione elevata. Il veicolo è dotato di quasi completa autonomia e può gestire il controllo totale della vettura anche in zone di traffico congestionato e durante la percorrenza di strade extraurbane. L'utente può completamente disinteressarsi dell'ambiente circostante, anche se è necessaria una manovra complessa. Il guidatore diventa principalmente un passeggero, anche se ha ancora la possibilità di intervenire in situazioni che potrebbero superare le capacità del sistema.

Con il Livello 5 si ha automazione totale. Un veicolo di questo livello sarebbe in grado di gestire tutte le funzioni legate alla guida senza alcun intervento umano. Componenti come volante, pedali e cambio diventerebbero superflui poiché il sistema di guida automatica sarebbe in grado di affrontare qualsiasi scenario in completa autonomia, indipendentemente dalla località e dalle

condizioni stradali. Questo livello di tecnologia è ancora un obiettivo futuro per l'industria automobilistica.

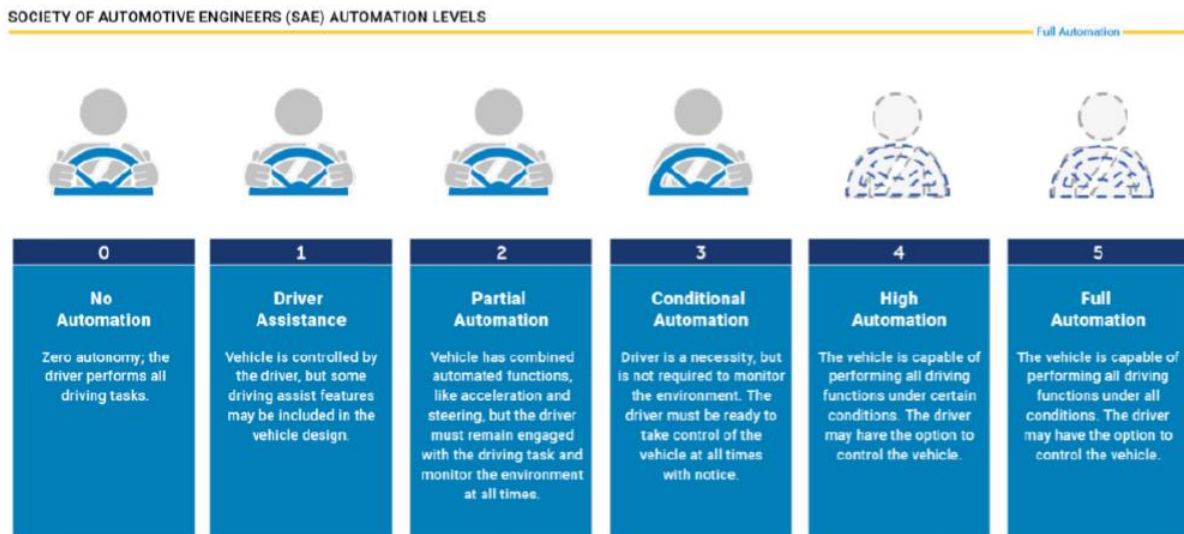


Figura 1.1: Livelli di automazione SAE

1.2 Sistemi ADAS

L'interazione con l'ambiente circostante richiede software specializzati in grado di trasformare la grande quantità di informazioni raccolte in dati utili per il sistema, permettendo così di intervenire a livello meccanico sul veicolo. L'insieme di hardware e software che gestisce specifiche funzionalità è noto come ADAS (Advanced Driver Assistance System). Questi sistemi di sicurezza sono progettati per prevenire collisioni e incidenti, fornendo tecnologie che avvertono il guidatore riguardo a potenziali rischi e problemi [3]. Di seguito sono elencati alcuni degli ADAS attualmente implementati nei veicoli in circolazione.

- L'Adaptive Cruise Control (ACC) permette al veicolo di mantenere una velocità costante senza l'uso del pedale dell'acceleratore e di rallentare automaticamente quando un altro mezzo si avvicina oltre il limite di distanza di sicurezza impostato;
- Il parcheggio automatico è un sistema che assiste il guidatore durante le manovre di parcheggio grazie a sensori di prossimità. Il sistema avvisa l'utente attraverso segnali acustici quando è disponibile uno spazio sufficiente per il parcheggio e controlla lo sterzo. Il conducente è comunque responsabile dell'inserimento delle marce, dell'accelerazione e della frenata;
- Il sistema anticollisione è una funzione di sicurezza che aiuta il guidatore in caso di impatto imminente. Nel caso in cui i sensori rilevino un avvicinamento rapido a un oggetto senza che il guidatore freni, il sistema attiva misure precauzionali come il tensionamento della cintura di sicurezza e l'innescio della frenata;
- Lane Keeping Assist (LKA) è una funzione che attraverso l'utilizzo di sensori permette di avvisare il guidatore, tramite segnali acustici o vibrazioni, di un'eventuale invasione della corsia adiacente. Alcune versioni sono in grado di correggere la traiettoria del veicolo riportandolo in corsia;
- L'Anti-lock Braking System (ABS) è un sistema di sicurezza che previene il bloccaggio delle ruote durante una frenata. Nel momento in cui viene rilevato un bloccaggio il

sistema interviene per ridurre la forza di frenata, consentendo al guidatore di frenare in sicurezza anche in situazioni di emergenza;

- Il sistema di visione notturna (NVS) si divide in due categorie. I sistemi attivi utilizzano dispositivi a infrarossi per rilevare le temperature degli oggetti sulla carreggiata e rappresentarle su un display LCD, con una portata di 150-200 metri. I sistemi passivi, sebbene abbiano una portata intorno ai 300 metri e siano più efficaci in condizioni atmosferiche avverse, presentano limitazioni ad alte temperature;
- Il sistema di riconoscimento del semaforo (TLRS). Questa funzione consente all'auto di prevedere se il prossimo semaforo sarà rosso o verde mentre si avvicina, permettendo così di regolare la velocità per ottimizzare il risparmio energetico;
- Il sistema di navigazione tramite il GPS del veicolo assiste il conducente nella navigazione stradale, fornendo indicazioni sul percorso da seguire verso una destinazione preimpostata.

1.3 Dispositivi utilizzati

Le autovetture per poter sfruttare le tecnologie sopra descritte devono essere dotate di dispositivi che controllano l'ambiente circostante e, quindi, permettono al veicolo di comprendere le azioni che devono essere intraprese [4]. La Figura 1.2 mostra una panoramica dei sensori che vengono utilizzati.

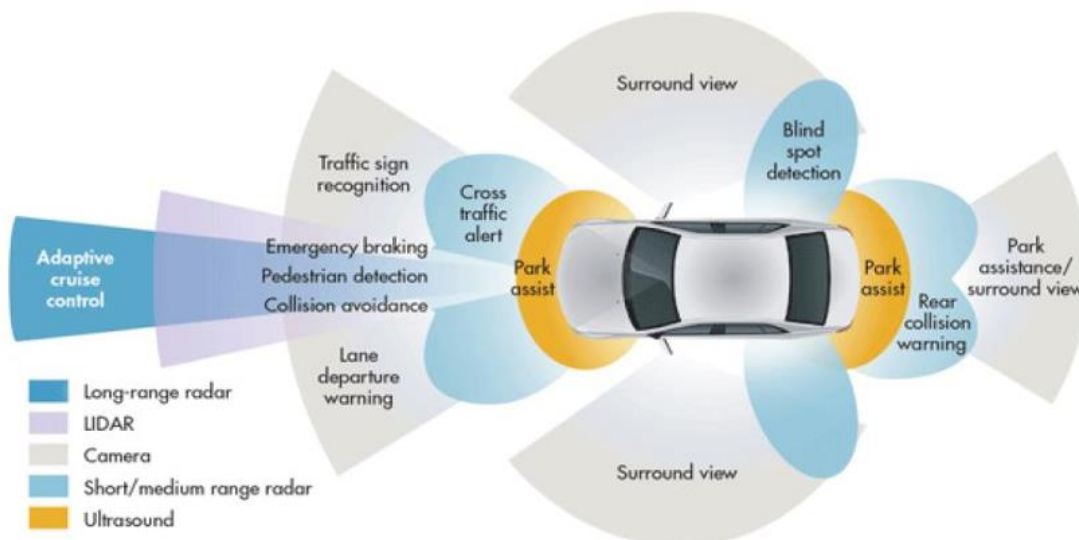


Figura 1.2: Panoramica dei sensori

- Le telecamere prendono le immagini dall'ambiente circostante, processandole successivamente tramite software di decodifica ed elaborazione e, quindi, consentendo al sistema di rilevare tutti i possibili ostacoli lungo il percorso;
- Il radar è un dispositivo che utilizza l'emissione di onde elettromagnetiche per determinare la posizione, le dimensioni e la velocità degli oggetti all'interno del suo raggio d'azione, come mostrato dalla Figura 1.3. Il radar rimane efficace anche in condizioni atmosferiche avverse come pioggia, neve e polvere grazie alla capacità di penetrazione delle sue lunghezze d'onda. Questo permette di migliorare e completare il

lavoro svolto dalle telecamere, rendendolo particolarmente utile per rilevare veicoli e ostacoli in autostrada, in cui le velocità elevate richiedono una risposta al rilevamento rapida e precisa;



Figura 1.3: Radar anticollisione

- Il Lidar (Light Detection and Ranging) è un sistema di scansione che utilizza la luce laser (solitamente nelle frequenze comprese tra l'ultravioletto e l'infrarosso) per determinare la posizione, la velocità e le dimensioni di un oggetto nel raggio d'azione. Esso emette, tramite una sorgente, un fascio laser che rimbalza sugli oggetti da scansionare e viene assorbito da un ricevitore posto in corrispondenza dell'emettitore. Il tempo che intercorre tra l'emissione e il riassorbimento della luce permette di calcolare la distanza da un determinato oggetto/superficie. Questa tecnologia è particolarmente indicata per la creazione di modelli 3D ad alta risoluzione degli oggetti circostanti, fornendo all'unità di controllo dati dettagliati e identificativi su pedoni, veicoli, muri e altri elementi dell'ambiente circostante, come mostra la Figura 1.4. A differenza del radar, che è ottimo per rilevamenti a lunga distanza, il Lidar eccelle nel dettaglio e nella precisione dei dati raccolti a medie e corte distanze. Di solito il Lidar è montato sul tettuccio del veicolo, da dove può avere una visione chiara e libera su 360 gradi, con una portata tipica di circa 200 metri;



Figura 1.4: Immagine 3D fornita dal Lidar

- I sensori ad ultrasuoni utilizzano onde sonore ad alta frequenza, non percepibili dall'orecchio umano, per rilevare oggetti nelle vicinanze del veicolo. Tuttavia, il loro principale limite è il ridotto raggio d'azione, che si estende solo per pochi metri. Inoltre, le informazioni fornite dai sensori ad ultrasuoni sono meno precise rispetto a quelle dei sistemi lidar. Questi sensori sono comunemente impiegati nelle manovre di parcheggio per avvisare il conducente della presenza di veicoli o ostacoli nelle immediate vicinanze;
- Il sistema di geo-localizzazione riceve informazioni dal GPS integrato nell'auto. Il 'Global Positioning System' è un sistema di posizionamento e navigazione basato sull'utilizzo di satelliti. Le stime del GPS possono essere imprecise a causa di disturbi al segnale o da interferenze atmosferiche; quindi, i dati forniti vengono confrontati con una mappa digitale del luogo, ottenuta in precedenza grazie ad altri sensori.

1.4 Adaptive Cruise Control

L'Adaptive Cruise Control, un sistema avanzato di assistenza alla guida (ADAS), modifica automaticamente la velocità del veicolo principale per seguire quello che precede, mantenendo una certa distanza di sicurezza [4], come mostra la Figura 1.5. L'obiettivo principale dei sistemi ACC è quello di migliorare le funzionalità dei tradizionali sistemi di cruise control (CCC - Conventional Cruise Control). Il cruise control consente al conducente di selezionare e mantenere una velocità desiderata senza richiedere un input continuo sul pedale dell'acceleratore. Questo, a differenza dell'Adaptive Cruise Control, non modifica automaticamente la velocità del veicolo in base alla distanza dal veicolo che precede [5].

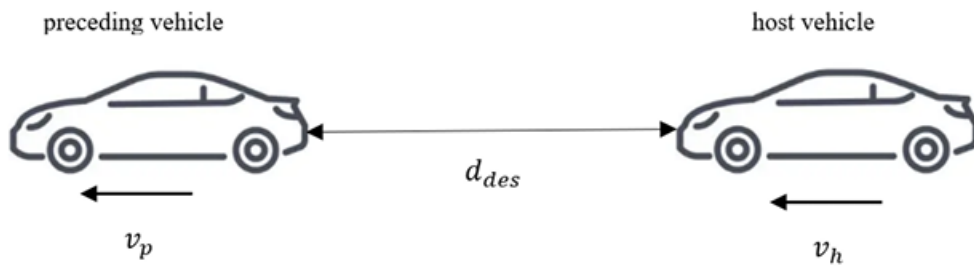


Figura 1.5: Scenario con due veicoli

L'Adaptive Cruise Control utilizza sensori come telecamere, laser, radar e algoritmi intelligenti per regolare la velocità del veicolo in base alle diverse condizioni del traffico, offrendo maggiore sicurezza e comodità ai conducenti. In un tipico sistema ACC vengono utilizzate due modalità di controllo longitudinale. La modalità di controllo della velocità (modalità cruise) viene utilizzata se nessun veicolo si trova davanti al veicolo principale. La modalità di controllo della distanza (modalità following) si attiva quando un veicolo si trova davanti [6].

Un tipico sistema ACC è composto da quattro parti principali: sensori, attuatori, interfaccia uomo-macchina e controllore, come mostra Figura 1.6 [7].

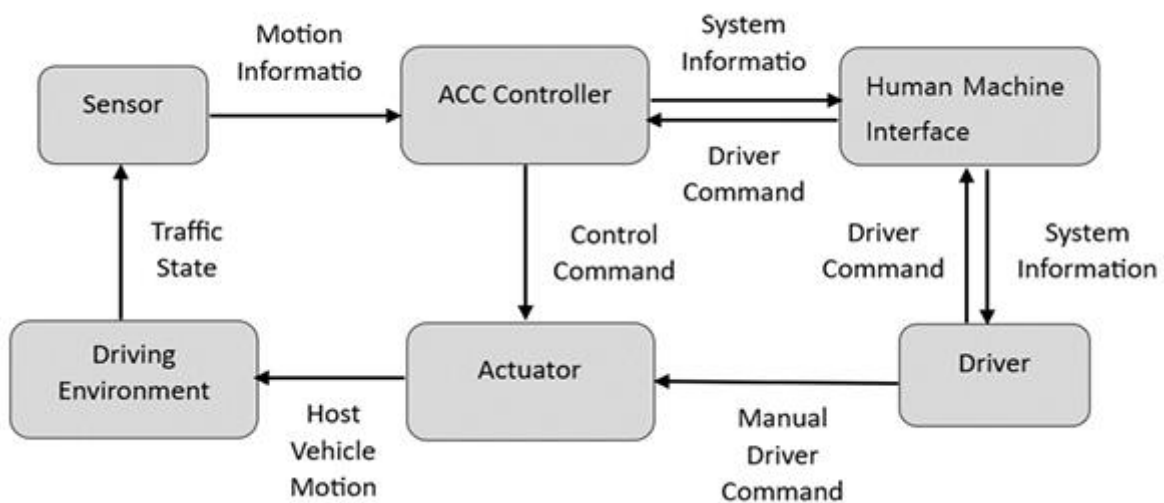


Figura 1.6: Componenti di un sistema ACC

I sensori raccolgono e trasferiscono le informazioni sul movimento del veicolo principale e di quello che precede agli altri componenti del sistema. I sensori di distanza, come telecamere, radar o lidar, inviano informazioni sul veicolo che precede. Altri tipi di sensori, come il sensore di velocità del veicolo e il sensore della posizione dell'acceleratore, inviano informazioni sul veicolo principale. Successivamente, in base ai dati elaborati dai sensori, viene impartito un comando appropriato ai freni e all'acceleratore. L'unità di controllo è quindi responsabile dell'invio dei comandi richiesti al gruppo propulsore e al sistema frenante. Infine, gli attuatori intervengono implementando il comando del controllore per seguire la velocità del veicolo che precede o mantenere la velocità necessaria. Questi attuatori possono essere alimentati da gruppi propulsori a motore a combustione interna (ICE) per veicoli a combustibile convenzionale,

gruppi propulsori elettrici per veicoli elettrici (EV) o entrambi i tipi di gruppi propulsori per veicoli ibridi elettrici (HEV). Inoltre, gli attuatori possono essere freni elettronici, freni idraulici e sistemi di frenatura rigenerativa. L'unità di interfaccia uomo-macchina (HMI) informa il conducente sulle condizioni del veicolo e sui possibili avvisi, in modo che possa intraprendere le azioni necessarie in base alla situazione specifica [7].

1.4.1 Tecniche di controllo

Esistono due strutture di controllo nella letteratura: end-to-end e gerarchico.

Il controller a un livello (end-to-end) invia un segnale di controllo all'attuatore direttamente dalle informazioni di stato del sensore, come mostrato nella Figura 1.7.

Il controller gerarchico è costituito da due o tre livelli, come illustrato nella Figura 1.8. Il livello superiore viene utilizzato a scopo decisionale, mentre il livello inferiore viene utilizzato a scopo di controllo. Quest'ultimo opererà in modalità di controllo della velocità e procederà a una velocità preimpostata dal conducente quando non ci sono veicoli davanti al veicolo controllato (host). Invece, quando si ha un veicolo che precede, l'host vehicle entrerà in modalità di controllo della distanza, "car-following".

Il processo di car-following include un modello di car-following, un algoritmo di ottimizzazione multi-obiettivo e una strategia di regolazione dei pesi.

Il controller di alto livello determina l'accelerazione (o la velocità) ottimizzata da applicare al controller inferiore in base allo stato di movimento dell'host vehicle e del veicolo che lo precede. D'altra parte, il controller di basso livello comanda gli attuatori per seguire questa accelerazione (o velocità) prodotta [7], [8]. Ad esempio, nello studio di Zhu *et al.* [9], il controller superiore valuta lo spazio di sicurezza teorico tra due automobili e l'accelerazione, mentre il controller integrato di basso livello rappresenta la dinamica longitudinale.

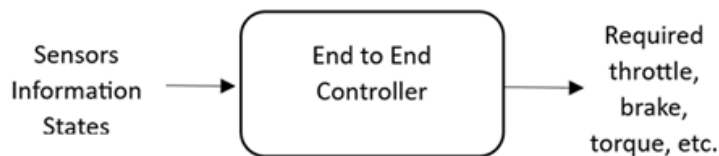


Figura 1.7: Controller a un livello

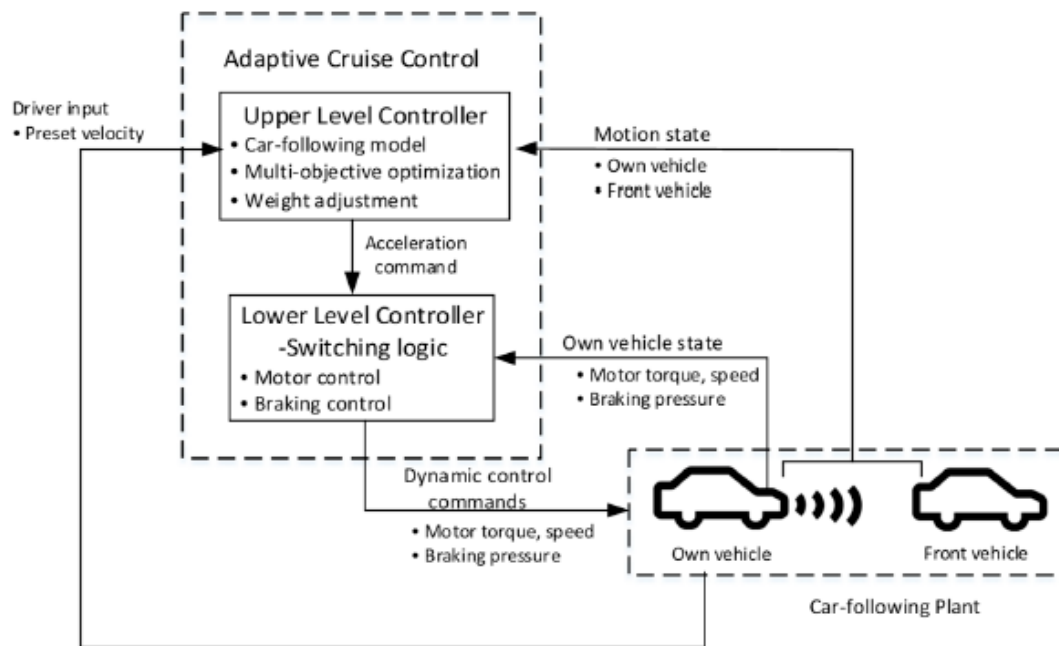


Figura 1.8: Controller gerarchico

2. Energy Efficient Adaptive Cruise Control

Un potenziale sistema ADAS è l'Energy Optimal Adaptive Cruise Control (EACC), ossia un sistema di cruise control adattivo che utilizza informazioni provenienti dall'ambiente esterno, come la topografia della strada, i limiti di velocità e le informazioni sul traffico, per pianificare in tempo reale la velocità che minimizza il consumo energetico dell'host vehicle [10].

Durante un viaggio si prevede che l'host vehicle, controllato in modo ottimale, abbia un consumo di energia minore rispetto al veicolo che lo precede. Oltre all'obiettivo principale di risparmiare energia, l'EACC ha altri due obiettivi di controllo per l'host vehicle:

- mantenere una distanza minima di sicurezza dal veicolo che lo precede;
- seguire il veicolo che lo precede fintanto che non viene violato il limite di velocità [11].

L'EACC, come altri sistemi ADAS, deve essere eseguito in tempo reale dal sistema di calcolo incorporato nel veicolo.

Yanzhao Jia *et al.* ha riportato una revisione dei sistemi di gestione energetica dei veicoli riportando tre approcci utilizzati per i controller operanti in tempo reale:

- leggi di controllo precomutate;
- leggi di controllo calcolate online;
- leggi di controllo miste, precomutate e calcolate online [12].

Le leggi di controllo precomutate sono strategie di controllo in cui le azioni vengono stabilite in anticipo e memorizzate, anziché essere calcolate dinamicamente durante l'esecuzione. Questo approccio prevede che il comportamento del sistema venga analizzato e ottimizzato in diverse condizioni di funzionamento e le regole che governano le azioni di controllo vengano determinate e codificate in anticipo. Esempi di leggi di controllo precomutate includono controller basati su regole e logica fuzzy.

Nei controller basati su regole, le azioni di controllo sono definite da un insieme di regole "se-allora" (if-then). Ad esempio:

- Se la distanza dal veicolo davanti è inferiore a una soglia, *allora* riduci la velocità;
- Se la pendenza della strada aumenta, *allora* aumenta la coppia motrice.

Questi controller sono semplici da implementare, ma richiedono che ogni regola sia progettata manualmente e che tutte le combinazioni di condizioni siano considerate in anticipo.

La logica fuzzy utilizza insiemi fuzzy¹ per rappresentare variabili non esatte, come "velocità elevata" o "distanza molto ravvicinata", e definisce regole basate su tali insiemi. Ad esempio:

- Se la distanza dal veicolo davanti è "media" e la velocità è "elevata", *allora* riduci l'accelerazione.

È utile per gestire sistemi complessi in cui non è semplice definire leggi di controllo esatte [13], [14], [15].

Il problema degli approcci sopracitati riguarda l'eccessivo sforzo richiesto per tarare i parametri di controllo e la sub-ottimalità in nuovi cicli di guida. Infatti, le leggi di controllo precomutate sono ottimizzate per scenari di guida specifici (città, autostrada, traffico regolare); quando il sistema si trova ad affrontare condizioni non previste (ad esempio un nuovo ciclo di guida), le performance possono peggiorare significativamente.

Un esempio di controller misto, precomputato e calcolato online, è il Model Predictive Control, affrontato nel prossimo paragrafo.

¹ Gli insiemi fuzzy sono un concetto matematico utilizzato per rappresentare informazioni che non possono essere descritte in modo preciso o binario, ma che hanno una natura vagamente definita. Sono stati introdotti per trattare situazioni in cui i confini tra appartenenza e non appartenenza a un insieme non sono netti [16].

2.1 Model Predictive Control

Nella progettazione di questo Energy Efficient Adaptive Cruise Control l'approccio di controllo utilizzato è il Model Predictive Control. L'idea del controllo predittivo è quella di utilizzare un modello semplificato del processo al fine di prevedere e ottimizzare il comportamento futuro del sistema. Lo definiamo predittivo perché il modello interno al controllore fa una previsione su un orizzonte di previsione (prediction horizon), cioè su una certa distanza temporale nel futuro.

All'interno del controllore è presente un modello semplificato che, tramite equazioni fisiche o reti neurali, va ad approssimare il comportamento del veicolo nel prediction horizon. Più il modello sarà preciso, più la risposta del controllore sarà buona perché approssimerà meglio il futuro del comportamento del veicolo; quindi, potrà calcolare delle azioni di controllo più precise e migliori sul plant.

Il controllo predittivo basato su modello è un metodo in cui l'input di controllo ottimale viene calcolato basandosi su previsioni dell'evoluzione del sistema, fornite appunto da un modello del sistema.

La Figura 2.1 mostra un ciclo di controllo generico con un controllore MPC. L'MPC è composto da un modello del sistema e da una parte di ottimizzazione. I segnali di controllo ottimali " u " sono calcolati su un orizzonte discreto di lunghezza " N ".

La funzione di costo è l'elemento chiave che guida il processo decisionale per il controllo del sistema. Essa quantifica l'obiettivo del controllo in termini matematici. L'ottimizzazione mira a minimizzare una funzione obiettivo (funzione di costo) che è progettata per penalizzare l'errore tra uno stato e il riferimento dello stato stesso. Può essere una funzione multi-obiettivo, ossia contenere diversi obiettivi di controllo.

Una proprietà dell'MPC è che nel processo di ottimizzazione si può imporre dei vincoli sul segnale di controllo e sugli stati.

Gli stati " y " lungo il prediction horizon sono previsti dal modello del sistema sulla base dello stato iniziale misurato e dei segnali di controllo ottimizzati [17].

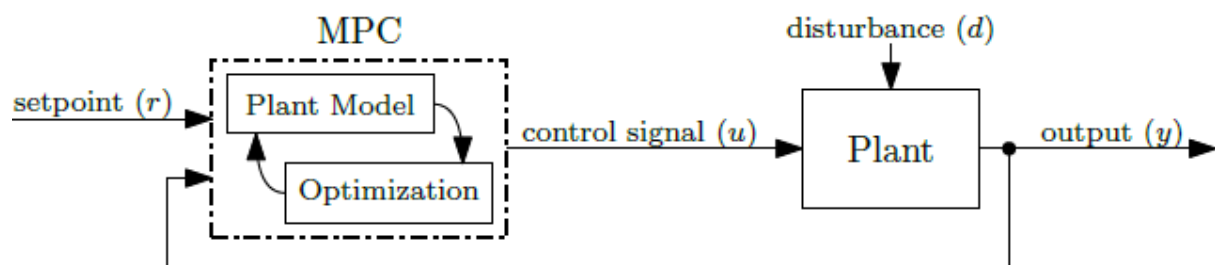


Figura 2.1: MPC control loop

2.2 Formulazione del problema di ottimizzazione

La formulazione del problema di ottimizzazione consiste nel definire la funzione obiettivo o funzione di costo da minimizzare e i vincoli a cui sono soggette le variabili di ottimizzazione.

2.2.1 Funzione di costo

La funzione di costo è l'elemento chiave che guida il processo decisionale per il controllo del sistema. Essa quantifica l'obiettivo del controllo in termini matematici.

La funzione di costo comprende:

- l'errore di tracciamento: termine che indica la variazione degli stati del sistema rispetto ai valori desiderati (di riferimento);

$$J = \sum_{k=0}^{N-1} (x_k - x_{ref,k})^2 * Q \quad (2.1)$$

x_k è il vettore degli stati al tempo k ;

$x_{ref,k}$ è il vettore degli stati di riferimento al tempo k ;

Q è una matrice di pesi che penalizza le deviazioni degli stati dai valori di riferimento.

- costo degli input di controllo: penalizza l'uso degli input di controllo, facendo sì che il sistema non utilizzi valori eccessivi degli input.

$$J_{input} = \sum_{k=0}^{N-1} u_k^2 * R \quad (2.2)$$

u_k è il vettore degli input di controllo al tempo k ;

R è una matrice di pesi che penalizza gli input di controllo.

- costo dello stato finale: questo termine penalizza la deviazione dell'ultimo stato previsto rispetto al riferimento finale.

$$J_{final} = (x_N - x_{ref,N})^2 * Q_f \quad (2.3)$$

x_N è il vettore degli stati al tempo finale dell'orizzonte di previsione;

$x_{ref,N}$ è il vettore degli stati di riferimento al tempo finale;

Q_f è una matrice di pesi per lo stato finale.

La funzione di costo deve essere minimizzata per trovare la sequenza ottimale di input di controllo. In seguito all'ottimizzazione solo il primo input di controllo della sequenza ottimale calcolata viene applicato al sistema. Gli altri input non vengono direttamente applicati, ma serviranno come punto di partenza per la prossima iterazione.

Le matrici dei pesi sono utilizzate per ponderare i vari termini nella funzione di costo. L'obiettivo è attribuire maggiore o minore importanza a determinati aspetti del comportamento del sistema.

Per quanto riguarda la variabile di controllo, maggiore sarà il coefficiente di peso attribuito ad essa, maggiore è la volontà di ottenere dal controllore una risposta in termini di variabile di controllo. Contrariamente, se viene dato un peso basso al termine di controllo, si lascia il sistema libero di operare nel modo che ritiene più opportuno per raggiungere l'obiettivo impostato nella funzione di costo.

2.2.2 Receding Control Horizon

L'orizzonte su cui l'MPC ottimizza i controlli e gli stati può essere visto come una finestra temporale di durata finita. Dopo ogni istante temporale discreto, questa finestra si sposta di un passo in avanti nel tempo.

Bisogna fare una distinzione tra passi temporali discreti “t” che si riferiscono agli istanti di tempo “reali” e passi temporali discreti “k” che si riferiscono ai passi temporali nell'orizzonte di previsione dell'MPC.

Per uno step di ottimizzazione MPC all'istante temporale discreto “t” sono definiti due orizzonti, come mostrato nella Figura 2.2.

L'MPC deve ottimizzare i segnali di controllo futuri e le uscite previste, che sul grafico sono rappresentati con una linea tratteggiata. Il riferimento “r” è un input al sistema, quindi è noto al tempo “t”. Per ogni iterazione dell'MPC, “k” è impostato a zero al tempo “t”.

L'orizzonte di previsione è definito da “ N_p ” e consiste negli istanti temporali $k \in \{1, \dots, N_p\}$ su cui gli stati saranno previsti in base ai controlli che saranno ottimizzati.

L'altro orizzonte è l'orizzonte di controllo, che comprende gli istanti temporali discreti $k \in \{N_d, \dots, N_c-1\}$. Esso contiene tutti i “k” per cui vengono determinati gli input di controllo ottimali. L'offset “ N_d ” all'inizio dell'orizzonte di controllo rappresenta il ritardo tra l'inizio dell'ottimizzazione e l'applicazione dell'azione di controllo. Deve sempre valere $N_d \leq N_c \leq N_p$, poiché è impossibile ottimizzare un input di controllo a un istante per cui non c'è previsione. Per la previsione ai passi temporali prima dell'orizzonte di controllo, $k \in \{0, N_d-1\}$, vengono utilizzati gli input di controllo applicati in precedenza o pianificati per essere applicati [17].

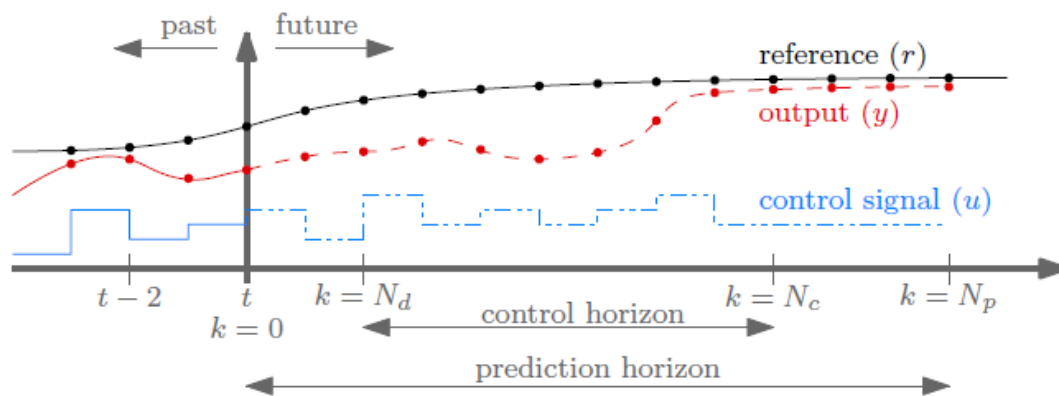


Figura 2.2: Control and prediction horizon in MPC

2.2.3 Vincoli

Nel Model Predictive Control si possono utilizzare dei vincoli per garantire che i valori delle variabili di ottimizzazione, stati e controlli, rispettino i limiti fisici del sistema. Si possono utilizzare i vincoli di uguaglianza (=) e disuguaglianza (stretta: $<$, $>$ e non stretta: $\leq$, $\geq$). I vincoli possono essere hard o soft.

Solitamente si mettono hard constraints sugli output del controllore, cioè sui control inputs; ad esempio, si va ad applicare un valore massimo e minimo alla coppia di un motore elettrico perché essa non può eccedere oltre un certo valore.

Le slack variables possono essere utilizzate per rilassare i vincoli rigidi e quindi creare dei soft constraints. L'applicazione delle slack variables agli input di controllo spesso non è attuabile, poiché, come detto in precedenza, gli attuatori nella pratica accettano solo valori entro un certo intervallo. Quindi, i vincoli su questi segnali devono rimanere "rigidi".

Il rilassamento dei vincoli può essere utilizzato per garantire la fattibilità e far sì che il controllore non dia problemi durante l'applicazione del controllo nel caso in cui una variabile,

come ad esempio la velocità o l'accelerazione, sfiora rispetto ai valori definiti dal vincolo rigido. Quindi, garantendo una certa violabilità del vincolo rigido si ottiene un vincolo soft. Per rilassare un vincolo viene aggiunta la variabile ϵ che permette di estendere i limiti del vincolo. Ad esempio:

$$y_{lb} - \epsilon_k \leq y_k \leq y_{ub} + \epsilon_k \quad (2.4)$$

$$\epsilon_k \geq 0 \quad (2.5)$$

È necessario che la variabile ϵ sia positiva, altrimenti il vincolo verrebbe rafforzato anziché rilassato. L'aggiunta della variabile ϵ alla funzione di costo avviene attribuendo un fattore di peso che varierà a seconda del comportamento che si vuole ottenere. Un elevato peso attribuito alla slack variable agirà in modo tale che il sistema, appena superato il valore limite impostato, opererà nella zona di tolleranza in modo tale da mantenersi in essa, per poi rientrare nella zona caratterizzata dal vincolo forte che sarebbe la normale zona di operabilità. Il fattore di peso deve essere abbastanza alto da non consentire violazioni dei vincoli se non strettamente necessario [17].

