

POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Civile



Tesi di Laurea Magistrale

Simulazione del traffico con veicoli dotati di sistemi cooperativi di supporto alla guida in scenari autostradali

Relatore

Candidato

Francesco Paolo DEFLORIO

Francesco VARONE

Marzo 2021

Sommario

Negli ultimi anni l'Unione Europea ha promosso lo sviluppo di sistemi di trasporto intelligenti e cooperativi (C-ITS), ovvero quei sistemi che permetteranno agli utenti della strada e ai gestori del traffico di interagire tra loro, condividere informazioni in tempo reale e coordinare le loro azioni a vicenda. Tali sistemi porteranno un aumento dell'efficienza dei trasporti, del traffico e a migliorare la sicurezza stradale, non solo per gli utenti tradizionali, ma anche per i futuri veicoli automatici e garantire la loro completa integrazione nel sistema stradale.

In un certo senso, i veicoli di oggi sono già connessi, ma presto anche loro interagiranno con altri veicoli (da veicolo a veicolo V2V) e con l'infrastruttura stradale (da veicolo a infrastruttura V2I).

I sistemi C-ITS consentono di fare un passo avanti rispetto ai già esistenti sistemi ITS, fornendo informazioni in tempo reale e strategie di controllo su misura per conducenti specifici.

Lo scopo di questa tesi è verificare i vantaggi di tali sistemi cooperativi, in particolare, l'attenzione è concentrata su sistemi di cruise control adattivo cooperativo (CACC), che permettono una comunicazione di tipo V2V. Nel seguente elaborato viene descritto il funzionamento di tale sistema e vengono analizzati i benefici del CACC nei confronti delle caratteristiche di traffico (riduzione dei fenomeni di congestione o stabilità del flusso) e nei confronti della sicurezza stradale.

L'efficienza di questi sistemi è stata testata attraverso simulazioni in diversi scenari di traffico all'interno del simulatore PTV VISSIM, integrando in esso modelli di guida esterni, mediante algoritmi implementati in C ++.

Indice

| | |
|---|-----|
| Elenco delle tabelle | VI |
| Elenco delle figure | VII |
| Glossario | X |
| 1 Introduzione | 1 |
| 1.1 Background | 2 |
| 1.2 Obiettivi della Tesi | 3 |
| 1.3 Struttura della Tesi | 3 |
| 2 Analisi della Letteratura | 4 |
| 3 Strumenti della Simulazione | 9 |
| 3.1 Background | 9 |
| 3.1.1 CARLA Simulator | 10 |
| 3.1.2 Modifica dei parametri di VISSIM | 11 |
| 3.1.3 Integrazione di un modello di guida esterno in VISSIM | 12 |
| 3.2 Sviluppo dei Veicoli Automatici Connessi | 13 |
| 4 Metodologia per la Simulazione dei Sistemi di Guida | 15 |
| 4.1 Struttura dei modelli di guida esterni | 15 |
| 4.2 Cruise Control Adattivo Cooperativo | 20 |
| 4.2.1 Metodologia per il CACC | 21 |
| 4.2.2 Test all'interno di un Anello | 31 |
| 4.2.3 Condizioni del Modello di Guida CACC | 36 |
| 4.2.4 Algoritmi per i comportamenti di guida automatica | 36 |
| 4.3 Parametri del modello di guida interno | 40 |
| 5 Simulazione degli scenari di Traffico | 46 |
| 5.1 Scenario 1 | 46 |
| 5.1.1 Input della Simulazione | 47 |

| | |
|--|-----------|
| 5.2 Scenario 2 | 50 |
| 5.2.1 Input della Simulazione | 51 |
| 6 Risultati e Considerazioni | 54 |
| 6.1 Scenario 1 | 54 |
| 6.2 Scenario 2 | 62 |
| 6.2.1 Analisi sulla Sicurezza in un Bottleneck | 68 |
| 7 Conclusioni | 70 |
| A CACC Driver Model | 74 |
| B ACC Driver Model | 85 |
| Bibliografia | 92 |

