



AUTONOMOUS DRIVING PROJECT

GRUPPO 12

a.a. 2022/2023

AMATO MARIO	m.amato72@studenti.unisa.it - 1670
AVITABILE MARGHERITA	m.avitabile6@studenti.unisa.it -1825
BATTIPAGLIA LUCIA	l.battipaglia6@studenti.unisa.it -1758

SOMMARIO

1	CAPITOLO 1	1
1.1	INTRODUZIONE	1
1.2	HARDWARE SETUP	1
1.3	OPERATIONAL DESIGN DOMAIN	2
1.3.1	ROUTE 0	2
1.3.2	ROUTE 1	2
1.3.3	ROUTE 2	3
1.3.4	ROUTE 3	3
1.3.5	ROUTE 4	3
1.4	VALUTAZIONE E METRICHE	3
1.5	BASELINE.....	4
1.6	CONSIDERAZIONI SULLA BASELINE	6
1.6.1	COLLISIONI CON ALTRI VEICOLI.....	6
1.6.2	RESTRINGIMENTO DELLA CARREGGIATA	6
1.6.3	COLLISIONI CON ELEMENTI STATICI.....	7
1.6.4	IGNORARE I SEGNALE DI STOP	7
1.6.5	GESTIONE DEGLI INCROCI	7
1.6.6	GUIDA FUORI STRADA.....	7
1.7	BEHAVIOUR PLANNER DELLA BASELINE.....	7
2	ARCHITETTURA SOFTWARE	10
2.1	TARATURA DEI CONTROLLORI.....	10
2.1.1	CONTROLLO LONGITUDINALE	10
2.1.2	CONTROLLO LATERALE	11
2.2	BEHAVIOUR PLANNER DELLA ENDLINE	13
2.3	FUNZIONE OVERTAKE_VEHICLES	14
2.4	FUNZIONE OVERTAKE_BICYCLES	16
2.5	FUNZIONE OVERTAKE_STATIC_OBSTACLE	19
2.6	FUNZIONE LANE_NARROWING_AND_STATIC_OBSTACLE.....	20
2.7	FUNZIONE COMPUTE_PARAMETERS_FOR_CRASHED_VEHICLES.....	20
2.8	FUNZIONE COLLISION_AND_CAR_AVOID_MANAGER	21
2.9	FUNZIONE STOP_MANAGER	21
2.10	GESTIONE INCROCI	22
3	CAPITOLO 3	24
3.1	RISULTATI OTTENUTI	24
3.2	TEST EFFETTUATI	25
3.3	ANALISI DEI RISULTATI.....	28

3.3.1	LIMITAZIONI DEL SIMULATORE: RESTRIZIONI E VINCOLI	28
4	CAPITOLO 4	31
4.1	ANALISI QUALITATIVA E SVILUPPI FUTURI	31
4.1.1	SORPASSO DI AUTO.....	31
4.1.2	SORPASSO DI OGGETTI STATICI	31
4.1.3	SORPASSO DI BICICLETTE	32
4.1.4	RESTRINGIMENTI	32
4.1.5	INCROCI.....	33
4.2	CONCLUSIONI	33

1 CAPITOLO 1

1.1 INTRODUZIONE

Il presente progetto si propone di sviluppare un software per la guida autonoma all'interno dell'ambiente di simulazione CARLA. L'obiettivo del software è consentire al veicolo di muoversi in modo autonomo, evitando pedoni e altri veicoli, rispettando le regole della strada e fermandosi in caso di semaforo rosso o di stop, il tutto nei limiti delle regole della strada. È richiesto di eseguire e valutare il nostro agente su cinque scenari della CARLA Leaderboard, ognuno avente una o più situazioni diverse da affrontare.

La baseline di partenza implementa un agente che esplora la scena per raggiungere una determinata destinazione. Questo agente è in grado di seguire correttamente i limiti di velocità, i semafori, tenendo conto anche dei veicoli nelle vicinanze. Oltre a questi possibili comportamenti, l'agente può anche mantenere una distanza di sicurezza da un'auto che gli sta davanti, monitorando il tempo istantaneo alla collisione e mantenendolo in un certo intervallo. Infine, nell'agente sono codificati diversi insiemi di comportamenti, da quelli cauti a quelli più aggressivi. Tale agente, tuttavia, non gestisce il comportamento agli incroci, il cambio corsia e alcune situazioni di emergenza che possono capitare nel contesto di guida. Per tale ragione non permette il raggiungimento di uno score soddisfacente al fine della valutazione finale.

La nostra soluzione cerca di limitare al minimo le infrazioni che il veicolo può commettere, implementando la funzionalità del cambio corsia, di gestione degli incroci, di gestione dei segnali di stop, di gestione dei ciclisti e di gestione delle situazioni di emergenza.

Di seguito analizzeremo più nel dettaglio l'Operational Design Domain (ODD), le scelte implementative che derivano da questa analisi, le soluzioni proposte, i test effettuati e i risultati ottenuti. Infine, analizzeremo i limiti della nostra soluzione ed effettueremo un'analisi qualitativa del risultato ottenuto.

1.2 HARDWARE SETUP

```
processor      : 16
vendor_id     : AuthenticAMD
cpu family    : 23
model         : 113
model name    : AMD Ryzen 7 3800X 8-Core Processor
stepping      : 0
microcode     : 0x8701013
cpu MHz       : 2033.553
cache size    : 512 KB
physical id   : 0
siblings      : 16
core id       : 7
cpu cores     : 8
apicid        : 15
initial apicid : 15
fpu           : yes
fpu_exception : yes
cpuid level   : 16
wp            : yes
```

MEMORIA

MemTotal: 16326188 kB
MemFree: 1858048 kB
MemAvailable: 7258768 kB

SCHEDA VIDEO

description: VGA compatible controller
product: GP104 [GeForce GTX 1080]
vendor: NVIDIA Corporation
physical id: 0
bus info: pci@0000:07:00.0
version: a1
width: 64 bits
clock: 33MHz
capabilities: pm msi pciexpress vga_controller bus_master cap_list rom
configuration: driver=nvidia latency=0

1.3 OPERATIONAL DESIGN DOMAIN

All'interno della baseline sono implementati tutti i meccanismi di mission planning e di navigazione all'interno della mappa. Effettuiamo un'analisi del contesto di guida in cui ci muoviamo (ODD), in termini di:

- Scena
- Condizioni Ambientali
- Elementi dinamici

per ognuno dei 5 scenari della CARLA Leaderboard presenti all'interno del file **route_avddiem_exam.xml**.

1.3.1 ROUTE 0

Questo scenario è caratterizzato da strade curve e tratti ad alta velocità, è uno scenario campestre.

Scenario	Condizioni Ambientali	Elementi Dinamici
Curve e tratti ad alta velocità	Passaggio da condizioni di buona illuminazione a condizioni di scarsa illuminazione , dovuto al calare della notte	Gruppo di 3 auto ferme lungo la carreggiata
Lunghezza 4211m	Condizioni di tempo nuvoloso	4 auto singole ferme lungo la carreggiata
1 incrocio con segnale di stop		4 gruppi di ciclisti
1 Traffic warning		2 pedoni che attraversano la strada
4 restringimenti della corsia opposta (l'altro verso di percorrenza)		

1.3.2 ROUTE 1

Questo scenario è caratterizzato da strade curve.

Scenario	Condizioni Ambientali	Elementi Dinamici
Presenza di curve	Passaggio da condizioni di buona illuminazione a condizioni di	Gruppo di 3 auto ferme lungo la carreggiata

	scarsa illuminazione , dovuto al maltempo	
Lunghezza 3207m	Passaggio da condizioni di tempo sereno a nuvoloso , con presenza di pioggia	1 auto singola ferma lungo la carreggiata
3 incroci , due dei quali con segnale di stop		5 gruppi di ciclisti
1 Traffic warning		2 pedoni che attraversano la strada
6 restringimenti della corsia opposta (l'altro verso di percorrenza)		

1.3.3 ROUTE 2

Questo scenario rappresenta un percorso urbano incentrato su aree ad alta densità con ostacoli.

Scenario	Condizioni Ambientali	Elementi Dinamici
Aree ad alta densità con ostacoli	Scena diurna con condizioni di buona illuminazione	
Lunghezza 884m	Condizioni di tempo sereno	
5 (riconta) incroci gestiti dai semafori		
2 incroci , di cui uno con segnale di stop		

1.3.4 ROUTE 3

Questo scenario rappresenta un percorso urbano incentrato su aree ad alta densità con ostacoli.

Scenario	Condizioni Ambientali	Elementi Dinamici
Aree ad alta densità con ostacoli	Scena diurna con condizioni di buona illuminazione	
Lunghezza 1231m	Condizioni di tempo sereno	
9 incroci gestiti dai semafori		

1.3.5 ROUTE 4

Percorso rurale incentrato sui pedoni che attraversano la strada.

Scenario	Condizioni Ambientali	Elementi Dinamici
Curve e tratti ad alta velocità	Scena diurna con condizioni di buona illuminazione	Gruppo di 3 auto ferme lungo la carreggiata
Lunghezza 900m	Condizioni di tempo nuvoloso	1 ciclista
4 incroci con segnale di stop		3 pedoni che attraversano la strada

1.4 VALUTAZIONE E METRICHE

La competenza di guida di un agente può essere caratterizzata da più metriche. Per questa classifica sono state selezionate una serie di metriche che aiutano a comprendere diversi aspetti della guida. Le metriche specifiche sono le seguenti:

- **Driving score:** $R_i P_i$ – La metrica principale della classifica, che funge da prodotto tra il completamento del percorso e la penalità per infrazioni. R_i è la percentuale di completamento della route i – esima, e P_i è la penalità di infrazione della route i -esima.
- **Route completion:** Percentuale della distanza del percorso completata da un agente.
- **Infraction penalty:** $\prod_j^{ped,...,stop} (P_{io}^j)^{\#infrazioni_j}$. La classifica tiene traccia di diversi tipi di infrazioni e questa metrica aggrega tutte queste infrazioni attivate da un agente come una serie geometrica. Gli agenti iniziano con un punteggio base ideale di 1.0, che viene ridotto a ogni tipo di infrazione commessa.

La media di ciascuna metrica di valutazione viene calcolata quando tutti i percorsi sono stati completati. Ciò che sarà oggetto di classifica dei progetti è il **GLOBAL DRIVING SCORE**. Di seguito riportiamo la penalità assegnata a ciascuna infrazione:

Infraction	Penalty	Description
Collisions with pedestrians	0.50	
Collisions with other vehicles	0.60	
Collisions with static elements	0.65	
Running a red light	0.70	
Running a stop sign	0.80	
Scenario Timeout	0.70	Vehicle stuck for 4 minutes. Raise Shutdown Event
Failure to maintain minimum speed	0.70	Fail to maintain the speed of other vehicles in traffic
Failure to yield to emergency vehicle	0.70	Fail to allow to pass an emergency vehicle
Off-road driving	No addition to Route Completion	

Per maggiori informazioni consultate il seguente [link](#).

1.5 BASELINE

La baseline di partenza implementa un agente che è già in grado di gestire molti degli scenari che possono presentarsi nel contesto di guida, tra cui:

- Gestione dei semafori
- Riconoscimento dei limiti di velocità
- Mantenimento della distanza di sicurezza da un'auto che gli sta davanti
- Rilevamento di pedoni
- Gestione di alcune situazioni di emergenza

Tuttavia, all'avvio della simulazione, si ottiene un punteggio insufficiente sia in termini di classifica per il progetto, sia per quanto riguarda il contesto dell'autonomia di guida che stiamo cercando di raggiungere.

Di seguito riportiamo i risultati ottenuti eseguendo tutte e cinque le route:

GLOBAL METRICS	
SCORE_COMPOSED	23.13
SCORE_ROUTE	82.42
SCORE_PENALTY	0.31

Lo **Score Composed**, indicato nella tabella come **Global Driving Score**, rappresenta la nostra metrica principale di interesse che cerchiamo di massimizzare. Questo punteggio rappresenta una media dei punteggi di guida di tutti gli scenari, offrendoci una visione delle nostre performance complessive. Quello che ci interessa capire è in quali scenari il nostro agente commette più infrazioni, quali sono le infrazioni che ci fanno

accumulare più penalità e quali sono le route che non vengono completate. Nella tabella di seguito, analizziamo, quindi, i punteggi ottenuti per le singole route.