2.3 Tuning an MPC controller

L'MPC, come detto in precedenza, è caratterizzato dall'avere un internal model (internamente al controllore) e un plant. Quest'ultimo potrebbe rappresentare un veicolo reale con tutte le sue non linearità, per cui il plant potrebbe essere molto più complesso come fisica che sta dietro, rispetto l'internal model. Però, più è preciso l'internal model, migliore sarà il comportamento del plant lungo il prediction horizon e questo permetterà al problema di ottimizzazione di trovare l'azione di controllo ottimale lungo l'orizzonte di predizione.

La complessità del modello influenza il carico computazionale, però modificandolo o semplificandolo potrebbe peggiorare le prestazioni. Questo tipo di regolazione è piuttosto indiretta, poiché non consiste semplicemente nel regolare un valore, bensì nel cambiare l'intero modello.

Un parametro di regolazione più diretto è la lunghezza del prediction horizon, che come già esplicitato risulta essere il periodo di tempo futuro durante il quale il sistema prevede gli effetti delle azioni di controllo. Avere un orizzonte di previsione più lungo consente al controllore di considerare in maniera migliore gli effetti a lungo termine delle sue decisioni. Questo è utile in sistemi complessi in cui le decisioni prese in un determinato momento possono influenzare le prestazioni future. Tuttavia, un orizzonte troppo lungo potrebbe aumentare il costo computazionale e rendere il problema di ottimizzazione più complesso.

L'orizzonte di controllo definisce per quanto tempo nel futuro il controller calcola le azioni da applicare. Può essere inferiore all'orizzonte di previsione, poiché non è necessario ottimizzare tutte le azioni future con lo stesso livello di dettaglio. Questo approccio riduce il numero di variabili di ottimizzazione, diminuendo così il carico computazionale. Ad esempio, il controller può calcolare con precisione le azioni per i primi passi temporali, ma i passi che verranno eseguiti in un lasso temporale successivo verranno affidati dal sistema alla semplice previsione. Analizzando la letteratura in materia è stato deciso di impostare i parametri dell'MPC come in Tabella 2.1 [4], [18], [19], [20]:

Tabella 2.1: Parametri impostati nell'MPC

Parameter	Symbol	Value	Unit
Steps along prediction horizon	N	20	[-]
Length of single step time	dT	200	[ms]

2.4 Implicit and explicit MPC

L'MPC implicito (MPC online) fa riferimento a un'implementazione in tempo reale in cui il problema di ottimizzazione viene risolto a ogni passo temporale. In questo caso, il tempo richiesto per l'ottimizzazione deve essere inferiore rispetto al tempo di campionamento del controllore. In alcuni casi non è possibile soddisfare gli obiettivi di controllo con una formulazione MPC che agisca in tempo reale; quindi, la soluzione può essere l'MPC esplicito (MPC offline). Nell'MPC esplicito le soluzioni ottimali per un dato intervallo di parametri sono calcolate in anticipo e memorizzate in una tabella di ricerca (look-up table). Siccome la dimensione della tabella di ricerca cresce rapidamente per problemi più complessi, l'MPC esplicito può essere applicato solo su un intervallo ristretto di stati possibili, il che può limitarne il potenziale e compromettere la robustezza del controllore.

L'MPC esplicito è utile, ad esempio, nell'elettronica di potenza, in cui i modelli sono relativamente semplici e i tempi di campionamento devono essere così ridotti da rendere l'ottimizzazione implicita troppo onerosa computazionalmente [17].

2.5 Optimization algorithms, solvers and toolboxes for MPC

L'MPC può contare su una vasta gamma di algoritmi e risolutori per i problemi di ottimizzazione vincolata. L'MPC basato su Non-Linear-Programming (programmazione non lineare) include i metodi dei punti interni, i metodi dei set attivi e i metodi di programmazione quadratica sequenziale. Nella programmazione non lineare vengono utilizzati anche metodi meta-euristici che esplorano lo spazio di ricerca per trovare una soluzione sufficientemente buona. Un esempio, è l'ottimizzazione con sciame di particelle (PSO) utilizzata per l'MPC non lineare in [21].

I risolutori sono spesso progettati per uno specifico problema di ottimizzazione (ad esempio, programmazione quadratica) e richiedono che il problema sia espresso in una certa forma. Un esempio di risolutore è qpOASES [22] che si basa su una strategia di set attivi online per la programmazione quadratica (QP). Un altro risolutore è OSQP [23] il quale utilizza un "metodo alternato dei moltiplicatori di direzione" per risolvere problemi quadratici. OSQP permette la generazione del codice nel linguaggio di programmazione C, utile per applicazioni MPC. La maggior parte dei toolbox di ottimizzazione spesso offrono anche interfacce con software come Matlab, Octave e linguaggi di programmazione come Python. Esempi di toolbox di ottimizzazione sono acados [24], YALMIP [25] e CasADi [26], [27].

2.6 Vantaggi e svantaggi dell'MPC

Uno dei principali vantaggi del controllo predittivo basato su modello rispetto alle tecniche di controllo classiche risiede nell'aspetto predittivo. A condizione che i modelli predittivi utilizzati per l'MPC siano sufficientemente buoni, ciò porta a un controllo più robusto e accurato.

L'MPC permette di considerare vincoli fisici e operativi del sistema nel problema di ottimizzazione, come limiti di velocità, accelerazione o capacità dell'attuatore. Questo è un aspetto unico rispetto ad altri metodi di controllo.

La funzione di costo può contenere diverse variabili che si vuole siano minimizzate, in questo modo l'MPC può bilanciare obiettivi multipli, come l'efficienza energetica e la sicurezza, rendendolo ideale per applicazioni come i sistemi di guida assistita o la gestione energetica.

L'MPC è particolarmente vantaggioso per sistemi con più ingressi e uscite (MIMO - Multiple Input, Multiple Output), in cui le interazioni tra variabili possono essere gestite in modo efficace.

Infine, la capacità di aggiornare il controllo in tempo reale, considerando nuove informazioni o variazioni nell'ambiente, rende l'MPC dinamico e adattabile.

Uno degli svantaggi nell'utilizzo dell'MPC riguarda il fatto che l'ottimizzazione possa essere computazionalmente costosa, in particolare per problemi non lineari. Per ridurre il carico computazionale, il problema di ottimizzazione può essere semplificato andando a modificare la lunghezza dell'orizzonte di predizione e la frequenza di campionamento. Un altro possibile svantaggio è la complessità della modellazione, poiché in casi estremi potrebbero essere necessari modelli molto complessi per ottenere prestazioni eccellenti. Tuttavia, per la maggior parte dei sistemi, modelli relativamente semplici sono sufficienti. Per sistemi molto veloci, sono addirittura necessari modelli più semplici per ridurre i costi computazionali [17].

2.7 Plant model

In questo lavoro le equazioni all'interno del plant sono identiche a quelle definite nell'internal model.

Le equazioni all'interno del VEHICLE PLANT nell'ambiente Simulink, come si vedrà nel capitolo 3, sono quelle relative al moto longitudinale, e un'equazione in cui si tiene conto del delay 'tau' tra la forza richiesta al motore elettrico F_{req} e la forza meccanica effettivamente applicata F_{mc} . In seguito, si spiegherà come sono state ottenute le mappe del motore elettrico e del gearbox.

2.7.1 Longitudinal dynamics

La forza risultante sul veicolo lungo la direzione longitudinale è una combinazione della forza generata dal motore elettrico, F_{mc} , della forza di resistenza al rotolamento, F_{roll} , della forza di resistenza aerodinamica F_{aero} , e della forza gravitazionale, F_{slope} (in cui si tiene conto della pendenza della strada). Qui, per semplicità, è stata considerata strada rettilinea, quindi $\theta=0$. La forza del motore può essere sia positiva che negativa a seconda che il motore elettrico venga utilizzato per la propulsione o per la frenata rigenerativa. Utilizzando la seconda legge di Newton, si ottiene l'equazione del moto nella direzione longitudinale:

$$\sum F = F_{mc} - F_{roll} - F_{aero} - F_{slope} = m_{eq} * a \quad (2.6)$$

In cui:

$$m_{eq} = \frac{1}{4} * (M_v + (4 * \frac{J_w}{R_w^2}) + (4 * \frac{J_{EM} * i_{tot}^2}{R_w^2}) + (4 * \frac{J_{gear2}}{R_w^2}) + (4 * \frac{J_{gear1} * i_{tot}^2}{R_w^2})) \quad (2.7)$$

m_{eq} combina la massa reale del veicolo, M_v , con l'effetto dell'inerzia rotazionale del sistema di trazione (ruote, motore, differenziale). Ciò, rappresenta la massa effettiva che deve essere accelerata o decelerata. La Tabella 2.2 descrive le variabili che ne definiscono l'equazione. Si parla di *Tire inertia* per stimare l'energia cinetica rotazionale delle ruote; *Rotor inertia* influenza la risposta dinamica del motore durante le variazioni di velocità; $J_{gear,2}$, $J_{gear,1}$ sono importanti per determinare la resistenza interna del sistema di trasmissione quando si modifica la coppia trasmessa.

Tabella 2.2: Variabili che definiscono l'equazione per la massa equivalente

Parameter	Symbol	Value	Unit
Vehicle mass	M_v	2350	[Kg]
Wheel radius	R_w	0.3708	[m]
Transmission ratio	i_{tot}	10.5	[-]
Tire inertia	J_w	$25 * \frac{R_w^2}{2}$	[Kgm ²]
Rotor inertia	J_{EM}	0.045/2	[Kgm ²]
Gear inertia	$J_{gear,2}$	0.0015	[Kgm ²]
Gear inertia	$J_{gear,1}$	0.001	[Kgm ²]

Per quanto riguarda le forze esercitate vi sono la forza di resistenza al rotolamento, la forza aerodinamica di cui si riportano le relative variabili (rispettivamente Tabella 2.3 e Tabella 2.4) e la forza gravitazionale dovuta alla pendenza della strada, le cui componenti vengono mostrate nella Figura 2.3.

$$F_{roll} = m_{eq} * g * \cos(\theta_{rad}) * (f_0 + (f_2 * v_h^2)) \quad (2.8)$$

Tabella 2.3: Variabili che definiscono la forza di resistenza al rotolamento

Parameter	Symbol	Value	Unit
Gravitational acceleration	g	9.81	[m/s ²]
Rolling resistance constant coefficient	f_0	0.008	[-]
Rolling resistance quadratic coefficient	f_2	0.0002	[-]
Ego vehicle speed	v_h		[m/s]
Road slope	θ	0	[°]

$$F_{aero} = 0.5 * \rho * c_d * A * v_h^2 \quad (2.9)$$

Tabella 2.4: Variabili che definiscono la forza di resistenza aerodinamica

Parameter	Symbol	Value	Unit
Air density	ρ	1.18	[Kg/m ³]
Airdrag coefficient	c_d	0.3	[-]
Vehicle frontal area	A	2.5	[m ²]

$$F_{slope} = m_{eq} * g * \sin(\theta_{rad}) \quad (2.10)$$

$$F_{res} = F_{roll} + F_{aero} + F_{slope} \quad (2.11)$$

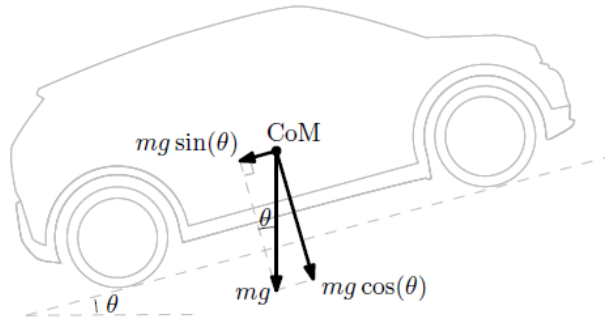


Figura 2.3: Componenti della forza gravitazionale agenti su un veicolo che percorre una pendenza

Successivamente, dall'equazione del moto longitudinale si ricava l'accelerazione dell'ego vehicle:

$$a_h = \frac{F_{mc} - F_{res}}{m_{eq}} \quad (2.12)$$

Viene, poi, utilizzata un'equazione per tenere conto del delay tra la forza richiesta al motore elettrico F_{req} e la forza meccanica effettivamente applicata F_{mc} . Tale equazione permetterà di ricavare la derivata della forza meccanica $F_{mc,d}$.

$$F_{req} = F_{mc} + (F_{mc,d} * \tau) \quad (2.13)$$

$$F_{mc,d} = \frac{(F_{req} - F_{mc})}{\tau} \quad (2.14)$$

Infine, viene definita la velocità relativa tra il veicolo che precede e l'ego vehicle:

$$v_{rel} = v_p - v_h \quad (2.15)$$

2.7.2 Powertrain model

In questo lavoro non vengono considerati gli effetti delle dinamiche della batteria (ad esempio, lo stato di carica SoC). Si partirà considerando il flusso di potenza in ingresso all'inverter, che transiterà poi attraverso motore elettrico e differenziale, fino alla strada.

Un breve cenno alle curve caratteristiche di potenza e coppia di un motore elettrico vengono mostrate nella Figura 2.4. Entrambe le quantità sono funzione della velocità di rotazione ω dell'albero motore.

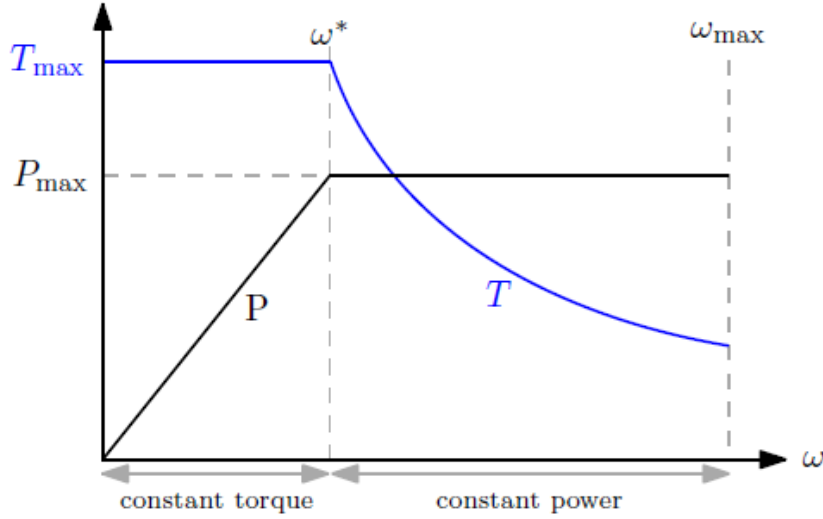


Figura 2.4: Mappa caratteristica di coppia e potenza per un motore elettrico [28]

Poiché la potenza erogata da un motore elettrico è teoricamente costante a $P = P_{max}$ e può essere espressa come $P = T * \omega$, la coppia diventerebbe infinitamente alta quando $\omega \rightarrow 0$. In pratica, il controller del motore realizza una regione di coppia costante in cui $T = T_{max}$ al di sotto di una velocità di rotazione ω^* . Questo ha come conseguenza che la potenza cresce linearmente con la velocità di rotazione quando $0 < \omega < \omega^*$, successivamente rimane costante alla potenza P_{max} fino alla velocità di rotazione massima del motore ω_{max} .

Le perdite energetiche nel powertrain sono causate dall'inverter, che si assume avere un'efficienza costante $\eta_{inv} = 0.98$, dall'efficienza del motore elettrico $\eta_{EM}(T_{EM}, \omega_{EM})$ e dall'efficienza del gearbox $\eta_{gearbox}(T_{EM}, \omega_{EM})$.

La forza di trazione esercitata, F_{mc} , è associata alla coppia, $T_{mc,EM}$, che il motore può fornire a una certa velocità angolare $\omega_{mc,EM}$.

$$T_{mc,EM} = \left(\frac{R_w}{i_{tot}} \right) * F_{mc} \quad (2.16)$$

$$\omega_{mc,EM} = \left(\frac{i_{tot}}{R_w} \right) * v_h \quad (2.17)$$

$$\omega_{mc,EM,rpm} = \omega_{mc,EM} * \left(\frac{30}{\pi} \right) \quad (2.18)$$

Infine, si considera la potenza del motore:

$$P_{mc,EM} = T_{mc,EM} * \omega_{mc,EM} \quad (2.19)$$

Si possono distinguere due casi. Quando la coppia è positiva, ossia il motore fornisce potenza alle ruote, la potenza elettrica estratta dalla batteria è la potenza meccanica divisa per l'efficienza totale. Quando la coppia è negativa, ossia durante la frenata rigenerativa, la potenza immagazzinata è pari al prodotto tra la potenza meccanica e l'efficienza totale.

$$P_{inv,traction} = \frac{P_{mc,EM}}{\eta_{tot}} \quad (2.20)$$

$$P_{inv,reg} = P_{mc,EM} * \eta_{tot} \quad (2.21)$$

Per quanto riguarda l'efficienza del gearbox è stata fornita una mappa in funzione della coppia T_{EM} e della velocità angolare w_{EM} del motore.

La mappa originale definisce i valori di efficienza in una matrice ottenuta dai vettori di coppia e velocità, T_{eff} e w_{eff} .

$$\begin{aligned} T_{eff} &= [-310 -290 -270 -250 -230 -210 -190 -170 -150 -130 -110 -90 -70 -50 -30 -10 \ 10 \\ &\quad 30 \ 50 \ 70 \ 90 \ 110 \ 130 \ 150 \ 170 \ 190 \ 210 \ 230 \ 250 \ 270 \ 290 \ 310].'; \\ w_{eff} &= [10 \ 1000 \ 2000 \ 3000 \ 4000 \ 5000 \ 6000 \ 7000 \ 8000 \ 9000 \ 10000].'; \end{aligned}$$

Poiché la mappa è definita in punti discreti, effettuando un'interpolazione si può andare a considerare valori di efficienza in punti intermedi, non direttamente mappati.

La mappa è stata interpolata su una griglia più densa tramite interpolazione lineare.

È stata creata una griglia interpolata tramite un vettore delle velocità $w_{eff,interp}$ e un vettore delle coppie $T_{eff,interp}$ rispettivamente con 1000 e 300 punti, in modo da generare una superficie continua su tutto il dominio della mappa.

$$\begin{aligned} w_{eff_interp} &= \text{linspace}(\min(w_{eff}), \max(w_{eff}), 1000).'; \\ T_{eff_interp} &= \text{linspace}(\min(T_{eff}), \max(T_{eff}), 300).'; \\ [X_interp, Y_interp] &= \text{meshgrid}(w_{eff_interp}, T_{eff_interp}); \end{aligned}$$

Successivamente, tramite la funzione *interp2*, è stata effettuata l'interpolazione lineare sulla mappa iniziale, ottenendo così un campo di valori di efficienza eff_interp del gearbox che copre qualsiasi coppia e velocità angolare all'interno dell'intervallo di dati fornito.

$$eff_interp = \text{interp2}(w_{eff}, T_{eff}, eff_gearbox, X_interp, Y_interp, 'linear');$$

La superficie risultante può essere visualizzata tramite il comando *surf*, come mostra Figura 2.5:

$$\text{surf}(X_interp, Y_interp, eff_interp);$$

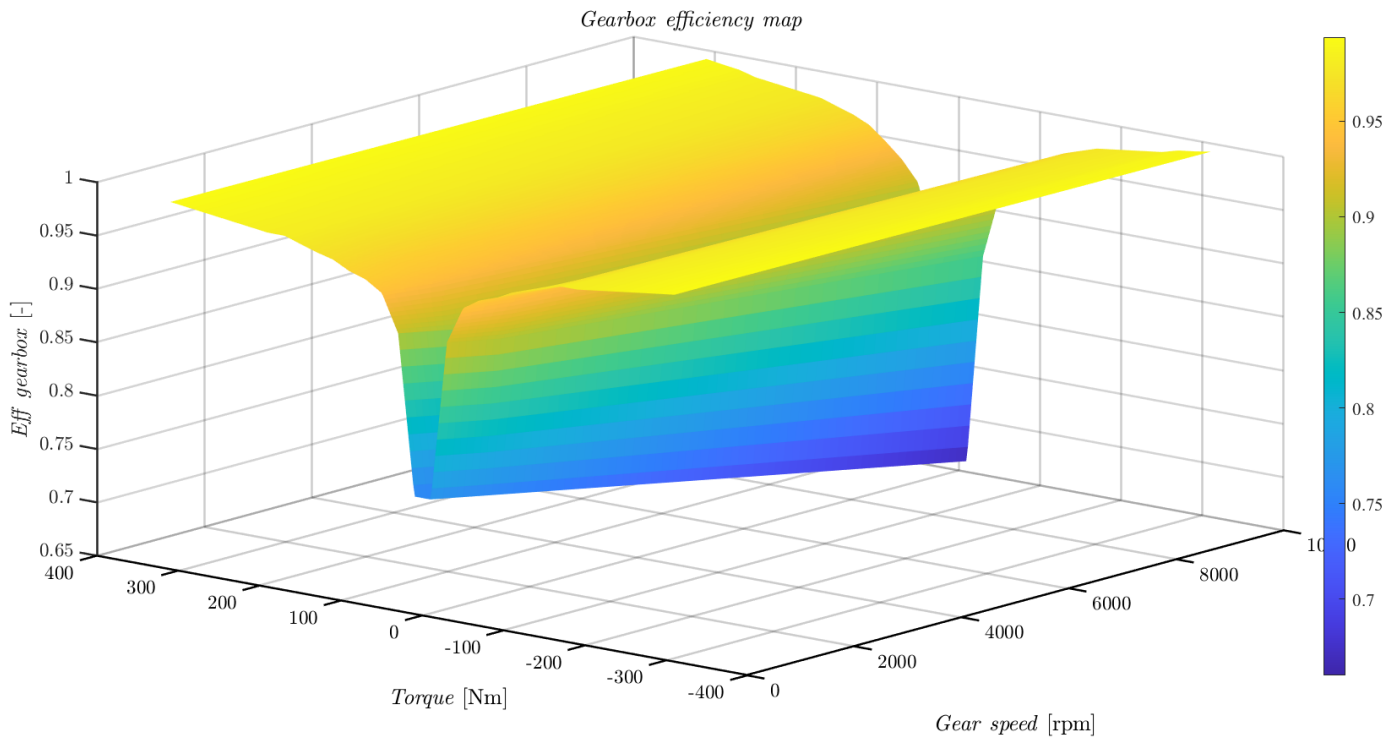


Figura 2.5: Gearbox efficiency map

Per il motore elettrico è stata utilizzata una mappa di efficienza fornita su un file Excel. Sono stati estratti i vettori di velocità di rotazione, coppia e la matrice delle efficienze tramite la funzione *readmatrix*.

```
speed_vector = readmatrix('EM_efficiency.xlsx','Range','B2:DR2').';
torque_vector = readmatrix('EM_efficiency.xlsx','Range','A3:A74');
efficiency_motor = readmatrix('EM_efficiency.xlsx','Range','B3:DR74');
```

Anche in questo caso è stata effettuata un'interpolazione per adattare la mappa a una griglia più densa di valori di coppia e velocità di rotazione. Il procedimento è identico a quello usato per l'efficienza del gearbox.

```
w_eff_interp_EM = linspace(min(w_eff),max(w_eff),1000).';
T_eff_interp_EM = linspace(min(T_eff),max(T_eff),300).';
[X_interp_EM,Y_interp_EM] = meshgrid(w_eff_interp_EM,T_eff_interp_EM);
```

Si è poi ottenuta la matrice interpolata *eff_interp_EM*, tramite la funzione *interp2*, e la visualizzazione della mappa tramite il comando *surf*, come mostra Figura 2.6.

```
eff_interp_EM =
interp2(speed_vector,torque_vector,efficiency_motor,X_interp_EM,Y_interp_EM,'linear'
);
surf(X_interp_EM, Y_interp_EM, eff_interp_EM);
```

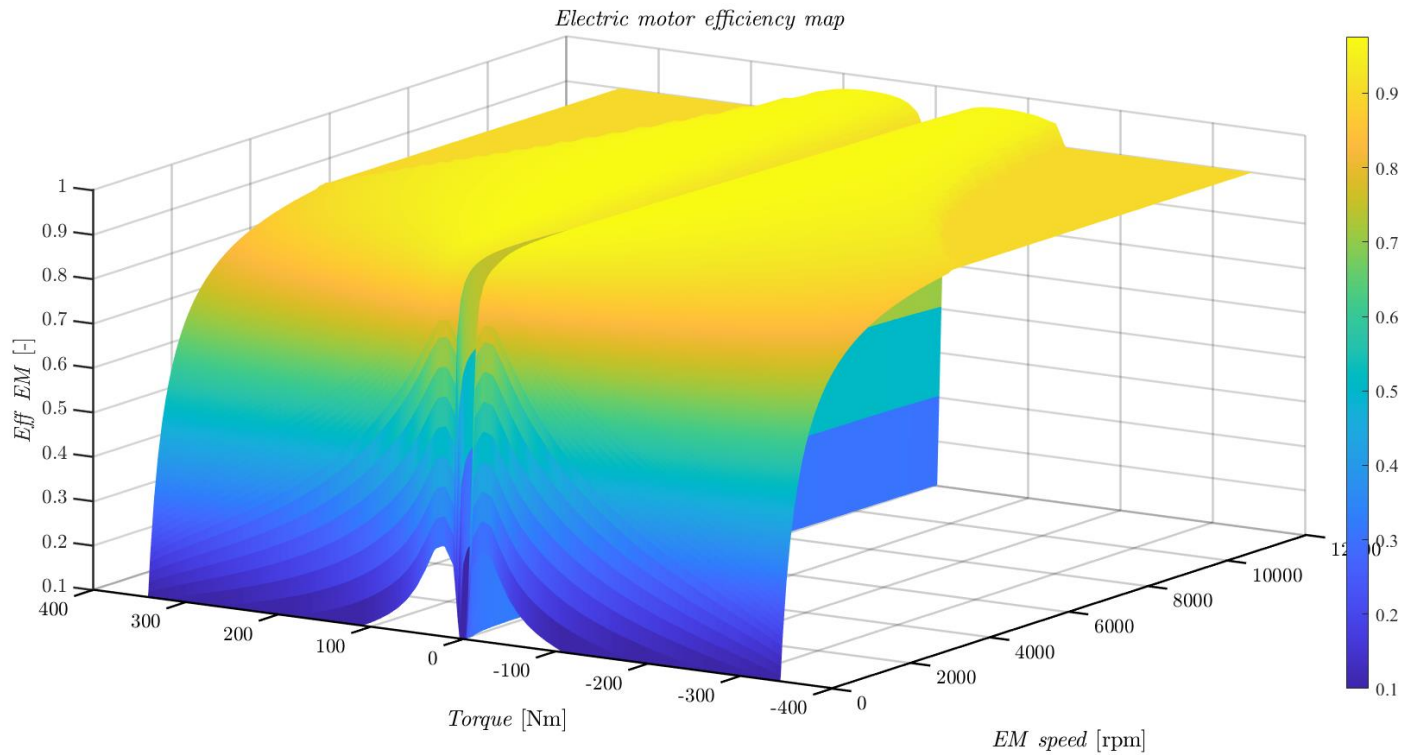


Figura 2.6: Electric motor efficiency map

2.8 Internal Model

L'MPC opera risolvendo iterativamente molti problemi di controllo ottimale (OCP – Optimal Control Problem) sull'orizzonte di predizione, adattando i risultati in base alle condizioni attuali del sistema. Questo richiede una notevole rapidità computazionale per aggiornare continuamente gli input di controllo durante l'esecuzione. Per semplificare i calcoli e garantire una soluzione in tempo reale, le dinamiche del sistema e i vincoli, se non lineari, vengono spesso linearizzati attorno allo stato corrente. Questa approssimazione consente di trasformare il problema in una programmazione quadratica (QP), più semplice e veloce da risolvere.

2.8.1 CasADi e acados

In questo lavoro la risoluzione del problema di ottimizzazione non lineare (NLP) è avvenuto utilizzando in Matlab la libreria CasADi in combinazione con acados. CasADi è una libreria open-source dedicata alla modellazione e risoluzione di problemi di ottimizzazione non lineare, che consente di formulare problemi di controllo ottimo con vincoli e funzioni obiettivo complesse.

Acados, integrato con CasADi, offre solutori efficienti e ottimizzati per problemi di controllo predittivo non lineare in tempo reale (NMPC) e per risolvere problemi di controllo ottimo (OCP) in modo accurato e rapido.

Acados è stato configurato per il problema di ottimizzazione non lineare utilizzando le seguenti opzioni, come mostra Tabella 2.5:

Tabella 2.5: Impostazioni del solver

Parameter	Symbol	Value
Solver NLP	<i>nlp_solver</i>	sqp
Solver for Quadratic Programming	<i>qp_solver</i>	partial_condensing_hpipm
New horizon after partial condensing	<i>qp_solver_cond_N</i>	20
Integrator type	<i>sim_method</i>	irk
Maximum number of NLP iterations	<i>nlp_max_iter</i>	20
Constraints type	<i>constr_type</i>	auto
Cost function type	<i>cost_type</i>	auto

Le impostazioni sopra riportate si riferiscono a:

- Solver NLP: è stato utilizzato ‘*sqp*’ come metodo di risoluzione, poiché il *Sequential Quadratic Programming* (SQP) è particolarmente efficace per problemi non lineari e con vincoli;
- Solver per la Programmazione Quadratica (QP): come solutore è stato scelto ‘*partial_condensing_hpipm*’. Questo approccio riduce la complessità computazionale sfruttando una rappresentazione compatta dell'orizzonte. Nella forma condensata stati e output vengono espressi in funzione degli input di controllo precedenti. Nella forma non condensata vengono esplicitati per ogni istante di tempo;
- Metodo di integrazione: è stato scelto ‘*irk*’ (Implicit Runge-Kutta) per la discretizzazione delle dinamiche non lineari del sistema; offre stabilità e precisione nelle simulazioni numeriche di sistemi rigidi;
- Numero massimo di iterazioni NLP: è stato limitato il numero massimo di iterazioni del solutore NLP a 20 per contenere i tempi computazionali;
- Tipo di vincoli e funzione di costo: il tipo di vincoli e la tipologia della funzione di costo sono stati impostati su ‘auto’, permettendo ad acados una selezione automatica delle opzioni più appropriate sulla base della struttura del problema [24].

2.8.2 Optimal Control Problem

In un problema di ottimizzazione è necessario definire le variabili di stato e le variabili di controllo. Una Matlab function è stata creata per riportare le informazioni essenziali che descrivono il suddetto problema. Queste devono essere definite nel linguaggio leggibile da acados. Gli stati scelti, la variabile di controllo, i parametri esterni e le variabili non lineari vengono riportati in Tabella 2.6.

Tabella 2.6: Definizione degli stati, variabile di controllo, parametri esterni e variabili non lineari

Parameter	Symbol	Value
Number of states	nx	4
Number of control input	nu	1
Number of external parameters	nw	3
Number of non-linear variables	nh	4

Gli stati scelti sono la distanza percorsa dall'ego vehicle x_h , la velocità dell'ego vehicle v_h , la distanza relativa tra l'ego vehicle e il veicolo che lo precede x_{rel} e la forza del motore elettrico effettivamente applicata F_{mc} .

L'input di controllo selezionato è la forza longitudinale richiesta al motore elettrico F_{req} , che può essere positiva o negativa a seconda che ci si trovi in una condizione di trazione o di frenata. Per quanto riguarda gli external parameters vengono considerati la posizione del veicolo precedente x_p , la velocità del veicolo precedente v_p e la variabile θ per rappresentare la pendenza della strada.

I termini di efficientamento energetico che entrano nella funzione di costo e che permettono lo sviluppo dell'Energy Efficient Adaptive Cruise Control sono la potenza del motore elettrico $P_{EM,cont}$, l'efficienza globale del powertrain Eff_{cont} , le perdite di potenza lungo la powertrain $P_{loss,cont}$ e l'energia per unità di distanza $E_{un,dist}$.

Infine, viene definito il numero di parametri:

$$np = nw + nwe + nb + (nmaxn + nminn + nmedn)$$

I termini nwe , nb , $nmaxn$, $nminn$, $nmedn$ fanno riferimento all'utilizzo di una rete neurale che verrà descritta successivamente.

In seguito, c'è bisogno di definire gli stati, l'azione di controllo e i parametri tramite variabili simboliche CasADi, riportate in Tabella 2.7. La definizione delle dinamiche del sistema in termini simbolici permette di applicare metodi di integrazione numerica, come quelli discussi nel capitolo 2.8.1, per ottenere una rappresentazione discreta del modello.

Tabella 2.7: Variabili simboliche CasADi

Parameter	Symbol	Value	Unit
Ego vehicle position	x_h	$SX.sym('x_h')$	[m]
Ego vehicle speed	v_h	$SX.sym('v_h')$	[m/s]
Relative displacement	x_{rel}	$SX.sym('x_{rel}')$	[m]
Effective actuated mechanical force	F_{mc}	$SX.sym('F_{mc}')$	[N]
Requested mechanical traction/braking force	F_{req}	$SX.sym('F_{req}')$	[N]
Preceding vehicle position	x_p	$SX.sym('x_p')$	[m]
Preceding vehicle speed	v_p	$SX.sym('v_p')$	[m/s]
Road slope	$teta$	$SX.sym('teta')$	[°]
Initialize All NN-WEIGHTS	we	$SX.sym('we', nwe, l)$	[-]
Initialize All NN-BIAS	b	$SX.sym('b', nb, l)$	[-]
Max NN inputs variables vector	mx_n	$SX.sym('mx_n', nmaxn, l)$	[-]
Min NN inputs variables vector	mn_n	$SX.sym('mn_n', nminn, l)$	[-]
Med NN inputs variables vector	md_n	$SX.sym('md_n', nmedn, l)$	[-]

Si è proseguito con la definizione dei vettori simbolici per rappresentare gli stati, le derivate degli stati, l'azione di controllo e i parametri del sistema.

Con ' sym_x ' è stato creato un vettore contenente le variabili simboliche degli stati del sistema.

$$sym_x = vertcat(x_h, v_h, x_{rel}, F_{mc});$$

Il vettore delle derivate degli stati permette di esprimere il cambiamento degli stati in funzione delle azioni di controllo.

$$sym_xdot = SX.sym('xdot', nx, l);$$

Poi sono stati rinominati la variabile di controllo e i parametri:

$$sym_u = vertcat(F_req);$$

$$sym_p = vertcat(x_p, v_p, teta, we, b, mx_n, mn_n, md_n);$$

Il modello interno utilizzato dal controllore MPC è usato per prevedere l'evoluzione degli stati futuri in base agli input di controllo e per calcolare la soluzione ottimale del problema di controllo predittivo. Le equazioni del modello interno potrebbero essere semplificate rispetto a quelle del *plant* per ridurre il carico computazionale e per permettere al controllore di operare in tempo reale.

Il *plant* rappresenta il modello realistico e completo del sistema fisico che viene simulato nell'ambiente Simulink per replicare il comportamento reale del veicolo. In questo lavoro le equazioni usate nel modello interno sono le stesse di quelle utilizzate nel *plant*.

