

POLITECNICO DI TORINO

Laurea Magistrale in Ingegneria Meccanica

Propulsione dei veicoli terrestri



Tesi di Laurea Magistrale

CONFRONTO TRA LE STRATEGIE DI CONTROLLO PID, LMPC E NMPC APPLICATE AL PATH TRACKING

Relatore

Prof. Aldo SORNIOTTI

Corelatori

Cecilia FORMENTO

Marino Massimo COSTANTINI

Candidato

Riccardo MOCAVERO

23/07/2025

Indice

Introduzione	5
---------------------------	---

1 Stato dell'Arte	6
--------------------------------	---

1.1 Evoluzione e stato dell'arte dei veicoli a guida autonoma	6
---	---

1.2 Il path tracking nei veicoli autonomi.....	8
--	---

1.3 Tecniche di controllo nel path tracking.....	10
--	----

1.3.1 Controllori classici	10
----------------------------------	----

1.3.2 Controllori predittivi.....	10
-----------------------------------	----

1.3.3 Tecniche avanzate e ibride	11
--	----

1.4 Confronto in letteratura tra PID, Pole Placement, LQR, LMPC e NMPC	12
--	----

1.4.1 Controllore PID	12
-----------------------------	----

1.4.2 Controllore Pole Placement	13
--	----

1.4.3 Controllore LQR.....	14
----------------------------	----

1.4.4 Controllore LMPC	15
------------------------------	----

1.4.5 Controllore NMPC	16
------------------------------	----

2 Metodologie	18
----------------------------	----

2.1 Ambiente di simulazione: MATLAB e Simulink	19
--	----

2.2 ACADOS tool-kit.....	21
--------------------------	----

2.3 Modello di validazione	22
----------------------------------	----

2.3.1 Dinamica del veicolo.....	23
---------------------------------	----

2.3.2 Generazione della traiettoria di riferimento	24
--	----

2.3.3 Calcolo degli errori cinematici.....	24
--	----

2.4 Modello interno LMPC.....	25
-------------------------------	----

2.5 Modello interno NMPC.....	29
-------------------------------	----

2.6 Struttura del codice di simulazione	30
--	----

2.6.1 Script principale di simulazione ...	30
--	----

2.6.2 Script di generazione della traiettoria.....	30
--	----

2.7 Layout del percorso di prova	31
--	----

2.8 Key Performance Indicators	32
--------------------------------------	----

3 Controllori	33
----------------------------	----

3.1 Controllore PID	33
---------------------------	----

3.1.1 Vincoli sull'attuatore	33
------------------------------------	----

3.1.2 Risultati delle simulazioni PID	34
---	----

3.1.3 Considerazioni	41
----------------------------	----

3.2 Controllore LMPC.....	42
---------------------------	----

3.2.1 Configurazione del controllore	42
---	----

3.2.2 Risultati delle simulazioni LMPC ..	44
---	----

3.2.3 Considerazioni	51
----------------------------	----

3.3 Controllore NMPC.....	52
---------------------------	----

4 Analisi dei KPI – Confronto tra PID e LMPC	54
---	----

4.1 Errore laterale massimo	54
-----------------------------------	----

4.2 Errore angolare massimo.....	55
----------------------------------	----

4.3 RMS dell'errore laterale.....	56
-----------------------------------	----

4.4 RMS dell'errore angolare.....	57
-----------------------------------	----

5 Conclusioni	58
----------------------------	----

Bibliografia	60
---------------------------	----

Indice delle figure

Figura 1: schema path tracking	8
Figura 2: sviluppo di un sistema di controllo.....	19
Figura 3: sistema di riferimento del veicolo	25
Figura 4: sistema di riferimento della ruota.....	25
Figura 5: cambio di coordinate.....	27
Figura 6: layout del percorso di prova.....	31
Figura 7: PID, v_x e v_y nel tempo	34
Figura 8: PID, andamento del tracking	34
Figura 9: PID, andamento degli errori nel tempo.....	34
Figura 10: PID, v_x e v_y nel tempo	35
Figura 11: PID, andamento del tracking	35
Figura 12: PID, andamento degli errori nel tempo.....	35
Figura 13: PID, v_x e v_y nel tempo	36
Figura 14: PID, andamento del tracking	37
Figura 15: PID, andamento degli errori nel tempo.....	36
Figura 16: PID, v_x e v_y nel tempo	37
Figura 17: PID, andamento del tracking	37
Figura 18: PID, andamento degli errori nel tempo.....	37
Figura 19: PID, v_x e v_y nel tempo	38
Figura 20: PID, andamento del tracking	38
Figura 21: PID, andamento degli errori nel tempo.....	38
Figura 22: PID, v_x e v_y nel tempo	39
Figura 23: PID, andamento del tracking	39
Figura 24: PID, andamento degli errori nel tempo.....	39
Figura 25: PID, v_x e v_y nel tempo	40
Figura 26: PID, andamento del tracking	40
Figura 27: PID, andamento degli errori nel tempo.....	40
Figura 28: LMPC, v_x e v_y nel tempo	44
Figura 29: LMPC, andamento del tracking	44
Figura 30: LMPC, andamento degli errori nel tempo.....	44
Figura 31: LMPC, v_x e v_y nel tempo	45
Figura 32: LMPC, andamento del tracking	45
Figura 33: LMPC, andamento degli errori nel tempo.....	45
Figura 34: LMPC, v_x e v_y nel tempo	46
Figura 35: LMPC, andamento del tracking	46
Figura 36: LMPC, andamento degli errori nel tempo.....	46
Figura 37: LMPC, v_x e v_y nel tempo	47
Figura 38: LMPC, andamento del tracking	47
Figura 39: LMPC, andamento degli errori nel tempo.....	47
Figura 40: LMPC, v_x e v_y nel tempo	48
Figura 41: LMPC, andamento del tracking	48
Figura 42: LMPC, andamento degli errori nel tempo.....	48

Figura 43: LMPC, v_x e v_y nel tempo	49
Figura 44: LMPC, andamento del tracking	49
Figura 45: LMPC, andamento degli errori nel tempo.....	49
Figura 46: LMPC, v_x e v_y nel tempo	50
Figura 47: LMPC, andamento del tracking	50
Figura 48: LMPC, andamento degli errori nel tempo.....	50
Figura 49: errore laterale massimo	54
Figura 50: errore angolare massimo	55
Figura 51: RMS dell'errore laterale	56
Figura 52: RMS dell'errore angolare	57

Indice delle tabelle

Tabella 1: sezioni del percorso di riferimento.....	31
Tabella 2: guadagni del PID	33
Tabella 3: setup LMPC.....	42

Introduzione

Negli ultimi anni, il settore dei veicoli autonomi ha conosciuto un rapido sviluppo, spinto dalla crescente domanda di soluzioni di mobilità sicure, efficienti e automatizzate. Una delle sfide principali nella progettazione di veicoli autonomi riguarda la capacità di seguire traiettorie predefinite con precisione, garantendo stabilità, comfort e rispetto dei vincoli ambientali e normativi.

In questo contesto, il problema del path tracking riveste un ruolo centrale. Il path tracking consiste nel seguire una traiettoria di riferimento minimizzando lo scostamento laterale e longitudinale e mantenendo l'orientamento desiderato, in presenza di vincoli di velocità, accelerazione e sicurezza. Si tratta di un problema di controllo dinamico complesso, che richiede al veicolo di correggere continuamente la propria posizione e direzione in funzione della traiettoria prestabilita reagendo a disturbi esterni, incertezze del modello e variazioni dell'ambiente circostante. Questo problema è stato affrontato in letteratura mediante diverse tecniche di controllo, ognuna caratterizzata da un differente compromesso tra prestazioni, robustezza e complessità computazionale. Tra i metodi più diffusi si trovano i controllori a retroazione classica come il PID, i controllori lineari predittivi (LMPC) e i controllori predittivi non lineari (NMPC).

L'obiettivo principale di questa tesi è analizzare e confrontare il comportamento di queste tre categorie di controllori, applicate a un modello dinamico di veicolo autonomo che percorre un tracciato definito e soggetto a vincoli normati. In particolare, l'analisi si concentra sulla capacità di ciascun controllore di mantenere la traiettoria, gestire perturbazioni e rispettare i vincoli.

1 Stato dell'Arte

1.1 Evoluzione e stato dell'arte dei veicoli a guida autonoma

Negli ultimi decenni, il tema della guida autonoma è emerso come uno dei più importanti ambiti di ricerca e sviluppo nell'ingegneria dei trasporti e dell'automazione. L'obiettivo è quello di realizzare veicoli in grado di muoversi in completa autonomia, percependo l'ambiente circostante, pianificando percorsi e prendendo decisioni in tempo reale, senza l'intervento umano.

I primi contributi significativi risalgono agli anni '80 e '90, con prototipi sperimentali sviluppati in centri di ricerca come l'Università di Carnegie Mellon e il progetto europeo PROMETHEUS [1]. Tuttavia, un punto di svolta è stato il DARPA Grand Challenge del 2004, competizione internazionale organizzata dal Dipartimento della Difesa statunitense, che ha dimostrato per la prima volta la possibilità di affrontare percorsi su terreni accidentati con veicoli autonomi. Il successo di Stanford Racing Team e Carnegie Mellon nel DARPA Urban Challenge 2007 ha ulteriormente accelerato l'interesse verso questa tipologia [2].

Negli anni successivi, grandi aziende automotive e tecnologiche hanno avviato programmi di ricerca avanzati: Waymo (gruppo Alphabet) ha avviato sperimentazioni di taxi autonomi a Phoenix (Arizona) [3], Tesla ha introdotto funzionalità di guida assistita con il sistema Autopilot (livello 2) [4], Cruise (General Motors), Aurora, Zoox (Amazon) e Baidu Apollo in Cina sono altri esempi di aziende che hanno investito fortemente nel settore [5].

La classificazione dei veicoli a guida autonoma è definita dallo standard SAE J3016 [6], che distingue sei livelli di automazione, dal livello 0 (assenza di automazione) al livello 5 (guida completamente autonoma in qualsiasi scenario operativo). Attualmente i sistemi più avanzati disponibili sul mercato si collocano tra i livelli 2 e 3, in cui il veicolo può eseguire alcune manovre in autonomia ma la supervisione attiva del conducente è ancora necessaria. La ricerca industriale e accademica è invece focalizzata al raggiungimento del livello 4 (autonomia completa in scenari specifici) e, a lungo termine, del livello 5.

Dal punto di vista dell'architettura di controllo, un sistema di guida autonoma si compone di quattro sottosistemi principali [7], dove viene evidenziato il ruolo centrale di ciascun modulo nel garantire un comportamento sicuro e predicibile del veicolo:

1. Percezione: riguarda la raccolta e l'elaborazione delle informazioni tramite sensori come lidar (Light Detection and Ranging), radar millimetrici, camere e moduli GPS-IMU per la stima della posizione [8].
2. Localizzazione: consente di determinare con precisione la posizione del veicolo rispetto all'ambiente e alle mappe utilizzando algoritmi di fusione sensoriale come il Monte Carlo Localization o EKF-SLAM [9].

3. Pianificazione: definisce la traiettoria ottimale da seguire, tenendo conto di vincoli ambientali, regole stradali e ostacoli. I planner più diffusi includono algoritmi come Hybrid A*, RRT* e MPC-based planners [10].
4. Controllo: si occupa di tradurre la traiettoria pianificata in comandi di attuazione (sterzo, accelerazione e frenata), garantendo precisione, stabilità e risposta dinamica ottimale.

Nonostante i notevoli progressi, permangono ancora diverse sfide aperte, la sicurezza dei veicoli autonomi non può prescindere da un approccio multidisciplinare che includa aspetti ingegneristici, etici e normativi [11]. Analogamente, la robustezza dei controllori è un elemento cruciale per garantire che il veicolo segua con precisione le traiettorie previste, anche in presenza di disturbi o variazioni delle condizioni operative [12].

1.2 Il path tracking nei veicoli autonomi

Il path tracking è una delle funzioni più importanti per i veicoli a guida autonoma. Con questo termine si intende la capacità del veicolo di seguire un percorso di riferimento definito in fase di pianificazione, mantenendo uno scostamento minimo in termini di posizione laterale e angolo di orientamento rispettando i vincoli prestabiliti. Ad esempio, in uno scenario urbano, un veicolo autonomo che percorre una curva deve seguire con precisione la corsia, evitando di oltrepassare la linea di mezzzeria o avvicinarsi troppo al bordo del marciapiede. Il path tracking consente di mantenere la traiettoria prevista anche in presenza di variazioni di velocità o fondo stradale irregolare.

La traiettoria può essere generata in tempo reale sulla base delle informazioni di percezione e localizzazione oppure predefinita in funzione di un percorso pianificato a priori. L'obiettivo è garantire il tracciamento preciso del percorso di riferimento, minimizzando le oscillazioni indesiderate e i ritardi eccessivi, garantendo stabilità e comfort per i passeggeri.

Dal punto di vista matematico, il problema del path tracking, può essere formalizzato come la minimizzazione di due variabili definite come errori:

- l'errore laterale e_y , ossia la distanza trasversale rispetto alla traiettoria di riferimento;
- l'errore angolare (heading error) e_ψ , cioè la differenza tra l'orientamento desiderato e quello attuale del veicolo.

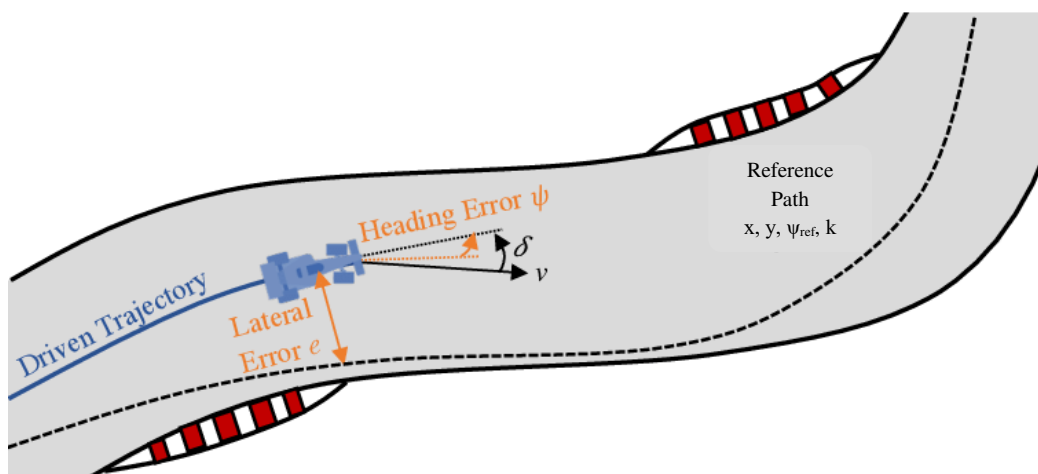


Figura 1: schema path tracking

Tali indicatori sono comunemente utilizzati come parametri di riferimento nella valutazione delle prestazioni dei controllori [13]. La progettazione di controllori efficaci per il path tracking si basa su modelli dinamici del veicolo, che variano in complessità a seconda dell'approccio adottato. Uno dei modelli più utilizzati è il modello bicicletta, che consente di rappresentare il comportamento del veicolo semplificando le due ruote anteriori e le due posteriori in un'unica ruota virtuale anteriore e posteriore. Tale modello, è adatto per

velocità moderate e consente di descrivere la relazione tra comandi di sterzo e variazione della posizione e dell'orientamento in modo accurato [14].

Oltre al modello bicicletta, esistono modelli dinamici più avanzati che tengono conto di effetti aggiuntivi come la dinamica laterale completa, le forze d'aderenza pneumatiche e le variazioni di carico sugli assi. In questi modelli più complessi, le equazioni di moto vengono derivate a partire dalla seconda legge di Newton applicata separatamente in direzione longitudinale, laterale e attorno all'asse di imbardata. La rappresentazione più completa può includere la modellizzazione delle forze laterali sulle gomme utilizzando modelli empirici, come il modello di Pacejka, che consente di descrivere il comportamento non lineare del pneumatico in funzione dello slip angle [15]. Inoltre, le accelerazioni trasversali e longitudinali non sono più considerate trascurabili, e il controllo deve gestire forti non linearità e saturazioni.

Tali modelli dinamici avanzati trovano applicazione soprattutto nello sviluppo di controllori di tipo non lineare o più avanzati, nella simulazione di scenari ad alta velocità o in manovre dinamiche critiche. Sebbene siano più onerosi dal punto di vista computazionale, offrono una rappresentazione più realistica delle dinamiche del veicolo, fondamentale per applicazioni su veicoli reali o simulazioni di livello elevato.

Per affrontare il problema del path tracking sono state sviluppate numerose tecniche di controllo [10]:

- approcci classici come il controllore PID e il controllore LQR
- approcci predittivi come il Model Predictive Control (MPC)
- tecniche avanzate come i controllori adattativi o i metodi basati su intelligenza artificiale

La selezione della strategia di controllo più adatta dipende da molteplici fattori quali ad esempio il livello di precisione richiesto, le capacità computazionali a disposizione, la complessità del modello o la necessità di rispettare vincoli stringenti su accelerazioni, velocità e comfort. Il confronto tra approcci classici e predittivi rappresenta oggi un tema centrale in letteratura, con l'obiettivo di identificare i contesti applicativi ideali per ciascuna metodologia.

1.3 Tecniche di controllo nel path tracking

Come anticipato nel paragrafo precedente il problema del path tracking richiede la progettazione di controllori in grado di garantire precisione, stabilità e capacità di adattamento alle condizioni variabili del veicolo e dell'ambiente circostante. Negli ultimi anni sono stati sviluppati numerosi approcci di controllo, ciascuno con specifici punti di forza e limitazioni. Le principali categorie di controllori utilizzati nel path tracking possono essere suddivise in tre gruppi:

1.3.1 Controllori classici

I controllori classici comprendono approcci lineari come il PID (Proporzionale-Integrale-Derivativo) e il LQR (Linear Quadratic Regulator).

- Il PID rappresenta una delle tecniche più utilizzate in ambito industriale, grazie alla semplicità di implementazione e alla bassa richiesta computazionale. Tuttavia, la sua capacità di gestire scenari complessi o non lineari è limitata [16].
- L'LQR offre un controllo ottimale per sistemi lineari, minimizzando una funzione di costo quadratica, ma richiede una precisa conoscenza del modello e non gestisce vincoli in modo esplicito.

Questi controllori sono ancora oggi largamente impiegati in applicazioni dove la semplicità e la velocità di calcolo sono fondamentali, come nel caso di veicoli autonomi in scala o robot mobili operanti in ambienti strutturati.

1.3.2 Controllori predittivi

Il Model Predictive Control (MPC) è una delle tecniche più avanzate e flessibili, in quanto consente di risolvere un problema di ottimizzazione a ogni passo di controllo, prevedendo il comportamento futuro del sistema e gestendo in modo esplicito i vincoli sugli stati e sugli input.

- L'LMPC (Linear MPC) utilizza un modello lineare del sistema e può essere applicato con successo in scenari dove le condizioni operative rimangono vicine al punto di linearizzazione.
- L'NMPC (Nonlinear MPC) estende questo approccio all'uso di modelli non lineari, consentendo una maggiore accuratezza e una migliore gestione delle dinamiche complesse, a scapito di un carico computazionale più elevato come discusso in [17].

I controllori predittivi trovano applicazione in scenari in cui è necessario ottimizzare le prestazioni in presenza di vincoli stringenti e prevedere l'evoluzione del sistema in condizioni dinamiche variabili.

1.3.3 Tecniche avanzate e ibride

Recentemente sono state sviluppate tecniche di controllo avanzato che combinano diversi approcci o utilizzano metodi innovativi:

- Controllori adattativi: in grado di modificare i propri parametri in tempo reale in funzione delle variazioni delle condizioni operative.
- Controllori fuzzy e metodi basati su logiche inferenziali: particolarmente utili in presenza di incertezze e modelli incompleti.
- Reti neurali e apprendimento automatico: utilizzati per realizzare controllori che apprendono direttamente dal comportamento ottimale di un controllore esperto (approccio imitation learning) o che approssimano funzioni di costo e dinamiche complesse non note. Un'analisi dettagliata di queste tecniche è disponibile in [18].

Sebbene queste tecniche siano promettenti, la loro implementazione su veicoli reali è spesso limitata dalla necessità di garantire robustezza, trasparenza del funzionamento (explainability) e capacità di operare in tempo reale, tutti requisiti fondamentali in ambito automotive.

1.4 Confronto in letteratura tra PID, Pole Placement, LQR, LMPC e NMPC

In questo paragrafo verranno presentate diverse strategie di controllo (PID, Pole Placement, LQR, LMPC e NMPC) che possono essere adottate per il path tracking, tuttavia, nella presente tesi saranno approfonditi esclusivamente i controllori PID, LMPC e NMPC.

1.4.1 Controllore PID

Il controllore Proporzionale-Integrale-Derivativo (PID), è uno dei più utilizzati nei sistemi di controllo industriale grazie alla sua semplicità implementativa ed efficacia in una vasta gamma di applicazioni. Il PID rappresenta una soluzione molto robusta per sistemi stazionari e lineari, ma presenta limiti significativi in scenari dinamici e non lineari, come quelli tipici della guida autonoma [19]. L'uscita del controllore $u(t)$ è data dalla combinazione lineare di tre termini:

1. Azione Proporzionale (P): proporzionale all'errore attuale $e(t)$:

$$u_P(t) = K_P \cdot e(t)$$

dove K_P è il guadagno proporzionale.