	Route0	Route1	Route2	Route3	Route4	Totale
EVALUATION METRICS						
DRIVING SCORE	9.6e-05	5.52	46.92	60.00	3.20	23.13
ROUTE COMPLETION	100%	12.14%	100%	100%	100%	82.42%
INFRACTION PENALTY	1e-06	0.45	0.46	0.6	0.03	0.31
INFRACTIONS						
NUM INFRACTIONS	32	3	3	1	9	48
Collisions with pedestrians (0.50)	2	X	X	X	2	4
Collisions with other vehicles (0.60)	14	X	1	1	2	18
Collisions with static elements (0.65)	5	1	X	X	X	6
Running a red light (0.70)	X	X	X	X	X	X
Running a stop sign (0.80)	1	X	1	X	3	5
Scenario Timeout (0.70)	8	1	X	X	1	10
Failure to maintain minimum speed (0.70)	1	X	1	X	1	3
Failure to yield to emergency vehicle (0.70)	X	X	X	X	X	X
Off-road driving (0.0)	1	X	X	X	X	1
SHUTDOWN EVENTS						
Route Deviation	X	X	X	X	X	X
Agent Blocked	X	X	X	X	X	X
Simulation Timeout	X	X	X	X	X	X
Route Timeout	X	1	X	X	X	1

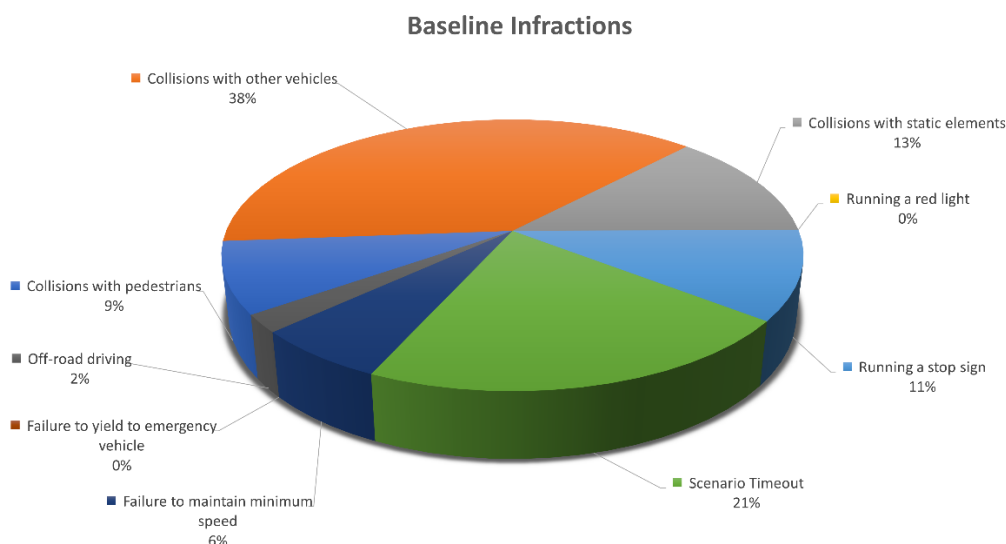
Dai risultati riportati nella tabella, emerge in modo più chiaro il motivo per cui abbiamo ottenuto un punteggio così basso. La baseline presenta significativi problemi nelle collisioni con altri veicoli e nei time-out di scenario, poiché non ha ancora implementate le funzionalità di:

- 1) **Gestione degli stop:** La baseline non è in grado di gestire correttamente le situazioni in cui è necessario fermarsi completamente ai segnali di stop. Questo porta a collisioni e a violazioni delle norme del codice stradale, come accade nella route 0, route 2 e route 4.
- 2) **Gestione degli incroci:** La baseline non gestisce in modo adeguato la dinamica degli incroci stradali, come la precedenza e le regole di destra. Questa mancanza di abilità causa incidenti nelle situazioni di intersezione, che si verificano in quasi tutti gli scenari.
- 3) **Sorpasso di biciclette:** La baseline non è dotata di una strategia per sorpassare in modo sicuro e corretto le biciclette sulla strada. Questo comporta situazioni di collisione, in particolare nelle route 0 e 1.
- 4) **Sorpasso di veicoli incidentati:** La baseline non è in grado di gestire efficacemente le situazioni in cui un veicolo è coinvolto in un incidente o si trova in difficoltà sulla strada. Questa mancanza di abilità porta a collisioni che creano ostacoli per il flusso del traffico.

- 5) **Comportamento agli ostacoli statici:** La baseline non tiene conto in modo adeguato dei segnali di avvertimento del traffico o dei coni per il restringimento della strada. Questa mancanza di considerazione porta a collisioni con i veicoli che si muovono lungo la strada.

1.6 CONSIDERAZIONI SULLA BASELINE

Considerando quanto emerso dall'analisi precedente, è evidente che il nostro punto di partenza presenta alcune limitazioni che abbiamo cercato di superare durante l'implementazione. Abbiamo dato priorità alle modifiche che potessero migliorare il punteggio globale finale (**Global Driving Score**). Per guidare questa decisione, abbiamo condotto un'analisi basata sulle penalità derivanti da ogni tipo di infrazione commessa sugli scenari in analisi e sulla frequenza con cui si verificano. In questo modo, abbiamo identificato gli aspetti critici su cui concentrarci per ottenere un miglioramento significativo delle prestazioni generali. Attraverso il diagramma riportato di seguito si riesce ad avere un'idea più chiara del punto di partenza.



1.6.1 COLLISIONI CON ALTRI VEICOLI

Possiamo osservare che su un totale di 48 infrazioni commesse, 18 di queste sono il risultato di collisioni del nostro veicolo con altri. Questo comportamento porta a una diminuzione dell'infraction penalty (riduzione del punteggio per collisione con altri veicoli di 0.6) e conseguentemente del driving score, poiché queste due misure sono direttamente proporzionali. Abbiamo analizzato i casi in cui si verificano tali collisioni e abbiamo constatato che in 6 occasioni il nostro veicolo entra in collisione con veicoli parcheggiati sulla corsia, altre 6 volte si scontra con biciclette che si trovano sulla stessa corsia, mentre il resto delle volte le collisioni avvengono a causa del restringimento della carreggiata, che provoca un'invasione di corsia da parte dei veicoli provenienti dalla direzione opposta. È raro che si verifichino collisioni agli incroci. La collisione con altri veicoli comporta un blocco del veicolo per un periodo di 4 minuti, dopodiché l'ego-veicolo viene rilasciato per continuare il percorso. Tuttavia, viene applicata una sanzione che si traduce in eventi di Timeout di scenario.

1.6.2 RESTRINGIMENTO DELLA CARREGGIATA

Il problema del restringimento della carreggiata provoca un aumento delle collisioni con i veicoli provenienti dalla corsia opposta e comporta una significativa diminuzione dello score (riduzione del punteggio per collisione con altri veicoli di 0.6), spesso seguiti da eventi di Timeout di scenario (riduzione del punteggio dello scenario di 0.7).

1.6.3 COLLISIONI CON ELEMENTI STATICI

Tra tutte le infrazioni commesse, un numero significativo è causato dalle collisioni con elementi statici. Pur essendo presenti solamente due traffic warning all'interno di tutti gli scenari, risultano 6 collisioni con oggetti statici. Questo perché, nel colpire i traffic warning, l'ego-veicolo sposta il traffic warning, arrivando a colpire i coni dietro quest'ultimo. La penalità assegnata a questo tipo di infrazioni è leggermente inferiore rispetto a quella assegnata alle collisioni con altri veicoli, ma comunque rilevante nel calcolo del punteggio finale, poiché uno scontro con un traffic warning comporta una penalità sia per la collisione con un elemento statico (riduzione del punteggio di 0.65) che per il fatto che l'auto non riesce a proseguire (riduzione del punteggio dello scenario di 0.7).

1.6.4 IGNORARE I SEGNALI DI STOP

Un'altra infrazione che ha un impatto significativo sul calcolo del punteggio finale è quella relativa ai segnali di stop. Sebbene sia considerata una violazione meno grave rispetto ad altre (riduzione del punteggio di 0.8), si verifica con una frequenza elevata. Pertanto, è importante gestire adeguatamente questa situazione.

1.6.5 GESTIONE DEGLI INCROCI

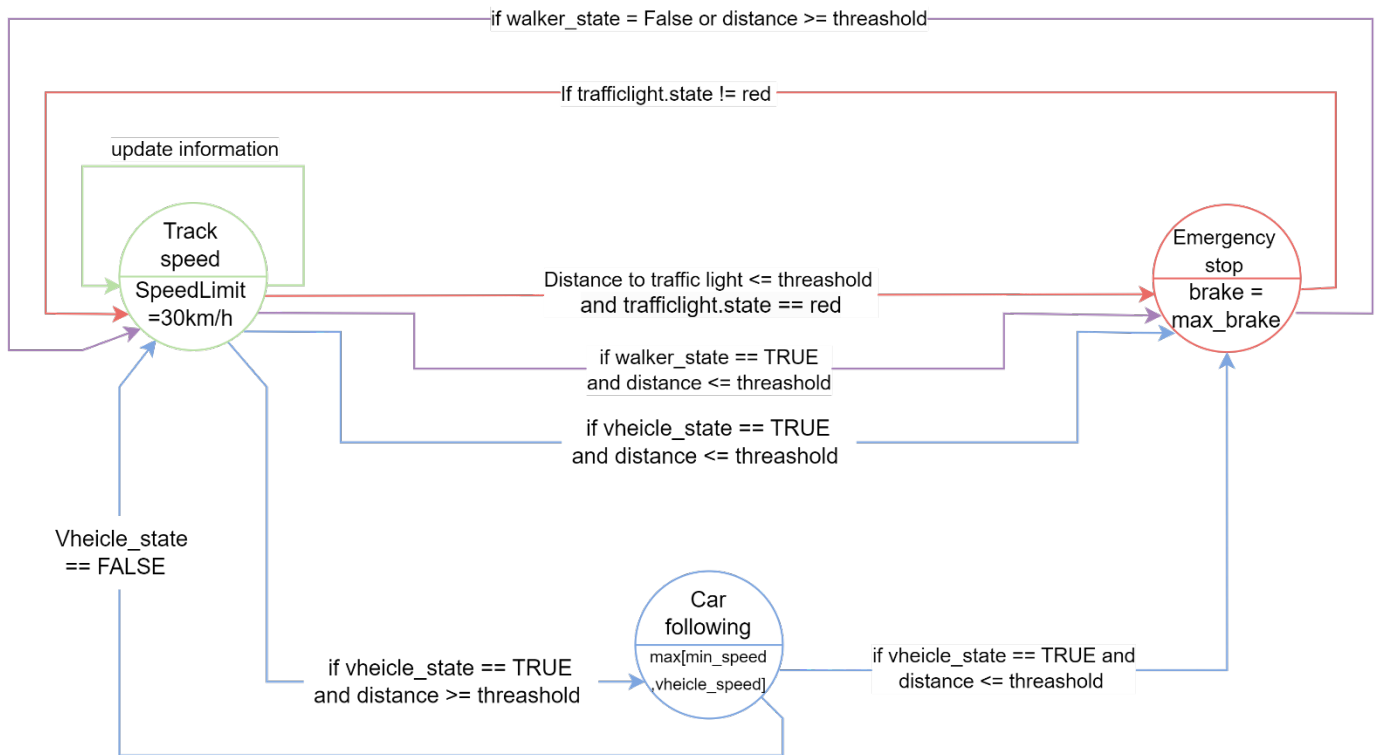
La mancata gestione delle intersezioni causa inevitabilmente una collisione con gli altri veicoli che occupano l'incrocio (riduzione del punteggio per collisione con altri veicoli di 0.6), e conseguentemente può far scattare un evento di timeout dello scenario (riduzione del punteggio dello scenario di 0.7).

1.6.6 GUIDA FUORI STRADA

Oltre alle infrazioni menzionate in precedenza, c'è un'ulteriore infrazione che non ha coefficiente, e invece influisce sul calcolo del completamento del percorso (route completion). Se un agente guida fuoristrada, la percentuale di percorso effettuata in tale modalità non verrà considerata ai fini del calcolo del punteggio di completamento del percorso. Nel codice fornito dalla baseline, si verifica un solo evento di guida fuori strada, nella route 0.

1.7 BEHAVIOUR PLANNER DELLA BASELINE

Nel **file behaviour agent** della baseline sono presenti l'insieme di regole necessarie per risolvere la selezione del comportamento appropriato e che possiamo rappresentare attraverso un automa a stati finiti.



Ad ogni step di navigazione, vengono monitorate una serie di condizioni di transizione, che se soddisfatte, permettono di passare allo stato successivo. In particolare, nell'implementazione della baseline l'automa può trovarsi nei seguenti tre stati:

- Il **"Track speed"** è lo stato che gestisce il comportamento normale del nostro veicolo, determinando la velocità che il veicolo deve seguire. Questa velocità è calcolata come il minimo tra due valori: la massima velocità consentita in base alla tipologia di guida selezionata e la differenza tra il limite di velocità della strada e un margine specifico.
 - Per la tipologia di guida "cauta", il limite di velocità imposto è di 40 km/h e il margine è di 6 metri.
 - Per la tipologia di guida "normale", il limite di velocità imposto è di 50 km/h e il margine è di 3 metri.
 - Per la tipologia di guida "aggressiva", il limite di velocità imposto è di 70 km/h e il margine è di 1 metro.
- L' **"Emergency stop"** è lo stato che gestisce il comportamento del veicolo in relazione ai semafori, all'evitamento dei pedoni e all'inseguimento dell'auto che precede. Per quanto riguarda i semafori, le condizioni di transizione dal comportamento normale allo stato di arresto controllano lo stato del semaforo che si trova entro una distanza di soglia dalla nostra corsia attuale. Per quanto riguarda l'evitamento dei pedoni e dei veicoli che precedono il nostro, le condizioni di transizione dal comportamento normale allo stato di frenata di emergenza verificano se un pedone o un veicolo (a seconda dell'attore identificato) si trova entro una distanza di soglia nella nostra corsia attuale.
- Lo stato di **"Car following"** si occupa della gestione del comportamento di accodamento del veicolo, nel caso in cui rilevi la presenza di un veicolo che lo precede ad una distanza superiore a una certa soglia. Inoltre, se il veicolo si avvicina troppo al veicolo che lo precede, il sistema attiva le condizioni per passare allo stato di frenata di emergenza.