Elenco delle tabelle

| | | |
|-----|--|----|
| 4.1 | Input degli scenari di test | 26 |
| 4.2 | Parametri del modello di Wiedemann | 41 |
| 4.3 | Parametri di Car-Following in VISSIM | 43 |
| 4.4 | Parametri del modello Wiedemann 99 in VISSIM | 45 |
| 5.1 | Parametri della simulazione | 48 |
| 5.2 | Composizioni degli input dei veicoli | 48 |
| 5.3 | Input della simulazione | 50 |
| 5.4 | Parametri della simulazione | 52 |
| 5.5 | Composizioni degli input dei veicoli | 52 |
| 5.6 | Input della simulazione | 53 |
| 6.1 | Velocità medie in km/h per i differenti livelli di MPR dei CACC . . | 55 |
| 6.2 | Variazione % delle velocità medie per i differenti livelli di MPR dei CACC rispetto allo scenario "10% CV" | 55 |
| 6.3 | Densità medie in veh/km per i differenti livelli di MPR dei CACC . | 57 |
| 6.4 | Variazione % delle densità medie per i differenti livelli di MPR dei CACC rispetto allo scenario "10% CV" | 57 |
| 6.5 | Tempi di viaggio medi in secondi per i differenti livelli di MPR dei CACC | 59 |
| 6.6 | Variazione % dei tempi di viaggio medi per i differenti livelli di MPR dei CACC rispetto allo scenario "10% CV" | 59 |
| 6.7 | Numero di possibili fenomeni di conflitto per le diverse composizioni veicolari | 68 |
| 6.8 | Variazione percentuale del numero di possibili fenomeni di collisioni per le diverse composizioni veicolari rispetto allo scenario "10% CV" | 68 |

Elenco delle figure

| | | |
|------|---|----|
| 3.1 | Architettura CARLA Simulator | 10 |
| 3.2 | Impostazioni del comportamento di guida in VISSIM | 12 |
| 4.1 | Visualizzazione 2D e 3D delle diverse tipologie di veicoli | 21 |
| 4.2 | Profili di velocità di due veicoli dotati di CACC | 25 |
| 4.3 | Profili di accelerazione di due veicoli dotati di CACC | 25 |
| 4.4 | Time Gap veicolo dotato di CACC | 26 |
| 4.5 | Platoon di veicoli a guida manuale e leader più lento | 27 |
| 4.6 | Platoon di veicoli CACC e leader più lento | 27 |
| 4.7 | Velocità del platoon di veicoli a guida manuale | 28 |
| 4.8 | Accelerazione del platoon di veicoli a guida manuale | 28 |
| 4.9 | Velocità del platoon di veicoli dotati di ACC | 29 |
| 4.10 | Accelerazione del platoon di veicoli dotati di ACC | 29 |
| 4.11 | Velocità del platoon di veicoli dotati di CACC | 30 |
| 4.12 | Accelerazione del platoon di veicoli dotati di CACC | 30 |
| 4.13 | Schema iniziale per il test su strada anulare | 31 |
| 4.14 | Profili di velocità dei veicoli a guida manuale all'interno dell'anello | 32 |
| 4.15 | Profili di velocità dei veicoli dotati di CACC all'interno dell'anello | 32 |
| 4.16 | Time gap dei veicoli a guida manuale all'interno dell'anello | 33 |
| 4.17 | Time gap dei veicoli dotati di CACC all'interno dell'anello | 33 |
| 4.18 | Clearance dei veicoli a guida manuale all'interno dell'anello | 34 |
| 4.19 | Clearance dei veicoli dotati di CACC all'interno dell'anello | 34 |
| 4.20 | Diagramma di flusso per CACC | 37 |
| 4.21 | Diagramma di flusso per la formazione del platoon | 39 |
| 4.22 | Rappresentazione grafica del modello di Wiedemann | 40 |
| 4.23 | Schermata dei parametri del comportamento di guida di VISSIM | 42 |
| 5.1 | Probabilità cumulata per velocità a guida manuale | 49 |
| 5.2 | Probabilità cumulata per velocità ACC, CACC | 50 |
| 6.1 | Velocità medie in km/h per i differenti livelli di MPR dei CACC | 56 |