2. Azione Integrale (I): proporzionale all'integrale dell'errore nel tempo, utile per eliminare l'errore stazionario:

$$u_I(t) = K_I \cdot \int_0^t e(\tau) d\tau$$

dove K_I è il guadagno integrale.

3. Azione Derivativa (D): proporzionale alla derivata dell'errore, fornisce una previsione dell'andamento futuro dell'errore:

$$u_D(t) = K_D \cdot \frac{de(t)}{dt}$$

dove K_D è il guadagno derivativo.

Combinando queste tre azioni, l'uscita totale del controllore PID è:

$$u(t) = K_P \cdot e(t) + K_I \cdot \int_0^t e(\tau) d\tau + K_D \cdot \frac{de(t)}{dt}$$

In ambito di path tracking, il PID è spesso utilizzato per il controllo dell'angolo di sterzo basandosi sull'errore laterale e angolare. Tuttavia, la sua efficacia può essere limitata in

presenza di dinamiche non lineari e inoltre non dispone di una capacità predittiva e né di una gestione esplicita dei vincoli, elementi oggi ritenuti fondamentali nelle applicazioni avanzate.

1.4.2 Controllore Pole Placement

Il pole placement è una tecnica di retroazione di stato che consente di posizionare gli autovalori (poli) del sistema in retroazione dove si desidera, modellando le proprietà dinamiche del sistema chiuso. Partendo da un modello lineare continuo

$$\dot{x} = Ax + Bu$$

Si assume una legge di controllo lineare $u = -Kx$ tale che la matrice del sistema in retroazione chiusa (closed-loop)

$$A_{cl} = A - BK$$

Abbiamo un insieme di poli $\{p_1, p_2, \dots, p_n\}$ prestabiliti nel semipiano sinistro. La condizione fondamentale per poter spostare liberamente i poli è che la coppia A, B sia controllabile, il che si verifica quando la matrice di controllabilità $[B \ AB \ A^2B \ \dots]$ ha rango pieno.

Il calcolo matriciale di K , può effettuarsi in due modi principali:

1. Metodo algebrico: si impone che il polinomio caratteristico di A_{cl} coincida con il polinomio desiderato $(s - p_1)(s - p_2) \dots (s - p_n)$ e si risolve il sistema di equazioni risultante
2. Trasformazione in forma canonica: mediante una trasformazione di coordinate si porta il modello in una forma controllabile semplificata, si calcolano i guadagni nello spazio trasformato e si risale allo spazio originale

Tra i vantaggi principali di questa strategia di controllo vi sono la chiarezza e la rapidità con cui il requisito di prestazione si traduce in un posizionamento diretto dei poli e il basso costo computazionale. Dal momento che non esiste una funzione obiettivo automatica, la scelta dei poli spesso richiede esperienza e calibrazioni successive. Inoltre, in presenza di disturbi o incertezze parametriche, il sistema può risultare meno robusto se confrontato con altre soluzioni che ad esempio incorporano esplicitamente vincoli e/o ottimizzazioni.

Nel contesto del path tracking, il pole placement può essere applicato modellando la dinamica degli errori di posizione e scegliendo i poli in modo da ottenere tempi di assestamento rapidi e uno smorzamento adeguato, riducendo così l'errore sulla traiettoria. Questa tecnica pertanto, data la sua chiarezza, semplicità e basso costo computazionale risulta particolarmente utile in fase di progettazione preliminare o nel caso in cui si disponga di requisiti puntuali sul comportamento dei poli.

1.4.3 Controllore LQR

Il controllore lineare-quadratico LQR (Linear Quadratic Regulator) è una tecnica di retroazione di stato ottimale per sistemi lineari, partendo dal modello

$$\dot{x} = Ax + Bu$$

Si assume una legge di controllo lineare $u = -Kx$ e si definisce un costo complessivo J

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt$$

Dove Q semidefinita positiva e R definita positiva, per ponderare rispettivamente i pesi relativi agli scostamenti di stato e all'intensità del segnale di controllo.

Nel contesto del path tracking con LQR si definisce un vettore di stato con gli errori che definiscono lo scostamento laterale e angolare del veicolo rispetto alla traiettoria desiderata. A partire da un modello lineare, si sceglie una cost function quadratica che penalizza simultaneamente gli errori e l'azione di controllo. Risolvendo l'equazione si ricava offline la matrice di guadagno K , che in tempo reale viene moltiplicata per lo stato, generando le azioni di controllo.

Tra i principali vantaggi del controllore LQR vi è la capacità di fornire, attraverso la scelta delle matrici di peso Q e R , un bilanciamento ottimale tra prestazioni dinamiche e costo dell'azione di controllo, il metodo risolve automaticamente il compromesso tra rapidità di risposta e intensità dell'azione di comando, evitando lunghe fasi di taratura empirica. Inoltre, a condizione che il modello lineare soddisfi i requisiti di stabilità, l'LQR garantisce stabilità asintotica del sistema in retroazione e possiede una robustezza interna che tollera piccole incertezze. Dal punto di vista dell'implementazione, la risoluzione dell'equazione di Riccati avviene offline rendendo il calcolo del guadagno K rapido, ripetibile e con un basso costo computazionale. Nonostante questi punti di forza, l'LQR presenta alcuni limiti, innanzitutto la sua efficacia dipende fortemente dalla validità del modello lineare, in presenza di non linearità marcate o di variazioni sostanziali delle condizioni operative, le prestazioni possono discostarsi da quelle previste. Inoltre la determinazione delle matrici Q e R richiede un processo di tuning iterativo che può risultare oneroso. Infine l'LQR non include meccanismi per la gestione dei vincoli, il che impone di ricorrere a metodologie di controllo più sofisticate qualora questi vincoli siano rilevanti.

1.4.4 Controllore LMPC

Il controllo predittivo basato su modello lineare (Linear Model Predictive Control, LMPC) utilizza un modello lineare del sistema per predire il comportamento futuro e risolvere il problema di ottimizzazione su un orizzonte di predizione finito. Questo approccio minimizza una cost function che penalizza l'allontanamento dello stato del sistema da un comportamento desiderato e anche l'uso degli input di controllo, gestendo in modo esplicito vincoli sugli stati e comandi [20]. La formulazione generale del problema di ottimizzazione è:

$$\min_u \sum_{k=0}^{N-1} (x_{k+1}^T Q x_{k+1} + u_k^T R u_k)$$

soggetto a:

$$x_{k+1} = Ax_k + Bu_k$$

$$x_{min} \leq x_k \leq x_{max}$$

$$u_{min} \leq u_k \leq u_{max}$$

dove:

- x_k è il vettore degli stati al passo k ,
- u_k è il vettore degli input di controllo al passo k ,
- **A** e **B** sono le matrici del sistema lineare,
- **Q** e **R** sono le matrici di peso per gli stati e gli input nel costo,
- N è l'orizzonte di predizione,
- $x_{min}, x_{max}, u_{min}, u_{max}$ rappresentano i vincoli sugli stati e sugli input.

L'LMPC è efficace nel gestire sistemi con dinamiche lineari o linearizzabili e può incorporare vincoli in modo esplicito. Tuttavia, la sua performance può degradare se il sistema presenta forti non linearità o se opera lontano dal punto di linearizzazione.

1.4.5 Controllore NMPC

Il controllo predittivo basato su modello non lineare (Nonlinear Model Predictive Control, NMPC) estende l'approccio MPC utilizzando un modello non lineare per predire il comportamento futuro. Si tratta di una tecnica sviluppata per affrontare scenari in cui le dinamiche del sistema non sono linearizzabili o variano significativamente nel tempo, questo approccio è ampiamente trattato nel volume [21], dove viene evidenziato il ruolo dell'NMPC nella gestione di sistemi altamente dinamici. La formulazione del problema di ottimizzazione è simile a quella dell'LMPC, ma con un modello non lineare:

$$\min_u \sum_{k=0}^{N-1} (x_{k+1}^T Q x_{k+1} + u_k^T R u_k)$$

soggetto a:

$$x_{k+1} = f(x_k, u_k)$$

$$x_{min} \leq x_k \leq x_{max}$$

$$u_{min} \leq u_k \leq u_{max}$$

dove:

- x_k è il vettore degli stati al passo k ,
- u_k è il vettore degli input di controllo al passo k ,
- $f(x_k, u_k)$ funzione non lineare che descrive la dinamica del sistema,
- Q è la matrice di peso degli stati nel costo, viene utilizzata per penalizzare deviazioni dai valori desiderati degli stati, come ad esempio l'errore laterale e l'errore angolare,
- R è la matrice di peso sugli input di controllo. Serve a penalizzare variazioni troppo brusche dei comandi e a contenere lo sforzo di attuazione
- N è l'orizzonte di predizione, cioè il numero di passi futuri su cui viene calcolata l'ottimizzazione,
- $x_{min}, x_{max}, u_{min}, u_{max}$ rappresentano i vincoli sugli stati e sugli input di controllo.

L'NMPC offre una maggiore accuratezza nel controllo di sistemi con dinamiche non lineari complesse e può gestire vincoli in modo diretto. Tuttavia, il principale limite dell'NMPC risiede nell'elevata complessità computazionale e nel fatto che è richiesta una precisa conoscenza del modello del sistema, che ne rende difficile l'impiego in tempo reale su hardware a risorse limitate. In particolare, l'implementazione su microcontrollori o hardware embedded impone spesso una riformulazione del problema di ottimizzazione come ad esempio accorciando l'orizzonte predittivo, utilizzando modelli semplificati oppure limitando il numero di iterazioni dell'algoritmo risolutivo [22].

L'analisi della letteratura ha mostrato come il problema del path tracking sia stato affrontato con numerosi approcci di controllo, ciascuno con caratteristiche, vantaggi e limitazioni ben definite:

- i controllori PID, grazie alla loro semplicità di implementazione e basso costo computazionale, rappresentano una soluzione valida per applicazioni in cui il sistema opera in condizioni stabili e le dinamiche sono poco complesse. Tuttavia, la mancanza di capacità predittiva e la difficoltà di gestione dei vincoli ne limitano l'applicabilità in scenari più dinamici o ad alta velocità.
- Il pole placement, sfrutta la retroazione di stato per collocare i poli del sistema chiuso in posizioni prestabilite e garantisce un basso onere computazionale e un'interpretazione intuitiva dei requisiti dinamici, pur non garantendo una robustezza ottimale contro errori e variazioni del modello
- I controllori LQR, basati sulla minimizzazione di un costo quadratico su stati e azioni di comando tramite la risoluzione dell'equazione di Riccati, offrono un compromesso ottimale tra velocità di risposta e intensità del comando. L'implementazione numerica è efficiente ma richiede un modello lineare accurato e un tuning esperto delle matrici di peso.
- i controllori LMPC, basati su modelli lineari predittivi, permettono di ottenere prestazioni più elevate e gestire vincoli in modo esplicito. La loro efficacia è però strettamente legata alla qualità del modello lineare e alla vicinanza al punto di linearizzazione. In condizioni fortemente non lineari o in presenza di variazioni rapide delle dinamiche, le prestazioni possono risultare inferiori alle aspettative.
- i controllori NMPC rappresentano la soluzione più avanzata, in grado di utilizzare modelli non lineari complessi e ottimizzare il comportamento futuro del sistema rispettando vincoli stringenti. Il principale ostacolo è costituito dalle elevate richieste computazionali e dalla necessità di un tuning molto accurato, che ne rendono difficile l'implementazione in tempo reale su hardware limitato. Negli ultimi anni, ai controllori NMPC sono state affiancate avanzate tecniche di machine learning, consentendo sia di ottimizzare le prestazioni dei sistemi di controllo in tempo reale e sia di adattarsi dinamicamente a scenari complessi e a variazioni improvvise delle condizioni operative [23].

La presente tesi ha come obiettivo quello di effettuare un confronto tra i controllori PID, LMPC e NMPC, mediante simulazioni numeriche condotte a differenti velocità su un percorso normato e prestabilito. Il confronto verrà condotto sulla base di specifici indicatori di prestazione (KPI) precedentemente descritti, tra cui l'errore laterale massimo, l'errore angolare massimo e i corrispondenti valori RMSE. L'analisi si propone di valutare l'efficacia, la robustezza e la fattibilità computazionale di ciascun approccio, con l'obiettivo di selezionare il controllore più adatto in relazione alle specifiche esigenze.

2 Metodologie

Questo capitolo descrive l'architettura di controllo adottata per l'analisi comparativa di tre diversi controllori di path tracking all'interno di un ambiente di simulazione sviluppato con MATLAB e Simulink. Lo studio si basa esclusivamente su simulazioni, senza implementazioni su veicoli reali, e si concentra sulla valutazione delle prestazioni dei controllori che simulano il percorso del veicolo lungo il tracciato di riferimento.

Viene inoltre descritto il modello di validazione, sviluppato su Simulink, che rappresenta il comportamento del veicolo durante l'esecuzione della manovra. Tale modello è stato utilizzato come riferimento per testare le diverse strategie di controllo e valutare le risposte dinamiche in condizioni standardizzate e normate per garantire ripetibilità e confrontabilità dei risultati. La simulazione si svolge su un percorso definito in conformità alla norma ISO 3888-2-2011, che specifica i vincoli geometrici e il sistema di riferimento utilizzati.

Il progetto dei controllori si basa su un modello dinamico laterale del veicolo, linearizzato o non linearizzato a seconda del tipo di controllore. L'implementazione e l'automatizzazione della generazione delle funzioni di costo e dei vincoli sono state gestite tramite il software ACADO Toolkit, integrato nell'ambiente MATLAB/Simulink.

I principali indicatori chiave di prestazione (KPI) utilizzati per confrontare le strategie di controllo:

- errore laterale massimo
- errore angolare massimo
- RMSE dell'errore laterale
- RMSE dell'errore angolare

Questi indicatori consentono di valutare in modo oggettivo l'efficacia dei controllori in termini di precisione e intensità degli interventi richiesti agli attuatori.

2.1 Ambiente di simulazione: MATLAB e Simulink

L'intero lavoro di simulazione è stato sviluppato utilizzando la piattaforma MATLAB® con il supporto di Simulink®, ambienti di calcolo tecnico e modellazione sviluppati da MathWorks. Questi strumenti sono largamente impiegati in diversi ambiti tra cui quello accademico e industriale per l'analisi, la simulazione e il controllo di sistemi dinamici, grazie alla loro flessibilità, all'interfaccia user-friendly e alla vasta integrazione di tool esterni avanzati [24][25].

Nel contesto di questa tesi, MATLAB è stato utilizzato per:

- definizione e inizializzazione dei parametri del veicolo e della traiettoria;
- gestione degli script della simulazione e l'elaborazione automatica dei risultati;
- calcolo degli indicatori di prestazione (KPI) e elaborazione dei risultati;
- generazione dei grafici per il confronto delle prestazioni dei controllori

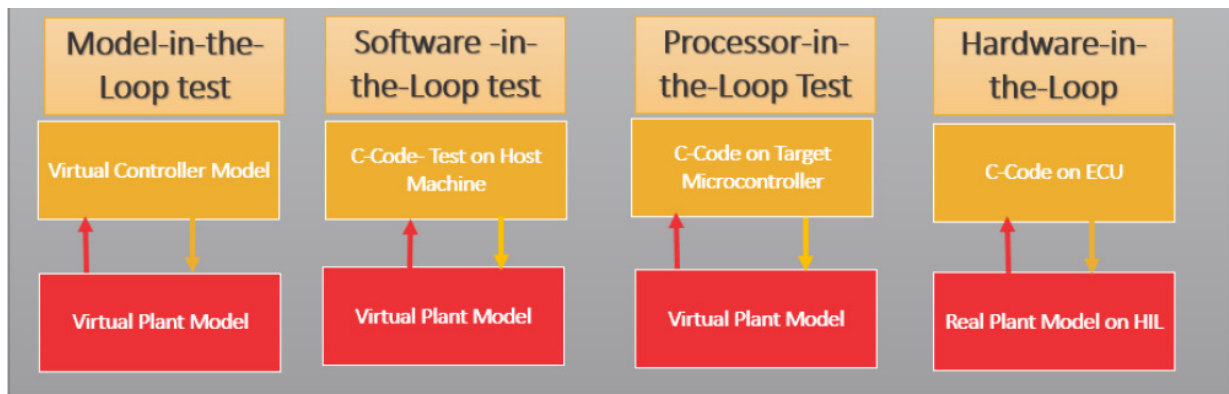


Figura 2: sviluppo di un sistema di controllo

Nel processo di sviluppo di un sistema di controllo, si susseguono quattro fasi fondamentali:

- **Model in the Loop (MiL)**
In questa prima fase, sia il controllore sia l'impianto sono rappresentati da modelli matematici eseguiti in ambienti virtuali. L'obiettivo è verificare la correttezza della logica di controllo, prima ancora della generazione del codice e trascurando eventuali limitazioni dell'hardware. In questa fase vengono esplorati algoritmi e strategie di regolazione in modo rapido e flessibile, sperimentando possibili varianti del controllore direttamente sul modello.
- **Software in the Loop (SiL)**
Una volta superato la fase di modellistica pura, si passa a eseguire il codice generato dal controllore su un normale computer di sviluppo (host). L'impianto rimane ancora un modello virtuale ma l'esecuzione reale del software permette di individuare problemi di compilazione, di librerie oppure di integrazione tra codice e modello.

- **Processor in the Loop (PiL)**
In questa fase il codice viene caricato e fatto girare su un microcontrollore simile a quello che verrà impiegato nell'ECU definitivo. Pur mantenendo il modello dell'impianto in ambiente virtuale, l'esecuzione su hardware reale permette di valutare con maggiore precisione le prestazioni, i tempi di elaborazione e i vincoli di calcolo e memoria.
- **Hardware in the Loop (HiL)**
Nella fase di validazione, il controllore gira sull'ECU definitiva o una equivalente, mentre il virtual plant è sostituito da un modello fisico ad alta fedeltà installato su un simulatore. Questa configurazione consente di testare in sicurezza scenari complessi prima di passare all'installazione su un veicolo o su un sistema reale.

Nella presente tesi verrà affrontata la prima delle quattro fasi appena descritte, il Model in the Loop test, concentrando l'attenzione sulla progettazione e la validazione del controllore in ambiente puramente modellistico.

Simulink è stato utilizzato per realizzare il modello dinamico del veicolo e costruire un ambiente di simulazione completo. Il modello è strutturato in sottosistemi che rappresentano le diverse componenti del sistema: generazione del percorso, dinamica del veicolo, controllore adottato, blocco di calcolo degli errori e modulo di valutazione dei KPI. Questa struttura modulare permette di testare diverse strategie di controllo senza modificare l'architettura globale, facilitando il cambio di controllore o l'introduzione di nuove funzionalità.

Un ulteriore vantaggio dell'integrazione MATLAB/Simulink risiede nella possibilità di combinare la potenza del calcolo numerico con la rappresentazione visuale dei modelli. Questa sinergia è particolarmente utile nel modello di validazione, poiché consente di monitorare con facilità l'evoluzione temporale degli stati del veicolo insieme ad altre variabili. Inoltre, l'utilizzo degli script MATLAB per il controllo delle simulazioni permette di automatizzare l'esecuzione di scenari multipli, migliorando l'efficienza nella fase di confronto tra i diversi controllori.

L'ambiente così costruito è facilmente estendibile, per esempio con l'introduzione di ulteriori complessità nei modelli, o con l'implementazione di strategie di controllo alternative. Inoltre, la compatibilità con toolbox avanzati esterni, come ACADOS e Vehicle Dynamics Blockset, consente di affrontare anche problemi complessi di ottimizzazione in tempo reale e simulazione avanzata [26].

2.2 ACADOS tool-kit

Per la risoluzione dei problemi di controllo in tempo reale, è stato utilizzato il software ACADOS (ACADOS Toolkit Advanced Control and Optimization Software) [27], una libreria open-source per la soluzione di problemi di controllo predittivo non lineare (NMPC), controllo lineare (LMPC) e problemi di programmazione quadratica sequenziale (SQP). L'ambiente è pensato per essere utilizzato in applicazioni embedded, real-time e automotive grazie alla rapidità di calcolo e alla possibilità di personalizzare ogni elemento del processo di ottimizzazione.

Offre un framework versatile, che include diversi algoritmi per il controllo diretto, comprendendo il controllo predittivo basato su modello (MPC), metodi di integrazione Runge-Kutta e BDF per la simulazione di equazioni differenziali ordinarie (ODE) ed equazioni differenziali algebriche (DAE). L'approccio consente di ottenere soluzioni con un buon compromesso tra accuratezza e tempo computazionale, rendendolo adatto all'utilizzo in controllori real-time come quelli adottati in questa tesi.

