

UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA DELL'AUTOMAZIONE

**SVILUPPO DI UN MODELLO PER IL CONTROLLO
PREDITTIVO NONLINEARE DI UN VEICOLO
VIRTUALE**

RELATORE:

PROF. ALESSANDRO BEGHI
UNIVERSITÀ DI PADOVA

CORRELATORI:

PHD ING. MATTIA BRUSCHETTA
DOTT. ING. ENRICO PICOTTI
UNIVERSITÀ DI PADOVA

LAUREANDO:

LUIGI VIANELLO

PADOVA, 16 DICEMBRE 2019

ANNO ACCADEMICO 2018/2019

Sommario

Questa tesi presenta lo sviluppo di un modello di veicolo da utilizzare in uno schema di controllo predittivo non lineare in un contesto racing.

Dopo una premessa sui fondamenti teorici del controllo predittivo, la prima parte del lavoro descrive il modello di veicolo allo stato dell'arte, a cui segue una validazione del modello volta a studiarne le maggiori criticità.

Al fine di ottenere una migliore descrizione della dinamica, nella seconda parte del lavoro vengono proposte delle modifiche al modello, ma solo alcune vengono effettivamente implementate nel controllore. Il modello proposto risulta adeguato nella descrizione di un veicolo che esegue manovre al limite, con una complessità computazionale adatta ad un'implementazione *real-time*.

Infine vengono esposti i risultati delle simulazioni, così da dimostrare che le modifiche introdotte migliorino effettivamente le performance di controllo.

Indice

1	INTRODUZIONE	I
2	CONTROLLO PREDITTIVO	3
2.1	NMPC	4
2.1.1	Problema di controllo ottimo	4
2.1.2	Direct Multiple Shooting	5
2.1.3	Algoritmo di programmazione nonlineare	7
2.2	MATMPC	8
3	MODELLIZZAZIONE	II
3.1	Bicycle model	12
3.2	Modello di veicolo allo stato dell'arte	12
3.2.1	Forze longitudinali	15
3.2.2	Forze laterali	16
3.2.3	Forze normali	17
3.3	Validazione del modello	19
3.3.1	Accelerazioni	19
3.3.2	Drag Force	21
3.3.3	Forze longitudinali	21
3.3.4	Forze laterali	23
3.3.5	Forze normali	27
4	INTRODUZIONE DELLA DINAMICA COMBINATA DELLE FORZE LATERALI E LONGITUDINALI	29
4.1	Analisi della forza longitudinale	29
4.1.1	Validazioni preliminari	30
4.1.2	Modelli proposti per il calcolo della forza longitudinale	32
4.1.3	Approssimazione lineare della forza longitudinale	33
4.1.4	Calcolo della forza longitudinale con sistema in retroazione	36
4.2	Miglioramento della forza laterale	39
4.2.1	Mappa degli angoli di toe per lo slip laterale	40
4.2.2	Introduzione dello slip longitudinale	42
4.2.3	Formula di Pacejka completa per F_y	43
4.3	Ellissoide di aderenza	44
4.3.1	Calcolo dei coefficienti d'attrito	47

4.3.2	Calcolo dell'ellisse tramite la formula di Pacejka	48
4.3.3	Relazione tra le ellissi calcolate tramite Pacejka	49
4.3.4	Proiezione delle forze all'interno dell'ellissoide di aderenza	50
4.3.5	Validazione dell'ellissoide	51
5	SIMULAZIONE E RISULTATI	53
5.1	Implementazione in MATMPC	54
5.1.1	Modello	54
5.1.2	Funzione obiettivo	55
5.1.3	Vincoli	57
5.1.4	Parametri del solver	58
5.2	Cosimulazione in Simulink	59
5.3	Simulazioni	60
5.3.1	Confronto con MATMPC_MB	60
5.3.2	Controllo con riferimento di velocità aumentato	67
6	CONCLUSIONI	75
A	RIFORMULAZIONE SPAZIALE DELLA DINAMICA	77
	BIBLIOGRAFIA	80

1

Introduzione

Il settore automotive da tempo investe ingenti risorse sullo sviluppo di veicoli a guida autonoma, in particolare l'attenzione delle case automobilistiche è rivolta alla realizzazione di veicoli dotati di sistemi altamente tecnologici in grado di migliorare la sicurezza alla guida e di assistere il conducente in particolari situazioni di emergenza.

Si sta cercando infatti di ridurre il numero di incidenti stradali mediante regolamentazioni riguardo l'automazione di veicoli : le automobili di nuova produzione sono spesso dotate di sistemi ADAS, acronimo di "Advanced Driver Assistance Systems", cioè di quei dispositivi che incrementano il comfort e i livelli di sicurezza alla guida.

Al giorno d'oggi si possono trovare veicoli già equipaggiati con un gran numero di sistemi di controllo che contribuiscono ad avere una guida più sicura, come l'anti-bloccaggio delle ruote (ABS) o il controllo della trazione (TCS) e si possono trovare inoltre delle tecnologie di assistenza alla guida, come il Cruise Control, il Lane Keep Assist e il Park Assist che correggono i comandi di sterzo, accelerazione e frenata dati dal conducente. [1, 2]

In questi sistemi di controllo si possono distinguere tre livelli principali: percezione dell'ambiente, generazione del riferimento e controllo del veicolo lungo tale riferimento.

Questo progetto si concentra sull'ultimo livello di controllo ed in particolare sullo sviluppo di un modello da utilizzare all'interno di uno schema di controllo basato sul *Model Predictive Control* (MPC), il controllore riceve in input la traiettoria di riferimento e calcola gli ingressi al sistema (sterzo, accelerazione e frenata) richiesti per assicurarne la corretta percorrenza, supponendo perfettamente note le grandezze del sistema stesso.

Nel nostro caso si fa riferimento ad un contesto *racing*, per cui il modello che descrive il veicolo deve essere sufficientemente preciso da poter eseguire manovre al limite, ma più complesso è il modello maggiori sono i tempi di calcolo richiesti per risolvere il problema di controllo, perciò il modello scelto deve rappresentare il giusto compromesso tra complessità e accuratezza.

In questo lavoro verrà discusso il progetto di un *virtual driver* che utilizza uno schema di controllo predittivo basato su un modello non lineare (NMPC). Il controllore così ottenuto si è rivelato capace di funzionare con tempistiche *real time* e competere nelle situazioni estreme con le performance di driver commerciali.

Questo elaborato è suddiviso in sei sezioni.

Nel Capitolo 2 viene trattato il problema del controllo predittivo e l'utilizzo del pacchetto MATMPC per il calcolo della soluzione.

Nel Capitolo 3 viene invece descritto il modello del veicolo allo stato dell'arte per il controllo del contesto *racing*, dopo un breve cenno sui modelli di veicolo noti in letteratura, successivamente viene proposta una procedura di validazione del modello.

Nel Capitolo 4 viene proposta l'introduzione della dinamica combinata per le forze longitudinali e laterali degli pneumatici nel modello, con una contestuale analisi sulla possibile efficienza di tale implementazione.

Nel Capitolo 5 viene descritto l'ambiente di simulazione e vengono illustrate le simulazioni effettuate ed i risultati ottenuti.

Nel Capitolo 6 vengono infine discussi gli esiti delle simulazioni e menzionati i possibili sviluppi del progetto.

2

Controllo predittivo

Il *Model Predictive Control* (MPC) è una tecnica di controllo che utilizza il modello dell'impianto ed eventuali vincoli per predire l'evoluzione futura del sistema. La funzione principale del MPC è quella di determinare un ingresso di controllo in modo da minimizzare una certa funzione obiettivo che rappresenta il comportamento futuro del sistema su uno specifico orizzonte di predizione $[t_0; t_f]$. Tuttavia, solo il primo passo dell'ingresso ottimo è effettivamente applicato al sistema, dopodichè l'orizzonte di predizione viene spostato in avanti di un passo e il problema di minimizzazione viene risolto nuovamente usando i nuovi stati del sistema come condizioni iniziali. Questa procedura è ripetuta ad ogni istante di campionamento, perciò rappresenta una tecnica di controllo che risolve una sequenza di problemi di ottimizzazione *online* basati sull'ultima misurazione dello stato del sistema. Tale approccio consente in realtà di formulare il problema di controllo aggiungendo facilmente vincoli statici e dinamici sul valore assunto dalle variabili di controllo o dalle variabili di stato lungo tutto l'orizzonte predittivo; inoltre è possibile inserire nell'ottimizzazione vincoli sia di tipo hard, ovvero il cui rispetto deve essere garantito, sia di tipo soft, implementati come termine di penalizzazione all'interno della cifra di merito. Un diagramma a blocchi che mostra lo schema di funzionamento del MPC è riportato in Figura 2.1.

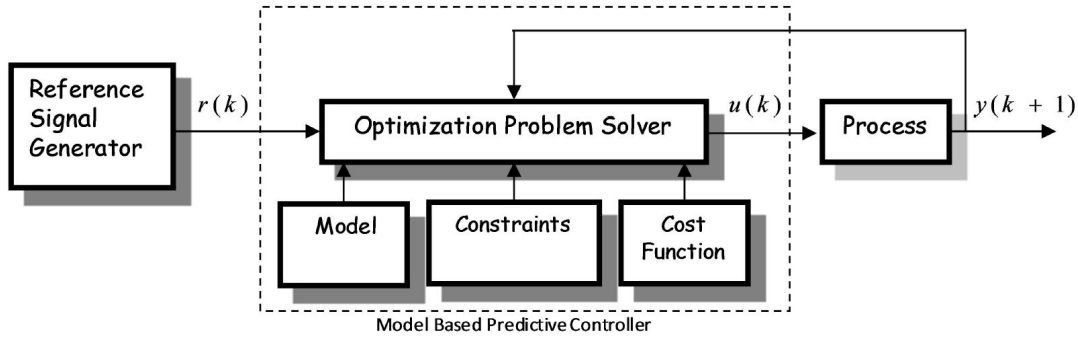


Figura 2.1: Schema di funzionamento del MPC

2.1 NMPC

Quando si considera un sistema descritto da un modello nonlineare si parla di *Nonlinear Model Predictive Control* (NMPC).

Questa tecnica permette di valutare in modo accurato i comportamenti non lineari del sistema e di conseguenza di migliorare la precisione del controllo.

Tuttavia una descrizione non lineare implica una maggiore complessità computazionale che può comportare l'impossibilità di utilizzare questo metodo in un contesto *real time*.

2.1.1 PROBLEMA DI CONTROLLO OTTIMO

Consideriamo un sistema nonlineare modellizzato in questa forma

$$\dot{x}(t) = f(t, x(t), u(t); p) \quad (2.1)$$

dove $x \in \mathbb{R}^{n_x}$ è il vettore degli stati, $u \in \mathbb{R}^{n_u}$ è il vettore degli ingressi di controllo e $p \in \mathbb{R}^{n_p}$ i parametri.

L'equazione 2.1 si riferisce ad un sistema di equazioni differenziali ordinarie e rappresenta la dinamica dell'impianto da controllare.

Data una condizione iniziale $x(t_0) = x_0$ e una traiettoria di controllo $u(t)$, l'esistenza di un'unica soluzione $x(t)$ su un certo intervallo $t \in [t_0, t_f]$ è assicurata dal teorema di esistenza di Picard, sotto la condizione che f sia lipschitziana in x ed u e continua in t [3]. Ad ogni

istante di campionamento il controllore deve risolvere il seguente problema di ottimizzazione

$$\begin{aligned}
\min_{x(t), u(t)} \quad & \tilde{J} = \int_{t_0}^{t_f} \phi(t, x(t), u(t); p) dt + \Phi(x(t_f)) \\
\text{s.t.} \quad & x(t_0) = x_0, \\
& \dot{x}(t) = f(t, x(t), u(t); p) \quad \forall t \in [t_0, t_f], \\
& r(t, x(t), u(t); p) \leq 0 \quad \forall t \in [t_0, t_f], \\
& l(t_f) \leq 0
\end{aligned} \tag{2.2}$$

dove ϕ e Φ sono le funzioni obiettivo, r i vincoli delle traiettorie, l le condizioni limite. Dato lo stato attuale x_0 può quindi essere calcolata una soluzione ottima per $(x(t), u(t))$ attraverso 2.2 nell'orizzonte di predizione $[t_0, t_f]$.

2.1.2 DIRECT MULTIPLE SHOOTING

Per risolvere i problemi di controllo ottimo ci sono vari metodi, tra i quali *Dynamic Programming, Indirect Methods* e *Direct Methods*, per questo progetto verrà usato un metodo diretto, il *direct multiple shooting*, che si è dimostrato molto efficace nella soluzione di problemi di NMPC [4]. Un metodo diretto è basato su una parametrizzazione a dimensione finita del problema a tempo continuo 2.2, nel nostro caso parametrizzato attraverso il *multiple shooting*.

Nel direct multiple shooting il periodo di predizione viene diviso in N intervalli di *shooting* definiti come $[t_k, t_{k+1}]$, $k = 0, 1, \dots, N - 1$ e risulta quindi scomposto in $N + 1$ punti $t_0 < t_1 < \dots < t_N = t_f$ di una griglia temporale.

La traiettoria di controllo viene quindi parametrizzata in questi intervalli tramite una rappresentazione costante a tratti come

$$u(t) = u_k \quad \forall t \in [t_k, t_{k+1}) \tag{2.3}$$

Anche lo stato $x(t)$ viene parametrizzato su N intervalli, e $N + 1$ punti di shooting s_k (con $k = 1, \dots, N$) sono introdotti come variabili di ottimizzazione aggiuntive. Ognuno di questi punti s_k è definito esattamente sui punti t_k della griglia temporale, e rappresenta la condizione iniziale del successivo problema ai valori iniziali sull'intervallo di shooting

$[t_k, t_{k+1})$:

$$\dot{x}_k(t) = f(t, x_k(t), u_k(t); p) \quad \forall t \in [t_k, t_{k+1}), \quad x_k(t_k) = s_k \quad (2.4)$$

e perciò i vincoli dinamici diventano vincoli di continuità

$$s_{k+1} = \Xi(t_k, s_k, u_k; p) \quad k = 0, 1, \dots, N-1 \quad (2.5)$$

dove $\Xi(\cdot)$ è un operatore di integrazione che risolve il problema ai valori iniziali 2.4 e restituisce la soluzione nel punto finale t_{k+1} . Allo stesso modo vengono parametrizzati i vincoli delle traiettorie tramite

$$r(s_k, u_k; p) \leq 0, \quad k = 0, 1, \dots, N-1. \quad (2.6)$$

Infine la funzione obiettivo \tilde{J} viene riformulata nel modo seguente

$$\sum_{k=0}^{N-1} \bar{J}_k = \sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} \phi(t_k, x_k(t), u_k(t); p) dt + \Phi(s_N) \quad (2.7)$$

che può essere approssimata con una somma discreta

$$\sum_{k=0}^{N-1} \bar{J}_k \cong \sum_{k=0}^{N-1} \phi(t_k, x_k(t), u_k(t); p) + \Phi(s_N) \quad (2.8)$$

Il problema di programmazione non lineare è quindi riformulato come

$$\begin{aligned} \min_{\mathbf{s}, \mathbf{u}} \quad & \sum_{k=0}^{N-1} \phi(t_k, s_k, u_k; p) + \Phi(s_N) \\ \text{s.t.} \quad & s_0 = x_0, \\ & s_{k+1} = \Xi(t_k, s_k, u_k; p) \quad k = 0, 1, \dots, N-1, \\ & r(s_k, u_k; p) \leq 0 \quad k = 0, 1, \dots, N-1, \\ & l(s_N) \leq 0 \end{aligned} \quad (2.9)$$

dove $\mathbf{s} = [s_0^T, s_1^T, \dots, s_N^T]^T$ e $\mathbf{u} = [u_0^T, u_1^T, \dots, u_{N-1}^T]^T$ sono rispettivamente le variabili discrete di stato e di ingresso.

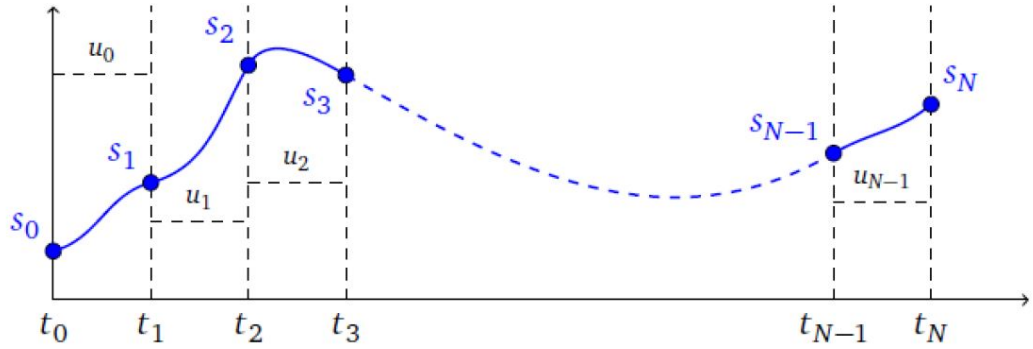


Figura 2.2: La traiettoria parametrizzata

2.1.3 ALGORITMO DI PROGRAMMAZIONE NONLINEARE

Consideriamo ora il problema 2.9 riscritto in forma standard come

$$\begin{aligned}
 \min_{\mathbf{z}} \quad & a(\mathbf{z}) \\
 \text{s.t.} \quad & b(\mathbf{z}) = 0, \\
 & c(\mathbf{z}) \leq 0
 \end{aligned} \tag{2.10}$$

dove

$$\begin{aligned}
 \mathbf{z} &= [z_0^T, z_1^T, \dots, z_{N-1}^T, s_N]^T \in \mathbb{R}^{n_z} \\
 z_k &= [s_k^T, u_k^T]^T \in \mathbb{R}^{n_x+n_u} \quad k = 0, 1, \dots, N-1
 \end{aligned} \tag{2.11}$$

è un vettore che comprende tutte le variabili di ottimizzazione, mentre le funzioni $b : \mathbb{R}^{n_z} \rightarrow \mathbb{R}^{n_b}$, $c : \mathbb{R}^{n_z} \rightarrow \mathbb{R}^{n_c}$ raccolgono tutti i vincoli, che sono dati da

$$b(\mathbf{z}) = \begin{bmatrix} x_0 - s_0 \\ \Xi(s_0, u_0) - s_1 \\ \vdots \\ \Xi(s_{N-1}, u_{N-1}) - s_N \end{bmatrix} \quad c(\mathbf{z}) = \begin{bmatrix} r(s_0, u_0) \\ \vdots \\ r(s_{N-1}, u_{N-1}) \\ l(s_N) \end{bmatrix} \tag{2.12}$$

Un modo possibile per risolvere un tale problema di programmazione nonlineare è attraverso la tecnica del *Sequential Quadratic Programming* (SQP) [5], in cui la soluzione

viene raggiunta iterativamente approssimando localmente la funzione obiettivo come quadratica e linearizzando i vincoli.

Tale strategia di ottimizzazione è stata implementata usando *qpOASES* [6], un pacchetto software open-source che implementa la risoluzione di problemi convessi di programmazione quadratica sfruttando le soluzioni passate come supposizione iniziale dello stato corrente. Il raggiungimento della soluzione ottima \mathbf{z}^* viene stabilito quando la condizione necessaria di ottimalità del primo ordine di Karush-Kuhn-Tucker (KKT) relativa al problema 2.10 è soddisfatta con approssimazione a scelta dell'utente.

L'algoritmo così illustrato può essere migliorato considerando che il problema di programmazione nonlineare non varia molto a due istanti di campionamento consecutivi (con una frequenza di campionamento adeguatamente alta) e inoltre una minore precisione della soluzione ottima può risultare sufficiente dato che spesso non è richiesta con accuratezza perfetta. Lo schema *Real-Time Iteration* (RTI) [4] sfrutta queste considerazioni e permette di ottenere un'implementazione molto rapida interrompendo la tecnica SQP alla prima iterazione e restituendo una soluzione subottima del problema.

2.2 MATMPC

Di recente sono stati sviluppati molti software open-source per la risoluzione di problemi MPC, ma non molti sono adatti a problemi di NMPC che siano allo stesso tempo facili da usare ed efficienti dal punto di vista computazionale. La difficoltà di una linearizzazione efficiente e la complessità dell'ottimizzazione non convessa rendono infatti particolarmente complicato adattare in modo semplice un algoritmo per MPC in un'implementazione valida per NMPC.

In questo progetto è stato utilizzato MATMPC, un pacchetto per la soluzione di problemi NMPC basato su MATLAB e sviluppato all'Università degli Studi di Padova da Yutao Chen [3]. Tra i vantaggi nell'utilizzare MATLAB come software di base vi sono principalmente la facilità di utilizzo, la presenza di molte routine di algebra lineare numerica già implementate e l'efficienza computazionale fornita dalle funzioni *MATLAB Executable* (MEX) che permettono di utilizzare codici in C o Fortran direttamente da MATLAB, come se fossero delle funzioni integrate. Lo schema di funzionamento MATMPC è illustrato in Figura 2.3.

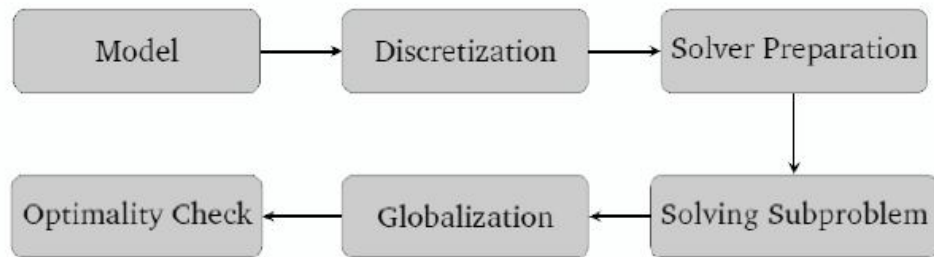


Figura 2.3: Struttura di MATMPC

Per il funzionamento di MATMPC viene richiesto un modello a tempo continuo, realizzato in linguaggio CasADi [7], che viene discretizzato attraverso la tecnica del *multiple shooting* descritta nel paragrafo 2.1.2 sfruttando l'integratore numerico *Runge-Kutta*. A questo punto il problema viene preparato per un risolutore specifico, in particolare per questa tesi viene costruito un *condensed QP problem* per il pacchetto qpOASES. Viene poi eseguita una procedura iterativa per arrivare ad una soluzione ottima dove viene applicato un algoritmo di *line search* per trovare un punto di minimo locale, valutando poi le condizioni di ottimalità del primo ordine di KKT in tale punto: più basso è il valore di KKT, più la soluzione trovata si avvicina a quella reale

3

Modellizzazione

Al giorno d'oggi sono stati sviluppati vari modelli di veicolo, diversificati in termini di complessità e in base ai gradi di libertà (GDL) considerati [8, 9].

Per gestire situazioni di guida al limite viene richiesto solitamente un modello più complesso che permetta al controllore di gestire ogni situazione, mentre nel caso di una guida più moderata dove velocità e accelerazioni sono più contenute non è necessaria una descrizione del modello così precisa.

Perciò la modellizzazione richiesta per un problema di NMPC è una questione che richiede di saper scegliere il giusto compromesso tra la precisione del modello e la complessità computazionale.

Nei capitoli seguenti verranno esaminati alcuni modelli di veicolo che si possono trovare in letteratura.

3.1 BICYCLE MODEL

Il modello più semplice e conosciuto è il *bicycle model*, il cui veicolo è rappresentato da sole due ruote (Figura 3.1).

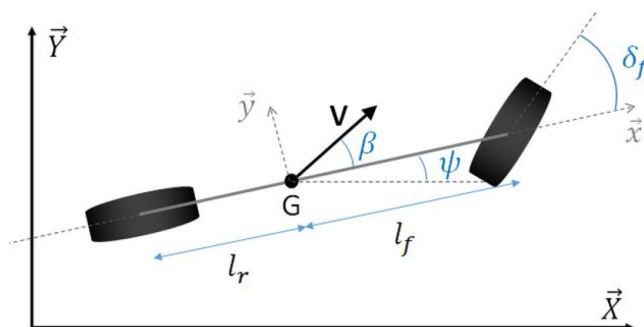


Figura 3.1: Schema del *bicycle model*

In questo modello la larghezza del veicolo viene considerata nulla e le due ruote anteriori e posteriori sono ammassate in un'unica ruota all'anteriore e una al posteriore, le sterzate vengono applicate solo alla ruota anteriore e non vengono considerati i movimenti verticali, la rotazione attorno l'asse longitudinale (*roll*) e quella attorno l'asse laterale (*pitch*), in questo modo vengono considerati solamente 3 gradi di libertà: lo spostamento longitudinale x , lo spostamento laterale y e la rotazione ψ attorno all'asse verticale (*yaw*).

Questo modello è stato usato in numerosi sistemi di controllo con buone prestazioni nonostante la sua semplicità, ciò nonostante presenta numerosi limiti ed incongruenze con il comportamento reale del veicolo soprattutto durante manovre che producono forti accelerazioni laterali.