| | | |
|------|--|----|
| 6.2 | Variazione % delle velocità medie per i differenti livelli di MPR dei CACC rispetto allo scenario 10% CV | 56 |
| 6.3 | Densità medie in veh/km per i differenti livelli di MPR dei CACC . | 58 |
| 6.4 | Variazione % delle densità medie per i differenti livelli di MPR dei CACC rispetto allo scenario "10% CV" | 58 |
| 6.5 | Tempi di viaggio medi in s per i differenti livelli di MPR dei CACC | 60 |
| 6.6 | Variazione % dei tempi di viaggio per i differenti livelli di MPR dei CACC rispetto allo scenario "10% CV" | 60 |
| 6.7 | Dettaglio n. 1 sul restringimento di corsia nello scenario "10% CV" . | 63 |
| 6.8 | Dettaglio n. 2 sul restringimento di corsia nello scenario "10% CV" . | 63 |
| 6.9 | Dettaglio n. 3 sul restringimento di corsia nello scenario "50% CV" . | 64 |
| 6.10 | Dettaglio n. 4 sul restringimento di corsia nello scenario "50% CV" . | 64 |
| 6.11 | Dettaglio n. 5 sul restringimento di corsia nello scenario "50% CV" . | 64 |
| 6.12 | Velocità media dei vari MPR lungo la sezione | 65 |
| 6.13 | Densità media dei vari MPR lungo la sezione | 65 |
| 6.14 | Ritardo medio in % per veicolo | 66 |
| 6.15 | Numero medio di Stop per veicolo | 66 |
| 6.16 | Diagramma spazio-tempo dei veicoli nello scenario "10%CV" | 67 |
| 6.17 | Diagramma spazio-tempo dei veicoli nello scenario "90%CV" | 67 |
| 6.18 | Numero di possibili fenomeni di conflitto per le diverse composizioni veicolari | 69 |
| 6.19 | Variazione percentuale del numero di possibili fenomeni di conflitto per le diverse composizioni veicolari rispetto allo scenario "10% CV" | 69 |

Glossario

ACC

Adaptive Cruise Control

ADAS

Advanced Driver Assistance Systems

CACC

Cooperative Adaptive Cruise Control

CAN

Controller Area Network

CAV

Connected Autonomous Vehicles

C-ITS

Cooperative Intelligent Transport Systems

DLL

Dynamic-Link Library

DSRC

Dedicated Short Range Communication

GPS

Global Positioning Systems

GUI

Graphic User Interface

ITS

Intelligent Transport Systems

MPR

Market Penetration Rate

V2I

Vehicle to Infrastructure

V2V

Vehicle to Vehicle

WSU

Wireless Safety Unit

Capitolo 1

Introduzione

Uno dei principali obiettivi dell'Ingegneria dei Trasporti è assicurare un servizio di trasporto sempre più sicuro ed efficace, sfruttando l'esperienza e le competenze dei professionisti e al tempo stesso sviluppando le nuove opportunità offerte dalla ricerca e dal progresso tecnologico. Secondo le ultime stime dell'Organizzazione mondiale della sanità, la morte per incidente stradale risulta essere la settima causa di decesso al mondo [1] e, solo in Europa, il 95 % degli incidenti stradali è causato da errore umano [2].

È inevitabile, dunque, implementare sistemi di sicurezza innovativi di supporto alla guida, che riescano a limitare o escludere del tutto il fattore umano. In tale ambito si configurano i già noti ADAS (Advanced Driver Assistance Systems) o ITS (Intelligent Transport Systems), sistemi che permettono l'automazione di processi indispensabili per la sicurezza e il comfort del conducente (Cruise control, mantenimento della corsia, assistenza alla frenata d'emergenza, riconoscimento sonnolenza e soglia di attenzione, sensori di prossimità e altro). Se tali sistemi rappresentano una realtà già effettiva sul mercato, uno step ulteriore, oggetto di studio, è la possibilità di far comunicare tra loro i veicoli e scambiarsi informazioni, implementando i cosiddetti C-ITS (Cooperative Intelligent Transport Systems). La Commissione europea ha adottato una nuova strategia per i C-ITS. L'iniziativa rappresenta una svolta importante verso la mobilità cooperativa, connessa e automatizzata e permetterà di immettere sul mercato veicoli in grado di "dialogare" tra loro (V2V: Vehicle to Vehicle) e con le infrastrutture di trasporto (V2I: Vehicle to Infrastructure).