In pratica, quando il veicolo si trova ad una distanza sufficientemente lontana dal veicolo che lo precede, lo stato di "Car following" consente al veicolo di adattare la sua velocità in modo da seguire il veicolo davanti a una distanza di sicurezza predefinita (in base al comportamento scelto).

Tuttavia, se la distanza tra il veicolo e quello che lo precede diventa inferiore a una soglia critica, il sistema attiva le condizioni per passare immediatamente allo stato di frenata di emergenza, al fine di evitare una possibile collisione.

- Per quanto riguarda la **gestione delle intersezioni**, il comportamento viene gestito nella baseline in modo simile allo stato che gestisce il comportamento normale del veicolo; pertanto, non è stato riportato nella figura dell'automa presentata sopra. Tuttavia, vi sono alcune differenze nella determinazione della velocità da seguire.

Quando il veicolo si avvicina a un'intersezione e questa viene rilevata tramite i waypoint, viene calcolata la velocità che il veicolo deve seguire come il minimo tra la massima velocità consentita in base alla tipologia di guida selezionata e la differenza tra il limite di velocità della strada e un valore specifico, impostato a 5. In altre parole, la velocità di attraversamento dell'intersezione viene regolata tenendo conto del limite di velocità della strada, riducendo la velocità di 5km/h rispetto a tale limite.

2 ARCHITETTURA SOFTWARE

In questo capitolo ci focalizzeremo sull'architettura software adottata per la nostra soluzione, nonché sui dettagli implementativi dei moduli che compongono il sistema. L'obiettivo principale di questo capitolo è fornire una visione approfondita della struttura del software, svelando i componenti chiave e le interazioni tra di essi.

Infine, concluderemo il capitolo con una panoramica generale sull'architettura software nel suo complesso, evidenziando i punti di forza e le possibili aree di miglioramento.

2.1 TARATURA DEI CONTROLLORI

Il controllo della velocità longitudinale e il controllo laterale svolgono un ruolo fondamentale nelle prestazioni complessive del veicolo su strada e sono una componente essenziale per la guida autonoma. Al fine di ottenere un feedback tempestivo dell'auto e garantire un controllo preciso, è necessario introdurre dei controllori in grado di regolare la velocità (accelerazione e freno) e la sterzata del veicolo. Per il controllo laterale, viene utilizzato un controllore PID (Proporzionale-Integrale-Derivativo), mentre per il controllo longitudinale viene impiegato il controllore di Stanley.

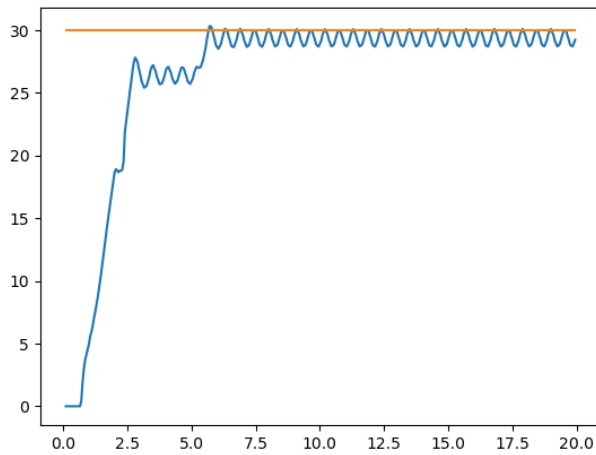
2.1.1 CONTROLLO LONGITUDINALE

Questo controllo riguarda la capacità del veicolo di gestire il movimento in direzione longitudinale, ovvero in avanti o all'indietro lungo l'asse della strada. I valori di K_p , K_I e K_D utilizzati per la taratura del controllore PID sono stati determinati attraverso un processo iterativo che ha coinvolto l'analisi delle risposte del sistema e l'ottimizzazione dei parametri al fine di raggiungere le prestazioni desiderate. Di seguito sono riportati i valori ottenuti dopo tale processo di taratura.

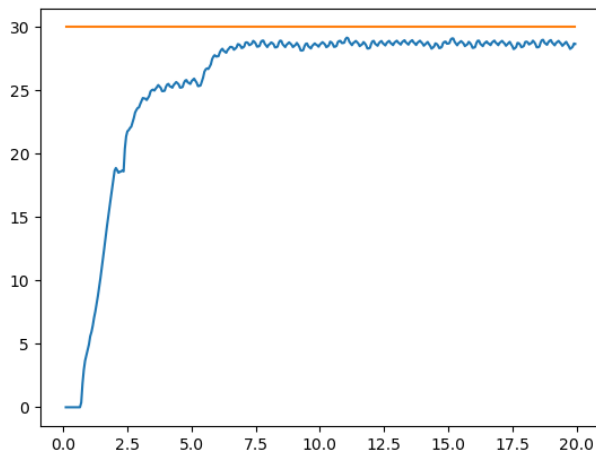
K_p	K_I	K_D	dt
0.75	0.02	0.4	0.05

Essendo la simulazione di tipo fixed time-step con valore fixed_delta_seconds pari a 0.05s, ovvero 20 fps, si è deciso di impostare il dt uguale a tale valore; quindi, abbiamo scelto $dt = \frac{1}{20fps} = 0.05$.

Inizialmente, abbiamo adottato un controllore puramente proporzionale, concentrandoci sul valore di K_p per fornire al veicolo la giusta accelerazione. Abbiamo variato il valore di K_p cercando di ottenere un tempo di salita basso e un sistema oscillante, seguendo le indicazioni della taratura di Ziegler-Nichols. Di seguito riportiamo il grafico che rappresenta l'andamento utilizzando la sola azione proporzionale.



Dall'analisi del grafico, si osserva che il tempo di salita è di circa 2,5 secondi, tuttavia si verificano sia overshoot sia uno steady state error. Successivamente, sono stati introdotti i termini integrali e derivativi per migliorare ulteriormente le prestazioni del sistema. Durante il processo di taratura, abbiamo notato che aumentando il valore di K_I si riduceva lo steady state error, ma si verificava un aumento dell'overshoot. Per risolvere questa problematica, è stato necessario tarare anche il termine derivativo, che ha permesso di bilanciare gli effetti indesiderati di K_I , migliorando anche il tempo di salita. Dopo una serie di prove, è stato raggiunto il risultato riportato nella tabella precedente. Applicando le azioni integrali e derivate, si possono notare ulteriori miglioramenti all'interno del grafico illustrato di seguito.



Il risultato di questa taratura ha evidenziato che il nostro controllore risulta essere reattivo, stabile a regime e poco oscillante.

2.1.2 CONTROLLO LATERALE

Per il controllo laterale, il nostro scopo era quello di ottenere un'azione di controllo proporzionale rispetto alla velocità e inversa tra l'entità della sterzata e la velocità a cui si sta navigando. Per farlo, abbiamo cercato di minimizzare i due errori: l'heading error e il crosstrack error.

I valori di K_V e K_S utilizzati per la taratura del controllore di Stanley, come per il controllore PID sono stati determinati attraverso un processo iterativo che ha coinvolto l'analisi delle risposte del sistema e l'ottimizzazione dei parametri al fine di raggiungere le prestazioni desiderate. Di seguito sono riportati i valori ottenuti dopo tale processo di taratura.

K_V	K_S	dt
3.0	2.0	0.05

L'errore di direzione è pari alla differenza tra la direzione del percorso e la direzione del veicolo nel punto di riferimento lungo il percorso. Con un valore troppo basso di K_V avevamo una risposta lenta del veicolo alla velocità desiderata, al contrario un valore troppo alto causava delle leggere oscillazioni.

Per quanto riguarda il guadagno di sterzata, un valore troppo basso di K_S causava una risposta lenta del veicolo all'angolo di sterzata desiderato, questo comportamento era particolarmente evidente nelle situazioni di sorpasso, in cui il veicolo doveva poi rientrare nella sua carreggiata.

Con i valori dei guadagni riportati nella tabella sopra, siamo riusciti ad ottenere un controllo più reattivo e, nel complesso, più stabile.

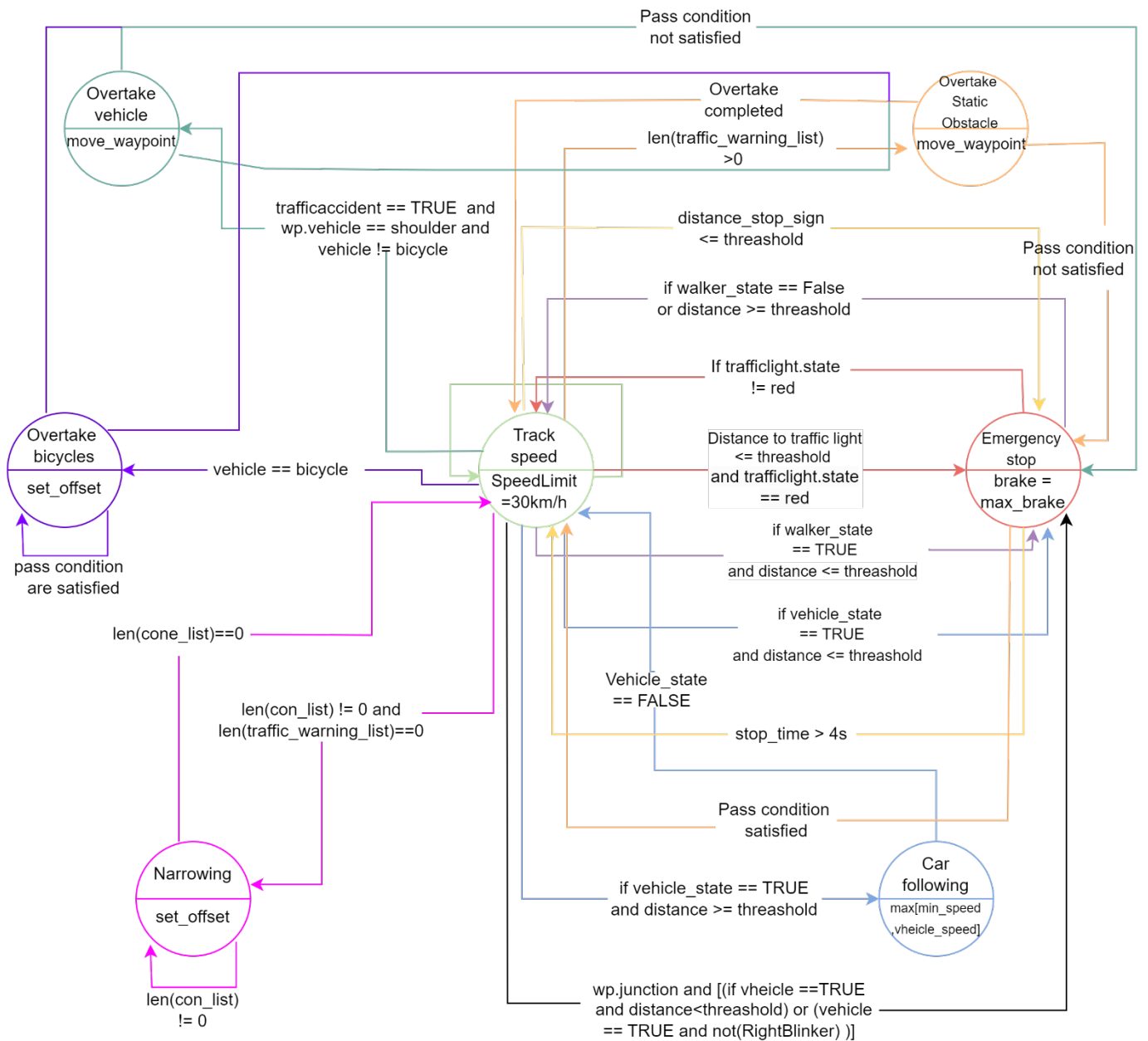
Infine, all'interno del controllore di Stanley è stato introdotto un nuovo parametro denominato "offset" per l'errore di crosstrack. Questo parametro consente di regolare la distanza tra il veicolo principale e il centro della carreggiata.

Modificando il valore dell'offset, è possibile regolare la distanza desiderata del veicolo principale rispetto al centro della carreggiata. Un valore positivo dell'offset sposterà il veicolo verso destra, mentre un valore negativo lo sposterà verso sinistra.

2.2 BEHAVIOUR PLANNER DELLA ENDLINE

Nel paragrafo 1.7 del capitolo 1 abbiamo cercato di concettualizzare il comportamento del behaviour planner della baseline fornitaci attraverso un automa a stati finiti. In particolare, l'automa era pensato per gestire solo determinate situazioni. Rispetto all'automa a stati finiti della baseline iniziale, la nostra soluzione introduce nuove fasi di gestione, come l'"overtake vehicle", l'"overtake static obstacle" e l'"overtake bicycles", per affrontare situazioni specifiche come sorpasso di veicoli fermi lungo la carreggiata, ciclisti o oggetti statici. Inoltre, la gestione delle intersezioni è stata affinata, seppur la sua gestione è ancora incompleta.