Tali imprecisioni spesso vengono corrette tramite l'implementazione di algoritmi nei quali le forze laterali vengono ricalcolate dinamicamente in base alle misurazioni dell'accelerazione laterale del veicolo [10].

3.2 MODELLO DI VEICOLO ALLO STATO DELL'ARTE

In Figura 3.2 è riportato lo schema del modello di veicolo a quattro ruote, che è una naturale evoluzione del *bicycle model*.

Una buona riproduzione di un veicolo reale è data da un modello avente quattordici gradi di libertà: movimento longitudinale, laterale, verticale, di *roll*, *pitch* e *yaw* costituiscono i 6

GDL relativi al telaio, mentre movimenti verticali e rotazionali delle quattro ruote costituiscono gli ulteriori 8 GDL [11, 12].

In un contesto di controllo *real-time* il modello necessita delle semplificazioni per garantire tempi computazionali ridotti, è infatti comune in letteratura trovare descritto un modello di veicolo a quattro ruote con 3 gradi di libertà, caratterizzato dalla dinamica laterale, longitudinale e di *yaw*.

In particolare in questa tesi si fa riferimento al lavoro di tesi di Enrico Picotti dell'Università degli Studi di Padova, che ha sviluppato un veicolo virtuale controllato tramite MPC nonlineare [13], congiuntamente ai miglioramenti apportati al modello da parte di Michele Bevilacqua durante una tesi svolta sempre all'Università degli Studi di Padova [14].

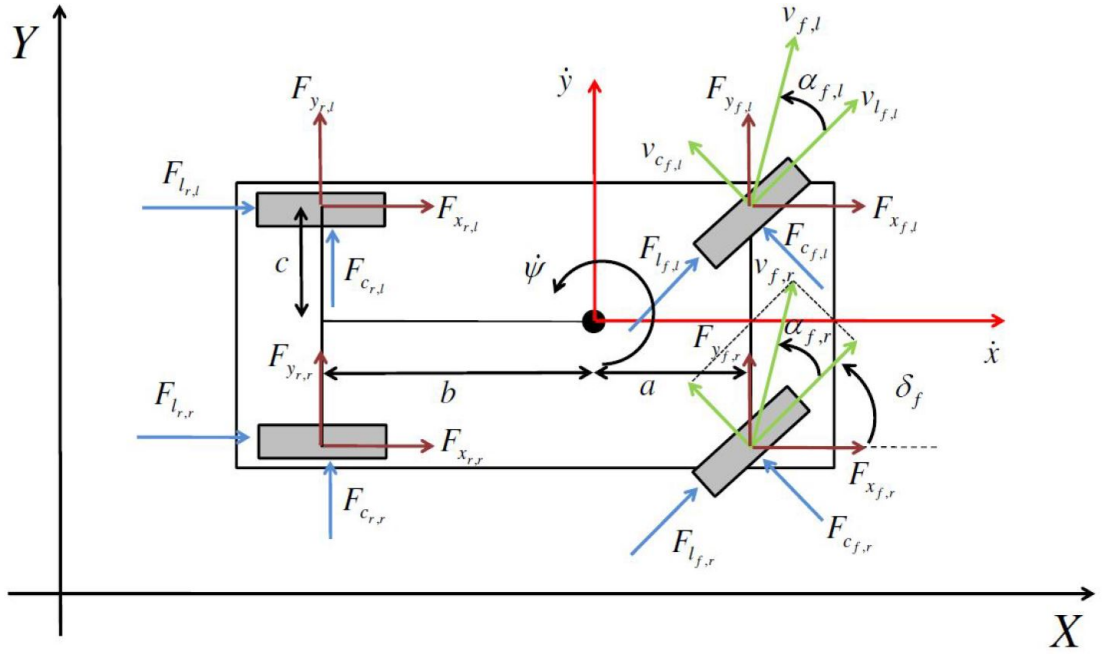


Figura 3.2: Schema del modello di veicolo a quattro ruote

Facendo riferimento alla Figura 3.2, viene ora descritta la dinamica del veicolo usata in un contesto *racing*, scritta in forma compatta come

$$\dot{\xi} = f(\xi(t), u(t)) \quad (3.1)$$

dove $\xi(t) \in \mathbb{R}^{n_x}$ è lo stato del sistema e $u(t) \in \mathbb{R}^{n_u}$ è l'input al sistema.

L'equazione 3.1 può essere derivata utilizzando le equazioni del moto del centro di massa del

veicolo che risultano

$$\begin{aligned}
\ddot{x} &= \dot{y}\dot{\psi} + \frac{1}{m} \left(\sum_{i,j} F_{x_{i,j}} - F_x^d \right) \\
\ddot{y} &= -\dot{x}\dot{\psi} + \frac{1}{m} \left(\sum_{i,j} F_{y_{i,j}} \right) \\
\ddot{\psi} &= \frac{1}{I_z} \left[a \left(\sum_j F_{y_{f,j}} \right) - b \left(\sum_j F_{y_{r,j}} \right) + c \left(\sum_i F_{x_{i,r}} - \sum_i F_{x_{i,l}} \right) \right]
\end{aligned} \tag{3.2}$$

dove $i \in \{f, r\}$ indica le ruote anteriori o posteriori, $j \in \{l, r\}$ indica le ruote di sinistra o di destra, a, b, c indicano rispettivamente la distanza dal centro di massa delle ruote anteriori, la distanza dal centro di massa di quelle posteriori e la distanza delle ruote dall'asse longitudinale, $F_{\{x,y\}_{\{i,j\}}}$ sono le forze laterali e longitudinali sulle quattro ruote nel sistema di riferimento del veicolo, F_x^d è la forza di resistenza aerodinamica (*drag force*) longitudinale e m e I_z sono rispettivamente la massa e l'inerzia del veicolo .

Per il calcolo di F_x^d è stata considerata la seguente formula

$$F_x^d = \frac{1}{2} \rho C_d A \dot{x}^2 \tag{3.3}$$

dove ρ è la densità del fluido in cui si muove il corpo, C_d è il coefficiente di *drag*, un valore adimensionale che incorpora informazioni sia sull'attrito superficiale che sulla forma del corpo, A è l'area della sezione frontale e \dot{x} è la velocità relativa di movimento tra il corpo e il fluido.

Per il calcolo delle forze laterali e longitudinali nel sistema di riferimento del veicolo si assume che lo sterzo possa agire solo sulle ruote anteriori ($\delta_{r,j} = 0$) e che queste ruotino dello stesso angolo ($\delta_{f,l} = \delta_{f,r} = \delta_f$). Tramite le relazioni geometriche di proiezione si ottiene

$$\begin{aligned}
F_{x_{f,j}} &= F_{l_{f,j}} \cos(\delta_f) - F_{c_{f,j}} \sin(\delta_f) \\
F_{y_{f,j}} &= F_{l_{f,j}} \sin(\delta_f) + F_{c_{f,j}} \cos(\delta_f) \\
F_{x_{r,j}} &= F_{l_{r,j}} \\
F_{y_{r,j}} &= F_{c_{r,j}}
\end{aligned} \tag{3.4}$$

dove $F_{l_{i,j}}$ e $F_{c_{i,j}}$ sono rispettivamente la forza longitudinale e la forza laterale (o di *cornering*) nel sistema di riferimento solidale con la ruota.

Nei prossimi paragrafi verranno descritti i metodi con cui vengono calcolate le forze longitudinali, laterali e normali.

3.2.1 FORZE LONGITUDINALI

Le forze longitudinali $F_{l_{i,j}}$ sono calcolate tramite la seguente relazione

$$F_{l_{i,j}} = \frac{\tau_{i,j}}{r_w} \cdot \mu \quad (3.5)$$

dove μ è il coefficiente d'attrito, r_w è il raggio nominale della ruota e $\tau_{i,j}$ è la sommatoria dei torques di accelerazione (T_{drive}^w) e di frenata (T_{brk}^w) applicati alla ruota $\{i, j\}$

$$\tau_{i,j} = T_{drive_{i,j}}^w + T_{brk_{i,j}}^w \quad (3.6)$$

Assumendo un veicolo con differenziali aperti ¹, il torque di accelerazione viene trasmesso dal motore attraverso la seguente relazione

$$T_{drive_{i,j}}^w = \frac{1}{4} \cdot T^w \cdot i_x \cdot i_{diff} \quad (3.7)$$

dove i_x è il rapporto trasmesso dal cambio di marcia, i_{diff} è il rapporto del differenziale e T^w è il torque trasmesso dal motore.

La coppia trasmessa effettivamente alle ruote è data da una combinazione convessa tra la coppia massima erogabile $T_+^w(\omega_w)$ e quella massima frenante $T_-^w(\omega_w)$:

$$T^w = \beta_t \cdot T_+^w(\omega_w) + (1 - \beta_t) \cdot T_-^w(\omega_w) \quad (3.8)$$

dove $\beta_t \in \{0, 1\}$ è il *throttle demands*, ovvero la richiesta di accelerazione da parte del controllore.

Le coppie $T_+^w(\omega_w)$ e $T_-^w(\omega_w)$ vengono calcolate dal controllore attraverso una funzione ricavata dall'interpolazione della curva di coppia del motore in funzione della velocità angolare delle ruote ω_w [14], la quale è ottenuta da:

$$\omega_w = \frac{\dot{x}}{r_w} \cdot \frac{360}{2\pi} \quad (3.9)$$

¹ Il differenziale aperto ripartisce equamente la coppia motrice tra le quattro ruote

Per quanto riguarda il sistema frenante, si considera un impianto composto da freni a disco su tutte le ruote: all'azionamento del freno viene generata una pressione all'interno di un pistone che porta a comprimere sul disco le due pastiglie dei freni.

La coppia generata dal sistema frenante è divisa tra ruote anteriori e posteriori ed è ottenuta da:

$$T_{brk_F}^w(\beta) = \beta_b \cdot 2A_{pist} \cdot \mu_{brk} \cdot r_{pist_F} \cdot bias_F \cdot K_{brk} \quad (3.10)$$

$$T_{brk_R}^w(\beta) = \beta_b \cdot 2A_{pist} \cdot \mu_{brk} \cdot r_{pist_F} \cdot (1 - bias_F) \cdot K_{brk} \quad (3.11)$$

dove β_b è l'input di brake, A_{pist} è l'area del pistone che muove le due pastiglie dei freni, μ_{brk} è il coefficiente d'attrito tra pastiglia e disco, $r_{pist_{F,R}}$ è il raggio del pistone, $bias_F$ è il coefficiente di ripartizione della coppia frenante tra ruote anteriori e posteriori, K_{brk} è un coefficiente di gain che mette in relazione la pressione esercitata sul pedale del freno a quella trasferita al cilindro.

Le forze longitudinali sono definite perciò a partire da velocità longitudinale \dot{x} , input di *throttle* e *brake* β_t, β_b e dal coefficiente d'attrito μ :

$$F_{l_{i,j}} = f_l(\dot{x}, \beta_t, \beta_b, \mu) \quad (3.12)$$

3.2.2 FORZE LATERALI

Per il calcolo delle forze laterali $F_{c_{i,j}}$ è stato considerato il modello di pneumatico di Pacejka disaccoppiato [15], dipendente da forza normale $F_{z_{i,j}}$, angoli di slittamento laterale $\alpha_{i,j}$ e coefficiente d'attrito μ :

$$F_{c_{i,j}} = f_c(\alpha_{i,j}, F_{z_{i,j}}, \mu) \quad (3.13)$$

Nello specifico, queste forze sono calcolate attraverso la *Pacejka magic formula* :

$$F_{c_{i,j}} = D \sin(C \arctan(B \alpha_{i,j} - E (B \alpha_{i,j} - \arctan(B \alpha_{i,j})))) \mu \quad (3.14)$$

dove B è lo *stiffness factor*, C è lo *shape factor*, D è il *peak factor*, E è il *curvature factor*. Questi termini sono ricavati a partire da specifici parametri basati sul modello di pneumatico *PAC2002Tire Model* e la forza normale $F_{z_{i,j}}$.

Gli angoli di slittamento laterale $\alpha_{i,j}$ (o semplicemente angoli di slip) usati all'interno dell'eq.

3.14 sono gli angoli che si formano tra la direzione longitudinale della gomma e la direzione reale del moto e vengono calcolati attraverso

$$\alpha_{i,j} = \arctan \left(\frac{v_{c_{i,j}}}{v_{l_{i,j}}} \right) \quad (3.15)$$

dove $v_{c_{i,j}}$ e $v_{l_{i,j}}$ sono rispettivamente la velocità laterale e longitudinale della ruota, espresse nel sistema di riferimento della ruota stessa. L'angolo di slittamento complessivo invece è definito come

$$\alpha = \arctan \left(\frac{\dot{y}}{\dot{x}} \right) \quad (3.16)$$

Le velocità $v_{c_{i,j}}$ e $v_{l_{i,j}}$ sono calcolate con le seguenti formule geometriche di proiezione

$$\begin{aligned} v_{c_{f,j}} &= v_{y_{f,j}} \cos(\delta_f) - v_{x_{f,j}} \sin(\delta_f) \\ v_{c_{r,j}} &= v_{y_{r,j}} \\ v_{l_{f,j}} &= v_{y_{f,j}} \sin(\delta_f) + v_{x_{f,j}} \cos(\delta_f) \\ v_{l_{r,j}} &= v_{x_{r,j}} \end{aligned} \quad (3.17)$$

dove $v_{y_{i,j}}$ e $v_{x_{i,j}}$ sono rispettivamente la velocità laterale e longitudinale della ruota nel sistema di riferimento del veicolo, calcolate a partire dalla velocità \dot{x} , \dot{y} , $\dot{\psi}$ del veicolo secondo le equazioni

$$\begin{aligned} v_{y_{f,j}} &= \dot{y} + a\dot{\psi} \\ v_{y_{r,j}} &= \dot{y} - b\dot{\psi} \\ v_{x_{f,j}} &= \dot{x} - c\dot{\psi} \\ v_{x_{r,j}} &= \dot{x} + c\dot{\psi} \end{aligned} \quad (3.18)$$

3.2.3 FORZE NORMALI

Per quanto riguarda le forze normali è stato introdotto nel modello il trasferimento di carico sia laterale che longitudinale [14], in modo da poter calcolare ad ogni istante la forza normale

$F_{z_{i,j}}$ agente su ogni ruota.

La descrizione della forza normale agente su ogni ruota è

$$\begin{aligned}
F_{z_{f,l}} &= F_{z_{f,l}}^0 + \Delta_{lon}^z - \Delta_{lat}^z \\
F_{z_{f,r}} &= F_{z_{f,r}}^0 + \Delta_{lon}^z + \Delta_{lat}^z \\
F_{z_{r,l}} &= F_{z_{r,l}}^0 - \Delta_{lon}^z - \Delta_{lat}^z \\
F_{z_{r,r}} &= F_{z_{r,r}}^0 - \Delta_{lon}^z + \Delta_{lat}^z
\end{aligned} \tag{3.19}$$

dove $F_{z_{i,i}}^0$ è il peso statico relativo alla ruota $\{i, j\}$, mentre Δ_{lon}^z e Δ_{lat}^z rappresentano rispettivamente il trasferimento di carico longitudinale e laterale. In particolare, le forze normali a riposo agenti sulle ruote sono definite come segue:

$$F_{z_{f,j}}^0 = \frac{b m g}{2(a+b)} \qquad F_{z_{r,j}}^0 = \frac{a m g}{2(a+b)} \tag{3.20}$$

invece il trasferimento di carico longitudinale Δ_{lon}^z e quello laterale Δ_{lat}^z vengono calcolati a partire dalla coppia di *pitch* T^p e da quella di *roll* T^r :

$$\begin{aligned}
\Delta_{lon}^z &= \frac{T^p}{2(a+b)} \\
\Delta_{lat}^z &= \frac{T^r}{4c}
\end{aligned} \tag{3.21}$$

La coppia di *pitch* T^p viene esercitata attorno all'asse laterale, mentre la coppia di *roll* T^r è esercitata attorno all'asse longitudinale, ed entrambe sono indotte dalle forze agenti sulle ruote, che per un veicolo avente altezza h del centro di massa risultano

$$\begin{aligned}
T^p &= - (F_{x_{f,l}} + F_{x_{f,r}} + F_{x_{r,l}} + F_{x_{r,r}}) h \\
T^r &= (F_{y_{f,l}} + F_{y_{f,r}} + F_{y_{r,l}} + F_{y_{r,r}}) h
\end{aligned} \tag{3.22}$$

Si noti che questa formulazione presenta un loop algebrico, poichè le forze laterali $F_{y_{i,j}}$ dipendono dalle forze normali secondo l'eq. 3.14 di Pacejka.

Perciò sono stati introdotti due nuovi stati $\tilde{\Delta}_{lon}^z$ e $\tilde{\Delta}_{lat}^z$ per il trasferimento di carico longitu-

dinale e laterale, descritti dalle seguenti equazioni differenziali del primo ordine:

$$\begin{aligned}\dot{\tilde{\Delta}}_{lon}^z &= \frac{1}{\tau_{LT}} \left(\Delta_{lon}^z - \tilde{\Delta}_{lon}^z \right) \\ \dot{\tilde{\Delta}}_{lat}^z &= \frac{1}{\tau_{LT}} \left(\Delta_{lat}^z - \tilde{\Delta}_{lat}^z \right)\end{aligned}\tag{3.23}$$

dove τ_{LT} è un'opportuna costante di tempo.

3.3 VALIDAZIONE DEL MODELLO

Viene proposta ora una procedura volta a validare le equazioni descritte nella sezione 3.2. La validazione permette di capire se il modello utilizzato risulta adeguato per una specifica applicazione e, dal momento che la precisione del modello incide sull'accuratezza delle predizioni generate, risulta di particolare rilievo in un contesto MPC. Inoltre, la validazione consente di mettere in evidenza gli aspetti più critici del nostro modello, fornendo così delle analisi su cui basarsi per il perfezionamento del modello.

In questa tesi viene considerato un veicolo virtuale che percorre un giro di un tracciato grazie ad un controllore basato su NMPC; viene utilizzato un profilo di velocità che porta il veicolo a eseguire manovre al limite, portandolo così a lavorare in zone fortemente nonlineari che richiedono una conoscenza più accurata del modello.

Tale simulazione è stata resa disponibile da Enrico Picotti, che ha implementato quanto appena descritto nel suo lavoro di tesi magistrale tramite l'utilizzo di *VI-Car RealTime* (VI-CRT), software che fa parte del pacchetto *VI-Grade*, lavoro che è stato poi migliorato da Michele Bevilacqua [13, 14].

Dai risultati della simulazione verranno estratti una serie di dati che saranno utilizzati per il calcolo diretto delle varie equazioni che descrivono la dinamica. I risultati ottenuti vengono poi confrontati con i corrispondenti valori generati dalla simulazione, che è una riproduzione molto fedele del comportamento di un veicolo reale.

3.3.1 ACCELERAZIONI

Le accelerazioni risultanti calcolate secondo l'eq. 3.2 sono mostrate in Figura 3.3 (in arancione), dove vengono confrontate con il valore calcolato dalla simulazione di CRT (in blu).

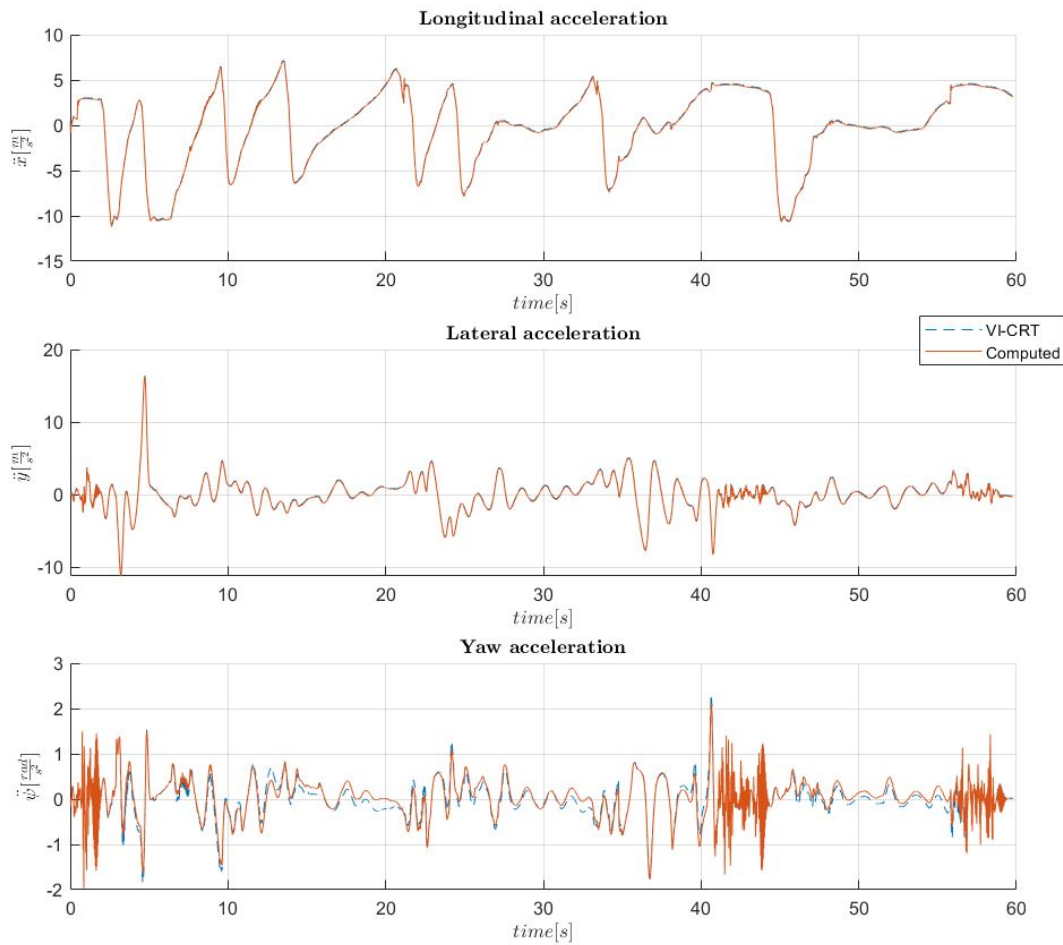


Figura 3.3: Calcolo delle accelerazioni

In Tabella 3.1 sono riportati gli errori massimo e medio dell'accelerazione di *yaw*.

	e_{max}	$e_{max}(\%)$	e_{mean}	$e_{mean}(\%)$
$\ddot{\psi}$	$0.54 \left[\frac{rad}{s^2} \right]$	183.93	$0.2 \left[\frac{rad}{s^2} \right]$	68.52

Tabella 3.1: Errore massimo e medio nel calcolo dell'accelerazione di *yaw*

L'errore percentuale $e\%$ viene calcolato facendo riferimento al valore medio assunto da una determinata accelerazione, ossia:

$$e\% = \frac{e}{X_m} \cdot 100 \quad (3.24)$$

Le formule usate per il calcolo dell'accelerazione longitudinale e laterale sono corrette, mentre risulta esserci un'evidente inadeguatezza dell'equazione per il calcolo dell'accelerazione di *yaw*, poichè non prende in considerazione lo spostamento del centro di gravità del veicolo quando questo è in moto ed il momento generato da ogni gomma lungo il suo asse verticale (chiamato *aligning moment*).

3.3.2 DRAG FORCE

I risultati del calcolo della forza di resistenza aerodinamica sono riportati in Figura 3.4. Il calcolo è molto preciso e gli errori valutati risultano essere $e_{mean}\% = 0.54N$ ed $e_{max}\% = 2.34N$.

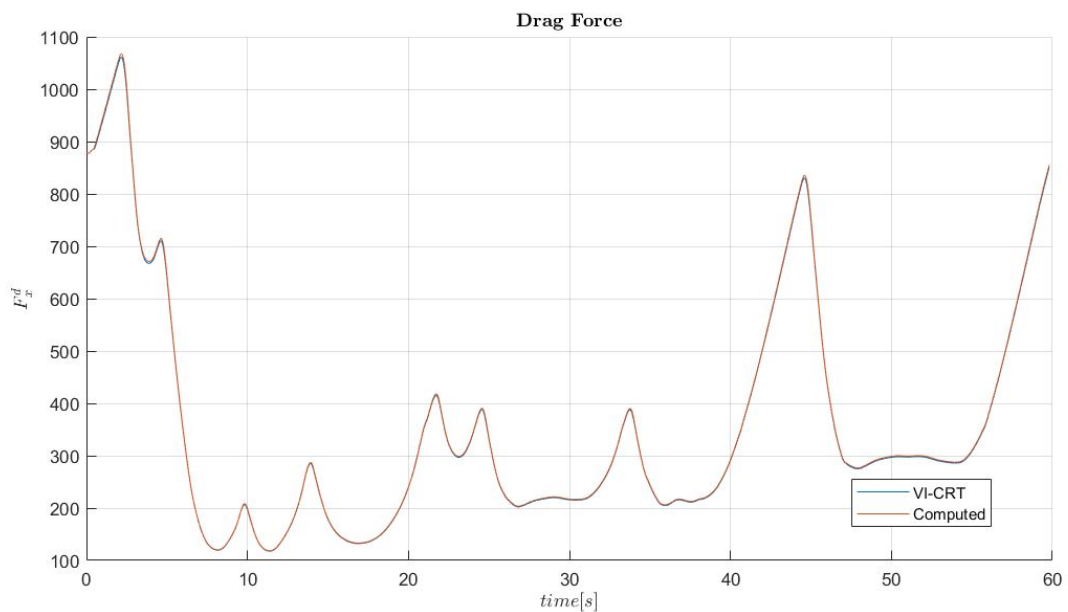


Figura 3.4: Calcolo della forza di resistenza aerodinamica

3.3.3 FORZE LONGITUDINALI

Le forze longitudinali calcolate secondo l'eq. 3.12 sono mostrate in Figura 3.5.