La forza di resistenza al rotolamento:

$$\mathbf{F}_{roll} = m_{eq} * g * \cos(\theta_{rad}) * (f_0 + (f_2 * v_h^2)) \quad (2.22)$$

La forza di resistenza aerodinamica:

$$\mathbf{F}_{aero} = 0.5 * \rho * c_d * A * v_h^2 \quad (2.23)$$

La forza gravitazionale dovuta alla pendenza:

$$\mathbf{F}_{slope} = m_{eq} * g * \sin(\theta_{rad}) \quad (2.24)$$

Conversione dei gradi in radianti:

$$\theta_{rad} = \frac{\theta * \pi}{180} \quad (2.25)$$

La forza di resistenza totale:

$$\mathbf{F}_{res} = F_{roll} + F_{aero} + F_{slope} \quad (2.26)$$

La velocità relativa:

$$\mathbf{v}_{rel} = v_p - v_h \quad (2.27)$$

Si tiene conto del delay τ tra la forza richiesta al motore elettrico F_{req} e la forza meccanica effettivamente applicata F_{mc} .

La derivata della forza meccanica applicata:

$$\mathbf{F}_{mc,d} = \frac{(F_{req} - F_{mc})}{\tau} \quad (2.28)$$

L'accelerazione dell'ego vehicle:

$$\mathbf{a}_h = (F_{mc} - F_{res})/m_{eq} \quad (2.29)$$

La velocità angolare del motore elettrico:

$$\mathbf{w}_{mc,EM} = \left(\frac{i_{tot}}{R_w}\right) * v_h \quad (2.30)$$

La velocità di rotazione del motore elettrico:

$$\mathbf{w}_{mc,EM,rpm} = \mathbf{w}_{mc,EM} * \left(\frac{30}{\pi}\right) \quad (2.31)$$

La coppia in uscita dal motore elettrico:

$$\mathbf{T}_{mc,EM} = \left(\frac{R_w}{i_{tot}}\right) * F_{mc} \quad (2.32)$$

La potenza in uscita dal motore elettrico:

$$\mathbf{P}_{mc,EM} = \mathbf{T}_{mc,EM} * \mathbf{w}_{mc,EM} \quad (2.33)$$

La potenza all'inverter:

$$\mathbf{P}_{inv} = \mathbf{P}_{mc,EM} * \left(eta_{tot}^{-sign_app(T_{mc,EM})}\right) \quad (2.34)$$

Le perdite di potenza lungo la powertrain:

$$\mathbf{P}_{loss} = \mathbf{P}_{inv} - \mathbf{P}_{mc,EM} \quad (2.35)$$

Ora verranno riportate le equazioni presenti nell'internal model che descrivono la rete neurale che ha l'obiettivo di fornire l'efficienza globale della powertrain in funzione dei valori di coppia e velocità di rotazione del motore. Verrà considerato il rendimento dell'inverter ($\eta_{inv} = 0.98$), il rendimento del motore elettrico $\eta_{EM} = (T_{EM}, w_{EM})$ e il rendimento del gearbox $\eta_{gearbox} = (T_{EM}, w_{EM})$.

La rete neurale è caratterizzata da un input layer con due nodi, uno per i valori di coppia motore T_{EM} e uno per i valori di velocità di rotazione del motore, $w_{EM,rpm}$. Poi sono presenti due hidden layers con 16 nodi ciascuno. La struttura della rete neurale è riportata in Tabella 2.8. Ogni nodo applica una funzione di attivazione per modellare relazioni non lineari tra gli input e l'output. Infine, si ha un nodo di output che restituisce l'efficienza globale. Nella fase di training la rete viene addestrata attraverso i valori di coppia e velocità di rotazione del motore a cui corrispondono determinati valori di efficienza presi dalla mappa sperimentale.

L'obiettivo del processo di training è ottimizzare i pesi (weights) e i bias della rete neurale per minimizzare l'errore tra i valori di efficienza previsti dalla rete e quelli sperimentali. Questo processo avviene attraverso un algoritmo di ottimizzazione (ad esempio l'algoritmo di retro-

propagazione, *backpropagation*), che aggiorna iterativamente i weights e i bias per migliorare le prestazioni della rete.

Tabella 2.8: Struttura della rete neurale

Parameter	Symbol	Value
Number of inputs to the Neural Network	n_inputs	2
Nodes in the first hidden layer	n_layer1	16
Nodes in the second hidden layer	n_layer2	16
Nodes in the Output layer	$n_outputs$	1

Successivamente viene definito nwe , rappresentante il numero totale di pesi nella rete neurale. Essi vengono calcolati rispettivamente alla connessione tra input layer e primo hidden layer, tra primo e secondo hidden layer, tra secondo hidden layer e output layer.

$$nwe = (n_inputs * n_layer1 + n_layer1 * n_layer2 + n_layer2 * n_outputs);$$

Poi viene definito nb , rappresentante il numero totale di bias nella rete. Ogni nodo nei livelli nascosti e nell'output layer ha un bias.

$$nb = (n_layer1 + n_layer2 + n_outputs);$$

Ogni input w_{EM} , T_{EM} verrà normalizzato. Quindi vengono definite le dimensioni dei vettori di normalizzazione.

$$\begin{aligned} nmaxn &= n_inputs; \\ nminn &= n_inputs; \\ nmedn &= n_inputs; \end{aligned}$$

$nmaxn$, $nminn$, $nmedn$ indicano il numero di elementi nei vettori di normalizzazione. Poiché ci sono due ingressi, ciascun vettore ha due elementi.

I vettori successivi contengono i valori sperimentali utilizzati per normalizzare gli input della rete.

$$\begin{aligned} max_norm &= norm_vect.max_norm'; \\ min_norm &= norm_vect.min_norm'; \\ med_norm &= norm_vect.med_norm'; \end{aligned}$$

Poi si hanno gli input data in ingresso alla rete neurale:

$$data = [w_mc_EM_rpm; T_mc_EM];$$

In seguito, vengono definiti i pesi, we , e i bias, b , della rete neurale per ogni layer.

$we1$ è la matrice dei pesi per il primo hidden layer con dimensione $(n_{inputs} * n_{layer,1})$.

$$we1 = reshape(sym_p(nw+1:nw+n_inputs*n_layer1),n_layer1,n_inputs);$$

$we2$ è la matrice dei pesi per il secondo hidden layer con dimensione $(n_{inputs} * n_{layer,2})$.

$$we2 = reshape(sym_p(nw+n_inputs*n_layer1+1:nw+n_inputs*n_layer1+n_layer1*n_layer2),n_layer2,n_layer1);$$

$we3$ è la matrice dei pesi per l'output layer con dimensione $(n_{layer,2} * n_{outputs})$.

$$we3 = reshape(sym_p(nw+n_inputs*n_layer1+n_layer1*n_layer2+1:nw+n_inputs*n_layer1+n_layer1*n_layer2+n_layer2*n_outputs),n_outputs,n_layer2);$$

$b1$, $b2$, $b3$ rappresentano i bias per il primo, secondo hidden layer e per l'output layer. Sono vettori di dimensione pari al numero di nodi del layer corrispondente.

$$\begin{aligned} b1 &= sym_p(nw+nwe+1:nw+nwe+n_layer1); \\ b2 &= sym_p(nw+nwe+n_layer1+1:nw+nwe+n_layer1+n_layer2); \\ b3 &= sym_p(nw+nwe+n_layer1+n_layer2+1:nw+nwe+n_layer1+n_layer2+n_outputs); \end{aligned}$$

I dati di input alla rete vengono normalizzati per migliorare il training. Per la normalizzazione si ha il vettore dei valori massimi, minimi e medi.

$$\begin{aligned} mx_n &= sym_p(nw+nwe+nb+1:nw+nwe+nb+nmaxn); \\ mn_n &= sym_p(nw+nwe+nb+nmaxn+1:nw+nwe+nb+nmaxn+nminn); \\ md_n &= sym_p(nw+nwe+nb+nmaxn+nminn+1:nw+nwe+nb+nmaxn+nminn+nmedn); \end{aligned}$$

Si va a normalizzare gli input in un intervallo [0-1], utile per le funzioni di attivazione della rete.

$$NNin = 2.*(data - md_n)./(mx_n - mn_n);$$

Viene poi definito il primo hidden layer:

$$hd1 = we1 * NNin + b1;$$

Al vettore $hd1$ viene applicata una funzione di attivazione (è stata creata questa funzione di attivazione siccome di default non era presente) che permette di dare una forma logaritmica. Infatti, vedendo la mappa sperimentale lungo l'asse delle velocità di rotazione si nota un andamento logaritmico.

$$hd1 = \log(abs(hd1)+1);$$

Viene definito il secondo hidden layer:

$$hd2 = we2 * hd1 + b2;$$

La seconda funzione di attivazione, applicata al secondo hidden layer, è la funzione Swish che permette di introdurre nella rete delle non linearità. Effettivamente la mappa di efficienza è altamente non lineare.

$$hd2 = hd2. * (1. / (1 + \exp(-hd2)));$$

Infine, viene valutata l'efficienza totale. La somma pesata degli output del secondo hidden layer, $hd2$, viene effettuata tramite i pesi $w3$ e $b3$. Poi, il valore minimo per l'efficienza viene impostato a 0.1.

$$\begin{aligned} eta_tot &= we3 * hd2 + b3; \\ eta_tot &= \max(eta_tot, 0.1); \end{aligned}$$

Successivamente, vengono riportati i quattro termini di efficientamento energetico. La potenza del motore elettrico normalizzata:

$$P_{EM,cont} = \frac{P_{inv}}{P_{EM,max}} \quad (2.36)$$

L'efficienza globale della driveline:

$$Eff_{cont} = 1 - eta_{tot} \quad (2.37)$$

La perdita di potenza lungo la powertrain normalizzata:

$$P_{loss,cont} = \frac{P_{loss}}{P_{loss,max}} \quad (2.38)$$

L'energia per unità di distanza:

$$E_{un,dist} = \left(\frac{T_{mc,EM} * i_{tot}}{R_w * N * 3600} \right) * eta_{tot}^{-sign_app(T_{mc,EM})} \quad (2.39)$$

Alcune variabili sono state normalizzate in modo tale da ottenere delle grandezze comparabili. Le variabili utilizzate per la normalizzazione sono riportate in Tabella 2.9. Questo favorisce il controllore dal punto di vista computazionale e semplifica anche il lavoro dell'algoritmo di ottimizzazione dei pesi che ha il compito di definire dei pesi adeguati per i termini inseriti nella

funzione di costo, a seconda delle performance che si vuole ottenere. Della modalità in cui i pesi sono stati ottenuti tramite l'algoritmo di ottimizzazione ne parleremo successivamente.

Tabella 2.9: Variabili di normalizzazione inerenti la potenza del motore elettrico e la perdita di potenza

Parameter	Symbol	Value	Unit
Maximum electric motor Power output	$P_{EM,max}$	134000	[W]
Maximum value of Power losses	$P_{loss,max}$	31302	[W]

Di seguito viene spiegato come è stato ricavato il termine dell'energia per unità di distanza valutato in $\left[\frac{kWh}{km}\right]$.

$$E_{unit\ distance,k} = \frac{E'}{v_h dT N} = \frac{P_{EM} dT}{v_h dT \eta_{tot} N} = \frac{T_{EM} \omega_{EM}}{v_h \eta_{tot} N} = \frac{T_{EM} i_{tot} v_h}{v_h \eta_{tot} N R} = \frac{T_{EM} i_{tot}}{\eta_{tot} N R 3600} \quad (2.40)$$

È stata considerata l'energia spesa per ogni time step del prediction horizon (tenendo conto anche del rendimento globale) rapportata alla distanza percorsa, relativa allo specifico time step. È stata aggiunta la variabile N, che rappresenta il numero di step del prediction horizon, in modo da ottenere un'energia valutata a ogni step del prediction horizon. Poi, integrando tutti i contributi, lungo il prediction horizon, si ottiene un'energia lungo l'orizzonte di predizione. Ogni time step che caratterizza il prediction horizon viene valutato tenendo conto del tempo e del numero di passi che caratterizza l'orizzonte di predizione, come riportano i dati di Tabella 2.9.

$$dT = \frac{t_{pred}}{N} \quad (2.41)$$

Tabella 2.10: Dati relativi al prediction horizon

Parameter	Symbol	Value	Unit
Prediction horizon	t_{pred}	4	[s]
Prediction horizon steps	N	20	[-]

La funzione *sign_app* viene utilizzata per approssimare la funzione *sign(x)* che non è una funzione continua.

$$sign_app = \tanh(50 * x) \quad (2.42)$$

Dopo aver definito le equazioni del modello dinamico è necessario definirne la forma esplicita ed implicita, definire le variabili che devono essere vincolate, e definire le variabili che compongono la funzione di costo.

- La forma esplicita, '*expr_f_expl*', rappresenta le equazioni di stato in una struttura semplice, che permette di calcolare i valori di ogni variabile di stato in base agli input e agli stati correnti.

$$expr_f_expl = [v_h; a_h; v_rel; F_mc_d];$$

- La forma implicita, '*expr_f_impl*', è ottenuta sottraendo il vettore delle derivate di stato '*sym_xdot*'. La forma implicita è utile per quei solutori che richiedono la descrizione delle dinamiche attraverso vincoli di uguaglianza tra derivate e funzioni di stato.

$$expr_f_impl = expr_f_expl - sym_xdot;$$

Successivamente vengono definiti i vincoli imposti lungo il percorso, '*expr_h*', e al passo finale, '*expr_h_e*', della predizione.

La velocità dell'ego vehicle, la distanza relativa, la forza longitudinale effettivamente applicata, l'accelerazione dell'ego vehicle, la derivata della forza longitudinale, la forza richiesta al motore elettrico, e i quattro termini di efficientamento: potenza del motore elettrico, efficienza globale, perdite di potenza della powertrain, energia per unità di distanza; tutte queste variabili devono rispettare determinate condizioni, che verranno poi definite, lungo tutto il percorso.

All'ultimo step della predizione non si vincola la variabile di controllo F_{req} perché non si ha interesse che venga valutato un ulteriore input di controllo in quanto la predizione è terminata. Inoltre, non vengono vincolate l'accelerazione dell'ego vehicle a_h e la derivata della forza meccanica $F_{mc,d}$ per non appesantire la formulazione.

$$expr_h = [v_h; x_rel; F_mc; a_h; F_mc_d; F_req; P_EM_cont; Eff_cont; P_loss_cont; E_un_dist];$$

$$expr_h_e = [v_h; x_rel; F_mc; P_EM_cont; Eff_cont; P_loss_cont; E_un_dist];$$

In seguito, viene impostata la struttura per la funzione di costo. Essa contiene dei termini di penalità che il solutore cerca di minimizzare. Include variabili di stato e di controllo. La funzione di costo è così costruita:

- *cost_expr_y_0*: variabili che partecipano alla funzione di costo al primo step dell'orizzonte predittivo;

$$cost_expr_y_0 = vertcat(v_h, F_mc, a_h, F_mc_d, F_req, P_EM_cont, Eff_cont, P_loss_cont, E_un_dist);$$

- *cost_expr_y*: variabili che partecipano alla funzione di costo lungo l'intero orizzonte predittivo;

$$cost_expr_y = vertcat(v_h, F_mc, a_h, F_mc_d, F_req, P_EM_cont, Eff_cont, P_loss_cont, E_un_dist);$$

- *cost_expr_y_e*: variabili che caratterizzano la funzione di costo all'ultimo step della predizione. In questo caso, saranno escluse le stesse variabili non vincolate all'ultimo step: forza richiesta al motore elettrico (variabile di controllo), accelerazione dell'ego vehicle, derivata della forza meccanica effettivamente applicata.

$$cost_expr_y_e = vertcat(v_h, F_mc, P_EM_cont, Eff_cont, P_loss_cont, E_un_dist);$$

Nella funzione di costo entrano i termini di accelerazione dell'ego vehicle e della derivata della forza meccanica effettivamente applicata (che ne rappresenta la sua variazione) con lo scopo di minimizzare le oscillazioni nella velocità.

Si riporta l'equazione che caratterizza la funzione di costo per una miglior comprensione dal punto di vista matematico. Rispettivamente lungo l'orizzonte di predizione e all'ultimo step.

$$\begin{aligned} J = \sum_{k=1}^{N-1} [& (v_{h,k} - v_{ref,h,k})^2 * W_{v_h} + (F_{mc,k} - F_{mc,ref,k})^2 * W_{F_{mc}} + (a_{h,k} - a_{ref,h,k})^2 * \\ & W_{a_h} + (F_{mc,d,k} - F_{mc,d})^2 * W_{F_{mc,d}} + (F_{req,k}^2 * W_{F_{req}}) + (P_{EM,cont,k} - P_{EM,cont,ref,k})^2 * \\ & W_{P_{EM,cont}} + (Eff_{cont,k} - Eff_{cont,ref,k})^2 * W_{Eff_{cont}} + (P_{loss,cont,k} - P_{loss,cont,ref,k})^2 * \\ & W_{P_{loss,cont}} + (E_{un,dist,k} - E_{un,dist,ref,k})^2 * W_{E_{un,dist}}] \end{aligned} \quad (2.43)$$

$$\begin{aligned} J_{final} = \sum_{k=N} [& (v_{h,k} - v_{ref,h,k})^2 * W_{v_h} + (F_{mc,k} - F_{mc,ref,k})^2 * W_{F_{mc}} + (P_{EM,cont,k} - \\ & P_{EM,cont,ref,k})^2 * W_{P_{EM,cont}} + (Eff_{cont,k} - Eff_{cont,ref,k})^2 * W_{Eff_{cont}} + (P_{loss,cont,k} - \\ & P_{loss,cont,ref,k})^2 * W_{P_{loss,cont}} + (E_{un,dist,k} - E_{un,dist,ref,k})^2 * W_{E_{un,dist}}] \end{aligned} \quad (2.44)$$

A ogni termine presente nella funzione di costo deve essere attribuita una determinata importanza in funzione del comportamento che si vuole ottenere dal sistema.

Perciò vengono definiti i pesi W_{v_h} , $W_{F_{mc}}$, W_{a_h} , $W_{F_{mc,d}}$, $W_{F_{req}}$, $W_{P_{EM,cont}}$, $W_{Eff_{cont}}$, $W_{P_{loss,cont}}$, $W_{E_{un,dist}}$.

Nella funzione di costo non viene coinvolto lo stato relativo alla posizione assunta dall'ego vehicle, x_h , perché il veicolo deve essere libero di percorrere la distanza che vuole, per cui non la si vincola.

Il termine di distanza relativa viene controllato tramite l'impostazione dei vincoli di distanza relativa minima e di distanza relativa massima come vedremo successivamente. Inserendo la distanza relativa nella funzione di costo, il controllore farebbe rispettare all'ego vehicle la distanza che si andrebbe a imporre come riferimento, questo vorrebbe dire che l'ego vehicle sarebbe costretto a inseguire il veicolo di fronte indipendentemente dalla velocità che quest'ultimo andrebbe ad assumere. Questa cosa non è voluta in quanto tramite l'Adaptive Cruise Control si vuole far seguire all'ego vehicle una certa velocità imposta dall'utente, se poi si presenterà un veicolo di fronte entrerà in gioco la distanza minima di sicurezza che dovrà essere rispettata.

È necessario, poi, definire le matrici contenenti i pesi:

```

W_x = diag([W_v_h, W_F_mc]);
W_u = diag([W_a_h, W_F_mc_d, W_F_req, W_P_EM_cont, W_Eff_cont, W_P_loss_cont,
W_E_un_dist]);
W = blkdiag(W_x, W_u);
W_0 = W;
W_e = blkdiag(W_x, W_u(4:end,4:end));

weight_path =
[W_v_h;0;0;0;0;0;0;0;0;0;W_F_mc;0;0;0;0;0;0;0;0;W_a_h;0;0;0;0;0;0;0;0;W_F_m
c_d;0;0;0;0;0;0;0;0;0;W_F_req;0;0;0;0;0;0;0;0;W_P_EM_cont;0;0;0;0;0;0;0;0;W_
Eff_cont;0;0;0;0;0;0;0;0;0;W_P_loss_cont;0;0;0;0;0;0;0;0;W_E_un_dist];

weight_term =
[W_v_h;0;0;0;0;0;0;0;W_F_mc;0;0;0;0;0;0;W_P_EM_cont;0;0;0;0;0;0;W_Eff_cont;0;0;0;
0;0;0;W_P_loss_cont;0;0;0;0;0;0;W_E_un_dist];

```

La matrice W raccoglie i pesi di tutti i termini che compongono la funzione di costo in una struttura combinata (utilizzando *blkdiag*), andando a rappresentare la matrice di costo complessiva per l'intero orizzonte predittivo. W_0 e W_e rappresentano i pesi applicati alla funzione di costo al primo e ultimo step del prediction horizon. Nella matrice W_e vengono esclusi i pesi relativi all'accelerazione dell'ego vehicle, alla derivata della forza longitudinale e alla forza richiesta al controllore.

I vettori *'weight_path'* e *'weight_term'* rappresentano le componenti della matrice dei pesi *W* rispettivamente lungo il percorso e al passo finale. Questi vettori sono input specifici per il blocco S-Function che gestisce il controllore MPC in Simulink.

Di seguito vengono riportati i pesi associati alle slack variables. Le slack variables sono state introdotte ai seguenti stati: velocità dell'ego vehicle v_h , distanza relativa rispetto al veicolo precedente x_{rel} , forza longitudinale effettivamente applicata F_{mc} . Questo consente di ottenere dei soft constraints e quindi avere una flessibilità riguardo i limiti imposti sui vincoli rigidi, garantendo che il sistema rimanga all'interno di una tolleranza prestabilita, anche in condizioni operative estreme.

Il vettore z_l contiene i pesi lineari per le slack variables, come mostra Tabella 2.11. I valori sono impostati a zero per evitare costi aggiuntivi sulle deviazioni minori rispetto i limiti. Questo vale anche per gli altri termini lineari z_u , z_{l_e} , z_{u_e} .

Tabella 2.11: Pesi lineari per le slack variables

Parameter	Value
zl	$0.*[1e4, 1e5, 1e2]'$
zu	zl
zl_e	zl
zu_e	zu

Per quanto riguarda i termini quadratici sono state create delle matrici diagonali contenenti i pesi definiti in Tabella 2.12. 10^4 è il peso della slack variable riferito alla velocità dell'ego vehicle; 10^5 è il peso della slack variable riferito alla distanza relativa; 10^2 è il peso della slack

variable riferito alla forza longitudinale effettivamente applicata. C'è una forte penalizzazione riguardo le violazioni sui vincoli di distanza relativa e di velocità dell'ego vehicle.

Tabella 2.12: Termini quadratici per le slack variables

Parameter	Value
Zl	$diag([1e4, 1e5, 1e2])$
Zu	Zl
Zl_e	Zl
Zu_e	Zu

Poi c'è una sezione del codice che si occupa di popolare la struttura '*model_eacc*' con le informazioni necessarie alla configurazione e parametrizzazione del modello di veicolo utilizzato nell'Energy Efficient Adaptive Cruise Control.

Vengono ripresi gli stati, la variabile di controllo, i parametri esterni, le variabili non lineari e i parametri che definiscono la rete neurale.

```

model_eacc.nx = nx;
model_eacc.nu = nu;
model_eacc.nw = nw;
model_eacc.np = np;
model_eacc.nh = nh;
model_eacc.nwe = nwe;
model_eacc.nb = nb;
model_eacc.n_inputs = n_inputs;
model_eacc.n_layer1 = n_layer1;
model_eacc.n_layer2 = n_layer2;
model_eacc.n_outputs = n_outputs;
model_eacc.nmaxn = nmaxn;
model_eacc.nminn = nminn;
model_eacc.nmedn = nmedn.

```

Vengono riprese il numero di variabili che andranno a definire i vincoli:

```

model_eacc.nbx = nbx;
model_eacc.nbu = nbu;
model_eacc.nbh = nbh;
model_eacc.nbh_e = nbh_e;
model_eacc.nsbx = nsbx;
model_eacc.nsbh = nsbh;
model_eacc.nsbh_e = nsbh_e.

```

Vengono ripresi i termini costituenti la funzione di costo:

```

model_eacc.ny_0 = ny_0;
model_eacc.ny = ny;
model_eacc.ny_e = ny_e.

```

Vengono ripresi i vettori normalizzati:

```
model_eacc.max_norm = max_norm;  
model_eacc.min_norm = min_norm;  
model_eacc.med_norm = med_norm.
```

Vengono ripresi i dati del veicolo e i parametri che caratterizzano l'MPC:

```
model_eacc.N = N;  
model_eacc.dT = dT;  
model_eacc.M_v = M_v;  
model_eacc.J_em = J_em;  
model_eacc.J_g1 = J_g1;  
model_eacc.J_g2 = J_g2;  
model_eacc.Rw = Rw;  
model_eacc.J_w = J_w;  
model_eacc.i_tot = i_tot;  
model_eacc.rho = rho;  
model_eacc.g = g;  
model_eacc.c_d = c_d;  
model_eacc.A = A;  
model_eacc.taum = taum;  
model_eacc.f0 = f0;  
model_eacc.f2 = f2;  
model_eacc.M_eq = M_eq;  
model_eacc.P_EM_max = P_EM_max;  
model_eacc.d_min = d_min;  
model_eacc.t_h = t_h;
```

Vengono ripresi i vettori simbolici che rappresentano gli stati, le derivate degli stati, l'azione di controllo e i parametri del sistema:

```
model_eacc.sym_x = sym_x;  
model_eacc.sym_xdot = sym_xdot;  
model_eacc.sym_u = sym_u;  
model_eacc.sym_p = sym_p;
```

Vengono riprese le equazioni del modello dinamico definite nella forma esplicita ed implicita:

```
model_eacc.expr_f_expl = expr_f_expl;  
model_eacc.expr_f_impl = expr_f_impl;
```

Vengono ripresi i vincoli imposti lungo il percorso e all'ultimo step della predizione:

```
model_eacc.expr_h = expr_h;  
model_eacc.expr_h_e = expr_h_e;
```

Viene ripresa la struttura che definisce la funzione di costo:

```
model_eacc.cost_expr_y_0 = cost_expr_y_0;  
model_eacc.cost_expr_y = cost_expr_y;  
model_eacc.cost_expr_y_e = cost_expr_y_e;
```

Vengono riprese le matrici dei pesi e gli elementi che le compongono:

```
model_eacc.W_0 = W_0;  
model_eacc.W = W;  
model_eacc.W_e = W_e;  
model_eacc.W_x = W_x;  
model_eacc.weight_path = weight_path;  
model_eacc.weight_term = weight_term;  
model_eacc.W_v_h = W_v_h;  
model_eacc.W_F_mc = W_F_mc;  
model_eacc.W_a_h = W_a_h;  
model_eacc.W_F_mc_d = W_F_mc_d;  
model_eacc.W_F_req = W_F_req;  
model_eacc.W_P_EM_cont = W_P_EM_cont;  
model_eacc.W_Eff_cont = W_Eff_cont;  
model_eacc.W_P_loss_cont = W_P_loss_cont;  
model_eacc.W_E_un_dist = W_E_un_dist;
```

Vengono riprese le slack variables definite su tre stati:

```
model_eacc.zl = zl;  
model_eacc.zu = zu;  
model_eacc.zl_e = zl_e;  
model_eacc.zu_e = zu_e;  
model_eacc.Zl = Zl;  
model_eacc.Zu = Zu;  
model_eacc.Zl_e = Zl_e;  
model_eacc.Zu_e = Zu_e;
```

La rete neurale è stata utilizzata per ottenere una rappresentazione fedele della mappa di efficienza del motore elettrico. Infatti, come si vede in Figura 2.7, la superficie ottenuta con la Neural Network copia bene la superficie che si ottiene tramite l'interpolazione dei dati sperimentali.

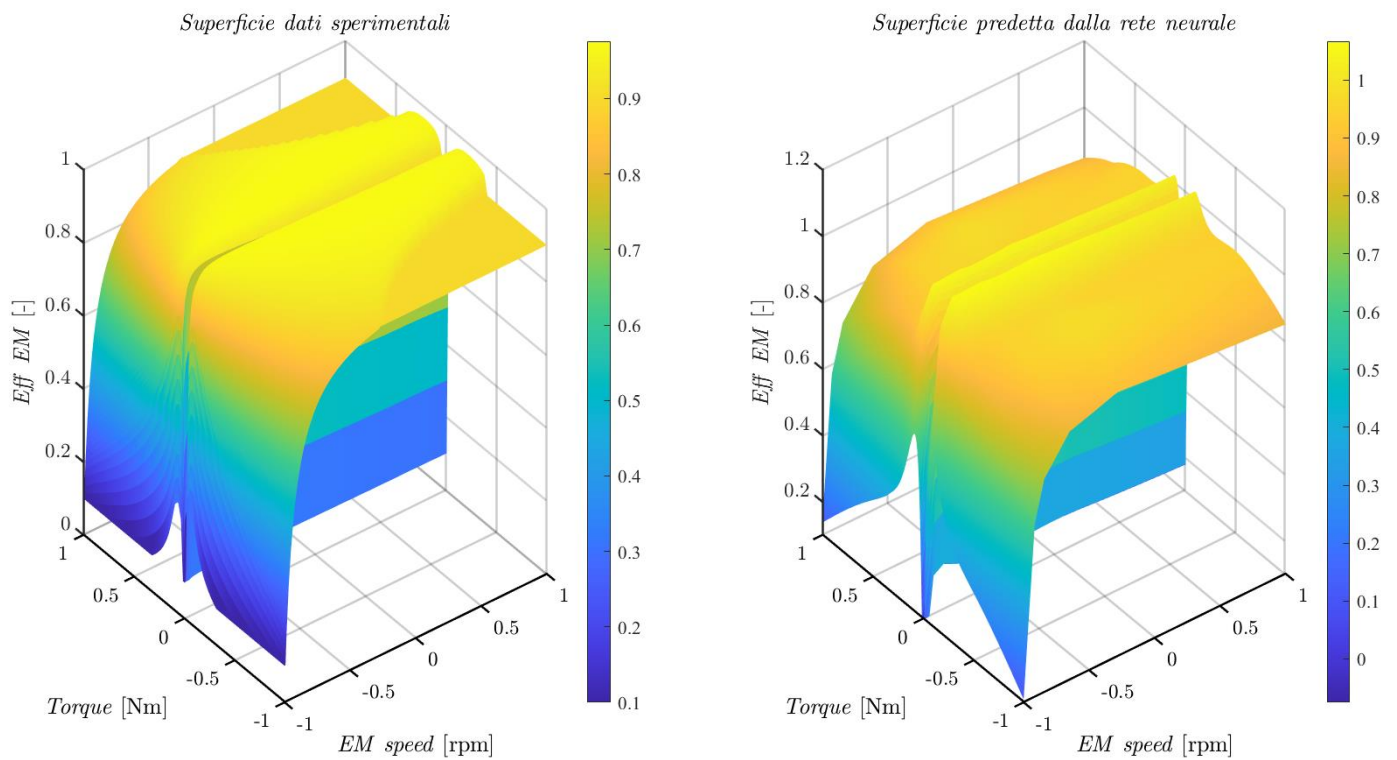


Figura 2.7: A sinistra la mappa di efficienza del motore elettrico ottenuta con i dati sperimentali; a destra la mappa di efficienza del motore elettrico ottenuta tramite la rete neurale

Inizialmente, prima dell'implementazione della rete neurale, si aveva provato a replicare la superficie ottenuta con i dati sperimentali tramite una funzione polinomiale di ottavo grado. I risultati, però, non erano stati soddisfacenti come si può vedere in Figura 2.8.

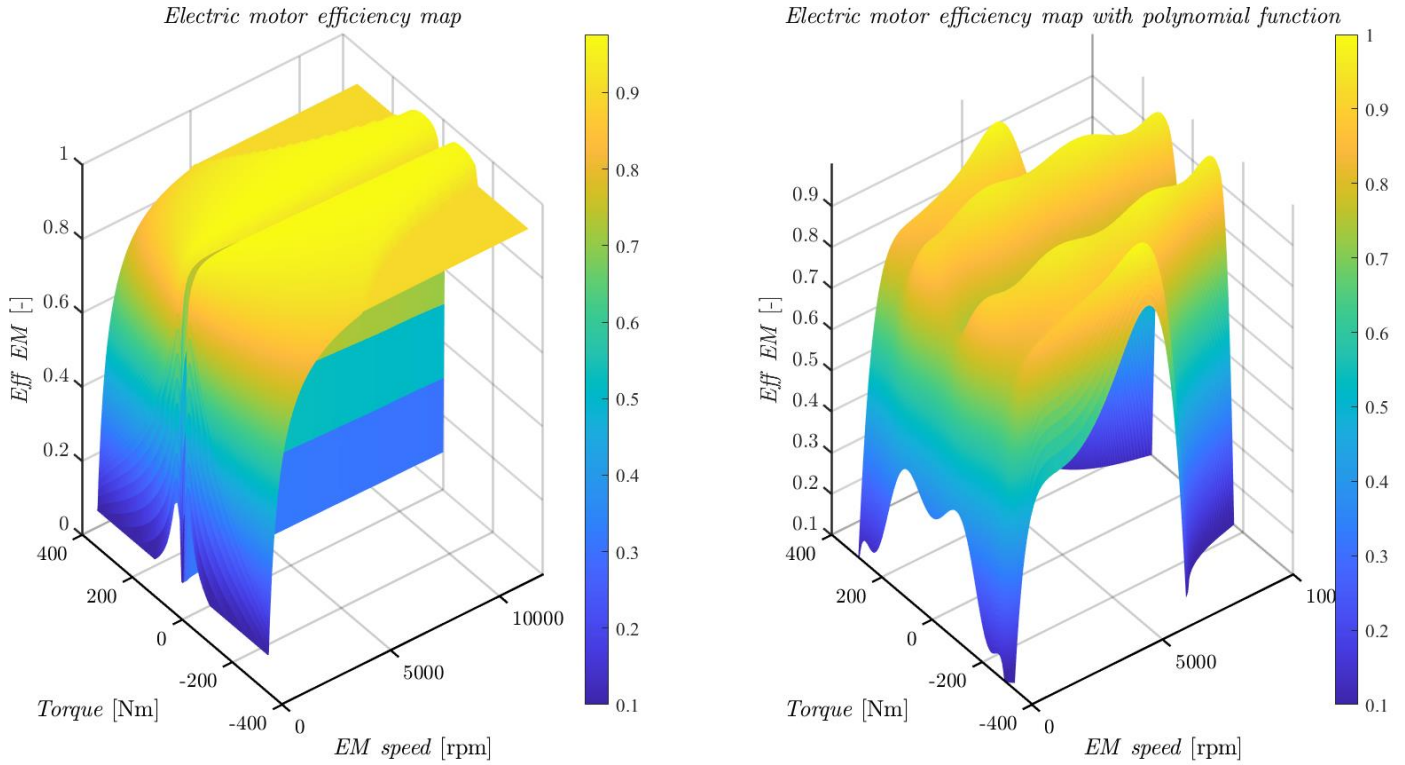


Figura 2.8: A sinistra la mappa di efficienza del motore elettrico ottenuta con i dati sperimentali; a destra la mappa di efficienza del motore elettrico ottenuta tramite la funzione polinomiale

Lo sviluppo della funzione polinomiale era avvenuto in questo modo:
Infittimento dei vettori di velocità di rotazione e di coppia del motore elettrico.

```
w_eff_interp_EM = linspace(min(w_eff),max(w_eff),1000).';
T_eff_interp_EM = linspace(min(T_eff),max(T_eff),300).';
```

Definizione di una griglia con i punti di velocità di rotazione e di coppia del motore e interpolazione di essi.

```
[X_interp,Y_interp] = meshgrid(w_eff_interp_EM,T_eff_interp_EM);
Z_interp_EM =
interp2(w_vec_EM_eff,T_vec_EM_eff,Eff_mat_EM_eff,X_interp,Y_interp,'linear');
Z_interp_EM_mat =
reshape(Z_interp_EM_vec,[size(Z_interp_EM,1),size(Z_interp_EM,2)]);
```

Lo stesso procedimento era stato effettuato per i valori di efficienza del gearbox, in modo poi da definire l'efficienza globale della powertrain considerando anche il rendimento dell'inverter $\eta_{inv} = 0.98$.

```
Z_interp_DRIVE = (Z_interp_GEAR_mat.*Z_interp_EM_mat)*Z_inver;
```

Infine, si va a introdurre la funzione polinomiale di ottavo grado. La funzione *polyfitn* genera il polinomio che approssima l'efficienza della powertrain in funzione di coppia e velocità di rotazione del motore.

```
Z_interp_DRIVE_vec = Z_interp_DRIVE(:);
p = polyfitn([X_interp_vec,Y_interp_vec],Z_interp_DRIVE_vec,8);
c = p.Coefficients;
f = @(x,y) c(1)*x.^8 + c(2)*x.^7.*y + c(3)*x.^7 + c(4)*x.^6.*y.^2 + c(5)*x.^6.*y +
c(6)*x.^6 + c(7)*x.^5.*y.^3 + c(8)*x.^5.*y.^2 + c(9)*x.^5.*y + c(10)*x.^5 + ...
c(11)*x.^4.*y.^4 + c(12)*x.^4.*y.^3 + c(13)*x.^4.*y.^2 + c(14)*x.^4.*y + ...
c(15)*x.^4 + c(16)*x.^3.*y.^5 + c(17)*x.^3.*y.^4 + c(18)*x.^3.*y.^3 + ...
c(19)*x.^3.*y.^2 + c(20)*x.^3.*y + c(21)*x.^3 + c(22)*x.^2.*y.^6 + ...
c(23)*x.^2.*y.^5 + c(24)*x.^2.*y.^4 + c(25)*x.^2.*y.^3 + c(26)*x.^2.*y.^2 + ...
c(27)*x.^2.*y + c(28)*x.^2 + c(29)*x.*y.^7 + c(30)*x.*y.^6 + ...
c(31)*x.*y.^5 + c(32)*x.*y.^4 + c(33)*x.*y.^3 + c(34)*x.*y.^2 + ...
c(35)*x.*y + c(36)*x + c(37)*y.^8 + c(38)*y.^7 + c(39)*y.^6 + c(40)*y.^5 + ...
c(41)*y.^4 + c(42)*y.^3 + c(43)*y.^2 + c(44)*y + c(45);
```

La stesura del codice che caratterizza il problema di ottimizzazione continua con la definizione del numero di porte di ingresso e di uscita che deve avere il blocco S-function, che rappresenta il controllore in Simulink.