Di seguito viene proposto un confronto tra l'automa a stati finiti di partenza iniziale e quello ottenuto con la nostra soluzione. I dettagli implementativi dei singoli moduli verranno spiegati in modo approfondito nei paragrafi successivi.



Ad ogni step di navigazione, oltre ad effettuare i controlli per gli stati già implementati nell'automa di partenza, il nostro automa gestisce i seguenti nuovi stati:

- Lo stato “**Overtake static obstacle**” permette il sorpasso dei ‘traffic_warning’ presenti lungo il percorso. Le condizioni di transizione dal comportamento normale allo stato di sorpasso controllano se è presente un ostacolo entro una distanza di soglia (30m) nella nostra corsia e, se le condizioni di sorpasso sono rispettate, sposta la traiettoria dei waypoint in modo tale da permettere la manovra di sorpasso. Qualora fosse presente un ostacolo entro quella distanza di soglia ma le condizioni di sorpasso non sono rispettate, quando il veicolo raggiunge i 12 metri di distanza dall’ostacolo deve arrestarsi, passando nello stato di emergency stop.
- Lo stato “**Overtake vehicle**” consente di superare i veicoli incidentati o fermi lungo la carreggiata. Questo stato viene raggiunto quando viene rilevato un cartello di segnalazione che indica la presenza di un veicolo fermo lungo la strada prima dell’incidente. Se le condizioni di sorpasso sono verificate e viene rilevato il cartello di segnalazione, il veicolo può procedere con il sorpasso spostando la traiettoria dei waypoint. Tuttavia, se le condizioni per il sorpasso non sono verificate, quando il veicolo si avvicina a una distanza di 3 metri dall’ostacolo, viene attivato lo stato di “emergency stop” (arresto di emergenza). Questo significa che il veicolo deve fermarsi in modo sicuro per evitare la collisione con l’ostacolo.
- Lo stato “**Overtake bicycles**” permette il sorpasso dei ciclisti. Le condizioni di transizione dal comportamento normale allo stato di sorpasso controllano se è presente un ciclista entro una distanza di soglia (60 metri) nella nostra corsia e, se le condizioni di sorpasso sono rispettate, modifica la distanza dal centro di corsia, in modo tale da permettere la manovra di sorpasso.
- Lo stato “**Narrowing**” permette di gestire il restringimento della carreggiata a causa della presenza di coni. Se questi ultimi sono presenti, infatti, si passa dallo stato di comportamento normale allo stato di gestione del restringimento, nel quale la distanza dell’ego-veicolo dal centro della corsia viene modificata per evitare collisioni con auto provenienti dalla corsia opposta. Quando la lista di coni che si trovano entro una distanza di soglia sulla corsia opposta è vuota, allora vuol dire che è possibile uscire da quello stato.
- Per quanto riguarda la **gestione degli incroci**, quando il veicolo si avvicina a un’intersezione e questa viene rilevata tramite i waypoint, viene eseguito un controllo per identificare la presenza di un ostacolo entro una distanza di soglia davanti al veicolo. Se è presente un veicolo entro una certa distanza di soglia e non indica l’intenzione di svoltare a destra (riconoscibile tramite una freccia), passiamo allo stato di frenata di emergenza.

2.3 FUNZIONE OVERTAKE_VEHICLES

La funzione denominata “overtake_vehicles” viene richiamata dalla funzione “collision_and_car_avoid_manager” e si occupa di gestire il comportamento relativo al sorpasso di veicoli incidentati. Tale funzione accetta come parametri la distanza dal primo veicolo fermo che verrà incontrato, la distanza dall’ultimo veicolo fermo, il numero di veicoli fermi da superare e la posizione del nostro veicolo.

Inizialmente, vengono eseguite operazioni di filtraggio e ordinamento per selezionare tutti i veicoli situati a una distanza di 150 metri dal veicolo principale che non abbiamo *lane_type = Shoulder* (così da non considerare i veicoli incidentati).

Successivamente, attraverso l’utilizzo della funzione “vehicle_obstacle_detected” e impostando il parametro “lane_offset” a -2, è possibile identificare il veicolo più prossimo posizionato nella corsia opposta. Per filtrare il veicolo da superare, consideriamo anche la sua posizione rispetto al nostro veicolo. Lo rileviamo solo se si trova entro un campo visivo di 45 gradi in avanti rispetto a noi. In questo modo, se il veicolo si trova di fianco, possiamo considerarlo come assente poiché al termine delle valutazioni per iniziare il sorpasso non rappresenterà più un rischio di collisione con il nostro veicolo.

```

582 # Simplified version for outside junctions
583 if not ego_wpt.is_junction or not target_wpt.is_junction:
584     if target_wpt.road_id != ego_wpt.road_id or target_wpt.lane_id != ego_wpt.lane_id + lane_offset:
585         next_wpt = self._local_planner.get_incoming_waypoint_and_direction(steps=3)[0]
586         if str(target_wpt.lane_type) == "Shoulder":
587             #print("Actor: " + str(target_vehicle) + " Target waypoint: " + str(target_wpt.lane_type))
588             pass
589         elif not next_wpt:
590             continue
591         elif target_wpt.road_id != next_wpt.road_id or target_wpt.lane_id != next_wpt.lane_id + lane_offset:
592             continue
593
594 target_forward_vector = target_transform.get_forward_vector()
595 target_extent = target_vehicle.bounding_box.extent.x
596 target_rear_transform = target_transform
597 target_rear_transform.location -= carla.Location(
598     x=target_extent * target_forward_vector.x,
599     y=target_extent * target_forward_vector.y,
600 )

```

La foto mostra un estratto della funzione "vehicle_obstacle_detected" che gestisce quando una macchina viene considerata o meno. Nel caso in cui ci interessi rilevare un veicolo proveniente dalla corsia opposta, vogliamo fare in modo che l'istruzione "if" a riga 584 sia falsa. Per farlo, è necessario avere due condizioni false. La prima condizione è sempre non verificata in quanto i veicoli si trovano sulla stessa strada. Per far sì che la seconda condizione sia falsa, è necessario che i due "lane_id" - uno relativo al veicolo principale e l'altro relativo alla macchina nell'altra corsia - siano uguali. Applicando un "lane_offset" pari a -2, ipotizzando che il "lane_id" del veicolo target sia -1 e che il "lane_id" del veicolo principale sia 1, otteniamo che:

$$-1 \neq 1 - 2 \rightarrow \text{False}$$

Di conseguenza, le due condizioni saranno false e si procederà alla fase di verifica per determinare se il veicolo si trova nel range visivo da considerare.

Nella funzione, viene definita una variabile chiamata "look_ahead_distance" che viene inizializzata diversamente in base al numero di veicoli incidentati:

```

# Modify the look ahead distance based on the number of vehicles
if num_of_vehicle > 1:
    look_ahead_distance = 35 * num_of_vehicle
else:
    look_ahead_distance = 1.5 * self._speed_limit

```

Nel caso in cui siano presenti più veicoli da sorpassare, abbiamo condotto degli esperimenti e abbiamo verificato che guardare ad una distanza pari a 35 metri per il numero di veicoli ci fornisce un range di visibilità sufficiente. Durante questi test, abbiamo raggiunto tale compromesso per evitare di guardare troppo lontano, in quanto in molti sorpassi capita che il simulatore generi nuovi veicoli ad una distanza tale da rendere impraticabile il sorpasso.

Nel caso in cui ci sia un solo veicolo da superare, abbiamo osservato che è sufficiente utilizzare una "look_ahead_distance" leggermente superiore al limite di velocità della strada per effettuare un sorpasso in modo sicuro, evitando collisioni con i veicoli provenienti dall'altra corsia. Abbiamo quindi deciso di impostare il valore di questa distanza a 1,5 volte il limite di velocità.

Successivamente, vengono effettuati controlli per verificare se nella corsia accanto non ci sono veicoli, o se ci sono veicoli ma sono ad una distanza superiore alla "look_ahead_distance". Inoltre, viene controllato se la distanza dal veicolo di fronte è minore di un quarto del limite di velocità (speed_limit/4) e se non stiamo già effettuando un altro sorpasso.

Se tutte queste condizioni sono verificate, spostiamo il nostro veicolo nella corsia di sinistra spostando un numero di punti di riferimento (waypoint) pari alla distanza tra il nostro veicolo e il veicolo incidentato più distante. Inoltre, impostiamo la variabile "overtake_counter" a 200. Questo significa che nelle prossime 200 iterazioni del ciclo "run_step", non effettueremo nuovamente questa valutazione. In pratica, evitiamo di

ripetere la procedura di sorpasso per un certo numero di passi successivi per garantire una corretta esecuzione del sorpasso senza interruzioni o sovrapposizioni indesiderate.

2.4 FUNZIONE OVERTAKE_BICYCLES

La funzione denominata "overtake_bicycle" viene richiamata dalla funzione "collision and car avoid manager" e si occupa di gestire il comportamento relativo al sorpasso di gruppi di biciclette. L'obiettivo del sorpasso è quello di replicare il comportamento di una situazione reale, evitando di invadere la corsia opposta. Per fare ciò, viene impostato un offset all'interno del controllore di Stanley, consentendo di sorpassare le biciclette spostandosi sull'estremità della corsia di percorrenza.

Per sopperire alle mancanze delle leaderboard di Carla per le biciclette sono state fatte delle assunzioni.

- Lunghezza di una bicicletta = metà della lunghezza dell'attore principale.
- Larghezza di una bicicletta = $\frac{1}{4}$ della larghezza dell'attore principale.
- Un gruppo di biciclette è formato sempre da due biciclette.
- Le biciclette si trovano sempre sul lato estremo della carreggiata (importante per il sorpasso)
- La larghezza di una carreggiata standard è pari a 3 metri.

Tale funzione accetta come parametri la distanza dalla prima bicicletta che verrà incontrata, la posizione del nostro veicolo e la posizione della bicicletta.

Inizialmente, vengono eseguite operazioni di filtraggio e ordinamento per selezionare tutti i veicoli situati a una distanza di 150 metri dal veicolo principale e che non siano biciclette.

Successivamente, attraverso l'utilizzo della funzione "vehicle_obstacle_detected" e impostando il parametro "lane_offset" a -2, è possibile identificare il veicolo più prossimo posizionato nella corsia opposta. Per filtrare il veicolo da superare, consideriamo anche la sua posizione rispetto al nostro veicolo. Lo rileviamo solo se si trova entro un campo visivo di 60 gradi in avanti rispetto a noi.

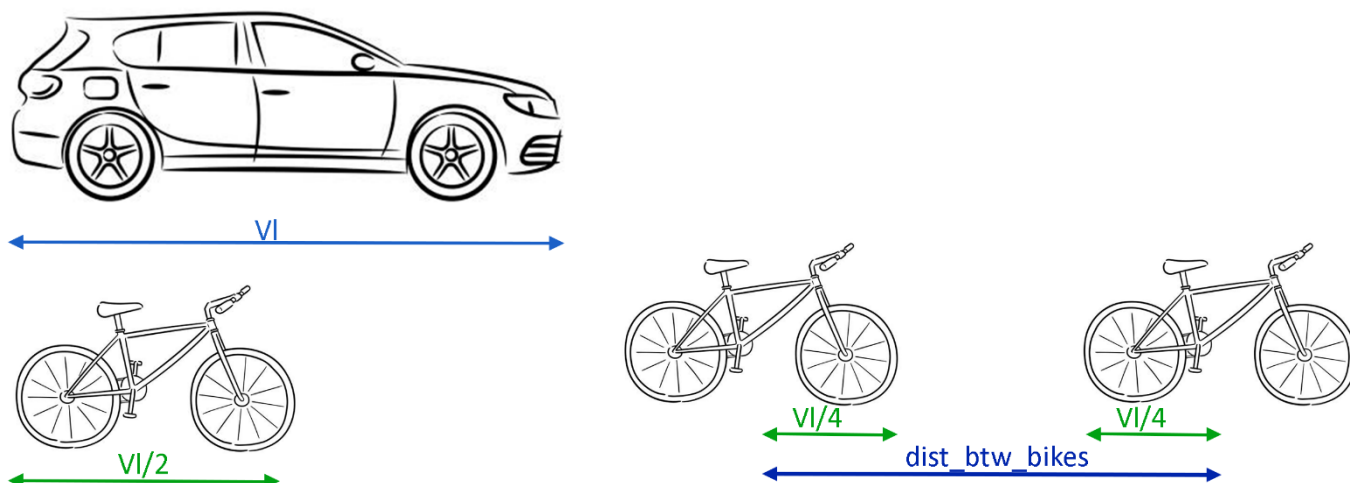
Il comportamento della funzione è suddiviso in tre fasi:

- Calcolo della distanza tra le biciclette;
- Valutazione della presenza e distanza della macchina nella corsia di sorpasso e valutazione della larghezza della carreggiata;
- Sorpasso.

Nella prima fase, viene calcolata la distanza tra le due biciclette al fine di valutare la possibilità di un rientro tra di esse per effettuare il sorpasso in due fasi ([qui](#) il video).

```
if len(bike_list) >= 2:  
    dist_bt看_bikes = compute_distance(bike_list[0].get_location(), bike_list[1].get_location()) - self._vehicle.bounding_box.extent.x/4 * 2
```

Il calcolo della distanza avviene utilizzando la funzione "compute_distance", che calcola la distanza euclidea in 3D tra due posizioni (Location). Tuttavia, poiché il calcolo della distanza tiene conto anche della parte anteriore della prima bicicletta e della parte posteriore della seconda bicicletta, è necessario sottrarre questa quantità dal calcolo della distanza effettiva.



