

Collegio di ingegneria meccanica e aerospaziale

Corso di Laurea di II livello in Ingegneria Meccanica



**Politecnico
di Torino**

Tesi di Laurea Magistrale

**Simulazione di scenari di guida complessi per lo
studio di un sistema ADAS di sorpasso automatico**

Relatori

Prof. Enrico GALVAGNO

Ing. Luca ZERBATO

Candidato

Luca VELLA

Matricola S295393

Luglio 2024

Sommario

Questa tesi di laurea magistrale presenta uno studio incentrato su un sistema assistenziale alla guida in grado di analizzare lo scenario circostante e di sviluppare una traiettoria utile ad eseguire una manovra di sorpasso, qualora questa venga ritenuta possibile e necessaria al raggiungimento di un determinato obiettivo. Il sorpasso è una fase della guida associata ad un elevato rischio per il conducente; conseguentemente, risulta essere opportuno sviluppare sistemi in grado di ridurre errori ed incertezze del pilota. Il tutto è stato prodotto all'interno di un contesto in rapido sviluppo come quello dell'automatizzazione dei vicoli; che, a partire dalla fine del 2023, ha permesso l'introduzione di veicoli con livello di automazione SAE superiore al livello 2 all'interno del parco auto circolante.

In questo lavoro di tesi sono state analizzate le caratteristiche di cui deve essere necessariamente fornito un sistema come quello descritto; all'analisi teorica di ogni suo componente ne è stata affiancata una possibile realizzazione all'interno di un opportuno ambiente di simulazione. Con il fine di presentare un sistema in grado di fornire una risposta solida, indipendentemente dallo scenario in cui viene utilizzato, è stato introdotto un set di casistiche in grado di sollevare potenziali criticità del sistema stesso; analizzando in tal modo le possibili soluzioni teoriche e pratiche.

Summary

This Master of Science thesis presents a study focused on a driver assistance system that is able to analyse the surrounding scenario and develop a useful trajectory to perform an overtaking manoeuvre, if this is considered possible and necessary to achieve a given goal. The overtaking manoeuvre is a phase of driving associated with a high risk for the driver; consequently, it is appropriate to develop systems capable of reducing driver errors and uncertainty. This was produced within the rapidly developing context of automotive automation; which, since the end of 2023, has allowed the introduction of vehicles with an SAE level of automation higher than level 2 within the current car population.

In this thesis work were analysed the characteristics with which a system such as the one described must necessarily be equipped; the theoretical analysis of each of its components was accompanied by a possible realisation within an appropriate simulation environment. With the aim of presenting a system capable of providing a solid response, regardless of the scenario in which it is used, a set of cases was introduced to raise potential criticalities of the system itself; analysing possible theoretical and practical solutions.

Indice

Elenco delle tabelle	X
Elenco delle figure	XII
Glossario	XXI
1 Introduzione	1
1.1 Tecnologie di guida automatica	2
1.1.1 Principali sistemi ADAS in dotazione	2
1.1.2 Sensoristica dei veicolo autonomi	6
1.2 Livelli di automazione SAE	9
1.3 Contesto normativo ed aspetto etico	10
1.3.1 Normativa UE - Introduzione nuovi ADAS obbligatori	10
1.3.2 Decreto Smart Road	11
1.3.3 Aspetto etico	12
1.4 Struttura della tesi	13
2 Pianificazione della traiettoria	15
2.1 Generazione della traiettoria	16
2.1.1 Metodologie di generazione della traiettoria	16
2.1.2 Sistema di riferimento Frenet	21
2.1.3 Strumenti di generazione della traiettoria in ambiente di simulazione	23
2.2 Valutazione di fattibilità delle traiettorie alternative	29

2.2.1	Analisi non lineare della stabilità del veicolo	30
2.3	Valutazione sulle possibili collisioni delle traiettorie alternative . . .	37
2.3.1	Metodo capsule 2D	37
3	Ambiente di simulazione	47
3.1	Modello Simulink del sistema assistenziale	48
3.1.1	Highway Lane Change Planner Test Bench	49
3.1.2	Scenario and Enviroment Subsystem	53
3.1.3	Planner Configuration Parameters Subsystem	56
3.1.4	Highway Lane Change Planner Subsystem	59
3.1.5	Pack Actor	93
3.1.6	Metrics Assessment	94
3.1.7	Visualization	100
3.2	Modello di veicolo e controllore	104
3.2.1	Architettura del modello	105
3.2.2	Lane Change Controller	108
3.2.3	Modello di veicolo	129
3.2.4	Parametri di configurazione veicolo	136
3.2.5	Caratterizzazione del modello di veicolo	138
3.3	Scenario della simulazione	151
3.3.1	Driving Scenario Designer	151
3.4	Segnali ottenuti da simulazione	155
3.4.1	Richiami di dinamica del veicolo	155
3.4.2	Segnali di analisi strategia del pianificatore	158
4	Casi studio e risultati ottenuti	165
4.1	Introduzione dei casi studio	166
4.2	Caso 1 - Singolo cambio di corsia	169
4.2.1	Risultati simulazione - caso 1	171
4.3	Caso 2 - Sorpasso con veicolo in senso di marcia opposto	186
4.3.1	Risultati simulazione - caso 2	189
4.3.2	Adattamento del sistema alle criticità presentate - Caso 2 . .	200

4.3.3	Risultati simulazione - caso 2 con risoluzione criticità	207
4.4	Caso 3 - Sorpasso di una fila di veicoli	217
4.4.1	Risultati simulazione - caso 3	221
4.4.2	Adattamento del sistema alle criticità presentate - Caso 3 . .	230
4.4.3	Risultati simulazione - caso 3 con risoluzione criticità	235
4.5	Caso 4 - Logiche di scelta corsia per il veicolo ego	241
4.5.1	Risultati simulazione - caso 4	244
4.5.2	Adattamento del sistema alle criticità presentate - Caso 4 . .	251
4.5.3	Risultati simulazione - caso 4 con risoluzione criticità	258
4.6	Caso 5 - Modalità di cambio corsia rapido	263
4.6.1	Introduzione strategia di cambio corsia rapido	265
4.6.2	Risultati simulazione - caso 5	266
4.6.3	Implementazione verifica di stabilità	272
4.6.4	Risultati simulazione caso 5 con risoluzione criticità	287
4.7	Simulazione test del controllore e del modello di veicolo	295
4.7.1	Test 1 - Verifica funzionamento modello complessivo con scenario a bassa richiesta di prestazioni	296
4.7.2	Test 2 - Introduzione caratteristiche rigidzze in deriva nel controllore MPC	300
4.8	Confronto tra sistema di attuazione dello sterzo <i>slow/fast</i>	303
5	Conclusioni	309
5.1	Futuri sviluppi	310
A	Functions	311
	Bibliografia	349

Elenco delle tabelle

2.1	Stati terminali - scenario "sorpasso del veicolo <i>lead</i> ".	27
2.2	Coordinata <i>d</i> singole corsie - scenario "sorpasso del veicolo <i>lead</i> ". . .	27
2.3	Geometria capsule - Ex. Dynamic Capsule-based List	38
3.1	Planning Configuration Parameters values	58
3.2	Condizioni logiche per definizione <i>IsValid</i>	81
3.3	Parametro limite verifiche <i>Metrics Assessment</i>	99
3.4	Parametri di configurazione veicolo.	136
3.5	Parametri di configurazione manovra <i>ramp steer</i>	138
3.6	Parametri di configurazione manovra <i>step steer</i>	145
3.7	Condizioni logiche per definizione <i>Safe/Unsafe</i>	163
3.8	Limiti di default su TTC e distanza di sicurezza	164
4.1	Dati strada - scenario "Singolo cambio di corsia"	169
4.2	Dati veicoli - scenario "Singolo cambio di corsia"	169
4.3	Parametro di costo set traiettorie alternative ($t=2.1s$) - scenario "Singolo cambio di corsia"	175
4.4	Valori di picco grandezze veicolo <i>ego</i> - scenario "Singolo cambio di <i>corsia</i> "	178
4.5	Confronto valori di picco - scenario "Singolo cambio di corsia" . . .	185
4.6	Dati strada - scenario "Sorpasso con veicolo in senso di marcia opposto"	186
4.7	Dati veicoli - scenario "Sorpasso con veicolo in senso di marcia opposto"	186
4.8	Valori di picco grandezze veicolo <i>ego</i> - scenario "Sorpasso con veicolo <i>in senso di marcia opposto</i> "	191

4.9	Valori di picco grandezze veicolo <i>ego</i> - scenario " <i>Sorpasso con veicolo in senso di marcia opposto</i> "	209
4.10	Studio sensibilità $TTC_{oncoming}$ - scenario " <i>Sorpasso di una fila di veicoli</i> "	215
4.11	Dati strada - scenario " <i>Sorpasso di una fila di veicoli</i> "	217
4.12	Dati veicoli - scenario " <i>Sorpasso di una fila di veicoli</i> "	218
4.13	Valori di picco grandezze veicolo <i>ego</i> - scenario " <i>Sorpasso di una fila di veicoli</i> "	225
4.14	Valori di picco grandezze veicolo <i>ego</i> - scenario " <i>Sorpasso di una fila di veicoli</i> "	235
4.15	Dati strada - " <i>Logiche di scelta corsia per il veicolo ego</i> "	241
4.16	Dati veicoli - scenario " <i>Sorpasso di una fila di veicoli</i> "	242
4.17	Valori di picco grandezze veicolo <i>ego</i> - scenario " <i>Logiche di scelta corsia per il veicolo ego</i> "	246
4.18	Valori di picco grandezze veicolo <i>ego</i> - scenario " <i>Logiche di scelta corsia per il veicolo ego</i> "	258
4.19	Dati strada - scenario " <i>Modalità di cambio corsia rapido</i> ".	263
4.20	Dati veicoli - scenario " <i>Modalità di cambio corsia rapido</i> ".	264
4.21	Variazione del parametro " <i>timeWeight</i> ".	265
4.22	Valori di picco grandezze veicolo <i>ego</i> - scenario " <i>Modalità di cambio corsia rapido</i> ".	271
4.23	Confronto errori di traiettoria prima e dopo l'inserimento della verifica di stabilità - scenario " <i>Modalità di cambio corsia rapido</i> ". . .	289
4.24	Confronto richiesta di prestazioni prima e dopo l'inserimento della verifica di stabilità - scenario " <i>Modalità di cambio corsia rapido</i> ". . .	291
4.25	Confronto richiesta di prestazioni prima e dopo l'inserimento della verifica di stabilità - scenario " <i>Modalità di cambio corsia rapido</i> ". . .	292
4.26	Costante temporale adottata per funzione di trasferimento angolo di sterzo.	303

Elenco delle figure

1.1	Porsche LCA - rilevazione veicoli nei punti ciechi.	4
1.2	Sensori ADAS e Hardware informatico per una percezione a 360°.	6
2.1	Esempio curva Sigmoide.	19
2.2	Traiettorie possibili nel reticolo di stati o <i>State Lattice</i>	20
2.3	Generazione di una traiettoria con coordinate di Frénet	21
2.4	Traiettoria di riferimetno espressa mediante coordinate veicolo.	22
2.5	Traiettoria di riferimento espressa mediante coordinate di Frenét.	22
2.6	Stati iniziali $t = 0 s$ - scenario "sorpasso del veicolo <i>lead</i> ".	25
2.7	Set traiettorie alternative- scenario "sorpasso del veicolo <i>lead</i> ".	28
2.8	Piano delle fasi $\beta - \dot{\psi}$	32
2.9	Influenza della velocità del veicolo sul piano delle fasi $\beta - \dot{\psi}$	34
2.10	Influenza dell'angolo di sterzo del veicolo sul piano delle fasi $\beta - \dot{\psi}$. δ_f grafico a sinistra 0°, δ_f grafico a destra 5°.	34
2.11	Correlazione dell'accelerazione laterale a_y e dell'indice LTR con i punti di funzionamento riportati sul piano delle fasi $\beta - \dot{\psi}$	35
2.12	Geometria delle capsule 2D [20].	38
2.13	Evoluzione nel tempo degli stati dei veicoli e verifica sulle collisioni - scenario "sorpasso del veicolo <i>lead</i> ".	41
2.14	Evoluzione nel tempo degli stati dei veicoli e verifica sulle collisioni - scenario "sorpasso del veicolo <i>lead</i> con sorpasso".	43
2.15	Stati iniziali $t = 0 s$ - scenario "sorpasso con veicolo in senso di marcia opposto".	44

2.16	Evoluzione nel tempo degli stati dei veicoli e verifica sulle collisioni - scenario "sorpasso con veicolo in senso di marcia opposto".	44
3.1	Struttura generale Highway Lane Change Planner.	49
3.2	Diagramma di flusso funzionamento generale <i>Highway Lane Change Palmer model</i>	50
3.3	Scenario and Environment subsystem.	53
3.4	Scheda di configurazione blocco Scenario Reader.	54
3.5	Planner Configuration Parameters subsystem.	56
3.6	Highway Lane Change subsystem.	59
3.7	Diagramma di flusso Frenét State Converter	61
3.8	Most Important Objects [24].	62
3.9	Diagramma di flusso Find MIOs	64
3.10	Terminal State Sampler Subsystem.	65
3.11	Diagramma di flusso calcolo corsia preferenziale	67
3.12	Diagramma di flusso <i>function cruiseControlSampler</i>	70
3.13	Diagramma di flusso <i>function leadCarFollowingSampler</i>	71
3.14	Diagramma di flusso <i>function laneChangeSampler</i>	72
3.15	Diagramma di flusso calcolo traiettorie future veicoli MIO	74
3.16	Motion Planner Subsystem	75
3.17	Diagramma di flusso generazione traiettorie associate agli stati terminali	79
3.18	Validity Checker subsystem.	80
3.19	Diagramma di flusso verifica collisioni sulle traiettorie alternative	83
3.20	Rappresentazione processo di determinazione traiettoria di riferimento.	85
3.21	Rappresentazione grafica individuazione punto di riferimento.	88
3.22	Distanze tra punti della traiettoria di riferimento.	92
3.23	Pack Actor subsystem.	93
3.24	Metrics Assessment subsystem.	94
3.25	Collision Detection subsystem.	96
3.26	Visualization subsystem - <i>Lane Change Status Plot</i>	100
3.27	Visualization subsystem - <i>Lane Change Simulation Analysis</i>	102

3.28	Schema a blocchi modello con controllore	105
3.29	Struttura modello Simulink con controllore	107
3.30	Sottosistema Simulink controllore	108
3.31	Controllore path following [26].	109
3.32	Ciclo base di controllo [27].	111
3.33	Modello a blocchi funzione di trasferimento.	113
3.34	Modello a blocchi funzionamento modello di veicolo integrato.	116
3.35	Errori di posizionamento modello <i>single track</i>	120
3.36	Sistema di riferimento solidale al veicolo e solidale alla traiettoria di riferimento.	121
3.37	Andamento rigidzze in deriva degli assali.	125
3.38	Confronto traiettoria effettiva/riferimento ad elevate performance richieste.	127
3.39	Grandezze veicolo <i>ego</i> per traiettoria ad elevate performance richieste.	128
3.40	Errori controllore traiettoria ad elevate performance richieste.	128
3.41	<i>Vehicle Dynamics subsystem</i>	129
3.42	Modello di veicolo.	130
3.43	Diagramma di corpo libero piano X-Y.	131
3.44	Diagrammi di corpo libero piano Y-Z.	131
3.45	Andamento angoli di deriva α manovra <i>ramp steer</i>	138
3.46	Andamento forze laterali F_y manovra <i>ramp steer</i>	139
3.47	Andamento forze laterali F_y in funzione degli angoli di deriva α manovra <i>ramp steer</i>	139
3.48	Rigidezza in deriva manovra <i>ramp steer</i>	141
3.49	Caratteristiche di sovrasterzo e di assetto veicolo.	142
3.50	Forze longitudinali e verticali manovra <i>ramp steer</i>	143
3.51	Scorrimenti longitudinali pneumatici manovra <i>ramp steer</i>	143
3.52	Velocità e accelerazione laterale veicolo manovra <i>ramp steer</i>	144
3.53	Andamento angoli di deriva α manovra <i>step steer</i>	145
3.54	Andamento forze laterali F_y manovra <i>step steer</i>	146
3.55	Andamento angolo di assetto e velocità di imbardata manovra <i>step steer</i>	146

3.56	Andamento angoli di deriva α manovra <i>step steer</i>	147
3.57	Andamento forze laterali F_y manovra <i>step steer</i>	148
3.58	Andamento angolo di assetto e velocità di imbardata manovra <i>step steer</i>	148
3.59	Posizione dei punti di equilibrio al variare della velocità $\delta_{SW} = 5\ deg$ - metodo piano delle fasi.	150
3.60	Velocità critica del veicolo al variare dell'angolo volante δ_{SW} - metodo piano delle fasi.	150
3.61	Interfaccia Driving Scenario Designer	151
3.62	Sovrapposizione traiettorie Driving Scenario Designer	153
3.63	Sistema di riferimento veicolo	155
3.64	Diagramma di flusso scelta <i>Safety gap</i>	158
3.65	Distanza relativa veicoli ego/MIO	159
3.66	Diagramma di flusso valutazione veicoli MIO	162
4.1	Scenario " <i>Singolo cambio di corsia</i> "	166
4.2	Scenario " <i>Sorpasso con veicolo in senso di marcia opposto</i> "	167
4.3	Scenario " <i>Sorpasso di una fila di veicoli</i> "	168
4.4	Stati iniziali $t=0s$ e collisione - scenario "Singolo cambio di corsia" .	170
4.5	Traiettoria veicolo <i>ego</i> - scenario "Singolo cambio di corsia"	171
4.6	Corsia veicolo <i>ego</i> - scenario "Singolo cambio di corsia"	172
4.7	Step traiettoria veicolo <i>ego</i> - scenario "Singolo cambio di corsia" . .	173
4.8	Analisi traiettorie future $t=2.1s$ - scenario "Singolo cambio di corsia"	174
4.9	Andamenti grandezze moto veicolo <i>ego</i> - scenario "Singolo cambio di corsia"	177
4.10	Accelerazione longitudinale veicolo <i>ego</i> - scenario "Singolo cambio di corsia"	178
4.11	Andamenti grandezze moto veicolo <i>ego</i> (2) - scenario "Singolo cambio di corsia"	179
4.12	Valutazione veicolo MIO - scenario "Singolo cambio di corsia"	180
4.13	Confronto tra stati terminali - scenario "Singolo cambio di corsia" .	183
4.14	Confronto modalità - scenario "Singolo cambio di corsia"	185

4.15	Stati iniziali $t=0s$ - scenario " <i>Sorpasso con veicolo in senso di marcia opposto</i> "	187
4.16	Traiettoria veicolo <i>ego</i> - scenario " <i>Sorpasso con veicolo in senso di marcia opposto</i> "	189
4.17	Frame traiettoria veicolo <i>ego</i> - scenario " <i>Sorpasso con veicolo in senso di marcia opposto</i> "	190
4.18	Andamenti grandezze moto veicolo <i>ego</i> - scenario " <i>Sorpasso con veicolo in senso di marcia opposto</i> "	191
4.19	Andamenti grandezze moto veicolo <i>ego</i> (2) - scenario " <i>Sorpasso con veicolo in senso di marcia opposto</i> "	192
4.20	Punti della traiettoria invalidanti i limiti imposti - scenario " <i>Sorpasso con veicolo in senso di marcia opposto</i> "	193
4.21	Status veicolo <i>ego</i> - scenario " <i>Sorpasso con veicolo in senso di marcia opposto</i> "	194
4.22	Corsia veicolo <i>ego</i> - scenario " <i>Sorpasso con veicolo in senso di marcia opposto</i> "	195
4.23	Valutazione veicoli MIO - scenario " <i>Sorpasso con veicolo in senso di marcia opposto</i> "	196
4.24	Diagramma di flusso funzione " <i>oncoming Check</i> " - scenario " <i>Sorpasso con veicolo in senso di marcia opposto</i> "	203
4.25	Traiettoria veicolo <i>ego</i> - scenario " <i>Sorpasso con veicolo in senso di marcia opposto</i> "	207
4.26	Frame traiettoria veicolo <i>ego</i> - scenario " <i>Sorpasso con veicolo in senso di marcia opposto</i> "	208
4.27	Andamenti grandezze moto veicolo <i>ego</i> - scenario " <i>Sorpasso con veicolo in senso di marcia opposto</i> "	209
4.28	Andamenti grandezze moto veicolo <i>ego</i> (2) - scenario " <i>Sorpasso con veicolo in senso di marcia opposto</i> "	210
4.29	Corsia veicolo <i>ego</i> - scenario " <i>Sorpasso con veicolo in senso di marcia opposto</i> "	211
4.30	Valutazione veicoli MIO- scenario " <i>Sorpasso con veicolo in senso di marcia opposto</i> "	212

4.31	Stati iniziali $t=0s$ - scenario " <i>Sorpasso di una fila di veicoli</i> "	214
4.32	Stati iniziali $t=0s$ - scenario " <i>Sorpasso di una fila di veicoli</i> "	219
4.33	Fila di veicoli - scenario " <i>Sorpasso di una fila di veicoli</i> "	220
4.34	Traiettoria veicolo <i>ego</i> - scenario " <i>Sorpasso di una fila di veicoli</i> "	221
4.35	Frame traiettoria veicolo <i>ego</i> - scenario " <i>Sorpasso di una fila di veicoli</i> "	222
4.36	Corsia veicolo <i>ego</i> - scenario " <i>Sorpasso di una fila di veicoli</i> "	226
4.37	Valutazione veicoli MIO - scenario " <i>Sorpasso di una fila di veicoli</i> "	227
4.38	Diagramma di flusso <i>function: leadExtensionCheck</i> - scenario " <i>Sorpasso di una fila di veicoli</i> "	232
4.39	Traiettoria veicolo <i>ego</i> - scenario " <i>Sorpasso di una fila di veicoli</i> "	235
4.40	Frame traiettoria veicolo <i>ego</i> - scenario " <i>Sorpasso di una fila di veicoli</i> "	236
4.41	Corsia veicolo <i>ego</i> - scenario " <i>Sorpasso di una fila di veicoli</i> "	237
4.42	Valutazione veicoli MIO - scenario " <i>Sorpasso di una fila di veicoli</i> "	238
4.43	Stati iniziali $t=0s$ - scenario " <i>Logiche di scelta corsia per il veicolo ego</i> "	243
4.44	Traiettoria veicolo <i>ego</i> - scenario " <i>Logiche di scelta corsia per il veicolo ego</i> "	244
4.45	Frame traiettoria veicolo <i>ego</i> - scenario " <i>Logiche di scelta corsia per il veicolo ego</i> "	245
4.46	Corsia veicolo <i>ego</i> - scenario " <i>Logiche di scelta corsia per il veicolo ego</i> "	247
4.47	Valutazione veicoli MIO - scenario " <i>Logiche di scelta corsia per il veicolo ego</i> "	248
4.48	Diagramma di flusso <i>function: helperFindPreferredLane</i> - scenario " <i>Logiche di scelta corsia per il veicolo ego</i> "	254
4.49	Diagramma di flusso <i>function: helperFindPreferredLane</i> con rientro in corsia - scenario " <i>Logiche di scelta corsia per il veicolo ego</i> "	256
4.50	Traiettoria veicolo <i>ego</i> - scenario " <i>Logiche di scelta corsia per il veicolo ego</i> "	258
4.51	Frame traiettoria veicolo <i>ego</i> - scenario " <i>Logiche di scelta corsia per il veicolo ego</i> "	259
4.52	Corsia veicolo <i>ego</i> - scenario " <i>Logiche di scelta corsia per il veicolo ego</i> "	260
4.53	Valutazione veicoli MIO - scenario " <i>Logiche di scelta corsia per il veicolo ego</i> "	261

4.54	Stati iniziali $t = 0s$ - scenario "Modalità di cambio corsia rapido".	264
4.55	Traiettoria veicolo <i>ego</i> - scenario "Modalità di cambio corsia rapido".	266
4.56	Frame traiettoria veicolo <i>ego</i> - scenario "Modalità di cambio corsia rapido".	267
4.57	Errori traiettoria veicolo <i>ego</i> - scenario "Modalità di cambio corsia rapido".	268
4.58	Segnali in output dal controllore <i>ego</i> - scenario "Modalità di cambio corsia rapido".	268
4.59	Confronto grandezze traiettoria di riferimento e traiettoria veicolo <i>ego</i> - scenario "Modalità di cambio corsia rapido".	269
4.60	Punti di lavoro traiettoria di riferimento - scenario "Modalità di cambio corsia rapido".	270
4.61	Punti di lavoro traiettoria veicolo <i>ego</i> - scenario "Modalità di cambio corsia rapido".	271
4.62	Caratteristica angolo di deriva posteriore.	273
4.63	Stima Fy_R - manovra <i>Ramp steer</i> .	275
4.64	Stima β - manovra <i>Ramp steer</i> .	275
4.65	Stima Fy_R - manovra <i>Step steer</i> .	277
4.66	Stima β - manovra <i>Step steer</i> .	277
4.67	Stima Fy_R - manovra <i>Sweep steer</i> .	278
4.68	Stima β - manovra <i>Sweep steer</i> .	278
4.69	Confronto funzioni di trasferimento angolo di assetto/angolo volante - manovra <i>Sweep steer</i> .	279
4.70	Stima Fy_R - manovra <i>Sorpasso semplice</i> .	280
4.71	Stima β - manovra <i>Sorpasso semplice</i> .	280
4.72	Posizione punti di equilibrio instabile al variare della velocità del veicolo.	281
4.73	Ordinata β dei punti di equilibrio instabile al variare della velocità.	282
4.74	Ascissa $\dot{\psi}$ dei punti di equilibrio instabile al variare della velocità.	282
4.75	Esempio piano delle fasi.	283
4.76	Esempio piano delle fasi con zona di stabilità.	284
4.77	Diagramma di flusso della verifica della condizione di stabilità.	285

4.78	Lunghezze utili al calcolo dell'indice di stabilità relativo a $\dot{\psi}$	286
4.79	Traiettoria veicolo <i>ego</i> - scenario "Modalità di cambio corsia rapido".	287
4.80	Frame traiettoria veicolo <i>ego</i> - scenario "Modalità di cambio corsia rapido".	288
4.81	Errori traiettoria veicolo <i>ego</i> - scenario "Modalità di cambio corsia rapido".	289
4.82	Confronto grandezze traiettoria ottimale e traiettoria veicolo <i>ego</i> - scenario "Modalità di cambio corsia rapido".	290
4.83	Confronto della traiettoria effettivamente percorsa dal veicolo <i>ego</i> prima e dopo l'introduzione della verifica - scenario "Modalità di cambio corsia rapido".	291
4.84	Segnali in output dal controllore <i>ego</i> - scenario "Modalità di cambio corsia rapido".	292
4.85	Confronto della posizione nel piano delle fasi dei punti delle traiettorie di riferimento ed effettiva - scenario "Modalità di cambio corsia rapido".	293
4.86	Confronto SI delle traiettorie di riferimento ed effettiva - scenario "Modalità di cambio corsia rapido".	294
4.87	Confronto traiettorie veicolo <i>ego</i> di riferimento/effettiva	296
4.88	Andamento errori laterale ed angolare traiettoria effettiva veicolo <i>ego</i> al confronto con traiettoria di riferimento.	297
4.89	Andamento output prodotti dal controllore nel tempo.	297
4.90	Andamento grandezze descrittive il moto del veicolo <i>ego</i> ed i rispettivi riferimenti.	298
4.91	Andamento indice di stabilità traiettoria di riferimento/traiettoria effettiva.	299
4.92	Confronto traiettorie veicolo <i>ego</i> con C_α costante/variabile.	300
4.93	Confronto errori di traiettoria con C_α costante/variabile.	301
4.94	Angolo di sterzo confronto attuazione rapida/lenta.	304
4.95	Traiettoria veicolo <i>ego</i> confronto attuazione di sterzo rapida/lenta. .	304
4.96	Grandezze moto veicolo <i>ego</i> confronto attuazione di sterzo rapida/lenta.	306
4.97	Angoli di deriva confronto attuazione di sterzo rapida/lenta.	306
4.98	Andamento forze laterali confronto attuazione di sterzo rapida/lenta.	307

Glossario

ABS

Anti-lock Braking System

ACC

Adaptive Cruise Control

ADAS

Advanced Driver Assistance Systems

AEB

Automatic Emergency Braking

AWD

All-Wheel Drive

CC

Cruise Control

ESP

Electronic Stability Program

FWD

Front Wheel Drive

GPS

Global Positioning System

LC

Lane Change

LCA

Lane Change Assist

LCF

Lead Car Following

LKA

Lane Keeping Assist

MIO

Most Important Object

MPC

Model Predictive Controller

RWD

Rear Wheel Drive

TCS

Traction Control System

TSR

Traffic-sign recognition

TTC

Time To Collision

V2X

Vehicle to Everything

Capitolo 1

Introduzione

Con l'obiettivo di facilitare l'esperienza di guida e di ottenere una concomitante riduzione degli incidenti stradali, l'industria dell'automobile si sta spostando verso un crescente livello di automazione. Nel 2022 i dati raccolti dall'ISTAT imputano l'89,7% degli incidenti stradali al comportamento scorretto del conducente [1]. Questo dato riesce a spiegare il motivo per cui la ricerca sui sistemi assistenziali alla guida avanzati (ADAS) si fa sempre più frenetica. Gli studi sono focalizzati sull'influenza dei vari sistemi sul veicolo in termini di sicurezza; con l'obiettivo di migliorare l'efficienza e di ampliare il ventaglio di funzioni che possono essere svolte in assenza della supervisione del conducente. Esaminando gli scenari tipici in cui si può trovare un automobilista, certamente fra le fasi di guida più rischiose concorre la manovra di sorpasso. La criticità è data da velocità generalmente elevate, a cui possono essere associate l'invasione della corsia con senso di marcia opposto o l'esecuzione della manovra in curva. In determinati contesti il guidatore riduce la propria capacità di anticipare i segnali provenienti dal traffico e dallo scenario in cui si trova; conseguentemente, incertezze o errori possono portare a risvolti decisamente gravi. In un ambito scientifico in rapido sviluppo come quello dell'automatizzazione dei veicoli, risulta essere inevitabile l'introduzione di ADAS in grado di agevolare o svolgere in maniera autonoma una manovra alla quale viene associata una percezione del rischio tanto elevata.

1.1 Tecnologie di guida automatica

Si definiscono "*tecnologie di guida automatica*" le tecnologie innovative per la guida automatica basate su sensori di vario tipo, *software* per l'elaborazione dei dati dei sensori e l'interpretazione di situazioni nel traffico, *software* di apprendimento, *software* per assumere decisioni di guida e per la loro attuazione, componenti per l'integrazione con il veicolo tradizionale, che rientrano nell'oggetto della sperimentazione su strada [2]. Gli ADAS fanno parte delle tecnologie di guida automatica, dunque, sono basati sugli input provenienti dalla sensoristica del veicolo. Tali segnali vengono poi processati dai vari sistemi assistenziali, i quali in taluni casi cooperano fra loro, in modo da poter valutare la sicurezza del veicolo e dei suoi occupanti, il raggiungimento di un determinato obiettivo di guida o ancora dare il via a processi decisionali.

1.1.1 Principali sistemi ADAS in dotazione

In relazione a quanto appena descritto, risulta utile per la progettazione di un sistema di sorpasso automatico effettuare una disamina sulle principali tecnologie con cui può essere equipaggiato un veicolo; nell'ottica di individuare i sensori necessari a ricavare gli input richiesti ed i principali ADAS con cui entrare in comunicazione. Fra i principali sistemi avanzati in possibile dotazione al parco auto attuale si annoverano:

- ABS: Anti-lock Braking System. Lavora sull'efficienza della frenata evitando il bloccaggio delle ruote ed evita la perdita di direzionabilità del veicolo.
- TCS: Traction Control System. Impedisce il pattinamento delle ruote durante la fase di accelerazione.
- ESP: Electronic Stability Program. Agisce in caso di sbandata regolando la potenza derivante dal motore e la coppia frenante sulle singole ruote.

Si specifica che si tratta solo di una piccola parte fra tutti gli ADAS in dotazione alle vetture moderne. Quelli sopra citati rappresentano i sistemi di sicurezza fra i più importanti che prevedono l'installazione obbligatoria per legge da diverso tempo.

Per le logiche di sviluppo di sistemi assistenziali come quello oggetto di studio di questa tesi risultano essere fondamentali anche gli ADAS riportati di seguito.

- ACC: Adaptive Cruise Control. Permette al veicolo di impostare una data velocità, mantenerla ed eventualmente adattarla alle condizioni di traffico rilevate.
- LKA: Lane Keeping Assist. Consente al conducente di mantenere il veicolo all'interno della corsia in un ampio range di velocità [3].

ACC e LKA si rivelano particolarmente utili per un sistema di guida automatica, in quanto la loro cooperazione permette di controllare allo stesso tempo sia la dinamica longitudinale che quella laterale [3].

Tra i più diffusi sistemi che attuano dei comandi direttamente sul veicolo è presente il *Park Assist*. Esso è dedicato a effettuare in maniera autonoma le manovre di ingresso e uscita relative ad uno spazio di parcheggio. Risulta particolarmente interessante la cooperazione con i sensori del veicolo, al fine di identificare con precisione lo scenario circostante.

Per un sistema assistenziale come quello oggetto di questo studio, in cui esso deve stabilire la manovra da eseguire in relazione all'analisi dello scenario, risulta particolarmente utile la cooperazione con il sistema *Traffic-sign recognition* (TSR). Esso sfrutta una telecamera frontale per identificare tutti i segnali stradali presenti.

Infine, uno dei sistemi obbligatori che ha un grande impatto sulla sicurezza dei veicoli è l'*Automatic Emergency Braking* (AEB). Esso sfrutta sensori e telecamere per rilevare la presenza di ostacoli nell'area antistante il veicolo; emettendo un segnale acustico e visivo qualora venisse rilevato un ostacolo e, se il conducente non interviene, attua un'azione frenante per evitare la collisione o ridurre la velocità nel momento dell'eventuale impatto.

Lane Change Assist (LCA)

Fra i molti ADAS presenti sulle nuove vetture risulta essere sempre più popolare l'LCA.

- LCA: Lane Change Assist. Permette all'auto di cambiare corsia autonomamente per effettuare un sorpasso. Generalmente il conducente deve solo azionare l'indicatore di direzione o un pulsante [4].

Si tratta di una tecnologia sviluppata per permettere un cambio corsia in tutta sicurezza; si sfrutta l'insieme dei sensori per individuare e segnalare veicoli presenti nelle corsie adiacenti. Oltre a fornire aiuto al conducente durante una manovra che richiede la massima attenzione, permette anche di ampliare la percezione di ciò che accade nell'ambiente circostante, poiché grazie a tale sistema possono essere identificati veicoli che si trovano nei così detti *punti ciechi* del veicolo *ego* [5].

Al giorno d'oggi sono state sviluppate due differenti tipologie di LCA; le quali possono essere identificate come *LCA passivo* ed *LCA attivo*. Come lascia intendere la nomenclatura, la differenza sta nella possibilità del sistema di agire attivamente sullo sterzo e sulla velocità del veicolo [5]. L'*LCA passivo* è un sistema destinato a fornire informazioni aggiuntive sull'ambiente circostante al conducente intenzionato ad effettuare un sorpasso; dunque, esso è un sistema di allarme ormai facente parte della dotazione standard, in quanto sviluppato e introdotto da tutte le case automobilistiche.

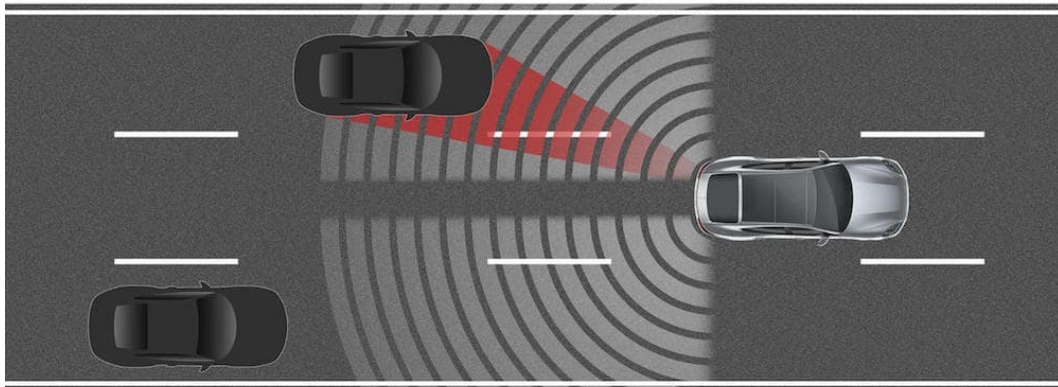


Figura 1.1: Porsche LCA - rilevazione veicoli nei punti ciechi.

L'*LCA passivo* si limita a segnalare la criticità della manovra mediante una spia luminosa posta nel finestrino e l'avvertimento viene fornito se vengono rilevati dei veicoli che procedono a velocità incompatibili con un cambio corsia [6]. L'*LCA attivo* permette al veicolo, tramite l'assistenza di sensori ed attuatori, di cambiare corsia in maniera totalmente autonoma quando questa manovra viene richiesta dal conducente e dopo aver svolto tutti i controlli di sicurezza necessari. Dunque, l'*LCA attivo* è considerabile come l'anticipazione di un sistema di sorpasso totalmente autonomo, come quello integrato in un veicolo con un livello di automazione superiore al secondo livello.

Attualmente molti costruttori hanno equipaggiato le loro vetture con sistemi di tipo *LCA attivo*; fra i quali si annoverano: *Tesla, Cadillac, BMW, Audi, Volkswagen, Subaru e Mercedes-Benz* [5][7].

1.1.2 Sensoristica dei veicolo autonomi

I sistemi assistenziali alla guida avanzati (ADAS) si basano, oltre che sui segnali che descrivono le caratteristiche del veicolo ed il suo funzionamento, sulla ricostruzione dello scenario in cui il veicolo si trova. Affinché si possano ricevere input precisi circa ciò che circonda il veicolo, per poi analizzarli con l'obiettivo di permettere ai sistemi dedicati di valutare la sicurezza del veicolo stesso e dei suoi occupanti, è necessario dotare la vettura di un insieme di sensori in grado di ricostruire con accuratezza lo scenario. Con il fine di comprendere il funzionamento degli ADAS, di cui alcuni esempi sono stati riportati in precedenza, si effettua di seguito una disamina circa la sensoristica dei veicoli moderni.

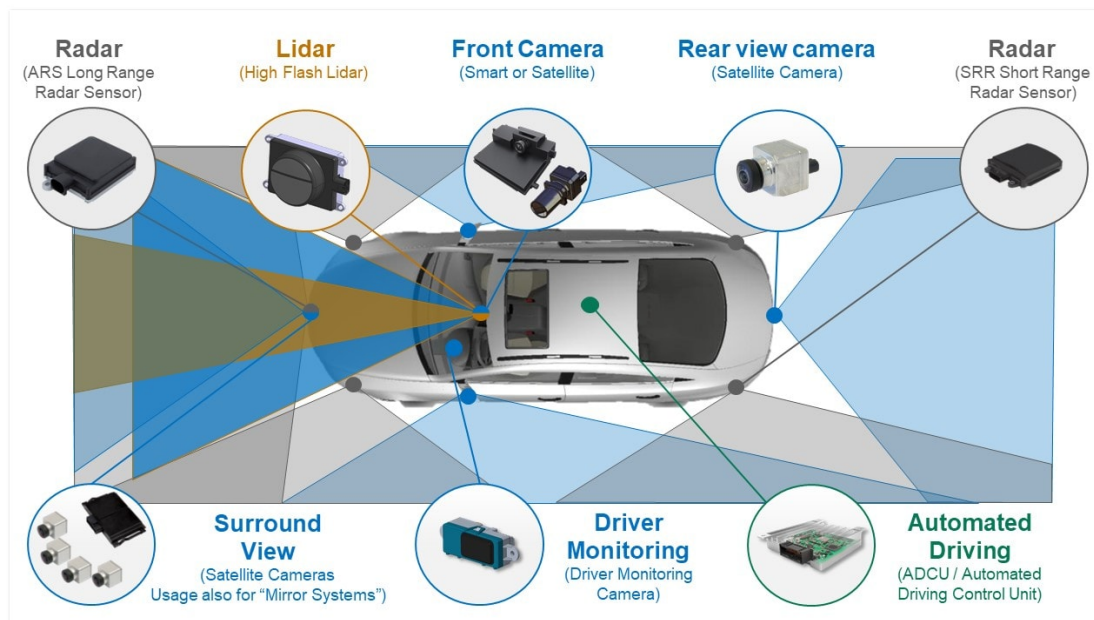


Figura 1.2: Sensori ADAS e Hardware informatico per una percezione a 360°.

In figura 1.2 si ha una rappresentazione dei principali componenti della sensoristica di un veicolo in grado di avere una percezione a 360° dello scenario circostante [3]. La dotazione comprende:

- Un set di camere che copre interamente il campo visivo circostante della vettura.

- Un sistema di radar che copre i cosiddetti "*blind spot*", punti ciechi in corrispondenza dei montanti posteriori, e la parte frontale del veicolo.
- Un sensore LiDAR con campo di azione frontale.
- Un set di camere che forniscono la cosiddetta "*surround view*", ovvero la vista dal satellite del veicolo.
- Una camera che monitori il comportamento del pilota.

Oltre quelli riportati in figura 1.2 solitamente viene anche utilizzato un sistema di geo-localizzazione (GPS) ed in alcuni casi anche un set di sensori ad ultrasuoni. Per un sistema come quello oggetto di questo studio i dati utili ricavabili dai sensori del veicolo sono: la localizzazione della corsia, la localizzazione degli altri veicoli presenti nello scenario e la definizione geometrica dello scenario stesso [8].

Radar

Il principale tipo di dispositivo da cui ricavare i segnali necessari ad un sistema di sorpasso automatico è il radar. Si tratta di un dispositivo in grado di rilevare la distanza, la velocità e l'angolo azimutale di ogni oggetto che riflette al ricevitore una parte dell'energia emessa dal trasmettitore [9]. Di derivazione militare è stato ampiamente utilizzato negli ultimi anni nelle vetture prodotte, in quanto necessario a sistemi ADAS ben noti quali ACC e LCA; dunque, il vantaggio nel suo utilizzo è indubbio, in quanto già in dotazione alle vetture moderne.

LiDAR

Il LiDAR, il cui acronimo sta per "*light detection and range*", è un dispositivo che permette di misurare con elevata precisione le distanze tra il sensore e gli oggetti dello scenario, utilizzando una tecnologia laser. A differenza delle microonde utilizzate dal sensore radar, il mezzo con cui il LiDAR effettua la misurazione, ovvero la luce riflessa, permette una maggiore velocità dell'acquisizione dei dati ed una miglior precisione degli stessi. Tuttavia, il raggio d'azione è limitato a pochi metri.

Vehicle to Everything

In termini di metodologie di acquisizione dei dati relativi allo scenario risulta essere di rilevante impatto, nel contesto del progressivo sviluppo delle tecnologie di automatizzazione dei veicoli, il "*Vehicle to Everything*" (V2X). Si tratta di un sistema di comunicazione che sfrutta la rete WLAN per uno scambio di informazioni in tempo reale tra tutti i componenti dello scenario stradale. Esso è formato da differenti tecnologie, ognuna delle quali sfrutta il medesimo sistema per scambiare informazioni con una specifica categoria di attori o oggetti, le quali influenzano e/o aiutano le analisi dati condotte dai sistemi assistenziali. I singoli canali di comunicazione sono i seguenti:

- V2D: Vehicle to Device.
- V2G: Vehicle to Grid.
- V2I: Vehicle to Infrastructure.
- V2N: Vehicle to Network.
- V2P: Vehicle to Pedestrian.
- V2V: Vehicle to Vehicle.

Sviluppare sistemi di questo tipo, portando ogni veicolo ad un'onniscenza relativa al contesto in cui si trova, porterebbe a benefici enormemente impattanti in termini di sicurezza e di potenzialità di sviluppo dei sistemi assistenziali.

1.2 Livelli di automazione SAE

La presenza di sistemi e dispositivi utili ad automatizzare i veicoli ha portato alla creazione di un parco auto con diverse potenzialità in termini di manovre realizzabili senza l'ausilio del guidatore. Questa varietà ha portato alla necessità di una tassonomia con definizioni dettagliate per ogni livello di automazione; la quale è stata introdotta mediante la norma *J3016* dalla *SAE International* [10]. I livelli che definisce i seguenti:

- Livello 0: No Driving Automation.
- Livello 1: Driver Assistance.
- Livello 2: Partial Driving Automation
- Livello 3: Conditional Driving Automation.
- Livello 4: High Driving Automation.
- Livello 5: Full Driving Automation.

Come si può evincere dalle brevi descrizioni, partendo dal Livello 0 con automazione totalmente nulla e procedendo verso livelli crescenti, l'importanza di sistemi e dispositivi di assistenza alla guida all'interno del veicolo ha una rilevanza sempre maggiore. Il massimo livello raggiungibile è il livello 5 che identifica un veicolo completamente autonomo. Dopo aver definito i singoli livelli, la SAE ha effettuato una ulteriore distinzione in due gruppi. Sotto la dicitura "*Driver Support*" si trovano i primi tre livelli (da 0 a 2) e gli ultimi tre (da 3 a 5) vengono identificati come "*Automated Driving*". Questa distinzione è stata effettuata per far immediatamente comprendere chi fra Driver e Sistema di guida automatizzata ha l'effettivo controllo del veicolo. Di particolare rilevanza per questa tesi sono i veicoli appartenenti al gruppo Automated Driving, in quanto la loro guida automatizzata implica la capacità di effettuare dei sorpassi quando necessario. I veicoli di livello 3 sono progettati in modo da poter sorpassare il veicolo che li precede durante un viaggio autostradale; aumentando il livello lo scenario in cui tale manovra può essere effettuata è chiaramente più vasto.

1.3 Contesto normativo ed aspetto etico

L'impatto sulla sicurezza alla guida dei sistemi ADAS risulta essere talmente evidente da ritenerli ormai indispensabili per le auto moderne. Da diversi anni alcuni fra i sistemi citati sono diventati obbligatori per le auto da introdurre nel commercio, elevando in tal modo gli standard richiesti relativi al livello di automazione dei nuovi veicoli.

1.3.1 Normativa UE - Introduzione nuovi ADAS obbligatori

La normativa *UE 2019/2144* a partire dal 7 luglio 2024 impone come obbligatoria la presenza dei seguenti sistemi assistenziali per tutti i veicoli da immatricolare, che siano di nuova o precedente omologazione [11]:

- *Adattamento intelligente della velocità (ISA)*: il quale integra i sistemi ACC e TSR in un unico sistema. In tal modo il veicolo riesce ad adattare la propria velocità di crociera sia alla presenza di altri veicoli nello scenario sia ai limiti imposti dalla segnaletica stradale.
- *Interfaccia per l'etilometro*: Viene resa obbligatoria la predisposizione dei veicoli ad installare un dispositivo di alcolock.
- *Monitoraggio dell'attenzione e della stanchezza*: il quale rileva tramite la posizione delle mani sul volante e/o il movimento dello sterzo la stanchezza del conducente.
- *Segnalatore di arresto di emergenza*: il già citato AEB.
- *Rilevamento oggetti e pedoni in retromarcia*: il quale sfrutta appositi sensori per individuare la presenza di ostacoli in retromarcia.
- *Registrazione dei dati relativi agli eventi*: il quale all'interno della così detta "scatola nera" registra tutti i dati relativi ad eventi straordinari, ad esempio gli incidenti. Tale dispositivo permette l'accesso ai dati solo alle forze dell'ordine.
- *Mantenimento di corsia*: il già citato LKA.

- *Misuratore di pressione degli pneumatici (TPMS)*: il quale informa il conducente della pressione degli pneumatici, ricavata a partire dai dati della centralina ABS.

1.3.2 Decreto Smart Road

Ad un progressivo sviluppo delle tecnologie in dotazione alle moderne vetture deve necessariamente essere accompagnato lo sviluppo delle infrastrutture sulle quali esse transitano. In seguito a direttive UE il Ministero delle Infrastrutture e dei Trasporti ha pubblicato il 28 febbraio 2018 il decreto "*Smart Road*" [2].

La legge stabilisce gli standard funzionali per strade più connesse e sicure che, grazie all'introduzione di nuove tecnologie, permettono di avvicinarsi al sistema V2X presentato in precedenza. La direzione adottata è quella di un progressivo miglioramento delle infrastrutture per riuscire ad ottenere una comunicazione in tempo reale tra gli utenti a bordo del veicolo e le strade; veicolando informazioni relative al traffico, al meteo, ad incidenti e così via. Gli obiettivi pianificati a lungo termine riguardano l'implementazione di interventi sulle velocità dei veicoli, la gestione di parcheggi e rifornimenti ed anche il suggerimento di traiettorie.

La novità apportata dal decreto che è più inerente con la tematica trattata in questo studio è la possibilità, da parte del Ministero dei Trasporti, di autorizzare la sperimentazione di veicoli a guida automatica su alcuni tratti di strada ed in assoluta sicurezza; la quale può essere richiesta da università, enti pubblici e privati [12].

1.3.3 Aspetto etico

L'introduzione di tecnologie di guida automatica nel parco veicoli circolante dà vita ad un numero cospicuo di dilemmi di carattere etico. Lo sviluppo di questi sistemi richiede investimenti massivi su veicoli e infrastrutture; di conseguenza esso si accompagna a dibattiti anche di carattere pubblico e politico.

Una tematica critica è l'attribuzione della responsabilità relativa ad incidenti e alla sicurezza nel traffico. In quanto attualmente esse sono imputabili al conducente; tuttavia, con il progredire del livello di automazione assumono una rilevanza tangibile i sistemi di comunicazione, le strade ed il veicolo stesso. Deve, dunque, essere compito del progettista quello di tarare i principi che guidano i processi decisionali del sistema di guida autonoma, in modo da ponderare le logiche di scelta su un numero n di fattori; con l'obiettivo di minimizzare il sacrificio effettuato nel caso di dilemmi etici.

L'opinione pubblica ha un peso rilevante sulla tematica; in quanto in campo etico non si ha mai una risposta oggettiva al quesito posto. Questo comporta una tolleranza attesa, su episodi di incidenti di veicoli con guida autonoma, inferiore rispetto a quella che accompagna episodi che coinvolgono veicoli tradizionali. Queste considerazioni possono portare a ritardare l'introduzione di queste tecnologie nel mercato o alla richiesta di limitazioni sulle relative funzionalità [13].

1.4 Struttura della tesi

L'obiettivo di questa tesi è quello di realizzare una disamina circa i principi e le componenti che stanno alla base di un sistema assistenziale alla guida avanzato e dedicato al sorpasso automatico.

Nello specifico ci si predilige di offrire una visione teorica e completa delle tematiche che è necessario comprendere per progettare un ADAS di questo genere. Le medesime tematiche sono state analizzate da una differente prospettiva, quella pratica, in un ambiente di simulazione dedicato. A tale scopo sono stati raccolti degli strumenti utili ad ottenere ed analizzare i dati necessari al suddetto sistema ADAS, fra cui un set di funzioni dedicate alla simulazione veicolo ed un modello di base, a partire dal quale è stato costruito il modello completo utilizzato per le simulazioni riportate.

Come suggerisce il titolo della ricerca "*Simulazione di scenari di guida complessi per lo studio di un sistema ADAS di sorpasso automatico*", la simulazione svolge il ruolo di attore protagonista. Gli strumenti presentati nel corso della ricerca sono stati sfruttati per realizzare delle simulazioni in grado di fornire dei risultati utili ad evidenziare le criticità di un sistema come quello oggetto di studio. Queste sono state poi approfondite sia sul piano teorico che su quello pratico; ovvero proponendo una corretta metodologia di manipolazione dei segnali all'interno dell'ambiente di simulazione, per poi verificarne l'efficienza attraverso l'analisi dei risultati. Nella parte conclusiva del lavoro è quindi stato introdotto un set di scenari complessi, con l'obiettivo di testare le risposte del sistema a fronte di input diversi; come diversi sicuramente saranno in un contesto applicativo reale.

In conclusione, si può affermare che la struttura della tesi è stata sviluppata con l'intento di fornire, procedendo nei vari capitoli, una visione sempre più dettagliata della tematica in esame; con l'obiettivo ultimo di fornire una prospettiva tale da permettere lo sviluppo di un sistema che abbia una logica solida, quindi indipendente dallo scenario adottato.

Capitolo 2

Pianificazione della traiettoria

La parte di maggior rilievo di un sistema assistenziale alla guida destinato al sorpasso automatico è senza alcun dubbio il *motion planner* o *path planner*. Si tratta dell'area del sistema dedicata alla pianificazione della traiettoria di riferimento. Le caratteristiche che deve indispensabilmente avere il sistema per raggiungere tale obiettivo sono le seguenti:

- Avere la capacità di generare n differenti traiettorie alternative che portino il veicolo *ego* dal punto corrente ad un numero n di punti differenti.
- Disporre di un algoritmo in grado di valutare l'effettiva percorribilità di ciascuna delle n traiettorie.
- Disporre di un algoritmo in grado di valutare il rischio di collisione per ciascuna delle n traiettorie.

All'interno di questo capitolo sono state analizzate nel dettaglio le tre tematiche esposte.

2.1 Generazione della traiettoria

Il compito principale del sistema assistenziale è quello di generare una traiettoria da far seguire al veicolo *ego*. Si specifica che ciò che nel concreto viene pianificato è un insieme di segmenti di traiettoria di riferimento che, se uniti, congiungono il punto di partenza con il punto di arrivo del veicolo. Ognuno di questi segmenti viene pianificato in un dato istante che nel corso di questa trattazione verrà chiamato "istante di pianificazione".

Nell'istante di pianificazione viene prodotto un nuovo segmento di traiettoria di riferimento; si può procedere tramite due differenti strategie:

- Con un campionamento *spaziale*: attraverso il quale ogni segmento viene prodotto in modo tale da far percorrere al veicolo una distanza stabilita.
- Con un campionamento *temporale*: attraverso il quale ogni segmento viene prodotto in modo tale da farlo percorrere dal veicolo *ego* in un determinato intervallo di tempo; tale intervallo viene chiamato orizzonte temporale o *Time Horizon*.

Nell'ambito di questo studio è stata adottata la seconda soluzione per una maggiore compatibilità con gli strumenti simulativi a disposizione.

2.1.1 Metodologie di generazione della traiettoria

Il letteratura possono essere riscontrate differenti alternative in merito alla tecnica di generazione del segmento di traiettoria che unisce il punto di partenza P_0 ad un punto di arrivo P_1 . La soluzione adottata in questo caso studio consiste nel pianificare i segmenti mediante l'utilizzo di curve polinomiali del quinto ordine. Tra le varie alternative disponibili è stata scelta quella citata in quanto è possibile dimostrare che, a parità di estremi P_0 e P_1 e di altre condizioni, essa minimizza la seguente funzione di costo C [14].

$$C = \frac{1}{2} \int_{t_0}^{t_1} (\ddot{x}^2 + \ddot{y}^2) dt \quad (2.1)$$

Tale funzione di costo viene definita come l'integrale del quadrato del jerk, il quale viene calcolato rispetto ad un sistema di riferimento di tipo inerziale nel piano XY ; a tale grandezza viene generalmente associata la percezione del comfort. Le equazioni che descrivono la traiettoria all'interno del medesimo sistema di riferimento hanno la seguente forma.

$$\begin{cases} x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \\ y(t) = b_0 + b_1 t + b_2 t^2 + b_3 t^3 + b_4 t^4 + b_5 t^5 \end{cases} \quad (2.2)$$

Di conseguenza generare il corretto segmento di traiettoria vuol dire individuare i 12 coefficienti del sistema, a partire dalle seguenti condizioni iniziali.

$$\begin{cases} x(t_0) = X_0 \\ \dot{x}(t_0) = V_{x,0} \\ \ddot{x}(t_0) = A_{x,0} \\ x(t_1) = X_1 \\ \dot{x}(t_1) = V_{x,1} \\ \ddot{x}(t_1) = A_{x,1} \\ y(t_0) = Y_0 \\ \dot{y}(t_0) = V_{y,0} \\ \ddot{y}(t_0) = A_{y,0} \\ y(t_1) = Y_1 \\ \dot{y}(t_1) = V_{y,1} \\ \ddot{y}(t_1) = A_{y,1} \end{cases} \quad (2.3)$$

Dove:

- t_0 e t_1 : sono gli estremi dell'intervallo temporale associato al percorso di estremi P_0 e P_1 .
- $[X, Y]$, $[V_x, V_y]$ e $[A_x, A_y]$: sono le componenti rispettivamente di posizione,

velocità e accelerazione lungo gli assi del sistema di coordinate fisso.

Nel caso di utilizzo di un sistema di riferimento espresso in coordinate di *Frenét*, le condizioni iniziali che permettono di ricavare i coefficienti delle equazioni che descrivono la variazione delle due coordinate $s(t)$ e $d(t)$ sono i seguenti:

$$\left\{ \begin{array}{l} s(t_0) = s_0 \\ \dot{s}(t_0) = \dot{s}_0 \\ \ddot{s}(t_0) = \ddot{s}_0 \\ s(t_1) = s_1 \\ \dot{s}(t_1) = \dot{s}_1 \\ \ddot{s}(t_1) = \ddot{s}_1 \end{array} \right. \quad (2.4)$$

$$\left\{ \begin{array}{l} d(t_0) = d_0 \\ d'(t_0) = d'_0 \\ d''(t_0) = d''_0 \\ d(t_1) = d_1 \\ d'(t_1) = d'_1 \\ d''(t_1) = d''_1 \end{array} \right.$$

Dove:

- $[s_0, \dot{s}_0, \ddot{s}_0, d_0, d'_0, d''_0]$: Indica lo stato del sistema nel punto iniziale P_0 del segmento di traiettoria pianificato.
- $[s_1, \dot{s}_1, \ddot{s}_1, d_1, d'_1, d''_1]$: Indica lo stato del sistema nel punto finale P_1 del segmento di traiettoria pianificato.

Si rimanda al paragrafo 2.1.2 per il confronto tra i due sistemi di riferimento.

Differenti soluzioni presenti in letteratura

Come anticipato, quella della polinomiale del quinto ordine non rappresenta l'unica strategia utilizzata per calcolare la traiettoria in sistemi di *path planning*. Tra le soluzioni più diffuse si annoverano:

- *Curva Sigmoide*: la quale viene utilizzata esclusivamente per pianificare traiettorie dedicate al cambio corsia; a differenza della soluzione adottata che ha permesso di implementare una singola metodologia di pianificazione, valida per tutte le tipologie di traiettoria pianificabili.

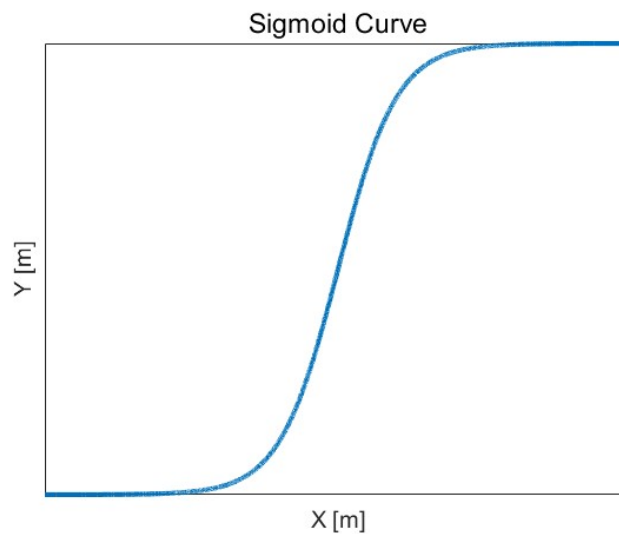


Figura 2.1: Esempio curva Sigmoide.

L'equazione che la descrive è la seguente.

$$y(x) = \frac{1}{1 + e^{-x}} \quad (2.5)$$

- *Reticolo di stati* o *State Lattice*: tale metodologia consiste nella discretizzazione dello spazio nel quale il veicolo può spostarsi, con il fine di formare un reticolo di punti in cui esso può essere posizionato. Il vantaggio nell'adottare questa soluzione sta nell'inferiore costo computazionale, in quanto le manovre tra i punti risultano essere predeterminate, come si può vedere in figura 2.2 [15][16].

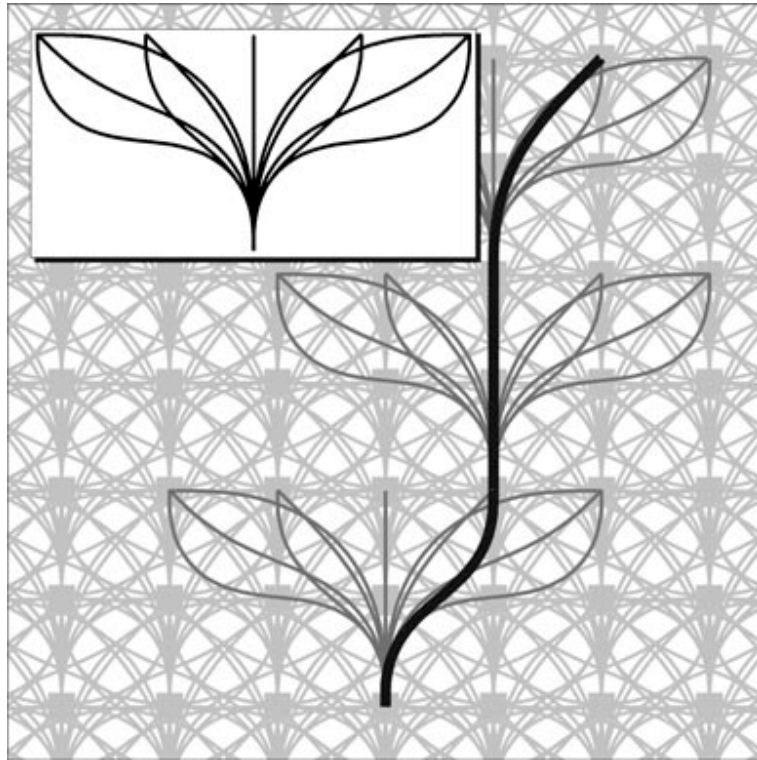


Figura 2.2: Traiettorie possibili nel reticolo di stati o *State Lattice*.

Tuttavia, tale strategia non risulta ottimale, in quanto in tal modo si limitano le manovre del veicolo anche in termini di angolo di sterzo attuabili [15].

2.1.2 Sistema di riferimento Frenet

Un approccio ben noto al problema del *Trajectory planning* è il cosiddetto *Frenét Frame method*. Esso consiste nell'adottare un sistema di riferimento mobile dove il punto considerato in un determinato istante viene descritto dal vettore normale e da quello tangente, \vec{t}_r e \vec{n}_r , rispetto ad un determinato punto di una curva di riferimento nota come *center line*. La *center line* viene scelta arbitrariamente dal progettista; essa può rappresentare la traiettoria ideale da seguire, la linea di mezzzeria della corsia o il risultato di un differente algoritmo di *path planning* [14].

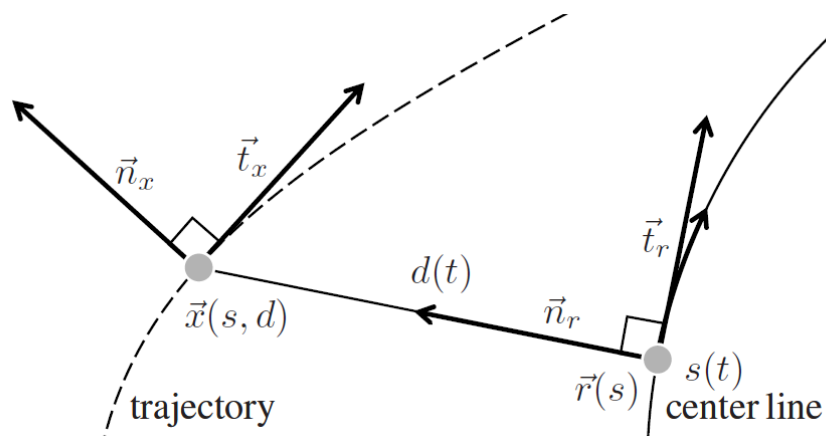


Figura 2.3: Generazione di una traiettoria con coordinate di Frénet

Conseguentemente piuttosto che utilizzare direttamente le coordinate cartesiane \vec{x} , si cambia il sistema di riferimento generando una traiettoria monodimensionale, dipendente dal punto \vec{r} sulla *center line* e dall'offset d ; come si può apprezzare in figura 2.3 e secondo la seguente relazione:

$$\vec{x}(s(t), d(t)) = \vec{r}(s(t)) + d(t)\vec{n}_r(s(t)) \quad (2.6)$$

Dove con s si indica la lunghezza dell'arco della *center line* che è stato percorso e \vec{t}_x e \vec{n}_x sono i vettori tangente e normale alla traiettoria risultante $\vec{x}(s(t), d(t))$ [14]. Tutte le traiettorie vengono quindi descritte dalle coordinate $[s, d]$, che vengono riferite alla *center line* prestabilita.

Il vantaggio di questo metodo sta nella più semplice descrizione dei punti di una determinata traiettoria; la quale risulta di più immediata comprensione anche dal punto di vista numerico. In figura 2.4 è stato riportato un esempio grafico della traiettoria di riferimento del veicolo *ego*, i cui punti sono definiti rispetto ad un sistema di tipo non inerziale, la cui origine coincide con la posizione del veicolo *ego*. Si noti come risulta essere numericamente complesso descrivere l'evoluzione degli stati associati alla traiettoria in blu, quando questo viene fatto con il sistema di riferimento veicolo. In questo caso, anche quantificare lo scostamento del veicolo dalla traiettoria di riferimento è più complesso.

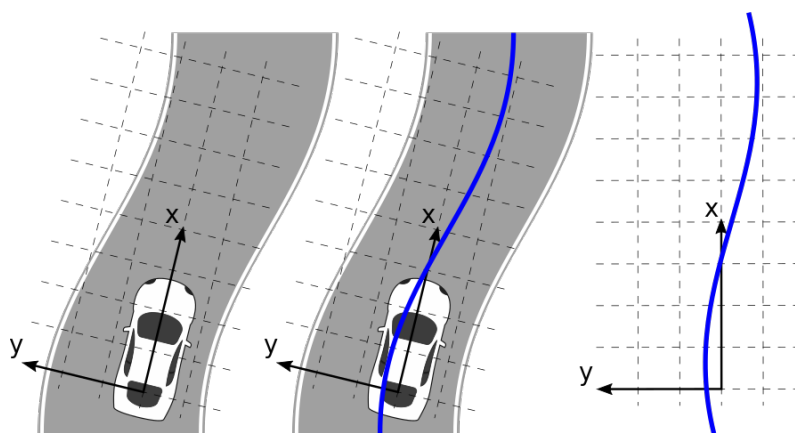


Figura 2.4: Traiettoria di riferimtno espressa mediante coordinate veicolo.

In figura 2.5 si ha la rappresentazione del medesimo scenario espressa mediante sistema di riferimento di Frenét.

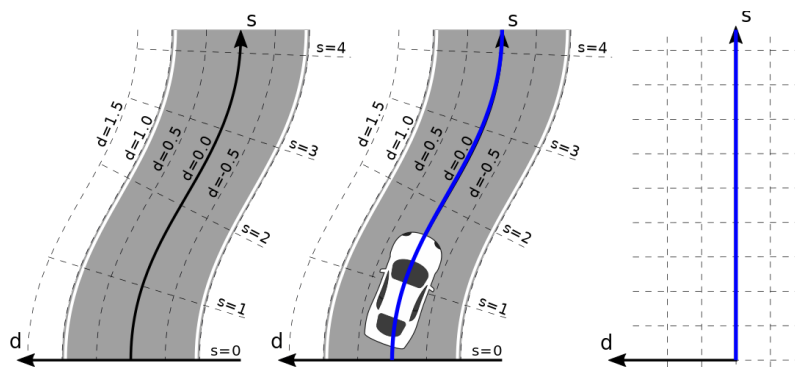


Figura 2.5: Traiettoria di riferimento espressa mediante coordinate di Frenét.

Si nota che la traiettoria di riferimento del sistema di Frenét coincide con la linea di mezzzeria della strada e con la traiettoria di riferimento del veicolo. Esprimere la posizione del vicolo *ego* all'interno della carreggiata risulta essere notevolmente più semplice; in quanto ipotizzando un suo scostamento laterale rispetto alla linea di mezzzeria, la coordinata d quantifica già tale grandezza.

Nel corso di questo studio è stata sempre adottata come traiettoria di riferimento la linea di mezzzeria della carreggiata. Dunque, utilizzando questo metodo risulta essere di facile implementazione una strategia di campionamento dei punti terminali dei segmenti di traiettoria di riferimento anche nel caso di manovra di sorpasso. Poiché per identificare un punto P_1 in cui far terminare la traiettoria con cui eseguire il cambio corsia basta esprimere la quantità Δd di cui si vuole spostare il veicolo rispetto alla relativa corsia attuale. Per una migliore comprensione di tali tematiche teoriche, nel paragrafo successivo è stato realizzato un esempio di costruzione della traiettoria di cambio corsia.

2.1.3 Strumenti di generazione della traiettoria in ambiente di simulazione

Nelle librerie del *software* MatLab & Simulink è disponibile un insieme di funzioni dedicate agli argomenti trattati e coerenti con le scelte fatte in termini di sistema di riferimento e metodologia di generazione delle traiettorie.

global2frenet: La funzione *global2frenet* permette di passare dal sistema di riferimento globale al sistema di riferimento di Frenét, fornendo in input alla funzione i waypoints della curva scelta come riferimento [17]. Il cambio di coordinate è il seguente:

$$[x, y, \psi, \kappa, v, a] \rightarrow [s, \dot{s}, \ddot{s}, d, d', d''] \quad (2.7)$$

Con:

- $[x, y]$: coordinate globali del dato punto P.
- ψ : angolo di imbardata.

- κ : curvatura della traiettoria nel dato punto P.
- v ed a : rispettivamente velocità ed accelerazione.
- s : lunghezza dell'arco di curva di riferimento percorso.
- d : distanza in direzione normale dalla curva di riferimento.

Si rimanda ai riferimenti bibliografici per una descrizione più dettagliata dell'algoritmo di dedicato al cambio di coordinate [17] [14].

trajectoryGeneratorFrenet: Nelle librerie dedicate al *Navigation Toolbox* è presente un insieme di funzioni dedicate alla generazione di traiettorie, le quali si basano sui principi esposti in precedenza. La funzione *trajectoryGeneratorFrenet* permette di generare le traiettorie che uniscono il punto P_0 , attualmente occupato dal veicolo *ego*, con un set di n punti in cui fare terminare gli n segmenti di traiettoria prodotti. La generazione di traiettorie viene effettuata utilizzando delle curve polinomiali del quinto ordine, alle quali viene associato un orizzonte temporale $Th = (t_1 - t_0)$ specificato.

Esempio generazione di traiettorie

Con il fine di rendere la spiegazione di più immediata comprensione, è stato realizzato un caso studio esempio; il quale si basa su uno scenario che, con le modifiche progressivamente introdotte in questo capitolo, ha portato alla definizione dello scenario analizzato nel capitolo quarto e denominato "*Sorpasso con veicolo in senso di marcia opposto*".

La configurazione iniziale dello scenario prevede la presenza di due soli veicoli:

- Veicolo *ego*: veicolo oggetto della simulazione. Il quale in questo caso ha una velocità $v_{ego} = 20m/s$.
- Veicolo *lead*: veicolo che precede il veicolo *ego* transitando nella medesima corsia. Il quale in questo caso ha una velocità $v_{lead} = 15m/s$.

Essi transitano lungo una strada rettilinea lunga $300m$, nella medesima corsia, con il medesimo senso di marcia e sono posti ad una distanza iniziale pari a $40m$.

Si riporta in figura una rappresentazione dello scenario all'istante iniziale $t = 0$ s.

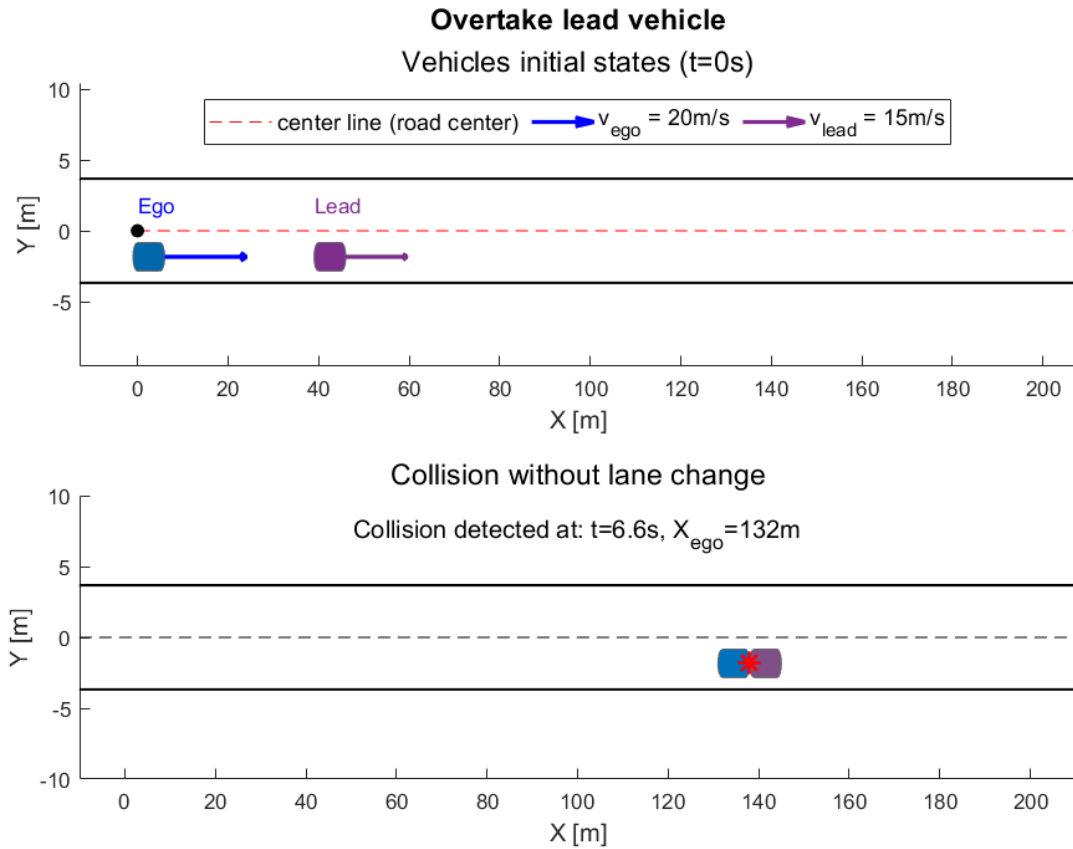


Figura 2.6: Stati iniziali $t = 0$ s - scenario "sorpasso del veicolo *lead*".

Nel secondo grafico è stata riportata l'eventuale collisione che si riscontrerebbe qualora si mantenesse una modalità di Cruise Control (CC), secondo la quale il veicolo *ego* mantiene la propria traiettoria e la propria velocità di crociera inalterate.

L'intervento di un sistema assistenziale come quello oggetto di questo studio è quello di individuare due differenti soluzioni:

- Modalità LCF: Viene identificata come traiettoria di riferimento una traiettoria che preveda il mantenimento della corsia e la riduzione della velocità del veicolo *ego* di un'entità tale da mantenere costante la distanza dal veicolo *lead*.

- Modalità LC: Viene identificata come traiettoria di riferimento una traiettoria che porti il veicolo *ego* in una corsia differente rispetto a quella attualmente occupata. L'assunzione fatta in questo studio è quella di pianificare le traiettorie di cambio corsia ad una velocità \dot{s} costante.

Dopo aver presentato il caso studio, si passa di seguito a pianificare un set di traiettorie mediante l'utilizzo delle funzioni citate, in modo da fornire un riscontro pratico della metodologia implementata all'interno del modello presentato nel capitolo terzo.

Per ognuna delle due modalità è stato scelto di pianificare due differenti segmenti di traiettoria, uno per ogni valore di orizzonte temporale scelto ed espresso in secondi.

$$TimeHorizons = [1.5, 3] \quad (2.8)$$

Il primo step è quello di definire i *way points* della curva di riferimento del sistema di coordinate *Frenét*. Definiti attraverso coordinate cartesiane nel piano $X - Y$. Come anticipato, in questa trattazione la curva di riferimento scelta è sempre la linea di mezzzeria della carreggiata. Rappresentata in rosso nella figura 2.6.

$$wayPoints = [0, 0; 300, 0] \quad (2.9)$$

Dopodiché si crea un oggetto contenente una curva che congiunge i *way points* mediante la funzione dedicata *referencePathFrenet*.

$$refPath = referencePathFrenet(wayPoints) \quad (2.10)$$

Tale curva è necessaria a creare l'oggetto denominato "connector" che ospita al proprio interno tutte le informazioni relative alla curva di riferimento del sistema in coordinate di Frenét. Esso viene creato attraverso la funzione *trajectoryGeneratorFrenet*.

$$connector = trajectoryGeneratorFrenet(refPath) \quad (2.11)$$

L'ultimo step consiste nell'ottenere il set di traiettorie, espresso in coordinate globali $[x, y, \psi, \kappa, v, a]$, attraverso la funzione *connect*. Essa ha il compito di collegare lo

stato iniziale con il set di stati terminali; tali traiettorie vengono percorse in un tempo pari all'orizzonte temporale specificato.

Il set di stati terminali, arbitrariamente creato, è composto dagli stati espressi in coordinate di Frenét $[s, \dot{s}, \ddot{s}, d, d', d'']$, ai quali viene associato un determinato orizzonte temporale. Esso è stato così composto:

Modalità	Stato terminale	Orizzonte temporale
LCF	$[NaN, \dot{s}_{lead,0}, 0, -laneWidth/2, 0, 0]$	1.5 s
LCF	$[NaN, \dot{s}_{lead,0}, 0, -laneWidth/2, 0, 0]$	3 s
LC	$[NaN, \dot{s}_{ego,0}, 0, +laneWidth/2, 0, 0]$	1.5 s
LC	$[NaN, \dot{s}_{ego,0}, 0, +laneWidth/2, 0, 0]$	3 s

Tabella 2.1: Stati terminali - scenario "sorpasso del veicolo *lead*".

Si specifica che il valore s di ciascun stato terminale non è stato specificato, in quanto in tal caso viene calcolato in maniera autonoma dalla funzione a partire dai valori di orizzonte temporale e velocità. Lo stato iniziale viene numericamente espresso come:

$$initialState = [s_0, \dot{s}_{ego,0}, 0, -laneWidth/2, 0, 0] \quad (2.12)$$

Dove $\dot{s}_{ego,0}$ coincide con $v_{ego,0}$. Si specifica che la coordinata d associata al centro delle due corsie è pari a:

Corsia	d
<i>Destra</i>	$-laneWidth/2$
<i>Sinistra</i>	$+laneWidth/2$

Tabella 2.2: Coordinata d singole corsie - scenario "sorpasso del veicolo *lead*".

Dove "destra" e "sinistra" sono state definite dal punto di vista del veicolo *ego* e "laneWidth" è l'ampiezza di una singola corsia.

Utilizzando la funzione *connect* si ottiene il set di traiettorie espresso in coordinate globali.

$$trajGlobal = connect(connector, initState, terminalStates, timeHorizons) \quad (2.13)$$

Si riporta in figura 2.7 un riscontro grafico delle traiettorie ottenute a partire dal set di stati terminali descritto in precedenza.

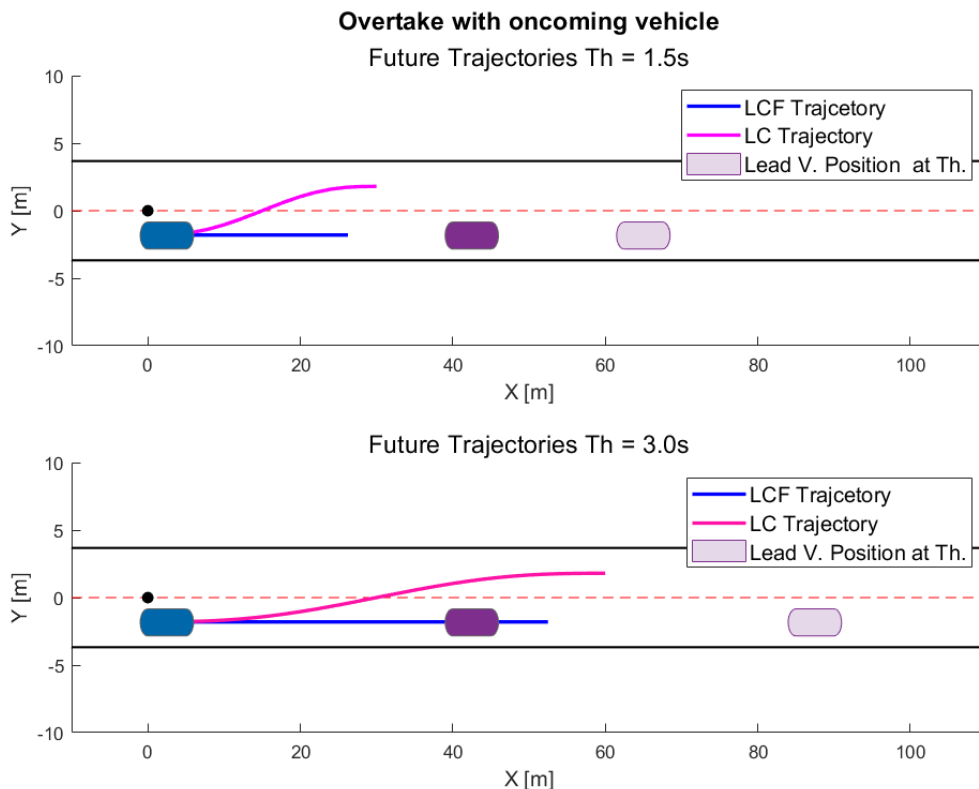


Figura 2.7: Set traiettorie alternative- scenario "sorpasso del veicolo *lead*".

Si noti che è stata inserita una rappresentazione della posizione del veicolo *lead* al dato orizzonte temporale, riportata con colorazione semitrasparente, in modo da poterla confrontare con l'ultimo punto delle traiettorie rappresentate. Inoltre, si noti che il punto terminale delle traiettorie di LC ha una coordinata s maggiore rispetto alle traiettorie calcolate con modalità LCF, in quanto il veicolo *ego* in modalità LC ha una velocità \dot{s} superiore.

Tramite questo esempio è possibile comprendere come sia effettivamente semplice definire gli stati del veicolo in coordinate di *Frenét*.

2.2 Valutazione di fattibilità delle traiettorie alternative

Il passo immediatamente successivo alla creazione di un set di possibili traiettorie da seguire è la valutazione dei risultati ottenuti. Ciò che generalmente viene definito come traiettoria di riferimento non corrisponde effettivamente alla migliore fra le infinite combinazioni possibili ma a quella che, partendo da un set composto da n alternative e rispettando tutte le verifiche imposte, minimizza una data funzione di costo.

La prima verifica che viene generalmente effettuata riguarda la percorribilità della traiettoria; dove con tale termine si intende l'effettiva capacità del veicolo di riuscire a sopperire alla richiesta di prestazioni necessaria a seguire, il più fedelmente possibile, il riferimento fornito.

Bisogna tener conto che le grandezze che descrivono il comportamento del veicolo, le quali vengono associate alle traiettorie di riferimento campionate, sono frutto di numerose semplificazioni in termini dinamici e spesso vengono calcolate con metodologie puramente cinematiche. Ciò è correlato ad una tematica ricorrente nel mondo delle simulazioni, in quanto si cerca sempre il corretto punto di equilibrio tra accuratezza dei risultati, costo computazionale e semplicità delle modellizzazioni introdotte.

Verifica cinematica di base

Nel caso più semplice la verifica sulla fattibilità può essere limitata all'inserimento di alcune soglie limite da rispettare per tutti i punti campionati di una data traiettoria. Generalmente i parametri su cui viene condotta l'analisi sono associati a grandezze quali: accelerazione longitudinale e laterale, jerk, velocità di imbardata, curvatura della traiettoria e così via. Bisogna necessariamente tener conto del fatto che si tratta di stime, in quanto tali valori vengono calcolati in previsione dell'effettiva risposta del veicolo.

Qualora i valori limite fossero il frutto di una caratterizzazione della risposta del veicolo a differenti input, tale metodologia potrebbe essere utilizzata per scartare tutte le alternative i cui valori associati sono ampiamente al di fuori della soglia tollerata. Il vantaggio di questa soluzione è insito nella rapidità e nel basso costo computazionale della verifica. Tuttavia, in condizioni di funzionamento più estreme, la dinamica del veicolo influenzerebbe in maniera massiva tutte le grandezze calcolate; dunque, per tali condizioni è necessario attuare una tecnica di verifica di fattibilità più avanzata.

2.2.1 Analisi non lineare della stabilità del veicolo

Nelle condizioni di lavoro in cui la prestazione richiesta risulta molto elevata il veicolo può incorrere nel fenomeno dell'instabilità direzionale. Di conseguenza per sistemi avanzati è importante includere una strategia che sia in grado di valutare la richiesta prestazionale delle traiettorie alternative, depennando dal set quelle che porterebbero ad instabilità, dunque, ad una condizione di elevatissimo rischio in termini di sicurezza per il veicolo ed i suoi occupanti.

Affinché si verifichi tale condizione è necessario che il veicolo sia in condizioni di funzionamento sovrasterzante. In tal caso è possibile che esso presenti una condizione di equilibrio instabile, la quale richiede un angolo volante di segno opposto rispetto a quello corrente. Nei casi più estremi si può incorrere nella divergenza degli stati del veicolo, i quali portano in una condizione in cui la traiettoria ha un raggio di curvatura nullo, comunemente chiamata "testa-coda".

Metodo del piano delle fasi

Un approccio ben noto al problema è il "metodo del piano delle fasi". Si tratta di un metodo grafico incentrato su studi della dinamica del veicolo in campi di funzionamento fortemente non lineari. Tale metodo è comunemente utilizzato per la problematica esposta in quanto non possono essere utilizzati i criteri di stabilità ricavati dai modelli di dinamica lineare, essendo il fenomeno dell'instabilità direzionale centrato sul campo di funzionamento non lineare degli pneumatici [18].

Secondo questa teoria devono essere selezionati una serie di parametri, opportunamente scelti poiché influenzati dalla dinamica laterale del veicolo, per tracciare l'evoluzione delle traiettorie in un determinato piano. I parametri generalmente utilizzati nel corso degli studi riportati in letteratura sono i seguenti:

- v_y : velocità laterale.
- $\dot{\psi}$: velocità di imbardata.
- β : angolo di assetto, definito $\beta = \text{atan}(v_y/v_x)$
- $\dot{\beta}$: variazione dell'angolo di assetto nel tempo.
- α_f e α_r : angoli di deriva medi dei due assali.

I piani delle fasi all'interno dei quali è possibile descrivere le cosiddette traiettorie di fase sono i seguenti:

- $v_y - \dot{\psi}$
- $\beta - \dot{\psi}$
- $\beta - \dot{\beta}$
- $\alpha_f - \alpha_r$

Sebbene ogni coppia di parametri descriva le stesse informazioni in forma grafica, vi sono delle differenze imputabili alle seguenti caratteristiche:

- Misurabilità: in quanto alcuni parametri come $\dot{\psi}$ possono essere misurati direttamente, a differenza di stime che devono essere opportunamente calibrate ed effettuate su altri parametri, come ad esempio β .
- Rilevanza: in quanto vi sono parametri come β che hanno un'influenza maggiore sulle condizioni di stabilità del veicolo.
- Sensibilità: la quale viene misurata in termini di variazione della zona di stabilità in seguito a variazioni di condizione di funzionamento.

Nella maggior parte dei casi vengono utilizzati i piani $\beta - \dot{\psi}$ e $\beta - \dot{\beta}$ [18]. Deve essere specificato che la caratterizzazione degli pneumatici accoppiata a questo tipo di analisi deve essere di tipo non lineare, come quella ricavata con l'utilizzo del metodo *magic formula* sviluppato da Pacejka. Inoltre, è necessario precisare che questa tipologia di analisi viene condotta attraverso modelli 2DOF che isolano la dinamica laterale del veicolo, rendendo tali piani delle fasi dipendenti da un numero inferiore di parametri.

L'obiettivo di questo metodo è quello di ricavare delle mappe nei piani citati che mostrino graficamente l'evoluzione dei parametri associati a differenti traiettorie, ognuna delle quali avente stati iniziali differenti. L'evoluzione degli stati, a partire da una data condizione iniziale, viene rappresentata mediante un campo vettoriale. Si riporta un esempio di piano delle fasi $\beta - \dot{\psi}$ in figura 2.8.

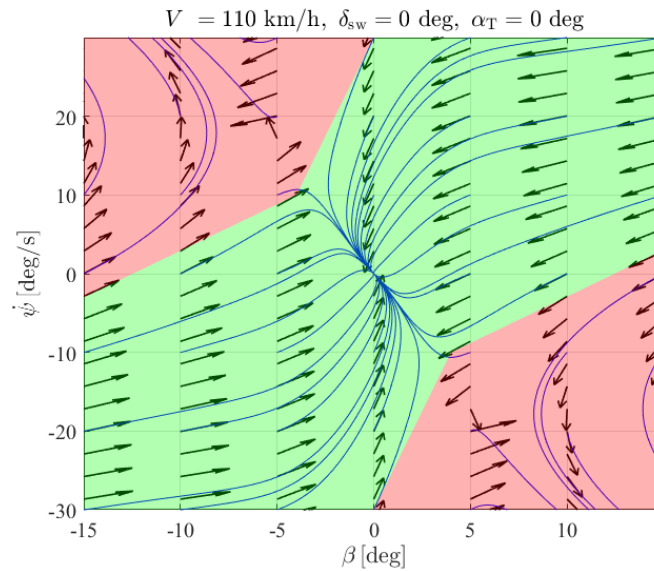


Figura 2.8: Piano delle fasi $\beta - \dot{\psi}$.

Per realizzare la mappa riportata in figura è stato necessario condurre un numero n di simulazioni variando gli stati di partenza β_0 e $\dot{\psi}_0$ e registrando l'evoluzione di tali grandezze. Tutte le simulazioni sono state effettuate a parità di altre condizioni quali velocità, angolo di sterzo, coefficiente di attrito dell'asfalto e inclinazione della superficie stradale.

Grazie alla rappresentazione grafica del campo vettoriale si nota che alcune delle traiettorie descritte all'interno del piano portano ad una convergenza degli stati verso il punto di equilibrio stabile. La porzione di grafico contenente le traiettorie che convergono verso tale punto è la zona di stabilità del veicolo, anche definita "Dominio di attrazione". Le zone che devono assolutamente essere evitate sono quelle che portano alla divergenza degli stati, anche chiamate zone di instabilità. L'approccio grafico consiste nell'individuazione dei limiti che separano le zone di stabilità da quelle di instabilità. In letteratura sono stati individuati differenti approcci al problema; si citano il *Diamond method* che consiste nel delimitare la zona che contorna il punto di equilibrio stabile con un rombo ed il *Two line method* che consiste nell'individuazione di due rette che al proprio interno contengono soltanto punti appartenenti al dominio di attrazione. Ciò che preme sottolineare è che non esiste un metodo universale di delimitazione delle zone, tuttavia, esse sono strettamente dipendenti dalla posizione dei due punti di equilibrio instabile. Individuarli vuol dire poter sfruttare la metodologia di delimitazione ritenuta più adeguata ed implementare la verifica nel processo di selezione della traiettoria di riferimento. La posizione dei due punti di equilibrio instabile è strettamente dipendente da diversi parametri [19]:

- V_x : velocità del veicolo. Incrementando la velocità a parità di altri parametri la zona di stabilità riduce la propria ampiezza ed i punti di equilibrio instabile acquisiscono una distanza inferiore dal punto di equilibrio stabile. In figura 2.9 è stato riportato un confronto tra due mappe ottenute rispettivamente a 50 km/h e 110 km/h .
- μ : coefficiente di attrito della strada. Peggiorando l'aderenza degli pneumatici la zona di stabilità riduce la propria ampiezza ed i punti di equilibrio instabile acquisiscono una distanza inferiore dal punto di equilibrio stabile.
- δ_f : angolo di sterzo delle ruote. Modificando l'angolo δ_f la zona stabile diventa asimmetrica. La condizione limite di questo fenomeno è la scomparsa totale del dominio di attrazione per valori di angolo di sterzo elevati. Si ha un riscontro grafico del fenomeno in figura 2.10, nella quale sono state riscontrate

le mappe ottenute nel piano delle fasi $\beta - \dot{\psi}$ con angolo di sterzo δ_f pari a 0° , grafico a sinistra, e 5° , grafico a destra.

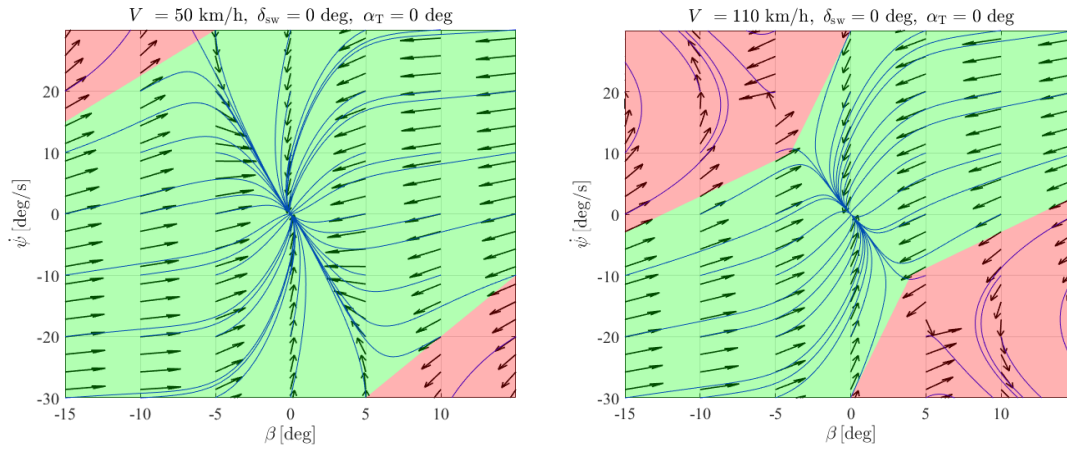


Figura 2.9: Influenza della velocità del veicolo sul piano delle fasi $\beta - \dot{\psi}$.

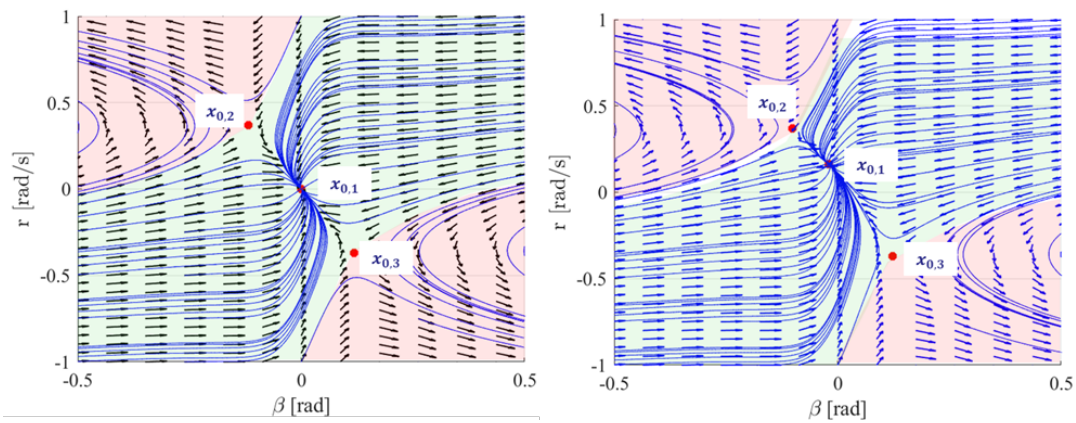


Figura 2.10: Influenza dell'angolo di sterzo del veicolo sul piano delle fasi $\beta - \dot{\psi}$. δ_f grafico a sinistra 0° , δ_f grafico a destra 5° .

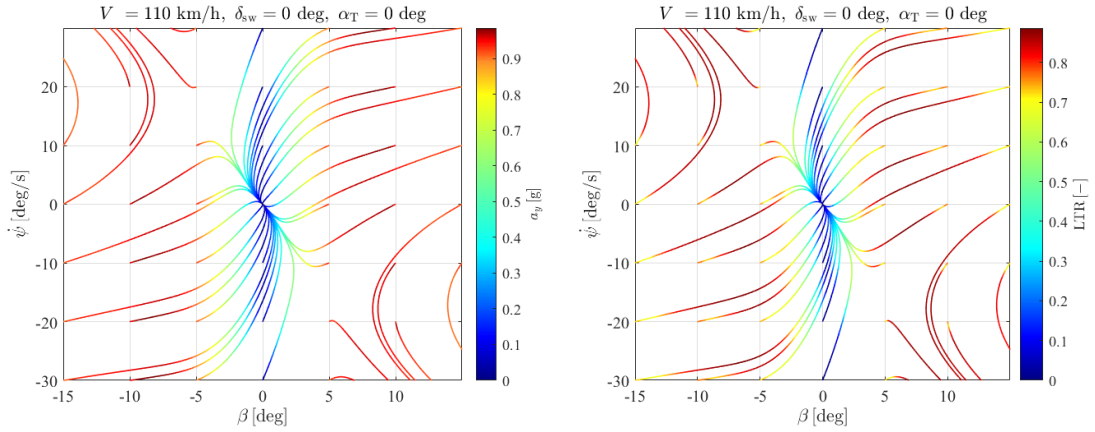


Figura 2.11: Correlazione dell'accelerazione laterale a_y e dell'indice LTR con i punti di funzionamento riportati sul piano delle fasi $\beta - \dot{\psi}$.

In figura 2.11 sono stati riportati i risultati di differenti simulazioni sul piano delle fasi e, attraverso un gradiente di colore, è stata riportata la variazione rispettivamente di accelerazione laterale a_y e di indice di trasferimento di carico verticale (LTR), il quale da un'indicazione accurata sul trasferimento di carico verticale nel corso della manovra.

$$LTR = \frac{Fz_L - Fz_R}{Fz_{tot}} \quad (2.14)$$

Dove Fz_L , Fz_R e Fz_{tot} sono le forze verticali rispettivamente agenti su: coppia di pneumatici a sinistra, coppia di pneumatici a destra e tutti e quattro gli pneumatici. Dall'esame dei grafici riportati emerge che ad una maggiore distanza dal punto di equilibrio corrisponde una criticità più elevata dello stato.

Tra le metodologie scelte per stimare i limiti delle zone possono essere utilizzati approcci maggiormente conservativi, in quanto in determinati punti di funzionamento all'interno del dominio di attrazione, i quali risultano comunque distanti dal punto di equilibrio stabile, il transitorio che porta il veicolo in condizione stabile risulta essere eccessivamente lento.

Cenni sull'implementazione della verifica di stabilità

Implementare questa verifica all'interno del modello implica l'identificazione di corrette metodologie di stima delle grandezze che descrivono gli stati all'interno del sistema e la scelta di della tecnica di identificazione dei limiti delle zone stabile/instabile.

La verifica avviene con una logica analoga a quella descritta nel caso cinematico, ovvero si confronta ciascun punto di ciascuna traiettoria con i rispettivi valori limite. Qualora si dovesse riscontrare un punto di una data traiettoria alternativa all'interno della zona definita instabile, essa deve essere necessariamente scartata dal set.

Le modalità di implementazione della verifica nel modello creato e la manipolazione dei dati ricavati dai piani delle fasi sono state spiegate nel dettaglio all'interno dei capitoli successivi, nei quali è stato illustrato l'approccio pratico della verifica.

2.3 Valutazione sulle possibili collisioni delle traiettorie alternative

Un'ulteriore verifica che deve necessariamente essere inserita all'interno del processo di selezione della traiettoria di riferimento è quella sulle collisioni con gli altri veicoli presenti nello scenario. Il funzionamento di tale verifica si basa su un algoritmo che sia in grado di stimare l'evoluzione delle traiettorie degli altri attori e sulla conoscenza dei relativi ingombri.

Supponendo di avere a disposizione tali dati, una possibile soluzione è quella di effettuare un confronto istante per istante fra le traiettorie dei corpi presenti nello scenario. Disponendo dei parametri descrittivi degli ingombri dei vari attori si possono escludere tutte le traiettorie che contengono una sovrapposizione dei corpi in un dato istante temporale.

2.3.1 Metodo capsule 2D

In ambiente MATLAB & Simulink il già citato *Navigation Toolbox* fornisce gli strumenti che permettono di effettuare la verifica sulle collisioni a partire dai dati citati. Il set di funzioni a disposizione risulta essere ben ottimizzato in quanto permette di adempiere allo scopo in maniera rapida e semplice.

La funzione principale si chiama *dynamicCapsuleList*, la quale crea due liste di oggetti separate. Tali liste contengono rispettivamente i corpi del veicolo *ego* e quelli degli altri attori presenti nello scenario; tali oggetti sono rappresentati come capsule in uno spazio 2D. Ognuno di essi viene identificato da tre elementi chiave [20]:

1. Un indice con il quale identificare il determinato veicolo all'interno della lista.
2. Una matrice contenente gli stati del veicolo in coordinate globali $[x, y, \psi]$.
3. Una struttura contenente i parametri descrittivi della geometria del veicolo, quali: la lunghezza della capsula che lo rappresenta, il suo raggio ed una matrice 3x3 rappresentante la trasformata utile a spostare l'origine locale della

capsula in un punto differente da quello di default (punto nero di coordinata (x, y) in figura 2.12).

La rappresentazione grafica della capsula è stata riportata in figura 2.12.

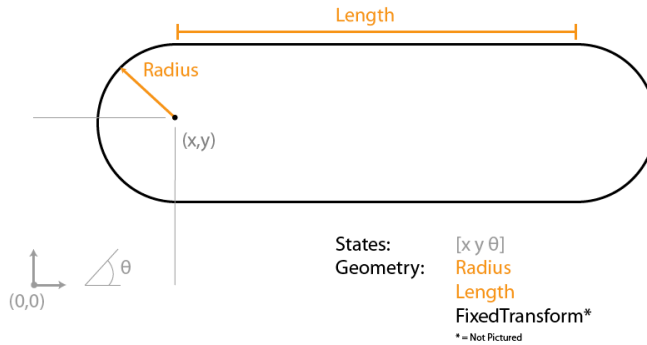


Figura 2.12: Geometria delle capsule 2D [20].

Nel successivo sottoparagrafo è stato riportato un esempio appositamente creato al fine di poter facilitare la comprensione dello strumento *dynamicCapsuleList* e di dare una dimostrazione tangibile del processo di verifica delle collisioni. Lo scenario di partenza è quello esaminato nel paragrafo primo in modo da dare continuità alla trattazione.

Esempio valutazione delle collisioni

L'esempio creato si articola in tre differenti sotto casi, attraverso i quali lo scenario è stato progressivamente modificato. Tutti i veicoli hanno le seguenti caratteristiche geometriche:

Lunghezza	5m
Raggio	1m
Trasformata	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Tabella 2.3: Geometria capsule - Ex. Dynamic Capsule-based List

Caso 1 - verifica di eventuali collisioni scenario di partenza: Si riportano in breve le caratteristiche dello scenario di partenza, la cui rappresentazione grafica è apprezzabile in figura 2.6:

- Una strada rettilinea a due corsie, ciascuna larga $3.6 m$.
- Un veicolo *ego* che percorre una delle due corsie ad una velocità costante pari a $20 m/s$
- Un veicolo che precede il veicolo *ego* percorrendo la strada nella medesima corsia e con la medesima direzione ad una velocità costante pari a $10 m/s$.
- Una distanza iniziale tra i due veicoli pari a $40 m$.

L'obiettivo posto è quello di verificare l'eventuale presenza di collisioni, lasciando inalterate velocità e traiettorie iniziali dei veicoli. In tal modo è possibile comprendere la metodologia da utilizzare con gli strumenti a disposizione. Si procede per punti.

1. La fase iniziale di questo metodo consiste nel creare l'oggetto contenente la struttura inizializzata delle due liste di attori richiamando la funzione *dynamicCapsuleList*:

$$obsList = dynamicCapsuleList; \tag{2.15}$$

Dopodiché si determina arbitrariamente un Δt come unità di campionamento temporale e si crea il vettore temporale. Calcolato il numero di elementi del vettore, che corrisponde al numero di step temporali della simulazione, si registra l'informazione all'interno dell'oggetto *dynamicCapsuleList*.

2. Il veicolo *ego* deve essere aggiunto all'interno dell'apposita lista nell'oggetto *dynamicCapsuleList*, fornendo tutte le informazioni che lo descrivono. Per la definizione degli stati del veicolo *ego* del primo scenario è stata utilizzata la funzione *linspace*, in modo da creare una matrice "states" con ascissa crescente in maniera costante e di una quantità congrua alla velocità del veicolo. La geometria è stata creata inserendo in un'unica struttura "geom" i tre parametri

descritti nella tabella 2.3. Ottenuti i tre elementi è stato possibile creare la struttura dedicata alla capsula del veicolo *ego*.

$$\begin{aligned} egoCapsule = struct('ID', egoID, 'States', states, 'Geometry', geom) \\ addEgo(obsList, egoCapsule) \end{aligned} \quad (2.16)$$

3. In maniera analoga si procede con l'inserimento degli ostali, ovvero degli altri attori presenti nello scenario, all'interno dell'apposita lista. In questo caso anche l'evoluzione degli stati del veicolo *lead* è stata descritta mediante la funzione *linspace*. La capsula del veicolo *lead* è stata creata ed inserita nella lista con le seguenti funzioni.

$$\begin{aligned} obsCapsule1 = struct('ID', obsID1, 'States', obsState1, ... \\ ...'Geometry', geom); \\ addObstacle(obsList, obsCapsule1); \end{aligned} \quad (2.17)$$

4. Infine viene effettuata la verifica sulla collisione dei veicoli nel corso dell'intervallo temporale descritto attraverso il seguente semplice comando:

$$collisions = checkCollision(obsList); \quad (2.18)$$

L'output "collisions" è un vettore di variabili booleane associate ad ogni singolo step temporale ed al valore 1 corrisponde lo stato di collisione ovvero una sovrapposizione delle aree occupate dai veicoli. Conseguentemente una data traiettoria verrà scartata in fase di verifica se vi sarà anche un singolo elemento *true* nell'output della funzione *checkCollision*.