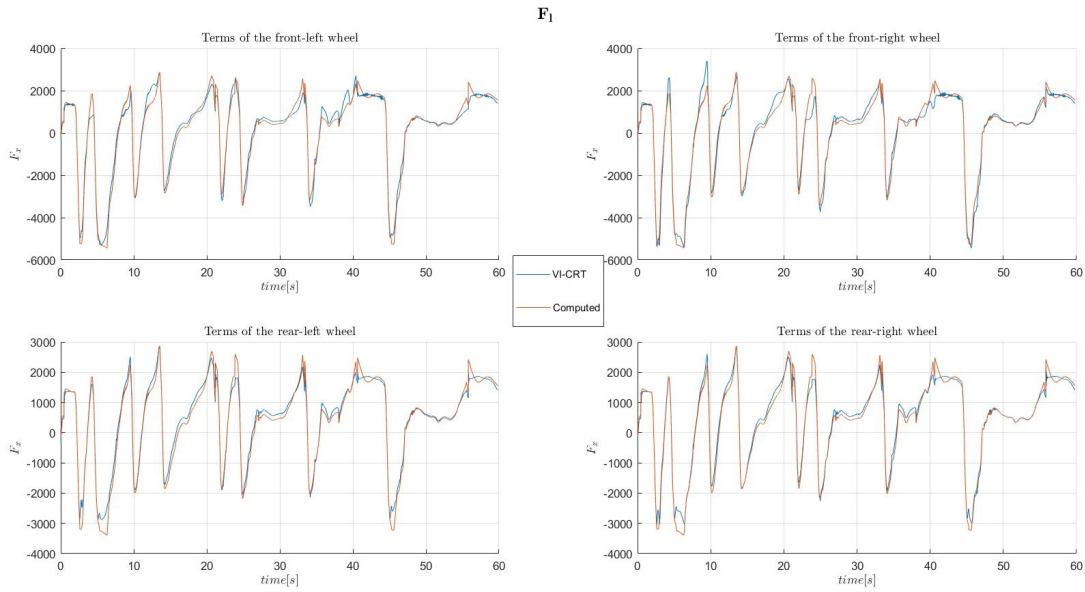


Figura 3.5: Calcolo delle forze longitudinali (nel sistema di riferimento della ruota stessa)

In Tabella 3.2 sono riportati gli errori massimo e medio delle forze longitudinali calcolate, suddivisi tra ruote anteriori e posteriori.

	$e_{max}[N]$	$e_{max}(\%)$	$e_{mean}[N]$	$e_{mean}(\%)$
$F_{x_{f,j}}$	1275.9	86.37	215.23	14.98
$F_{x_{r,j}}$	1260.2	100.34	154.47	12.63

Tabella 3.2: Errore massimo e medio nel calcolo delle forze longitudinali

Si può notare come le forze longitudinali abbiano lo stesso andamento di quelle di CRT, ma siano presenti alcuni picchi di errore significativi.

Tra i principali miglioramenti che potrebbero essere apportati per migliorare il calcolo della forza longitudinale vi sono:

- introduzione della *rolling resistance* tra i termini dell'eq. 3.6: l'assenza di tale termine comporta lievi sovrastime nei punti in cui si ha una forza longitudinale costante;
- prendere in considerazione il raggio reale della ruota R_l (*loaded radius*) anziché il raggio nominale r_w , che influirebbe in particolare su alcuni picchi di forza in frenata, quando le gomme sono sottoposte ad alti valori di forze normali;

- migliorare la funzione per il calcolo di $T_+^w(\omega_w)$ e $T_-^w(\omega_w)$, l'attuale approssimazione di queste funzioni comporta discostamenti in molti punti in cui abbiamo una forza positiva;

Un'alternativa sarebbe quella di introdurre la formula di Pacejka anche per la forza longitudinale, tale tecnica richiede il calcolo dello slip longitudinale, che ha una dinamica molto veloce e richiede quindi l'integrazione a frequenze troppo elevate per l'algoritmo di controllo.

3.3.4 FORZE LATERALI

Le forze laterali calcolate secondo l'eq. di Pacejka 3.13 sono derivate dapprima considerando il valore reale di ogni singola componente dell'eq. di Pacejka (Figura 3.6), in seguito vengono determinate considerando i valori di $\alpha_{i,j}$, $v_{c_{i,j}}$ e $v_{l_{i,j}}$ calcolati secondo le eq. 3.15 e 3.17 (Figura 3.7).

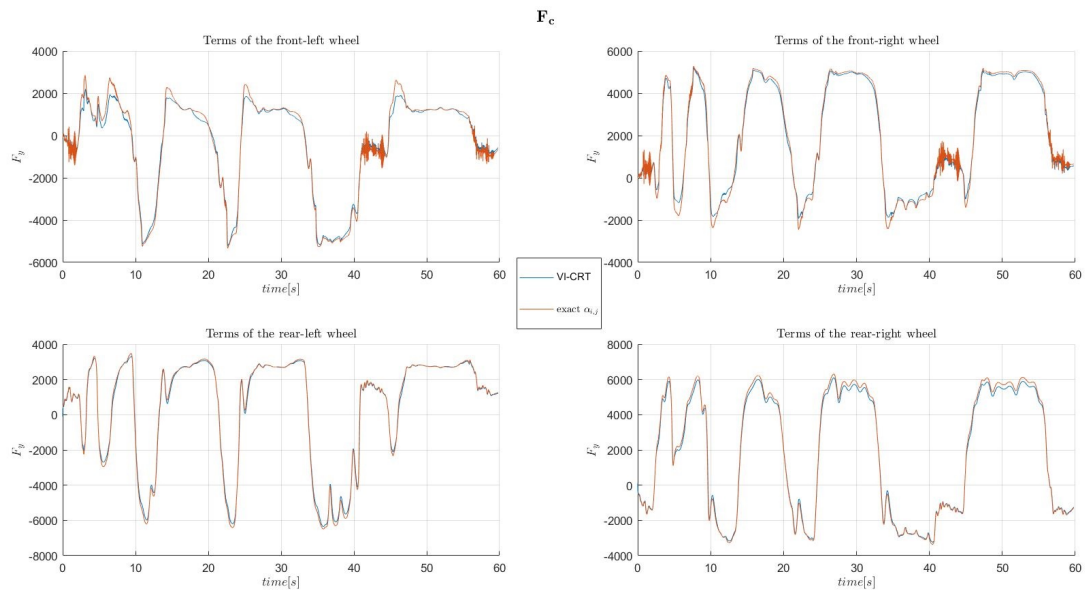


Figura 3.6: Calcolo delle forze laterali considerando $\alpha_{i,j}$ reale

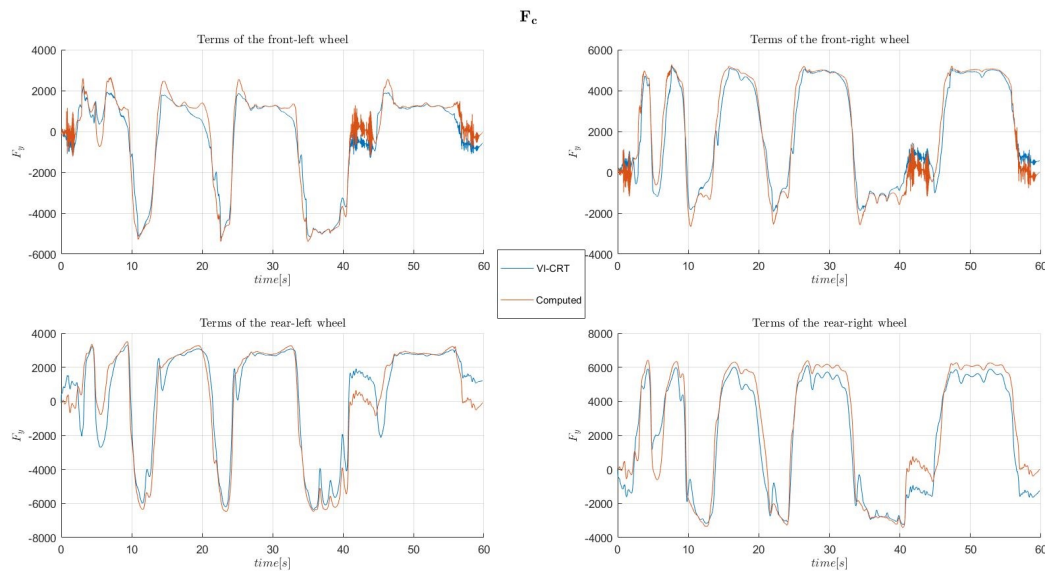


Figura 3.7: Calcolo delle forze laterali (nel sistema di riferimento della ruota stessa)

Dalla Figura 3.6 possiamo vedere come l'eq. di Pacejka disaccoppiata approssima molto bene il comportamento della forza laterale, ci sono però alcune piccole differenze e alcune sovrastime importanti in prossimità di molti picchi che sono da attribuire principalmente ai seguenti fattori:

1. la formula di Pacejka disaccoppiata (eq. 3.14) non tiene conto della dinamica longitudinale della gomma, in particolare del suo slittamento longitudinale;
2. non sono stati considerati gli angoli di *camber* $\gamma_{i,j}$ delle ruote ²

Se consideriamo invece la Figura 3.7 notiamo che il calcolo delle forze laterali diventa molto impreciso, in particolare abbiamo delle situazioni in cui la forza laterale stimata è negativa quando invece quella reale è positiva.

La maggiore causa di tale imprecisioni è dovuta dall'errore di stima delle velocità laterali, come si può vedere dalla Figura 3.8, le quali servono per calcolare gli slip laterali (eq. 3.15, Figura 3.9).

²L'angolo di *camber* è quello formato tra il piano che contiene la ruota e l'asse verticale, e varia in base alla forza normale agente sul giunto della sospensione

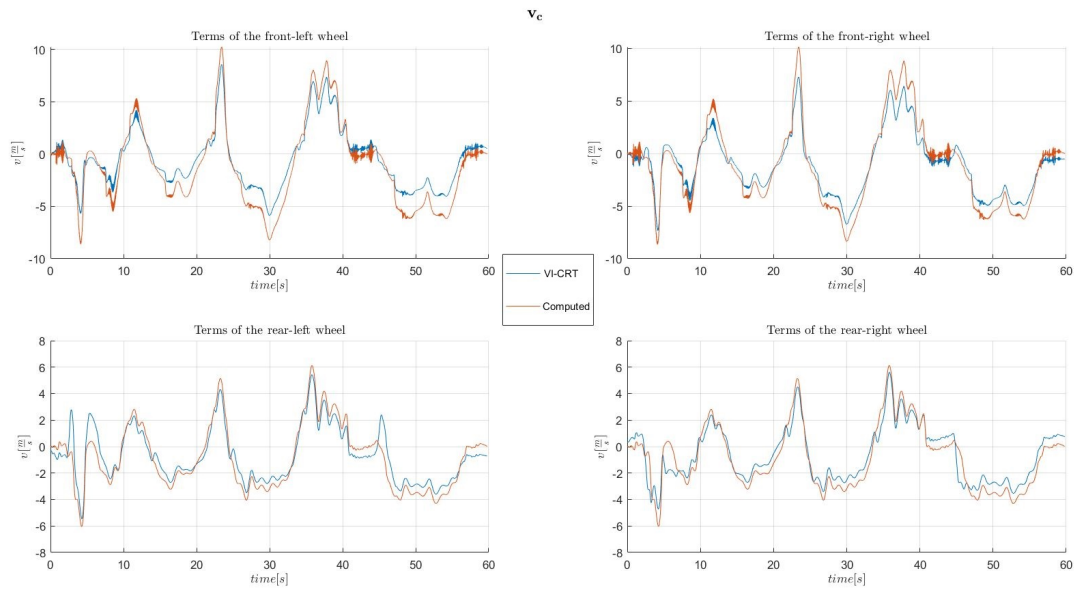


Figura 3.8: Calcolo delle velocità laterali delle ruote

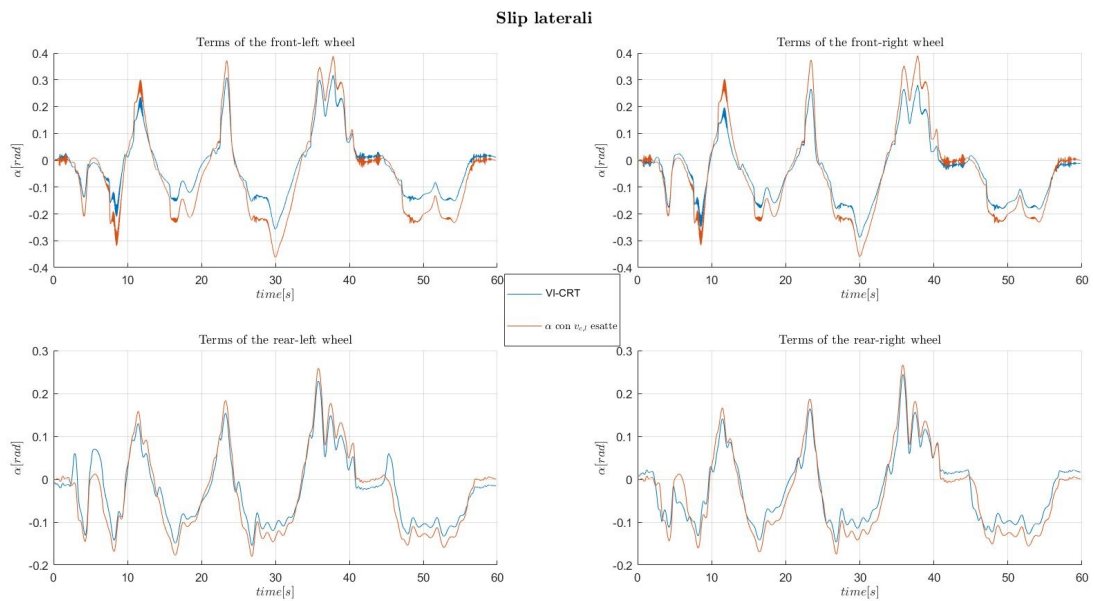


Figura 3.9: Calcolo degli angoli di slip laterale

Tali velocità infatti prendono in considerazione solo la sterzata del veicolo δ_f , tuttavia non è l'unico elemento che determina l'orientamento delle ruote, ma un ruolo fondamentale viene svolto anche dal sistema di sospensioni.

Infatti, al variare della forza normale esercitata su una gomma, la sospensione modifica il suo orientamento rispetto all'asse verticale e a quello longitudinale. L'angolo formato tra il piano contenente la ruota e l'asse longitudinale è chiamato *toe angle*, ed è lo stesso tipo di angolo che si crea quando vengono girate le ruote agendo sullo sterzo.

Invece l'angolo creato tra il piano che contiene la ruota e l'asse verticale è chiamato *camber angle*.

Al fine di ottenere delle proiezioni corrette (e quindi calcolare le velocità laterali delle ruote $v_{c_{i,j}}$ corrette) bisognerebbe tenere in considerazione anche questi angoli.

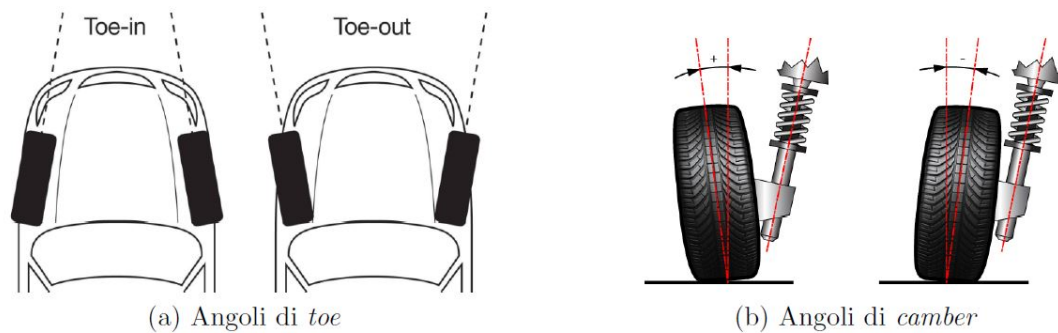


Figura 3.10: Angoli generati dal sistema di sospensioni

3.3.5 FORZE NORMALI

Le forze normali agenti sulle quattro ruote calcolate secondo l'eq. 3.19 sono mostrate in Figura 3.11.

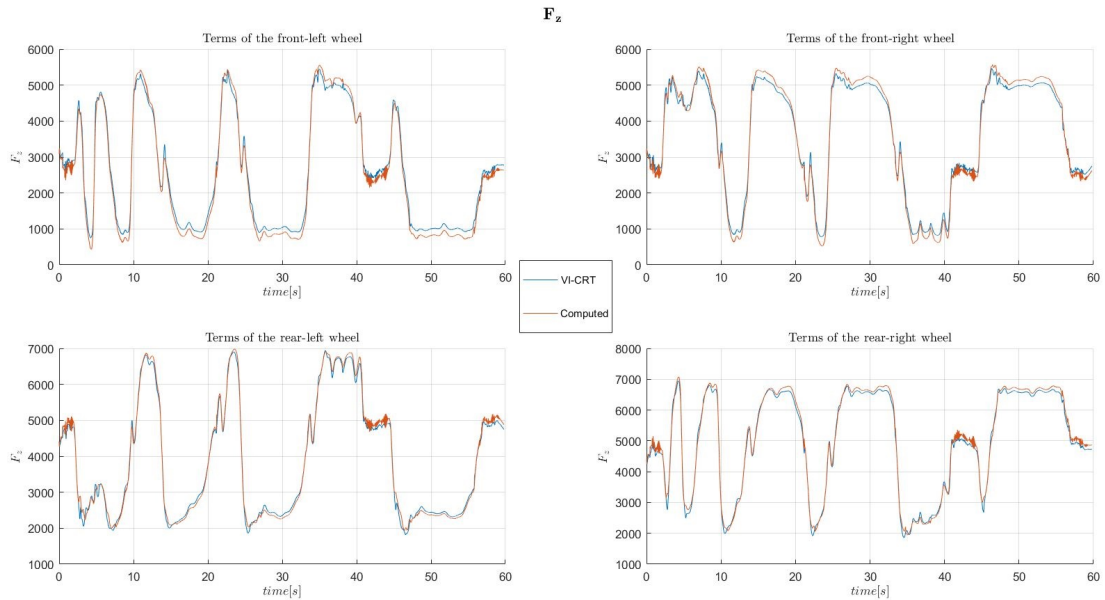


Figura 3.11: Calcolo delle forze normali delle ruote

In Tabella 3.3 sono riportati gli errori massimo e medio delle forze normali calcolate, suddivisi tra ruote anteriori e posteriori.

	$e_{max}[N]$	$e_{max}(\%)$	$e_{mean}[N]$	$e_{mean}(\%)$
$F_{z_{f,j}}$	448.81	18.4	158.49	6.25
$F_{z_{r,j}}$	630.03	16.28	92.06	2.39

Tabella 3.3: Errore massimo e medio nel calcolo delle forze normali

Le forze normali presentano un errore medio percentuale molto basso, questo sta a significare che l'accuratezza nel calcolo delle forze normali potrebbe risultare sufficientemente adeguata per il controllo, nonostante alcuni picchi di errore che sono comunque contenuti sotto il 18.4%.

4

Introduzione della dinamica combinata delle forze laterali e longitudinali

In questo capitolo viene introdotta la dinamica combinata delle forze laterali e longitudinali. Sulla base delle considerazioni esposte nel capitolo precedente, verranno considerati i seguenti miglioramenti da apportare al veicolo:

1. migliore caratterizzazione delle forze longitudinali e laterali;
2. introduzione della dinamica combinata delle forze laterali e longitudinali.

Nei prossimi paragrafi verranno perciò analizzati principalmente questi 2 aspetti, volti a sviluppare lo stesso modello di veicolo descritto nella sezione 3.2.

Nel corso di questo capitolo le forze laterali e longitudinali nel sistema di riferimento della ruota $F_{c_{i,j}}$ e $F_{l_{i,j}}$ verranno indicate semplicemente come F_y ed F_x , al fine di rendere più chiare le dinamiche della gomma.

4.1 ANALISI DELLA FORZA LONGITUDINALE

Con l'obiettivo di migliorare l'accuratezza del calcolo della forza longitudinale, vengono illustrati in questa sezione alcuni metodi alternativi per il calcolo di F_x utilizzando i valori degli slip longitudinale e laterale e la velocità di rotazione Ω che verranno introdotti nel paragrafo

4.1.1 .

4.1.1 VALIDAZIONI PRELIMINARI

Per le validazioni illustrate in questa sezione continueremo ad usare gli output di VI-CRT come valori di riferimento.

CALCOLO DELLO SLIP LONGITUDINALE k

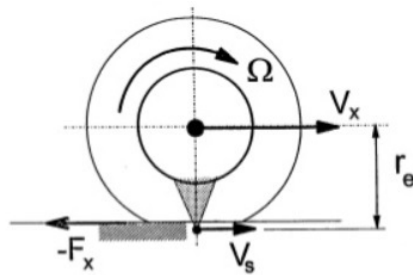


Figura 4.1: Rappresentazione della dinamica longitudinale della gomma

Lo slip longitudinale viene calcolato attraverso

$$k = -\frac{v_s}{v_x} = \frac{R_e \cdot \Omega - v_x}{v_x} \quad (4.1)$$

dove v_x ed R_e sono rispettivamente la velocità longitudinale e il raggio effettivo della ruota. A partire da questa formula sono stati ricavati gli slip longitudinali delle quattro ruote, i cui grafici sono stati riportati in Figura 4.2 .

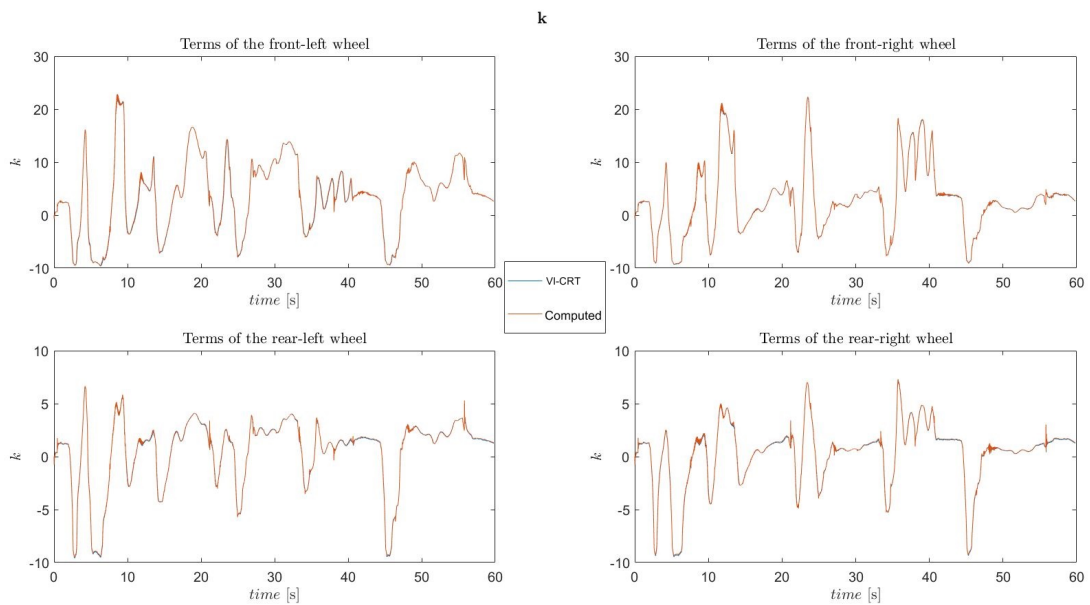


Figura 4.2: Calcolo degli slip longitudinali

DINAMICA DELLA ROTAZIONE DELLA GOMMA

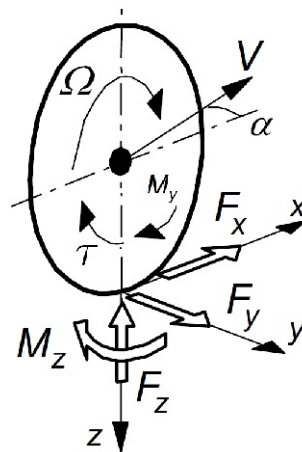


Figura 4.3: Rappresentazione della dinamica della gomma

La dinamica della rotazione della gomma è descritta dalla seguente equazione:

$$I \cdot \dot{\Omega} = -F_x \cdot R_l + \tau + M_y \quad (4.2)$$

dove R_l indica il *loaded radius*, M_y la *rolling resistance*, I il momento d'inerzia dello pneumatico e τ indica la somma dei torques di accelerazione e di frenata.