Si vanno, perciò, a definire nell'ordine le porte di ingresso all'S-function, riportate in Tabella 2.13:

- due input che riguardano l'ingresso degli stati che provengono dal *plant* e che vengono valutati a ogni step della simulazione;
- un input che riguarda i parametri provenienti dal blocco *parameters*;
- tre input delle variabili di riferimento;
- quattro input che rappresentano i vincoli massimi e minimi degli stati lungo l'orizzonte di predizione e all'ultimo step della predizione;
- due input relativi al vincolo massimo e minimo riferito alla variabile di controllo lungo l'orizzonte di predizione;
- quattro input che riguardano i vincoli massimi e minimi legati alle quattro variabili di efficientamento lungo il prediction horizon e all'ultimo step. Inoltre, due input inerenti i vincoli massimi e minimi riguardanti l'accelerazione dell'ego vehicle e la derivata della forza meccanica, lungo il prediction horizon, perché all'ultimo step queste due variabili non entrano nella funzione di costo;
- tre input legati alle componenti che popolano la matrice dei pesi globale. Al primo step della predizione, come durante tutto il prediction horizon entrano in gioco tutti i pesi appartenenti ai termini della funzione di costo; all'ultimo step entrano in gioco tutti i pesi tranne quelli relativi alla variabile di controllo W_{F_req} , all'accelerazione dell'ego vehicle W_{a_h} e alla derivata della forza meccanica W_{F_mc}, d ;
- un input relativo all'inizializzazione del risolutore alla prima esecuzione;
- un'ultima porta di ingresso che definisce lo stato iniziale del sistema in modo che la simulazione parta da una condizione ben definita.

Tabella 2.13: Porte di ingresso dell'S-function

Parameter	Value
<i>simulink_opts.inputs.lbx_0</i>	1
<i>simulink_opts.inputs.ubx_0</i>	1
<i>simulink_opts.inputs.parameter_traj</i>	1
<i>simulink_opts.inputs.y_ref_0</i>	1
<i>simulink_opts.inputs.y_ref</i>	1
<i>simulink_opts.inputs.y_ref_e</i>	1
<i>simulink_opts.inputs.lbx</i>	1
<i>simulink_opts.inputs.ubx</i>	1
<i>simulink_opts.inputs.lbx_e</i>	1
<i>simulink_opts.inputs.ubx_e</i>	1
<i>simulink_opts.inputs.lbu</i>	1
<i>simulink_opts.inputs.ubu</i>	1
<i>simulink_opts.inputs.lg</i>	1
<i>simulink_opts.inputs.ug</i>	1
<i>simulink_opts.inputs.lh</i>	1
<i>simulink_opts.inputs.uh</i>	1
<i>simulink_opts.inputs.lh_e</i>	1
<i>simulink_opts.inputs.uh_e</i>	1
<i>simulink_opts.inputs.cost_W_0</i>	1
<i>simulink_opts.inputs.cost_W</i>	1
<i>simulink_opts.inputs.cost_W_e</i>	1
<i>simulink_opts.inputs.reset_solver</i>	1
<i>simulink_opts.inputs.x_init</i>	1

Per quanto riguarda le porte di uscita si ha:

- la prima uscita del blocco S-function rappresenta l'azione di controllo che viene inviata al plant;
- un output che rappresenta la traiettoria delle azioni di controllo ottimizzate nel corso della simulazione;
- un output relativo alla traiettoria degli stati del sistema nel tempo;
- un output che indica il valore della funzione di costo ottimizzata dal controllore;
- un output che fornisce il residuo KKT su tutta la traiettoria, utile per valutare il livello di accuratezza e l'aderenza ai vincoli del problema di ottimizzazione risolto.

Esse vengono riportate in Tabella 2.14.

Tabella 2.14: Porte di uscita dell'S-function

Parameter	Value
<i>simulink_opts.outputs.utraj</i>	1
<i>simulink_opts.outputs.xtraj</i>	1
<i>simulink_opts.outputs.cost_value</i>	1
<i>simulink_opts.outputs.KKT_residual</i>	0
<i>simulink_opts.outputs.KKT_residuals</i>	1

Nel codice vengono ripresi i parametri di discretizzazione dell'orizzonte di predizione definiti nella Matlab function "*model.eacc*" e di cui si è discusso in precedenza.

$$\begin{aligned} N &= \text{model_eacc}.N; \\ dT &= \text{model_eacc}.dT; \end{aligned}$$

Viene poi definito il tempo che caratterizza il prediction horizon. In questo caso, siccome si assume $N = 20$ step nel prediction horizon e $dT = 0.2$ s come singolo time step, si avrà $T = 4$ s.

$$T = dT * N \quad (2.45)$$

Modificare il time step dT ha ripercussioni sulla frequenza di aggiornamento del controllo.

Impostando un time step caratterizzato da $dT > 0.2$ s si ha un vantaggio che riguarda il minore carico computazionale. Si hanno meno aggiornamenti del controllo in un dato intervallo di tempo, riducendo il numero di problemi di ottimizzazione da risolvere. Poi si avrà un prediction horizon più lungo che permette al sistema di prevedere il comportamento futuro su un periodo più ampio, utile per scenari con variazioni lente (ad esempio la guida in autostrada).

Lo svantaggio è rappresentato dal fatto che con time step maggiore, la rete di controllo risponde meno frequentemente, introducendo possibili ritardi nell'adattamento alle condizioni di guida. Inoltre, si riduce la precisione del modello, poiché variazioni rapide del sistema non vengono catturate accuratamente.

Contrariamente diminuire il time step a $dT < 0.2$ s, porta ad avere una maggior precisione e un controllo più reattivo. Si hanno predizioni più frequenti che permettono di catturare dinamiche veloci e rispondere rapidamente a variazioni improvvise, come frenate improvvise del veicolo davanti. Un aggiornamento più rapido consente al sistema di adattarsi meglio a situazioni variabili. Lo svantaggio riguarda l'aumento del carico computazionale, in quanto riducendo il time step si aumenta il numero di calcoli in un intervallo di tempo. Inoltre, si ridurrà il tempo totale del prediction horizon, limitando la capacità di predire comportamenti futuri a lungo termine [17].

In seguito, viene definita una parte di codice relativa ai parametri iniziali e impostazioni temporali, fondamentali alla simulazione del sistema nell'ambiente Simulink.

- $x0 = [0; 0; 200; 0]$: rappresenta il vettore che contiene gli stati iniziali all'interno del modello interno del controllore. In ordine si ha: la posizione iniziale dell'ego vehicle, la velocità iniziale dell'ego vehicle, la distanza relativa rispetto al veicolo che precede e la forza longitudinale effettivamente applicata;
- $tsim = 500$ s: è la durata complessiva della simulazione per valutare la risposta del controllore (questo valore varierà a seconda dello scenario che si andrà a testare);
- $dt_sim = 0.001$ s: è il tempo di campionamento per la simulazione in Simulink. Un passo di 1 ms garantisce un'alta risoluzione temporale, che permette una valutazione accurata della risposta dinamica del sistema;
- $time = (0:dt_sim:tsim)'$: crea un vettore temporale che rappresenta la serie temporale della simulazione, da $t = 0$ a $t = tsim$ con intervallo di 1 ms. Questo vettore è utilizzato per definire gli intervalli temporali su cui vengono registrati i dati durante la simulazione.

Si definisce, poi, i valori di riferimento rispetto alle variabili che entrano nella funzione di costo, essenziali affinché il sistema insegua l'obiettivo stabilito. Quindi vengono definiti i riferimenti per tutta la durata della simulazione, rispettivamente per la velocità dell'ego vehicle, per la forza longitudinale, per l'accelerazione dell'ego vehicle, per la derivata della forza longitudinale, per la forza richiesta al controllore. Vi sono altri quattro riferimenti relativi ai termini di efficientamento: la potenza del motore elettrico, l'efficienza globale della powertrain, le perdite di potenza lungo la powertrain, l'energia per unità di distanza. Essi vengono riportati in Tabella 2.15.

La velocità di riferimento v_{h_ref} che si andrà a inserire nel codice sarà un valore costante (uno scalare) oppure variabile (considerando un vettore di velocità) a seconda dei test che vengono effettuati. Nei test finali, come riferimento, verranno forniti dei vettori di velocità in modo che il veicolo segua un dato profilo di velocità. In quest'ultimo caso si avrà una velocità di riferimento variabile in base alla parte di ciclo guida che il veicolo sta affrontando.

Tabella 2.15: Valori di riferimento delle variabili che compongono la funzione di costo

Parameter	Symbol	Value	Unit
Actuated reference mechanical force	F_{mc_ref}	0	[N]
Acceleration reference	a_{h_ref}	0	[m/s ²]
Variation of the force	$F_{mc_d_ref}$	0	[N/s]
Requested reference EM force	F_{req_ref}	0	[N]
Reference normalized electric motor power	$P_{EM_cont_ref}$	0	[-]
Reference Efficiency contribution	Eff_{cont_ref}	0	[-]
Reference normalized power losses	$P_{loss_cont_ref}$	0	[-]
Energy per unit distance	$E_{un_dist_ref}$	0	[kWh/km]

I valori dei riferimenti, eccetto quello della velocità, sono impostati a zero perché idealmente si vorrebbe che il controllore applicasse all'ego vehicle la minima forza longitudinale, la minima potenza al motore elettrico, la minima energia in funzione della distanza percorsa; si vorrebbe avere perdite di potenza vicine allo zero per ottimizzare i consumi. I riferimenti impostati a zero per l'accelerazione e la derivata della forza longitudinale per minimizzare le oscillazioni nella velocità dell'ego vehicle. L'efficienza impostata a zero perché nella funzione di costo si

considera il termine $(1 - \eta_{a_{tot}})$, esso si vuole minimizzarlo al fine di massimizzare il rendimento globale.

In seguito, vengono creati dei vettori contenenti i valori di riferimento caratterizzanti il primo step, gli step intermedi e l'ultimo step della predizione. Essi saranno utilizzati durante l'impostazione del Simulink.

$$\begin{aligned}
y_{ref_0} &= [v_{h_ref}; F_{mc_ref}; a_{h_ref}; F_{mc_d_ref}; F_{req_ref}; P_{EM_cont_ref}; \\
&\quad Eff_{cont_ref}; P_{loss_cont_ref}; E_{un_dist_ref}]; \\
y_{ref} &= [v_{h_ref}; F_{mc_ref}; a_{h_ref}; F_{mc_d_ref}; F_{req_ref}; P_{EM_cont_ref}; \\
&\quad Eff_{cont_ref}; P_{loss_cont_ref}; E_{un_dist_ref}]; \\
y_{ref_e} &= [v_{h_ref}; F_{mc_ref}; P_{EM_cont_ref}; Eff_{cont_ref}; P_{loss_cont_ref}; \\
&\quad E_{un_dist_ref}];
\end{aligned}$$

Si prosegue con l'assegnazione dei vincoli relativi agli stati e alla variabile di controllo sull'orizzonte di predizione al fine di mantenere il sistema entro i limiti desiderati. I dati sono riportati in Tabella 2.16. Come detto in precedenza non si definisce nessun vincolo sulla posizione assunta dall'ego vehicle perché non si ha interesse che venga rispettata una certa distanza percorsa. Gli stati sono caratterizzati da soft constraints, permettendo, quindi, una certa flessibilità al vincolo, rispetto al valore massimo e minimo definito. La variabile di controllo è caratterizzata da un hard constraint in quanto si ha un limite operativo del motore elettrico in un range di coppia massima e minima. Hard constraints sono applicati anche per i quattro termini di efficientamento, per l'accelerazione dell'ego vehicle e per la derivata della forza longitudinale.

Tabella 2.16: Vincoli sugli stati, sulla variabile di controllo e sulle variabili che compongono la funzione di costo

Parameter	Symbol	Value	Unit
Maximum vehicle speed	v_{h_max}	36.9	[m/s]
Minimum vehicle speed	v_{h_min}	0	[m/s]
Maximum distance between host and preceding vehicle	x_{rel_max}	1000	[m]
Minimum distance between host and preceding vehicle	x_{rel_min}	10	[m]
Maximum actuated force	F_{mc_max}	5500	[N]
Minimum actuated force	F_{mc_min}	-5500	[N]
Maximum acceleration	a_{h_max}	5.8949	[m/s ²]

Minimum acceleration	a_{h_min}	-8.8874	[m/s ²]
Maximum variation of the force	$F_{mc_d_max}$	200000	[N/s]
Minimum variation of the force	$F_{mc_d_min}$	-200000	[N/s]
Maximum requested force	F_{req_max}	5500	[N]
Minimum requested force	F_{req_min}	-5500	[N]
Maximum Normalized electric motor power	$P_{EM_cont_max}$	1	[-]
Minimum Normalized electric motor power	$P_{EM_cont_min}$	0	[-]
Maximum Efficiency contribution	Eff_{cont_max}	1	[-]
Minimum Efficiency contribution	Eff_{cont_min}	0	[-]
Maximum Normalized power loss	$P_{loss_cont_max}$	1	[-]
Minimum Normalized power loss	$P_{loss_cont_min}$	0	[-]
Maximum Energy for unit distance	$E_{un_dist_max}$	20	[kWh/km]
Minimum Energy for unit distance	$E_{un_dist_min}$	-20	[kWh/km]

Il valore impostato sul limite massimo di velocità proviene dal limite che si ha sul numero di giri massimo del motore elettrico (che determina la velocità angolare massima del motore), dal rapporto di trasmissione e dal raggio della ruota.

Per quanto riguarda la distanza relativa massima non si è interessati a rispettarne una perché il veicolo che precede può allontanarsi di quanto vuole. È stato impostato un valore elevato in quanto a cosa vuole che sia impostata necessariamente una variabile, senza ometterla. La distanza relativa minima, come distanza di sicurezza, è stata impostata a 10 m; successivamente nell'ambiente Simulink il vincolo sulla distanza relativa minima da rispettare verrà sovrascritto. Infatti, si vuole ottenere una distanza dal veicolo precedente che tenga conto di una distanza minima $d0 = 10m$ più un termine di spazio ($v_h * t_h$) che tenga conto della velocità dell'ego

vehicle e dell'headway time, cioè il tempo in cui l'ego vehicle percorrerebbe l'attuale distanza se la sua velocità rimanesse costante [18], [29].

I limiti sulla forza longitudinale sono impostati in funzione della coppia massima e minima del motore elettrico, rapporto di trasmissione e raggio della ruota.

I limiti sull'accelerazione dell'ego vehicle sono impostati in funzione dei limiti sulla forza longitudinale, sulle forze di resistenza (forza di rotolamento e aerodinamica; della forza di resistenza dovuta alla pendenza della strada non se ne tiene conto, siccome la pendenza è impostata a zero e il termine della forza va con la funzione $\sin(\theta)$) legate ai valori assunti dalla velocità, e alla massa del veicolo.

I limiti imposti sulla variazione della forza longitudinale sono lasciati ampi perché comunque si è visto che il controllore si comporta bene; quindi, non sono stati applicati vincoli stringenti. Infine, i vincoli sulla potenza del motore elettrico, efficienza globale, perdite di potenza che sono stati normalizzati, rispetto al valore massimo assunto, sono stati impostati tra 0 e 1; mentre all'energia per unità di distanza è stata lasciata libertà non restringendo troppo il vincolo.

All'ultimo step della predizione vengono impostati i vincoli in Tabella 2.16 con esclusione del vincolo sull'accelerazione dell'ego vehicle, sulla derivata della forza longitudinale e sulla variabile di controllo, ossia la forza richiesta al controllore.

Si ha poi la configurazione del tipo di vincolo che viene applicato al modello di controllo predittivo durante l'orizzonte di previsione e all'ultimo step.

```
ocp_model.set('constr_type', constr_type);  
ocp_model.set('constr_type_e', constr_type);
```

L'impostazione *constr_type* a ogni istante dell'orizzonte di predizione è stata impostata su *auto*, permettendo ad acados una selezione automatica delle opzioni più appropriate sulla base della struttura del problema.

Per quanto riguarda gli stati viene inizializzata la matrice *Jbx* come matrice di zeri di dimensione (nbx, nx) . $nbx = 3$ è il numero di vincoli, $nx = 4$ è il numero di stati. Si imposta la matrice in modo da attivare i vincoli solo su determinati stati.

In seguito, vengono assegnati i valori ai vincoli specificati in *Jbx*. Quindi si avranno i lower boundary *lbx* e gli upper boundary *ubx* che rappresentano i vettori dei limiti inferiori e superiori per ogni variabile vincolata.

Le righe di codice *ocp_model.set* permettono di assegnare, lungo l'orizzonte di predizione e all'ultimo step, i limiti imposti sui vincoli, *lbx*, *ubx*, *lbx_e*, *ubx_e*, alle matrici di vincoli *Jbx*, *Jbx_e*.

```

Jbx = zeros(nbx,nx);
Jbx(1,2) = 1;
Jbx(2,3) = 1;
Jbx(3,4) = 1;
lbx = [v_h_min; x_rel_min; F_mc_min];
ubx = [v_h_max; x_rel_max; F_mc_max];
ocp_model.set('constr_Jbx', Jbx);
ocp_model.set('constr_lbx', lbx);
ocp_model.set('constr_ubx', ubx);
ocp_model.set('constr_Jbx_e', Jbx);
ocp_model.set('constr_lbx_e', lbx);
ocp_model.set('constr_ubx_e', ubx);

```

Lo stesso concetto viene applicato per la variabile di controllo. Viene inizializzata la matrice *Jbu* come matrice di zeri di dimensione (*nbu, nu*). *nbu* = 1 rappresenta l'unico vincolo, *nu* = 1 rappresenta l'unica variabile di controllo. Successivamente, viene assegnato un valore specifico nella matrice per attivare il vincolo sulla variabile di controllo.

Si definiranno il lower boundary *lbu* e l'upper boundary *ubu* che rappresentano i termini riguardanti il limite inferiore e superiore per la variabile vincolata.

Le righe di codice *ocp_model.set* permettono di assegnare, lungo l'orizzonte di predizione, i limiti imposti *lbu, ubu*, alla matrice dei vincoli *Jbu*.

```

Jbu = zeros(nbu,nu);
Jbu(1,1) = 1;
lbu = F_req_min;
ubu = F_req_max;
ocp_model.set('constr_Jbu', Jbu);
ocp_model.set('constr_lbu', lbu);
ocp_model.set('constr_ubu', ubu);

```

C'è poi una sezione di codice che definisce i vincoli non lineari nel modello di controllo predittivo. *expr_h* ed *expr_h_e* specificano le espressioni dei vincoli non lineari lungo l'orizzonte di predizione e all'ultimo step. Lungo l'orizzonte e con l'applicazione dei relativi limiti inferiori *lh* e superiori *uh*, si ritrovano gli stati, la variabile di controllo, l'accelerazione dell'ego vehicle, la derivata della forza longitudinale e le quattro variabili di efficientamento che caratterizzano il modello.

All'ultimo step della predizione si ritrovano i tre stati e le quattro variabili di efficientamento con l'applicazione dei limiti inferiori *lh_e* e dei limiti superiori *uh_e*.

```

ocp_model.set('constr_expr_h', model_eacc.expr_h);
lh = [v_h_min, x_rel_min, F_em_min, F_req_min]';
uh = [v_h_max, x_rel_max, F_em_max, F_req_max]';
ocp_model.set('constr_lh', lh);
ocp_model.set('constr_uh', uh);
ocp_model.set('constr_expr_h_e', model_eacc.expr_h_e);
lh_e = [v_h_min_e, x_rel_min_e, F_em_min_e]';
uh_e = [v_h_max_e, x_rel_max_e, F_em_max_e]';
ocp_model.set('constr_lh_e', lh_e);
ocp_model.set('constr_uh_e', uh_e);

```

In un'altra sezione del codice vengono applicate le slack variables sui seguenti stati: velocità dell'ego vehicle v_h , distanza relativa rispetto al veicolo precedente x_{rel} , forza longitudinale applicata F_{mc} . Questo consente di ottenere dei soft constraints e, quindi, avere una flessibilità riguardo i limiti rigidi imposti sui vincoli.

Come fatto in precedenza, viene inizializzata la matrice $Jsbx$ come matrice di zeri di dimensione $(nbx, nsbx)$. $nbx = 3$ è il numero di vincoli soft, $nsbx = 3$ è il numero di stati a cui viene applicato il vincolo soft. Successivamente la matrice viene popolata per assegnare agli stati del sistema le relative slack variables.

Le righe di codice `ocp_model.set` assegnano le matrici delle slack variables 'Jsbx' e 'Jsbx_e' all'orizzonte di predizione e all'ultimo step.

```

Jsbx = zeros(nbx,nsbx);
Jsbx(1,1) = 1;
Jsbx(2,2) = 1;
Jsbx(3,3) = 1;
ocp_model.set('constr_Jsbx', Jsbx);
ocp_model.set('constr_Jsbx_e', Jsbx);

```

In seguito, vengono definiti i pesi associati alle slack variables distinguendo tra termini lineari e quadratici. Il termine lineare cerca di limitare le deviazioni minime del sistema, mentre il termine quadratico penalizza le deviazioni maggiori. L'applicazione di questi pesi permette di bilanciare l'importanza nel rispetto dei vincoli.

I termini lineari comprendono:

```

ocp_model.set('cost_zl', zl);
ocp_model.set('cost_zu', zu);

```

I termini quadratici comprendono:

```

ocp_model.set('cost_Zl', Zl);
ocp_model.set('cost_Zu', Zu);

```

I termini lineari all'ultimo step della predizione comprendono:

```
ocp_model.set('cost_zl_e', zl_e);  
ocp_model.set('cost_zu_e', zu_e);
```

I termini quadratici all'ultimo step della predizione comprendono:

```
ocp_model.set('cost_Zl_e', Zl_e);  
ocp_model.set('cost_Zu_e', Zu_e);
```

Nella parte finale del codice vengono prelevati da un file esterno i parametri della rete neurale addestrata che serviranno alla costruzione di un vettore che verrà utilizzato nel blocco Parameters dell'ambiente Simulink, al primo step della simulazione.

Viene effettuata un'estrazione dei weights e bias dai layer della rete:

```
we1 = reshape(double(net.Layers(2,1).Weights),n_inputs*n_layer1,1);  
we2 = reshape(double(net.Layers(4,1).Weights),n_layer1*n_layer2,1);  
we3 = reshape(double(net.Layers(6,1).Weights),n_layer2*n_outputs,1);  
weights = [we1; we2; we3];  
bi1 = double(net.Layers(2,1).Bias);  
bi2 = double(net.Layers(4,1).Bias);  
bi3 = double(net.Layers(6,1).Bias);  
bias = [bi1; bi2; bi3];
```

net.Layers(2,1).Weights, accede ai pesi del secondo layer (primo hidden layer) della rete. Questi pesi sono rappresentati come una matrice ($n_{\text{layer1}} * n_{\text{inputs}}$) definita come il numero di nodi del layer rispetto agli input.

we2, we3 rappresentano rispettivamente i pesi del quarto layer (secondo hidden layer) e i pesi del sesto layer (output layer). Infine, i pesi della rete vengono concatenati nel vettore *weights*. L'estrazione dei bias avviene allo stesso modo: *bi1, bi2, bi3* accedono al secondo, quarto, sesto layer (che rappresentano rispettivamente primo, secondo hidden layer e output layer); in seguito vengono concatenati nel vettore *bias*. Poi vengono ripresi i vettori normalizzati, definiti in precedenza nella Matlab function e viene creato un unico vettore *NNfactors* contenente tutti i parametri necessari per utilizzare la rete neurale.

```
max_norm = model_eacc.max_norm;  
min_norm = model_eacc.min_norm;  
med_norm = model_eacc.med_norm;  
NNfactors = [weights; bias; max_norm; min_norm; med_norm];
```

Come ultima cosa, viene definito un vettore dei parametri iniziali necessario per dare l'avvio alla simulazione nell'ambiente Simulink. Si inizializza un vettore colonna di dimensioni $((N + 1) * np)$, in cui $(N + 1)$ è il numero di passi lungo il prediction horizon e np rappresenta il numero di parametri da considerare (vettore definito in precedenza nella Matlab function).

```
Param_init = zeros((N+1)*np,1);
```

Viene creato un vettore temporale caratterizzante l'orizzonte predittivo, da 0 fino a T (lunghezza dell'orizzonte temporale), con incrementi di dT , che rappresentano la durata di ciascun passo di previsione.

$$t_init = (0:dT:T)';$$

Con $x_p_prev_0$ si calcola la posizione iniziale del veicolo precedente aggiungendo la distanza relativa iniziale alla posizione iniziale del veicolo. Vengono utilizzati i valori definiti nel vettore delle condizioni iniziali $x0$.

$$x_p_prev_0 = (x0(1,1)+x0(3,1)).*ones(length(t_init),1);$$

x_p_prev , invece, rappresenta la posizione stimata del veicolo precedente in ciascun passo lungo l'orizzonte di previsione.

$$x_p_prev = x_p_prev_0 + (v_prec(1,1)).*t_init);$$

Si definisce il vettore che rappresenta la pendenza della strada. Impostato a zero lungo l'intero orizzonte di previsione per semplicità.

$$teta_sim = 0.*ones(length(time_1),1);$$

Infine, tramite il ciclo *for* si riempie il vettore $Param_init$ con la posizione e velocità del veicolo precedente, la pendenza della strada e il vettore dei parametri della rete neurale. Il vettore viene riempito a blocchi. Ogni blocco di 4 righe viene riempito con i parametri per ciascun istante temporale, per $(N+1)$ volte.

```
for i = 1:1:(N+1)
    Param_init((np*(i-1)+1):(np*i),1) = [x_p_prev(i); v_prec(i); teta_sim(1,1);
    NNfactors];
end
```

3. Modello Simulink

Il modello Simulink è caratterizzato da due blocchi principali: EACC e Plant come illustrato nella Figura 3.1.

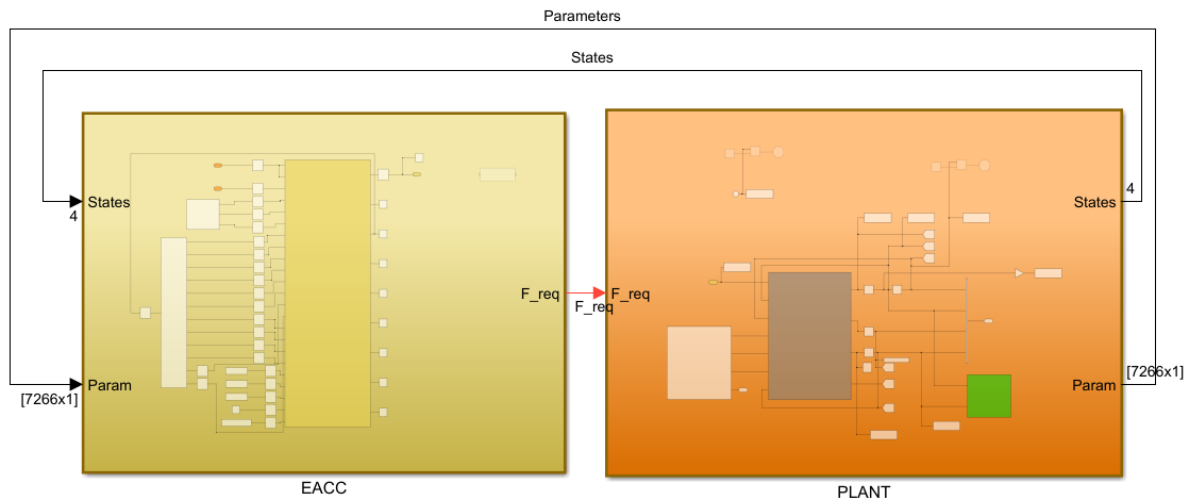


Figura 3.1: Panoramica sul modello Simulink

Nel blocco *EACC* si trova il controllore dell'Energy Efficient Adaptive Cruise Control. Da esso provengono le azioni di controllo in cui la forza richiesta al motore elettrico viene inviata al modello di veicolo definito nel *Vehicle Plant*, all'interno del blocco *Plant*. Dal *Vehicle Plant* provengono gli stati che vengono valutati a ogni step della simulazione e vengono mandati al controllore affinché esso calcoli la forza longitudinale di controllo successiva. Inoltre, è presente il blocco *Parameters* che fornisce al *Vehicle Plant* le informazioni relative alla posizione del veicolo precedente x_p , velocità del veicolo precedente v_p e pendenza della strada θ . Il controllore è rappresentato dal blocco S-function, il quale, come detto in precedenza (nella parte di codice dedicata alla definizione del numero di porte di ingresso e di uscita), è caratterizzato da 23 porte di ingresso e 9 porte di uscita, come mostra la Figura 3.2.

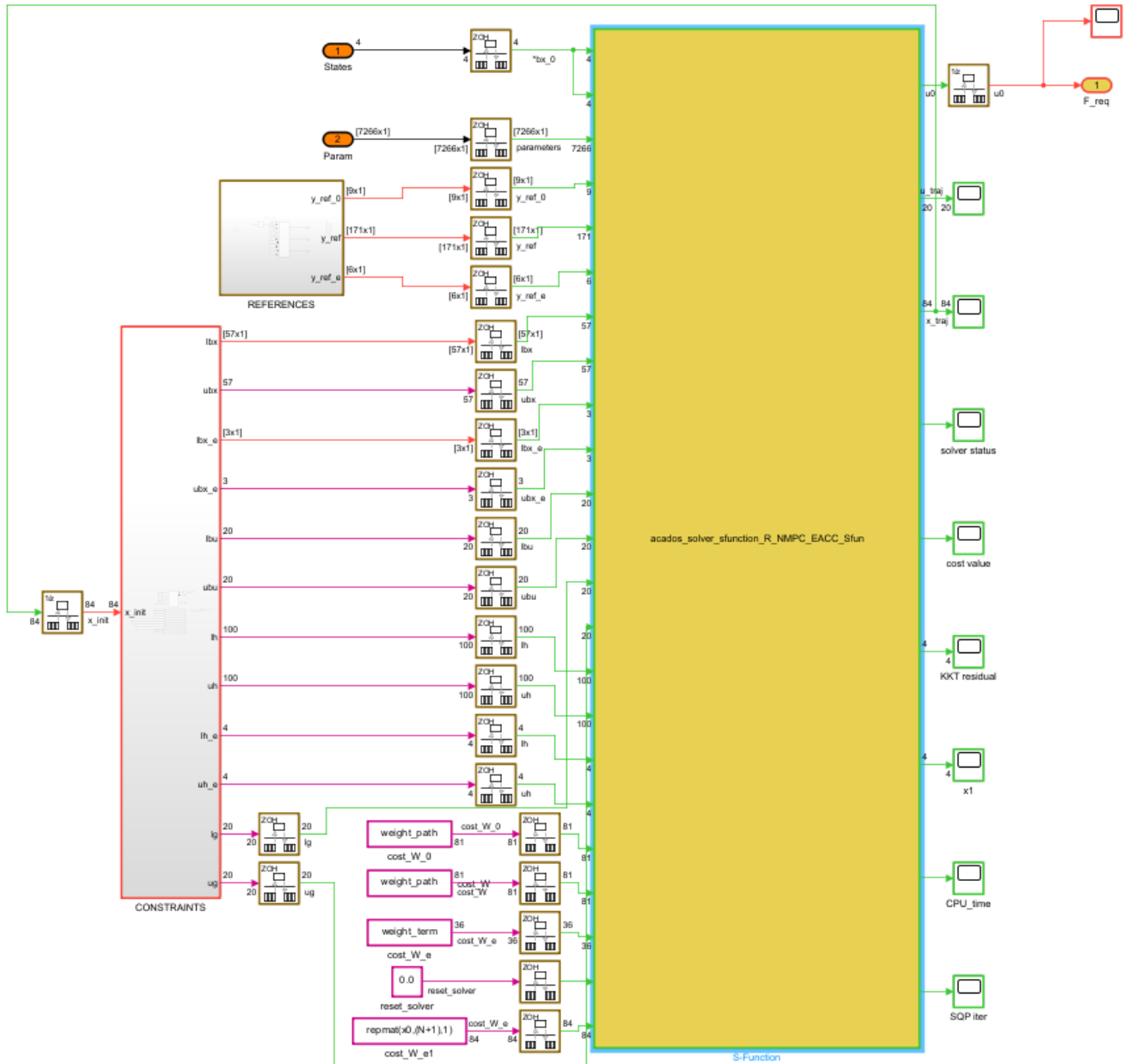


Figura 3.2: Blocco EACC contenente il controllore

Degli output, si è principalmente interessati al primo che fornisce l'azione di controllo e al terzo che contiene la traiettoria degli stati del sistema nel tempo. Siccome il vincolo sulla distanza minima da rispettare viene sovrascritto per tenere conto di una distanza variabile in funzione della velocità dell'ego vehicle, dal vettore contenente la traiettoria degli stati viene prelevato il valore di velocità dell'ego vehicle predetto durante la simulazione. Esso viene inviato, successivamente, al blocco *Constraints* in cui vengono assegnati i vincoli che il sistema deve rispettare. Il blocco *Constraints* viene presentato in Figura 3.3.