Si riporta in figura 2.13 l'evoluzione degli stati del sistema ottenuta mediante rappresentazione capsule 2D.

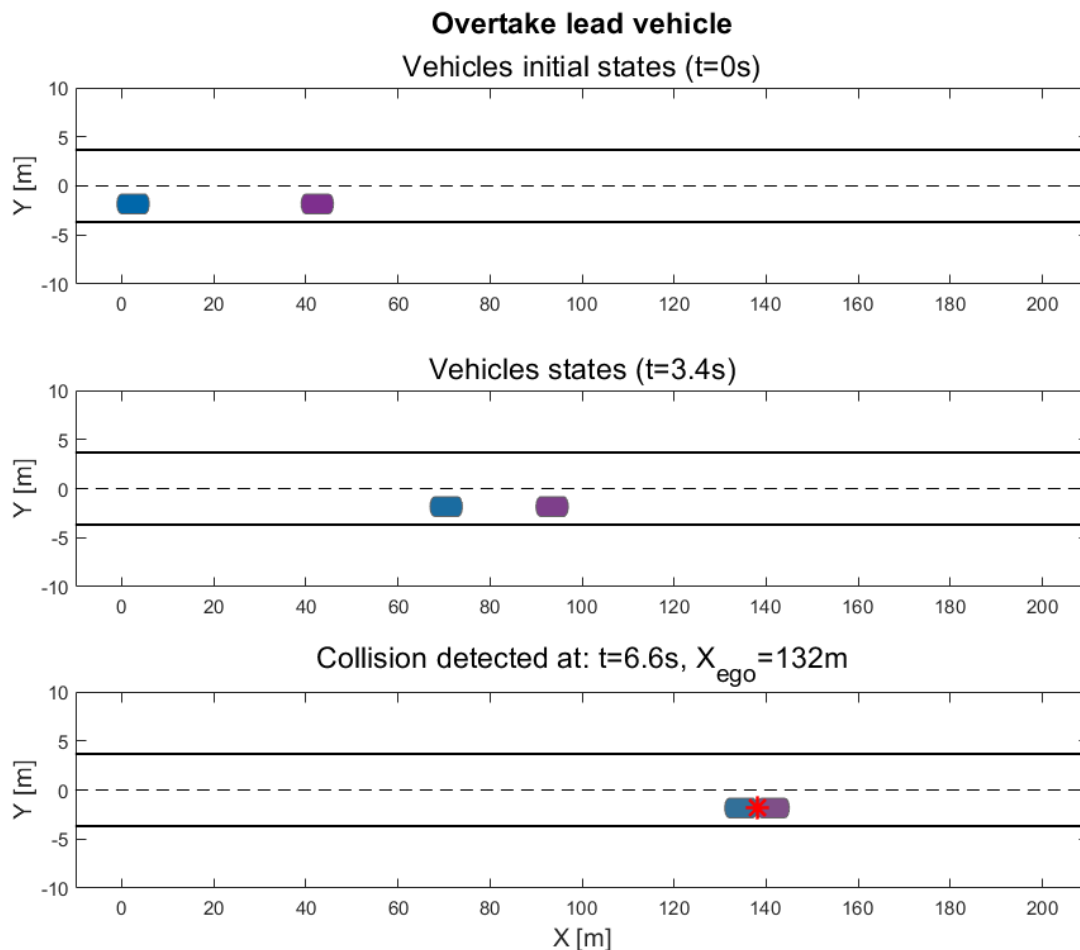


Figura 2.13: Evoluzione nel tempo degli stati dei veicoli e verifica sulle collisioni - scenario "sorpasso del veicolo *lead*".

Si nota nell'ultimo grafico il rilevamento della collisione dovuto alla differenza di velocità dei due veicoli. Si specifica che la distanza iniziale dei due veicoli è stata riferita alla distanza tra i punti a cui tali veicoli vengono associato ovvero il rispettivo centro dell'assale posteriore; tuttavia, per entrare in collisione i veicoli devono avvicinarsi di una distanza inferiore rispetto a quella riportata come distanza iniziale, in quanto basta che la parte frontale del veicolo *ego* entri in contatto con la parte posteriore del veicolo *lead*.

Caso 2 - verifica di eventuali collisioni con traiettoria di sorpasso:
Ipotizzando una reazione del sistema alla rilevazione di un'incipiente collisione come quella rappresentata nel primo caso, gli scenari del caso secondo e terzo di questo esempio sono stati costruiti in modo che il veicolo *ego* effettui una manovra di sorpasso ai danni del veicolo *lead*.

Il veicolo identificato *lead* è rimasto inalterato in tutti e tre gli elementi che ne descrivono la modellizzazione in capsula. Invece, il veicolo *ego* ha subito delle modifiche alla matrice degli stati, in quanto la sua traiettoria è deifferente. Per sviluppare una traiettoria idonea ad eseguire un sorpasso è stato utilizzato il metodo *connect* descritto in precedenza. La logica con la quale è stata costruita la traiettoria di cambio e rientro corsia è stata la seguente:

- Si effettua un primo cambio di corsia all'istante $t = 0s$ con orizzonte temporale $Th = 3.0s$.
- Si effettua la seconda manovra di cambio corsia solo dopo aver fatto completamente sfilare il veicolo *lead*. Anche in questo caso è stato adottato un orizzonte temporale $Th = 3.0s$.
- Per semplicità è stato scelto di mantenere costante la velocità del veicolo *ego* lungo la coordinata x , nonché lungo la coordinata s con sistema di riferimento di Frenét.

Dopo aver aggiornato la matrice "states" essa è stata memorizzata nella struttura rappresentante la relativa capsula.

In questo caso non sono state riscontrate collisioni, come si può verificare dai grafici in figura 2.14.

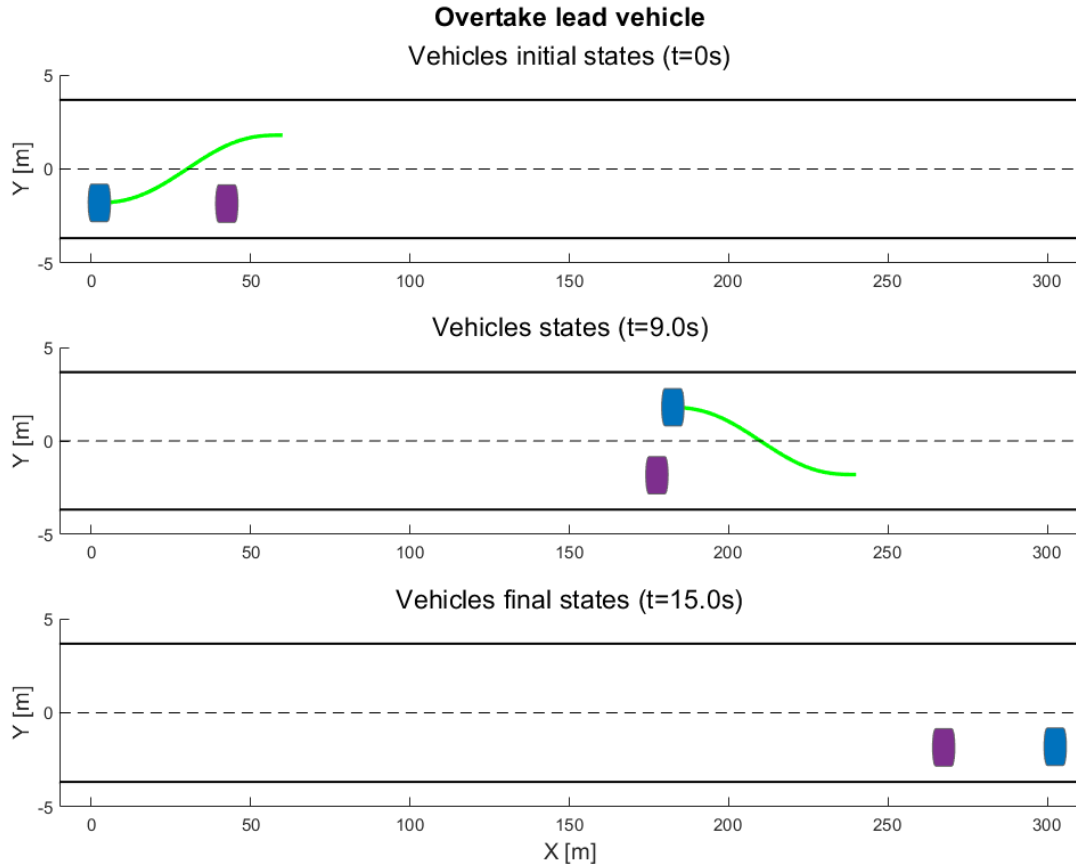


Figura 2.14: Evoluzione nel tempo degli stati dei veicoli e verifica sulle collisioni - scenario "sorpasso del veicolo *lead* con sorpasso".

Caso 3 - verifica di eventuali collisioni con veicolo transitante nel senso di marcia opposto: In questa terza ed ultima versione dell'esempio lo scenario è stato modificato ulteriormente, aumentandone la complessità mediante l'aggiunta di un ulteriore veicolo transitante nella corsia con senso di marcia opposto. Esso è stato denominato veicolo *oncoming* ed ha una velocità pari a 10 m/s ed una distanza iniziale dal veicolo *ego* pari a $\Delta s = 200\text{ m}$.

Si riporta in figura 2.15 una rappresentazione dello stato iniziale dello scenario in assenza di collisioni.

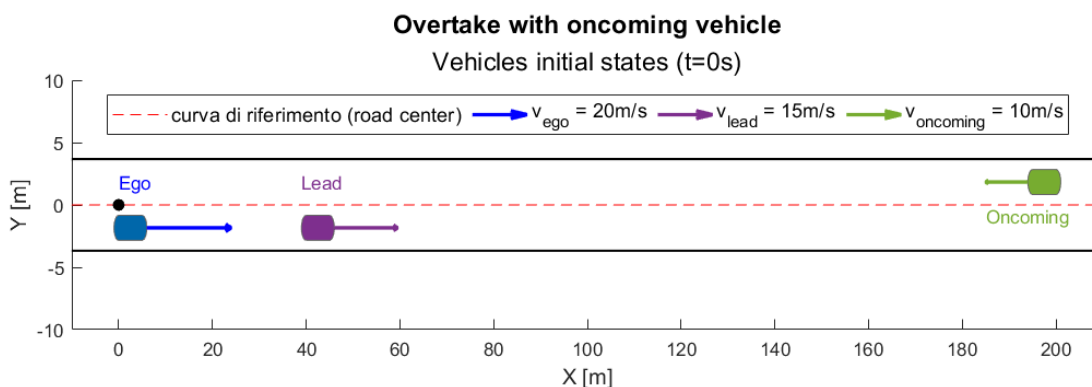


Figura 2.15: Stati iniziali $t = 0 s$ - scenario "sorpasso con veicolo in senso di marcia opposto".

In questo caso le traiettorie dei veicoli *ego* e *lead* sono le medesime del caso precedente. In figura 2.16 si riporta il risultato dell'analisi sulle collisioni.

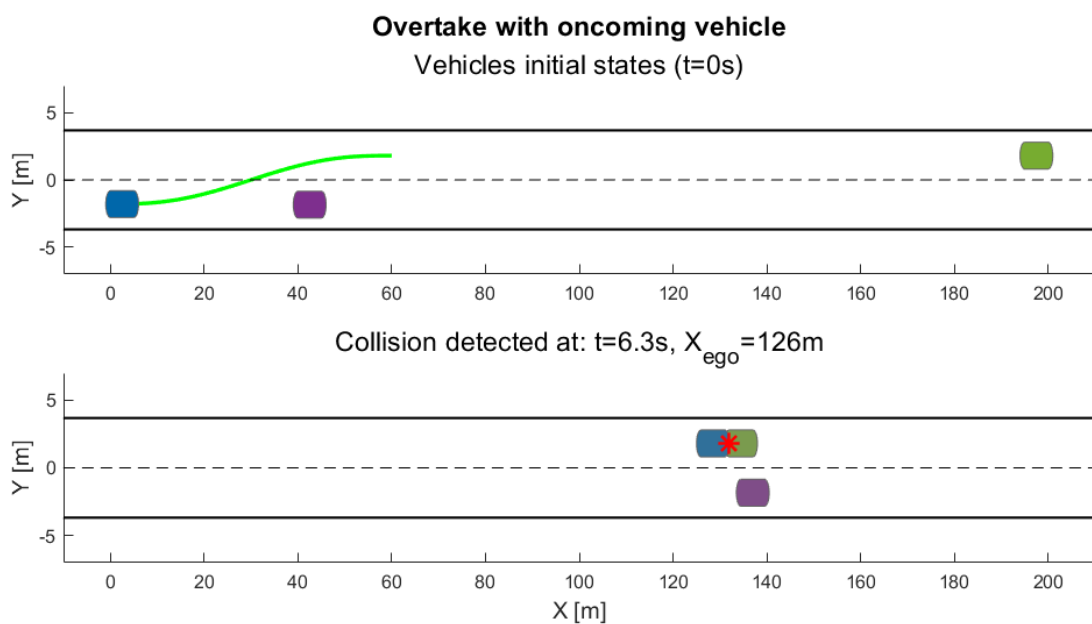


Figura 2.16: Evoluzione nel tempo degli stati dei veicoli e verifica sulle collisioni - scenario "sorpasso con veicolo in senso di marcia opposto".

Come si può notare viene riscontrata una collisione nel corso dell'esecuzione della manovra di sorpasso mediante sovrapposizione delle aree occupate dalle capsule.

La configurazione attuale dello scenario è stata analizzata mediante simulazioni complesse nel capitolo quarto, attraverso le quali sono state implementate strategie in grado di riconoscere la criticità presentata e di modificare la traiettoria e/o la velocità del veicolo *ego* in modo tale da non incorrere in collisioni.

Tale caso studio è stato sviluppato con l'intento di evidenziare la criticità presentata; in quanto per essere superata è necessario dotare il veicolo di un sistema assistenziale come quello oggetto di questo caso studio.

Capitolo 3

Ambiente di simulazione

L'ambiente industriale richiede notoriamente un'accelerazione costante dei processi di ricerca e sviluppo; dunque, risulta essere di fondamentale importanza disporre dei corretti strumenti per ridurre i tempi di progettazione, massimizzandone l'efficienza e l'accuratezza dei risultati.

Non fa eccezione il settore automobilistico, le cui aziende basano la propria progettazione per la maggior parte in ambienti virtuali sempre più accurati. Le simulazioni computerizzate hanno ridotto drasticamente i costi ed i tempi di risoluzione di problemi complessi. Le nuove frontiere della simulazione in campo automobilistico stanno portando ad un'integrazione sempre maggiore della stessa all'interno dell'automobile. In veicoli ad alto livello di automazione, risulta essere necessario avere dei sistemi *on-board* che siano in grado di analizzare i segnali provenienti dai sensori e di stabilire i comandi da dover attuare, prevedendo l'evoluzione dello scenario ed il conseguente comportamento del veicolo.

Le fasi di progettazione e sviluppo di un sistema assistenziale avanzato, come quello oggetto di questo studio, sono basate interamente su ambienti di simulazione; in quanto, la necessità di effettuare le già citate previsioni, relative a scenario e veicolo, è parte integrante e fondamentale del sistema assistenziale stesso.

3.1 Modello Simulink del sistema assistenziale

Con il fine di proporre una ricerca totalmente incentrata sullo studio della manovra di sorpasso e sulla possibilità di realizzarlo in maniera autonoma, è stato scelto di non costruire ex novo lo strumento di simulazione ma di partire dalla base di un modello sviluppato dalla MathWorks e disponibile nelle proprie librerie. Nella fase iniziale dello studio la priorità è stata attribuita a quei componenti del modello dedicati al *path planning* (pianificazione della traiettoria), ritenendo poi indispensabile la possibilità di poter intervenire sui parametri di calibrazione e sulle logiche di funzionamento degli stessi. Per tale motivo la scelta è inizialmente ricaduta sullo sviluppo di un modello comprensivo del solo pianificatore; in maniera tale da poter valutare al meglio gli output prodotti dal sistema assistenziale, ovvero le traiettorie da seguire, eliminando dai risultati le influenze di altri componenti come controllore e modello di veicolo.

Il modello utilizzato fa parte dei modelli forniti con l'*Automated Driving Toolbox* prodotto dalla MathWorks per i software MatLab & Simulink. Si tratta di un tool dedicato alla progettazione, alla simulazione e ai test di sistemi ADAS e di guida autonoma. Esso permette di poter lavorare su progetti dedicati ai sensori dei veicoli, ai controllori o al *path planning* come in questo caso [21]. Oltre a fornire modelli veicolo, come quello oggetto di questo paragrafo, l'*Automated Driving Toolbox* risulta essere particolarmente efficiente per lo studio di questa tematica, in quanto comprendere applicazioni totalmente dedicate ad alcuni aspetti della simulazione in ambito Automotive, come il *Driving Scenario Designer* [22]. Inoltre, è particolarmente utile la compatibilità con molteplici formati, ad esempio *OpenDRIVE*, che permette di importare file utili per la creazione di appositi scenari. Al fine di poter sfruttare tutti gli strumenti dedicati allo studio della tematica in questione è stato utilizzato anche il *Navigation Toolbox* [23]. Si tratta di un tool prodotto dalla MathWorks dedicato agli algoritmi e agli strumenti per l'analisi di *motion planning*, localizzazione simultanea e navigazione inerziale.

All'interno di questo paragrafo verrà analizzato nel dettaglio il modello citato; tuttavia, nel corso dello studio, esso è stato integrato all'interno di un modello più complesso e completo, presentato nel paragrafo 3.2.

3.1.1 Highway Lane Change Planner Test Bench

Il modello utilizzato in questa ricerca è l'*Highway Lane Change Planner*, disponibile all'interno dell'*Automated Driving Toolbox* di MatLab [24]. Questo modello permette al veicolo *ego* di cambiare corsia in maniera autonoma all'interno di scenari autostradali e gestisce differenti comportamenti del pilota per un medesimo scenario. Ad esempio, in base al contesto il veicolo può attivare le funzioni di cruise control (CC), di lead car following (LCA) o di sorpasso automatico (LC)[24]. Il modello è stato sviluppato in ambiente MatLab & Simulink e se ne può apprezzare la struttura generale in figura 3.1. Ognuno dei blocchi rappresentati è stato sviluppato in sottosistemi ed il funzionamento dell'intero modello è regolato da *script MatLab*.

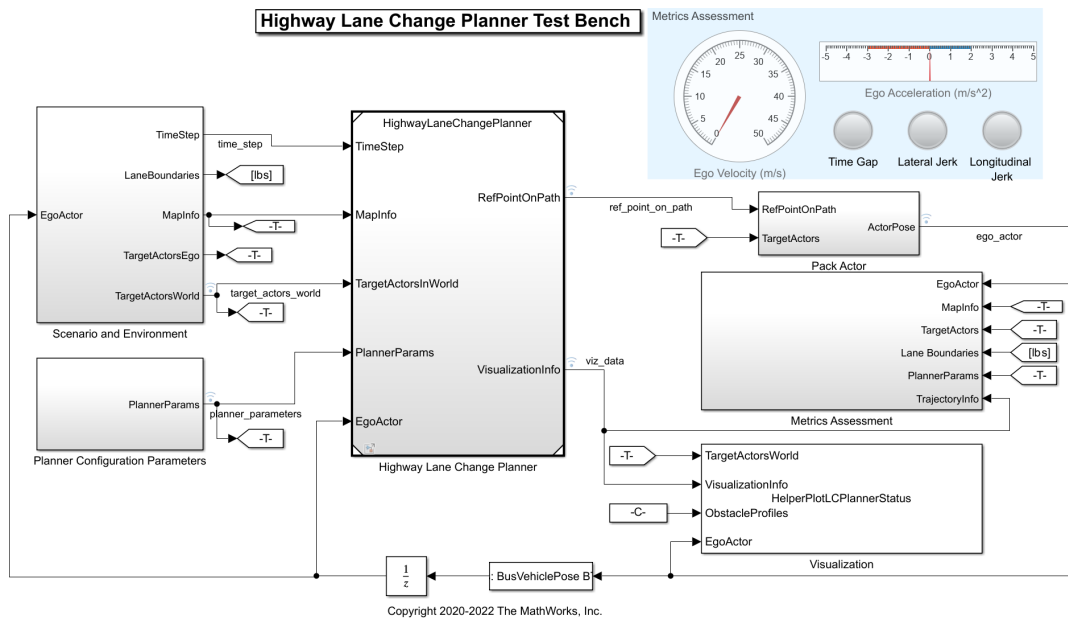


Figura 3.1: Struttura generale Highway Lane Change Planner.

L'input principale del sistema è lo scenario, il quale viene caricato mediante un apposito *script*. Si tratta di un oggetto con formato *drivingScenario* che fornisce informazioni circa la strada e gli attori principali, quali: veicolo ego, altri veicoli presenti nello scenario, barriere, ostacoli e così via. Le caratteristiche del *drivingScenario object*, le relative metodologie di creazione e di inserimento all'interno del modello sono state discusse nel paragrafo 3.3. Come si può notare in figura 3.1 il modello si compone di vari sottosistemi principali, ognuno dei quali è stato analizzato nel dettaglio in seguito. In figura 3.2 è stato riportato un diagramma di flusso realizzato con lo scopo di mettere in mostra la sequenza con la quale vengono processate le informazioni all'interno del modello.

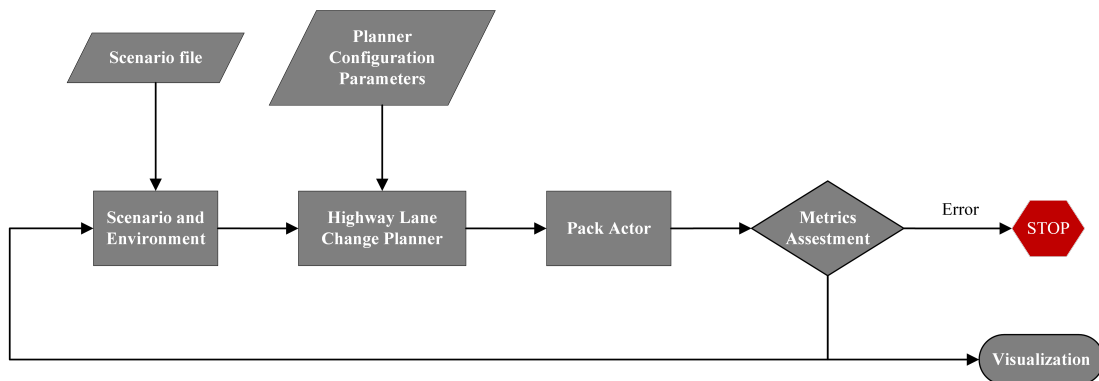


Figura 3.2: Diagramma di flusso funzionamento generale *Highway Lane Change Palnner model*.

Si riporta una breve descrizione dei singoli sottosistemi e delle logica di funzionamento del modello rappresentata nel diagramma.

- Scenario and Environment: Sottosistema dedicato alla lettura del *drivingScenario object* ricevuto in input, utile a produrre i segnali necessari alla simulazione.
- Planner Configuration Parameters: Sottosistema dedicato a tradurre in segnali i parametri necessari a configurare il pianificatore di traiettoria; tali dati vengono importati direttamente dal Workspace.

- Highway Lane Change Planner: Sottosistema contenente il pianificatore di traiettoria.
- Pack Actor: Sottosistema dedicato alla creazione del segnale di tipo Bus contenente lo stato attuale del veicolo *ego*.
- Metrics Assessment: Sottosistema dedicato alla valutazione dei segnali prodotti nel corso della simulazione. Esso genera degli allarmi o interrompe la simulazione nel caso in cui i parametri limite imposti non venissero rispettati.
- Visualization: Sottosistema dedicato alla creazione e all'aggiornamento dell'interfaccia grafica.

Per eseguire un'analisi sintetica di come vengono processati i segnali in input al modello è necessario partire dall'istante $t=0$, in cui le coordinate attuali del veicolo coincidono con quelle iniziali "egoInitialPose". Le quali vengono prelevate direttamente dal Workspace tramite il blocco *Scenario and Environment*. Tali coordinate vengono confrontate con il file "scenario", anch'esso prelevato dal Workspace, per produrre gli stati di tutti gli attori in coordinate sia globali che locali (calcolate rispetto alla posizione del veicolo *ego*). Quanto prodotto viene poi elaborato dal blocco *Highway Lane Change Planner*, il cui scopo è quello di fornire in output un segnale contenente la traiettoria che il veicolo deve seguire, calcolata dopo avere:

1. Confrontato lo stato attuale con quello dei veicoli ad esso più vicini.
2. Eliminato tutte le possibili soluzioni appartenenti a traiettorie concludenti con collisioni o non compatibili con la dinamica del veicolo.
3. Valutato quelle percorribili in base all'obiettivo prescelto dal pianificatore, la valutazione viene effettuata associando a ciascuna traiettoria un determinato parametro di costo.

Deve essere necessariamente ricordato che in questa versione del modello, dove è stato isolato il pianificatore di traiettorie, si assume che il veicolo segua puntualmente la traiettoria fornita in output dal blocco *Highway Lane Change Planner*; in quanto non sono presenti controllore e modello di veicolo.

Tutte le costanti utili a calibrare il pianificatore vengono importate dal blocco *Planner Configuration Parameters*, il quale preleva i dati dal *WorkSpace*. Dopo aver ottenuto il segnale contenente la traiettoria da seguire, il blocco *Pack Actor* convoglia i relativi segnali all'interno di un apposito Bus. Quest'ultimo è poi valutato dal blocco *Metrics Assessment*, all'interno del quale vengono effettuati controlli su un ampio range di parametri. Qualora venissero riscontrati errori la simulazione si interrompe; in caso contrario, il segnale viene fornito in input al blocco *scenario and Environment* come nuovo stato attuale del veicolo *ego*. In tale blocco viene effettuato un confronto tra la nuova posizione e lo scenario, in modo da ottenere i Bus contenenti le nuove posizioni relative tra veicolo *ego* e tutti gli altri veicoli presenti nello scenario. Dopodiché il meccanismo si ripete fino al raggiungimento dello "StopTime" o di un eventuale errore.

3.1.2 Scenario and Environment Subsystem

Il sottosistema *Scenario and Environment* si occupa di fornire al modello i segnali relativi agli attori, ai confini delle corsie e alle strade.

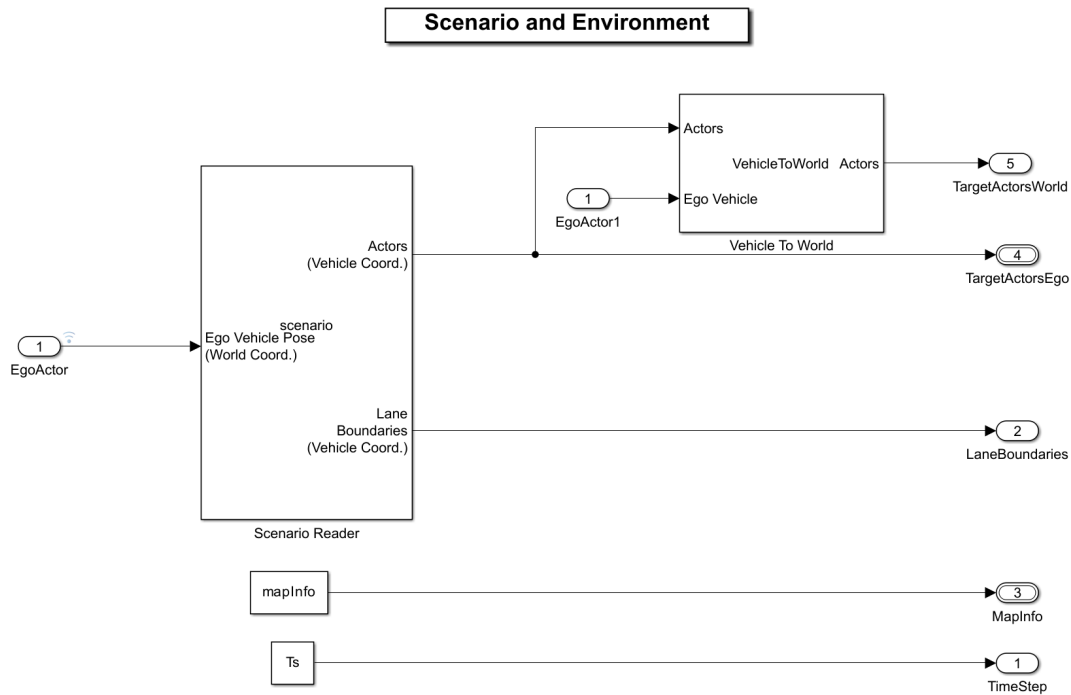


Figura 3.3: Scenario and Environment subsystem.

In figura 3.3 si possono notare due ulteriori blocchi di cui si compone il sottosistema. Sia il blocco *Scenario Reader* che il blocco *Vehicle To World* fanno parte delle librerie Simulink dedicate all'*Automated Driving Toolbox*. Si procede di seguito con l'analisi dettagliata del loro funzionamento.

Scenario Reader: Si tratta di un blocco dedicato alla lettura di strade e attori da uno *scenario file*; il quale può essere creato come *drivingScenario object* o direttamente come output del *Driving Scenario Designer tool* [25]. Quest'ultima è un'applicazione dedicata alla creazione di scenari per simulazioni in ambito automotive, il cui funzionamento è stato discusso nel paragrafo 3.3. Come si nota in figura 3.3, questo blocco fornisce come output le coordinate degli attori e la posizione dei limiti delle corsie, in questo caso con sistema di riferimento locale

(rispetto al veicolo *ego*). L'input del blocco è costituito dal segnale "EgoActor", contenente le coordinate del veicolo *ego* con sistema di riferimento globale, dal segnale "egoInitialPose", contenente lo stato di partenza del veicolo, e dal file scenario presente nel Workspace. La scheda di configurazione del blocco è stata riportata in figura 3.4 al fine di permettere una maggiore comprensione del funzionamento.

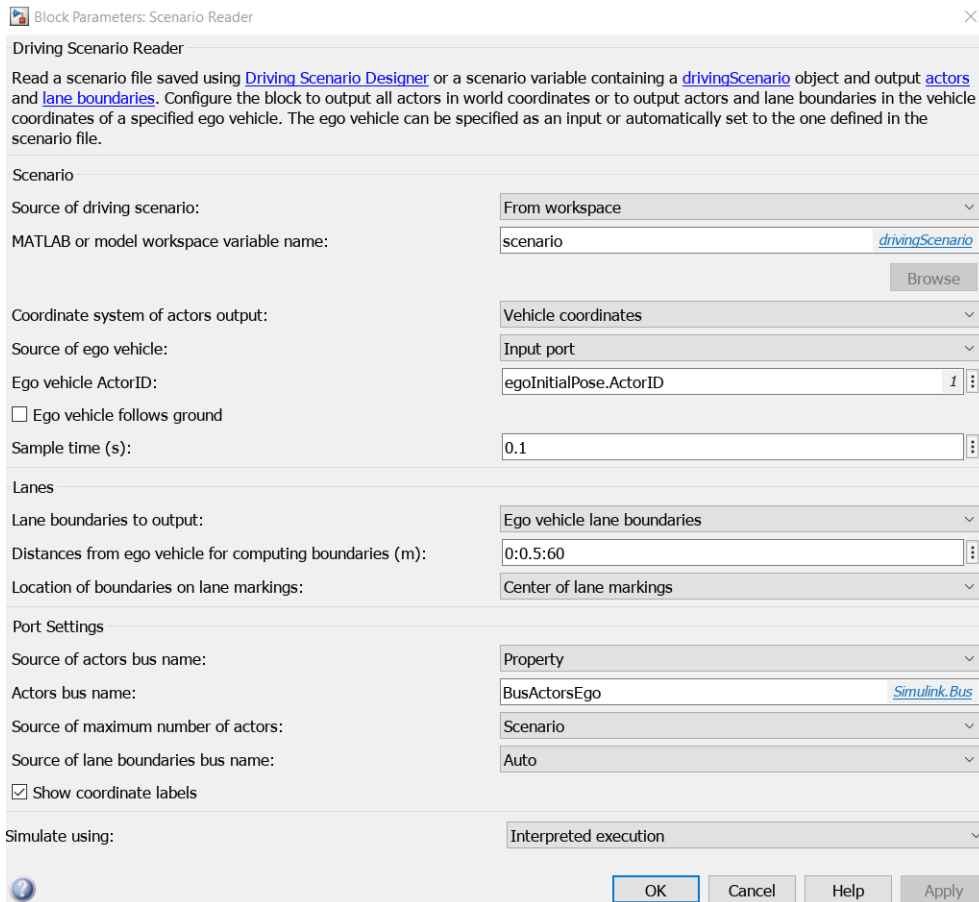


Figura 3.4: Scheda di configurazione blocco Scenario Reader.

Fra le possibilità di configurazione del blocco analizzato spiccano la possibilità di scegliere il tipo di coordinate dell'output (locale o globale) e la distanza dal veicolo *ego* da tenere in considerazione per il calcolo dei limiti di carreggiata.

Vehicle To World: Questo blocco viene utilizzato per modificare il tipo di coordinate degli attori. Riceve in input le coordinate locali degli attori e le coordinate globali del veicolo *ego*; computando i due segnali ne ricava le coordinate di tipo globale di tutti gli attori.

Oltre ai segnali provenienti dai blocchi precedentemente analizzati, il sottosistema *Scenario and Environment* fornisce come output "MapInfo" e "Ts". Si tratta di due segnali provenienti da due blocchi di tipo *Constant*, grazie ai quali vengono importati dal Workspace il tempo di campionamento e le informazioni relative alla strada. "MapInfo" contiene:

- Il numero di corsie della strada.
- L'ampiezza delle corsie.
- I waypoints che compongono la linea di simmetria della carreggiata denominati "roadCenters".
- Il numero di waypoints.

Dalla struttura generale del modello riportata in figura 3.1 si nota che il segnale "EgoActor" proviene dall'output del sottosistema *Pack Actor* processato attraverso un blocco *delay*, come illustrato nella analisi sintetica del modello effettuata nel paragrafo 3.1.1.

3.1.3 Planner Configuration Parameters Subsystem

Nel sottosistema *Planner Configuration Parameters* alcune delle variabili presenti nel Workspace vengono fornite in input al modello tramite dei blocchi *Constant*. I cui segnali a loro volta compongono una *Bus structure* creata mediante il blocco *Bus Creator*. Il Bus che ne deriva viene poi collegato alla parte di modello dedicata al pianificatore di traiettoria.

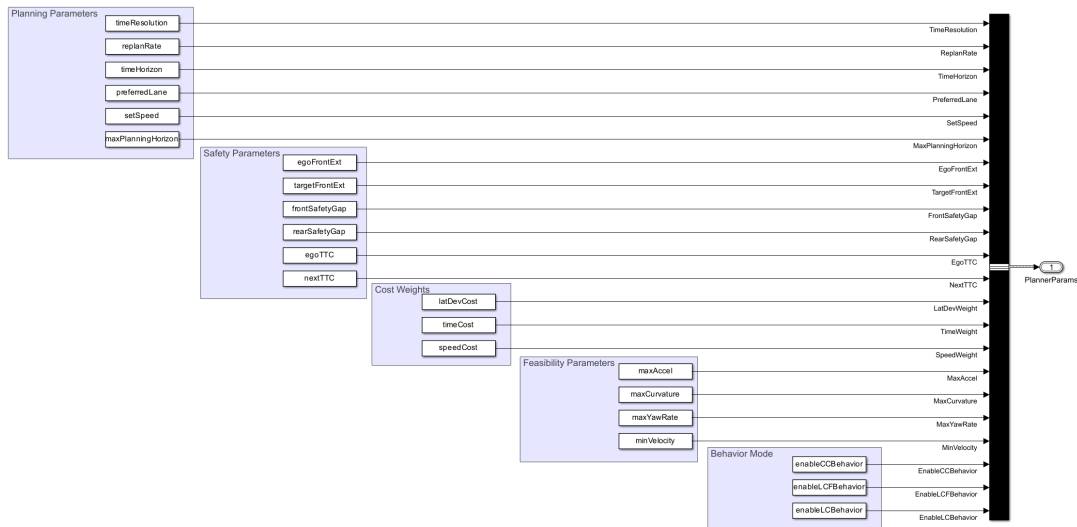


Figura 3.5: Planner Configuration Parameters subsystem.

Come mostrato in figura 3.5 i parametri di configurazione sono stati raggruppati in aree. Si riporta di seguito l'elenco e la descrizione dei segnali che compongono il Bus.

- *Planning Parameters*: I primi tre parametri di questa area definiscono delle variabili temporali in secondi. "timeResolution" viene utilizzato per il tempo di campionamento, "replaneRate" definisce il tempo di ripianificazione della traiettoria e "timeHorizon" è un vettore di orizzonti temporali per il campionamento delle traiettorie. Come parametri di pianificazione sono stati creati: "preferredLane" che identifica la corsia di preferenza per il veicolo ego, "setSpeed" che ne definisce la velocità ed infine "maxPlanningHorizon" utile a specificare la massima distanza longitudinale di pianificazione.

- *Safety Parameters*: In quest'area vengono definiti i parametri utili ai controlli di sicurezza della manovra. Nello specifico "egoFrontExt" e "targetFrontExt" identificano l'estensione frontale dei veicoli ego e target, dati utilizzati durante la verifica di collisione. "frontSafetyGap" e "rearSafetyGap" sono stati introdotti per specificare la distanza di sicurezza fra il veicolo ego e gli altri veicoli presenti nello scenario. Infine, "egoTTC" e "nextTTC" definiscono i limiti relativi al tempo di collisione con altri veicoli dello scenario.
- *Cost Weights*: All'interno di quest'area sono state raggruppate le variabili utilizzate per stabilire il peso dei contributi della funzione di costo, utilizzata per ordinare secondo una logica preferenziale gli stati terminali campionati dal pianificatore. Queste variabili sono: "latDevCost", "timeCost" e "speedCost"; le quali identificano il peso rispettivamente di: distanza laterale, orizzonte temporale e velocità.
- *Feasibility Parameters*: Nell'area in esame sono stati introdotti i parametri: "maxAccel", "maxCurvature", "maxYawRate" e "minVelocity". Si tratta di variabili che definiscono limiti utili alla verifica di fattibilità di una certa traiettoria in termini cinematici. Essi sono relativi a: accelerazione, curvatura, velocità di imbardata e velocità del veicolo.
- *Behavior Mode*: Quest'ultima area è dedicata al comportamento del modello. Come detto nel paragrafo 3.1.1, esso è in grado di operare nelle modalità: cruise control (CC), lead car following (LCF) e lane change (LC). Tramite le variabili "enableCCBehavior", "enableLCFBehavior" ed "enableLCBehavior" vengono introdotti dei cosiddetti flag che permettono di disabilitare la possibilità di usufruire di una data modalità di funzionamento.

Nella tabella 3.1 sono stati riportati i valori di default dei parametri elencati che sono stati scelti per effettuare le simulazioni. Tali parametri sono modificabili; tuttavia, è stato ritenuto utile fornire un esempio dei valori scelti; in modo da fornire un parametro di confronto per possibili studi di sensitività del modello.

Area	Parametro	Valore	u.tà
Panning Parameters	timeResolution	0.1	
	replaneRate	1	<i>s</i>
	timeHorizon	[1, 2, 3]	<i>s</i>
	preferredLane	/	
	setSpeed	/	<i>m/s</i>
	maxPlanningHorizon	80	<i>m</i>
Safety Parameters	egoFrontExt	5	<i>m</i>
	targetFrontExt	5	<i>m</i>
	frontSafetyGap	30	<i>m</i>
	rearSafetyGap	10	<i>m</i>
	egoTTC	4	<i>s</i>
	nextTTC	4	<i>s</i>
Cost Weights	latDevCost	1	
	timeCost	-1	
	speedCost	1	
Feasibility Parameters	maxAccel	5	<i>m/s²</i>
	maxCurvature	1	<i>1/m</i>
	maxYawRate	80	<i>°/s</i>
	minVelocity	0	<i>m/s</i>
Behavior Mode	enableCCBehavior	1	
	enableLCFBehavior	1	
	enableLCBehavior	1	

Tabella 3.1: Planning Configuration Parameters values

Si specifica che a differenza di tutti gli altri parametri elencati, i quali sono definiti in fase di calibrazione del pianificatore, il parametro "setSpeed" ed il valore iniziale del parametro "preferredLane" vengono definiti all'interno dello scenario. Per una definizione più dettagliata delle grandezze elencate si rimanda all'analisi, riportata di seguito, delle singole funzioni presenti nel modello, nelle quali tali parametri vengono processati.

3.1.4 Highway Lane Change Planner Subsystem

Il sottosistema *Highway Lane Change Planner* è la parte di maggior importanza dell'omonimo modello, poiché è quella dedicata al pianificatore di traiettoria. Dallo schema a blocchi generale in figura 3.1 emerge che l'insieme dei segnali in input del sottosistema in esame è formato dagli output prodotti dai sottosistemi *Scenario and Environment* e *Planner Configuration Parameters*, rispettivamente analizzati nei paragrafi 3.1.2 e 3.1.3, e dal Bus "egoActor", output del sottosistema *Pack Actor* contenente lo stato attuale del veicolo *ego*.

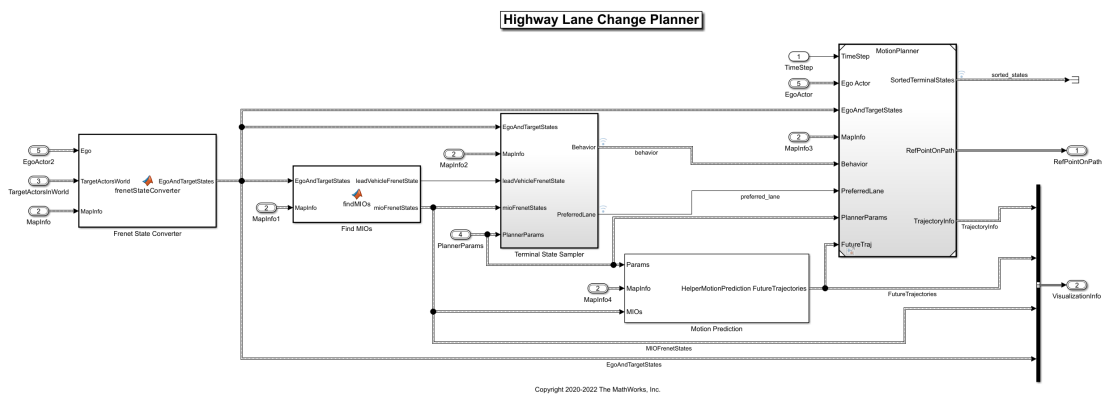


Figura 3.6: Highway Lane Change subsystem.

Come si può notare in figura 3.6, il primo blocco a processare i segnali in input al sottosistema è il *Frénet State Converter*. Si tratta di un blocco *MatLab function*, il quale al proprio interno ospita un codice *MatLab* che comunica con il modello *Simulink* esclusivamente attraverso i segnali di input e di output della funzione. Esso genera un cambio di coordinate degli stati contenuti nei segnali in input, modificando il sistema di riferimento da globale a Frénet. L'espedito utilizzato è di uso comune negli studi dedicati alla tematica del *path planning* e consiste nell'adoptare un sistema di riferimento mobile descritto da una coordinata curvilinea e da una normale a quest'ultima. Lo specifico argomento del sistema di riferimento Frénet è stato analizzato nel dettaglio all'interno del capitolo secondo.

Frénet State Coverter: Come anticipato, si tratta di una funzione dedita al cambio di coordinate (da globali a Frénet); tuttavia, non si tratta semplicemente di un isolato cambio di coordinate ma della creazione di un Bus "EgoAndTargetStates".

Esso convoglia in un'unica uscita tutte le informazioni dello scenario, tradotte nel nuovo sistema di riferimento. Inoltre, è bene precisare che il modello è stato progettato in modo da associare la curva di riferimento delle coordinate di Frenét alla *center line* della carreggiata; questa è un'assunzione effettuata ogni qual volta viene effettuato questo cambio di coordinate. La posizione del veicolo viene quindi descritta dalla coordinata curvilinea s e dalla distanza d dalla linea di mezzzeria della strada.

$$[x, y, \psi, \kappa, v, a] \rightarrow [s, \dot{s}, \ddot{s}, d, d', d''] \quad (3.1)$$

Con il fine di agevolarne la comprensione, è stato realizzato un diagramma di flusso, riportato in figura 3.7, che sintetizza i principali passaggi logici dell'algoritmo realizzato dalla funzione *frenetStateConverter*.

Per un'analisi di maggior dettaglio si rimanda all'appendice A, all'interno della quale sono state analizzate tutte le funzioni richiamate dal modello. In tal caso si noti che all'interno della funzione ne è stata richiamata un'altra denominata *Helper* e nello specifico *HelperLCPlannerDefaultData*. Si tratta di un espediente utilizzato per evitare di appesantire eccessivamente il corpo delle funzioni all'interno del modello Simulink. Il progetto è stato sviluppato corredando gli script principali e la parte sviluppata su Simulink di un certo numero di funzioni *Helper*, utili ad assolvere compiti richiesti in più parti del modello.

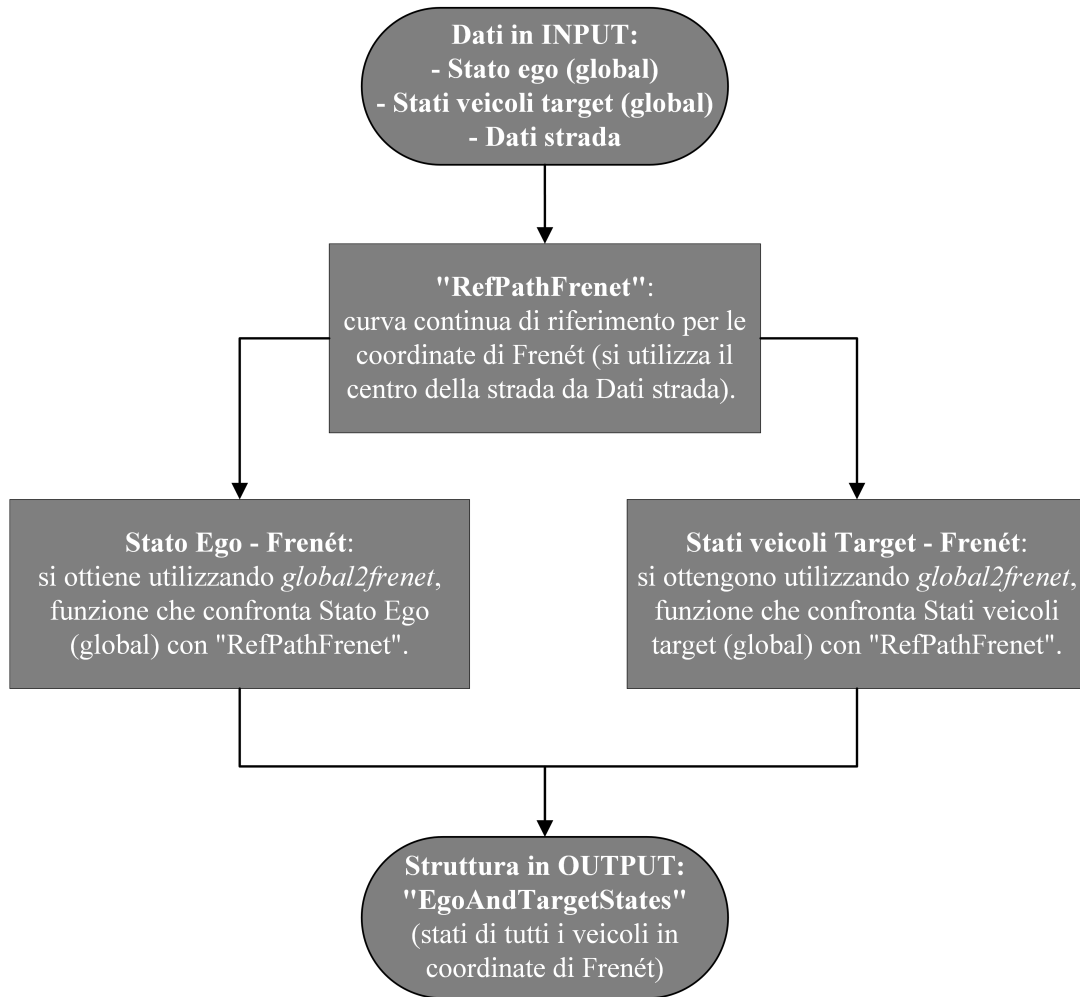


Figura 3.7: Diagramma di flusso Frenét State Converter

Find MIOs: Il segnale bus creato attraverso il blocco precedente viene poi rielaborato in un'ulteriore blocco *MATLAB function*. Lo scopo del blocco *Find MIOs*, dove MIO sta per *Most Important Objects*, è quello di individuare i veicoli considerati più rilevanti per lo sviluppo della manovra all'interno dello scenario.

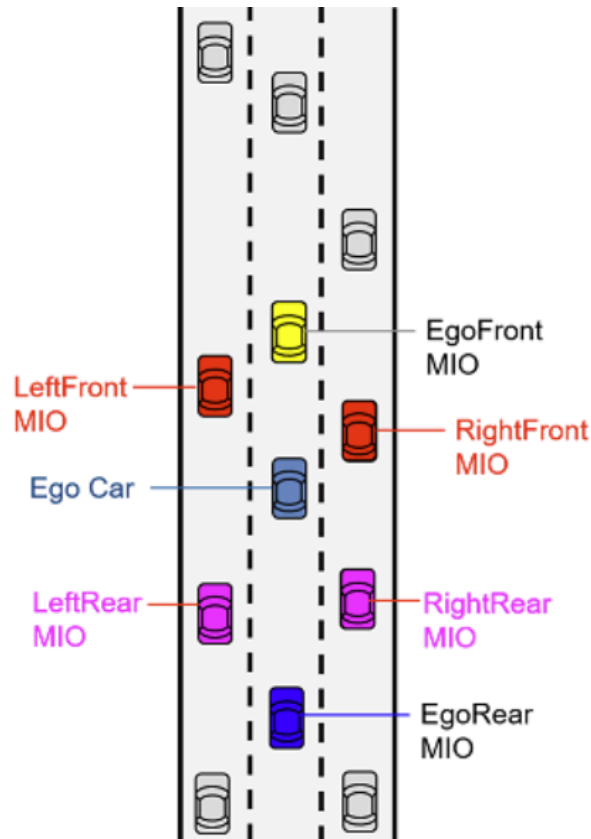


Figura 3.8: Most Important Objects [24].

Come si può notare in figura 3.8, esistono diversi veicoli che possono essere dichiarati "importanti"; quello che li contraddistingue è l'immediata prossimità al veicolo ego secondo una determinata direzione. La funzione contenuta all'interno di tale blocco è denominata *findMIOs* e si occupa di allocare nelle opportune variabili i segnali che riceve in input, dopodiché richiama la funzione *helperfindMIOs*, la quale contiene l'algoritmo di individuazione dei veicoli MIO. Entrambe le funzioni sono state analizzate e riportate nell'appendice A.

La funzione *helperFindMios* è una delle funzioni *Helper* che fanno parte del progetto ma che sono state descritte in script dedicati. La sua utilità è quella di verificare la presenza di attori che precedono o seguono il veicolo ego e che si trovano nella medesima corsia o in una di quelle adiacenti. Per illustrare i passaggi principali dell'algoritmo della funzione è stato realizzato un diagramma di flusso, riportato in figura 3.9.

Esaminando quest'ultimo è possibile notare che anche in questo caso è stata utilizzata una funzione *Helper*, nello specifico la *helperDetectLaneNumber*; poiché l'identificazione dell'indice della corsia è un'operazione più volte ripetuta all'interno del modello. Si tratta di una funzione che riceve in input il già noto segnale "mapInfo", contenente le informazioni della strada, e la distanza laterale d dalla linea di riferimento. Confrontando la distanza d dalla linea di riferimento con l'ampiezza delle corsie si riesce ad individuare quella occupata dall'attore selezionato.

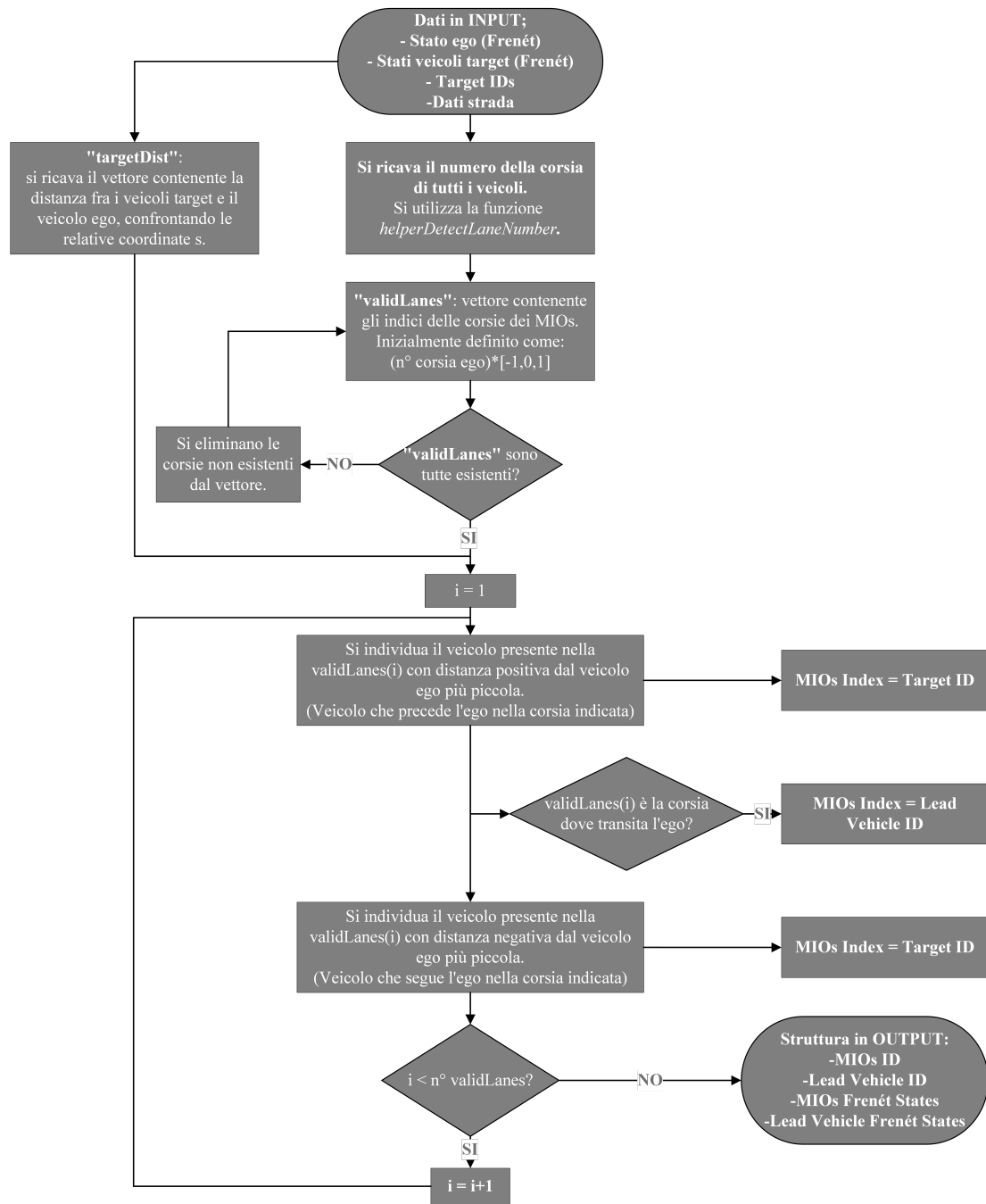


Figura 3.9: Diagramma di flusso Find MIOs

Terminal State Sampler Subsystem

Il blocco denominato *Terminal State Sampler* fa parte del sottosistema *Highway Lane Change Planner* ed è esso stesso un sottosistema, a sua volta composto da diversi blocchi *MATLAB function*. Esso elabora i segnali provenienti dai due blocchi precedentemente esposti, *Frénet State Converter* e *Find MIOs*, utilizzando anche i segnali "mapInfo" e "PlannerParameters", con l'obiettivo di campionare stati terminali differenti a seconda della modalità di funzionamento (CC, LC e LCF). Con stato terminale si indica l'insieme dei parametri che identificano lo stato del veicolo *ego* nel punto finale del segmento di traiettoria in output dal pianificatore. Vengono inizialmente individuati diversi stati terminali, ognuno dei quali verrà successivamente valutato e classificato in base ad un parametro di costo associato. Ogni stato terminale viene associato ad un dato orizzonte temporale, ovvero l'intervallo temporale che intercorre tra l'istante attuale e quello in cui il veicolo *ego* si troverà nel punto in cui termina l'eventuale traiettoria pianificata, e ad una data modalità di funzionamento del pianificatore (CC, LCF e LC). L'insieme di tutte le combinazioni porta alla creazione del set di stati terminali. Da questo set verrà poi selezionato quello corrispondente alla traiettoria effettivamente pianificata e fornita come output del sottosistema *Highway Lane Change Planner*. Si riporta in figura 3.10 lo schema a blocchi del sottosistema.

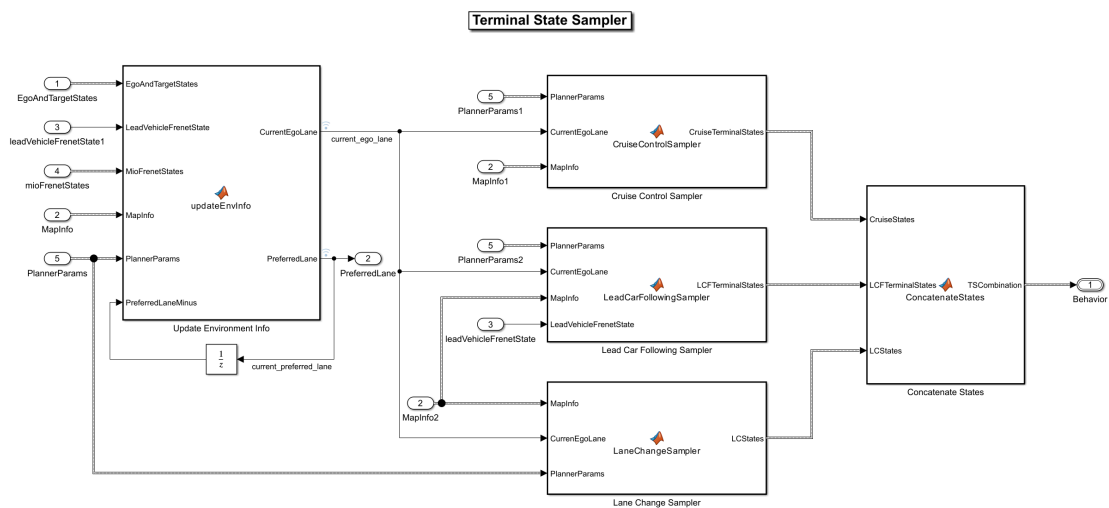


Figura 3.10: Terminal State Sampler Subsystem.

Seguendo il flusso di elaborazione dei segnali in input, il sottosistema si compone dei seguenti blocchi:

- *Update Environment Info* è il blocco dedicato all'individuazione della corsia corrente del veicolo ego e di quella definita come preferenziale.
- *Cruise ControlSampler*, *Lead Car Following Sampler* e *Lane Change Sampler* sono i tre blocchi dedicati al campionamento degli stati terminali, ognuno dedicato ad una singola modalità di funzionamento del pianificatore.
- *Concatenate States* è il blocco destinato alla concatenazione degli stati terminali calcolati.

Dopo una sintetica panoramica di quelli che sono i componenti del sottosistema *Terminal State Sampler*, si passa di seguito ad un'analisi maggiormente dettagliata.

La funzione contenuta nel blocco *updateEnvInfo* è stata creata con il fine di processare i segnali che vengono forniti in input per fornire correttamente i dati ad una seconda funzione, denominata *updateEnvironmentInfo*. Inoltre, questa funzione, la quale si trova a monte dell'intero processo di determinazione dell'indice di corsia preferenziale, si occupa di riportare l'indice di *preferred line* al valore definito nel blocco *Planner configuration parameters*, ovvero il valore corrispondente alla traiettoria del veicolo *ego* descritta nello scenario. Qualora la successiva verifica di tale indice dovesse essere positiva esso viene mantenuto; in tal modo si dà sempre la preferenza alla corsia corrispondente a quella della traiettoria descritta all'interno dello scenario.

La seconda funzione, *updateEnvironmentInfo*, si occupa di ricavare l'attuale corsia del veicolo *ego* richiamando la funzione *helperDetectLaneNumber* e di fornire i segnali necessari alla funzione *helperFindPreferredLane*, la quale contiene il cuore dell'algoritmo di determinazione della corsia preferenziale.

Effettuando analisi sul tempo di collisione (TTC) e sulla distanza tra i veicoli MIO ed il veicolo *ego*, la funzione *helperFindPreferredLane* rimpolpa l'insieme degli script *Helper* annessi al modello, con lo scopo di individuare la corsia definita come preferenziale.

Tutte e tre le funzioni citate sono state analizzate nel dettaglio nell'appendice A. Inoltre, si riporta in figura 3.11 il diagramma di flusso realizzato con l'obiettivo di esporre in maniera grafica e sintetica la logica attraverso la quale viene individuata la corsia preferenziale.

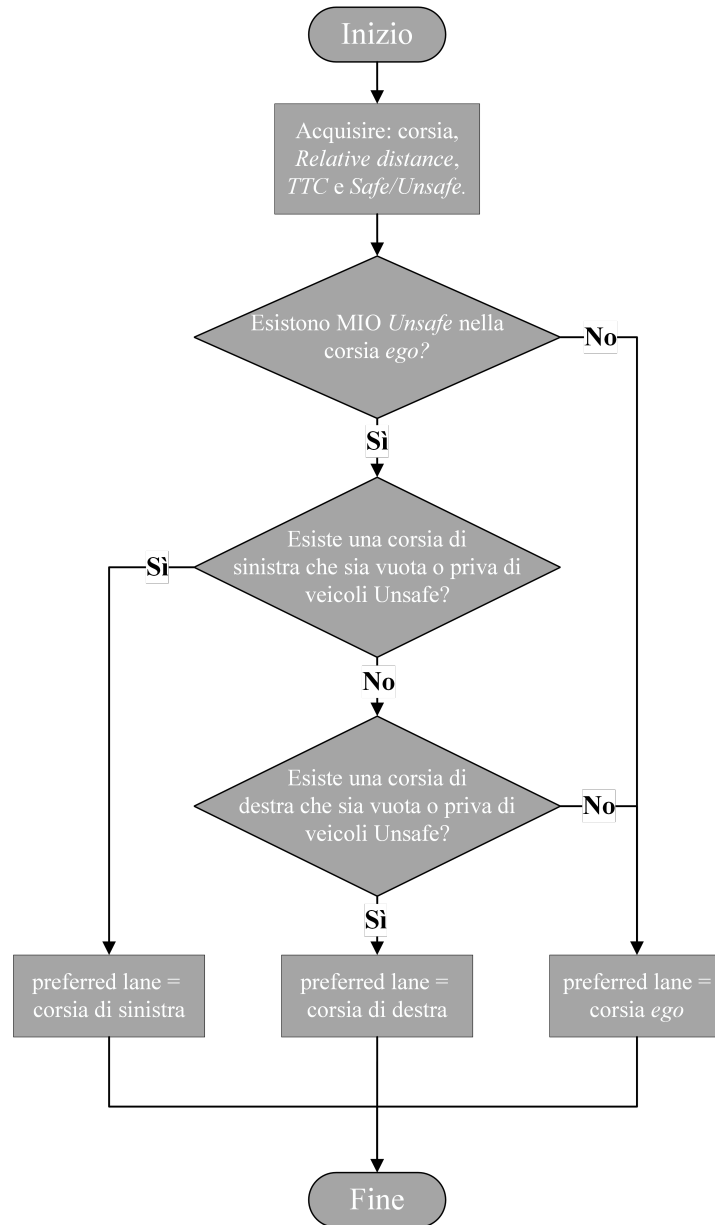


Figura 3.11: Diagramma di flusso calcolo corsia preferenziale

Per comprendere al meglio quanto descritto all'interno del diagramma di flusso, risulta essere di fondamentale importanza approfondire il significato della definizione *Safe/Unsafe* e dei parametri da cui essa dipende. Una corsia viene dichiarata *Unsafe* quando uno dei veicoli MIO che transitano al suo interno viene dichiarato *Unsafe*. Ciò avviene se almeno uno fra i valori di *TTC* e *Relative distance*, calcolati fra il suddetto MIO ed il veicolo *ego*, oltrepassa i valori limite imposti.

- *Relative distance*: Distanza in termini assoluti tra i veicoli *ego* e MIO.
- *TTC*: Tempo necessario alla collisione tra i veicoli *ego* e MIO in assenza di variazioni di velocità e direzione dei singoli veicoli. Esso viene calcolato come il rapporto tra la *Relative distance* e la velocità di avvicinamento dei due veicoli, ovvero la proiezione della velocità relativa dei due veicoli lungo la loro congiungente. Dove per congiungente si identifica il segmento di retta che passa per il centro dell'assale posteriore dei due veicoli.

Queste grandezze risultano essere di primaria importanza per comprendere le logiche di funzionamento del pianificatore; dunque, nel paragrafo 3.4 è stata realizzata un'analisi più approfondita delle logiche di valutazione *Safe/Unsafe*.

I due parametri presentati vengono calcolati utilizzando la funzione *helperCalculateTTC*, la quale è stata analizzata in appendice A. Per stabilire se il veicolo sia *Safe* o *Unsafe* i due segnali vengono processati dalla funzione *checkSafety*. Essa ha il compito di confrontare i parametri di distanza *Relative distance* con i cosiddetti *SafetyGap*, distanze di sicurezza in metri definite nel blocco *Planner Configuration Parameters*. Sempre la funzione *checkSafety* effettua una verifica sul tempo alla collisione (TTC); anche in questo caso tramite confronto con parametri definiti nel blocco citato. Qualora una delle due verifiche condotte rispetto ad un veicolo *MIO* desse esito negativo, la corsia corrente del veicolo MIO viene dichiarata non sicura.

Se la corsia preferenziale in un dato step temporale diventa *Unsafe*, si innesca la procedura di cambio indice "preferredLine"; in caso contrario l'indice attuale viene mantenuto.

Dopo aver terminato il calcolo della corsia definita preferita, quest'ultima viene fornita come input nuovamente alla stessa funzione sotto il nome di "PreferredLaneMinus" nel successivo step temporale, come si può notare dalla figura 3.10.

I tre blocchi dedicati alla modalità di simulazione del veicolo ego, *Cruise Control Sampler*, *Lead Car Following Sampler* e *Lane Change Sampler*, come già detto, sono finalizzati al campionamento degli stati terminali per le relative modalità. Ognuno di essi si occupa di costruire una matrice che per ogni riga ha le coordinate di *Frenét* del dato stato terminale nella data modalità di funzionamento; l'ultimo elemento della riga rappresenta il valore dell'orizzonte temporale associato al singolo stato terminale.

La funzione *CruiseControlSampler*, contenuta all'interno dell'omonimo blocco, permette di ottenere gli stati terminali relativi ad una modalità di CC rielaborando i segnali relativi alle informazioni della strada, all'indice della corsia del veicolo ego e ai parametri di configurazione del pianificatore. La modalità CC prevede che il veicolo *ego* mantenga la corsia attuale e venga portato ad una velocità stabilita dal parametro "setSpeed". Il singolo vettore avrà la seguente composizione:

$$[s, \dot{s}, \ddot{s}, d, d', d'', th] \quad (3.2)$$

Dunque, gli stati terminali differiranno in base al valore del "time horizon" ma avranno tutti $\dot{s} = setSpeed$ e $d = d(currentLane)$.

La funzione è stata analizzata nel dettaglio in appendice A e ne è stata realizzata una schematizzazione grafica in figura 3.12.

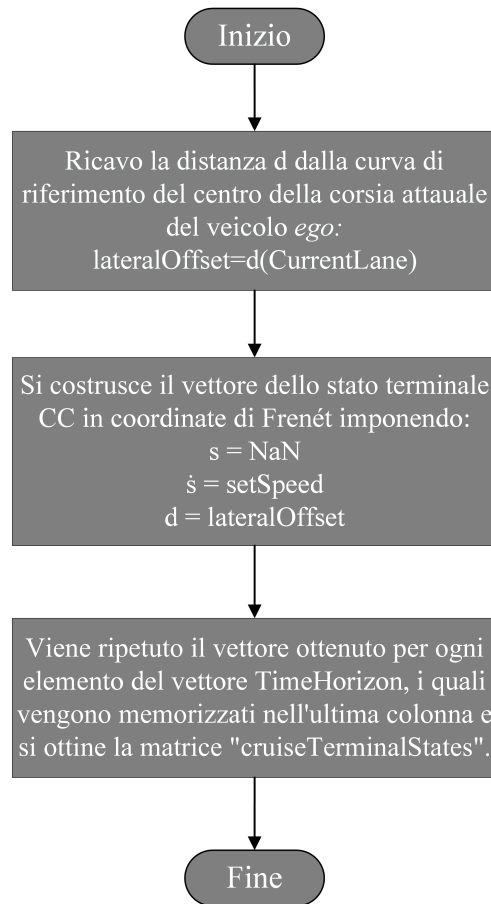


Figura 3.12: Diagramma di flusso *function cruiseControlSampler*

Nel medesimo modo si procede all'analisi del blocco dedicato alla modalità LCF. Questo riceve come input i segnali relativi alle informazioni della strada, ai parametri di configurazione del pianificatore, all'indice della corsia del veicolo *ego* e allo stato di Frénet del veicolo che precede il veicolo *ego*, nonché il veicolo da seguire nella modalità LCF. Questi dati sono utili in quanto la modalità LCF prevede che il veicolo *ego* mantenga la corsia attualmente occupata e adatti la propria velocità a quella del veicolo che lo precede, denominato veicolo *lead*. Dunque, gli stati terminali differiranno in base al valore del time horizon ma avranno tutti $\dot{s} = \dot{s}_{Lead}$ e $d = d(currentLane)$.

La funzione è stata analizzata nel dettaglio in appendice A e ne è stata realizzata una schematizzazione grafica in figura 3.13.

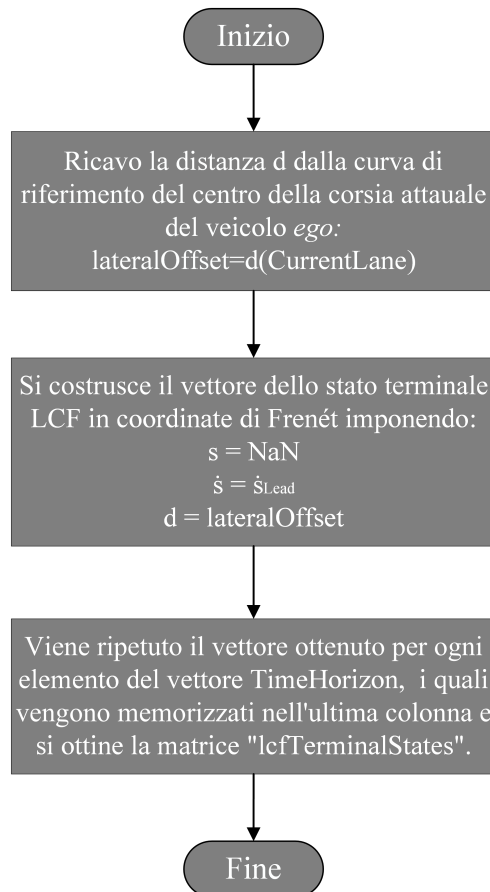


Figura 3.13: Diagramma di flusso *function leadCarFollowingSampler*

Infine, si procede con l'analisi del blocco dedicato alla modalità di LC. Gli input della funzione sono i medesimi e anch'essa è stata definita in *HelperTerminalStateSampler*. In quest'ultimo caso il numero di stati terminali potrebbe aumentare, in quanto per ogni orizzonte temporale si hanno tanti stati terminali quante sono le corsie prese in esame "adjacentLanes".

La modalità LC prevede che il veicolo *ego* mantenga la propria velocità di crociera, definita dal parametro "setSpeed", e che si sposti in una delle corsie adiacenti a quella occupata. Dunque, gli stati terminali differiranno in base al valore del "time horizon" e della distanza laterale, tale che $d = d(\text{adjacentLanes})$, ma avranno tutti $\dot{s} = \text{setSpeed}$.

La funzione è stata analizzata nel dettaglio in appendice A e ne è stata realizzata una schematizzazione grafica in figura 3.14.

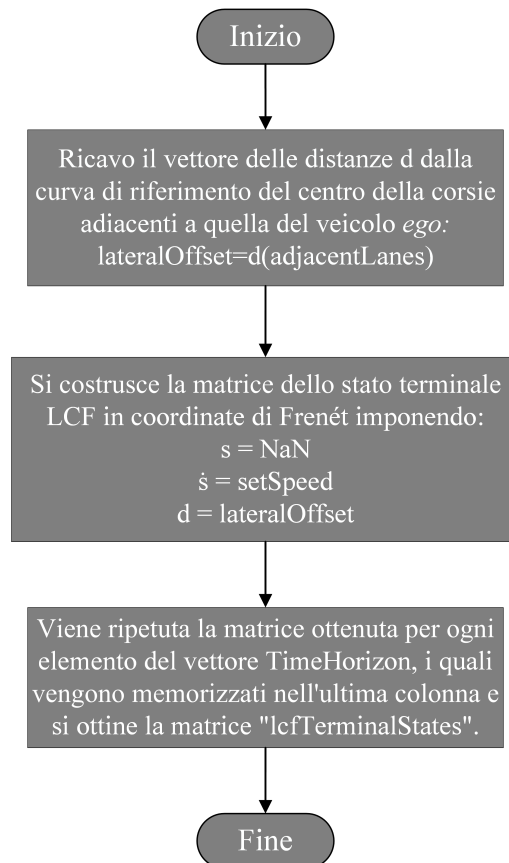


Figura 3.14: Diagramma di flusso *function laneChangeSampler*

Dopo aver campionato gli stati terminali per ognuna delle tre modalità, i segnali in output a ciascun blocco vengono processati dal blocco *Concatenation States*. Esso si occupa della loro concatenazione per produrre il bus contenente la combinazione degli stati terminali. Inoltre, viene realizzato un vettore che per ognuno degli stati appartenenti al set identifica la modalità di funzionamento (CC, LCF e LC).

Questo blocco assume una notevole rilevanza per le future implementazioni del modello, in quanto permette di manipolare tutti gli stati del set facilmente, avendo accesso ad ogni loro informazione. Per un'analisi maggiormente dettagliata della funzione *ConcatenateStates* si rimanda all'appendice A.

Motion Prediction: Il blocco *Motion Prediction* è dedicato al calcolo degli stati futuri dei veicoli identificati come MIO, il quale viene effettuato supponendo costanti le due componenti della velocità e partendo dai relativi stati attuali. Fanno parte degli input del blocco esaminato i parametri di configurazione del pianificatore definiti nel blocco *Planning Configurator Parameters*, di cui al paragrafo 3.5, le informazioni relative alla strada e gli stati in coordinate di Frénet dei veicoli MIO. La funzione utilizzata per processare i dati riportati è denominata *stepImpl* e fa parte della *function HelperMotionPrediction*.

Si specifica che le manipolazioni dei dati vengono effettuate mantenendo le coordinate di Frénet; tuttavia, l'output del blocco restituisce le traiettorie future di tutti i veicolo MIO in coordinate globali. Questo è reso necessario in quanto le coordinate di Frénet vengono utilizzate esclusivamente per generare le traiettorie, in quanto la manipolazione dei dati risulta essere notevolmente facilitata; tuttavia, le successive valutazioni vengono tutte effettuate mediante coordinate globali.

In figura 3.15 è stato riportato un diagramma di flusso realizzato con l'obiettivo di sintetizzare l'algoritmo che esegue la funzione citata; qualora si volesse procedere con un'analisi più approfondita si rimanda all'appendice A.

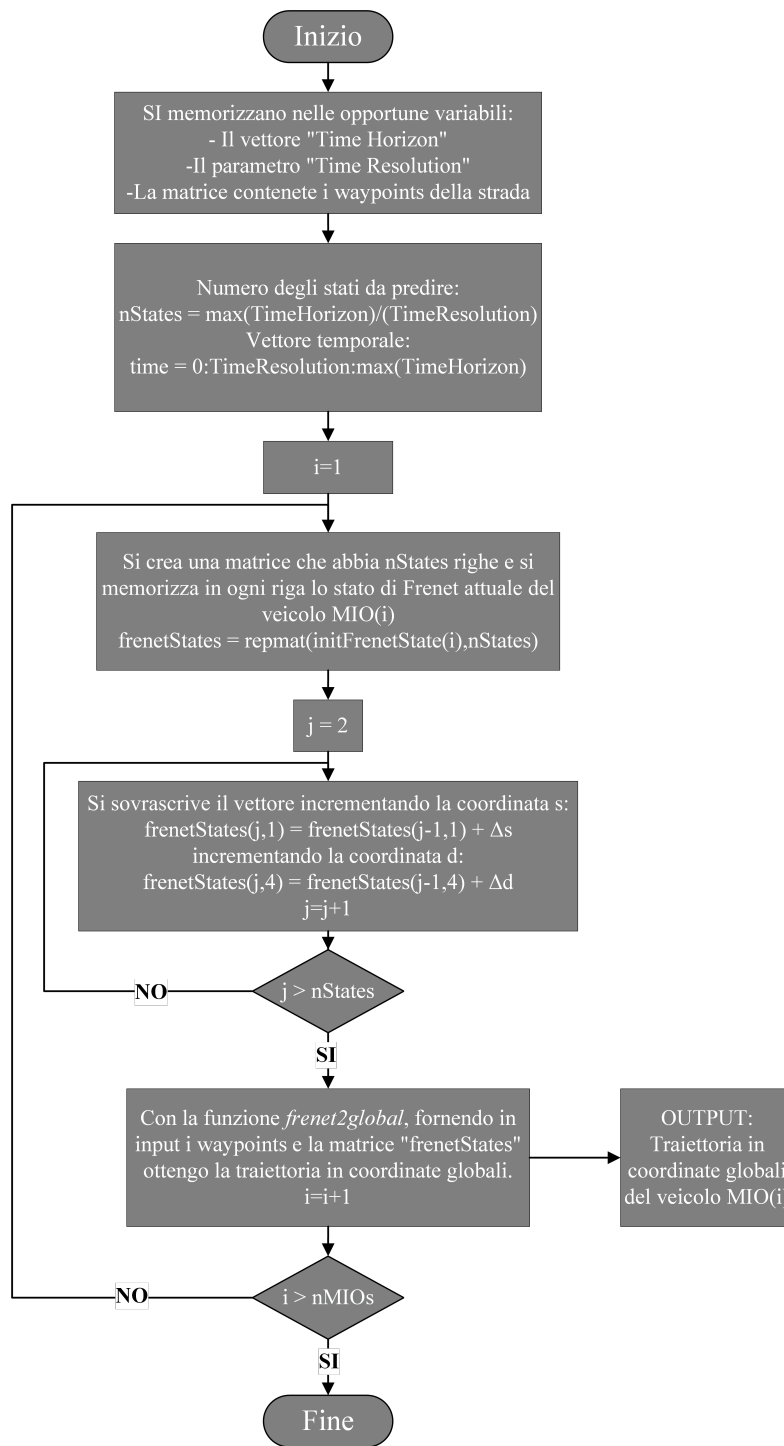


Figura 3.15: Diagramma di flusso calcolo traiettorie future veicoli MIO

Motion Planner Subsystem

Il sottosistema *Motion Planner* è la parte centrale del pianificatore, in quanto dedicata alla generazione della traiettoria del veicolo *ego*. Per generare una traiettoria è necessario partire dagli stati terminali campionati e dai parametri di configurazione del pianificatore. La pianificazione avviene con una determinata frequenza e, dovendo seguire una singola traiettoria, è necessario implementare delle funzioni di costo con lo scopo di scartare le opzioni peggiori. Dopo aver generato le traiettorie alternative è necessario verificarne la fattibilità e la presenza di eventuali collisioni; a valle di ciò si otterrà la traiettoria da seguire. In figura 3.16 si può apprezzare lo schema a blocchi del sottosistema.

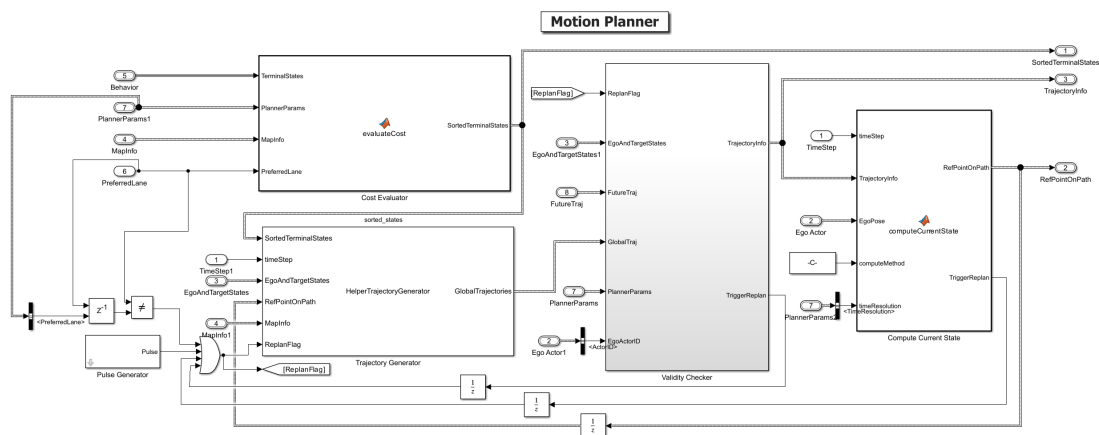


Figura 3.16: Motion Planner Subsystem

Anche in questo caso il sottosistema è caratterizzato dalla presenza di un diverso numero di blocchi, ognuno di essi ha una specifica funzione in relazione alle operazioni svolte dal sottosistema che sono state descritte in precedenza. Analizzando gli input del sistema, apprezzabili in figura 3.6, si nota che il primo blocco a processare i segnali è quello destinato alla valutazione delle funzioni di costo. Si tratta di un blocco MatLab *function* che riceve in input gli stati terminali concatenati relativi alle tre modalità di guida, le informazioni sulla strada, l'indice della corsia considerata come preferita e i parametri di configurazione del pianificatore. La funzione al proprio interno è la *helperEvaluateCost* e si occupa di riordinare gli stati terminali seguendo una logica preferenziale.

Ogni stato presente nel set viene associato ad un parametro di costo. La funzione adottata per calcolare tali grandezze è la seguente:

$$costTS = latCost + timeCost + speedCost \quad (3.3)$$

Ciascun addendo viene calcolato singolarmente mediante le seguenti formule:

$$\begin{aligned} latCost &= latDeviation * latDevWeight \\ timeCost &= timeHorizon * timeWeight \\ speedCost &= |\dot{s} - setSpeed| * speedWeight \end{aligned} \quad (3.4)$$

- *latCost*: Quantifica la distanza dalla corsia indicata come preferenziale della corsia che il veicolo *ego* andrebbe ad occupare nel punto terminale del segmento di traiettoria da pianificare .
- *timeCost*: Tale parametro risulta essere tanto maggiore quanto minore è l'orizzonte temporale del dato stato terminale. Questo è possibile in quanto il valore di "timeWeight" è negativo. In tal modo si preferisce pianificare dei segmenti di traiettoria più lunghi piuttosto che tanti brevi segmenti.
- *speedCost*: Quantifica la differenza in valore assoluto tra la velocità del veicolo *ego* nel punto terminale del segmento di traiettoria e la velocità di crociera "setSpeed" del veicolo *ego*, definita nel blocco *Planner Configuration Parameters*.

Come si può notare dalle equazioni 3.4, ogni contributo della funzione di costo ha nella propria formulazione un parametro *costWeight*. Essi vengono definiti da un apposito script MatLab e permettono di variare il peso relativo degli addendi del parametro "costTS".

Replan flag

Per attivare il generatore di traiettorie è stata inserita una variabile booleana chiamata "ReplanFlag", che quando assume il valore logico "true" (1) esprime la necessità di analizzare il set di stati terminali per pianificare un nuovo segmento di traiettoria.

Il segnale "ReplanFlag" è l'output di un blocco *OR* con quattro segnali in input; trattandosi di un blocco che ha la funzione di operatore logico, i segnali in input sono anch'essi delle variabili booleane. Come si può notare in figura 3.16 gli input del blocco citato sono:

- Segnale output del blocco *NotEqual*; il quale confronta il parametro "Preferred Lane" proveniente dal segnale in output al sottosistema *Terminal State Sampler* con il medesimo segnale processato da un blocco *Delay*. Conseguentemente, non appena si ha una variazione del valore "Preferred Lane" viene richiesta una nuova pianificazione della traiettoria.
- Segnale output del blocco *Pulse Generator*; il quale genera una variabile booleana positiva con una frequenza fissa. Il valore di *replane rate* impostato è pari a $1s^{-1}$. Tale grandezza è regolata da un parametro definito nel blocco *Planner Parameters*.
- Segnale output del sottosistema *Validity Checker*.
- Segnale output del sottosistema *Compute Current State*.

Il segnale "Replan flag" e la struttura "SortedTerminalStates" fanno parte del gruppo di input del sistema *Trejectory Generator*.

Trajectory Generator System

Il sottosistema *Trajectory Generator* è dedicato alla creazione di segmenti di traiettoria di riferimento per il veicolo *ego*. Essi vengono generati a partire dal set di stati terminali ordinati in base alla funzione di costo, mediante la funzione *trajectoryGeneratorFrenet* fornita dall'*Automated Driving Toolbox* di MatLab. L'utilizzo della funzione citata e le logiche che ne governano il funzionamento sono state approfondite all'interno del capitolo secondo.

La funzione che esegue la parte centrale dell'algoritmo è stata denominata *stepImpl* e fa parte della *function HelperTrajectoryGenerator*. Essa viene attivata quando il segnale denominato "replanFlag" assume il valore logico "true".

Attraverso una retroazione del segnale denominato "RefPointOnPath", opportunamente processato da un blocco *Delay*, si ottiene lo stato attuale di riferimento del veicolo *ego*. Esso viene utilizzato come primo punto del nuovo segmento di traiettoria. Dunque, i nuovi segmenti di traiettoria da seguire vengono pianificati in modo che la traiettoria di riferimento sia un'unica curva continua.

Per generare le n traiettorie che congiungono il punto di riferimento agli n stati terminali viene utilizzato il "connect method". A partire dai waypoints della traiettoria di riferimento, la quale anche in questo caso è stata associata alla linea di mezzzeria della carreggiata, si ottiene un oggetto denominato "connector". Questo viene poi processato dalla funzione *connect* che ne estrapola le informazioni circa la curva di riferimento, per generare le n traiettorie in coordinate globali.

Ultimata la creazione delle traiettorie associate agli stati terminali, il flusso di informazioni procede attraverso il sottosistema *Validity Checker*, dedicato all'esclusione delle traiettorie non percorribili.

Per una maggiore comprensione della funzione *HelperTrajectoryGenerator* è stato creato un diagramma di flusso, opportunamente riportato in figura 3.17. Per un'analisi maggiormente dettagliata si rimanda all'appendice A.

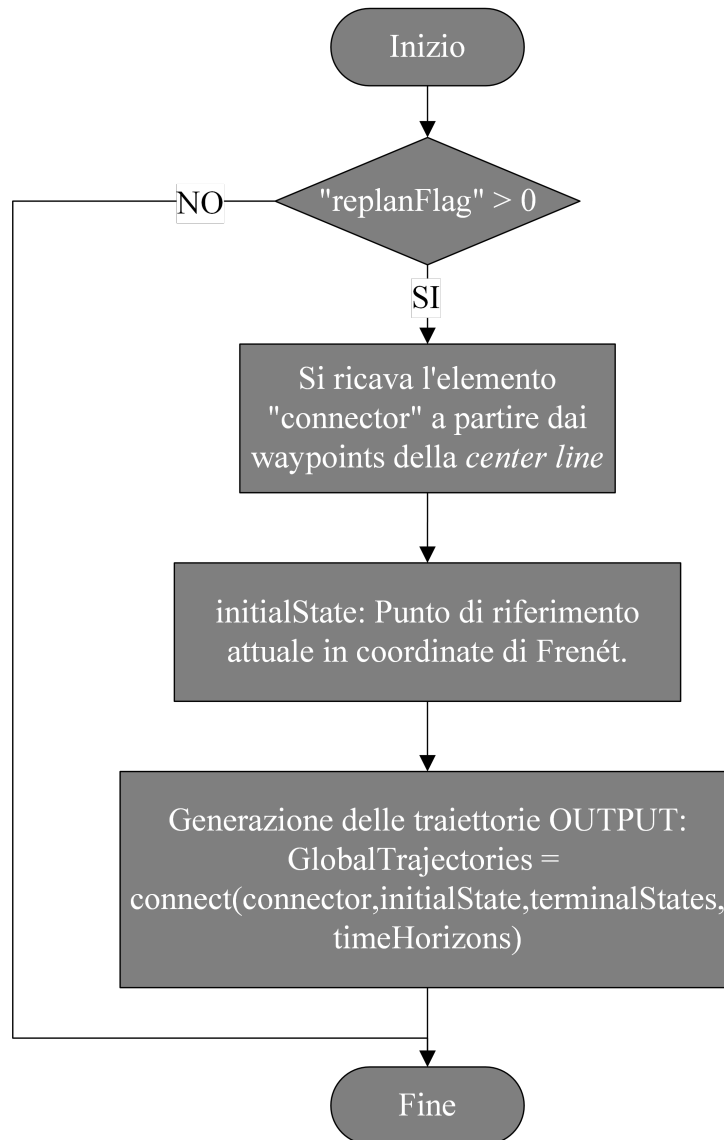


Figura 3.17: Diagramma di flusso generazione traiettorie associate agli stati terminali

Validity Checker Subsystem

Il sottosistema *Validity Checker* è dedicato alla verifica di percorribilità delle traiettorie generate; questa viene effettuata su due fronti, uno puramente cinematico e l'altro legato ad eventuali collisioni con i veicoli target.

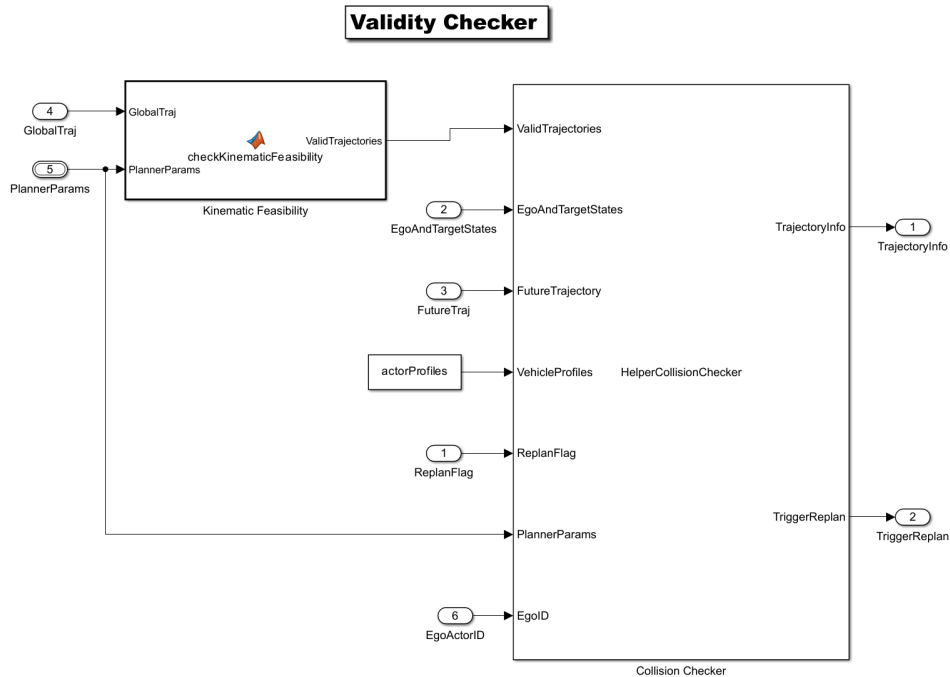


Figura 3.18: Validity Checker subsystem.

Come si nota in figura 3.18, il sottosistema si compone di due differenti blocchi, ognuno dei quali dedicato ad una specifica verifica. Il blocco *kinematic Feasibility* è del tipo MatLab *function* ed ospita la *function helperKinematicFeasibility*. Essa ha il compito di stabilire se una traiettoria è valida o meno, in base alla relativa la fattibilità cinematica.

Al suo interno vengono analizzati i valori di picco delle principali grandezze associate alle traiettorie alternative, quali accelerazione, velocità, velocità di imbardata e curvatura. Se tutte le grandezze citate rispettano i parametri limite definiti nel blocco *Planner parameters configuration* la traiettoria viene dichiarata "valida".

Nella tabella 3.2 sono state riportate tutte le condizioni logiche da rispettare, per un'analisi maggiormente dettagliata dell'algoritmo della funzione si rimanda all'appendice A.

Logical condition	Description
A	$\max(\textit{Longitudinal acceleration}) < \textit{Max Acceleration}$
B	$\max(\textit{Curvature}) < \textit{Max Curvature}$
C	$\max(\textit{Yaw rate}) < \textit{Max Yaw rate}$
D	$\min(\textit{Velocity}) > \textit{Min Velocity}$
E	$\max(\textit{Lateral acceleration}) < \textit{Max Acceleration}$

Tabella 3.2: Condizioni logiche per definizione *IsValid*.

$$\textit{IsValid} = A \wedge B \wedge C \wedge D \wedge E \quad (3.5)$$

Si specifica che la condizione *E* è stata aggiunta in seguito, in quanto ritenuta di primaria importanza nel corso della valutazione di una traiettoria.

La verifica sulle eventuali collisioni viene effettuata all'interno del blocco *Collision Checker*; esso contiene un algoritmo in grado di confrontare le traiettorie generate con quelle dei veicoli definiti come MIO.

L'algoritmo permette di identificare gli ostacoli presenti nello scenario (veicoli MIO) e tiene conto delle relative geometrie al fine di poter valutare il rischio di collisione con la traiettoria del veicolo *ego*. Tale scopo viene raggiunto grazie all'ausilio della *Dynamic capsule-based obstacle list*, le cui logiche di funzionamento sono state descritte dettagliatamente nel capitolo secondo, paragrafo 2.2. Si tratta di una modellizzazione dei veicoli in capsule bidimensionali. Ognuno dei corpi viene descritto da tre elementi chiave, quali: l'indice identificativo, la matrice contenente l'evoluzione degli stati nel tempo e la struttura contenente la geometria del corpo. A sua volta la geometria viene descritta da tre parametri: la lunghezza della capsula, il raggio e la trasformata utile a modificare il posizionamento locale della capsula.

Ciascuna fra le traiettorie alternative viene analizzata mediante un confronto con le previsioni delle traiettorie dei veicoli MIO, le quali, si ricorda, vengono individuate dal blocco *Motion Prediction*. Utilizzando la lista delle capsule è poi possibile confrontare istante per istante l'estensione dei corpi alla ricerca di sovrapposizioni, le quali corrispondono alle eventuali collisioni fra i veicoli. Qualora una traiettoria dovesse portare ad una collisione, la funzione *StepImpl* contenuta all'interno della *function HelperCollisionChecker* associa ad una variabile booleana denominata "isColliding" il valore logico "true".

Per una maggiore comprensione della funzione *HelperCollisionCheck* è stato realizzato un digramma di flusso che ne schematizza l'algoritmo, riportato in figura 3.19. Per ulteriori approfondimenti si rimanda all'appendice A.

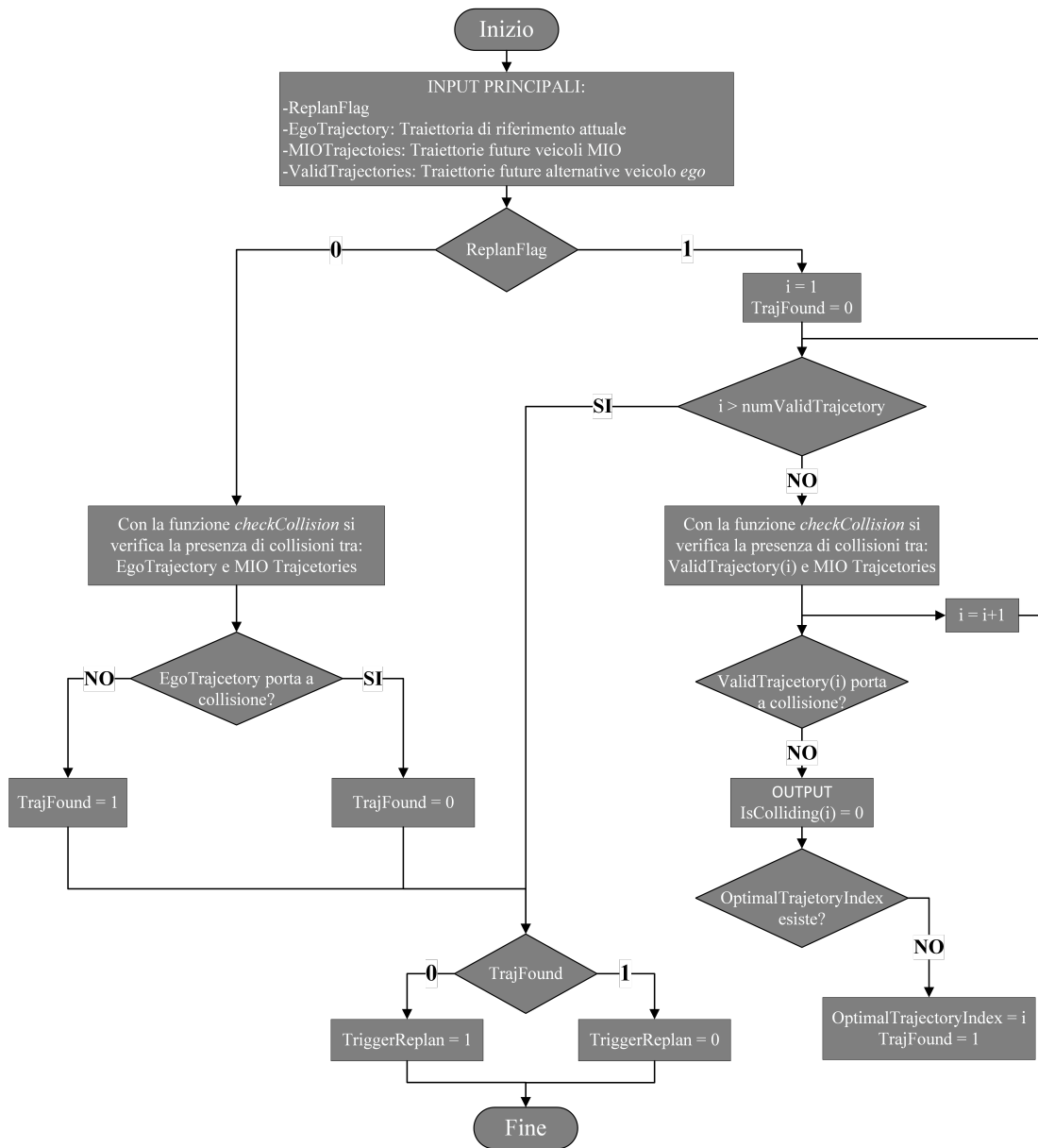


Figura 3.19: Diagramma di flusso verifica collisioni sulle traiettorie alternative

Modifiche introdotte Esaminando gli algoritmi delle due funzioni esposte ci si accorge che una traiettoria può essere identificata come riferimento pur non superando la verifica sulla fattibilità cinematica. In tal modo il parametro "IsValid" avrebbe un'utilità di tipo esclusivamente informativo e non influenzerebbe alcun processo decisionale sulla scelta della traiettoria da seguire.

Dunque, la non esclusione delle traiettorie non valide cinematicamente dal processo di selezione della traiettoria di riferimento potrebbe portare ad un output del pianificatore errato. Nell'eventualità in cui ciò accedesse il controllore tenterebbe comunque di fornire i corretti comandi al veicolo; tuttavia, quest'ultimo potrebbe non riuscire a mantenere la stabilità nei casi più estremi.

Tale considerazione ha portato ad una variazione dell'algoritmo, tale per cui tutte le traiettorie dichiarate non valide non possono essere identificate come ottimali. Dunque, la traiettoria di riferimento è la traiettoria con il minore parametro di costo associato fra quelle che non portano ad una collisione e che superano la verifica di fattibilità cinematica. In figura 3.20 è stata riportata una rappresentazione grafica del processo di determinazione della traiettoria di riferimento, dove ciascuna traiettoria alternativa è stata rappresentata da un rettangolo, la cui altezza simboleggia il parametro di costo associato.

Deve essere compreso che non si può parlare di ricerca dell'ottimo, in quanto la traiettoria frutto di tale processo di selezione viene scelta da un set fortemente limitato di alternative. Il numero N di alternative dipende da:

- n numero di differenti modalità: CC, LCF, LC a sinistra e LC a destra.
- m numero di differenti valori di orizzonte temporale inseriti nel vettore *TimeHorizons*.

Tale per cui:

$$N = n * m \tag{3.6}$$

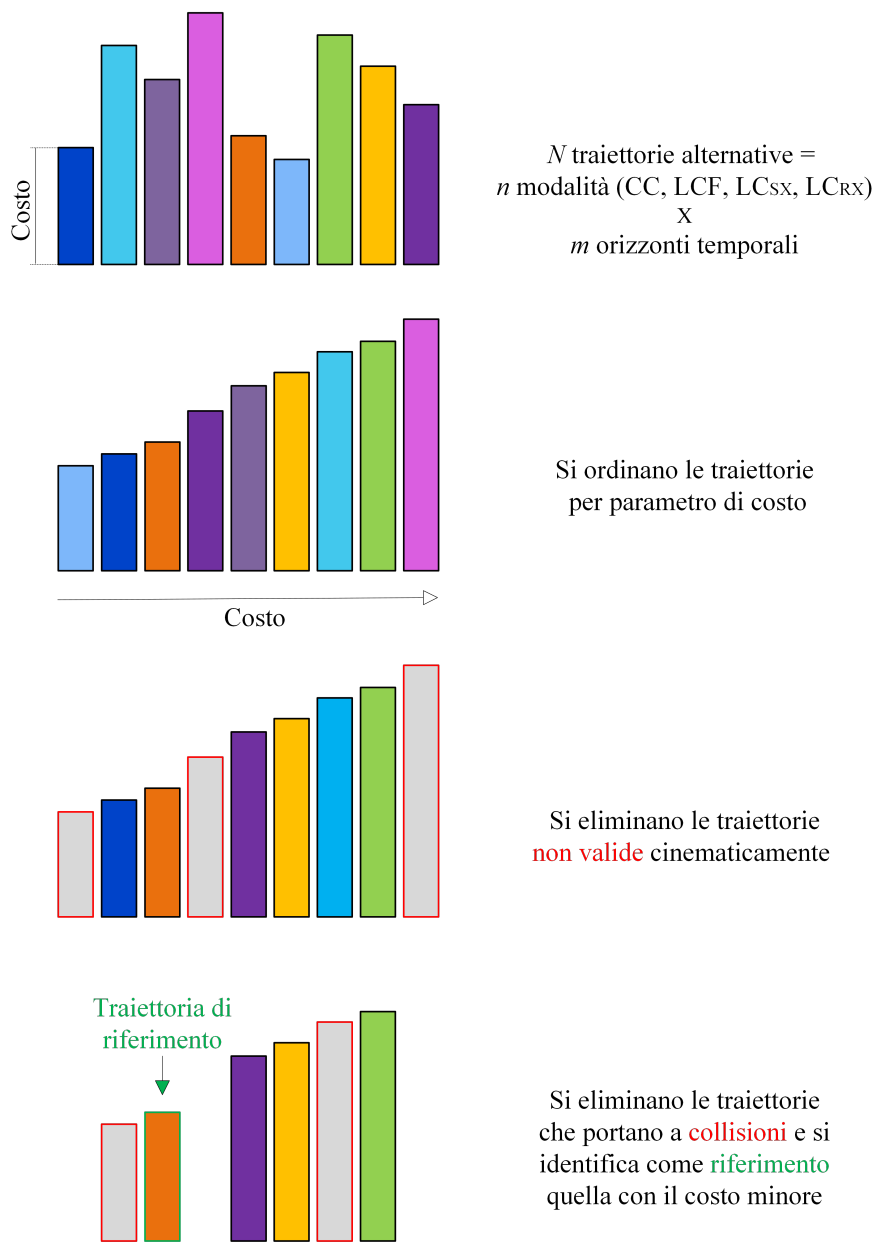


Figura 3.20: Rappresentazione processo di determinazione traiettoria di riferimento.