Nel caso in esame ACADOS è stato utilizzato per implementare i controllori LMPC e NMPC, risolvendo in tempo reale il problema di minimizzazione di una cost function soggetta alle dinamiche del veicolo e ai vincoli di ingresso e di stato. La libreria permette la definizione del modello dinamico, della funzione obiettivo e dei vincoli in ambiente MATLAB, con successiva generazione del codice utilizzato in linguaggio C. Per facilitare l'integrazione in ambiente MATLAB/Simulink, ACADOS for MATLAB funge da interfaccia. Le principali caratteristiche di ACADOS for MATLAB includono:

- Stesse proprietà dell'ACADOS Toolkit: l'interfaccia eredita tutte le proprietà e le funzionalità dell'ACADOS Toolkit, senza introdurre nuovi algoritmi o funzionalità aggiuntive
- Non è richiesta nessuna conoscenza del linguaggio C++, ma solo familiarità con l'ambiente MATLAB: l'utente quindi non deve necessariamente possedere alcuna conoscenza del C++, né della sua sintassi o delle regole di compilazione per utilizzare l'interfaccia. L'interfaccia adotta notazioni nello stile MATLAB e consente l'utilizzo diretto di variabili e matrici memorizzate nell'area di lavoro di MATLAB.
- integrazione con modelli black-box MATLAB: mentre l'ACADOS Toolkit supporta una sintassi simbolica per l'espressione di equazioni differenziali e algebriche, l'interfaccia consente la connessione diretta di modelli black-box esistenti in MATLAB all'ACADOS Toolkit
- compatibilità multiplatforma: l'interfaccia è compatibile con diversi sistemi operativi, inclusi Windows, Mac e Linux.

2.3 Modello di validazione

Il modello di validazione rappresenta l'ambiente simulato complesso, sviluppato con l'obiettivo di descrivere nel modo più fedele possibile il comportamento dinamico del veicolo in risposta agli input generati dal controllore. Questo modello è stato sviluppato in Simulink ed è integrato con lo script MATLAB, che gestisce l'inizializzazione dei parametri, l'avvio delle simulazioni e la raccolta dei risultati.

Il blocco ValidationVehicleModel riceve in ingresso due vettori strutturati:

- ReferenceINFO che include le informazioni riguardo la traiettoria di riferimento, come le coordinate di riferimento (x_{ref} , y_{ref}) e l'angolo di imbardata (ψ_{ref})
- ControlAction, che contiene le azioni di controllo calcolate dal controllore, come angoli di sterzo (δ) e le coppie motrici (torque)

Il sistema all'interno del blocco di validazione implementa la dinamica del veicolo:

- ReferencePath, funzione che calcola, istante per istante, i valori target della traiettoria sulla base delle reference info
- vehicleDynamicalStateDerivative che calcola le derivate dello stato del veicolo in base al modello cinematico-dinamico tenendo conto dell'interazione tra input di controllo e condizione attuali
- ZOH che integra le derivate per ottenere gli stati aggiornati e sincronizza il passo della simulazione
- VehicleAlgebraicState che calcola variabili come slip angle, forze agenti sulle ruote, coppie e altri parametri
- ErrorState che valuta gli errori tra stato simulato e di riferimento

L'output del blocco di validazione è costituito dal blocco ErrorState, che raccoglie gli errori cinematici e il vettore VehicleINFO, che riunisce tutte le variabili dinamiche calcolate:

- coordinate spaziale (x , y)
- coordinate angolari (yaw angle, roll angle, yaw rate)
- variabili legate alle ruote (δ , ω , ecc)
- forze agenti (F_x , F_y)
- parametri cinematici (α , β , σ)

Tutti i dati vengono poi registrati e inviati ai blocchi di post-processing per consentire un'analisi dettagliata delle prestazioni del controllore adottato.

2.3.1 Dinamica del veicolo

Il nucleo del modello è costituito da un sistema dinamico che si pone l'obiettivo di descrivere nel dettaglio la risposta del veicolo agli input di controllo, esso include:

- la posizione e l'orientamento nello spazio;
- la velocità longitudinale, laterale e angolare;
- lo stato delle singole ruote (velocità angolare, angoli di sterzo, forze agenti);
- parametri cinematici aggiuntivi come slip angle, camber e accelerazioni.

La dinamica è modellata in modo completo, considerando l'interazione tra le azioni di controllo (angolo di sterzo e coppie motrici) e la risposta del veicolo. Si sottolinea che solo le ruote anteriori sono sterzanti, mentre tutte e quattro contribuiscono alla generazione delle forze di trazione e frenata. Il modello tiene conto inoltre della distribuzione delle masse, delle inerzie, delle rigidità laterali dei pneumatici e della geometria del veicolo, rendendolo adatto a simulazioni realistiche di manovre critiche.

Nel contesto della dinamica del veicolo presentata in questo paragrafo, un elemento chiave è rappresentato dalla modellazione realistica del comportamento dei pneumatici. A tal fine, il modello di validazione implementa la formulazione empirica di Pacejka, nota anche come "Magic Formula", utilizzata per calcolare in modo accurato le forze laterali e longitudinali generate dalle ruote. Questo approccio consente di descrivere la relazione non lineare tra angolo di deriva, slip ratio e le forze risultanti, tenendo conto del comportamento reale degli pneumatici, soprattutto in condizioni di guida dinamiche o al limite di aderenza. La formulazione adottata permette inoltre di modellare l'influenza del carico verticale sulla generazione delle forze, aspetto essenziale per simulare correttamente trasferimenti di carico e variazioni transitorie durante le manovre.

2.3.2 Generazione della traiettoria di riferimento

Il sistema di generazione della traiettoria è progettato per fornire, istante per istante, una sequenza di valori target coerenti con la dinamica del veicolo. Le principali funzioni svolte dal modulo includono:

- interpolazione spaziale a partire da un set discreto di punti (ricavati da uno script, da una norma o da tracciati personalizzati);
- aggiornamento temporale, il sistema è in grado di fornire in ogni istante il punto di riferimento più vicino al veicolo, aggiornando i valori target (posizione e orientamento) sulla base della posizione attuale e di quella successiva lungo il tracciato di prova;
- output normalizzato, che restituisce variabili fondamentali per la fase di controllo (coordinate di riferimento, altre grandezze spaziali di riferimento);

Il modulo è implementato come sottosistema indipendente nel modello di validazione e viene richiamato ad ogni passo di simulazione. Nel caso di questa di tesi la traiettoria è definita secondo la norma ISO 3888-2-2011, che verrà approfondita in seguito.

2.3.3 Calcolo degli errori cinematici

Per valutare l'efficacia del tracking nella fase di post-processing, viene utilizzato il modulo ErrorState. Si sottolinea come questo blocco sia separato dal modulo di simulazione della dinamica del veicolo. Esso calcola:

- errore laterale, distanza trasversale tra la posizione attuale del veicolo e la traiettoria di riferimento
- errore di orientamento, differenza tra angolo di imbardata attuale del veicolo e l'angolo di imbardata desiderato
- ulteriori parametri utili per la definizione degli indicatori di performance

Questi errori sono essenziali per valutare le prestazioni del controllore, ottimizzarne i parametri e garantire la sicurezza assicurandosi che rimangano entro i limiti accettabili.

2.4 Modello interno LMPC

Per la progettazione dei controllori è stato utilizzato un modello interno che descrive la dinamica laterale del tipo single-track model. Questa scelta consente di rappresentare in maniera compatta le forze laterali, le variazioni di angolo di imbardata e i principali effetti dinamici mantenendo contenuta la complessità computazionale [28].

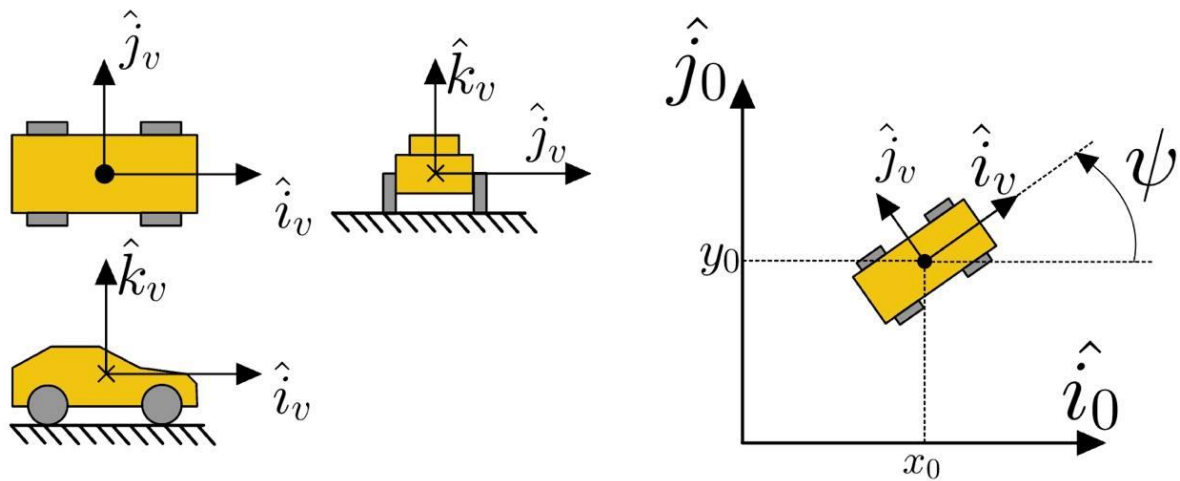


Figura 3: sistema di riferimento del veicolo

Il veicolo è descritto in un sistema di riferimento solidale al veicolo stesso, con origine nel centro di gravità (CoG), asse longitudinale X diretto verso la parte anteriore e asse trasversale Y rivolto verso sinistra. L'errore laterale rispetto alla traiettoria di riferimento è indicato come e_y , mentre l'errore di orientamento (yaw error) è definito come la differenza tra l'angolo di imbardata del veicolo ψ e la direzione della traiettoria ψ_{ref} .

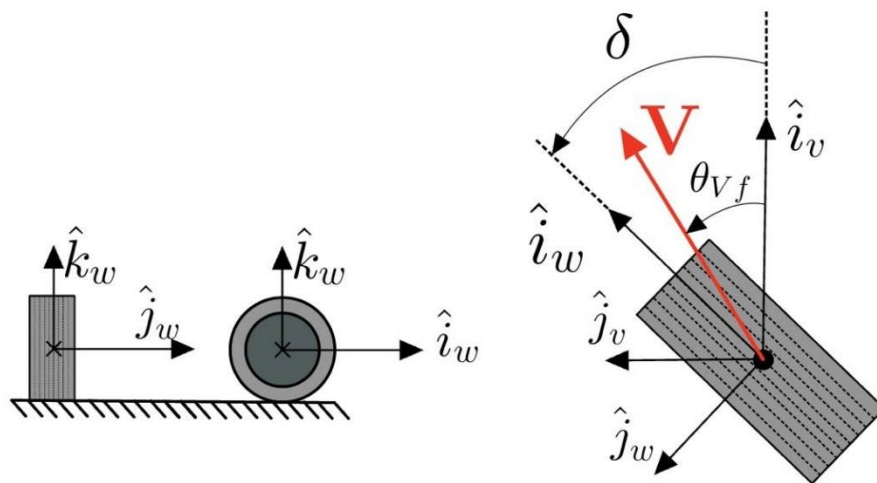


Figura 4: sistema di riferimento della ruota

Il modello interno utilizza come stato un vettore 4 x 1 composto dalle seguenti variabili:

$$x = \begin{bmatrix} y \\ \dot{y} \\ \psi \\ \dot{\psi} \end{bmatrix}$$

dove:

- y è la coordinata trasversale della posizione del veicolo;
- $\dot{y}$ è la velocità laterale del centro di gravità;
- ψ è l'angolo di imbardata rispetto al sistema globale;
- $\dot{\psi}$ è la velocità angolare d'imbardata.

L'ingresso di controllo è rappresentato dall'angolo di sterzo anteriore:

$$u = [\delta]$$

Il sistema dinamico lineare può essere rappresentato nella forma:

$$\dot{x} = Ax + Bu$$

Questa formulazione rappresenta il punto di partenza per il successivo cambio di variabile, che consente di esprimere il modello in coordinate d'errore rispetto alla traiettoria di riferimento, come illustrato nel paragrafo successivo.

Per esprimere il problema di controllo in termini funzionali al tracking della traiettoria, il modello dinamico descritto precedentemente viene convertito da coordinate assolute a coordinate di errore rispetto al reference path. Si parte dalla descrizione della posizione del veicolo nel sistema globale e si proietta il veicolo sulla traiettoria, questo cambio consente di lavorare su errori geometrici e cinematici che il controllore dovrà poi minimizzare, migliorando la struttura del problema di ottimizzazione. Le variabili di controllo sono definite come:

$$e_1 = (p - p_{ref}) \cdot \hat{j}_v$$

$$e_2 = \psi - \psi_{ref}$$

Dove p è la posizione del veicolo, p_{ref} è il punto di proiezione della traiettoria e j è il versore normale alla traiettoria, ψ è l'angolo di imbardata e ψ_{ref} è l'angolo di imbardata di riferimento, cioè l'angolo tangente alla traiettoria nel punto di riferimento.

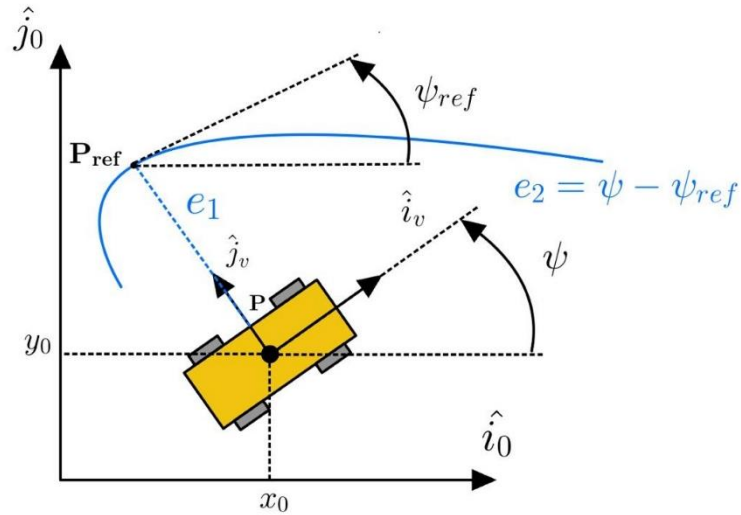


Figura 5: cambio di coordinate

Il nuovo vettore degli stati è definito come:

$$e = \begin{bmatrix} e_1 \\ \dot{e}_1 \\ e_2 \\ \dot{e}_2 \end{bmatrix}$$

L'ingresso di controllo è espresso come:

$$u_e = \begin{bmatrix} \delta \\ \psi_{ref} \end{bmatrix}$$

Il sistema dinamico lineare può essere rappresentato nella forma:

$$\dot{e} = Ae + B$$

dove le matrici A e B sono definite come:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{2(C_{af} + C_{ar})}{mv_x} & \frac{2(C_{af} + C_{ar})}{m} & -\frac{2(C_{af}l_f - C_{ar}l_r)}{mv_x} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{2(C_{af}l_f - C_{ar}l_r)}{I_z v_x} & \frac{2(C_{af}l_f - C_{ar}l_r)}{I_z} & -\frac{2(C_{af}l_f^2 + C_{ar}l_r^2)}{I_z v_x} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ \frac{2C_{af}}{m} & -\frac{2(C_{af}l_f - C_{ar}l_r)}{mv_x} \\ 0 & 0 \\ \frac{2C_{af}l_f}{I_z} & -\frac{2(C_{af}l_f^2 + C_{ar}l_r^2)}{I_z v_x} \end{bmatrix} v_x$$

Dove:

- C_{af} e C_{ar} rappresentano le rigidezze laterali dei pneumatici anteriori e posteriori [N/rad]
- m è la massa del veicolo [kg]
- v_x denota la velocità longitudinale [m/s]
- I_z è il momento d'inerzia di imbardata [kg m²]
- l_f e l_r rappresentano la distanza del baricentro dagli assi anteriore e posteriore [m]

2.5 Modello interno NMPC

Nel caso di controllore NMPC si è adottata un'assunzione significativa: il modello interno utilizzato dall'NMPC coincide con il modello di validazione. Questa scelta nel contesto di questo studio simulativo permette di sfruttare appieno la ricchezza descrittiva del modello.

Il modello di validazione, sviluppato in ambiente Simulink, è altamente dettagliato ed è in grado di rappresentare con elevata fedeltà il comportamento dinamico del veicolo. Include variabili relative allo stato delle singole ruote, agli angoli di sterzo, alle forze agenti, ai parametri cinematici locali e alle accelerazioni, il tutto consente di descrivere scenari operativi realistici come manovre complesse, transitori dinamici rapidi e variazioni significative delle condizioni di guida.

L'adozione dello stesso modello sia per la simulazione che per la predizione offre l'importante vantaggio di valutare le performance del controllore NMPC nelle condizioni ideali, ovvero in assenza di disallineamenti tra il modello predittivo e quello reale. Va tuttavia sottolineato che questa impostazione è stata adottata esclusivamente per il controllore NMPC. I controllori LMPC e PID utilizzano modelli interni semplificati, scelti per contenere la complessità computazionale.

2.6 Struttura del codice di simulazione

In questa sezione vengono descritti due dei diversi script MATLAB su cui si basa l'ambiente simulativo sviluppato. L'obiettivo è fornire una panoramica del file principale di simulazione, responsabile della gestione del processo, e dello script dedicato alla generazione della traiettoria di riferimento, fondamentale per la definizione del percorso da seguire durante le varie prove.

2.6.1 Script principale di simulazione

Lo script `main.m` rappresenta il nucleo centrale del processo di simulazione. La sua funzione principale è quella di inizializzare l'ambiente MATLAB, configurare i parametri della simulazione e caricare i modelli necessari per ciascuna combinazione di velocità e controllore. Dopo una fase iniziale setup, lo script consente all'utente di selezionare il tipo di traiettoria da utilizzare tra quelle a disposizione (nel caso in esame cambio di corsia). A seconda della scelta, viene richiamato uno specifico script di generazione del percorso e in seguito, vengono poi impostate le condizioni di test, come la velocità iniziale e la directory del controllore da testare.

Per ciascuna configurazione, lo script esegue il caricamento dei dati del veicolo e delle interfacce di controllo, avvia la simulazione del modello di validazione in Simulink e raccoglie i risultati. L'output della simulazione è poi utilizzato per calcolare gli indicatori di prestazione (KPI) e generare grafici comparativi. La struttura modulare dello script, che include chiamate a funzioni esterne e l'uso di percorsi dinamici garantisce flessibilità e riutilizzabilità del codice.

2.6.2 Script di generazione della traiettoria

Lo script `pathCreator.m` è responsabile della creazione del tracciato di riferimento utilizzato durante la simulazione. L'utente seleziona la tipologia di percorso da generare, tra le opzioni disponibili vi è il tracciato conforme alla norma ISO 3888-2-2011, utilizzato per testare le manovre di evitamento ostacolo (*obstacle avoidance*), oltre ad eventuali configurazioni personalizzate.

In funzione del tracciato scelto, lo script richiama la funzione di generazione alla quale vengono passati parametri geometrici come ampiezza dell'offset, lunghezza delle sezioni e posizionamento dei tratti iniziali e finali. Il risultato è una traiettoria densa in coordinate spaziali generata tramite interpolazione e visualizzata graficamente per verificarne la correttezza. Lo script si configura come uno strumento indipendente, utile sia per generare tracciati normati, sia per sviluppare percorsi personalizzati. I dati generati sono poi utilizzati come input nella simulazione, costituendo la base su cui i controllori devono effettuare il tracking.

2.7 Layout del percorso di prova

Per la valutazione delle prestazioni dei controllori implementati e per garantire risultati confrontabili e ripetibili, il percorso di prova adottato è stato definito in conformità della norma internazionale ISO 3888-2-2011 [29]. La norma descrive un percorso standardizzato utilizzato per testare la capacità del veicolo di eseguire manovre improvvise di cambio corsia per evitare un possibile ostacolo.

Il percorso presenta una lunghezza complessiva di 61 metri, suddiviso in cinque sezioni:

Sezione	Lunghezza (m)	Larghezza (m)	Descrizione
1	12	$1,1 \times \text{larghezza veicolo} + 0,25\text{m}$	Ingresso iniziale
2	13,5	Non specificata	Cambio di corsia
3	11	$\text{larghezza veicolo} + 1\text{m}$	Corsia di offset
4	12,5	Non specificata	Ritorno in corsia
5	12	$1,3 \times \text{larghezza veicolo} + 0,25\text{m}$	Uscita finale

Tabella 1: sezioni del percorso di riferimento

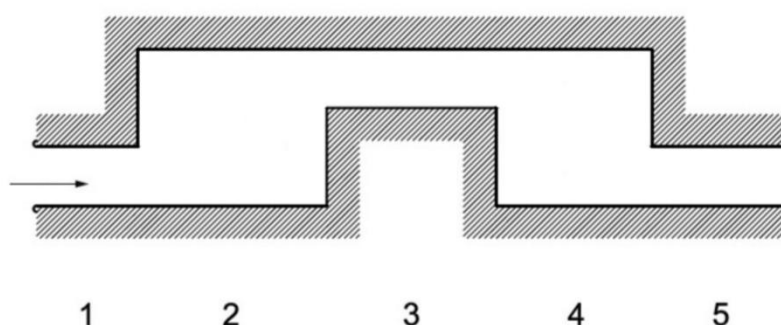


Figura 6: layout del percorso di prova

Un elemento chiave del percorso di prova è rappresentato dall'offset laterale tra le corsie, che ha un valore fisso di 5 metri. Questo disassamento costringe il veicolo a compiere un rapido spostamento laterale, mettendo alla prova il controllore in termini di capacità di reazione, precisione di manovra e gestione delle forze laterali. Il percorso descritto è delimitato da coni di altezza minima pari a 500 mm, disposti secondo quanto prescritto dalla norma, cioè con le basi tangenti ai limiti del tracciato.