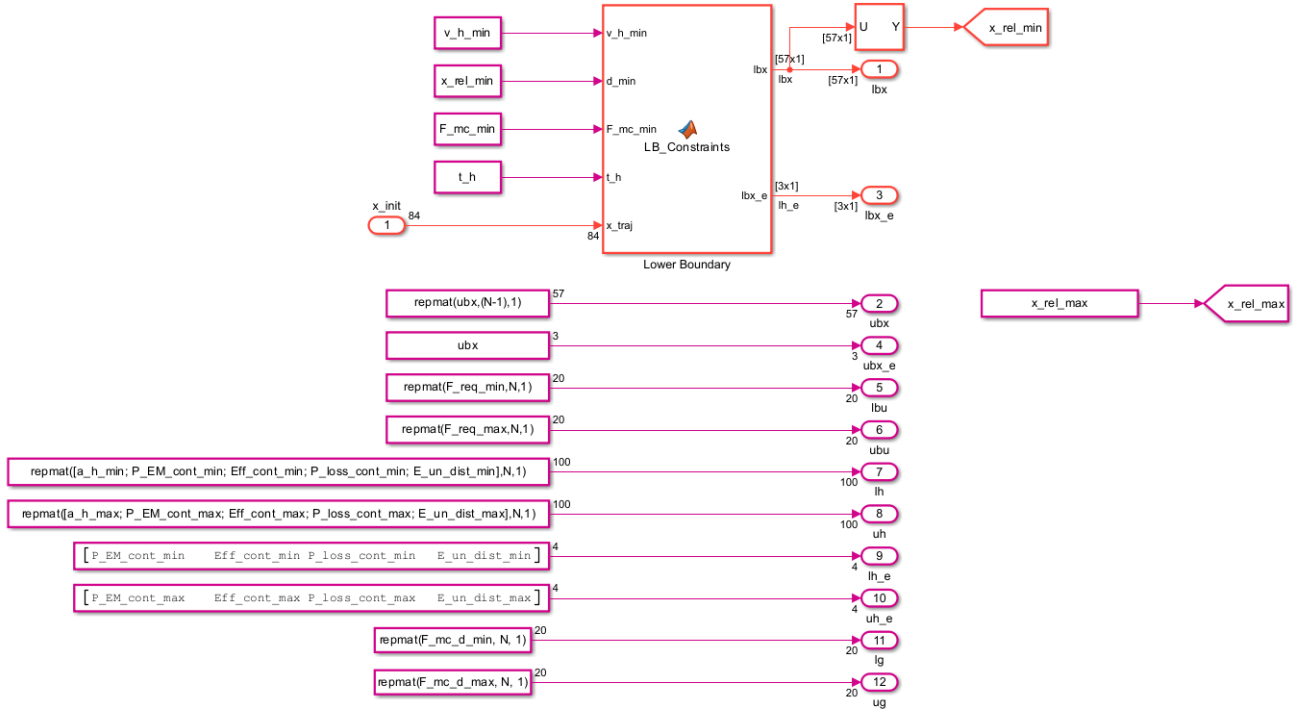


Figura 3.3: Blocco in cui vengono assegnati i vincoli (Constraints)

In primis si ha la funzione *LB_Constraints*, tramite la quale vengono applicati i lower boundary *lbx*, *lbx_e* lungo il prediction horizon e all'ultimo step. Si ha come input i seguenti parametri:

- v_h_min : velocità minima consentita per l'ego vehicle;
- d_min : distanza minima di sicurezza;
- F_mc_min : forza minima attuabile dal sistema di controllo;
- t_h : headway time (tempo di reazione);
- x_traj : vettore degli stati previsti dal controller MPC lungo l'orizzonte di predizione.

Il codice definito nella funzione prevede innanzitutto l'estrazione degli elementi corrispondenti alla velocità dell'ego vehicle v_h dal vettore degli stati previsti x_traj , considerando solo le velocità previste dal secondo step fino allo step (N+1).

$$v_h_prev = x_traj(6:4:end,1);$$

Poi viene creato il vettore d_min_vect di lunghezza pari al vettore di velocità v_h_prev contenente i valori di distanza minima.

$$d_min_vect = d_min.*ones(length(v_h_prev),1);$$

In seguito, viene definita la distanza relativa minima x_rel_min tramite i due vettori sopra definiti, il cui valore viene aggiornato in funzione della velocità assunta dall'ego vehicle. Questo garantisce che la distanza relativa sia sempre sufficientemente grande per evitare collisioni.

$$x_rel_min = d_min_vect + (t_h.*v_h_prev);$$

Poi si inizializza il vettore dei vincoli inferiori lbx che caratterizzano gli stati lungo l'orizzonte di predizione. Viene impostato un ciclo *for* per ogni passo dell'orizzonte (dal primo al penultimo passo) e vengono assegnati i vincoli inferiori a blocchi di tre. Il vettore ripetuto 19 volte ha la seguente forma: [velocità minima dell'ego vehicle; distanza relativa minima; forza attuata minima]. Quindi l'outport 1 è caratterizzato da 57 elementi (19*3).

```

    lbx = zeros((N-1)*3,1);
    for i = 1:1:(N-1)
        lbx(3*i-2:3*i,1) = [v_h_min; x_rel_min(i); F_mc_min];
    end

```

Infine, il vettore di 3 elementi lbx_e contiene i vincoli inferiori applicati all'ultimo passo della predizione.

```

    lbx_e = [v_h_min; x_rel_min(end); F_mc_min];

```

La stessa logica viene applicata per i vincoli superiori. La funzione *repmat* effettua la ripetizione per 19 volte del vettore ubx contenente $[v_h_max; x_rel_max; F_mc_max]$ lungo l'orizzonte di predizione.

Identico procedimento per il vettore dei vincoli superiori all'ultimo step della predizione, ubx_e .

Successivamente sono stati impostati i vincoli inerenti alla variabile di controllo F_{req} . Si è interessati ad avere il vincolo solo lungo l'orizzonte di predizione; quindi, vengono definiti i vettori lbu e ubu , contenenti i 20 vincoli inferiori e superiori lungo gli step del prediction horizon.

In seguito, vengono applicati i vincoli alle variabili mancanti. Lungo il prediction horizon con lh e uh si ripetono per 20 volte i limiti inferiori e superiori relativi all'accelerazione dell'ego vehicle e ai quattro termini di efficientamento. Procedimento equivalente per il vincolo inferiore e superiore lg , ug riguardante la derivata della forza longitudinale $F_{mc,d}$. Infine, si ha l'applicazione dei limiti inferiori e superiori lh_e , uh_e all'ultimo step della predizione per le quattro variabili di efficientamento.

Come detto in precedenza, le prime due porte del controllore (S-function) hanno in ingresso i quattro stati provenienti dal *Plant*. Da quest'ultimo provengono anche i *Parameters* che rappresentano l'input della terza porta (riferendosi alla Figura 3.2). Alla quarta, quinta e sesta porta si introducono le variabili di riferimento (provenienti dal blocco *References*) dei termini che costituiscono la funzione di costo. I riferimenti sono forniti per il primo step, per gli step intermedi e per l'ultimo step della predizione.

In Figura 3.4 viene mostrata la generazione dei vettori contenenti i riferimenti.

Gli input di riferimento vengono utilizzati per creare i seguenti vettori nella Matlab function *y_ref_generator* :

```

    y_ref_0 = [v_h_ref; F_mc_ref; a_h_ref; F_mc_d_ref; F_req_ref; P_EM_cont_ref;
        Eff_cont_ref; P_loss_cont_ref; E_un_dist_ref];
    y_ref = repmat(y_ref_0,(N-1),1);
    y_ref_e = [v_h_ref; F_mc_ref; P_EM_cont_ref; Eff_cont_ref; P_loss_cont_ref;
        E_un_dist_ref];

```

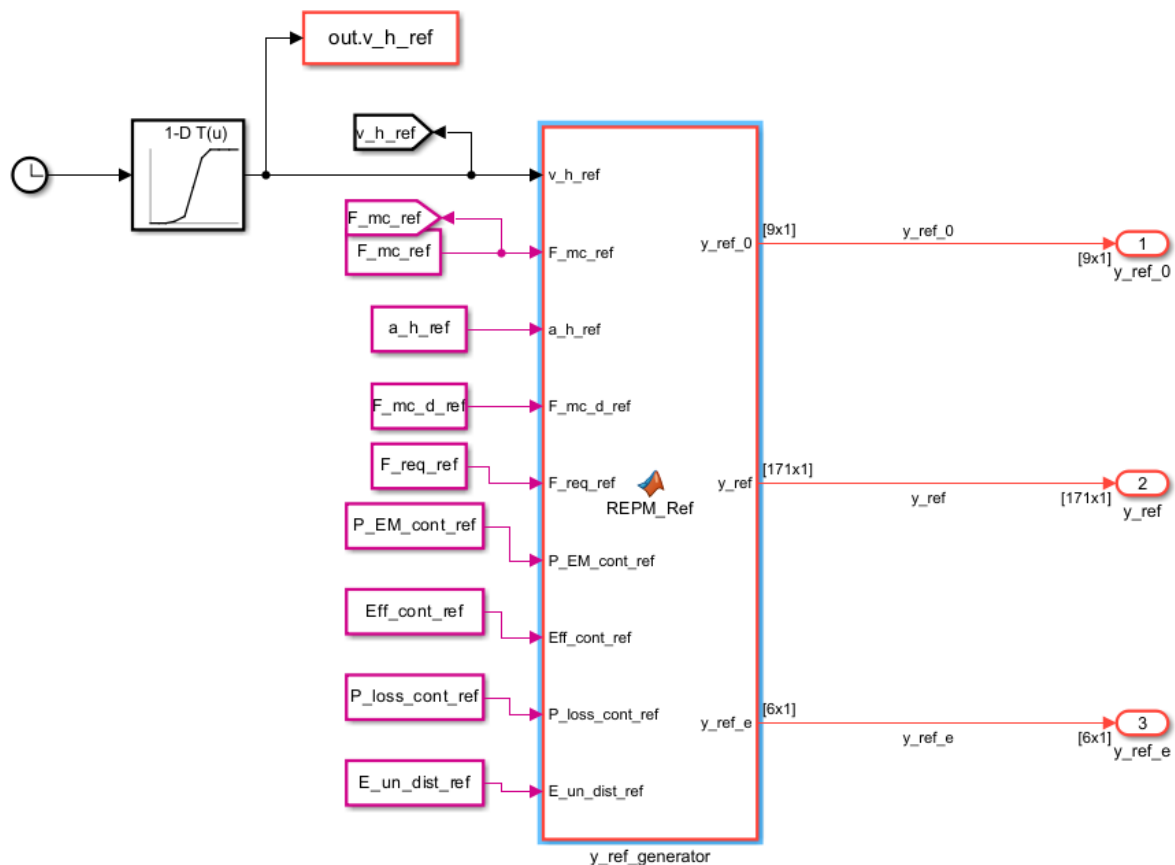


Figura 3.4: Valori di riferimento per i termini caratterizzanti la funzione di costo

Infine, alle porte 19, 20, 21 vengono forniti i vettori *weight path* e *weight term* contenenti i pesi dei termini che caratterizzano la funzione di costo. Il penultimo input all'S-function è relativo all'inizializzazione del risolutore alla prima esecuzione. L'ultima porta di ingresso definisce lo stato iniziale del sistema in modo che la simulazione parta da una condizione ben definita.

Si nota come tra le variabili di ingresso e il blocco S-function vengano interposti dei blocchetti denominati *rate transition* che permettono la comunicazione tra blocchi caratterizzati da time step diversi. Questo determina un'adeguata velocità di esecuzione della simulazione.

La Figura 3.5 raffigura una parte del contenuto del blocco *Plant*; successivamente si analizzerà il contenuto del blocco *Energy_consumption*.

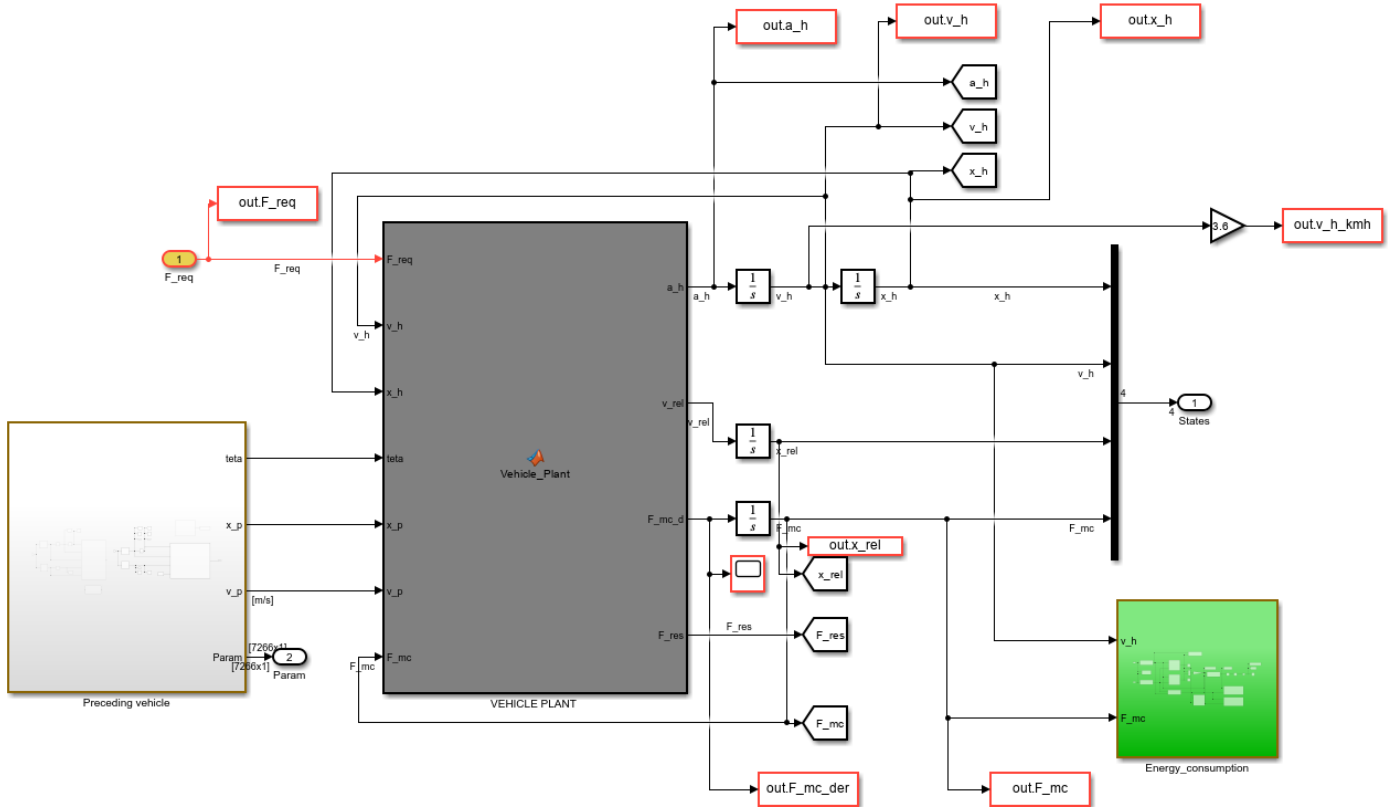


Figura 3.5: Caratterizzazione del blocco Plant

Il *Vehicle Plant* rappresenta il modello di veicolo e come detto in precedenza le equazioni utilizzate sono le stesse di quelle definite per il modello interno del controllore come mostra Figura 3.6. Le variabili di input fornite al blocco sono:

- la forza richiesta al motore elettrico F_{req} che è la variabile di controllo proveniente dal controllore;
- la velocità dell'ego vehicle v_h proveniente dall'integrazione dell'accelerazione dell'ego vehicle a_h ricavata dall'equazione della dinamica:

$$a_h = (F_{mc} - F_{res})/M_{eq};$$

- la distanza percorsa dall'ego vehicle x_h proveniente dall'integrazione della velocità v_h ;
- gli input riguardanti la pendenza della strada θ , la posizione del veicolo precedente x_p e la velocità del veicolo precedente v_p provenienti dal blocco *Preceding vehicle*.

La velocità del veicolo precedente è un segnale generato da una Look-up Table che ha come input il vettore v_prec e il vettore $time = (0:dt_sim:tsim)$. Questi vettori erano stati definiti in precedenza nel codice Matlab per definire la variazione di velocità del veicolo che precede e per creare la serie temporale della simulazione con intervallo di 1 ms.

La stessa logica viene applicata per la Look-up Table che fornisce la pendenza della strada.

Si osserva, poi, che i segnali $v_p, x_p, teta$ vengono inviati fuori, tramite gli outport, affinché siano disponibili al *Vehicle Plant*.

Nella Matlab function *X_p_preview_evaluation* viene creato un vettore t con un passo temporale di 0.2 s, che parte da 0 fino a 2s, il che corrisponde ad ottenere 20 step di predizione (2/0.1), in accordo con quanto impostato nel modello interno del controllore. Il valore iniziale della posizione del veicolo precedente x_p viene replicato corrispondentemente alla lunghezza del vettore temporale, creando un vettore di dimensione (21x1). Questo rappresenta la posizione iniziale del veicolo precedente in tutti gli step del prediction horizon.

$$t = (0:dt:(N*dt))';$$

$$x_p = x_p * ones(length(t), 1);$$

Successivamente viene valutata la posizione prevista per il veicolo precedente x_p_prev tramite i segnali in ingresso di velocità del veicolo precedente (proveniente dalla Look-up Table) e il segnale di posizione del veicolo precedente fornito dall'integrazione della suddetta velocità.

$$x_p_prev = x_p + (v_p * t);$$

Infine, si inizializza il vettore *Param* di dimensione ((21 * np), 1) che verrà compilato opportunamente al fine di inviare i parametri al controllore. Il ciclo *for* itera per ciascuno degli step di predizione, da 1 a N+1.

Per ogni iterazione il vettore inviato corrisponderà a contenere:

- la posizione prevista del veicolo precedente;
- la velocità del veicolo precedente;
- la pendenza della strada;
- i parametri caratterizzanti la rete neurale.

$$Param = zeros((N+1)*np, 1);$$

$$for i = 1:(N+1)$$

$$Param((np*(i-1)+1):(np*i), 1) = [x_p_prev(i); v_p; teta; NNfact];$$

$$end$$

Riprendendo la Figura 3.5, all'uscita del *Vehicle Plant* si ha l'accelerazione dell'ego vehicle a_h integrata per ottenere la velocità v_h , a sua volta integrata per ottenere la distanza percorsa x_h . Successivamente si ha la velocità relativa v_{rel} , integrata per ottenere la posizione relativa x_{rel} ; infine, si ha la derivata della forza longitudinale applicata $F_{mc,d}$ anch'essa integrata in modo da ottenere la forza longitudinale che viene rinviata all'ingresso del blocco affinché possa essere effettuato il calcolo dell'accelerazione.

A questo punto gli stati del sistema sopracitati, ottenuti tramite integrazione, vengono raggruppati nel vettore *States* e inviati al controllore.

Il blocco *Energy_consumption* di Figura 3.5, contiene lo schema che è possibile osservare nella Figura 3.8.

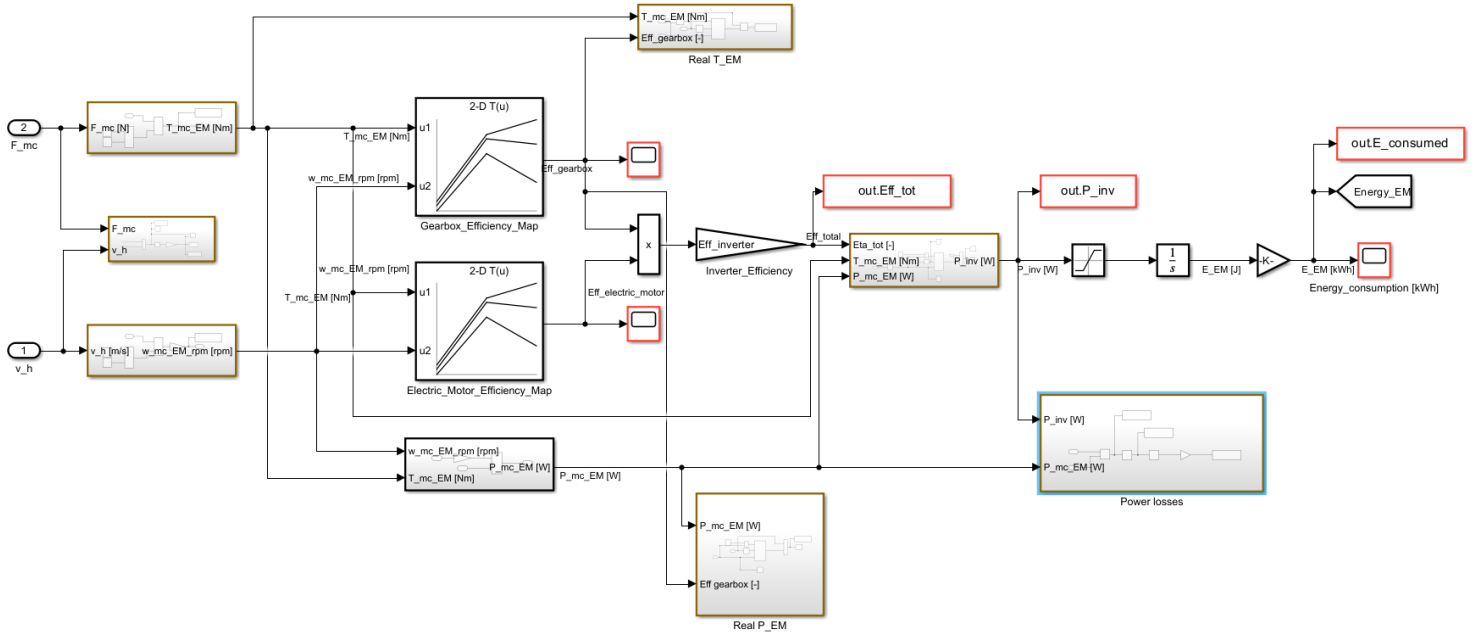


Figura 3.8: Energy consumption block

Gli input di questo schema sono la forza meccanica applicata F_{mc} e la velocità dell'ego vehicle v_h , tramite le quali si può andare a ricavare subito la potenza meccanica $P_{mc} = F_{mc} * v_h$ [W], la quale, integrata e divisa per $(3.6 * 10^6)$, permette di ricavare l'energia meccanica in [kWh]. Infatti, si è a conoscenza che $1kWh = 3.6 * 10^6 J$.

Successivamente viene calcolata la coppia $T_{mc,EM} = \frac{(F_{mc} * R_w)}{i_{tot}}$ [Nm] e la velocità angolare

$w_{mc,EM} = \frac{v_h * i_{tot}}{R_w}$ [rad/s] viene trasformata in [rpm] moltiplicando per il termine $(\frac{60}{2 * \pi})$. Coppia e velocità di rotazione, poi, entrano nelle Look-up Table che definiranno il rendimento del gearbox e del motore elettrico. I rendimenti ricavati verranno moltiplicati per il rendimento costante dell'inverter $\eta_i = 0.98$, in modo da ottenere l'efficienza globale.

In seguito, la potenza meccanica $P_{mc,EM} = T_{mc,EM} * w_{mc,EM}$ [W] viene divisa o moltiplicata per l'efficienza globale in modo da ottenere la potenza all'inverter. Nel caso di trazione si ha la divisione per l'efficienza globale in quanto la potenza che si ha a monte è maggiore in seguito alle perdite lungo la driveline. Nel caso di rigenerazione si ha la moltiplicazione per l'efficienza globale perché ora la potenza maggiore è quella meccanica e per essere immagazzinata nella batteria deve transitare attraverso la driveline, per cui una parte viene persa.

In seguito, nel blocco *Power losses*, si ricava le perdite di potenza dalla differenza tra la potenza all'inverter e la potenza meccanica: $P_{loss} = P_{inv} - P_{mc,EM}$ [W]. Facendo, poi, l'integrale e dividendolo per $(3.6 * 10^6)$ si ottiene l'energia persa in [kWh].

Infine, il consumo di energia viene valutato facendo l'integrale della potenza all'inverter e dividendo per $(3.6 * 10^6)$ in modo da ottenere l'energia consumata in [kWh].

4. Algoritmo di ottimizzazione dei pesi

L'algoritmo basato sul solver *surrogateopt* è un risolutore globale per funzioni di costo che richiedono tempi di calcolo elevati, soprattutto quando queste non hanno derivate analitiche disponibili o sono altamente non lineari. Il risolutore cerca il minimo globale di una funzione di costo a valori reali in più dimensioni, soggetta a vincoli su tutte le variabili.

Il risolutore può opzionalmente mantenere un file di checkpoint per consentire il recupero in caso di arresti anomali, esecuzioni parziali, o per continuare l'ottimizzazione dopo aver raggiunto una condizione di arresto.

La funzione di costo $f(x)$ può essere vuota ($[]$), in tal caso *surrogateopt* cercherà di trovare un punto che soddisfi tutti i vincoli.

Viene utilizzato un modello surrogato, ossia una rappresentazione semplificata della funzione obiettivo reale, con lo scopo di approssimare la funzione di costo e ridurre il numero di simulazioni necessarie, risparmiando tempo computazionale.

Il modello viene costruito utilizzando un piccolo numero di valutazioni della funzione obiettivo reale e serve per stimare il valore della funzione in altre regioni dello spazio delle variabili, senza dover effettuare una simulazione completa. L'algoritmo utilizza il modello surrogato per decidere dove esplorare lo spazio delle variabili, evitando di valutare la funzione obiettivo in ogni punto [30].

L'algoritmo è stato scelto in quanto la funzione di costo, multi-obiettivo, dell'MPC per un Energy Efficient Adaptive Cruise Control contiene nove termini, i cui pesi

$$W_{v_h}, W_{F_{mc}}, W_{a_h}, W_{F_{mc,d}}, W_{F_{req}}, W_{P_{EM,cont}}, W_{Eff_{cont}}, W_{P_{loss,cont}}, W_{E_{un,dist}}$$

devono essere bilanciati a seconda dell'obiettivo che viene imposto. Nei test eseguiti, l'algoritmo è stato utilizzato per ottenere i pesi ottimizzati in due casi: nel primo si aveva come obiettivo lo Speed Tracking, quindi l'inseguimento, da parte dell'ego vehicle, della velocità imposta dall'utente; nel secondo caso l'obiettivo era dare priorità ai quattro termini di efficientamento presenti nella funzione di costo in modo da ridurre il consumo energetico. Questo approccio è ideale per effettuare il tuning di parametri in scenari ingegneristici complessi.

Inizialmente viene definito il vettore contenente i pesi che devono essere ottimizzati e viene definito il range di grandezza entro cui l'algoritmo può esplorare durante l'ottimizzazione. I pesi sulla forza longitudinale e sulla forza richiesta al controllore sono stati impostati pari a 10^{-6} in modo da permettere al sistema di fornire le forze che ritiene opportune e anche per non complicare e sovraccaricare la ricerca da parte dell'algoritmo.

$$\begin{aligned} initial_weights &= [W_{v_h}, W_{a_h}, W_{F_{mc,d}}, W_{P_{EM_cont}}, W_{Eff_cont}, \\ &W_{P_{loss_cont}}, W_{E_{un_dist}}]; \\ lb &= [1, 1e3, 1e-6, 1e3, 1e1, 1e3, 1e4]; \\ ub &= [1e3, 1e5, 1e-3, 1.5e6, 5e4, 7.5e6, 1.4e7]; \end{aligned}$$

Successivamente vengono definite le impostazioni dell'algoritmo:

- *'BatchUpdateInterval',9*: definisce il numero di iterazioni che devono essere eseguite prima che l'algoritmo aggiorni il modello con un nuovo set di punti;
- *'Display','iter'*: a ogni iterazione vengono fornite informazioni nella command window;
- *'InitialPoints', initial_weights*: vettore contenente i pesi iniziali;

- In seguito, viene definita una funzione obiettivo J_cost che prende in ingresso il vettore dei pesi $weights_opt$ che devono essere ottimizzati.

Successivamente, vengono definiti i vettori dei pesi *weight path* e *weight term*, rispettivamente lungo il prediction horizon e all'ultimo step di esso. Dopo di che, questi vettori, tramite la funzione *assignin*, vengono salvati nel workspace di Matlab, affinché le variabili siano disponibili per il modello Simulink.

70

Successivamente viene eseguita la simulazione del modello Simulink, (le opzioni in *simOut* specificano la modalità di simulazione e indicano che le variabili utilizzate dalla simulazione vengono prese dal workspace di Matlab):

```
simOut = sim('R_NMPC_EACC_simulink.slx', 'SimulationMode', 'normal',  
'SrcWorkspace', 'base');
```

In seguito, vengono definiti i pesi $W_1, W_2, W_3, W_4, W_5, W_6$ applicati ai termini che andranno a comporre la funzione di costo J_{cost} . Essi assumeranno dei valori in funzione dell'obiettivo che si vuole ottenere, che sia lo speed tracking oppure la minimizzazione del consumo energetico. Dopo di che, si estrae dalla simulazione le variabili necessarie alla definizione dei performance indicators tramite la funzione *simOut.get*, come mostra Tabella 4.1.

Tabella 4.1: Variabili necessarie alla definizione dei performance indicators

Parameter	Symbol	Value	Unit
Simulation time	<i>sim_time</i>	<i>simOut.get('sim_time')</i>	[s]
Longitudinal speed	<i>v_h</i>	<i>simOut.get('v_h')</i>	[m/s]
Consumed energy	<i>E_consumed</i>	<i>simOut.get('E_consumed')</i>	[kWh/km]
Global efficiency	<i>Eff_tot</i>	<i>simOut.get('Eff_tot')</i>	[-]
Energy losses	<i>E_loss</i>	<i>simOut.get('E_loss')</i>	[kWh/km]
Longitudinal strength derivative	<i>F_mc_der</i>	<i>simOut.get('F_mc_der')</i>	[N/s]
Acceleration	<i>a_h</i>	<i>simOut.get('a_h')</i>	[m/s ²]

I termini che compongono la funzione di costo sono valutati in base agli obiettivi che si vuole ottenere dal sistema, quindi si avrà:

il termine *delta_speed* in modo da far seguire al veicolo la velocità di riferimento impostata:

$$\text{delta_speed} = (v_{\text{expected}} - v_{h_mean});$$

il termine *consumed_energy* in modo da minimizzare l'energia consumata:

$$\text{consumed_energy} = (E_{\text{consumed}}(\text{end}));$$

il termine *eff* si vuole minimizzarlo in modo da massimizzare l'efficienza:

$$\text{eff} = (1 - \text{Eff_tot_mean});$$

il termine *energy_loss* in modo da minimizzare l'energia persa:

$$\text{energy_loss} = (E_{\text{loss}}(\text{end}));$$

per minimizzare la variazione della forza meccanica applicata si considera il suo valore quadratico medio (Root Mean Square), che è una misura statistica utilizzata per rappresentare l'ampiezza media di un segnale o di un insieme di valori, indipendentemente dal segno positivo o negativo:

$$\begin{aligned} F_mc_d_tot &= cumtrapz(sim_time, F_mc_der.^2); \\ F_mc_d_RMS &= sqrt(mean(F_mc_d_tot)); \end{aligned}$$

per minimizzare l'accelerazione dell'ego vehicle si può utilizzare la media del valore assoluto dell'accelerazione, questo penalizza il comportamento oscillatorio:

$$\begin{aligned} a_h_tot &= cumtrapz(sim_time, abs(a_h)); \\ avg_a_h &= mean(a_h_tot); \end{aligned}$$

Infine, viene definita la funzione di costo che deve essere minimizzata:

$$\begin{aligned} J_cost &= (W1*delta_speed) + (W2*consumed_energy) + (W3*eff) + (W4*energy_loss) \\ &+ (W5*F_mc_d_RMS) + (W6*avg_a_h); \end{aligned}$$

5. Test e risultati

È stato deciso di effettuare i test su tre scenari e per ciascuno di essi valutare due casi. Nel primo caso dare la priorità al tracking della velocità, in modo tale che l'ego vehicle segua la velocità impostata dall'utente; nel secondo caso dare la priorità ai termini di efficientamento energetico in modo da diminuire l'energia consumata.

Per fare ciò sono stati impostati i pesi dei termini costituenti la funzione di costo dell'algoritmo di ottimizzazione. Nel primo caso verrà dato maggior peso al termine di velocità per dare priorità al mantenimento della velocità di riferimento impostata dall'utente. Nel secondo caso si darà maggior peso ai termini di efficientamento energetico per cercare di minimizzare l'energia consumata, l'energia persa e massimizzare il rendimento complessivo della driveline. Si passa, quindi, all'impostazione dei pesi caratterizzanti i termini della funzione di costo nel caso di Speed Tracking, come mostra Tabella 5.1. Si impone un peso al termine di velocità di due ordini di grandezza superiore rispetto ai pesi relativi al consumo di energia, efficienza media ed energia persa; i pesi attribuiti alla derivata della forza longitudinale e all'accelerazione sono tali per cui il loro valore moltiplicato per la relativa variabile assegni un'importanza elevata nella funzione di costo in modo da minimizzare le due variabili ed evitare oscillazioni nella velocità dell'ego vehicle.

Tabella 5.1: Pesi caratterizzanti la funzione di costo dell'algoritmo di ottimizzazione nel caso di Speed Tracking

Parameter	Symbol	Value
Weight for delta speed	W1	500
Weight for energy consumption	W2	1
Weight for efficiency	W3	1
Weight for energy losses	W4	1
Weight for RMS value of derivative of longitudinal strength	W5	0.5
Weight for acceleration	W6	1000

L'impostazione dei pesi caratterizzanti i termini della funzione di costo nel caso in cui la priorità venga data alle variabili di efficientamento è avvenuta imponendo dei pesi di tre ordini di grandezza superiori rispetto alla variabile che rappresenta la velocità dell'ego vehicle; mentre i pesi relativi alla variazione della forza longitudinale e all'accelerazione sono rimasti invariati. I dati sono osservabili nella Tabella 5.2.