Abbiamo deciso di inserire questa possibilità nel comportamento in quanto all'interno degli scenari non è sempre possibile sorpassare l'intero gruppo in una sola fase data la presenza di curve e veicoli che arrivano nell'altra corsia ad una distanza molto piccola tra di loro.

Nella fase successiva, vengono valutati diversi parametri per determinare i comportamenti di sorpasso appropriati:

- Se la larghezza della corsia supera la larghezza standard (3 metri), è possibile effettuare il sorpasso senza la necessità di controllare la presenza di veicoli nella carreggiata opposta. Questo perché la larghezza della corsia consente di sorpassare le biciclette senza invadere l'altra carreggiata.

```
389 # if the size of the roadway permits it, it allows overtake the bicycles
390 if bike_waypoint.lane_width > 3 and distance_front_vehicle < 8:
391     self.behavior.overtake_counter = 80
392     self.lateral_controller.set_offset(-self.lane_width/2 + 0.3)
393     return
```

- Se la larghezza della corsia non permette un sorpasso diretto, viene verificata la distanza tra le biciclette per determinare se è possibile tornare alla corsia di origine dopo aver superato la prima bicicletta. In tal caso, vengono impostati i parametri "look_ahead_distance" e "overtake_counter" su valori inferiori rispetto alla situazione opposta.

```
397 # change the overtaking parameters allowing the return if the distance between the
398 # bicycles permits it
399 if dist_bt看 bikes > self.vehicle.bounding_box.extent.x * 3 and self._speed != 0:
400     counter = 50
401     print("SPEED LIMIT: " + str(self._speed_limit))
402     look_ahead_distance = self._speed_limit + distance_front_vehicle
403 elif self._speed != 0:
404     counter = 80
405     print("SPEED LIMIT: " + str(self._speed_limit))
406     look_ahead_distance = 1.3 * self._speed_limit + dist_far_bycycle
```

- I valori di questi due parametri ("look_ahead_distance" e "counter") sono stati determinati mediante una serie di esperimenti. Per la "look_ahead_distance" è stato utilizzato il valore di "speed_limit", che rappresenta la velocità con cui i veicoli si muovono nella corsia opposta, trasformata in una distanza di riferimento. La differenza tra i due casi sta nel fatto che nel primo caso viene aggiunta la distanza tra il veicolo principale e la bicicletta di fronte, mentre nel secondo caso viene aggiunta la distanza tra il veicolo principale e la bicicletta più lontana. Per quanto riguarda il parametro "counter", esso

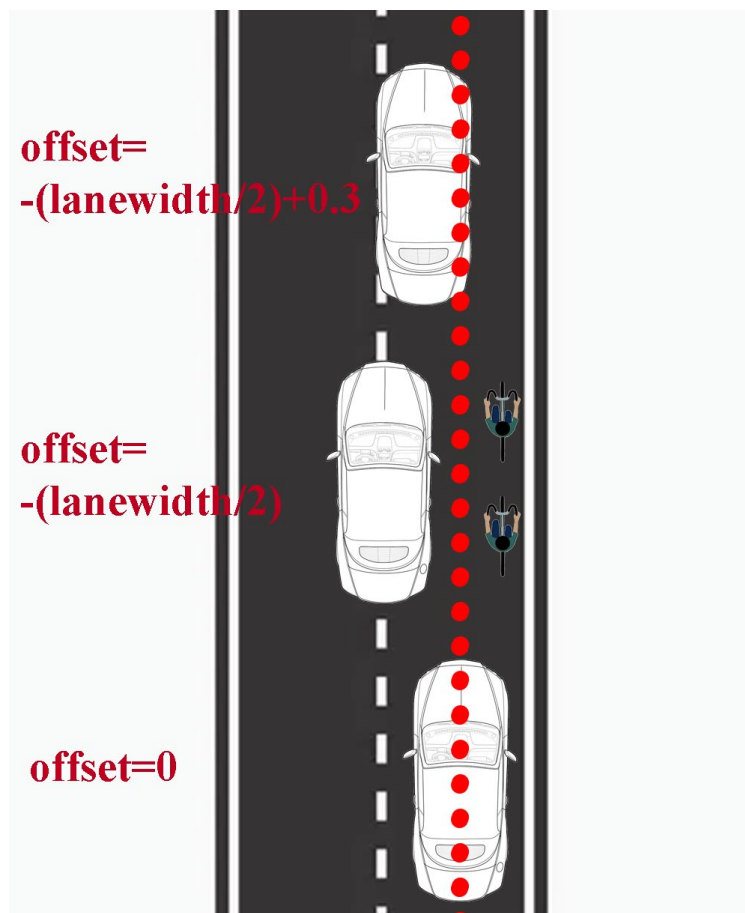
assumerà un valore maggiore nel caso in cui dovrà effettuare un sorpasso più lungo (ovvero di entrambe le biciclette).

- Si osserva che nel caso in cui il veicolo si fermi dietro alle biciclette, la distanza tra le biciclette non viene più considerata, ma viene utilizzato un valore standard di "look_ahead_distance" e "overtake_counter". Questa scelta è dovuta al fatto che le biciclette non sono sempre oggetti dinamici e una volta ferme, anche il nostro veicolo si ferma. Ciò comporta che in alcuni casi, in particolare nello scenario 1, il sorpasso non avveniva poiché le condizioni di "look_ahead_distance" non venivano soddisfatte. Pertanto, abbiamo deciso di utilizzare valori di "look_ahead_distance" e "overtake_counter" inferiori rispetto al caso dinamico.

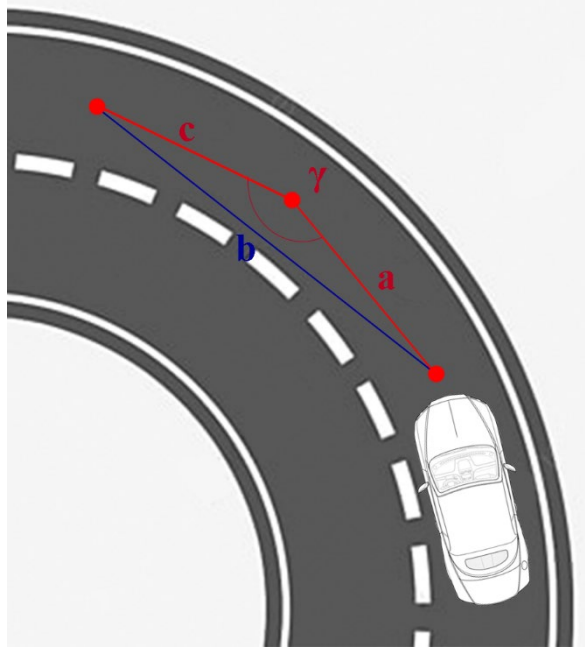
```
395 look_ahead_distance = 35 + dist_far_bycycle
396 counter = 70
```

- Successivamente, viene effettuato il sorpasso dopo aver verificato che:
 - Non ci sono veicoli nella corsia opposta o che il veicolo rilevato si trova a una distanza superiore alla "look_ahead_distance";
 - Il veicolo non si trova in una curva (ovvero l'angolo è maggiore di 173 gradi);
 - La distanza dalla prima bicicletta è minore di 8 metri;
 - Non stiamo già sorpassando un altro veicolo.

Il sorpasso viene eseguito impostando l'offset ad un valore ottenuto mediante esperimenti ed è pari a $lane_width/2 + 0.3$.



La motivazione per includere la condizione di non essere in curva è basata sulla metodologia di calcolo della distanza tra il nostro veicolo principale e l'auto nella corsia opposta, che utilizza una misura euclidea. Pertanto, nel caso in cui il veicolo principale si trovi in una curva, la distanza ottenuta potrebbe non corrispondere effettivamente alla distanza reale poiché viene calcolata in linea retta, ignorando le deviazioni del percorso. Di conseguenza, potrebbe verificarsi una discrepanza tra la distanza calcolata e quella reale, compromettendo la sicurezza nel caso di manovre di sorpasso con la possibilità di generare collisioni. Per determinare se il veicolo principale si trova in curva, applichiamo il teorema dei coseni su tre waypoint: il waypoint attuale, il quinto waypoint successivo e il decimo waypoint successivo a quello attuale.



In questo modo possiamo calcolare l'angolo che ha come vertice il waypoint centrale utilizzando la seguente formula:

$$b^2 = a^2 + c^2 - 2 * ac * \cos (\gamma)$$

dove a, b, c sono le lunghezze dei tre lati ottenuti mediante calcolo della distanza tra due punti. Da qui otteniamo:

$$\gamma = \arccos \left(\frac{a^2 + c^2 - b^2}{2ac} \right)$$

2.5 FUNZIONE OVERTAKE_STATIC_OBSTACLE

La funzione “overtake_static_obstacle” viene richiamata dalla funzione “lane narrowing and static obstacle” e si occupa di gestire il comportamento relativo al sorpasso del cartello di traffic warning. L’obiettivo della funzione è replicare il comportamento di una situazione reale, pertanto, il veicolo si ferma ad una certa distanza dal traffic warning e dopo aver effettuato gli opportuni controlli effettua il sorpasso quando ne ha occasione.

Tale funzione accetta come parametri la posizione del veicolo principale, la lista contenente gli oggetti di traffic_warning e la lista contenente i coni stradali.

Inizialmente, vengono eseguite operazioni di filtraggio e ordinamento per selezionare tutti i veicoli situati a una distanza di 100 metri dal veicolo principale.

Inizialmente la funzione crea una lista con tutti i veicoli presenti in un raggio di 100 metri e controlla la presenza di veicoli sulla corsia adiacente tramite la funzione “vehicle_obstacle_detected”. Dopodiché, il sorpasso avverrà quando:

- la distanza dell'ostacolo sarà inferiore a un quarto del limite di velocità (speed_limit/4)
- non sarà già in corso un altro sorpasso
- non ci saranno veicoli accanto a noi o saranno ad una distanza maggiore della somma tra il limite di velocità attuale della strada (self._speed_limit) e la distanza dall'ultimo cono che compone il traffic warning (dist_c).

Se tutte queste condizioni sono soddisfatte, procediamo spostando nella corsia adiacente i successivi punti di riferimento (waypoint) che l'auto dovrà seguire. Il numero di punti di riferimento da spostare sarà determinato dalla variabile “dist_c”. Una volta che l'auto terminerà i waypoint da seguire sulla corsia adiacente ritornerà nella corsia di partenza.

2.6 FUNZIONE LANE_NARROWING_AND_STATIC_OBSTACLE

La funzione denominata "lane_narrowing_and_static_obstacle" viene richiamata dalla funzione “run_step” ed è responsabile della gestione del comportamento relativo al restringimento di una carreggiata e al sorpasso dei cartelli di avvertimento del traffico che occupano la stessa carreggiata. Nel contesto di Carla, il restringimento della carreggiata viene segnalato attraverso l'utilizzo di coni stradali disposti lungo la carreggiata opposta rispetto a quella percorsa dal veicolo principale, mentre il cartello di avvertimento del traffico è posizionato al centro della carreggiata, seguito da coni stradali.

È quindi necessario distinguere tra questi due scenari, poiché la presenza di soli coni stradali indica inequivocabilmente un restringimento di carreggiata, mentre la presenza di un cartello di avvertimento del traffico richiede la gestione del sorpasso.

Tale funzione accetta come parametro la posizione del nostro veicolo.

Inizialmente, vengono eseguite operazioni di filtraggio e ordinamento per selezionare tutti i coni situati in un range di 25 metri dal veicolo principale e tutti i traffic warning situati in un range di 30 metri dal veicolo principale.

```
558 cone_list = self.sort_vehicle_list(self._world.get_actors().filter('static.prop.constructioncone'), ego_vehicle_wp, 25)
559 traffic_warning_list = self.sort_vehicle_list(self._world.get_actors().filter('static.prop.trafficwarning'), ego_vehicle_wp, 30)
```

Durante ogni iterazione, la funzione controlla se ci sono coni stradali nel suo campo visivo e, contemporaneamente, se non vi sono segnali di "traffic_accident".

Nel caso in cui si verifichi questa condizione, l'offset del controllore di Stanley viene impostato in modo tale che il veicolo si sposti sul lato destro della carreggiata, evitando così di scontrarsi con le auto provenienti dalla carreggiata opposta che invadono la sua corsia. Il valore specifico scelto per l'offset pari a $lane_width/2 - 0.25$ è stato determinato tramite esperimenti, tenendo conto anche dell'obiettivo di mantenere il veicolo sulla strada.

D'altra parte, se viene rilevato un segnale di cartello di avvertimento del traffico, la funzione richiama la funzione "overtake_static_obstacle" per procedere con il sorpasso.

2.7 FUNZIONE COMPUTE_PARAMETERS_FOR_CRASHED_VEHICLES

La funzione "compute_parameters_for_crashed_vehicles" viene richiamata dalla funzione “collision_and_car_avoid_manager” ed è responsabile del calcolo dei parametri relativi ai veicoli coinvolti in un incidente stradale.