Compute Current State

L'ultimo blocco da descrivere per completare l'analisi del sottosistema *Motion Planner* è denominato *Compute Current State*. Si tratta di un blocco MatLab *function* dedicato alla stima dello stato successivo del veicolo; ovvero a partire dalla traiettoria di riferimento, output del blocco *Validity checker*, individua le coordinate del punto di riferimento per lo step temporale successivo.

La funzione che assolve il compito descritto differisce in base al modello in cui il pianificatore di traiettorie viene inserito; ciò avviene poiché sono necessari output differenti. Possiamo distinguere due differenti casi:

- *Caso A*: Viene utilizzato il modello con il solo pianificatore di traiettorie. In tal caso si assume che la traiettoria di riferimento, output del pianificatore, sia esattamente la traiettoria del veicolo *ego*, la funzione che viene utilizzata è la *helperComputeCurrentStateBasedOnTime*.
- *Caso B*: Viene utilizzato il modello che integra il pianificatore con un controllore ed un modello di veicolo. La traiettoria di riferimento, output del pianificatore, viene utilizzata per calcolare gli errori da un confronto con la posizione effettiva del veicolo *ego*, la funzione che viene utilizzata è la *helperComputeCurrentStateBasedOnPose*.

Come si può vedere in figura 3.16 gli input del blocco sono:

- *Trajectory Info*: Struttura output del sottosistema *validity checker* contenente la traiettoria di riferimento.
- *timeStep*: Parametro contenente l'unità di campionamento temporale.
- *EgoPose*: Struttura contenente lo stato attuale del veicolo *ego* in coordinate globali.
- *computeMethod*: Flag che indica quale delle due funzioni citate deve essere utilizzata.

Caso A - Computazione temporale

Assumendo che la traiettoria effettiva del veicolo *ego* sia puntualmente coincidente con la traiettoria di riferimento, per individuare il punto del successivo step temporale e le grandezze descrittive del moto del veicolo in tale punto, è necessario basare il calcolo sul tempo. Risultano essere di fondamentale importanza tre variabili:

- *timeResolution*: Parametro contenente l'unità di campionamento temporale, ovvero il tempo che intercorre tra lo stato attuale ed il successivo stato temporale.
- *replanTime*: Istante temporale in cui viene pianificato il nuovo segmento di traiettoria.
- *timeDelta*: Tempo che intercorre tra lo step in cui viene pianificato il nuovo segmento di traiettoria ed il successivo step temporale. Esso viene calcolato con la seguente formula.

$$timeDelta = time - replanTime + timeResolution \quad (3.7)$$

Per ottenere le grandezze associate al successivo istante temporale del veicolo *ego* è necessario interpolare linearmente le grandezze della traiettoria di riferimento su base temporale. Il valore di tempo utilizzato per l'interpolazione è il parametro "timeDelta". Infine, tutte le informazioni circa gli stati del veicolo nel punto calcolato vengono memorizzate nella struttura "RefPointOnPath". Si rimanda all'appendice A per un'analisi più approfondita.

Caso B - Computazione spaziale

Utilizzando un modello che integri un controllore ed un modello di veicolo dedicato, l'output del blocco *validityChecker*, contenente la traiettoria ideale, deve essere confrontato con lo stato attuale del veicolo per quantificare gli errori di orientazione, posizionamento e velocità. Essi servono al controllore per correggere la traiettoria effettiva del veicolo *ego*, fornendo al modello di veicolo i corretti comandi.

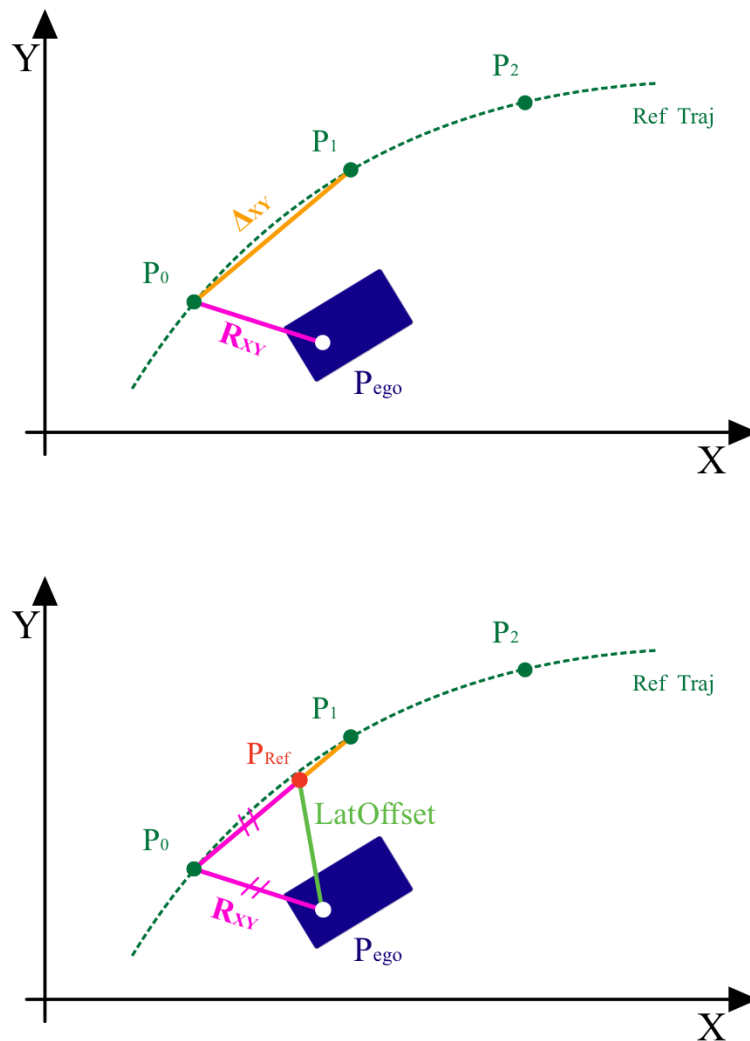


Figura 3.21: Rappresentazione grafica individuazione punto di riferimento.

A differenza del caso precedente, il punto di riferimento viene calcolato come interpolazione effettuata geometricamente e non mediante il vettore temporale. In figura 3.21 è stata riportata una rappresentazione grafica delle principali distanze coinvolte. Ad ogni step temporale, è necessario individuare il punto della traiettoria di riferimento più vicino, nell'esempio riportato esso è stato denominato P_0 . Dopodiché si procede calcolando la quantità "u", tale per cui:

$$u = \left| \frac{R_{XY}}{\Delta_{XY}} \right| \quad (3.8)$$

Dove il numeratore R_{XY} rappresenta la distanza tra il punto P_0 e la posizione del veicolo *ego* P_{ego} ; invece, il denominatore Δ_{XY} rappresenta la distanza tra il punto P_0 ed il punto successivo della traiettoria di riferimento P_1 . Attraverso la quantità u è possibile definire due "pesi":

$$\begin{aligned} W1 &= 1 - u \\ W2 &= u \end{aligned} \quad (3.9)$$

Attraverso queste quantità viene calcolato il punto di riferimento P_{ref} corrente, rappresentato in rosso. Esso si trova lungo il segmento $\overline{P_0P_1}$ ad una distanza da P_0 pari a R_{XY} .

$$\begin{aligned} X_{ref} &= W1 * X_0 + W2 * X_1 \\ Y_{ref} &= W1 * Y_0 + W2 * Y_1 \\ \theta_{ref} &= W1 * \theta_0 + W2 * \theta_1 \end{aligned} \quad (3.10)$$

Dopo il calcolo del punto di riferimento corrente è necessario procedere con il calcolo delle grandezze di riferimento da fornire in output, quali: velocità, angolo di imbardata e così via. Come nel caso precedente, l'output del blocco contiene le grandezze associate al punto di riferimento da inseguire e non a quello corrente; dunque, le grandezze di riferimento del moto del veicolo sono calcolate rispetto al punto di riferimento successivo a quello corrente. La metodologia con la quale vengono calcolati tali dati è la medesima; tuttavia, i valori utilizzati per l'interpolazione sono corrispondenti ai punti P_1 e P_2 . Si riporta la formula con la

quale viene calcolato il valore di una grandezza generica "value".

$$value_{ref} = W1 * value_1 + W2 * value_2 \quad (3.11)$$

Tuttavia, il calcolo del punto di riferimento corrente P_{ref} risulta necessario per il calcolo degli errori di posizione e di orientamento da fornire al controllore. Nel modello utilizzato l'errore di velocità viene calcolato direttamente all'interno del blocco dedicato al controllore.

$$\begin{aligned} LatOffset &= \sqrt{(X_{ego} - X_{ref})^2 + (Y_{ego} - Y_{ref})^2} \\ RefYawDiff &= Yaw_{ref} - Yaw_{ego} \end{aligned} \quad (3.12)$$

Infine, tutti i dati calcolati vengono memorizzate nella struttura "RefPointOnPath". Si rimanda all'appendice A per un'analisi più approfondita della funzione.

Distanza tra i punti di una traiettoria: La distanza tra i punti della traiettoria di riferimento dipende da diversi fattori e per comprenderne l'influenza è necessario precisare che, per costruzione del modello e della funzione utilizzata, i punti vengono definiti su base temporale. Per percorrere la distanza tra un punto ed il suo successivo, il veicolo *ego* impiega una quantità di tempo pari all'unità di campionamento temporale *TimeStep*. Dunque, il numero di punti di un dato segmento di traiettoria è pari a:

$$n_{pt} = \frac{TimeHorizon}{TimeStep} + 1 \quad (3.13)$$

Di conseguenza è possibile affermare che:

- A parità di altri fattori, diminuendo la velocità si ottengono delle distanze tra i punti minori. In quanto per un medesimo orizzonte temporale la distanza percorsa è minore ma il numero di punti rimane costante. Quanto descritto è apprezzabile confrontando il primo ed il secondo grafico riportati in figura 3.22.
- A parità di altri fattori, diminuendo il *TimeStep* la distanza tra i punti si riduce poiché aumenta il numero di punti. Quanto descritto è apprezzabile confrontando il secondo ed il terzo grafico riportati in figura 3.22.
- A parità di altri fattori, diminuendo il *Time Horizon* la distanza tra i punti aumenta quando è presente una velocità laterale \dot{d} non nulla. Prendendo in considerazione un cambio di corsia come quello riportato nel primo grafico in figura 3.22, il veicolo che percorre la traiettoria in blu deve percorrere la medesima distanza laterale d in tempi minori rispetto al veicolo che percorre la traiettoria in arancio, in quanto ha un *Time Horizon* minore. Ciò porterà a intervalli Δd maggiori, a causa di un numero di punti inferiore.

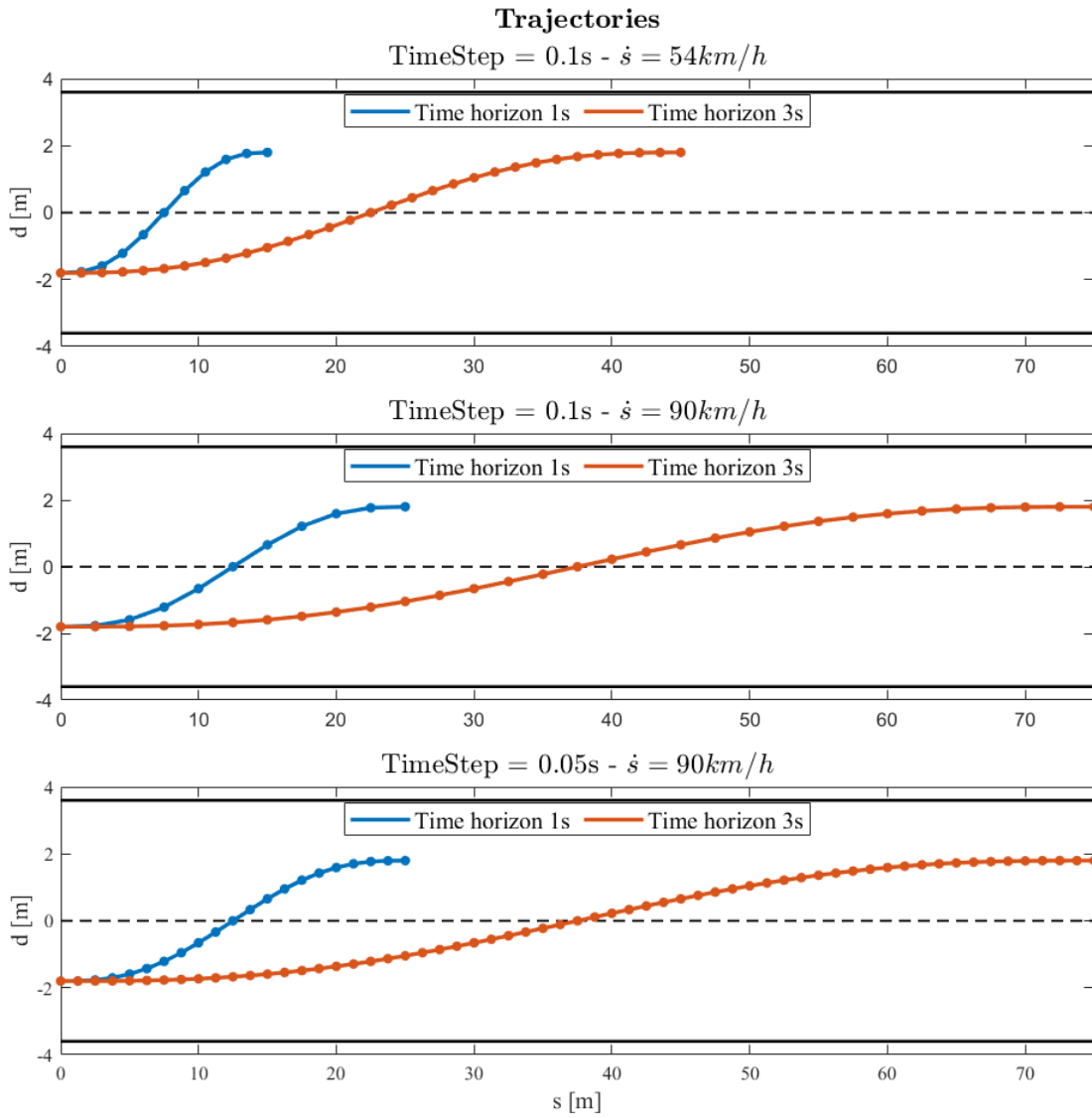


Figura 3.22: Distanze tra punti della traiettoria di riferimento.

3.1.5 Pack Actor

Il sottosistema *Pack Actor* rappresenta l'area del modello dedicata alla costruzione del Bus relativo allo stato attuale del veicolo *ego*. Se ne può apprezzare la schematizzazione a blocchi in figura 3.23.

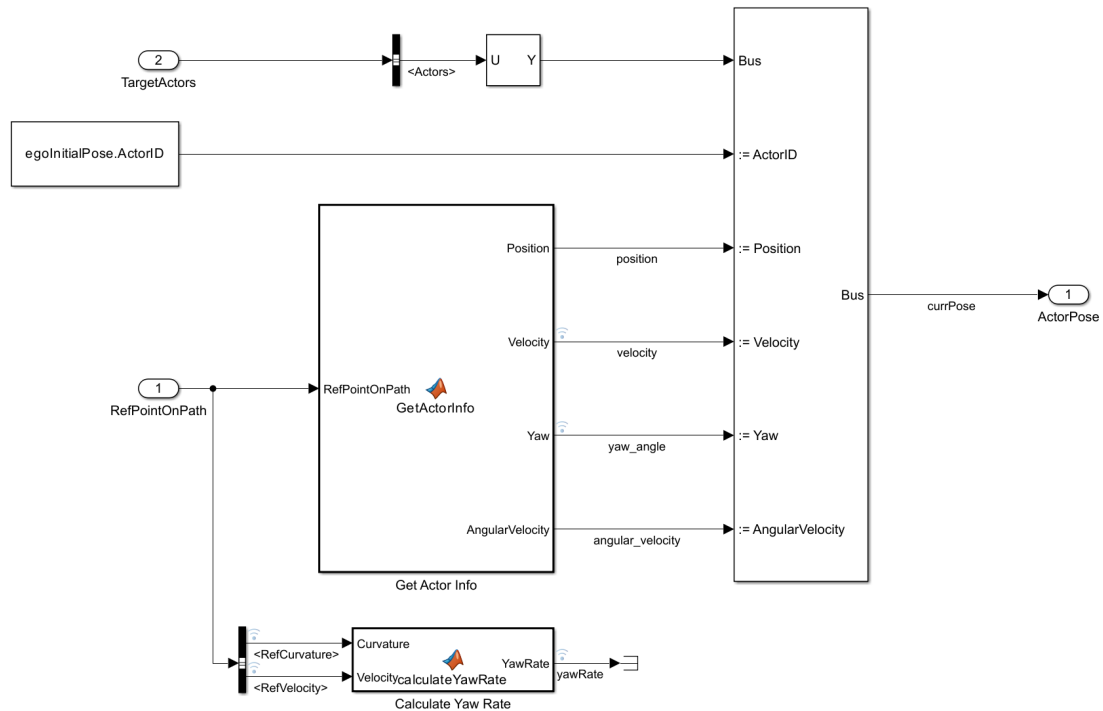


Figura 3.23: Pack Actor subsystem.

L'output del sottosistema è rappresentato dal segnale Bus "ActorPose", il quale al proprio interno ospita i segnali relativi a:

- Indice di identificazione del veicolo *ego* "ActorID";
- Vettore posizione, vettore velocità, angolo di imbardata e velocità di imbardata. Tali parametri sono estratti dalla struttura "RefPointOnPath" mediante uno script contenuto all'interno del blocco *GetActorInfo*.

Il blocco *PackActor* è presente esclusivamente nel modello con il solo pianificatore; in quanto, qualora presente, i dati contenenti lo stato attuale del veicolo *ego* vengono processati ed estratti dal modello di veicolo.

3.1.6 Metrics Assessment

Il sottosistema *Metrics Assessment* non fornisce direttamente un contributo attivo al sistema assistenziale, in quanto non altera i segnali presenti nel modello. Ad ogni step temporale esso effettua una verifica dello scenario e delle grandezze che caratterizzano il moto del veicolo; qualora non venissero rispettate le condizioni limite imposte vengono prodotti dei segnali di allarme e viene interrotta la simulazione in caso di collisione fra veicoli. La struttura generale del sottosistema è rappresentata in figura 3.24.

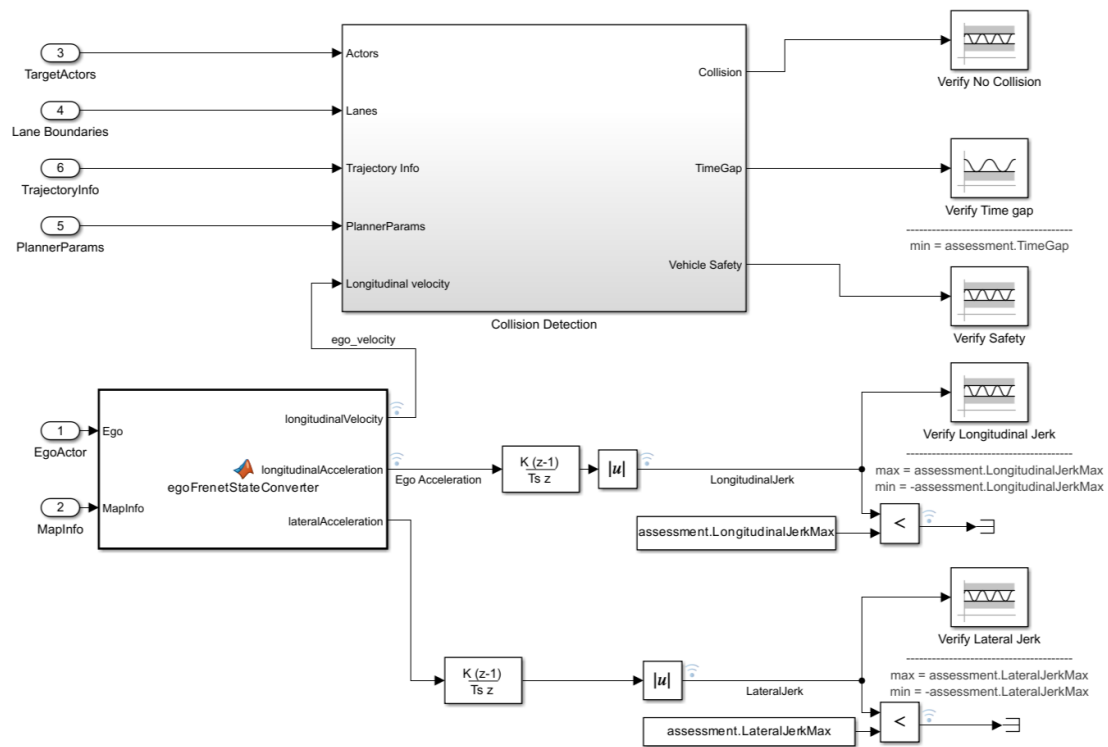


Figura 3.24: Metrics Assessment subsystem.

egoFrenetStateConverter

Il blocco *MatLab function* denominato *egoFrenetStateConverter* si occupa di ricavare dal Bus "EgoActor", contenente la traiettoria del veicolo *ego* in coordinate globali, e dal Bus "MapInfo", contenente le informazioni relative alla strada, la velocità longitudinale e le accelerazioni longitudinale e laterale del veicolo *ego*. La funzione contenuta all'interno del blocco è la già nota *global2frenet*; la quale ricevendo in input una traiettoria di riferimento e le coordinate globali fornisce in output le coordinate di Frenét [$s, \dot{s}, \ddot{s}, d, d', d''$]. Ottenute tali coordinate vengono fornite in output dal blocco le quantità \dot{s} , \ddot{s} e d'' . Dunque, si precisa che la verifica viene effettuata su grandezze espresse con sistema di riferimento di Frenét.

A partire dalle due accelerazioni vengono ricavate le quantità J_{long} e J_{lat} , le quali indicano rispettivamente il jerk longitudinale ed il jerk laterale.

Collision Detection

Il sottosistema *Collision Detection* si occupa di verificare l'assenza di collisioni, diversificare la distanza effettiva tra il veicolo *ego* ed il veicolo *lead* e di stabilire se il veicolo *ego* è in condizioni di sicurezza. In figura 3.25 ne è stata riportata la struttura generale.

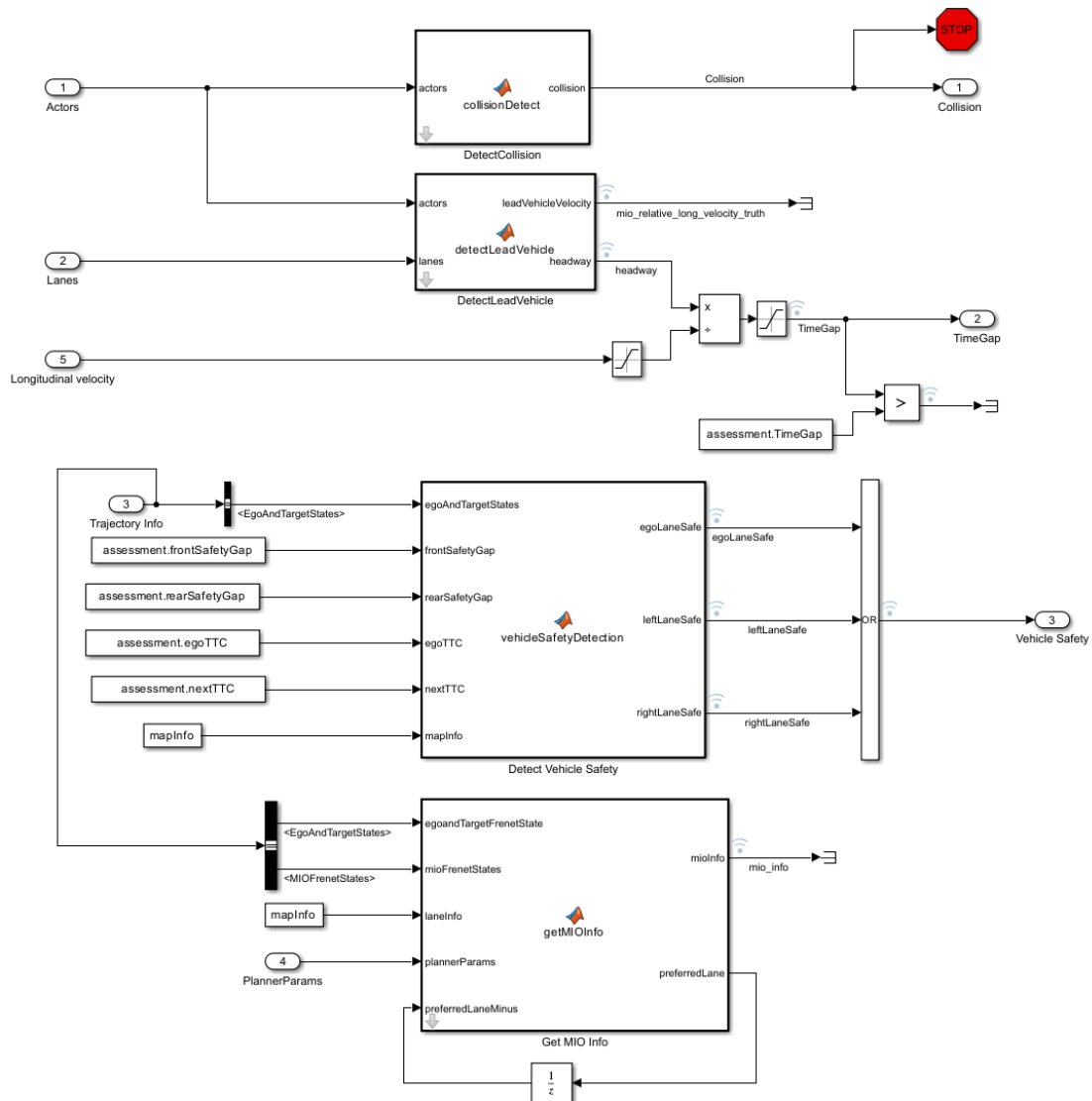


Figura 3.25: Collision Detection subsystem.

Esso si compone di diversi blocchi MatLab *function*; per ognuno di essi ne è stata riportata una descrizione sintetica:

- *Detect Collision*: Il quale riceve in input la posizione e le caratteristiche geometriche degli attori presenti nello scenario. Viene verificata la presenza di sovrapposizioni mediante un confronto tra le aree occupate dai singoli veicoli. In caso di aree sovrapposte nel dato step temporale, si verifica una collisione; dunque, la simulazione viene interrotta mediante un blocco *Stop*.
- *Detect Lane Vehicle*: Il quale riceve in input la posizione e le caratteristiche geometriche degli attori presenti nello scenario. Viene individuato il veicolo *lead*, mediante tecniche già illustrate in precedenza, e si calcola il parametro *headway*; il quale indica la distanza tra la parte frontale del veicolo *ego* e la parte posteriore del veicolo *lead*. Dopodiché si calcola il parametro "TimeGap" con la seguente formula:

$$TimeGap = \frac{Headway}{Longitudinal\ velocity} \quad (3.14)$$

Esso indica il tempo necessario al veicolo *ego* per raggiungere il punto attualmente occupato dal veicolo *lead*.

- *Detect Vehicle Safety*: Il quale si occupa di stabilire se la corsia attualmente occupata dal veicolo *ego* e/o se le corsie ad essa adiacenti sono definite *Safe/Unsafe*. La metodologia e le funzioni utilizzate per definire se una corsia sia sicura sono le medesime definite nel corso dell'analisi del sottosistema *Terminal State Sampler Subsystem*. Per approfondimenti maggiori si rimanda all'analisi eseguita in merito nel paragrafo 3.4. Mediante l'operatore logico *OR* il parametro "Vehicle Safety" assume il valore logico "true" (1) quando una delle tre corsie citate risulta *Safe*.

- *Get MIO Info*: Il quale è dedicato alla creazione della struttura "mioInfo" in output. Tale segnale non viene processato all'interno del modello ma mediante un *inspector* viene salvato istante per istante nel Workspace, in modo da poter essere esaminato nel corso del *post processing*. La struttura in output contiene le informazioni relative ad ogni singolo veicolo identificato come MIO: distanza, velocità relativa e TTC calcolati rispetto al veicolo *ego* e il parametro "IsSafe". Quest'ultimo assume il valore logico "false", quando una delle grandezza citate non rispetta il relativo valore limite. La struttura viene ricavata utilizzando la funzione *helperFindPreferredLane*, la quale è già stata esaminata in precedenza; per un'analisi maggiormente approfondita della stessa si rimanda all'appendice A.

Segnali di allarme

Per le grandezze ricavate dai due blocchi analizzati vengono effettuate delle verifiche che portano alla generazione di segnali di allarme se non rispettate; i quali sono visibili in tempo reale nel Workspace.

Le verifiche effettuate sono le seguenti:

- *Verify No Collision*: Qualora il segnale fosse diverso da 0, dunque venisse rilevata una collisione, la simulazione viene interrotta.
- *Verify Time Gap*: Qualora il segnale fosse inferiore alla soglia minima imposta, nel Workspace viene generata una stringa che comunica all'utente l'istante in cui il segnale oltrepassa la soglia.
- *Verify Safety*: Qualora il segnale fosse diverso da 1, nel Workspace viene generata una stringa che comunica all'utente l'istante in cui sia la corsia attualmente occupata dal veicolo *ego* che le due corsie adiacenti sono contemporaneamente non sicure.
- *Verify Longitudinal Jerk* e *Verify Lateral Jerk*: Qualora i segnali uscissero dalla regione compresa tra la soglia massima e la soglia minima, nel Workspace viene generata una stringa che comunica all'utente l'istante in cui una delle due soglie viene oltrepassata.

Si specifica che i limiti sul *Time Gap* e sul *Jerk* vengono definiti manualmente mediante uno *script* dedicato. I parametri di default sono riportati nella tabella 3.3.

Parameter	Value	Unit
Time Gap	0.8	<i>s</i>
J_{long} min	-5	m/s^3
J_{long} max	5	m/s^3
J_{lat} min	-5	m/s^3
J_{lat} max	5	m/s^3

Tabella 3.3: Parametri limite verifiche *Metrics Assessment*

3.1.7 Visualization

Il blocco *Visualization* è dedicato alla creazione di un riscontro grafico della simulazione, fornito ed aggiornato in tempo reale. Esso riceve in input i principali segnali di tipo Bus del modello, contenenti dati relativi al veicolo *ego*, agli altri veicoli presenti nello scenario, agli output del pianificatore, incluse tutte le traiettorie non ottimali, e le caratteristiche geometriche degli attori.

Al proprio interno ospita un codice MatLab dedicato al *post processing* dei segnali ricevuti in input. Non apportando modifiche ai segnali citati, non risulta essere un componente attivo del sistema assistenziale ma uno strumento utile al suo sviluppo; in quanto permette di analizzare visivamente l'evoluzione degli stati del veicolo all'interno dello scenario. Per tale motivo è stato ritenuto opportuno soffermarsi sull'output del blocco. Esso consiste in un *live plot*, denominato *Lane Change Status Plot*, la cui interfaccia è riportata in figura 3.26.

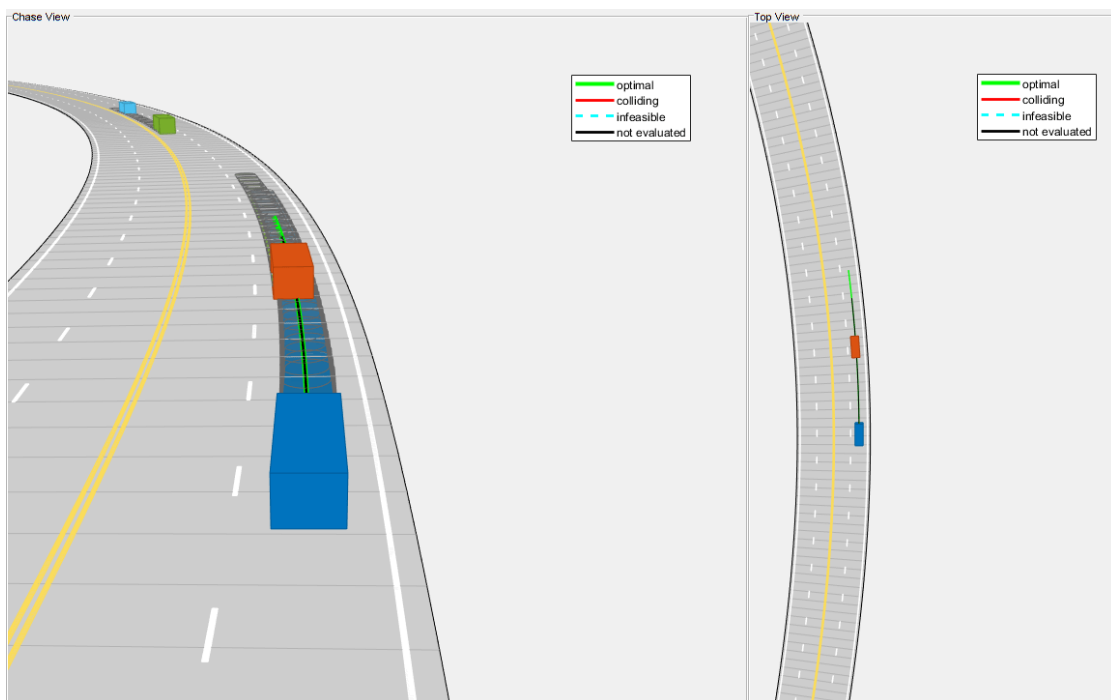


Figura 3.26: Visualization subsystem - *Lane Change Status Plot*.

Sono disponibili due differenti visuali:

- "chase view": vista dell'inseguimento. La quale permette di avere una vista ravvicinata del veicolo *ego*, ma con un'angolazione tale da visualizzare un'ampia parte dello scenario.
- "top view": vista dall'alto. Utile per comprendere le distanze effettive tra i veicoli.

Le informazioni tradotte in forma grafica sono le seguenti:

- Tutte le traiettorie pianificate: Ad ogni step si ha un riscontro grafico di tutte le traiettorie disponibili nel set di alternative; la cui differente colorazione ne indica lo status. I segmenti in rosso indicano le traiettorie che portano ad una collisione, quelle in azzurro indicano le traiettorie che non rispettano i limiti sulla fattibilità cinematica, quelle in nero indicano le traiettorie non valutate. Le traiettorie che non vengono valutate sono quelle che hanno un parametro di costo superiore a quella scelta come riferimento.
- La traiettoria di riferimento: Essa viene rappresentata con un segmento di colore verde.
- I veicoli nello scenario: Essi vengono rappresentati attraverso dei parallelogrammi di colorazioni differenti.
- Capsule 2D: Sulla superficie stradale vengono riportate le rappresentazioni in capsule utilizzate per la verifica sulle collisioni. Nello specifico si hanno tutti gli stati futuri dei veicoli presenti nello scenario associati alla traiettoria corrente.

Un ulteriore strumento disponibile per il post processing è il *Lane Change Simulation Analysis*. Si tratta di un *tool* che analizza la struttura "logsout", la quale racchiude l'andamento di tutti i segnali del modello nel corso del tempo. Essi vengono salvati mediante *inspector* all'interno di appositi campi della struttura "logsout" e possono essere aggiunti o rimossi manualmente. Il *Lane Change Simulation Analysis* ha anch'esso un'interfaccia grafica che include la visuale *chase view*. Ne è stato riportato un esempio in figura 3.27

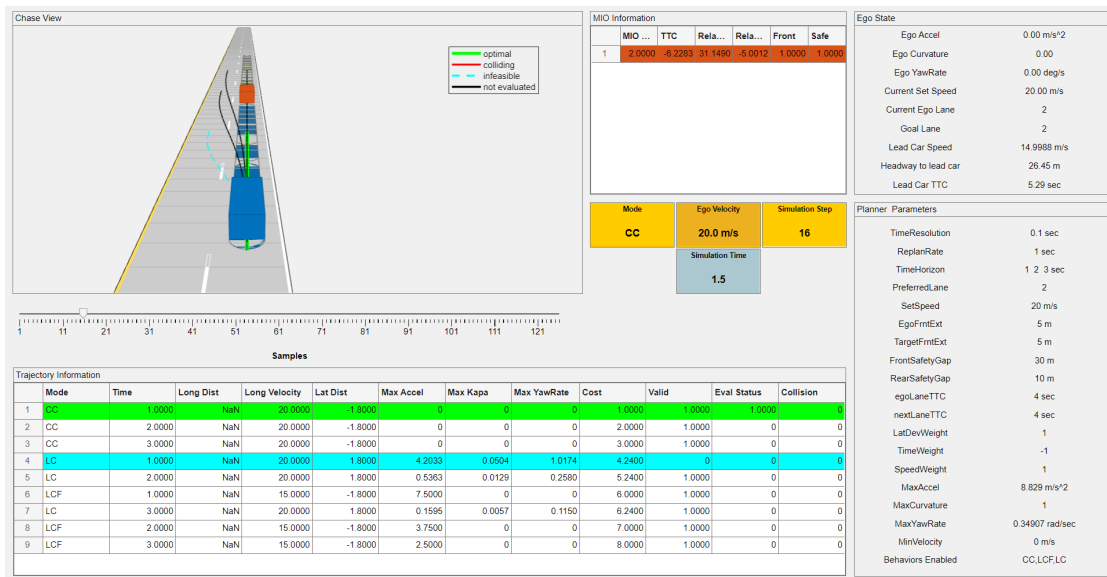


Figura 3.27: Visualization subsystem - Lane Change Simulation Analysis.

L'interfaccia dispone di un cursore che permette di modificare lo step da analizzare. Le informazioni sono raggruppate nelle seguenti schede:

- *Chase view*: La quale fornisce un riscontro visivo dello scenario nello step selezionato.
- *Trajectory Information*: La quale mostra, per lo step selezionato, il set di traiettorie disponibili ordinate secondo una logica preferenziale. Per ognuna di esse è possibile visualizzare: l'orizzonte temporale, il valore del parametro di costo, i falg "IsValid", "IsEvaluated" e "IsColliding", la velocità, la distanza del punto terminale rispetto al centro della corsia ed i valori di picco delle grandezze ad essa associate.
- *MIO Information*: La quale fornisce le informazioni dei veicoli *MIO* presenti nello scenario, quali: l'indice, il TTC, distanza e velocità relative e valutazione *Safe/Unsafe*.
- *Ego State*: la quale fornisce informazioni sulle principali grandezze associate al moto del veicolo *ego*.

- *Planner Parameters*: La quale fornisce i valori dei parametri di configurazione del pianificatore di traiettorie, utilizzati per la simulazione
- Infine, attraverso delle apposite finestre è possibile vedere la modalità attualmente utilizzata (CC, LCF e LC), la velocità del veicolo *ego*, lo step ed il tempo di simulazione attuali.

3.2 Modello di veicolo e controllore

Nel corso degli studi effettuati in questa tesi è stato ritenuto opportuno inserire all'interno del modello Simulink precedentemente analizzato un modello di veicolo ed il relativo controllore. Tale scelta viene motivata con l'esigenza di testare le traiettorie definite dal sistema di *path planning* mediante un modello di veicolo e non esclusivamente attraverso la verifica di fattibilità, da effettuare sui valori di picco delle grandezze descrittive della traiettoria da seguire.

L'introduzione dei due nuovi componenti, controllore e modello di veicolo, ha portato alla disponibilità di numerosi nuovi segnali, anch'essi sfruttabili dal pianificatore per poter ottenere delle traiettorie da seguire sempre più precise. In tal senso è stato possibile mettere a punto una strategia di differenziazione degli obiettivi di pianificazione, effettuata mediante l'analisi delle grandezze in output dal modello di veicolo.

In continuità con quanto esposto fin'ora, il primo passo effettuato nella direzione di un modello con una complessità superiore è stato quello di prendere visione delle opzioni disponibili per l'ambiente di simulazione utilizzato. Attraverso tale ricerca è stato individuato il modello *Highway Lane Change Planner and Controller* sviluppato dalla *MathWorks*, il quale è l'evoluzione del modello analizzato nel dettaglio all'interno del paragrafo precedente. Esso collega il pianificatore di traiettoria ad un controllore di tipo *Model Predictive Control* (MPC) ed un modello di veicolo abbastanza semplificato, quale il modello a bicicletta [24].

Tuttavia, in seguito ad alcune considerazioni effettuate a valle di un'analisi dettagliata di tutte le parti, la strategia adottata è stata quella di mantenere l'architettura generale del modello, variandone i componenti. Nello specifico, il modello di veicolo utilizzato è stato sostituito in toto, poiché frutto di ipotesi eccessivamente stringenti che non avrebbero permesso di implementare alcune delle funzionalità poste come obiettivo di questo lavoro di tesi. Il pianificatore, essendo identico a quello analizzato all'interno del caso precedente, è stato aggiornato con tutte le funzionalità introdotte nel capitolo quarto, le quali sono state prima ottimizzate con il modello comprensivo del solo pianificatore. Infine, il controllore MPC è stato adattato alle variazioni apportate agli altri blocchi del sistema.

3.2.1 Architettura del modello

Con l'introduzione dei due nuovi componenti, controllore e modello di veicolo, l'architettura generale del modello varia considerevolmente. Precedentemente alla suddetta modifica, la traiettoria del veicolo *ego* risultava coincidente in ogni suo punto con la traiettoria in output al pianificatore. La presenza di un modello di veicolo permette di avvicinarsi ulteriormente ad un sistema reale; in quanto le traiettorie pianificate vengono processate da un controllore, il quale fornisce i comandi al veicolo *ego*. Grazie alla presenza del modello di veicolo è possibile verificare se le traiettorie di riferimento prodotte dal pianificatore risultano fisicamente percorribili dal veicolo *ego*, tenendo conto dei relativi ritardi di attuazione e di tutte le caratteristiche fisiche che ne descrivono il comportamento.

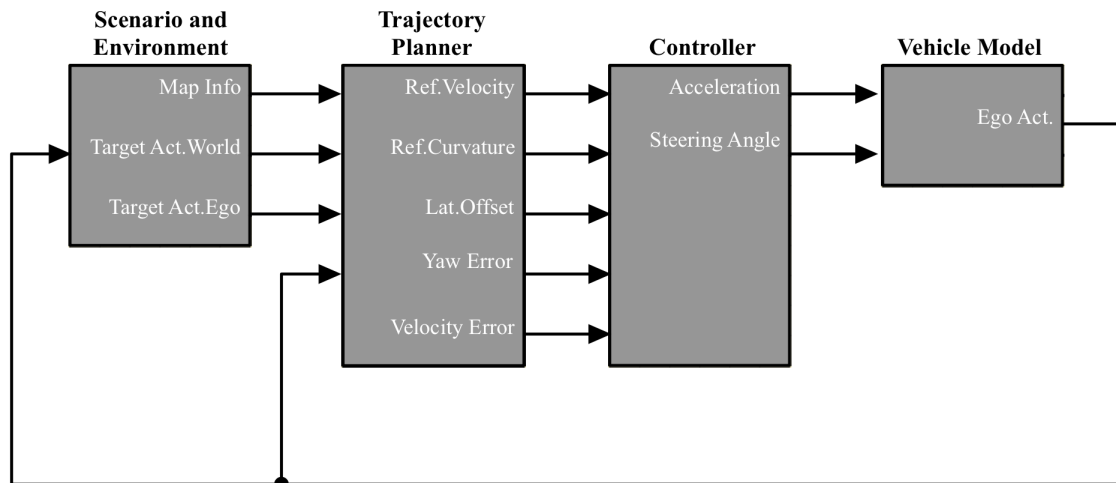


Figura 3.28: Schema a blocchi modello con controllore

Nello schema riportato in figura 3.28 si ha una rappresentazione grafica del funzionamento del modello utilizzato. Nello specifico, sono stati riportati i quattro principali blocchi che cooperano nel corso della simulazione ed i segnali che ne regolano la reciproca comunicazione. Si tratta di una rappresentazione semplificata, realizzata in quanto chiarisce le connessioni necessarie tra le varie parti e la logica che regola il funzionamento dei principali attori del sistema assistenziale.

- *Scenario and Environment*: In questo blocco viene confrontata la posizione del veicolo *ego*, memorizzata nel segnale *Ego Actor*, con lo scenario e con i veicoli Target presenti. Si producono in output i segnali contenenti la posizione dei veicoli Target, calcolata rispetto ad un sistema di riferimento globale (*Target Actor World*) e rispetto alla posizione del veicolo *ego* (*Target Actor Ego*). Inoltre, il segnale *MapInfo* contiene tutte le informazioni descrittive dello scenario.
- *Trajectory Planner* o *Highway Lane Change Planner*: In questo blocco è contenuto il fulcro del sistema assistenziale. Vengono analizzati i segnali provenienti dal blocco precedente, confrontati con i dati che descrivono il moto del veicolo *ego* (segnale *Ego Actor*), per stabilire la traiettoria da seguire. I segnali in output sono la velocità di riferimento (*Ref. Velocity*), la curvatura di riferimento (*Ref. Curvature*) e gli errori, laterale (*Lat. Offset*), angolare (*Yaw Error*) e di velocità (*Velocity Error*).
- *Controller* o *Lane Change Controller*: Contiene al proprio interno il controllore MPC. Confronta i dati in output al blocco precedente e la velocità longitudinale attuale del veicolo *Longitudinal Velocity* per stabilire l'accelerazione (*Acceleration*) e l'angolo di sterzo (*Steering Angle*) necessari per seguire la traiettoria stabilita dal pianificatore ed a mantenere la velocità di riferimento.
- *Vehicle Model* o *Vehicle Dynamics*: Si tratta del blocco contenente il modello di veicolo. Quello utilizzato è della tipologia *Double Track* con modellizzazione non lineare di pneumatici. Esso riceve i segnali di comando e permette di calcolare posizione, velocità ed accelerazione del veicolo *ego*. Tali parametri vengono convogliati nel segnale *Ego Actor*, il quale viene successivamente fornito in input ai blocchi *Scenario and Environment* e *Trajectory Planner*.

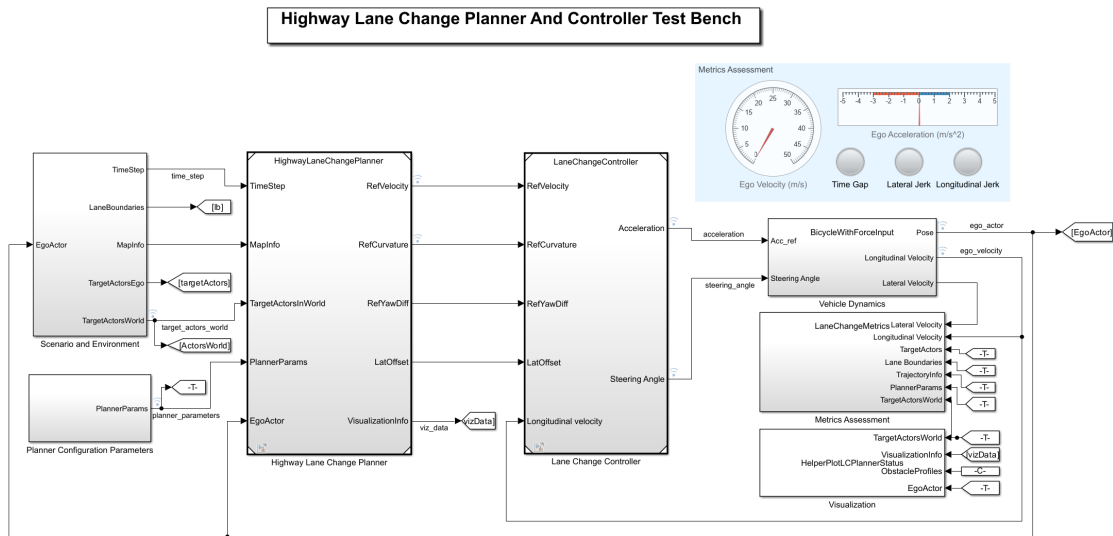


Figura 3.29: Struttura modello Simulink con controllore

Nella figura 3.29 è stata riportata la struttura generale del modello Simulink utilizzato per le simulazioni con modello di veicolo. Come si può notare i quattro blocchi principali e le relative connessioni sono i medesimi descritti attraverso la schematizzazione precedente; tuttavia, sono presenti ulteriori blocchi già introdotti e descritti all'interno del paragrafo precedente, quali: *Planner Configuration Parameters*, *Visualization* e *Metrics Assessment*.

Il modello è regolato dai medesimi script del modello del solo pianificatore di traiettoria. Tuttavia, essi sono stati rimpolpati da un file Matlab che permette di variare i parametri di configurazione del pianificatore e dai file che contengono i parametri descrittivi del veicolo *ego*.

3.2.2 Lane Change Controller

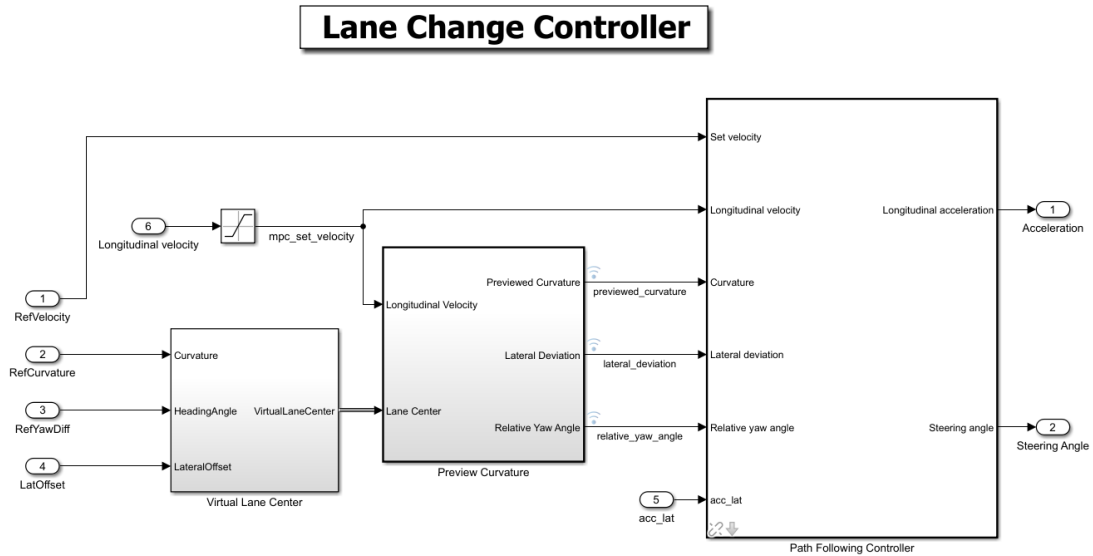


Figura 3.30: Sottosistema Simulink controllore

Il sottosistema *Lane Change Controller* ha il compito di processare i segnali che vengono forniti in input, contenenti informazioni circa la traiettoria da seguire, e produce i comandi utili a governare il veicolo *ego*.

I segnali che il sottosistema riceve in input sono i seguenti:

- *LatOffset*: Distanza in termini assoluti tra la posizione effettiva del veicolo *ego* e la posizione di riferimento nel medesimo istante temporale. Tale valore quantifica l'errore laterale. In figura 3.31 è stato rappresentato come $E_{lateral}$. La formula utilizzata per il calcolo è la seguente:

$$LatOffset = \sqrt{(X_{ego} - X_{ref})^2 + (Y_{ego} - Y_{ref})^2} \quad (3.15)$$

I dati necessari a stimare tale grandezza vengono generalmente ricavati da un sistema GPS con correzione *Real Time Kinematic* (RTK) o da telecamere in grado di riconoscere la segnaletica orizzontale.

- *Heading Angle*: Differenza tra il valore dell'angolo di imbardata effettivo del veicolo *ego* e angolo di imbardata di riferimento nel medesimo istante temporale. Tale valore quantifica l'errore angolare. In figura 3.31 è stato rappresentato come E_{yaw} . La formula utilizzata per il calcolo è la seguente:

$$\text{Heading Angle} = \psi_{ref} - \psi_{ego} \quad (3.16)$$

I dati necessari a stimare tali grandezze vengono generalmente ottenuti da sensori in grado di leggere la posizione del veicolo all'interno della strada, come ad esempio le telecamere in grado di riconoscere la segnaletica orizzontale.

- *RefCurvature* e *RefVelocity*: Segnali di curvatura e velocità di riferimento.
- *Longitudinal velocity* e *Lateral acceleration*: Segnali di velocità longitudinale e accelerazione laterale associati allo stato attuale effettivo del veicolo *ego*.

Nella precedente disamina sono stati citati i sensori di uso comune per la stima delle grandezze con le quali vengono calcolati gli errori. All'interno di questo modello essi vengono calcolati a partire dai dati esatti di posizione e orientamento in output dal modello di veicolo. La quale rappresenta un'ipotesi importante e da tenere sempre in considerazione nel corso dell'analisi del modello.

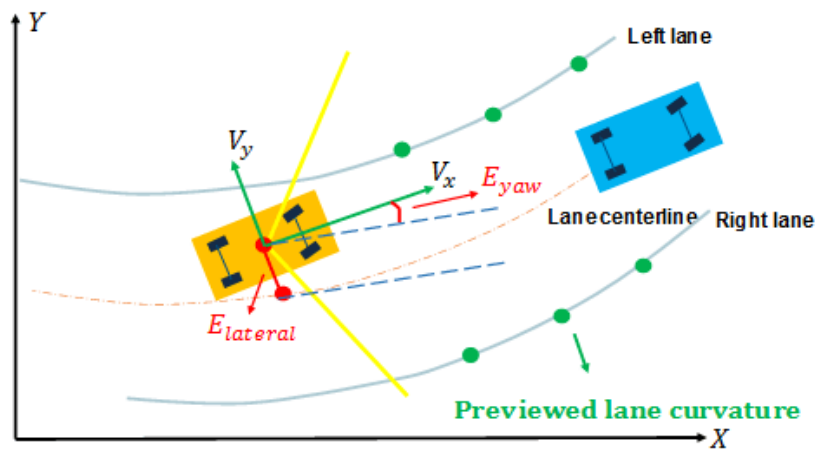


Figura 3.31: Controllore path following [26].

I segnali che quantificano gli errori laterale ed angolare sono fondamentali per un controllore; in quanto permettono di correlare l'entità della correzione sulla traiettoria e sull'accelerazione da dover apportare. Inoltre, la presenza dei due segnali di velocità longitudinale permette al sistema di calcolarne il relativo errore; quest'operazione viene effettuata all'interno del blocco *Path Following Controller*.

Il blocco *Virtual Lane Center* è dedicato alla creazione di un segnale Bus che contiene i singoli segnali relativi a: offset laterale, errore angolare e curvatura della traiettoria di riferimento. Essi vengono processati all'interno del blocco *Preview Curvature*, adattandoli alle convenzioni di segno richieste dal blocco *Path Following Controller*. Quest'ultimo rappresenta il cuore del sottosistema dedicato al controllore. Si tratta di un blocco disponibile nelle librerie dell'*Automated Driving Toolbox*, il quale simula un controllore *path following* che fornisce i comandi utili ad inseguire la traiettoria di riferimento e mantiene una distanza di sicurezza dal veicolo che lo precede [21] [26]. Tuttavia, all'interno del modello la funzione dedicata al mantenimento della distanza di sicurezza è stata disabilitata; in quanto è il pianificatore a stabilire la corretta traiettoria, in relazione al posizionamento di tutti i veicoli presenti nello scenario.

MPC Controller

Il *Path Following Controller* combina le funzionalità di un LKA e di un ACC, utilizzando un controllore di tipo *adaptive model predictive controller* o MPC adattivo.

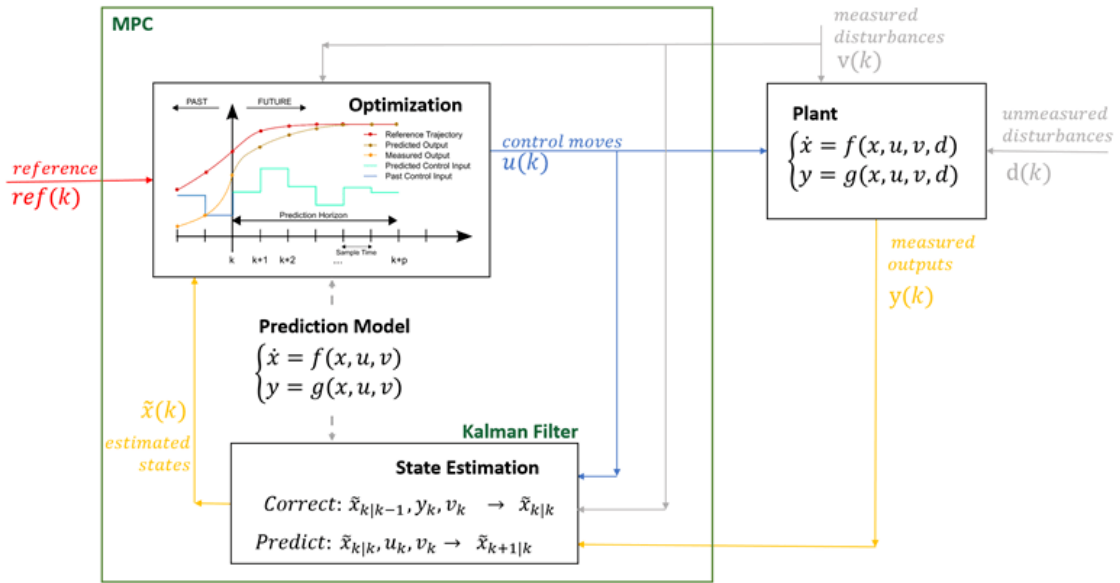


Figura 3.32: Ciclo base di controllo [27].

Nella figura 3.32 è stato riportato lo schema del funzionamento di base di un controllore MPC. Dato un modello da controllare (*Plant*), l'MPC è in grado di stimare gli stati del controllore e di predire gli output del *Plant* grazie alla presenza di un ulteriore modello di veicolo integrato. Utilizzando gli output previsti, esso programma una sequenza di azioni di controllo mirate a risolvere un problema di ottimizzazione che minimizza una funzione di costo su un dato orizzonte temporale. Tale funzione è tipicamente finalizzata a ridurre gli errori tra il riferimento e l'output del *Plant*. Ad ogni step temporale il controllore attua la prima azione di controllo programmata, inviando i segnali elaborati al modello *Plant*. Nel passo temporale successivo l'azione si ripete [27]. Bisogna precisare che nello specifico caso di questo studio non sono stati inclusi i disturbi misurati e non misurati all'interno del modello e, come già anticipato, che è stata introdotta l'ipotesi di conoscere

con esattezza gli stati effettivi del veicolo; dunque, le stime vengono effettuate solo nella fase di pianificazione.

Il controllore di tipo MPC prevede il comportamento futuro del modello di veicolo, utilizzando un ulteriore modello integrato che sia di tipo lineare-invariante nel tempo (LTI). Dove per LTI si definisce un sistema che produce segnali di output a partire da segnali input soggetti a vincoli di linearità ed invarianza temporale.

L'MPC di tipo adattivo rappresenta un upgrade del controllore MPC standard, finalizzato a lavorare in combinazione con modelli con caratteristiche non lineari o che varino fortemente nel tempo. L'MPC adattivo utilizza una struttura di modello fissa, anch'essa di tipo LTI, ma a differenza del controllore MPC standard aggiorna ad ogni step temporale i parametri del modello integrato e le condizioni nominali, migliorando la risposta complessiva del controllore [28]. Il parametro che nel modello originale viene aggiornato ad ogni step temporale è la velocità longitudinale v_x . Successivamente è stato implementato l'aggiornamento anche dei parametri di rigidità in deriva, data la forte variazione che essi subiscono in campo di funzionamento non lineare. L'MPC adattivo utilizzato in questo studio, come anticipato, combina le funzionalità di un ACC, dedicato a produrre un comando sull'acceleratore, con quelle di un LKA, dedicato a produrre un comando sul volante.

I due modelli matematici ed il modello integrato, comprensivo sei modelli ACC e LKA, vengono scritti utilizzando lo spazio degli stati:

$$\begin{cases} \dot{x}(t) &= A x(t) + B u(t) \\ y(t) &= C x(t) + D u(t) \end{cases} \quad (3.17)$$

Dove:

- A, B, C, D: matrici che descrivono il sistema.
- $x(t)$: stati del sistema dinamico.
- $u(t)$: ingresso del sistema.
- $y(t)$: uscita del sistema.

Adaptive Cruise Control Predictive Model

L'*Adaptive Cruise Control Predictive Model* integrato nel controllore, fornisce in output la velocità longitudinale v_x , ricevendo in input l'accelerazione longitudinale a_x . Si riportano le equazioni di stato in forma estesa:

$$\begin{cases} \tau \dot{a}_{x\ out} + a_{x\ out} = a_{x\ in} \\ \dot{v}_{x\ out} = a_{x\ out} \end{cases} \quad (3.18)$$

Dove τ è la costante temporale. Si definiscono gli stati del sistema dinamico $x(t)$, l'output $y(t)$ e l'input $u(t)$:

$$x(t) = \begin{bmatrix} a_{x\ out} \\ v_{x\ out} \end{bmatrix} \quad y(t) = v_{x\ out} \quad u(t) = a_{x\ in} \quad (3.19)$$

Le matrici del modello *state-space* sono così definite:

$$A_1 = \begin{bmatrix} -\frac{1}{\tau} & 0 \\ 1 & 0 \end{bmatrix} \quad B_1 = \begin{bmatrix} \frac{1}{\tau} \\ 0 \end{bmatrix} \quad C_1 = [0 \quad 1] \quad D_1 = 0 \quad (3.20)$$

Il fine di questo modello è quello di introdurre un ritardo nel segnale di accelerazione, in modo da simulare il ritardo di attuazione del comando e l'inerzia del veicolo. Dunque, la dinamica fra accelerazione e velocità del veicolo è governata dalla seguente funzione di trasferimento:

$$\frac{v_{x\ out}}{a_{x\ in}} = G(s) = \frac{1}{s(\tau s + 1)} \quad (3.21)$$

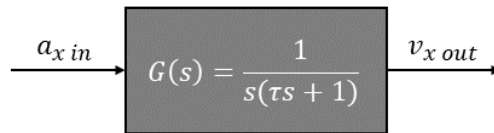


Figura 3.33: Modello a blocchi funzione di trasferimento.

Lane-Keeping Predictive Model

Il *Lane-Keeping Predictive Model* integrato nel controllore, fornisce in output la velocità laterale v_y e la velocità angolare $\dot{\psi}$, ricevendo in input l'angolo di sterzo alle ruote δ . Si tratta delle equazioni che descrivono il moto di un modello di veicolo *single track* o bicicletta. Si riportano le equazioni di stato in forma estesa:

$$\begin{aligned} m(\dot{v}_y + v_x \dot{\psi}) &= -C_r \frac{v_y - b \dot{\psi}}{v_x} + C_F \delta_f - C_F \frac{v_y + a \dot{\psi}}{v_x} \\ I_Z \ddot{\psi} &= b C_R \frac{v_y - b \dot{\psi}}{v_x} + C_F a \delta_f - C_F a \frac{v_y + a \dot{\psi}}{v_x} \end{aligned} \quad (3.22)$$

Dove:

- v_x : Velocità longitudinale del veicolo.
- m : Massa totale del veicolo.
- I_Z : Momento d'inerzia attorno all'asse z .
- a : Semipasso anteriore.
- b : Semipasso posteriore.
- C_F : Rigidezza in deriva dell'assale anteriore.
- C_R : Rigidezza in deriva dell'assale posteriore.

Si definiscono gli stati del sistema dinamico $x(t)$, l'output $y(t)$ e l'input $u(t)$:

$$x(t) = \begin{bmatrix} v_y \\ \dot{\psi} \end{bmatrix} \quad y(t) = \begin{bmatrix} v_y \\ \dot{\psi} \end{bmatrix} \quad u(t) = \delta_f \quad (3.23)$$

Le matrici del modello *state-space* sono così definite:

$$\begin{aligned}
 A_2 &= \begin{bmatrix} \frac{-(C_F + C_R)}{m v_x} & \frac{-(C_F a - C_R b)}{m v_x} - v_x \\ \frac{-(C_F a - C_R b)}{I_Z v_x} & \frac{-(C_F a^2 + C_R b^2)}{I_Z v_x} \end{bmatrix} \\
 B_2 &= C_F \begin{bmatrix} 1 \\ m \\ a \\ I_Z \end{bmatrix} \\
 C_2 &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\
 D_2 &= \begin{bmatrix} 0 \\ 0 \end{bmatrix}
 \end{aligned} \tag{3.24}$$

Dove:

- v_y : Velocità laterale del veicolo.
- $\dot{\psi}$: Velocità di imbardata del veicolo.
- δ_f : Angolo di sterzo.

Combined Model

I due modelli vengono combinati in un unico modello *state-space* integrato avente i seguenti vettore di stato e vettore di uscite.

$$x(t) = \begin{bmatrix} a_{x\ out} \\ v_{x\ out} \\ v_{y\ out} \\ \dot{\psi}_{out} \end{bmatrix} \quad u(t) = \begin{bmatrix} a_{x,in} \\ \delta_f \end{bmatrix} \tag{3.25}$$

Le matrici del modello *state-space* complessivo sono così definite.

$$\begin{aligned}
 A &= \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} \\
 B &= \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix} \\
 C &= \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix} \\
 D &= \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix}
 \end{aligned}
 \tag{3.26}$$

Il modello a blocchi riportato in figura 3.34, fornisce una rappresentazione grafica del suo funzionamento.

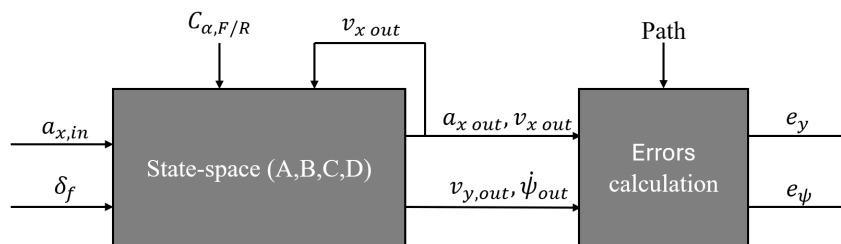


Figura 3.34: Modello a blocchi funzionamento modello di veicolo integrato.

Come più volte detto esso viene utilizzato per prevedere la risposta del veicolo a fronte di determinati output del controllore, i quali sono l'accelerazione longitudinale e l'angolo di sterzo. Conseguentemente tale modello integrato riceve proprio questi due parametri come segnali di input ($a_{x,in}$, δ_f). Essi vengono elaborati dal modello *state-space* per produrre i segnali di: accelerazione longitudinale, la quale viene processata da una funzione di trasferimento $G(s)$, velocità longitudinale e laterale e velocità di imbardata. Si ricorda che il modello è di tipo LTI, dunque, i parametri di configurazione delle matrici (A, B, C e D) sono costanti. Tuttavia, il controllore è di tipo adattivo, dunque, alcuni dei parametri delle matrici citate vengono aggiornati ad ogni step.

Per tale motivo si ha una retroazione della velocità longitudinale calcolata e l'ingresso delle rigidezze in deriva nel blocco *state-space*, le quali vengono calcolate in base al valore corrente di accelerazione laterale. Infine, tali dati vengono successivamente processati per calcolare gli errori laterale ed angolare, rispettivamente e_y ed e_ψ .

Funzione di costo controllore MPC

Il controllore di tipo *Model Predictive Control* (MPC) risolve un problema di ottimizzazione quadratica (QP) ad ogni intervallo di controllo. La soluzione del problema permette di identificare le variabili in output da fornire al modello *Plant* nello step di simulazione successivo. Il problema viene risolto con l'obiettivo di minimizzare una data funzione di costo.

La funzione di costo standard $J(z_k)$ si compone di quattro termini, ognuno dei quali centrato su un particolare aspetto del controllore.

$$J(z_k) = J_y(z_k) + J_u(z_k) + J_{\Delta u}(z_k) + J_\epsilon(z_k) \quad (3.27)$$

Dove z_k rappresenta la soluzione del problema di ottimizzazione quadratica risolto nell'istante k di simulazione, anche denominato "vettore delle variabili decisionali del QP".

$$z_k^T = [u(k|k)^T \ u(k+1|k)^T \ \dots \ u(k+p-1|k)^T \ \epsilon_k] \quad (3.28)$$

Tracciamento del riferimento di uscita ($J_y(z_k)$): Il controllore MPC deve mantenere le uscite del sistema vicine a valori di riferimento, per cui viene effettuata una misura delle performance in termini di distanza dal dato riferimento. Tale grandezza è il primo dei quattro termini che compongono la funzione di costo [29].

$$J_y(z_k) = \sum_{j=1}^{n_y} \sum_{i=1}^p \left\{ \frac{w_{i,j}^y}{s_j^y} [r_j(k+i|k) - y_j(k+i|k)] \right\}^2 \quad (3.29)$$

Dove:

- k : Step attuale della simulazione.

- p : Orizzonte predittivo.
- n_y : Numero di variabili in output dal *Plant*.
- z_k : Vettore delle variabili decisionali del QP.
- $y_j(k+i|k)$: Valori previsti del j-esimo output dell'i-esimo step di predizione.
- $r_j(k+i|k)$: Valori di riferimento del j-esimo output dell'i-esimo step di predizione.
- s_j^y : Fattore di scala del j-esimo output del *Plant*.
- $w_{i,j}^y$: Parametro di calibrazione che stabilisce il peso del j-esimo output del plant nell'i-esimo step di predizione.
- ϵ_k : Variabile slack nello step della simulazione k , la quale fornisce un'indicazione sullo scostamento rispetto al riferimento attuale.

Il controllore riceve i valori di riferimento $r_j(k+i|k)$ per l'intero orizzonte temporale. Esso calcola in previsione gli output del *Plant* $y_j(k+i|k)$, i quali dipendono dalle variabili manipolate z_k , dai disturbi e dalla stima degli stati; ricordando che nel caso in esame vengono trascurati i disturbi e si assume la conoscenza esatta degli stati attuali del *Plant*.

Tracciamento delle variabili manipolate ($J_u(z_k)$):

$$J_u(z_k) = \sum_{j=1}^{n_u} \sum_{i=0}^{p-1} \left\{ \frac{w_{i,j}^u}{s_j^u} [u_j(k+i|k) - u_{j,target}(k+i|k)] \right\}^2 \quad (3.30)$$

Dove:

- n_u : Numero di variabili manipolate.
- $u_{j,target}(k+i|k)$: Valore di riferimento della j-esima variabile manipolata nell'i-esimo istante step di predizione.
- s_j^u : Fattore di scala della j-esima variabile manipolata.

- $w_{i,j}^u$: Parametro di calibrazione che stabilisce il peso della j -esima variabile manipolata nell' i -esimo step di predizione.

Il controllore riceve i valori di riferimento delle variabili manipolate $u_{j,target}(k+i|k)$ per l'intero orizzonte temporale.

Riduzione di variazione della variabile manipolata nell'intervallo T_s ($J_{\Delta u}(z_k)$):

$$J_{\Delta u}(z_k) = \sum_{j=1}^{n_u} \sum_{i=0}^{p-1} \left\{ \frac{w_{i,j}^{\Delta u}}{s_j^u} [u_j(k+i|k) - u_j(k+i-1|k)] \right\}^2 \quad (3.31)$$

Dove $w_{i,j}^{\Delta u}$ è il parametro di calibrazione che stabilisce il peso della variazione della j -esima variabile manipolata nell' i -esimo step di predizione.

Violazione dei vincoli ($J_\epsilon(z_k)$): Nella pratica di utilizzo di questo controllore si può incorrere inevitabilmente nella violazione delle soglie imposte. Conseguentemente il modello utilizza dei vincoli *soft* che in taluni casi possono essere violati. La variabile slack inserita ϵ_k quantifica la violazione del vincolo nel caso peggiore.

$$J_\epsilon(z_k) = \rho_\epsilon \epsilon_k^2 \quad (3.32)$$

Dove ρ_ϵ rappresenta il peso della penalità per la violazione del dato vincolo.

Si rimanda alla letteratura per l'analisi di funzioni di costo alternative a quella standard [29].

Calcolo degli errori controllore MPC

Esaminando quanto descritto fin'ora emerge che per ogni istante di simulazione il controllore stabilisce una strategia da seguire in base alle previsioni effettuate con il modello integrato, minimizzando la funzione di costo esposta nel sottoparagrafo precedente. Come si può notare dall'equazione 3.30, all'interno della funzione di costo è presente un contributo che è dipendente dall'errore calcolato nel corso di tutto l'orizzonte di previsione per ciascun parametro output j -esimo y_j del modello di veicolo integrato rispetto al relativo riferimento r_j .

Tuttavia, in letteratura sono presenti differenti approcci per il calcolo degli errori. Ad esempio Snider propone la riscrittura delle matrici del modello *state-space* rispetto alla traiettoria di riferimento.

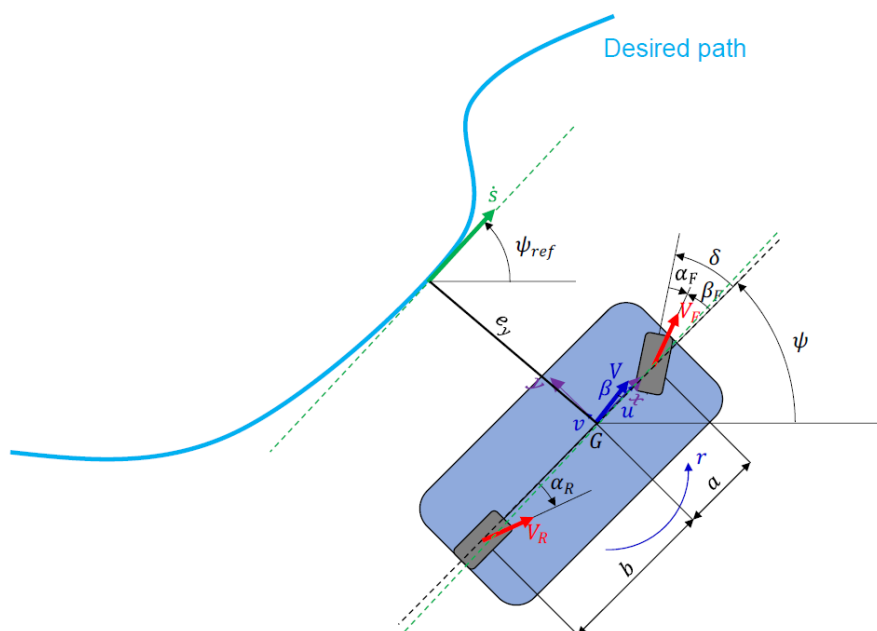


Figura 3.35: Errori di posizionamento modello *single track*.

L'obiettivo posto è quello di scrivere il sistema di equazioni di stato in funzione dell'errore di orientamento e_ψ e di quello di posizione e_y , in modo da rendere più semplice l'implementazione degli stessi all'interno della funzione di costo.

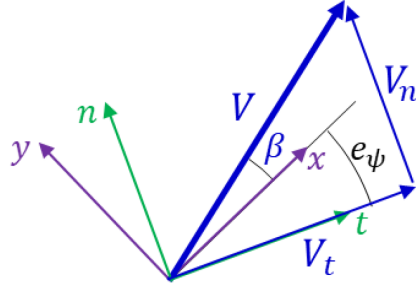


Figura 3.36: Sistema di riferimento solido al veicolo e solido alla traiettoria di riferimento.

Come primo step è necessario scrivere le componenti della velocità del veicolo *ego* rispetto ad un sistema di riferimento mobile e solido con la traiettoria di riferimento, il quale è stato rappresentato in verde in figura 3.36.

$$\begin{cases} V_t = V \cos(\beta + e_\psi) \\ V_n = V \sin(\beta + e_\psi) \end{cases} \quad (3.33)$$

Quindi si applica l'equazione fondamentale della cinematica.

$$V_t = \dot{s} \pm \dot{\psi}_{ref} e_y = \dot{s} \pm \rho_{ref} \dot{s} e_y \quad (3.34)$$

L'errore di deviazione laterale può essere espresso in tal modo.

$$V_n = \dot{e}_y = V \sin(\beta + e_\psi) \quad (3.35)$$

Dunque, può essere scritto il seguente sistema contenente la relazione tra la velocità del veicolo V e la velocità lungo la traiettoria di riferimento \dot{s} .

$$\begin{cases} \dot{s} = \frac{V \cos(\beta + e_\psi)}{1 \pm \rho_{ref} e_y} \\ \dot{e}_y = V \sin(\beta + e_\psi) \end{cases} \quad (3.36)$$

Tali equazioni possono essere linearizzate assumendo angoli di assetto β ed errori di orientamento e_ψ piccoli nell'intorno delle condizioni di equilibrio.

Quindi le funzioni trigonometriche possono essere scritte nel seguente modo.

$$\begin{aligned} V \cos(\beta + e_\psi) &= V[\cos(\beta)\cos(e_\psi) - \sin(\beta)\sin(e_\psi)] \simeq V - V\beta e_\psi = u - ve_\psi \\ V \sin(\beta + e_\psi) &= V[\sin(\beta)\cos(e_\psi) + \cos(\beta)\sin(e_\psi)] \simeq v + ue_\psi \end{aligned} \quad (3.37)$$

Dunque, si riscrive il sistema precedente.

$$\begin{cases} \dot{s} = \frac{u - ve_\psi}{1 \pm \rho_{ref} e_y} \\ \dot{e}_y = v + ue_\psi \end{cases} \quad (3.38)$$

Si ricava la velocità dell'errore di deviazione laterale.

$$\ddot{e}_y = \dot{v} + ue_\psi \quad (3.39)$$

Si ricavano analogamente velocità ed accelerazione dell'errore angolare.

$$\begin{aligned} \dot{e}_\psi &= \dot{\psi} - \dot{\psi}_{ref} \\ \ddot{e}_\psi &= \ddot{\psi} - \ddot{\psi}_{ref} \end{aligned} \quad (3.40)$$

Quindi, sostituendo le espressioni di v , \dot{v} , ψ , $\dot{\psi}$ come funzione dei due errori ricavati nelle equazioni di stato del modello *single-track*, è possibile scrivere il sistema *state-space* associato nel seguente modo.

$$\dot{\mathbf{e}} = \mathbf{A}_p \mathbf{e} + \mathbf{B}_{p1} \delta + \mathbf{B}_{p2} \dot{\psi}_{ref} + \ddot{\psi}_{ref} \quad (3.41)$$

Dove:

$$\begin{aligned}
 \mathbf{e} &= [e_y, \dot{e}_y, e_\psi, \dot{e}_\psi] \\
 \mathbf{A} &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{C_F + C_R}{mV} & \frac{C_F + C_R}{m} & \frac{C_R b - C_F a}{mV} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{C_R b - C_F a}{J_z V} & \frac{C_F a - C_R b}{J_z} & -\frac{C_F a^2 + C_R b^2}{J_z V} \end{bmatrix} \\
 \mathbf{B}_{p1} &= \begin{bmatrix} 0 \\ \frac{C_F}{m} \\ 0 \\ \frac{C_F a}{m} \end{bmatrix} \\
 \mathbf{B}_{p2} &= \begin{bmatrix} 0 \\ \frac{C_R b - C_F a}{mV} - V \\ 0 \\ -\frac{C_F a^2 + C_R b^2}{J_z V} \end{bmatrix}
 \end{aligned} \tag{3.42}$$

Modifiche introdotte

Rigidezze in deriva variabili: Sebbene il controllore MPC di tipo adattivo sia più congeniale rispetto ad un MPC standard per modelli non lineari, le predizioni vengono comunque effettuate utilizzando un modello lineare *state-space*, differente rispetto a quello utilizzato come modello di veicolo nelle ipotesi semplificative introdotte e nella struttura. Conseguentemente, i risultati sono influenzati da approssimazioni che, nei casi più estremi, hanno un impatto elevato.

Per tale ragione è stato ritenuto opportuno modificare la struttura del controllore, inserendo le caratteristiche di rigidezza in deriva degli assali proprie del modello di veicolo utilizzato. Nella configurazione originaria del controllore, i parametri di rigidezza in deriva richiesti come input sono costanti. Tuttavia, per migliorare le prestazioni del modello ad elevate accelerazioni laterali, sono state inserite le caratteristiche di ciascun assale mediante *look-up table*. Nello specifico le *look-up table* ricevono in input il valore di accelerazione laterale effettivo del veicolo a_y , restituendo gli attuali valori di $C_{\alpha F}$ e $C_{\alpha R}$.

Questo non modifica la tipologia di modello integrato nel controllore; in quanto, tutte le previsioni vengono effettuate utilizzando i valori di rigidezza in deriva forniti dalle *look-up table* anche per gli istanti futuri. Di conseguenza, ciò che nel concreto introduce questa modifica è un ulteriore adattamento dei parametri del modello integrato alle differenti condizioni del veicolo nei vari step della simulazione. Esaminando la lista dei parametri che descrivono il modello *state-space*, si può notare che essi sono costanti ad eccezione della velocità longitudinale e dei valori di rigidezza in deriva, nel caso di campi di funzionamento con spiccate non linearità. Con la soluzione proposta si introduce nel modello integrato nel controllore una sensibilità alla risposta di tipo non lineare degli pneumatici. Le caratteristiche introdotte sono riportate in figura 3.37.

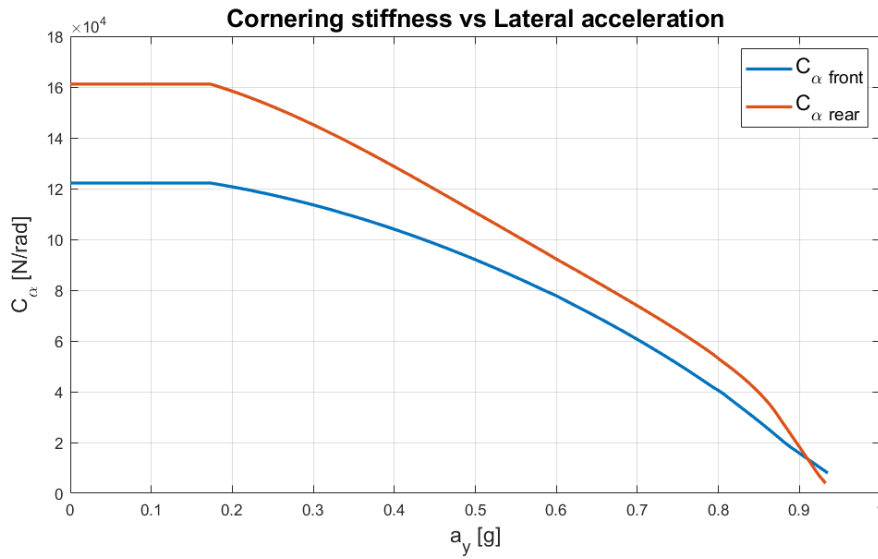


Figura 3.37: Andamento rigidezze in deriva degli assali.

In seguito all'implementazione di tale modifica sono state eseguite delle simulazioni con il fine di testarne gli effettivi benefici. I risultati ottenuti sono stati riportati all'interno del capitolo quarto, nello specifico nel paragrafo 4.7, in quanto ritenuto più opportuno presentare tali dati non prima di aver introdotto le tecniche di simulazione e di interpretazione dei risultati ottenuti.

Limiti del controllore

La modifica introdotta circa le rigidzze in deriva $C_{\alpha,F/R}$ risulta avere alcune limitazioni sul campo di funzionamento poiché la formula utilizzata per il calcolo delle forze laterali nel descrivere lo spazio degli stati del modello a bicicletta è la seguente:

$$F_{yR} = C_{\alpha_R} \alpha_R \quad (3.43)$$

Nel caso di funzionamento in campo fortemente non lineare essa restituisce un valore di F_y più elevato di quello reale. Ciò accade poiché in tale campo di funzionamento l'equazione della forza laterale non assume più la forma di una retta passante per l'origine, ma è non lineare. La sua linearizzazione nell'intorno di un punto di equilibrio che non sia l'origine ($\alpha = 0$) sarà una retta non passante per l'origine:

$$F_{yR} = C_{\alpha_R,0} (\alpha_R - \alpha_{R,0}) + F_{yR,0} \quad (3.44)$$

Di conseguenza è possibile affermare che questa criticità rappresenta un limite dell'attuale modello dal quale poter ripartire nel caso di future implementazioni.

Sebbene le modifiche introdotte abbiano esteso il campo del corretto funzionamento del controllore MPC, quando le caratteristiche del veicolo si scostano eccessivamente dalle relative approssimazioni lineari gli output del controllore non sono sufficientemente accurati. Ciò è dovuto all'esacerbazione delle differenze tra modello integrato nel controllore e modello di veicolo, nei suddetti campi di funzionamento più estremi. Per dare un riscontro pratico di quanto appena detto, è stata condotta una simulazione fornendo una traiettoria ideale che richiede elevate performance del controllore, caratterizzata da accelerazioni laterali fino a $1g$.

Lo scenario adottato è una strada rettilinea nella quale transitano un veicolo *ego* ed un veicolo *lead*. La manovra eseguita dal veicolo *ego* è un doppio cambio di corsia, analogo a quello svolto nell'esempio introdotto nel capitolo precedente. Si riporta in figura 3.38 un confronto tra la traiettoria di riferimento e la traiettoria effettiva.

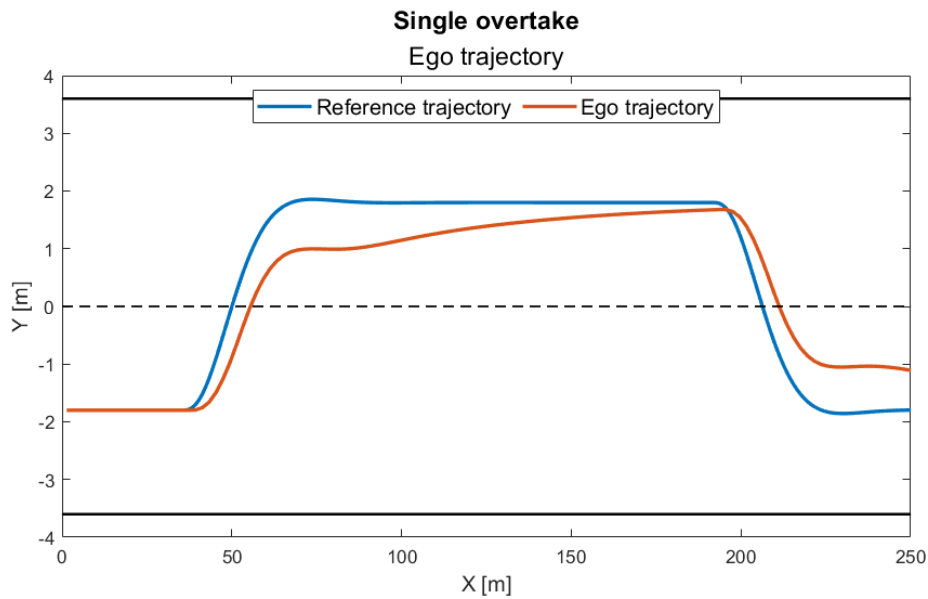


Figura 3.38: Confronto traiettoria effettiva/riferimento ad elevate performance richieste.

Come si può vedere dal confronto delle traiettorie e dall'andamento degli errori di posizionamento ed orientazione in figura 3.40, in seguito ad una richiesta di accelerazione laterale tanto elevata, il veicolo non riesce a seguire in maniera precisa la traiettoria di riferimento. L'errore di orientazione viene compensato rapidamente in seguito all'attuazione del comando di sterzo, ben prima che il veicolo *ego* raggiunga il centro della corsia di sorpasso. Ciò provoca un errore di distanza laterale residuo che viene compensato con transitori eccessivamente lunghi. Tale comportamento viene esacerbato quando la richiesta in termini di performance è ancora maggiore. Deve necessariamente essere sottolineato che nonostante il veicolo *ego* fatichi a seguire la traiettoria di riferimento non incorre in instabilità direzionale, dunque, gli stati del sistema non divergono. Ciò è dovuto all'azione di controllo effettuata, la quale, sebbene in anello aperto la richiesta di performance porti alla teorica instabilità, rende il veicolo *ego* stabile grazie all'architettura ad anello chiuso.

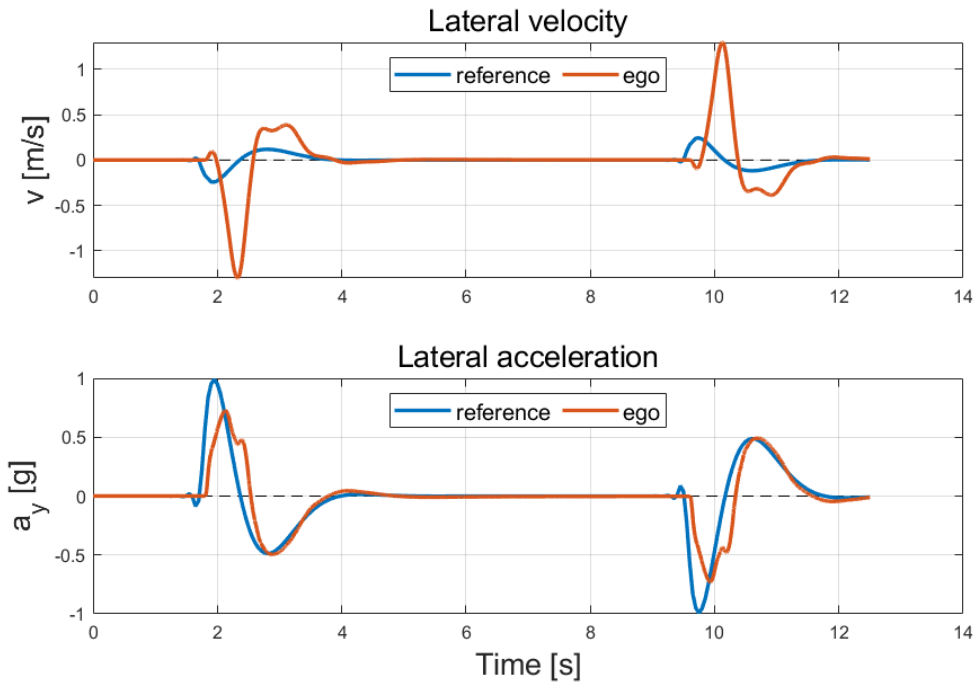


Figura 3.39: Grandezze veicolo *ego* per traiettoria ad elevate performance richieste.

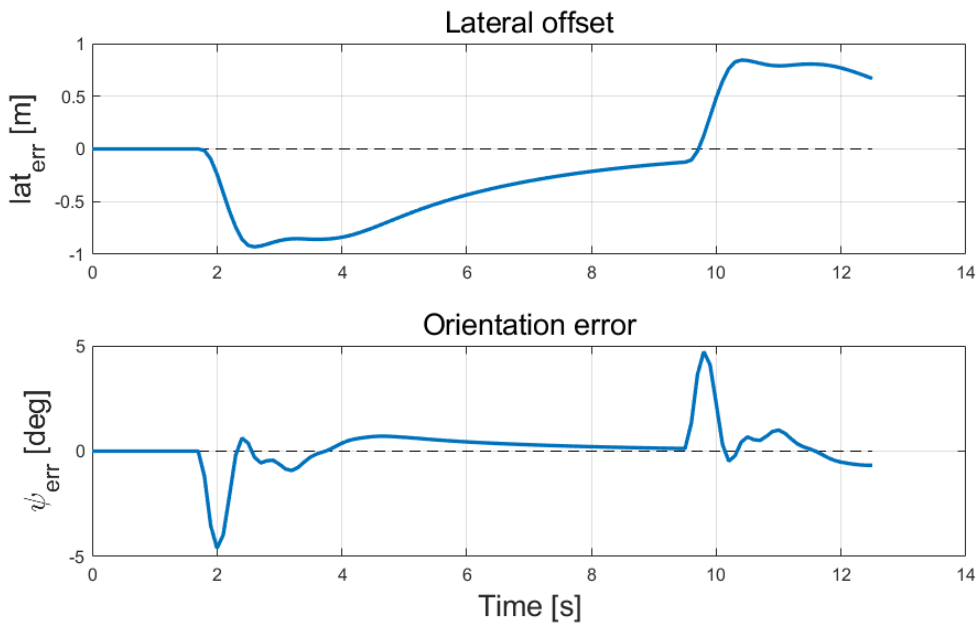


Figura 3.40: Errori controllo traiettoria ad elevate performance richieste.

3.2.3 Modello di veicolo

All'interno del sottosistema *Vehicle dynamics* è stato integrato il modello di veicolo adottato per simulare il comportamento del veicolo *ego* in relazione agli input ricevuti dal controllore. Esso è stato sviluppato dal gruppo di ricerca di dinamica del veicolo del Politecnico di Torino ed è un modello a 4 gradi di libertà, di tipo *Doppia traccia* con rollio e modello di pneumatici non lineare. Si riporta la struttura generale del sottosistema in figura 3.41.

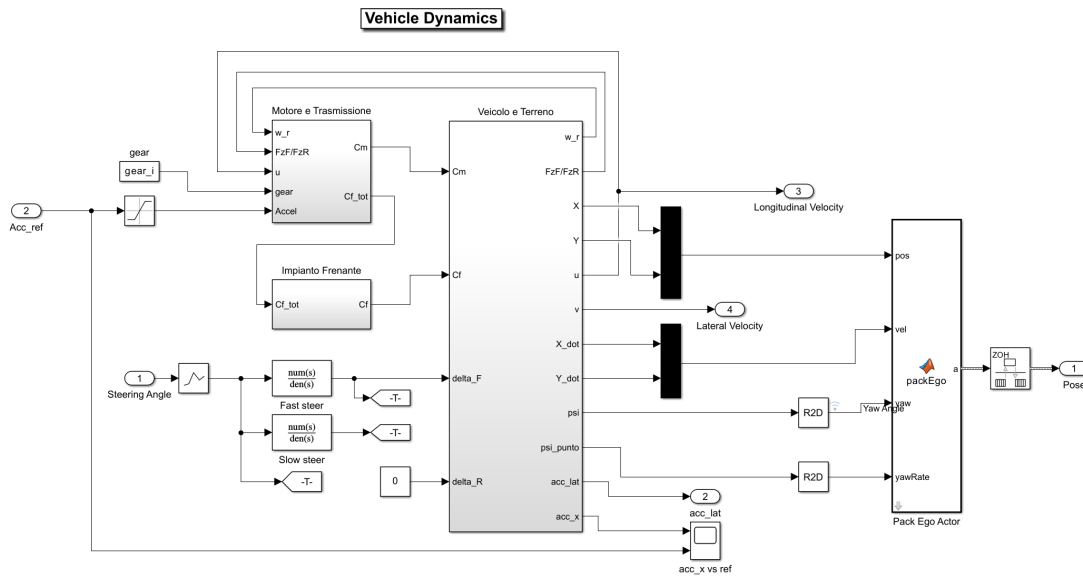


Figura 3.41: *Vehicle Dynamics* subsystem.

I sottosistemi *Motore e Trasmissione* ed *Impianto Frenante* si occupano di tradurre il segnale di accelerazione ricevuto dal controllore in coppia motrice C_m e coppia frenante C_f . Essi hanno subito modifiche massive rispetto alla versione originale per permettere l'accoppiamento del modello di veicolo con il modello complessivo, le quali sono state espone in seguito. Il blocco *Pack Ego Actor* ha la stessa funzionalità svolta dal sottosistema *Pack Actor*, analizzato nel paragrafo 3.1.5. Esso si occupa di convogliare i segnali contenenti le grandezze descrittive del moto del veicolo *ego*, nella struttura *EgoActor*.

Il sottosistema *Veicolo e Terreno* è dedicato al calcolo di tutte le grandezze descrittive del moto del veicolo, quali: posizione, velocità espressa con sistema di riferimento veicolo, velocità espressa con sistema di riferimento globale e accelerazione. In figura 3.42 è stata riportata la struttura generale.

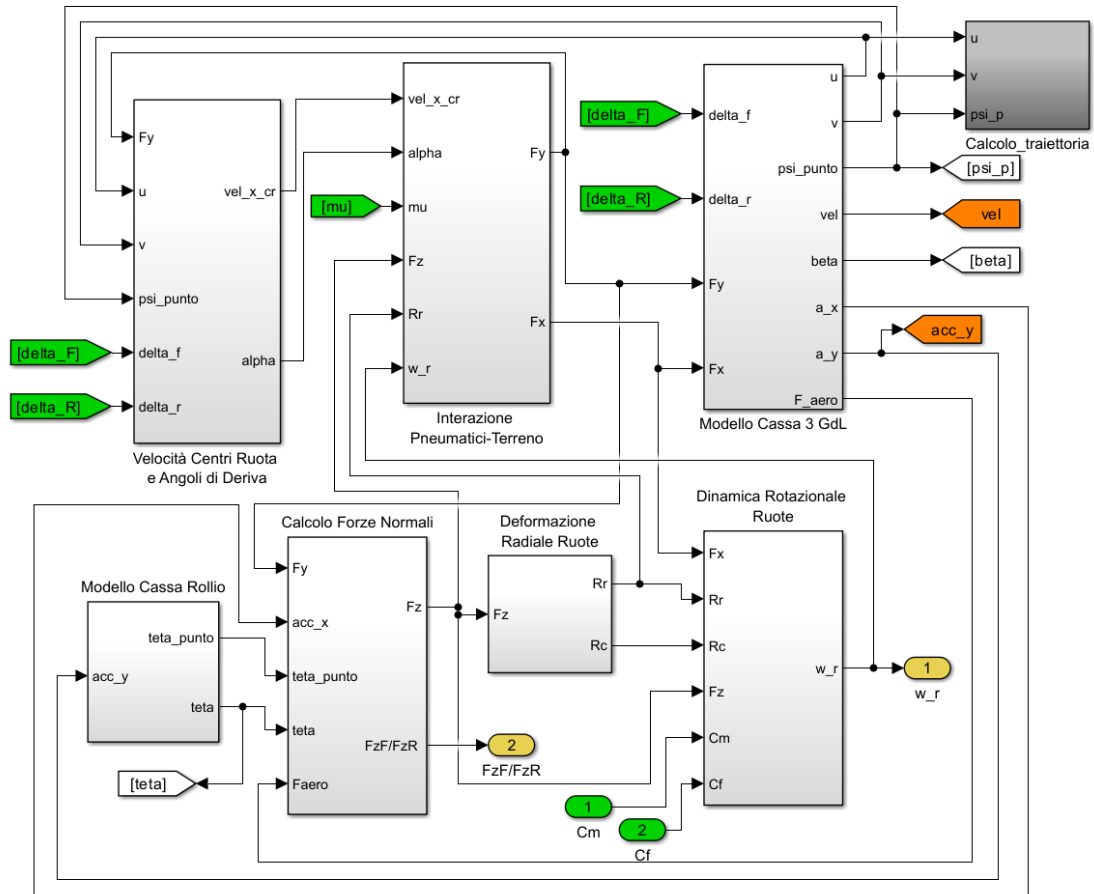


Figura 3.42: Modello di veicolo.

Nell'analisi proposta di seguito non sono state analizzate tutte le equazioni utilizzate per il calcolo delle singole grandezze; tuttavia, è stata proposta una disamina sulle ipotesi introdotte e repute di maggior rilievo per la comprensione del grado di accuratezza del modello. In figura 3.43 e 3.44 sono stati riportati i diagrammi di corpo libero del veicolo, realizzati rispettivamente nei piani X-Y e Y-Z.

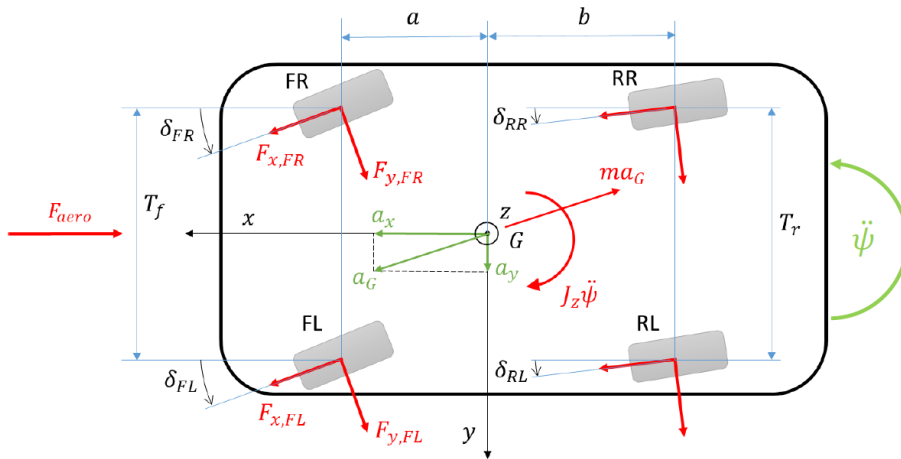


Figura 3.43: Diagramma di corpo libero piano X-Y.

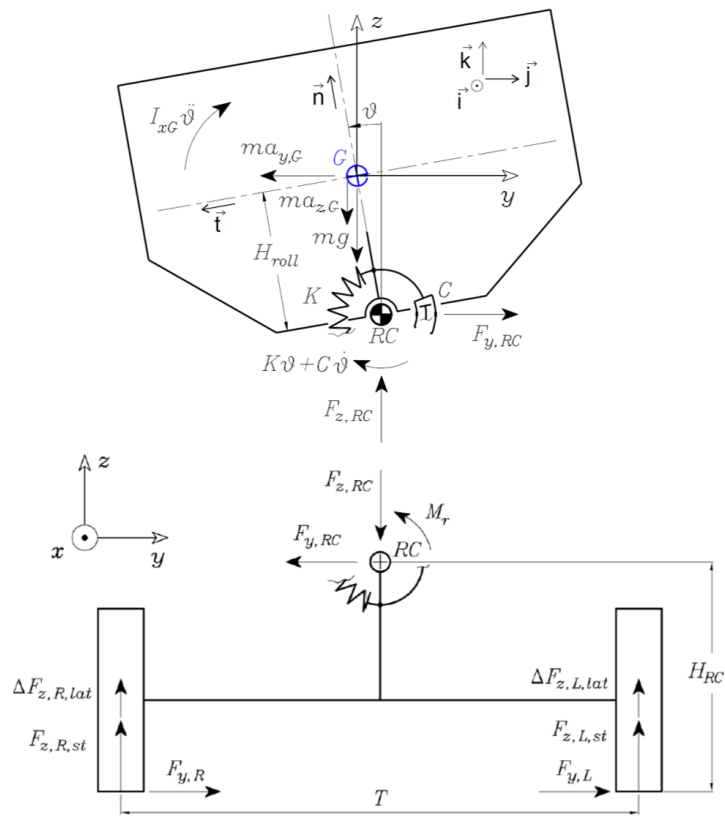


Figura 3.44: Diagrammi di corpo libero piano Y-Z.

Modello Cassa 3 Gdl: Le principali grandezze fornite in output, quali posizione, velocità e accelerazione, vengono calcolate attraverso le equazioni che descrivono il moto di un veicolo dotato di quattro punti di contatto con il terreno e che si sposta lungo il piano orizzontale. All'interno del modello è stata integrata la possibilità di avere un angolo di sterzo anche per le ruote dell'assale posteriore; tuttavia, è stato liberamente scelto di limitare la possibilità di sterzare esclusivamente alle ruote dell'assale anteriore.

$$\delta_R = 0 \quad (3.45)$$

Le grandezze citate vengono calcolate mediante la combinazione delle equazioni di equilibrio alla traslazione e alla rotazione. Infine, viene fornito in output il valore dell'angolo di assetto valutato nel baricentro.

$$\beta = \arctan \frac{u}{v} \quad (3.46)$$

Dove u e v rappresentano le velocità rispettivamente longitudinale e laterale del veicolo, cioè le componenti della velocità del baricentro del veicolo calcolate rispetto ad un sistema di riferimento non inerziale.

Velocità Centri Ruota e Angoli di Deriva: Conoscendo la velocità longitudinale u e laterale v , calcolate nel blocco precedente, è possibile calcolare il loro valore nei singoli centri ruota. Ottenendo l'angolo di assetto β valutato nei centri ruota si fornisce come output il relativo angolo di deriva α_i .

$$\alpha_i = \beta_i - \delta_i \quad (3.47)$$

Interazione Pneumatici - Terreno: Una delle caratteristiche principali del modello è quella di avere integrato un modello non lineare di pneumatico. Nello specifico, viene adottato il modello *Magic Formula* di Pacejka. All'interno di un apposito blocco *MatLab function* sono contenuti tutti i coefficienti della formula di Pacejka, la quale permette di calcolare le forze laterali, longitudinali ed il momento di autoallineamento che agiscono sulle singole ruote. Il vantaggio nell'utilizzo di questo modello è una maggiore accuratezza dei risultati per angoli di deriva più

elevati, per i quali non sarebbe corretto considerare un legame di tipo lineare tra la forza laterale F_y e l'angolo di deriva α .

Inoltre, per rendere maggiormente realistico il modello di pneumatico sono stati introdotti dei ritardi di risposta sulle forze longitudinali e laterali. Le forze effettive vengono calcolate con la seguente equazione differenziale:

$$\frac{1}{\tau} \dot{F}_{rit} + F_{rit} = F_{pac} \quad (3.48)$$

Dove la costante temporale τ è il rapporto tra la velocità longitudinale del centro ruota e la relativa lunghezza di rilassamento (longitudinale o trasversale).

Modello Cassa Rollio e Calcolo Forze Normali: Le ipotesi semplificative introdotte nella modello di rollio della massa sospesa sono le seguenti:

- Rigidezza a rollio di ciascuna sospensione: contributo molle e barre stabilizzatrici.

$$K_{sosp} = K_{spring-roll} + K_{ARB} \quad (3.49)$$

- Smorzamento a rollio di ciascuna sospensione: contributo ammortizzatori.
- Rigidezze e smorzamenti a rollio delle sospensioni anteriore e posteriore tali per cui:

$$\begin{aligned} K &= K_{sosp F} + K_{sosp R} \\ C &= C_{damp F} + C_{damp R} \end{aligned} \quad (3.50)$$

- Asse di rollio fisso.