Tabella 5.2: Pesi caratterizzanti la funzione di costo dell'algoritmo di ottimizzazione nel caso in cui la priorità venga data alle variabili di efficientamento

Parameter	Symbol	Value
Weight for delta speed	W1	0.1
Weight for energy consumption	W2	6000
Weight for efficiency	W3	6000
Weight for energy losses	W4	23000
Weight for RMS value of derivative of longitudinal strength	W5	0.5
Weight for acceleration	W6	1000

Infine, l'algoritmo di ottimizzazione è stato eseguito nelle due condizioni sopra definite sui primi 4 km del ciclo guida WLTP, in modo da fornire i pesi ottimizzati per i termini costituenti la funzione di costo dell'MPC per l'Energy Efficient Adaptive Cruise Control. Sono stati selezionati solo i primi 4 km poiché, altrimenti, l'ottimizzazione sull'intero ciclo avrebbe richiesto molto tempo.

5.1 Analisi di sensitività sul ciclo WLTP

A questo punto, si è passati ad effettuare un'analisi di sensitività andando a variare del -3000%, -50%, +50%, +250%, +1000%, +3000% la soluzione di pesi proposta dall'algoritmo di ottimizzazione nel caso in cui la priorità sia data all'efficientamento energetico. Quindi, per ognuno dei quattro termini di efficientamento (potenza del motore elettrico, efficienza globale considerando l'inverter, il motore elettrico e la trasmissione, perdite di potenza ed energia per unità di distanza) costituenti la funzione di costo dell'Energy Efficient Adaptive Cruise Control è stata eseguita questa analisi per comprendere le tendenze dei seguenti performance indicators:

- velocità media dell'ego vehicle [km/h];
- consumo di energia [kWh];
- rendimento medio [-];
- media delle perdite di potenza [W];
- potenza media del motore elettrico (valutata solo nel caso relativo alla variazione del peso inerente la potenza del motore elettrico) [W].

Le variazioni percentuali per effettuare l'analisi vengono applicate alla soluzione di pesi proposta dall'algoritmo di ottimizzazione dando priorità alle variabili di efficientamento, come mostra Tabella 5.3. Con W_x si fa riferimento ai pesi applicati ai termini della funzione di costo lungo il prediction horizon; $W_{x,end}$ si riferisce ai pesi applicati all'ultimo step del prediction horizon; mentre W_u si riferisce all'applicazione del peso alla variabile di controllo.

Tabella 5.3: Pesi ottimizzati nel caso di priorità data ai termini di efficientamento

Weight	W_{v_h}	$W_{F_{mc}}$	W_{a_h}	$W_{F_{mc_d}}$	$W_{F_{req}}$	$W_{P_{EM,cont}}$	$W_{Eff,cont}$	$W_{P_{loss,cont}}$	$W_{E_{un,dist}}$
W_x	10^2	10^{-6}	10^3	$2.5075 * 10^{-4}$	/	10^3	10	10^3	10^4
$W_{x,end}$	10^2	10^{-6}	/	/	/	10^3	10	10^3	10^4
W_u	/	/	/	/	10^{-6}	/	/	/	/

La prima analisi è stata effettuata variando il peso relativo alla potenza del motore elettrico, $W_{P_{EM,cont}}$, e sono stati ricavati i vari performance indicators, come mostra Tabella 5.4. Essi sono stati poi confrontati con i performance indicators relativi al test effettuato sul ciclo WLTP con i pesi di Tabella 5.3.

Tabella 5.4: Analisi di tendenza dei performance indicators nel caso di variazione del termine di potenza del motore elettrico

Performance indicators	Unit	WLTP Test with priority to Efficiencies terms	$W_{P_{EM,cont}}$ (-3000%) [10^{-6}]	$W_{P_{EM,cont}}$ (-50%) [500]	$W_{P_{EM,cont}}$ (+50%) [1500]	$W_{P_{EM,cont}}$ (+250%) [3500]	$W_{P_{EM,cont}}$ (+1000%) [11000]	$W_{P_{EM,cont}}$ (+3000%) [31000]
$v_{h,mean}$ (Average vehicle speed)	[km/h]	73,40	73,97 (0,78%)	73,68 (0,38%)	73,13 (0,37%)	72,16 (1,69%)	69,43 (5,41%)	65,20 (11,17%)
E_{tot} (Consumed energy)	[kWh]	5,8080	5,9147 (1,84%)	5,8599 (0,89%)	5,7585 (0,85%)	5,5817 (3,90%)	5,1087 (12,04%)	4,4459 (23,45%)
η_{mean} (Mean efficiency)	[-]	70,12%	70,25% (0,19%)	70,18% (0,09%)	70,05% (0,10%)	69,81% (0,44%)	69,03% (1,55%)	67,45% (3,81%)
P_{losses} (Power losses)	[W]	4045,9	4098,3 (1,30%)	4071,4 (0,63%)	4021,7 (0,60%)	3935,8 (2,72%)	3715,1 (8,18%)	3450,7 (14,71%)
P_{EM} (Electric motor power)	[W]	21598	22179 (2,69%)	21880 (1,31%)	21331 (1,24%)	20390 (5,59%)	17967 (16,81%)	14794 (31,50%)

Siccome la potenza del motore elettrico è una variabile compresa nella funzione di costo, in questa analisi è stato inserito anche il performance indicator relativo alla potenza del motore elettrico in modo da analizzare se il suo comportamento corrispondesse con le aspettative. Gli andamenti di Figura 5.1 ottenuti con i dati di Tabella 5.4 corrispondono alle aspettative. Infatti, siccome il valore di riferimento per la potenza del motore elettrico è stato impostato a zero (in quanto l'obiettivo per minimizzare il consumo è diminuire la potenza utilizzata dal motore), si osserva che all'aumentare del peso $W_{P_{EM,cont}}$ si ha una riduzione del performance

indicator direttamente associato *EM Power*; si ha un decremento della velocità media dell'ego vehicle spostandosi in punti della mappa motore a rendimento minore. Inoltre, si ha una riduzione delle perdite di potenza e la variabile di maggior interesse è rappresentata dalla diminuzione nel consumo di energia.

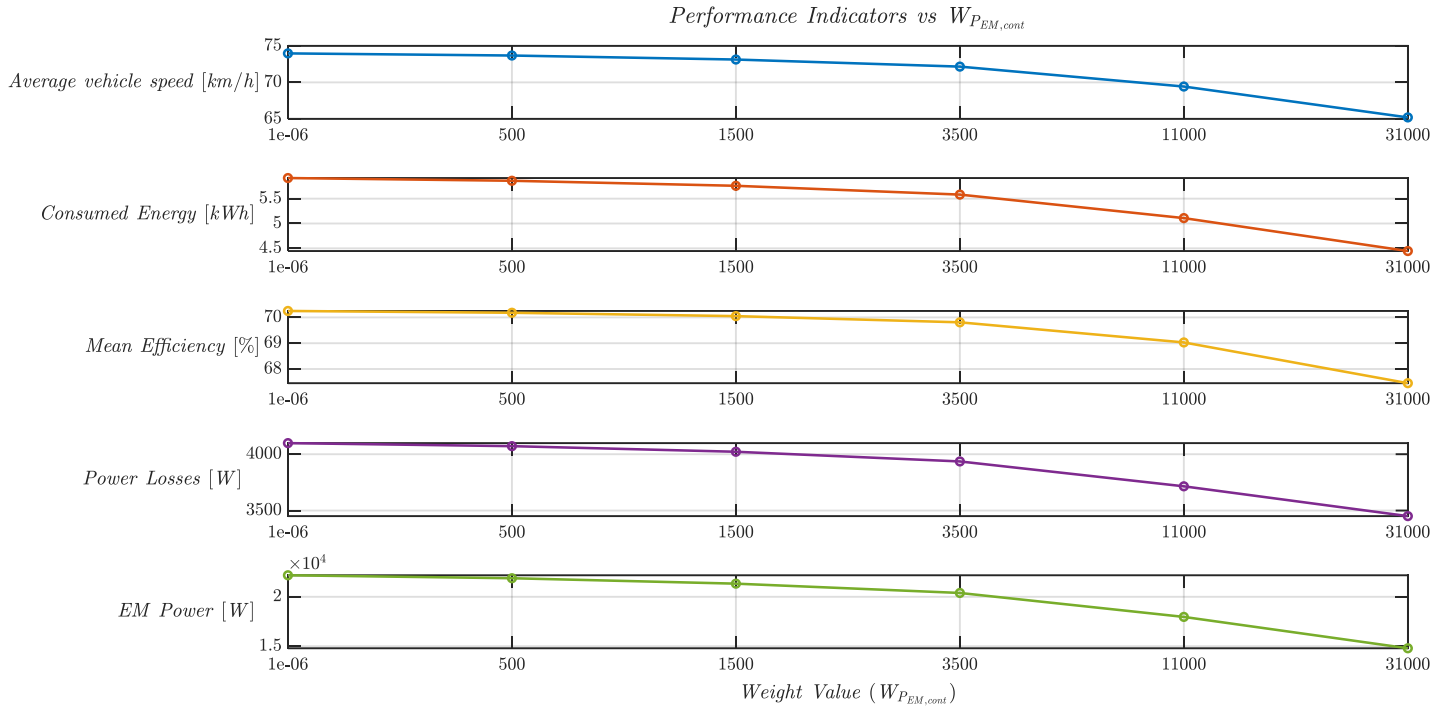


Figura 5.1: Performance indicators relativi alla variazione del termine di potenza del motore elettrico

Si è proseguito con l'analisi sul peso relativo all'efficienza globale della powertrain $W_{Eff,cont}$, comprendente l'efficienza costante dell'inverter, quella del motore elettrico e quella del gearbox. Sono stati ricavati i vari performance indicators, come mostra Tabella 5.5, poi confrontati con i performance indicators relativi al test effettuato sul ciclo WLTP con i pesi di Tabella 5.3.

Tabella 5.5: Analisi di tendenza dei performance indicators nel caso di variazione del termine di efficienza globale della powertrain

Performance indicators	Unit	WLTP Test with priority to Efficiencies terms	$W_{Eff,cont}$ (-3000%) [10 ⁻⁶]	$W_{Eff,cont}$ (-50%) [5]	$W_{Eff,cont}$ (+50%) [15]	$W_{Eff,cont}$ (+250%) [35]	$W_{Eff,cont}$ (+1000%) [110]	$W_{Eff,cont}$ (+3000%) [310]
v_{hmean} (Average vehicle speed)	[km/h]	73,40	73,37 (0,04%)	73,37 (0,04%)	73,39 (0,01%)	73,43 (0,04%)	73,51 (0,15%)	73,65 (0,34%)
E_{tot} (Consumed energy)	[kWh]	5,8080	5,8062 (0,03%)	5,8070 (0,02%)	5,8089 (0,02%)	5,8126 (0,08%)	5,8251 (0,29%)	5,8518 (0,75%)
η_{mean} (Mean efficiency)	[-]	70,12%	70,11% (0,01%)	70,11% (0,01%)	70,12%	70,13% (0,01%)	70,20% (0,11%)	70,44% (0,46%)
P_{losses} (Power losses)	[W]	4045,9	4044,5 (0,03%)	4044,9 (0,02%)	4045,6 (0,01%)	4047,4 (0,04%)	4050,5 (0,12%)	4055,7 (0,24%)

In questa analisi, il performance indicator direttamente associato al peso variato, presente nella funzione di costo, è rappresentato dall'efficienza media del powertrain.

I dati in Tabella 5.5 definiscono gli andamenti di Figura 5.2 e sono conformi alle aspettative. Infatti, il valore di riferimento per l'efficienza globale è stato impostato a zero, in quanto nella funzione di costo viene considerato il termine $(1 - \eta_{a_{tot}})$, quindi minimizzarlo corrisponde a massimizzare il rendimento della powetrain. All'aumentare del peso $W_{Eff,cont}$ si ha un aumento dell'efficienza media e corrispondentemente si verifica un incremento della velocità media dell'ego vehicle in quanto i punti a maggior rendimento, nella mappa motore, si spostano a coppie intermedie e velocità di rotazione maggiori. Inoltre, far lavorare la powertrain nel punto di maggiore efficienza non permette di ridurre il consumo di energia, come si nota dall'andamento del performance indicator associato. Infine, anche le perdite di potenza tendono ad aumentare.

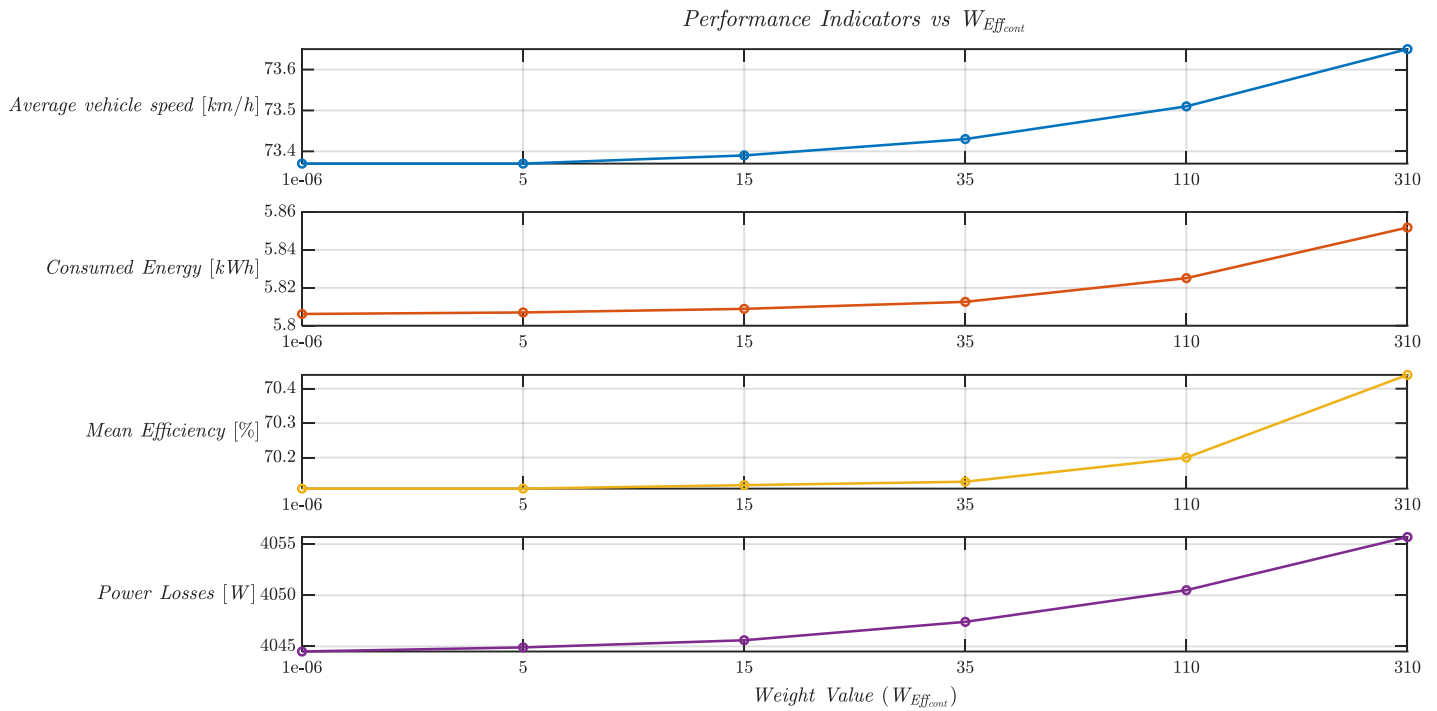


Figura 5.2: Performance indicators relativi alla variazione del termine inerente l'efficienza globale

Successivamente l'analisi si è spostata al peso relativo alle perdite di potenza $W_{P_{loss,cont}}$. Sono stati ricavati i vari performance indicators, come mostra Tabella 5.6, poi confrontati con i performance indicators relativi al test effettuato sul ciclo WLTP con i pesi di Tabella 5.3.

Tabella 5.6: Analisi di tendenza dei performance indicators nel caso di variazione del termine inerente le perdite di potenza nella powertrain

Performance indicators	Unit	WLTP Test with priority to Efficiencies terms	$W_{P_{loss,cont}}$ (-3000%) [10 ⁻⁶]	$W_{P_{loss,cont}}$ (-50%) [500]	$W_{P_{loss,cont}}$ (+50%) [1500]	$W_{P_{loss,cont}}$ (+250%) [3500]	$W_{P_{loss,cont}}$ (+1000%) [11000]	$W_{P_{loss,cont}}$ (+3000%) [31000]
v_{hmean} (Average vehicle speed)	[km/h]	73,40	73,48 (0,11%)	73,44 (0,05%)	73,36 (0,05%)	73,20 (0,27%)	72,59 (1,10%)	70,84 (3,49%)
E_{tot} (Consumed energy)	[kWh]	5,8080	5,8235 (0,27%)	5,8157 (0,13%)	5,8002 (0,13%)	5,7691 (0,67%)	5,6542 (2,65%)	5,3517 (7,86%)
η_{mean} (Mean efficiency)	[-]	70,12%	70,17% (0,07%)	70,14% (0,03%)	70,09% (0,04%)	69,99% (0,19%)	69,55% (0,81%)	67,84% (3,25%)
P_{losses} (Power losses)	[W]	4045,9	4053,5 (0,19%)	4049,7 (0,09%)	4042,1 (0,09%)	4026,8 (0,47%)	3975,7 (1,74%)	3878,8 (4,13%)

In questa analisi il performance indicator direttamente associato al peso variato, presente nella funzione di costo, è rappresentato dalla media delle perdite di potenza lungo la powertrain. I dati in Tabella 5.6 definiscono gli andamenti di Figura 5.3 e corrispondenti alle aspettative. Infatti, siccome il valore di riferimento per le perdite di potenza è stato impostato a zero (in quanto l'obiettivo per minimizzare il consumo è diminuire le perdite di potenza che si hanno lungo la powertrain), si riscontra che all'aumentare del peso $W_{P_{loss,cont}}$ si verifica una riduzione del performance indicator direttamente associato *Power Losses*. Il rendimento medio tende a diminuire e con esso la velocità media dell'ego vehicle siccome ci si sta spostando in punti della mappa motore a coppie e velocità di rotazione inferiori. Infine, anche il performance indicator che rappresenta l'energia consumata tende a diminuire come ci si aspettava.

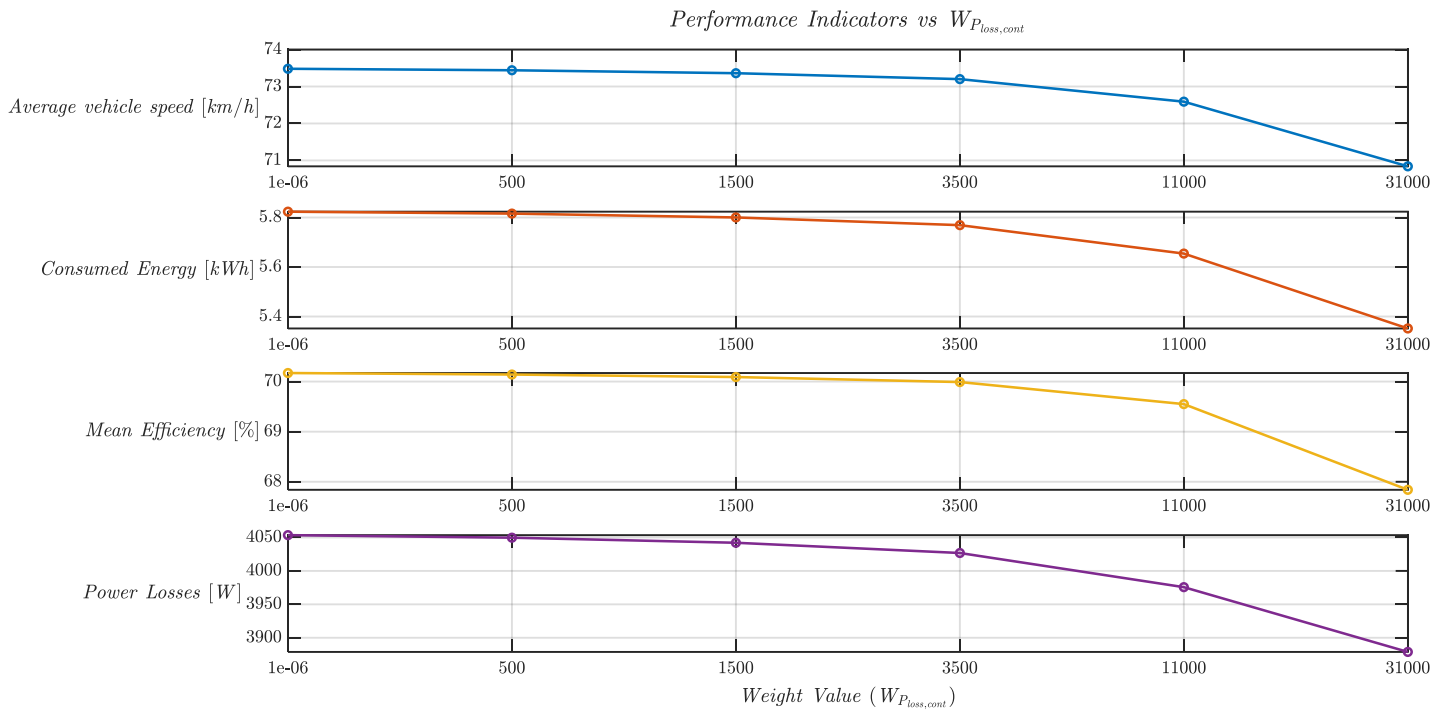


Figura 5.3: Performance indicators relativi alla variazione del termine inerente le perdite di potenza lungo la powertrain

Infine, l'ultima analisi è stata effettuata sul peso relativo all'energia consumata per unità di distanza $W_{E_{un,dist}}$. Sono stati ricavati i vari performance indicators, come mostra Tabella 5.7, poi confrontati con i performance indicators relativi al test effettuato sul ciclo WLTP con i pesi di Tabella 5.3.

Tabella 5.7: Analisi di tendenza dei performance indicators nel caso di variazione del termine inerente l'energia consumata per unità di distanza

Performance indicators	Unit	WLTP Test with priority to Efficiencies terms	$W_{Eun,dist}$ (-3000%) [1e-6]	$W_{Eun,dist}$ (-50%) [5e3]	$W_{Eun,dist}$ (+50%) [1,5e4]	$W_{Eun,dist}$ (+250%) [3,5e4]	$W_{Eun,dist}$ (+1000%) [1,1e5]	$W_{Eun,dist}$ (+3000%) [3,1e5]
v_{hmean} (Average vehicle speed)	[km/h]	73,40	73,42 (0,03%)	73,41 (0,01%)	73,39 (0,01%)	73,36 (0,05%)	73,23 (0,23%)	72,88 (0,71%)
E_{tot} (Consumed energy)	[kWh]	5,8080	5,8116 (0,06%)	5,8098 (0,03%)	5,8062 (0,03%)	5,7989 (0,16%)	5,7723 (0,61%)	5,7035 (1,80%)
η_{mean} (Mean efficiency)	[-]	70,12%	70,12%	70,12%	70,11% (0,01%)	70,10% (0,03%)	70,06% (0,09%)	69,92% (0,29%)
P_{losses} (Power losses)	[W]	4045,9	4047,3 (0,03%)	4046,6 (0,02%)	4045,2 (0,02%)	4042,6 (0,08%)	4032,1 (0,34%)	4003,9 (1,04%)

In questa analisi il performance indicator direttamente associato al peso variato, presente nella funzione di costo, è rappresentato dal consumo di energia.

I dati in Tabella 5.7 definiscono gli andamenti di Figura 5.4 e corrispondenti con le aspettative. Infatti, siccome il valore di riferimento riferito all'energia spesa per unità di distanza è stato impostato a zero (in quanto l'obiettivo per minimizzare il consumo è diminuire l'energia spesa sulla distanza percorsa), si osserva che all'aumentare del peso $W_{Eun,dist}$ si verifica una riduzione del performance indicator direttamente associato *Consumed Energy*. Il rendimento medio tende a diminuire e con esso la velocità media dell'ego vehicle siccome ci si sta spostando in punti della mappa motore a coppie e velocità di rotazione inferiori. Infine, anche il performance indicator che rappresenta la media delle perdite di potenza tende a diminuire come ci si aspettava.

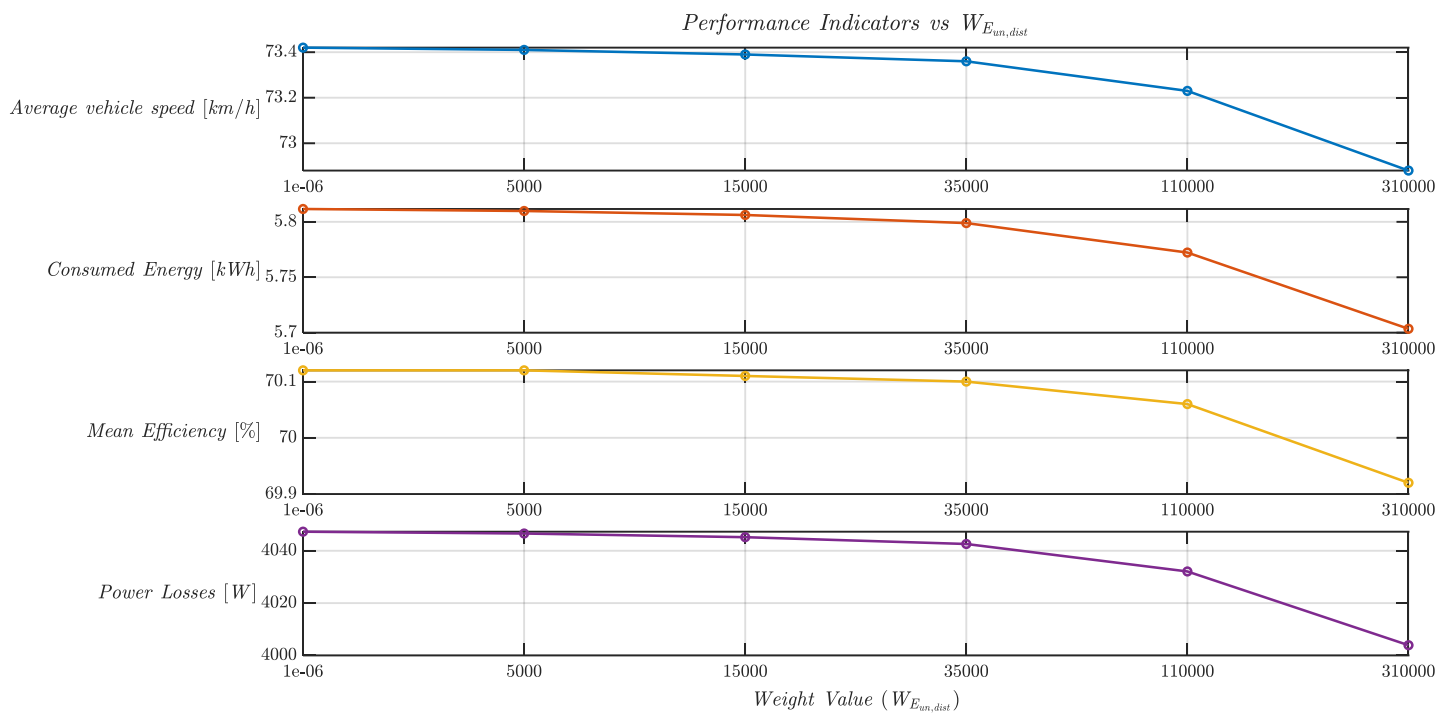


Figura 5.4: Performance indicators relativi alla variazione del termine inerente l'energia spesa per unità di distanza

5.2 Scenari di test

Il primo scenario rappresenta una manovra di accelerazione del veicolo che precede su 1 km di distanza, come illustrato nella Figura 5.5, [31]. Esso varia la sua velocità tra 0 e 40 km/h, mentre le condizioni iniziali imposte all'ego vehicle sono partenza da fermo e distanza dal veicolo che precede di 200 m.

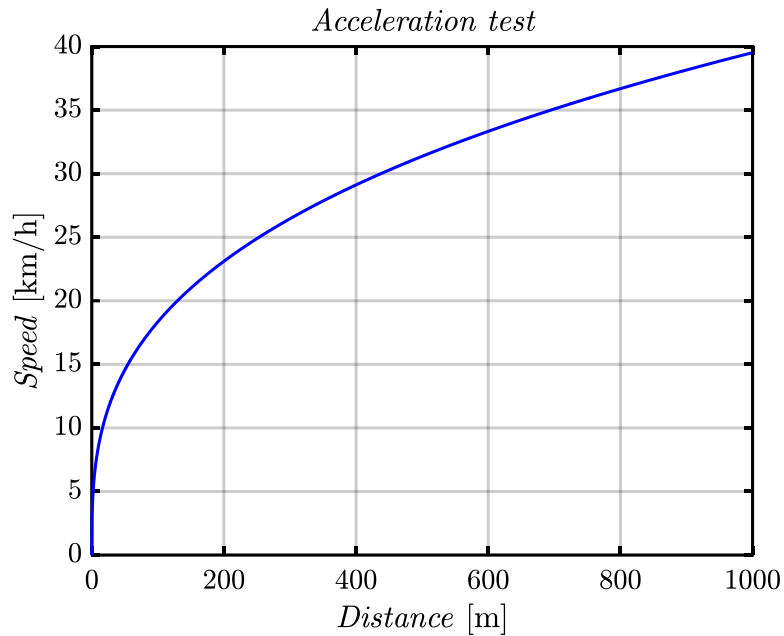


Figura 5.5: Primo scenario di test

Il secondo scenario è rappresentato da Figura 5.6, ed è stato scelto analizzando la letteratura [32]. Il veicolo che precede varia la sua velocità tra 0 e 67 km/h. Il test viene effettuato su 1200 s di simulazione che equivalgono a circa 12 km di distanza percorsa.

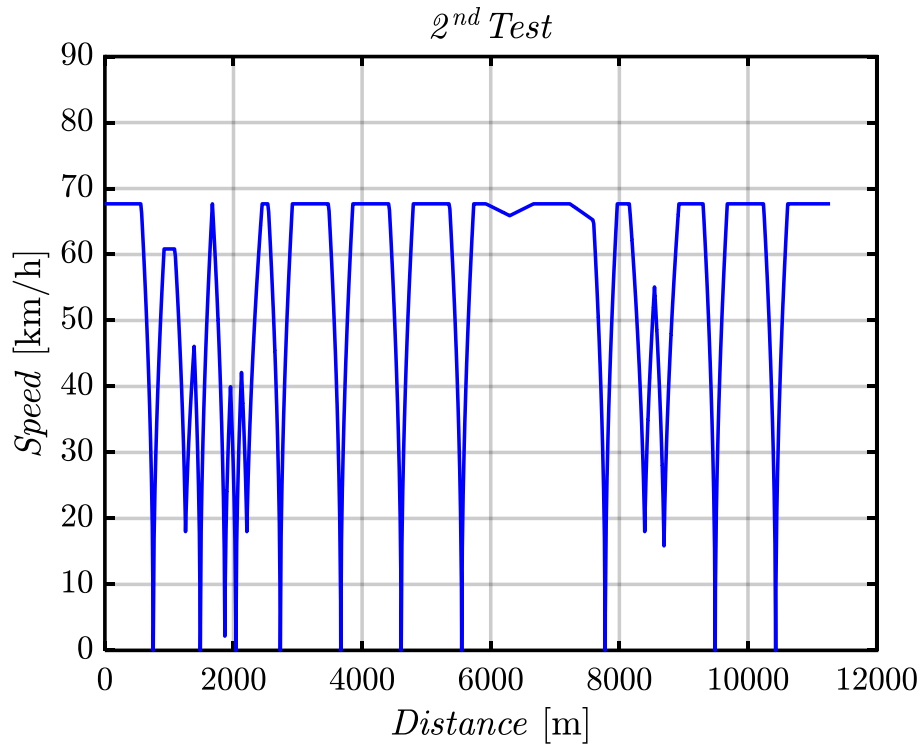


Figura 5.6: Secondo scenario di test

Il terzo scenario è rappresentato dal ciclo guida WLTP, in Figura 5.7, utilizzato nella letteratura [33], [34]. Il WLTP (Worldwide Harmonized Light Vehicles Test Procedure) è un ciclo di guida standardizzato a livello globale, progettato per fornire una valutazione più realistica delle prestazioni e delle emissioni dei veicoli, rispetto a cicli precedenti, come il NEDC (New European Driving Cycle). È stato introdotto nel 2017 nell'Unione Europea per omologare i veicoli leggeri. Rispetto al NEDC, il WLTP, basandosi sui dati raccolti da veicoli reali in diverse aree del mondo, è un ciclo più lungo che include condizioni di guida a velocità più elevata, offrendo transitori di accelerazione e decelerazione più realistici.

Il WLTP è suddiviso in quattro fasi che rappresentano diverse condizioni di guida. La prima fase simula la guida urbana a basse velocità, fino a 60 km/h, con frequenti arresti e partenze; la seconda fase simula la guida suburbana con velocità fino a 80 km/h; la terza fase rappresenta la guida su strade extraurbane con velocità fino a 120 km/h; la quarta fase simula la guida in autostrada con velocità fino a 131 km/h.

La durata complessiva del ciclo WLTP è di 30 minuti, con una distanza totale percorsa di circa 23 km [35].

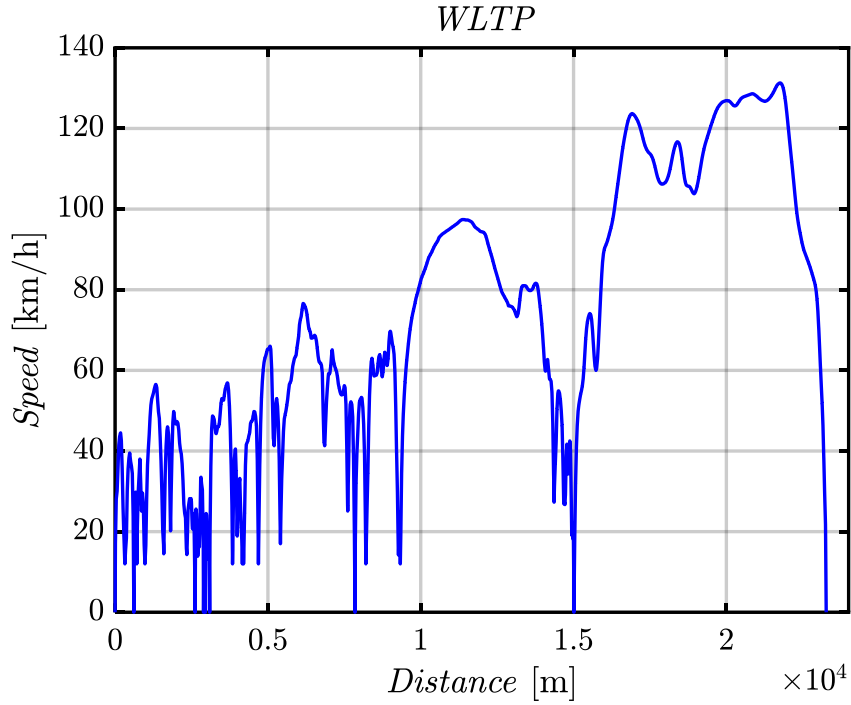


Figura 5.7: Ciclo guida WLTP

5.2.1 Risultati sul primo scenario

Nell'internal model sono stati definiti i pesi dei termini costituenti la funzione di costo al fine di eseguire il test sul tracking della velocità, come riportato in Tabella 5.8. I termini W_x , $W_{x,end}$, W_u rappresentano rispettivamente i pesi forniti lungo il prediction horizon, all'ultimo passo di esso e il peso sulla variabile di controllo.