I C-ITS permettono agli utenti della strada e della gestione della circolazione di condividere informazioni e utilizzarle per coordinare i loro spostamenti. Questo dovrebbe migliorare in modo significativo la sicurezza stradale, l'efficienza del traffico e il comfort di guida aiutando il conducente a prendere le decisioni giuste e adattarsi alla situazione del traffico.

La tendenza costante al miglioramento della sicurezza stradale in UE ha visto

negli ultimi dieci anni un rallentamento. I C-ITS possono sicuramente aiutare a rilanciare una dinamica positiva nella riduzione delle vittime della strada, al fine di raggiungere l'obiettivo dell'UE, nei prossimi anni, di dimezzare il numero complessivo delle vittime della strada nell'Unione europea tra il 2010 e il 2020.

La comunicazione tra veicoli, infrastrutture e altri utenti della strada è altresì prodromica per lo sviluppo dei futuri mezzi automatizzati, per aumentare la sicurezza e garantire la loro piena integrazione nel sistema globale dei trasporti. In altre parole, la connettività e la cooperazione sono prerequisiti per l'automazione sicura. La mobilità cooperativa permetterà, inoltre, un sensibile miglioramento delle caratteristiche del traffico e dell'impatto ambientale.

1.1 Background

La mobilità cooperativa basata su sistemi C-ITS contribuisce non solo a migliorare la sicurezza stradale, ma anche a migliorare le condizioni di traffico. Uno dei principali sistemi di guida cooperativa è rappresentato dal CACC (Cooperative Adaptive Cruise Control), un'evoluzione dell'ACC (Adaptive Cruise Control) già presente sul mercato. Il Cruise Control Adattivo permette di regolare la propria velocità in funzione della velocità del veicolo che precede, basandosi su un sistema di rilevamento mediante radar e sensori installati a bordo; il veicolo, dotato di ACC, dunque, manterrà la stessa velocità del veicolo che lo precede, tenendo dunque una distanza di sicurezza pressoché costante.

Il CACC, basato invece su una comunicazione wireless con il veicolo che lo precede (purché anch'esso sia dotato di CACC), permette di ricevere informazioni quali accelerazione e/o velocità in tempo reale dal veicolo, e adeguare tali valori alla guida, definendo quindi un sistema che riesca ad anticipare meglio e in tempi più rapidi i comportamenti di guida del veicolo che lo precede.

Il principale beneficio, inoltre, di suddetto sistema cooperativo, è la riduzione del time gap che sussiste tra due veicoli in marcia. Il time gap è inteso come intervallo temporale, misurato quindi in secondi, che c'è tra il passaggio di due successivi veicoli misurato in una sezione stradale dal paraurti posteriore del primo veicolo al paraurti anteriore del secondo veicolo. Tale valore è strettamente collegato alla velocità e al tipo di contesto di guida (urbano, extraurbano, ecc.), ma generalmente, tra due veicoli a guida tradizionale questo valore si aggira intorno ai 2,5 secondi; valore che scende a circa 1,4 se i veicoli sono dotati di ACC e a 0,5 circa se dotati di CACC.

Tale riduzione, come si vedrà in seguito, comporta un beneficio evidente sia per la sicurezza e il comfort di guida, che anche per l'efficienza della rete stradale, permettendo, ad esempio, di ridurre i fenomeni di congestione.

1.2 Obiettivi della Tesi

L'obiettivo di questa tesi è simulare lo sviluppo di sistemi di guida di tipo cooperativo nell'ambito di un contesto extraurbano per apprezzarne i benefici in termini di caratteristiche di traffico. La riproduzione di modelli di guida cooperativi è, dunque, effettuata mediante un simulatore di guida, PTV Vissim, che permette di implementare modelli di guida esterni, creati appositamente secondo le proprie esigenze, tramite librerie dinamiche programmate in C++.

Dopo aver istruito il simulatore sugli algoritmi di riproduzione del sistema di cruise control cooperativo, i veicoli connessi vengono introdotti nello scenario di simulazione con diverse percentuali in coesistenza con sistemi ACC e veicoli a guida tradizionale e se ne valutano gli effetti in diversi scenari. Lo scopo sarà quindi verificare e misurare l'efficienza dei sistemi di guida cooperativa in un contesto extraurbano e il loro impatto sulle caratteristiche di flusso.