La scelta di questo percorso non è casuale in quanto consente di analizzare la risposta del veicolo in presenza di variazioni repentine di traiettoria, valutando l'accuratezza nel mantenimento della traiettoria di riferimento e la stabilità dinamica. L'offset laterale, in particolare, rappresenta la condizione più sfidante e discriminante poiché pone l'accento sulla prontezza del sistema di controllo nel gestire manovre critiche. Infine, l'impiego di un percorso normato e riconosciuto a livello internazionale, garantisce la possibilità di confrontare i risultati ottenuti con quelli presenti in letteratura e/o in altre campagne sperimentali, rafforzando la validità e la replicabilità delle prove effettuate.

2.8 Key Performance Indicators

Per la valutazione comparativa delle prestazioni, sono stati definiti quattro indicatori chiave di prestazione (Key Performance Indicators, KPI), mirati a quantificare la qualità del tracking e la stabilità del sistema, i KPI sono fondamentali nella valutazione dell'efficacia di una strategia di controllo, soprattutto quando si effettuano confronti tra diverse strategie, come nel caso di questo studio. In questo contesto, i KPI utilizzati sono:

1. Errore laterale massimo (Maximum Lateral Error), misura il massimo scostamento laterale rispetto alla traiettoria di riferimento durante tutta la simulazione:

$$e_{y,max} = \max_{i \in [1, N]} |y_{ref}(i) - y(i)|$$

2. Errore angolare massimo (Maximum Heading Angle Error), misura la massima deviazione dell'orientamento del veicolo rispetto all'orientamento desiderato:

$$e_{\psi,max} = \max_{i \in [1, N]} |\psi_{ref}(i) - \psi(i)|$$

3. RMS dell'errore laterale (Root Mean Square Error of Lateral Error), rappresenta la radice quadrata della media del quadrato degli scarti laterali:

$$RMSE_{e_y} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_{ref}(i) - y(i))^2}$$

4. RMS dell'errore angolare (Root Mean Square Error of Heading Angle Error), rappresenta la radice quadrata della media del quadrato della differenza tra orientamento desiderato e reale:

$$RMSE_{e_{\psi}} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\psi_{ref}(i) - \psi(i))^2}$$

I quattro KPI descritti consentiranno di valutare le prestazioni dei diversi controllori non solo in termini di errori massimi, che evidenziano eventi critici, ma anche in termini di errori medi globali, attraverso gli RMSE, offrendo una visione completa dell'efficacia, stabilità e precisione dei controllori testati.

3 Controllori

3.1 Controllore PID

Il controllore PID è stato utilizzato come soluzione di riferimento per il controllo di path tracking.

L'azione di controllo calcola l'angolo di sterzo anteriore δ sulla base della combinazione di due errori principali:

1. errore laterale e_y , ovvero la distanza laterale rispetto alla traiettoria di riferimento;
2. errore di orientamento e_ψ , ossia la differenza tra l'angolo di imbardata del veicolo e l'orientamento della traiettoria.

La legge di controllo è data da:

$$\delta = \underbrace{(K_{p,ey}e_y + K_{i,ey}e_y \int e_y dt + K_{d,ey}\dot{e}_y)}_{\text{contributo errore laterale}} + \underbrace{(K_{p,e\psi}e_\psi + K_{i,e\psi}e_\psi \int e_\psi dt + K_{d,e\psi}\dot{e}_\psi)}_{\text{contributo errore di imbardata}}$$

Per garantire prestazioni ottimali al variare della velocità, il controllore è stato testato su un intervallo di velocità compreso tra 10 km/h e 40 km/h, con passi di 5 km/h. Per ciascuna velocità, sono stati selezionati e ottimizzati set di guadagni PID specifici.

Guadagni PID adottati alle varie velocità

	Guadagni errore laterale			Guadagni errore di imbardata		
Velocità (km/h)	$K_{p,ey}$	$K_{i,ey}$	$K_{d,ey}$	$K_{p,e\psi}$	$K_{i,e\psi}$	$K_{d,e\psi}$
10	0.8	0.8	0.3	0.7	0.8	0.2
15	0.8	0.7	0.3	0.7	0.7	0.3
20	0.6	0.7	0.3	0.7	0.5	0.15
25	0.8	0.7	0.2	0.7	0.65	0.1
30	0.55	0.5	0.15	1.1	0.8	0.2
35	0.8	0.55	0.2	1.2	0.95	0.6
40	0.35	0.2	0.15	1.1	0.65	0.65

Tabella 2: guadagni del PID

3.1.1 Vincoli sull'attuatore

Per motivi di sicurezza e realismo, è stato imposto un vincolo fisico sull'angolo massimo di sterzo:

$$|\delta| \leq \pi/3 \quad (\approx 60^\circ)$$

Inoltre, il comando è aggiornato ogni 0.02 secondi per garantire fluidità e stabilità della risposta. Questa necessità di ritardatura a ogni variazione della velocità dimostra il limite principale del PID, cioè che non è facilmente scalabile e richiede un tuning dedicato in funzione della velocità e delle condizioni operative. Questo ne limita la versatilità rispetto a controllori avanzati come LMPC e NMPC, in grado di adattarsi a scenari variabili in modo predittivo ed efficiente.

3.1.2 Risultati delle simulazioni PID

In questo paragrafo sono riportati i grafici delle simulazioni effettuate con il controllore PID per ciascuna velocità.

- 10 km/h

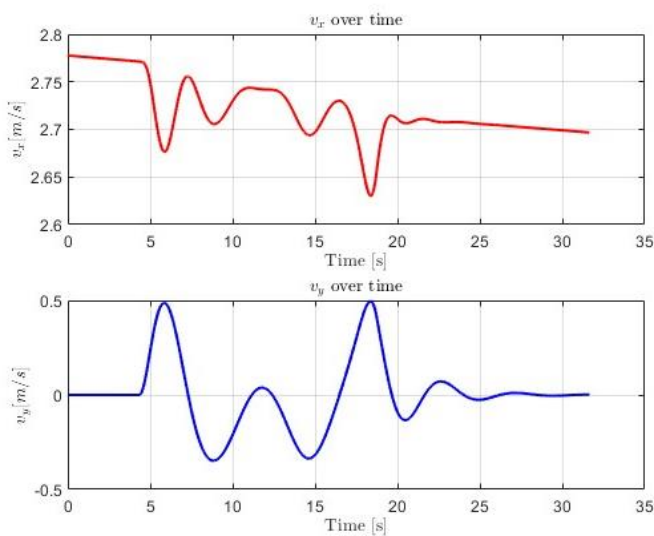


Figura 7: PID, v_x e v_y nel tempo

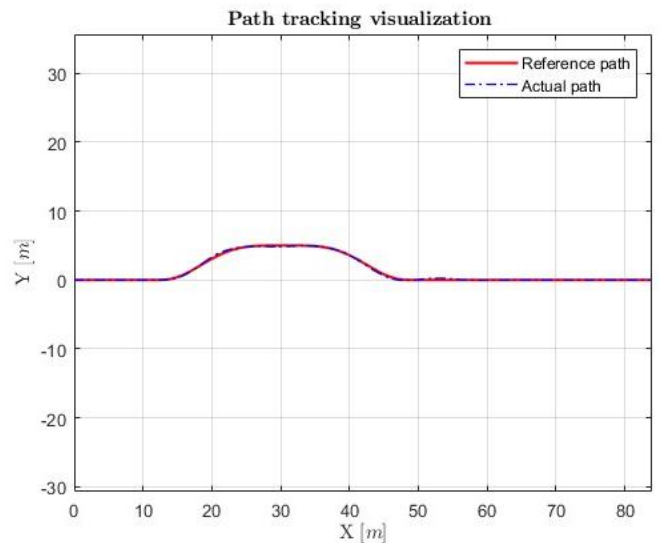


Figura 8: PID, andamento del tracking

Errors over time visualization

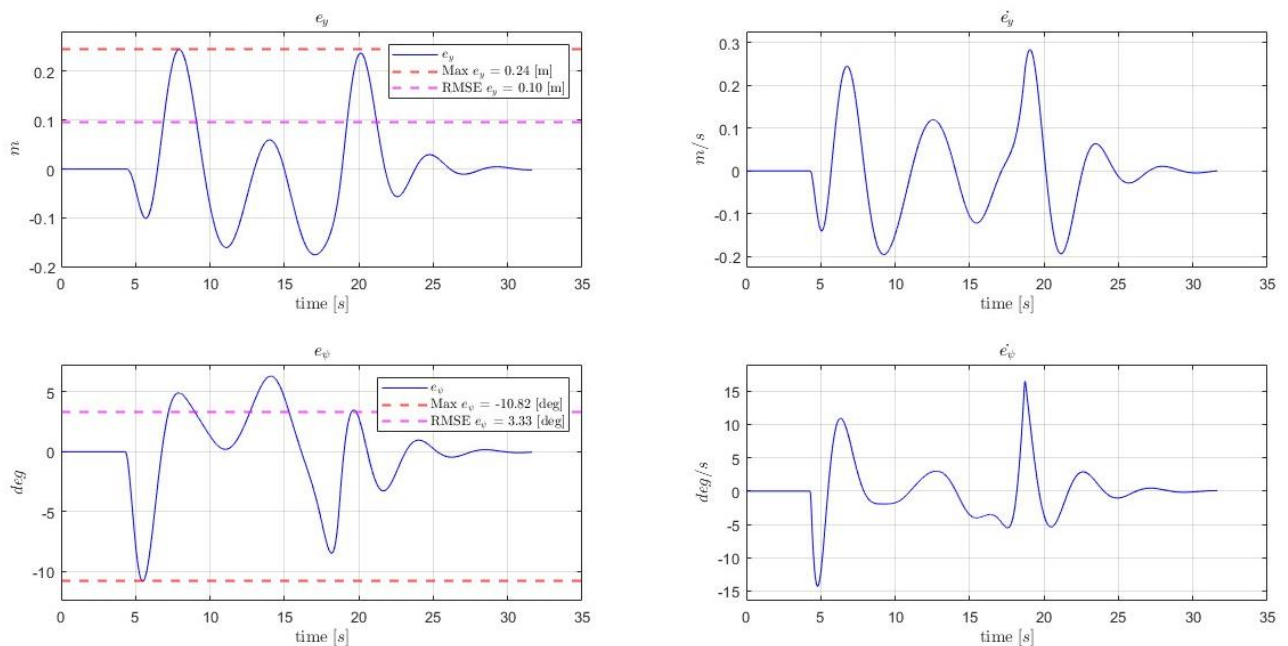


Figura 9: PID, andamento degli errori nel tempo

- 15 km/h

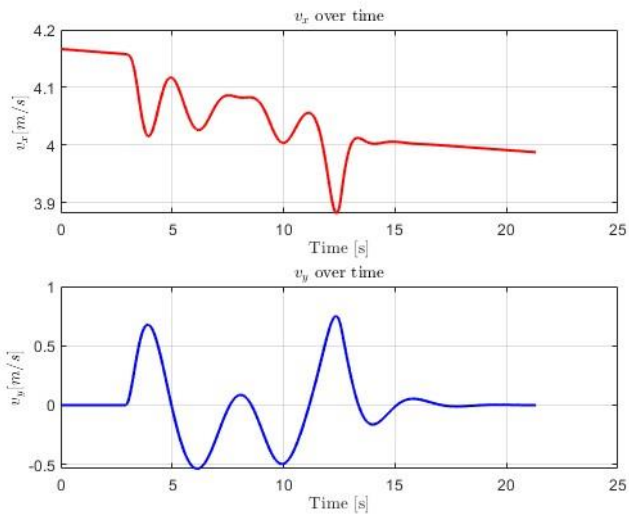


Figura 10: PID, v_x e v_y nel tempo

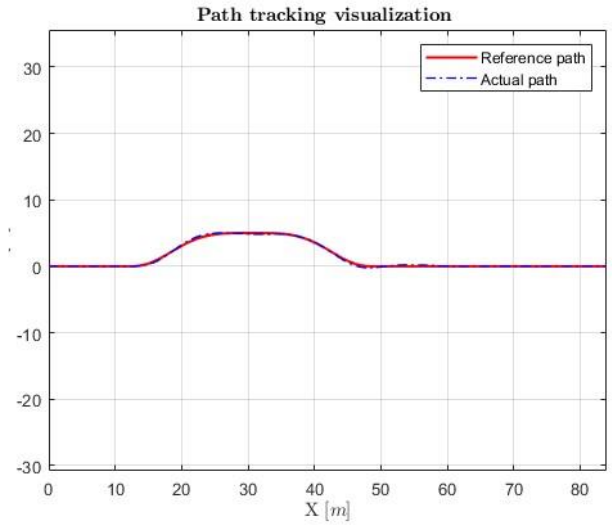


Figura 11: PID, andamento del tracking

Errors over time visualization

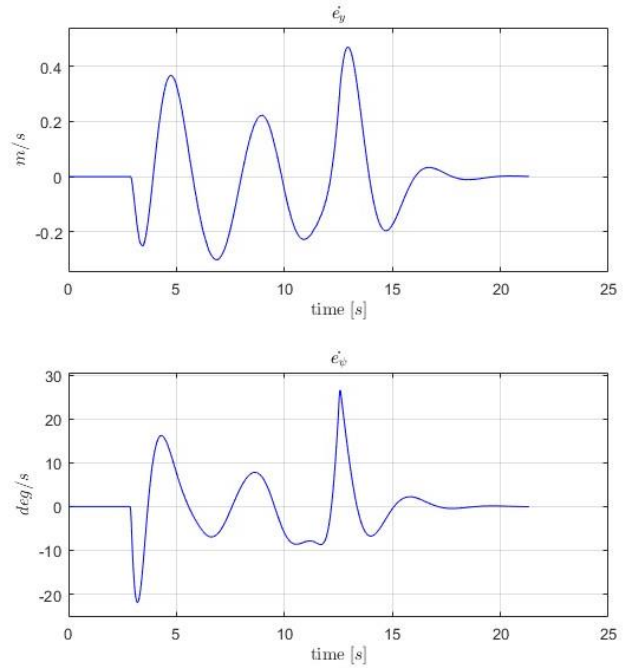
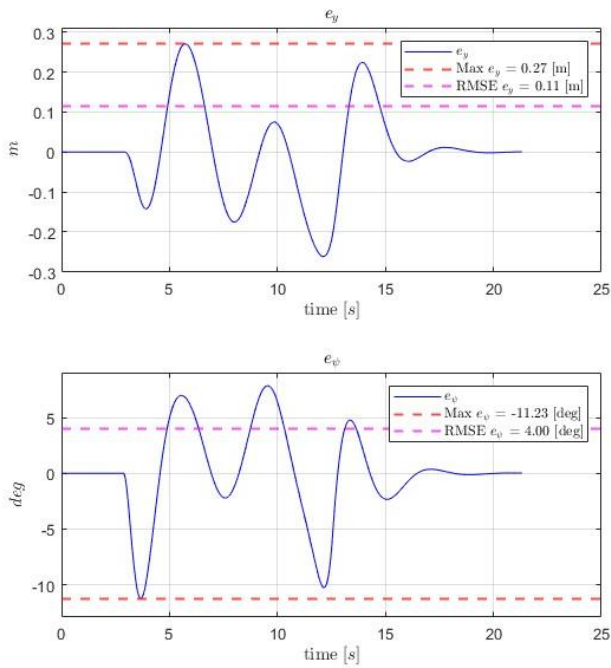