Tabella 5.8: Pesi dei termini costituenti la funzione di costo al fine di eseguire il test sul tracking della velocità nel primo scenario

Weight	W_{v_h}	$W_{F_{mc}}$	W_{a_h}	$W_{F_{mc_d}}$	$W_{F_{req}}$	$W_{P_{EM,cont}}$	$W_{Eff,cont}$	$W_{P_{loss,cont}}$	$W_{E_{un,dist}}$
W_x	10^3	10^{-6}	10^3	$2.5075 * 10^{-4}$	/	10^3	10	10^3	10^4
$W_{x,end}$	10^3	10^{-6}	/	/	/	10^3	10	10^3	10^4
W_u	/	/	/	/	10^{-6}	/	/	/	/

Una volta effettuato il test, sono stati ricavati i performance indicators, riportati in Tabella 5.9. I valori di velocità dell'ego vehicle, efficienza globale e perdite di potenza sono stati mediati rispetto la distanza percorsa. Inoltre, è stata valutata l'energia consumata.

Tabella 5.9: Performance indicators on the Tracking Speed test

Performance indicator	Symbol	Value	Unit
Average vehicle speed	$v_{h,mean}$	29,66	[km/h]
Consumed energy	E_{tot}	0,1162	[kWh]
Mean efficiency	η_{mean}	43,37%	[-]
Drivetrain Power losses	P_{losses}	1903,6	[W]

Successivamente, sono stati variati i pesi dei termini costituenti la funzione di costo per effettuare il test nel quale viene data priorità ai termini di efficientamento energetico, come mostra la Tabella 5.10.

Tabella 5.10: Pesi dei termini costituenti la funzione di costo al fine di eseguire il test sul primo scenario dando priorità ai termini di efficientamento energetico

Weight	W_{v_h}	$W_{F_{mc}}$	W_{a_h}	$W_{F_{mc,d}}$	$W_{F_{req}}$	$W_{P_{EM,cont}}$	$W_{Eff,cont}$	$W_{P_{loss,cont}}$	$W_{E_{un,dist}}$
W_x	10^2	10^{-6}	10^3	$2.5075 * 10^{-4}$	/	10^3	10	10^3	10^4
$W_{x,end}$	10^2	10^{-6}	/	/	/	10^3	10	10^3	10^4
W_u	/	/	/	/	10^{-6}	/	/	/	/

In seguito, in Tabella 5.11, vengono riportati i relativi performance indicators e viene effettuato un confronto dei risultati evidenziando le variazioni percentuali.

Tabella 5.11: Performance indicators relativi al test sul primo scenario. Sono riportate le variazioni percentuali relative al confronto tra il caso in cui si dà priorità all'inseguimento della velocità e il caso in cui viene data priorità ai termini di efficientamento

Performance indicator	Symbol	Test with priority to efficiencies terms	Test with priority to speed tracking	Percentage change	Unit
Average vehicle speed	$v_{h,mean}$	29,61	29,66	0,17%	[km/h]
Consumed energy	E_{tot}	0,1162	0,1162	0%	[kWh]
Mean efficiency	η_{mean}	43,24%	43,37%	0,30%	[-]
Drivetrain Power losses	P_{losses}	1906,7	1903,6	0,16%	[W]

L'analisi dei risultati non mostra risultati significativi per quanto riguarda la riduzione nel consumo di energia. Infatti, come mostrano gli andamenti in Figura 5.8, dopo aver percorso i primi 100 m, il profilo di velocità ottenuto nel test di speed tracking è sovrapposto al profilo di

velocità ottenuto dando priorità ai termini di efficientamento. L'efficienza media peggiora dello 0.30% e a questa è collegato un leggero decremento della velocità media dell'ego vehicle dello 0.17%. Questi ultimi due termini sono legati in quanto, in un motore elettrico, le zone a rendimento inferiore si trovano in punti di lavoro a velocità di rotazione e coppia inferiori. Inoltre, si nota un aumento dello 0.16% del termine riguardante la media delle perdite di potenza, contrastante con il comportamento che ci si aspetterebbe. Infatti, si vorrebbe avere un decremento di esse siccome il relativo valore di riferimento impostato nella funzione di costo è zero, volto a ridurle il più possibile.

Da questo test su 1 km non si riesce ad ottenere i risultati attesi inerenti ad una diminuzione nel consumo di energia, probabilmente a causa della breve distanza percorsa.

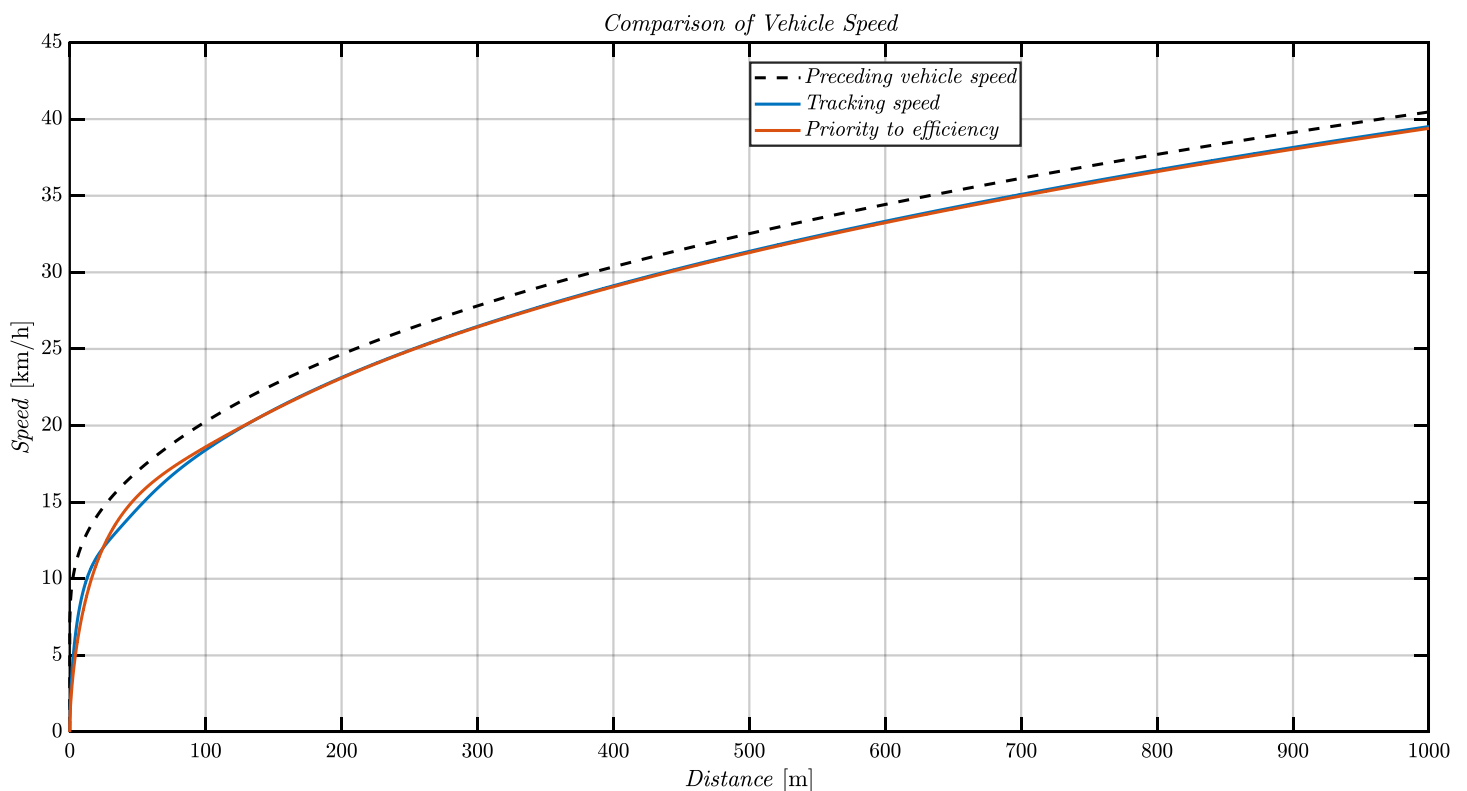


Figura 5.8: Profili di velocità nel test di speed tracking e nel caso di priorità data all'efficientamento

Inizialmente l'ego vehicle dista 200 m dal veicolo che precede. Il profilo blu, rappresentante il test inerente allo speed tracking e il profilo arancio inerente al test dando priorità ai termini di efficientamento, mostrano un comportamento diverso solo nei primi 100 m di distanza percorsi. Nella fase iniziale, il profilo blu presenta un incremento più rapido nella velocità rispetto al profilo arancio. Quest'ultimo, intorno ai 50 m di distanza percorsa, mostra dei valori di velocità superiori. Successivamente, dopo i primi 100 m percorsi, la variazione di velocità dei profili è quasi identica, con il profilo blu che ha dei valori leggermente superiori. Questo lo si comprende dal grafico che rappresenta la distanza relativa (Figura 5.9) in cui si nota che la distanza tracciata dal profilo arancio tende ad aumentare rispetto quella tracciata dal profilo blu. Ciò è sinonimo del fatto che la variazione di velocità del profilo arancio è inferiore in quanto cerca di consumare meno energia. Questo lo si nota anche dal grafico in Figura 5.11, rappresentante la coppia

applicata. Nei primi metri percorsi la coppia applicata nel test di speed tracking ha un picco maggiore rispetto al test in cui si dà la priorità all'efficientamento energetico; successivamente si ha un tratto in cui le curve si accavallano, per poi crescere linearmente dopo i 200 m di distanza percorsa. I dati riguardanti il consumo di energia rispetto al chilometro percorso non hanno riportato variazioni come detto in precedenza e come mostrato dal grafico di Figura 5.10.

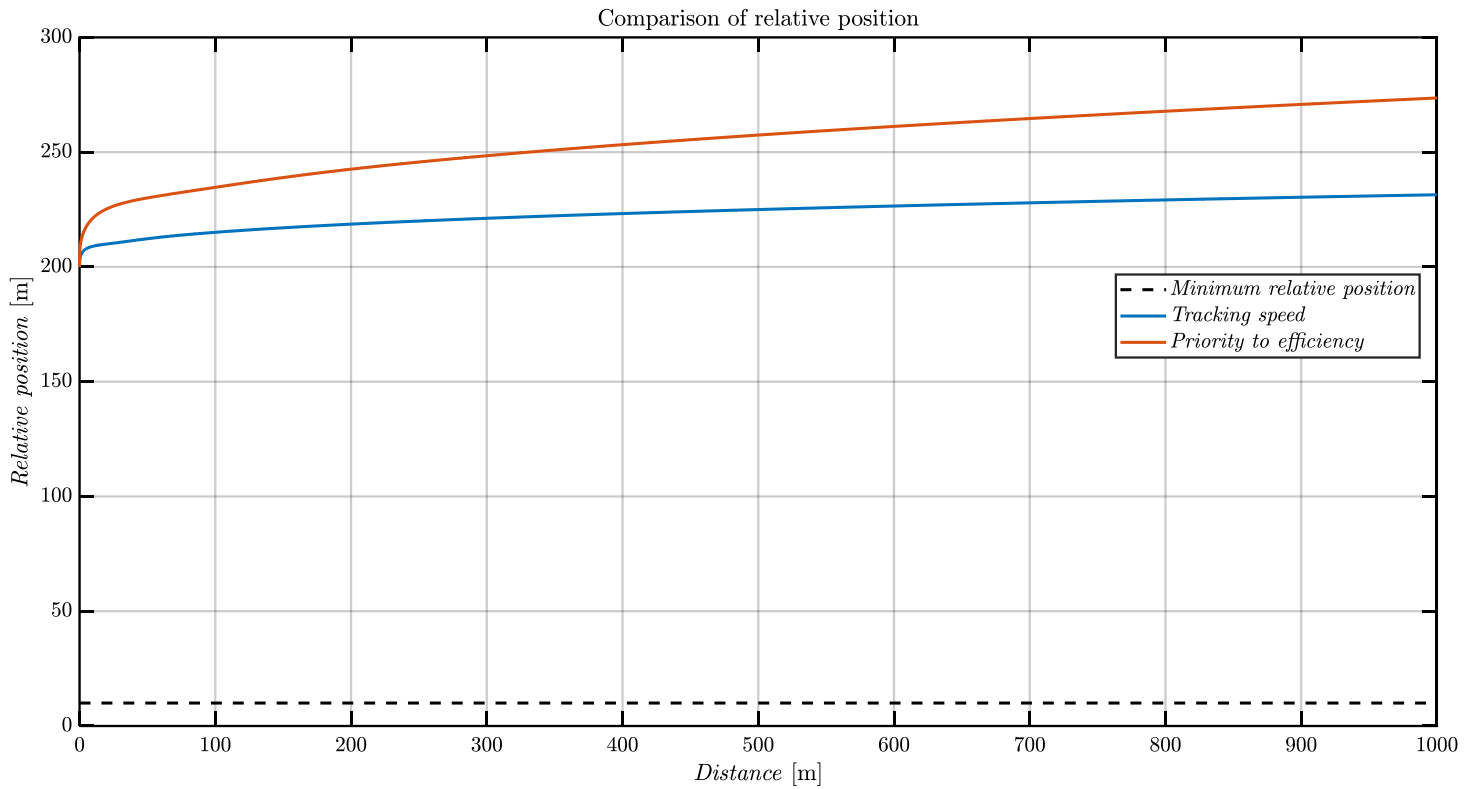


Figura 5.9: Andamenti della distanza relativa nel test di speed tracking e nel caso di priorità data all'efficientamento

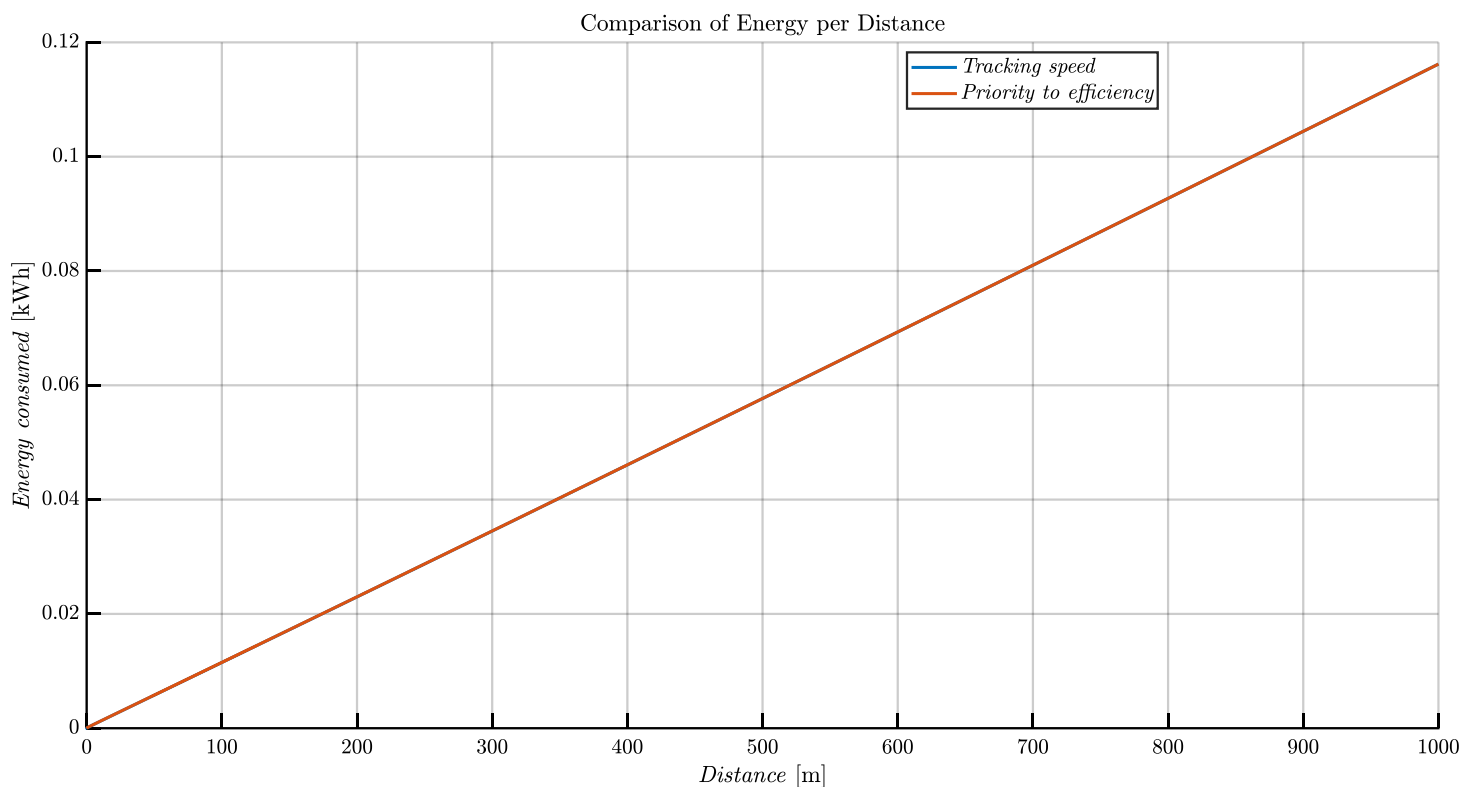


Figura 5.10: Andamenti dell'energia consumata nel test di speed tracking e nel caso di priorità data all'efficientamento

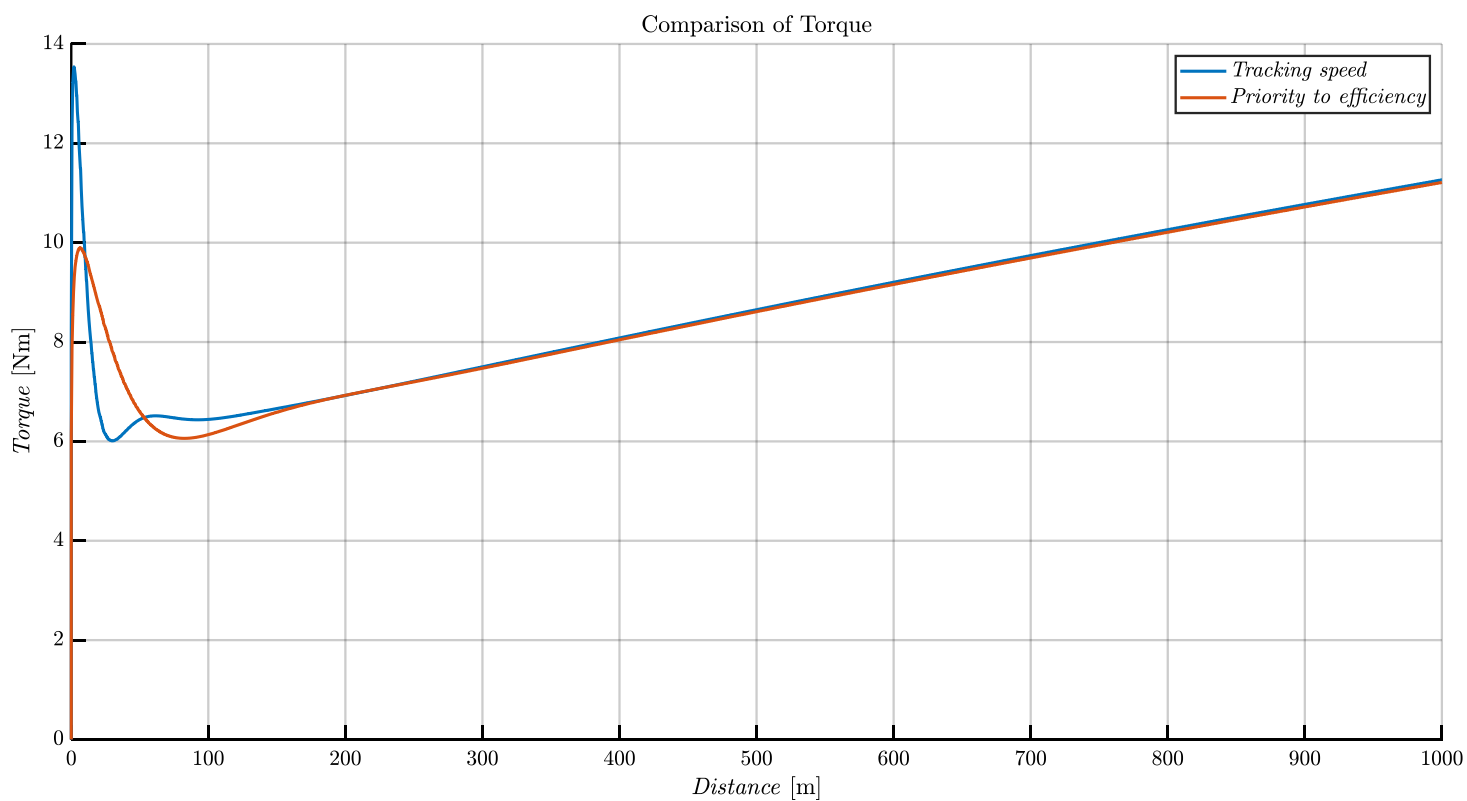


Figura 5.11: Profili di coppia nel test di speed tracking e nel caso di priorità data all'efficientamento

5.2.2 Risultati sul secondo scenario

Nell'internal model sono stati definiti i pesi dei termini costituenti la funzione di costo al fine di eseguire il test sul tracking della velocità; i pesi impostati sono identici a quelli di Tabella 5.8.

I performance indicators ricavati sono riportati in Tabella 5.12:

Tabella 5.12: Performance indicators on the Tracking Speed test

Performance indicator	Symbol	Value	Unit
Average vehicle speed	$v_{h_{mean}}$	57,29	[km/h]
Consumed energy	E_{tot}	1,9659	[kWh]
Mean efficiency	η_{mean}	70,44%	[-]
Drivetrain Power losses	P_{losses}	2881,4	[W]

Successivamente, sono stati variati i pesi dei termini costituenti la funzione di costo per effettuare il test nel quale viene data priorità ai termini di efficientamento energetico; i pesi impostati sono identici a quelli di Tabella 5.10.

In seguito, vengono riportati i relativi performance indicators e viene effettuato un confronto dei risultati evidenziando le variazioni percentuali, come mostrato in Tabella 5.13.

Tabella 5.13: Performance indicators relativi al test sul secondo scenario. Sono riportate le variazioni percentuali relative al confronto tra il caso in cui si dà priorità all'inseguimento della velocità e il caso in cui viene data priorità ai termini di efficientamento

Performance indicator	Symbol	Test with priority to efficiencies terms	Test with priority to speed tracking	Percentage change	Unit
Average vehicle speed	$v_{h_{mean}}$	56,42	57,29	1,52%	[km/h]
Consumed energy	E_{tot}	1,9558	1,9659	0,51%	[kWh]
Mean efficiency	η_{mean}	67,59%	70,44%	4,05%	[-]
Drivetrain Power losses	P_{losses}	2871,0	2881,4	0,36%	[W]

I risultati ottenuti sono in linea con le aspettative. Analizzando i dati si evince una riduzione dell'energia consumata dello 0.51%, e l'andamento di quest'ultima in funzione della distanza percorsa è riportato in Figura 5.13.

Nei primi 4 km percorsi il profilo arancio mostra una distanza relativa maggiore rispetto al profilo blu, come mostra Figura 5.14, il che significa che nel test con priorità all'efficientamento energetico l'ego vehicle sta andando a velocità inferiore per ridurre il consumo di energia. Infatti, questo si nota anche dal grafico che riporta i profili di velocità in Figura 5.12. Inoltre, i picchi di coppia del profilo blu, rappresentante il test sullo speed tracking, sono maggiori rispetto al profilo arancio, osservabile alla Figura 5.15.

Successivamente, dopo 5 km percorsi, la distanza relativa dal veicolo che precede, per il profilo arancio, tende a essere inferiore rispetto a quella del profilo blu. Ciò accade perché i profili di

velocità diventano sfalsati, cioè il profilo arancio cercherà di seguire il profilo di velocità del veicolo precedente, però questo avviene in modo meno preciso rispetto al profilo blu in cui l'obiettivo era il tracking della velocità. A causa di questo sfalsamento, ad una fissata distanza percorsa, corrisponderà una velocità del profilo arancio maggiore a quella del profilo blu. L'efficienza globale della driveline peggiora del 4.05%, e collegato a questo valore si ha un decremento della velocità media dell'ego vehicle dell'1.52%. Infatti, al diminuire del rendimento, i punti di lavoro sulla mappa motore tendono verso coppie e velocità di rotazione inferiori. Inoltre, è presente una leggera diminuzione delle perdite di potenza dello 0.36%.

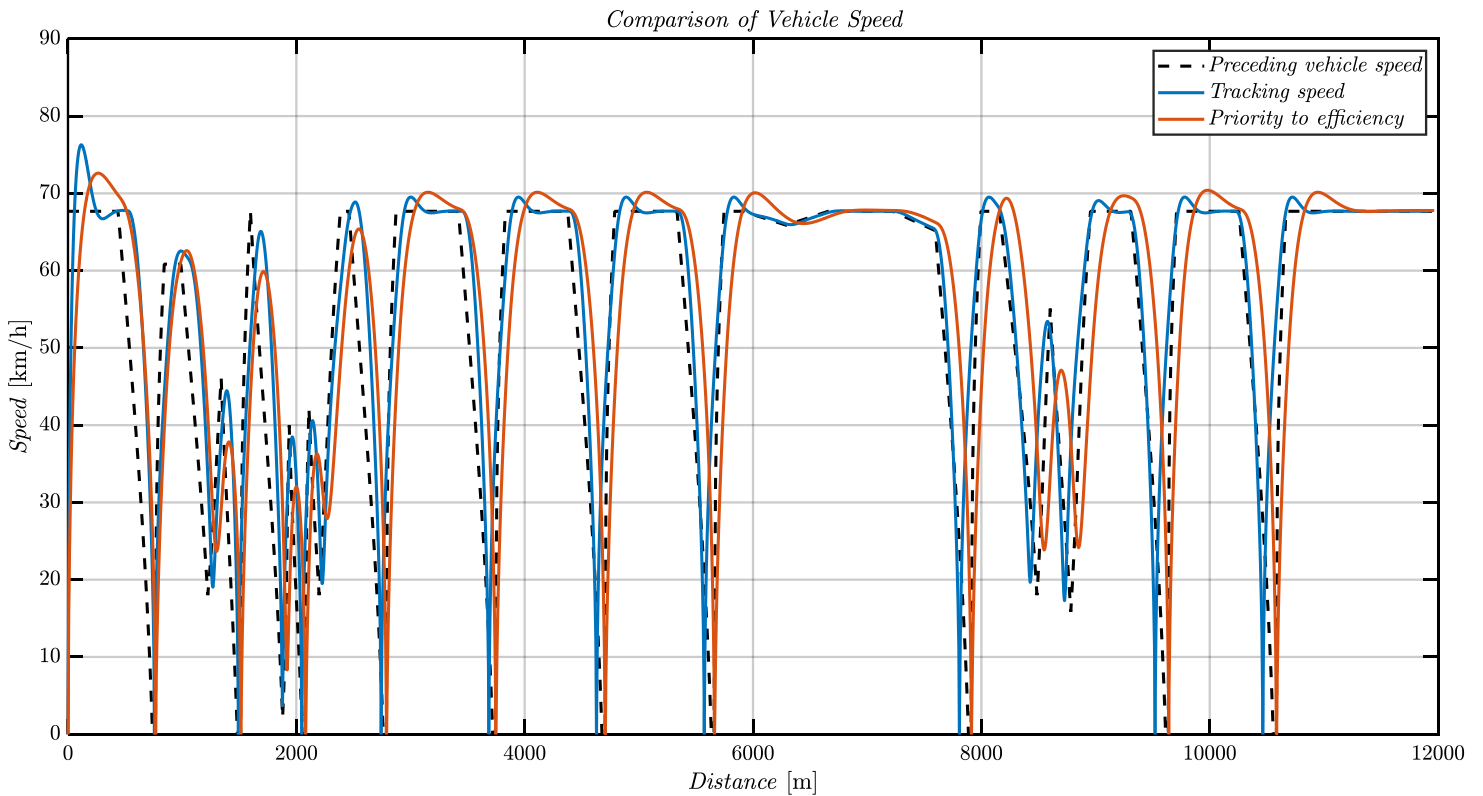


Figura 5.12: Profili di velocità nel test di speed tracking e nel caso di priorità data all'efficientamento

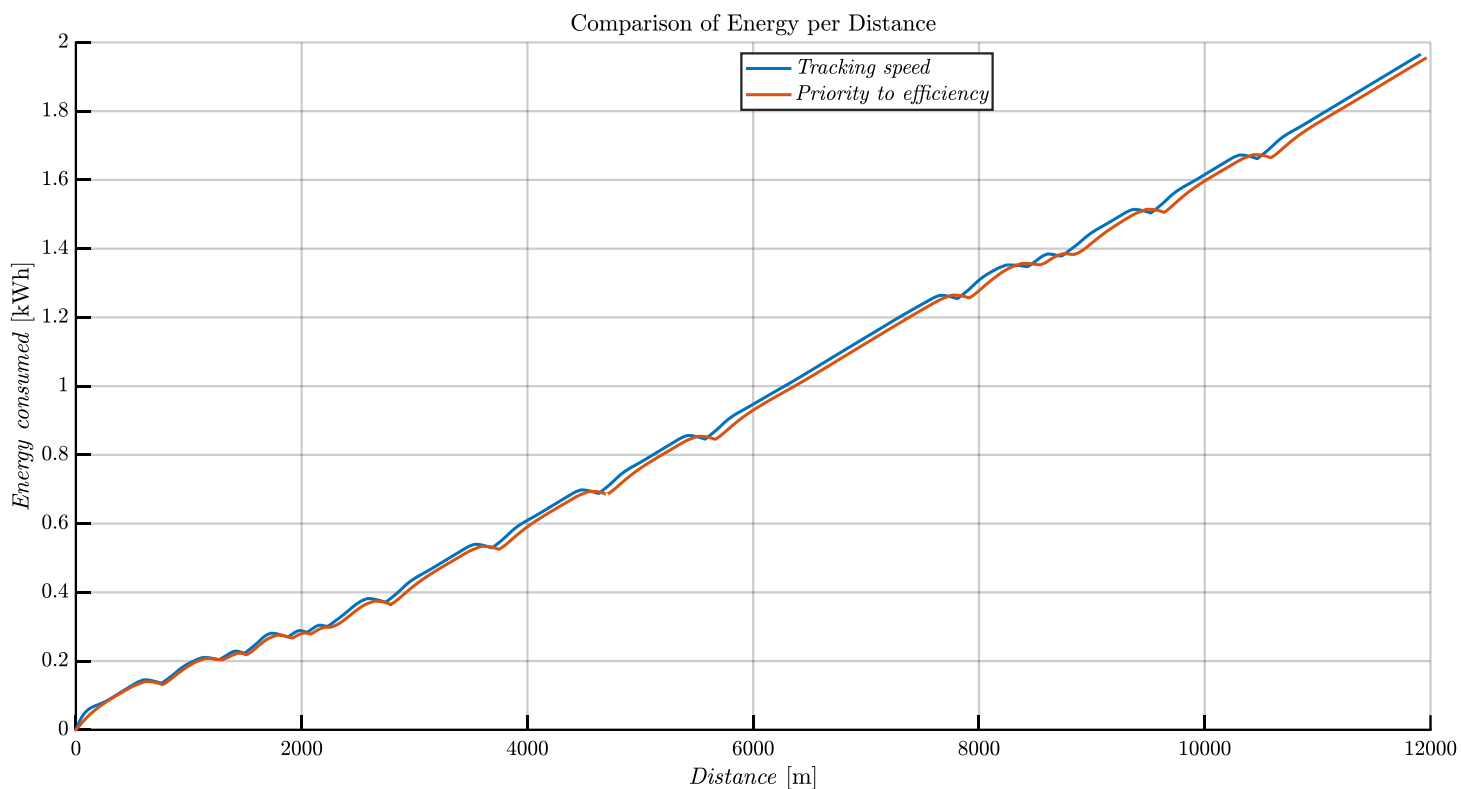


Figura 5.13: Andamenti dell'energia consumata nel test di speed tracking e nel caso di priorità data all'efficientamento

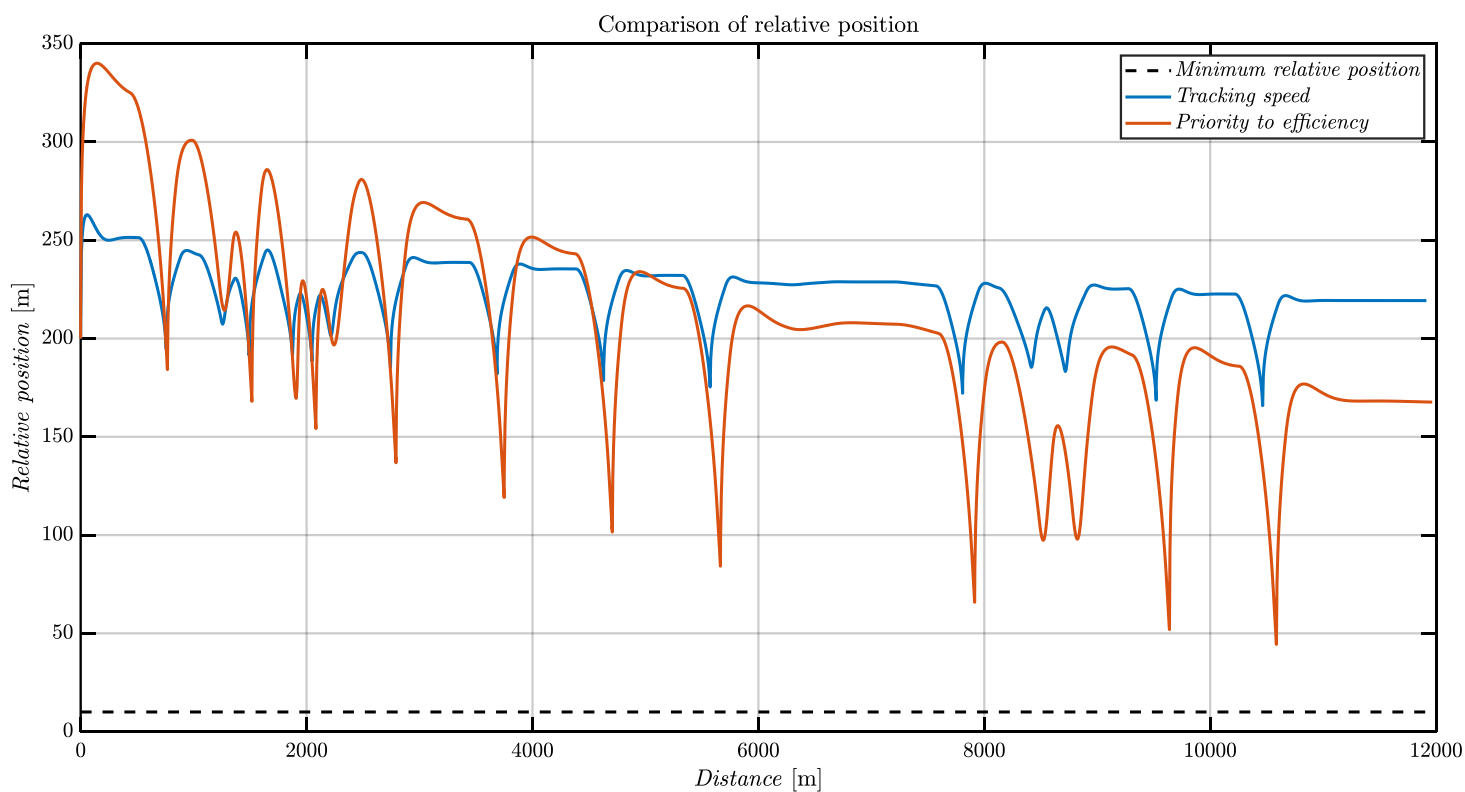


Figura 5.14: Andamenti della distanza relativa nel test di speed tracking e nel caso di priorità data all'efficientamento