Inoltre, si definiscono le seguenti grandezze:

- Percentuale rigidezza a rollio barre su rigidezza a rollio totale.

$$\%ARB = 100 \frac{k_{ARB}}{k_{sosp}} \quad (3.51)$$

- Percentuale rigidezza a rollio barra anteriore su rigidezza barre totale.

$$\%ARB_{front} = 100 \frac{k_{ARB,R}}{k_{ARB,tot}} \quad (3.52)$$

Grazie alle equazioni introdotte in questi due blocchi è possibile ricavare le forze verticali agenti sulle singole ruote e su ciascun assale.

Dinamica Rotazionale Ruote: Quest'ultimo blocco si occupa di calcolare la velocità angolare delle singole ruote, il cui raggio è stato correttamente ricavato in seguito all'introduzione dell'ipotesi di deformazione radiale dello pneumatico.

Motore e Trasmissione ed Impianto frenante: Nel modello originario questi due blocchi si occupano di tradurre in coppia frenante e motrice i comandi provenienti rispettivamente da pedale del freno e dell'acceleratore; i quali sono due dei tre input richiesti dal modello. La *driveline* del modello è stata costruita in maniera tale da poter modificare la modalità di trazione del veicolo (FWD, RWD, AWD) ed è stato inserito un differenziale di tipo *open*. L'impianto frenante è stato costruito in modo da ripartire equamente tra le quattro ruote l'azione frenante.

Come anticipato, questi due blocchi sono stati modificati in maniera massiva in seguito all'inserimento del suddetto modello di veicolo nel modello complessivo. Ciò è stato necessario per adattare l'output del controllore, l'accelerazione longitudinale del veicolo, agli input del modello esaminato, quali la percentuale di attuazione dei pedali di freno ed acceleratore.

La soluzione individuata come ottimale è stata quella di eliminare la modellizzazione del motore e dell'impianto frenante, calcolando la coppia necessaria per il dato valore di accelerazione in input.

$$C = R_0 (m a_x + F_{aero} + F_r) \quad (3.53)$$

Con

- C. Coppia necessaria al raggiungimento della data accelerazione a_x .
- m. Massa del veicolo.

- a_x . Accelerazione longitudinale in output dal controllore.
- F_{aero} . Forza che esprime la resistenza aerodinamica.
- F_r . Forza che esprime la resistenza al rotolamento
- R_0 . Raggio ruota indeformato.

Qualora la coppia calcolata fosse positiva ciò implicherebbe una richiesta di coppia al motore. Per cui si procede calcolando la coppia al motore C_m , conoscendo il rapporto finale dell'intera trasmissione. Inoltre, mediante due *look-up table*, vengono calcolati i valori di coppia minima e massima erogabili dal motore al dato numero di giri; i quali sono i valori di saturazione del parametro C_m calcolato.

Nel caso in cui il valore di coppia fosse negativo, questa viene equamente suddivisa nelle singole ruote come coppia frenante C_f .

Sebbene questa sia una semplificazione importante rispetto al modello originario, l'introduzione di un sistema di attuazione più complesso non sarebbe affine con le prestazioni richieste al modello complessivo nella fase di sviluppo del sistema assistenziale. In quanto, generare dei ritardi sull'attuazione dei comandi porterebbe il modello ad una maggior vicinanza rispetto alle condizioni reali del veicolo; tuttavia, nella fase di sviluppo attuale, è preferibile non introdurre ulteriori alterazioni ai segnali, con il fine di comprendere al meglio gli effetti delle variazioni apportate alle aree tematiche dedicate al pianificatore, al controllore ed al modello di veicolo.

Attuazione angolo di sterzo: L'ultimo input del modello è l'angolo di sterzo alle ruote δ_f . Questo viene processato da una funzione di trasferimento, la quale simula il ritardo del sistema di attuazione dell'angolo di sterzo. Tuttavia, analogamente a quanto espresso circa l'attuazione dell'accelerazione, è stato preferito non introdurre ritardi tali da alterare considerevolmente i segnali di output del modello. La funzione di trasferimento introdotta è la seguente.

$$\frac{\delta_f}{\delta_{f \text{ controll}}} = \frac{1}{\tau s + 1} \quad (3.54)$$

Sono stati assunti due differenti valori di costante temporale τ per valutare il corretto ritardo di attuazione, uno corrispondente ad un sistema di attuazione rapido (*fast*) e l'altro ad un sistema più lento (*slow*). In seguito ad opportuni test effettuati mediante apposite simulazioni è stato scelto il sistema di attuazione rapido, tale che $\tau = 0.01$ s. I risultati delle simulazioni citate sono stati riportati nel paragrafo 4.7.

3.2.4 Parametri di configurazione veicolo

Il veicolo utilizzato per la simulazione è dotato di sistema di trazione posteriore (RWD) ed è di tipo sovrasterzante. Le principali specifiche tecniche sono riportate nella tabella 3.4.

Parameter	Value	Unit
m	1600	kg
Jx	600	$kg\ m^2$
Jz	2860	$kg\ m^2$
a	1.56	m
b	1.04	m
Hg	0.65	m
$H_{RC\ front}$	$4\ 10^{-2}$	m
$H_{RC\ rear}$	$10\ 10^{-2}$	m
K	$1.3\ 10^5$	$N\ m/rad$
% ARB	70%	
% ARB_{front}	30 %	
R_S	13	
TBR	1	
R_0	$3.28\ 10^{-1}$	m
toe_F	-0.1	deg
toe_R	0.1	deg

Tabella 3.4: Parametri di configurazione veicolo.

Dove:

- m. Massa totale del veicolo.
- Jx. Momento d'inerzia a rollio.

- J_z . Momento d'inerzia calcolato rispetto all'asse verticale che passa per il baricentro del veicolo.
- a. Semipasso anteriore.
- b. Semipasso posteriore.
- H_g . Altezza del baricentro dal suolo.
- $H_{RC\ front}$. Altezza del centro di rollio anteriore dal suolo.
- $H_{RC\ rear}$. Altezza del centro di rollio posteriore dal suolo.
- K . Rigidezza a rollio totale.
- % ARB. Ripartizione rigidezza a rollio barre su rigidezza totale.
- % ARB_{front} . Rapporto rigidezza a rollio barra anteriore sul rigidezza barre totale.
- R_S . Rapporto di sterzo.
- TBR. Torque Bias Ratio: Rapporto tra la coppia massima e la coppia minima del differenziale tra le ruote.
- R_0 . Raggio indeformato.
- toe_F . Angolo di convergenza ruote anteriori.
- toe_R . Angolo di convergenza ruote posteriori.

Dai parametri inseriti si può notare che in questa fase è stato preferito un differenziale *open* di tipo ideale. Inoltre, la ripartizione della rigidezza delle barre tra anteriore e posteriore è tale da aumentare il trasferimento di carico al posteriore, riducendo la rigidezza in deriva dell'assale. Il trasferimento di carico al posteriore viene incrementato anche dall'aumento di altezza del centro di rollio posteriore. Queste due condizioni portano il veicolo ad avere un comportamento più sovrasterzante.

3.2.5 Caratterizzazione del modello di veicolo

Caratterizzazione statica

Per ricavare tutte le caratteristiche proprie del veicolo utilizzato è stata effettuata in primo luogo una simulazione mediante modello di veicolo isolato. La manovra effettuata è una *ramp steer*, le cui caratteristiche sono riportate nella tabella 3.5.

Parameter	Value	Unit
V_{start}	50	km/h
gear	3	
δ_{swgain}	15	deg/s

Tabella 3.5: Parametri di configurazione manovra *ramp steer*.

Si riportano nelle figure 3.45 e 3.46, gli andamenti degli angoli di deriva α e delle forze laterali F_y agenti sulle singole ruote e sugli assali, calcolati rispetto al tempo.

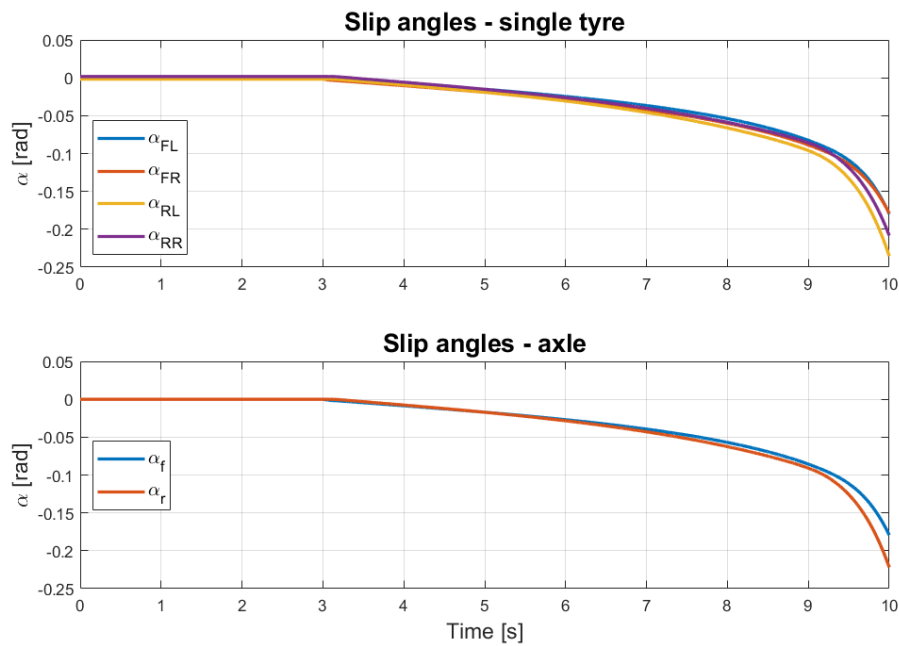


Figura 3.45: Andamento angoli di deriva α manovra *ramp steer*.

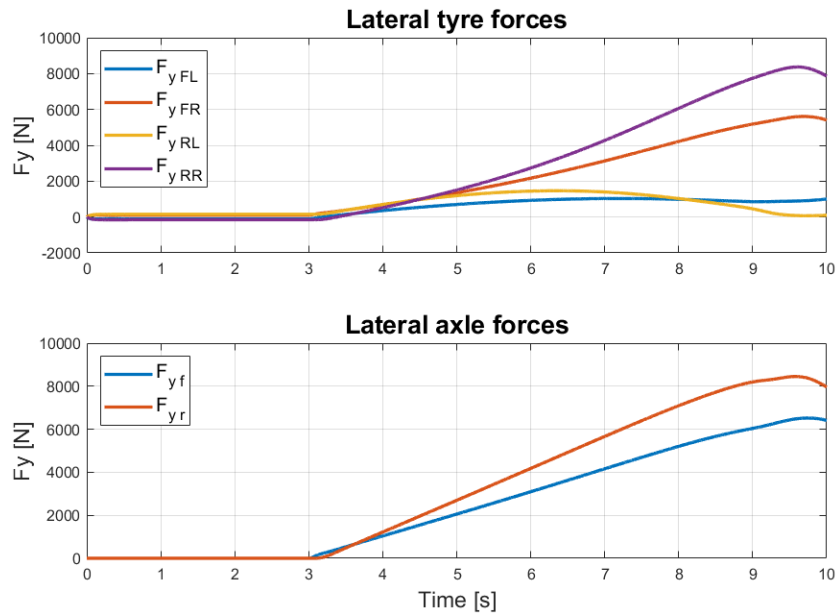


Figura 3.46: Andamento forze laterali F_y manovra *ramp steer*.

Gli andamenti riportati sono stati utilizzati per ricavare l'andamento delle forze laterali F_y al variare dell'angolo di deriva, riportato in figura 3.47.

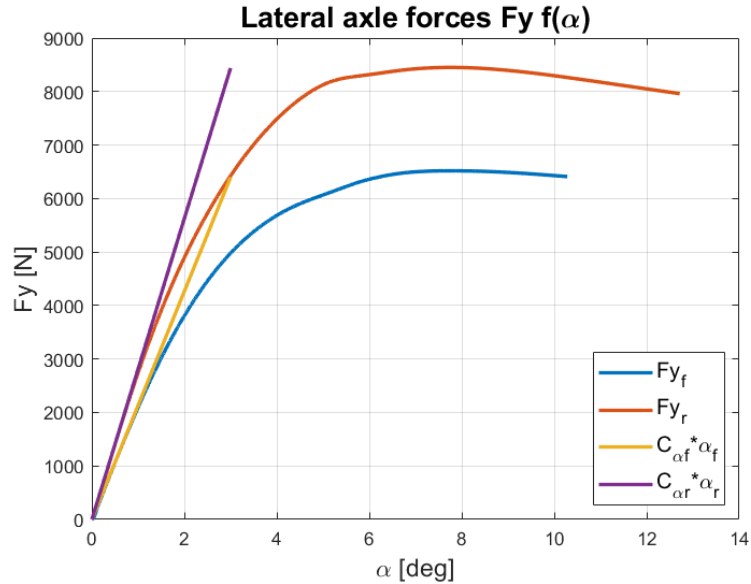


Figura 3.47: Andamento forze laterali F_y in funzione degli angoli di deriva α manovra *ramp steer*.

Si noti che le rette, le cui equazioni sono riportate in legenda, approssimano in maniera ottimale l'andamento delle curve in campo lineare. Si definisce "rigidezza in deriva" la pendenza della retta che approssima la caratteristica $Fy(\alpha)$ nell'origine. Tale per cui, nel caso di piccoli angoli di deriva (campo di funzionamento lineare), vale la seguente equazione.

$$Fy_{F/R}(\alpha) = C_{\alpha,F/R} \cdot \alpha_{F/R} \quad (3.55)$$

I valori di rigidezza in deriva corrispondenti sono stati utilizzati per calcolare il gradiente di sottosterzo, valutato al volante. Essendo il risultato minore di zero, il veicolo ha un comportamento tendenzialmente sovrasterzante.

$$K_{us} = \frac{m}{l} \left(\frac{b}{C_F} - \frac{a}{C_R} \right) \cdot R_s = -0.93 \cdot 10^{-2} \text{ rad}/(m/s^2) \quad (3.56)$$

Il gradiente di assetto è pari a:

$$K_{\beta} = -\frac{m a}{l C_R} = -0.6 \cdot 10^{-2} \text{ rad}/(m/s^2) \quad (3.57)$$

A partire dal valore di K_{us} è stato calcolato il valore di velocità critica in campo lineare.

$$V_{cr} = \sqrt{-\frac{l}{\frac{m}{l} \left(\frac{b}{C_F} - \frac{a}{C_R} \right)}} = 216.62 \text{ km/h} \quad (3.58)$$

La velocità critica individua la soglia di instabilità direzionale del veicolo. Per velocità superiori ad essa il veicolo presenta la condizione di equilibrio instabile per la quale è necessario il così detto "controsterzo", per cui, per mantenere il veicolo su una traiettoria a curvatura costante, deve essere fornito un angolo volante di segno opposto a quello cinematicamente necessario.

Processando questi dati è poi stata ricavata la caratteristica di rigidità in deriva C_α di ciascun assale al variare dell'angolo di deriva α e dell'accelerazione laterale a_y , i cui andamenti sono riportati in figura 3.48.

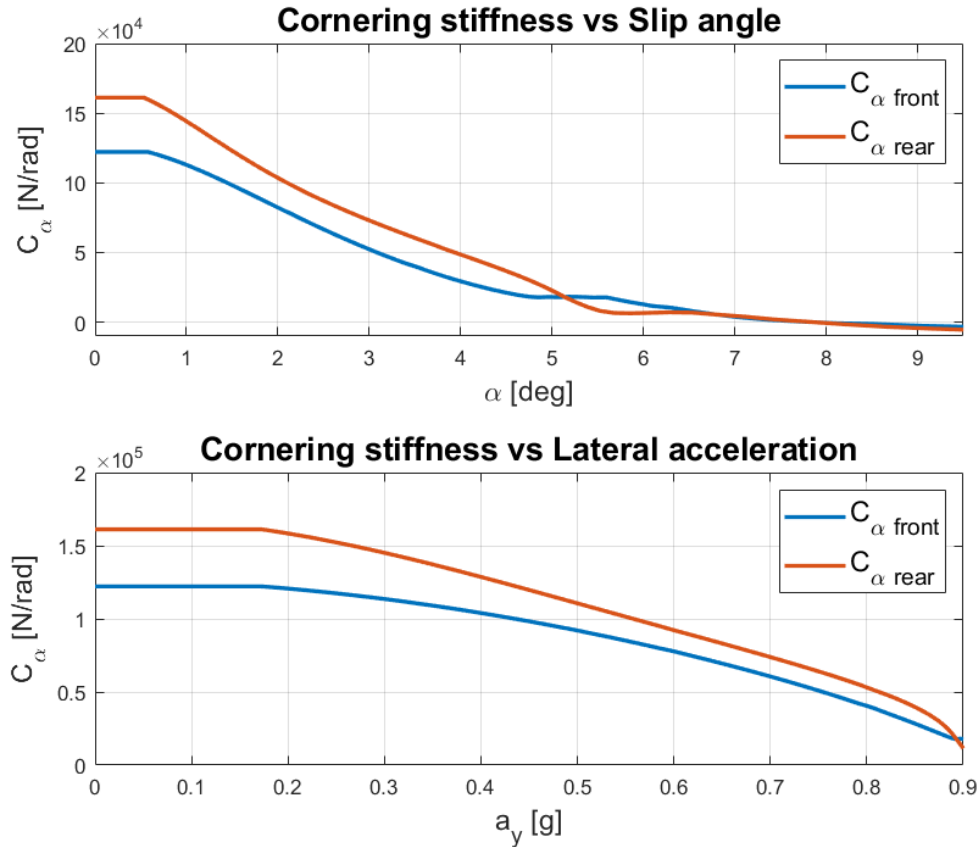


Figura 3.48: Rigidezza in deriva manovra *ramp steer*.

Si può notare che le caratteristiche hanno un tratto inizialmente rettilineo. Questo è dovuto ad un'alterazione dei segnali effettuata manualmente durante la fase di *post processing*; la quale è risultata necessaria per eliminare tutti i disturbi dei segnali presenti a piccoli angoli di deriva e piccole accelerazioni laterali. I Valori di saturazione adottati sono pari alle pendenze delle rette riportate in figura 3.47.

Nella figura 3.49 sono state riportate le caratteristiche di sottosterzo e di assetto. Esse descrivono la differenza degli angoli di sterzo e di assetto, δ_{vol} e β , calcolata rispetto al loro corrispettivo cinematico, $\delta_{vol,0}$ e β_0 , al variare dell'accelerazione laterale a_y .

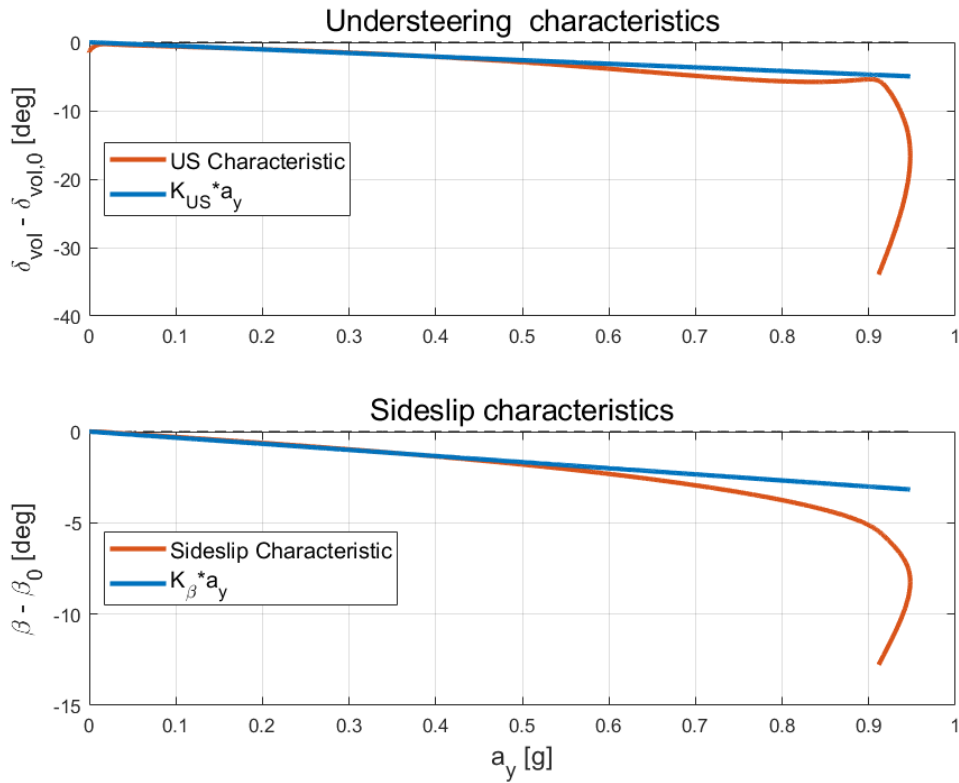


Figura 3.49: Caratteristiche di sovrasterzo e di assetto veicolo.

Per ognuna delle curve è stato riportato il prodotto tra il relativo gradiente e l'accelerazione laterale. Come ci si può aspettare, per valori di a_y bassi l'andamento di questo prodotto è sovrapponibile all'andamento della relativa caratteristica.

Per completezza si riportano nelle successive figure 3.50 e 3.51 gli andamenti delle forze longitudinali e verticali agenti sulle singole ruote e lo scorrimento di ciascun pneumatico.

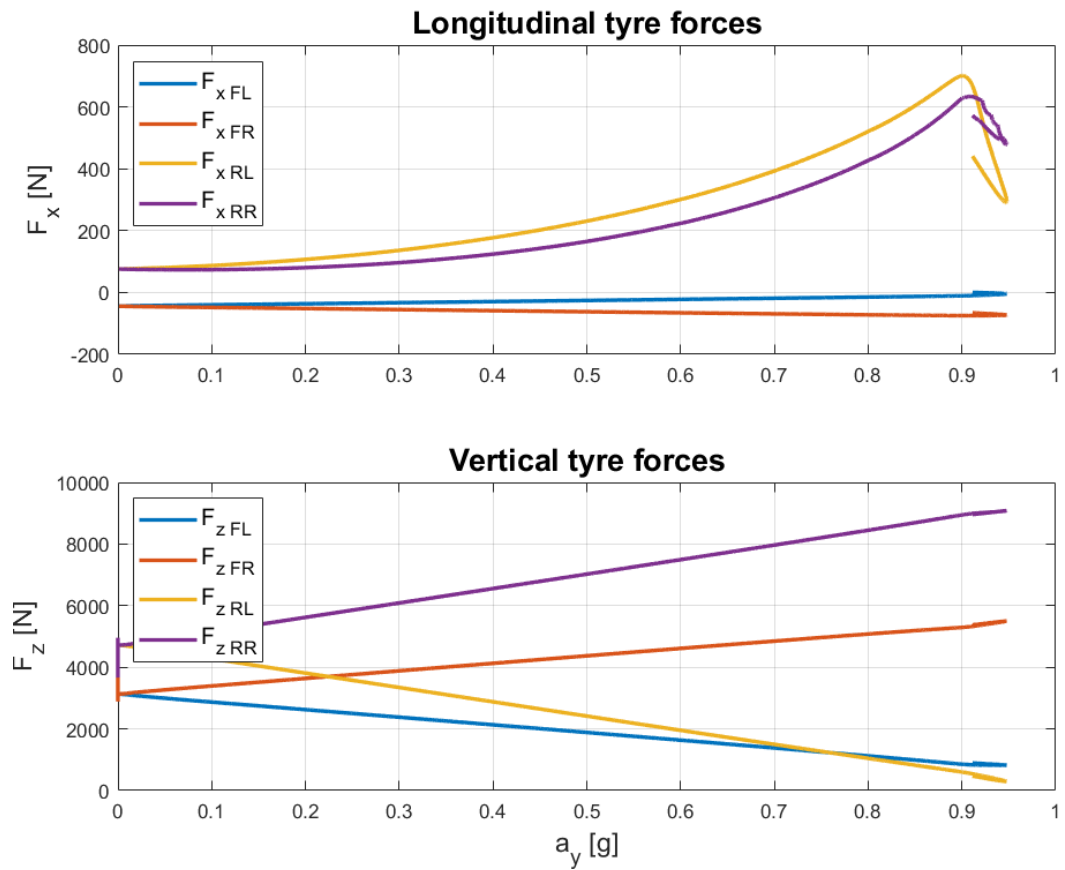


Figura 3.50: Forze longitudinali e verticali manovra *ramp steer*.

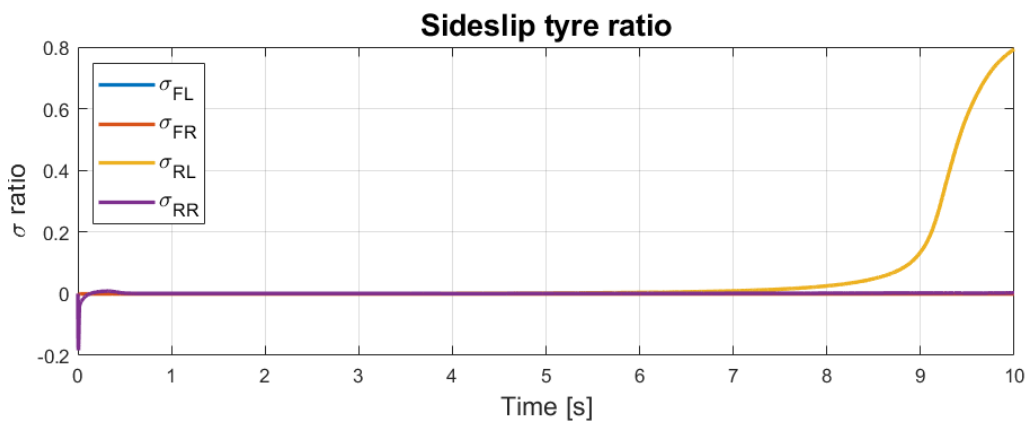


Figura 3.51: Scorrimenti longitudinali pneumatici manovra *ramp steer*

Nella seguente figura 3.52 sono stati riportati gli andamenti della velocità e dell'accelerazione laterale nel corso del tempo.

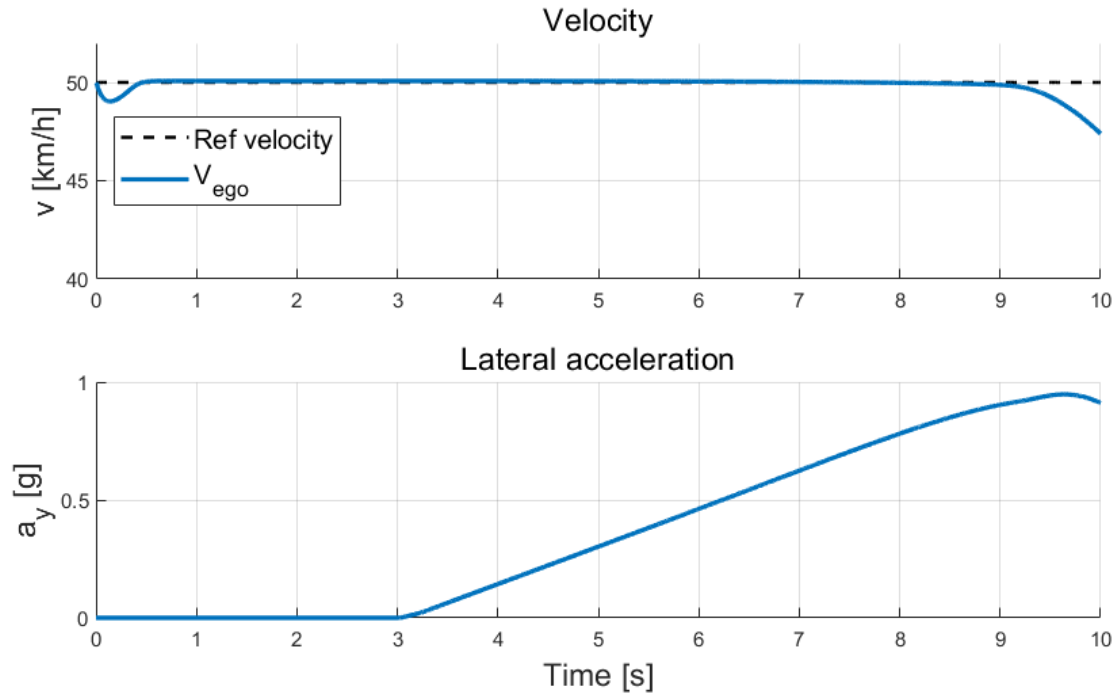


Figura 3.52: Velocità e accelerazione laterale veicolo manovra *ramp steer*.

Si nota che negli istanti finali della prova il veicolo perde aderenza nella ruota posteriore sinistra (interno curva). Ciò porta ad un conseguente decremento di velocità ed accelerazione laterale, oltre che ad un angolo volante δ_{vol} elevato rispetto al corrispettivo cinematico.

Caratterizzazione in transitorio del veicolo

Per ottenere un quadro completo delle caratteristiche del veicolo è stata successivamente effettuata una seconda simulazione con il modello veicolo isolato. La manovra utilizzata è una *step steer* mediante la quale, nell'istante della simulazione pari a $t = 2.0$ s, si passa istantaneamente da una condizione con angolo volante nullo ad una con angolo volante pari a $\delta_{SW} \neq 0$. Le caratteristiche della manovra sono riportate nella tabella 3.6.

Parameter	Value	Unit
V_{start}	100	km/h
gear	5	
δ_{SW}	10	deg

Tabella 3.6: Parametri di configurazione manovra *step steer*.

Si riportano nelle figure 3.53 e 3.54, gli andamenti degli angoli di deriva α e delle forze laterali F_y agenti sulle singole ruote e sugli assali, calcolati rispetto al tempo.

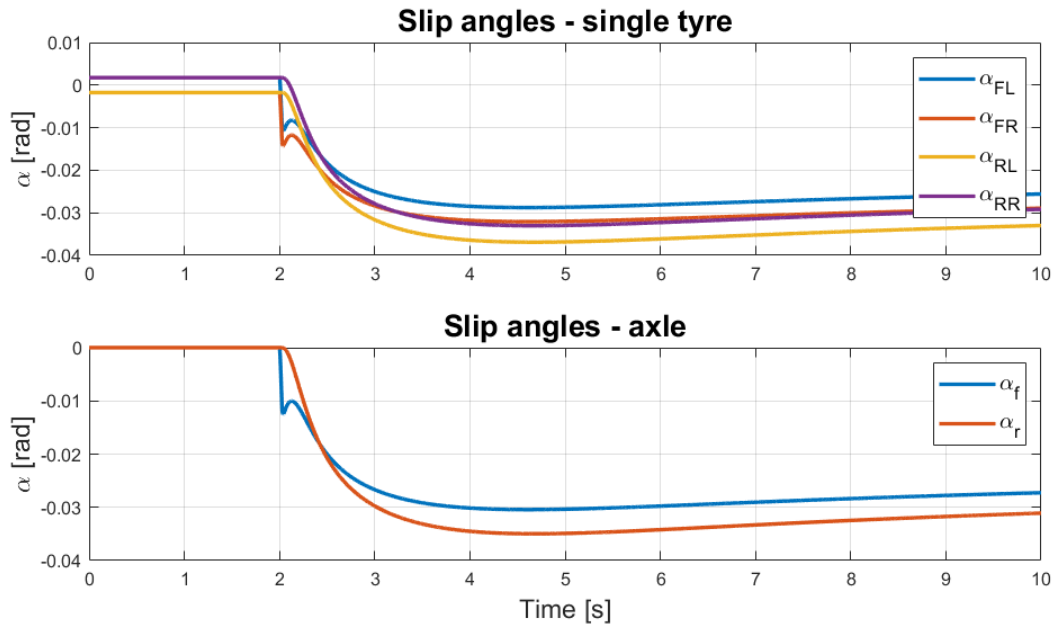


Figura 3.53: Andamento angoli di deriva α manovra *step steer*.

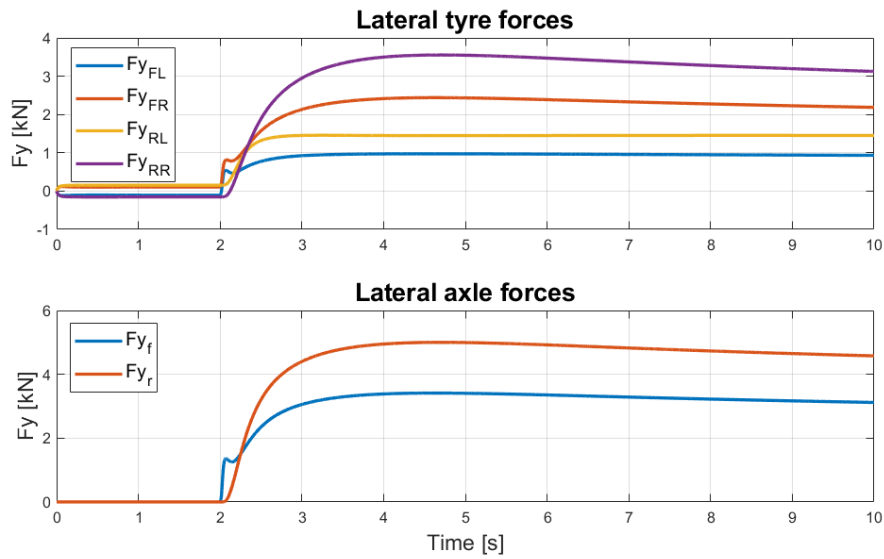


Figura 3.54: Andamento forze laterali F_y manovra *step steer*.

Si riportano in figura 3.55 gli andamenti dell'angolo di assetto e della velocità di imbardata nel corso della manovra.

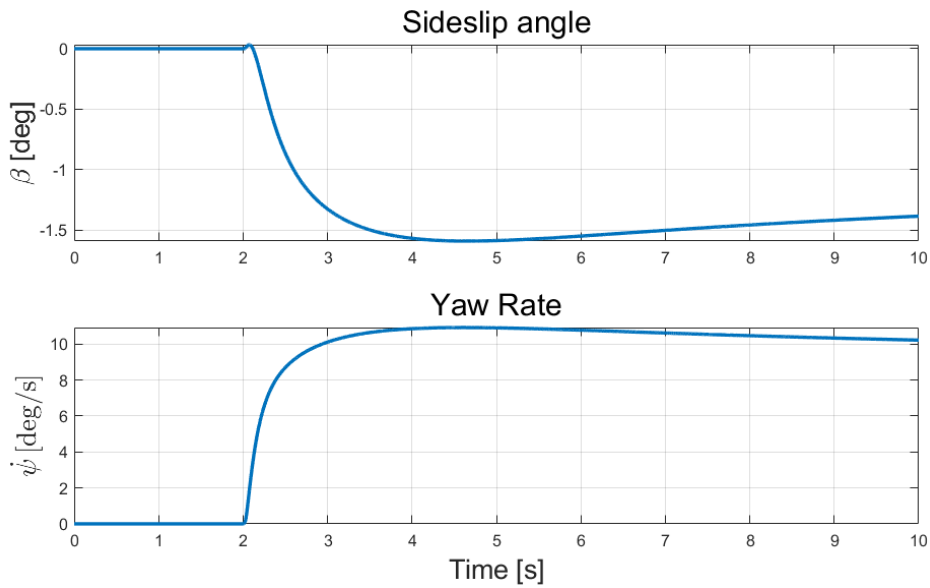


Figura 3.55: Andamento angolo di assetto e velocità di imbardata manovra *step steer*.

Risulta interessante ai fini di introdurre un'analisi di stabilità del veicolo, effettuare una seconda prova *step steer* con un angolo volante δ_{SW} maggiore. L'angolo scelto è pari a $\delta_{SW} = 15 \text{ deg}$. Si riportano nelle figure 3.56 e 3.57, gli andamenti degli angoli di deriva α e delle forze laterali F_y agenti sulle singole ruote e sugli assali, calcolati rispetto al tempo

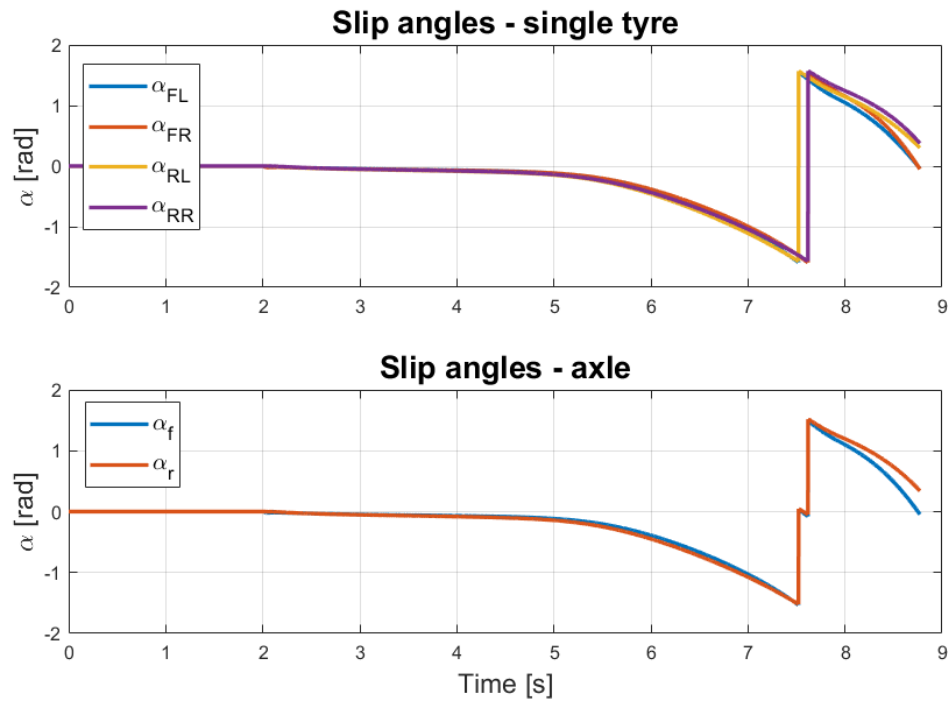


Figura 3.56: Andamento angoli di deriva α manovra *step steer*.

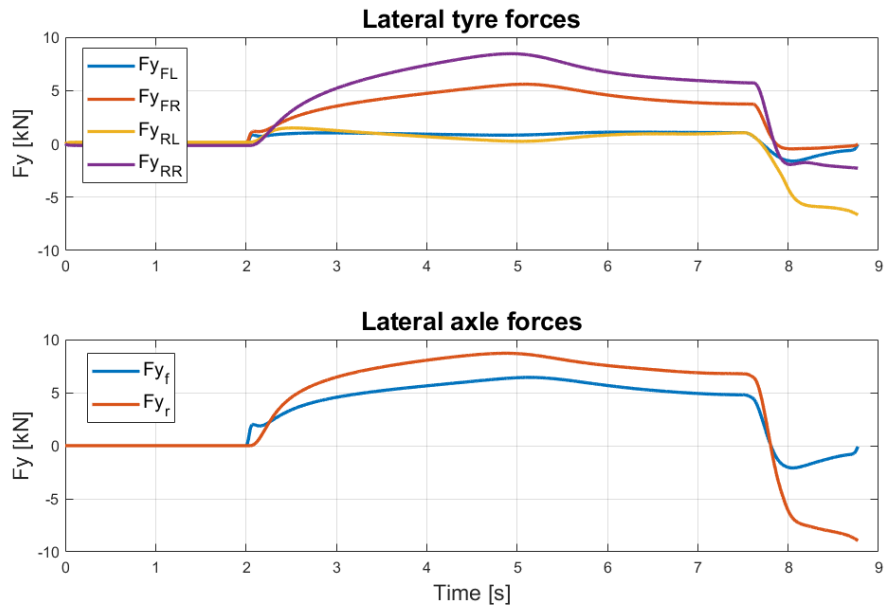


Figura 3.57: Andamento forze laterali F_y manovra *step steer*.

Si riportano in figura 3.59 gli andamenti dell'angolo di assetto e della velocità di imbardata nel corso della manovra.

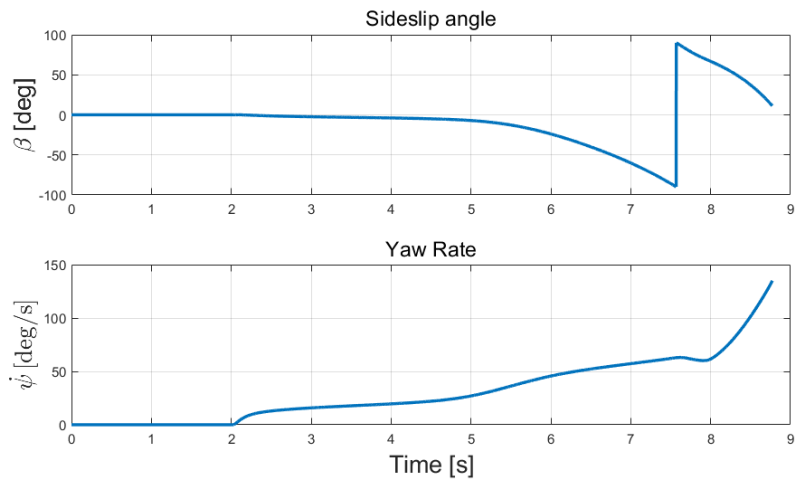


Figura 3.58: Andamento angolo di assetto e velocità di imbardata manovra *step steer*.

Si nota che la manovra con angolo volante maggiore presenta instabilità dinamica. Per entrambe le prove la velocità viene mantenuta costante e pari a $V = 100 [km/h]$

a meno di piccole oscillazioni, motivo per il quale non ne sono stati riportati gli andamenti. Ciò che si evince dal confronto tra le due prove, le quali vengono effettuate a pari velocità, è che incrementando l'angolo volante δ_{SW} si ha una netta diminuzione della velocità critica; la quale porta alla divergenza degli stati del sistema, come si vede in figura 3.59.

Studi analoghi possono essere condotti sfruttando il metodo del piano delle fasi presentato nel capitolo secondo. Utilizzando il piano delle fasi $\dot{\psi} - \beta$ è possibile valutare la posizione dei punti di equilibrio instabile al variare della velocità del veicolo. Si raggiunge la velocità critica per quel dato valore V_{cr} in cui non si ha più un punto di equilibrio stabile poiché esso si unisce con un punto di equilibrio instabile.

Richiamando i concetti teorici esposti nel capitolo precedente, la velocità del veicolo V_x è uno dei parametri che maggiormente influenzano il dominio di attrazione. All'aumentare della stessa, la zona stabile riduce progressivamente la sua area fino a scomparire. Per ottenere questa condizione sono state effettuate diverse simulazioni variando la velocità del veicolo ma mantenendo fissi i parametri di coefficiente di attrito e angolo volante. Dopodiché sono stati realizzati dei grafici nel piano $V - \beta$, riportando il valore β_i associato ai tre punti di equilibrio al variare della velocità.

Si riporta di seguito un esempio della metodologia citata. Si specifica che il veicolo utilizzato per ricavare le mappe ed i valori riportati ha parametri e caratteristiche differenti da quelli presentati in precedenza. La prova è stata condotta ad angolo volante costante e pari a $\delta_{SW} = 5 \text{ deg}$, da essa è stata ricavato il grafico riportato in figura 3.59. Dall'analisi del grafico risulta chiara la velocità alla quale si ha l'unione del punto di equilibrio stabile con uno dei due punti di equilibrio instabile; tale valore viene quindi associato alla velocità critica.

Un altro parametro di notevole influenza sulla posizione dei punti di equilibrio è l'angolo volante δ_{SW} , il quale deve essere mantenuto costante per ricavare le traiettorie che compongono la mappa all'interno del piano delle fasi. Tuttavia, richiamando nuovamente i concetti teorici esposti nel secondo capitolo, incrementando l'angolo di sterzo iniziale il dominio di attrazione si riduce e contemporaneamente si sposta di una quantità pari all'angolo di assetto. Il caso limite è rappresentato dal collasso di uno dei punti di equilibrio instabile sul punto di equilibrio stabile, concetto che

bene interpreta la differenza di comportamento del veicolo tra le due prove di *step steer* condotte a differente angolo volante.

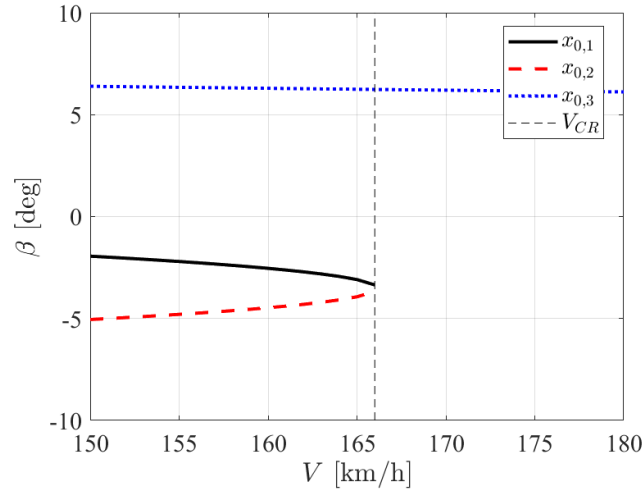


Figura 3.59: Posizione dei punti di equilibrio al variare della velocità $\delta_{SW} = 5 \text{ deg}$ - metodo piano delle fasi.

Ricavando n grafici analoghi a quello riportato in figura 3.59 per un numero n di angoli volante iniziali differenti, è stato possibile caratterizzare in campo non lineare la velocità critica del veicolo al variare del valore di angolo volante. Se ne riporta la caratteristica in figura 3.60.

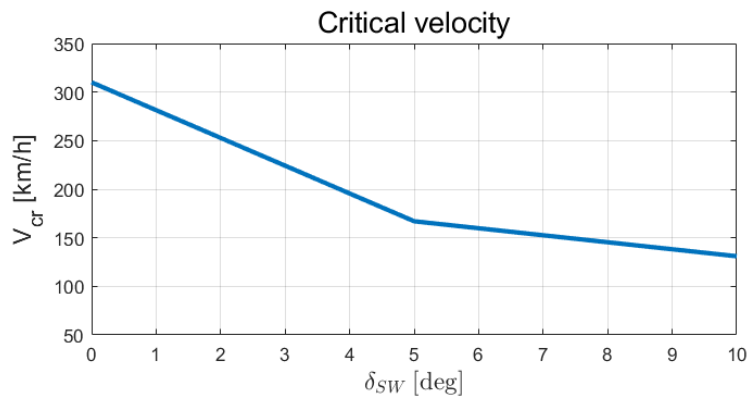


Figura 3.60: Velocità critica del veicolo al variare dell'angolo volante δ_{SW} - metodo piano delle fasi.

3.3 Scenario della simulazione

L'input primario del modello precedentemente descritto è lo scenario della simulazione. Nel corso dello sviluppo di questa tesi il modello è stato testato con diverse varianti di scenario, le quali sono state opportunamente descritte in seguito. Tutte le varianti sono state create ed introdotte come *scenario file*; la compatibilità del modello con questo tipo di file ha permesso di utilizzare il *Driving Scenario Designer Tool*, rendendo la creazione degli scenari un'operazione semplificata dall'interfaccia grafica e dall'intuitività del tool di MATLAB.

3.3.1 Driving Scenario Designer

Come anticipato, il *Driving Scenario Designer* è un'applicazione del software MATLAB che permette all'utente di progettare scenari di guida in modo da poter testare i propri sistemi di guida autonoma [22]. Tramite questo tool è possibile creare da zero uno scenario utilizzando un'apposita interfaccia *drag-and-drop*.

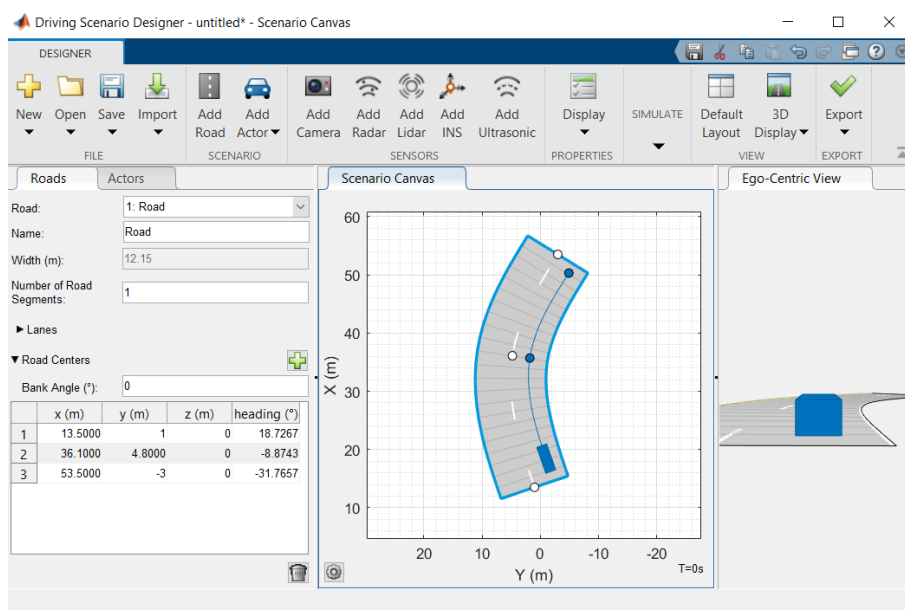


Figura 3.61: Interfaccia Driving Scenario Designer

In figura 3.61 si può apprezzare un semplice esempio di realizzazione di una traiettoria curvilinea, con un singolo veicolo (il veicolo *ego*) che percorre una delle due corsie. Gli elementi indispensabili alla creazione di uno scenario destinato a questa tipologia di uso sono la strada e gli attori; ovvero automobili, camion, pedoni, biciclette, barriere, guardrail e quant'altro.

Il primo passo per realizzare uno scenario, anche uno molto semplice come quello mostrato in figura, risulta essere la creazione della superficie stradale. Tramite il tab *Add Road* si accede alla scheda di configurazione dell'elemento strada, visibile sulla sinistra. L'applicazione da la possibilità di specificare, tra i vari parametri, il numero di segmenti che si intende creare, il numero di corsie, la loro ampiezza e l'angolo di inclinazione della superficie. Tuttavia, l'input primario risulta essere l'elenco dei cosiddetti *road centers*, ovvero i punti che si trovano sulla linea di mezzzeria della superficie stradale; questi possono essere inseriti mediante un'apposita tabella nella scheda di configurazione sulla sinistra o attraverso la già citata interfaccia *drag-and-drop*. Quest'ultima metodologia consiste nel posizionare i punti graficamente nel piano cartesiano presente nella scheda dal nome *Scenario Canvas*. Dopo averli posizionati, questi possono essere trascinati in altre posizioni, modificando di conseguenza la linea di mezzzeria della carreggiata. Anche utilizzando questa funzione la tabella relativa ai dati dai *road centers* è disponibile e viene compilata in maniera automatica; essa potrà essere modificata manualmente, permettendo in tal modo di correggere le imprecisioni del posizionamento grafico. Dopo aver definito i *road centers*, il software in maniera automatica genera una linea di mezzzeria e successivamente una superficie stradale comprensiva del numero di corsie specificato; la quale potrà essere visionata nella scheda *Scenario Canvas*. Risulta essere particolarmente funzionale la possibilità del software di generare una linea continua in maniera autonoma utilizzando un'interpolazione mediante clotoidi [22] e di modificare l'*heading angle* della linea creata in ogni punto inserito.

L'inserimento degli attori all'interno dello scenario segue una procedura molto simile a quella appena descritta. Se viene inserito un unico attore, come nell'esempio riportato in figura 3.61, questo verrà identificato come veicolo *ego*; in caso contrario è possibile specificare quale fra gli attori presenti è da considerare tale. Tramite l'apposita scheda è possibile modificare parametri del veicolo quali: dimensioni, tipologia di vettura, tipo di rappresentazione 3D ed altri. Tuttavia, l'aspetto principale di questa fase è la possibilità di definire una traiettoria per un dato veicolo in maniera analoga a quanto effettuato per la creazione della linea di mezzzeria della superficie stradale. Dunque, si procede con l'inserimento dei punti percorsi in successione dalla vettura attraverso l'inserimento di dati in un'apposita tabella o mediante approccio grafico; anche in questo caso questi punti verranno interpolati automaticamente. Inoltre, viene fornita la possibilità di inserire un limite sul Jerk massimo delle traiettorie generate. La velocità del veicolo può essere settata manualmente per ogni punto creato e ciò permette all'utente di calibrare anche le accelerazioni e le decelerazioni delle varie vetture.

Dopo aver inserito tutti gli attori dello scenario, può essere avviata una simulazione direttamente dall'applicazione; la quale può essere visualizzata nella scheda *Ego-Centric View* che si trova alla destra dell'interfaccia.

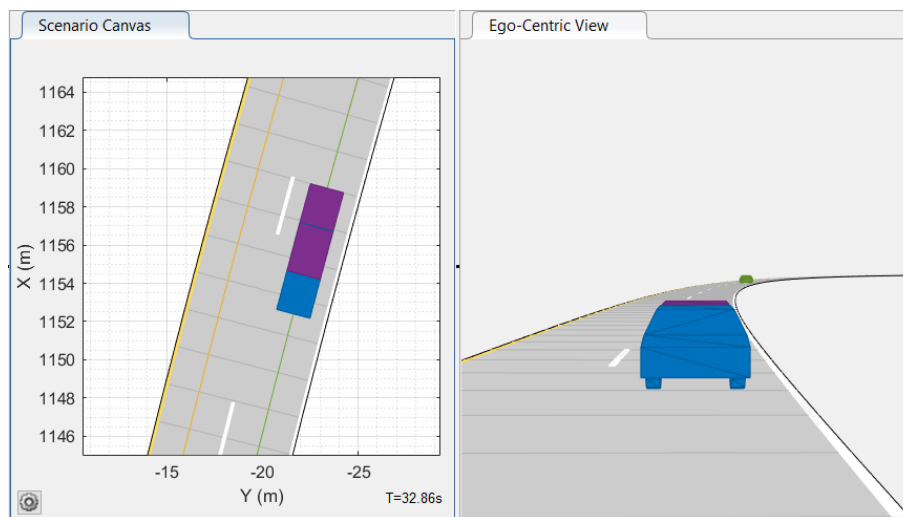


Figura 3.62: Sovrapposizione traiettorie Driving Scenario Designer

Negli scenari creati per testare il sistema di sorpasso automatico, la traiettoria del veicolo ego risulta essere sovrapposta in alcuni punti a quella di un altro veicolo. Si tratta di un aspetto voluto, in quanto, una volta inserito lo scenario all'interno del modello, il compito del sistema assistenziale alla guida è quello di riconoscere questa caratteristica ed ovviare al problema modificando la traiettoria del veicolo ego e/o modificandone la velocità. Tuttavia, all'interno della simulazione avviata mediante il *Driving Scenario Designer* questo non rappresenta un errore tale da interromperla, poiché i corpi risultano essere compenetrabili, come si può apprezzare osservando i veicoli blu e viola in figura 3.62.

Dopo aver esposto il procedimento da utilizzare per creare uno scenario partendo da zero, è necessario riportare alcune delle principali funzionalità del software descritto, al fine di poter sfruttare al meglio le relative potenzialità. Ciò a cui si fa riferimento è la possibilità di inserire e settare sensori del veicolo quali: camere, radar, Lidar, ultrasuoni e sistemi di navigazione inerziale. Inoltre, non è sempre necessario creare strade e/o scenari completi in maniera manuale, in quanto è possibile importarne alcuni precostruiti; ad esempio, è particolarmente interessante la possibilità di importare gli scenari dei test descritti nei protocolli dell'*European New Car Assessment Programme (Euro NCAP)* [22]. Il *Driving Scenario Designer Tool* supporta anche i file formato *OpenDRIVE*, il quale fornisce le specifiche necessarie a descrivere la rete stradale per applicazioni di simulazioni di guida [30]. Questa caratteristica apre le porte alla possibilità di importare le strade all'interno dello scenario direttamente da *ASAM OpenDRIVE*, *OpenStreetMap* o altri servizi web [22].

Come detto in precedenza, il vantaggio dell'usare questo tool è insito nell'immediatezza di utilizzo e nella compatibilità del tipo di output con il modello utilizzato. Il *Driving Scenario Designer* permette di ottenere un output contenente tutte le informazioni dello scenario nel cosiddetto *Scenario file*; come alternativa i dati possono essere inseriti in una function da poter richiamare nel modello, la quale genera autonomamente lo *Scenario file*. Nel modello utilizzato è stata preferita quest'ultima opzione.

3.4 Segnali ottenuti da simulazione

All'interno di questo capitolo sono stati illustrati i risultati di ogni caso studio in forma grafica. Con il fine di comprendere al meglio le grandezze descritte, si riportano di seguito dei richiami di teoria della dinamica del veicolo e la spiegazione delle logiche legate all'ottenimento e all'analisi dei segnali utili alle valutazioni effettuate dal pianificatore del modello. Nel corso della valutazione dei risultati la traiettoria è stata rappresentata all'interno di un piano cartesiano, nel quale sono stati riportati anche i limiti delle corsie della strada.

3.4.1 Richiami di dinamica del veicolo

Per definire le grandezze analizzate si descrive prima il sistema di riferimento adottato.

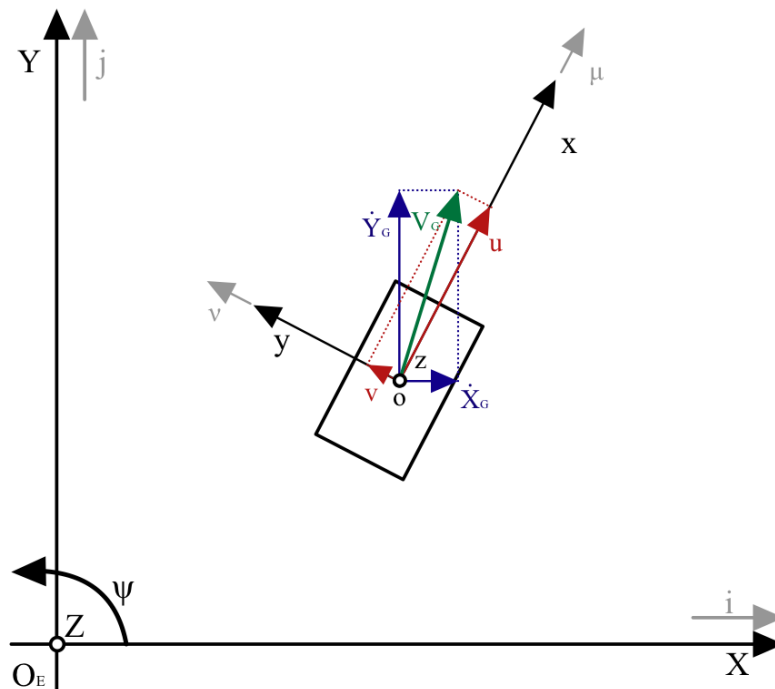


Figura 3.63: Sistema di riferimento veicolare

Si definisce un sistema di riferimento $(X, Y, Z; O_E)$ globale, grazie al quale è possibile definire la posizione del veicolo in termini assoluti. Inoltre, si definisce un sistema di riferimento $(x, y, z; O)$ locale, ovvero solidale al veicolo; nello specifico l'asse x coincide con l'asse longitudinale del veicolo, l'asse y con quello laterale. L'origine di tale sistema di riferimento è coincidente con il baricentro del veicolo e i suoi versori sono (ν, μ, λ) . Il sistema di riferimento globale è anche chiamato sistema inerziale, quello locale è conosciuto come sistema veicolo. Quest'ultimo viene ottenuto rispetto al primo mediante una rototraslazione; trasformazione in cui l'origine del sistema di riferimento trasla fino a coincidere con il baricentro del veicolo ed il sistema di riferimento globale ruota attorno all'asse Z in senso antiorario fin quando l'asse X è parallelo all'asse x . L'angolo percorso durante quest'ultima fase è l'angolo di imbardata ψ . Si riporta la matrice di rotazione R .

$$R = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.59)$$

Si definiscono le componenti di velocità del baricentro del veicolo rispetto al sistema di riferimento inerziale (\dot{X}_G e \dot{Y}_G). Le componenti di velocità del baricentro lungo il sistema di riferimento veicolo sono $(u$ e $v)$, rispettivamente chiamate velocità longitudinale e laterale. Le equazioni per ottenere le velocità del baricentro calcolate rispetto al sistema di riferimento globale, partendo da quelle del sistema di riferimento locale e dall'angolo di imbardata sono le seguenti:

$$\begin{cases} \dot{X}_G = u \cos \psi - v \sin \psi \\ \dot{Y}_G = v \cos \psi + u \sin \psi \end{cases} \quad (3.60)$$

Analogamente si può scrivere la formulazione che permette di effettuare il passaggio inverso:

$$\begin{cases} u = \dot{X}_G \cos \psi + \dot{Y}_G \sin \psi \\ v = \dot{Y}_G \cos \psi - \dot{X}_G \sin \psi \end{cases} \quad (3.61)$$

La velocità assoluta del veicolo (V_G) si può esprimere in tal modo:

$$\mathbf{V}_G = u\boldsymbol{\mu} + v\boldsymbol{\nu} \quad (3.62)$$

Per ottenere l'accelerazione del veicolo e le relative componenti lungo il sistema di riferimento veicolo si deriva la velocità, definita come nell'equazione 3.62.

$$\begin{aligned} \frac{d\mathbf{V}_G}{dt} &= \dot{u}\boldsymbol{\mu} + u\frac{d\boldsymbol{\mu}}{dt} + \dot{v}\boldsymbol{\nu} + v\frac{d\boldsymbol{\nu}}{dt} = \\ &= \dot{u}\boldsymbol{\mu} + u\dot{\psi}\boldsymbol{\lambda} \wedge \boldsymbol{\mu} + \dot{v}\boldsymbol{\nu} + v\dot{\psi}\boldsymbol{\lambda} \wedge \boldsymbol{\nu} \\ &= (\dot{u} - \dot{\psi}v)\boldsymbol{\mu} + (\dot{v} + \dot{\psi}u)\boldsymbol{\nu} \end{aligned} \quad (3.63)$$

Conseguentemente si può scrivere l'accelerazione assoluta del veicolo come:

$$\mathbf{a}_G = \frac{d\mathbf{V}_G}{dt} = a_x\boldsymbol{\mu} + a_y\boldsymbol{\nu} = (\dot{u} - \dot{\psi}v)\boldsymbol{\mu} + (\dot{v} + \dot{\psi}u)\boldsymbol{\nu} \quad (3.64)$$

Le componenti a_x e a_y sono rispettivamente l'accelerazione longitudinale e l'accelerazione laterale calcolate rispetto al sistema di riferimento mobile solidale al veicolo. Dopodiché è utile riportare il Jerk calcolato nelle componenti longitudinale e laterale, rispettivamente J_x e J_y , in quanto parametro strettamente correlato al comfort.

$$\mathbf{J}_x = \frac{da_x}{dt}, \quad \mathbf{J}_y = \frac{da_y}{dt} \quad (3.65)$$

3.4.2 Segnali di analisi strategia del pianificatore

Risulta particolarmente utile analizzare la logica del sistema circa le valutazioni sulla sicurezza del veicolo *ego* rispetto agli altri veicoli presenti nello scenario. Questi segnali sono quelli che il pianificatore valuta istante per istante ed il loro confronto con valori di soglia porta il sistema a valutare la migliore soluzione in termini di modalità di guida adottata; ovvero le modalità CC, LCF e LC.

Per aiutare l'assimilazione di tali concetti è stato realizzato un diagramma di flusso, riportato in figura 3.66. Ad ogni step temporale e per ogni veicolo identificato come veicolo MIO deve essere eseguita la valutazione rappresentata. L'output di questo processo logico informa il sistema circa la sicurezza di ogni veicolo MIO; dove per MIO sicuro (*Safe*) si intende un veicolo che rispetta i limiti imposti dal sistema riguardanti il tempo di collisione (TTC) e i cosiddetti *Safe Ranges*, ovvero delle distanze di sicurezza minime, da mantenere, fra il veicolo *ego* e gli altri veicoli nello scenario.

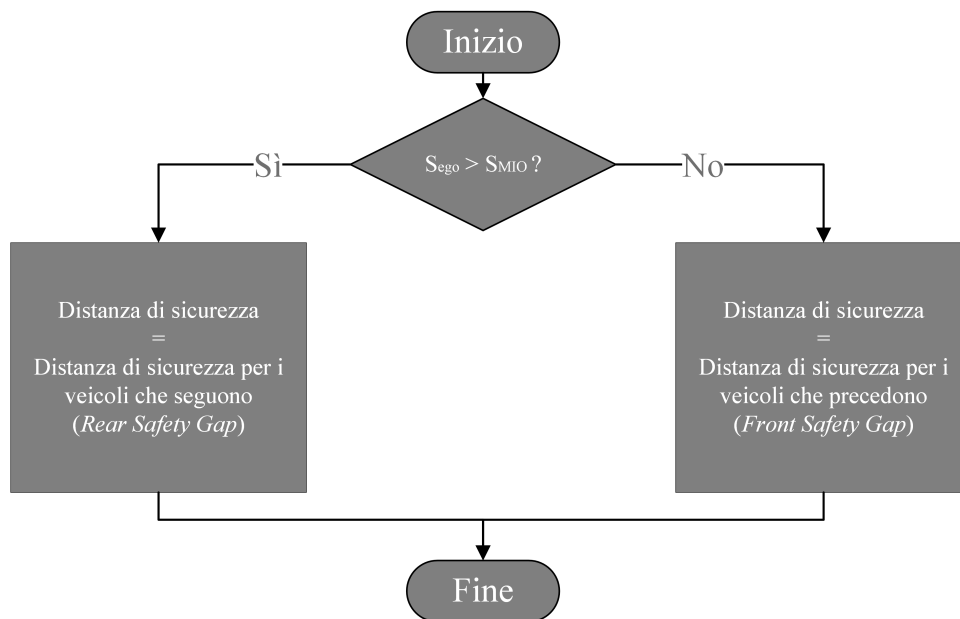


Figura 3.64: Diagramma di flusso scelta *Safety gap*

Nel modello analizzato è stata implementata la possibilità di differenziare il limite sulla distanza relativa imposto per i veicoli che seguono e che precedono il veicolo *ego*. Conseguentemente; viene effettuato un confronto tra le coordinate s dei due veicoli, con s coordinata curvilinea calcolata rispetto ad un sistema di riferimento di Frenét. In figura 3.64 è riportato un diagramma di flusso che illustra la logica di scelta del valore di distanza di sicurezza fra i due veicoli.

Una prima valutazione viene effettuata confrontando il corretto limite con la distanza relativa (*Relative distance*) tra i veicoli *ego* e MIO, calcolata in termini assoluti. La formula adottata per il calcolo di tale parametro è quella riportata nell'equazione 3.66. Si precisa che le traiettorie vengono pianificate considerando i veicoli come puntiformi, facendo poi coincidere tale punto con il centro dell'assale posteriore del dato veicolo. Nella rappresentazione realizzata la posizione dei due veicoli viene identificata dai punti P_{ego} e P_{MIO} .

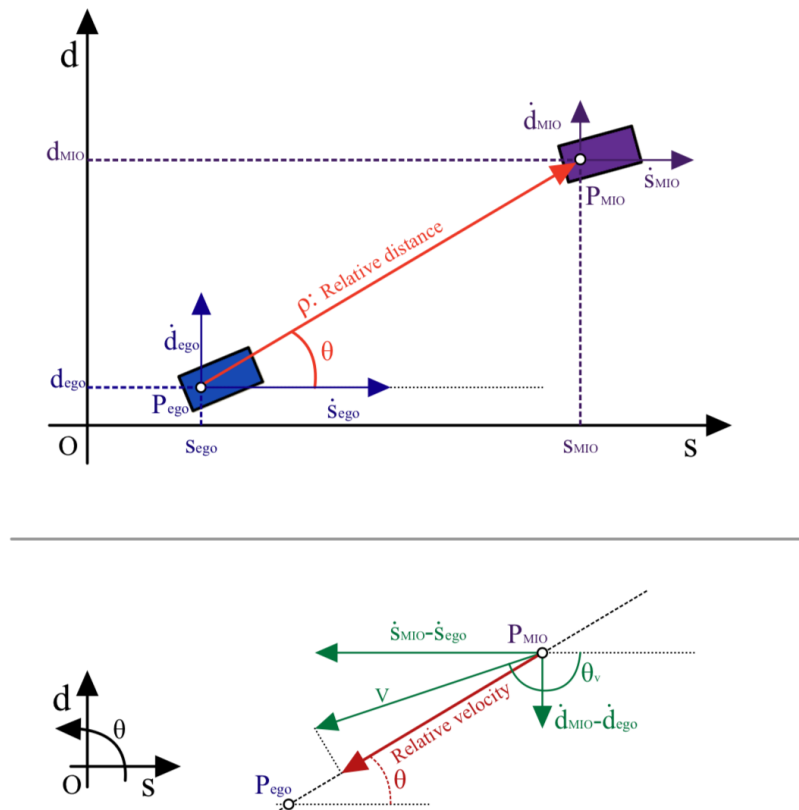


Figura 3.65: Distanza relativa veicoli ego/MIO

La seconda valutazione viene effettuata monitorando il tempo necessario alla collisione (TTC) fra i due veicoli, *ego* e MIO, date la loro velocità relativa e la loro distanza. Nella seguente figura 3.65, si ha un riscontro grafico delle grandezze utili a calcolare il TTC. Il veicolo *ego* è rappresentato in blu, il veicolo MIO in viola. Si noti che il sistema di riferimento utilizzato è un sistema di Frenét; dunque, descritto dalla coordinata curvilinea s e dalla distanza in direzione normale alla curva di riferimento d .

Si riportano di seguito le formule utilizzate per il calcolo delle grandezze rappresentate:

$$\text{Relative distance} = \sqrt{(s_{MIO} - s_{ego})^2 + (d_{MIO} - d_{ego})^2} \quad (3.66)$$

$$V = \sqrt{(\dot{s}_{MIO} - \dot{s}_{ego})^2 + (\dot{d}_{MIO} - \dot{d}_{ego})^2} \quad (3.67)$$

$$\text{Relative velocity} = V \cos(\theta - \theta_V) \quad (3.68)$$

L'angolo θ_V viene calcolato nel seguente modo:

$$\theta_V = \text{atan2}\left(\frac{\dot{d}_{MIO} - \dot{d}_{ego}}{\dot{s}_{MIO} - \dot{s}_{ego}}\right) \quad (3.69)$$

Dove *atan2* indica l'arcotangente a quattro quadranti o arcotangente2; la quale è una funzione a due argomenti che rappresenta una variazione della funzione arcotangente. Tale funzione restituisce un valore compreso nell'intervallo $(-\pi, \pi]$, a differenza dell'arcotangente che ha valori in $(-\frac{\pi}{2}, \frac{\pi}{2})$. La funzione *atan2*(y/x) calcola l'angolo in radianti tra il semiasse positivo delle X di un piano cartesiano e un generico punto (x,y) giacente sul medesimo piano. L'angolo è positivo se percorso in senso orario, negativo se percorso in senso antiorario, come nel caso rappresentato in figura 3.65. La funzione risulta essere sempre definita tranne che per il punto $(0,0)$; tuttavia, la funzione Matlab *atan2*(y,x) ricevendo in input $(0,0)$ restituisce il valore 0 .

La velocità relativa *Relative velocity* tra i due veicoli è definita come la proiezione del vettore velocità V lungo la congiungente dei centri dell'assale posteriore dei due veicoli. Il vettore velocità V è a sua volta definito mediante le due componenti di velocità relativa, rispettivamente lungo s e lungo d . Dunque, si tratta di un parametro che esprime la velocità con cui i due veicoli si avvicinano o si allontanano lungo la direzione che li congiunge. Qualora il valore del parametro *Relative velocity* fosse negativo, ciò indicherebbe un avvicinamento dei due veicoli; viceversa, un allontanamento.

Dopodiché è possibile calcolare il tempo necessario all'ipotetica collisione fra i due veicoli, *ego* e MIO, espresso in secondi nel seguente modo:

$$TTC = \frac{Relative\ distance}{Relative\ velocity} \quad (3.70)$$

Per valori di *Relative velocity* troppo piccoli si ottengono valori di *TTC* in modulo molto elevati. Per rimediare a tale problema all'interno del modello è stato inserito un filtro che nel caso di valori più piccoli della soglia li sostituisce con il valore della soglia stesso. Nel modello è stata imposta una soglia pari a $0.05\ m/s$.

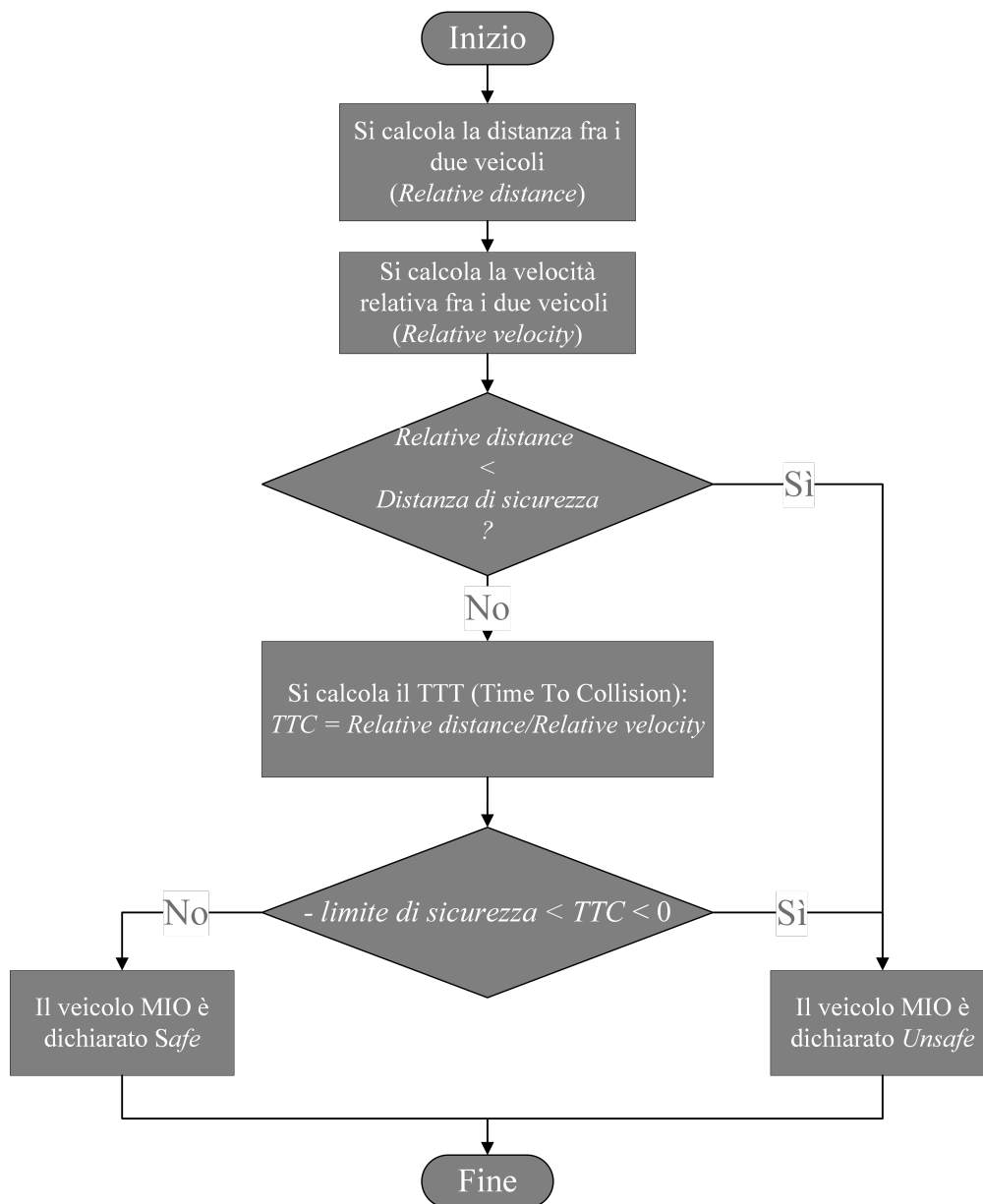


Figura 3.66: Diagramma di flusso valutazione veicoli MIO

Come anticipato, il diagramma di flusso in figura 3.66 riporta la logica di valutazione di un veicolo MIO implementata nel modello utilizzato. Un veicolo MIO viene definito *Unsafe* se rispetta almeno una delle seguenti condizioni:

- La distanza relativa fra tale veicolo MIO ed il veicolo ego risulta essere inferiore alla distanza di sicurezza *Rear Safety Gap/Front Safety Gap*.
- Il TTC assume un valore negativo e contemporaneamente inferiore in modulo al limite di sicurezza.

Si riportano nella seguente tabella 3.7 le condizioni logiche precedentemente descritte; identificandole con le lettere *A*, *B* e *C*.

Logical condition	Description
A	$Relative\ Distance < Safety\ Gap$
B	$TTC < 0$
C	$ TTC < TTC\ limit$

Tabella 3.7: Condizioni logiche per definizione *Safe/Unsafe*

Tali condizioni logiche permettono di definire mediante equazioni logiche gli stati di veicolo *Safe* e *Unsafe*.

$$\begin{aligned}
 Unsafe &= A \vee (B \wedge C) \\
 Safe &= \neg Unsafe
 \end{aligned}
 \tag{3.71}$$

Esaminando la formula 3.70 si può affermare che un valore negativo di TTC implica un valore di velocità relativa dei due veicoli *Relative velocity* negativo; che, come commentato in precedenza, implica un progressivo avvicinamento tra i due, ovvero una progressiva riduzione del parametro *Relative distance*. I limiti di sicurezza citati e imposti all'interno del modello sono riportati nella seguente tabella 3.8.

Parameter	Limit	Unit
Front Safety Gap	30	<i>m</i>
Rear Safety Gap	10	<i>m</i>
TTC limit	4	<i>s</i>

Tabella 3.8: Limiti di default su TTC e distanza di sicurezza

Stabilire se un veicolo MIO è definito *Safe* o *Unsafe* è necessario per valutare la corsia sulla quale spostare il veicolo *ego* o se eventualmente mantenere la corsia attuale. Nello specifico, il pianificatore valuta i veicoli MIO presenti nella corsia del veicolo *ego*; qualora fosse esistente il rischio di collisione, ovvero fosse presente nella medesima corsia un veicolo MIO definito *Unsafe*, allora esso procederebbe con la medesima analisi anche sulle corsie adiacenti. Se nelle corsie adiacenti non vi è alcun veicolo MIO *Unsafe*, a quel punto il pianificatore valuta in base ad una data funzione di costo se mantenere la propria corsia, abilitando la modalità modalità CC o LCF, oppure se abilitare la strategia di cambio corsia. Dunque, si sottolinea che la presenza di un veicolo *Unsafe* nella corsia del veicolo *ego* non implica necessariamente l'immediata abilitazione della modalità di cambio corsia.

Le definizioni di *Safe/Unsafe* vanno ad influenzare l'indice della corsia preferenziale. Tuttavia, come anticipato, un eventuale cambio di corsia preferenziale non determina necessariamente un cambio corsia del veicolo; ciò dipende da valutazioni sulle alternative legate alla fattibilità, al rischio di collisione ed al parametro di costo associato. Ciò che la variazione di corsia preferenziale implica necessariamente è l'attivazione del flag di pianificazione della traiettoria; ovvero viene dato un segnale di input al modello, in modo che venga valutato in quel dato istante il set di alternative disponibili, ognuna corrispondente ad una data modalità (CC, LCF e LC), e che ne venga individuata la migliore.

Capitolo 4

Casi studio e risultati ottenuti

Con il fine di fornire una visione completa dello studio di un sistema di sorpasso automatico, sono stati analizzati gli aspetti più critici della manovra e successivamente testati mediante opportune simulazioni. Per effettuare i test e per sviluppare opportune metodologie in grado di arginare le problematiche evidenziate, sono stati utilizzati gli strumenti analizzati all'interno dei capitoli precedenti, messi a disposizione per l'ambiente di simulazione MatLab & Simulink. Ciò che nel concreto è stato prodotto è un'insieme di scenari, ognuno dei quali sviluppato con la finalità di mettere in risalto un determinato aspetto o una determinata problematica del sistema assistenziale; i quali hanno poi portato ad effettuare delle considerazioni teoriche, allo sviluppo di soluzioni da implementare nell'ambiente di simulazione adottato e all'esposizione delle stesse mediante risultati grafici e numerici.

4.1 Introduzione dei casi studio

Lo studio portato avanti nei capitoli 2 e 3 ha permesso di comprendere le principali caratteristiche di cui fornire un sistema dedicato alla progettazione di una traiettoria di sorpasso e di acquisire una conoscenza dettagliata sulla costruzione di un modello destinato a questo scopo. All'interno di questo capitolo sono state approfondite alcune delle tematiche inerenti all'argomento, tutte analizzate con l'ausilio del simulatore presentato nel capitolo 3. Tutta la trattazione inerente ai casi studio dal primo al quarto è stata prodotta utilizzando il modello contenente il solo pianificatore di traiettorie, con il fine di evitare alterazioni dei segnali prodotti, quindi per calibrare con maggiore accuratezza le strategie implementate. L'ultimo scenario ha richiesto l'analisi mediante modello completo, in quanto la strategia implementata è strettamente correlata al comportamento effettivo del veicolo. Le tematiche analizzate all'interno di questo capitolo sono le seguenti:

- **Caso 1 - Singolo cambio di corsia:** Per rendere chiara la logica con cui sono state studiate le singole casistiche è stato ritenuto opportuno analizzare il comportamento del modello applicato ad un caso banale. Nello specifico è stato scelto uno scenario con un solo veicolo da sorpassare, mediante un singolo cambio di corsia ed in totale assenza di altri attori. In tal modo è stato possibile mostrare il funzionamento del modello per simulazioni prive di criticità ed effettuare una disamina sulla tipologia di risultati ottenibili.



Figura 4.1: Scenario "*Singolo cambio di corsia*"

- **Caso 2 - Sorpasso con veicolo in senso di marcia opposto:** Un'ulteriore simulazione è stata effettuata su uno scenario introdotto all'interno del capitolo 2 con il nome di "*Sorpasso con veicolo in senso di marcia opposto*". Si tratta di uno scenario utilizzato nella precedente trattazione per facilitare la comprensione, mediante un riscontro pratico, di alcune funzioni o strategie illustrate. Conseguentemente è stato scelto di espandere la gamma delle simulazioni riportate includendo tale scenario, sia per ottenere una trattazione completa del caso studio sia per sviluppare una strategia a fronte della criticità rappresentata dal sopraggiungere di veicoli con senso di marcia opposto. In tal modo è stato possibile ottimizzare il sistema assistenziale per scenari non esclusivamente autostradali ma con carreggiate a doppio senso di marcia.



Figura 4.2: Scenario "*Sorpasso con veicolo in senso di marcia opposto*"

- **Caso 3 - Sorpasso di una fila di veicoli:** Una situazione di guida alquanto critica, sia per un pilota sia per il simulatore, è quella in cui i veicoli da sorpassare sono molteplici e molto vicini fra loro. Nello specifico, è stato scelto questo caso studio per analizzare le risposte del modello, nel caso di scenari in cui il posizionamento di veicoli non identificati inizialmente come MIO non è irrilevante per il pianificatore; con il fine ultimo di sviluppare le corrette strategie da implementare per migliorare le performance del sistema assistenziale.



Figura 4.3: Scenario "*Sorpasso di una fila di veicoli*"

- **Caso 4 - Logiche di scelta corsia per il veicolo *ego*:** Con questo caso è stata dettagliata l'analisi circa la determinazione di una corretta strategia per il posizionamento del veicolo *ego* all'interno della carreggiata, dettata non solo dalla posizione degli altri veicoli ma anche dal rispetto delle norme stradali. Nello specifico è stata implementata una strategia di rientro in corsia dopo l'esecuzione della manovra di sorpasso.
- **Caso 5 - Modalità di cambio corsia rapido:** In alcuni contesti viene richiesto al pilota o al simulatore di effettuare un cambio di corsia nel minor tempo possibile, penalizzando il comfort dei passeggeri pur di abbandonare rapidamente l'attuale corsia occupata. Il sistema assistenziale è stato sviluppato di conseguenza, in modo da includere una strategia in grado di dare maggior peso all'obiettivo temporale; la quale può essere attivata in casi di emergenza.

4.2 Caso 1 - Singolo cambio di corsia

Lo scenario di questo caso studio è stato progettato con lo scopo di fornire un esempio semplice dal quale ottenere ed analizzare i risultati di una simulazione effettuata tramite il modello presentato nel capitolo 3. Esso si compone di un veicolo *ego* e di un veicolo target che transitano lungo la medesima corsia di una strada rettilinea a doppia corsia.

Lunghezza strada	250m
n° corsie	2

Tabella 4.1: Dati strada - scenario "Singolo cambio di corsia"

Il veicolo target è stato posizionato in modo da precedere il veicolo *ego*, assumendo quindi la denominazione di *lead vehicle*. Inoltre; con il fine di creare la necessità di valutare se effettuare o meno la manovra di sorpasso, i due veicoli sono stati caratterizzati da due velocità costanti e differenti, quella del veicolo lead inferiore rispetto a quella del veicolo ego. Infine, ad entrambi gli attori è stata associata la medesima geometria. Si riportano nella seguente tabella 4.2 i dati numerici.

Distanza iniziale ego - lead	40m
v_{ego}	20m/s
v_{lead}	15m/s

Tabella 4.2: Dati veicoli - scenario "Singolo cambio di corsia"

In figura 4.4 sono stati riportati due grafici utili a comprendere la costruzione dello scenario. Il grafico in alto riporta lo stato iniziale dei due veicoli; le cui velocità sono state rappresentate mediante due vettori, aventi diverso modulo ma medesima direzione e verso.

Il grafico in basso mostra l'evoluzione dello scenario in assenza di un intervento del sistema assistenziale, dunque, di una variazione della traiettoria e/o della velocità del veicolo *ego*. In particolare viene registrata una collisione fra i due veicoli dopo 6.6 s.

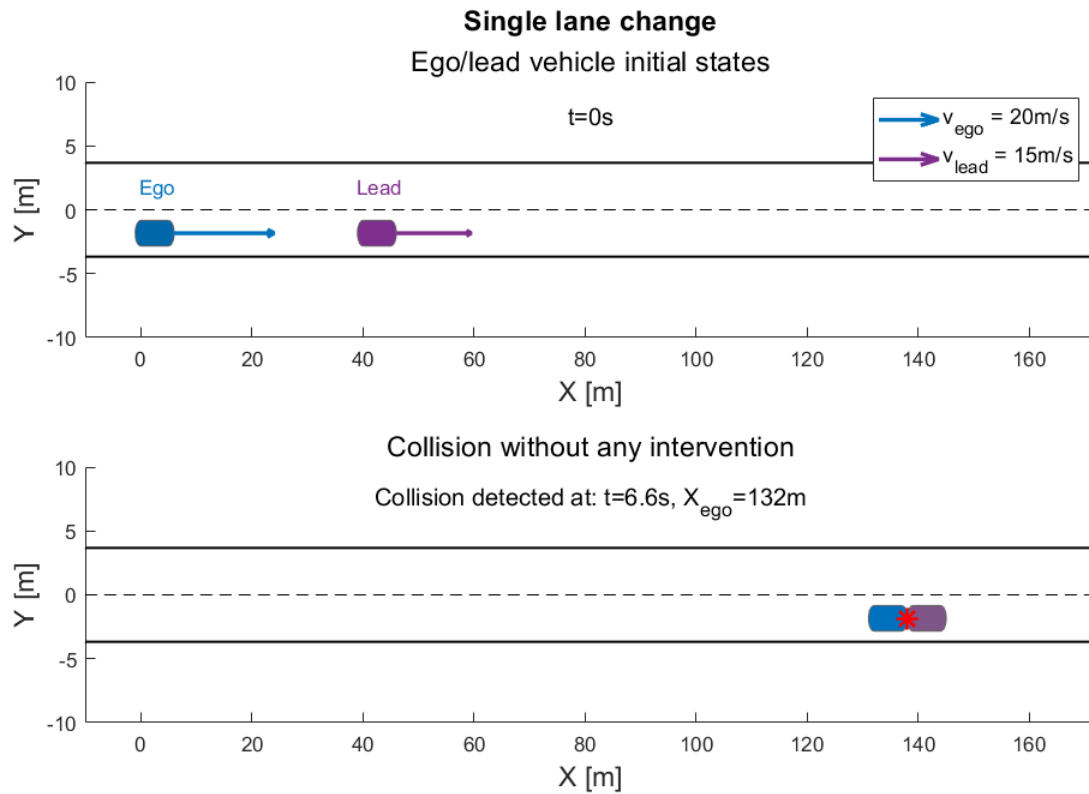


Figura 4.4: Stati iniziali $t=0s$ e collisione - scenario "Singolo cambio di corsia"

Per poter testare il modello di sorpasso automatico con questo specifico scenario è stato necessario riprodurlo mediante l'applicazione *Driving scenario designer*, in modo da ottenere il corretto input. Dopodiché, la *function* prodotta dal *designer*, contenente l'insieme dei dati di attori e strada, è stata caricata all'interno del modello ed è stata effettuata la simulazione. Essa ha dato esito positivo; ovvero, il veicolo ego ha effettuato il sorpasso e non sono stati registrati errori durante la simulazione. Si specifica che la simulazione è stata eseguita mantenendo tutti i parametri di configurazione del pianificatore di default descritti in tabella 3.1. Si riporta di seguito l'analisi dei risultati.

4.2.1 Risultati simulazione - caso 1

Come anticipato, durante la simulazione non sono stati riscontrati errori e la strategia utilizzata dal pianificatore per evitare la collisione con il veicolo *lead* è stata quella di abilitare la funzione di cambio corsia. In figura 4.5 è riportata la traiettoria del veicolo ego.

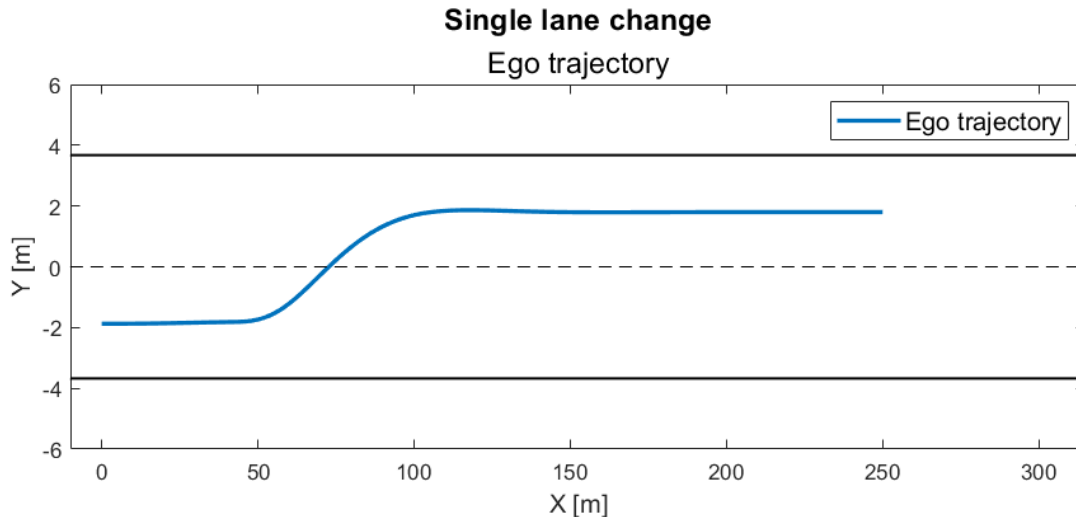


Figura 4.5: Traiettoria veicolo *ego* - scenario "Singolo cambio di corsia"

Si specifica che in tutti i grafici riportati in questo capitolo contenenti traiettorie o segmenti di traiettoria, è necessario prestare attenzione alla differenza di scala tra l'asse delle ascisse e quello delle ordinate. A causa di diversi ordini di grandezza di differenza tra lunghezza e larghezza del tratto di strada esaminato, le traiettorie riportate risultano essere "schiacciate"; dunque, non bisogna porre affidamento alla curvatura delle traiettorie rappresentate, come in figura 4.5, o alla forma dei veicoli nella rappresentazione mediante capsule, come in figura 4.7.

In figura 4.6 nel primo grafico, si ha una rappresentazione dell'andamento degli indici legati alla corsia del veicolo *ego*. Nello specifico si nota nell'istante $t=2.1$ s il pianificatore varia la corsia preferenziale, identificandola con la corsia di sinistra; corsia che viene poi raggiunta nell'istante $t=3.7$ s. All'interno del grafico in basso viene rappresentata la modalità di funzionamento del modello istante per istante. In seguito alle considerazioni effettuate in merito alla potenziale collisione rilevata,

il modello valuta come stato terminale ottimale quello che prevede un cambio corsia (LC), la preferenza passa alla modalità di cruise control (CC) solo quando il veicolo transita lungo la corsia di sinistra.

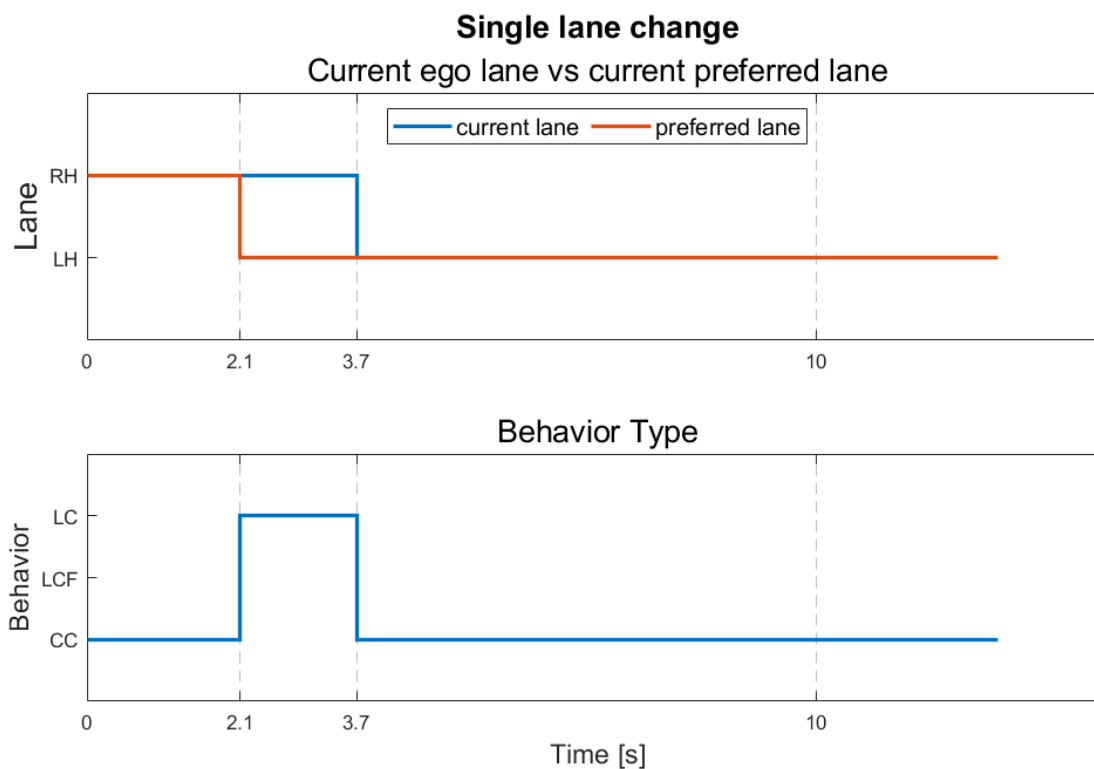


Figura 4.6: Corsia veicolo *ego* - scenario "Singolo cambio di corsia"

In figura 4.7 sono stati riportati alcuni step della traiettoria percorsa dal veicolo *ego*, mediante una rappresentazione in capsule dei veicoli. Nello specifico sono stati rappresentati: l'istante iniziale ($t=0s$), l'istante di inizio cambio corsia ($t=2.1s$), l'istante di raggiungimento del centro della corsia di sorpasso ($t=6.0s$), l'istante in cui il veicolo *ego* sorpassa il veicolo *lead* rispetto la *center line* ($t=8.1s$) e, per concludere, l'istante di fine simulazione ($t=12.5s$).

Nella rappresentazione corrispondente all'inizio del cambio corsia ($t=2.1s$) si ha un segmento della traiettoria futura del veicolo *ego*. Tale segmento è stato rappresentato in quanto l'istante in esame è quello in cui si ha la variazione di corsia preferenziale ed una conseguente pianificazione del segmento di traiettoria futura. Entrando maggiormente nel dettaglio, circa i meccanismi di pianificazione,

la variazione di corsia preferenziale porta, in seguito ad un opportuno confronto fra tutte le varie alternative, l'individuazione della modalità LC come ottimale. Questo avviene in quanto, per costruzione del modello, ogniqualvolta varia la *preferred line* viene aggiornato l'insieme degli stati terminali ed ognuno di questi viene valutato con il fine di identificare un nuovo segmento di traiettoria, se fra le alternative disponibili ne esiste una valida.

Di conseguenza, il tratto in verde rappresenta la traiettoria che il veicolo *ego* dovrà seguire fino al dato orizzonte temporale; istante in cui esso sarà collocato al centro della corsia di sinistra, in quanto la modalità con la quale è stato pianificato questo segmento di traiettoria è la LC.

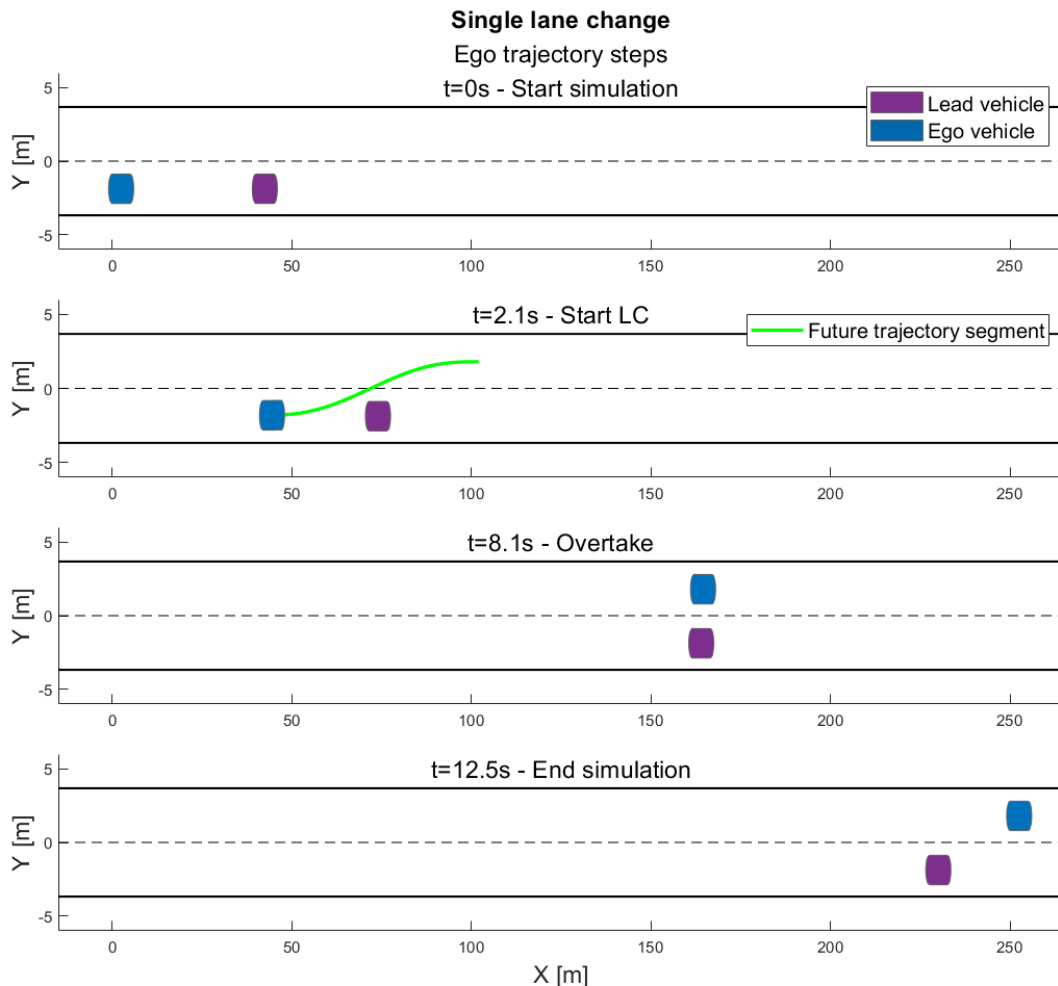


Figura 4.7: Step traiettoria veicolo *ego* - scenario "Singolo cambio di corsia"

Con il fine di illustrare nel dettaglio i meccanismi di selezione del segmento di traiettoria futura, in figura 4.8 sono state riportate graficamente le traiettorie corrispondenti a tutti gli stati terminali presenti nel set di alternative nell'istante $t=2.1s$. Nel terzo grafico è stata riportata la posizione futura del veicolo *lead* per ognuno dei tre orizzonti temporali, cosicché venga evidenziata l'assenza di collisioni mediante un confronto con il corrispettivo punto terminale di uno dei segmenti di traiettoria rappresentati e calcolato con il medesimo orizzonte temporale.

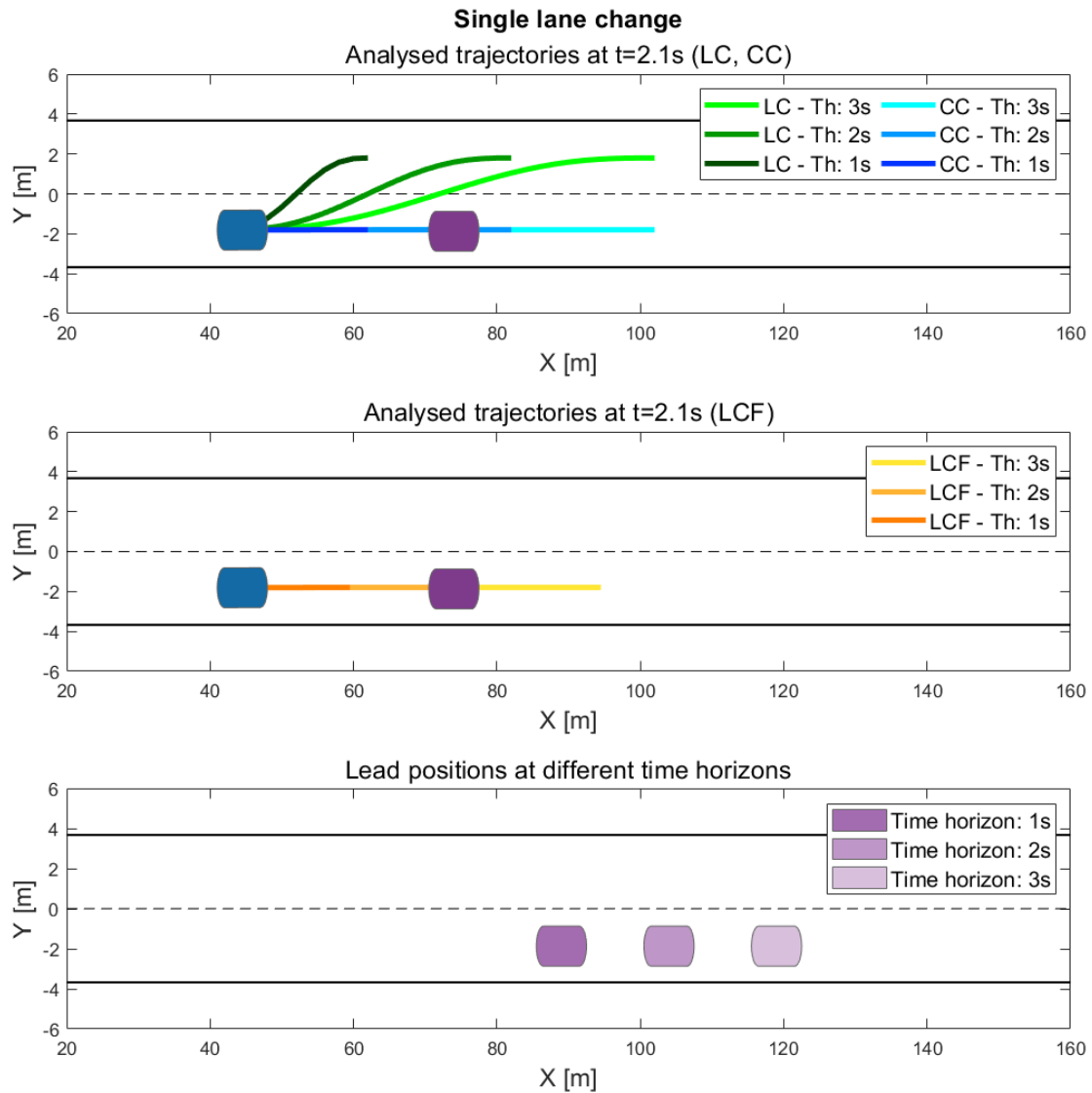


Figura 4.8: Analisi traiettorie future $t=2.1s$ - scenario "Singolo cambio di corsia"

Le alternative disponibili sono le seguenti:

- modalità LC: per ognuno dei tre orizzonti temporali ($1s$, $2s$ e $3s$), sono state rappresentate con diverse sfumature di verde. Ognuna delle quali colloca nel proprio punto terminale il veicolo *ego* al centro della corsia di sinistra con una manovra svolta mantenendo la velocità costante.
- modalità CC: per ognuno dei tre orizzonti temporali ($1s$, $2s$ e $3s$), sono state rappresentate con diverse sfumature di blu. Ognuna delle quali colloca nel proprio punto terminale il veicolo *ego* al centro della corsia di destra con una manovra svolta mantenendo la velocità costante.
- modalità LCF: per ognuno dei tre orizzonti temporali ($1s$, $2s$ e $3s$), sono state rappresentate con diverse sfumature di arancione. Ognuna delle quali colloca nel proprio punto terminale il veicolo *ego* al centro della corsia di destra con una manovra svolta riducendo progressivamente la velocità del veicolo *ego* fino a che essa non eguaglia quella del veicolo *lead*. Si noti come, a causa della riduzione di velocità, il punto terminale sia collocato ad un'ascissa inferiore rispetto ai segmenti di traiettoria realizzati con le due modalità differenti ma corrispondenti al medesimo orizzonte temporale.

mode	\dot{s} [m/s]	d [m]	time hzn. [s]	lat. cost	time cost	speed cost	cost
LC	20	1.80	3	0	-3	0	-3
LC	20	1.80	2	0	-2	0	-2
LC	20	1.80	1	0	-1	0	-1
CC	20	-1.80	3	3.6	-3	0	0.6
CC	20	-1.80	2	3.6	-2	0	1.6
CC	20	-1.80	1	3.6	-1	0	2.6
LCF	15	-1.80	3	3.6	-3	5	5.6
LCF	15	-1.80	2	3.6	-2	5	6.6
LCF	15	-1.80	1	3.6	-1	5	7.6

Tabella 4.3: Parametro di costo set traiettorie alternative ($t=2.1s$) - scenario "Singolo cambio di corsia"

La tabella 4.3 riporta il set stati terminali relativi all'istante $t=2.1s$. Ognuno di essi è stato descritto mediante la modalità, la velocità in coordinate di *Frenét*, la distanza dalla *center line*, l'orizzonte temporale, i contributi del parametro di costo associati alla deviazione laterale, al tempo e alla velocità, ed infine dal parametro di costo totale; essi sono stati ordinati in base a quest'ultimo valore. Come si può notare la soluzione individuata, quella con il parametro di costo inferiore, corrisponde al segmento di traiettoria calcolato con modalità LC e con un orizzonte temporale pari a $3s$. Il parametro di costo è viene calcolato con la seguente equazione.

$$cost = lat. cost + time cost + speed cost \quad (4.1)$$

Dove si ricorda che i singoli contributi rappresentano:

- *lat. cost*: la distanza tra la corsia preferenziale e la corsia nel quale viene collocato il veicolo *ego* al dato orizzonte temporale. Il dato calcolato viene moltiplicato per un coefficiente, in tal caso unitario.
- *time cost*: l'intervallo di tempo che separa l'istante corrente da quello in cui il veicolo si troverà nel punto terminale del segmento di traiettoria calcolato, ovvero l'orizzonte temporale. Tale dato viene poi moltiplicato per un coefficiente pari a -1, in modo tale da preferire le pianificazioni a lungo termine.
- *speed cost*: la differenza di velocità tra quella all'orizzonte temporale e quella stabilita in fase di descrizione dello scenario. Il dato calcolato viene moltiplicato per un coefficiente, in tal caso unitario.