È possibile perciò ricavare il valore di Ω integrando l'equazione 4.2, a tale scopo è stato usato per l'integrazione un integratore Runge-Kutta del quarto ordine con frequenza di 100Hz.

I risultati sono mostrati in Figura 4.4.

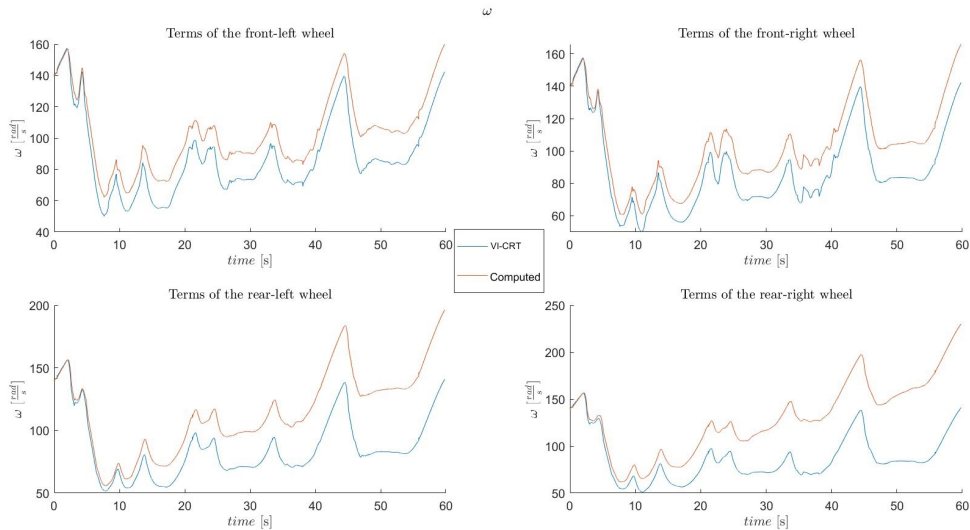


Figura 4.4: Calcolo della velocità di rotazione delle ruote

Questa volta il discostamento tra il termine reale e quello calcolato cresce con l'aumentare di t , ciò è dovuto principalmente alla deriva dell'integratore, che può essere eliminata utilizzando il sistema in retroazione.

4.1.2 MODELLI PROPOSTI PER IL CALCOLO DELLA FORZA LONGITUDINALE

Il modello utilizzato da VI-CRT per ricavare la forza longitudinale - in questo caso il PAC2002 Tire - usa la formula di Pacejka completa.

Al fine di trovare un modello il meno complesso possibile, andremo a validare 3 formule diverse per il calcolo della forza longitudinale con diversi gradi di approssimazione della formula di Pacejka:

- F_x con la sua approssimazione lineare;
- F_x con la formula di Pacejka disaccoppiata;
- F_x con la formula di Pacejka completa.

4.1.3 APPROSSIMAZIONE LINEARE DELLA FORZA LONGITUDINALE

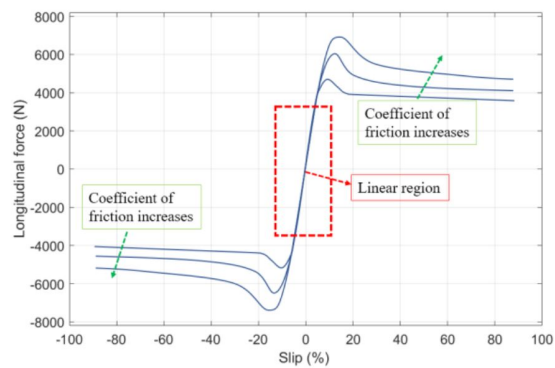


Figura 4.5: Curva tipica della formula di Pacejka

Un altro modo di calcolare la forza longitudinale è quello di supporre di avere uno slip longitudinale sufficientemente piccolo in maniera tale da trovarci nella regione lineare della formula di Pacejka, come illustrato in Figura 4.5.

L'equazione che esprime questa approssimazione lineare è data da:

$$F_x = F_z \cdot k \cdot l_s \quad (4.3)$$

dove l_s è la *longitudinal slip stiffness* di cui verrà considerato il suo valore statico. Applicando questa formula si ottengono i risultati in Figura 4.6, l'evidente inconsistenza dei valori calcolati è causata dalla mancata conoscenza del comportamento dello slip al di fuori della regione lineare della formula di Pacejka.

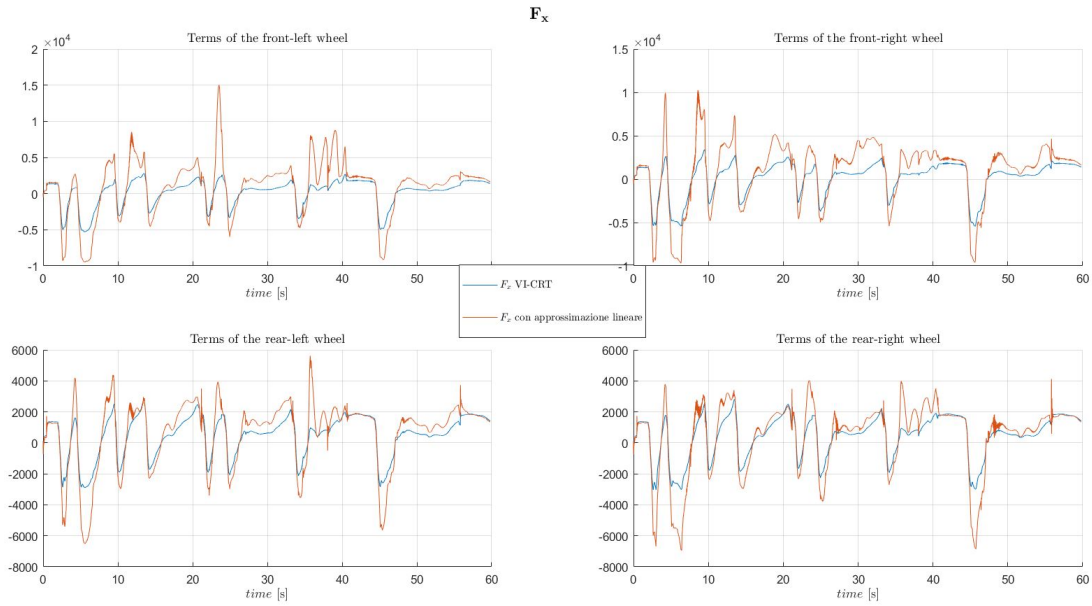


Figura 4.6: Calcolo di F_x con approssimazione lineare

FORMULA DI PACEJKA DISACCOPIATA

Riprendendo la formula di Pacejka già descritta nella sezione 3.2.2, per la dinamica longitudinale diventa:

$$F_{x_{i,j}}(k) = D \sin (C \arctan(B k_{i,j} - E (B k_{i,j} - \arctan(B k_{i,j})))) \mu \quad (4.4)$$

Applicando questa formula si ottengono i risultati in Figura 4.7. Si può notare come, rispetto alla F_x con approssimazione lineare, la formula di Pacejka disaccoppiata diminuisca il discostamento dalla F_x reale, in particolar modo i picchi sono meno accentuati. Tuttavia il calcolo rimane impreciso a causa dell'assenza dell'informazione combinata.

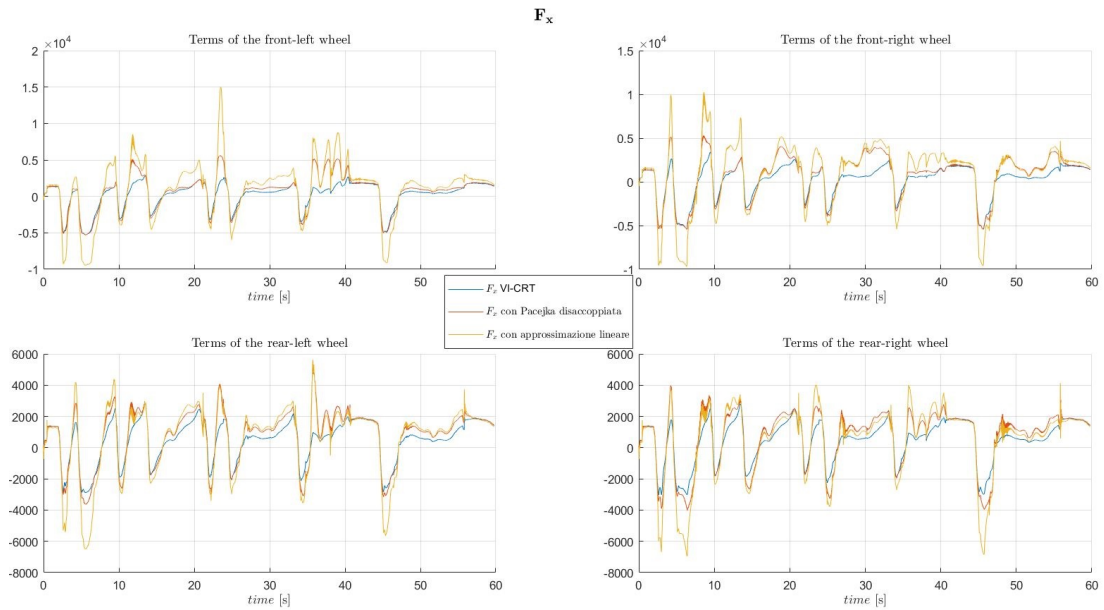


Figura 4.7: Calcolo di F_x con formula di Pacejka disaccoppiata

FORMULA DI PACEJKA COMPLETA

La formula di Pacejka completa considera sia lo slip longitudinale che lo slip laterale:

$$F_{x_{i,j}} = G_{xa}(\alpha) \cdot F_{x0}(k_{i,j}) \quad (4.5)$$

dove $F_{x0}(k_{i,j})$ è la forza longitudinale calcolata in 4.4, mentre $G_{xa}(\alpha)$ è una funzione dello slip laterale.

I risultati sono molto più precisi rispetto al calcolo con formula di Pacejka disaccoppiata, come si può notare in Figura 4.8.

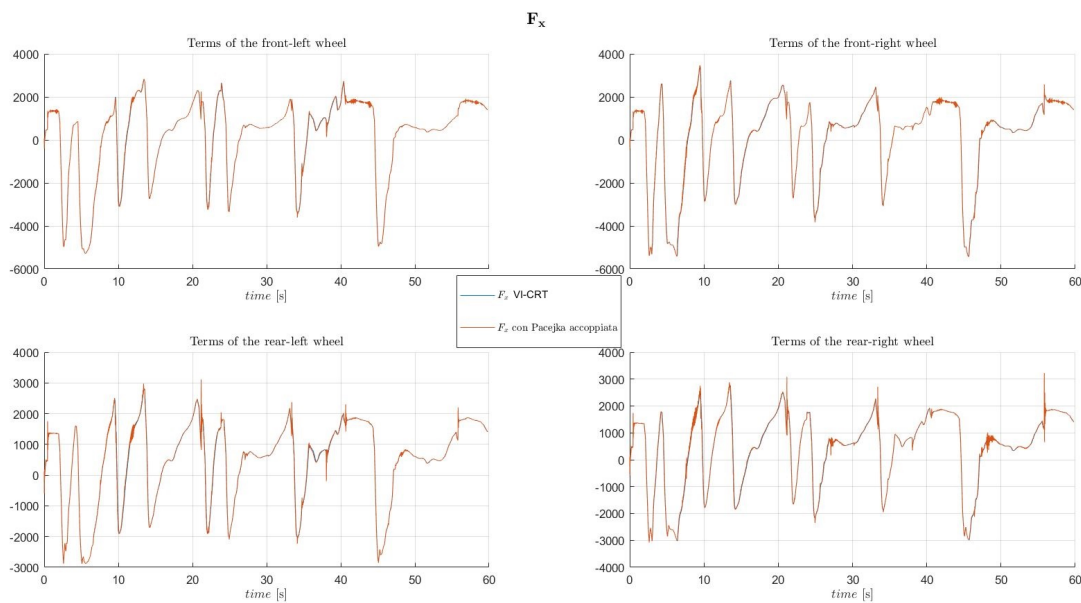


Figura 4.8: Calcolo di F_x con formula di Pacejka completa

4.1.4 CALCOLO DELLA FORZA LONGITUDINALE CON SISTEMA IN RETROAZIONE

Tra i metodi sopracitati per il calcolo di F_x notiamo che solamente la formula di Pacejka completa porta dei miglioramenti rispetto al metodo illustrato nel paragrafo 3.2.1 per il calcolo della forza longitudinale; verrà studiato perciò il comportamento di questo metodo in uno schema in retroazione.

Procediamo quindi al calcolo di F_x calcolando e aggiornando i valori di ω , k e F_x ad ogni iterazione del ciclo usando le equazioni 4.2 e 4.1, mentre per calcolare F_x verrà utilizzata la formula di Pacejka completa; in questo modo ω userà ad ogni iterazione il valore di F_x calcolato durante l'iterazione precedente, il quale utilizza lo slip longitudinale ricavato attraverso l'equazione 4.1.

Quanto detto è schematizzato in Figura 4.9, dove il calcolo di ω , k e F_x viene svolto nei rispettivi blocchi *Wheel Dynamics*, *k Computation* e *F_x Computation*.

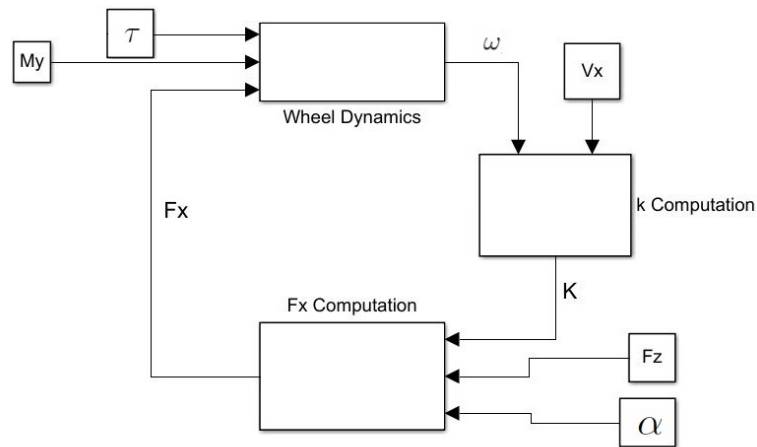


Figura 4.9: Schema a blocchi del calcolo della forza longitudinale

In Figura 4.10-4.11 sono mostrati i risultati delle iterazioni a 100 e a 500 Hz di F_x e k . Possiamo notare le molte oscillazioni presenti a 100Hz, mentre a 500Hz risulta avere un comportamento molto più stabile.

Perciò per un calcolo accurato di F_x bisognerebbe usare un modello con una frequenza di campionamento sufficientemente alta, altrimenti le predizioni risulterebbero instabili.

Ricordando però che il modello proposto deve cercare di mantenere una complessità adatta ad un'implementazione *real-time*, l'introduzione di tale descrizione della forza longitudinale viene esclusa dai miglioramenti che verranno apportati al modello del veicolo.

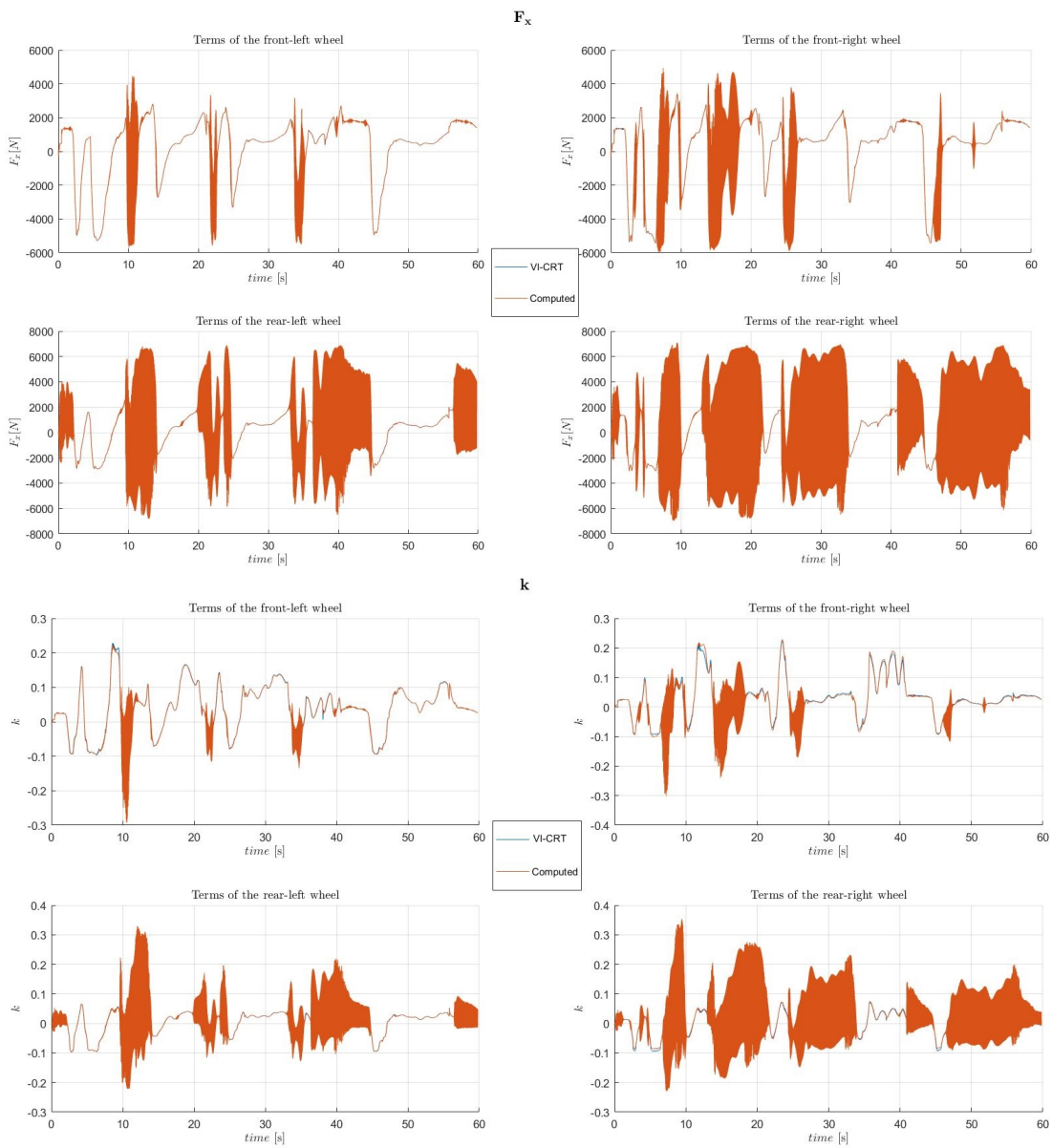


Figura 4.10: Plot di F_x e k con formula di Pacejka completa a 100Hz

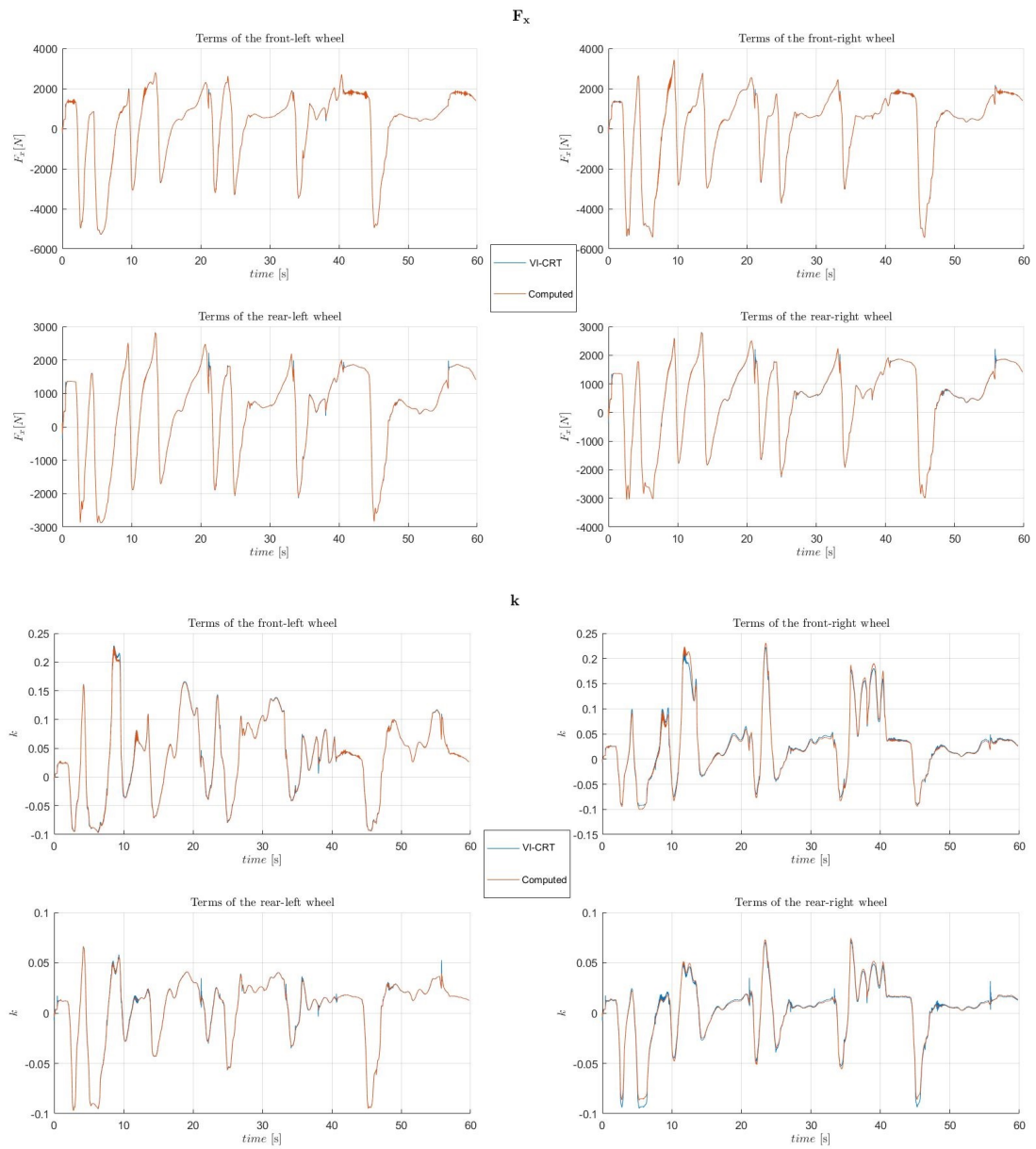


Figura 4.11: Plot di F_x e k con formula di Pacejka completa a 500Hz

4.2 MIGLIORAMENTO DELLA FORZA LATERALE

Al fine di migliorare il calcolo della forza laterale, in questa sezione verrà introdotto un metodo per migliorare il calcolo dello slip laterale che, come visto nella sezione 3.3.4, prende in considerazione solo la sterzata δ_f ma non l'angolo di toe .

In seguito verrà descritta una tecnica per ricavare lo slip longitudinale a partire dai valori di α e F_x , termine che servirà a introdurre la dinamica longitudinale nella formula di Pacejka per il calcolo di F_y .

4.2.1 MAPPA DEGLI ANGOLI DI TOE PER LO SLIP LATERALE

L'angolo di *toe* dipende dalla lunghezza del giunto della sospensione J_{susp} , che dipende a sua volta dalla forza F_{spr} agente sulla molla. Si suppone che tale forza sia determinata unicamente dall'accelerazione longitudinale \ddot{x} e laterale \ddot{y} del veicolo.

In base a queste assunzioni il metodo proposto mira a descrivere lo slip laterale delle quattro ruote del veicolo attraverso una funzione polinomiale dipendente dall'accelerazione longitudinale \ddot{x} e dallo stesso slip laterale:

$$\bar{\alpha} = f(\alpha, \ddot{x}) \quad (4.6)$$

A tale scopo ci serviremo del tool di matlab *Curve Fitting Toolbox*, con il quale possiamo ricavare la funzione ricercata attraverso il comando *surf fit*, che permette di specificare il grado polinomiale richiesto, 2 variabili in entrata e 1 variabile in uscita e come risultato restituisce un polinomio che descrive la relazione tra i 2 ingressi e l'uscita.