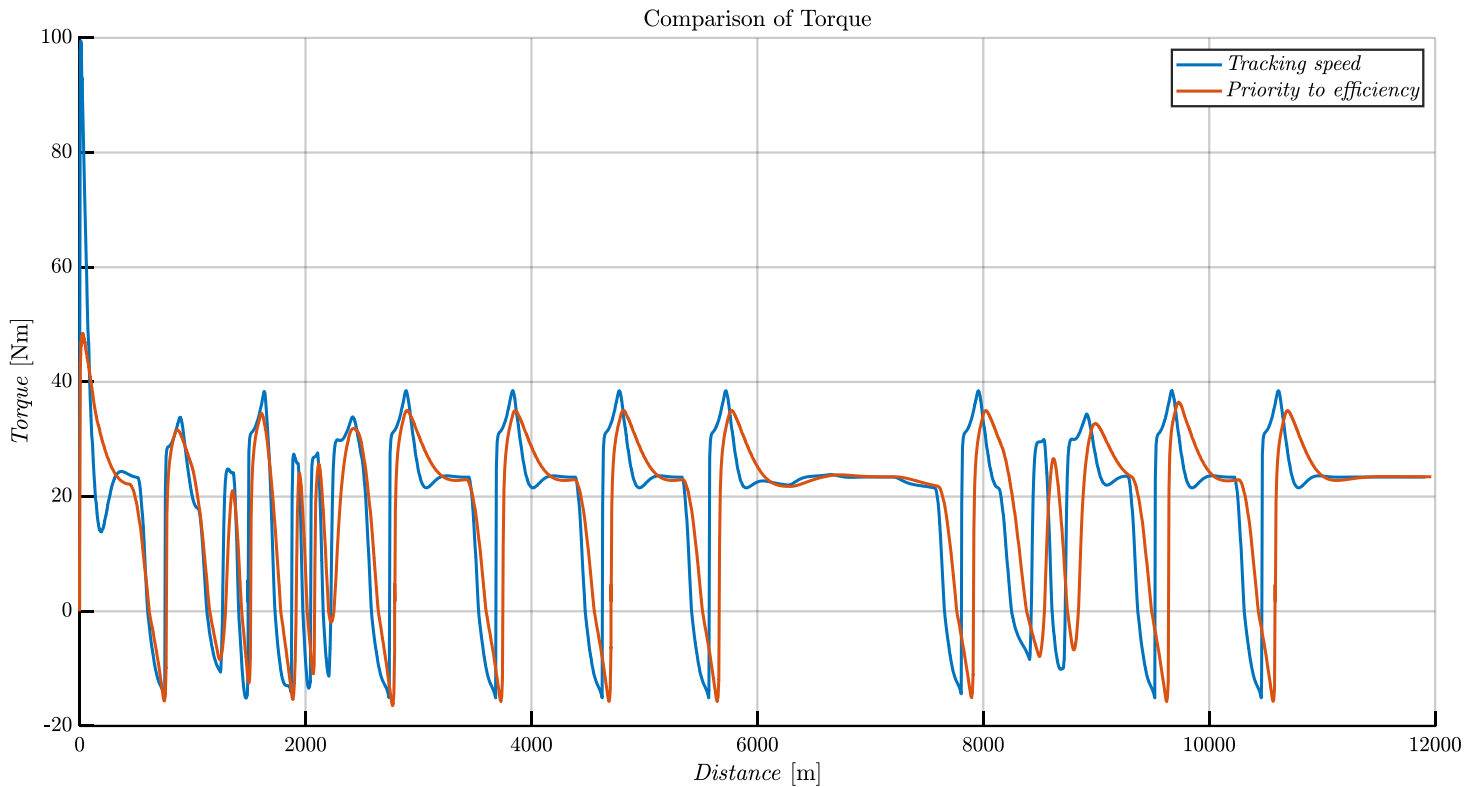


Figura 5.15: Profili di coppia nel test di speed tracking e nel caso di priorità data all'efficientamento

5.2.3 Risultati sul terzo scenario

Nell'internal model sono stati definiti i pesi dei termini costituenti la funzione di costo al fine di eseguire il test sul tracking della velocità; i pesi impostati sono identici a quelli visionabili nella Tabella 5.8.

I performance indicators ricavati sono riportati, invece, in Tabella 5.14:

Tabella 5.14: Performance indicators on the Tracking Speed test

Performance indicator	Symbol	Value	Unit
Average vehicle speed	v_{hmean}	74,59	[km/h]
Consumed energy	E_{tot}	5,9566	[kWh]
Mean efficiency	η_{mean}	72,69%	[-]
Drivetrain Power losses	P_{losses}	4142,6	[W]

Successivamente, sono stati variati i pesi dei termini costituenti la funzione di costo per effettuare il test nel quale viene data priorità ai termini di efficientamento energetico; i pesi impostati sono identici a quelli di Tabella 5.10.

In seguito, vengono riportati i relativi performance indicators e viene effettuato un confronto dei risultati evidenziando le variazioni percentuali, come mostrato in Tabella 5.15.

Tabella 5.15: Performance indicators relativi al test sul terzo scenario. Sono riportate le variazioni percentuali relative al confronto tra il caso in cui si dà priorità all'inseguimento della velocità e il caso in cui viene data priorità ai termini di efficientamento

Performance indicator	Symbol	Test with priority to efficiencies terms	Test with priority to speed tracking	Percentage change	Unit
Average vehicle speed	$v_{h_{mean}}$	73,40	74,59	1,60%	[km/h]
Consumed energy	E_{tot}	5,8080	5,9566	2,49%	[kWh]
Mean efficiency	η_{mean}	70,12%	72,69%	3,54%	[-]
Drivetrain Power losses	P_{losses}	4045,9	4142,6	2,33%	[W]

I risultati ottenuti sono in linea con le aspettative. Analizzando i dati si evince una riduzione dell'energia consumata del 2.49%, e l'andamento di quest'ultima in funzione della distanza percorsa è riportato in Figura 5.17. Il profilo arancio mostra una distanza relativa maggiore rispetto al profilo blu (Figura 5.18) il che significa che nel test con priorità all'efficientamento energetico l'ego vehicle sta andando a velocità inferiore per ridurre il consumo di energia. Infatti, questo si nota anche dal grafico che riporta i profili di velocità in Figura 5.16. Inoltre, il grafico che riporta la coppia in funzione della distanza percorsa (Figura 5.19) mostra, nel caso del profilo arancio relativo all'ottimizzazione volta all'efficientamento energetico, un andamento meno aggressivo della richiesta di coppia rispetto al profilo blu rappresentante il test inerente lo speed tracking.

L'efficienza globale della driveline peggiora del 3.54% e, collegato a questo valore, si ha un decremento della velocità media dell'ego vehicle dell'1.60%. Infatti, al diminuire del rendimento, i punti di lavoro sulla mappa motore tendono verso coppie e velocità di rotazione inferiori. Inoltre, si ha un decremento delle perdite di potenza del 2.33%.

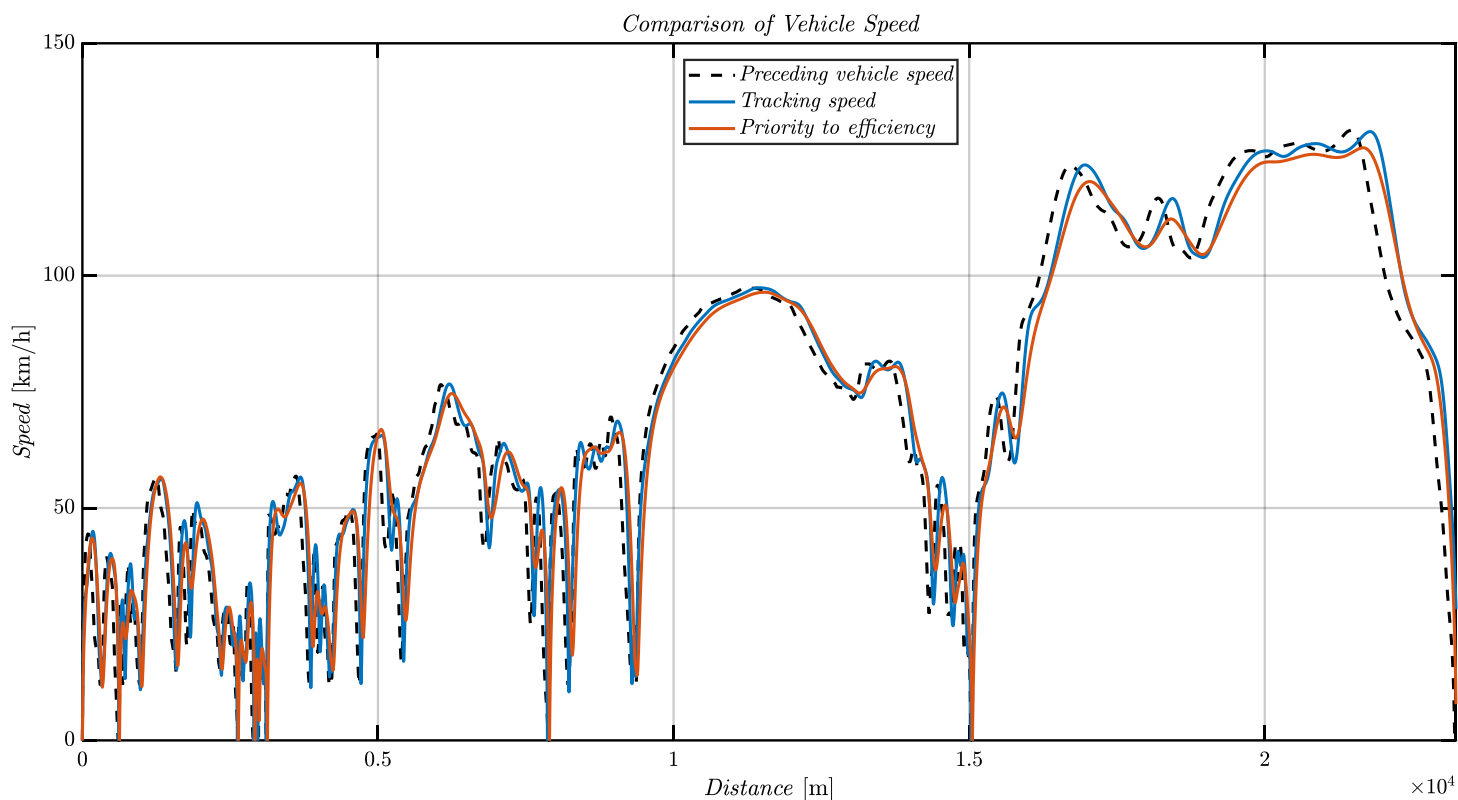


Figura 5.16: Profili di velocità nel test di speed tracking e nel caso di priorità data all'efficientamento

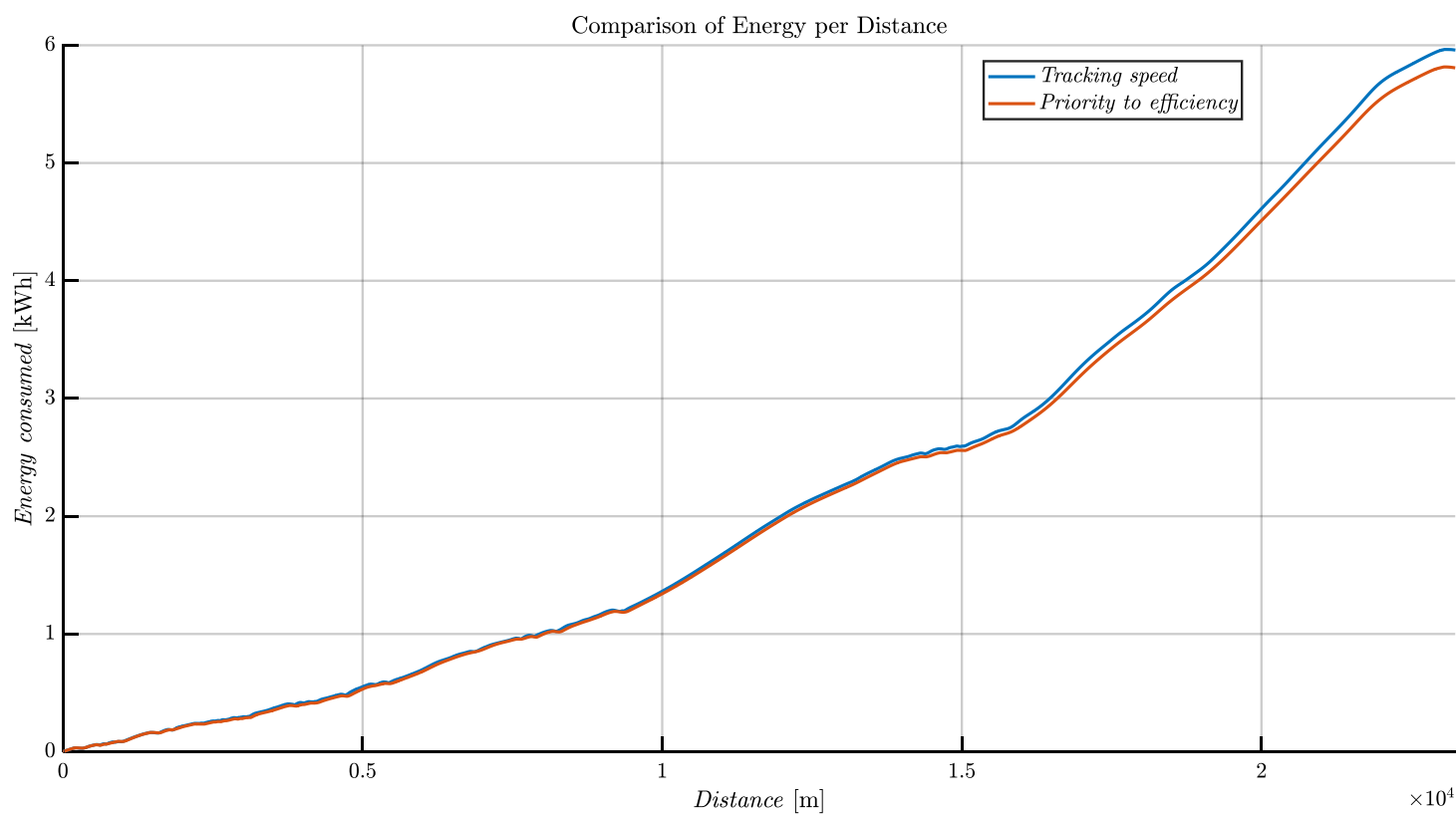


Figura 5.17: Andamenti dell'energia consumata nel test di speed tracking e nel caso di priorità data all'efficientamento

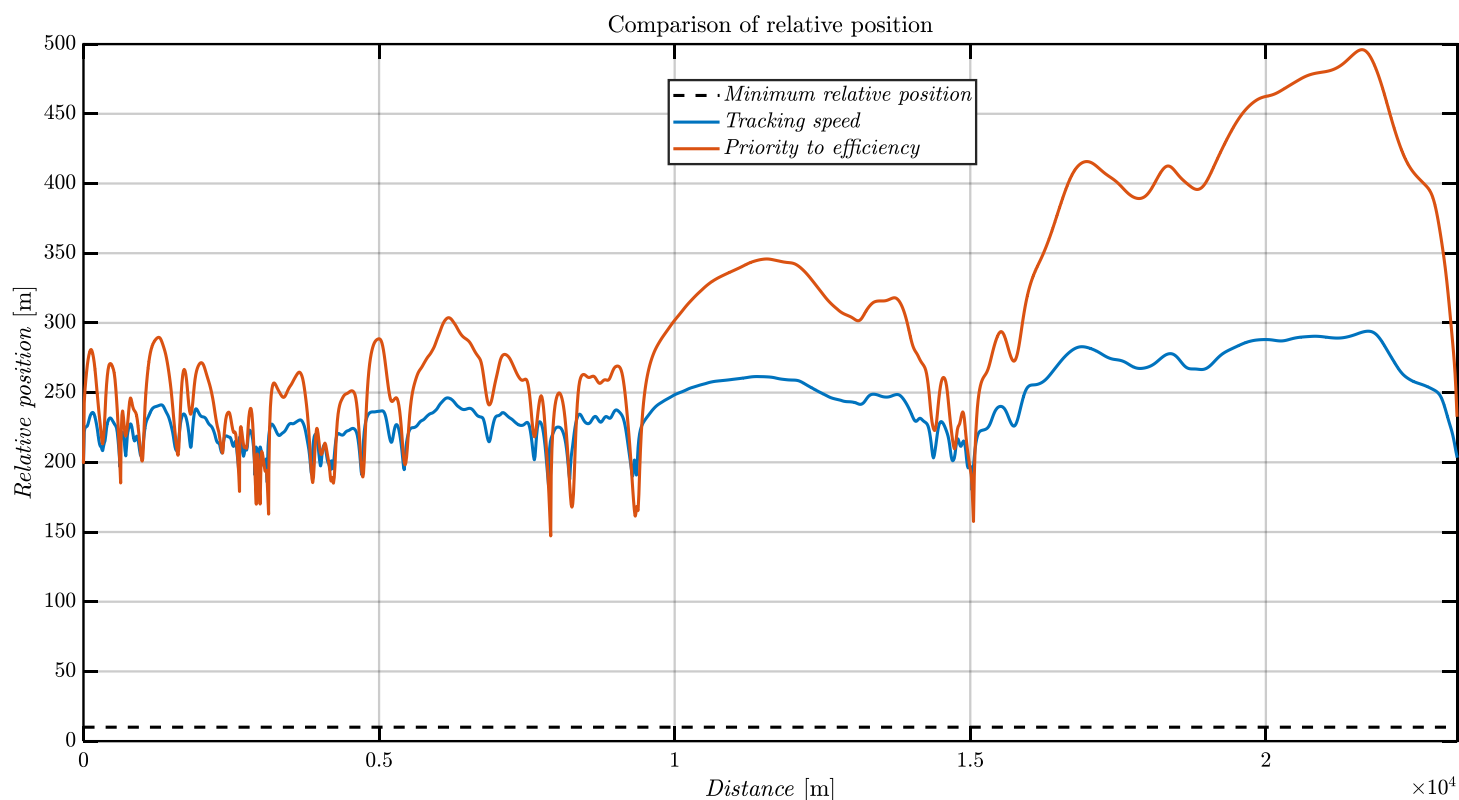


Figura 5.18: Andamenti della distanza relativa nel test di speed tracking e nel caso di priorità data all'efficientamento

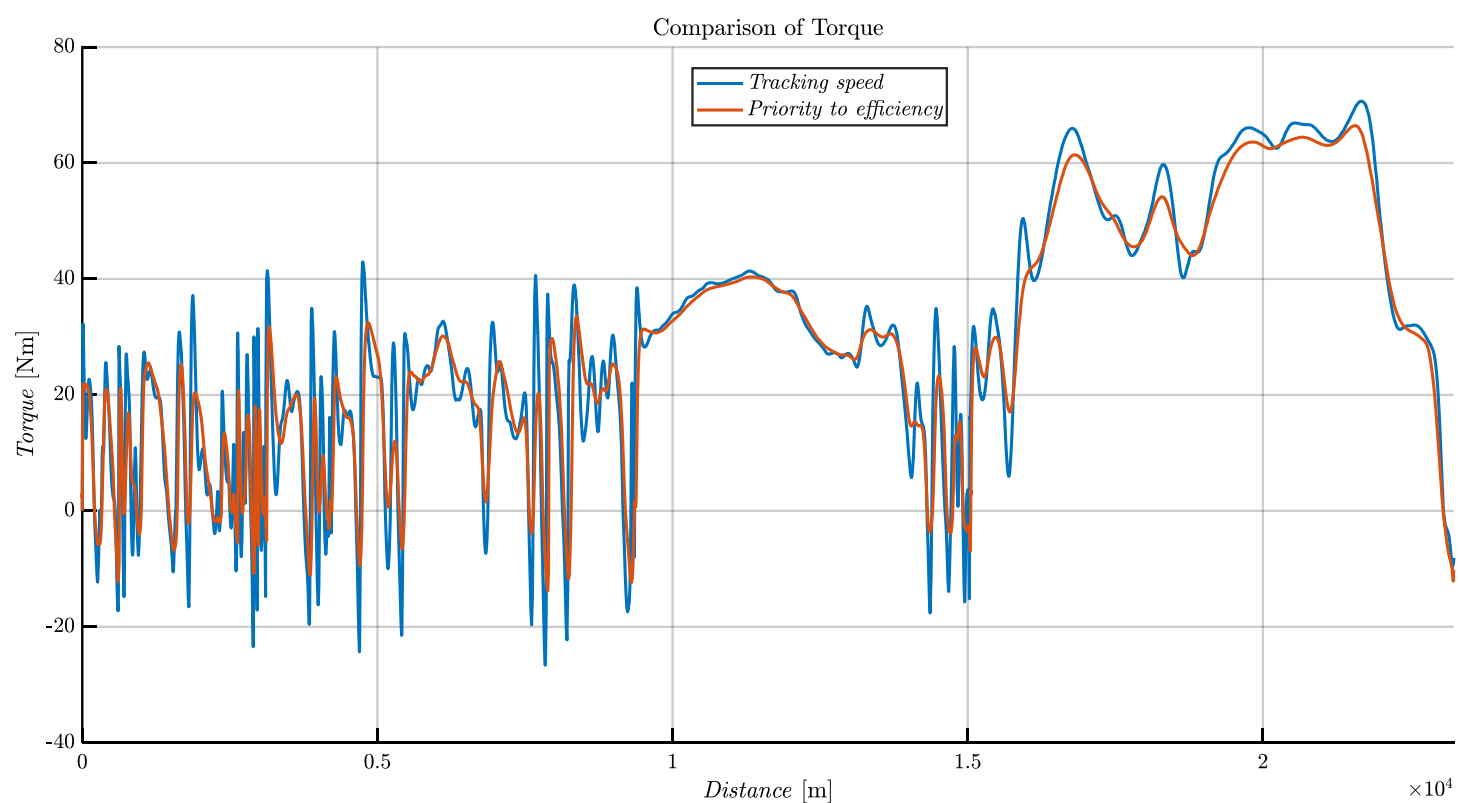


Figura 5.19: Profili di coppia nel test di speed tracking e nel caso di priorità data all'efficientamento

6. Conclusione

In questa tesi è stato sviluppato un Energy Efficient Adaptive Cruise Control basato su un controllore Model Predictive Control con l'obiettivo di minimizzare il consumo energetico. L'MPC è caratterizzato da un prediction horizon di 20 step, ciascuno della durata di 200 ms, per un orizzonte di predizione complessivo di 4 s. La funzione di costo è stata progettata in modo da ottenere un controllo sulla velocità dell'ego vehicle, sulla forza longitudinale applicata e sulla sua variazione, sull'accelerazione dell'ego vehicle, sulla forza richiesta al controllore. Sono stati inclusi nella funzione di costo i seguenti parametri energetici:

- potenza assorbita dal motore elettrico;
- efficienza globale della powertrain;
- perdite di potenza lungo la powertrain;
- energia consumata per unità di distanza.

I termini relativi alla derivata della forza longitudinale e all'accelerazione dell'ego vehicle sono stati inseriti per limitare le oscillazioni nella velocità.

L'MPC rappresenta una soluzione ideale per la gestione dell'EACC grazie alla sua capacità di considerare esplicitamente i vincoli del sistema in modo da garantire sicurezza e comfort. Inoltre, la taratura dei pesi relativi ai termini costituenti la funzione di costo permette di bilanciare le diverse priorità di guida.

L'ottimizzazione dei pesi caratterizzanti i termini nella funzione di costo, è stata effettuata tramite l'utilizzo dell'algoritmo di ottimizzazione basato sulla funzione *surrogateopt*. Tramite questo algoritmo sono stati individuati due set distinti di pesi: uno finalizzato al tracking della velocità e uno volto a massimizzare l'efficienza energetica. Questi set di pesi sono stati testati in tre scenari sperimentali:

1. scenario di accelerazione di 1 km;
2. scenario ricavato dalla letteratura di circa 12 km;
3. ciclo guida WLTP.

I risultati ottenuti sul secondo e terzo scenario confermano i benefici dell'approccio adottato. Infatti, si registra una riduzione del consumo energetico, una riduzione del rendimento medio a cui è associata una velocità media inferiore e una diminuzione delle perdite di potenza, confermando il corretto funzionamento della strategia di ottimizzazione. Come già detto in precedenza, il peggioramento del rendimento determina un decremento nella velocità dell'ego vehicle dovuta al fatto che i punti sulla mappa motore tendono verso valori di coppia e velocità di rotazione inferiori. Tuttavia, nel primo scenario di accelerazione di 1 km, non si riscontrano miglioramenti significativi. L'energia consumata rimane invariata, il rendimento peggiora leggermente dello 0.30%, la velocità media dell'ego vehicle diminuisce dello 0.17% e le perdite di potenza aumentano dello 0.16%. Questa mancanza di effetti rilevanti potrebbe essere dovuta alla brevità del test, insufficiente per apprezzare le dinamiche di ottimizzazione energetica.

Il presente lavoro può essere ulteriormente sviluppato integrando un modello della batteria, valutando la variazione della resistenza interna in funzione della temperatura e analizzando l'andamento dello State of Charge (SoC) per comprendere meglio l'impatto dell'EACC sulla gestione energetica complessiva del veicolo. Successivamente si potrebbe considerare l'introduzione di strategie di controllo per modificare dinamicamente i pesi della funzione di costo in base alle condizioni di guida e allo stato di carica della batteria. Un ulteriore sviluppo potrebbe riguardare l'integrazione con dati di connettività V2X, in modo da adattare il controllo in funzione delle condizioni del traffico e delle informazioni provenienti dall'infrastruttura stradale, migliorando ulteriormente l'efficienza e la sicurezza del sistema.

In conclusione, i risultati ottenuti dimostrano il potenziale ottenibile dall'approccio basato sul Model Predictive Control nella gestione di un Energy Efficient Adaptive Cruise Control, evidenziando la possibilità di ottenere un miglioramento nella gestione energetica che potrebbe portare ad una ottimizzazione del consumo energetico nei veicoli elettrificati.

Ringraziamenti

Ringrazio Gianluca, Luca, Fabio e i professori che mi hanno supportato nello sviluppo di questo lavoro; e un grazie a chi c'è stato durante tutto il percorso che mi ha portato fino a qui.

Bibliografia

- [1] “Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles,” Society of Automotive Engineers, Warrendale, PA, USA, Tech. Rep. J3016_202104, Apr. 2021.
- [2] SAE, «Normativa SAE J3061_202104,» 2021. [Online]. Available: https://www.sae.org/standards/content/j3016_202104/.
- [3] S. Aldo, *Technologies for autonomous vehicles: vehicle dynamics – adaptive cruise control and autonomous emergency braking*, Torino, 2023.
- [4] Guodong Yin *et al.*, “Collaborative Optimization of Energy Management Strategy and Adaptive Cruise Control Based on Deep Reinforcement Learning”, *ieee transactions on transportation electrification*, vol. 9, no. 1, march 2023.
- [5] Yinglong He *et al.*, “Adaptive Cruise Control Strategies Implemented on Experimental Vehicles: A Review”, *IFAC PapersOnLine* 52-5 (2019) 21–27.
- [6] Yang, Z.; Wang, Z.; Yan, M., “An Optimization Design of Adaptive Cruise Control System Based on MPC and ADRC”. *Actuators* 2021, 10, 110. <https://doi.org/10.3390/act10060110>
- [7] Yu, L. and Wang, R., 2022. “Researches on adaptive cruise control system: A state of the art review”. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 236(2-3), pp. 211-240. <https://doi.org/10.1177/09544070211019254>.
- [8] Sheng Zhang and Xiangtao Zhuan. “Study on Adaptive Cruise Control Strategy for Battery Electric Vehicle Considering Weight Adjustment”. *Symmetry* 2019, 11, 1516; doi:10.3390/sym11121516
- [9] Zhu, Z., Bei, S., Li, B., Liu, G., Tang, H., Zhu, Y., and Gao, C., 2023. “Research on robust control of intelligent vehicle adaptive cruise”. *World Electric Vehicle Journal*. 14(10), P. 268. <https://doi.org/10.3390/wevj14100268>.
- [10] Y. Jia, T. Saito, Y. Itoh, Y. Nukezhanov, and D. Görges, “Energy-optimal adaptive cruise control in time domain based on model predictive control,” *IFAC-PapersOnLine*, vol. 51, no. 31, pp. 846–853, 2018.
- [11] S. Cheng, L. Li, M.-M. Mei, Y.-l. Nie, and L. Zhao, “Multiple-objective adaptive cruise control system integrated with direct yaw moment control,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4550–4559, May 2019.
- [12] A. Biswas and A. Emadi, “Energy management systems for electrified powertrains: State-of-the-art review and future trends,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 6453–6467, Jul. 2019.
- [13] D. Bianchi *et al.*, “A rule-based strategy for a series/parallel hybrid electric vehicle: an approach based on dynamic programming,” in *Proc. ASME Dyn. Syst. Control Conf.* 2010, pp. 507–514.

- [14] R. Biasini, S. Onori, and G. Rizzoni, “A near-optimal rule-based energy management strategy for medium duty hybrid truck,” *Int. J. Powertrains*, vol. 2, no. 2-3, pp. 232–261, 2013.
- [15] J. Peng, H. He, and R. Xiong, “Rule based energy management strategy for a series–parallel plug-in hybrid electric bus optimized by dynamic programming,” *Appl. Energy*, vol. 185, pp. 1633–1643, 2017.
- [16] E. Mumolo, Lezione: “Agenti che ragionano logicamente; Logica Fuzzy”. <https://sites.units.it>
- [17] M. Schwenzer, M. Ay, T. Bergs, D. Abel, “Review on model predictive control: an engineering perspective,” 11 2021.
- [18] Chaofeng Pan, Aibao Huang, Jian Wang, Liao Chen, Jun Liang, Weiqi Zhou, Limei Wang, Jufeng Yang, “Energy-optimal adaptive cruise control strategy for electric vehicles based on model predictive control”, *Energy*, Volume 241, 2022, 122793, ISSN 0360-5442, <https://doi.org/10.1016/j.energy.2021.122793>.
- [19] Anye Zhou, Zejiang Wang, Adian Cook, “Model Predictive Control-Based Trajectory Shaper for Safe and Efficient Adaptive Cruise Control”, *2023 IEEE International Automated Vehicle Validation Conference (IAVVC)*.
- [20] Xiaohai Lin, Daniel Görges, Andreas Weißmann, “Simplified Energy-Efficient Adaptive Cruise Control based on Model Predictive Control”, *IFAC PapersOnLine* 50-1 (2017) 4794–4799.
- [21] F. Xu, Z. Guo, H. Chen, D. Ji, and T. Qu, “A custom parallel hardware architecture of nonlinear model-predictive control on fpga,” *IEEE Transactions on Industrial Electronics*, vol. 69, pp. 11569–11579, 11 2022.
- [22] H. Ferreau, A. Potschka, and C. Kirches, “qpOASES webpage.” <http://www.qpOASES.org/>, 2007–2014.
- [23] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, “OSQP: an operator splitting solver for quadratic programs,” *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020.
- [24] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. van Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, and M. Diehl, “acados – a modular open-source framework for fast embedded optimal control,” *Mathematical Programming Computation*, Oct 2021.
- [25] J. Lofberg, “YALMIP : a toolbox for modeling and optimization in matlab,” in *2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No.04CH37508)*, pp. 284–289, 2004.
- [26] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, “CasADi – A software framework for nonlinear optimization and optimal control,” *Mathematical Programming Computation*, In Press, 2018.

- [27] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, “Casadi: a software framework for nonlinear optimization and optimal control,” *Mathematical Programming Computation*, vol. 11, no. 1, pp.1–36, 2019.
- [28] A. Khajepour, S. Fallah, and A. Goodarzi, “Electric and Hybrid Vehicles Technologies, Modeling and Control: A Mechatronic Approach”. Wiley, 2014.
- [29] Minghao Shen, Robert Austin Dollar, Tamas G. Molnar, Chaozhe R. He, Ardalan Vahidi, Gábor Orosz, “Energy-Efficient Reactive and Predictive Connected Cruise Control”, *IEEE Transactions on intelligent vehicles*, vol. 9, no. 1, January 2024.
- [30] <https://it.mathworks.com/help/gads/surrogateopt.html>.
- [31] Van Hegelsom, J., “Real-Time MPC Strategy for Predictive Adaptive Cruise Control”, Master Thesis, 28 Nov 2023.
- [32] Yuxuan Liu, Jialing He, Jianjun Cai, Yitao Wu, Jie Li, Yonggang Liu, “Energy-Optimal Adaptive Cruise Control for Electric Vehicles Based on Model Predictive Control and Dynamic Programming”, *Proceedings of the 2023 7th CAA International Conference on Vehicular Control and Intelligence (CVCI)*, Changsha, China, October 27-29, 2023.
- [33] Qitao Li, Changfu Gong, Yuan Lin, “Co-Optimization of Adaptive Cruise Control and Hybrid Electric Vehicle Energy Management via Model Predictive Mixed Integer Control”, *Proceedings of the 42nd Chinese Control Conference*, July 24-26, 2023, Tianjin, China.
- [34] Borneo, A.; Zerbato, L.; Miretti, F.; Tota, A.; Galvagno, E.; Misul, D.A.; “Platooning Cooperative Adaptive Cruise Control for Dynamic Performance and Energy Saving: A Comparative Study of Linear Quadratic and Reinforcement Learning-Based Controllers”. *Appl. Sci.* 2023, 13, 10459. <https://doi.org/10.3390/app131810459>.
- [35] Federico Millo, Lezione: “Emissioni inquinanti, Introduzione, Il contesto ambientale e normativo”, Propulsori termici.