In figura 4.9 e 4.11 sono stati riportati gli andamenti delle principali grandezze che descrivono il moto del veicolo *ego* durante la simulazione. Si nota un picco della velocità di imbardata piuttosto consistente, tuttavia ben al di sotto dei limiti sulla fattibilità della traiettoria definiti. La velocità laterale da un contributo piuttosto piccolo alla velocità assoluta del veicolo, la quale, come si può vedere dal terzo grafico, ricopia fedelmente l'andamento della velocità longitudinale.

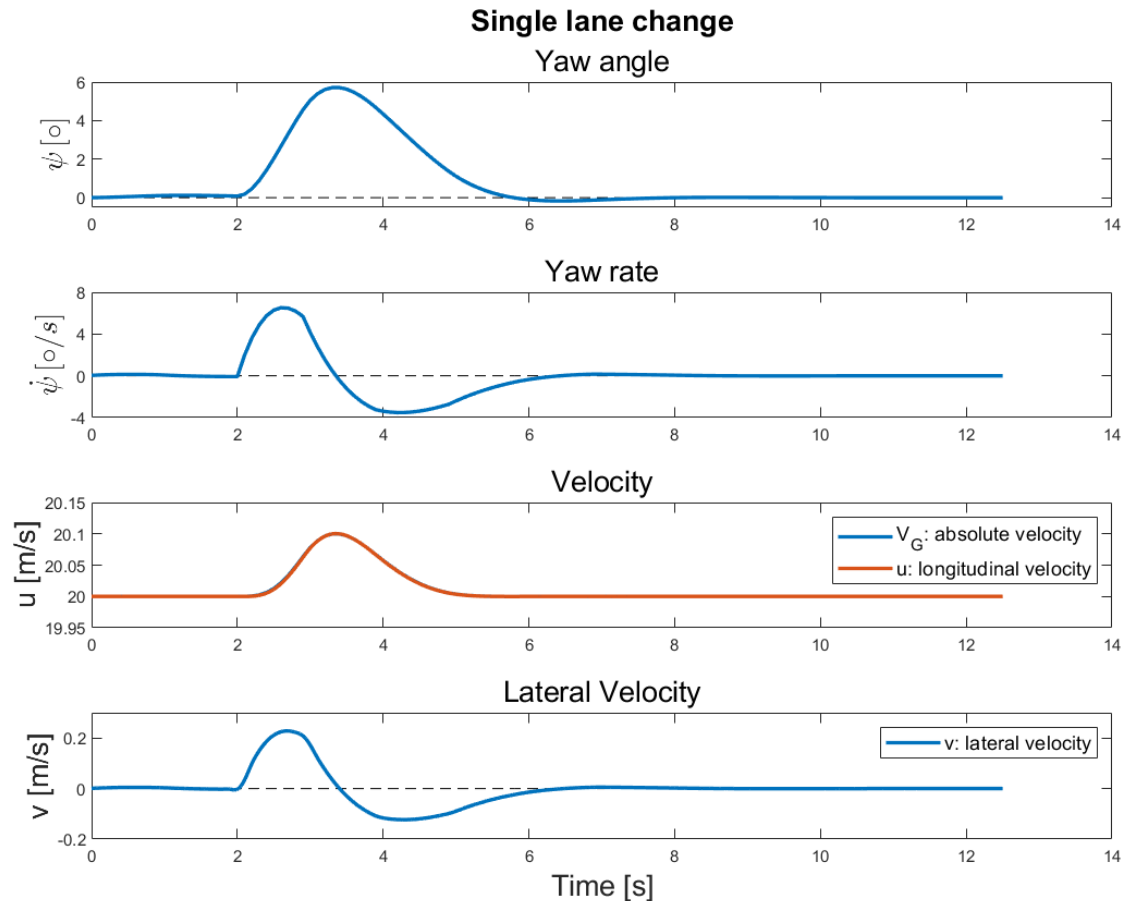


Figura 4.9: Andamenti grandezze moto veicolo *ego* - scenario "Singolo cambio di corsia"

Come si può notare dall'andamento delle accelerazioni in figura 4.11, l'accelerazione longitudinale, la cui equazione è riportata di seguito, assume valori di scostamento dallo zero molto piccoli rispetto all'accelerazione laterale, durante la manovra di cambio corsia.

$$\mathbf{a}_{long} = (\dot{u} - \dot{\psi}v)\boldsymbol{\mu} \quad (4.2)$$

Entrambe le componenti dell'accelerazione longitudinale sono state riportate graficamente nella seguente figura 4.10.

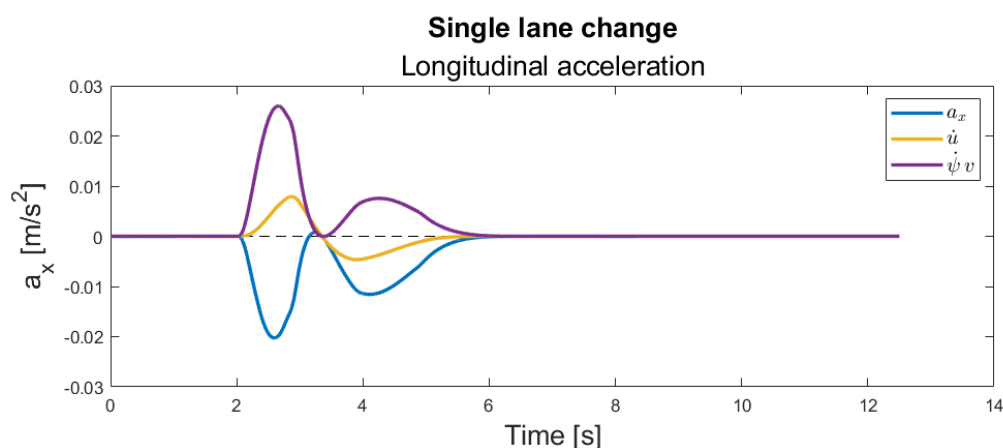


Figura 4.10: Accelerazione longitudinale veicolo *ego* - scenario "Singolo cambio di corsia"

Come ci si aspetta, anche nel caso del Jerk le componenti laterale e longitudinale della grandezza analizzata registrano valori con ordini di grandezza differenti. Il contributo della componente laterale del Jerk risulta essere il maggiore dei due; il quale, tuttavia, si mantiene ben al di sotto della soglia massima imposta.

Si riportano nella seguente tabella 4.4 i valori di picco, calcolati in modulo, delle grandezze riportate.

Parameter	Max	Limit	Unit
Yaw rate	6.52	20	$^{\circ}/s$
Long. Acceleration	$2.02 \cdot 10^{-2}$	5	m/s^2
Lat. Acceleration	2.29	5	m/s^2
Long. Jerk	$3.78 \cdot 10^{-3}$	5	m/s^3
Lat. Jerk	0.43	5	m/s^3

Tabella 4.4: Valori di picco grandezze veicolo *ego* - scenario "Singolo cambio di corsia"

Come detto inizialmente, dal confronto di ogni parametro con il relativo valore limite, non è stato registrato sfioramento alcuno rispetto la soglia; conseguentemente, la simulazione ha avuto esito positivo e la traiettoria individuata risulta essere valida.

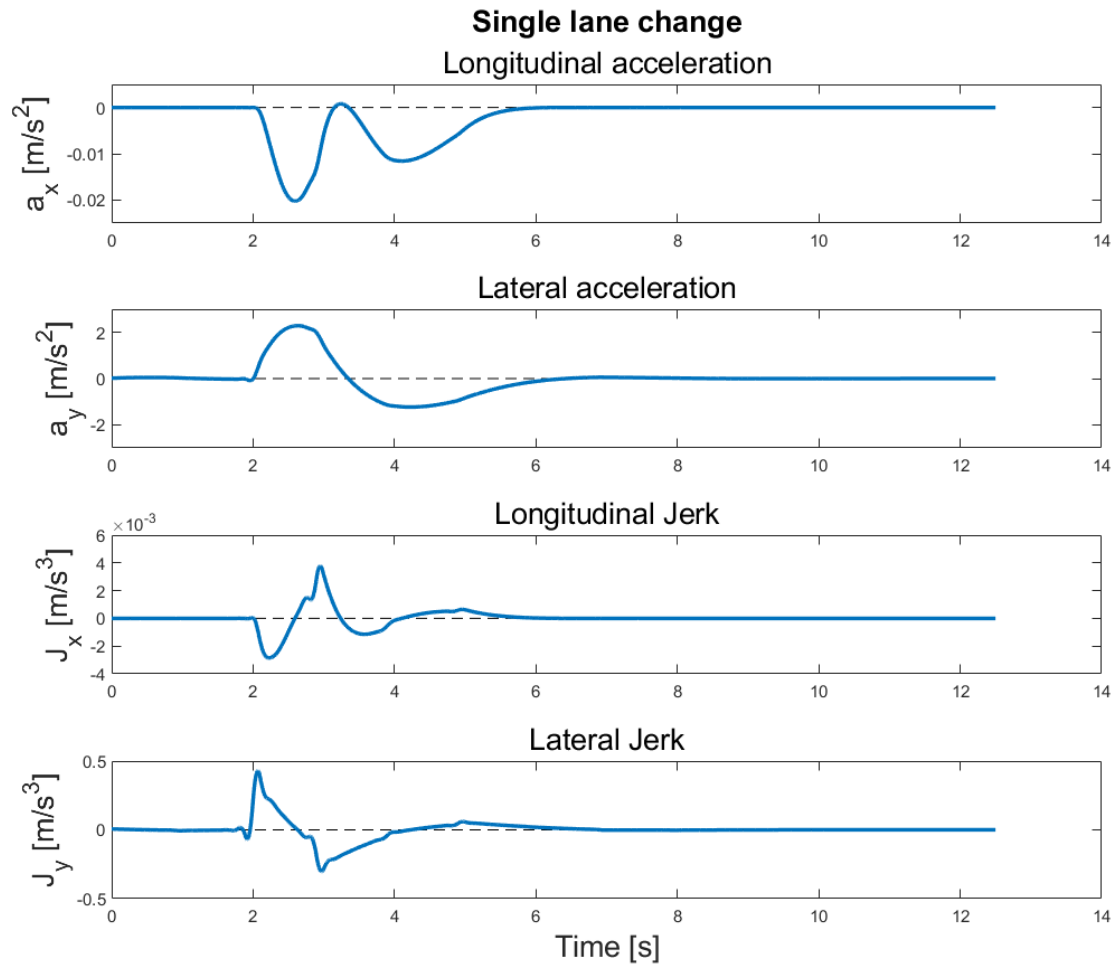


Figura 4.11: Andamenti grandezze moto veicolo *ego* (2) - scenario "Singolo cambio di corsia"

Dopo aver esaminato i valori delle grandezze che caratterizzano il moto del veicolo, risulta essere particolarmente interessante, ai fini di una maggiore comprensione delle logiche di un sistema assistenziale alla guida come quello in esame, analizzare i segnali che vengono valutati dal sistema per registrare la necessità di effettuare un cambio corsia.

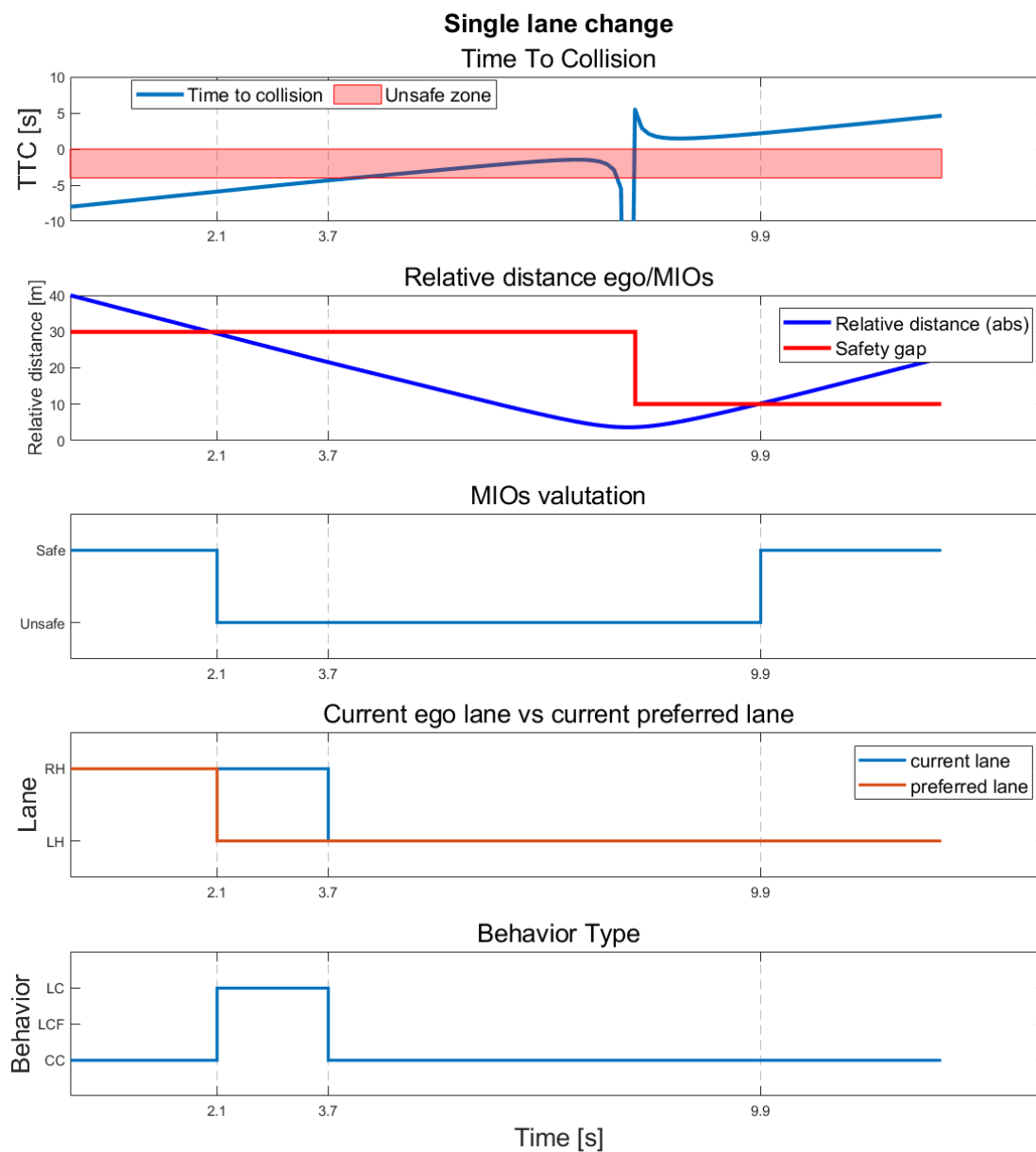


Figura 4.12: Valutazione veicolo MIO - scenario "Singolo cambio di corsia"

Esaminando i grafici riportati in figura 4.12 emerge che:

- Il veicolo MIO viene riconosciuto come *Unsafe* non appena viene oltrepassata la soglia definita in precedenza come distanza di sicurezza, indicata nel secondo grafico come *Safety gap* e rappresentata in rosso. Come si può notare, la linea rossa assume due valori differenti; ciò è dovuto ad un differente limite fra i veicoli che precedono il veicolo *ego* e quelli che lo seguono. La linea rossa, dunque, è stata costruita in modo da rappresentare il corretto limite, istante per istante, tenendo conto della posizione relativa tra veicolo *ego* e veicolo MIO. La logica di attribuzione del corretto limite è stata schematizzata nel diagramma di flusso in figura 3.64, i valori utilizzati come limite sono pari a quelli di default e definiti nella tabella 3.8.
- Nel medesimo istante in cui il valore di distanza relativa tra i due veicoli nello scenario oltrepassa il valore della soglia, riportato in rosso, il veicolo MIO viene dichiarato *Unsafe*, come rappresentato nel terzo grafico (*MIOs valuation*).
- In seguito a questa valutazione del veicolo MIO, chiamato anche veicolo *lead* o veicolo *Obstacle 1*, il pianificatore varia l'indice di *preferred lane* del veicolo *ego*. Il relativo segnale è riportato nel quarto grafico.
- Modificando l'indice di *preferred lane* il pianificatore modifica il comportamento del veicolo *ego*, dove per cambio di comportamento si intende il passaggio dalla modalità *cruise control* (CC) alla modalità *Lane change* (LC), come mostrato in figura 4.6. A ciò segue l'esecuzione della manovra di cambio corsia come si può notare nel grafico che riporta la traiettoria in figura 4.5.
- A tal punto il veicolo *ego* si trova nella corsia di sinistra in seguito alla manovra di cambio corsia. Questa è stata possibile in quanto non sono presenti veicoli MIO dichiarati *Unsafe* nella corsia di sinistra. Questo poiché, per definizione dello scenario, è presente un solo altro veicolo oltre il veicolo *ego*, ed esso mantiene la corsia di destra per tutta la durata della simulazione.

- Il grafico in alto riporta l'andamento del TTC, calcolato per i due veicoli nello scenario con l'equazione 3.70. Si nota che fin quando i due veicoli si trovano nella medesima corsia non viene mai oltrepassata la soglia imposta, il cui valore numerico è pari a quello assegnato di default dal modello e riportato in tabella 3.8. Questa viene oltrepassata esclusivamente quando i due veicoli transitano in corsie differenti.
- Analizzando il grafico si nota che per identificare il veicolo MIO come *Unsafe* valutando il TTC, il valore di questo parametro deve risiedere all'interno della zona rappresentata in rosso. Per dichiarare un veicolo MIO *Unsafe* il TTC deve essere inferiore in modulo alla soglia ed anche negativo; in quanto, essendo negativo, implica la presenza di un progressivo avvicinamento dei due veicoli, indispensabile per ottenere una collisione.
- Si nota che quando il veicolo *ego* ha superato il veicolo MIO, istante che graficamente coincide con il cambio di soglia sulla distanza riportata in rosso nel secondo grafico per i motivi esposti in precedenza, sia la distanza relativa che il TTC assumono un andamento crescente.
- Infine, esaminando il terzo grafico, emerge che negli istanti terminali della simulazione, il veicolo MIO assume nuovamente lo status *Safe*, in quanto entrambi parametri, distanza relativa e TTC, rispettano le condizioni imposte. Questo non interessa il pianificatore durante il mantenimento della modalità di CC, in quanto i veicoli transitano in corsie differenti, ma potrebbe essere un'informazione utile qualora venisse implementato all'interno del modello una strategia di rientro in corsia dopo aver eseguito la manovra di sorpasso.

Confronto tra le differenti modalità CC/LCF/LC

Per mettere in evidenza le caratteristiche del sistema assistenziale, è stato effettuato un confronto tra i risultati prodotti da simulazioni effettuate con il medesimo scenario ma eliminando progressivamente la possibilità di abilitare le modalità implementate. Sono stati esaminati tre differenti casi, il primo permette al sistema assistenziale di poter liberamente modificare il comportamento del veicolo tra le tre modalità, il secondo inibisce l'attivazione del cambio corsia e l'ultimo invece permette al veicolo *ego* di percorrere la propria traiettoria esclusivamente in modalità CC, ovvero viene impedito del tutto l'intervento del sistema assistenziale.

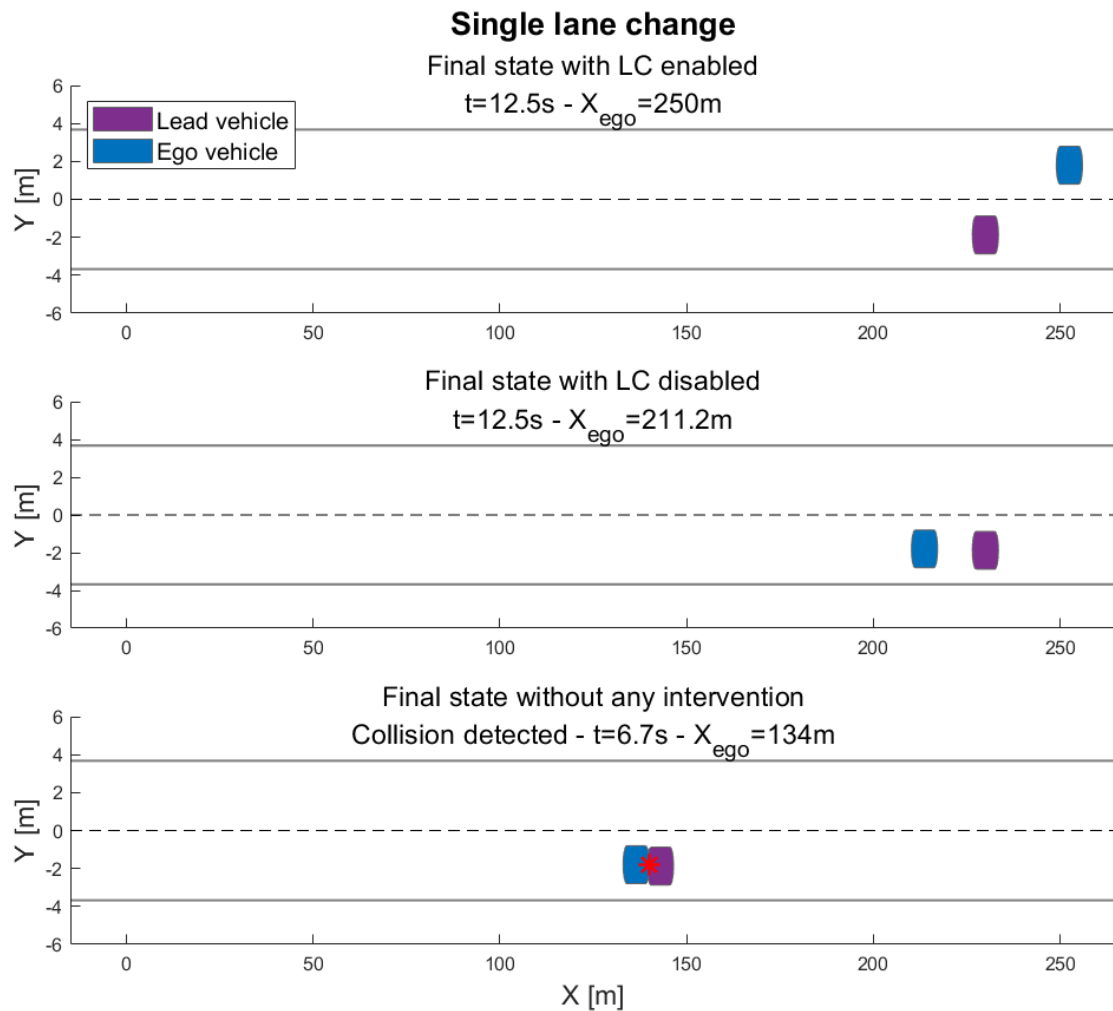


Figura 4.13: Confronto tra stati terminali - scenario "Singolo cambio di corsia"

In figura 4.13 è stato riportato lo scenario nell'istante finale della simulazione per ognuno dei tre casi analizzati. Da tale confronto emerge che:

- Il primo grafico mostra la posizione dei veicoli nel caso in cui tutte le modalità possono essere abilitate; esso è stato ampiamente analizzato precedentemente all'interno di questo paragrafo. Il veicolo *ego* percorre una distanza maggiore rispetto agli altri due casi e non vengono registrate anomalie o collisioni.
- Il secondo grafico mostra la posizione dei veicoli nel caso in cui non sia consentita una variazione di corsia. Si noti che la posizione del veicolo *lead* non varia, poiché definita con lo scenario, a differenza di quella del veicolo *ego*; il quale a causa dell'intervento del sistema assistenziale che abilita la modalità LCF, riduce la velocità del veicolo *ego* fino a che essa non eguaglia quella del veicolo *lead*.
- Il terzo grafico è relativo al caso in cui il veicolo *ego* viene lasciato procedere nel suo moto secondo quanto descritto in fase di definizione dello scenario.; ovvero viene inibito l'intervento di alcun tipo da parte del sistema assistenziale. La simulazione non viene portata a termine (fino a $t=12.5s$), in quanto si verifica una collisione tra i due veicoli.

In figura 4.14 è riportato l'andamento nel tempo della modalità di funzionamento del pianificatore per ognuno dei tre casi analizzati. I commenti relativi al primo caso sono già stati effettuati nel corso della disamina precedente. Per ciò che concerne il secondo caso, si noti che la modalità LCF viene abilitata in un istante successivo a $t=2.1s$; dove si ricorda che $t=2.1s$ è l'istante in cui la distanza tra i due veicoli è inferiore alla soglia *Front safety gap*. Ciò avviene in quanto, a differenza del primo caso, a tale step la modalità CC risulta essere quella con il parametro di costo inferiore. A testimonianza di quanto detto possono essere analizzati i dati riportati nella tabella 4.3, che risultano validi per tutti e tre i casi, in quanto fino all'istante $t=2.1s$ il veicolo *ego* si comporta in maniera analoga nelle tre casistiche.

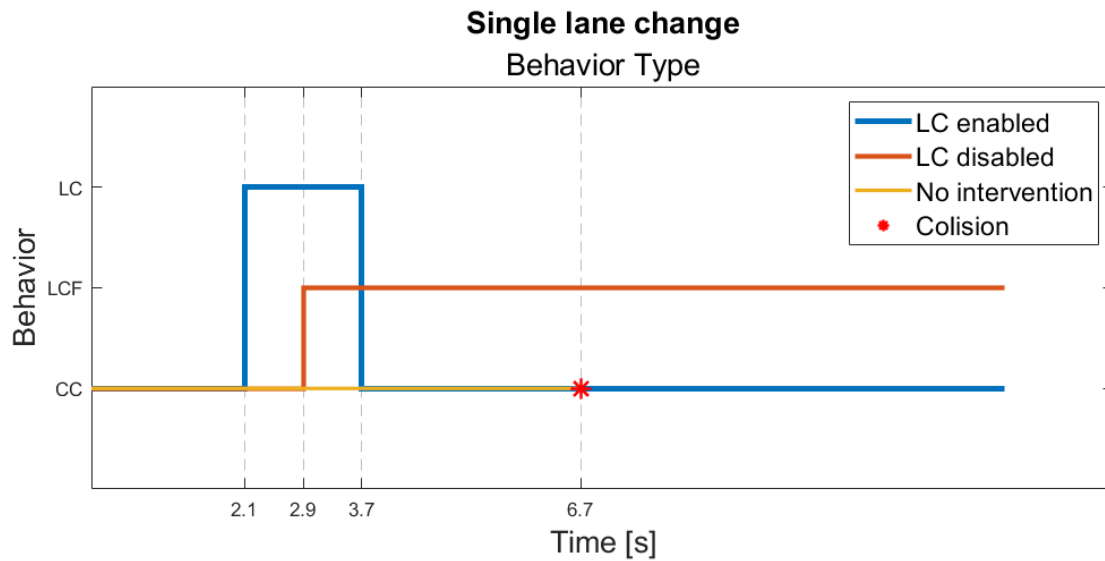


Figura 4.14: Confronto modalità - scenario "Singolo cambio di corsia"

Nella seguente tabella 4.5 è riportato il confronto tra i valori di picco delle grandezze che descrivono il moto del veicolo *ego* per i due casi che non hanno riscontrato collisioni.

	LC disabled	LC enabled	Unit
Max yaw rate	0.00	6.52	$^{\circ}/s$
Max long. acceleration	$9.43 \cdot 10^{-2}$	$2.03 \cdot 10^{-2}$	m/s^2
Max lat. acceleration	0.00	2.29	m/s^2
Max long. jerk	$8.98 \cdot 10^{-3}$	$3.78 \cdot 10^{-3}$	m/s^3
Max lat. jerk	0.00	0.43	m/s^3

Tabella 4.5: Confronto valori di picco - scenario "Singolo cambio di corsia"

Si può facilmente notare che la configurazione che inibisce il cambio corsia porta ad una performance peggiore in termini di distanza percorsa a tutto vantaggio del comfort; in quanto i benefici in termini di velocità angolare, accelerazione laterale e jerk non vengono compensati dai peggioramenti circa le componenti longitudinali di accelerazione e jerk. Tuttavia, il cambio di corsia permette di coprire una distanza superiore rimanendo al di sotto delle soglie limite imposte.

4.3 Caso 2 - Sorpasso con veicolo in senso di marcia opposto

Lo scenario denominato "*Sorpasso con veicolo in senso di marcia opposto*" è stato introdotto ed analizzato all'interno dei capitoli precedenti con il fine di fornire un caso studio su cui testare le funzioni e gli algoritmi introdotti. Se ne riportano di seguito le caratteristiche della strada e si specifica che quest'ultima ha andamento rettilineo.

Lunghezza strada	300m
n°corsie	2

Tabella 4.6: Dati strada - scenario "*Sorpasso con veicolo in senso di marcia opposto*"

All'interno di questo scenario sono presenti, oltre al veicolo ego, due veicoli target. Il primo, identificato come *obstacle vehicle 1* o *lead vehicle*, percorre la strada a velocità costante lungo la stessa corsia del veicolo ego, come nel Caso 1. Il secondo veicolo percorre la strada, anch'esso a velocità costante, ma lungo la corsia con senso di marcia opposto ed è stato nominato *obstacle vehicle 2* o *oncoming vehicle*. Si specifica che tale denominazione, definita durante la descrizione dello scenario all'istante iniziale, verrà mantenuta per tutto il corso della simulazione e per l'analisi dei risultati.

$corsia_{t=0s} \text{ ego}$	2
$corsia \text{ lead}$	2
$corsia \text{ oncoming}$	1
$s_{lead_0} - s_{ego_0}$	40m
$s_{oncoming_0} - s_{ego_0}$	200m
v_{ego}	20m/s
v_{lead}	15m/s
$v_{oncoming}$	10m/s

Tabella 4.7: Dati veicoli - scenario "*Sorpasso con veicolo in senso di marcia opposto*"

Esaminando i dati riportati in tabella 4.7 è necessario specificare che, assumendo il punto di vista del veicolo *ego*, la corsia con l'indice 2 è quella di destra, dove transita il veicolo *lead* ed inizialmente anche il veicolo *ego*; invece, la corsia con senso di marcia opposto, corsia di appartenenza del veicolo *obstacle 2*, è stata identificata con l'indice 1. Inoltre; s_{ego_0} , $s_{oncoming_0}$ e s_{lead_0} rappresentano la coordinata curvilinea dei veicoli all'istante iniziale. Dove per coordinata curvilinea si intende la lunghezza di arco calcolata rispetto ad una curva di riferimento stabilita per il sistema di riferimento adottato; in questo caso, è stata considerata come curva di riferimento quella che rappresenta il centro della carreggiata. Esprimendo il tutto in coordinate globali $[X,Y]$ nel piano XY , la curva di riferimento ha inizio nel punto $[0, 0]$ e termina nel punto $[300, 0]$.

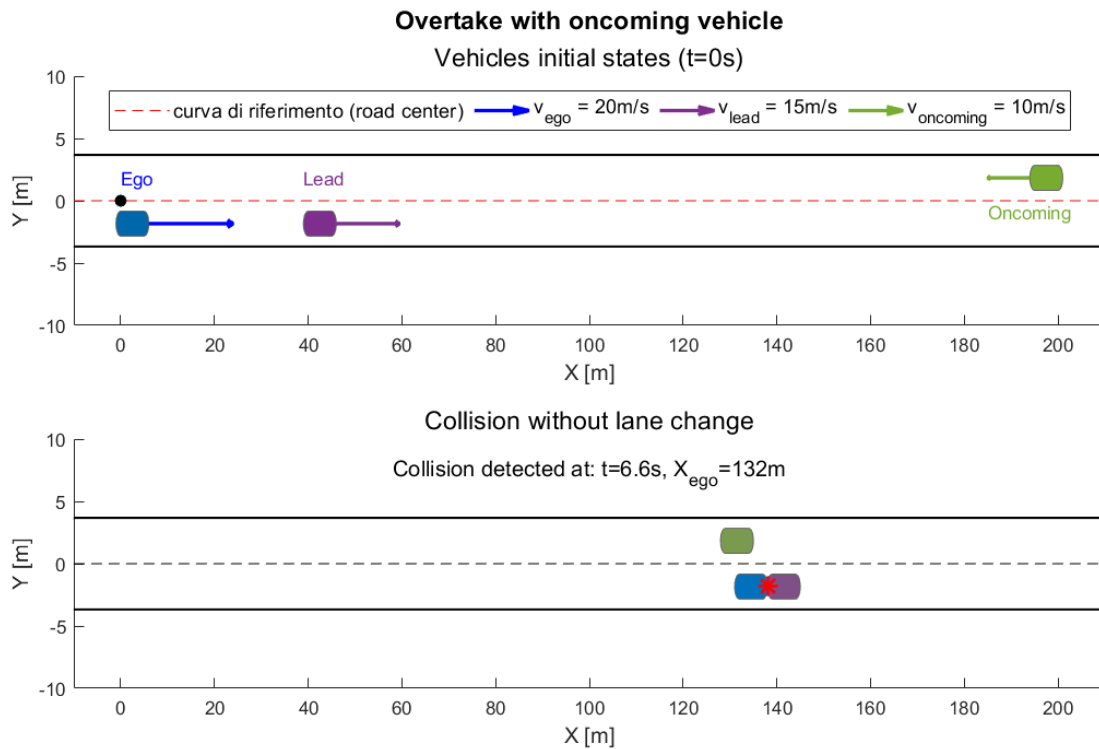


Figura 4.15: Stati iniziali $t=0s$ - scenario "Sorpasso con veicolo in senso di marcia opposto"

In figura 4.15 si ha una rappresentazione grafica dello scenario all'istante iniziale. Sono stati rappresentati i limiti delle due corsie con linee continue nere; la curva di riferimento è stata riportata come linea tratteggiata e di colore rosso, i cui waypoints sono stati indicati da punti neri.

Nel grafico in basso si ha una rappresentazione della collisione in cui si incorrerebbe in assenza di intervento sulla traiettoria del veicolo ego e/o sulla sua velocità.

Questo scenario è stato ideato con l'intento di riprodurre una situazione ad elevata criticità per eseguire una manovra di sorpasso. La criticità è insita nel sopraggiungere di un secondo veicolo *obstacle* (il veicolo *oncoming*), il quale percorre la corsia che il sistema utilizzerebbe come corsia di sorpasso ma con senso di marcia opposto.

Lo scenario è stato successivamente riprodotto all'interno dell'applicazione *Driving scenario designer*, così da ottenere l'input corretto da fornire al modello. La simulazione è stata portata a termine senza registrare collisioni ed il pianificatore ha valutato come alternativa migliore per il veicolo ego, in relazione allo scenario inserito, quella di effettuare un sorpasso ai danni del veicolo *lead*. Tuttavia, la traiettoria individuata non rispetta alcuni fra i limiti imposti; conseguentemente non può essere ritenuta una soluzione valida. I risultati della simulazione sono stati esaminati nel paragrafo successivo; a valle di tali analisi sono state analizzate le criticità emerse, proponendo infine una soluzione in grado di individuare una traiettoria valida.

4.3.1 Risultati simulazione - caso 2

Come anticipato, la simulazione è stata portata a termine in totale assenza di collisioni tra veicolo *ego* e gli altri due veicoli dello scenario. In seguito al rilevamento di un'incipiente collisione, in caso di mantenimento della traiettoria descritta all'interno dello scenario, il pianificatore è dovuto intervenire sugli stati del veicolo *ego* modificandone le traiettoria. Si riporta la traiettoria percorsa dal veicolo nel corso della simulazione in figura 4.16.

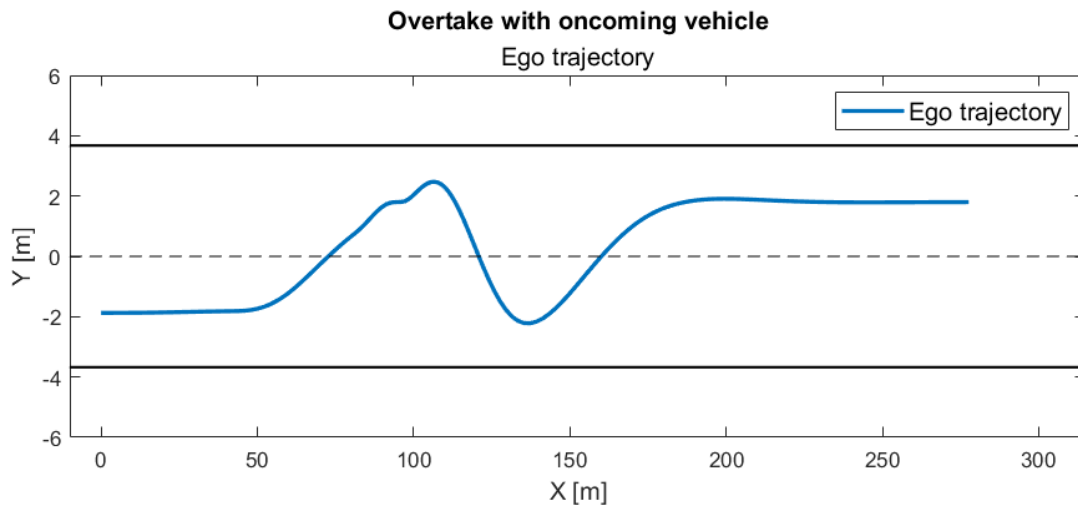


Figura 4.16: Traiettoria veicolo *ego* - scenario "Sorpasso con veicolo in senso di marcia opposto"

Si nota immediatamente che la traiettoria individuata non risulta essere ottimale, in quanto per essere realizzata richiede un triplo cambio di corsia; il quale non è in alcun modo necessario, poiché il veicolo *ego* si sposta nella corsia di sorpasso per poi ritornare nella corsia precedente, senza però aver effettivamente eseguito la manovra di sorpasso ai danni del veicolo *lead*.

Per una migliore comprensione dei risultati della simulazione, in figura 4.17 sono state sovrapposte le rappresentazioni dello scenario mediante capsule negli istanti di maggior rilievo. I segmenti di traiettoria prodotti dal pianificatore sono stati rappresentati in verde.

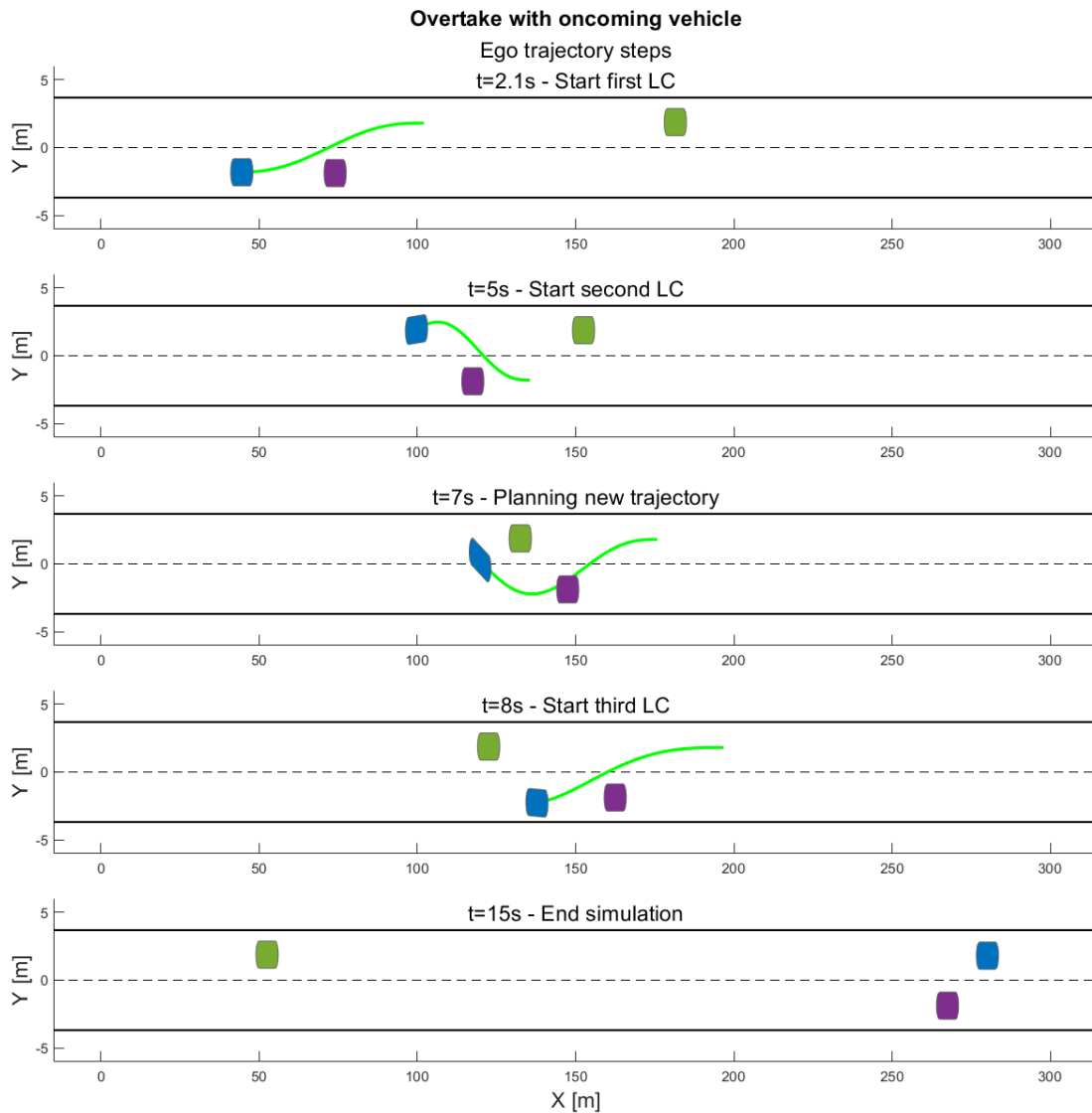


Figura 4.17: Frame traiettoria veicolo *ego* - scenario "Sorpasso con veicolo in senso di marcia opposto"

Oltretutto questa traiettoria non risulta effettivamente percorribile dal veicolo in quanto vengono oltrepassati i limiti di fattibilità imposti; si ha un riscontro numerico e grafico delle grandezze che caratterizzano il moto del veicolo nella tabella 4.8 e nelle figure 4.18 e 4.19. Nello specifico vengono oltrepassati i limiti circa la velocità di imbardata e l'accelerazione laterale in due differenti momenti della simulazione, nell'intorno dei 4.9 s e nell'intorno dei 7.7 s.

Parameter	Max	Limit	Unit
Yaw rate	26.99	20	$^{\circ}/s$
Long. Acceleration	0.83	5	m/s^2
Lat. Acceleration	7.84	5	m/s^2
Long. Jerk	0.14	5	m/s^3
Lat. Jerk	2.53	5	m/s^3

Tabella 4.8: Valori di picco grandezze veicolo *ego* - scenario "Sorpasso con veicolo in senso di marcia opposto"

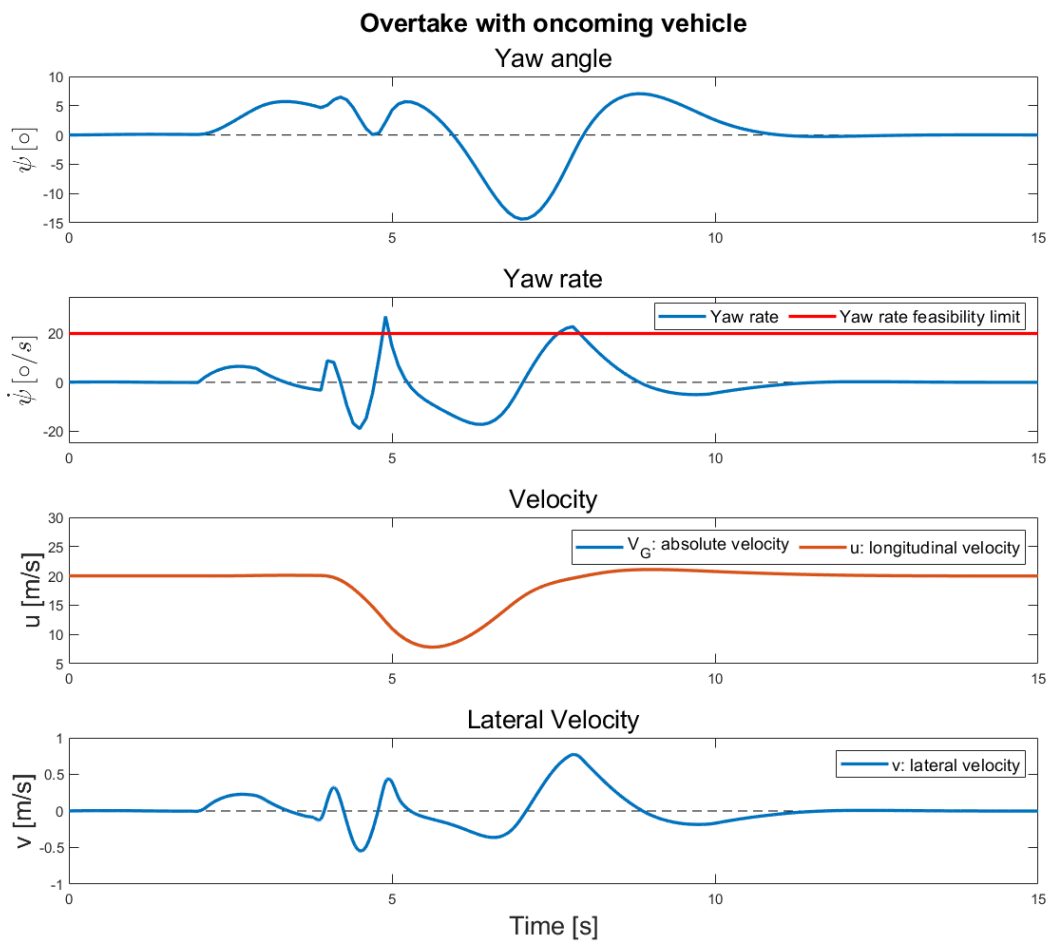


Figura 4.18: Andamenti grandezze moto veicolo *ego* - scenario "Sorpasso con veicolo in senso di marcia opposto"

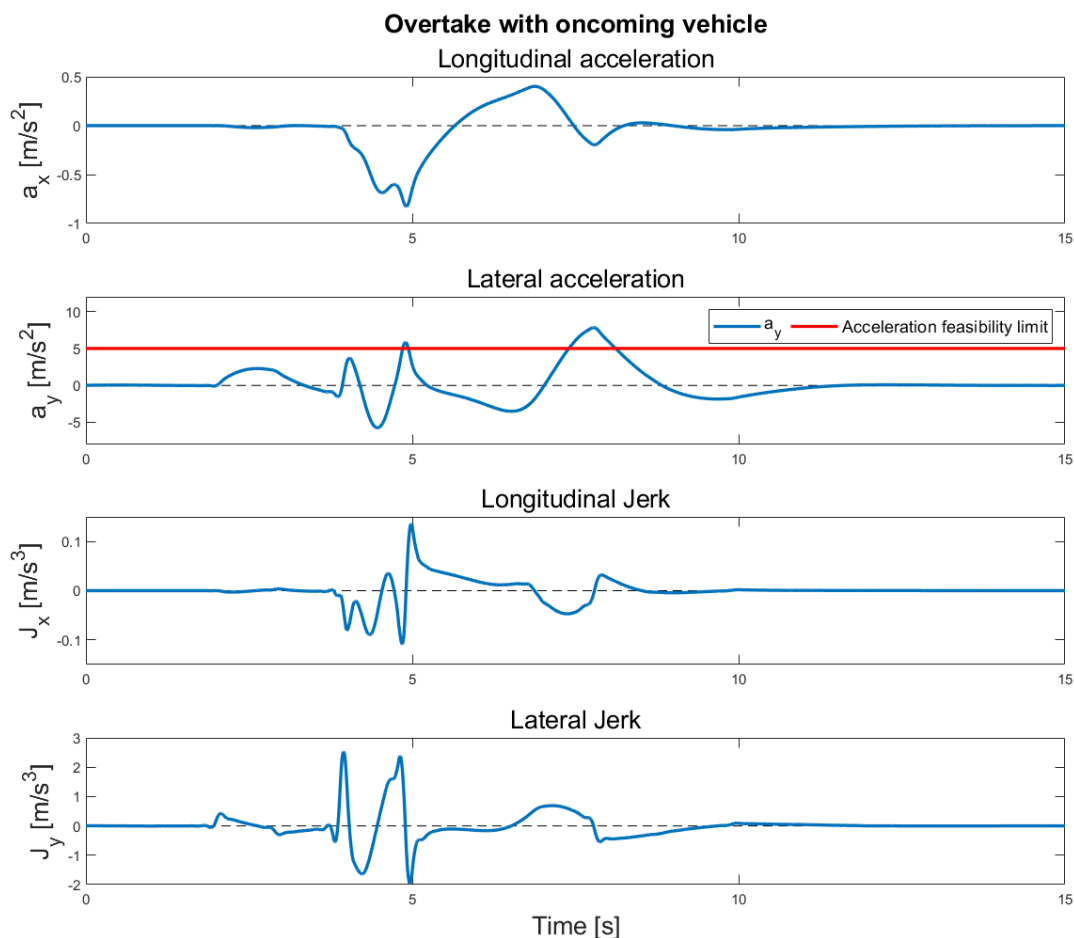


Figura 4.19: Andamenti grandezze moto veicolo *ego* (2) - scenario "*Sorpasso con veicolo in senso di marcia opposto*"

Analizzando con maggior dettaglio i risultati riportati, emerge che le due zone della traiettoria che non rispettano i limiti imposti sull'accelerazione laterale e sulla velocità di imbardata corrispondono ai repentini cambi di traiettoria necessari ad evitare collisioni con gli altri due veicoli presenti nello scenario. Per rendere la spiegazione più efficace, i punti "non validi" secondo i parametri di fattibilità imposti sono stati riportati graficamente sulla traiettoria in figura 4.20.

In prossimità di un'ascissa pari a 100 m, si nota come il veicolo *ego*, ormai completata la manovra di cambio corsia, avendo quindi raggiunto il centro della stessa con angolo di imbardata nullo, è costretto ad operare una manovra di emergenza per evitare lo scontro con il veicolo precedentemente denominato *oncoming*.

Nel corso di questa manovra la variazione di angolo di imbardata è talmente brusca da oltrepassare il limite imposto e da gravare sul computo dell'accelerazione laterale a_y portando anche il valore di quest'ultima oltre la soglia.

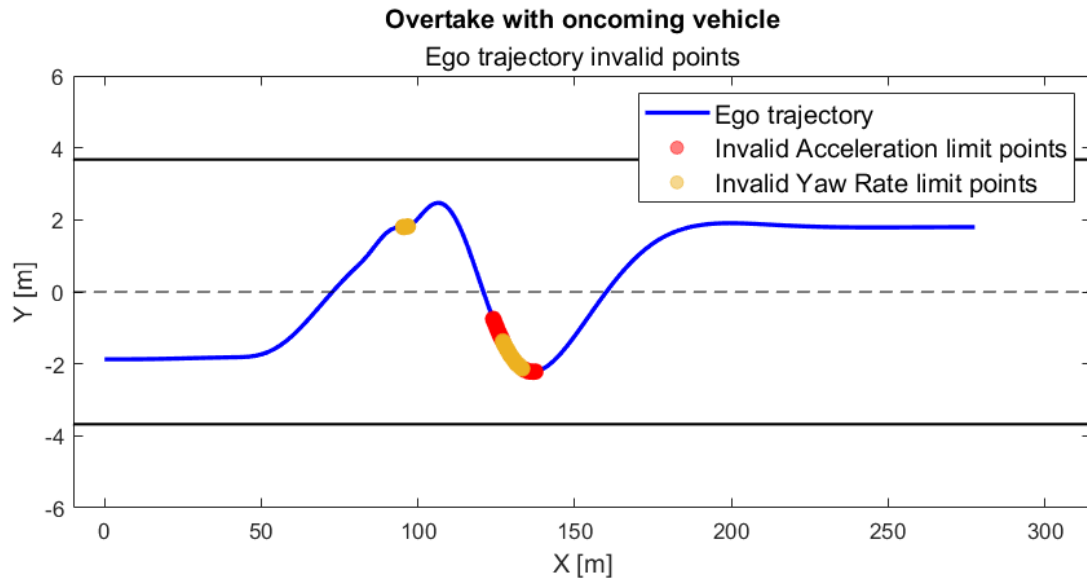


Figura 4.20: Punti della traiettoria invalidanti i limiti imposti - scenario "Sorpasso con veicolo in senso di marcia opposto"

Osservando i grafici dell'angolo di imbardata e della velocità di imbardata in figura 4.18, si nota che:

- Dopo aver eseguito questa prima manovra evasiva il veicolo *ego* si dirige verso la corsia del proprio senso di marcia, corsia sulla quale transita il veicolo *lead*, raggiungendo nell'istante in cui sono trascorsi 7.7 s dall'avvio della simulazione un minimo nell'andamento dell'angolo di imbardata, a cui si associa un valore pari a -14.37° .
- Dopo soli 1.1 s, all'istante della simulazione pari a 8.8 s, il veicolo *ego* ha un angolo di imbardata pari a 7.05° , nonché il massimo dell'andamento della grandezza esaminata.
- Questo dato implica che il veicolo, all'istante pari a 8.8 s è coinvolto nel suo terzo cambio di corsia, con il fine ultimo di portare a termine il sorpasso ai danni del veicolo *lead*.

- Raggiungendo effettivamente lo scopo negli istanti successivi della simulazione, questo rapido cambio di direzione porta però ad invalidare ancora una volta i limiti circa la velocità di imbardata e della connessa accelerazione laterale a_y .

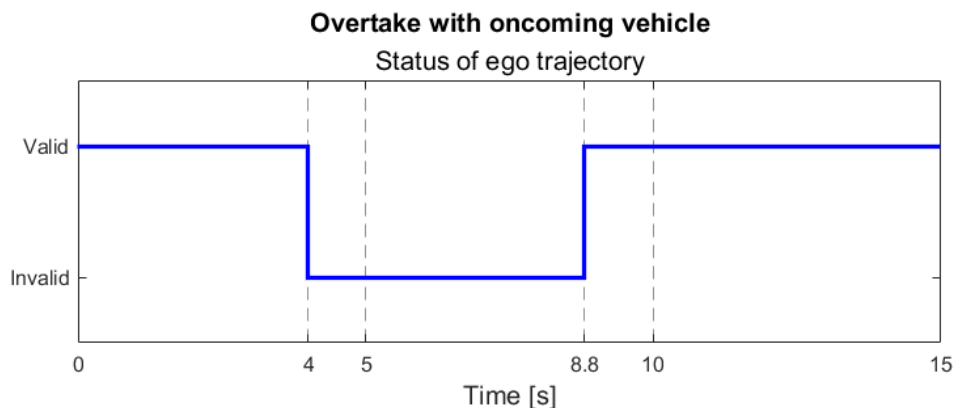


Figura 4.21: Status veicolo *ego* - scenario "*Sorpasso con veicolo in senso di marcia opposto*"

In seguito alle considerazioni introdotte si può commentare il grafico in figura 4.21. Esso riporta in forma grafica il segnale dello status del veicolo *ego* (*Valid/Invalid*). Per ottenere una simulazione con esito positivo, dunque una traiettoria effettivamente percorribile dal veicolo *ego*, è necessario che il segnale riportato sia per ogni istante temporale pari all'ascissa *Valid*. Tuttavia, per i motivi spiegati in precedenza, non è questo il caso in esame. Si nota però come gli istanti di inizio e fine status *Invalid* risultano essere rispettivamente precedenti e successivi ai due istanti temporali ricavati in precedenza, pari a 4.9 s e 7.7 s, circa l'effettivo superamento dei limiti legati alle grandezze descrittive del moto del veicolo. Questo è dovuto alla logica di costruzione del pianificatore, in quanto esso elabora una traiettoria da far seguire al veicolo *ego* a partire dall'istante corrente fino ad un successivo orizzonte temporale, per poi ricalcolarla se necessario o se viene raggiunto tale orizzonte temporale. Conseguentemente, la valutazione del set di alternative viene effettuata nel momento in cui si effettua la pianificazione del successivo tratto di traiettoria; se un punto della traiettoria migliore individuata risulta essere non valido allora il veicolo viene dichiarato *Invalid* dal momento stesso della pianificazione, fin quando non viene individuata una traiettoria *Valid* da seguire.

In figura 4.22 si ha una rappresentazione grafica dell'indice di corsia ottimale ed ideale, per ciò che concerne la traiettoria del veicolo *ego*; nel grafico in basso è rappresentato l'andamento della modalità di funzionamento del sistema nel corso della simulazione.

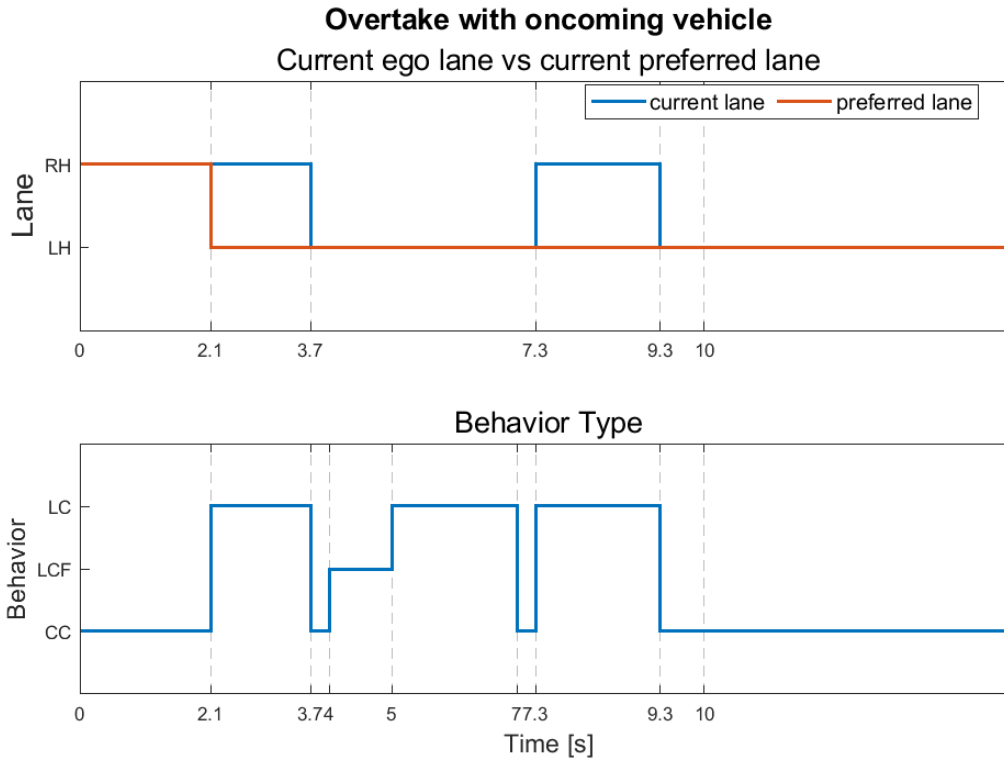


Figura 4.22: Corsia veicolo *ego* - scenario "Sorpasso con veicolo in senso di marcia opposto"

Come si nota dal grafico in alto in figura 4.22 dall'istante pari a 2.1 s la corsia di preferenza risulta essere quella di sorpasso, la corsia di sinistra, in seguito al rilevamento di un'incipiente collisione con il veicolo *lead* in mancanza di una variazione del moto del veicolo *ego* rispetto a quanto descritto nello scenario. Ciò è dovuto a due traiettorie coincidenti fra i due veicoli e ad una maggiore velocità del veicolo *ego* rispetto a quella del veicolo *lead*. La corsia di preferenza viene mantenuta uguale per tutto il resto della simulazione, al contrario della corsia corrente, in quanto il veicolo *ego* è costretto nuovamente a rientrare nella corsia di destra per evitare la collisione con il veicolo *oncoming*.

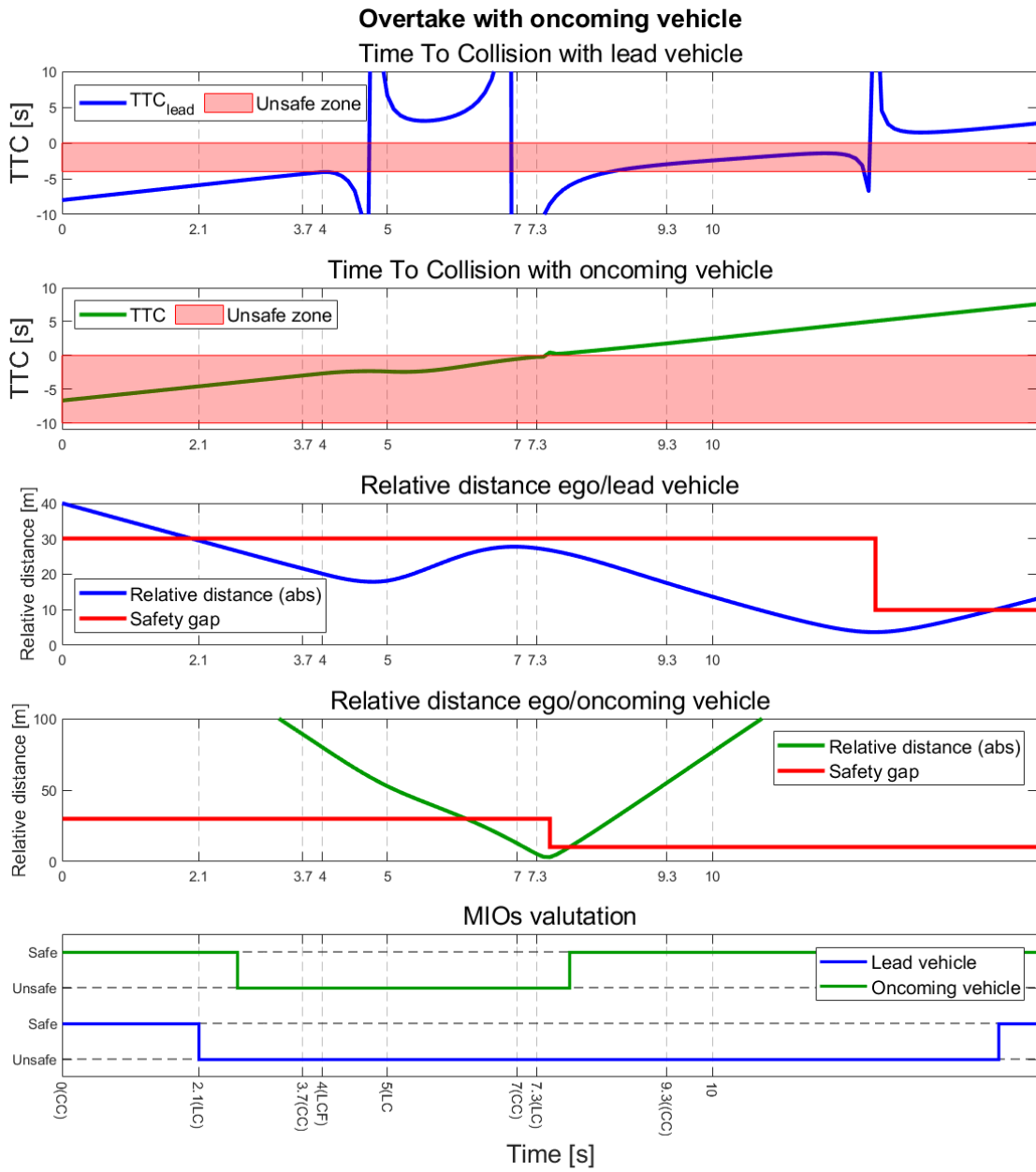


Figura 4.23: Valutazione veicoli MIO - scenario "Sorpasso con veicolo in senso di marcia opposto"

Si procede di seguito con l'analisi degli andamenti delle principali grandezze utilizzate dal pianificatore durante la fase di simulazione, riportati graficamente in figura 4.23. L'analisi verrà eseguita in ordine cronologico, commentando gli istanti di maggior rilievo. Con il fine di rendere più comprensibile tale disamina, nell'asse delle ascisse sono state riportate le modalità di funzionamento del pianificatore; il relativo andamento può essere analizzato anche mediante la rappresentazione grafica in figura 4.22.

- Il primo tratto è quello di funzionamento in modalità CC, in quanto il veicolo segue la traiettoria descritta all'interno dello scenario.
- Dopo 2.1s si passa in modalità LC, poiché viene dichiarato come *Unsafe* il veicolo *lead*. Questo accade poiché viene valicato il limite circa la *Relative Distance*, come si può apprezzare nel terzo grafico in figura 4.23. Come si nota dall'ultimo grafico, denominato "MIOs valuation", il veicolo *oncoming*, transitante lungo la corsia di sorpasso, è considerato *Safe* in tale istante. Il segmento di traiettoria pianificato può essere visionato nel primo grafico in figura 4.17.
- Il sorpasso viene eseguito, ed una volta che il veicolo *ego* si trova nella corsia di sinistra viene abilitata la funzione CC, utile a sorpassare il veicolo *lead* lungo la coordinata curvilinea di riferimento *s*.
- Durante il funzionamento in modalità LC, dunque durante la manovra di cambio corsia, il veicolo *oncoming* riduce progressivamente la propria distanza dal veicolo *ego*; in quanto, come si può notare dal secondo grafico in figura 4.23, il *TTC* risulta essere negativo. Il *TTC* si riduce progressivamente fin quando non viene oltrepassata la soglia, rappresentata in rosso e con il nome di *Unsafe zone*, ed il veicolo *oncoming* viene dichiarato *Unsafe*. Questo accade all'istante pari a 2.6 s, ovvero prima del completamento della manovra di sorpasso.
- Nel momento in cui il veicolo *oncoming* viene dichiarato *Unsafe* entrambi i veicoli, come si nota dall'ultimo grafico, non sono *Safe*. La manovra di cambio corsia iniziata nell'istante $t=2.1s$ viene completata e viene abilitata la modalità CC.

- Quella presentata al punto precedente risulta essere un'invalidante criticità; in quanto, se il veicolo *oncoming* fosse stato riconosciuto come non sicuro prima dell'inizio della manovra di cambio corsia, quest'ultima non sarebbe stata effettuata e non sarebbero state necessarie le successive manovre evasive descritte in precedenza.
- Nell'istante pari a 4 s viene abilitata la funzione LCF; tuttavia, nel medesimo istante la traiettoria viene dichiarata *Invalid*, come si nota dal grafico in figura 4.21. Quando la traiettoria da seguire è valutata non valida, ciò implica che tutti gli stati terminali portano a sviluppare traiettorie considerate non valide. Conseguentemente, viene individuata tale modalità (LCF) come quella da seguire in quanto la modalità CC porta prima alla collisione, essendo la velocità del veicolo *ego* maggiore di quella del veicolo *oncoming*. Si nota infatti una progressiva riduzione della velocità nel grafico in figura 4.18. Si ricorda che in modalità CC il veicolo *ego* adotta una velocità pari a quella inserita durante la descrizione dello scenario, 20 m/s in questo caso; invece, in modalità LCF, all'istante finale della traiettoria pianificata adotta una velocità uguale a quella del veicolo che lo precede lungo la medesima corsia, in tal caso il veicolo *oncoming*, il quale ha una velocità pari a 10 m/s e negativa secondo il dato sistema di riferimento. Come anticipato, una velocità del veicolo *ego* inferiore implica una *Relative Velocity* inferiore, dunque un *TTC* maggiore a parità di altre condizioni; è dunque questo il motivo per cui viene abilitata la modalità LCF.
- Come si nota dalla descrizione riportata nel punto precedente, il sistema assistenziale abilita la modalità LCF riconoscendo come veicolo da seguire il veicolo *oncoming*, il quale ha senso di marcia opposto al primo, semplicemente perché precede il veicolo *ego* nella medesima corsia. Questo è errato ed è un problema risolto con la soluzione adottata in seguito e presentata nel sottoparagrafo successivo.

- Nell'istante $t=5s$ si passa nuovamente alla modalità LC per evitare la collisione con il veicolo *oncoming*, fin quando si raggiunge la corsia di destra e viene attivata nuovamente la modalità CC. Tuttavia, è necessario sottolineare nuovamente che questo cambio di corsia viene effettuato prima di aver superato il veicolo *ego* rispetto alla coordinata curvilinea di riferimento s ; dunque, quando viene portato a termine il rientro in corsia il veicolo *lead* continua a precedere il veicolo *ego*, rendendo di fatto inutili le manovre svolte in precedenza. Questo risulta essere un problema del sistema sul quale è necessario intervenire. Il segmento di traiettoria pianificato può essere visionato nel secondo grafico in figura 4.17.
- Nell'istante $t=7s$ viene pianificato un nuovo segmento di traiettoria in modalità CC; questo poiché, come si può vedere dal terzo grafico in figura 4.17, il primo e l'ultimo punto della polinomiale del quinto ordine sono entrambi nella medesima corsia. Tuttavia nell'istante $t=7.3s$ il veicolo oltrepassa la linea di mezzzeria, passando così in modalità LC, in quanto la corsia corrente e quella dello stato terminale del segmento sono differenti.
- Nell'istante $t=8s$ il veicolo *ego* si trova nella corsia di destra, viene pianificata una nuova traiettoria in modalità LC per evitare una collisione con il veicolo *lead*. Il segmento di traiettoria pianificato può essere visionato nel quarto grafico in figura 4.17.
- Negli istanti successivi della simulazione viene eseguito il sorpasso e nuovamente abilitata la modalità di CC. Possibile in quanto il veicolo *oncoming* è transitato, dunque non più in avvicinamento e considerato *Safe*. Entrambi i veicoli vengono dichiarati *Safe* solo a partire dall'istante $14.3 s$, ovvero quando il veicolo *ego* supera, rispetto la coordinata curvilinea di riferimento s , il veicolo *lead*.

4.3.2 Adattamento del sistema alle criticità presentate - Caso 2

A valle delle criticità emerse dai risultati della simulazione, opportunamente analizzati nel sottoparagrafo precedente, è stata sviluppata una strategia in grado di adattare la risposta del sistema assistenziale avanzato alla guida a scenari simili a quello esposto in questo caso studio.

Esposizione del problema

L'analisi precedente ha portato all'individuazione del problema che porta all'impossibilità nell'identificazione di una traiettoria valida ed ottimale per tale caso studio; il quale si sviluppa con la concomitanza degli scenari descritti nei seguenti punti:

- Il sistema assistenziale avanzato alla guida di cui è dotato il veicolo *ego* definisce come *Unsafe* il veicolo *oncoming* dopo aver già eseguito la prima manovra di cambio corsia; ovvero, solo quando il veicolo *ego* è già transitante lungo la corsia di sorpasso, quella di sinistra.
- Il limiti sul tempo alla collisione *TTC* e sulla distanza relativa *Relative Distance*, sulla base dei quali viene effettuata la valutazione sulla sicurezza di un dato veicolo *Safe/Unsafe*, risultano essere non compatibili con il rientro in corsia ed un concomitante completamento della manovra di sorpasso ai danni del veicolo *lead*. L'incompatibilità nasce dal fatto che tali limiti risultano essere calibrati per il confronto tra la traiettoria del veicolo *ego* e quella di altri veicoli che percorrono la carreggiata con il medesimo senso di marcia.

Le due caratteristiche presentate devono coesistere per incorrere in una criticità tale da portare a risultati non validi, come quelli del caso in esame. Questo è necessario poiché il sorpasso può effettivamente essere portato a termine anche se durante tale manovra sopraggiungesse un veicolo nella medesima corsia e con senso di marcia opposto, purché le distanze siano compatibili con il superamento del veicolo *lead* ed con il successivo rientro in corsia.

Tuttavia, per permettere ciò dovrebbe venire a mancare la seconda condizione; ovvero, sarebbe necessario adottare dei limiti circa il *TTC* e la *Relative Distance* superiori, in modo da comunicare ad una *Relative Distance* ed un *TTC* maggiori l'esigenza di modificare la traiettoria. Ricordando che tale esigenza nasce dalla presenza di un veicolo *Unsafe* nella medesima corsia del veicolo *ego* e che i veicoli dello scenario vengono dichiarati *Unsafe* se valicano i limiti di *TTC* e *Relative Distance*.

In tal modo, anche se la definizione di "veicolo *Unsafe*" circa il veicolo *oncoming* avvenisse già a sorpasso avviato, come nel caso in esame, si avrebbero il tempo e la distanza necessari a portare a termine la manovra, in quanto il pianificatore varierebbe la traiettoria del veicolo *ego* con sufficiente anticipo. Tuttavia, quella che apparentemente può sembrare una soluzione, ovvero la modifica dei parametri citati, non risulta essere una strada efficace per i motivi spiegati in seguito.

Soluzione adottata

Per risolvere il problema presentato, l'obiettivo posto è stato quello di consentire al sistema di riconoscere la presenza di una simile criticità, fornendo una risposta sempre solida, ovvero non dipendente dalla calibrazione numerica dei parametri di configurazione caso per caso. Conseguentemente, è stata sviluppata una strategia sulla base dei principi esposti successivamente.

Per una migliore comprensione di quanto segue, si ricorda che la traiettoria da seguire fino al prossimo orizzonte temporale viene scelta da un set di alternative in base ad una data funzione di costo. Inoltre, tale set di alternative viene calcolato sulla base dei cosiddetti "stati terminali"; ovvero, partendo dalla posizione corrente del veicolo *ego*, il punto in cui lo si vuole collocare al dato orizzonte temporale ad ogni specifica alternativa. Dove per orizzonte temporale si intende l'istante t in cui termina il tratto di traiettoria pianificato; istante t calcolato addizionando un Δt , espresso in secondi, al "*current time*". Per ognuno di questi stati terminali, i quali si differiscono sia per l'orizzonte temporale che per la modalità di funzionamento (CC, LCF o LC), verrà calcolato un segmento di traiettoria.

Sulla base di quanto detto, la strategia è stata sviluppata in modo da:

1. Riconoscere la criticità imminente mediante l'individuazione di eventuali veicoli *oncoming*. Ciò è stato possibile implementando un'analisi delle grandezze che descrivono il moto dei veicoli presenti nello scenario e definiti MIO.
2. Basare il processo di cui al punto precedente su parametri dedicati, non quindi utili ad altre funzionalità del sistema assistenziale. Questo è stato ricercato in quanto il problema si sarebbe potuto risolvere aumentando il valore numerico dei limiti circa il *TTC* e la *Relative Distance*; tuttavia, tale soluzione avrebbe compromesso la pianificazione di traiettorie ottimali in scenari comuni come quello presentato nel "Caso 1". Dunque, è stato scelto di differenziare i limiti relativi ai veicoli *oncoming* rispetto a quelli del veicolo *lead*, in modo da evitare che la manovra di sorpasso ai danni di quest'ultimo abbia inizio con eccessivo anticipo, incrementando inutilmente la permanenza del veicolo *ego* nella corsia con senso di marcia opposto.
3. Stabilire un range in cui la manovra può comunque essere effettuata in sicurezza se presente un veicolo nella medesima corsia, con senso di marcia opposto ed in avvicinamento; ovvero quando quest'ultimo ha una distanza o una velocità compatibili con la manovra da eseguire in sicurezza.
4. Agire a monte dell'intero processo di pianificazione; ovvero, evitando che gli stati terminali che prevedono la presenza del veicolo *ego* in una corsia in cui è transitante un veicolo "*oncoming*", il quale non rientra nei parametri imposti per dichiarare la manovra sicura, partecipino alle successive fasi della pianificazione di traiettoria, quindi depennandole dal set di alternative.

In figura 4.24 è stato riportato un digramma di flusso in cui sono schematizzati i passaggi logici eseguiti dalla funzione sviluppata. All'interno dei vari blocchi sono espressi in forma pratica i principi appena esposti.

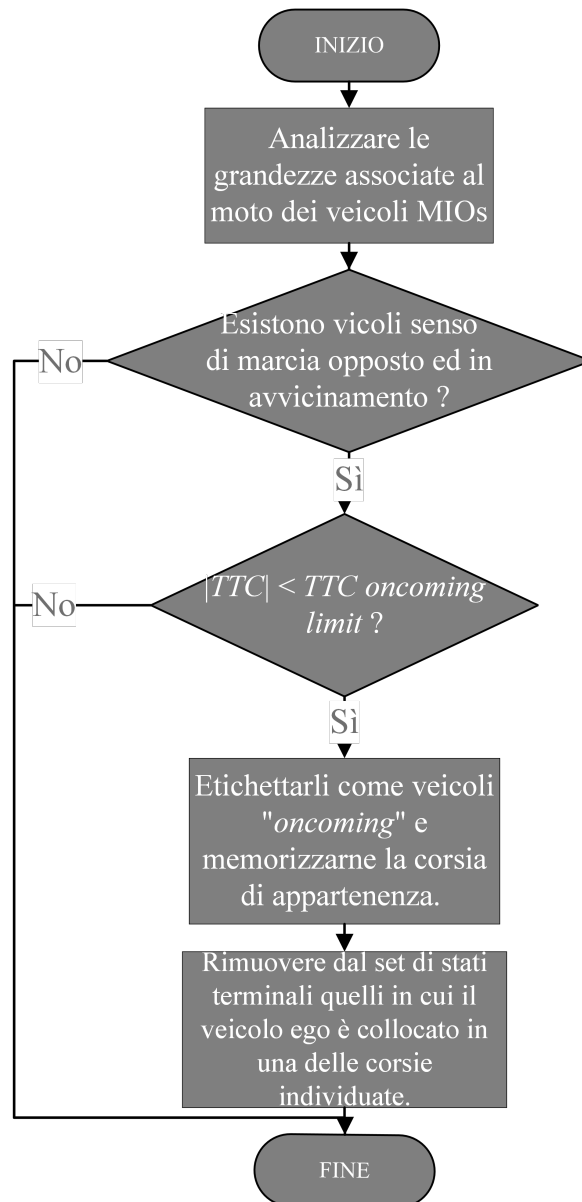


Figura 4.24: Diagramma di flusso funzione "oncoming Check" - scenario "Sorpasso con veicolo in senso di marcia opposto"

Per riconoscere l'eventuale presenza di criticità come quella del caso presentato è innanzitutto necessario verificare la presenza di veicoli con senso di marcia opposto. In tal senso sono stati sfruttati i segnali contenenti gli stati dei veicoli presenti nello scenario in coordinate di Frenét. Entrando maggiormente nello specifico, è stata confrontata la velocità dei veicoli esaminati \dot{s} con quella del veicolo *ego*; qualora il segno del valore analizzato fosse di segno opposto a quello del rispettivo parametro del veicolo *ego*, ciò implicherebbe una direzione di marcia opposta.

$$\frac{\dot{s}_{ego}}{\dot{s}_{MIO}} < 0 \implies \text{senso di marcia opposto} \quad (4.3)$$

Si è proseguito con l'introduzione di una valutazione necessaria all'esclusione di tutti quei veicoli che si stanno progressivamente allontanando, e non avvicinando, al veicolo *ego*. In tal senso è stato sapientemente sfruttato un segnale già esistente, ovvero quello del *TTC*; ricordando che un *TTC* minore di zero implica una condizione di progressivo avvicinamento tra due veicoli, come dimostrato in precedenza.

$$TTC < 0 \implies \text{veicolo in avvicinamento} \quad (4.4)$$

Tra i veicoli che rispettano entrambe le condizioni precedenti è stato necessario escludere i veicoli che non implicano un rischio per la manovra di sorpasso seppur in avvicinamento e transitanti nella medesima corsia ma con senso di marcia opposto. Tale esclusione è imprescindibile per consentire al sistema di operare cambi di corsia in condizioni in cui il veicolo *oncoming* risulta essere sufficientemente "lento" o "distante". Tale valutazione è stata effettuata sul parametro relativo al tempo alla collisione *TTC*; in quanto si tratta di un valore che tiene conto sia della distanza relativa che della velocità relativa tra i due veicoli. In tal senso il valore soglia è stato denominato $TTC_{oncoming}$ ed è un parametro espresso in secondi. In valore numerico adottato è pari a 10 s; esso è stato scelto arbitrariamente ma risulta essere compatibile con i tempi necessari al sistema per effettuare un cambio corsia.

$$|TTC| < TTC_{oncoming} \implies \text{criticità} \quad (4.5)$$

Come anticipato, in caso di rilevamento di tale criticità è stato scelto di agire a monte dell'intero sistema di pianificazione. Nello specifico, si eliminano tutti gli stati terminali che implicano la presenza del veicolo *ego* nella medesima corsia di tutti quei veicoli che rispettano tutte le condizioni precedentemente esposte nelle equazioni 4.3, 4.4 e 4.5.

I segnali di input della *function* creata per assolvere il compito esposto risultano essere gli stati terminali relativi alle modalità di CC, LCF e LC, il segnale contenente le informazioni relative alla strada, gli stati dei veicoli *ego* e MIO, ed infine il parametro $TTT_{oncoming}$. Si riportano di seguito i passaggi logici eseguiti all'interno della funzione.

```
function [CruiseStatesCheck,LCFTerminalStatesCheck,LCStatesCheck] = ...  
...oncomingCheck(CruiseStates, LCFTerminalStates, LCStates, road,...  
...mioFrenetStates, EgoAndTargetStates, limitTTC);
```

1. Si inizializzano le strutture in output *CruiseStatesCheck*, *LCFTerminalStatesCheck*, *LCStatesCheck*, le quali conterranno gli stati terminali che hanno superato la verifica circa i veicoli *oncoming*. Vengono inizializzati con la struttura *initStruct*, la quale viene definita richiamando la "funzione helper" *HelperLCPlannerDefaultData*. Inizializzando tali strutture viene associato un numero pari a zero di stati terminali.
2. Se presenti veicoli MIO nello scenario, per ognuno di essi si memorizza nel vettore "mioInfo.TTC" il relativo *TTC* con il veicolo *ego*; il quale viene ottenuto utilizzando la "funzione helper" *helperCalculateTTC*.
3. Se il veicolo rispetta le condizioni riportate nelle equazioni 4.3, 4.4 e 4.5, viene memorizzata la corsia del veicolo *MIO* nel vettore "oncomingLane". Essa a sua volta viene ricavata utilizzando la "funzione helper" *helperDetectLaneNumber*.
4. Si analizzano gli stati terminali in input ed attraverso la "funzione helper" *helperDetectLaneNumber* e se ne ricava la corsia del veicolo *ego*. Qualora l'indice di corsia di un dato stato terminale coincidesse con un elemento presente nel vettore "oncomingLane" precedentemente ricavato, per quella data modalità verrebbe fornito in output il vettore "initStruct", contenente un

numero di stati terminali nullo. Si ricorda che gli stati terminali di una data modalità per costruzione, hanno tutti la medesima coordinata d ; ovvero per una data modalità il veicolo *ego* transita lungo la medesima corsia.

5. Qualora non venissero ricavati indici coincidenti in output viene fornita la struttura relativa ad una data modalità di funzionamento del veicolo che si ha in input.

4.3.3 Risultati simulazione - caso 2 con risoluzione criticità

Dopo aver introdotto all'interno del modello la *function* presentata nel sottoparagrafo precedente, con il fine di renderlo in grado di superare criticità come quelle che hanno portato alla produzione di risultati non validi, esposti nel sottoparagrafo 4.3.1, è stata eseguita una simulazione utilizzando come input lo scenario presentato come *caso 2*. La simulazione ha avuto esito positivo. Il sistema è stato in grado di produrre un risultato perfettamente valido in tutti i suoi punti e non sono state riscontrate collisioni. Il veicolo *ego* ha effettuato il sorpasso del veicolo *lead* esclusivamente dopo il passaggio del veicolo *oncoming*; questo è dovuto al fatto che il sistema è stato in grado di riconoscere la criticità imminente, valutando la presenza del veicolo *oncoming* non compatibile con l'immediata manovra di sorpasso ai danni del veicolo *lead*.

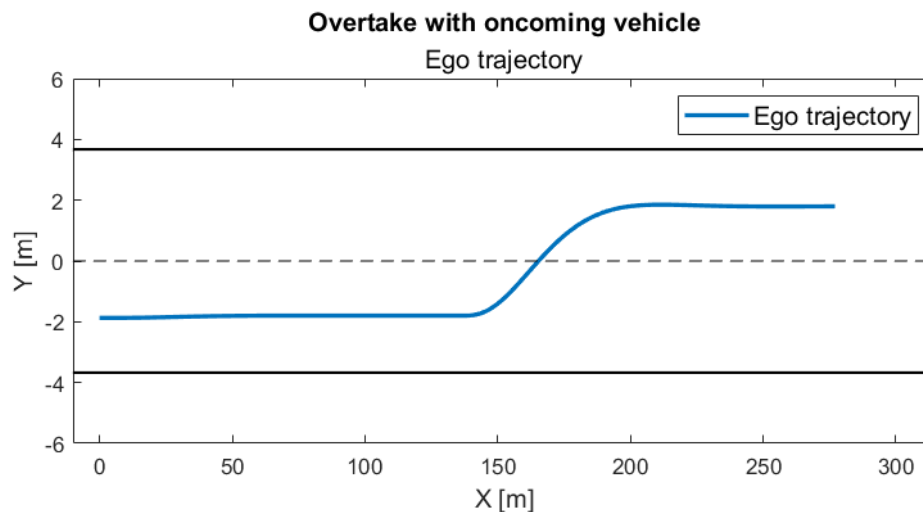


Figura 4.25: Traiettoria veicolo *ego* - scenario "Sorpasso con veicolo in senso di marcia opposto"

In figura 4.25 si ha una rappresentazione della traiettoria effettivamente percorsa dal veicolo *ego*. Si nota immediatamente come sia assente il triplo cambio di corsia ottenuto dalla simulazione in assenza della funzione *oncomingCheck*.

Per una migliore comprensione dei risultati della simulazione, in figura 4.26 sono state sovrapposte le rappresentazioni dello scenario negli istanti di maggior rilievo.

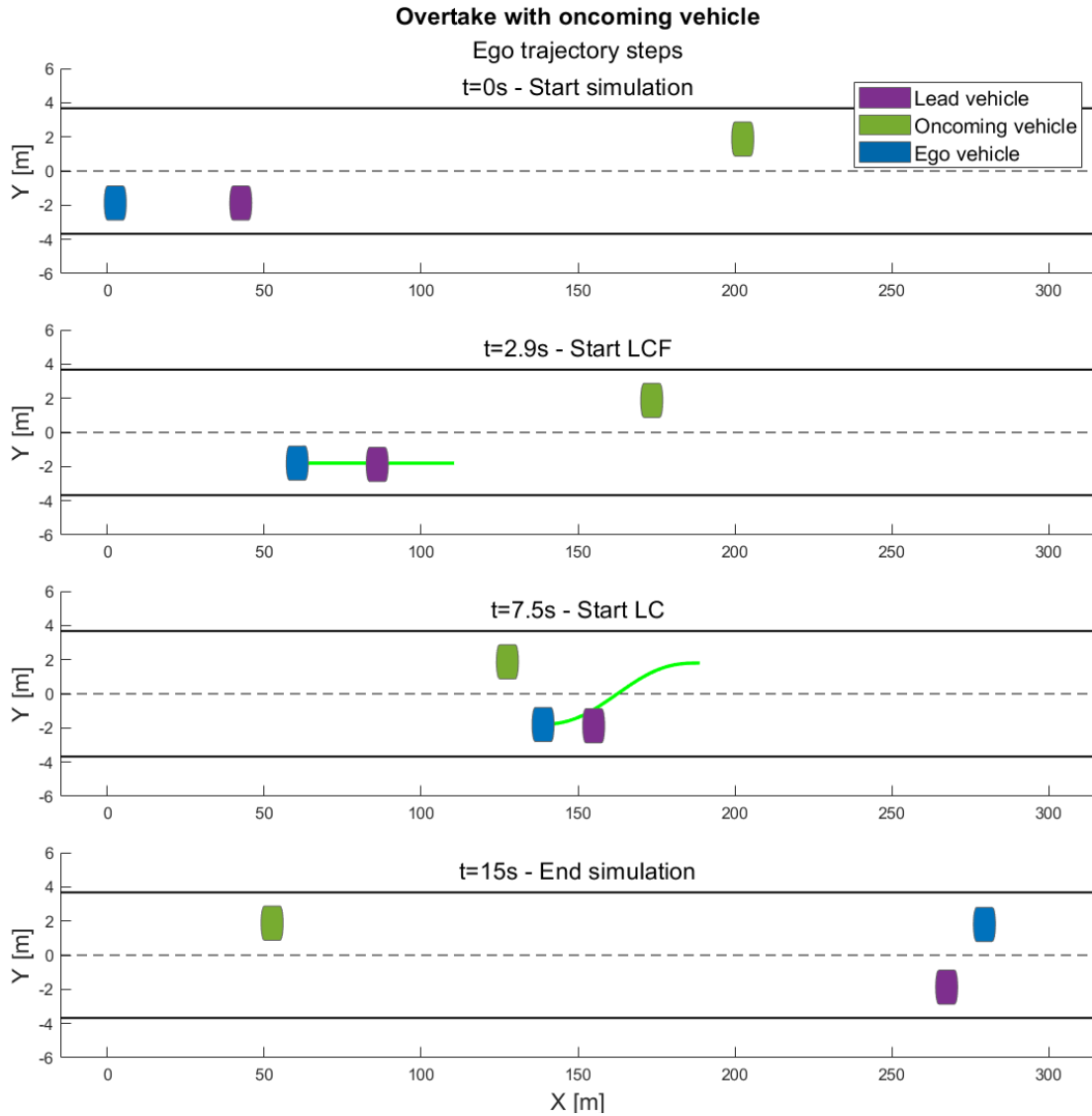


Figura 4.26: Frame traiettoria veicolo *ego* - scenario "Sorpasso con veicolo in senso di marcia opposto"

Nella tabella 4.9 sono riportati i parametri di picco, relativi alle grandezze che descrivono il moto del veicolo. Si nota come nessun parametro limite è stato superato.

Parameter	Max	Limit	Unit
Yaw rate	6.07	20	$^{\circ}/s$
Long. Acceleration	$9.41 \cdot 10^{-2}$	5	m/s^2
Lat. Acceleration	1.64	5	m/s^2
Long. Jerk	$8.53 \cdot 10^{-3}$	5	m/s^3
Lat. Jerk	0.28	5	m/s^3

Tabella 4.9: Valori di picco grandezze veicolo *ego* - scenario "Sorpasso con veicolo in senso di marcia opposto"

Si riportano nelle figure 4.27 e 4.28 i grafici delle principali grandezze descrittive del moto del veicolo *ego*.

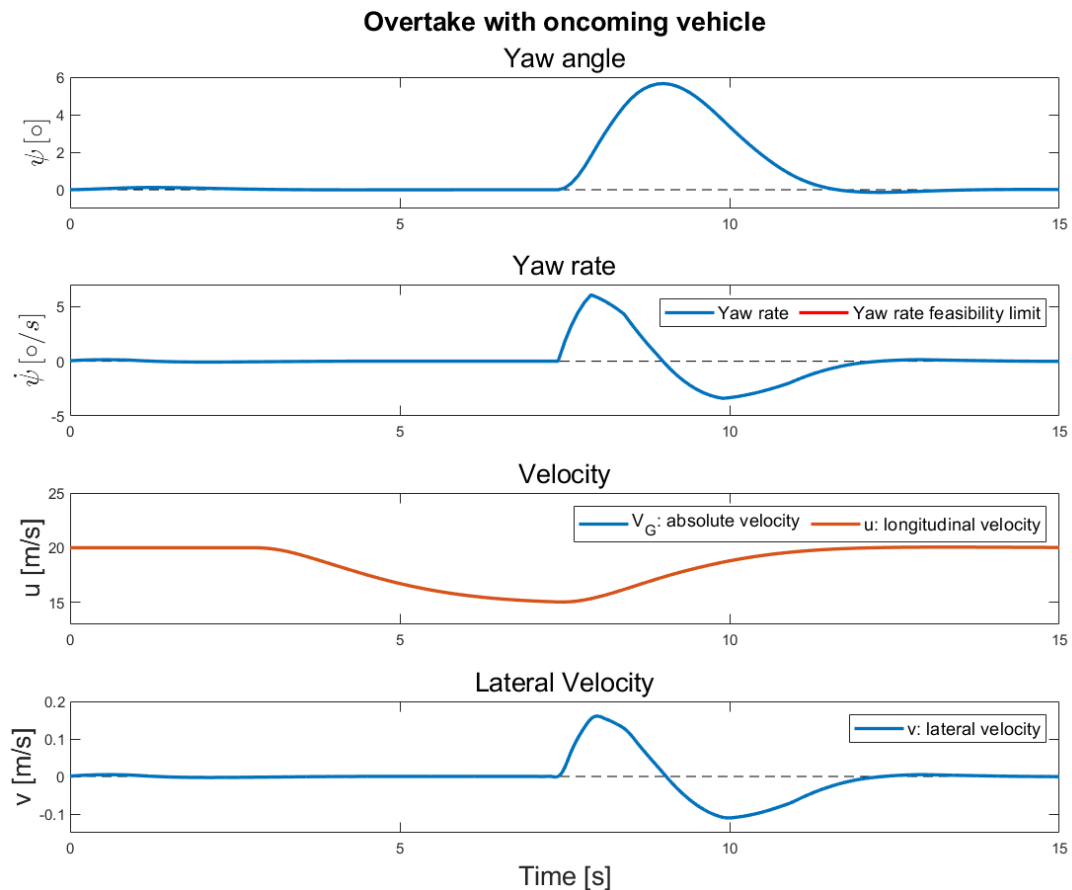


Figura 4.27: Andamenti grandezze moto veicolo *ego* - scenario "Sorpasso con veicolo in senso di marcia opposto"

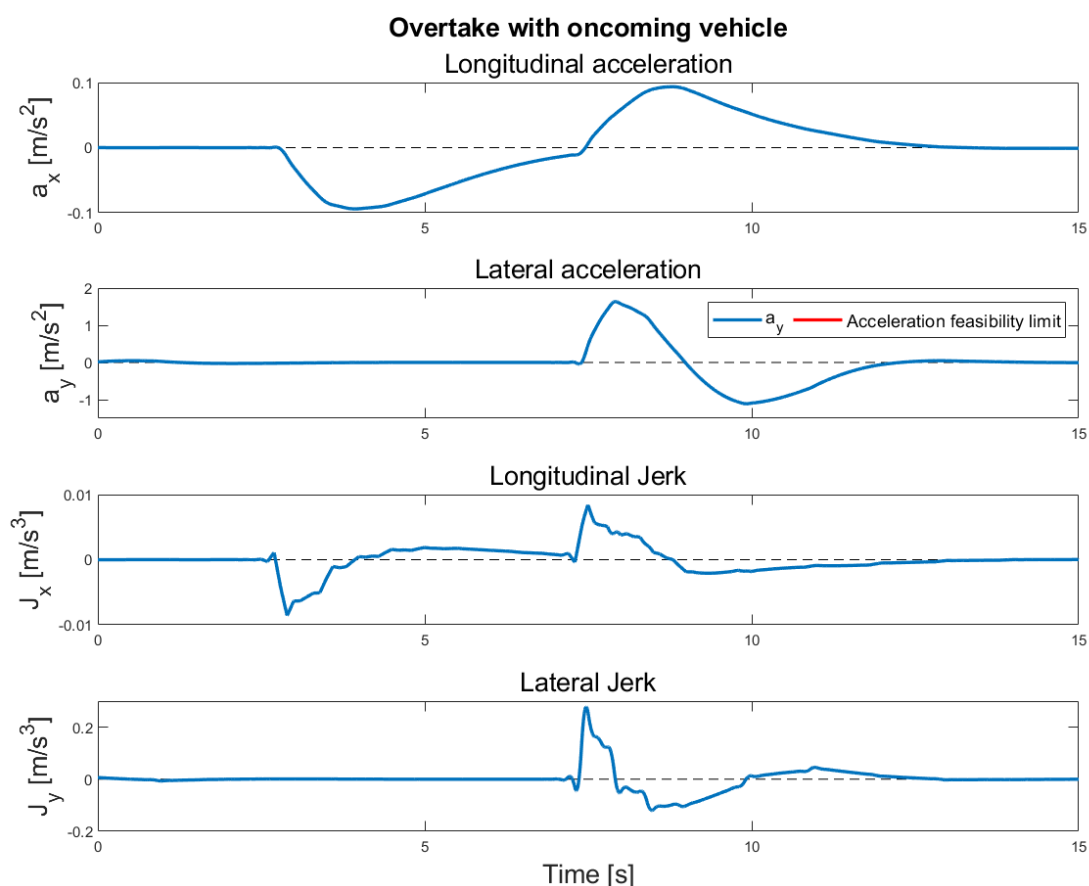


Figura 4.28: Andamenti grandezze moto veicolo *ego* (2) - scenario "*Sorpasso con veicolo in senso di marcia opposto*"

Per non incorrere in ripetizioni, si rimanda al paragrafo 4.2 per i commenti relativi a tali grafici, poiché in parte analoghi a quelli ottenuti dalla simulazione dello scenario "*Singolo cambio di corsia*". In quanto, dopo l'introduzione della funzione *oncomingCheck*, il veicolo *ego* percorre la traiettoria di un semplice sorpasso, simile a quella percorsa del veicolo *ego* nel *caso 1*.

Tuttavia, una differenza fra gli andamenti riportati per il *caso 1* la si ha circa la velocità assoluta. Si può notare una ingente decelerazione, dovuta all'impossibilità di sorpassare il veicolo *lead* e ad un conseguente cambio di modalità di funzionamento, da CC a LCF; come si nota in figura 4.29. Di conseguenza, rispetto al caso 1 si hanno andamenti differenti per le grandezze correlate alla velocità assoluta; in tal caso le componenti longitudinali di velocità, accelerazione e jerk.

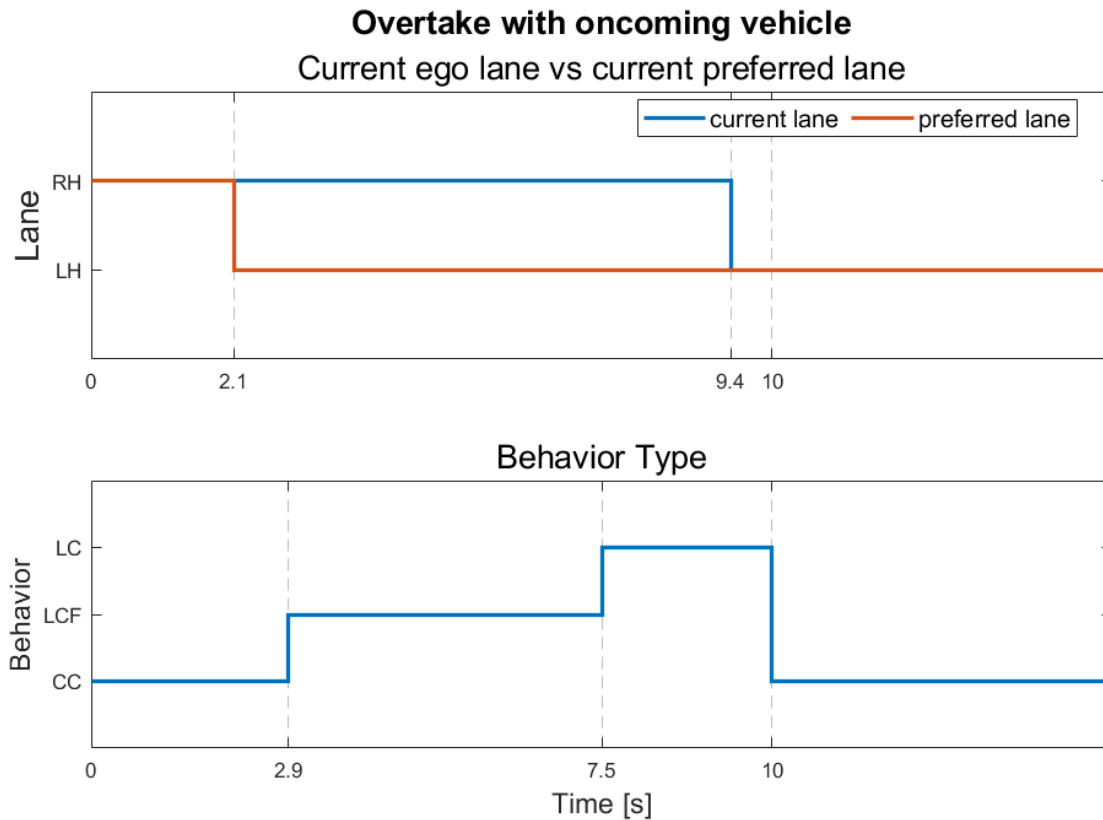


Figura 4.29: Corsia veicolo *ego* - scenario "Sorpasso con veicolo in senso di marcia opposto"

Dalla figura 4.29 si può notare come la corsia di preferenza del veicolo *ego* sia, dopo 2.1 s di simulazione, la corsia di sinistra, quindi differente dalla corsia corrente del veicolo; ciò è dovuto al rilevamento di una possibile collisione fra veicolo *ego* e *lead* con il mantenimento della traiettoria e/o della velocità descritte all'interno dello scenario. Tuttavia, dato che la funzione *oncomingCheck* impedisce al sistema di abilitare la funzione LC, l'unica opzione che non porta alla collisione è quella di abilitare la funzione LCF; la quale adegua la velocità del veicolo *ego* a quella del veicolo *lead*. La funzione introdotta inibisce la modalità LC poiché in tal caso il veicolo si ritroverebbe a transitare nella medesima corsia del veicolo *oncoming*; il quale è in avvicinamento, con senso di marcia opposto e con un *TTC* inferiore alla soglia, ovvero a 10 s, come si può vedere dal secondo grafico in figura 4.30.