Tale funzione accetta come parametri la lista di veicoli (filtrati ed ordinati in un range di 60 metri dal nostro veicolo) e la posizione del nostro veicolo.

All'interno della funzione, vengono inizializzate due variabili: "distance_far_vehicle" e "num_of_vehicles". Entrambe le variabili vengono inizializzate con il valore 0. Queste variabili verranno utilizzate per calcolare la distanza dal veicolo più lontano coinvolto nell'incidente e il numero totale di veicoli presenti di fronte al veicolo principale.

Successivamente, la funzione procede a individuare il numero di veicoli incidentati e la distanza dal veicolo fermo più lontano. Per distinguere i veicoli incidentati dagli altri veicoli, viene utilizzato il tipo di waypoint "Shoulder", che in questo caso indica che una macchina è ferma sulla carreggiata a seguito dell'incidente.

Infine, la funzione restituisce il numero di veicoli e la distanza dal veicolo incidentato più lontano.

2.8 FUNZIONE COLLISION_AND_CAR_AVOID_MANAGER

La funzione "collision_and_car_avoid_manager" viene richiamata dalla funzione "run_step" e svolge diverse funzioni:

- Gestisce il comportamento per tutte le situazioni di sorpasso dei veicoli.
- Gestisce il comportamento per il "tailgating".
- Restituisce l'informazione sulla presenza o assenza di un veicolo nella nostra corsia.

Tale funzione accetta come parametro la posizione del nostro veicolo.

Nella prima fase, la funzione crea due elenchi: uno contenente tutti i veicoli all'interno di un raggio di 60 metri e l'altro contenente i segnali di "traffic accident" entro un raggio di 10 metri. I segnali di "traffic accident" indicano la presenza, poco avanti, di uno o più veicoli incidentati fermi a bordo della carreggiata. Successivamente, la funzione verifica in quale corsia deve controllare la presenza di veicoli confrontando l'attributo "self._direction" con il valore di "RoadOption", che determina se la funzione deve guardare a destra, sinistra o in avanti. Se il veicolo deve proseguire lungo la propria corsia viene chiamata la funzione "vehicle_obstacle_detected" con un angolo di rilevamento di 90 gradi. A partire da questo punto, inizia la fase di controllo per valutare la possibilità di effettuare il sorpasso, che comprende diversi casi:

- Se il tipo di waypoint è "Shoulder" e il veicolo non è una bicicletta, e in precedenza è stato rilevato un segnale di "traffic accident", ci troviamo nella situazione in cui vi sono veicoli fermi sulla carreggiata. Viene chiamata la funzione "compute_paramters_for_crashed_vehicles" per calcolare quanti veicoli ci sono davanti al veicolo e anche distanza, e successivamente la funzione "overtake_vehicles" per avviare la procedura di sorpasso.
- Se viene rilevata una bicicletta, viene chiamata la funzione "overtake_bicycles" per gestire il sorpasso di tale veicolo.
- Se il tipo di waypoint è "Shoulder", ma non è stato rilevato alcun segnale di "traffic accident" in precedenza, significa che il veicolo rilevato si trova fuori dalla carreggiata e quindi deve essere ignorato.
- Se nessuno di questi casi si verifica, viene restituito semplicemente il veicolo rilevato e la distanza a cui viene rilevato.

2.9 FUNZIONE STOP_MANAGER

La funzione "stop_manager" viene chiamata dalla funzione "run_step" ed è responsabile di verificare se il nostro veicolo è influenzato dalla presenza di un segnale di stop. Per effettuare questa verifica, richiama la funzione "affected_by_stop_sign", definendo una distanza massima chiamata "max_distance" entro la quale il segnale di stop deve trovarsi per essere preso in considerazione.

Viene quindi creata una lista dei segnali di stop presenti nello scenario e, ad ogni iterazione, vengono esclusi quelli che si trovano ad una distanza superiore alla "max_distance" e quelli posizionati su strade diverse da quella in cui si trova il nostro veicolo.

Successivamente, viene effettuata una valutazione utilizzando la funzione "is_within_distance", che controlla se l'ostacolo si trova a una determinata distanza da noi ed entro l'angolo tra la posizione del nostro veicolo e quella di riferimento.

Nel caso in cui la valutazione restituisca "true", viene salvato il segnale di stop nella variabile "_last_stop_sign" (in modo da poterlo ignorare nelle iterazioni successive della funzione) e viene salvato anche il tempo di simulazione a cui ci siamo fermati nella variabile "_last_time_stop_signal".

All'inizio della funzione, viene valutato se sono trascorsi quattro secondi dal momento in cui ci siamo fermati, basandosi su "_last_time_stop_signal". In caso contrario, viene restituito il valore "true" insieme al segnale di stop, indicando che l'auto è ancora influenzata da quel segnale e deve rimanere ferma. Altrimenti, viene restituito il valore "false", indicando che l'auto non è più influenzata dal segnale di stop e può procedere.

Abbiamo definito una "max_distance" che di default ha valore 5, ovvero permette all'auto di rilevare tutti i segnali di stop in un range di 5 metri da essa.

2.10 GESTIONE INCROCI

Sono state effettuate diverse considerazioni per implementare il comportamento relativo alla gestione degli incroci:

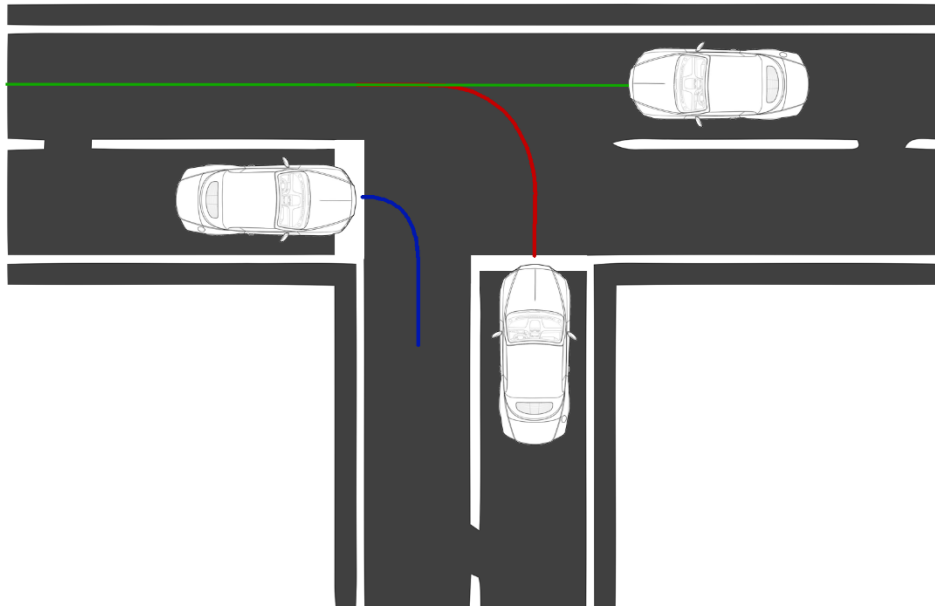
1. Attualmente, la funzione gestisce solo gli incroci a T, in cui le macchine arrivano da entrambi i lati della strada.
2. Le macchine che percorrono altre corsie segnalano le loro intenzioni di girare a destra o sinistra utilizzando le frecce direzionali del veicolo.
3. Le macchine si fermano quando devono dare la precedenza a destra, ma non si fermano quando hanno la precedenza a sinistra. Ciò significa che se il veicolo principale si ferma nel mezzo della strada, le macchine alla sua sinistra si fermano, mentre quelle alla sua destra continuano a procedere.

Inizialmente, vengono eseguite operazioni di filtraggio e ordinamento per selezionare tutti i veicoli situati a una distanza di 70 metri dal veicolo principale.

Il funzionamento dell'algoritmo si divide nei seguenti casi:

1. Rilevazione di un veicolo utilizzando la funzione "vehicle_obstacle_detected". Poiché ci troviamo in un'intersezione, la funzione non si basa sull'informazione dei waypoint per rilevare la presenza di un veicolo, ma analizza l'intersezione tra il poligono del veicolo considerato e il poligono del nostro veicolo principale. Se i due poligoni si intersecano, viene restituito il veicolo con cui si intersecano.
2. Successivamente, verifichiamo se il veicolo restituito ha attivata la freccia di svolta a destra. Se è vero, seguiamo lungo il percorso; altrimenti, ci fermiamo e aspettiamo.

3. Se la funzione non restituisce un veicolo o se la distanza a cui si trova il veicolo è maggiore della metà del limite di velocità attuale, continuiamo ad avanzare nell'intersezione.



Sarà quindi opportuno considerare queste osservazioni per una corretta gestione degli incroci nel contesto dell'algoritmo implementato.

Il comportamento della funzione non è perfetto, ma nella maggior parte dei casi ci consente di completare l'intersezione senza problemi. Questo è possibile grazie al fatto che ci sono macchine a sinistra che utilizzano la freccia a destra per indicare la loro intenzione di girare, e noi sfruttiamo queste informazioni per procedere lungo l'intersezione. Tuttavia, se notiamo la presenza di macchine che arrivano dalla destra del veicolo principale, le quali ricordiamo non si fermano, ci fermiamo. Per poter avanzare nell'incrocio teniamo conto che le auto provenienti dalla sinistra dell'ego-veicolo gli daranno la precedenza se questo ha già impegnato l'incrocio, mentre non accadrà lo stesso per i veicoli provenienti dalla destra. Pertanto, aspettiamo fino a quando non abbiamo l'opportunità di proseguire l'intersezione.

È importante notare che, sebbene questa strategia permetta di gestire alcune situazioni nell'intersezione, potrebbero ancora verificarsi casi in cui sia necessaria una valutazione più complessa o ulteriori considerazioni per garantire una corretta e sicura navigazione dell'intersezione stessa.

3 CAPITOLO 3

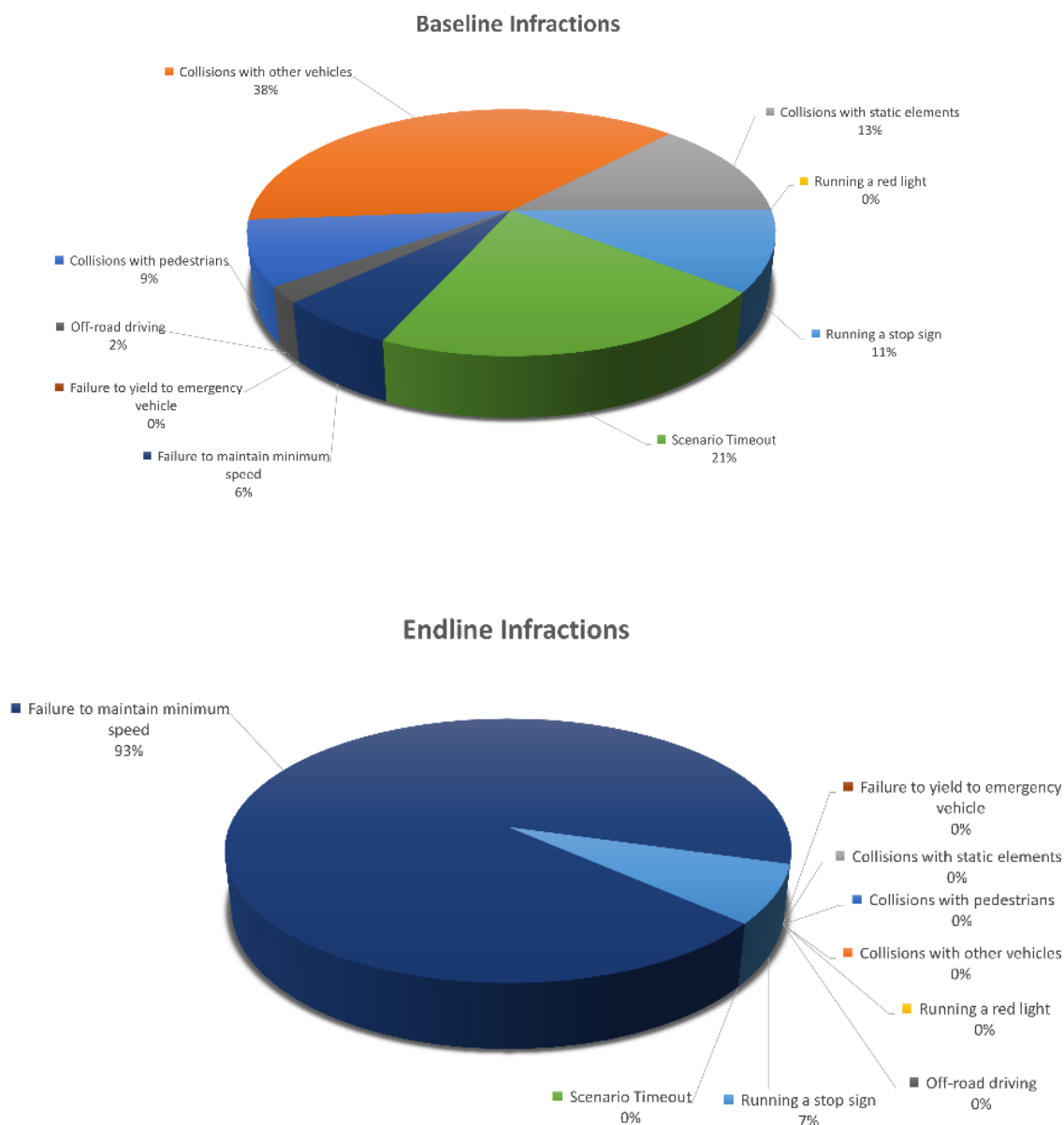
3.1 RISULTATI OTTENUTI

Di seguito riportiamo, oltre allo score globale, i punteggi migliori ottenuti per le singole route.