Figura 12: PID, andamento degli errori nel tempo

- 20 km/h

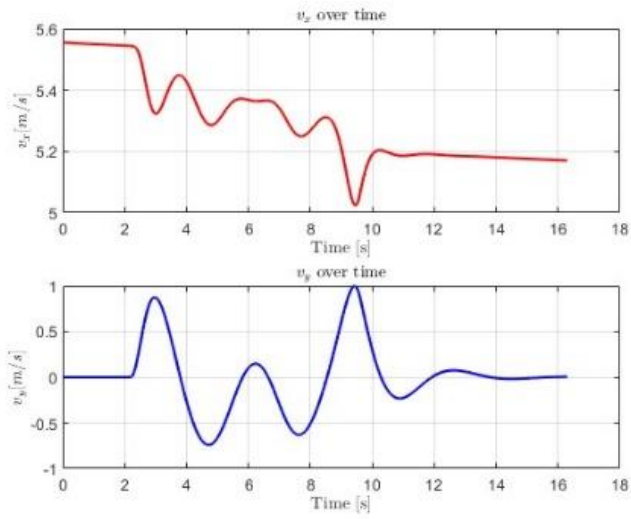


Figura 13: PID, v_x e v_y nel tempo

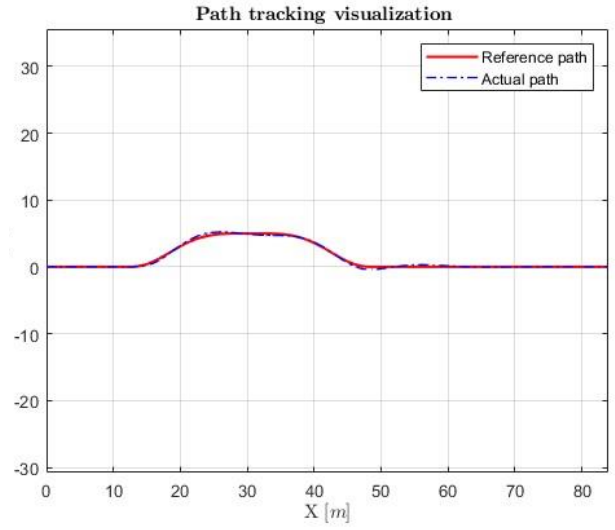


Figura 14: PID, andamento del tracking

Errors over time visualization

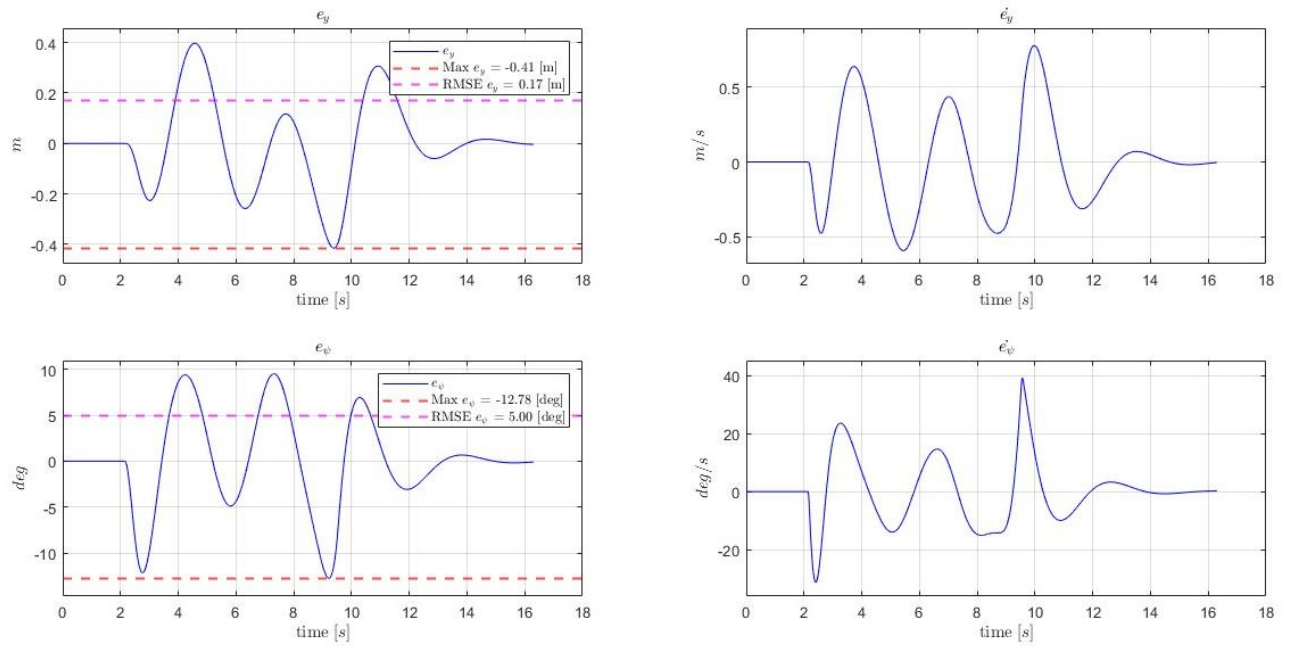


Figura 15: PID, andamento degli errori nel tempo

- 25 km/h

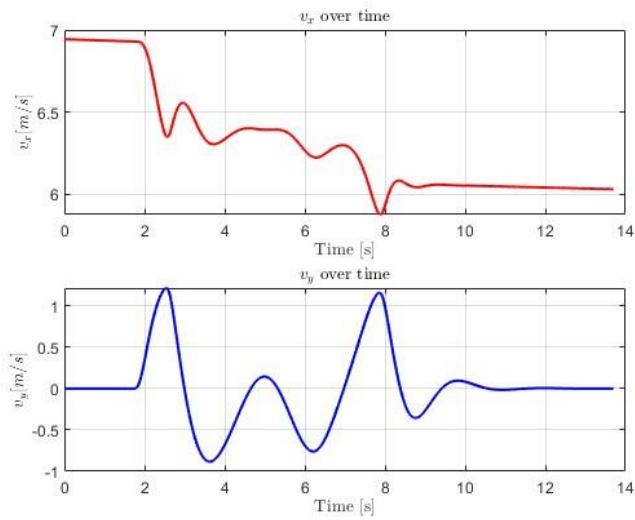


Figura 16: PID, v_x e v_y nel tempo

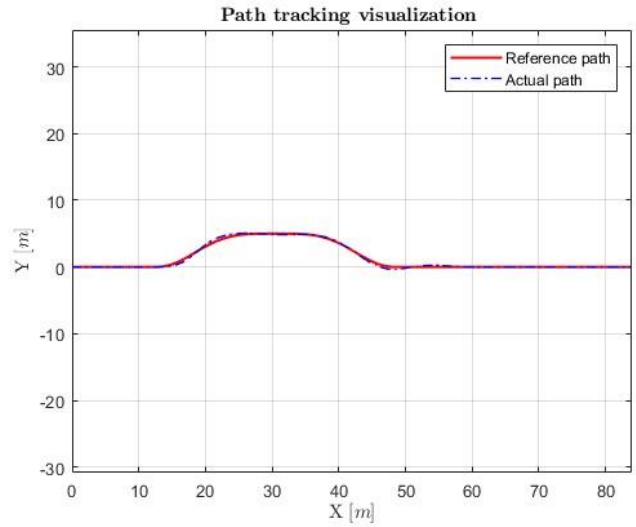


Figura 17: PID, andamento del tracking

Errors over time visualization

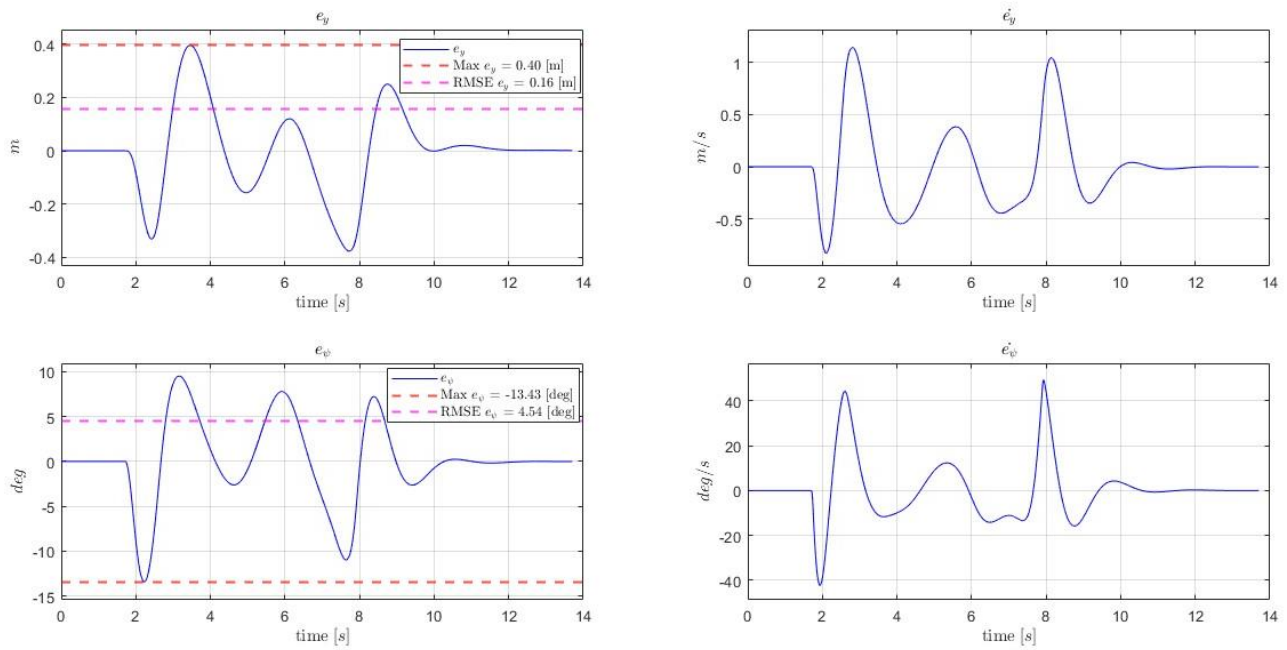


Figura 18: PID, andamento degli errori nel tempo

- 30 km/h

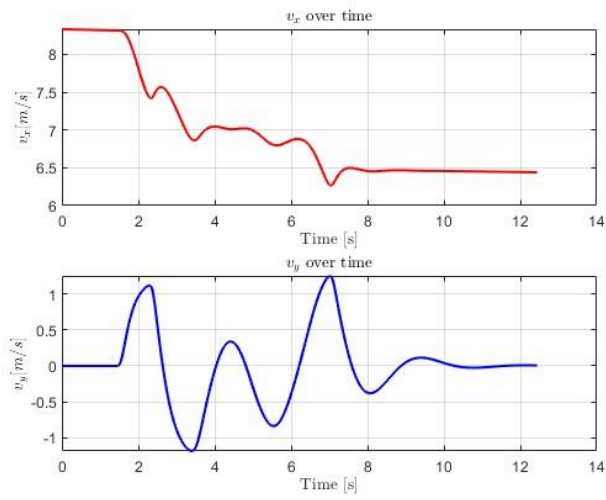


Figura 19: PID, v_x e v_y nel tempo

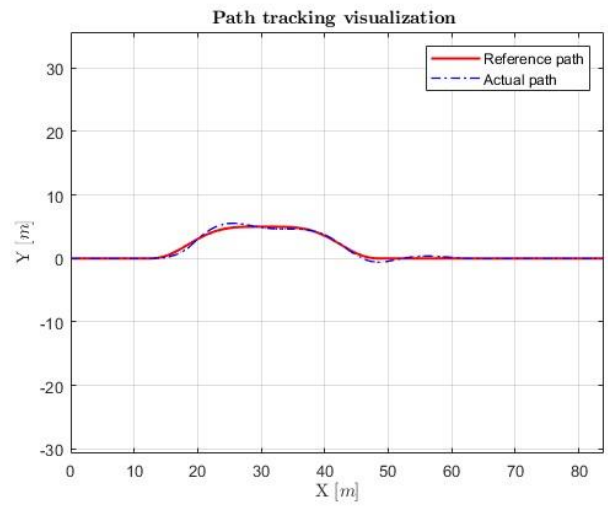


Figura 20: PID, andamento del tracking

Errors over time visualization

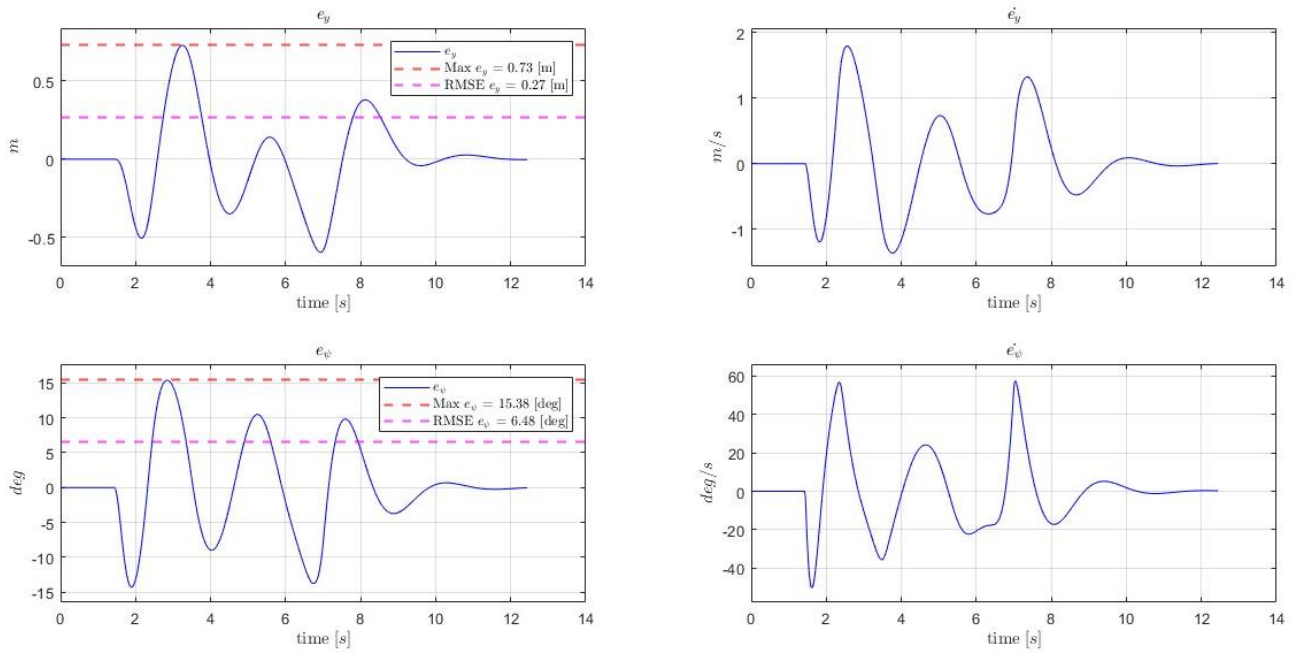


Figura 21: PID, andamento degli errori nel tempo

- 35 km/h

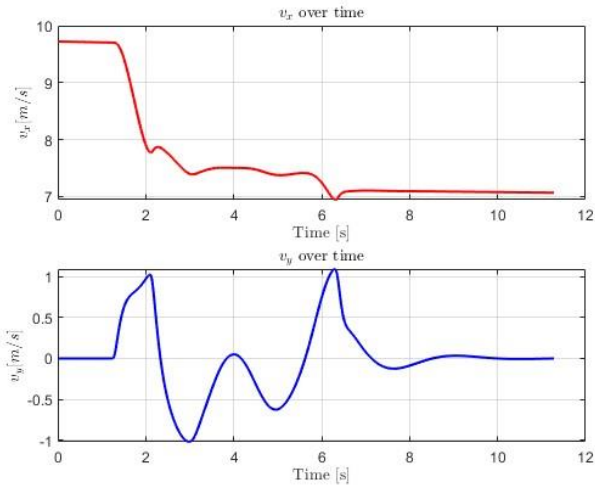


Figura 22: PID, v_x e v_y nel tempo

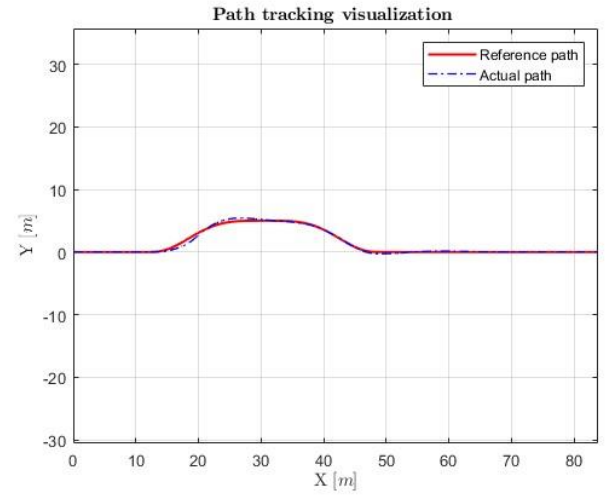


Figura 23: PID, andamento del tracking

Errors over time visualization

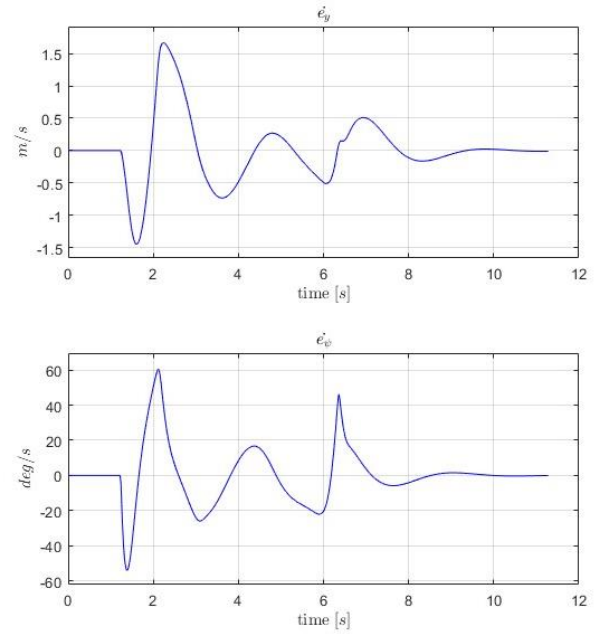
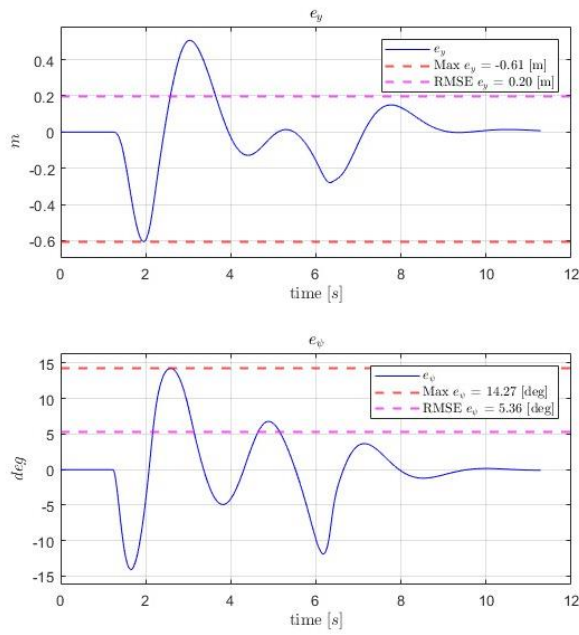


Figura 24: PID, andamento degli errori nel tempo

- 40 km/h

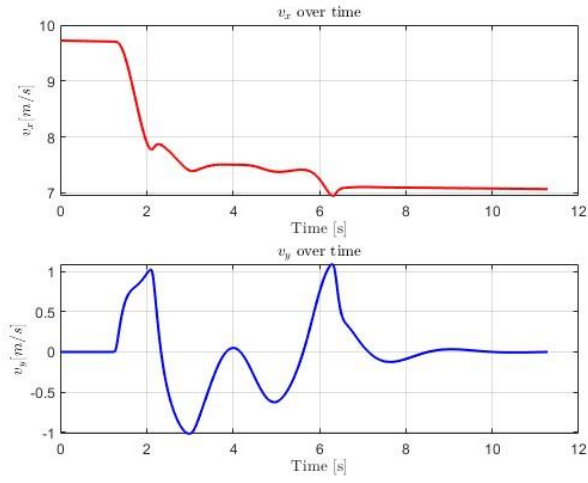


Figura 25: PID, v_x e v_y nel tempo

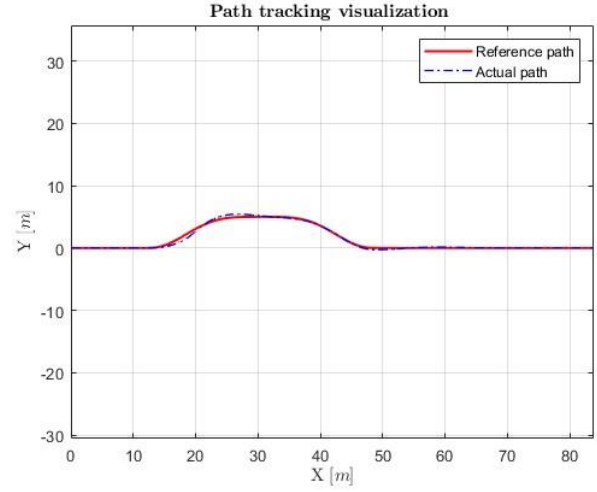


Figura 26: PID, andamento del tracking

Errors over time visualization

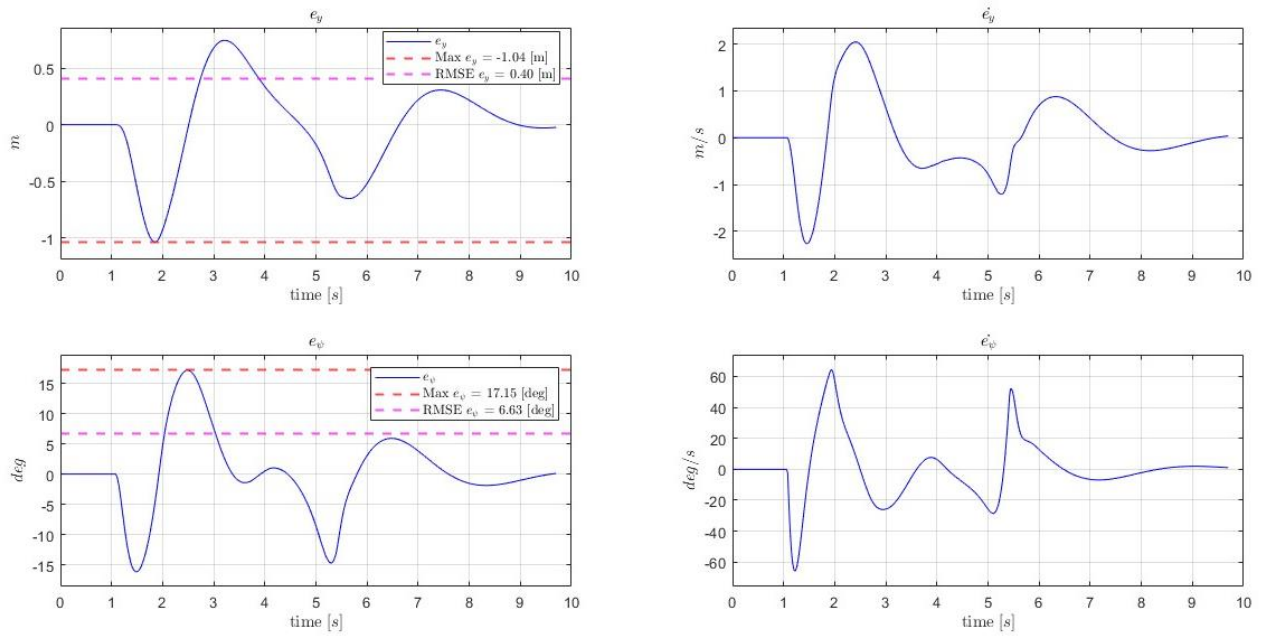


Figura 27: PID, andamento degli errori nel tempo

3.1.3 Considerazioni

Dai grafici relativi alle simulazioni effettuate con il controllore PID emergono sia i punti di forza sia i limiti di questa soluzione al variare della velocità di marcia. A velocità contenute (10 - 15 km/h) il PID mostra un ottimo comportamento in termini di tracking della traiettoria, infatti l'errore laterale rimane ridotto e il termine derivativo riesce ad attenuare le oscillazioni residue. In queste condizioni, il veicolo segue con precisione la traiettoria di riferimento e non si osservano fenomeni di instabilità, a conferma del buon tuning del controllore adottato.

Quando si passa a velocità intermedie (20 – 30 km/h), l'incremento dell'inerzia complessiva del veicolo si traduce in un lieve aumento dell'errore laterale massimo e dei tempi di assestamento, le oscillazioni risultano più pronunciate e l'azione derivativa fatica a smorzarle completamente.

La situazione tende a essere critica alle alte velocità (35 – 40 km/h) dove l'angolo di sterzo tende a saturare raggiungendo il vincolo imposto e l'errore di imbardata presenta picchi considerevoli. Si nota come sia presente un comportamento oscillatorio persistente, segno che né il termine proporzionale né quello derivativo sono in grado di compensare efficacemente le elevate inerzie del sistema.

3.2 Controllore LMPC

Il controllore LMPC rappresenta un approccio avanzato basato sulla previsione futura del comportamento del veicolo lungo un orizzonte temporale finito. A ogni passo di simulazione, il controllore calcola la sequenza ottimale di comandi di sterzo che minimizza una funzione di costo costruita sulla distanza tra la traiettoria di riferimento e quella prevista dal modello del veicolo, penalizzando anche l'utilizzo eccessivo degli attuatori.

Il problema viene risolto in modalità receding horizon, a ogni intervallo di campionamento viene calcolata una sequenza ottimale di comandi di controllo lungo l'intero prediction horizon, ma viene applicato solo il primo comando della sequenza. Il processo viene poi ripetuto al passo successivo, aggiornando lo stato del veicolo e ricalcolando una nuova ottimizzazione, in modo da adattarsi costantemente alle condizioni correnti.