Notiamo dalla Figura 4.12 che una buona approssimazione dell'equazione 4.6 potrebbe essere un piano (diverso per le ruote anteriori/posteriori), dunque la mappa di toe può essere rappresentata da una mappa affine dipendente da α e da \ddot{x} :

$$\bar{\alpha}_{f,j} = 0.009612 + 0.81 \cdot \alpha_{f,j} + 2.602 \cdot 10^{-3} \cdot \ddot{x} \quad (4.7a)$$

$$\bar{\alpha}_{r,j} = 8.039 \cdot 10^{-4} + 0.8043 \cdot \alpha_{r,j} - 4.143 \cdot 10^{-3} \cdot \ddot{x} \quad (4.7b)$$

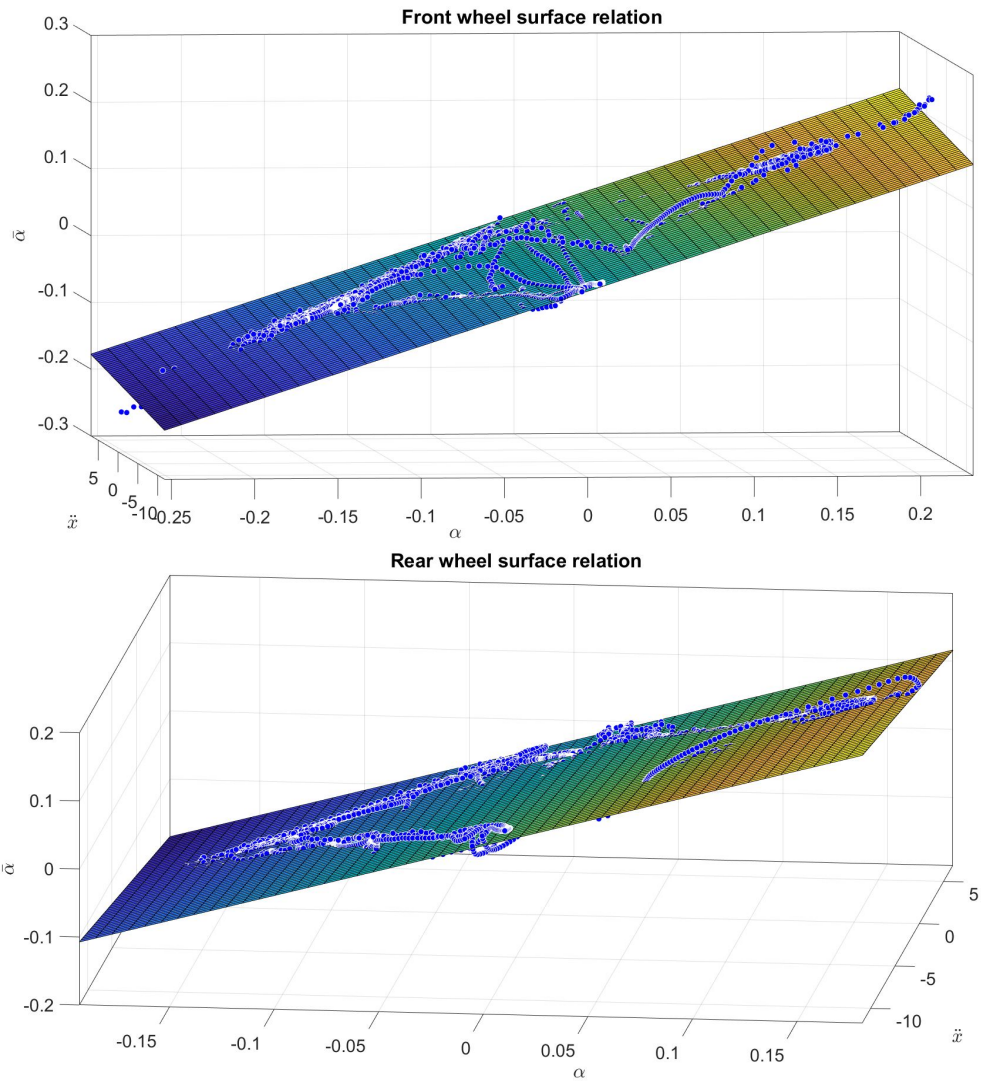


Figura 4.12: Plot delle 2 superfici trovate

Le relazioni 4.7a e 4.7b sono state adottate per le 4 ruote applicando così le correzioni agli slip laterali, i cui risultati sono mostrati in Figura 4.13.

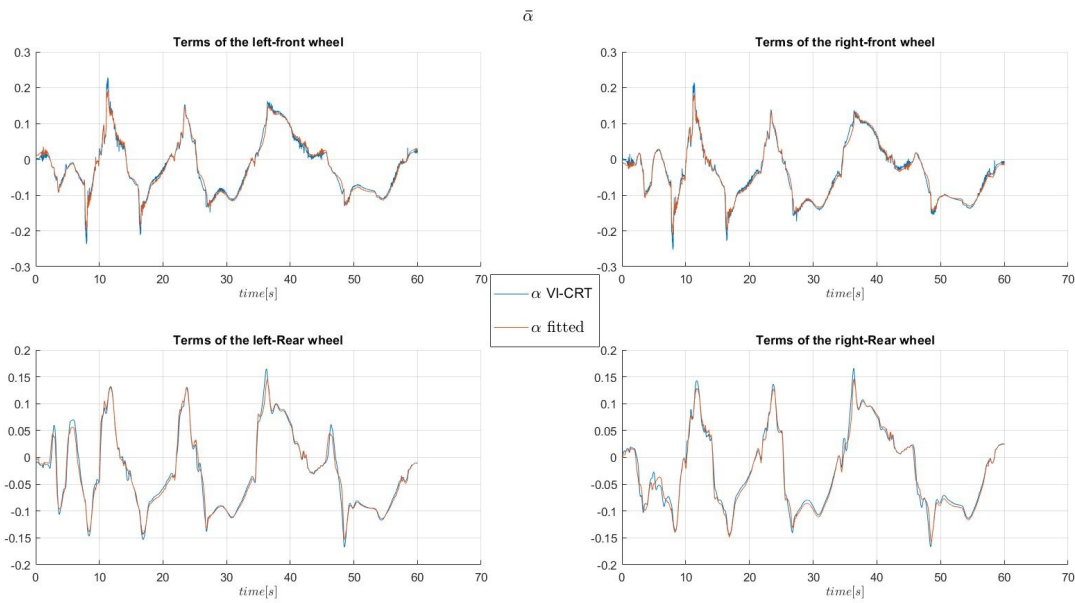


Figura 4.13: Plot degli slip longitudinali calcolati tramite fit polinomiale

Si nota un miglioramento significativo nel calcolo di α , con gli errori $e_{mean} = 0.2296\%$ e un $e_{max} = 0.0396 \frac{rad}{s}$ che sono molto bassi.

4.2.2 INTRODUZIONE DELLO SLIP LONGITUDINALE

Con un ragionamento analogo al metodo descritto nel paragrafo 4.2.1, verrà ricavato lo slip longitudinale k .

La forza laterale e longitudinale viene calcolata da VI-CRT attraverso la formula di Pacejka completa a partire dai valori di α , k , F_z e γ , ciò equivale a scrivere:

$$[F_y, F_x] = f(\alpha, k, F_z, \gamma) \quad (4.8)$$

La formula di Pacejka non è invertibile, si noti comunque che una tale formula implicherebbe calcolare 2 volte la forza laterale F_y : una prima volta attraverso la formula di Pacejka disaccoppiata e, dopo aver ricavato il termine k , una seconda volta tramite la formula di Pacejka completa.

Ciò comporta complicare il problema dal punto di vista computazionale, per questo motivo

è stato proposto un fit polinomiale del quinto ordine che a partire dai valori di α e F_x calcoli k :

$$k = f_k(\alpha, F_x) \quad (4.9)$$

In Figura 4.14 vengono mostrati i valori di k derivati con l'eq. 4.9. Come si può notare k calcolato è coerente con il comportamento tenuto dal corrispettivo valore di VI-CRT ma risulta impreciso per grandi valori dello slip longitudinale, dove le dinamiche che lo descrivono sono più complesse. Per questo motivo lo slip longitudinale non può essere usato per il calcolo della forza longitudinale, il suo uso sarà limitato per il solo calcolo di F_y tramite la formula di Pacejka.

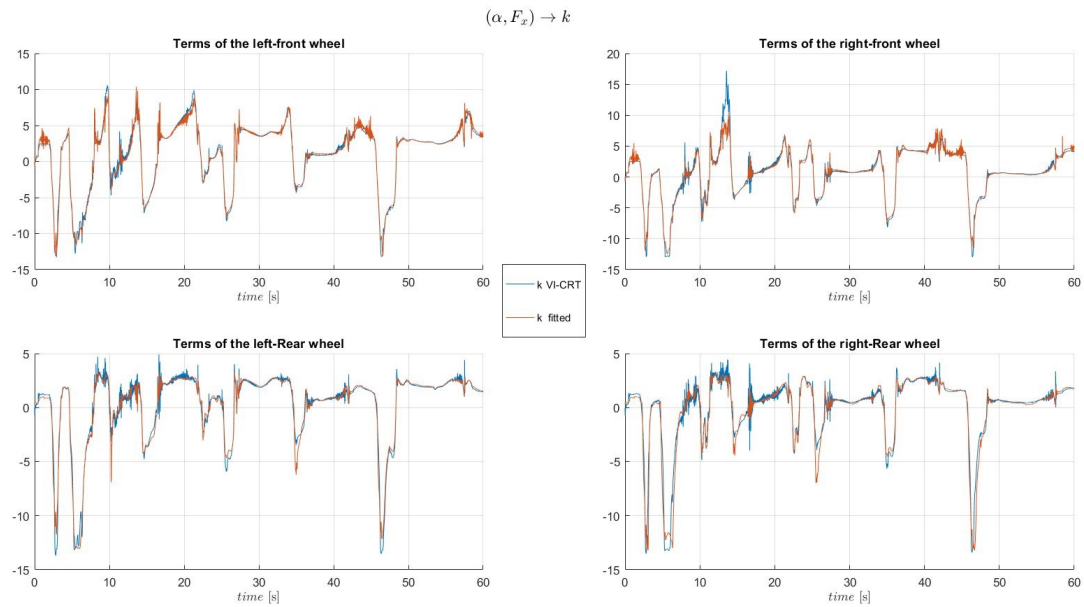


Figura 4.14: Plot degli slip longitudinali calcolati tramite fit polinomiale

4.2.3 FORMULA DI PACEJKA COMPLETA PER F_y

Avendo a disposizione lo slip longitudinale, si può applicare la formula di Pacejka completa per il calcolo della forza laterale:

$$F_{y_{i,j}} = G_{yk}(k) \cdot F_{y0}(\alpha_{i,j}) \quad (4.10)$$

dove $F_{y0}(\alpha_{i,j})$ è la forza laterale calcolata in 3.14, mentre $G_{yk}(k)$ è una funzione dello slip longitudinale.

Le forze laterali calcolate tramite 4.10, utilizzando l'eq. 4.7 per gli slip laterali, sono mostrate in Figura 4.15.

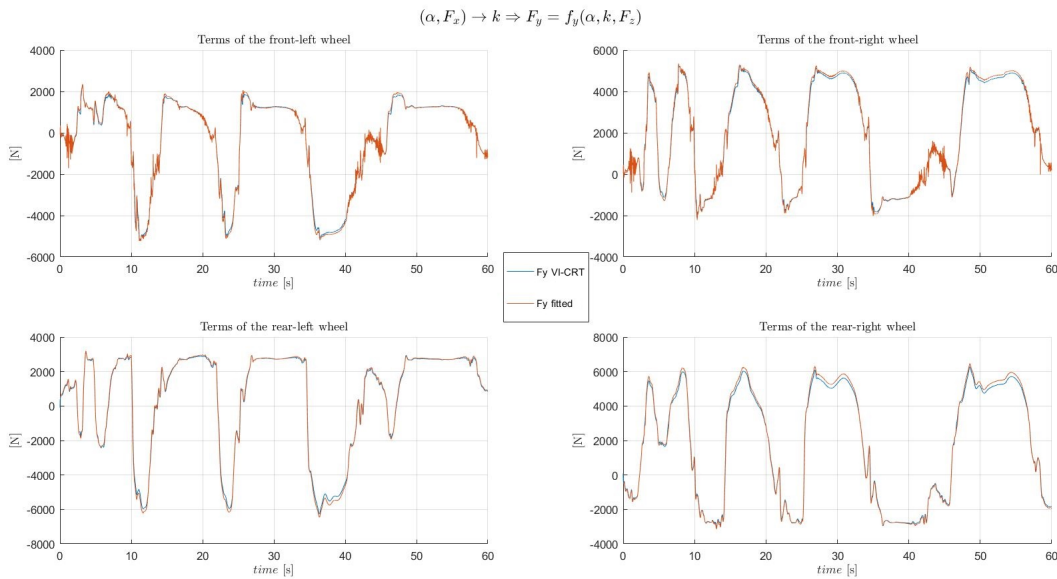


Figura 4.15: Calcolo delle forze laterali utilizzando la formula di Pacejka completa

I risultati ottenuti sono significativamente più precisi rispetto alla validazione effettuata sulle forze laterali in 3.3.4, questo è dovuto sia all'introduzione della dinamica longitudinale nella formula di Pacejka e sia alla maggior precisione degli slip laterali.

4.3 ELLISSOIDE DI ADERENZA

In questa sezione verrà descritto come verrà ricavato l'ellissoide di aderenza (*friction ellipse*) [16].

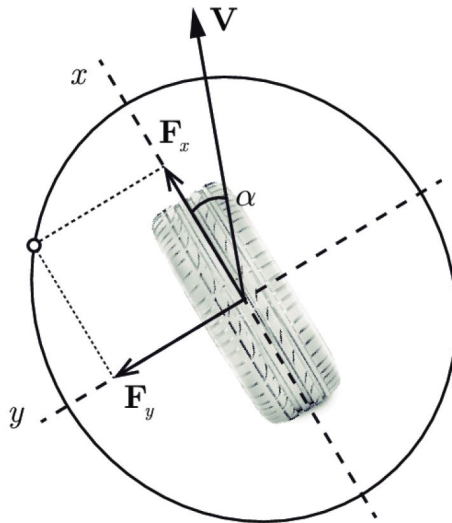


Figura 4.16: Schema semplificato dell'ellissoide di aderenza

L'ellissoide di aderenza rappresenta la forza laterale F_y e longitudinale F_x che uno pneumatico riesce a trasmettere a terra per un dato carico F_z .

L'equazione generale che descrive l'ellisse nominale delle forze agenti esprimibili da uno pneumatico è data da:

$$\frac{F_y^2}{\mu_y^2 F_z^2} + \frac{F_x^2}{\mu_x^2 F_z^2} \leq 1 \quad (4.11)$$

dove F_z indica la forza normale e μ_x e μ_y sono i coefficienti d'attrito longitudinale e laterale dello pneumatico.

In particolare gli assi dell'ellisse, che corrispondono alle forze massime longitudinali e laterali applicabili allo pneumatico, si trovano attraverso:

$$asse F_x = \mu_x F_z \quad (4.12)$$

$$asse F_y = \mu_y F_z \quad (4.13)$$

quindi più in generale vale:

$$\frac{F_x}{F_z} \leq \mu_x \quad (4.14)$$

$$\frac{F_y}{F_z} \leq \mu_y \quad (4.15)$$

In Figura 4.17 vengono mostrate le ellissi nominali normalizzate su F_z , dove gli assi delle ellissi corrispondono perciò a μ_x e μ_y ed i relativi valori di forza sono stati normalizzati secondo le eq. 4.14 e 4.15.

Notiamo grazie a questi grafici che le forze esercitate sugli pneumatici superano le soglie ammesse dalle ellissi, infatti i coefficienti d'attrito non sono costanti, ma sono dipendenti dalla forza normale agente sullo pneumatico e la cui espressione è ricavata in sezione 4.3.1.

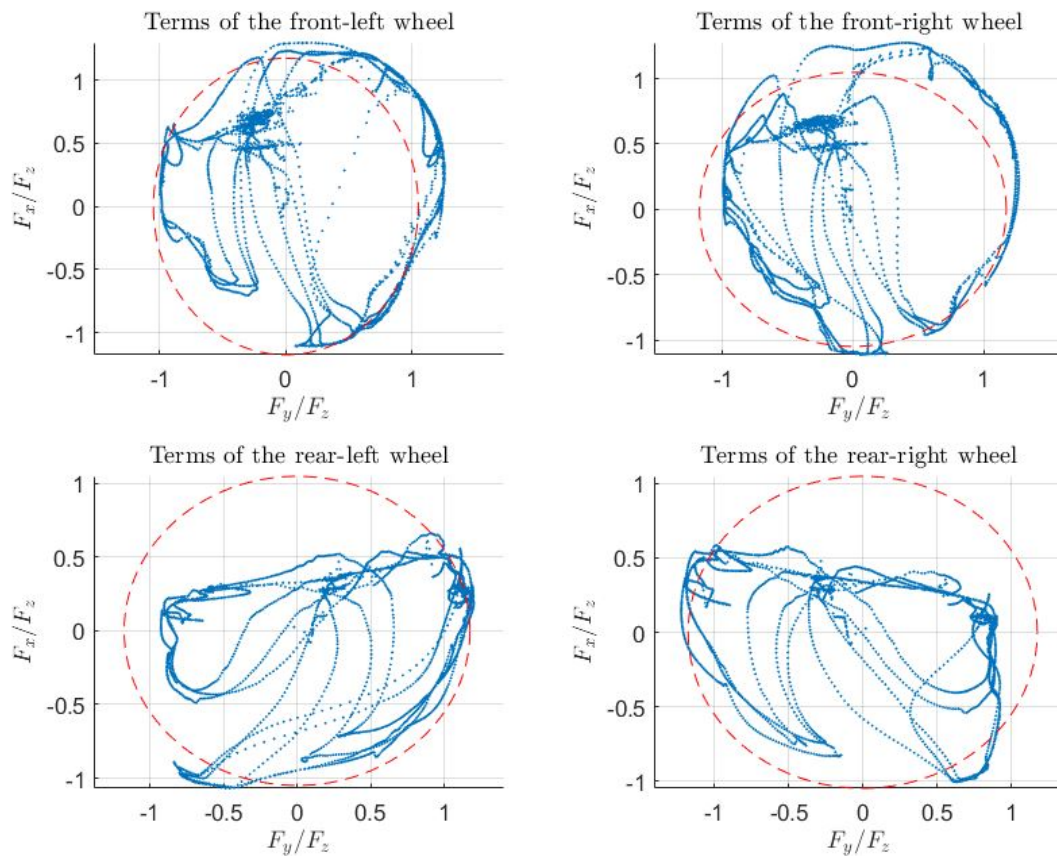


Figura 4.17: Ellisse nominale normalizzata per le quattro ruote

4.3.1 CALCOLO DEI COEFFICIENTI D'ATTRITO

I coefficienti d'attrito saranno ricavati tramite le formule:

$$\mu_{x_{max,min}} = \frac{F_{x_{max,min}}}{F_z} \quad , \quad \mu_{y_{max,min}} = \frac{F_{y_{max,min}}}{F_z} \quad (4.16)$$

I valori di $F_{x_{max,min}}$ e $F_{y_{max,min}}$ sono i valori massimi e minimi di forza esprimibile dalla gomma a determinati valori di F_z , calcolati a camber nullo.

Un secondo test è stato fatto considerando anche l'angolo di camber γ . I risultati sono mostrati in Figura 4.18 e 4.19.

Si può notare in Figura 4.18 che, senza considerare l'angolo di camber, i coefficienti d'attrito sono linearmente dipendenti dalla forza normale, mentre in Figura 4.19 dipendenza che scompare se viene considerato l'angolo di camber (lineare a tratti, con coefficiente che cambia a $F_z = 2500$).

Nelle prossime sezioni verranno usate le approssimazioni lineari per il calcolo dei coefficienti d'attrito in entrambi i casi.

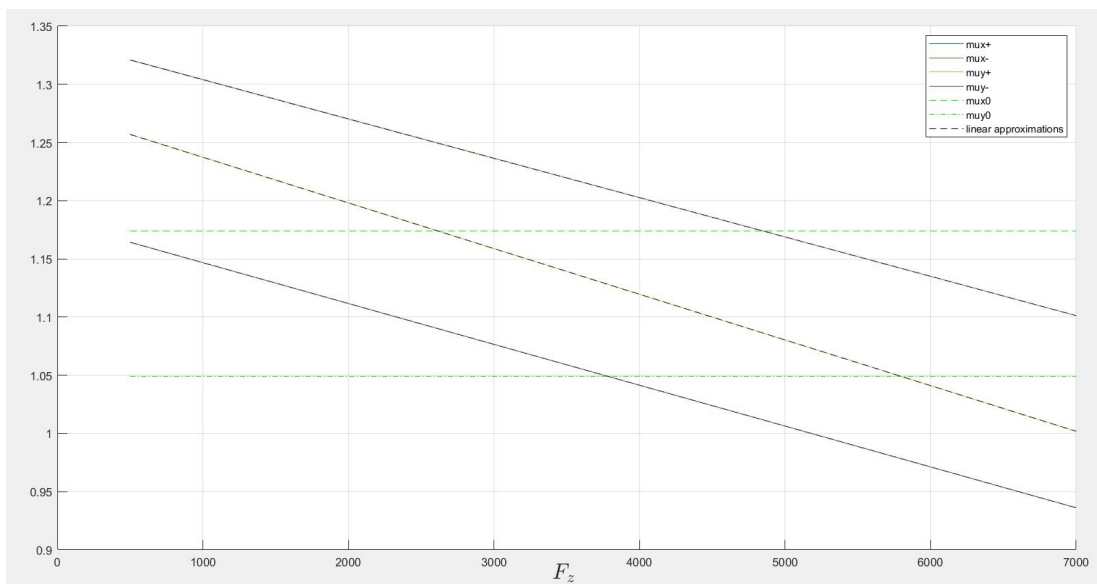


Figura 4.18: Coefficienti d'attrito calcolati senza considerare l'angolo di camber

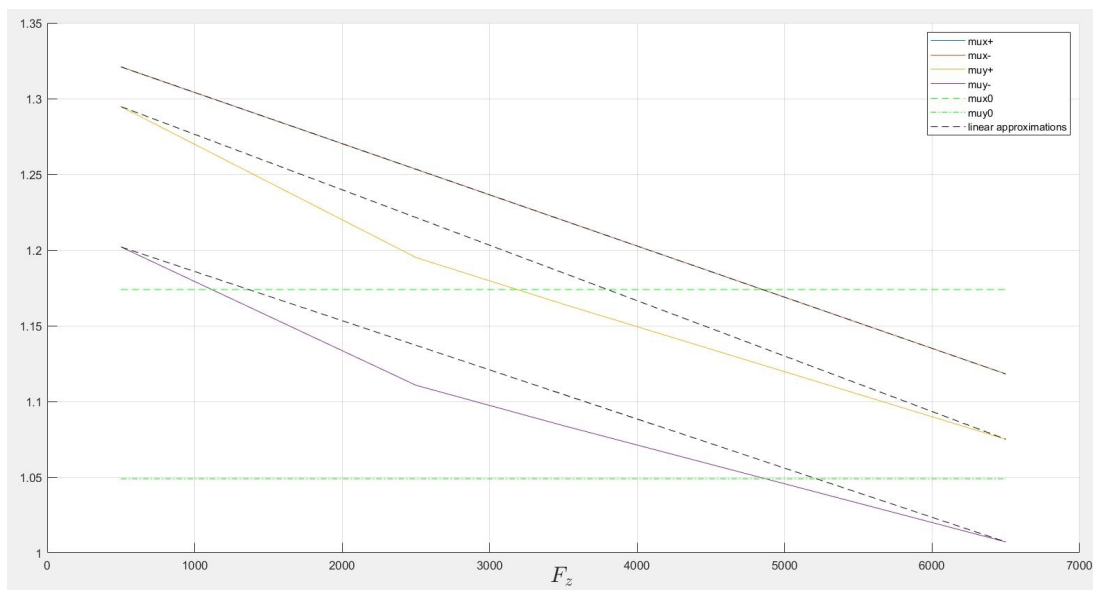


Figura 4.19: Coefficienti d'attrito calcolati considerando l'angolo di camber

4.3.2 CALCOLO DELL'ELLISSE TRAMITE LA FORMULA DI PACEJKA

Il calcolo delle forze effettivamente esprimibili dalla gomma, e quindi l'ellisse effettivo di aderenza, può essere fatto tramite la formula di Pacejka combinata, calcolando, per ogni valore di forza normale, la massima forza laterale e longitudinale esprimibili contemporaneamente. In Figura 4.20 vengono mostrate per il caso $F_z = 500N$ l'ellisse nominale e l'ellisse calcolata tramite Pacejka, considerando l'angolo di camber variabile.