	Route0	Route1	Route2	Route3	Route4	Totale
EVALUATION METRICS						
DRIVING SCORE	85.27	83.99	73.38	100	79.82	84.49
ROUTE COMPLETION	100%	100%	100%	100%	100%	100%
INFRACTION PENALTY	0.85	0.83	0.73	1	0.79	0.84
INFRACTIONS						
NUM INFRACTIONS	4	5	3	0	2	14
Collisions with pedestrians (0.50)	X	X	X	X	X	X
Collisions with other vehicles (0.60)	X	X	X	X	X	X
Collisions with static elements (0.65)	X	X	X	X	X	X
Running a red light (0.70)	X	X	X	X	X	X
Running a stop sign (0.80)	X	X	X	X	1	1
Scenario Timeout (0.70)	X	X	X	X	X	X
Failure to maintain minimum speed (0.70)	4	5	3	X	1	13
Failure to yield to emergency vehicle (0.70)	X	X	X	X	X	X
Off-road driving (0.0)	X	X	X	X	X	X
SHUTDOWN EVENTS						
Route Deviation	X	X	X	X	X	X
Agent Blocked	X	X	X	X	X	X
Simulation Timeout	X	X	X	X	X	X
Route Timeout	X	X	X	X	X	X

Questi risultati rappresentano il punteggio più alto ottenuto durante la fase dei test. Confrontando i risultati ottenuti con quelli iniziali della baseline, notiamo un netto miglioramento del numero di infrazioni totali, che dalle 48 iniziali sono diminuite a 14. Di queste 14 infrazioni, 13 sono state causate dall'incapacità di mantenere una velocità minima. Tuttavia, queste infrazioni sono inevitabili in situazioni di sorpasso, in cui l'auto deve aspettare il momento opportuno per eseguire la manovra e quindi rallentare, violando così il limite di velocità minima. Abbiamo categorizzato l'infrazione riguardante la mancata frenata ad un segnale di stop come un comportamento anomalo e abbiamo fornito una spiegazione dettagliata del motivo nel punto 2 del paragrafo 3.3.1.

Per fornire una panoramica immediata dei miglioramenti apportati, presentiamo un confronto tra il diagramma delle infrazioni commesse con la baseline e il diagramma delle infrazioni ottenute con la endline. In questo modo sarà possibile visualizzare chiaramente i cambiamenti e i progressi realizzati nell'ottimizzazione delle prestazioni.



3.2 TEST EFFETTUATI

La seguente tabella mostra i risultati ottenuti durante la fase di test, in cui abbiamo eseguito l'algoritmo 5 volte per ogni scenario. Ogni riga della tabella rappresenta il risultato ottenuto durante una simulazione. Nella tabella delle infrazioni e degli "Shutdown Event", nel caso in cui non ci siano state infrazioni viene segnata una **X**.

	Route 0	Route 1	Route 2	Route 3	Route 4
EVALUATION METRICS					
Driving Score	48.99	79.31	74.79	100.00	24.00
	83.13	76.90	27.78	60.00	79.82
	78.63	40.16	73.38	41.75	29.95
	49.53	82.17	47.51	60.00	29.95
	10.28	84.00	60.00	70.00	47.89
	100%	100%	100%	100%	100%

Route Completion	100%	100%	77.90%	100%	100%
	100%	100%	100%	100%	100%
	100%	100%	100%	100%	100%
	100%	100%	100%	100%	100%
Infraction Penalty	0.48	0.79	0.74	1.00	0.24
	0.83	0.76	0.35	0.60	0.79
	0.78	0.40	0.73	0.41	0.29
	0.49	0.82	0.47	0.60	0.29
	0.10	0.84	0.60	0.70	0.47
INFRACTIONS					
NUM INFRACTIONS	5	5	3	0	4
	4	5	4	1	2
	4	6	3	4	3
	5	5	3	1	3
	8	5	1	3	4
Collisions with pedestrians (0.50)	X	X	X	X	1
	X	X	X	X	X
	X	X	X	X	1
	X	X	X	X	1
	X	X	X	X	1
Collisions with other vehicles (0.60)	1	X	X	X	1
	X	X	2	1	X
	X	X	X	X	1
	1	X	1	1	1
	4(sempré la stessa)	X	1	X	1
Collisions with static elements (0.65)	X	X	X	X	X
	X	X	X	X	X
	X	X	X	X	X
	X	X	X	X	X
	X	X	X	X	X
Running a red light (0.70)	X	X	X	X	X
	X	X	X	X	X
	X	X	X	1	X
	X	X	X	X	X
	X	X	X	1	X
Running a stop sign (0.80)	X	X	X	X	1
	X	X	X	X	1
	X	X	X	X	X
	X	X	1	X	X
	X	X	X	X	1
Scenario Timeout (0.70)	X	X	X	X	X
	X	X	X	X	X
	X	X	X	X	X
	X	X	X	X	X
	X	X	X	X	X
Failure to maintain minimum speed (0.70)	4	5	3	X	X
	4	5	X	X	1
	4	6	3	2	1
	4	5	1	X	1

	4	5	X	2	1
Failure to yield to emergency vehicle (0.70)	X	X	X	X	X
	X	X	X	X	X
	X	X	X	X	X
	X	X	X	X	X
	X	X	X	X	X
Off-road Driving (0.0)	X	X	X	X	X
	X	X	1	X	X
	X	X	X	X	X
	X	X	X	X	X
	X	X	X	X	X
SHUTDOWN EVENT					
Route Deviation	X	X	X	X	X
	X	X	X	X	X
	X	X	X	X	X
	X	X	X	X	X
	X	X	X	X	X
Agent Blocked	X	X	X	X	X
	X	X	1	X	X
	X	X	X	X	X
	X	X	X	X	X
	X	X	X	X	X
Simulation Timeout	X	X	X	X	X
	X	X	X	X	X
	X	X	X	X	X
	X	X	X	X	X
	X	X	X	X	X
Route Timeout	X	X	X	X	X
	X	X	X	X	X
	X	X	X	X	X
	X	X	X	X	X
	X	X	X	X	X

Dalla tabella dei test, si evince che il nostro agente fallisce nel completare il percorso in una prova su 25(5 test per 5 scenari). Per quanto riguarda le infrazioni commesse, si può notare che:

- Nel caso specifico dello scenario 4, il nostro agente è stato coinvolto in quattro collisioni con pedoni. È importante notare che queste collisioni non sono state causate direttamente dal nostro veicolo, ma sono state il risultato di un altro veicolo con cui il pedone è entrato in collisione, facendolo rimbalzare sull'ego-veicolo. Questo spiega il motivo per cui si sono verificate ripetute collisioni con il pedone nello stesso scenario ([qui](#) il video).
- Nel caso delle collisioni con altri veicoli, su 25 simulazioni, si sono verificate 13 collisioni con altri veicoli. È importante notare che le 4 collisioni nello scenario 0 sono state considerate come un'unica collisione. Questo perché, quando le due auto si scontrano in un incrocio, la nostra auto tende a trascinare l'altra per un breve tratto di strada. Anche in questo caso i risultati non sorprendono, poiché la maggior parte delle collisioni sono avvenute nelle intersezioni, le quali sono ancora imperfette e la cui gestione non è soddisfacente. Alcune di queste collisioni si verificano anche durante il sorpasso di veicoli fermi lungo la carreggiata, In particolare, in una situazione in cui abbiamo tre veicoli fermi e l'auto sull'altra carreggiata viene generata dopo che il nostro veicolo ha già iniziato la manovra di

sorpasso. Questo caso è stato incluso tra i casi anomali, i quali vengono discussi nel dettaglio nel paragrafo 3.3.1.

- Nel caso di collisioni con elementi statici, il nostro agente risulta robusto rispetto a questo tipo di infrazione, al punto da non accumulare nessuna penalità per tutte e 25 le simulazioni.
- Il nostro agente ha commesso infrazioni relative al mancato rispetto di semafori o segnali di stop in diversi scenari. Durante le 25 simulazioni totali, l'agente ha commesso due volte l'infrazione di non rispettare uno stop, nello scenario 3. Inoltre, l'agente ha commesso l'infrazione di non rispettare uno stop per quattro volte su 25, negli scenari 2 e 4.
- Durante le 25 simulazioni, non sono stati registrati eventi di timeout di scenario nonostante le collisioni con altri veicoli. Questo può essere attribuito al fatto che la maggior parte delle collisioni si è verificata nelle intersezioni, dove entrambi i veicoli erano in movimento. Di conseguenza, non si è creata una situazione di blocco come accadeva nella baseline, in cui gli eventi di timeout erano spesso causati da collisioni con auto ferme lungo la carreggiata.
- L'infrazione di guida fuori strada, su 25 simulazioni totali si verifica una sola volta nello scenario 2 a causa di una collisione all'incrocio. Questo evento blocca la simulazione, generando anche uno "Shutdown event", ossia l'evento di "Agent Blocked".

3.3 ANALISI DEI RISULTATI

I risultati ottenuti sono, nel complesso, soddisfacenti, seppur con margine di miglioramento. Dato il tempo e l'hardware a disposizione limitato, abbiamo dovuto effettuare scelte mirate per ottenere il miglior compromesso tra la risoluzione dei problemi presenti nella baseline e un punteggio elevato. Nelle sezioni seguenti, ci concentreremo sull'analisi della endline, considerando anche i vincoli e le restrizioni del simulatore che hanno inevitabilmente influenzato le nostre decisioni.

3.3.1 LIMITAZIONI DEL SIMULATORE: RESTRIZIONI E VINCOLI

In questo capitolo discuteremo dei problemi che abbiamo riscontrato in CARLA durante gli esperimenti che abbiamo condotto portandoci ad ottenere dei risultati che non sono dipesi dall'implementazione del nostro algoritmo.

1. In una prima istanza dell'implementazione del sorpasso delle biciclette, al fine di generalizzare il valore dell'offset da utilizzare nel controllore di CARLA, avevamo deciso di utilizzare l'attributo "lane_width" del waypoint delle biciclette per farci restituire la grandezza della carreggiata. In questo modo, quando l'auto si trovava in una posizione favorevole per il sorpasso, l'offset veniva impostato al valore di " $-bike_waypoint.lane_width/2 + 0.3$ ". Tale approccio consentiva di ottenere un valore dinamico dell'offset, che variava in base alla dimensione effettiva della carreggiata. Tuttavia, durante l'analisi della mappa, abbiamo riscontrato che in alcune aree l'attributo "lane_width" non era correttamente configurato e restituiva un valore di larghezza pari a 0.05 metri.

```

agent.py M X | $ run_test_localsh M | config_agent_basicon M |
avior_agent | BehaviorAgent | > overtake_bikes_new
target_wpt.lane.type: " + str(l
print("Distanza dell'ego vehicle dalla macchina
if side_vehicle.state:
    if(self.debug):
        print("Ci sta un veicolo non posso sorpassa
if(self.debug):
    print("Distanza del veicolo dall'ego vehicle
if dist_bt看_bikes > self_vehicle.bounding_box.ex
    if (not side vehicle state or \
        (distance_side_vehicle > self_speed_lim
        and distance_front_vehicle < 8 \
        and distance_front_vehicle < 0 :
        if(self.debug):
            print("Overtaking to the left! ")
            self.local_planner.set_speed(self
            self.behavior_overtake_counter = 50
            diff position = ego waypoint.transform
            print("diff position: " + str(diff_posi
            self.lateral_controller.set_offset_bike
            input("OFFSET: " + str(self.lateral_c
else:

```

L'origine di questo valore anomalo è attribuibile al fatto che alcuni gruppi di biciclette risultavano essere posizionati su una corsia alternativa sul lato destro della carreggiata, il che causava l'errata

indicazione della sua larghezza. Questo valore anomalo provocava inevitabilmente una collisione con le biciclette. Di conseguenza, abbiamo deciso di rendere fisso tale valore, assegnandolo all'attributo "self.lane_width", e di considerarlo come la dimensione standard della carreggiata.