3.2.1 Configurazione del controllore

Nel presente lavoro, il LMPC è stato configurato con le seguenti impostazioni, direttamente derivate dalla struttura implementata nei file di progetto:

Setup LMPC	
Modello interno	Single-track dynamical lateral model
Sample time	0,02 s
Discretisation time	0,02 s
Prediction horizon (N)	20 steps
Intervallo di velocità (km/h)	$v = [10, 15, 20, 25, 30, 35, 40]$

Tabella 3: setup LMPC

Dove:

- Sample time: è l'intervallo di tempo tra due aggiornamenti consecutivi del comando di controllo. Rappresenta la frequenza con cui il controllore calcola una nuova azione di controllo e la applica al sistema. Nel contesto dell'MPC, è il tempo di esecuzione con cui il problema di ottimizzazione viene risolto e il comando aggiornato.
- Discretisation time: è il passo temporale con cui il modello dinamico interno viene discretizzato lungo l'orizzonte predittivo. Corrisponde alla suddivisione dell'orizzonte totale in un certo numero di passi N. In molti casi, come nel caso attuale, coincide con il sample time, ma può essere anche inferiore, nel caso si voglia avere una discretizzazione più fine rispetto alla frequenza di aggiornamento del controllo

- Prediction horizon: è definito come il numero di passi N in avanti lungo i quali il controllore predittivo effettua la stima degli stati futuri e ottimizza la sequenza dei comandi. Il controllore calcola una traiettoria predetta composta da N punti, ciascuno separato da un intervallo di tempo costante. Oltre che in termini di passi, l'orizzonte predittivo può essere espresso anche come una durata temporale totale. Un orizzonte predittivo più lungo consente di anticipare meglio le variazioni della traiettoria, ma aumenta la complessità computazionale.

Il modello lineare utilizzato nel LMPC viene calcolato mediante linearizzazioni locali attorno a diversi punti di funzionamento, ciascuno corrispondente a una diversa velocità di marcia. Nel progetto considerato, sono stati generati controllori LMPC per velocità comprese tra 10 km/h e 40 km/h, con incrementi di 5 km/h. In fase di simulazione, il controllore seleziona dinamicamente la linearizzazione più adatta alla velocità corrente.

3.2.2 Risultati delle simulazioni LMPC

In questo paragrafo sono riportati i grafici delle simulazioni effettuate con il controllore LMPC per ciascuna velocità.