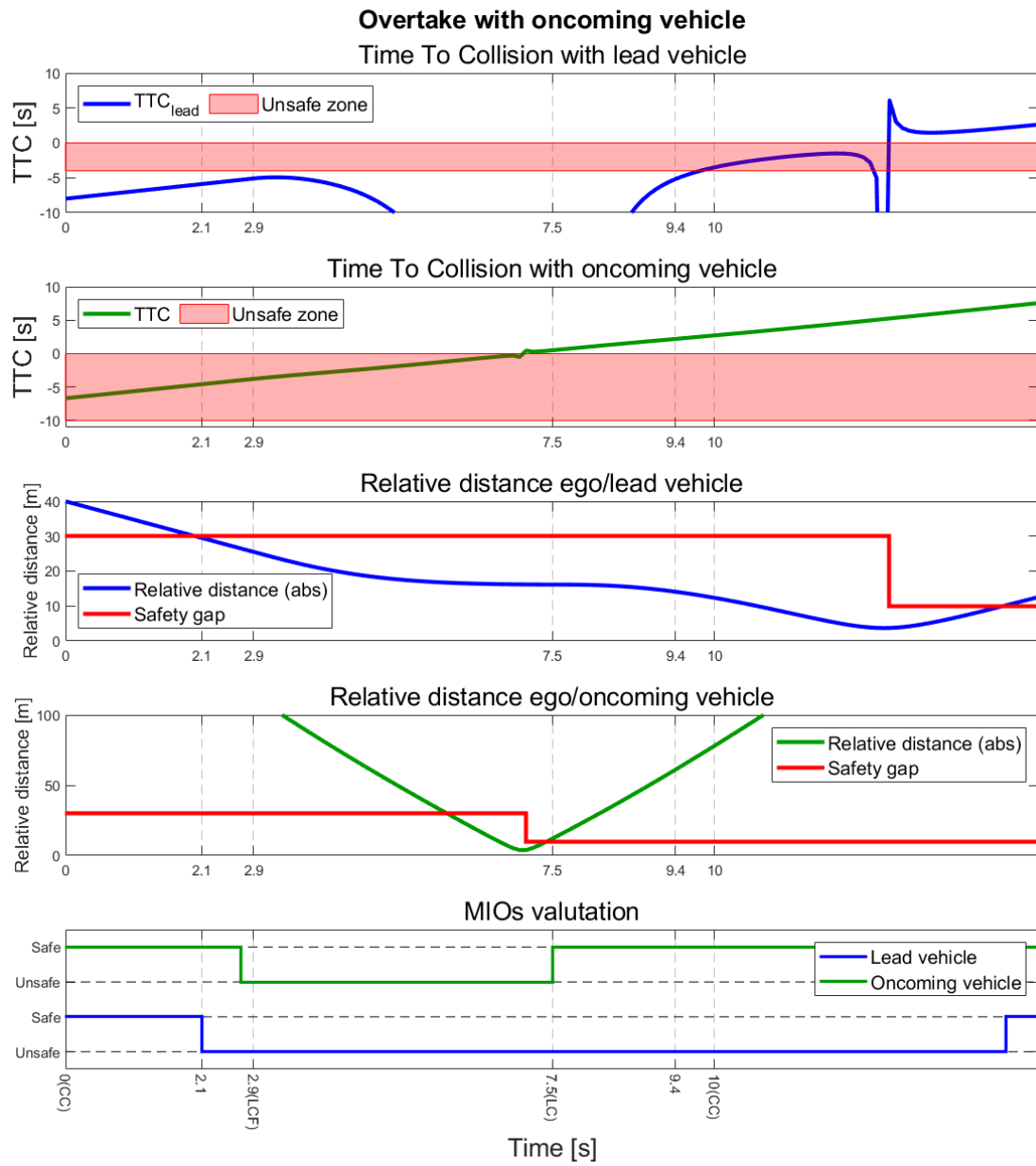


Figura 4.30: Valutazione veicoli MIO- scenario "Sorpasso con veicolo in senso di marcia opposto"

Analizzando nel dettaglio i grafici in figura 4.30 emerge che:

- Da $t=0$ s a $t=2.0$ s il veicolo procede indisturbato in modalità CC.
- Nell'istante $t=2.1$ s viene superata la soglia *Front safety gap* calcolata fra il veicolo *ego* ed il veicolo *lead*. Tuttavia; ciò non implica un immediato passaggio alla modalità LCF, poiché ad essa corrisponde un parametro di costo più elevato rispetto a quello associato agli stati calcolati mediante modalità CC. Nonostante la corsia attuale del veicolo *ego* sia definita *Unsafe*, in quanto in essa transita un veicolo dichiarato come tale, non è possibile abilitare la modalità LC. Ciò accade poiché, come si può notare dal secondo grafico in figura 4.30, il *TTC* calcolato tra il veicolo *ego* ed il veicolo *oncoming* sin dai primi istanti della simulazione risulta essere inferiore alla soglia imposta; dunque la funzione introdotta inibisce la possibilità di un cambio corsia.
- Nell'istante $t=2.9$ s viene abilitata la modalità LCF, poiché ad essa corrisponde un parametro di costo inferiore rispetto alla modalità CC. A ciò consegue un progressivo rallentamento del veicolo *ego*, con l'obiettivo di poter adeguare la sua velocità a quella del veicolo *lead*. Conseguentemente il *TTC* calcolato tra i due veicoli menzionati cresce in modulo e la distanza relativa viene mantenuta costante dopo un transitorio iniziale.
- Dal quarto grafico si può ricavare l'istante in cui il parametro di *Relative Distane* calcolato fra il veicolo *ego* ed il veicolo *oncoming* rispetta nuovamente la soglia imposta (*Rear safety gap*). Tale istante è pari a 7.5 s. Quindi si nota dal quinto grafico come per $t=7.5$ s il veicolo *oncoming* venga dichiarato *Safe* ed il concomitante valore di *TTC* positivo permette al sistema di ottenere nel set di alternative anche gli stati terminali che coinvolgono la modalità LC e di abilitarla.
- Negli istanti successivi viene eseguita regolarmente la manovra di cambio corsia ed il veicolo *ego* procede lungo la corsia di sinistra fino al termine della simulazione.

Studio del parametro $TTC_{oncoming}$

Al termine di questo caso studio è stato effettuato un'analisi sul valore della soglia $TTC_{oncoming}$ impostato. Come anticipato nel sottoparagrafo precedente ad esso è stato associato il valore $TTC_{oncoming} = 10 s$ in maniera arbitraria. Tuttavia; come si può notare dal secondo grafico in figura 4.30, il valore del TTC calcolato rispetto al veicolo *oncoming* risulta essere ben al di sotto della soglia imposta sin dai primi istanti della simulazione; ciò può essere dovuto ad un eccesso di conservatività.

Lo studio presentato è stato effettuato variando il parametro della soglia e la velocità del veicolo *ego*; inoltre, lo scenario testato è analogo a quello presentato in questo caso studio, se non per la posizione iniziale del veicolo *ego*.

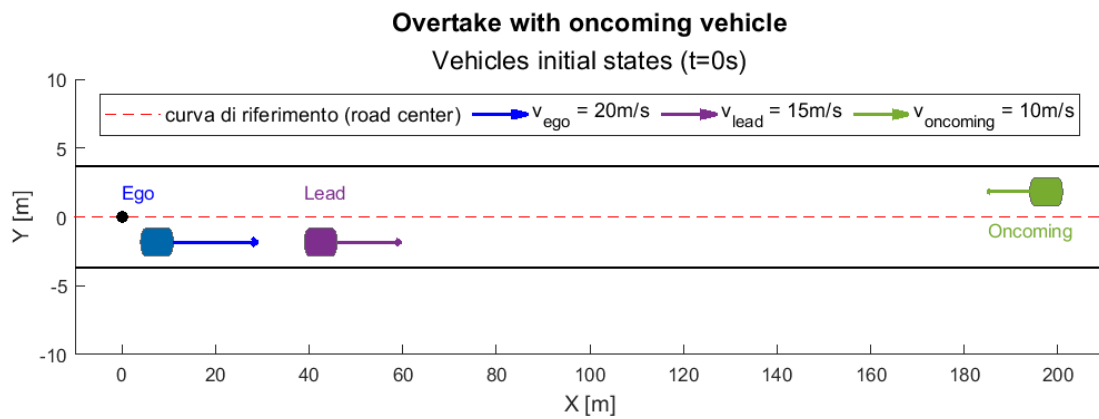


Figura 4.31: Stati iniziali $t=0s$ - scenario "Sorpasso di una fila di veicoli"

Come si nota dalla figura 4.31, esso è stato spostato in avanti di 5m. Tale modifica è stata effettuata con l'intento di diminuirne la distanza dal veicolo *lead*, rendendo più imminente la necessità di un sorpasso. Se la manovra di sorpasso viene anticipata, mantenendo invariati tutti gli altri parametri descrittivi dello scenario, ciò implica un valore maggiore del TTC calcolato rispetto al veicolo *oncoming* quando viene attivata la modalità LC. Di conseguenza, è preferibile effettuare lo studio in tale condizione, in quanto garantisce una maggiore sensibilità al parametro $TTC_{oncoming}$.

Si riportano nella tabella 4.10 i risultati ottenuti dalle simulazioni.

		$TTC_{oncoming}$			
		4s	5s	6s	8s
v_{ego}	60 km/h	Triple LC	Active	Active	Active
	70 km/h	Stop	Triple LC	Active	Active
	80 km/h	Colliding	Colliding	Active	Active
	90 km/h	Disactive	Disactive	Active	Active

Tabella 4.10: Studio sensibilità $TTC_{oncoming}$ - scenario "Sorpasso di una fila di veicoli"

Si può notare che sono stati registrati cinque differenti comportamenti del sistema; essi devono essere interpretati nel seguente modo:

- *Active*: La soglia $TTC_{oncoming}$ è stata superata, conseguentemente è stata inibita la modalità LC ed il veicolo *ego* ha superato il veicolo *lead* solo dopo che il veicolo *oncoming* è sfilato alla propria sinistra. In maniera analoga a quanto visto in questo paragrafo.
- *Triple LC*: La soglia $TTC_{oncoming}$ è stata superata dopo aver già iniziato il cambio di corsia; tuttavia, essendo intercorso poco tempo dall'inizio della manovra è stato possibile effettuare un rientro in corsia in sicurezza.
- *Stop*: Il sistema ha interrotto il moto del veicolo *ego* non avendo individuato una traiettoria sicura e percorribile. La natura di questo errore verrà approfondita nel paragrafo successivo.
- *Colliding*: Il veicolo *ego* effettua la manovra di sorpasso; tuttavia, non avendo possibilità di rientro in corsia entra in collisione con il veicolo *oncoming*.
- *Disactive*: La soglia $TTC_{oncoming}$ non viene superata ed il veicolo *ego* effettua regolarmente la manovra di sorpasso, non entrando in collisione con gli altri veicoli dello scenario.

A valle di quanto mostrato in tabella 4.10 è possibile affermare che il valore associato inizialmente alla soglia (10s) risulta essere molto conservativo; poiché assumendo un valore pari a $TTC_{oncoming} = 6\text{ s}$ non viene registrato alcun errore. Tuttavia, con l'obiettivo di far funzionare correttamente la strategia implementata anche per gli scenari maggiormente critici è consigliabile adottare un valore superiore, in modo da aumentare la conservatività del sistema assistenziale.

4.4 Caso 3 - Sorpasso di una fila di veicoli

Lo scenario "*Sorpasso di una fila di veicoli*" è stato introdotto con l'obiettivo di testare e migliorare le capacità del sistema assistenziale nel pianificare la manovra corretta a fronte di un insieme di veicoli da sorpassare con un ingombro complessivo elevato. Definendo come "fila di veicoli" un gruppo di veicoli che viaggiano con una medesima traiettoria e con vincoli di distanza ristretti. La problematica principale portata dallo scenario è data dall'impossibilità da parte del veicolo ego di un rientro in corsia tra un veicolo e l'altro della fila, a causa di uno spazio tra i veicoli non compatibile con tale manovra.

L'obiettivo dell'analisi riportata è quello di costruire un meccanismo di pianificazione della traiettoria che sia in grado di valutare eventualità come quella in esame; in modo da valutare la manovra tenendo conto di un ingombro superiore rispetto al solo veicolo *lead*.

Entrando maggiormente nel dettaglio, tale scenario si compone di sei differenti veicoli che transitano lungo una strada rettilinea a due corsie, una per senso di marcia.

Lunghezza strada	700 m
n° corsie	2

Tabella 4.11: Dati strada - scenario "Sorpasso di una fila di veicoli"

I veicoli che compongono lo scenario sono i seguenti:

- un veicolo *ego*;
- un veicolo *lead*, transitante lungo la medesima corsia del veicolo ego e con una velocità inferiore allo stesso;
- tre veicoli *obstacle*, nello specifico chiamati *obstacle 1*, *obstacle 2* e *obstacle 3*, i quali precedono il veicolo *lead*, procedono con la medesima velocità di quest'ultimo e si distanziano l'un l'altro di un'eguale misura;

- un veicolo che procede con senso di marcia opposto e nella corsia opposta rispetto a quella in cui transitano i veicoli precedentemente descritti; il quale è stato denominato veicolo *oncoming*.

Si riportano nella tabella 4.7 i parametri di configurazione dello scenario; inoltre, si precisa che tutti i veicoli presenti hanno la medesima geometria.

$\text{lane}_{ego} (t = 0 s)$	2
lane_{lead}	2
$\text{lane}_{obstacle 1,2,3}$	2
$\text{lane}_{oncoming}$	1
$s_{lead_0} - s_{ego_0}$	35 m
$s_{oncoming_0} - s_{ego_0}$	480 m
$\text{distance}_{vehicles row}$	13 m
v_{ego}	20 m/s
v_{lead}	15 m/s
$v_{obstacle 1,2,3}$	15 m/s
$v_{oncoming}$	12 m/s

Tabella 4.12: Dati veicoli - scenario "Sorpasso di una fila di veicoli"

In riferimento alla tabella riportata si identifica con l'indice 2 la corsia di destra, assumendo il punto di vista del veicolo *ego*, e con l'indice 1 la corsia di sinistra; inoltre, con il termine $\text{distance}_{vehicles row}$ si intende la distanza tra i veicoli appartenenti alla fila, calcolata rispetto al relativo asse posteriore.

In figura 4.32 si ha una rappresentazione grafica dello scenario all'istante iniziale della simulazione. I limiti della carreggiata sono stati riprodotti mediante due linee continue nere e la linea che delimita le due corsie, dunque i due sensi di marcia, è stata realizzata con una linea tratteggiata. Quest'ultima coincide con la curva di riferimento delle coordinate di Frenét, o *center line*, la quale è rappresentata mediante una linea tratteggiata di colore rosso; i *way points* con cui essa è stata creata sono riportati in forma grafica mediante punti neri e sono: $[0, 0]$ e $[700, 0]$.

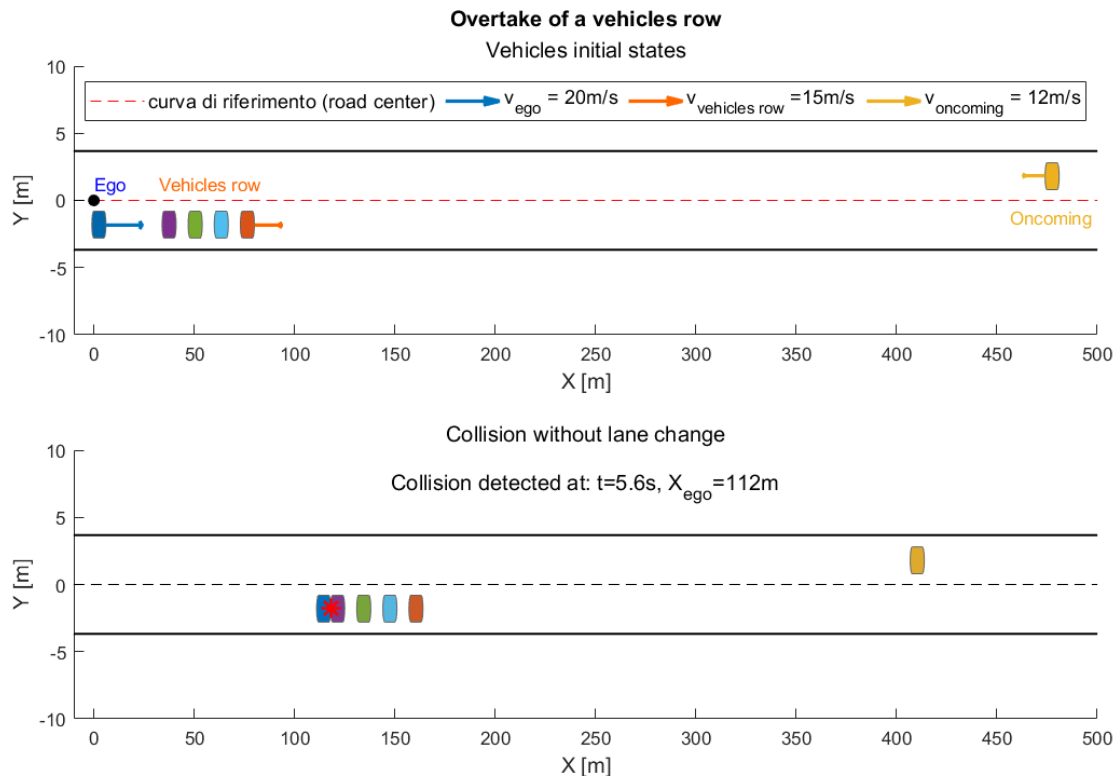


Figura 4.32: Stati iniziali $t=0\text{s}$ - scenario "Sorpasso di una fila di veicoli"

Con il termine $v_{vehicle\ row}$ si individua la velocità dei componenti della fila di veicoli; poiché da ipotesi iniziali questi si muovono mantenendo una velocità costante ed una distanza relativa costante. Tale condizione è espressa dalla seguente relazione:

$$v_{vehicles\ row} = v_{lead} = v_{obstacle\ 1} = v_{obstacle\ 2} = v_{obstacle\ 3} = 15\text{ m/s} \quad (4.6)$$

In figura 4.33 è stato riportato un ingrandimento dello scenario all'istante iniziale; grazie al quale è stato possibile identificare graficamente la fila di veicoli, i singoli componenti e le relative distanze. Per semplicità è stato rappresentato un unico vettore velocità $v_{vehicles\ row}$ di colore rosso, valido per tutti i veicoli appartenenti alla fila.

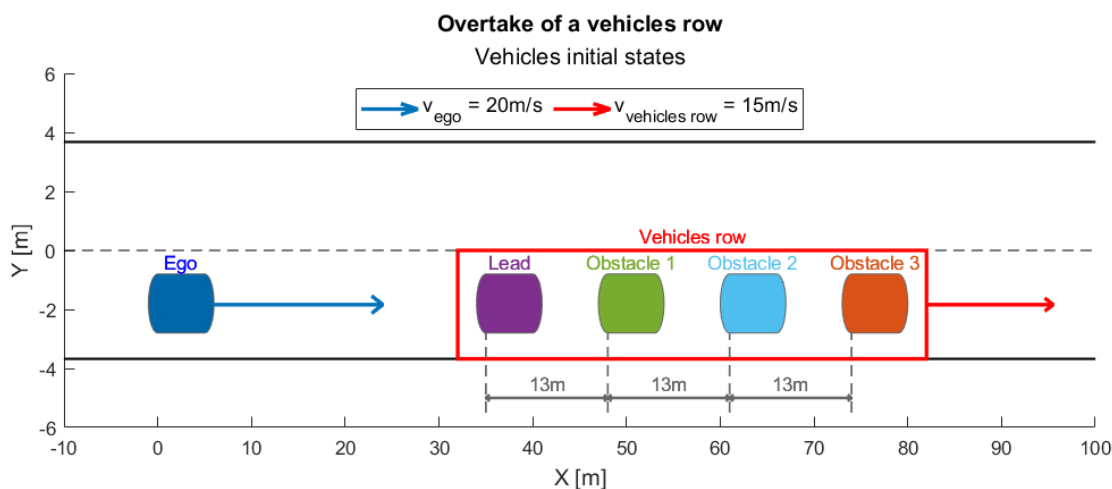


Figura 4.33: Fila di veicoli - scenario "*Sorpasso di una fila di veicoli*"

Anche in questo caso lo scenario è stato costruito in maniera tale da richiedere l'intervento del sistema assistenziale; ciò è dovuto ad un'incompatibilità tra la fila di veicoli ed il veicolo *ego* circa le traiettorie e le velocità. Tale incompatibilità, in assenza di interventi sulla traiettoria del veicolo *ego* e/o sulla sua velocità, porta ad una collisione dopo 5.6 s dall'istante iniziale della simulazione, come si può notare dal secondo grafico riportato in figura 4.32.

Si può facilmente notare, data la distanza ridotta tra i veicoli che viene mantenuta costante a causa dell'uguaglianza tra le velocità dei componenti fila, che il veicolo *ego* non può rientrare tra un veicolo e l'altro in caso di sorpasso ma può completare la manovra solo dopo aver superato il veicolo *front* (veicolo alla testa della fila). Nel caso in esame tale denominazione è associata al veicolo *obstacle 3*, rappresentato in arancione nel grafico in figura 4.33.

Tale scenario è stato successivamente riprodotto all'interno dell'applicazione *Driving scenario designer*, in modo da ottenere l'input corretto per il modello utilizzato. Si esaminano nel sotto-paragrafo successivo i risultati ottenuti dalla simulazione.

4.4.1 Risultati simulazione - caso 3

La simulazione esaminata di seguito è stata effettuata mantenendo attive le modifiche del modello che sono state introdotte in seguito all'analisi dei risultati del caso studio precedente, "*Sorpasso con veicoli in senso di marcia opposto*" - paragrafo 4.3. L'analogia con il caso precedente sta nel progressivo avvicinamento di un veicolo *oncoming* nella corsia con senso di marcia opposto; invece, la differenza è legata all'estensione del tempo di permanenza nella medesima corsia del veicolo *oncoming*. Poiché per completare la manovra di sorpasso il veicolo *ego* deve oltrepassare l'intera fila di veicoli.

La simulazione non è stata completata con successo, in quanto, per un determinato intervallo di tempo il modello non ha individuato una traiettoria valida tra le alternative di CC, LCF e LC. Durante tale intervallo di tempo il veicolo risulta essere fermo ed in attesa che una delle tre modalità di funzionamento venga considerata valida e percorribile fisicamente. L'analisi dei risultati riportata in seguito verterà sulle cause di questa anomalia e non approfondirà ciò che succede durante gli istanti successivi, in quanto tutti i valori registrati sono ritenuti non validi. Nella seguente figura 4.34 è stata riportata su un piano cartesiano la traiettoria effettivamente seguita dal veicolo *ego* durante la simulazione.

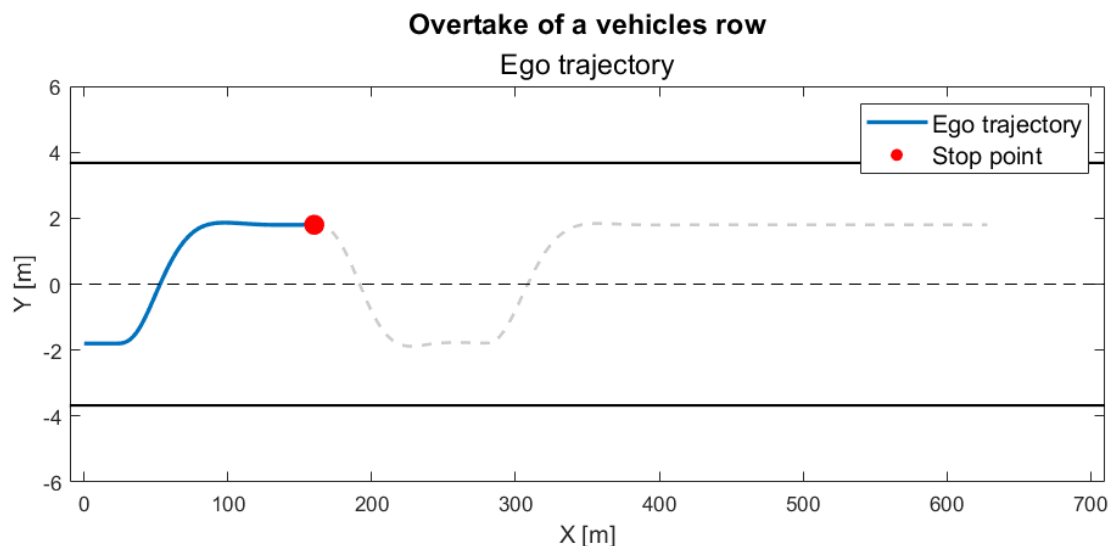


Figura 4.34: Traiettoria veicolo *ego* - scenario "*Sorpasso di una fila di veicoli*"

Per comprendere al meglio le ragioni per le quali non è possibile considerare accettabile il risultato della simulazione in esame è utile non limitarsi alla sola traiettoria percorsa dal veicolo *ego* ed analizzare l'evoluzione del suo stato nel tempo. A tale scopo è stato prodotto un grafico, riportato in figura 4.35, che mostra il posizionamento dei veicoli all'interno dello scenario in specifici istanti temporali.

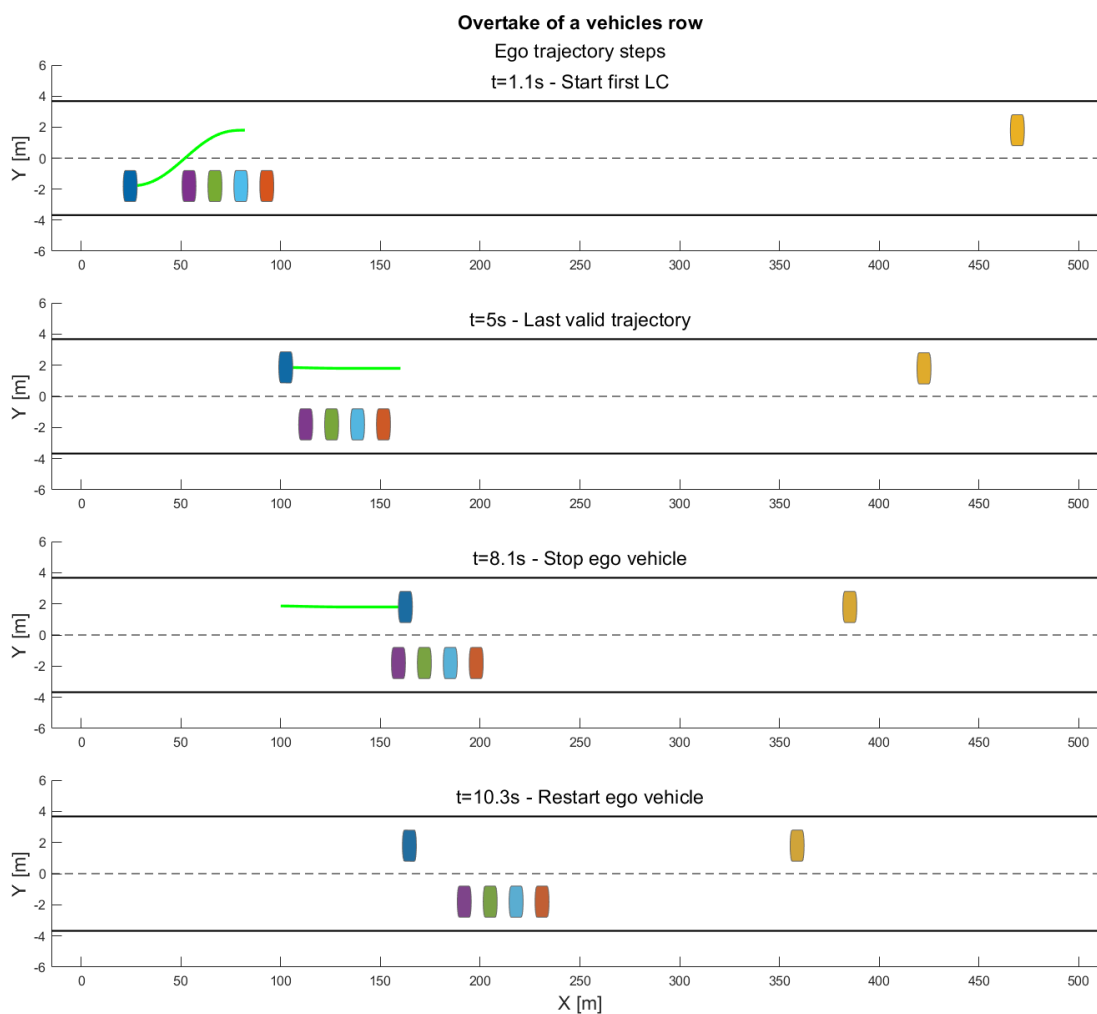


Figura 4.35: Frame traiettoria veicolo *ego* - scenario "Sorpasso di una fila di veicoli"

Si riporta di seguito l'analisi di ognuno dei grafici inseriti nelle figure 4.34 e 4.35:

- $t = 1.1 s$: Il veicolo *ego* raggiunge una distanza tale dal primo veicolo appartenente alla fila, il veicolo denominato *lead* e rappresentato in viola, da abilitare la modalità di cambio corsia.
- $t = 5 s$: il pianificatore individua come traiettoria di riferimento una traiettoria rettilinea che mantenga il veicolo *ego* al centro della corsia di sinistra. Tale segmento ha un orizzonte temporale pari a 3 s ed è stato riportato in verde nel secondo grafico in figura 4.35.
- Da $t = 5.1 s$ a $t = 8 s$: Il veicolo *ego*, dopo aver completato la manovra di cambio corsia, ha già iniziato a superare parte della fila di veicoli; tuttavia, l'imminente avvicinamento del veicolo *oncoming* (rappresentato in giallo) lungo la medesima corsia e con senso di marcia opposto porta ad un'incompatibilità con l'attuale configurazione del modello. Nell'istante $t = 5.1 s$ il TTC calcolato tra il veicolo *ego* ed il veicolo *oncoming* scende in modulo al di sotto della soglia $TTC_{oncoming}$ imposta. Ciò che consegue è l'eliminazione delle modalità che prevedono la pianificazione di traiettorie che implicano la presenza del veicolo *ego* nella medesima corsia del veicolo *oncoming*, in tal caso sono quelle relative alle modalità CC e LCF. Di conseguenza l'unica modalità che può essere utilizzata è la LC; ma anche in questo caso non è possibile pianificare le relative traiettorie, in quanto, a causa dell'ingombro della fila di veicoli, il veicolo *ego* non può effettuare il rientro in corsia senza impattare con uno dei suoi componenti. In tale intervallo temporale il veicolo *ego* continua a seguire il segmento di traiettoria individuato al punto precedente.
- $t = 8.1 s$: Non avendo a disposizione una traiettoria da seguire il modello congela la posizione del veicolo *ego*, arrestandone il moto nell'ultimo punto pianificato fin quando non viene individuata una nuova traiettoria da seguire.
- $t = 10.3 s$: Durante lo stop del veicolo *ego* la fila di veicoli continua ad avanzare lungo la propria traiettoria, fin quando non sorpassa rispetto la coordinata curvilinea s lo stesso veicolo *ego*. Nell'istante pari a 10.3 s si ha una distanza

sufficiente tra fila di veicoli ed *ego* da permettere a quest'ultimo di eseguire un rientro in corsia. Dunque, il veicolo *ego* torna istantaneamente in movimento per seguire la nuova traiettoria pianificata.

Grazie a questa panoramica generale del comportamento del veicolo *ego* nel corso della simulazione sono state individuate le due principali problematiche che hanno portato a ritenere non valido il risultato ottenuti:

- La strategia non efficace di pianificazione ha portato a realizzare un cambio di corsia per superare la fila di veicoli, in assenza delle condizioni adatte a portare a termine la manovra.
- La simulazione prevede uno stop del veicolo dall'istante $t = 8.1 s$ a $t = 10.3 s$. Come anticipato, questo è dovuto all'assenza di una soluzione percorribile fra il set di alternative; set dal quale sono state eliminate le soluzioni che prevedono la modalità CC e LCF. Non si tratta di uno stop effettivo del veicolo, come ad esempio una frenata di emergenza dovuta alla criticità della situazione, ma di un errore della simulazione. In altre parole, nell'istante in cui termina lo stop il veicolo riprende la marcia esattamente con gli stessi valori dei parametri caratterizzanti il proprio moto, come se lo stop non fosse mai avvenuto.

Al termine dello stop la simulazione va avanti, il veicolo *ego* rientra in corsia e successivamente, dopo aver fatto sfilare il veicolo *oncoming*, supera la fila di veicoli. La traiettoria descritta dopo lo stop è rappresentata in grigio in figura 4.34. Tuttavia, come anticipato, è stato scelto di non approfondire ciò che accade dopo il periodo di stop, in quanto i dati registrati durante tale criticità non hanno alcuna validità fisica; dunque, invalidano del tutto la simulazione.

La deduzione circa la natura dello stop del veicolo emerge dall'analisi dei parametri descrittivi del moto del veicolo *ego*. In particolare, le acquisizioni mostrano un congelamento di tutti i valori caratterizzanti, come ad esempio: posizione, angolo di imbardata, velocità e accelerazione; i quali rimangono costanti nel corso dell'intervallo citato (da $t = 8.1 s$ a $t = 10.3 s$). Tale accadimento non può che essere interpretato come un errore del sistema, in quanto sarebbe impossibile avere valori di velocità non nulli mantenendo il veicolo fermo nella medesima posizione.

Di conseguenza, il modello fa riprendere il moto al veicolo *ego* con uno stato coincidente con quello del primo istante di stop, come se non si fosse presentata alcuna irregolarità. Ciò implica che la natura dell'errore è imputabile esclusivamente alle logiche di pianificazione del modello; dunque, ad esclusione dell'intervallo in cui tutte le grandezze del moto rimangono costanti, non si riscontrano anomalie nei loro andamenti. Per tale motivo è stato ritenuto poco proficuo ai fini dell'analisi della problematica riportarne i relativi grafici, ma per completezza nella seguente tabella 4.13 sono riportati i valori di picco delle principali grandezze; i quali rispettano tutti i limiti imposti.

Parameter	Max	Limit	Unit
Yaw rate	6.59	20	$^{\circ}/s$
Long. Acceleration	0.11	5	m/s^2
Lat. Acceleration	2.31	5	m/s^2
Long. Jerk	$1.09 \cdot 10^{-2}$	5	m/s^3
Lat. Jerk	0.43	5	m/s^3

Tabella 4.13: Valori di picco grandezze veicolo *ego* - scenario "Sorpasso di una fila di veicoli"

Per una migliore comprensione delle problematiche evidenziate risulta essere utile analizzare l'andamento dei segnali utilizzati dal pianificatore per stabilire la traiettoria del veicolo *ego*. Si riporta in figura 4.36 l'andamento della corsia preferenziale e di quella corrente (primo grafico) e la modalità di funzionamento del veicolo nel corso del tempo (secondo grafico).

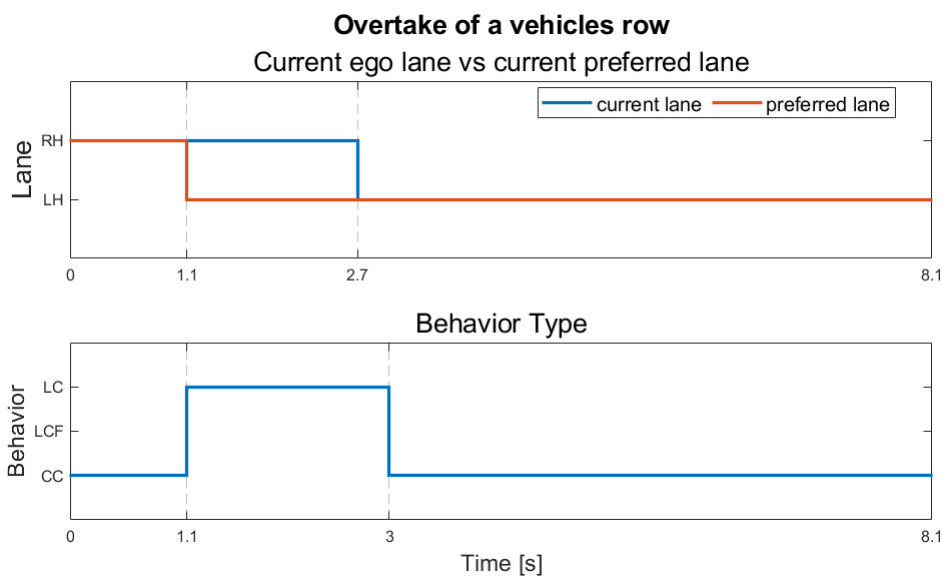


Figura 4.36: Corsia veicolo *ego* - scenario "*Sorpasso di una fila di veicoli*"

In figura 4.37 sono stati sovrapposti gli andamenti delle principali grandezze utili alle valutazioni del pianificatore di traiettoria in merito al confronto tra il veicolo *ego* e gli altri veicoli presenti nello scenario. Nello specifico sono stati riportati i grafici:

- Relativi al *TTC* e alla *Relative distance* con il veicolo *lead*, veicolo che chiude la fila, con cui il veicolo *ego* rischia di collidere in assenza di una variazione della traiettoria pianificata.
- Del *TTC* calcolato rispetto al veicolo *oncoming*. Si nota una zona *Unsafe* maggiore rispetto a quella relativa al veicolo *lead*, in quanto in questo caso si tratta di un veicolo che viaggia con senso di marcia opposto rispetto a quello del veicolo *ego*. Il valore soglia è quello stabilito a valle dello studio

portato avanti con l'analisi del caso precedente "Sorpasso con veicolo in senso di marcia opposto", paragrafo 4.3.

- Sulla valutazione della sicurezza (*Safe/Unsafe*) relativa ai due veicoli citati.
- Infine, si specifica che nell'asse delle ordinate sono stati evidenziati gli istanti temporali di maggior rilievo e la relativa modalità di funzionamento del pianificatore; con il fine di aiutare a comprendere come hanno influito i vari segnali alla pianificazione del veicolo *ego*.

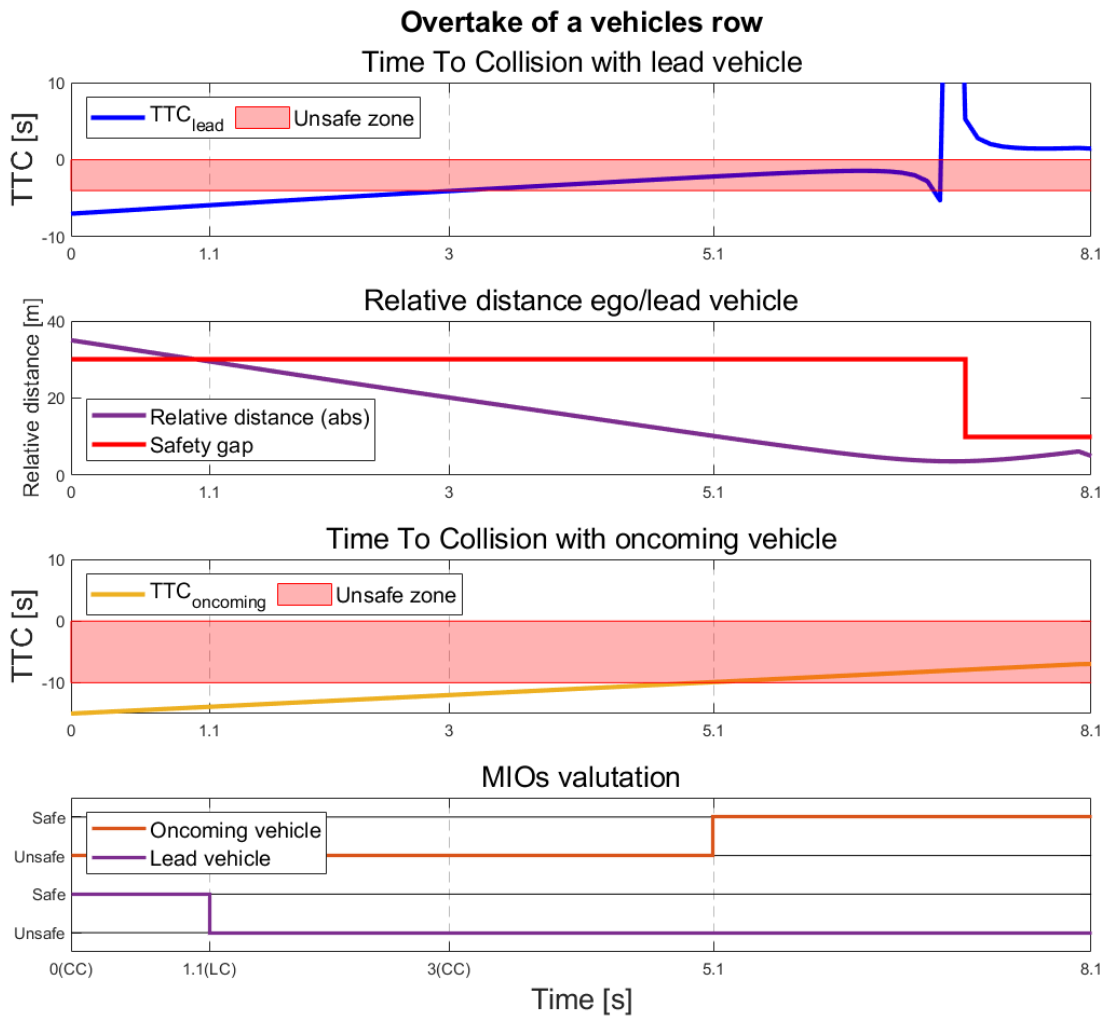


Figura 4.37: Valutazione veicoli MIO - scenario "Sorpasso di una fila di veicoli"

Confrontando i grafici presentati nelle figure 4.34, 4.35 e 4.37, è possibile fornire un'interpretazione più dettagliata delle cause che hanno portato alle due problematiche precedentemente esposte. Si riporta di seguito l'analisi per punti, effettuata seguendo la linea cronologica degli eventi.

- Da $t = 0 s$ a $t = 1.0 s$: Il veicolo *ego* segue la traiettoria preimpostata durante la fase di costruzione dello scenario senza riscontrare la necessità di una nuova pianificazione della stessa.
- $t = 1.1 s$: Il veicolo *ego* entra in modalità LC. Questo avviene poiché viene oltrepassato il limite imposto sulla *Relative distance* con il veicolo che lo precede (secondo grafico in figura 4.37), in questo caso il veicolo *lead*, colui il quale chiude la fila di veicoli. Ciò implica la variazione di *preferred lane* (primo grafico in figura 4.36); a causa di quest'ultima variazione il pianificatore individua come riferimento la traiettoria che porta al cambio corsia.
- $t = 2.7 s$: Viene nuovamente abilitata la modalità CC; poiché, come si vede dal grafico in figura 4.36, il veicolo *ego* ha ormai raggiunto la corsia di sinistra.
- $t = 5.1 s$: Il *TTC*, calcolato fra i veicoli *ego* e *oncoming*, risulta essere inferiore in modulo alla soglia imposta ($TTC_{oncoming}$). Ciò implica la catena logica introdotta con la funzione illustrata attraverso il caso studio precedente, paragrafo 4.3 "Sorpasso con veicolo in senso di marcia opposto"; la quale porta all'eliminazione di tutti gli stati terminali che implicano il collocamento al dato orizzonte temporale del veicolo *ego* nella medesima corsia del veicolo *oncoming*. Di conseguenza il pianificatore elimina tutti gli stati terminali che mantengono l'attuale corsia occupata dal veicolo *ego*, ovvero quelli che prevedono le modalità CC e LCF. Da questa esclusione rimangono tra il set di alternative esclusivamente gli stati terminali che prevedono un cambio di corsia (modalità LC). Tuttavia, come è possibile vedere in figura 4.35, il veicolo *ego* non ha a disposizione uno spazio sufficiente per poter rientrare in corsia.

- $t = 6.1 s$: Si ricorda che il modello pianifica una nuova traiettoria ogni qualvolta viene attivato un flag; quest'ultimo assume il valore logico 1 con una cadenza periodica di $1 s$ a meno di ulteriori allarmi provenienti dai segnali elaborati, ad esempio una variazione di *preferred lane*. Nel tempo che intercorre tra $t = 5.1 s$ ed $t = 6.1 s$ non vi sono variazioni di segnali che portano alla valutazione del set di alternative disponibili; conseguentemente solo dopo essere passato l'intervallo temporale di pianificazione del modello ($1s$), ovvero nell'istante pari a $t = 6.1 s$ vengono valutati gli stati terminali. Nel corso della valutazione emerge l'impossibilità di seguire una fra le traiettorie selezionate, in quanto tutte le alternative disponibili portano alla collisione con i veicoli della fila durante l'eventuale rientro in corsia. Il veicolo *ego*, in assenza di nuove traiettorie, continua a seguire l'ultima traiettoria individuata.
- $t = 8.1 s$: L'ultimo segmento di traiettoria seguito dal veicolo *ego* è stato pianificato nell'istante della simulazione $t = 5.0 s$ con il massimo orizzonte temporale disponibile ($3s$). Di conseguenza il veicolo segue la traiettoria a disposizione fino all'istante corrente per poi bloccarsi in quanto non ha a disposizione un nuovo segmento da seguire. Ciò accade poiché ad ogni istante sono state valutate le alternative disponibili; ma tutte portano alla collisione durante l'eventuale manovra di rientro in corsia non avendo ancora superato la fila di veicoli, come si può vedere in figura 4.35.

4.4.2 Adattamento del sistema alle criticità presentate - Caso 3

Le considerazioni effettuate nel corso dell'analisi dei risultati hanno portato all'individuazione di due criticità principali:

- La mancanza di una strategia di pianificazione ottimale, in quanto per effettuare il sorpasso vengono effettuati tre differenti cambi di corsia.
- La presenza di uno *stop* del moto del veicolo *ego*, che ha portato l'intera simulazione ad essere ritenuta non valida.

La cause dello stop è originata dall'impossibilità di seguire le traiettorie disponibili nel set di alternative. Quest'ultimo è stato ridotto a causa della presenza del veicolo *oncoming*; in seguito all'introduzione nel modello della funzione *oncomingCheck*. Tuttavia, le combinazioni disponibili, ovvero quelle che prevedono il cambio corsia, portano tutte ad una collisione con i veicoli della fila, in quanto il veicolo *ego* aveva già iniziato la manovra di sorpasso. Questa disamina porta a considerare le due problematiche enunciate correlate fra loro, poiché è a causa del primo cambio di corsia che il veicolo inizia una manovra di sorpasso che non può portare a termine.

Soluzione adottata

La logica con cui è stata affrontata questa criticità segue i principi adottati nel caso studio precedente. Si specifica che tale strategia è stata sviluppata supponendo di avere a disposizione i dati di ogni veicolo presente nello scenario; dunque, supponendo di equipaggiare il veicolo *ego* di un sistema V2X (*vehicle-to-everything*).

1. Riconoscere la criticità imminente. In tal caso si tratta di non limitare l'analisi dello scenario ai soli veicoli MIO, ma di individuare la presenza della fila di veicoli ed in particolare individuare quale fra i suoi componenti è il veicolo *front*.
2. Basare il processo su parametri dedicati e con una strategia dedicata a tale criticità. Questo è necessario poiché un risultato valido sarebbe ottenibile

incrementando il valore del parametro limite $TTC_{oncoming}$ utilizzato nella funzione *oncomingCheck*, introdotta nel caso studio "Sorpasso con veicolo in senso di marcia opposto". Tuttavia, tale soluzione non è stata adottata in quanto per scenari dove si ha un singolo veicolo che precede il veicolo *ego*, un valore del parametro $TTC_{oncoming}$ impedirebbe la manovra di sorpasso a distanze sufficienti per svolgere la manovra ancora in piena sicurezza.

3. Stabilire un criterio che permetta di discernere quando la manovra di sorpasso può comunque essere effettuata in sicurezza, anche se ciò che deve essere sorpassato è un'intera fila di veicoli.
4. Anche in questo caso è stato scelto di agire a monte dell'intero processo di pianificazione. Nello specifico è stato scelto di rimuovere dalle successive fasi di creazione e valutazione della traiettoria tutti quegli stati terminali che non rispettano i criteri imposti, ovvero quelli che portano ad incorrere in criticità analoghe a quella presentata.

In figura 4.38 è stato riportato un diagramma di flusso realizzato con l'obiettivo di aiutare la comprensione della funzione introdotta per adattare al modello anche a tale criticità. Essa ha il nome *leadExtensionCheck* poiché il suo scopo principale è quello di verificare l'ingombro longitudinale dei veicoli che devono essere sorpassati nel corso della manovra di sorpasso.

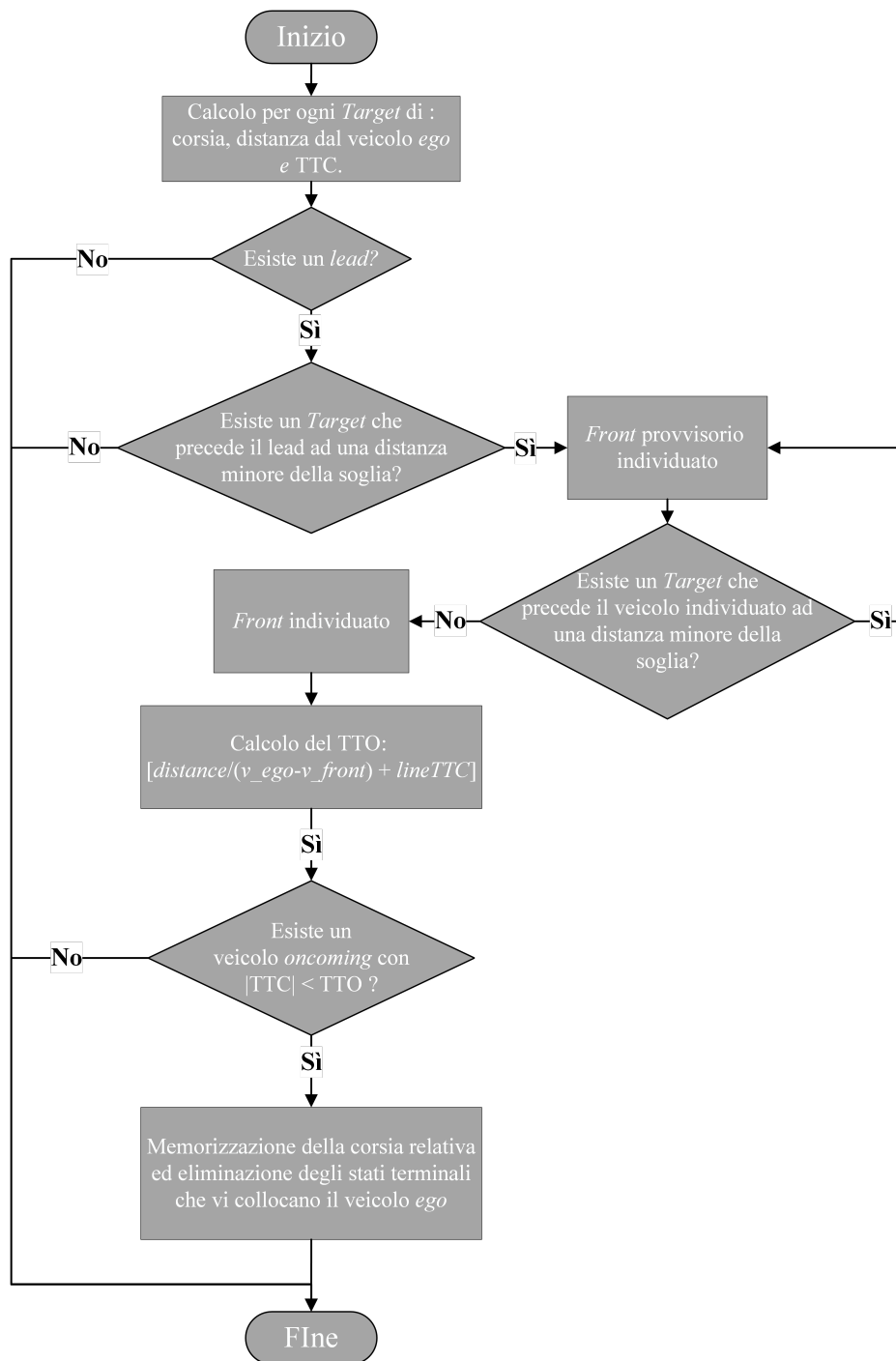


Figura 4.38: Diagramma di flusso *function: leadExtensionCheck* - scenario "Sorpasso di una fila di veicoli"

- Il primo compito della funzione è quello di ricavare tutti i parametri necessari alle valutazioni effettuate in seguito, come: corsia dei veicoli *Target*, distanza degli stessi dal veicolo *ego* ed il relativo *TTC*. Si noti che in questo caso, lavorando con informazioni relative a veicoli che possono avere altri veicoli interposti tra se ed il veicolo *ego*, non possono essere usate le informazioni relative ai soli veicoli MIO, ma devono essere verificati i dati di tutti i veicoli presenti nello scenario (veicoli *Target*).
- Successivamente è necessario verificare la presenza di un veicolo *lead* nello scenario; in caso contrario, non essendoci un veicolo da sorpassare non può presentarsi tale criticità.
- Nel caso fosse presente un veicolo *lead* è necessario verificare la presenza di una fila di veicoli; per far ciò è stata scelta una distanza soglia tra i veicoli per considerarli parte di una fila. La soglia scelta è la seguente:

$$\textit{Threshold distance} = \textit{Front SafetyGap} + \textit{Rear SafetyGap} \quad (4.7)$$

La ragione per la quale è stata scelta questa soglia è correlata all'eventuale impossibilità del veicolo *ego* di effettuare un rientro in corsia di emergenza tra i due veicoli; in quanto per poter eseguire tale manovra esso deve avere una distanza pari a *Front SafetyGap* dal veicolo che lo precede e pari a *Rear SafetyGap* dal veicolo che lo segue. Conseguentemente se la distanza tra due veicoli fosse inferiore alla somma delle due distanze di sicurezza il veicolo *ego* non potrebbe collocarsi nello spazio che li separa; dunque, i due veicoli possono essere considerati parte di una fila.

- Dopo aver individuato un eventuale veicolo che rispetta la precedente condizione, la verifica viene effettuata alla ricerca di un ulteriore mezzo che precede quello appena individuata. Il loop procede fino a quando non si individua il veicolo che "apre" la fila, veicolo che viene denominato *front*.

- Viene calcolata una stima approssimativa del tempo necessario al sorpasso dell'intera fila TTO (*time to overtake*).

$$TTO = \frac{s_{ego} - s_{front}}{\dot{s}_{ego} - \dot{s}_{front}} + lineTTC \quad (4.8)$$

I parametri sono stati calcolati rispetto al veicolo *front*, poiché esso è il più distante dal veicolo *ego* fra quelli appartenenti alla fila. Inoltre, è stata aggiunta una costante denominata $lineTTC$ che assume il valore del parametro $egoTTC$, ovvero la soglia standard sul TTC. Quest'ultimo ha un valore pari a 4s, il che è ritenuto sufficiente per il rientro in corsia dopo il superamento del veicolo *front*, in quanto il pianificatore ha un orizzonte temporale massimo di 3s.

- Viene verificata la presenza di veicoli transitanti con senso di marcia opposto ed in avvicinamento. Qualora tale verifica desse esito positivo la funzione memorizza la corsia dei veicoli che rispettano la seguente condizione:

$$|TTC| < TTO \quad (4.9)$$

- Dopo aver memorizzato le corsie dei veicoli ad elevato rischio di collisione, la funzione procede in maniera del tutto analoga a quella illustrata per la funzione *OncomingCheck*; ovvero, essa elimina dal set di alternative tutti gli stati terminali che prevedono il collocamento al dato orizzonte temporale del veicolo *ego* in una fra le corsie individuate a valle della verifica precedente.

4.4.3 Risultati simulazione - caso 3 con risoluzione criticità

Dopo aver approfonditamente esposto gli aggiornamenti apportati al modello con l'introduzione della funzione *leadExtensionCheck*, è stata eseguita una nuova simulazione dello scenario presentato in questo caso studio. Ciò è stato utile per poter valutare l'effettivo adattamento del modello alla criticità riscontrata. In figura 4.39 si ha una rappresentazione sul piano cartesiano *XY* della traiettoria effettivamente seguita dal veicolo *ego*.

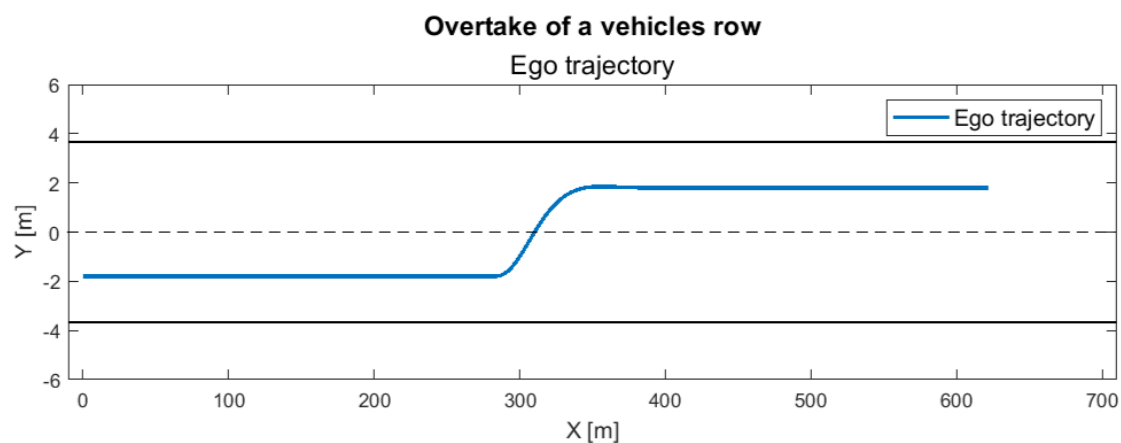


Figura 4.39: Traiettoria veicolo *ego* - scenario "Sorpasso di una fila di veicoli"

Si riportano di seguito nella tabella 4.14 i valori di picco delle grandezze che caratterizzano il moto del veicolo.

Parameter	Max	Limit	Unit
Yaw rate	6.05	20	$^{\circ}/s$
Long. Acceleration	$9.41 \cdot 10^{-2}$	5	m/s^2
Lat. Acceleration	1.64	5	m/s^2
Long. Jerk	$8.57 \cdot 10^{-3}$	5	m/s^3
Lat. Jerk	0.28	5	m/s^3

Tabella 4.14: Valori di picco grandezze veicolo *ego* - scenario "Sorpasso di una fila di veicoli"

La simulazione ha avuto esito positivo, non sono stati riscontrati stop come quello registrato in assenza della funzione *leadExtensionCheck*, la traiettoria non è caratterizzata da manovre prive di utilità come due dei tre cambi di corsia nella simulazione precedente e i parametri descrittivi del moto risultano essere ben al di sotto della soglia imposta.

Nella figura 4.40 sono rappresentati mediante capsule gli istanti di maggior rilievo della simulazione. Si precisa che non è stato incluso l'istante iniziale, poiché già presentato nella fase di introduzione del paragrafo.

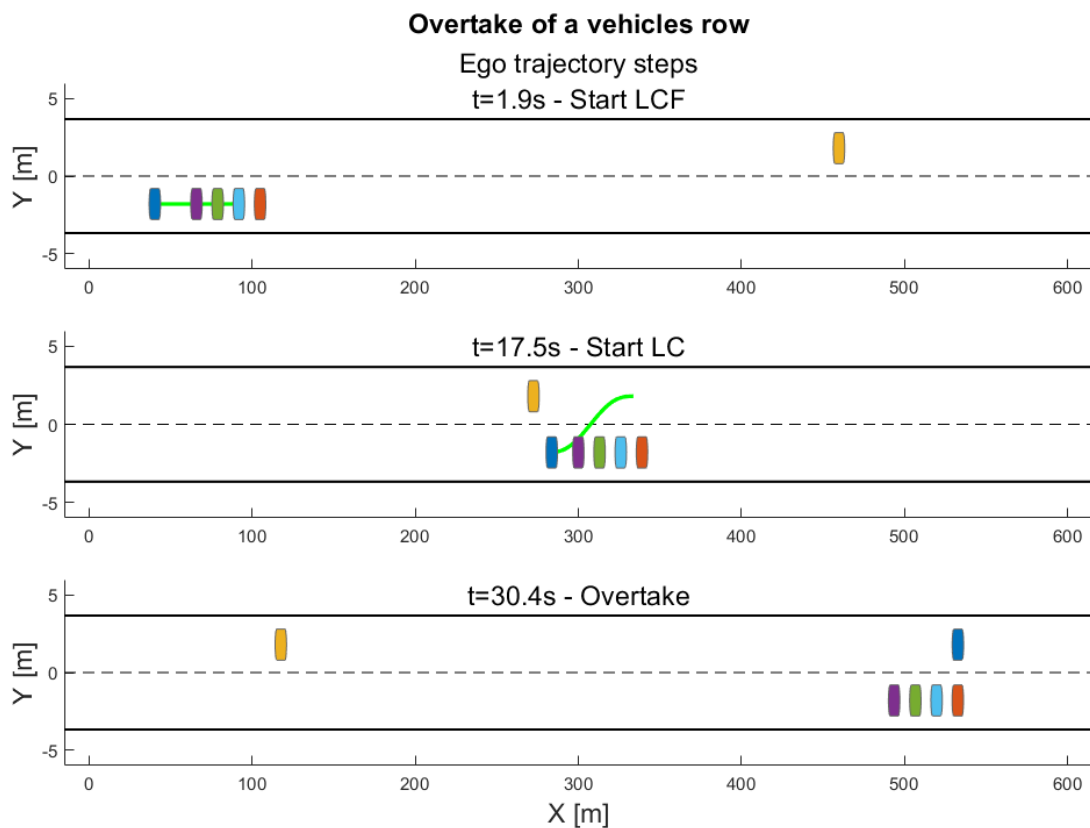


Figura 4.40: Frame traiettoria veicolo *ego* - scenario "Sorpasso di una fila di veicoli"

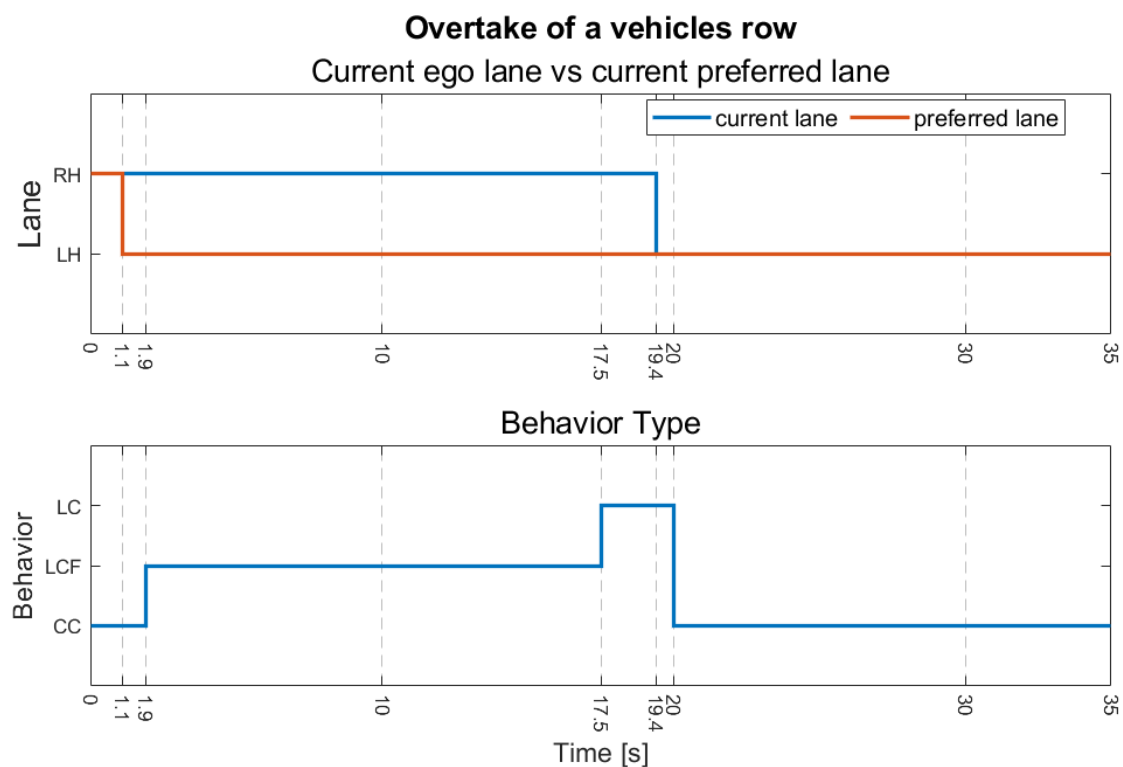


Figura 4.41: Corsia veicolo *ego* - scenario "Sorpasso di una fila di veicoli"

In figura 4.41 sono stati riportati gli andamenti relativi alla corsia corrente e preferenziale del veicolo *ego* e alla modalità di funzionamento dello stesso nel corso del tempo. Gli istanti più rilevanti sono stati riportati in valore numerico nell'asse delle ascisse. Dal primo grafico si può notare come la funzione di cambio corsia sia stata inibita nella fase iniziale della simulazione, poiché corsia preferenziale e corsia corrente risultano essere differenti per un intervallo non esiguo di tempo.

La figura 4.42 sovrappone gli andamenti di alcune grandezze rilevanti per il pianificatore. Nello specifico sono state riportate: la *Relative distance* tra i veicoli *ego* e *lead*, *ego* e *front* ed *ego* e *oncoming*, inoltre, è stato riportato anche l'andamento del *TTC* relativo al veicolo *oncoming* in comparazione con il *TTO* relativo alla fila di veicoli.

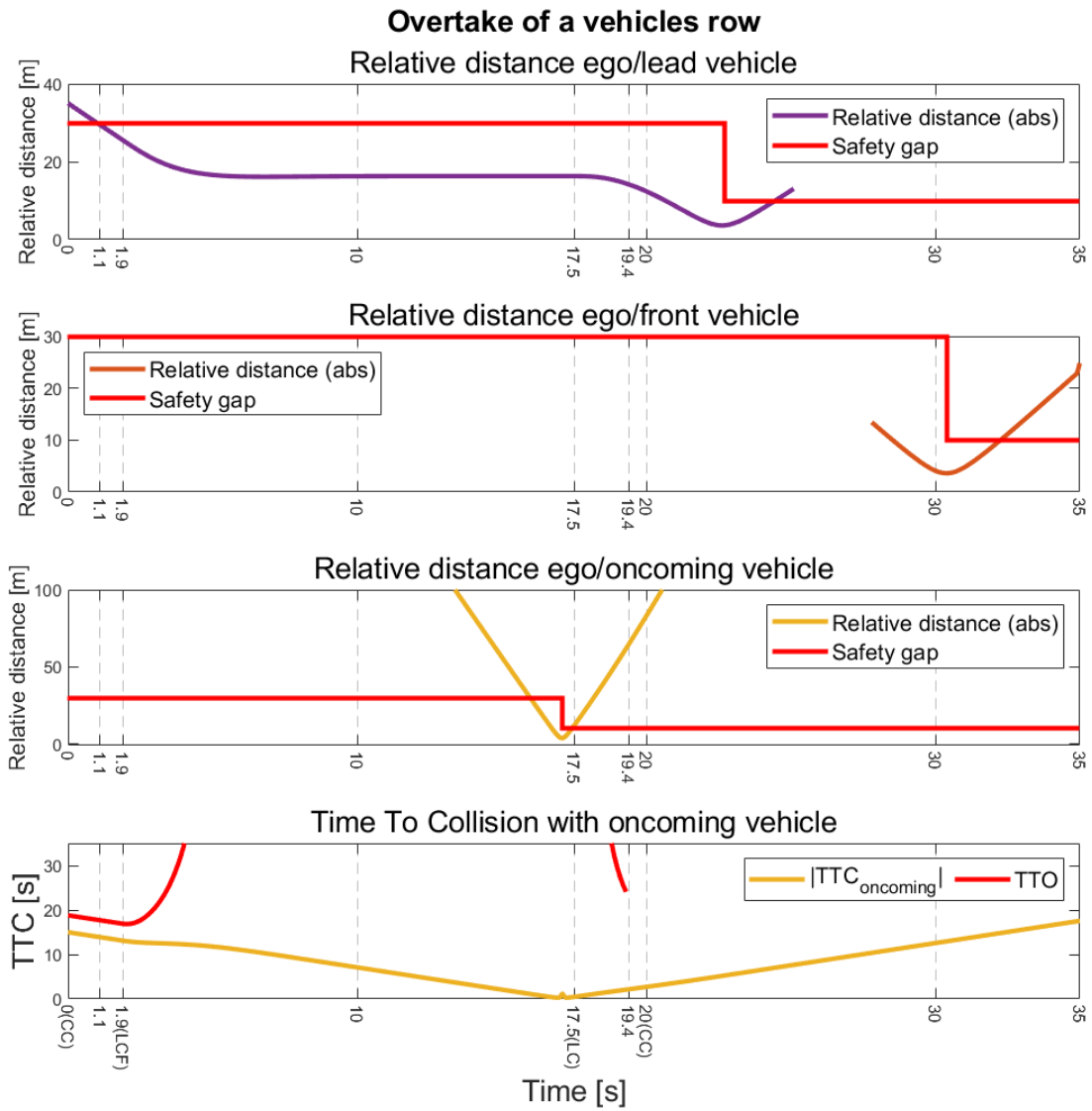


Figura 4.42: Valutazione veicoli MIO - scenario "Sorpasso di una fila di veicoli"

Si effettua di seguito un'analisi per punti dei risultati ottenuti dalla simulazione, commentando gli istanti di maggior rilievo in ordine cronologico.

- $t = 1.1$ s: Il veicolo *ego* si avvicina al veicolo *lead* tanto da oltrepassare la soglia minima sulla *Relative distance*. Coerentemente con quanto visto fin'ora ci si aspetterebbe l'attivazione della modalità LC a fronte di una variazione di una corsia preferenziale; tuttavia la neo-introdotta funzione *leadExtensionCheck* ne impedisce l'attivazione. Ciò accade poiché il valore del *TTC* calcolato rispetto al veicolo *oncoming* è inferiore al valore del parametro *TTO*, come mostrato nel quarto grafico in figura 4.42.
- $t = 1.9$ s: Non potendo variare la propria corsia e procedendo ad una velocità superiore rispetto alla fila di veicoli, il veicolo *ego* passa alla modalità LCF per non incorrere in una collisione. Questa tipologia di cambio di modalità avviene in seguito a valutazioni effettuate sul parametro di costo associato ad ogni stato terminale.
- $t = 17.1$ s: Il veicolo *oncoming* oltrepassa il veicolo *ego* rispetto alla coordinata curvilinea s ; come si nota dalla variazione di pendenza della *Relative distance* riportata nel terzo grafico in figura 4.42. Dunque, viene introdotta la modalità LC nel set di alternative di stati terminali.
- $t = 17.5$ s: Il valore della *Relative distance* calcolata rispetto al veicolo *oncoming* assume un valore superiore rispetto alla soglia *Rear SafetyGap*, ciò implica l'assenza di veicoli *Unsafe* nella corsia di sinistra. Nell'istante corrente viene pianificato il nuovo segmento di traiettoria, abilitando la modalità LC; disponibile poiché il veicolo *oncoming* è ormai transitato.
- $t = 19.4$ s: Il veicolo *ego* raggiunge la corsia di sinistra, come si può vedere dall'andamento della corsia corrente in figura 4.41. Questa coincide con la corsia preferenziale, conseguentemente il veicolo entra in modalità CC quando viene pianificato un nuovo segmento di traiettoria a $t = 20.0$ s.
- $t = 30.4$ s: Il veicolo *ego* supera rispetto alla curva di riferimento il veicolo *front*.

- $t = 35$ s: Fine della simulazione.

In conclusione, è possibile affermare che l'introduzione della funzione *leadExtensionCheck* ha avuto riscontro positivo; rendendo il modello in grado di adattarsi a scenari che presentano criticità come quelle illustrate in questo caso studio.

4.5 Caso 4 - Logiche di scelta corsia per il veicolo ego

Il caso "*Logiche di scelta corsia per il veicolo ego*" è stato introdotto con il fine di implementare all'interno del modello una strategia relativa al posizionamento del veicolo *ego*, che non sia esclusivamente dipendente dalla presenza di altri veicoli nelle vicinanze. Ciò è stato ritenuto opportuno poiché, come è possibile visionare dai risultati delle simulazioni condotte nel corso dei casi studio precedenti, analizzati nei paragrafi 4.2, 4.3 e 4.4, in seguito al superamento di un generico veicolo, il sistema pianifica la traiettoria del veicolo *ego* in modo che esso continui a transitare lungo la corsia di sorpasso. Questa pianificazione risulta essere errata sia perché non rispetta le norme del codice della strada, illustrate approfonditamente in seguito, sia perché in un caso reale si traduce in un elevato rischio per il veicolo ed i propri occupanti.

Lo scenario proposto colloca il veicolo *ego* in un tratto autostradale con tre corsie ad un unico senso di marcia. Tale alternativa è stata ritenuta la migliore per questo caso studio; in quanto, avendo a disposizione un numero maggiore di corsie rispetto agli scenari visti in precedenza, risulta avere un grado di complessità utile a sviluppare una strategia priva di lacune.

Entrando maggiormente nel dettaglio, lo scenario ospita quattro veicoli, compreso il veicolo *ego*, lungo un strada rettilinea a tre corsie ad un unico senso di marcia.

Lunghezza strada	1000 <i>m</i>
n°corsie	3

Tabella 4.15: Dati strada - "*Logiche di scelta corsia per il veicolo ego*"

I dati dati che caratterizzano il posizionamento e la velocità dei veicoli sono riportati in tabella 4.16. Si specifica che i veicoli *obstacle* mantengono velocità costante e traiettoria rettilinea lungo l'intera durata della simulazione. Inoltre, le corsie sono state numerate, adottando il punto di vista del veicolo *ego*, da sinistra verso destra.

$\text{lane}_{ego} (t = 0s)$	3
$\text{lane}_{obstacle 1}$	3
$\text{lane}_{obstacle 2}$	2
$\text{lane}_{obstacle 3}$	2
$s_{obstacle 1} - s_{ego} (t = 0s)$	100 m
$s_{obstacle 2} - s_{ego} (t = 0s)$	90 m
$s_{obstacle 3} - s_{ego} (t = 0s)$	225 m
v_{ego}	35 m/s
$v_{obstacle 1}$	20 m/s
$v_{obstacle 2}$	25 m/s
$v_{obstacle 3}$	25 m/s

Tabella 4.16: Dati veicoli - scenario "Sorpasso di una fila di veicoli"

Nel primo grafico in figura 4.43 si ha una rappresentazione mediante capsule dello scenario all'istante iniziale. Sono stati inseriti dei vettori rappresentanti in scala le velocità dei veicoli, ognuna delle quali è riportata nella legenda con l'associato valore numerico. In rosso si ha la curva di riferimento utilizzata per il sistema di riferimento di *Frenét*; per uniformità della trattazione anche in questo caso è stata scelta come curva di riferimento la *center line* della carreggiata.

Nel grafico in basso si ha una rappresentazione dello scenario in assenza di un intervento del sistema assistenziale. In particolare, è stato riportato lo scenario all'istante $t = 6.3 s$, istante in cui viene registrata una collisione fra il veicolo *ego* ed il veicolo *obstacle 1*. Dunque, risulta evidente la necessità di variare la traiettoria del veicolo *ego* o i valori associati alle grandezze che ne caratterizzano il moto.

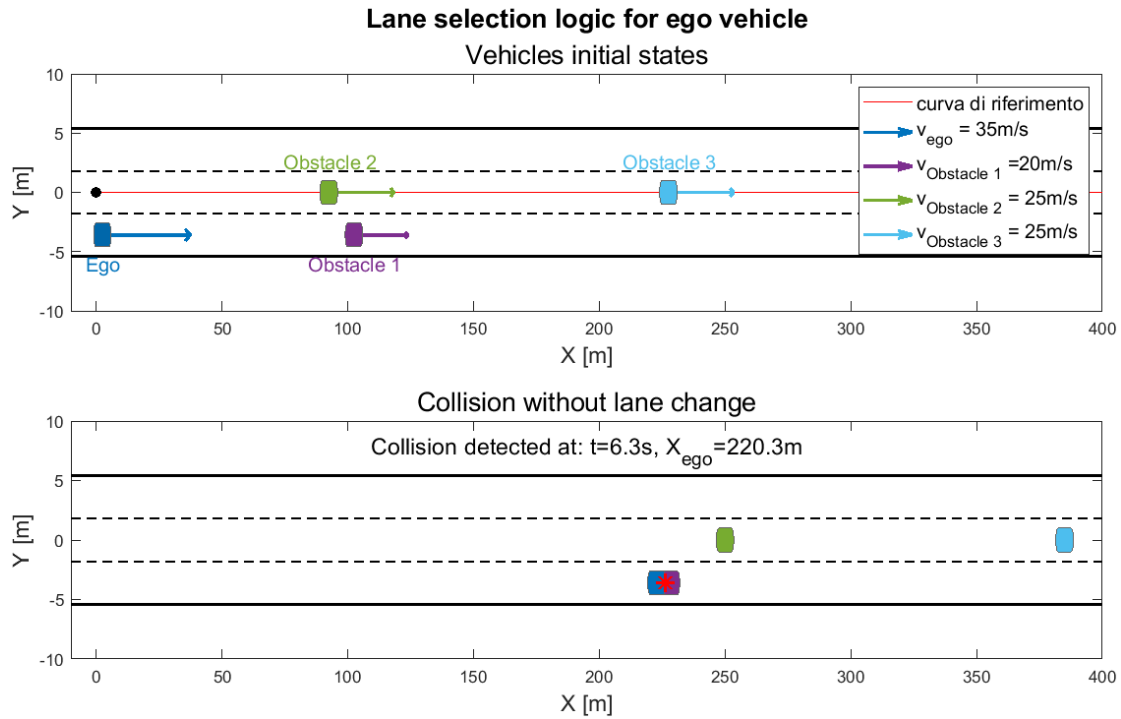


Figura 4.43: Stati iniziali $t=0s$ - scenario "Logiche di scelta corsia per il veicolo ego"

Come mostrato in figura 4.43, il veicolo *ego* sopraggiunge rapidamente al veicolo *obstacle 1* (in viola); contemporaneamente quest'ultimo è oggetto di sorpasso a beneficio del veicolo *obstacle 2* (in verde). Infine, ad una distanza maggiore viaggia irregolarmente lungo la corsia centrale il veicolo *obstacle 3* (in azzurro).

4.5.1 Risultati simulazione - caso 4

La simulazione è stata portata a termine in totale assenza di collisioni tra il veicolo *ego* e gli altri veicoli presenti nello scenario grazie all'intervento del sistema assistenziale. Esso è intervenuto in seguito al rilevamento di un'incipiente collisione con il veicolo *obstacle 1*, mutando la corsia preferenziale del veicolo *ego*, ed in un secondo caso a causa di un progressivo avvicinamento nei confronti del veicolo *obstacle 2*. Successivamente il veicolo ha mantenuto la corsia di sinistra fino al termine della simulazione.

Si riporta in figura 4.44 la traiettoria seguita dal veicolo *ego* durante la simulazione.

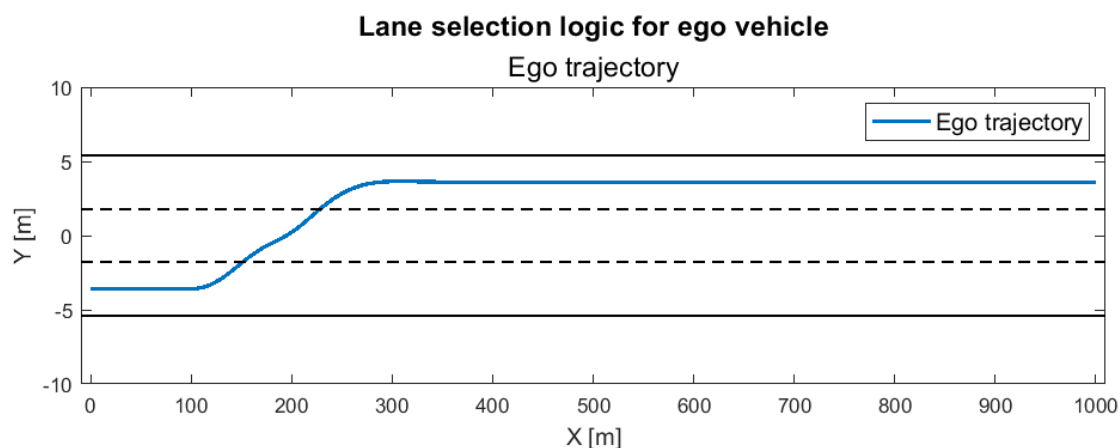


Figura 4.44: Traiettoria veicolo *ego* - scenario "*Logiche di scelta corsia per il veicolo ego*"

Per una migliore comprensione dell'evoluzione degli stati del veicolo *ego* nel corso della simulazione, in figura 4.45 sono state sovrapposte delle rappresentazioni mediante capsule del posizionamento dei veicoli all'interno dello scenario negli istanti di maggior rilievo. Inoltre, si specifica che non è stato inserito lo scenario iniziale in quanto esso è già stato presentato in figura 4.43. In ognuno dei grafici in figura è stata riportato in verde il segmento di traiettoria da seguire fornito dal pianificatore.

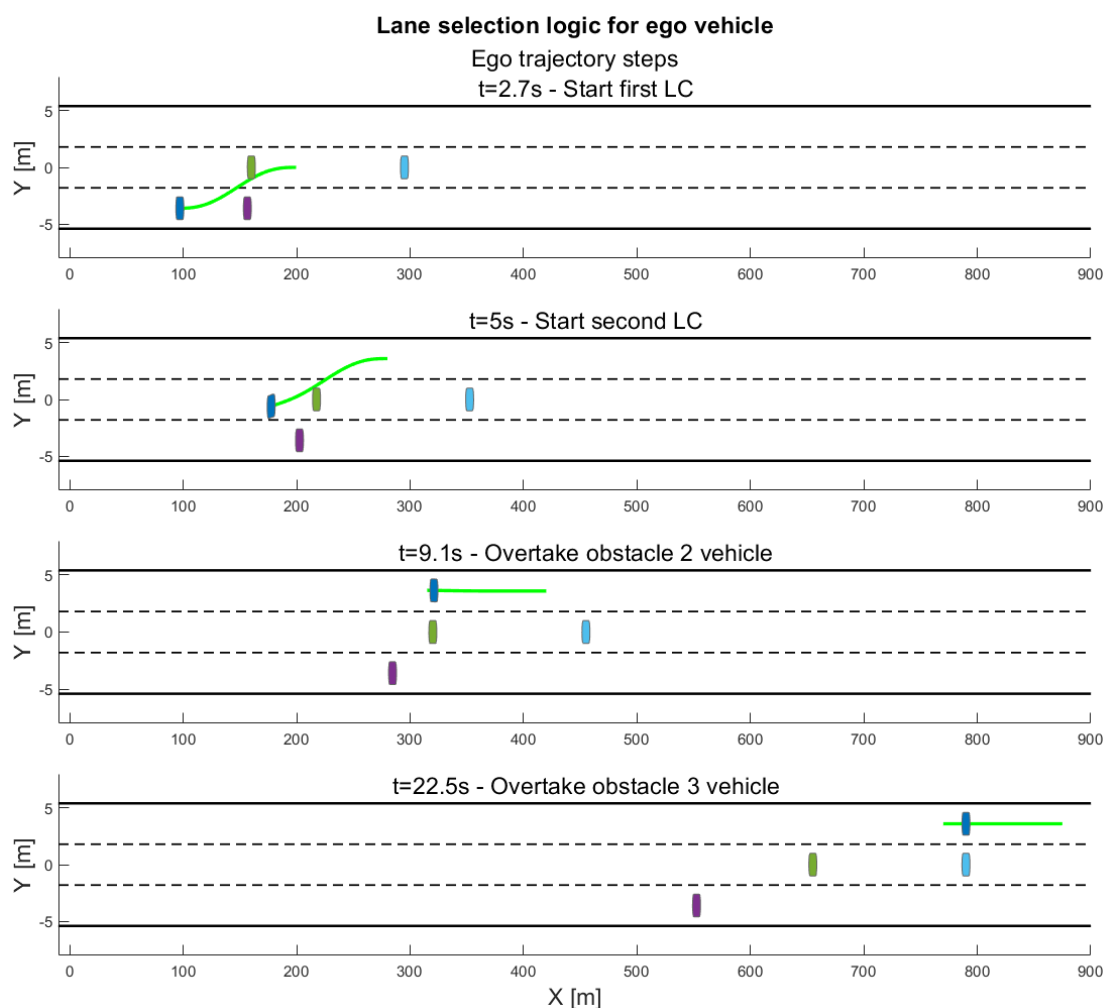


Figura 4.45: Frame traiettoria veicolo *ego* - scenario "Logiche di scelta corsia per il veicolo *ego*"

Coerentemente con quello che è lo scopo di questo caso studio, ovvero quello di sviluppare una strategia circa il corretto posizionamento del veicolo *ego* all'interno della carreggiata, viene proposta una trattazione incentrata sulle cause che hanno portato il pianificatore ad individuare le corsie sulle quali far transitare il veicolo *ego*. Dunque, è stato ritenuto non rilevante per tale obiettivo il contributo dell'analisi dei parametri descrittivi del moto del veicolo; di conseguenza i relativi andamenti non sono stati inclusi nella successiva analisi.

Di fatto, non si tratta di un'analisi votata all'individuazione di particolari problematiche che non hanno permesso di portare a termine la simulazione; in quanto, quest'ultima ha avuto esito positivo, non presentando errori e offrendo una traiettoria fisicamente percorribile dal veicolo. Si riportano nella tabella 4.17 i valori di picco delle grandezze descrittive del moto; si nota che nessuna grandezza oltrepassa il relativo limite imposto.

Parameter	Max	Limit	Unit
Yaw rate	3.14	20	$^{\circ}/s$
Long. Acceleration	$8.52 \cdot 10^{-3}$	5	m/s^2
Lat. Acceleration	1.93	5	m/s^2
Long. Jerk	$2.24 \cdot 10^{-3}$	5	m/s^3
Lat. Jerk	0.43	5	m/s^3

Tabella 4.17: Valori di picco grandezze veicolo *ego* - scenario "*Logiche di scelta corsia per il veicolo ego*"

In figura 4.46 sono stati riportati gli andamenti relativi alla corsia preferenziale, a quella corrente ed alle modalità di funzionamento del veicolo (CC, LCF e LC).

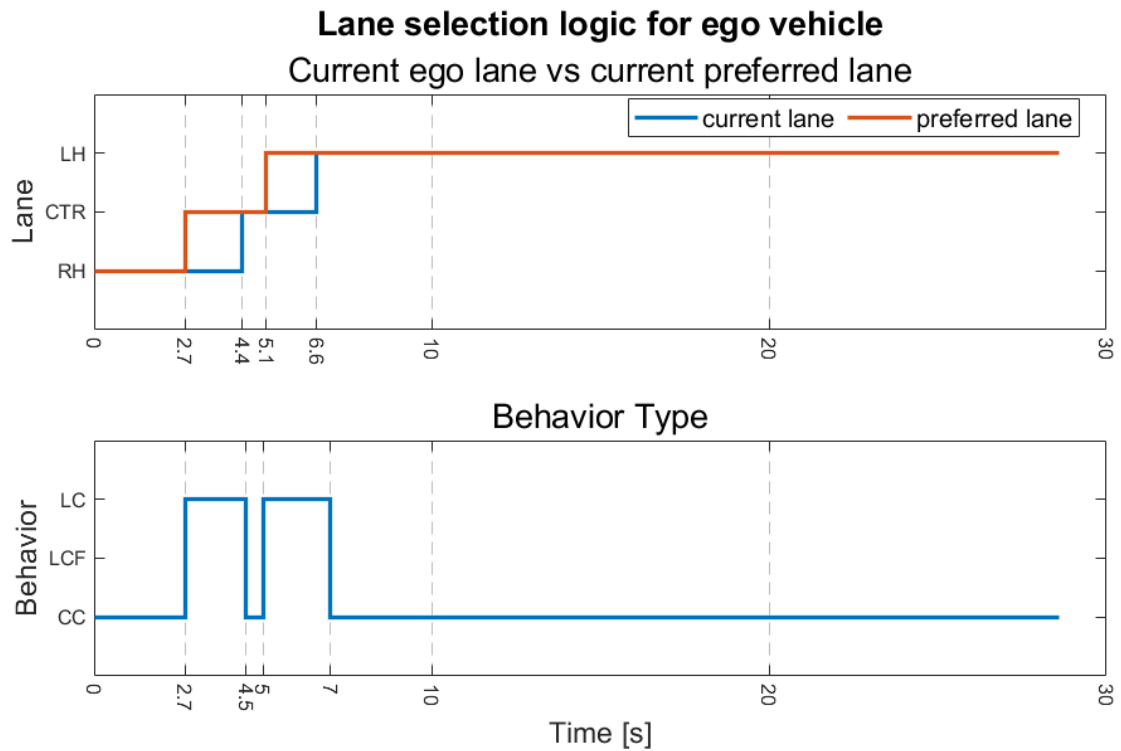


Figura 4.46: Corsia veicolo *ego* - scenario "Logiche di scelta corsia per il veicolo *ego*"

Nella seguente figura 4.47 si hanno gli andamenti dei segnali *TTC*, *Relative distance* e valutazione di sicurezza *Safe/Unsafe* calcolati rispetto ai veicoli *obstacle 1* (in viola) e *obstacle 2* (in verde).

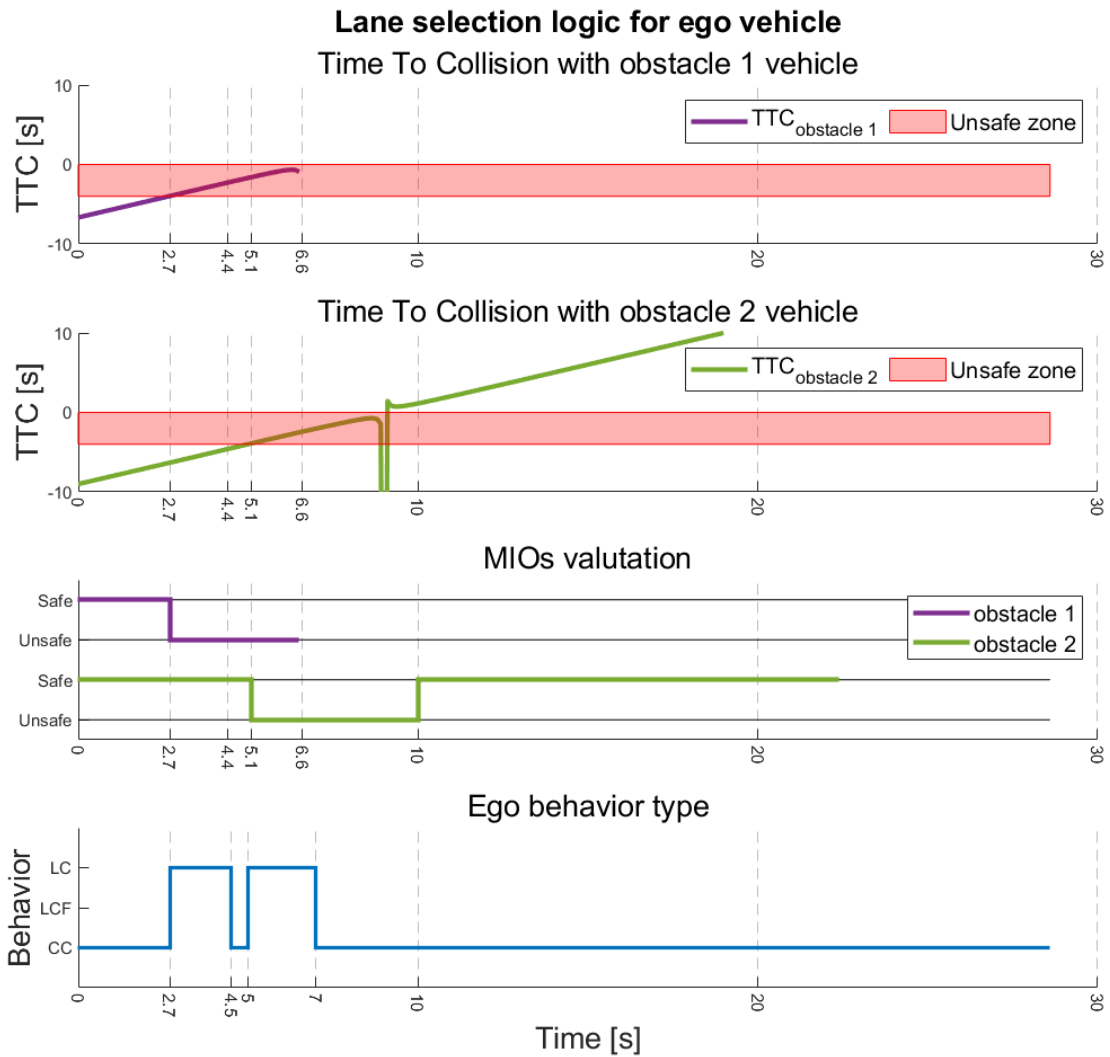


Figura 4.47: Valutazione veicoli MIO - scenario "Logiche di scelta corsia per il veicolo ego"

Si specifica che qualora i segnali riportati non dovessero essere visibili per tutto l'intervallo temporale della simulazione ($t = 28.5\ s$), ciò è dovuto a valori eccessivamente grandi del segnale rappresentato da non essere comparabili con la soglia imposta, dunque si ha una fuoriuscita del segnale dai limiti delle ordinate del relativo grafico, o alla perdita di status di veicolo *MIO*.

Non sono stati proposti i grafici che mostrano l'andamento del *TTC* e della *Relative distance* calcolati rispetto al veicolo *obstacle 3*, in quanto esso non si trova mai nella stessa corsia del veicolo *ego* se non prima del sorpasso ai danni del veicolo *obstacle 2* ($t = 5 s$). In tale circostanza i veicoli *ego* e *obstacle 3* sono separati dal veicolo *obstacle 2*; dunque, il veicolo *obstacle 3* non può essere definito un veicolo MIO.

Analizzando e confrontando fra loro i grafici riportati si hanno tutte le informazioni necessarie per fornire una completa interpretazione del comportamento del sistema assistenziale, dunque del veicolo *ego*, nel corso della simulazione. Si procede per punti, commentando gli istanti di maggior rilievo in ordine cronologico.

- Da $t = 0 s$ a $t = 2.6 s$: Il veicolo *ego* segue la traiettoria preimpostata durante la fase di costruzione dello scenario senza riscontrare la necessità di una nuova pianificazione della stessa.
- $t = 2.7 s$: Il *TTC* calcolato rispetto al veicolo *obstacle 1* assume un valore in modulo inferiore del valore soglia (primo grafico in figura 4.47); ciò porta alla definizione di veicolo *Unsafe*. Dunque, la corsia di destra, corsia nella quale transita il veicolo *obstacle 1* ed inizialmente anche il veicolo *ego*, viene dichiarata *Unsafe* data la presenza del veicolo *obstacle 1*. Ciò comporta una variazione di *preferred line*, la quale a sua volta porta alla pianificazione di una nuova traiettoria in modalità LC, come si può notare dagli andamenti in figura 4.46.
- $t = 4.4 s$: Il veicolo *ego*, dopo il cambio corsia, inizia a transitare lungo la corsia centrale; essendo quest'ultima la corsia di preferenza. Quando viene ripianificata la traiettoria ($t = 4.5 s$) viene utilizzata la modalità CC.
- $t = 5 s$: Nonostante corsia preferenziale e corrente siano uguali, si nota dal secondo grafico in figura 4.46 come venga utilizzata la modalità di cambio corsia in tale istante. Valutando il set di stati terminali disponibili, emerge che le traiettorie ottimali sono quelle pianificate con modalità CC, ovvero che prevedano un mantenimento della corsia attuale da parte del veicolo *ego*; tuttavia, tali stati terminali portano tutti a traiettorie che incorrono in collisioni

con il veicolo *obstacle 2*. Si ricorda che successivamente all'interpretazione dei segnali *TTC* e *Relative distance* calcolati rispetto ai veicoli MIO, una traiettoria viene verificata attraverso la funzione *checkCollision* fornita dal *Navigation Toolbox*, grazie alla quale è possibile confrontare le traiettorie future di due o più veicoli per verificarne un'eventuale futura collisione. Tale è il motivo per il quale viene utilizzata la modalità di cambio corsia, sebbene il veicolo *ego* si trovi già nella propria corsia preferenziale.

- $t = 5.1 s$: Il *TTC* calcolato rispetto al veicolo *obstacle 2* assume un valore in modulo inferiore del valore soglia (terzo grafico in figura 4.47); ciò implica una variazione della *preferred line*.
- $t = 6.6 s$: Il veicolo *ego*, dopo il cambio corsia, inizia a transitare lungo la corsia all'estrema sinistra; essendo quest'ultima la corsia di preferenza, la pianificazione viene effettuata in modalità CC. Il veicolo *ego* manterrà questa corsia fino alla fine della simulazione, in assenza di una strategia dedicata al rientro di corsia.
- $t = 22.5 s$: Come mostrato in figura 4.45 il veicolo *obstacle 3* viene superato da sinistra.

4.5.2 Adattamento del sistema alle criticità presentate - Caso 4

Come mostrano chiaramente i risultati esposti nel sottoparagrafo precedente, la traiettoria individuata non presenta criticità sulla fattibilità della traiettoria, legate a collisioni con altri veicoli dello scenario o criticità del funzionamento del pianificatore, come nei casi studio precedenti. Nonostante la traiettoria individuata sia effettivamente percorribile dal veicolo *ego*, ciò che in questo paragrafo rappresenta la problematica in esame è la mancanza di una strategia che porti il veicolo *ego* a modificare nuovamente la propria corsia, dopo aver superato il veicolo *obstacle 2* all'istante della simulazione $t = 9.1$ s. La necessità di implementare una strategia di rientro in corsia sta nell'importanza di adeguare il modello alle norme vigenti del codice della strada.

Norme del codice della strada e definizioni utili alla creazione della strategia

Per stabilire una corretta strategia circa la logica di attribuzione della corsia del veicolo è necessario analizzare le norme vigenti che regolano il comportamento dei veicoli. Tra le norme del codice della strada (dlgs. 285/1992, da qui in avanti anche c.d.s.) due articoli risultano essere particolarmente utili allo scopo proposto:

Art. 143, co. 5 c.d.s.: *«Salvo diversa segnalazione, quando una carreggiata é a due o più corsie per senso di marcia, si deve percorrere la corsia più libera a destra; la corsia o le corsie di sinistra sono riservate al sorpasso»* [31].

Art. 148, co. 3 c.d.s.: *«Il conducente che sorpassa un veicolo o altro utente della strada che lo precede sulla stessa corsia, dopo aver fatto l'apposita segnalazione, deve portarsi sulla sinistra dello stesso, superarlo rapidamente tenendosi da questo ad una adeguata distanza laterale e riportarsi a destra appena possibile, senza creare pericolo o intralcio. Se la carreggiata o semicarreggiata sono suddivise in più corsie, il sorpasso deve essere effettuato sulla corsia immediatamente alla sinistra del veicolo che si intende superare»* [32].

Il primo dei dei articoli riportati è stato ricavato dalla sezione dedicata alla posizione dei veicoli all'interno della carreggiata. Grazie ad esso è possibile stabilire che, in scenari come quello presentato in questo paragrafo, il veicolo deve necessariamente spostarsi nella corsia destra quando tale manovra è possibile.

Il secondo articolo è stato ricavato dalla sezione dedicata al sorpasso. Esso stabilisce che dopo aver superato un veicolo è necessario "*riportarsi a destra appena possibile*"; ciò è valido sia per strade a più corsie per senso di marcia che per strade ad una singola corsia per senso di marcia.

Infine, risulta essere importante dedicare attenzione alla tematica del *sorpasso da destra* in autostrada. Il sorpasso a destra, anche in autostrada, è espressamente vietato dall'articolo 148 del c.d.s. [32]; tuttavia, è fondamentale distinguere il *sorpasso da destra* dal *superamento a destra*.

I due termini apparentemente indicano la medesima manovra; tuttavia, la Cassazione ha chiarito la differenza; aggiungendo, in seguito ad alcune sentenze, che il superamento da destra è consentito a differenza del sorpasso. Il *sorpasso a destra* si ha quando il veicolo si sposta sulla destra per superare il mezzo più lento, per poi ritornare nuovamente a sinistra; invece, il *superamento a destra* si ha quando il veicolo sfilava alla destra del mezzo più lento, non spostandosi successivamente a sinistra [33]. Conseguentemente il superamento a destra è consentito quando gli spostamenti fra le corsie da parte del veicolo che supera non avvengono in sequenza tanto da poter essere considerate manovre differenti tra loro. Nel caso del "*superamento a destra*" è il mezzo che procede più lentamente a commettere un'infrazione, dunque è passibile di sanzione, poiché non rispetta l'articolo Art. 143, co. 5 del c.d.s. non mantenendo la destra.

Soluzione adottata

A valle di quanto appreso nel corso dello studio delle norme vigenti e inerenti alla tematica in esame; la soluzione adottata deve rispettare i seguenti punti:

- Deve permettere al veicolo di eseguire le manovre di spostamento nella corsia di destra appena possibile ed in sicurezza dopo un qualsiasi sorpasso.
- Deve posizionare il veicolo nella corsia di destra, se questa esiste e permette la manovra in sicurezza.
- Deve consentire il superamento e non il sorpasso a destra.

Lavorando a partire dal modello utilizzato in questa tesi, una soluzione che rispetta tutti i punti enunciati è la seguente: Verificare ad ogni step la possibilità di spostare il veicolo *ego* in una corsia alla destra di quella occupata; qualora questa esistesse, fosse libera o priva di veicoli *Unsafe* ed eseguire la manovra con una traiettoria che rispetti tutte le verifiche già predisposte all'interno del modello.

Adottando questa soluzione, in seguito ad un eventuale sorpasso da sinistra di un mezzo che procede più lentamente, il veicolo *ego* si porterà nella corsia di destra solamente dopo che questa verrà dichiarata *Safe*, ovvero priva di veicoli *Unsafe*. Questa condizione corrisponde al superamento da parte del valore di *Relative distance* della soglia denominata *Rear safety gap*; la quale indica la distanza minima dal veicolo che segue il veicolo *ego* per poter eseguire la manovra di rientro in corsia.

Analizzando il caso del *superamento a destra*, esso risulta praticabile in quanto, in maniera conforme con l'articolo Art. 143, co. 5 del c.d.s., il veicolo *ego* si sposta a destra appena possibile, indipendentemente dall'avvicinamento di un altro veicolo che lo precede; conseguentemente, se dovesse superare dalla destra un mezzo che procede più lentamente, lo farebbe in maniera conforme alle normative e non con una manovra di sorpasso irregolare.

Entrando in ambito pratico, tale soluzione è stata introdotta nel modello non mediante una nuova funzione come nei due casi studio precedenti, bensì modificando la funzione *helperFindPreferredLane* già presente nel modello.

Riassumendo la spiegazione fornita all'interno del paragrafo 3.1.4, essa è una funzione che confronta gli stati del veicolo ego e dei veicoli definiti MIO con i dati della strada, per fornire in output l'indice della corsia preferenziale. Per agevolare la comprensione è stato realizzato un diagramma di flusso in figura 4.48.

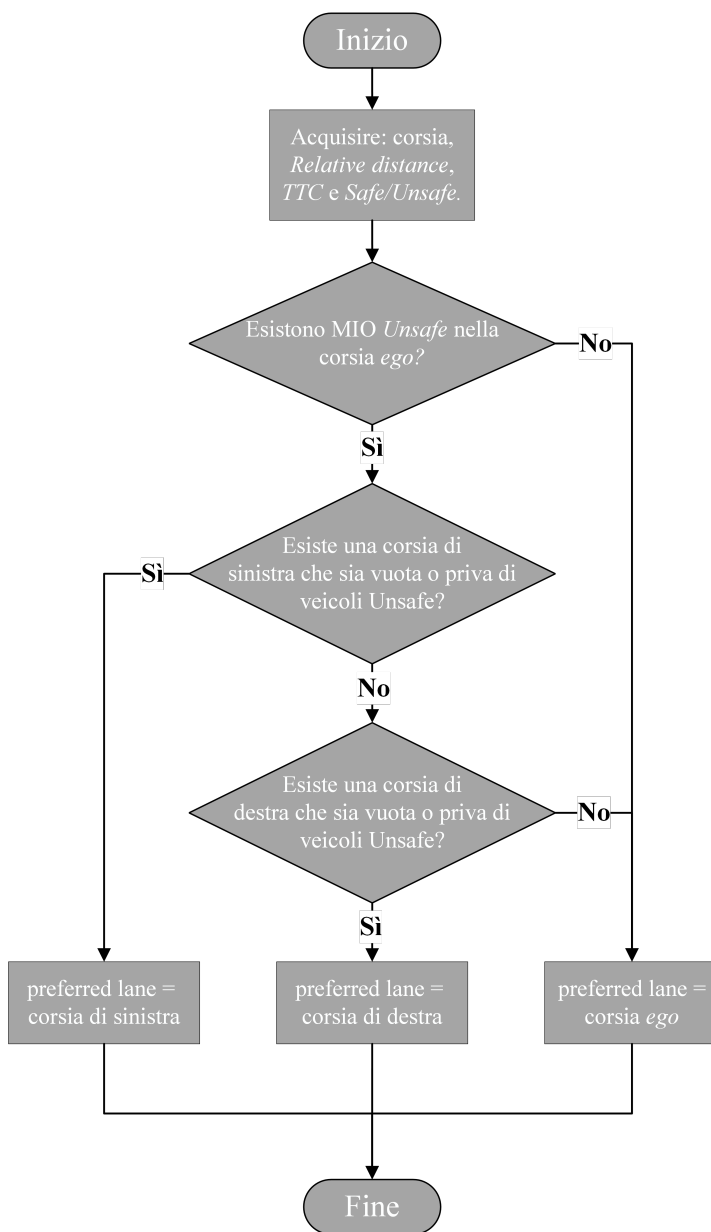


Figura 4.48: Diagramma di flusso *function: helperFindPreferredLane* - scenario "Logiche di scelta corsia per il veicolo ego"

Si fornisce di seguito l'interpretazione per punti del diagramma di flusso presentato in figura 4.48:

- Si acquisiscono i dati relativi a corsia, *Relative distance*, *TTC* e definizione *Safe/Unsafe* del veicolo *ego* e dei veicoli MIO.
- Se esistono veicoli MIO *Unsafe* nella corsia corrente del veicolo *ego* vengono valutate le corsie di destra e di sinistra; in caso contrario la corsia preferenziale viene associata a quella corrente.
- Viene valutata in primo luogo la corsia di sinistra, qualora fosse priva di veicoli *Unsafe* la corsia preferenziale viene associata a quella di sinistra; in caso contrario viene esaminata la corsia di destra.
- Viene valutata la corsia di destra, qualora fosse priva di veicoli *Unsafe* la corsia preferenziale viene associata a quella di destra; in caso contrario viene mantenuta la corsia corrente.