2. In alcuni casi, all'interno dei risultati del secondo e del quarto scenario viene segnalata un'infrazione relativa alla mancata frenata ad un segnale di stop. Tuttavia, tale infrazione non si verifica in tutti gli altri scenari che includono uno o più segnali di stop. Inoltre, osservando il video e analizzando i valori stampati sul terminale ([qui](#) il video), è evidente che il veicolo non solo rileva il segnale di stop, ma attende anche il tempo necessario prima di riprendere il normale comportamento dell'algoritmo. La ragione per cui viene rilevata questa infrazione potrebbe risiedere nel fatto che quel particolare segnale di stop in CARLA richiede che il veicolo principale sia più vicino al segnale stesso per poter rilevare effettivamente la frenata. Di conseguenza, abbiamo deciso di concentrarci su altre problematiche anziché adattare specificamente l'algoritmo per evitare questo problema.
3. Durante alcune simulazioni dello scenario 0, intorno ai tre quarti del percorso, è previsto un sorpasso di tre veicoli fermi nella nostra carreggiata. In circostanze normali, il nostro veicolo principale riesce a completare il sorpasso senza problemi. Tuttavia, in alcune simulazioni dello stesso scenario, il nostro veicolo è entrato in collisione con il primo veicolo fermo, causando un incidente. Dopo un'attenta analisi di questo fenomeno, abbiamo scoperto che la collisione non avviene perché l'algoritmo non rileva il veicolo fermo, ma perché viene rilevata al suo posto una bicicletta fantasma che non è presente sulla carreggiata. Poiché non era possibile escludere questa bicicletta dalla considerazione nell'algoritmo, in quanto avrebbe creato problemi nel sorpasso di gruppi di biciclette, abbiamo deciso di ignorare temporaneamente il problema, poiché non si è verificato in tutte le simulazioni.
4. Durante alcune simulazioni dello scenario 0, sono previsti dei sorpassi che coinvolgono tre veicoli fermi sulla carreggiata. Di solito, questi sorpassi vengono eseguiti correttamente senza problemi. Tuttavia, in alcune simulazioni abbiamo osservato che il nostro veicolo si scontra con un veicolo proveniente dalla direzione opposta, nonostante in prove precedenti i controlli effettuati abbiano consentito di completare con successo il sorpasso. Dopo un'attenta analisi e numerosi tentativi, abbiamo notato che il veicolo coinvolto è sempre lo stesso e abbiamo concluso che il veicolo compare nella simulazione dopo che i controlli per la fattibilità del sorpasso sono stati effettuati. Ciò provoca la situazione in cui il nostro veicolo principale non vede il veicolo in questione durante la valutazione del sorpasso e avvia la manovra. Tuttavia, durante l'esecuzione del sorpasso, il veicolo compare nella corsia opposta e si scontra inevitabilmente con il nostro veicolo principale prima che il sorpasso sia completato. In prove successive, abbiamo cercato di aumentare la "distanza di fattibilità del sorpasso" (`look_ahead_distance`), ma se lo facciamo, il nostro veicolo non inizia mai il sorpasso perché a quella distanza i veicoli vengono generati dalla simulazione. Pertanto, abbiamo deciso di considerare questa situazione come "anomala" e di non tenerla in considerazione, concentrandoci su altri aspetti più rilevanti. Una situazione simile si verifica quando si cerca di sorpassare gli oggetti "traffic_warning".
5. Nello scenario 4 è presente un incrocio controllato da uno stop, in cui è evidente una posizione asimmetrica dei waypoint rispetto al percorso che il veicolo principale dovrebbe seguire. Questa asimmetria dei waypoint genera un comportamento anomalo nel nostro veicolo, che non è stato possibile gestire. Infatti, per seguire correttamente questi waypoint, il veicolo si sposta fuori strada e, nel tentativo di correggere la traiettoria con il waypoint successivo, si scontra con uno o due veicoli nell'altra carreggiata. Nonostante abbiamo effettuato tentativi di regolare i parametri del controllore per ridurre questo problema, la situazione non è migliorata. Di conseguenza, abbiamo deciso di sospendere temporaneamente la risoluzione di questo problema, in quanto non era possibile modificare la posizione dei waypoint.

6. Nello scenario 1 si presenta la situazione in cui un pedone attraversa improvvisamente la carreggiata mentre il nostro veicolo principale sta procedendo. Il nostro veicolo si ferma correttamente, ma il pedone entra in collisione con un'auto proveniente dall'altra corsia e successivamente rimbalza sul nostro veicolo ([qui](#) il video). Nonostante sia riportata una collisione con il pedone nei risultati, non è causata dal nostro veicolo. Una situazione simile si verifica anche nello scenario 4 sia per i pedoni che per una bicicletta che attraversa sulle strisce pedonali. Poiché questa situazione non è influenzata dall'implementazione dell'algoritmo, abbiamo deciso di non considerare questo problema.

4 CAPITOLO 4

4.1 ANALISI QUALITATIVA E SVILUPPI FUTURI

Nell'implementazione attuale, l'obiettivo principale è stato quello di migliorare il punteggio complessivo di global driving score (riducendo il numero di infrazioni) tenendo conto delle priorità e del tempo a disposizione. Tuttavia, sono state necessarie alcune limitazioni e sono emerse alcune questioni che richiedono attenzione in una futura implementazione. Di seguito vengono elencate le questioni non affrontate:

- Assunzioni riguardanti il sorpasso: Nelle funzioni che gestiscono il sorpasso, si è fatto riferimento a una distanza specifica per determinare la fattibilità del sorpasso. Questa distanza è stata definita considerando l'assunzione che tutti gli altri veicoli rispettino i limiti di velocità imposti. Tuttavia, è necessario considerare scenari in cui i veicoli potrebbero superare i limiti di velocità, richiedendo una valutazione più dinamica e accurata.
- Velocità massima del veicolo: Attualmente, la velocità massima impostata per il nostro veicolo è di 30 km/h. Questo limite è stato scelto per garantire l'assenza di collisioni, ma non è stato gestito in modo dinamico in relazione alla distanza di frenata del veicolo. Pertanto, è necessario implementare una gestione dinamica della velocità per adattarsi alle diverse situazioni stradali.

4.1.1 SORPASSO DI AUTO

Il sorpasso di auto incidentate e del veicolo con segnale di avviso comporta lo spostamento del nostro veicolo nella corsia a sinistra fino a quando l'ostacolo non è più presente a destra. Sebbene questa funzione funzioni correttamente negli scenari specifici forniti, potrebbe essere generalizzata ulteriormente per tener conto del numero di corsie presenti su quella strada. Ad esempio, se ci sono due corsie per senso di marcia (quindi quattro in totale) e ci troviamo nella corsia di sinistra tra le due, sarebbe più corretto valutare uno spostamento nella corsia di destra prima di procedere con il sorpasso.

Inoltre, per effettuare il sorpasso di un'auto incidentata o del veicolo con segnale di avviso, ci siamo basati sulla presenza di un cartello stradale appropriato che lo permetta. Questo ci consente di distinguere le auto incidentate (che non ci consentono di procedere lungo la carreggiata evitando collisioni) dalle auto in sosta al di fuori della carreggiata (che non causano collisioni) e di evitare di gestire quest'ultime, poiché non è necessario.

Questi aspetti evidenziano opportunità di miglioramento nella gestione del sorpasso, considerando il numero di corsie e valutando la presenza di ostacoli in modo più preciso e dinamico. Inoltre, potrebbe essere utile implementare ulteriori criteri di valutazione, come il flusso del traffico e la velocità relativa degli altri veicoli, per garantire una navigazione sicura e realistica.

4.1.2 SORPASSO DI OGGETTI STATICI

Nell'implementazione attuale, il sorpasso del veicolo con segnale di avviso può essere effettuato solo spostandosi nella corsia di sinistra. Non sono stati gestiti casi in cui sono presenti più carreggiate o verificato se l'oggetto non si trova nella nostra stessa corsia. Questo aspetto richiede una gestione più sofisticata per affrontare scenari più complessi. In questa implementazione è stato preso in considerazione, come oggetto statico, il solo traffic warning, questo perché gli scenari forniti non richiedevano la gestione del sorpasso di altri oggetti statici. In futuro, si potrebbe valutare la possibilità di generalizzare la lista di oggetti statici e non limitarla al solo traffic warning.

4.1.3 SORPASSO DI BICICLETTE

Durante l'implementazione attuale, abbiamo assunto che le biciclette siano sempre presenti come un gruppo di due ciclisti, posizionati in fila indiana lungo il bordo destro della nostra carreggiata. Tuttavia, alcune considerazioni aggiuntive devono essere prese in considerazione per migliorare la gestione delle biciclette:

1. Posizione delle biciclette: Attualmente, le biciclette vengono rilevate anche se non sono posizionate lungo il bordo destro della carreggiata. Tuttavia, l'offset applicato al controllore del nostro veicolo potrebbe non essere sufficiente per evitare una collisione con le biciclette se si trovano in altre posizioni della corsia. È necessario sviluppare una gestione più robusta delle posizioni delle biciclette per garantire una navigazione sicura.
2. Sorpasso di una sola bicicletta: Nel caso in cui sia presente solo una bicicletta, il sorpasso avverrebbe correttamente senza collisioni. Tuttavia, la funzione corrente considera questa situazione come il caso in cui la distanza tra le due biciclette non è sufficientemente grande da permettere il rientro del nostro veicolo tra esse. Questo potrebbe portare all'applicazione dell'offset per un periodo più lungo del necessario. È necessario adattare la gestione del sorpasso per gestire correttamente il caso di una sola bicicletta.
3. Presenza di più biciclette: Nel caso in cui siano presenti più di due biciclette, il rientro verrebbe considerato solo tra le prime due biciclette, mentre le successive potrebbero essere a rischio di collisione poiché l'offset si azzererebbe prima che il sorpasso sia completato. Inoltre, non stiamo valutando le auto presenti nell'altra corsia ad una distanza sufficiente. È necessario sviluppare una gestione più completa delle situazioni in cui sono presenti più biciclette.
4. Durante l'attraversamento orizzontale di una bicicletta nella route 4, attualmente la situazione viene gestita semplicemente con l'emergency stop perché la bicicletta compare molto vicina al nostro veicolo e scompare immediatamente dopo. Tuttavia, questa situazione potrebbe essere gestita in modo più accurato attraverso l'utilizzo del bounding box delle biciclette (per intuire l'orientamento) e l'implementazione di un'operazione di tracking.

Calcolando il bounding box delle biciclette possiamo ottenere informazioni sull'orientamento della bicicletta e quindi comportarci in maniera diversa a seconda se la bici si muova orizzontalmente o verticalmente. Con l'uso di algoritmi di tracking, possiamo monitorare il movimento della bicicletta e prevedere il suo percorso durante l'attraversamento della carreggiata. Questo ci permetterebbe di adottare una strategia più precisa per gestire l'attraversamento orizzontale della bicicletta.

5. Gestione di motociclette: Attualmente, tutti i veicoli a due ruote vengono considerati come biciclette poiché nel nostro scenario non erano presenti motociclette. Tuttavia, nel caso in cui il nostro veicolo dovesse incontrare delle moto lungo il percorso, potrebbe essere necessaria una gestione specifica considerando la differenza di velocità tra il nostro veicolo e le moto, nonché la possibile differenza di dimensioni che potrebbe richiedere un offset diverso.

Queste considerazioni e questioni aggiuntive riguardanti le biciclette possono essere affrontate in una futura implementazione per migliorare la gestione delle situazioni stradali coinvolgendo queste tipologie di veicoli.

4.1.4 RESTRINGIMENTI

Nella versione attuale, si è ipotizzato che il restringimento stradale sia sempre presente nella corsia a sinistra della nostra. Pertanto, non sono stati gestiti i seguenti casi:

- Caso in cui il restringimento sia presente nella nostra corsia: In questa situazione, dovremmo applicare lo stesso offset utilizzato, ma con segno inverso. L'offset negativo ci consentirebbe di mantenere la distanza di sicurezza dalla barriera del restringimento.
- Nel caso in cui ci troviamo su strade a corsia multipla sia nello stesso verso che nel verso contrario, è necessario gestire il comportamento in maniera dinamica, tenendo conto di come il restringimento influenzi la nostra guida. Ciò richiede una valutazione più precisa e specifica delle condizioni stradali.

4.1.5 INCROCI

È corretto osservare che negli scenari forniti, in presenza di semafori, gli incroci sembrano essere correttamente gestiti grazie alla regolazione dei semafori stessi. Tuttavia, è importante considerare che l'implementazione del comportamento del veicolo negli incroci a T potrebbe essere ulteriormente migliorata per affrontare in modo più accurato e sicuro le diverse situazioni stradali. Alcuni aspetti da prendere in considerazione includono:

1. Gestione dei casi di incroci a 4 corsie e situazioni più complesse: attualmente, il comportamento del veicolo potrebbe non essere adatto per affrontare incroci a 4 corsie o situazioni stradali più complicate in cui è necessario dare la precedenza ai veicoli sulla destra. È importante sviluppare algoritmi e strategie di controllo appropriati per gestire queste situazioni in modo sicuro ed efficiente.
2. Miglioramento del rilevamento di veicoli provenienti dalle diverse corsie: l'utilizzo del tracking di oggetti potrebbe consentire una migliore rilevazione e comprensione della presenza di veicoli provenienti dalle varie corsie in prossimità dell'incrocio. Ciò consentirebbe di prendere decisioni dinamiche basate sulla situazione attuale, come l'adattamento della velocità, l'identificazione delle opportunità di attraversamento e la gestione delle precedenza.

4.2 CONCLUSIONI

Tutto il materiale riguardante il codice del nostro agente e i video dimostrativi sono contenuti all'interno della cartella condivisa su Google drive, consultabile al seguente [link](#). All'interno della cartella 'video' è possibile trovare il video completo contenente la simulazione di tutti gli scenari, oltre che le simulazioni dei singoli scenari. In più sono presenti spezzoni di video, che evidenziano dei casi anomali riscontrati durante la simulazione. All'interno del file 'AVD_Group12.zip' è possibile consultare il codice dell'algoritmo. Inoltre è presente un file 'simulation_result.json' contenente i risultati.