1.3 Struttura della Tesi

Il seguente elaborato è articolato in sei parti. Al capitolo 2 si approfondiscono precedenti studi presenti in letteratura. Al capitolo 3 sono descritti gli strumenti della simulazione. Nel capitolo 4 si descrivono concettualmente gli algoritmi utilizzati per la definizione dei modelli di guida adattiva e cooperativa e vengono altresì definite le impostazioni di settaggio del simulatore. Nel capitolo 5 sono descritte approfonditamente le simulazioni effettuate nei vari scenari, i cui risultati seguiti da ulteriori considerazioni sono invece riportate nel capitolo 6. Il capitolo 7 contiene le conclusioni del lavoro.

Capitolo 2

Analisi della Letteratura

Sebbene in letteratura siano presenti diversi studi sulla validità dei sistemi ACC all'interno di una rete di veicoli a guida tradizionale [3], [4], sono ancora pochi invece gli studi che analizzano gli effetti dei sistemi di guida cooperativa in situazioni di traffico misto.

Una scarsa percentuale di veicoli dotati di ACC non ha quasi nessun beneficio sulle condizioni di traffico [5], a meno di un leggero incremento della densità, a seguito della riduzione del time gap [6].

Numerosi studi, invece, si sono soffermati sulla descrizione dei sistemi da un punto di vista elettronico e/o tecnologico [7]. Alcuni studi analizzano i sistemi cooperativi, per la valutazione dei benefici di un platoon di pochi veicoli.

Secondo tali studi [8] in presenza di una infrastruttura dedicata e con l'utilizzo del sistema cooperativo da parte di tutti gli utenti della strada, i valori di capacità possono aumentare fino a 8500 veicoli all'ora per corsia.

L'aumento di veicoli dotati di CACC e ACC nella rete ha portato a un incremento della capacità secondo uno studio condotto in un simulatore di traffico [6]. La simulazione è stata condotta in uno scenario autostradale ad una corsia in prossimità di uno svincolo, con rampa in ingresso e in uscita. Le analisi sono state condotte nei diversi tassi di penetrazione (100 % veicoli a guida manuale, 100 % veicoli dotati di ACC con time gap di 1,4 s, 100 % veicoli dotati di CACC con time gap di 0,5 s) per verificare la robustezza del sistema con questi casi semplici. La capacità nominale per i veicoli manuali, ACC e CACC è stata stimata rispettivamente a valori di 2050, 2200 e 4550 veicoli all'ora. Successivamente è stata analizzata una popolazione di veicoli misti, arrivando a concludere che il cruise control cooperativo può incrementare la capacità della corsia autostradale del doppio.

Il progetto CHAUFFEUR 2 ha analizzato tre situazioni mirate a ridurre il carico di lavoro di un conducente di un camion sviluppando un sistema di truckplatooning [9]. Grazie alla comunicazione V2V la distanza di sicurezza è ridotta a circa 6-12 m (equivalente a un time gap di 0,3-0,6 s a 80 km/h). Le simulazioni sono state

effettuate sia su pista che ai simulatori VISSIM e FARSI, ottenendo come beneficio un incremento della capacità, una riduzione del 20% delle emissioni e in generale un aumento della sicurezza. In ogni caso, viene stabilito che il platooning è per lo più fattibile di notte o in condizioni di basso volume di traffico perché con un flusso più alto, la stabilità del traffico decresce.

I veicoli dotati di CACC possono ridurre le distanze tra essi rispetto ad altre tipologie di veicoli. Pertanto, quando i veicoli pesanti dotati di CACC si muovono in un platoon, sperimentano una riduzione della resistenza del vento, che si traduce in una riduzione della potenza complessiva che un veicolo deve esercitare per mantenere la stessa velocità o accelerare. Ciò riduce il consumo di carburante e le emissioni dallo scarico.

Alam et al. [10] hanno valutato platoon formati da mezzi pesanti dotati di ACC. Hanno aggiunto una funzione in modo da consentire al veicolo successivo di ottenere le informazioni sulle condizioni del traffico davanti al veicolo di testa. Questa caratteristica è simile a un controller CACC e quindi i risultati sperimentali possono essere estrapolati per i veicoli dotati di CACC. Gli autori hanno dimostrato che grazie alla riduzione delle distanze e della resistenza del vento si può ottenere una riduzione del consumo di carburante tra il 4,7 e il 7,7%.