Per realizzare la funzione basata sulle considerazioni esposte in precedenza, la funzione *helperFindPreferredLane* ha subito delle modifiche sulla catena logiche che la caratterizza; dove per catena logica si intende l'ordine e le correlazioni che caratterizzano le verifiche che vengono effettuate. Il diagramma di flusso che ne schematizza il funzionamento è riportato in figura 4.49. Inoltre, all'interno della funzione è stato inserito un flag, che se associato alla condizione logica 0 disattiva la funzione di rientro in corsia.

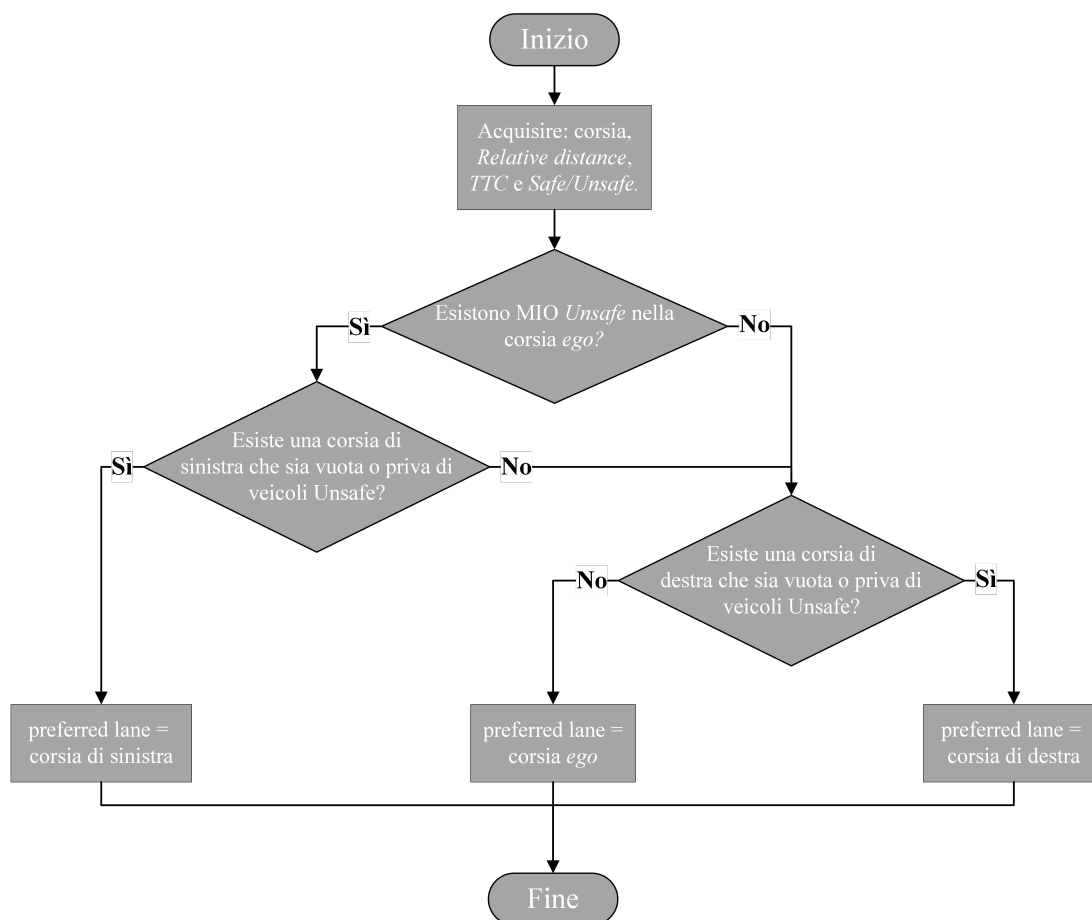


Figura 4.49: Diagramma di flusso *function: helperFindPreferredLane* con rientro in corsia - scenario "Logiche di scelta corsia per il veicolo ego"

Si fornisce di seguito l'interpretazione per punti del diagramma di flusso presentato in figura 4.49:

- Si acquisiscono i dati relativi a corsia, *Relative distance*, *TTC* e definizione *Safe/Unsafe* del veicolo *ego* e dei veicoli MIO.
- Se esistono veicoli *Unsafe* nella corsia corrente del veicolo *ego* la funzione verifica la possibilità di effettuare un cambio corsia destinato alla corsia di sinistra. Qualora questa seconda verifica desse esito positivo la corsia preferenziale viene associata a quella di sinistra; in caso contrario si passa alla verifica successiva.
- Se non esistono veicoli *Unsafe* nella corsia corrente del veicolo *ego* o se la corsia di sinistra non esiste o non risulta essere *Safe* la funzione valuta uno spostamento nella corsia di destra. Qualora questa verifica desse esito positivo la corsia preferenziale viene associata a quella di destra; in caso contrario si passa alla verifica successiva.

Da quanto descritto emerge che se la corsia del veicolo *ego* risulta essere *Unsafe* la funzione valuta un eventuale sorpasso da sinistra; inoltre, la funzione controlla costantemente se la corsia di destra risulta essere libera e ne richiede il passaggio appena possibile in maniera conforme alle norme vigenti.

Tale architettura permette al sistema assistenziale di discernere tra *superamento* e *sorpasso* da destra. Poiché se la distanza o la velocità in confronto al veicolo che lo precede sono tali da dichiarare la corsia *Unsafe*, il veicolo *ego* lo supererebbe da sinistra, in maniera conforme all'articolo Art. 148, co. 3 del codice della strada. Qualora il veicolo fosse dichiarato *Safe* il sistema sposta il veicolo *ego* sempre nella corsia libera di destra; dunque, successivamente lo farebbe semplicemente sfilare alla destra del mezzo più lento.

4.5.3 Risultati simulazione - caso 4 con risoluzione criticità

Dopo aver approfonditamente esposto gli aggiornamenti apportati al modello, è stata eseguita una nuova simulazione utilizzando il medesimo scenario presentato ad inizio paragrafo, in modo da poter valutare i risultati ottenuti a valle della nuova strategia di scelta corsia implementata. In figura 4.50 è stata rappresentata la traiettoria seguita dal veicolo *ego* all'interno del piano cartesiano XY .

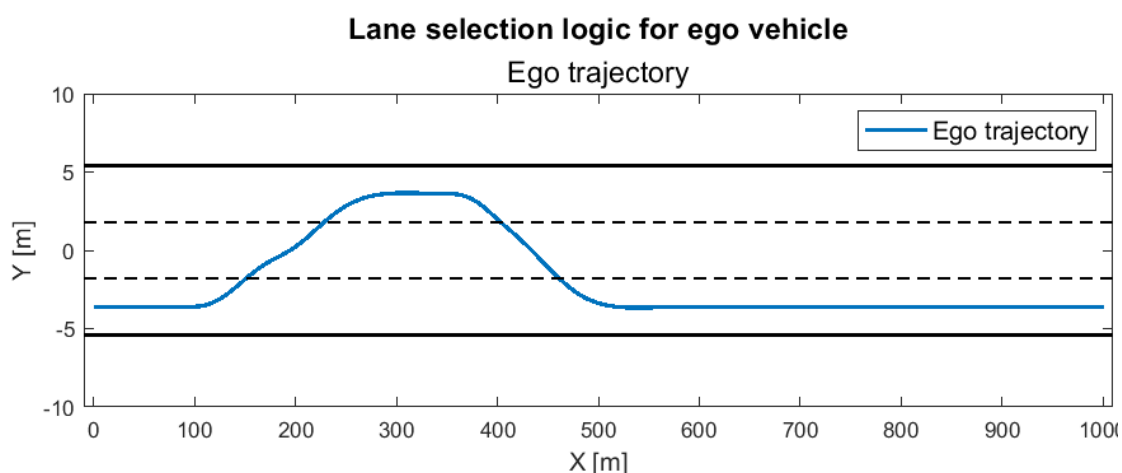


Figura 4.50: Traiettoria veicolo *ego* - scenario "Logiche di scelta corsia per il veicolo *ego*"

La simulazione ha avuto esito positivo, individuando una traiettoria priva di collisioni ed effettivamente percorribile, come mostrato in tabella 4.18, la quale riporta i valori di picco delle grandezze che descrivono il moto del veicolo.

Parameter	Max	Limit	Unit
Yaw rate	3.72	20	$^{\circ}/s$
Long. Acceleration	$1.15 \cdot 10^{-2}$	5	m/s^2
Lat. Acceleration	2.28	5	m/s^2
Long. Jerk	$2.24 \cdot 10^{-3}$	5	m/s^3
Lat. Jerk	0.43	5	m/s^3

Tabella 4.18: Valori di picco grandezze veicolo *ego* - scenario "Logiche di scelta corsia per il veicolo *ego*"

Inoltre, la nuova strategia implementata relativa alla logica di scelta della corsia del veicolo *ego* ha anch'essa avuto un riscontro positivo, come spiegato in seguito. Gli istanti di maggior rilievo sono stati riportati mediante rappresentazione con capsule in figura 4.51. Il veicolo ha effettuato un primo sorpasso ai danni del veicolo *obstacle 1*, successivamente ha sorpassato il veicolo *obstacle 2* per poi spostarsi progressivamente nella corsia libera alla propria destra fino a raggiungere la corsia di partenza. Corsia dalla quale ha superato il veicolo *obstacle 3*.

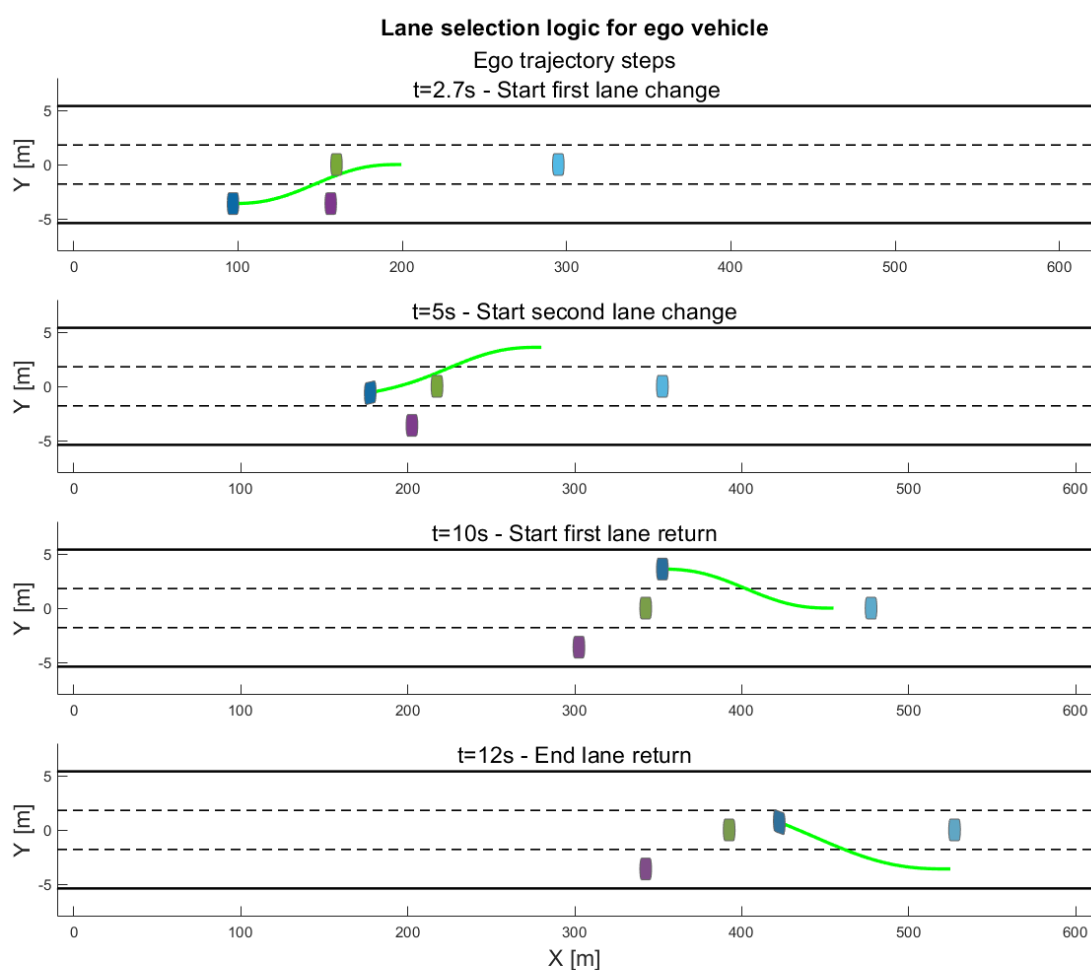


Figura 4.51: Frame traiettoria veicolo *ego* - scenario "Logiche di scelta corsia per il veicolo *ego*"

Si specifica che la trattazione da qui riportata parte dall'istante $t = 9.1 s$ in cui il veicolo *ego* supera il veicolo *obstacle 2*; in quanto, per ciò che concerne gli istanti precedenti a quello citato i risultati ottenuti sono i medesimi di quelli illustrati nel sottoparagrafo 4.5.1. Nella seguente figura 4.52 sono stati riportati gli andamenti dei segnali relativi alla corsia corrente e di preferenza del veicolo *ego* e la modalità di funzionamento adottata dal pianificatore.

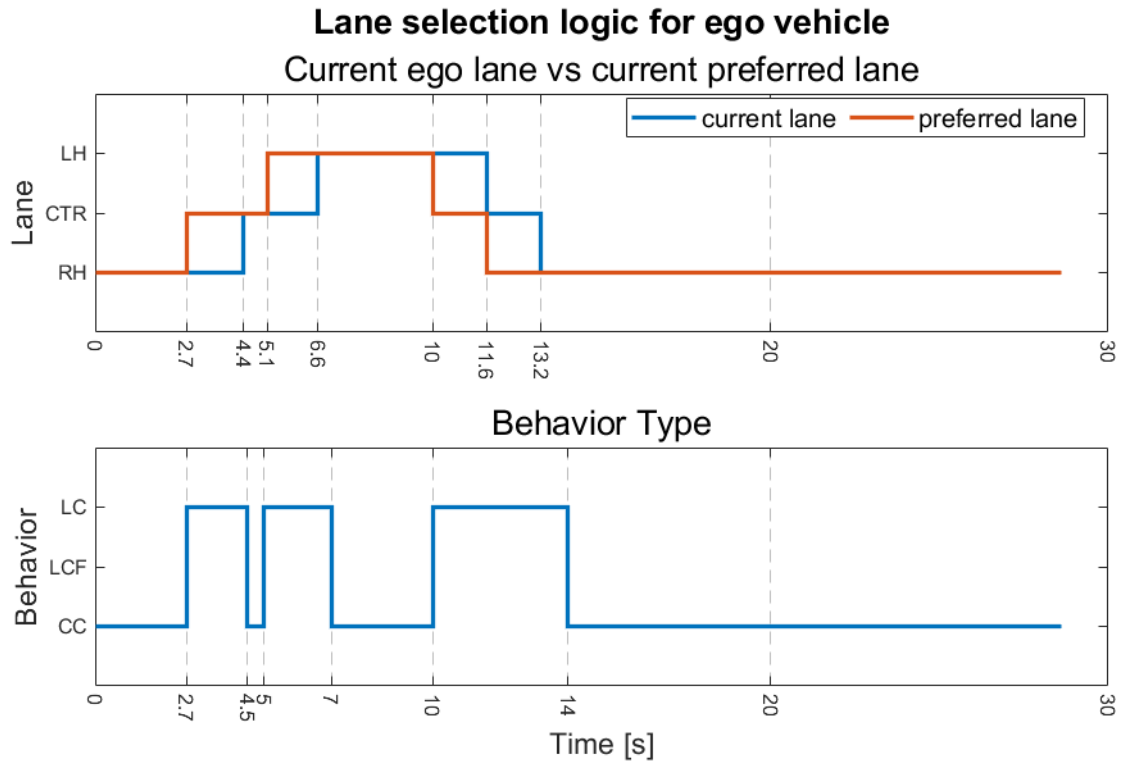


Figura 4.52: Corsia veicolo *ego* - scenario "Logiche di scelta corsia per il veicolo *ego*"

In figura 4.53 sono stati sovrapposti gli andamenti del *TTC*, della *Relative distance* e della valutazione di sicurezza (*Safe/Unsafe*) relativa rispettivamente ai veicoli *obstacle 2* e *obstacle 3*. Essi risultano utili a chiarire le ragioni di variazione della corsia corrente del veicolo *ego*.

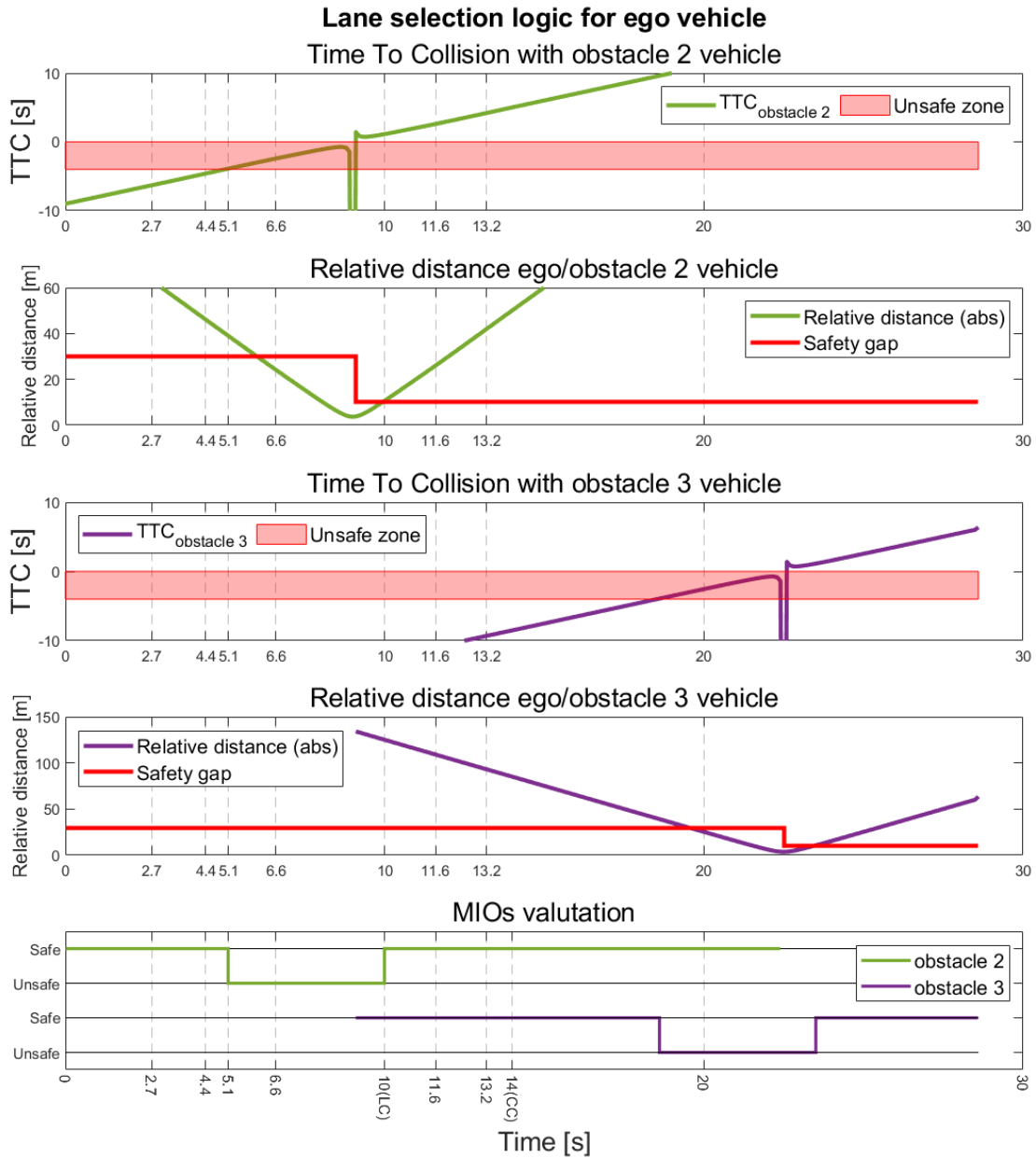


Figura 4.53: Valutazione veicoli MIO - scenario "Logiche di scelta corsia per il veicolo ego"

Si procede di seguito all'analisi per punti, commentando gli istanti di maggior rilievo in ordine cronologico; partendo come anticipato dall'istante $t = 9.1 s$, in cui il veicolo *ego* supera rispetto alla curva di riferimento il veicolo *obstacle 2*:

- $t = 9.1 s$: Il veicolo *ego* si trova nella corsia di estrema sinistra ed ha superato il veicolo *obstacle 2* rispetto alla curva di riferimento. La modalità di funzionamento è la CC; essa rimarrà tale fin quando la corsia preferenziale sarà coincidente con quella corrente, come mostrato dal grafico in figura 4.52.
- $t = 10 s$: Dall'ultimo grafico in figura 4.53 si nota che il veicolo *obstacle 2* viene dichiarato *Safe*; ciò accade in quanto la *Relative distance* oltrepassa la soglia *Rear safety gap*, rappresentata mediante una linea rossa nel secondo grafico della medesima figura. Come conseguenza, si ha una variazione istantanea di corsia preferenziale e l'attivazione della modalità LC (figura 4.52). Questo accade grazie alla nuova strategia di scelta corsia, la quale impone lo spostamento nella corsia di destra dopo aver eseguito il sorpasso.
- $t = 11.6 s$: Dal secondo grafico in figura 4.52 si nota come il veicolo *ego* oltrepassa la linea che divide la corsia di sinistra da quella centrale; contestualmente, essendo possibile spostarsi nuovamente a destra, il pianificatore mantiene la modalità LC, variando la corsia preferenziale. Si noti dalla figura 4.53 (quarto grafico) che il veicolo *obstacle 3* ha una distanza di oltre 100m dal veicolo *ego*; dunque, la manovra di rientro in corsia non si identifica come un sorpasso da destra ma come un corretto posizionamento all'interno della carreggiata.
- $t = 13.2 s$: Il veicolo *ego* ha raggiunto la corsia all'estrema destra, dunque prosegue la marcia in modalità CC.
- $t = 22.5 s$: Il veicolo *ego* sfila alla destra del veicolo *obstacle 3* che procede con una velocità inferiore ed irregolarmente nella corsia centrale.

In conclusione è possibile affermare che, come atteso, i risultati risultano essere perfettamente coerenti con le norme vigenti riportate in precedenza.

4.6 Caso 5 - Modalità di cambio corsia rapido

Lo scenario "*Modalità di cambio corsia rapido*" è stato introdotto con l'obiettivo di permettere al sistema assistenziale di abilitare, in casi di emergenza, una strategia che genera traiettorie in grado di effettuare un cambio di corsia nel minor tempo possibile. Quest'introduzione permette al sistema di fornire una corretta risposta anche nel caso in cui venisse impiegato in scenari con elevate criticità.

Tuttavia, è bene precisare che l'obiettivo proposto e raggiunto, nell'ambito di questo studio, è quello di sviluppare tale strategia e non di includere all'interno del modello le relative logiche di attivazione. Dunque; rimane come futura implementazione la necessità di stabilire le tecniche di riconoscimento della criticità discussa, sulla base dei segnali di analisi dello scenario.

Il caso studio oggetto di questo paragrafo è stato l'unico sviluppato ed ottimizzato per intero con l'utilizzo del modello completo, ovvero inclusivo del modello di veicolo e del relativo controllore. Questa necessità nasce dall'introduzione di una nuova strategia di valutazione della traiettoria; in quanto, per poter effettuare il cambio di corsia in tempi ristretti è necessario comprendere il limite in termini di richiesta di performance. Conseguentemente, la verifica della traiettoria deve essere effettuata su parametri massivamente influenzati dalla dinamica del veicolo; le cui caratteristiche vengono trascurate nel modello comprensivo del solo pianificatore.

In questo caso studio si usa come manovra di test quella di un sorpasso semplice. La configurazione dello scenario è stata già introdotta con il caso studio "*Singolo cambio di corsia*". Le caratteristiche della strada sono riportate nella tabella 4.19.

Lunghezza strada	350 m
n° corsie	2

Tabella 4.19: Dati strada - scenario "*Modalità di cambio corsia rapido*".

Lo scenario si compone di due veicoli, *ego* e *lead*, i quali viaggiano inizialmente lungo la medesima corsia. Successivamente il veicolo *ego* si sposta nella corsia di sinistra, destinata al sorpasso. Si riportano nella seguente tabella le velocità iniziali e la distanza relativa tra i due veicoli all'istante $t = 0s$.

In figura 4.54 si ha una rappresentazione dello scenario nell'istante iniziale della simulazione.

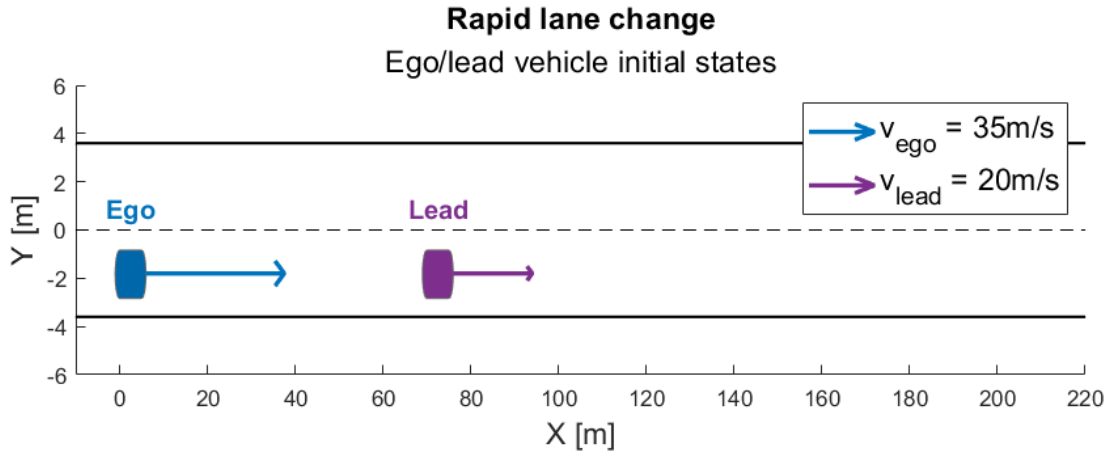


Figura 4.54: Stati iniziali $t = 0s$ - scenario "Modalità di cambio corsia rapido".

Nella seguente tabella 4.20 sono stati riportati i valori numerici della velocità iniziale di ciascun veicolo e della distanza relativa tra i due.

Distanza iniziale ego - lead	70 m
v_{ego}	35 m/s
v_{lead}	20 m/s

Tabella 4.20: Dati veicoli - scenario "Modalità di cambio corsia rapido".

4.6.1 Introduzione strategia di cambio corsia rapido

Per minimizzare il tempo impiegato ad effettuare la manovra di cambio corsia è necessario agire sul parametro che regola la durata della manovra, ovvero l'orizzonte temporale o "Time Horizon".

Si ricorda che l'insieme delle traiettorie alternative viene ordinato in base al parametro di costo associato. Da queste viene poi individuata come riferimento la traiettoria con il minor parametro di costo associato e che rispetta tutte le verifiche imposte. La funzione di costo è formata da diversi addendi, uno dei quali dipende dal tempo:

$$timeCost = timeHorizon * timeWeight \quad (4.10)$$

Il valore di default del coefficiente "timeWeight" è pari a -1, in modo da preferire le traiettorie con orizzonti temporali maggiori; a ciò corrispondono, nel caso della modalità LC, segmenti con accelerazioni e jerk laterali minori.

Per effettuare cambi di corsia rapidi, è stato scelto di inserire un *flag*, denominato "rapidLC", che modifica il segno del coefficiente "timeWeight". In tal modo si inverte l'ordine di preferenza in termini temporali.

<i>rapidLC</i>	<i>timeWeight</i>
0	-1
1	1

Tabella 4.21: Variazione del parametro "timeWeight".

La configurazione attuale del modello permette di modificare manualmente il valore del *flag* introdotto. Si rimanda a future implementazioni l'introduzione di strategie di analisi dei dati, provenienti dallo scenariol, che regolano l'attivazione della strategia introdotta.

4.6.2 Risultati simulazione - caso 5

La simulazione è stata condotta con un orizzonte temporale minimo pari a $Th_{min} = 1.3 s$. Analizzando i risultati emerge che la strategia implementata funziona correttamente, in quanto il sistema assistenziale ha pianificato la traiettoria di sorpasso con il minor orizzonte temporale possibile. Tuttavia, esaminando il confronto tra traiettoria di riferimento ed effettiva del veicolo *ego*, riportato in figura 4.55, ci si accorge che queste risultano essere sensibilmente differenti; la causa è correlata alle criticità esposte in seguito.

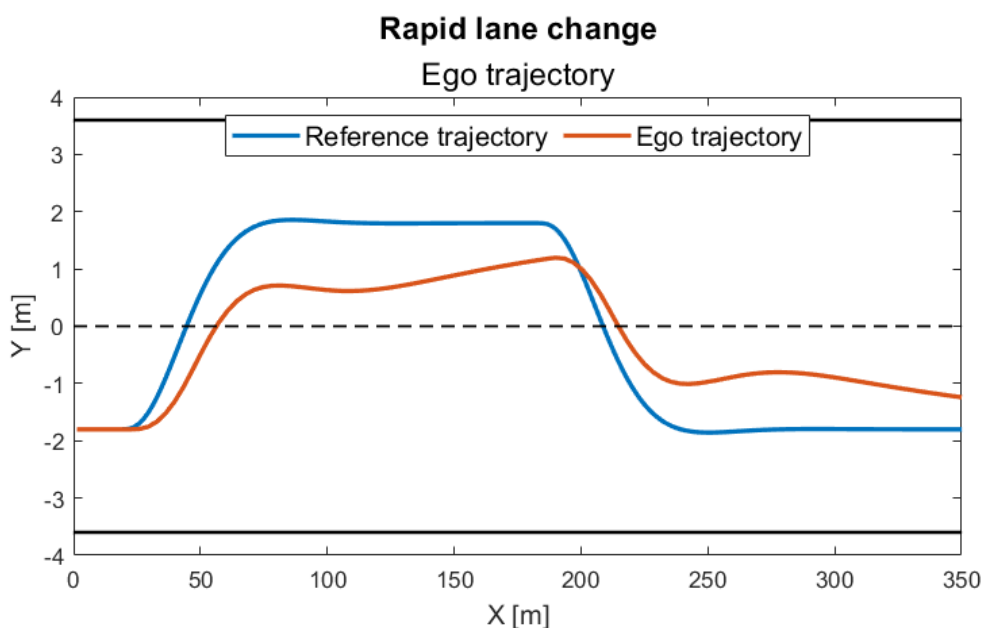


Figura 4.55: Traiettoria veicolo *ego* - scenario "Modalità di cambio corsia rapido".

Si specifica che l'obiettivo di questa disamina non è quello di valutare se il sistema assistenziale è in grado di governare il veicolo *ego* in maniera tale da non farlo collidere con il veicolo *lead*, bensì quello di valutare la traiettoria pianificata e la capacità del veicolo *ego* di seguirla. Conseguentemente, in fase di costruzione dello scenario, i due veicoli sono stati distanziati di una quantità tale da evitare la collisione; in modo da poter valutare l'evoluzione degli stati del veicolo *ego* per l'intero tratto stradale.

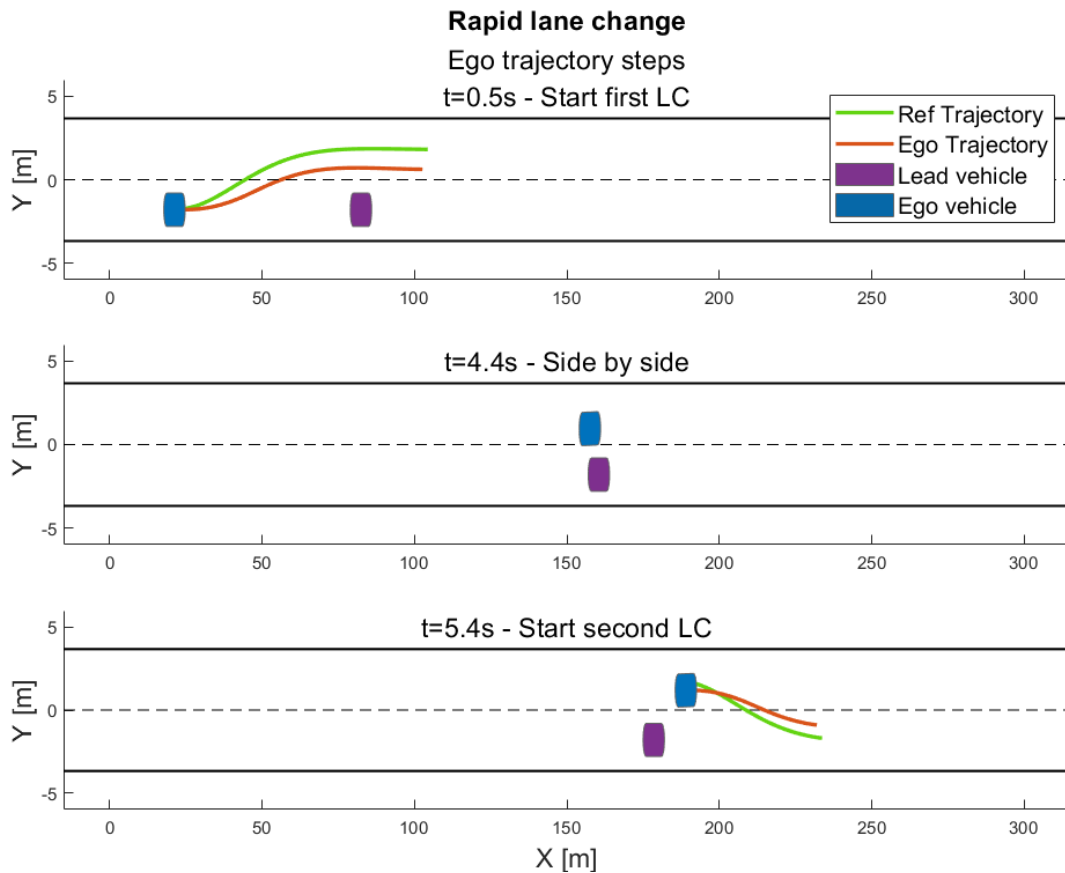


Figura 4.56: Frame traiettoria veicolo *ego* - scenario "Modalità di cambio corsia rapido".

Esaminando la configurazione dello scenario negli istanti di maggior rilievo, tramite i grafici riportati in figura 4.56, si può notare che nell'istante in cui i due veicoli si affiancano vi è sufficiente distanza laterale. Tuttavia, confrontando nell'istante di inizio cambio corsia il segmento di traiettoria pianificato con quello effettivamente percorso, emerge che lo scostamento è tale da non poter garantire la sicurezza del veicolo *ego* e dei suoi occupanti, nel caso di ostacoli maggiormente ravvicinati. Questo rischio è presente poiché la valutazione sulle collisioni viene effettuata sul percorso descritto dalla traiettoria di riferimento; dunque, maggiori sono gli errori calcolati dal controllore, maggiore è il rischio di trovarsi in una zona in cui l'assenza di collisioni non sia verificata. Si riportano gli errori della traiettoria in figura 4.57.

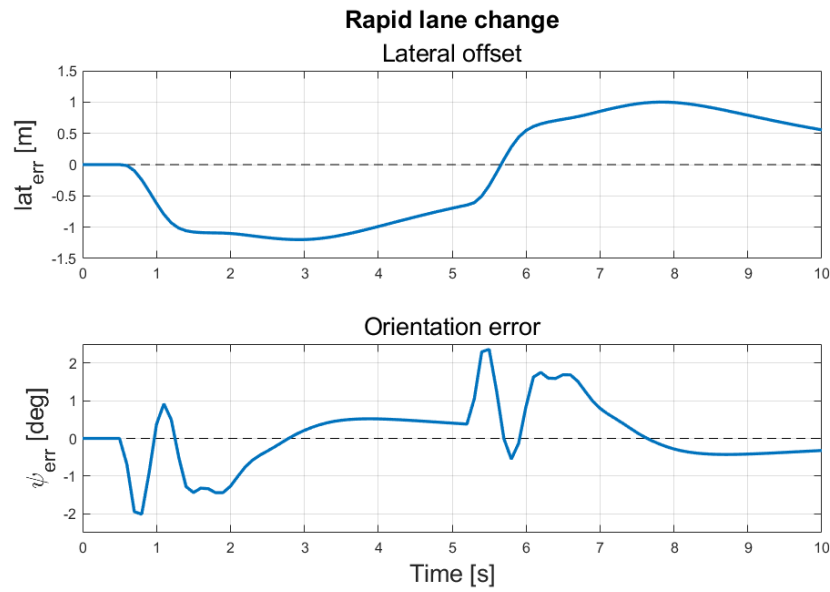


Figura 4.57: Errori traiettoria veicolo *ego* - scenario "Modalità di cambio corsia rapido".

Si può facilmente notare come l'entità degli errori è tale da non poter essere trascurata. Per comprenderne l'origine è necessario effettuare un'analisi incentrata sul confronto tra le grandezze associate alla traiettoria di riferimento e quelle associate alla traiettoria effettivamente percorsa dal veicolo *ego*.

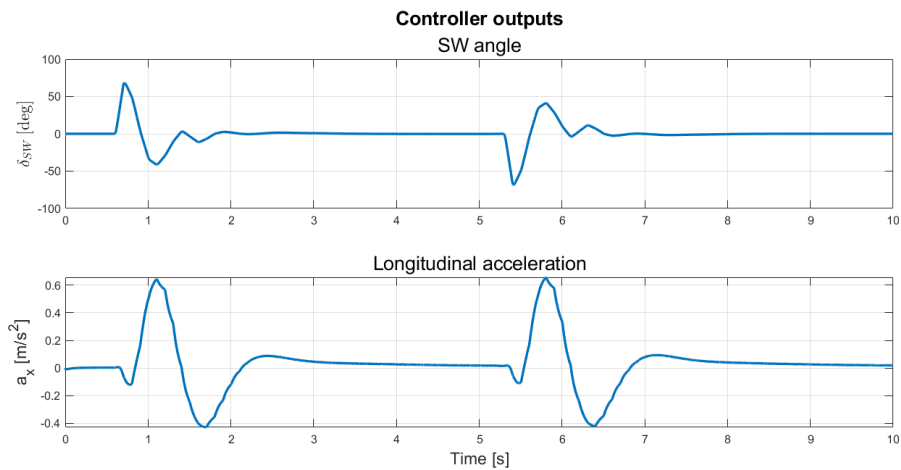


Figura 4.58: Segnali in output dal controllore *ego* - scenario "Modalità di cambio corsia rapido".

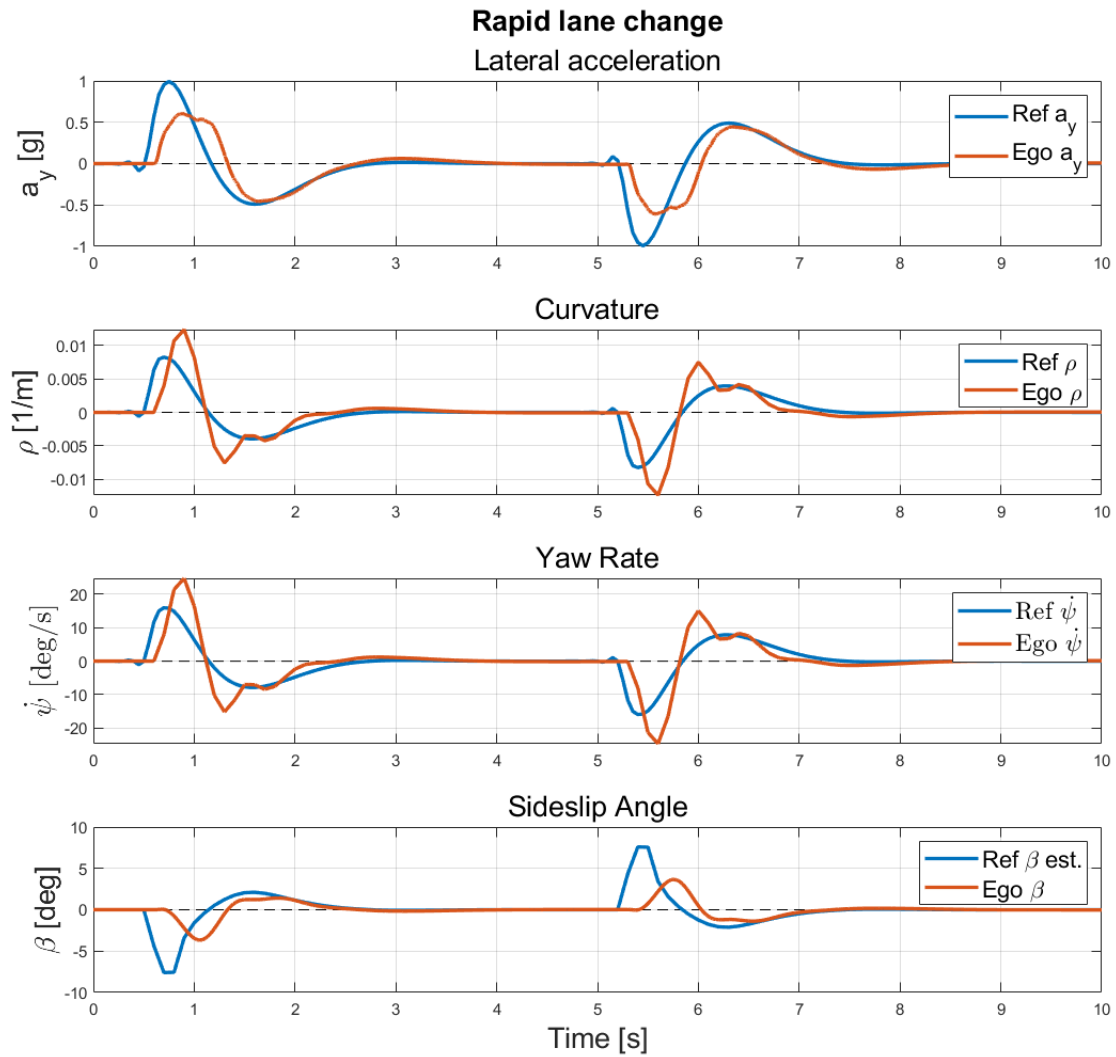


Figura 4.59: Confronto grandezze traiettoria di riferimento e traiettoria veicolo *ego* - scenario "Modalità di cambio corsia rapido".

Esaminando i grafici riportati in figura 4.59 emerge che è comune a tutte le grandezze riportate un ritardo di fase rispetto alla traiettoria di riferimento. Ciò è dovuto ad un ritardo di attuazione del controllore rispetto alla traiettoria di riferimento ricevuta in input.

La differenza di traiettoria può essere spiegata mediante l'analisi dei punti di lavoro della traiettoria di riferimento, riportati nel piano in figura 4.60.

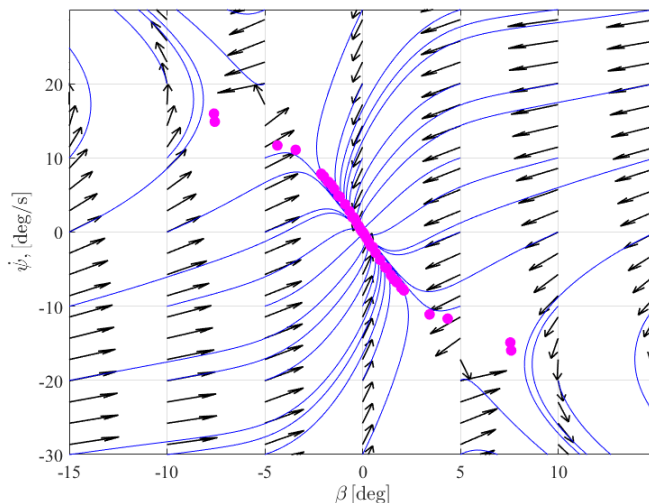


Figura 4.60: Punti di lavoro traiettoria di riferimento - scenario "Modalità di cambio corsia rapido".

All'interno del piano delle fasi $\dot{\psi} - \beta$ è possibile individuare ogni punto campionato della traiettoria di riferimento, essi sono stati rappresentati in fucsia in figura 4.60. I campi vettoriali tracciati dalle curve blu, indicano la convergenza/divergenza verso/dai punti di equilibrio stabile/instabile. Si nota che più punti fra quelli campionati vanno ad interessare aree della mappa dove si ha una divergenza degli stati del veicolo. In ambito pratico, ottenere i valori di $\dot{\psi}$ e β associati alle zone divergenti implicherebbe la perdita del controllo del veicolo.

Si ricorda che il controllore utilizzato è di tipo predittivo, ovvero contiene un modello di veicolo integrato in grado di prevedere gli stati futuri del veicolo *ego*, in risposta ai comandi di attuazione prodotti. Inoltre, il suo obiettivo principale è quello di minimizzare la funzione di costo associata agli errori di posizione ed orientamento. Conseguentemente, seguire una traiettoria che porta alla divergenza degli stati, come quella vista in questo esempio, porterebbe ad una divergenza degli errori. Per tale motivo, il controllore pianifica gli output in modo da avere una traiettoria differente da quella di riferimento, compensando l'instabilità della

traiettoria di riferimento ad anello aperto con un'azione di controllo che di fatto rende il modello ad anello chiuso.

Si riportano in figura 4.61 i punti di lavoro della traiettoria del veicolo *ego*. Si nota come essi siano tutti all'interno delle zone convergenti.

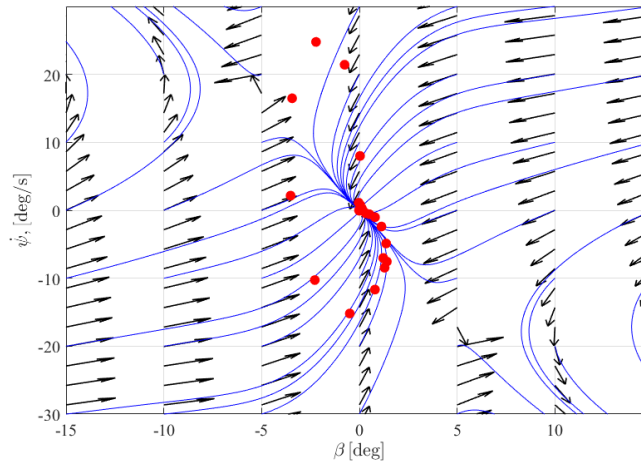


Figura 4.61: Punti di lavoro traiettoria veicolo *ego* - scenario "Modalità di cambio corsia rapido".

In conclusione, per effettuare dei cambi di corsia nel minor tempo possibile, riuscendo al contempo a seguire la traiettoria pianificata, è necessario implementare una strategia che sia in grado di eliminare dal set tutte quelle alternative che contengono punti di lavoro all'interno delle zone con stati divergenti. Risulta insufficiente la verifica attuale sulla fattibilità, la quale ha stabilito valida la traiettoria di riferimento esaminata, in base alle verifiche riportate nella tabella 4.22.

Parameter	Max	Limit	Unit
Yaw rate	15.97	20	<i>deg/s</i>
Long. Acceleration	0.07	1	<i>g</i>
Lat. Acceleration	0.99	1	<i>g</i>

Tabella 4.22: Valori di picco grandezze veicolo *ego* - scenario "Modalità di cambio corsia rapido".

4.6.3 Implementazione verifica di stabilità

Con il fine di effettuare una valutazione più precisa del set di traiettorie alternative, è stata implementata una verifica necessaria a scartare le traiettorie le cui variabili correlate alla dinamica laterale del veicolo *ego* lo porterebbero in condizioni di instabilità dinamica.

Il parametro sul quale viene effettuata la verifica è l'angolo di assetto β , la cui formulazione è la seguente:

$$\beta = \text{atan} \frac{v}{u} \quad (4.11)$$

Dove v ed u sono rispettivamente la velocità longitudinale e laterale del veicolo, calcolate nel baricentro e rispetto ad un sistema di riferimento non inerziale.

Calcolo dell'angolo di assetto β delle traiettorie alternative

Per eliminare tutte le traiettorie alle quali è associato un angolo di assetto al di fuori dei limiti massimo e minimo del parametro, in modo da evitare instabilità dinamica, è necessario calcolare in previsione l'angolo di assetto β associato a ciascuna traiettoria appartenente al set di alternative.

Tuttavia, non è possibile implementare il calcolo di β effettuato in previsione mediante la formulazione 4.11. Ciò è correlato alla metodologia con la quale vengono costruite le traiettorie alternative, ipotizzando un veicolo puntiforme, dunque, privo di tutte quelle caratteristiche che influenzano in modo massivo la dinamica laterale del veicolo, quindi la velocità laterale v .

Per far sì che il valore di β stimato sia il più vicino possibile a quello reale in tutti i campi di funzionamento è stato scelto un approccio di tipo non lineare. Attraverso il quale è stata implementata una delle caratteristiche stazionarie ricavate nel capitolo precedente all'interno del modello. Per testare la metodologia di stima dell'angolo di assetto sono stati valutati i risultati delle manovre di: *ramp steer*, *step steer*, *sweep steer* ed una manovra di sorpasso semplice.

Soluzione adottata: Per effettuare la stima del valore di angolo di assetto previsto è stato scelto un approccio di tipo non lineare. Nello specifico è stata ricavata ed implementata la caratteristica del veicolo $\alpha_R(F_{yR})$, riportata in figura 4.62.

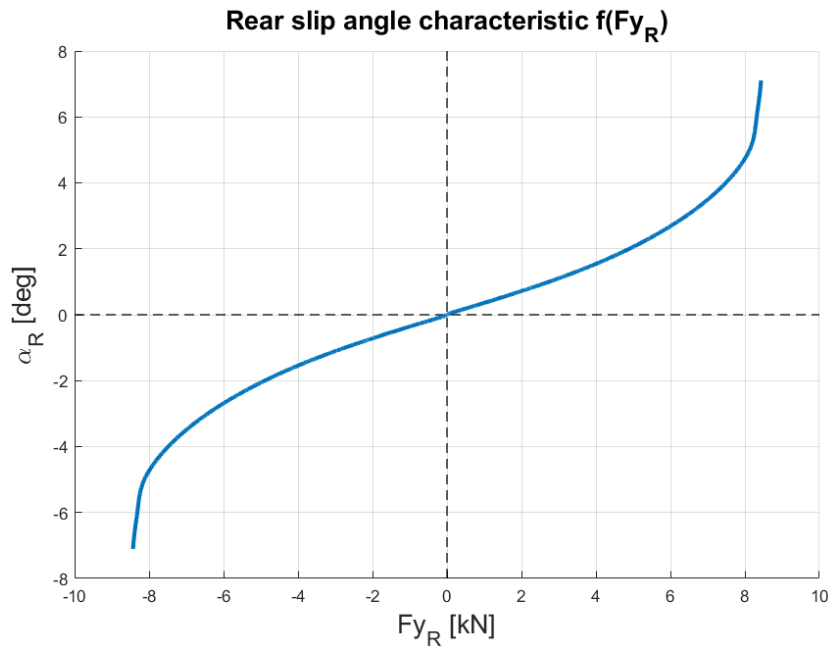


Figura 4.62: Caratteristica angolo di deriva posteriore.

Il valore puntuale della forza laterale è stato stimato mediante la seguente formula, la quale trascura la componente inerziale per ipotesi di stazionarietà.

$$F_{yR} = m_R a_y = \left(\frac{m a}{L} \right) a_y \quad (4.12)$$

Dunque, per ogni punto di ciascuna traiettoria viene calcolato il valore della forza laterale al posteriore e mediante la caratteristica implementata si ricava il valore dell'angolo di deriva medio degli pneumatici posteriori α_R . Quindi si ricava il valore dell'angolo di assetto con la seguente equazione:

$$\beta = \beta_0 + \delta_R - \alpha_R(m_R a_y) = b\rho + \delta_R - \alpha_R(m_R a_y) \quad (4.13)$$

Dove β_0 identifica l'angolo di assetto cinematico e δ_R l'angolo di sterzo al posteriore, il quale è imposto sempre nullo. Oltre all'ipotesi di stazionarietà, risulta essere importante specificare che per la stima di β vengono considerati angoli di convergenza e campanatura nulli.

Si riportano di seguito le stime di F_{yR} e β per ciascuna traiettoria testata. Si specifica che il confronto viene effettuato fra valore stimato a partire dai dati forniti in output dal modello di veicolo e valore effettivo. Dunque, non si hanno alterazioni dei segnali dovute al controllore avendo un sistema ad anello aperto. Le manovre testate sono le seguenti:

- *Ramp steer*: Manovra caratterizzata da un incremento nel tempo dell'angolo volante costante e pari a $\dot{\delta}_{vol} = 15 \text{ deg/s}$. La manovra viene eseguita a velocità costante e pari a $Vel = 50 \text{ km/h}$.
- *Step steer*: Manovra caratterizzata dalla variazione istantanea dell'angolo di sterzo che passa dal valore $\delta_{vol} = 0 \text{ deg}$ a $\delta_{vol} = 10 \text{ deg}$ nell'istante $t = 2.0 \text{ s}$. La manovra viene eseguita a velocità costante e pari a $Vel = 100 \text{ km/h}$.
- *Sweep steer*: Manovra caratterizzata da un input sinusoidale dell'angolo di sterzo con ampiezza fissa $\delta_{vol} \pm 20 \text{ deg}$ e da una frequenza che aumenta nel tempo. La manovra viene eseguita a velocità costante e pari a $Vel = 80 \text{ km/h}$.
- *Sorpasso semplice*: Manovra di sorpasso completo di un veicolo che precede il veicolo *lead*, con orizzonte temporale pari a $Th = 1.5 \text{ s}$ e velocità di crociera pari a $Vel = 126 \text{ km/h}$.

Tutte le manovre sono state eseguite con il medesimo veicolo, il quale ha un comportamento tendenzialmente sovrasterzante e le cui caratteristiche sono state dettagliate all'interno del capitolo terzo. Le condizioni dell'asfalto sono state mantenute uguali in tutte le prove, assunto un coefficiente di attrito pari a $\mu = 1$.

Risultati test *Ramp steer*

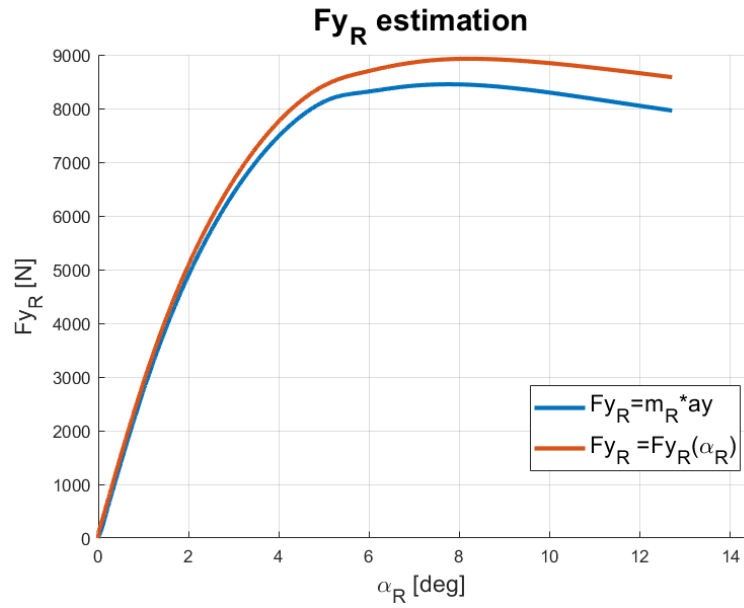


Figura 4.63: Stima F_{y_R} - manovra *Ramp steer*.

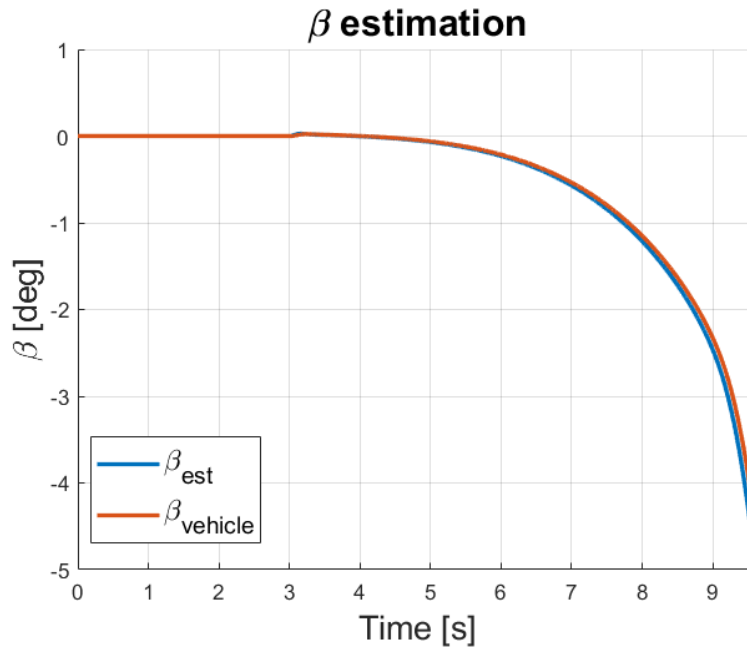
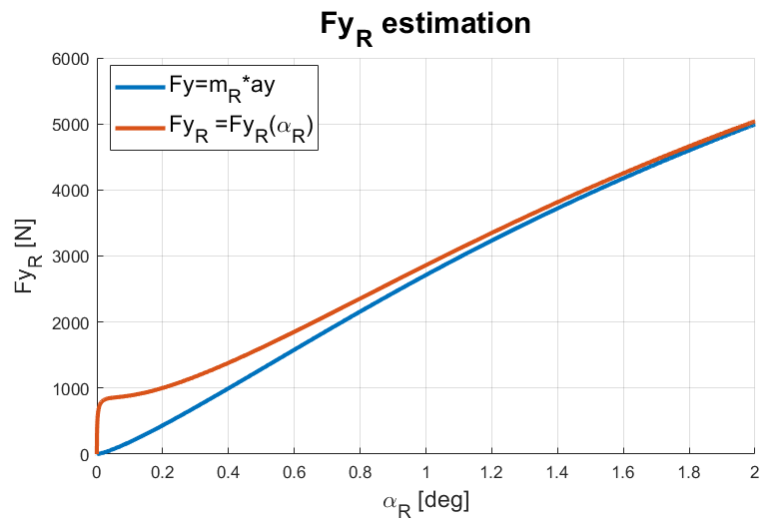
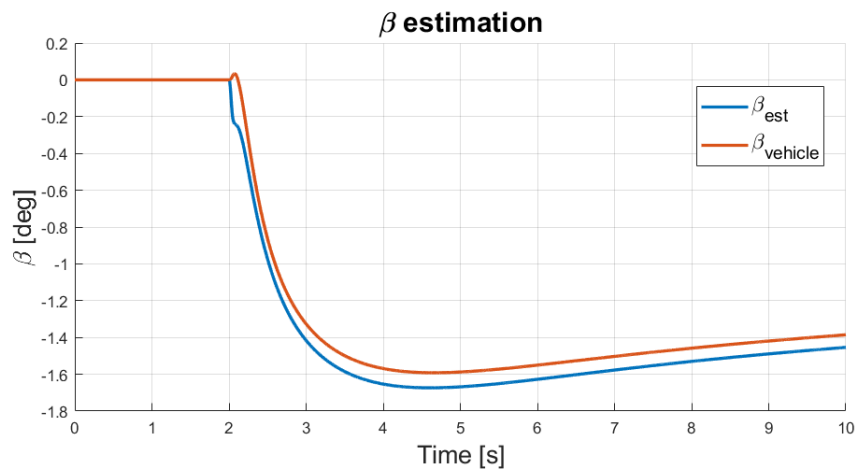


Figura 4.64: Stima β - manovra *Ramp steer*.

Nella figure 4.67 è stato riportato il confronto tra la stima di Fy_R e il valore effettivo in output dal modello di veicolo. Si nota come l'andamento della stima risulta essere coerente con il relativo valore effettivo; tuttavia, è presente uno scostamento tra i valori registrati. Esso può essere imputato alla convergenza statica, la cui influenza viene trascurata nel caso della stima della forza laterale. La convergenza positiva delle ruote aumenta la forza laterale sviluppata dall'assale Fy_R a parità angolo di deriva medio dell'assale α_R .

Nella figure 4.68 è stato riportato il confronto tra la stima dell'angolo di assetto ed il relativo valore effettivo. Si nota come anche in questo caso gli andamenti delle stime risultano essere coerenti con il relativo valore effettivo. Dunque, la metodologia implementata è ritenuta valida. Lo scostamento ed il ritardo di fase presente tra i due segnali possono essere attribuiti a due fattori:

- La coppia di imbardata $I_Z \ddot{\psi}$: la quale rappresenta il termine transitorio che viene trascurato per l'ipotesi di stazionarietà introdotta.
- La dinamica di rollio: la quale viene trascurata nella stima di β , ma introduce un ritardo tra accelerazione laterale e trasferimenti di carico.

Risultati test *Step steer*Figura 4.65: Stima F_{yR} - manovra *Step steer*.Figura 4.66: Stima β - manovra *Step steer*.

Anche in questo caso gli andamenti delle grandezze stimate risultano essere perfettamente coerenti con quelli delle grandezze effettive. Si nota uno scostamento maggiore dei risultati rispetto al test precedente, in quanto la manovra *step steer* presenta forti fenomeni transitori; tuttavia, i risultati possono essere ritenuti accettabili.

Risultati test *Sweep steer*

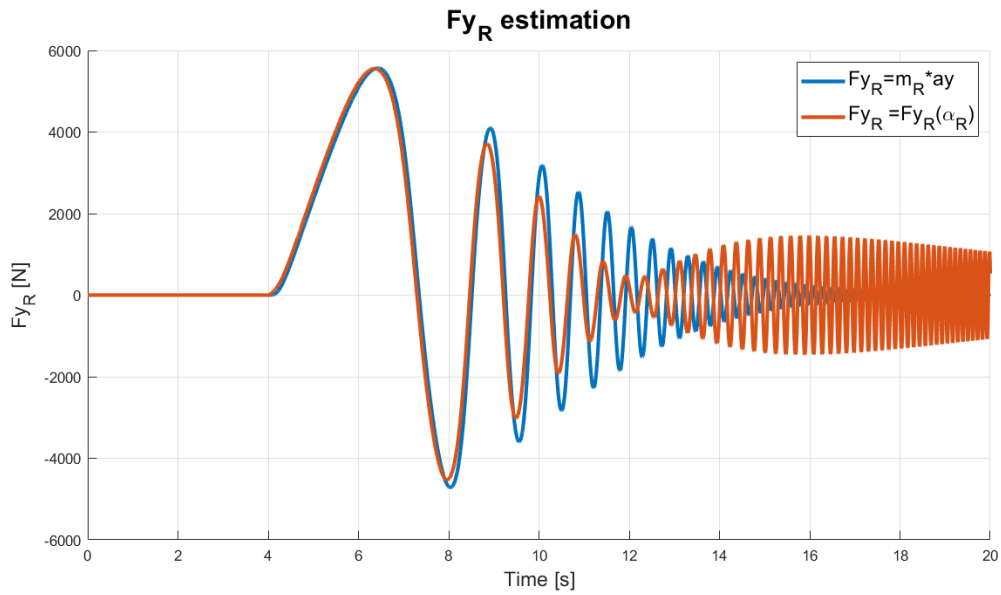


Figura 4.67: Stima Fy_R - manovra *Sweep steer*.

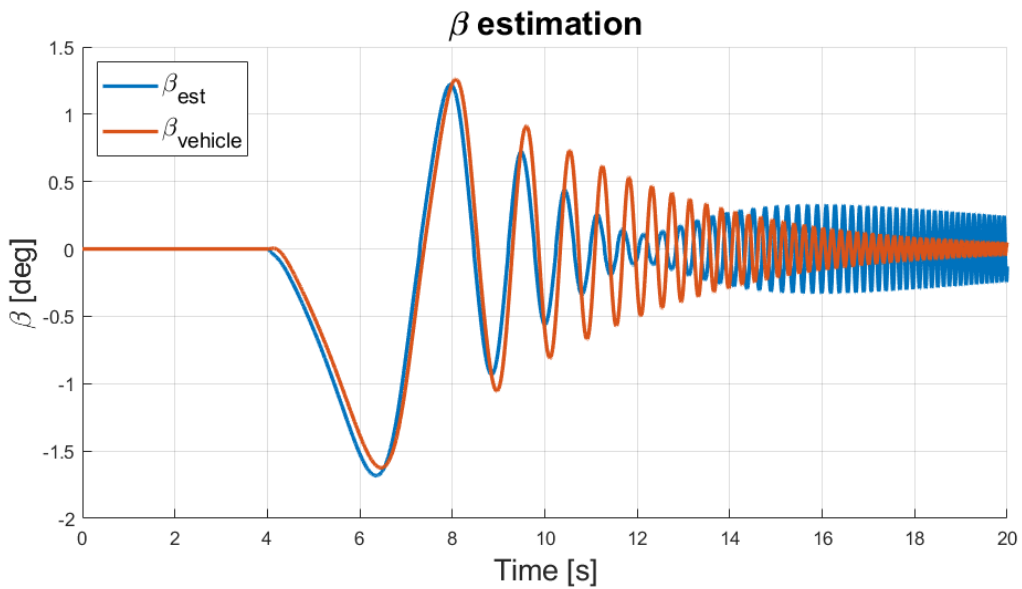


Figura 4.68: Stima β - manovra *Sweep steer*.

Esaminando i grafici riportati nelle figure 4.67 e 4.68 si può ottenere un riscontro del comportamento dell'algoritmo di stima dell'angolo di assetto al variare della frequenza con la quale il sistema viene sollecitato.

Per poter proporre un'analisi maggiormente dettagliata è stato ricavato un confronto tra la funzione di trasferimento che in uscita fornisce il segnale di angolo di assetto stimato β_{est} e quella che ha in uscita il segnale di angolo di assetto effettivo del veicolo β a parità di ingresso, ovvero l'angolo volante δ_{SW} .

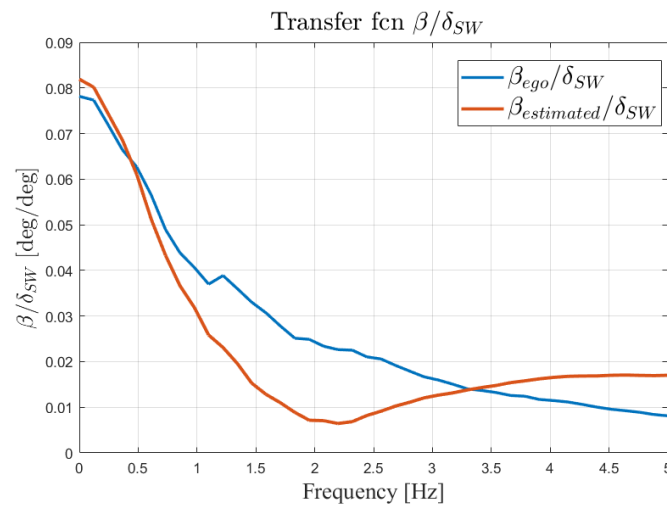
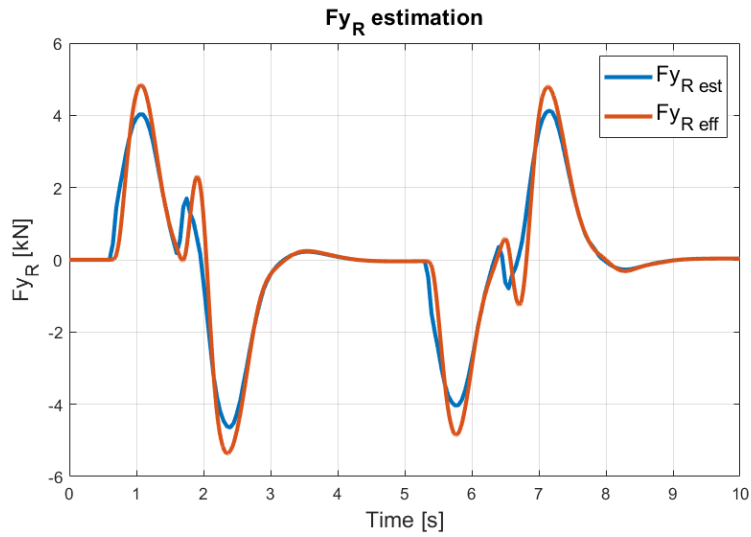
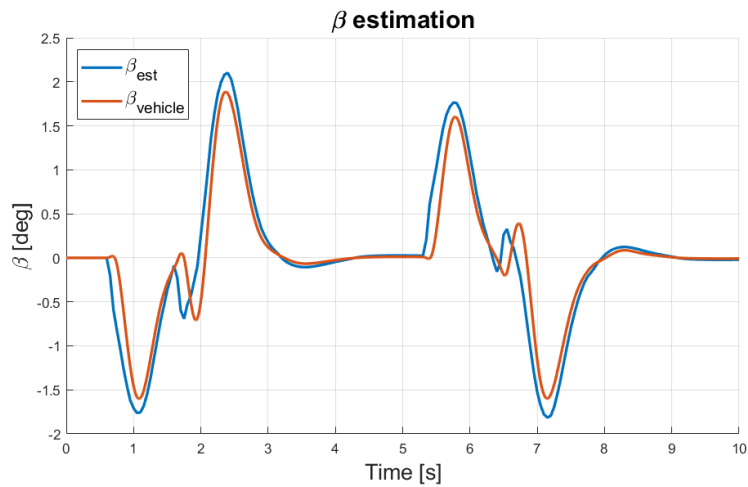


Figura 4.69: Confronto funzioni di trasferimento angolo di assetto/angolo volante - manovra *Sweep steer*.

Si nota come alle basse ed alle alte frequenze, facendo riferimento ad un range che va da 0 Hz a 5 Hz , il guadagno della funzione di trasferimento ottenuta con il valore di angolo di assetto stimato sia superiore rispetto al caso reale, viceversa alle frequenze intermedie.

Risultati manovra *Sorpasso semplice*Figura 4.70: Stima Fy_R - manovra *Sorpasso semplice*.Figura 4.71: Stima β - manovra *Sorpasso semplice*.

Nel caso della manovra di sorpasso le considerazioni effettuate sullo scostamento dei valori di β stimato ed effettivo sono analoghe a quelle effettuate per i casi precedenti. La stima presenta un errore contenuto, dunque, la metodologia è considerevolmente validata.

Individuazione dei limiti dell'angolo di assetto β

Il valore stimato dell'angolo di assetto β viene utilizzato per verificare, all'interno del piano delle fasi $\dot{\psi} - \beta$, se il punto di funzionamento è in una zona caratterizzata dalla convergenza degli stati verso un punto di equilibrio stabile o dalla loro divergenza. Le mappe utilizzate per la verifica sono state ricavate mediante una metodologia illustrata all'interno del capitolo secondo; tuttavia, ciò che è importante sottolineare in questa fase dello studio è che la posizione dei punti di equilibrio instabile varia al variare della velocità, come si può vedere in figura 4.72.

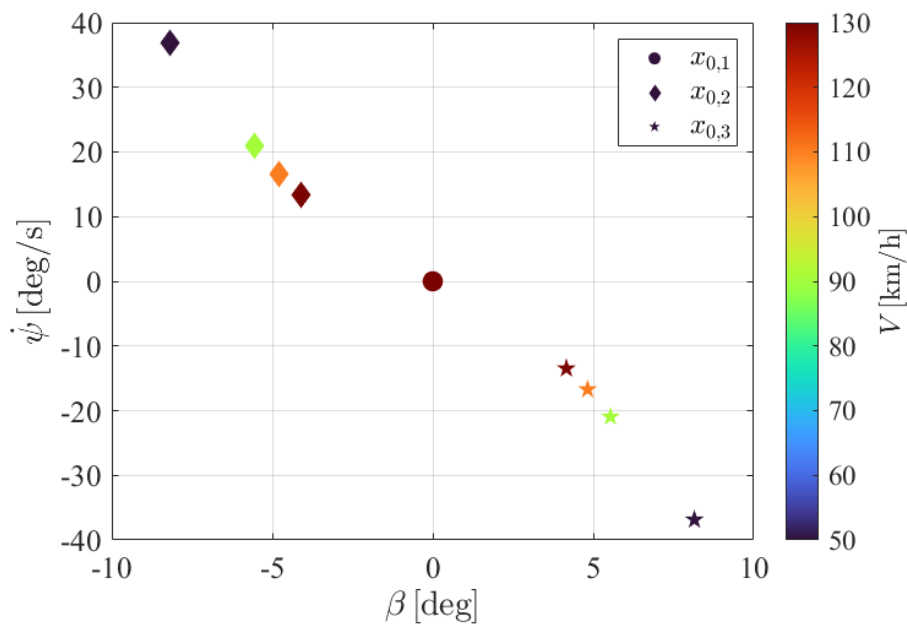


Figura 4.72: Posizione punti di equilibrio instabile al variare della velocità del veicolo.

Per introdurre i limiti da utilizzare nel corso della verifica di fattibilità sono state ricavate un numero n di mappe per un numero n di valori differenti di velocità del veicolo. Da ciascuna di queste mappe sono state ricavate le coordinate dei punti di equilibrio instabile, i cui valori di ascisse β ed ordinate $\dot{\psi}$ sono stati utilizzati per ricavare le caratteristiche riportate nelle figure 4.73 e 4.74. Dove $x_{0,1}$ è il punto di equilibrio stabile e $x_{0,2}$ e $x_{0,3}$ sono i punti di equilibrio instabile. Le mappe vengono ricavate introducendo un'ipotesi forte, quale quella di angolo di sterzo nullo.

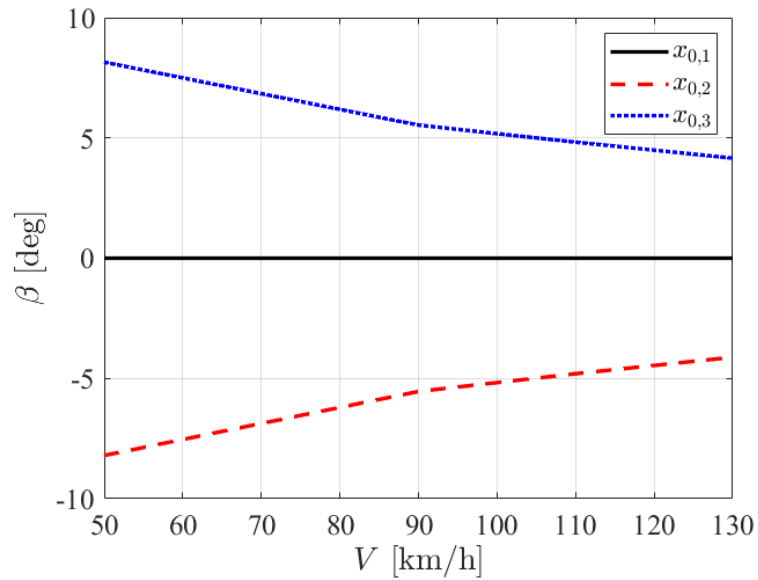


Figura 4.73: Ordinata β dei punti di equilibrio instabile al variare della velocità.

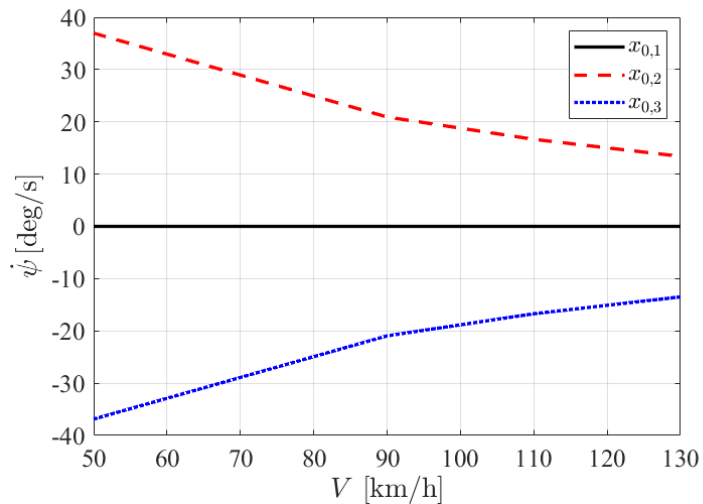


Figura 4.74: Ascissa ψ dei punti di equilibrio instabile al variare della velocità.

La verifica di stabilità viene effettuata su ciascun punto di ciascuna traiettoria alternativa. Essendo le coordinate dei punti di equilibrio instabile dipendenti dalla velocità, esse devono necessariamente variare in base al valore della velocità del punto della traiettoria da valutare. Vengono così ricavate mediante interpolazione lineare le coordinate dei punti $x_{0,1}$, $x_{0,2}$ e $x_{0,3}$.

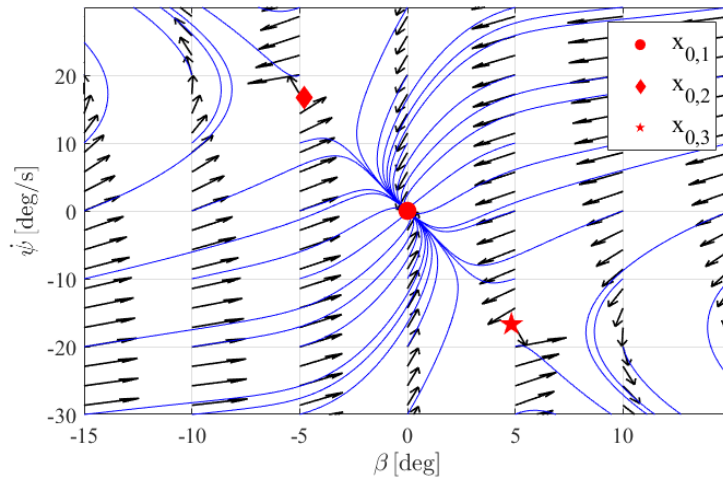


Figura 4.75: Esempio piano delle fasi.

Prendendo in esempio una mappa come quella riportata in figura 4.74 e gli annessi punti di equilibrio stabile $x_{0,1}$ ed instabile $x_{0,2}$ e $x_{0,3}$, si possono identificare dai campi vettoriali le zone stabili ed instabili.

- *Zona stabile:* tutta la porzione di mappa in cui il campo vettoriale è tale da far convergere gli stati verso il punto di equilibrio stabile $x_{0,1}$. Nel grafico in figura 4.75 la zona stabile si compone della porzione centrale della mappa e delle porzioni in basso a sinistra ed in alto a destra.
- *Zona instabile:* tutta la porzione di mappa in cui il campo vettoriale è tale da far divergere gli stati. Nel grafico in figura 4.75 la zona stabile si compone delle porzioni in alto a sinistra ed in basso a destra.

In letteratura vi sono differenti approcci per definire i limiti di stabilità/instabilità; quello scelto nell'ambito di questo studio è il più conservativo. Vengono considerati stabili tutti quei punti che risiedono all'interno del rettangolo che ha i punti di equilibrio instabile $x_{0,2}$ e $x_{0,3}$ agli estremi di una delle sue diagonali.

Con tale soluzione vengono escluse ampie porzioni della zona di stabilità; tuttavia, l'approccio conservativo è stato ricercato poiché sebbene alcuni punti di funzionamento facciano parte della zona stabile, quando la distanza da $x_{0,1}$ è elevata i transitori di convergenza sono eccessivamente lunghi.

Nella seguente figura 4.76 è stato riprodotto un esempio grafico della zona di stabilità adottata in questo studio.

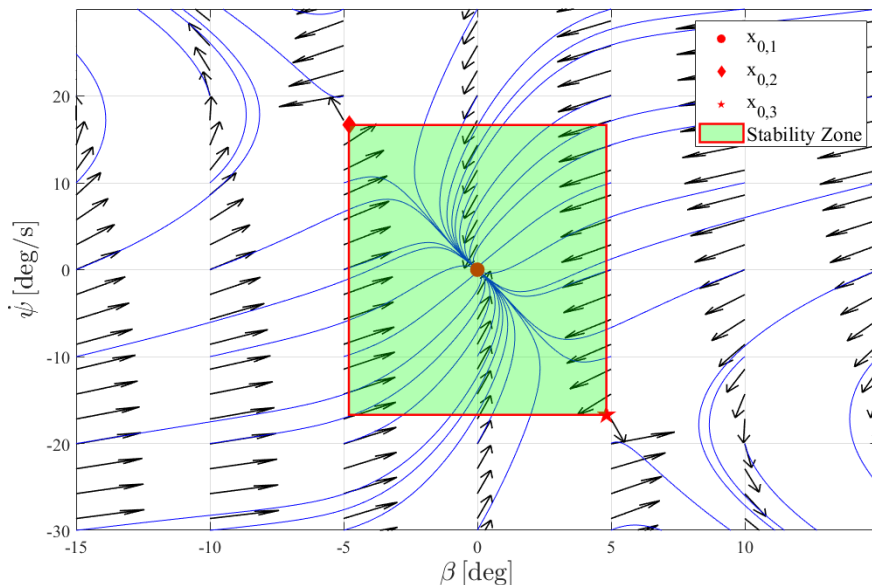


Figura 4.76: Esempio piano delle fasi con zona di stabilità.

Implementazione della strategia adottata

Quanto appena esposto è stato successivamente implementato all'interno della funzione *helperKinematicFeasibility*, la cui versione iniziale è stata analizzata in appendice A. La traiettoria viene ritenuta non valida se il *flag* associato a tale verifica, denominato "instabilityViolated", assume il valore logico *true* (1); ciò accade se almeno uno degli n punti della traiettoria esaminata non rispetta la seguente verifica.

$$(\beta_{min}(i) < \beta(i) < \beta_{max}(i)) \wedge (\dot{\psi}_{min}(i) < \dot{\psi}(i) < \dot{\psi}_{max}(i)) \quad (4.14)$$

Si riporta in figura 4.77 un diagramma di flusso che schematizza l'algoritmo di verifica delle condizioni di stabilità.

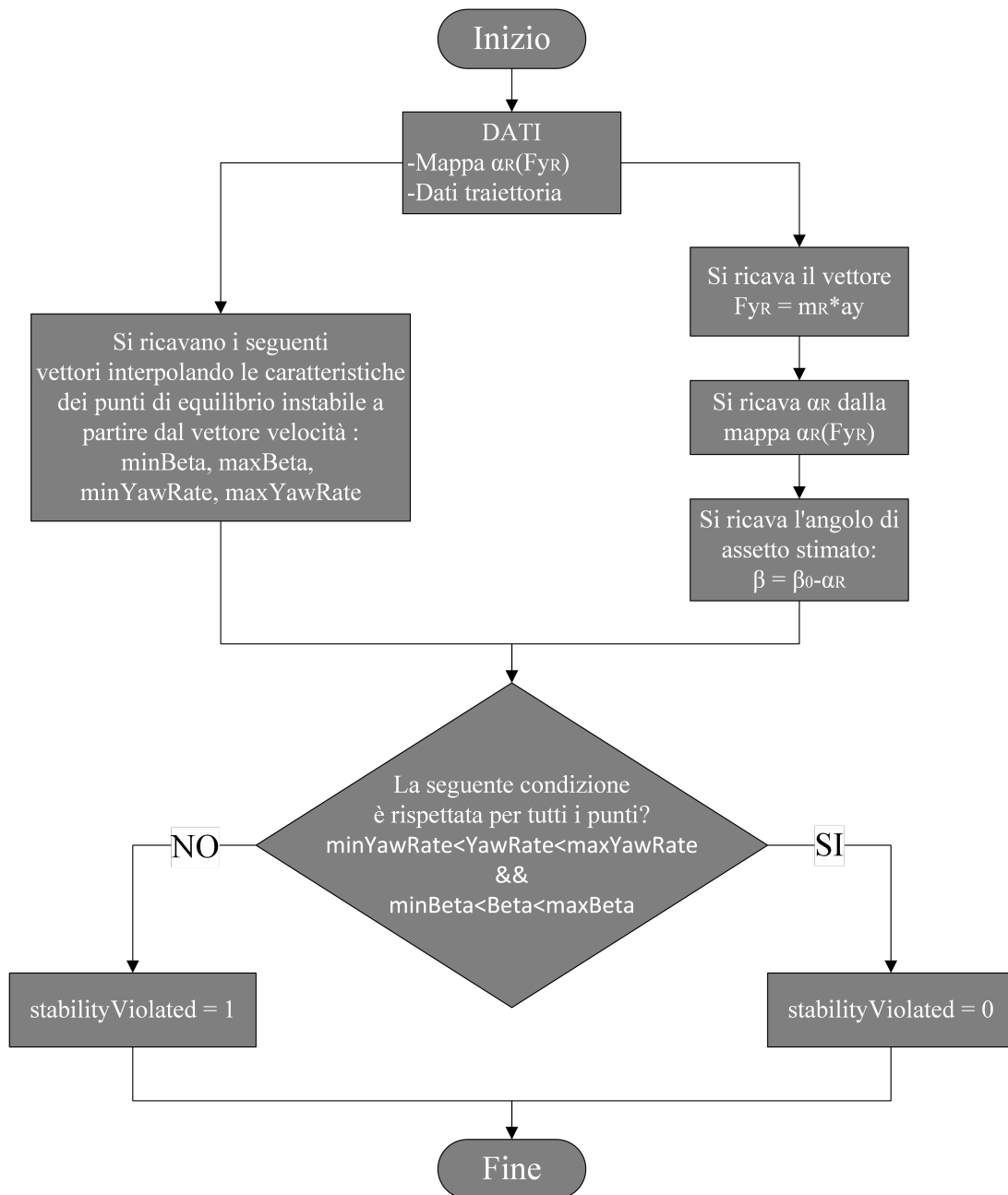


Figura 4.77: Diagramma di flusso della verifica della condizione di stabilità.

Stability Index: In studi di stabilità, analoghi a quello introdotto in questo studio, in letteratura viene usualmente individuato un indice che quantifica la vicinanza del punto esaminato P dal limite della zona di stabilità.

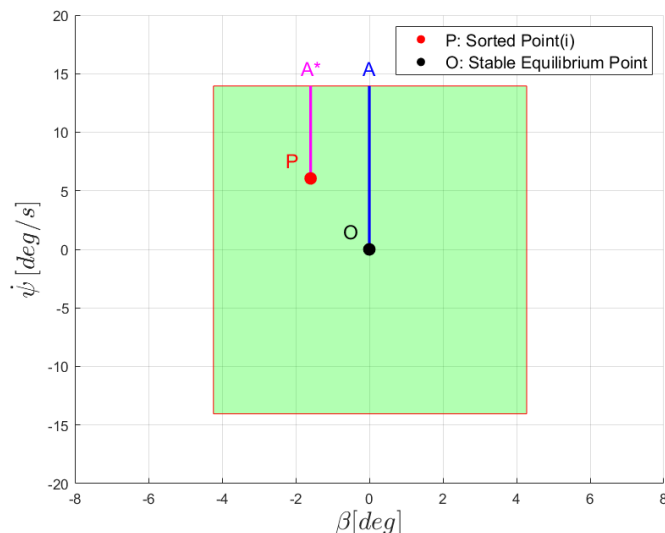


Figura 4.78: Lunghezze utili al calcolo dell'indice di stabilità relativo a $\dot{\psi}$.

Avendo utilizzato un approccio più conservativo, dunque, una definizione differente dei limiti rispetto agli esempi in letteratura, è necessario definire nuovamente l'indice di stabilità. Facendo riferimento alle lunghezze evidenziate nell'esempio in figura 4.78, esso viene calcolato per un generico punto P mediante la seguente equazione. Il calcolo deve essere ripetuto in direzione ortogonale a quella rappresentata, per valutare l'indice anche per β ; viene poi preso il valore corrispondente al caso maggiormente critico.

$$SI = 1 - \frac{|P A^*|}{|O A|} \quad (4.15)$$

Dove:

- P : Generico punto per cui calcolare l'indice SI.
- O : Punto di equilibrio stabile.
- A, A^* : Punti del lato del rettangolo limite più vicino al punto P.

4.6.4 Risultati simulazione caso 5 con risoluzione criticità

Dopo aver approfonditamente esposto gli aggiornamenti apportati al modello con l'introduzione della verifica di stabilità, è stata eseguita una nuova simulazione dello scenario presentato in questo caso studio. Ciò è stato utile per poter verificare il corretto funzionamento delle nuove funzionalità introdotte nel modello. Si riporta in figura 4.79, il confronto tra la traiettoria di riferimento e quella effettivamente seguita dal veicolo *ego*.

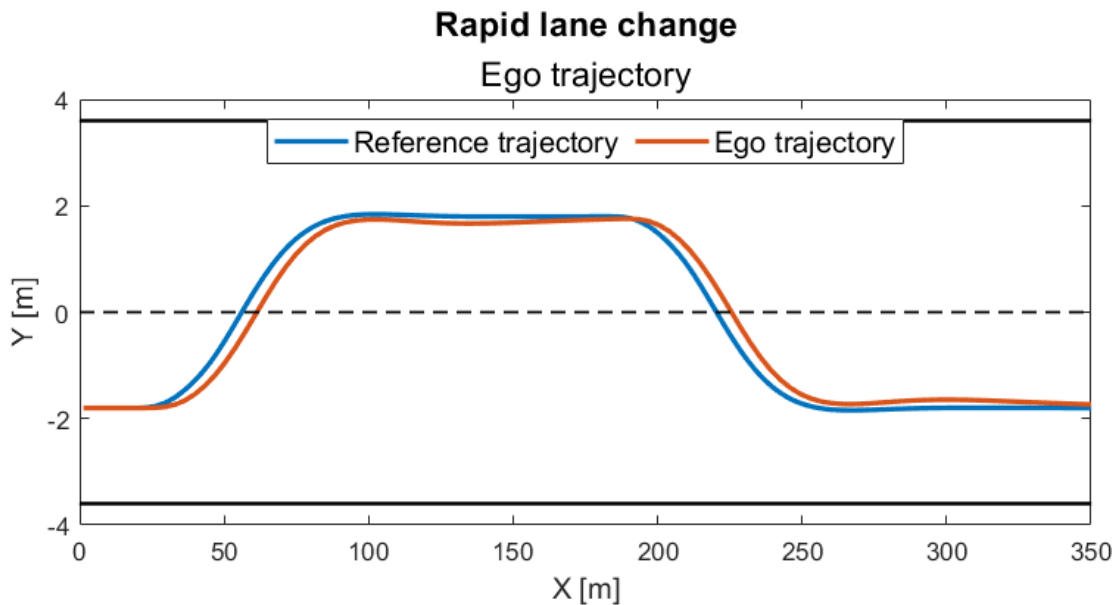


Figura 4.79: Traiettoria veicolo *ego* - scenario "Modalità di cambio corsia rapido".

Risulta visibilmente evidente il miglioramento, rispetto ai risultati ottenuti in assenza della verifica di stabilità, della capacità del veicolo di seguire la traiettoria fornita in output dal pianificatore. Esso è dovuto all'eliminazione dal set di traiettorie alternative, per ciascuna delle due manovre di cambio corsia, della traiettoria con orizzonte temporale $Th = 1.3s$, a causa dei valori di angolo di assetto stimato β eccessivamente elevati. La traiettoria che viene individuata come riferimento in questo caso ha un $Th = 2.0s$ e dei valori di angolo di assetto stimato β i quali rispettano la verifica di stabilità implementata.

Si riporta nella seguente figura 4.80 la rappresentazione in capsule 2D dello scenario negli istanti di maggior rilievo della simulazione.

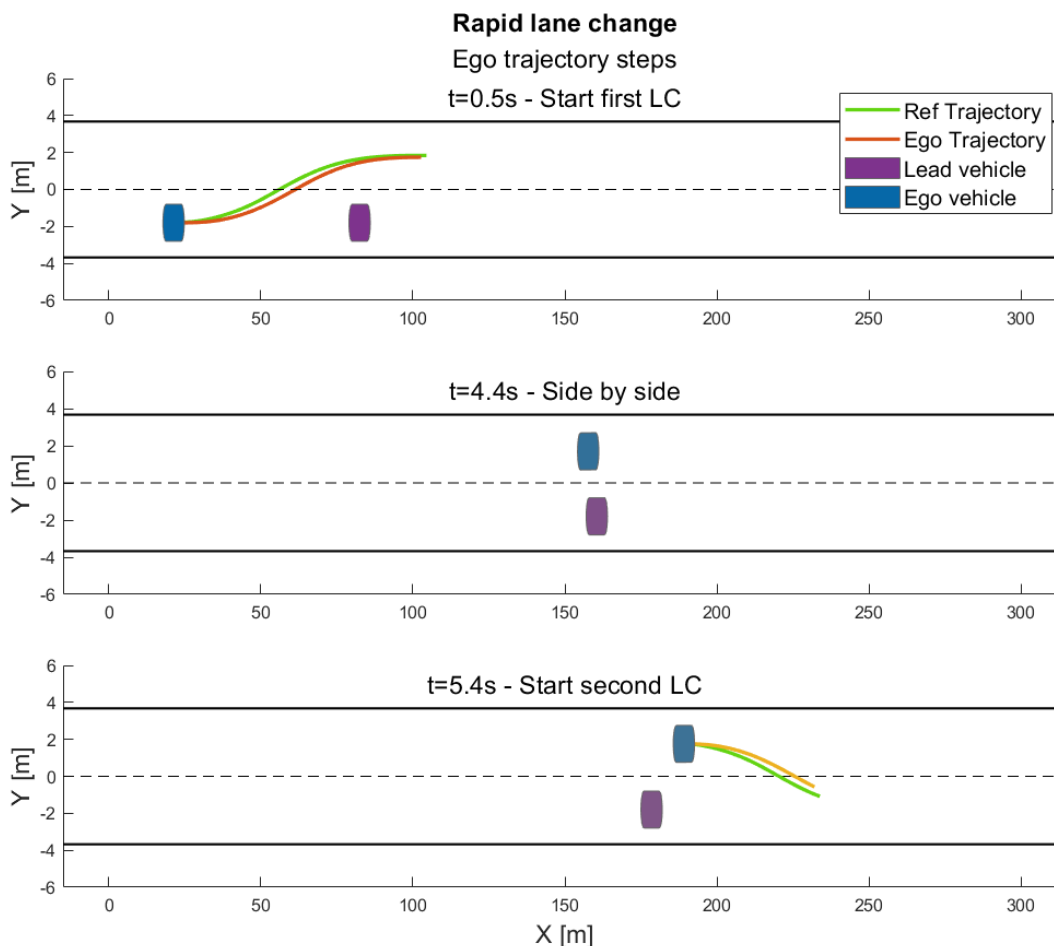


Figura 4.80: Frame traiettoria veicolo *ego* - scenario "Modalità di cambio corsia rapido".

Per entrambe le manovre di cambio corsia si nota una traiettoria effettivamente seguita, rappresentata in arancione, perfettamente coerente con quella di riferimento, rappresentata in verde. Inoltre, nell'istante in cui i due veicoli si affiancano, tra essi è presente una maggiore distanza.

Di conseguenza si può affermare che riuscire a seguire con maggiore accuratezza la traiettoria di riferimento porta a cospicui benefici in termini di sicurezza.

Anche se la traiettoria di riferimento pianificata in questo caso richiede un tempo pari a $Th = 2.0s$ per completare la manovra di cambio corsia, dunque maggiore rispetto al caso precedente, il veicolo *ego* riesce a seguirla con maggiore fedeltà, in quanto esso raggiunge il centro della corsia di sorpasso in tempi drasticamente ridotti.

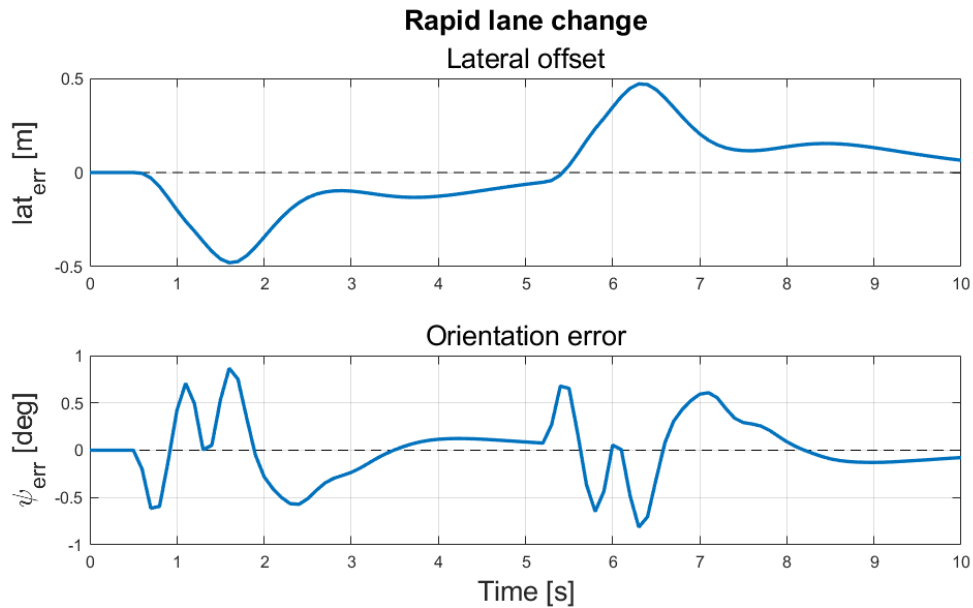


Figura 4.81: Errori traiettoria veicolo *ego* - scenario "Modalità di cambio corsia rapido".

In figura 4.81 sono stati riportati gli andamenti degli errori di posizionamento ed orientazione. Nella tabella sono stati riportati i risultati numerici del confronto tra i valori di picco degli errori associati alla traiettoria del veicolo *ego* nei due casi, prima e dopo l'introduzione della verifica di stabilità.

Parameter	No Stab. Check	With Stab. Check	%Variation	Unit
Max Lateral error	1.20	0.48	-60.0%	<i>m</i>
Max Yaw error	2.36	0.87	-63.1%	<i>deg</i>

Tabella 4.23: Confronto errori di traiettoria prima e dopo l'inserimento della verifica di stabilità - scenario "Modalità di cambio corsia rapido".

Quanto appena illustrato è imputabile ad una differente alterazione dei segnali da parte del controllore. In questo caso esso non deve più compensare l'errata pianificazione della traiettoria di riferimento, la quale nel caso precedente portava ad una divergenza degli stati. Si esamina il confronto tra le grandezze associate alla traiettoria di riferimento e quelle associate alla traiettoria effettivamente percorsa, il quale è stato riportato in figura 4.82.

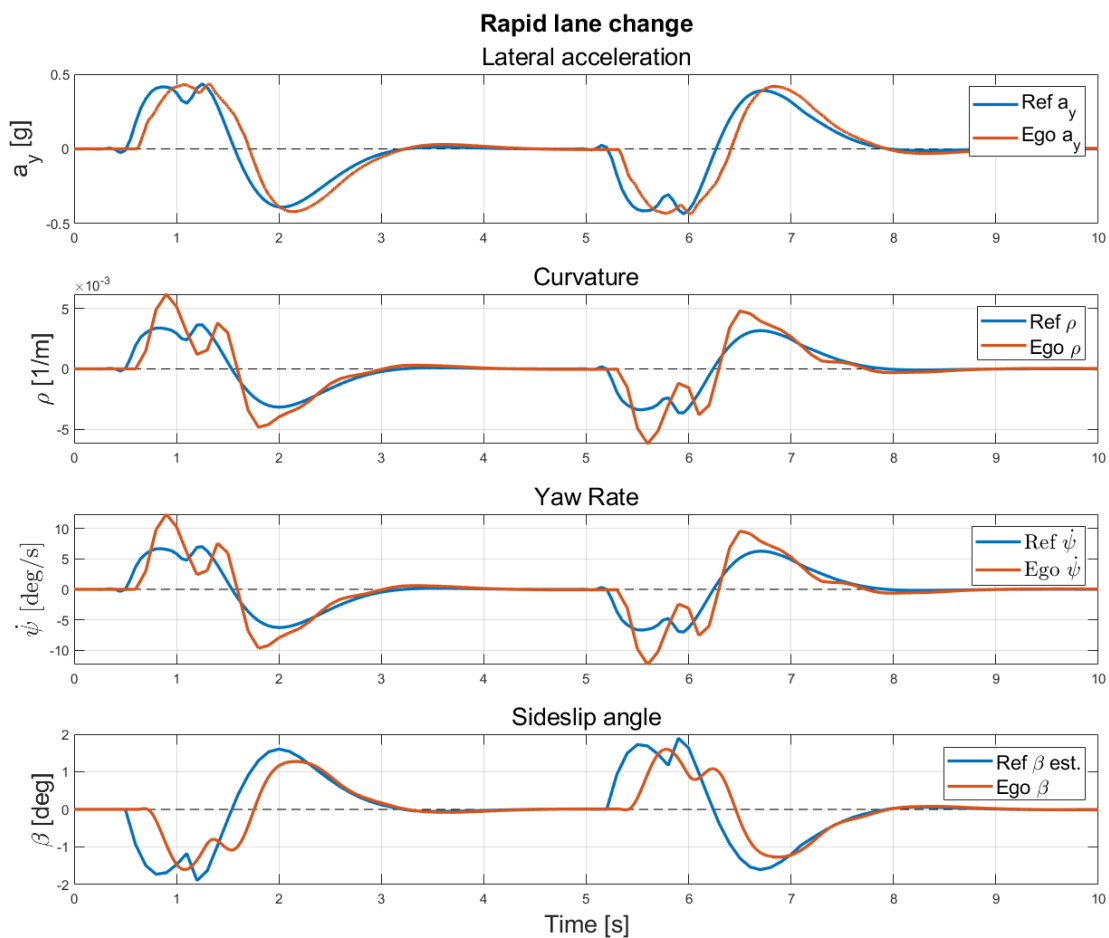


Figura 4.82: Confronto grandezze traiettoria ottimale e traiettoria veicolo *ego* - scenario "Modalità di cambio corsia rapido".

Le curve blu dei grafici riportati nelle figure 4.59 e 4.82 rappresentano le grandezze associate alla traiettoria di riferimento, rispettivamente prima e dopo l'inserimento della verifica di stabilità. Effettuando un loro confronto risulta evidente la minore richiesta di prestazioni nel caso con $Th = 2.s$.

Si riporta in tabella 4.24 un confronto tra i valori di picco delle grandezze riportate nei grafici citati.

Parameter	No Stab.Check	With Stab. Check	%Variation	Unit
Max $a_{y,Ref}$	0.99	0.44	-55.6%	g
Max ρ_{Ref}	$8.2 \cdot 10^{-3}$	$-3.6 \cdot 10^{-3}$	-56.1%	m^{-1}
Max $\dot{\psi}_{Ref}$	15.97	7.02	-56.0%	deg/s
Max β_{Ref}	7.60	1.89	-75.1%	deg

Tabella 4.24: Confronto richiesta di prestazioni prima e dopo l'inserimento della verifica di stabilità - scenario "Modalità di cambio corsia rapido".

Si riporta in figura 4.83, un confronto tra le traiettorie effettivamente percorse dal veicolo prima e dopo l'implementazione nel sistema della verifica di stabilità

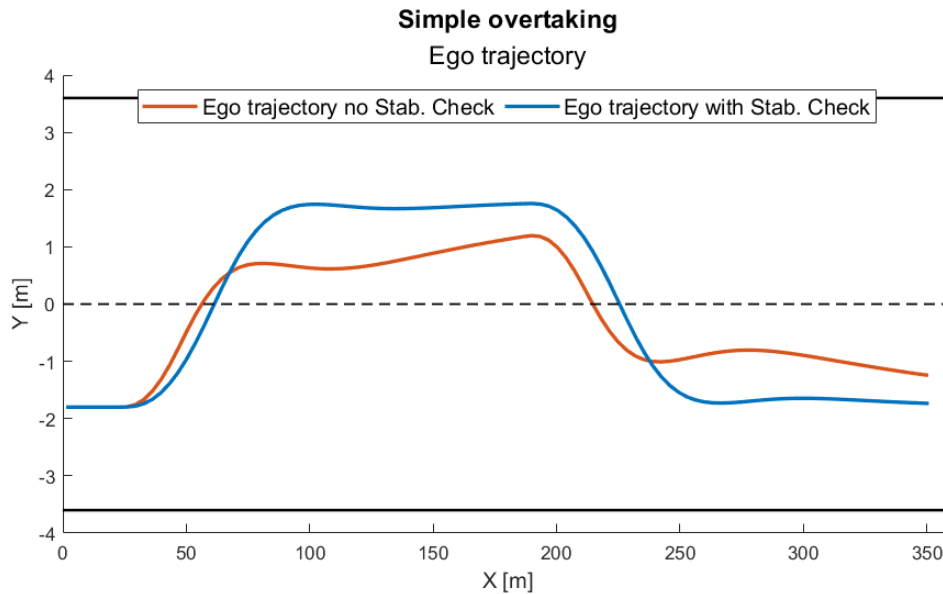


Figura 4.83: Confronto della traiettoria effettivamente percorsa dal veicolo *ego* prima e dopo l'introduzione della verifica - scenario "Modalità di cambio corsia rapido".

Per quantificare ulteriormente l'inferiorità della richiesta in termini di prestazione si riporta in figura 4.84 l'andamento dei segnali in output dal controllore dopo l'inserimento della verifica.