- 10 km/h

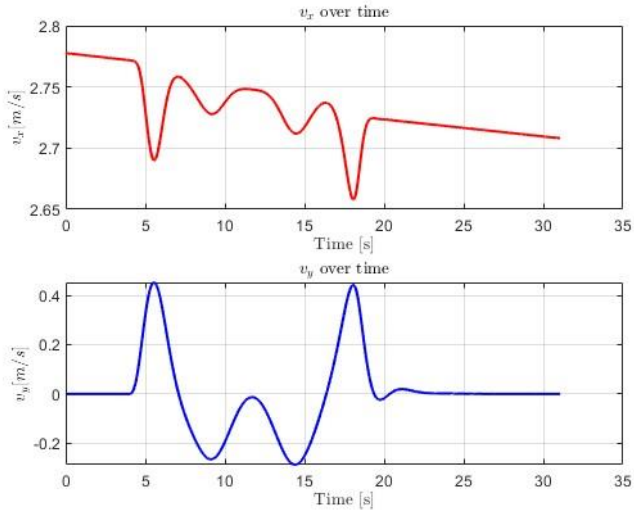


Figura 28: LMPC, v_x e v_y nel tempo

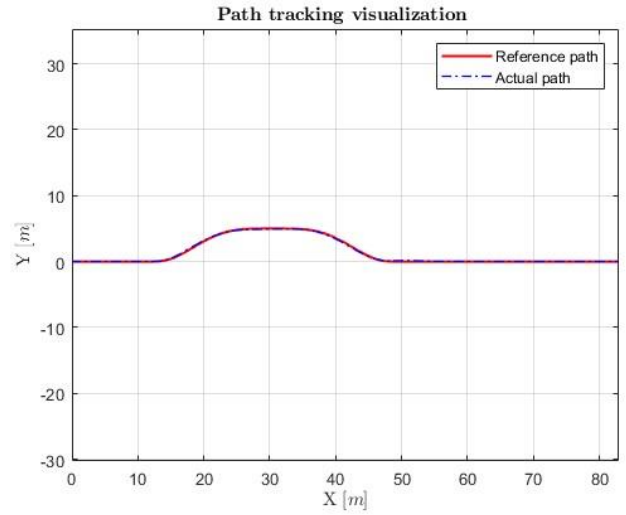


Figura 29: LMPC, andamento del tracking

Errors over time visualization

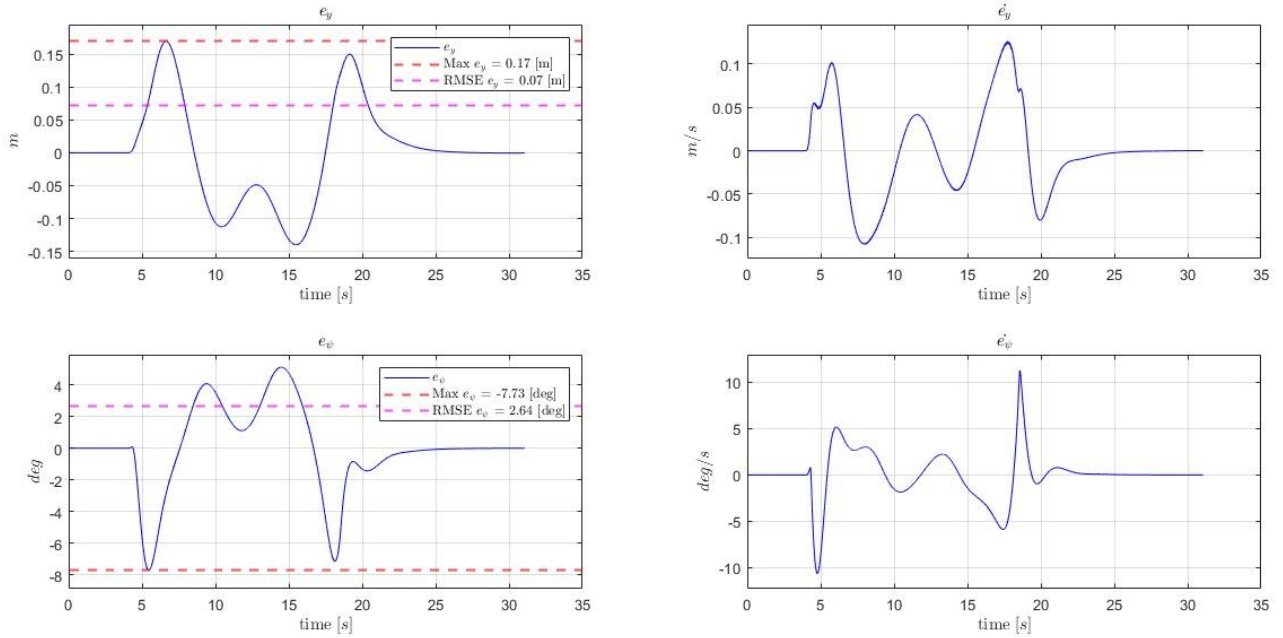


Figura 30: LMPC, andamento degli errori nel tempo

- 15 km/h

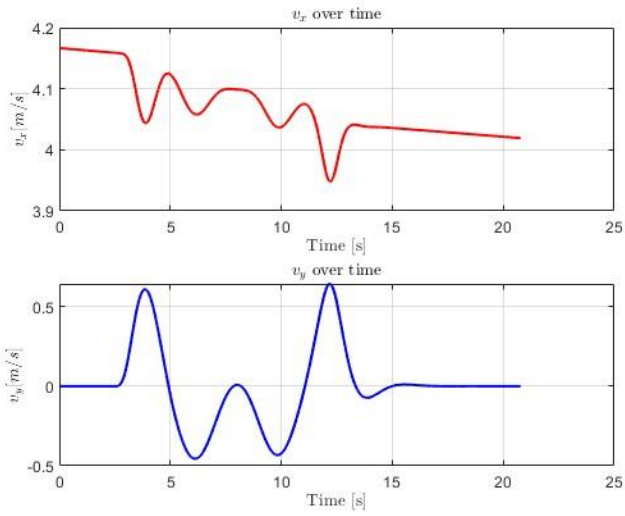


Figura 31: LMPC, v_x e v_y nel tempo

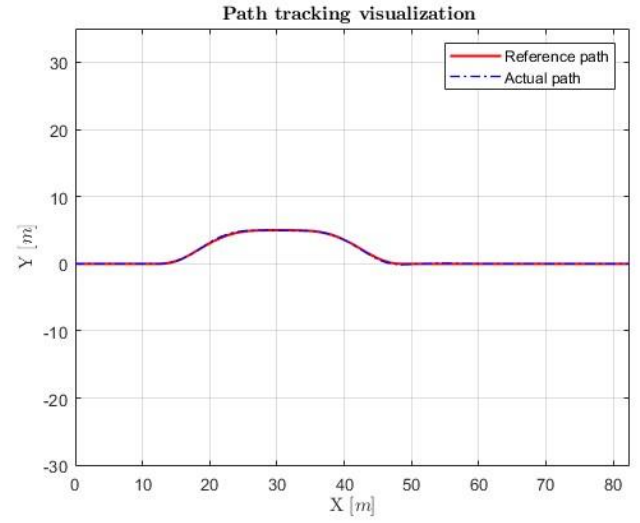


Figura 32: LMPC, andamento del tracking

Errors over time visualization

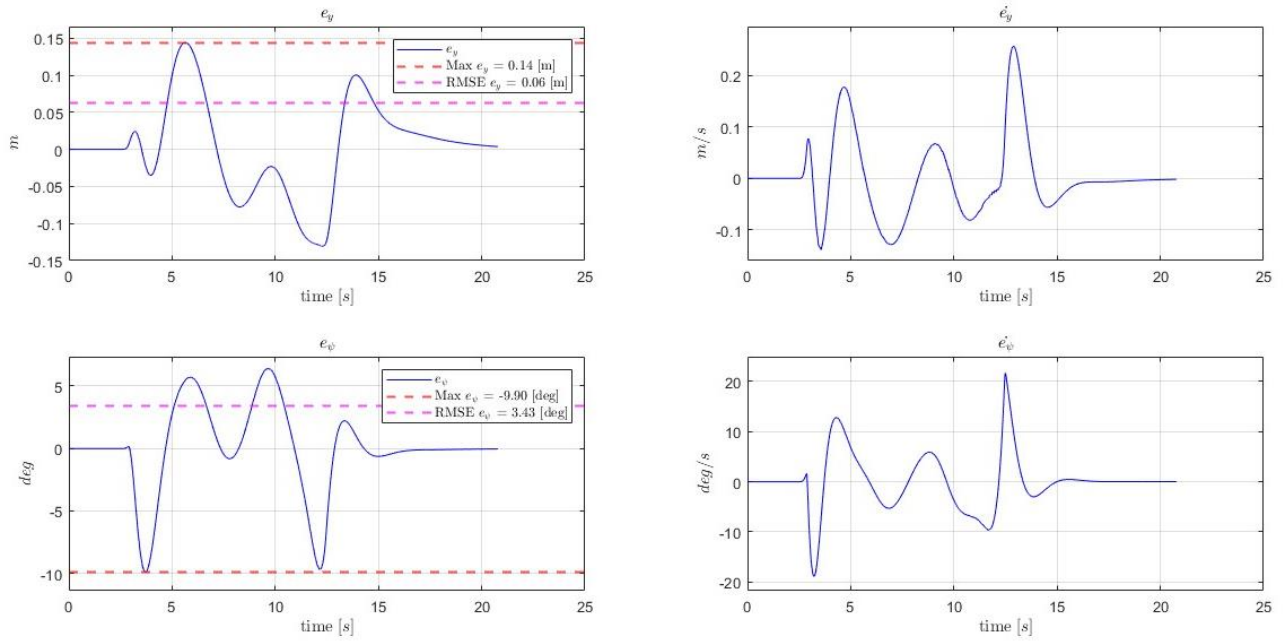


Figura 33: LMPC, andamento degli errori nel tempo

- 20 km/h

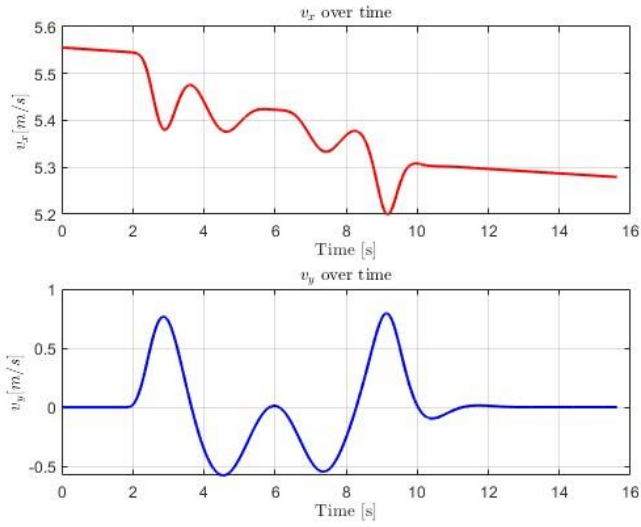


Figura 34: LMPC, v_x e v_y nel tempo

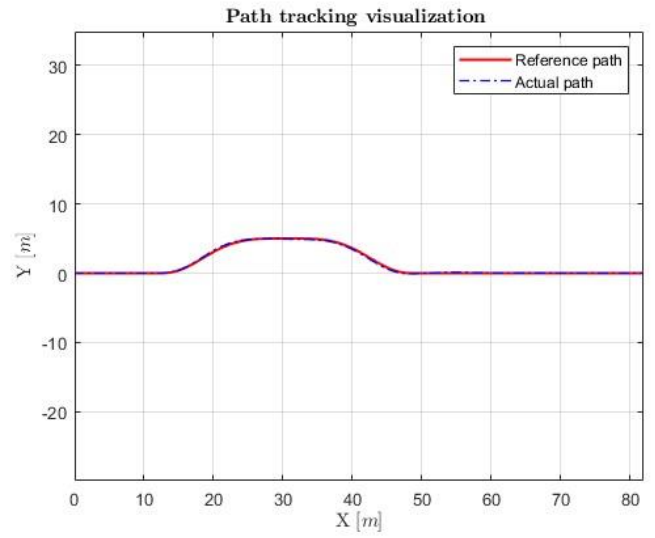


Figura 35: LMPC, andamento del tracking

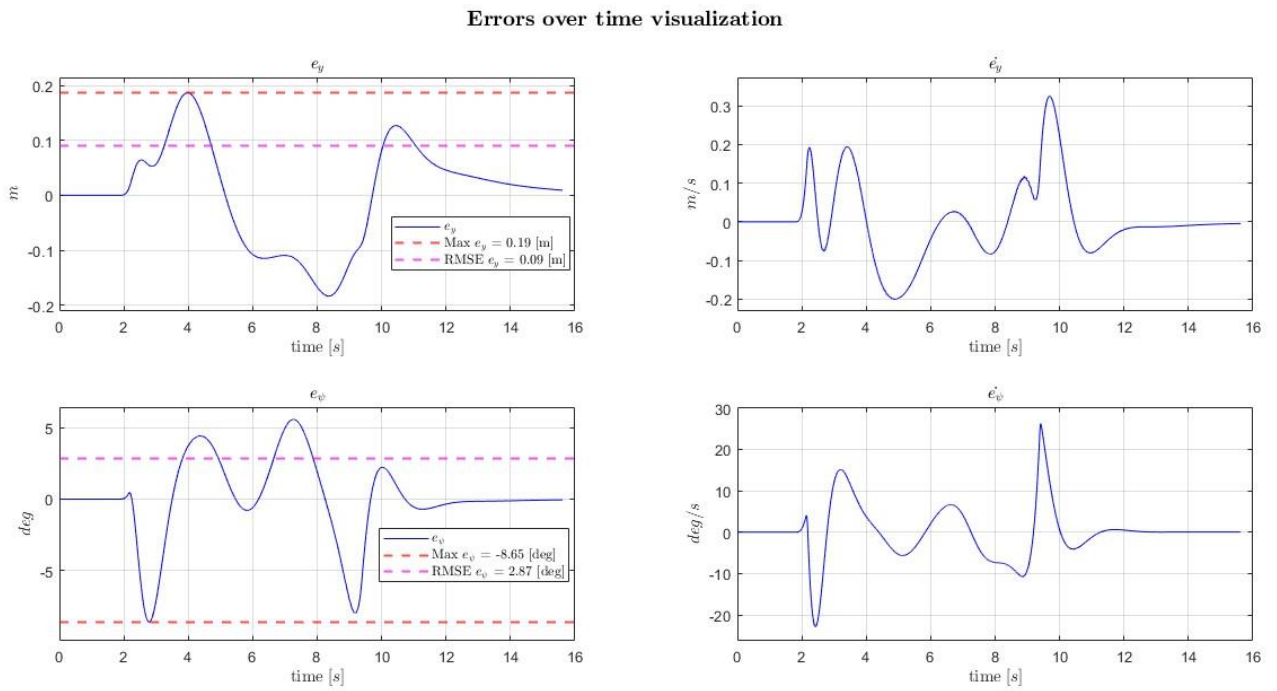


Figura 36: LMPC, andamento degli errori nel tempo

- 25 km/h

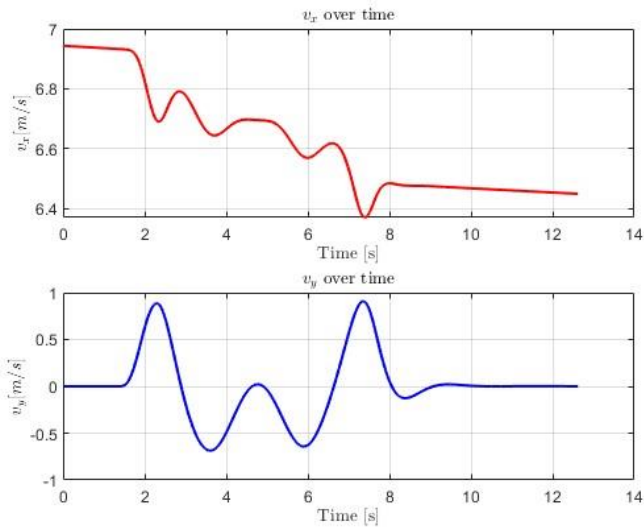


Figura 37: LMPC, v_x e v_y nel tempo

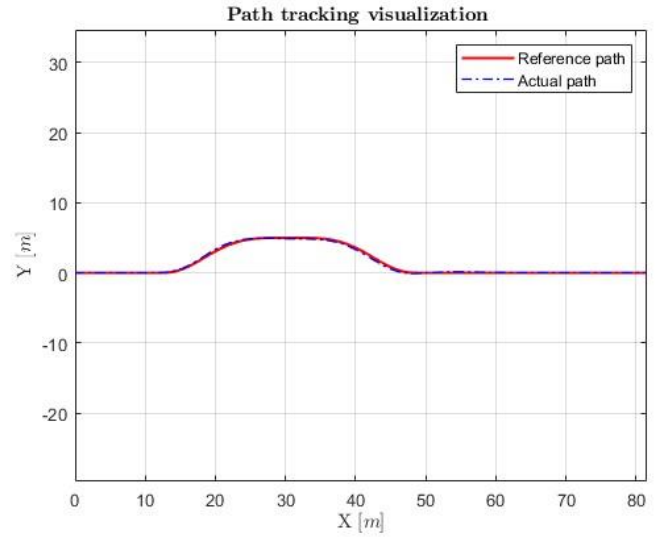


Figura 38: LMPC, andamento del tracking

Errors over time visualization

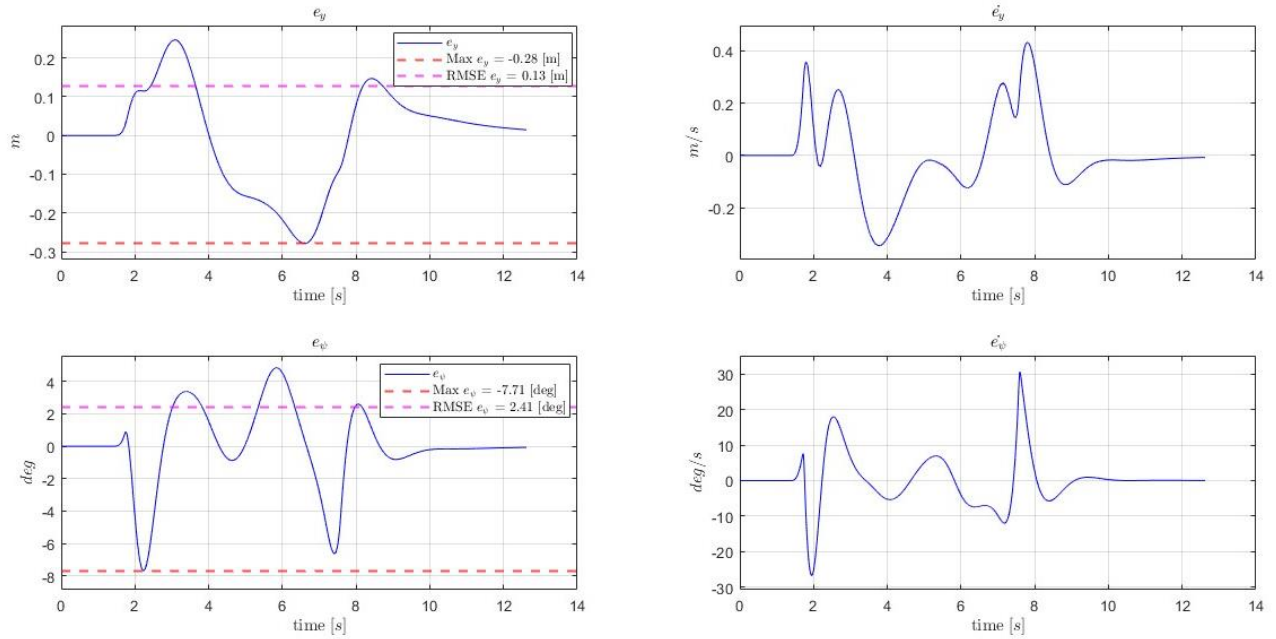


Figura 39: LMPC, andamento degli errori nel tempo

- 30 km/h

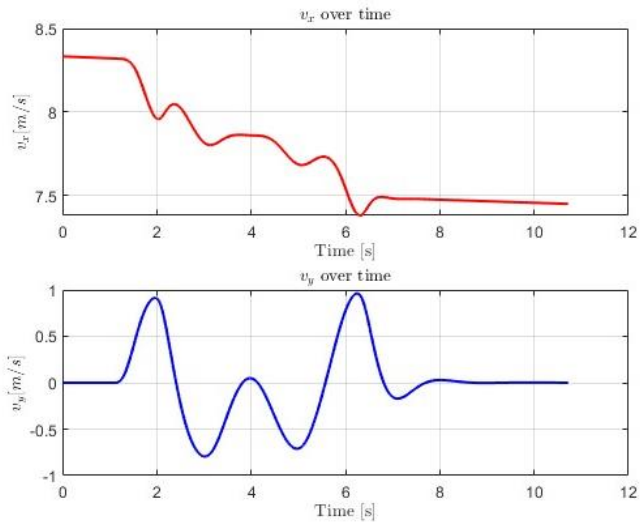


Figura 40: LMPC, v_x e v_y nel tempo

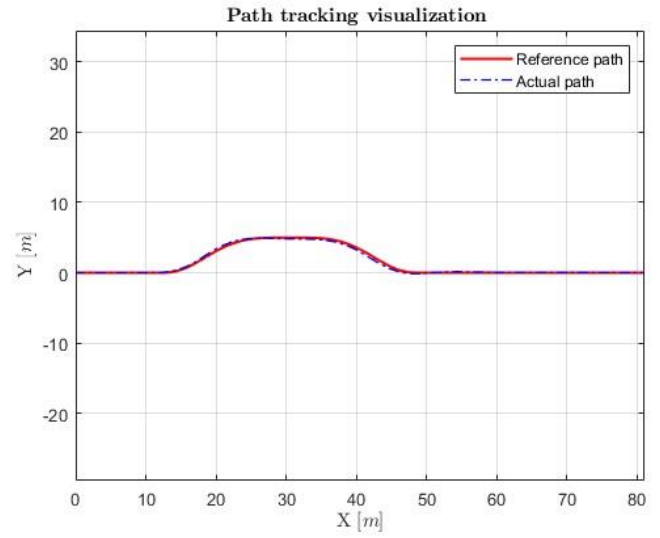


Figura 41: LMPC, andamento del tracking

Errors over time visualization

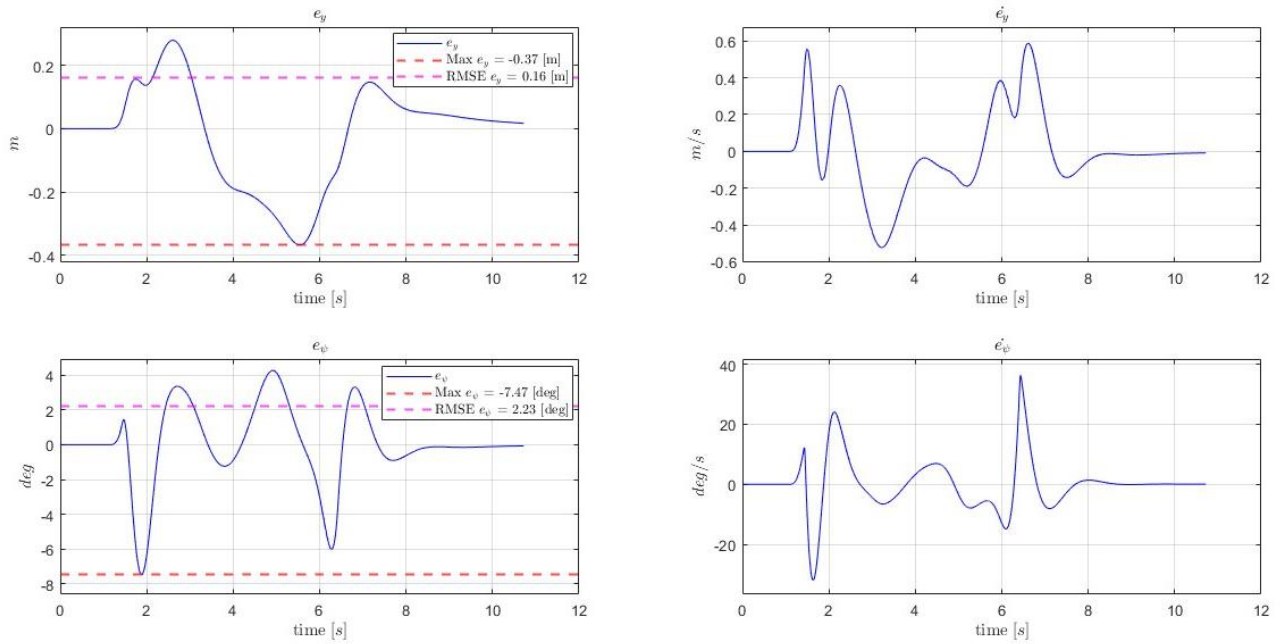


Figura 42: LMPC, andamento degli errori nel tempo

- 35 km/h

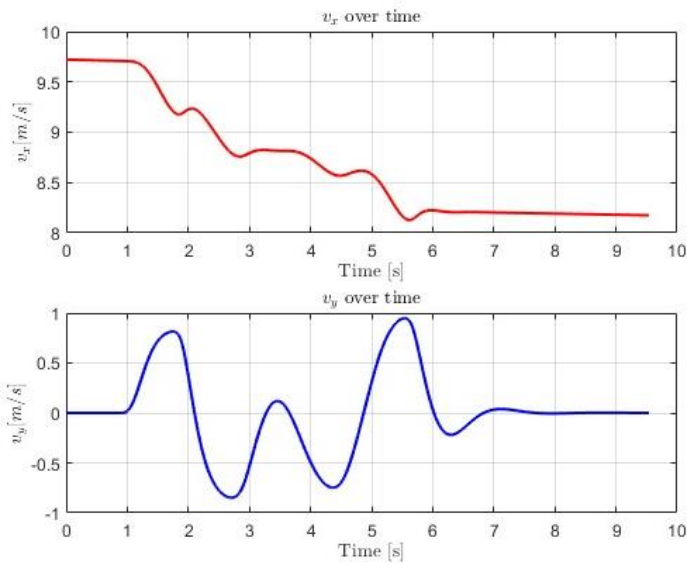


Figura 43: LMPC, v_x e v_y nel tempo

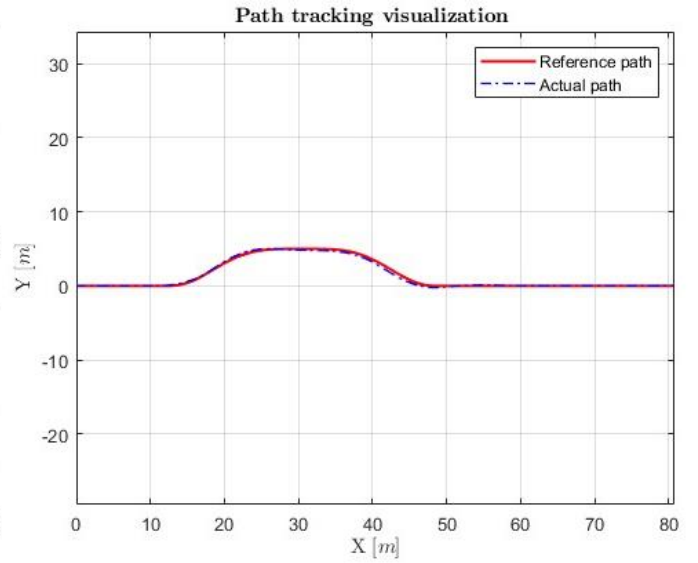


Figura 44: LMPC, andamento del tracking

Errors over time visualization

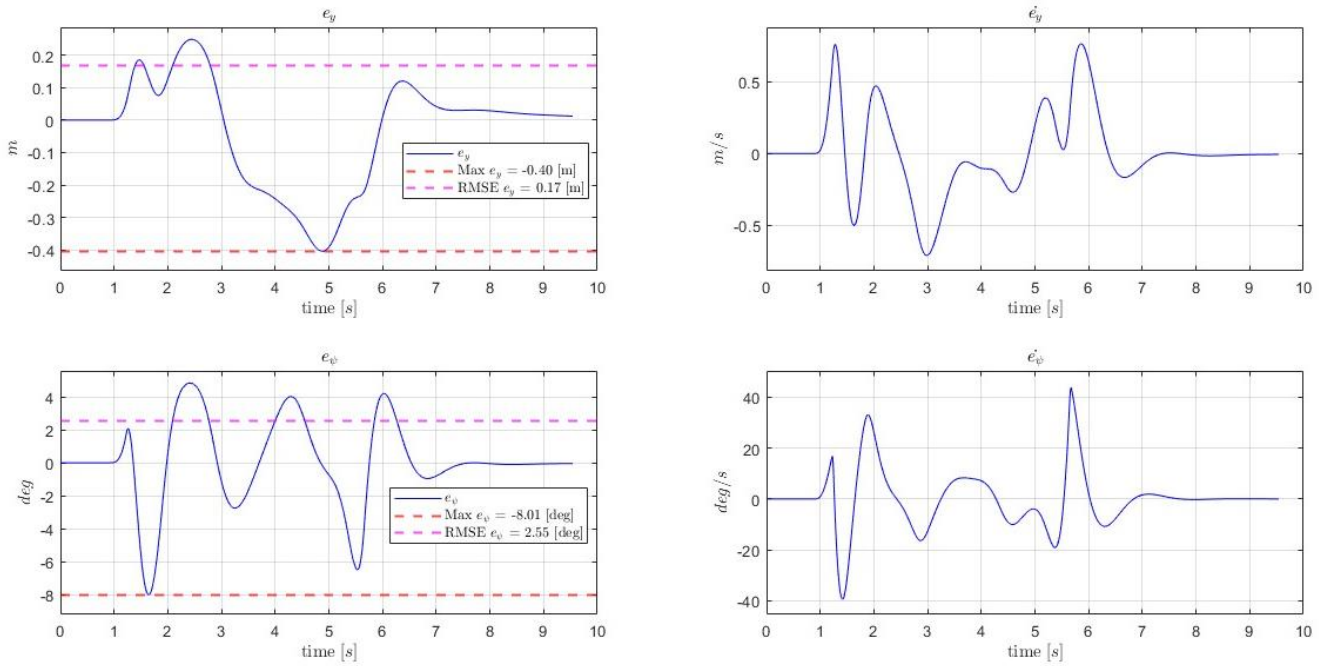


Figura 45: LMPC, andamento degli errori nel tempo

- 40 km/h

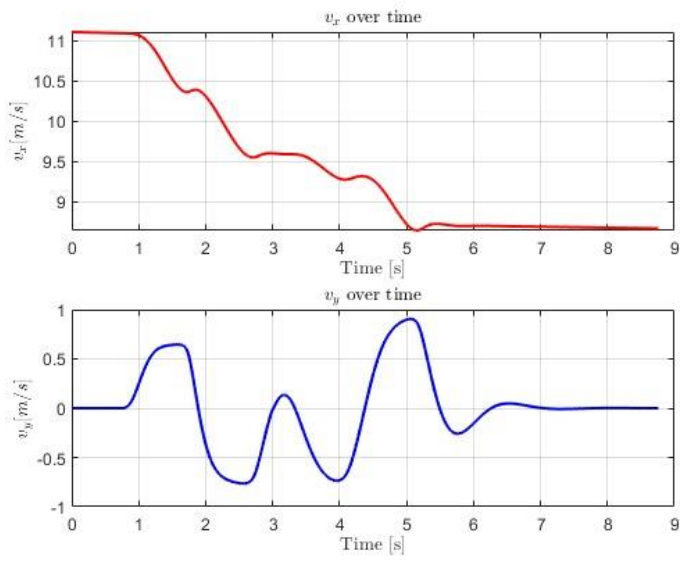


Figura 46: LMPC, v_x e v_y nel tempo

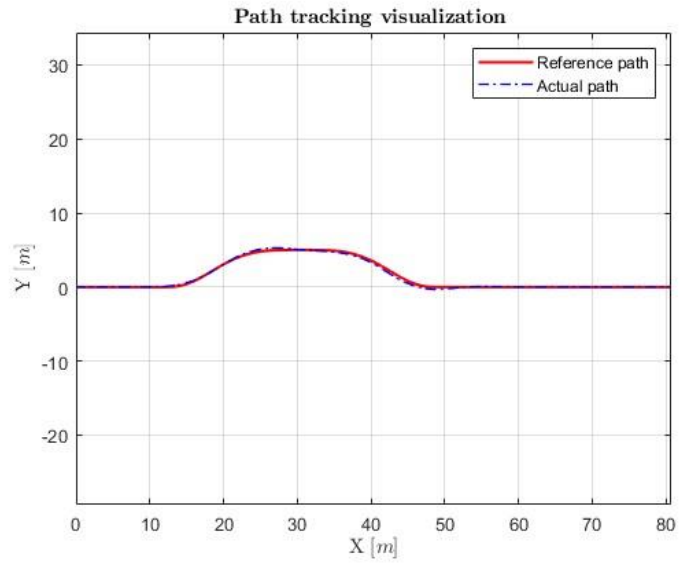


Figura 47: LMPC, andamento del tracking

Errors over time visualization

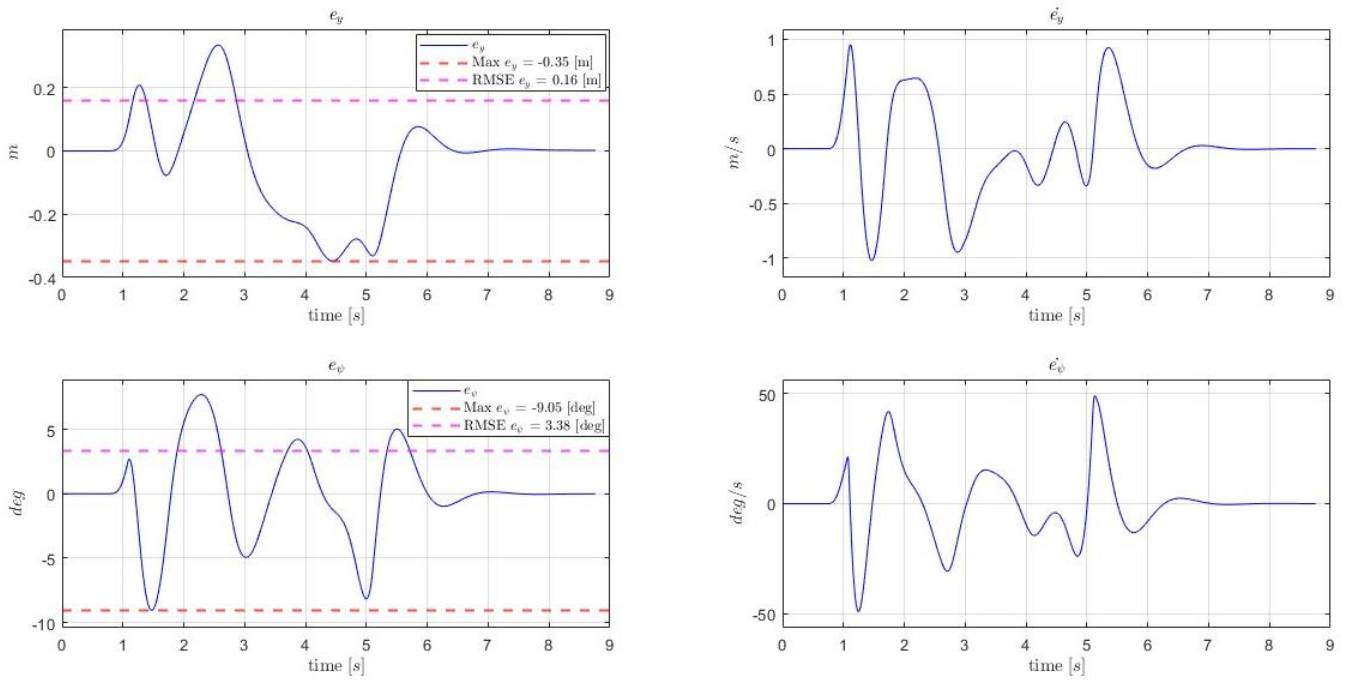


Figura 48: LMPC, andamento degli errori nel tempo

3.2.3 Considerazioni

Dai grafici delle simulazioni effettuate con il controllore LMPC emerge che, alle basse velocità (10 – 15 km/h), il controllore garantisce un tracking della traiettoria praticamente privo di errori significativi. L'errore laterale rimane trascurabile e non si osservano oscillazioni né picchi. Il vincolo sull'angolo di sterzo viene rispettato senza saturazioni, anche nelle curve più strette, dimostrando come l'orizzonte predittivo riesca a gestire vincoli fisici e cambi di direzione in modo efficace.

A velocità intermedie (20 – 30 km/h), nonostante l'aumento dell'inerzia, l'errore sulla traiettoria rimane contenuto. Le curve di risposta non mostrano picchi apprezzabili né oscillazioni marcate, segno che il controllore riesce ad anticipare le manovre necessarie e a modulare l'azione di controllo.

A velocità elevate (35 – 40 km/h) il LMPC riesce a mantenere un comportamento stabile e privo di saturazioni, l'angolo di sterzo segue in modo fluido la traiettoria di riferimento senza superare i limiti imposti e l'errore di imbardata rimane contenuto. Non si evidenziano inoltre fenomeni oscillatori rilevanti.

3.3 Controllore NMPC

Il Nonlinear Model Predictive Control (NMPC) è una strategia di controllo avanzata basata sull'ottimizzazione predittiva che utilizza un modello non lineare del sistema per prevedere l'evoluzione futura dello stato e generare una sequenza ottimale di azioni di controllo. Il controllore risolve un problema di ottimizzazione vincolata su un orizzonte finito a ogni istante di tempo, applicando al sistema solo la prima sequenza calcolata (receding horizon). Questo processo viene poi ripetuto ciclicamente per ogni passo temporale.

Nel contesto del path tracking di un veicolo autonomo, l'NMPC si dimostra particolarmente efficace grazie alla sua capacità di gestire accuratamente le dinamiche non lineari del veicolo, considerare esplicitamente i vincoli fisici e operativi e ancora di garantire un controllo preciso anche in scenari complessi o a velocità elevate.

Il problema di ottimizzazione risolto dall'NMPC si presenta nella forma descritta nella sezione 1.4.5:

$$\min_u \sum_{k=0}^{N-1} (x_{k+1}^T Q x_{k+1} + u_k^T R u_k)$$

Soggetto a:

$$x_{k+1} = f(x_k, u_k)$$

$$x_{min} \leq x_k \leq x_{max}$$

$$u_{min} \leq u_k \leq u_{max}$$

L'implementazione del controllore NMPC è ancora in fase di sviluppo, il progetto prevede l'utilizzo di un modello dinamico non lineare integrato in ambiente MATLAB/Simulink. Al momento non è ancora stato possibile completare la struttura del codice e i parametri del controllore per eseguire le simulazioni comparative e pertanto non ne verranno mostrati i risultati alle diverse velocità. Nonostante l'implementazione del controllore non sia stata ancora portata a termine risulta comunque utile riportare alcuni studi recenti che ne evidenziano le potenzialità applicative in scenari avanzati di path tracking e/o obstacle avoidance.

La presentazione di questi lavori ha pertanto lo scopo di fornire un riferimento concreto sui possibili risultati raggiungibili tramite NMPC.