Bonnet et al. [11] hanno utilizzato un traino elettronico per consentire a due veicoli pesanti di muoversi a una distanza ravvicinata. Il veicolo di testa era guidato manualmente e il veicolo successivo aveva un controller per seguire automaticamente il leader. Sono stati condotti una serie di esperimenti a diverse combinazioni di velocità e distanze con la spaziatura massima di 16 metri. Gli autori hanno riscontrato una riduzione del consumo di carburante a tutti i livelli. Si è visto che la riduzione del consumo di carburante aumenta con la diminuzione della distanza, ma la riduzione raggiunge un plateau a 10 metri di distanza. Pertanto, uno spazio libero di 10 metri è ottimale per il risparmio di carburante. In questo studio è stato osservato un risparmio di carburante compreso tra il 5 e il 10 %.

Tsugawa et al. [12] hanno condotto uno studio per valutare il platooning rispetto a tre veicoli pesanti equipaggiati con CACC e hanno trovato risultati simili agli studi di cui sopra. Gli autori hanno analizzato l'impatto sulle emissioni e sul consumo di energia a causa delle interdistanze ravvicinate e hanno riscontrato una riduzione del 2,1% della CO₂ a una distanza di 10 m e un tasso di penetrazione del mercato del 40%.

Ci sono stati molti studi simili a quelli sopra menzionati e quasi tutti hanno dimostrato che il CACC avrà un impatto positivo sul consumo di carburante e sulle emissioni. Una conclusione comune che si può trarre dagli studi di cui sopra è che con distanze più ravvicinate ci sarebbe un aumento della riduzione del consumo di carburante e della riduzione delle emissioni.

I ricercatori hanno sviluppato un'ampia gamma di modelli di guida per modellare il comportamento di car-following e di lane-change dei veicoli ACC e CACC. Di

seguito si descrivono alcuni modelli presenti in letteratura.

Intelligent driver model (IDM) di Treiber et al [13] è un modello di guida pensato per veicoli automatici, ed è costituito da sette parametri; velocità desiderata, tempo di avanzamento sicuro, accelerazione massima, decelerazione desiderata, esponente dell'accelerazione, densità degli ingorghi e lunghezza del veicolo. Questi parametri vengono utilizzati per stimare l'accelerazione in ogni time step. In questo modello l'accelerazione è una funzione della velocità del veicolo, del gap e della differenza di velocità con il veicolo precedente.

Questa funzione fornirà un valore di accelerazione quando il veicolo è su strada libera e decelerazione quando il veicolo si avvicina ad un altro veicolo. Il termine di decelerazione dipende dal rapporto tra gap desiderato e critico e varia con la velocità e la velocità di avvicinamento. Nel traffico in equilibrio questo modello fornisce valori di accelerazione tali che il proprio veicolo mantenga un divario dipendente dalla velocità con il veicolo che precede. A bassa densità di traffico i valori di accelerazione sono simili ai valori di accelerazione durante la guida su strada libera. In caso di elevata velocità di avvicinamento questo modello ha la capacità di distinguere tra frenata di emergenza e frenata normale. In caso di situazione di emergenza questo modello è progettato in modo tale che ci sia alcuna collisione. Considera la reazione eccessiva dei conducenti in situazioni di emergenza. Modella anche la tendenza dei conducenti a mantenere piccoli spazi vuoti con una piccola differenza di velocità con il veicolo che precede.

Kesting et al. [14] hanno modificato l'IDM di Treiber et al. [13] in modo da modellare veicoli dotati di ACC e anche per superare il difetto del modello IDM. Il modello IDM, al fine di prevenire la collisione, stima la decelerazione con valori non realistici. Per prevenire questo fenomeno, l'IDM necessita di un limite superiore per una decelerazione sicura. Per avere una decelerazione sicura, l'IDM migliorato incorpora un modello di accelerazione euristica costante (CAH). In questo modello il guidatore assume un'accelerazione costante rispetto al veicolo che precede. Inoltre, non è richiesto alcun progresso sicuro o distanza minima. Si presume che la reazione del conducente sia trascurabile. Dai risultati si è visto che la decelerazione del modello CAH è significativamente meno negativa rispetto al modello IDM. Tuttavia, questo modello non riesce a modellare l'accelerazione dei veicoli ACC, quindi il modello CAH viene utilizzato in combinazione con il modello IDM per formare il modello IDM avanzato.