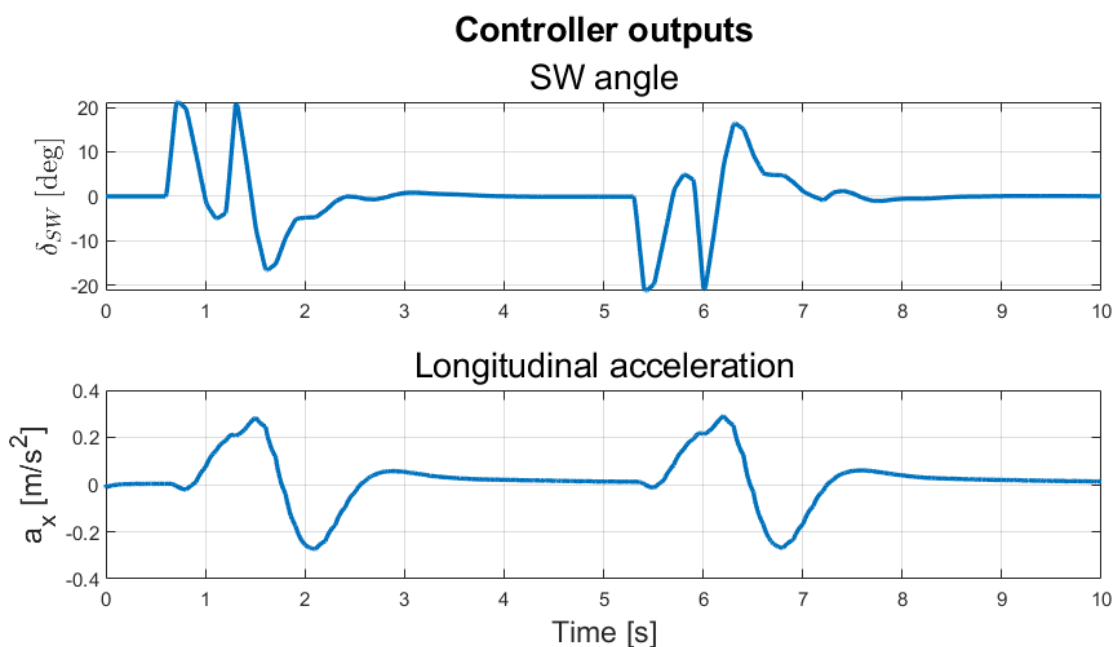


Figura 4.84: Segnali in output dal controllore *ego* - scenario "Modalità di cambio corsia rapido".

Nella tabella 4.25 sono stati inseriti i valori di picco dei segnali in output dal controllore prima e dopo l'inserimento della verifica. Essi sono stati calcolati in valore assoluto a partire dalle grandezze riportate nelle figure 4.58 e 4.84.

Parameter	No Stab.Check	With Stab. Check	%Variation	Unit
Max $a_{x,contr}$	0.66	0.29	-56.1%	<i>g</i>
Max $\delta_{SW,contr}$	68.50	21.20	-69.1%	<i>deg</i>

Tabella 4.25: Confronto richiesta di prestazioni prima e dopo l'inserimento della verifica di stabilità - scenario "Modalità di cambio corsia rapido".

A tale minore richiesta di prestazione segue la presenza di tutti i punti campionati della traiettoria individuata come riferimento nella zona di stabilità del piano delle fasi. Nella figura 4.85 si riporta la posizione dei punti della traiettoria di riferimento e di quelli della traiettoria effettiva all'interno del piano citato.

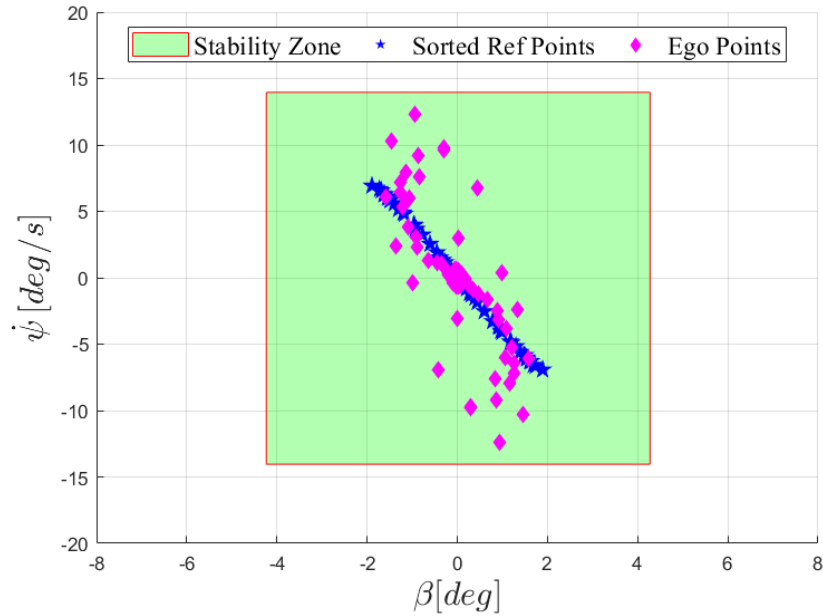


Figura 4.85: Confronto della posizione nel piano delle fasi dei punti delle traiettorie di riferimento ed effettiva - scenario "Modalità di cambio corsia rapido".

Si ricorda che la posizione dei limiti della zona di stabilità varia in relazione alla velocità del veicolo; tuttavia, nel caso presentato il veicolo mantiene la propria velocità longitudinale u costante, per tale motivo è stato possibile proporre la rappresentazione in figura.

Si nota che per entrambe le traiettorie tutti i punti sono all'interno della zona di stabilità. Nella seguente figura 4.86 si riporta il confronto tra l'andamento dello *Stability Index* associato ai punti delle due traiettorie, di riferimento ed effettiva..

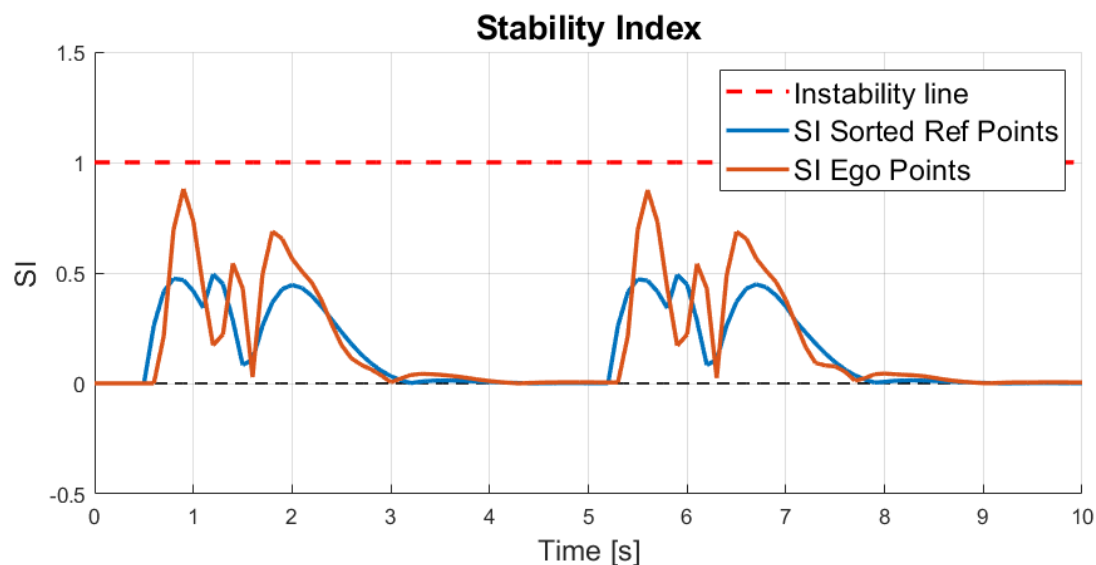


Figura 4.86: Confronto SI delle traiettorie di riferimento ed effettiva - scenario "Modalità di cambio corsia rapido".

La differenza di posizione e di *Stability Index* tra i punti delle due traiettorie è dovuta all'alterazione che i segnali subiscono in seguito all'azione del controllore e a tutte le semplificazioni introdotte per stimare i parametri associati alla traiettoria di riferimento.

Si consiglia per futuri sviluppi del sistema assistenziale di migliorare la stima delle grandezze citate, ad esempio introducendo dei contributi dipendenti dall'entità dell'alterazione di tali grandezze apportata dal controllore.

Mediante differenti simulazioni, le quali sono state condotte a differenti velocità, è stato individuato come minimo orizzonte temporale il valore $Th_{min} = 1.5 s$, poiché per valori inferiori la richiesta di prestazioni risulta troppo elevata nella maggior parte dei casi.

4.7 Simulazione test del controllore e del modello di veicolo

All'interno di questo paragrafo sono state riportate tutte le simulazioni condotte nel corso della creazione del modello complessivo. Tali risultati sono stati commentati alla fine dell'intera trattazione con il fine di evitare di esporre i relativi concetti non prima di aver introdotto tutte le tecniche di simulazione e di analisi dati, in modo da poterne facilitare la comprensione. Gli argomenti trattati sono i seguenti:

- Verifica funzionamento corretto modello complessivo con scenario a bassa richiesta di prestazioni.
- Introduzione delle caratteristiche di rigidità in deriva $C_{\alpha,F/R}$ nel modello integrato nel controllore MPC di tipo adattivo.
- Confronto risultati ottenuti con sistema di attuazione dello sterzo *slow/fast*.

4.7.1 Test 1 - Verifica funzionamento modello complessivo con scenario a bassa richiesta di prestazioni

Dopo aver completato l'accoppiamento di tutte le parti del modello completo sono stati eseguiti vari test con scenari poco critici per verificare l'effettivo funzionamento ed il corretto scambio di informazioni tra le parti. Lo scenario esaminato è quello di un doppio cambio di corsia, nel quale il veicolo *ego* effettua un sorpasso ai danni del veicolo *lead*. La manovra viene eseguita ad accelerazioni laterali massime non troppo elevate. Quest'ultima caratteristica è stata ricercata in quanto proporre uno scenario con un'impattante criticità avrebbe potuto produrre delle alterazioni dei segnali tali da non permettere di capire se effettivamente fossero stati presenti problemi di accoppiamento tra le parti del modello.

Si riporta in figura 4.87 il confronto tra la traiettoria di riferimento e quella effettivamente seguita dal veicolo *ego*.

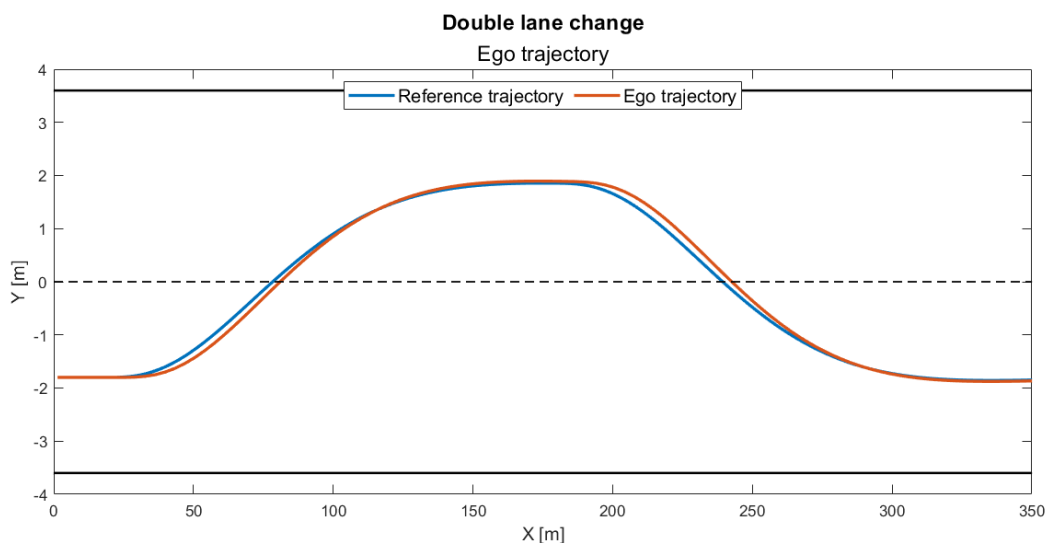


Figura 4.87: Confronto traiettorie veicolo *ego* di riferimento/effettiva .

Si nota immediatamente un'ottima corrispondenza tra le due traiettorie. Si riporta in figura 4.88 l'andamento degli errori di posizione ed orientamento nel corso del tempo.

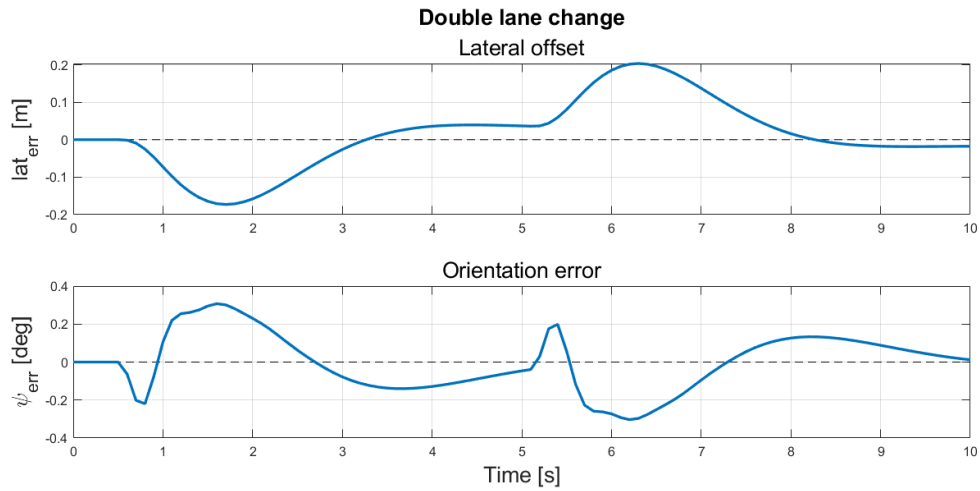


Figura 4.88: Andamento errori laterale ed angolare traiettoria effettiva veicolo *ego* al confronto con traiettoria di riferimento.

Si riportano nella seguente figura 4.89 gli output prodotti dal controllore nel corso della simulazione.

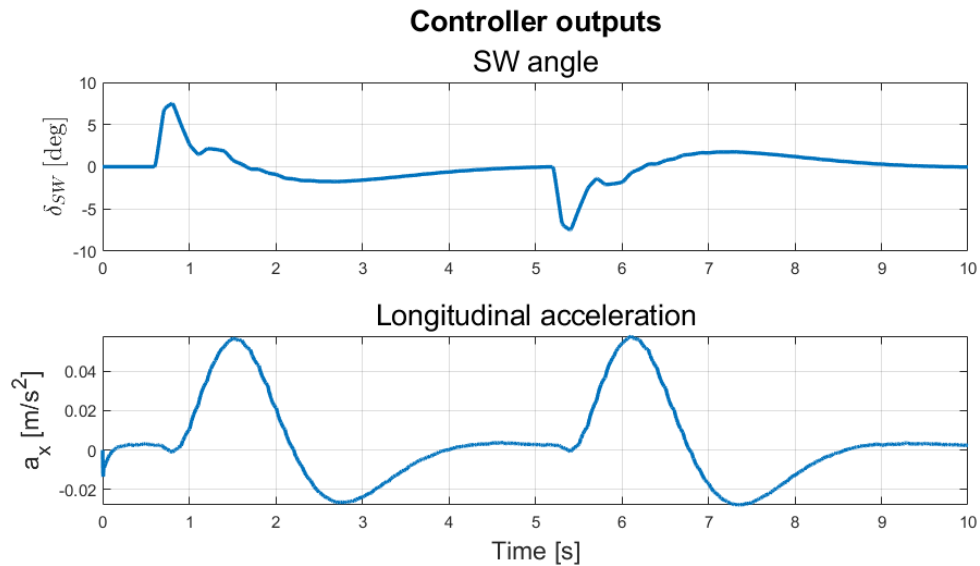


Figura 4.89: Andamento output prodotti dal controllore nel tempo.

Si riportano in figura gli andamenti delle principali grandezze del moto del veicolo *ego* e dei relativi riferimenti.

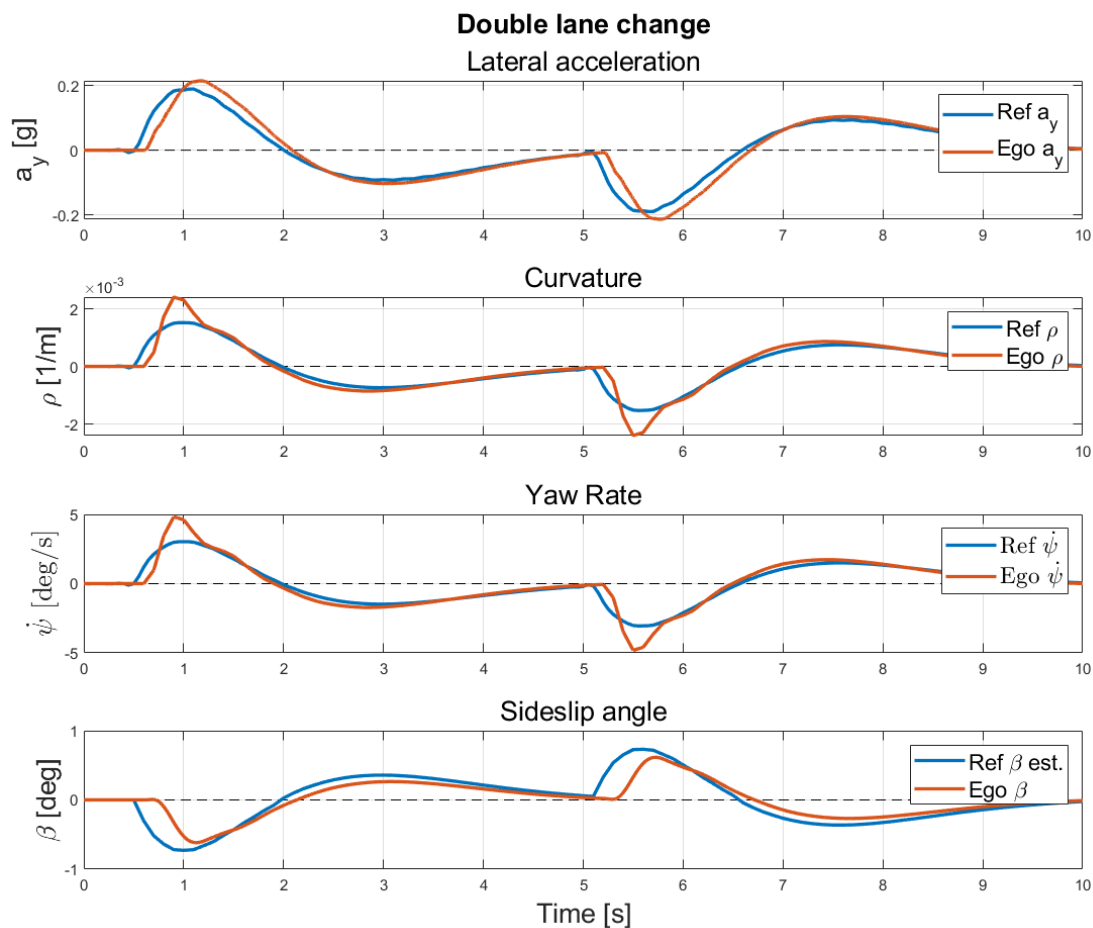


Figura 4.90: Andamento grandezze descrittive il moto del veicolo *ego* ed i rispettivi riferimenti.

Si nota la bassa richiesta in termini di prestazioni esaminando le curve blu di ogni grafico, le quali sono relative alla traiettoria di riferimento. In particolare si sottolinea l'accelerazione laterale massima pari a $a_{y_{max}} = 0.2g$. Tale considerazione è confermata anche dall'analisi dell'indice di stabilità, calcolato mediante la metodologia esposta nel paragrafo precedente, il quale rimane sempre molto al di sotto della *unsafe line*.

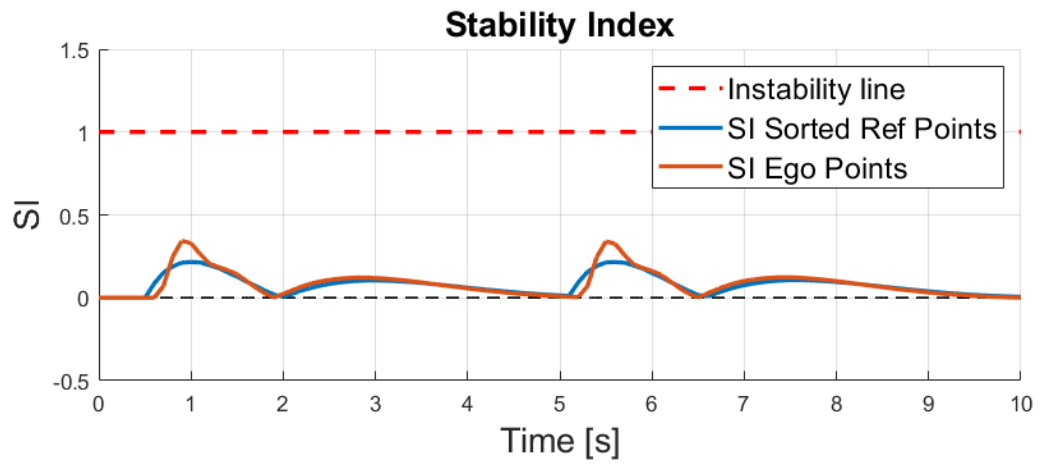


Figura 4.91: Andamento indice di stabilità traiettoria di riferimento/traiettoria effettiva.

4.7.2 Test 2 - Introduzione caratteristiche rigidzze in deriva nel controllore MPC

All'interno del capitolo terzo, nel paragrafo 3.2.2 dedicato al controllore, sono state espone da un punto di vista teorico le modifiche introdotte sul modello integrato nel controllore. Si tratta di un modello LTI i cui parametri vengono aggiornati ad ogni step temporale in quanto il controllore è di tipo adattivo. Per migliorare l'adattività del modello sono state inserite le mappe $C_{\alpha,F/R}(a_y)$ in grado di fornire i valori di rigidzza in deriva corrispondenti all'attuale valore di accelerazione laterale. I principi ed i limiti che accompagnano tale scelta sono stati esposti all'interno del paragrafo 3.2.2.

Per testare gli effetti della modifica introdotta, è stato eseguito un confronto tra due simulazioni effettuate rispettivamente a rigidzze in deriva costanti ed a rigidzze in deriva funzione dell'accelerazione laterale. Lo scenario adottato è una strada rettilinea nella quale transitano un veicolo *ego* ed un veicolo *lead*. La manovra eseguita dal veicolo *ego* è un doppio cambio di corsia, la cui accelerazione laterale massima richiesta è pari a $a_{y\ max} = 0.75\ g$.

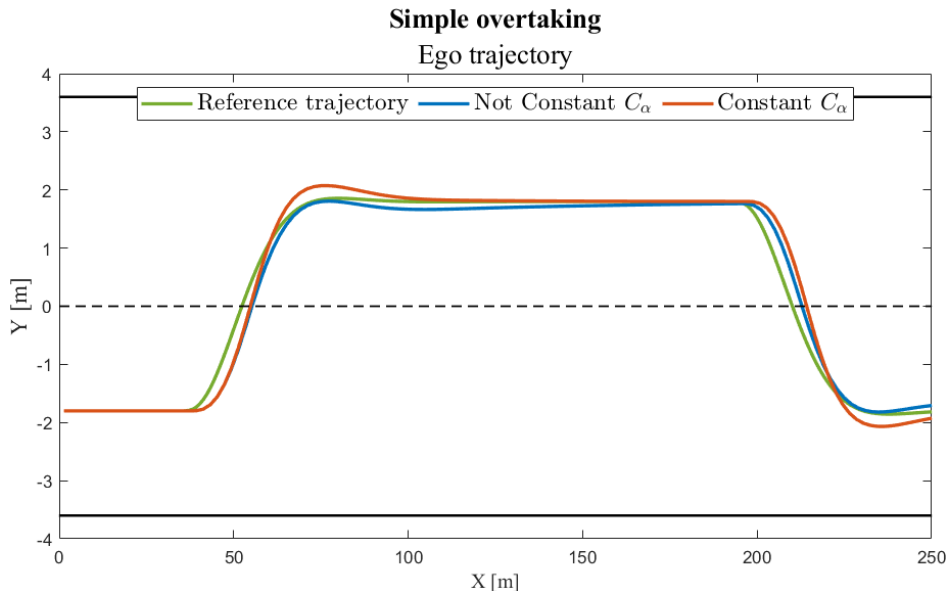


Figura 4.92: Confronto traiettorie veicolo *ego* con C_{α} costante/variabile.

In figura 4.92 è stato riportato un confronto tra le traiettorie effettive del veicolo *ego* e la traiettoria definita come riferimento. In blu si ha la traiettoria effettiva ottenuta dopo l'inserimento delle caratteristiche di rigidità in deriva all'interno del modello integrato nel controllore, in arancione quella ottenuta con rigidità in deriva costanti. La traiettoria di riferimento è stata progettata in modo da rientrare nel campo in cui i risultati di un modello di pneumatici lineare e di uno non lineare risultano differenti a causa di angoli di deriva abbastanza considerevoli. Come atteso, la traiettoria in blu risulta essere più fedele a quella di riferimento.

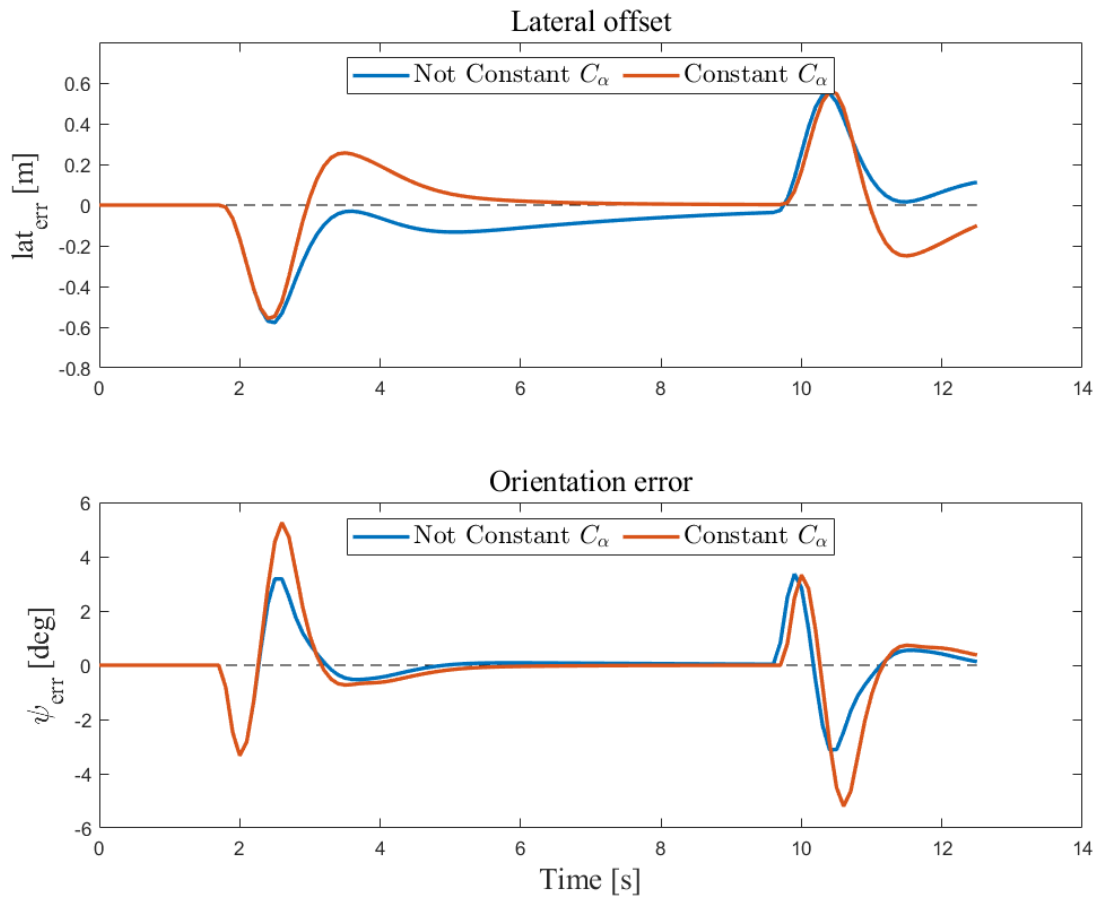


Figura 4.93: Confronto errori di traiettoria con C_α costante/variabile.

In figura 4.93 è stato riportato il confronto tra gli andamenti nel tempo degli errori di posizione e di orientamento. Il modello con rigidzze in deriva variabili, attua una correzione più rapida sull'angolo di imbardata, permettendo in tal modo di ridurre considerevolmente l'errore ad esso associato. Tuttavia, sebbene nella fase terminale della manovra di cambio corsia il modello a rigidzze variabili abbia uno scostamento laterale nettamente minore, la correzione rapida dell'angolo voltante fa in modo che il veicolo *ego* viaggi quasi parallelamente alla traiettoria di riferimento, con un errore laterale residuo che viene compensato dopo un lasso di tempo non esiguo.

4.8 Confronto tra sistema di attuazione dello sterzo *slow/fast*

All'interno del capitolo terzo è stata dettagliata la metodologia di accoppiamento tra le parti del modello complessivo. Nello specifico, i segnali con i quali comunicano il controllore ed il modello di veicolo sono l'accelerazione longitudinale a_x e l'angolo di sterzo δ . Nel caso del segnale di accelerazione, all'interno del capitolo terzo è stata dettagliata la logica che ha portato alla semplificazione della catena di attuazione di tale comando, per evitare di alterare i segnali in output al modello di veicolo in una fase dello studio dove esaminare con accuratezza i risultati delle operazioni di calibrazione effettuate sui tre principali componenti del modello complessivo (pianificatore, controllore e modello di veicolo) risulta di primaria importanza. Per tale motivo è stata svolta una sessione di test, relativa alla calibrazione della funzione di trasferimento associata all'output del controllore relativo all'angolo di sterzo.

$$\frac{\delta_f}{\delta_{f \text{ controll}}}} = \frac{1}{\tau s + 1} \quad (4.16)$$

Le costanti di tempo valutate sono riportate nella tabella 4.26. La costante che introduce il ritardo maggiore è stata chiamata τ_{slow} alla quale è stato associato un sistema di attuazione chiamato "lento"; la costante temporale più piccola τ_{fast} è stata associata ad un sistema di attuazione chiamato "veloce".

Parameter	Value	Unit
τ_{fast}	0.01	<i>s</i>
τ_{slow}	0.1	<i>s</i>

Tabella 4.26: Costante temporale adottata per funzione di trasferimento angolo di sterzo.

La manovra testata è la medesima per i due casi, ovvero una manovra di doppio cambio di corsia con la quale il veicolo *ego* esegue un sorpasso ai danni del veicolo *lead*.

Nelle figure 4.94 e 4.95 sono stati riportati gli andamenti dell'angolo di sterzo δ e la traiettoria dei due casi. Inoltre, è stato proposto il confronto con i rispettivi andamenti di riferimento.

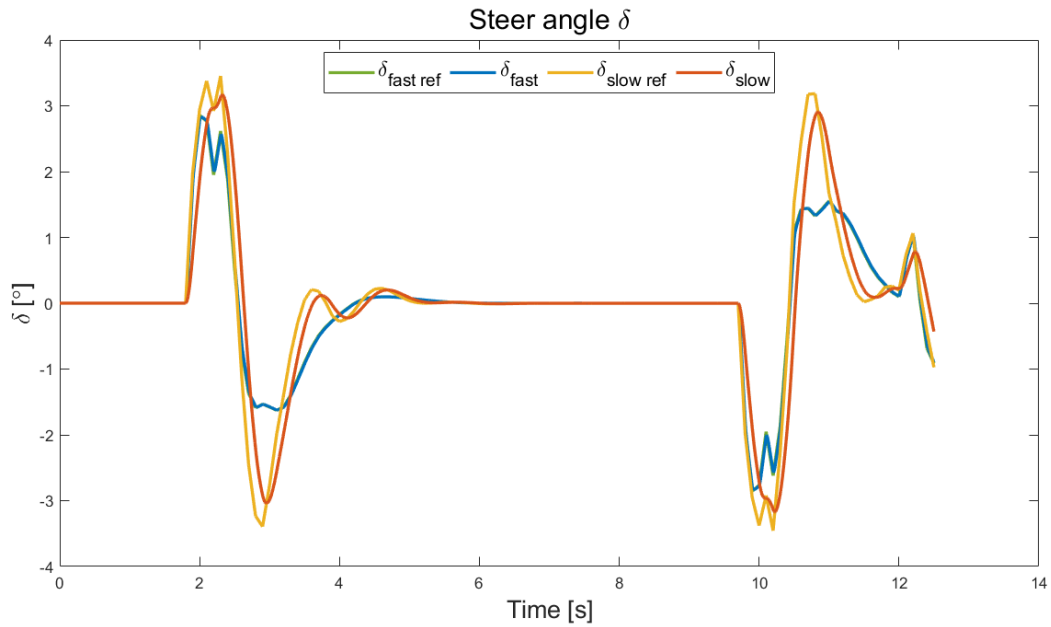


Figura 4.94: Angolo di sterzo confronto attuazione rapida/lenta.

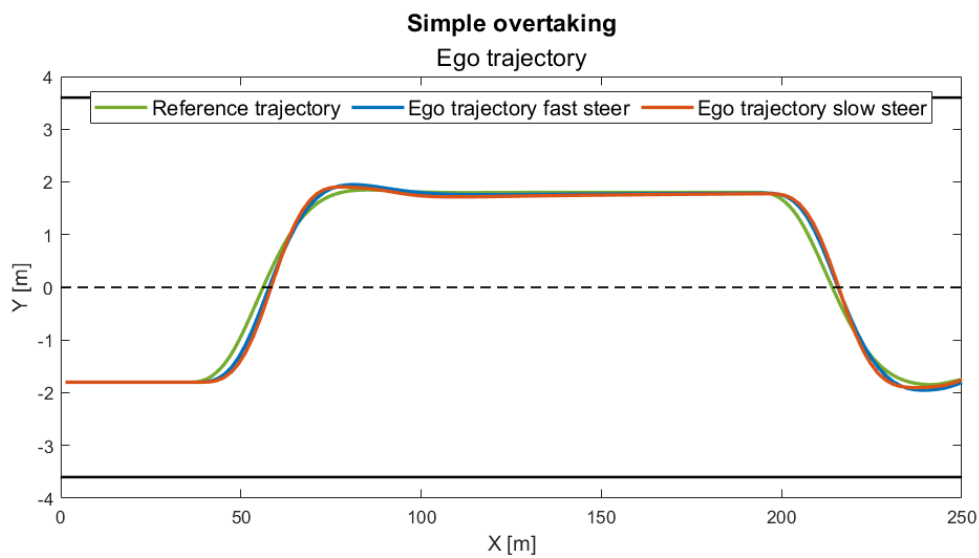


Figura 4.95: Traiettoria veicolo *ego* confronto attuazione di sterzo rapida/lenta.

Si noti che i riferimenti presenti nel grafico in figura 4.94 sono due, in quanto l'angolo di sterzo di riferimento non è altro che l'output del controllore. Quest'ultimo è differente nei due casi poiché essendo differenti le traiettorie effettive del veicolo, deve necessariamente esserlo anche il segnale fornito per correggere l'errore.

Confrontando i segnali dell'angolo di sterzo con i rispettivi riferimenti, si può notare come nel caso *fast* l'andamento dell'angolo δ fornito al modello si sovrappone puntualmente all'angolo δ_{ref} fornito dal controllore. Nel caso *slow* la differenza risulta maggiormente tangibile. Tuttavia, confrontando i segnali δ_{slow} e δ_{fast} si può notare come l'angolo di sterzo necessario a seguire la traiettoria di riferimento assume valori sensibilmente maggiori nel caso di attuazione lenta.

Confrontando le traiettorie si può notare come in entrambi casi il percorso di riferimento viene seguito abbastanza fedelmente, sebbene l'attuazione più rapida porti ad una maggiore precisione.

Risulta di particolare interesse esaminare le grandezze cinematiche e dinamiche le quali descrivono il moto del veicolo, si specifica che non sono stati riportati tutti i segnali ottenuti dal modello, in quanto l'intento della successiva disamina è quello di mostrare un esempio dell'alterazione degli andamenti che si otterrebbe ritardando l'attuazione del comando di sterzo. Sono stati riportati: l'andamento dell'angolo di imbardata, l'andamento della velocità, espressa nelle sue tre componenti nel sistema di riferimento non inerziale, l'andamento degli angoli di deriva e delle forze laterali delle singole ruote.

Osservando i grafici nelle figure 4.96, 4.97 e 4.98, è possibile notare che per tutte le grandezze il sistema di attuazione di sterzo più lento altera sensibilmente gli andamenti nei campi di funzionamento maggiormente critici.

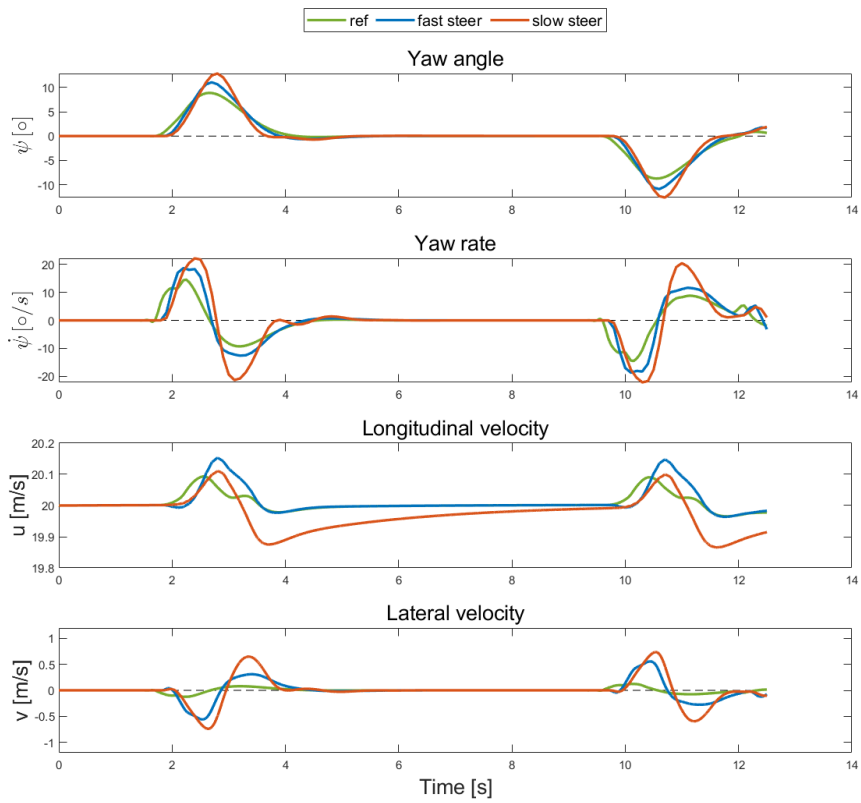


Figura 4.96: Grandezze moto veicolo *ego* confronto attuazione di sterzo rapida/-lenta.

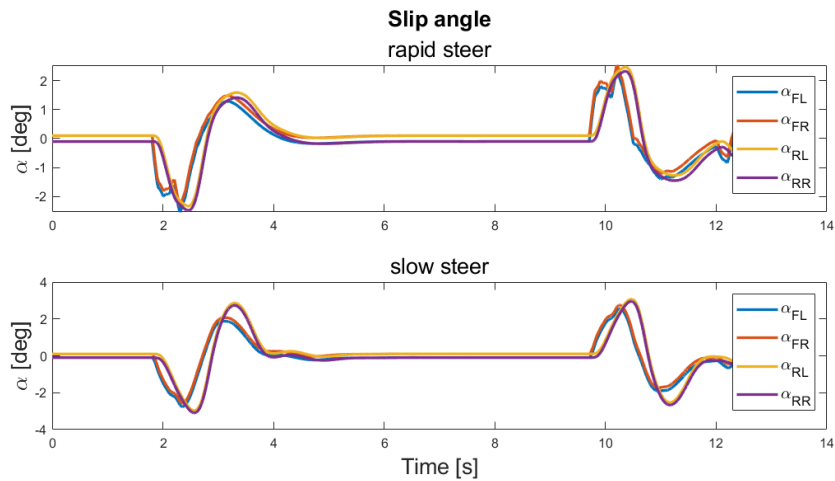


Figura 4.97: Angoli di deriva confronto attuazione di sterzo rapida/lenta.

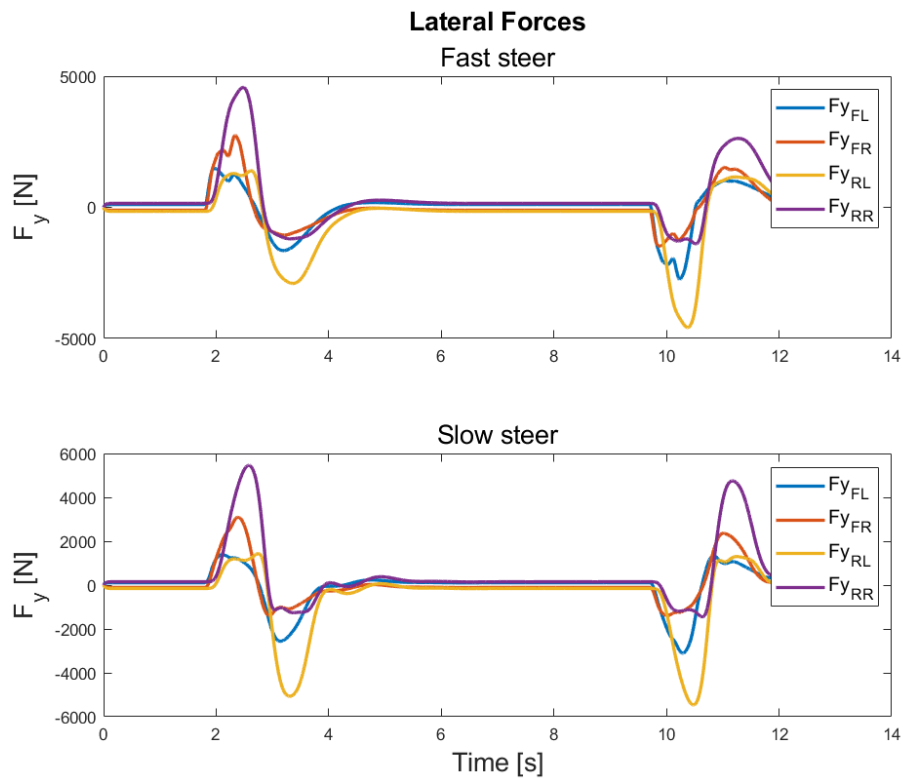


Figura 4.98: Andamento forze laterali confronto attuazione di sterzo rapida/lenta.

Coerentemente con quanto espresso in precedenza, è stato ritenuto opportuno adottare un sistema di attuazione *fast*, il quale è più distante da quello reale ma altera sensibilmente meno i segnali. Ciò è necessario per poter condurre in maniera più efficiente gli studi di sensitività sul pianificatore di traiettorie, sul controllore e sul modello di veicolo.

Capitolo 5

Conclusioni

In questa tesi sono state analizzate le caratteristiche di un sistema assistenziale avanzato in grado di: pianificare la corretta traiettoria di riferimento, stabilire la strategia di guida da attuare e, se ritenuto opportuno, effettuare una manovra di sorpasso. In primo luogo la trattazione ha esposto i principi su cui tale sistema si basa da un punto di vista puramente teorico. Su di essi è stato costruito un modello complesso e completo in ambiente di simulazione MatLab & Simulink. Il quale è formato da tre principali componenti:

- Pianificatore di traiettoria: in grado di analizzare lo scenario circostante, i parametri del veicolo e stabilire di conseguenza la traiettoria di riferimento.
- Controllore: di tipo adattivo e predittivo. Il quale minimizza gli errori di orientazione e posizionamento mediante una previsione degli stati futuri del veicolo *ego*, la quale viene effettuata attraverso un modello veicolo integrato.
- Modello veicolo: di tipo *Double Track*, con rollio e modello non lineare di pneumatici (modello *magic formula* di Pacejka).

Nella fase più avanzata dello studio è stata proposta una gamma di scenari, la quale è stata poi utilizzata per effettuare le simulazioni con il modello presentato. Esse hanno messo in evidenza alcune delle criticità del sistema assistenziale, portando a sviluppare strategie da implementare per renderlo capace di riconoscere tali criticità e di fornire una risposta adeguata.

Le strategie implementate hanno reso il sistema assistenziale in grado di:

- Effettuare manovre di sorpasso con un veicolo che transita in senso di marcia opposto.
- Effettuare manovre di sorpasso ai danni di una fila di veicoli, anche in presenza di un ulteriore veicolo che transita con senso di marcia opposto.
- Posizionare il veicolo *ego* correttamente all'interno della carreggiata. Avendo reso il sistema assistenziale coerente con le norme del c.d.s.
- Attuare in casi di emergenza una strategia di cambio di corsia rapido. Avendo introdotto una verifica sulla stabilità del veicolo, basata su tecniche di dinamica non lineare ed approcci grafici, la quale identifica la massima richiesta in termini di prestazioni.

5.1 Futuri sviluppi

Nel corso della trattazione sono stati evidenziati, qualora individuati, i limiti del sistema sviluppato; i quali possono essere considerati i punti di partenza di una successiva fase di studio. Nello specifico, gli sviluppi che porterebbero notevoli benefici in termini di potenzialità e funzionalità del sistema sono i seguenti:

- Implementazione di sensori del veicolo all'interno del modello utilizzato, in modo da far decadere l'ipotesi di V2X e di rendere il sistema utilizzabile non esclusivamente con infrastrutture avanzate.
- Aggiornamento del controllore utilizzato, con il fine di rendere il modello ad esso integrato maggiormente coerente con il modello veicolo utilizzato.
- Caratterizzazione dell'alterazione dei segnali apportata dal controllore, in modo da effettuare una stima maggiormente accurata dei parametri veicolo e di rendere maggiormente precise le verifiche sul set di traiettorie alternative.
- Sviluppo di nuove strategie, tali da ampliare la gamma di scenari correttamente analizzabili dal sistema. Nello specifico, risultano necessarie strategie da attuare in casi di emergenza, come l'arresto immediato del veicolo.

Appendice A

Functions

frenetStateConverter

input/output:

- *EgoAndTargetStates*: Bus output contenente gli stati in coordinate di Frenét di tutti i veicoli nello scenario.
- *Ego*: Segnale input contenente lo stato del veicolo *ego* in coordinate globali.
- *TargetActorsWorld*: Segnale input contenente gli stati dei veicoli *Target* in coordinate globali.
- *MapInfo*: Bus input contenente tutte le informazioni relative alla strada come: WayPoints della linea di mezzzeria, numero di punti, numero di corsie della strada e la relativa ampiezza.

```
function EgoAndTargetStates = frenetStateConverter(Ego, TargetActorsWorld,  
MapInfo)
```

1. Si definisce la variabile "RefPathFrenet" come persistente. In tal modo, ogni volta che la funzione viene utilizzata, la variabile "RefPathFrenet" contiene il vettore memorizzato in precedenza. Tale soluzione risulta essere molto utile, in quanto questa variabile ospita la traiettoria di riferimento del sistema di Frenét, la quale cambia quando cambia anche lo scenario.

Definendola come persistente si alleggerisce computazionalmente la funzione, evitando di doverla aggiornare ad ogni step temporale.

2. Si importano i punti della traiettoria di riferimento da "MapInfo.GlobalPoints" nella variabile "refPath"; dopodiché si esegue un check sulla traiettoria di riferimento in coordinate di Frénet ("RefPathFrenet"). Se quest'ultima variabile è vuota, ovvero all'avvio della simulazione, si impone:

$$\text{RefPathFrenet} = \text{referencePathFrenet}(\text{refPath});$$

"referencePathFrenet" è una funzione che adatta una curva continua ai punti che riceve in input con coordinate globali $[x, y]$ o $[x, y, \psi]$. I punti lungo la curva sono espressi come $[x, y, \psi, \kappa, \kappa', s]$, dove: x , y e ψ sono gli stati relativi alle coordinate globali; κ è la curvatura; κ' la derivata spaziale della curvatura rispetto alla lunghezza dell'arco, ed s è lunghezza dell'arco o distanza lungo il percorso dal punto di origine. Ciò che viene effettuato con questo passaggio è la creazione di una traiettoria di riferimento rispetto alla quale individuare mediante coordinate di Frénet tutti gli attori dello scenario. Inserendo come waypoints il vettore "MapInfo.GlobalPoints", contenente i "roadCenters", si sceglie come traiettoria di riferimento la linea di simmetria della strada.

3. Si inizializza il bus in output "EgoAndTargetStates" attraverso la funzione *HelperLCPlannerDefaultData*, utilizzata per restituire i dati di default dei segnali che si inviano come input.
4. Utilizzando il segnale in input "Ego" si crea il vettore "egoState" contenente lo stato del veicolo ego $[x, y, \psi, \kappa, v, 0]$; dopodiché si procede con il cambio di coordinate mediante la funzione *global2frenet* [17].

$$[x, y, \psi, \kappa, v, a] \rightarrow [s, \dot{s}, \ddot{s}, d, d', d''] \quad (\text{A.1})$$

Dove le derivate della coordinata curvilinea sono derivate temporali e le derivate dello scostamento sono derivate spaziali effettuate lungo la coordinata s .

A tale funzione vengono forniti in input lo stato del veicolo *ego* in coordinate globali "egoState" e la traiettoria a cui si vogliono riferire le coordinate di Frenét "RefPathFrenet".

5. Si procede ad eseguire la medesima operazione per tutti gli attori target presenti nello scenario, di cui si ricavano le informazioni dal segnale "TargetActorsWorld".
6. Infine, si convogliano tutti i segnali elaborati all'interno del bus "EgoAndTargetStates" inizializzato al punto terzo.

findMIOs

input/output:

- *leadVehicleFrenetState*: Segnale output contenente lo stato in coordinate di Frenét del veicolo *lead*, ovvero il veicolo che precede il veicolo *ego* nella medesima corsia di quest'ultimo.
- *mioFrenetStates*: Segnale output contenente tutti gli stati in coordinate di Frenét di tutti i veicoli definiti MIOs.
- *EgoAndTargetStates*: Bus input contenente gli stati in coordinate di Frenét di tutti i veicoli presenti nello scenario.
- *MapInfo*: Bus input contenente tutte le informazioni relative alla strada come: WayPoints della linea di mezzeria, numero di punti, numero di corsie della strada e la relativa ampiezza.

```
function [leadVehicleFrenetState, mioFrenetStates] = findMIOs(...
...EgoAndTargetStates, MapInfo)
```

1. Inizializzazione dei segnali di output mediante funzione *HelperLCPlannerDefaultData*, come nel caso precedente.
2. Si memorizzano in variabili dedicate i parametri derivati dai segnali di input, relativi agli stati del veicolo *ego* e dei veicoli *Target* in coordinate di Frénet, informazioni relative alla strada e la numerazione dei veicoli.
3. Si producono gli output della funzione elaborando i dati citati mediante la funzione *helperFindMIOs*.

helperFindMIOs

input/output:

- *leadVehicleFrenetState*: Segnale output contenente lo stato in coordinate di Frenét del veicolo *lead*, ovvero il veicolo che precede il veicolo *ego* nella medesima corsia di quest'ultimo.
- *mioFrenetStates*: Struttura output contenente tutti gli stati in coordinate di Frenét di tutti i veicoli definiti MIO, il numero totale di veicoli MIO e l'indice corrispondente ad ognuno di essi.
- *egoFrenetState*: Segnale input contenente lo stato in coordinate di Frenét del veicolo *ego*.
- *mapDB*: Segnale corrispondente a *MapInfo*. Bus input contenente tutte le informazioni relative alla strada come: WayPoints della linea di mezzzeria, numero di punti, numero di corsie della strada e la relativa ampiezza.
- *targetFrenetStates*: Segnale input contenente lo stato in coordinate di Frenét di tutti i veicoli *Target*.
- *targetIDs*: Segnale input contenente l'indice identificativo di ognuno dei veicoli *Target*.
- *initStruct*: Struttura input avente la stessa composizione della struttura in output, utile a manipolare i dati e ottenuta utilizzando la funzione *HelperLC-PlannerDefaultData*.

```
function [mioFrenetStates, leadVehicleFrenetState] = helperFindMIOs(...
...egoFrenetState, mapDB, targetFrenetStates, targetIDs, initStruct)
```

1. Attraverso l'ausilio della funzione *helperDetectLaneNumber*, utile ad identificare il numero della corsia dell'attore in input, si ricava la corsia del veicolo *ego* e dei veicoli *Target*.

2. Si memorizza il numero identificativo delle corsie interessate all'individuazione dei MIO nel vettore "validLanes". Si procede registrando la corsia del veicolo *ego* e le due che lo affiancano, per poi eliminare previa verifica quelle inesistenti qualora la corsia del veicolo *ego* fosse all'estrema destra o all'estrema sinistra.
3. Confrontando la posizione mediante la coordinata curvilinea s , si calcola la distanza dei veicoli *Target* dal veicolo *ego* e la si memorizza nel vettore "targetDist".
4. Si verifica per ogni corsia memorizzata nel vettore "validLanes" l'insieme dei veicoli presenti. Successivamente, si individua il veicolo con la minima distanza frontale dal veicolo *ego* e nel vettore "isMIO" l'indice corrispondente al veicolo target viene memorizzato come valore logico *true* (1).
5. Nel momento in cui, per una data corsia, viene identificato il veicolo che precede l'*ego* viene effettuata una seconda verifica; qualora la corsia analizzata fosse la medesima in cui transita il veicolo *ego* allora il veicolo target individuato viene registrato come *lead vehicle*. Ciò che nello specifico si realizza è la creazione di un vettore "leadVehicleFrenetState" in cui si memorizza lo stato del veicolo target con l'indice corrispondente a quello del *lead vehicle*.
6. Si ottiene con la medesima procedura l'identificazione dei veicoli target che precedono il veicolo *ego*.
7. Se sono stati individuati dei MIO, i dati raccolti vengono memorizzati nella struttura in output "mioFrenetStates". Nello specifico si riportano il numero totale di MIO individuati, l'indice corrispondente al veicolo target per ogni MIO ed il relativo stato.

updateEnvInfo

input/output:

- *CurrentEgoLane*: Segnale output che contiene l'indice della corsia corrente del veicolo *ego*.
- *PreferredLane*: Segnale output che contiene l'indice della corsia preferenziale del veicolo *ego*.
- *EgoAndTargetStates*: Bus input contenente gli stati in coordinate di Frenét di tutti i veicoli nello scenario.
- *LeadVehicleFrenetState*: Segnale input contenente lo stato in coordinate di Frenét del veicolo *lead*, ovvero il veicolo che precede il veicolo *ego* nella medesima corsia di quest'ultimo.
- *MioFrenetStates*: Struttura input contenente tutti gli stati in coordinate di Frenét di tutti i veicoli definiti MIO, il numero totale di veicoli MIO e l'indice corrispondente ad ognuno di essi.
- *MapInfo*: Bus input contenente tutte le informazioni relative alla strada come: WayPoints della linea di mezzzeria, numero di punti, numero di corsie della strada e la relativa ampiezza.
- *PlanneParams*: Bus input costruito all'interno del sottosistema *Planner configuration parameters*, contenente tutti i parametri di configurazione del pianificatore di traiettoria, originariamente importati dal Workspace.
- *PreferredLaneMinus*: Segnale input contenente l'indice della corsia preferenziale calcolato lo step temporale precedente.

```
function [CurrentEgoLane, PreferredLane] = updateEnvInfo(EgoAndTargetStates,
LeadVehicleFrenetState, MioFrenetStates, MapInfo, PlannerParams, PreferredLaneMinus)
```

1. Si rende persistente la variabile "obj".

2. Se ancora la variabile "obj" non esiste si impone:

```
obj.UserPreferredLane = PlannerParams.PreferredLane;
```

memorizzando di conseguenza la linea definita come preferita all'interno del WorkSpace nella struttura "obj".

3. Se la variabile "obj.UserPreferredLane" è diversa da quella ricevuta in input dal segnale "PlannerParams" (quella ricavata dal WorkSpace), allora si aggiorna "obj.UserPreferredLane" e si memorizza lo stesso valore nella variabile "PreferredLaneMinus" (sovrascrivendola, poiché la si riceve come input). Conseguentemente la funzione si occupa di riportare l'indice di *preferred line* al valore definito nel blocco *Planner configuration parameters*, ovvero il valore corrispondente alla traiettoria del veicolo *ego* descritta nello scenario..
4. Si inizializza la variabile "initstruct" come una struttura contenente le informazioni relative ai MIO mediante la già citata *HelperLCPlannerDefaultData*.
5. Dopo aver creato tutti gli input necessari nei precedenti passaggi, si utilizza la funzione *updateEnvironmentInfo*, definita in *HelperTerminalStateSampler*, richiedendo come output esclusivamente "CurrentEgoLane" e "PreferredLane".

updateEnvironmentInfo

input/output:

- *curEgoLane*: Segnale output che contiene l'indice della corsia corrente del veicolo *ego*.
- *preferredLane*: Segnale output che contiene l'indice della corsia preferenziale del veicolo *ego*.
- *distanceToLeadVehicle*: Segnale output contenente la distanza tra il veicolo *ego* ed il veicolo *lead*, calcolata rispetto alla coordinata curvilinea *s*.
- *leadVehicleTTC*: Segnale output contenente il valore del TTC calcolato fra i veicoli *ego* e *lead*.
- *obj*: Struttura input all'interno della quale viene memorizzato un valore iniziale di *PreferredLane* da sottoporre a successiva verifica.
- *egoFrenetState*: Bus input contenente lo stato in coordinate di Frenét del veicolo *ego*.
- *leadVehicleFrenetState*: Segnale input contenente lo stato in coordinate di Frenét del veicolo *lead*, ovvero il veicolo che precede il veicolo *ego* nella medesima corsia di quest'ultimo.
- *mioFrenetStates*: Struttura input contenente tutti gli stati in coordinate di Frenét di tutti i veicoli definiti MIO, il numero totale di veicoli MIO e l'indice corrispondente ad ognuno di essi.
- *laneInfo*: Struttura corrispondente a "MapInfo". Bus input contenente tutte le informazioni relative alla strada come: WayPoints della linea di mezzzeria, numero di punti, numero di corsie della strada e la relativa ampiezza.
- *plannerParams*: Bus input costruito all'interno del sottosistema *Planner configuration parameters*, contenente tutti i parametri di configurazione del pianificatore di traiettoria, originariamente importati dal Workspace.

- *preferredLaneMinus*: Segnale input contenente l'indice della corsia preferenziale calcolato lo step temporale precedente.
- *initStruct*: Struttura input generalmente utilizzata per la manipolazione dei dati ed avente la medesima composizione della struttura in output.

```
function [curEgoLane, preferredLane, distanceToLeadVehicle, leadVehicleTTC] =  
updateEnvironmentInfo(obj, egoFrenetState, leadVehicleFrenetState, mioFrenetStates, laneInfo, plannerParams, preferredLaneMinus, initStruct)
```

1. Si ricava la variabile "curEgoLane" mediante la funzione *Helper helperDetectLaneNumber*. La quale in input riceve le informazioni relative alla strada "laneInfo", corrispondente alla struttura "MapInfo", e la distanza del veicolo *ego* in direzione normale d dalla linea di riferimento.
2. Si ricava la variabile "preferredLane" mediante la funzione *helperFindPreferredLane* descritta in seguito.

helperFindPreferredLane

input/output:

- *preferredLane*: Segnale output che contiene l'indice della corsia preferenziale del veicolo *ego*.
- *mioInfo*: Struttura output contenente le informazioni relative ad ogni singolo veicolo identificato come MIO: distanza, velocità relativa e TTC calcolati rispetto al veicolo *ego* e il parametro "IsSafe". Quest'ultimo assume il valore logico "false", quando una delle grandezza citate non rispetta il relativo valore limite.
- *egoFrenetState*: Bus input contenente lo stato in coordinate di Frenét del veicolo *ego*.
- *mioFrenetStates*: Struttura input contenente tutti gli stati in coordinate di Frenét di tutti i veicoli definiti MIO, il numero totale di veicoli MIO e l'indice corrispondente ad ognuno di essi.
- *road*: Struttura corrispondente a "MapInfo". Bus input contenente tutte le informazioni relative alla strada come: WayPoints della linea di mezzzeria, numero di punti, numero di corsie della strada e la relativa ampiezza.
- *params*: Bus input costruito all'interno del sottosistema *Planner configuration parameters*, contenente tutti i parametri di configurazione del pianificatore di traiettoria, originariamente importati dal Workspace.
- *preferredLaneMinus*: Segnale input contenente l'indice della corsia preferenziale calcolato lo step temporale precedente.
- *outputStructObj*: Struttura input generalmente utilizzata per la manipolazione dei dati ed avente la medesima composizione della struttura in output.

```
function [preferredLane, mioInfo] = helperFindPreferredLane(egoFrenetState, mioFrenetState, road, params, preferredLaneMinus, outputStructObj)
```

1. Si importa nella variabile "preferredLane" la corsia preferita corrente memorizzata in "preferredLineMinus".
2. Si ottiene la corsia del veicolo *ego* mediante la funzione *helperDetectLaneNumber*.
3. Si inizializza una struttura con il nome di "mioInfo" ponendola uguale ad "outputStructObj", ricevuta in input come "initstruct". All'interno della stessa si memorizzano per ogni MIO la distanza relativa al veicolo *ego*, la velocità relativa, il TTC ed un flag "IsSafe", calcolati mediante la funzione *helperCalculateTTC*.
4. Utilizzando i dati appena ottenuti e processandoli attraverso la funzione *checkSafety* si sovrascrive il vettore "mioInfo.IsSafe" contenente il flag precedentemente citato. Esso assume valore logico "false" se la distanza relativa o il TTC non rispettano i parametri di sicurezza imposti.
5. Si crea un vettore logico "egoLane" della medesima lunghezza dei vettori contenuti nella struttura "mioInfo", ovvero con un numero di elementi pari al numero di MIO. Tale vettore ha l'elemento corrispondente al valore logico *true*, ovvero 1, quando il MIO con l'indice corrispondente si trova nella medesima corsia del veicolo *ego*. Esso viene utilizzato per comprendere se sono effettivamente presenti veicoli all'interno della corsia in cui transita il veicolo *ego* e se quest'ultimo risulta a rischio di collisione.
6. Qualora vi fossero uno o più veicoli nella corsia attuale del veicolo *ego* e fosse reale il rischio di collisione, si ripete la verifica per la corsia alla sinistra di quella attualmente occupata.
7. Se la corsia di sinistra risulta essere esistente e priva di rischio di collisione il vettore "preferredLane" viene aggiornato ponendolo uguale all'indice della corsia alla sinistra di quella attualmente occupata dal veicolo *ego*. In caso contrario si procede con la medesima operazione di verifica con la corsia di destra, e nel caso di verifica positiva si modifica la variabile "preferredLane".

helperCalculateTTC

input/output:

- *RangeMag*: Segnale output contenente il valore di *Relative distance* calcolato tra i due veicoli. Dove con *Relative distance* si identifica la distanza in termini assoluti tra i due punti associati ai due veicoli.
- *VelMag*: Segnale output contenente il valore di *Relative velocity* calcolato tra i due veicoli. Dove con *Relative velocity* si identifica la proiezione della velocità relativa dei due veicoli lungo la congiungente tra i due punti a loro associati.
- *ttc*: Segnale output contenente il tempo alla collisione (TTC). Calcolato come rapporto tra la *Relative distance* e la *Relative velocity*.
- *front*: Segnale flag output al quale viene associato il valore logico "true" se il suddetto veicolo MIO ha una coordinata curvilinea *s* superiore a quella del veicolo *ego*; ovvero se il veicolo MIO precede il veicolo *ego*.
- *egoFrenetState*: Bus input contenente lo stato in coordinate di Frenét del veicolo *ego*.
- *mioFrenetStates*: Struttura input contenente tutti gli stati in coordinate di Frenét di tutti i veicoli definiti MIO, il numero totale di veicoli MIO e l'indice corrispondente ad ognuno di essi.

```
function [RangeMag, ~, VelMag, ttc, front] = helperCalculateTTC(...
...egoFrenetState, mioFrenetState)
```

1. Si calcola la distanza relativa confrontando la posizione rispetto alla coordinata curvilinea *s*.
2. Si definisce il flag "front" come *true* se la distanza calcolata è maggiore di zero.

3. Confrontando gli stati di Frénet dei due veicoli si calcolano la distanza laterale d relativa, la velocità relativa longitudinale e quella laterale. Successivamente si processano i risultati modificando le coordinate da cartesiane a polari mediante la funzione di MatLab *cart2pol*; in tal modo e mediante ulteriori operazioni trigonometriche si ottengono le velocità relative e le distanze relative.
4. Il tempo di collisione (TTC) viene calcolato dividendo la distanza relativa con la velocità del veicolo.

cruiseControlSampler

input/output:

- *CruiseTerminalStates*: Struttura in output contenente le coordinate di Frenét di tutti gli stati terminali calcolati con la modalità CC.
- *obj*: Struttura input generalmente utilizzata per manipolare i dati all'interno della funzione.
- *CurrentEgoLane*: Segnale output che contiene l'indice della corsia corrente del veicolo *ego*.
- *PlannerParams.TimeHorizon*: Vettore input prelevato dal bus "PlannerParams" costruito all'interno del sottosistema *Planner configuration parameters*, contenente tutti i parametri di configurazione del pianificatore di traiettoria, originariamente importati dal Workspace. Il vettore "TimeHorizon" contiene il valore degli singoli orizzonti temporali per i quali si vuole calcolare uno stato terminale per ciascuna modalità. Conseguentemente ad un maggiore numero degli elementi del vettore corrisponderà un maggior numero di stati terminali per ciascuna modalità. Il valore del singolo elemento identifica il tempo che intercorre tra l'istante corrente e quello in cui il veicolo si troverà nell'ultimo punto del segmento di traiettoria pianificata.
- *PlannerParams.SetSpeed*: Parametro input prelevato dal bus "PlannerParams" costruito all'interno del sottosistema *Planner configuration parameters*, contenente tutti i parametri di configurazione del pianificatore di traiettoria, originariamente importati dal Workspace. Ad esso viene associato il valore della velocità \dot{s} alla quale verrà portato il veicolo *ego* in caso di funzionamento in modalità CC e LC.
- *MapInfo*: Bus input contenente tutte le informazioni relative alla strada come: WayPoints della linea di mezzzeria, numero di punti, numero di corsie della strada e la relativa ampiezza.

- *PlannerParams.EnableCCBehavior*: Parametro input prelevato dal bus "PlannerParams" costruito all'interno del sottosistema *Planner configuration parameters*, contenente tutti i parametri di configurazione del pianificatore di traiettoria, originariamente importati dal Workspace. Ad esso viene associata una variabile logica che se corrisponde al valore "false" inibisce la pianificazione di stati terminali nella suddetta modalità.
- *initStruct*: Struttura input generalmente utilizzata per la manipolazione dei dati ed avente la medesima composizione della struttura in output.

```
function CruiseTerminalStates = cruiseControlSampler(obj, CurrentEgoLane, PlannerParams.TimeHorizon, PlannerParams.SetSpeed, MapInfo,...
...PlannerParams.EnableCCBehavior, initStruct)
```

1. Si inizializza la struttura in output "CruiseTerminalStates".
2. Si crea la variabile "lateralOffsets" imponendo:

$$\text{lateralOffsets} = \text{laneInfo.LaneCenters}(\text{CurrentEgoLane});$$

Si impone che l'offset laterale dello stato terminale sia pari all'offset del centro della corsia in cui transita il veicolo *ego* rispetto al centro della strada.

3. Ricordando che per ogni colonna della struttura "CruiseTerminalStates" è presente uno stato e che gli stati di Frénet sono $[s, \dot{s}, \ddot{s}, d, d', d'']$, si impone \dot{s} pari a "setspeed", ovvero la velocità di crociera definita in *Planner Configuration Parameters* e d pari a "lateralOffsets". Si specifica che il numero di stati terminali che vengono calcolati è pari al numero di orizzonti temporali "timeHorizons" definito in *Planner Configuration Parameters* nel paragrafo 3.5. Ciò in cui gli stati terminali calcolati differiscono è il valore associato alla settima colonna, dedicata all'orizzonte temporale, la quale riporta il vettore "timeHorizons".

leadCarFollowingSampler

input/output:

- *lcfTerminalStates*: Struttura in output contenente le coordinate di Frenét di tutti gli stati terminali calcolati con la modalità LCF.
- *obj*: Struttura input generalmente utilizzata per manipolare i dati all'interno della funzione.
- *CurrentEgoLane*: Segnale output che contiene l'indice della corsia corrente del veicolo *ego*.
- *PlannerParams.TimeHorizon*: Vettore input prelevato dal bus "PlannerParams" costruito all'interno del sottosistema *Planner configuration parameters*, contenente tutti i parametri di configurazione del pianificatore di traiettoria, originariamente importati dal Workspace. Il vettore "TimeHorizon" contiene il valore degli singoli orizzonti temporali per i quali si vuole calcolare uno stato terminale per ciascuna modalità. Conseguentemente ad un maggiore numero degli elementi del vettore corrisponderà un maggior numero di stati terminali per ciascuna modalità. Il valore del singolo elemento identifica il tempo che intercorre tra l'istante corrente e quello in cui il veicolo si troverà nell'ultimo punto del segmento di traiettoria pianificata.
- *leadVehicleFrenetState*: Segnale input contenente lo stato in coordinate di Frenét del veicolo *lead*, ovvero il veicolo che precede il veicolo *ego* nella medesima corsia di quest'ultimo.
- *MapInfo*: Bus input contenente tutte le informazioni relative alla strada come: WayPoints della linea di mezzzeria, numero di punti, numero di corsie della strada e la relativa ampiezza.
- *PlannerParams.EnableLCFBehavior*: Parametro input prelevato dal bus "PlannerParams" costruito all'interno del sottosistema *Planner configuration parameters*, contenente tutti i parametri di configurazione del pianificatore di traiettoria, originariamente importati dal Workspace.

Ad esso viene associata una variabile logica che se corrisponde al valore "false", inibisce la pianificazione di stati terminali nella suddetta modalità.

- *initStruct*: Struttura input generalmente utilizzata per la manipolazione dei dati ed avente la medesima composizione della struttura in output.

```
function lcfTerminalStates = leadCarFollowingSampler(obj, currentEgoLane, timeHorizons, leadVehicleFrenetState, MapInfo, enableLCFBehavior, initStruct)
```

1. Si inizializza la struttura in output "lcfTerminalStates".
2. Si crea la variabile "lateralOffests" imponendo, come nel caso precedente, che sia pari alla distanza fra il centro della corsia del veicolo *ego* ed il centro della strada.
3. Si impone la velocità del veicolo pari alla velocità del veicolo da seguire, registrata nella seconda colonna della struttura "leadVehicleFrenetStates". Anche in questo caso si hanno un numero di stati terminali pari al numero di orizzonti temporali ed ognuno di essi differisce per l'orizzonte temporale definito nel vettore "timeHorizons".

laneChangeSampler

input/output:

- *LCStates*: Struttura in output contenente le coordinate di Frenét di tutti gli stati terminali calcolati con la modalità LC.
- *obj*: Struttura input generalmente utilizzata per manipolare i dati all'interno della funzione.
- *CurrentEgoLane*: Segnale output che contiene l'indice della corsia corrente del veicolo *ego*.
- *PlannerParams*: Bus input costruito all'interno del sottosistema *Planner configuration parameters*, contenente tutti i parametri di configurazione del pianificatore di traiettoria, originariamente importati dal Workspace.
- *MapInfo*: Bus input contenente tutte le informazioni relative alla strada come: WayPoints della linea di mezzzeria, numero di punti, numero di corsie della strada e la relativa ampiezza.
- *initStruct*: Struttura input generalmente utilizzata per la manipolazione dei dati ed avente la medesima composizione della struttura in output.

```
function LCStates = laneChangeSampler(obj, MapInfo, CurrenEgoLane, Planner-Params, initStruct);
```

1. Si inizializza la struttura in output "LCStates".
2. Si crea un vettore denominato "adjacentLanes" che contiene gli indici delle corsie esistenti adiacenti a quella del veicolo *ego*.
3. Si crea una variabile "lateralOffset" contenente l'offset dei centri corsia delle corsie adiacenti a quella del veicolo *ego*.

4. Nella struttura in output contenente gli stati terminali si memorizza la velocità s imponendola pari al valore "setSpeed" ricavato dal bus "PlanningParameters" e definito nel blocco *Planning Configuration Parameters*. Nella colonna della struttura relativa alla distanza d vengono memorizzati gli offset relativi a ciascuna corsia per ogni orizzonte temporale. Di conseguenza il numero di alternative presenti nel set di stati terminali corrispondenti alla modalità LC è pari al numero di elementi del vettore *timeHorizons* per il numero di corsie adiacenti.

ConcatenateStates

input/output:

- *TSCombination*: Struttura in output contenente il set di stati terminali, il numero totale e la relativa modalità (CC, LCF e LC).
- *CruiseStates*: Struttura in input contenente le coordinate di Frenét di tutti gli stati terminali calcolati con la modalità CC.
- *LCFTerminalStates*: Struttura in input contenente le coordinate di Frenét di tutti gli stati terminali calcolati con la modalità LCF.
- *LCStates*: Struttura in input contenente le coordinate di Frenét di tutti gli stati terminali calcolati con la modalità LC.

function TSCombination = ConcatenateStates(CruiseStates, LCFTerminalStates, LCStates)

1. Si inizializza la struttura in output "TSCombination".
2. Si calcola il numero totale degli stati sommando il numero di stati terminali realizzati dai tre blocchi descritti in precedenza.
3. Si concatenano gli stati in un'unica matrice "TSCombination.Combinations".
4. Si memorizza all'interno del vettore "TSCombination.BehaviorType" il tipo di modalità attivata in base all'indice dello stato terminato considerato.

stepImpl - HelperMotionPrediction

input/output:

- *FutureTrajectories*: Bus output contenente gli stati calcolati per ogni step di ciascun veicolo MIO, l'indice del relativo attore Target corrispondente ed il numero di punti di ciascuna traiettoria.
- *obj*: Struttura input generalmente utilizzata per manipolare i dati all'interno della funzione.
- *Params*: Bus input costruito all'interno del sottosistema *Planner configuration parameters*, contenente tutti i parametri di configurazione del pianificatore di traiettoria, originariamente importati dal Workspace.
- *MapInfo*: Bus input contenente tutte le informazioni relative alla strada come: WayPoints della linea di mezzeria, numero di punti, numero di corsie della strada e la relativa ampiezza.
- *MIOs*: Struttura input contenente tutti gli stati in coordinate di Frenét di tutti i veicoli definiti MIO, il numero totale di veicoli MIO e l'indice corrispondente ad ognuno di essi.

function FutureTrajectories = stepImpl(obj, Params, MapInfo, MIOs)

1. Inizializzando la struttura "obj", si memorizzano successivamente al suo interno i parametri relativi agli orizzonti temporali e alla risoluzione temporale; nello specifico "obj.TimeHorizon" e "obj.TimeResolution".
2. Si memorizza in "obj.RefPathFrenet.Waypoints" la traiettoria di riferimento estratta dal bus "MapInfo".
3. Si calcola il numero di stati che è necessario predire dividendo il più grande orizzonte temporale per la risoluzione temporale.

4. Si memorizza lo stato attuale in coordinate di Frénet nel vettore "initFrenetStates" e si crea il vettore temporale "time".
5. Successivamente si crea la matrice "frenetStates" con l'ausilio della funzione MatLab *repmat* che copia il vettore nelle righe successive un numero di volte pari a quanti sono gli stati da calcolare. Alla fine di tale operazione si avrà il medesimo stato iniziale ("initFrenetStates") in tutte le righe della matrice "frenetState"; le quali rappresentano gli istanti di tempo corrispondenti a quelli del vettore "time".
6. Attraverso un ciclo *for* si modificano gli stati riga per riga, aumentando la coordinata curvilinea s e la distanza dalla linea di riferimento d di quantità pari rispettivamente a Δs e Δd ad ogni istante di tempo. In tal modo si avrà un'evoluzione degli stati nel tempo a partire dalla prima riga. Le quantità Δs e Δd vengono calcolate nel seguente modo:

$$\begin{aligned}\Delta s &= \Delta t \dot{s} \\ \Delta d &= \Delta s d'\end{aligned}\tag{A.2}$$

7. Si utilizza la funzione MatLab *frenet2global* per ottenere la traiettoria futura del veicolo MIO in coordinate globali. Ad essa vengono forniti in input "obj.RefPathFrenet.Waypoints" e la traiettoria calcolata in coordinate di Frénet.
8. Attraverso un ciclo *for* si ripete tale operazione per ciascuno dei veicoli MIO.
9. Infine, si memorizzano nel bus in output "FutureTrajectories" i dati calcolati, quali: Traiettorie future, l'indice dei MIO corrispondente agli attori Target ed il numero di punti per ogni traiettoria.

helperEvaluateCost

input/output:

- *SortedTerminalStates*: Struttura in output contenente il set di stati terminali ordinato secondo una logica preferenziale, il numero totale e la relativa modalità (CC, LCF e LC).
- *TerminalStates*: Struttura in input contenente il set di stati terminali da ordinare, il numero totale e la relativa modalità (CC, LCF e LC).
- *PlannerParams*: Bus input costruito all'interno del sottosistema *Planner configuration parameters*, contenente tutti i parametri di configurazione del pianificatore di traiettoria, originariamente importati dal Workspace.
- *MapInfo*: Bus input contenente tutte le informazioni relative alla strada come: WayPoints della linea di mezzeria, numero di punti, numero di corsie della strada e la relativa ampiezza.
- *PlannerParams.SetSpeed*: Parametro input prelevato dal bus "PlannerParams". Ad esso viene associato il valore della velocità s alla quale verrà portato il veicolo *ego* in caso di funzionamento in modalità CC e LC. In tal caso, viene utilizzato questo valore come riferimento; in modo da poter valutare quanto si è distanti con una data traiettoria dalla velocità stabilita in fase di definizione dello scenario.
- *PreferredLane*: Segnale input che contiene l'indice della corsia preferenziale del veicolo *ego*.

```
function SortedTerminalStates = helperEvaluateCost(TerminalStates, PlannerParams, MapInfo, PlannerParams.SetSpeed, PreferredLane)
```

1. Si memorizza nella variabile "latDeviation" l'offset degli stati terminali inteso come distanza laterale d dal centro della corsia considerata come preferita. Il valore ottenuto è negativo per le corsie di sinistra.

2. Con lo scopo di dare preferenza ai cambi di corsia verso sinistra si moltiplicano per "-0.9" tutti gli elementi negativi del vettore "latDeviation", ovvero quelli relativi alle corsie alla sinistra del veicolo *ego*. In tal modo a parità di distanza, la funzione di costo porterà alla preferenza della corsia di sinistra piuttosto che di quella di destra.
3. Si procede con il calcolo del parametro di costo relativo allo scostamento laterale "latCost" di ognuno degli stati terminali. Si moltiplica "latDeviation" per il relativo peso introdotto nell'area *Cost Weights* del blocco *Planner Configuration Parameter* illustrato nel paragrafo 3.5.

$$latCost = latDeviation * latDevWeight \quad (A.3)$$

4. Si calcola il parametro di costo relativo al tempo "timeCost" moltiplicando il relativo parametro *Cost Weight* per l'orizzonte temporale del dato stato terminale.

$$timeCost = timeHorizon * timeWeight \quad (A.4)$$

5. Si calcola il parametro di costo relativo alla velocità "speedCost" moltiplicando il relativo parametro *Cost Weight* per la differenza in valore assoluto fra il parametro limite di velocità "speedLimit" e la velocità calcolata lungo l'ascissa curvilinea \dot{s} .

$$speedCost = |\dot{s} - setSpeed| * speedWeight \quad (A.5)$$

6. Dopo aver calcolato i tre parametri di costo si sommano all'interno della variabile "costTS" e si ordinano in maniera crescente mediante la funzione MatLab *sort*.

$$costTS = latCost + timeCost + speedCost \quad (A.6)$$

7. Nella nuova struttura vengono comunque specificati, per ogni parametro di costo, la modalità di simulazione (*BehaviorType*) e la relativa combinazione.

stepImp - HelperTrajectoryGenerator

input/output:

- *GlobalTrajectories*: Struttura in output contenente le n traiettorie in coordinate globali associate agli n stati terminali. Ad ognuna di esse viene associato: il vettore temporale, il numero di punti della traiettoria e la relativa modalità (CC, LCF e LC).
- *obj*: Struttura input generalmente utilizzata per manipolare i dati all'interno della funzione.
- *SortedTerminalStates*: Struttura in input contenente il set di stati terminali ordinato secondo una logica preferenziale, il numero totale e la relativa modalità (CC, LCF e LC).
- *timeStep*: parametro input che contiene l'unità di campionamento temporale Δt .
- *EgoAndTargetStates*: Bus input contenente gli stati in coordinate di Frenét di tutti i veicoli presenti nello scenario.
- *RefPointOnPath*: Bus input che contiene lo stato attuale di riferimento in coordinate globali.
- *MapInfo*: Bus input contenente tutte le informazioni relative alla strada come: WayPoints della linea di mezzzeria, numero di punti, numero di corsie della strada e la relativa ampiezza.
- *ReplanFlag*: Variabile booleana input che quando assume il valore logico "true" (1) esprime la necessità di analizzare il set di stati terminali per pianificare un nuovo segmento di traiettoria.

function GlobalTrajectories = stepImpl(obj, SortedTerminalStates, timeStep, EgoAndTargetStates, RefPointOnPath, MapInfo, ReplanFlag)

1. Si effettua una verifica su "ReplanFlag" e si entra all'interno del corpo della funzione se è positivo.

2. Al vettore "obj.RefPathPoints", contenente i waypoints della traiettoria di riferimento di Frenét, viene associato l'elemento di tipo "connector" "obj.Connector". Esso viene calcolato utilizzando la funzione *trajectoryGeneratorFrenet* e contiene le informazioni relative alla traiettoria di riferimento utili ai successivi step.
3. Attraverso le opportune manipolazioni dei segnali si ricava lo stato di riferimento del veicolo *ego* in coordinate di Frenét. Esso rappresenta il primo punto di tutte le traiettorie generate.
4. Si utilizza quello che dalla MathWorks viene chiamato il *connect method*; il quale permette di generare delle traiettorie mediante la funzione *connect*. La funzione viene fornita dal *Navigation Toolbox* ed è dedicata alla creazione di traiettorie in coordinate globali; queste vengono ottenute a partire da stati iniziali e finali in coordinate di Frenét e dagli archi temporali richiesti al raggiungimento degli stati terminali.
5. Dopo aver ricavato le traiettorie generate, per ognuna di esse si memorizza all'interno della struttura "GlobalTrajectories" il vettore temporale, il numero di punti della traiettoria, il *BehaviorType* con il quale è stata calcolata e le relative coordinate puntuali.

helperKinematicFeasibility

input/output:

- *validTrajectories*: Struttura in output contenente le n traiettorie in coordinate globali associate agli n stati terminali. Ad ognuna di esse viene associato: il parametro "IsValid", il vettore temporale, il numero di punti della traiettoria e la relativa modalità (CC, LCF e LC).
- *globalTraj*: Struttura in input contenente le n traiettorie in coordinate globali associate agli n stati terminali. Ad ognuna di esse viene associato: il vettore temporale, il numero di punti della traiettoria e la relativa modalità (CC, LCF e LC).
- *plannerParams*: Bus input costruito all'interno del sottosistema *Planner configuration parameters*, contenente tutti i parametri di configurazione del pianificatore di traiettoria, originariamente importati dal Workspace.

function validTrajectories = helperKinematicFeasibility(globalTraj, plannerParams)

1. Mediante un ciclo *for*, per ogni traiettoria si ricavano i vettori curvatura, velocità, accelerazione e velocità di imbardata.
2. Si ricavano i valori di picco di curvatura, accelerazione e velocità di imbardata.
3. Si esegue un'operazione di verifica confrontando i parametri ricavati con i valori limite ricavati dal segnale "plannerParams". I valori limite sono stati opportunamente definiti all'interno del sottosistema *Planner Configuration Parameters* descritto nel paragrafo 3.5. La verifica viene effettuata sui valori di picco raggiunti da accelerazione, curvatura e velocità di imbardata, e sui valori della velocità, i quali non devono mai essere inferiori alla soglia.
4. Se il confronto con i parametri limite ha esito positivo per tutte le grandezze riportate, la traiettoria esaminata viene dichiarata valida e l'informazione viene registrata all'interno del parametro "IsValid" con il valore logico "true".

sepImpl - HelperCollisionChecker

input/output:

- *TrajectoryInfo*: Struttura output contenente le n traiettorie in coordinate globali associate agli n stati terminali. Ad ognuna di esse viene associato: il parametro "IsValid", il parametro "IsColliding", il vettore temporale, il numero di punti della traiettoria e la relativa modalità (CC, LCF e LC). Inoltre al suo interno è presente il parametro "OptimalTrajectoryIndex" che memorizza l'indice della traiettoria ottimale.
- *TrajectoryReplan*: Variabile booleana output che quando assume il valore logico "true" (1) esprime la necessità di analizzare il set di stati terminali per pianificare un nuovo segmento di traiettoria.
- *obj*: Struttura input generalmente utilizzata per manipolare i dati all'interno della funzione.
- *ValidTrajectories*: Struttura in input contenente le n traiettorie in coordinate globali associate agli n stati terminali. Ad ognuna di esse viene associato: il parametro "IsValid", il vettore temporale, il numero di punti della traiettoria e la relativa modalità (CC, LCF e LC).
- *EgoAndTargetStates*: Bus input contenente gli stati in coordinate di Frenét di tutti i veicoli presenti nello scenario.
- *FutureTrajectory*: Bus input contenente gli stati calcolati per ogni step di ciascun veicolo MIO, l'indice del relativo attore Target corrispondente ed il numero di punti di ciascuna traiettoria.
- *VehicleProfiles*: Struttura input ottenuta mediante un blocco *Constant*, importando all'interno del modello la struttura "ActorProfiles" presente nel *WorkSpace*. Essa contiene le informazioni circa i singoli veicoli definite all'interno dello scenario.

- *ReplanFlag*: Variabile booleana input che quando assume il valore logico "true" (1) esprime la necessità di analizzare il set di stati terminali per pianificare un nuovo segmento di traiettoria.
- *PlannerParams*: Bus input costruito all'interno del sottosistema *Planner configuration parameters*, contenente tutti i parametri di configurazione del pianificatore di traiettoria, originariamente importati dal Workspace.
- *EgoID*: Parametro input che contenente l'indice identificativo del veicolo *ego* nell'elenco dei veicoli presenti nello scenario.

```
function [TrajectoryInfo, TriggerReplan] = stepImpl(obj, ValidTrajectories,...
...EgoAndTargetStates, FutureTrajectory, VehicleProfiles, ReplanFlag,...
...PlannerParams, EgoID)
```

1. Si memorizzano parametri utili alla definizione delle capsule all'interno della struttura "obj" quali: l'indice che identifica il veicolo *ego* ed i veicoli Target, l'estensione dei veicoli *ego* e target, il massimo orizzonte temporale di pianificazione e l'unità di campionamento temporale in secondi. Inoltre; si inizializzano gli output "TrajectoryInfo" e "TriggerReplan".
2. Vengono rimossi dalla lista *obstacle* tutti i veicoli non identificati come MIO.
3. Se il "ReplanFlag" restituisce il valore 1 (*true*) si procede come nei punti dal 4 al 7.
4. Si memorizzano all'interno dell'oggetto *dynamicCapsuleList* le geometrie del veicolo *ego* e dei veicoli MIO.
5. Si ricavano le coordinate dei MIO $[x, y, \psi]$ dalla struttura "FutureTrajectory" e si memorizzano come stati degli oggetti "obstacles" nella relativa lista dell'oggetto *dynamicCapsuleList*.

6. Si effettua la verifica delle traiettorie memorizzate nella struttura "ValidTrajectories" sulle collisioni con i veicoli MIO. Mediante un ciclo *for* ogni traiettoria viene analizzata singolarmente. La verifica viene effettuata aggiornando la matrice degli stati del veicolo *ego* nell'oggetto *dynamicCapsuleList* e processandolo mediante la funzione *checkCollision*. Quest'ultima è lo strumento utilizzato per verificare ad ogni step la presenza/assenza di sovrapposizione fra le aree delle capsule. L'output della funzione in questione è un vettore di variabili booleane, che ad un dato step associa il valore 1 (*true*) se viene rilevata una collisione, ergo una sovrapposizione fra le aree delle capsule.
7. Se il vettore in output alla funzione *checkcollision* risulta avere al proprio interno esclusivamente variabili booleane 0 (*false*), allora la traiettoria analizzata viene memorizzata nella struttura "obj.ExistingTrajectory" e con essa il relativo indice. Essendo le traiettorie ordinate in base al risultato della funzione di costo, la prima traiettoria che supera il check sulle collisioni sarà quella ottimale. Infine, viene associata la variabile logica *true* al flag "collisionFreeTrajectoryFound".
8. Se il "ReplanFlag" restituisce il valore 0 (*false*) si effettua una verifica sulla traiettoria attuale del veicolo *ego*. Essa viene effettuata utilizzando la medesima metodologia. Ovvero, si importa la matrice contenente la traiettoria futura del veicolo e la si memorizza all'interno dell'oggetto *dynamicCapsuleList*, in modo da poter utilizzare la funzione *checkCollision*. Qualora non venisse individuata una collisione viene memorizzata la variabile logica 1 (*true*) nel segnale "collisionFreeTrajectoryFound".
9. Infine, indipendentemente dal valore memorizzato in "ReplanFlag", viene effettuata una verifica sul parametro "collisionFreeTrajectoryFound". Qualora non fosse stata individuata alcuna traiettoria priva di collisioni, dunque "collisionFreeTrajectoryFound" restituisse il valore 0 (*false*), al parametro "TriggerReplan" verrebbe associata la variabile logica 1 (*true*). Tale segnale è collegato al blocco *OR* del sottosistema *motion Planner*, qualora fosse positivo esprimerebbe la necessità di pianificare un nuovo segmento traiettoria.

helperComputeCurrentStateBasedOnTime

input/output:

- *RefPoseRr*: Vettore output contenente la posizione del veicolo *ego* espressa dalle coordinate $[X, Y, \psi]$.
- *RefCurvature*: Parametro output contenente la curvatura della traiettoria del veicolo *ego*.
- *RefVelocity*: Parametro output contenente il valore della velocità del veicolo *ego*.
- *RefAccel*: Parametro output contenente il valore dell'accelerazione del veicolo *ego*.
- *RefAngVel*: Parametro output contenente il valore della velocità angolare del veicolo *ego*.
- *RefCurvatureDer*: Parametro output contenente il valore puntuale della derivata, calcolata rispetto al tempo, della curvatura della traiettoria veicolo *ego*.
- *TriggerReplan*: Variabile booleana output che quando assume il valore logico "true" (1) esprime la necessità di analizzare il set di stati terminali per pianificare un nuovo segmento di traiettoria.
- *time*: Parametro input contenente l'istante di simulazione attuale
- *replanTime*: Parametro input contenente l'istante temporale in cui viene pianificato il nuovo segmento di traiettoria.
- *numTrajPoints*: Parametro input che indica il numero di punti del segmento di traiettoria di riferimento.
- *optimalTrajectory*: Struttura input contenente i dati descrittivi del segmento di traiettoria di riferimento.

- *times*: Vettore input contenente gli istanti temporali associati ad ogni punto della traiettoria e il cui valore iniziale è nullo.
- *timeResolution*: Parametro input +contenente l'unità di campionamento temporale. Ovvero, il tempo che intercorre tra lo stato attuale ed il successivo stato temporale.

function [RefPoseRr, RefCurvature, RefVelocity, RefAccel, RefAngVel, RefCurvatureDer, TriggerReplan] = helperComputeCurrentStateBasedOnTime(time, replanTime, numTrajPoints, optimalTrajectory, times, timeResolution);

1. Si definisce il parametro "timeDelta" come:

$$timeDelta = time - replanTime + timeResolution; \quad (A.7)$$

Si tratta del parametro che identifica l'intervallo di tempo che intercorre fra lo stato attuale e quello da calcolare. Nel caso in cui si tratti del primo istante in cui viene analizzata una data traiettoria allora esso è uguale al valore di "timeResolution"; invece, per istanti successivi assume il valore del tempo trascorso dal primo stato in cui è stata analizzata la traiettoria.

2. Mediante interpolazione lineare effettuata utilizzando la matrice degli stati ed il vettore temporale, si ottengono i valori che assumono le grandezze che descrivono il moto del veicolo *ego* all'istante di tempo "timeDelta".
3. Qualora ci si avvicinasse all'orizzonte temporale senza ancora aver modificato la traiettoria, ovvero quando "timeDelta" è prossimo al valore finale del vettore "times", il valore di "replanFlag" viene modificato in 1 (*true*);

helperComputeCurrentStateBasedOnPose

input/output:

- RefPoseRr: Vettore output contenente le coordinate $[X, Y, \psi]$ del punto di riferimento corrente.
- RefCurvature: Parametro output contenente la nuova curvatura di riferimento da fornire al controllore.
- RefVelocity: Parametro output contenente la nuova velocità di riferimento da fornire al controllore.
- RefAccel: Parametro output contenente la nuova accelerazione di riferimento da fornire al controllore.
- RefAngVel: Parametro output contenente la nuova velocità di imbardata di riferimento da fornire al controllore.
- RefCurvatureDer: Parametro output contenente la nuova derivata della curvatura di riferimento da fornire al controllore.
- LatOffset: Parametro output errore di posizionamento del veicolo *ego* rispetto al punto di riferimento.
- RefYawDiff: Parametro output errore di orientamento del veicolo *ego* rispetto al riferimento.
- sectionStartIndex: Indice del punto di riferimento più prossimo al veicolo *ego*.
- currentPose: Vettore input contenente la posizione effettiva del veicolo *ego* espressa dalle coordinate $[X, Y, \psi]$.
- TrajectoryInfo.IsNew: Variabile booleana input che esprime se nel corrente step temporale il segmento di traiettoria di riferimento è nuovo. Qualora ad essa fosse associato il valore "false", implicherebbe la presenza del medesimo segmento di riferimento nel corso dello step temporale precedente.

- refPoses: Matrice input contenente traiettoria di riferimento espressa in coordinate $[X, Y, \psi]$.
- refCurva: Vettore input contenente la curvatura della traiettoria di riferimento.
- refVelo: Vettore input contenente il valore della velocità di riferimento.
- refAcc: Vettore input contenente il valore dell'accelerazione di riferimento.
- refAngVel: Vettore input contenente il valore della velocità angolare di riferimento.
- refCurvDer: Vettore input contenente il valore della derivata temporale della curvatura di riferimento.
- refPosePrev: Vettore input contenente le coordinate $[X, Y, \psi]$ del punto di riferimento precedente.
- refCurvPrev: Parametro input contenente la precedente curvatura di riferimento.
- refVelPrev: Parametro input contenente la precedente velocità di riferimento.
- refAccPrev: Parametro input contenente la precedente accelerazione di riferimento.
- sectionStartIndexPrev: Indice del punto di riferimento più prossimo al veicolo *ego* calcolato lo step precedente.

```
function [RefPoseRr, RefCurvature, RefVelocity, RefAccel, RefAngVel, RefCurvatureDer, LatOffset, RefYawDiff, sectionStartIndex] = helperComputeCurrentStateBasedOnPose(currentPose, TrajectoryInfo.IsNew, refPoses, refCurva, refVelo, refAcc, refAngVel, refCurvDer, refPosePrev, refCurvPrev, refVelPrev, refAccPrev, sectionStartIndexPrev);
```

1. Viene effettuata una verifica sul parametro "sectionStartIndex". Qualora fosse nullo viene posto pari ad 1. Nel corso della spiegazione tale valore verrà indicato con il nome "iStart".

2. Viene calcolata la distanza Δ_{XY} tra il punto P_{iStart} ed il suo successivo $P_{iStart+1}$.
I punti P_i sono i punti della traiettoria di riferimento "refPoses".

3. Viene calcolata la distanza R_{XY} tra il punto P_{iStart} ed il punto effettivo in cui si trova il veicolo *ego* P_{ego} .

4. Si calcola la quantità "u":

$$u = \left| \frac{R_{XY}}{\Delta_{XY}} \right| \quad (\text{A.8})$$

5. Si calcola la quantità "idxIncrement" arrotondando per difetto il parametro "u". E si aggiorna il valore "iStart", tale per cui:

$$iStart = iStart + idxIncrement \quad (\text{A.9})$$

6. Si calcolano i due parametri peso che verranno utilizzati in seguito:

$$\begin{aligned} Weight1 &= 1 - (u - idxIncrement) \\ Weight2 &= u - idxIncrement \end{aligned} \quad (\text{A.10})$$

7. Si calcolano gli indici del punto della traiettoria successivo al punto più vicino attuale, e del suo successivo "iNext":

$$\begin{aligned} iCurrent &= iStart + 1 \\ iNext &= iNext + 1 \end{aligned} \quad (\text{A.11})$$

8. Si calcolano le coordinate del punto di riferimento corrente:

$$\begin{aligned}
 XY_0 &= [X_{iStart}, Y_{iStart}] \\
 XY_1 &= [X_{iCurrent}, Y_{iCurrent}] \\
 Yaw_0 &= Yaw_{iStart} \\
 Yaw_1 &= Yaw_{iCurrent} \\
 XY_{Current} &= Weight1 * X_0 + Weight2 * X_1 \\
 Yaw_{Current} &= Weight1 * Yaw_0 + Weight2 * Yaw_1 \\
 ReferenceCurrentPosition &= [XY_{Current}, Yaw_{Current}]
 \end{aligned} \tag{A.12}$$

Si noti che i pesi inseriti agiscono in modo da interpolare spazialmente le grandezze

9. Si calcolano mediante la medesima interpolazione spaziale le grandezze descrittive del moto del veicolo. Si noti che in output vengono fornite le grandezze relative al punto da seguire e non a quello corrente; dunque, l'interpolazione viene effettuata tra i punti $P_{iStart+1}$ e $P_{iStart+2}$.

$$\begin{aligned}
 RefVelocity &= W1 * refVelo_{iCurrent} + W2 * refVelo_{iNext} \\
 RefAccel &= W1 * refAcc_{iCurrent} + W2 * refAcc_{iNext} \\
 RefAngVel &= W1 * refAngVel_{iCurrent} + W2 * refAngVel_{iNext} \\
 RefCurvature &= W1 * refCurva_{iCurrent} + W2 * refCurva_{iNext} \\
 RefCurvatureDer &= W1 * refCurvDer_{iCurrent} + W2 * refCurvDer_{iNext}
 \end{aligned} \tag{A.13}$$

10. Si calcolano gli errori di posizionamento e di orientamento:

$$\begin{aligned}
 LatOffset &= \sqrt{(X_{ego} - X_{Current})^2 + (Y_{ego} - Y_{Current})^2} \\
 RefYawDiff &= Yaw_{Current} - Yaw_{ego}
 \end{aligned} \tag{A.14}$$

11. Si memorizzano tutti le grandezze calcolate in nuovi parametri con la desinenza "Prev". Qualora nel successivo step il veicolo *ego* dovesse essere eccessivamente prossimo alla fine del segmento pianificato, essi verranno forniti come parametri delle grandezze in output.

Bibliografia

- [1] *Report incidenti stradali Italia 2022*. Rapp. tecn. ACI - ISTAT, 2023, p. 12 (cit. a p. 1).
- [2] *art.1, g, del decreto del Ministero delle infrastrutture e dei trasporti del 28 febbraio 2018 (c.d. smart road)*. Ministero delle infrastrutture e dei trasporti (cit. alle pp. 2, 11).
- [3] *ADAS, Advanced Driver Assistance Systems*. Continental Automotive. URL: <https://conti-engineering.com/areas-of-expertise/automotive/adas/> (cit. alle pp. 3, 6).
- [4] *Assistente al cambio corsia*. adasitalia.com. URL: <http://www.adasitalia.com/lca-lane-change-assist> (cit. a p. 4).
- [5] *Understanding ADAS: Lane Change Assist*. Caradas, apr. 2022. URL: <https://caradas.com/lane-change-assist/> (cit. alle pp. 4, 5).
- [6] *I sistemi di assistenza e comfort - Più sicuro per la guida quotidiana*. Porsche. URL: <https://media.porsche.com/mediakit/911-gts/it/the-new-porsche-911-gts-models/assistance-and-comfort-systems> (cit. a p. 5).
- [7] Alex Kierstein. *The BMW 5 Series Lets You Steer With Just Your Eyes*. Motortrend, mag. 2023. URL: <https://www.motortrend.com/news/2024-bmw-5-series-eye-lane-change-tech/> (cit. a p. 5).

- [8] Serdar Coskun. «Autonomous overtaking in highways: A receding horizon trajectory generator with embedded safety feature». In: *Engineering Science and Technology, an International Journal* (apr. 2021), pp. 1049–1058 (cit. a p. 7).
- [9] S. Zamfir, R. Drosescu e R. Gaiginschi. «Intelligent system for driver assistance in overtaking manoeuvres using multiple Camera and Radar Sensors». In: *IOP Conference Series: Materials* 997 (2020) (cit. a p. 7).
- [10] *SAE International J3016 - Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. Apr. 2021, p. 4 (cit. a p. 9).
- [11] Federico Cocianchich. «Sistemi ADAS obbligatori sulle auto nuove: cosa sono e perché servono». In: *Il Sole 24 Ore* (giu. 2024). URL: <https://www.ilsole24ore.com/art/sistemi-adas-obbligatori-auto-cosa-sono-e-perche-servono-AG7DdYb> (cit. a p. 10).
- [12] *Smart Road: Via libera in Gazzetta Ufficiale alle strade intelligenti*. Ministero delle Infrastrutture e dei Trasporti, feb. 2018 (cit. a p. 11).
- [13] Sven Ove Hansson, Matts-Åke Belin e Björn Lundgren. «Self-Driving Vehicles — an Ethical Overview». In: *Spinger Nature* (ago. 2021) (cit. a p. 12).
- [14] Moritz Werling, Julius Ziegler, Sören Kammel e Sebastian Thrun. «Optimal Trajectory Generation for Dynamic Street Scenarios in a Frenét Frame». In: (mag. 2010), pp. 987–993 (cit. alle pp. 16, 21, 24).
- [15] Alberto Ponso. «Decision Making and Path Planning algorithms for an Autonomous overtaking Vehicle on public road scenario». Tesi di laurea mag. Politecnico di Torino, 2022 (cit. alle pp. 19, 20).
- [16] Ross A. Knepper e Alonzo Kelly. «Differentially constrained mobile robot motion planning in state lattices. Journal of Field Robotics (JFR)». In: *Field Robotics* (2009) (cit. a p. 19).
- [17] *Convert global states to Frénet states*. MathWorks. URL: <https://it.mathworks.com/help/nav/ref/referencepathfrenet.global2frenet.html> (cit. alle pp. 23, 24, 312).

- [18] Zhewei Zhu, Xiaolin Tan, Yechen Qin, Yiwei Huang e Ehsan Hashemi. «A Survey of Lateral Stability Criterion and Control Application for Autonomous Vehicles». In: *IEEE* (ott. 2023) (cit. alle pp. 30, 32).
- [19] Zhongliang Han, Nan Xu, Hong Chen, Yanjun Huang e Bin Zhao. «Energy-efficient control of electric vehicles based on linear quadratic regulator and phase plane analysis». In: *elsevier* (dic. 2017) (cit. a p. 33).
- [20] *Dynamic capsule-based obstacle list*. MathWorks. URL: <https://it.mathworks.com/help/nav/ref/dynamiccapsulelist.html> (cit. alle pp. 37, 38).
- [21] *Automated Driving Toolbox*. MathWorks. URL: <https://it.mathworks.com/help/driving/index.html> (cit. alle pp. 48, 110).
- [22] *Driving Scenario Designer*. MarthWorks. URL: <https://it.mathworks.com/help/driving/ref/drivingsceniodesigner-app.html> (cit. alle pp. 48, 151, 152, 154).
- [23] *Navigation Toolbox*. MathWorks. URL: <https://it.mathworks.com/help/nav/index.html> (cit. a p. 48).
- [24] *Generate Code for Highway Lane Change Planner*. MathWorks. URL: <https://it.mathworks.com/help/driving/ug/generate-code-for-highway-lane-change-planner.html> (cit. alle pp. 49, 62, 104).
- [25] *Scenario Reader*. MathWorks. URL: <https://it.mathworks.com/help/driving/ref/scenarioreader.html> (cit. a p. 53).
- [26] *Path Following Control System*. MathWorks. URL: <https://it.mathworks.com/help/mpc/ref/pathfollowingcontrolsystem.html> (cit. alle pp. 109, 110).
- [27] *What is Model Predictive Control*. MathWorks. URL: <https://it.mathworks.com/help/mpc/ug/optimization-problem.html> (cit. a p. 111).
- [28] *Adaptive MPC*. MathWorks. URL: <https://it.mathworks.com/help/mpc/ug/adaptive-mpc.html> (cit. a p. 112).
- [29] *Optimization Problem*. MathWorks. URL: <https://it.mathworks.com/help/mpc/gs/what-is-mpc.html> (cit. alle pp. 117, 119).

- [30] *ASAM OpenDRIVE*. asam.net. URL: https://www.asam.net/fileadmin/Standards/OpenDRIVE/ASAM_OpenDRIVE_BS_V1-7-0.html (cit. a p. 154).
- [31] *Art.143.*Posizione dei veicoli sulla carreggiata*. Codice della strada (dlgs. 285/1992), TITOLO V - NORME DI COMPORTAMENTO (cit. a p. 251).
- [32] *Art.148.*Sorpasso*. Codice della strada (dlgs. 285/1992), TITOLO V - NORME DI COMPORTAMENTO (cit. alle pp. 251, 252).
- [33] Andrea Perfetti. «Polizia Stradale in azione: sorpasso e superamento a destra». In: *La Stampa* (giu. 2015). URL: <https://www.lastampa.it/motori/moto/2015/06/03/news/polizia-stradale-in-azione-sorpasso-e-superamento-a-destra-moto-it-1.35247989/> (cit. a p. 252).