Per l'ellisse teorica sono stati usati 2 valori di μ_y differenti, uno per valori di F_y negativi e uno per valori di F_y positivi.

Possiamo vedere come l'ellisse nominale non sia sufficiente a contenere tutti i possibili valori esercitati dalle forze laterali e longitudinali.

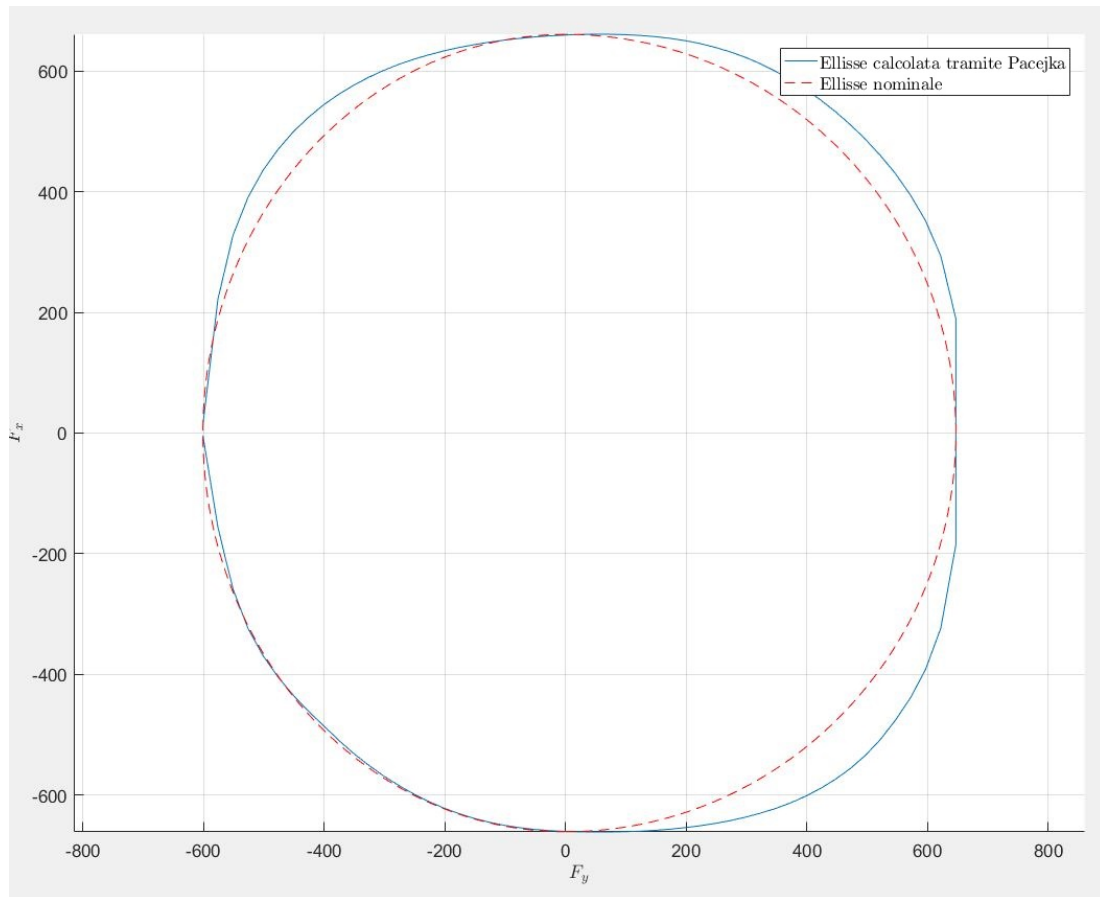


Figura 4.20: Ellisse nominale ed ellisse calcolata tramite Pacejka per $F_z = 500N$

4.3.3 RELAZIONE TRA LE ELLISSI CALCOLATE TRAMITE PACEJKA

È utile capire quale relazione di dipendenza esiste con la forza normale tra le ellissi di aderenza effettive.

Sono state quindi analizzate le ellissi effettive a differenti valori di F_z , calcolate tramite la formula 4.11 con i valori di $F_{x_{max,min}}$ e $F_{y_{max,min}}$ trovati in sezione 4.3.2.

In Figura 4.21 sono mostrati i risultati considerando variabile l'angolo di camber. Le ellissi risultano avere una buona correlazione, con qualche discostamento significativo dovuto all'approssimazione dei coefficienti d'attrito.

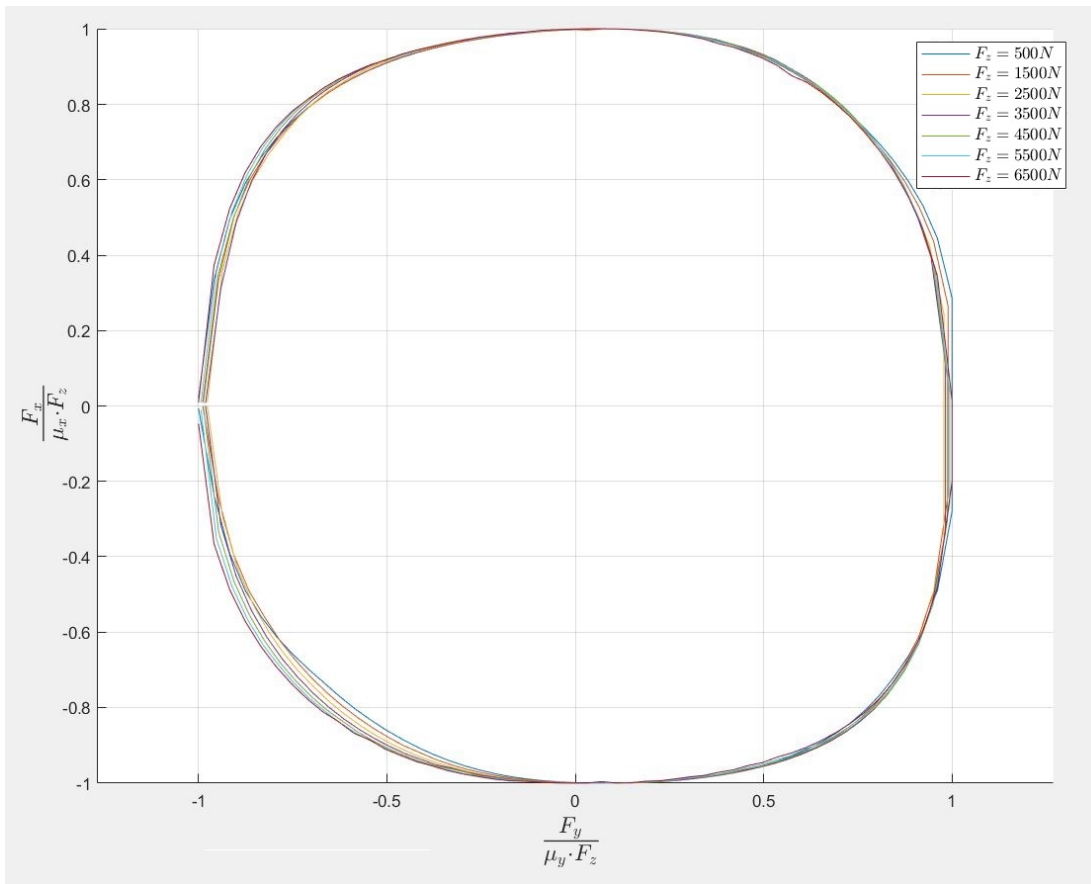


Figura 4.21: Relazione tra le ellissi (considerando γ)

4.3.4 PROIEZIONE DELLE FORZE ALL'INTERNO DELL'ELLISSOIDE DI ADERENZA

Verifichiamo che le ellissi siano effettivamente rappresentative della massima forza esprimibile dalle gomme, verificando quindi se le forze ottenute in simulazione e normalizzate sono contenute in esse.

A tale scopo utilizziamo il fascio di ellissi normalizzato trovato nella sezione 4.3.3 e normalizziamo allo stesso modo i valori F_x e F_y ricavati da VI-CRT.

Come si può vedere dalla Figura 4.22 ogni coppia di valori delle forze laterali e longitudinali è contenuta nel fascio di ellissi.

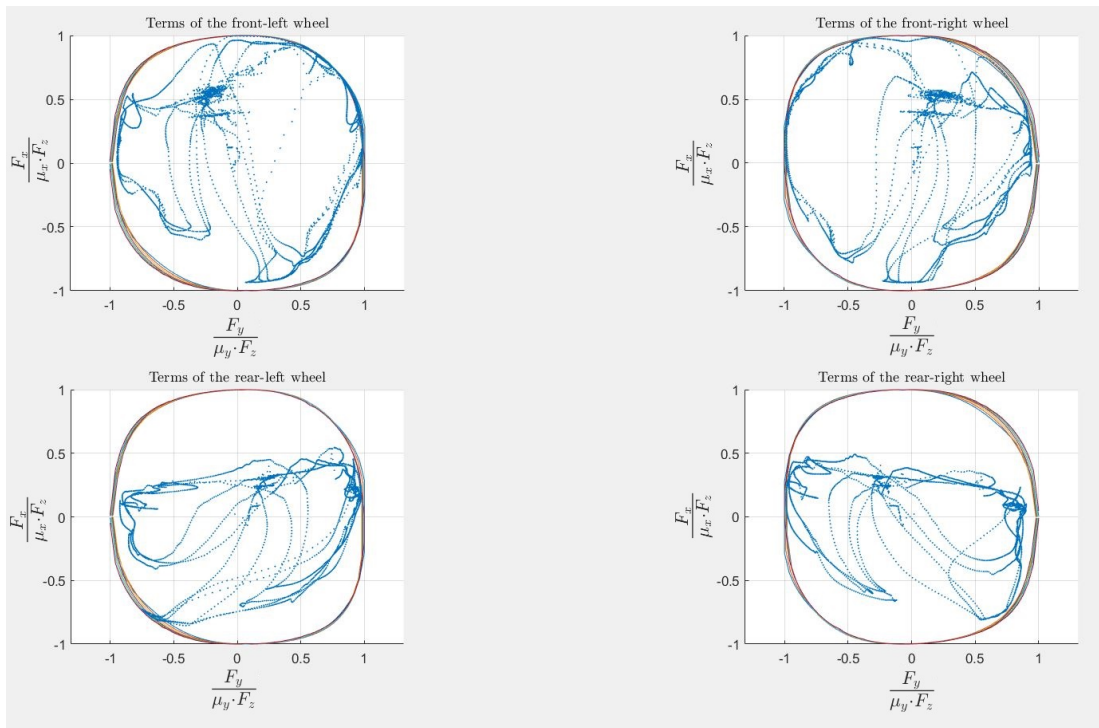


Figura 4.22: Scatter plot normalizzato

4.3.5 VALIDAZIONE DELL'ELLISSOIDE

Grazie alle modifiche proposte in questo capitolo, il modello ora risulta più accurato per quanto riguarda la dinamica laterale, ma presenta ancora delle forti lacune per quanto riguarda la dinamica longitudinale. Per inserire informazioni sulla massima forza esprimibile dalle gomme si è scelto quindi di introdurre le ellissi di aderenza nel modello.

Per semplicità, anziché utilizzare l'ellissoide di aderenza derivato in 4.3.3, verrà usata un'ellisse nominale analoga a quella descritta dall'eq. 4.11 con un raggio esteso a 1.1, la cui implementazione sarà basata su *soft constraint*.

In Figura 4.23 viene mostrato quanto appena descritto, le forze stimate con le tecniche proposte in questo capitolo (in rosso) sono messe a confronto con quelle di VI-CRT (in blu), assieme al fascio di ellissi normalizzate ricavate in 4.3.3, invece in verde è rappresentata l'ellisse nominale con raggio esteso a 1.1.

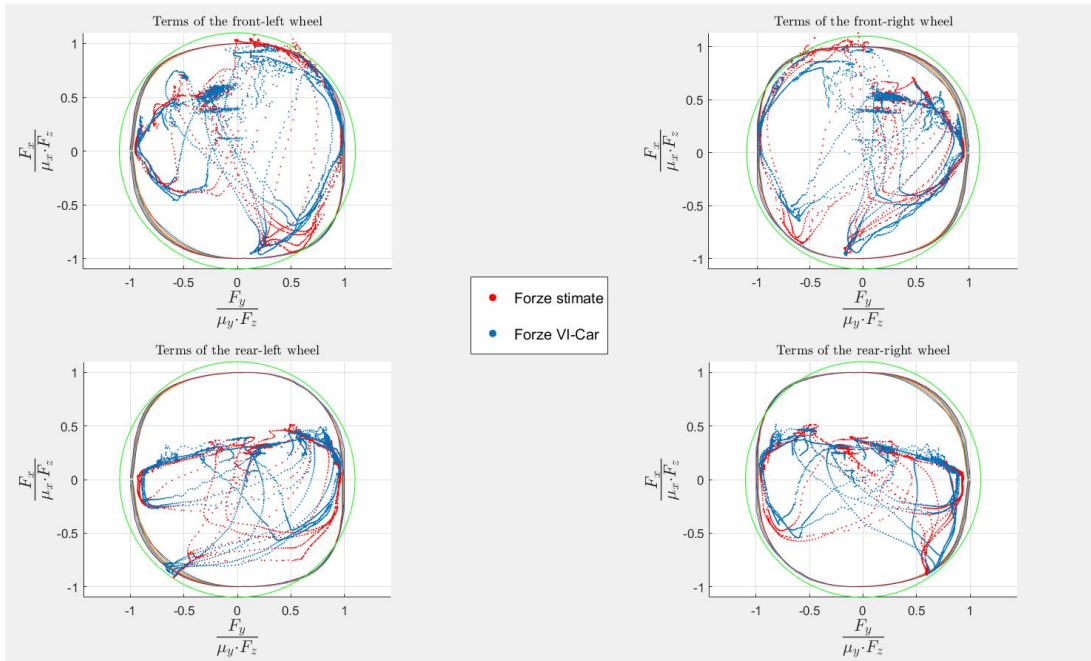


Figura 4.23: Ellissoidi di aderenza con forze stimate e reali

5

Simulazione e risultati

Verrà descritto in questo capitolo l'ambiente di simulazione utilizzato per il controllo del veicolo virtuale, in seguito verranno mostrati i risultati delle simulazioni e in particolare l'influenza delle modifiche apportate al modello.

Il tracciato che verrà utilizzato per le simulazioni di questa tesi è mostrato in Figura 5.1.

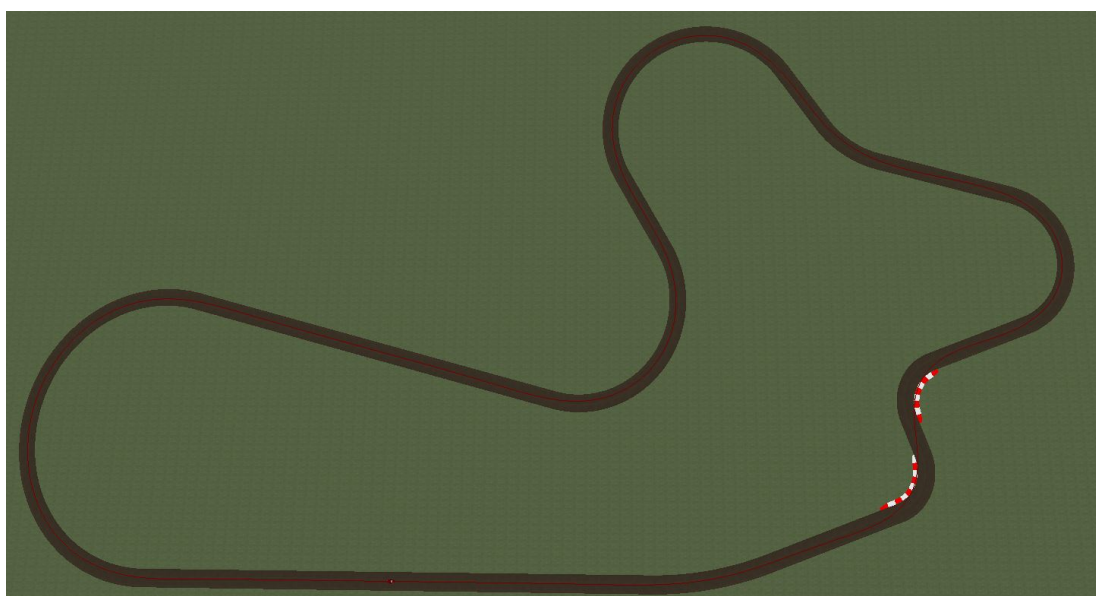


Figura 5.1: Vista dall'alto del circuito

Per effettuare le simulazioni è stato utilizzato MATLAB congiuntamente con il pacchetto software VI-Grade.

5.1 IMPLEMENTAZIONE IN MATMPC

5.1.1 MODELLO

Il modello descritto nel capitolo 4 è stato implementato in MATMPC, con una riformulazione spaziale della dinamica [17, 18]. In questo modo le equazioni della dinamica sono definite in base allo spazio percorso lungo la traiettoria di riferimento invece che rispetto al tempo.

Per una spiegazione dettagliata di questa riformulazione della dinamica si rimanda all'Appendice A.

Il modello effettivamente utilizzato, scritto in linguaggio CasADi, deriva direttamente dalla dinamica descritta nella sezione 3.2, a cui vengono aggiunti i termini $\tilde{\Delta}_{lon}^z$ e $\tilde{\Delta}_{lat}^z$ per poter calcolare il trasferimento di carico, gli errori di orientamento e_ψ e di posizione e_y . Si è scelto infine di aggiungere allo stato anche i termini δ_f , β_t e β_b , e considerare come ingressi le loro derivate; in questo modo è possibile ottenere delle funzioni di ingresso più regolari impostando dei pesi e limiti adeguati a $\dot{\delta}_f$ e $\dot{\beta}$.

Perciò il sistema risulta definito dai seguenti vettori di stato, di ingresso e dei parametri:

$$\xi = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \\ e_\psi \\ e_y \\ \tilde{\Delta}_{lon}^z \\ \tilde{\Delta}_{lat}^z \\ \delta_f \\ \beta_t \\ \beta_b \end{bmatrix} \quad u = \begin{bmatrix} \dot{\delta}_f \\ \dot{\beta}_t \\ \dot{\beta}_b \\ \varepsilon \end{bmatrix} \quad p = \begin{bmatrix} k \\ \mu \end{bmatrix} \quad (5.1)$$

dove k indica la curvatura del tracciato ed ε è una variabile di controllo usata per l'implementazione dei *soft constraint* descritti da:

$$\frac{F_y^2}{\mu_y^2 F_z^2} + \frac{F_x^2}{\mu_x^2 F_z^2} \leq 1.1 + \varepsilon \quad (5.2)$$

dove ε è una variabile di slack che permette di gestire il vincolo in modo soft, evitando condizioni di infeasibility.

La dinamica del veicolo espressa in coordinate spaziali è data dal seguente sistema di equazioni differenziali

$$\dot{\xi}^s = \begin{bmatrix} \dot{y}\dot{\psi} + \frac{1}{m} \left(\sum_{i,j} F_{x_{i,j}} - F_x^d \right) \\ -\dot{x}\dot{\psi} + \frac{1}{m} \left(\sum_{i,j} F_{y_{i,j}} \right) \\ \frac{1}{I_z} \left[a \left(\sum_j F_{y_{f,j}} \right) - b \left(\sum_j F_{y_{r,j}} \right) + c \left(\sum_i F_{x_{i,r}} - \sum_i F_{x_{i,l}} \right) \right] \\ \dot{\psi} - \frac{k}{1-k} e_y (\dot{x} \cos(e_\psi) - \dot{y} \sin(e_\psi)) \\ \dot{x} \cos(e_\psi) + \dot{y} \sin(e_\psi) \\ \frac{1}{\tau_{LT}} \left(-\frac{\sum_{i,j} F_{x_{i,j}}}{2(a+b)} h - \tilde{\Delta}_{lon}^z \right) \\ \frac{1}{\tau_{LT}} \left(\frac{\sum_{i,j} F_{y_{i,j}}}{4c} h - \tilde{\Delta}_{lat}^z \right) \\ \dot{\delta}_f \\ \dot{\beta}_t \\ \dot{\beta}_b \end{bmatrix} \cdot \frac{1}{\dot{s}} \quad (5.3)$$

Per una completa descrizione delle componenti in posizione 4,5 del vettore $\dot{\xi}^s$ e del termine \dot{s} si rimanda all'Appendice A.

5.1.2 FUNZIONE OBIETTIVO

La funzione obiettivo $obj(\cdot)$ è un elemento cruciale del sistema in quanto caratterizza la strategia di controllo minimizzando il problema di ottimizzazione descritto dall'eq. 2.9.

I vettori utilizzati sono

$$h(\cdot) = \begin{bmatrix} v \\ e_y \\ \dot{e}_y \\ e_\psi + \alpha \\ \dot{e}_\psi \\ \beta_t \cdot \beta_b \\ \dot{\delta}_f \\ \dot{\beta}_t \\ \dot{\beta}_b \\ \varepsilon \end{bmatrix} \quad h_N(\cdot) = \begin{bmatrix} v \\ e_y \\ \dot{e}_y \\ e_\psi + \alpha \\ \dot{e}_\psi \\ \beta_t \cdot \beta_b \end{bmatrix}. \quad (5.4)$$

La funzione obiettivo risultante è definita sia per ogni punto di shooting i

$$obj_i(\xi, u, p, ref) = \frac{1}{2} Q \|h(\xi, u, p) - ref\|_2^2 \quad (5.5)$$

che specificatamente nello stato finale N

$$obj_N(\xi_N, p_N, ref_N) = \frac{1}{2} Q_N \|h_N(\xi_N, p_N) - ref_N\|_2^2 \quad (5.6)$$

dove Q e Q_N sono delle opportune matrici di pesi, mentre ref è il vettore contenente i riferimenti. Quest'ultimo ha solo la prima componente diversa da zero, ovvero il riferimento di velocità, il quale viene generato direttamente dal software VI-Grade come una stima della velocità massima con cui è possibile percorrere il tracciato.

I valori inseriti nella matrice della funzione obiettivo meritano alcune considerazioni:

- la velocità del veicolo v è stata inserita perchè permette di assegnare un profilo di velocità che il driver cercherà di seguire;
- gli errori angolare $e_\psi + \alpha$, in cui viene considerata la direzione di marcia includendo lo slip del veicolo, e laterale e_y sono necessari per la corretta percorrenza della traiettoria e per evitare sbandate del veicolo;
- la derivata dell'errore longitudinale \dot{e}_y evita che il controllore preveda brusche sterzate;
- la derivata dell'errore angolare \dot{e}_ψ impone condizioni stabili di percorrenza ed è particolarmente importante nello stato finale perchè impone che la traiettoria prevista si concluda in condizione controllate di velocità angolare;

- il prodotto tra $\beta_t \cdot \beta_b$ evita che il controllore preveda situazioni in cui il veicolo sta sia accelerando che frenando;
- la derivata dell'ingresso $\dot{\delta}_f$ è necessaria affinché lo sterzo venga modificato solo in caso di discostamento significativo dalla traiettoria ed evita quindi instabilità e ondeggiamenti;
- le derivate del controllo $\dot{\beta}_t$ e $\dot{\beta}_b$ migliorano la regolarità dell'accelerazione/frenata.

5.1.3 VINCOLI

Nel controllore, oltre ai vincoli descritti dall'eq. 5.2, sono presenti sostanzialmente due tipi di vincoli: quelli sugli stati e quelli sugli ingressi. I vincoli sugli ingressi hanno la duplice funzione di aiutare il solver restringendo l'insieme delle possibili soluzioni e di impedire cambiamenti drastici degli input.

I vincoli imposti per gli stati sono

$$lb_x = \begin{bmatrix} -\frac{\pi}{2} \\ -10 \\ -\frac{\pi}{6} \\ 0 \\ 0 \end{bmatrix} \leq \begin{bmatrix} e_\psi \\ e_y \\ \delta_f \\ \beta_t \\ \beta_b \end{bmatrix} \leq \begin{bmatrix} +\frac{\pi}{2} \\ +10 \\ +\frac{\pi}{6} \\ 1 \\ 1 \end{bmatrix} = ub_x \quad (5.7)$$

dove i limiti su e_ψ e δ_f sono in $[rad]$, quello su e_y è in $[m]$, mentre quello su β_t e β_b è adimensionale. I limiti sugli errori permettono di decidere quanto deve essere preciso l'inseguimento della traiettoria, mentre i vincoli su δ_f , β_t e β_b sono fisiologici in quanto lo sterzo non può superare i 30° e la richiesta di accelerazione/frenata è normalizzata.

I vincoli imposti per gli ingressi sono invece sono

$$lb_u = \begin{bmatrix} -1 \\ -2 \\ -2 \end{bmatrix} \leq \begin{bmatrix} \dot{\delta}_f \\ \dot{\beta}_t \\ \dot{\beta}_b \end{bmatrix} \leq \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix} = ub_u \quad (5.8)$$

dove il limite su $\dot{\delta}_f$ è espresso in $[\frac{rad}{s}]$, mentre i limiti su $\dot{\beta}_t$ e $\dot{\beta}_b$ sono adimensionali.