1. Un primo esempio rilevante è offerto da Sathya et al. (2018) [29], i quali sviluppano un controllore NMPC embedded per veicoli terrestri e aerei, utilizzando il solver PANOC (Proximal Averaged Newton-type method for Optimal Control). L'approccio consente di affrontare in tempo reale la presenza di ostacoli con geometrie complesse come poligoni ed ellissi modellati tramite vincoli non lineari. Il controllore è stato implementato in linguaggio C con un'architettura compatibile con microcontrollori e validato su scenari sperimentali in laboratorio. I risultati mostrano come, anche con risorse computazionali limitate, sia possibile garantire traiettorie sicure e fluide, mantenendo una frequenza di aggiornamento superiore ai 20 Hz.
2. Un secondo contributo significativo è quello di Lindqvist et al. (2020) [30], che propongono l'uso dell'NMPC per veicoli aerei autonomi in ambienti dinamici, dove sono presenti ostacoli in movimento. Gli autori introducono un sistema predittivo che stima la traiettoria degli ostacoli e la integra nel problema di ottimizzazione del controllore. Anche in questo caso viene impiegato PANOC, con l'ausilio dell'ambiente OpEn (Optimization Engine) per la generazione automatica di codice ottimizzato. Il sistema è stato testato in real time su un drone dimostrando la capacità del controllore di adattarsi efficacemente a situazioni non previste e di pianificare manovre di evitamento robuste, con un tempo di campionamento di soli 50ms.
3. Infine Hermans et al. (2018) [31] propongono un approccio alternativo in cui l'NMPC affronta vincoli di collisione tramite un metodo a penalità. In particolare, gli ostacoli sono rappresentati come funzioni di costo aggiuntive, penalizzate progressivamente man mano che il veicolo si avvicina ai limiti critici. L'ottimizzazione viene risolta tramite PANOC e confrontata con altri solver classici (come IPOPT e SNOPT), evidenziando prestazioni computazionali competitive e maggiore adattabilità. I test vengono condotti su modelli cinematici complessi come il rimorchio articolato e dimostrano l'efficacia della penalizzazione nell'evitare blocchi locali e collisioni in ambienti ristretti.

Questi lavori evidenziano come l'NMPC, opportunamente formulato e risolto con metodi efficienti, rappresenti una soluzione avanzata e flessibile per il controllo di veicoli autonomi in scenari altamente dinamici. In particolare, la capacità di gestire vincoli non lineari, ostacoli in movimento e modelli complessi rende questa tecnica particolarmente promettente per future applicazioni, incluse quelle previste nel presente progetto.

4 Analisi dei KPI – Confronto tra PID e LMPC

Di seguito sono riportati e analizzati i quattro indicatori chiave di prestazione (KPI): errore laterale massimo, errore angolare massimo, RMS dell'errore laterale e RMS dell'errore angolare. Come detto nei paragrafi precedenti, le simulazioni sono state effettuate a diverse velocità comprese tra 10 km/h e 40 km/h, al fine di valutare l'evoluzione delle prestazioni al variare delle condizioni dinamiche.

4.1 Errore laterale massimo

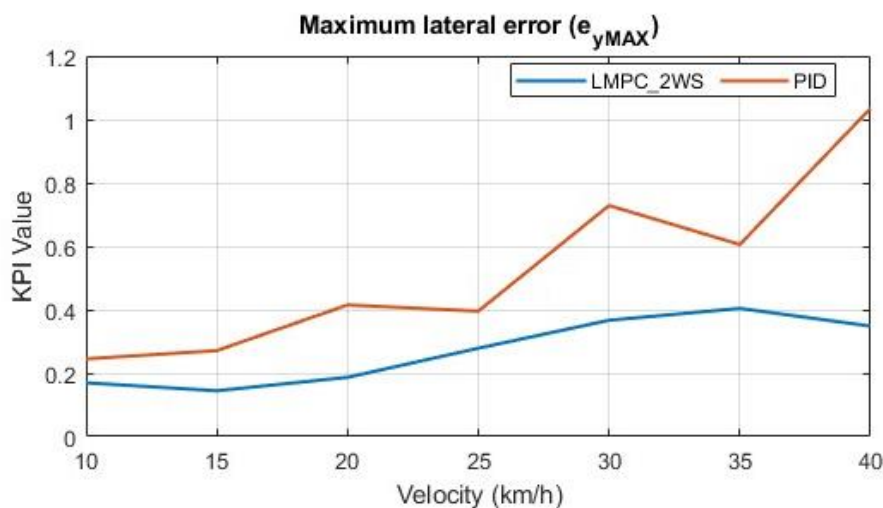


Figura 49: errore laterale massimo

Il grafico mostra l'andamento dell'errore laterale massimo, ovvero la deviazione del baricentro del veicolo rispetto alla traiettoria di riferimento. Questo indicatore è particolarmente rilevante in ambito di guida autonoma, in quanto riflette la capacità del controllore di mantenere il veicolo all'interno della carreggiata, aspetto cruciale nella manovra considerata.

L'analisi dei dati evidenzia come il controllore PID mostri una crescita marcata dell'errore massimo con l'aumentare della velocità. A 10 km/h, l'errore è contenuto entro 0.3 m, ma cresce progressivamente fino a superare di poco 1.0m a 40 km/h. Questo comportamento suggerisce una difficoltà del PID nel fronteggiare le maggiori inerzie e sollecitazioni dinamiche tipiche delle velocità più elevate.

Al contrario, il controllore LMPC mantiene un profilo di errore molto più contenuto e regolare. L'errore laterale massimo rimane stabile intorno ai 0.2–0.4m anche oltre i 30 km/h, segno che il controllore predittivo è in grado di anticipare le manovre e modulare le azioni di controllo in funzione dello scenario dinamico. La capacità dell'LMPC di apprendere dalle esecuzioni precedenti e di ottimizzare globalmente la traiettoria consente una significativa riduzione delle deviazioni critiche.

4.2 Errore angolare massimo

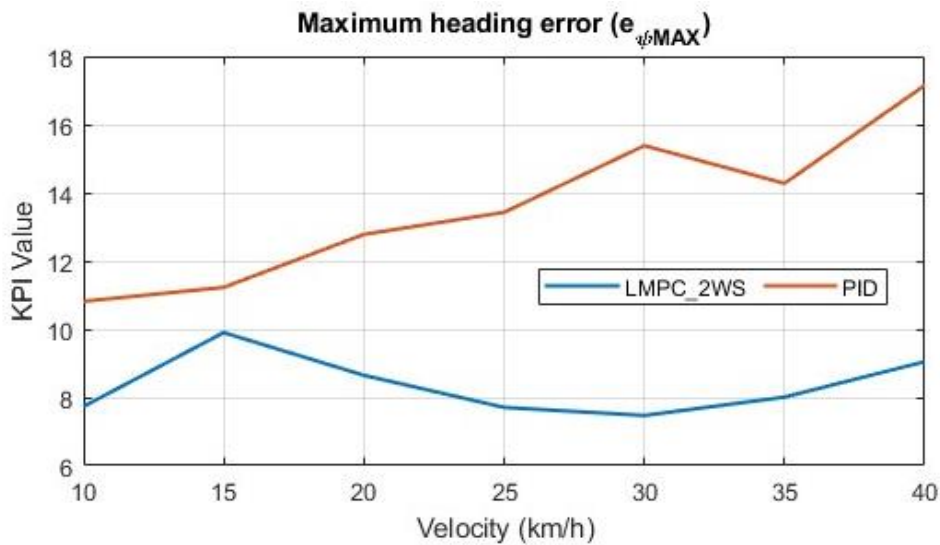


Figura 50: errore angolare massimo

Il grafico riguarda l'errore angolare massimo, cioè la massima differenza tra l'orientamento del veicolo e la direzione della traiettoria. Questo indicatore è particolarmente critico perché un elevato valore di heading error può compromettere la stabilità direzionale del veicolo e rendere difficile la correzione dell'errore laterale.

Il PID evidenzia un peggioramento significativo con l'aumentare della velocità: l'errore angolare massimo cresce da 11° a 17°, con un andamento quasi monotono. Questo testimonia l'incapacità del controllore di garantire un corretto allineamento direzionale quando il sistema è soggetto a dinamiche più rapide e complesse.

L'LMPC, invece, si mantiene su valori più contenuti, con un errore massimo che oscilla tra 7° e 10°, senza mostrare una tendenza fortemente crescente. Questo indica che il controllore è in grado di pianificare in anticipo le variazioni di sterzata, mantenendo il veicolo ben orientato lungo la traiettoria. La gestione efficace dell'angolo di imbardata è un ulteriore indicatore della superiorità dell'LMPC in contesti dinamicamente più critici.

4.3 RMS dell'errore laterale

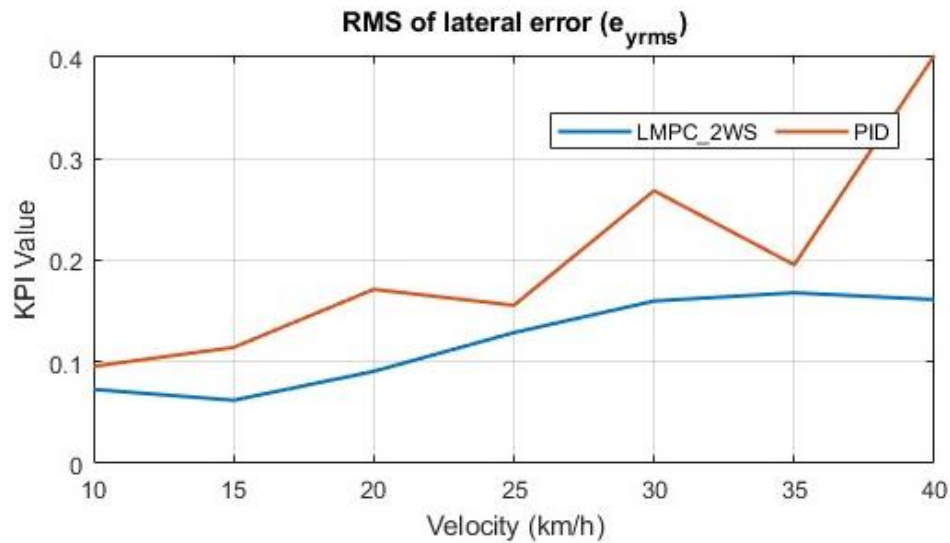


Figura 51: RMS dell'errore laterale

Il grafico rappresenta il valore quadratico medio dell'errore laterale, una misura più globale rispetto al massimo, in quanto tiene conto dell'andamento dell'errore su tutta la durata della simulazione. Esso fornisce una stima della precisione media con cui il veicolo segue la traiettoria.

Anche in questo caso, il PID mostra un andamento crescente con l'aumentare della velocità: il valore RMS passa da circa 0.12 m a oltre 0.35 m, con un'evidente variabilità nei risultati. Ciò suggerisce una mancanza di continuità e una maggiore sensibilità alle condizioni operative. L'oscillazione dei valori può anche essere indicativa di un effetto "nervoso" del controllore, il quale, non disponendo di informazioni predittive, reagisce in ritardo o in modo eccessivo agli scostamenti.

L'LMPC, invece, mostra una crescita più regolare e contenuta, mantenendosi sempre su valori inferiori rispetto al PID. I valori RMS restano al di sotto dei 0.2 m, a dimostrazione della sua efficacia nel seguire la traiettoria in modo fluido e costante. Tale comportamento è coerente con le aspettative di un controllore basato su ottimizzazione predittiva, che mira a minimizzare complessivamente il costo (lo scostamento) sull'intero orizzonte temporale.

4.4 RMS dell'errore angolare

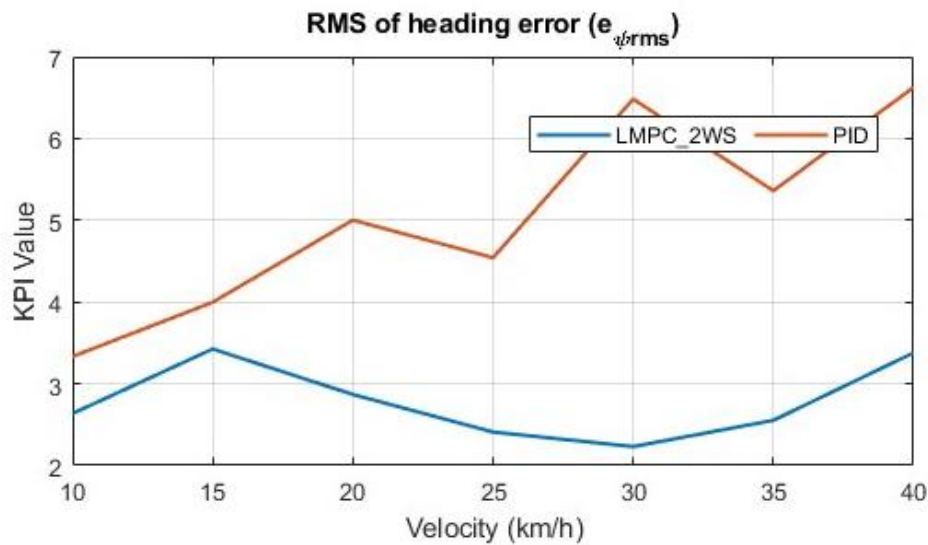


Figura 52: RMS dell'errore angolare

Infine, il quarto grafico riporta il valore quadratico medio dell'errore angolare, che rappresenta l'allineamento medio del veicolo rispetto alla traiettoria. Questo KPI permette di valutare la qualità complessiva del controllo direzionale, penalizzando oscillazioni prolungate o correzioni ripetute.

Il PID mostra valori inizialmente intorno a 4°, che salgono fino a oltre 6° a 40 km/h, con un andamento instabile che evidenzia una difficoltà nel mantenere una guida fluida e precisa. Il comportamento irregolare è sintomo di un sistema che reagisce con ritardo o con eccessiva intensità, compromettendo la stabilità e il comfort.

L'LMPC mostra invece un andamento decrescente fino a 30 km/h, segno di un controllo progressivamente più efficace, e una leggera risalita a 40 km/h. In ogni caso, i valori restano inferiori a quelli del PID, e in particolare tra 25 e 35 km/h si nota una maggiore efficienza nella gestione del controllo angolare. Questo conferma la qualità dell'approccio predittivo anche nella minimizzazione delle oscillazioni medie dell'assetto direzionale.

5 Conclusioni

Al termine del lavoro è possibile trarre alcune considerazioni significative sui risultati ottenuti e sulle prestazioni dei controllori. Le simulazioni condotte a diverse velocità hanno permesso di valutare il comportamento di ciascun controllore rispetto a una serie di indicatori chiave di performance (KPI), ovvero l'errore laterale massimo, l'errore angolare massimo e i rispettivi valori di RMSE.

Il controllore PID, pur nella sua semplicità strutturale e facilità di implementazione, ha mostrato evidenti limiti nel seguire con precisione il reference path, in particolar modo all'aumentare della velocità. Le simulazioni hanno evidenziato una crescita significativa degli errori, sia in ampiezza che in persistenza, con un controllo più reattivo che predittivo. Tale comportamento ha generato sollecitazioni non trascurabili sugli attuatori portando a una guida brusca e poco fluida, rendendo questo approccio poco adatto a scenari dinamici complessi. Rimane comunque un valido punto di partenza per applicazioni semplici o come benchmark per confronti più avanzati.

Il controllore LMPC ha rappresentato un notevole passo in avanti in termini di prestazioni. Grazie alla capacità di apprendere iterativamente e all'utilizzo di una formulazione predittiva, il controllore è riuscito a mantenere contenuti i valori di errore anche in condizioni dinamiche più sfidanti. In particolare, l'analisi dei KPI ha mostrato come l'LMPC sia in grado di ridurre drasticamente i valori di RMSE rispetto al PID, garantendo una traiettoria più stabile e aderente al percorso di riferimento. Inoltre, la capacità del controllore di anticipare le manovre necessarie, ha permesso una modulazione più fluida ed efficace dell'azione di controllo, riducendo così le sollecitazioni sugli attuatori e migliorando il comfort di guida.

Per quanto riguarda il controllore NMPC, come anticipato precedentemente, lo sviluppo del codice Matlab/Simulink nell'ambito della presente tesi è ancora in fase di completamento. Non è stato pertanto possibile presentare una serie completa di simulazioni alle diverse velocità come per gli altri due controllori. Tuttavia, sulla base della letteratura scientifica e degli studi citati, l'NMPC risulta essere potenzialmente la soluzione più avanzata tra quelle trattate. La possibilità di incorporare in modo esplicito i vincoli del sistema, di lavorare su un modello non lineare del veicolo e di ottimizzare il comportamento futuro a ogni passo temporale rappresentano caratteristiche ideali per un controllo più preciso in scenari complessi. L'attesa è che, una volta completato lo sviluppo, questo approccio possa garantire performance superiori sia in termini di aderenza al percorso di riferimento che nella gestione di vincoli e perturbazioni.

In sintesi, il confronto tra le strategie ha evidenziato come l'LMPC, a oggi, costituisca il miglior compromesso tra prestazioni, stabilità e complessità implementativa. Sebbene l'NMPC appaia il più promettente in ottica futura, resta da verificarne l'effettiva superiorità tramite un set completo di simulazioni. Il PID, infine, seppur largamente superato in termini

di efficacia, si conferma come base utile per comprendere le dinamiche fondamentali del controllo e per mettere in luce i vantaggi offerti da approcci più evoluti. Il lavoro svolto ha infine dimostrato come l'impiego di tecniche avanzate di controllo predittivo rappresenti una direzione concreta e necessaria per lo sviluppo di sistemi di guida autonoma sempre più sicuri, efficienti e adattivi, capaci di affrontare con successo le sfide poste dalla mobilità del futuro.

Bibliografia

- [1] Dickmanns, E. D. (2007). *Dynamic vision for perception and control of motion*. Springer
- [2] Buehler, M., Iagnemma, K., & Singh, S. (2009). *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*. Springer.
- [3] Waymo LLC. (2021). *Safety Report*. [Online] <https://waymo.com/safety>
- [4] Tesla Inc. (2023). *Autopilot and Full Self-Driving Capability*. [Online] <https://www.tesla.com/autopilot>
- [5] Burns, L. D., Jordan, W. C., & Scarborough, B. A. (2013). *Transforming personal mobility*. The Earth Institute, Columbia University
- [6] *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles* (SAE, 2021)
- [7] Paden et al. (2016) nell'articolo *A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles*
- [8] Levinson, J., et al. (2011). *Towards fully autonomous driving: Systems and algorithms*. IEEE IV Symposium.
- [9] Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic robotics*. MIT Press.
- [10] Paden, B., et al. (2016). *A survey of motion planning and control techniques for self-driving urban vehicles*. IEEE Transactions on Intelligent Vehicles
- [11] Koopman e Wagner (2017), nel lavoro *Autonomous vehicle safety: An interdisciplinary challenge*
- [12] Shalev-Shwartz et al. (2016), nell'articolo *On a Formal Model of Safe and Scalable Self-driving Cars*
- [13] cfr. *Review and performance evaluation of path tracking controllers of autonomous ground vehicles*, IET Intelligent Transport Systems, 2022
- [14] *Vehicle dynamics and control* di Rajesh Rajamani (2012)
- [15] *The Magic Formula Tire Model* di Hans B. Pacejka (2002)
- [16] *Feedback Systems: An Introduction for Scientists and Engineers* di Karl J. Åström e Richard M. Murray (2008)
- [17] Grüne e Pannek (2011) nel volume *Nonlinear Model Predictive Control: Theory and Algorithms*

- [18] *Deep Learning for Autonomous Vehicles* di Chen et al. (2020), pubblicato su *IEEE Transactions on Intelligent Transportation Systems*
- [19] Åström e Hägglund, nel testo *PID Controllers: Theory, Design, and Tuning* (1995)
- [20] *Model Predictive Control: Theory and Design* di Rawlings e Mayne (2009)
- [21] *Nonlinear Model Predictive Control: Theory and Algorithms* di Grüne e Pannek (2011)
- [22] Lucia et al. (2014) nell'articolo *Rapid development of nonlinear model predictive control using an auto-generated real-time iteration algorithm*
- [23] Zhe Wu, Panagiotis Christofides, Wenjie Wu, Yule Wang, Fadi Abdullah, Abdullah Alnajdi, Yash Kadakia, "A tutorial review of machine learning-based model predictive control methods", *Reviews in Chemical Engineering* (2025)
- [24] MathWorks, *MATLAB User's Guide*, The MathWorks, Inc., Natick, MA, USA
- [25] MathWorks, *Simulink User's Guide*, The MathWorks, Inc. Disponibile online
- [26] Gros, S., Zanon, M., & Diehl, M. (2020). *Numerical Optimal Control with ACADOS*
- [27] <https://docs.acados.org/citing/index.html>
- [28] Rajamani, R. (2012). *Vehicle Dynamics and Control*, 2nd ed., Springer.
- [29] ISO 3888-2-2011, "Passenger cars, test track for a severe lane-change manoeuvre – Part 2: obstacle avoidance"
- [30] A. Sathya, P. Sopasakis, R. Van Parys, A. Themelis, G. Pipeleers, and P. Patrinos, "Embedded nonlinear model predictive control for obstacle avoidance using PANOC," in *Proc. European Control Conference (ECC)*, Limassol, Cyprus, 2018, pp. 1523–1528, doi: 10.23919/ECC.2018.8550253.
- [31] B. Lindqvist, S. S. Mansouri, A.-A. Agha-Mohammadi, and G. Nikolakopoulos, "Nonlinear MPC for Collision Avoidance and Control of UAVs With Dynamic Obstacles," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6001–6008, 2020. Preprint: arXiv:2008.00792.
- [32] B. Hermans, P. Patrinos, and G. Pipeleers, "A Penalty Method Based Approach for Autonomous Navigation using Nonlinear MPC," *IFAC-PapersOnLine*, vol. 51, no. 20, 2018.