5.1.4 PARAMETRI DEL SOLVER

L'implementazione in MATMPC prevede l'utilizzo di un *solver*, ovvero una funzione che calcola la soluzione al problema di ottimizzazione avendo i riferimenti, i vincoli e lo stato attuale del veicolo. I parametri principali del solver sono i seguenti:

- il numero di iterazioni massimo per ogni chiamata al solver $sqp_{maxit} \in \mathbb{N}$, ossia il parametro che decreta l'uscita dalla funzione anche nel caso non sia stato raggiunto l'ottimo ed influenza profondamente il tempo di calcolo nelle situazioni più estreme;
- il numero di punti di shooting $N \in \mathbb{N}$, parametro fondamentale perchè è il valore che influenza più direttamente il tempo di calcolo richiesto per ogni passo ¹ ed inoltre, congiunto al passo di campionamento, stabilisce l'orizzonte di predizione;
- il passo di campionamento dell'integratore $T_s \in \mathbb{R}$ è il passo (spaziale ²), valore che permette una parametrizzazione più o meno fitta delle traiettorie;
- l'orizzonte di predizione $H = T_s \cdot N$, $H \in \mathbb{R}$, non è un parametro indipendente ma è necessario che risulti sufficientemente lungo per permettere al controllore di scegliere gli ingressi in base al comportamento futuro;
- il valore della *KKT* minimo per decretare di essere in un punto ottimo $kkt_{lim} \in \mathbb{R}$, è utilizzato solo nel caso $sqp_{maxit} > 1$ ed influenza, congiunto a sqp_{maxit} , il tempo di calcolo massimo.

In Tabella 5.1 sono riportati i valori dei parametri appena descritti, che verranno modificati nel corso delle varie simulazioni mostrate nel seguito.

Parametro	Valore
sqp_{maxit}	2
N	40
T_s	2 [m]
H	80 [m]
kkt_{lim}	10^3

Tabella 5.1: Parametri utilizzati nell'implementazione in MATMPC

¹il tempo di calcolo cresce cubicamente rispetto al numero di punti di shooting

²essendo il modello espresso come sistema di equazioni differenziali derivate rispetto allo spazio s , il passo di campionamento è espresso in [m].

5.2 COSIMULAZIONE IN SIMULINK

In questa sezione sono state effettuate le simulazioni che hanno portato al completamento effettivo dello sviluppo del driver ed ai relativi test.

Il software VI-Grade fornisce un blocco Simulink predisposto alla cosimulazione che necessita di un file, chiamato *send_svm*, generato in *Test Mode* da VI-CRT e che contiene tutte le informazioni relative al modello del veicolo, alla traiettoria da seguire e al controllore. In particolare la traiettoria può essere importata in MATLAB attraverso l'apposita funzione. Inoltre il blocco è dotato di bus di ingresso e uscita in cui si possono selezionare le variabili di interesse.

La struttura della cosimulazione, mostrata in Figura 5.2, prevede un loop in cui il blocco di VI-CRT fornisce lo stato della simulazione alla funzione *MATMPC_controller*, che provvede a calcolare gli ingressi di controllo.

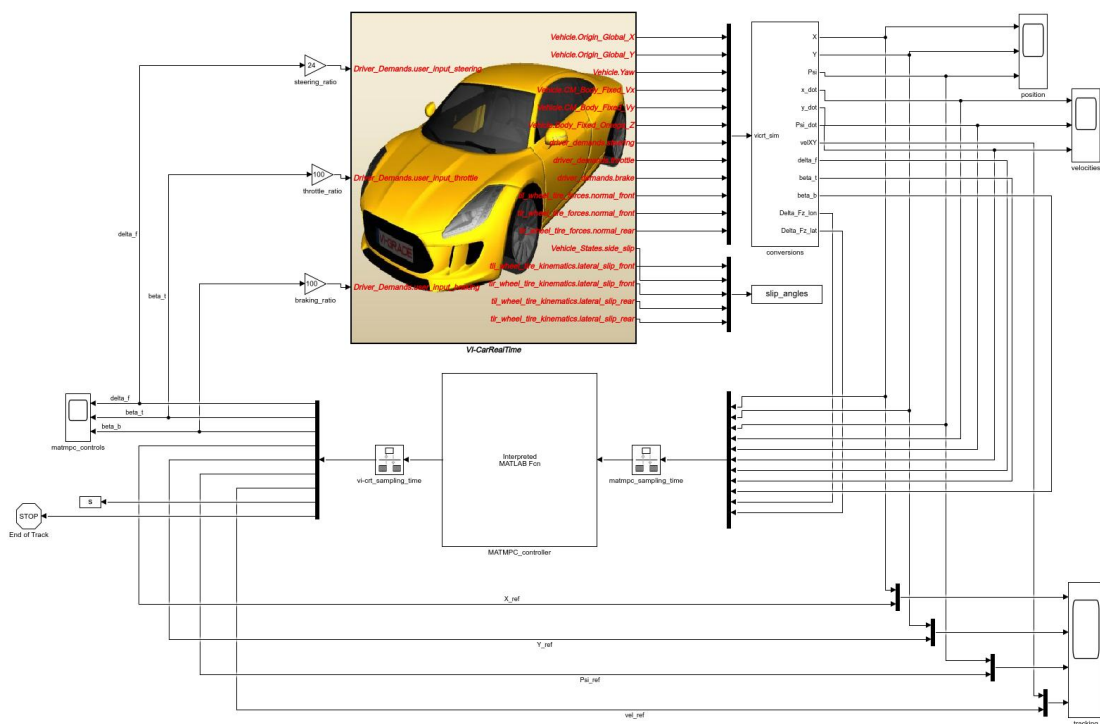


Figura 5.2: Schema Simulink della cosimulazione

Una volta terminata la cosimulazione viene generato un file *.res* che contiene tutte le informazioni riguardanti il veicolo durante il suo giro del tracciato. Questo file offre inoltre la pos-

sibilità di visualizzare un'animazione della simulazione effettuata, importandolo all'interno del software dedicato *VI-Animator*.

5.3 SIMULAZIONI

In questa sezione vengono esposte le prestazioni dello schema di controllo basato su NMPC utilizzando il modello di veicolo proposto nel Capitolo 4, considerando un giro del tracciato di Figura 5.1. Tutte le simulazioni sono state eseguite su un PC WIN10 x64 Intel® Core™ i7-8565U CPU @ 1.80GHz-1.99GHz.

Inizialmente il veicolo virtuale verrà controllato grazie all'implementazione in MATMPC del modello proposto nel corso del capitolo 4 (MATMPC_LV) e verrà poi confrontato con l'implementazione del modello allo stato dell'arte descritto nella sezione 3.2 (MATMPC_MB); entrambe le simulazioni seguiranno un riferimento di velocità aumentato dell'1%.

Successivamente il controllore verrà testato su un riferimento di velocità aumentato del 2% e verrà confrontato con quello all'1%.

5.3.1 CONFRONTO CON MATMPC_MB

La prima simulazione ha lo scopo di testare il controllo del veicolo utilizzando l'implementazione in MATMPC, ed allo stesso tempo verificare se con il controllore implementato sia possibile ottenere delle prestazioni comparabili a quelle ottenute da Michele Bevilacqua nel suo lavoro di tesi [14].

Entrambe le simulazioni utilizzano un profilo di velocità aumentato dell'1% con le seguenti

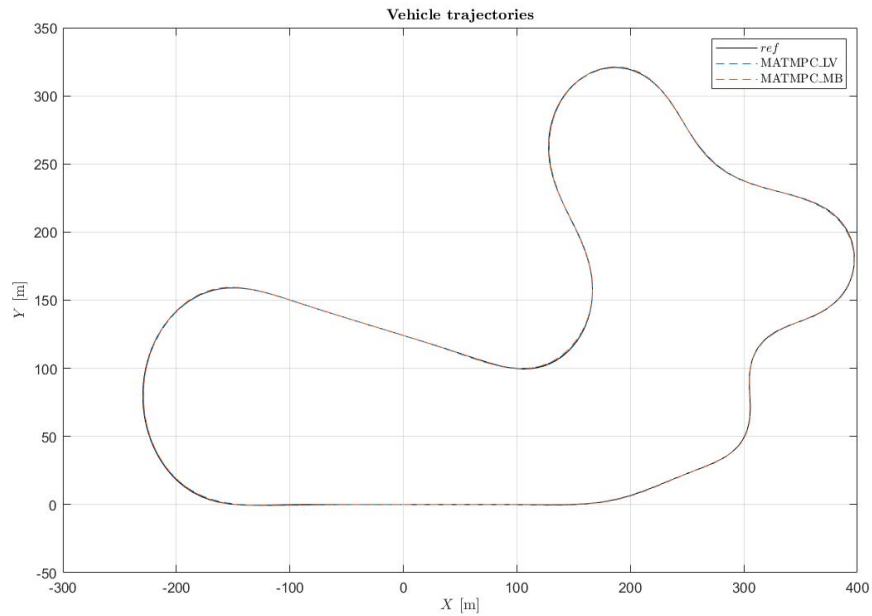


Figura 5.3: Confronto tra i tracciati percorsi da MATMPC_MB e MATMPC_LV

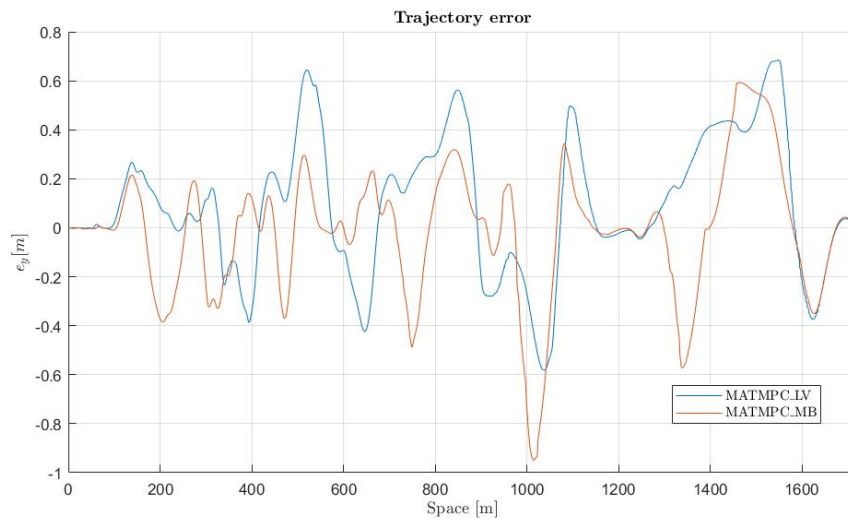


Figura 5.4: Errore di traiettoria del veicolo

In Figura 5.3 e 5.4 sono mostrate rispettivamente le traiettorie percorse insieme al riferimento generato da VI-CRT e gli errori di traiettoria dei due driver. Dai grafici si può notare come entrambi riescano a seguire bene la traiettoria di riferimento, con l'errore di traiettoria che è contenuto tra i valori $-1m$ e $+0.7m$.

In Figura 5.5 vengono mostrati i profili di velocità dei 2 veicoli. Si può notare come entrambi i driver riescano a mantenere un profilo di velocità molto vicino al riferimento, ad eccezione dei tratti in cui quest'ultimo presenta dei punti angolosi, poichè il peso sulle derivate degli ingressi definito in Q porta a questo comportamento; inoltre MATMPC_MB tende a rallentare troppo ad ogni frenata.

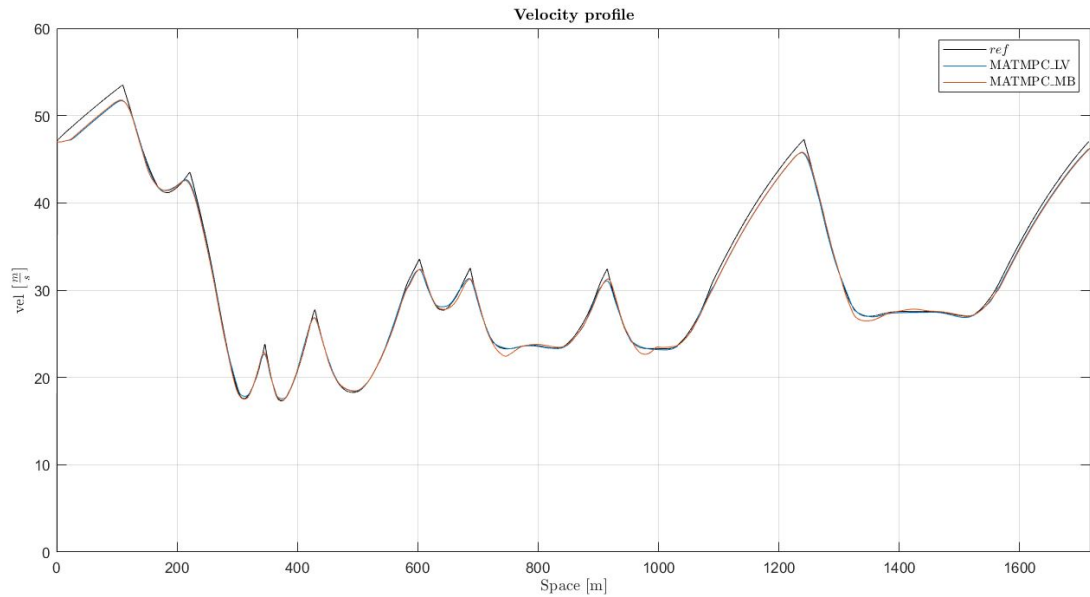


Figura 5.5: Confronto tra i profili di velocità

In Figura 5.6 vengono mostrati gli ingressi $\delta_f, \beta_t, \beta_b$ generati dai due driver: in blu quelli relativi a MATMPC_LV e in arancione quelli relativi a MATMPC_MB. Si può notare che l'azione di sterzo del MATMPC_LV è più composta (anche se presenta molte piccole oscillazioni), mentre MATMPC_MB effettua delle sterzate notevoli in un paio di casi.

Particolarmente interessante è la gestione delle accelerazioni/frenate da parte del controllore MATMPC_MB, quando il veicolo sta accelerando si ha una frenata praticamente nulla, al contrario il driver MATMPC_LV molto spesso si trova ad applicare delle frenate mentre è in fase di accelerazione.

Ciò è dovuto all'imprecisione del calcolo delle forze laterali implementate nel modello di MATMPC_MB che porta spesso il veicolo ad affrontare situazioni in cui arriva con una velocità troppo alta in curva senza riuscire a seguire la traiettoria di riferimento, di conseguenza tende a frenare per provare ad aggiustare la traiettoria.

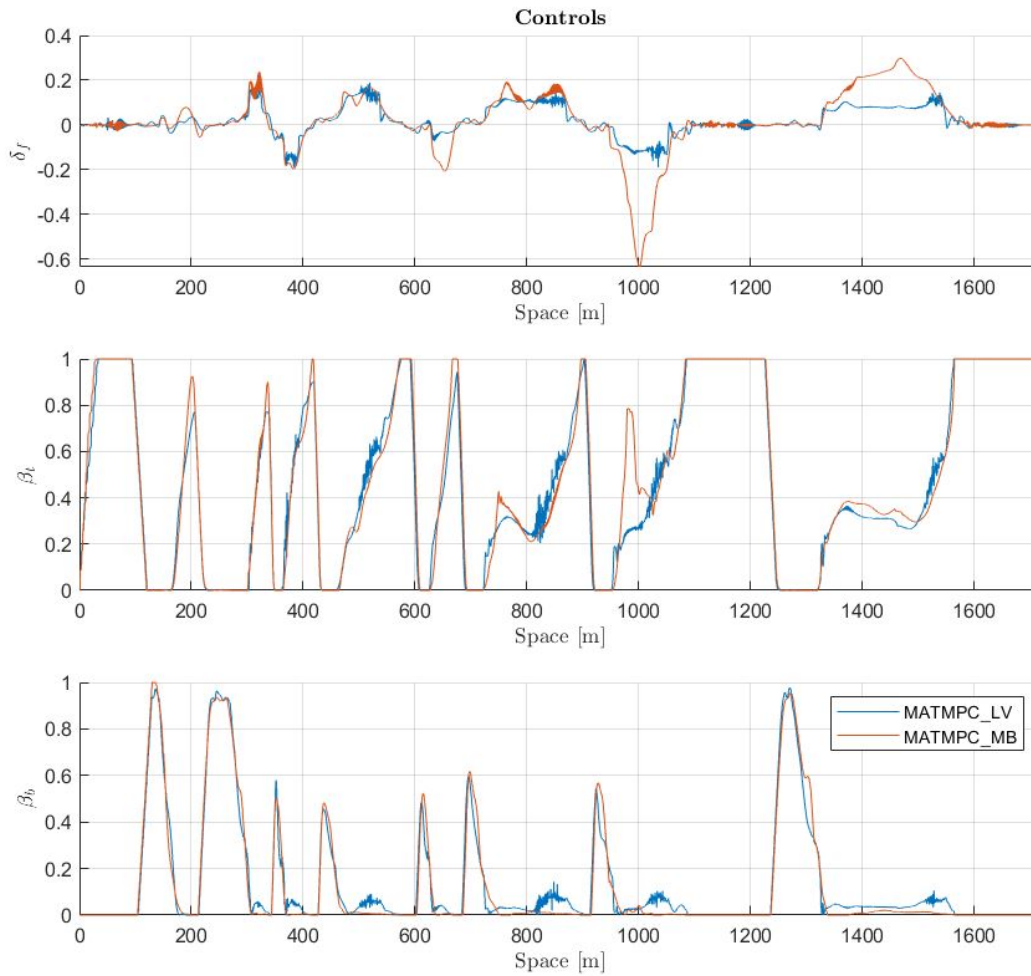


Figura 5.6: Confronto tra i controlli generati dai due driver

In Figura 5.7 viene mostrato l'angolo di slittamento del veicolo controllato da MATMPC_LV (in blu) e quello controllato da MATMPC_MB (in arancione). Si nota che il secondo presenta dei picchi più elevati in valore assoluto, segno che in questo caso il veicolo esibisce maggiormente la tendenza a sbandare.

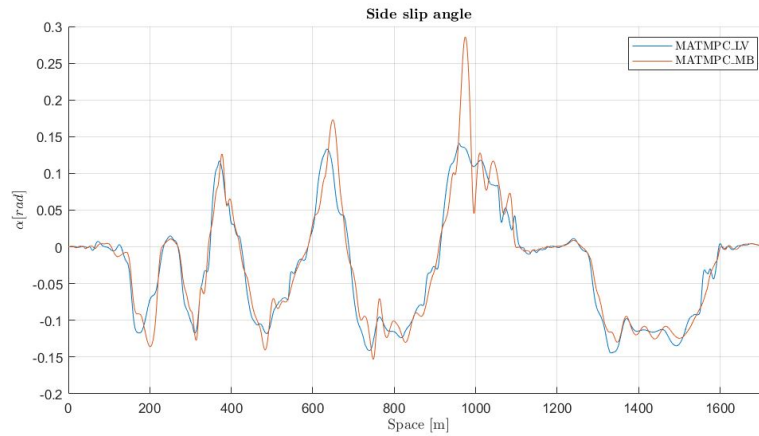


Figura 5.7: Angolo di slittamento del veicolo

In Figura 5.8 sono messe a confronto le forze stimate dal controllore MATMPC_LV (in rosso) con quelle di VI-CRT (in blu), assieme al fascio di ellissi normalizzate ricavate in 4.3.3, invece in verde è rappresentata l'ellisse nominale con raggio esteso a 1.1. Si può notare come l'imprecisione del calcolo della forza longitudinale porti il controllore a optare per una guida più conservativa, soprattutto in fase di frenata.

In Figura 5.9 viene mostrata la variabile di controllo ε dell'eq. 5.2; è interessante vedere come il controllore spesso decida di aumentare il raggio dell'ellissoide nominale.

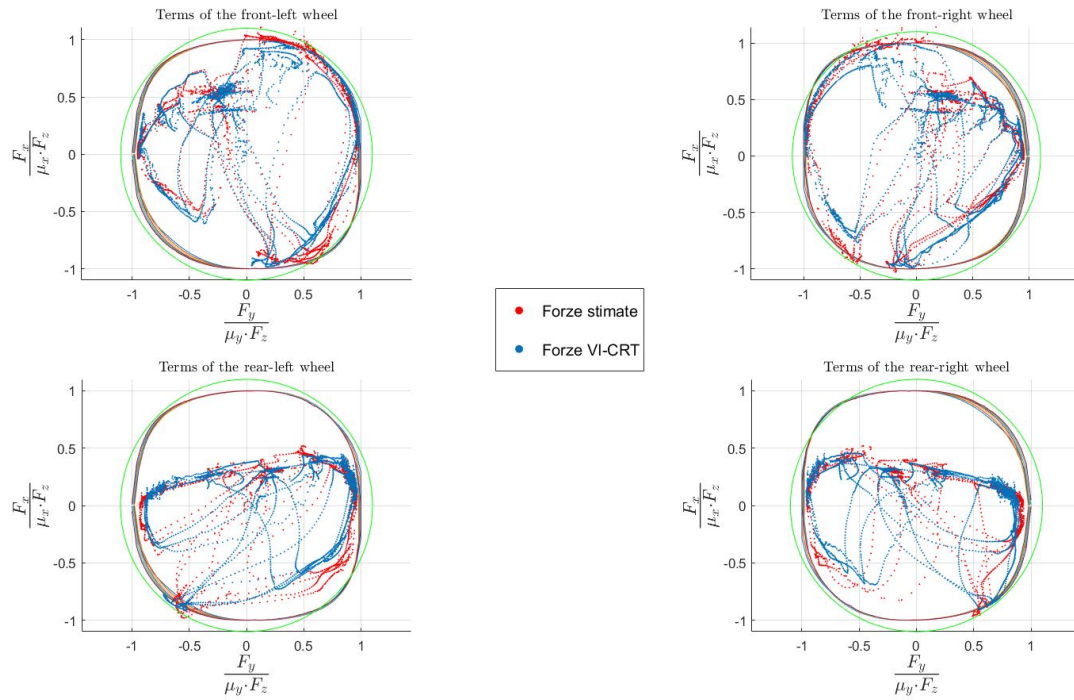


Figura 5.8: Ellissoidi di aderenza con forze stimate e reali

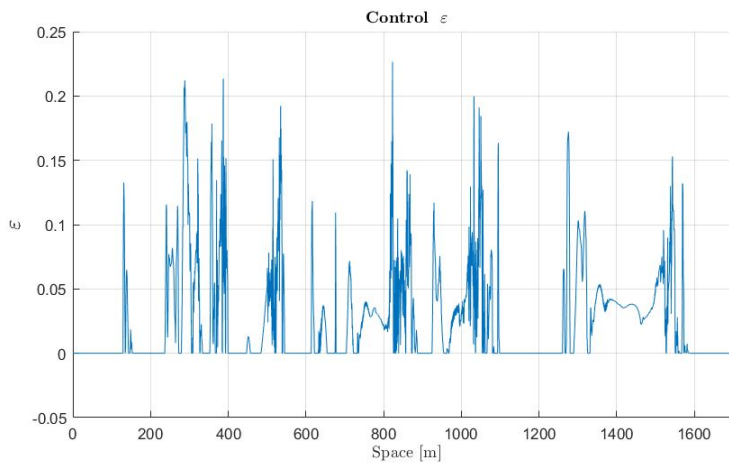


Figura 5.9: Variabile di controllo ε

Complessivamente, il tracciato viene percorso a velocità media leggermente maggiore dal driver MATMPC_LV, impiegando un tempo totale $T_{\text{MATMPC_LV}} = 59.64s$, mentre MATMPC_MB impiega un tempo totale $T_{\text{MATMPC_MB}} = 59.86s$; questa differenza di $0.22s$ corrisponde ad un distacco di circa $9m$, e sta a significare che il controllo basato su MATMPC_LV

di velocità aumentato del 102% (MATMPC_102) e la cosimulazione effettuata utilizzando lo stesso driver con profilo di velocità del 101% (MATMPC_101) è mostrato in Figura 5.10 e 5.11.

Come si può vedere MATMPC_102 segue una traiettoria leggermente diversa da quella di riferimento, infatti per tenere un profilo di velocità elevato il controllore tende a tagliare nelle curve maggiormente rispetto alla cosimulazione MATMPC_101.

Per quanto riguarda i controlli sono molto simili, si nota però una maggiore difficoltà da parte di MATMPC_102 di usare in maniera esclusiva l'acceleratore e il freno.

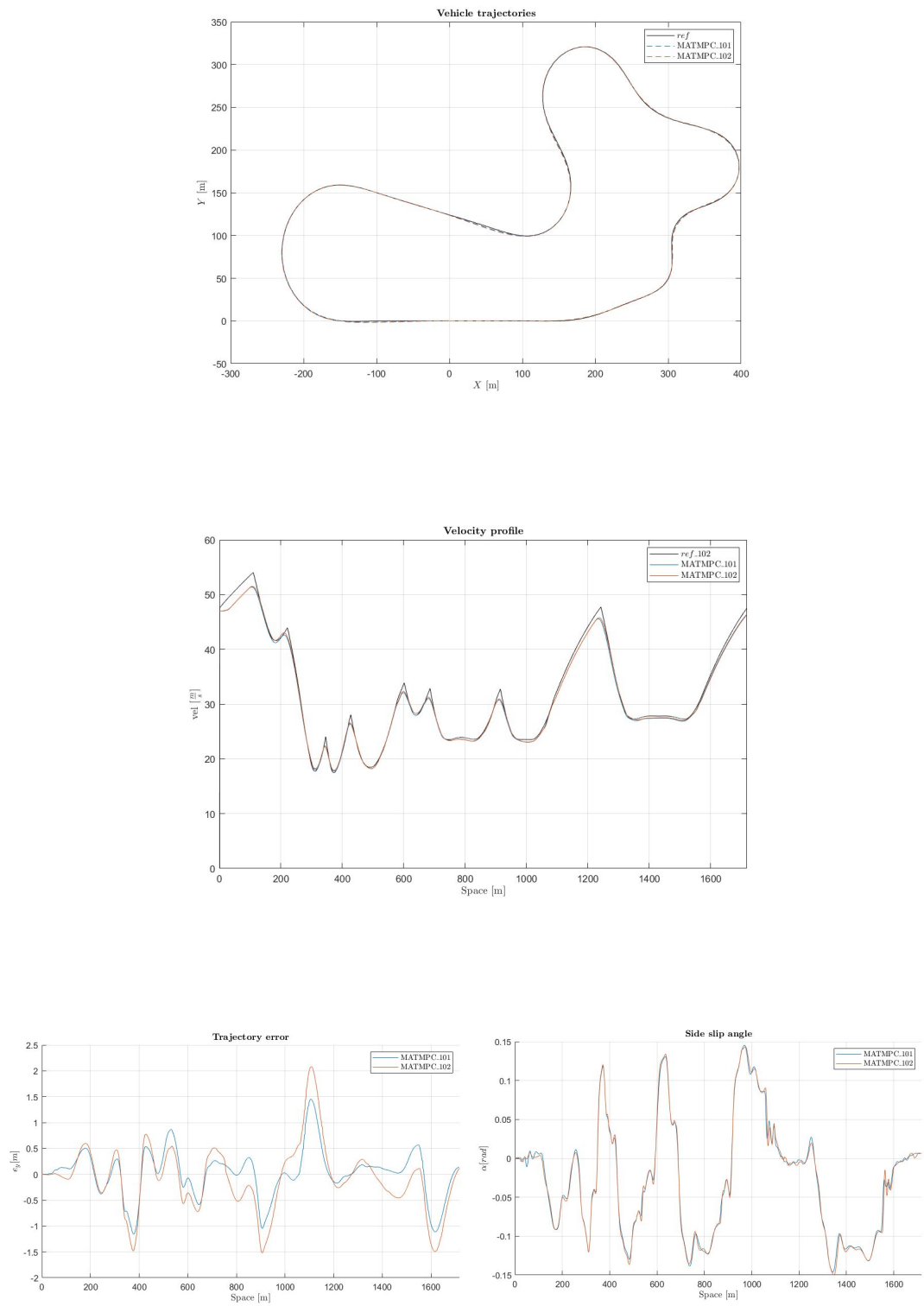


Figura 5.10: Confronto della traiettoria, velocità, errore della traiettoria e angolo di slittamento del veicolo

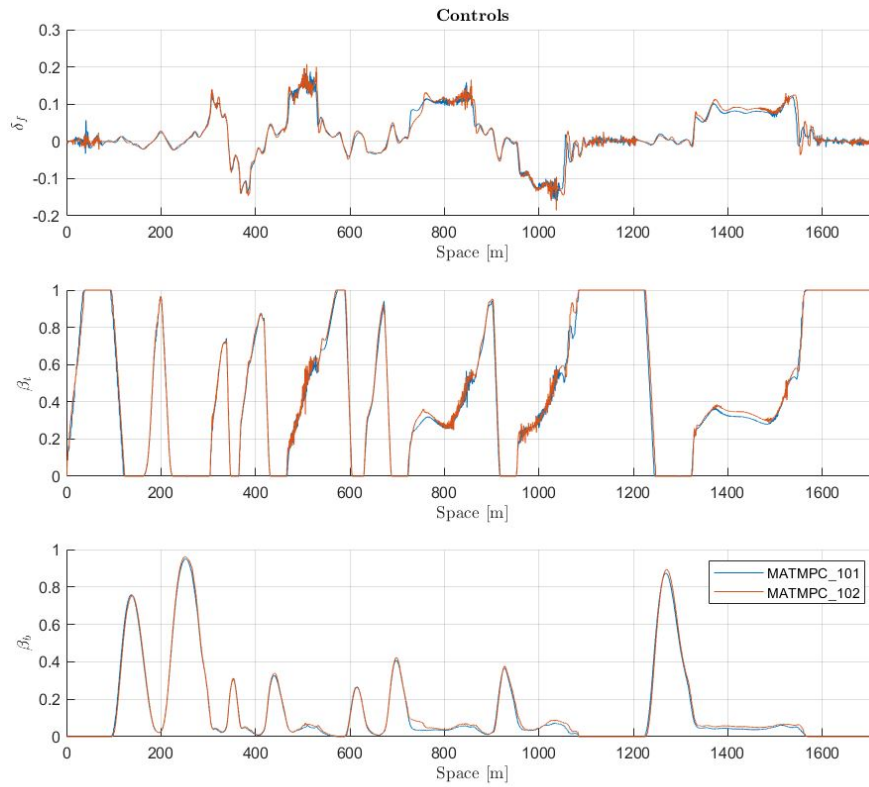


Figura 5.11: Confronto tra i controlli generati dai 2 driver

In Figura 5.12 e 5.13 viene mostrata l'analisi degli ellissoidi di aderenza delle 2 cosimulazioni. Si può vedere che anche per quanto riguarda l'applicazione delle forze laterali e longitudinali i due controllori agiscono in maniera analoga, anche se in realtà il driver MATMPC_102 tende ad aumentare di più la variabile di controllo ε per poter erogare più forza, come si può osservare in Figura 5.14.

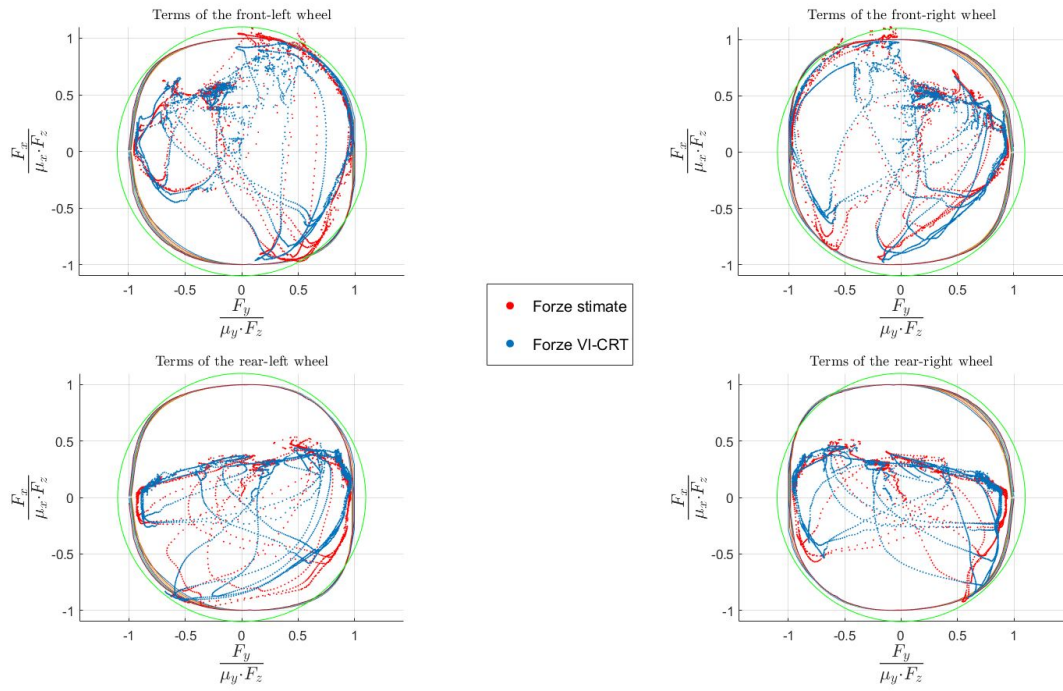


Figura 5.12: Ellissoidi di aderenza con forze stimate e reali per la simulazione MATMPC_102

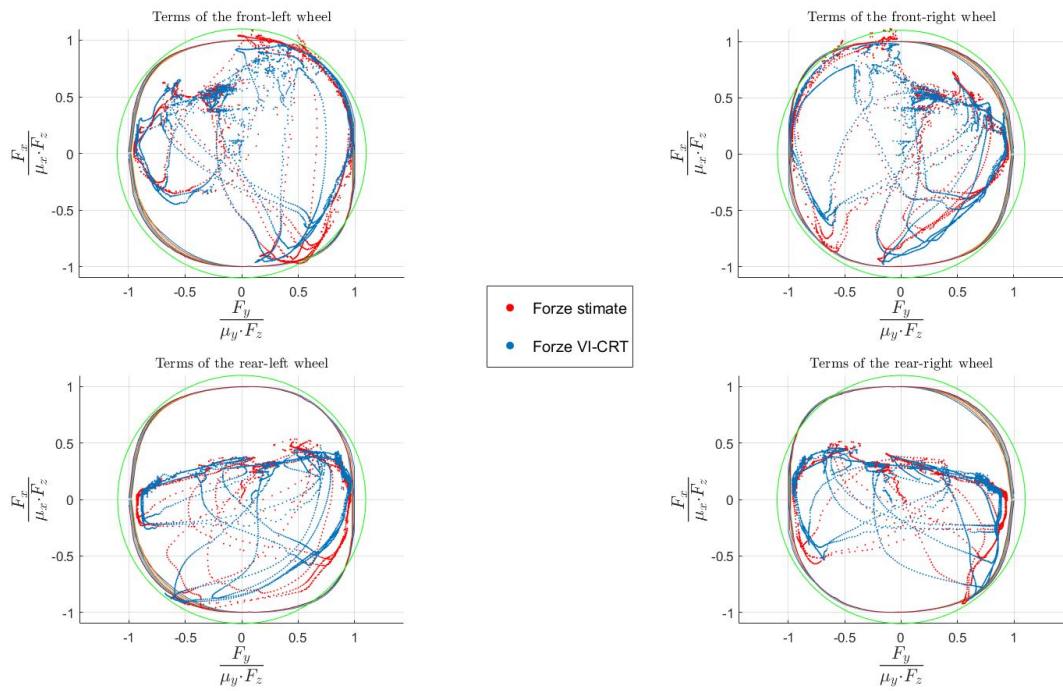


Figura 5.13: Ellissoidi di aderenza con forze stimate e reali per la simulazione MATMPC_101

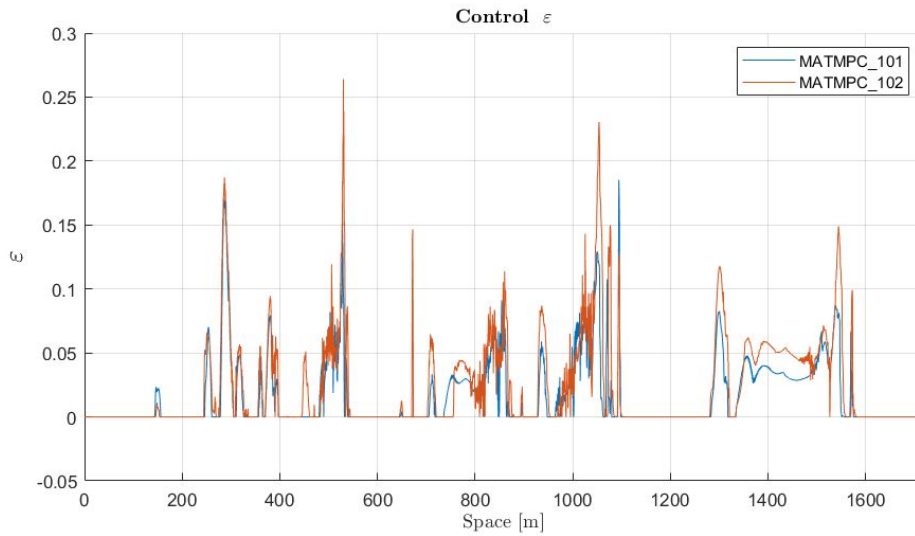


Figura 5.14: Confronto tra le variabili di controllo ε

In entrambi i casi i driver riescono a migliorare ulteriormente il tempo sul giro, impiegando $T_{\text{MATMPC}_{101}} = 59.561s$ e $T_{\text{MATMPC}_{102}} = 59.381s$.

Infine vengono riportati i tempi di calcolo del solver per arrivare alla soluzione del problema di ottimizzazione, escludendo la prima iterazione in cui avviene l'inizializzazione.

I tempi di calcolo medio e massimo risultano essere

- $T_{\text{MATMPC}_{101}}^{\text{mean}} = 39.5ms$,
- $T_{\text{MATMPC}_{101}}^{\text{max}} = 95.1ms$,
- $T_{\text{MATMPC}_{102}}^{\text{mean}} = 48.7ms$,
- $T_{\text{MATMPC}_{102}}^{\text{max}} = 128.6ms$.

Essendo la frequenza di controllo $f = 50$ Hz (ovvero la chiamata al solver avviene ogni $T = 20ms$), i controlli utilizzati per queste 2 cosimulazioni non permettono di essere usati in un contesto *real-time*.

Con l'obiettivo di ottenere un controllo utilizzabile in un contesto *real-time*, è stata eseguita una cosimulazione con i medesimi pesi di MATMPC_102, con riferimento di velocità diminuito al 101.5% e numero di punti di shooting ridotto a $N = 32$ (quindi un minore orizzonte di predizione). I tempi di calcolo richiesti dal controllore sono di $T_{\text{solv}}^{\text{mean}} = 17.6ms$ e $T_{\text{solv}}^{\text{max}} = 37.3ms$, ed il tracciato viene percorso in $T = 59.581$.

Perciò i tempi di calcolo rilevati dimostrano la possibilità di utilizzo di tale schema di controllo anche in un contesto *real-time*.

Nel complesso, il controllore con l'introduzione della dinamica combinata delle forze laterali e longitudinali riesce a generare degli ingressi di controllo più appropriati, volti ad evitare situazioni indesiderate (come per esempio una sbandata) ed a migliorare le performance in termini di tempo sul giro.

6

Conclusioni

In questa tesi è stato presentato un modello di veicolo da utilizzare in un contesto di controllo predittivo non lineare. Dopo la descrizione del modello di veicolo allo stato dell'arte, la procedura di validazione esposta ha fatto emergere alcune criticità rilevanti dall'utilizzo di tale modello. L'introduzione di elementi aggiuntivi ha permesso di ottenere un calcolo della dinamica laterale e longitudinale più accurato, permettendo di inserire efficacemente nel modello la dinamica combinata delle forze laterali e longitudinali delle gomme.

Il modello proposto è stato utilizzato per il controllo di un veicolo virtuale grazie all'implementazione in MATMPC, e le simulazioni effettuate mostrano l'efficacia dell'azione di controllo anche in condizioni di guida al limite.

Dal confronto tra MATMPC_MB e MATMPC_LV emerge come quest'ultimo riesca a seguire un riferimento di velocità in maniera più composta, riuscendo ad ottenere un tempo sul giro inferiore.

Le tempistiche computazionali compatibili al controllo *real-time* dimostrano l'effettiva realizzabilità del driver nel contesto della guida autonoma.

Questo progetto presenta numerosi aspetti che meriterebbero di essere studiati in maniera più approfondita, in particolare si fa riferimento a:

- introduzione di un modello di differenziale per ripartire la coppia tra ruote destre e sinistre;
- analisi della robustezza dello schema di controllo, testandolo su veicoli differenti;

- utilizzo di diverse implementazioni dell'ottimizzatore (ad es. hpipm) con tempistiche più rapide;
- implementazione di un algoritmo di *shifting*, allo scopo di velocizzare l'ottimizzazione;
- calcolo più preciso degli angoli di toe.



Riformulazione spaziale della dinamica

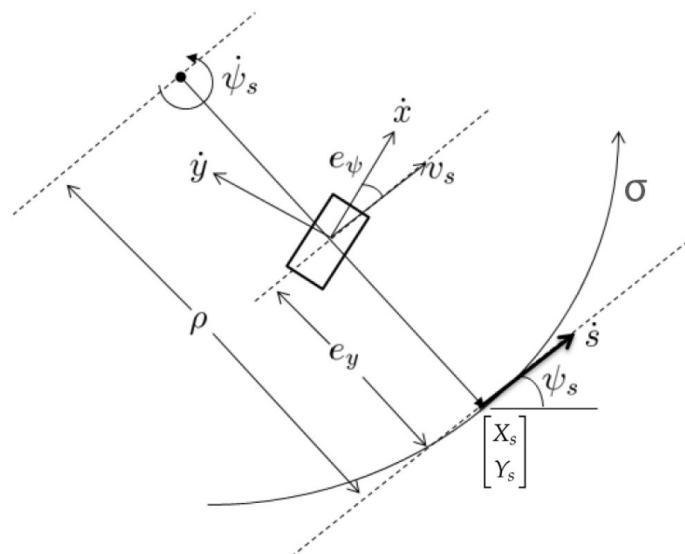


Figura A.1: Sistema definito da coordinate curvilinee

La formulazione temporale della dinamica impone di utilizzare il riferimento di traiettoria anch'esso dipendente dal tempo, questo significherebbe impostare una velocità ben definita di percorrenza del tracciato. Nelle situazioni limite in cui il sistema deve funzionare, è invece necessario lasciare al controllore una certa libertà nella scelta della velocità massima a cui

percorrere la traiettoria per non uscire dal percorso. per questo motivo la dinamica è stata riformulata nella variabile indipendente s , che è la lunghezza d'arco della proiezione delle coordinate $[X, Y]^T$ sulla curva σ , cioè il riferimento di traiettoria.

In questo modo, indicati con $[X_s, Y_s]^T$ e ψ_s la posizione e l'orientazione del punto di riferimento corrente sulla traiettoria in s , è possibile definire gli errori di inseguimento di orientazione e posizione

$$\begin{aligned} e_\psi &= \psi - \psi_s \\ e_y &= \pm \left\| \begin{bmatrix} X \\ Y \end{bmatrix} - \begin{bmatrix} X_s \\ Y_s \end{bmatrix} \right\|_2 \end{aligned} \quad (\text{A.1})$$

dove, nella definizione di e_y , il segno $+$ indica che la posizione attuale è alla sinistra della curva di riferimento, mentre il segno $-$ indica che è alla destra.

Inoltre, con riferimento alla Figura A.1, si possono ricavare le equazioni cinematiche per la proiezione della velocità lungo la direzione della traiettoria

$$\begin{aligned} v_s &= \dot{x} \cos(e_\psi) - \dot{y} \sin(e_\psi) \\ v_s &= (\rho - e_y) \dot{\psi}_s \end{aligned} \quad (\text{A.2})$$

dove ρ è il raggio di curvatura della curva σ . Da queste si può ottenere la velocità del veicolo lungo la traiettoria

$$\begin{aligned} \dot{s} &= \frac{ds}{dt} \\ &= \rho \dot{\psi}_s \\ &= \frac{\rho}{\rho - e_y} (\dot{x} \cos(e_\psi) - \dot{y} \sin(e_\psi)) \\ &= \frac{1}{1 - k e_y} (\dot{x} \cos(e_\psi) - \dot{y} \sin(e_\psi)) \end{aligned} \quad (\text{A.3})$$

dove, nell'ultimo passaggio, è stata utilizzata la relazione tra curvatura e raggio di curvatura $k = \rho^{-1}$.

In particolare, data una curva parametrizzata nella variabile s , $\sigma(s) = (x(s), y(s))$, la curva-

tura k è definita localmente con la formula

$$k = \frac{x'y'' - y'x''}{(x'^2 + y'^2)^{\frac{3}{2}}} \quad (\text{A.4})$$

dove $f' = \frac{df(s)}{ds}$ e $f'' = \frac{d^2f(s)}{ds^2}$.

Nello stato del sistema si possono quindi introdurre le dinamiche degli errori di inseguimento

$$\begin{aligned} \dot{e}_\psi &= \dot{\psi} - k \dot{s} \\ \dot{e}_y &= \dot{x} \cos(e_\psi) - \dot{y} \sin(e_\psi) \end{aligned} \quad (\text{A.5})$$

Il calcolo del vettore $\dot{\xi}^s$ si conclude, supponendo $\dot{s} \neq 0$ in ogni istante, attraverso la regola di derivazione

$$\dot{\xi}^s = \frac{d\xi}{ds} = \frac{d\xi}{dt} \frac{dt}{ds} = \frac{d\xi}{dt} \frac{1}{\dot{s}} = \frac{\dot{\xi}}{\dot{s}} \quad (\text{A.6})$$

che permette di descrivere completamente la dinamica del veicolo in una formulazione dipendente dallo spazio invece che dal tempo.

Bibliografia

- [1] A. Carvalho, Y. Gao, A. Gray, H. E. Tseng, and F. Borrelli, “Predictive control of an autonomous ground vehicle using an iterative linearization approach,” in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. IEEE, 2013, pp. 2335–2340.
- [2] D. Watzenig and M. Horn, “Automated driving: safer and more efficient future driving,” 2016.
- [3] Y. Chen, “Algorithms and applications for nonlinear model predictive control with long prediction horizon,” 2018.
- [4] M. Diehl, H. G. Bock, J. P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer, “Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations,” *Journal of Process Control*, vol. 12, no. 4, pp. 577–585, 2002.
- [5] S.-P. Han, “Superlinearly convergent variable metric algorithms for general nonlinear programming problems,” *Mathematical Programming*, vol. 11, no. 1, pp. 263–282, 1976.
- [6] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, “qpodes: A parametric active-set algorithm for quadratic programming,” *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.
- [7] J. Andersson, “A general-purpose software framework for dynamic optimization,” Ph.D. dissertation, PhD thesis, Arenberg Doctoral School, KU Leuven, Department of Electrical ..., 2013.
- [8] D. E. Smith and J. M. Starkey, “Effects of model complexity on the performance of automated vehicle steering controllers: Model development, validation and comparison,” *Vehicle System Dynamics*, vol. 24, no. 2, pp. 163–181, 1995.

- [9] R. Lot and N. Dal Bianco, “The significance of high-order dynamics in lap time simulations,” 2015.
- [10] H. Ren, T. Shim, J. Ryu, and S. Chen, “Development of effective bicycle model for wide ranges of vehicle operations,” SAE Technical Paper, Tech. Rep., 2014.
- [11] C. Ghike and T. Shim, “14 degree-of-freedom vehicle model for roll dynamics study,” SAE Technical Paper, Tech. Rep., 2006.
- [12] I. Youn, E. Youn, M. A. Khan, L. Wu, and M. Tomizuka, “Combined effect of electronic stability control and active tilting control based on full-car nonlinear model,” in *The Dynamics of Vehicles on Roads and Tracks: Proceedings of the 24th Symposium of the International Association for Vehicle System Dynamics (IAVSD 2015)*, Graz, Austria, 17-21 August 2015. CRC Press, 2016, p. 345.
- [13] E. Picotti, “Controllo predittivo non lineare di un veicolo virtuale,” *Master thesis in Automation Engineer, University of Padova*, 2018.
- [14] M. Bevilacqua, “Sviluppo di un modello per il controllo predittivo non lineare di un veicolo virtuale,” *Master thesis in Automation Engineer, University of Padova*, 2019.
- [15] H. Pacejka, *Tire and vehicle dynamics*. Elsevier, 2005.
- [16] R. Brach and M. Brach, “The tire-force ellipse (friction ellipse) and tire characteristics,” SAE Technical Paper, Tech. Rep., 2011.
- [17] Y. Gao, A. Gray, J. V. Frasch, T. Lin, E. Tseng, J. K. Hedrick, and F. Borrelli, “Spatial predictive control for agile semi-autonomous ground vehicles,” in *Proceedings of the 11th international symposium on advanced vehicle control*, no. 2, 2012, pp. 1–6.
- [18] J. V. Frasch, A. Gray, M. Zanon, H. J. Ferreau, S. Sager, F. Borrelli, and M. Diehl, “An auto-generated nonlinear mpc algorithm for real-time obstacle avoidance of ground vehicles,” in *2013 European Control Conference (ECC)*. IEEE, 2013, pp. 4136–4141.