Van Arem et al. [15] hanno analizzato l'effetto del CACC su un flusso di traffico utilizzando il modello di simulazione MIXIC. Per studiare l'impatto del CACC sul flusso di traffico, gli autori hanno aggiunto una componente per gestire il CACC. Analogamente alla componente ACC di MIXIC, la componente CACC riprende una parte del modello longitudinale di MIXIC. Il modello CACC calcola l'accelerazione di riferimento per il veicolo sulla base dei dati ottenuti dal veicolo che precede. Il modello del conducente in combinazione con il modello del veicolo calcola le

coordinate dei veicoli ad ogni time step. Come la maggior parte dei modelli di micro-simulazione, ha un modello di cambio corsia e un modello longitudinale che aiuta a determinare la posizione laterale e longitudinale di un veicolo. Oltre a questi modelli standard, MIXIC ha un modello per ACC, ovvero quando i veicoli circolano su ACC una parte del modello longitudinale è controllata dalla componente ACC di MIXIC.

Il modello MIXIC [16] che segue l'auto ha due comportamenti di guida. Mentre la guida non è congestionata dal traffico, viene impiegato un comportamento di guida in cui il conducente cerca di mantenere o raggiungere la velocità desiderata. E in condizioni di traffico congestionato viene impiegato il comportamento di approccio a un leader in cui il veicolo regola la sua velocità e/o la distanza in modo da poter mantenere una distanza desiderata tra il veicolo che precede e sé stesso.

MIXIC dispone inoltre di due modelli per determinare l'azione dei veicoli rispetto al movimento longitudinale. Uno è un modello di guida che valuta l'accelerazione desiderata dai conducenti in base alle condizioni attuali. Il secondo è il modello del veicolo che tiene conto dello stato attuale dei veicoli. Il modello del veicolo è costituito da parametri come la posizione del pedale dell'acceleratore, la posizione del cambio e la forza sul pedale del freno. Il modello del conducente insieme al modello del veicolo determina l'effettiva accelerazione del veicolo. In situazioni in cui non c'è traffico davanti a una lunga distanza, viene utilizzato il comportamento di guida libero del modello del conducente. Il modello confronta la velocità attuale con la velocità desiderata e se devia più di un dato valore dalla velocità desiderata, l'accelerazione viene resa proporzionale all'errore di velocità tenendo conto del tempo di reazione del conducente. Nel comportamento che segue l'auto il conducente stima la velocità del veicolo di testa e cerca di mantenere una velocità relativa pari a zero e una distanza desiderata con il veicolo di testa. La componente relativa alla velocità è stata aggiunta al modello originale per aumentare la stabilità del flusso di traffico. L'accelerazione desiderata viene calcolata tenendo conto dei parametri di cui sopra.

Esistono diversi modelli di guida per i veicoli dotati di CACC. Questi modelli si basano su requisiti diversi e possono essere suddivisi in due grandi categorie; alcuni richiedono che il veicolo mantenga un intervallo di tempo desiderato e la seconda categoria di modelli richiede che i veicoli mantengano una distanza desiderata quando seguono un altro veicolo. Kesting et al. [14] è un modello basato sul tempo. Il modello di guida di Van Arem et al. [15] è un modello di guida basato sulla distanza. Tutti i modelli di guida impongono un limite superiore e inferiore all'accelerazione in modo da fornire una guida stabile.

In conclusione, dai diversi studi presenti in letteratura, si evince come una comunicazione veicolo-veicolo possa migliorare un sistema di cruise control adattivo istruendo il modello di guida con molte più informazioni sul veicolo che segue e molto più accurate. Le informazioni includono valori di velocità puntuali o

variazioni di accelerazione. Con tali informazioni, il sistema può anticipare meglio e in tempi più rapidi eventuali problemi, rendendo il veicolo più sicuro, più fluido nella guida e più veloce a rispondere alle variazioni di guida del veicolo che segue. Inoltre, come risultato di ciò, tale sistema permette di ridurre la distanza tra veicoli in totale sicurezza, riducendo il time gap a 0,5 s.

In aggiunta, se da un lato, il CACC risulta essere un sistema affidabile e sicuro per il conducente, esso permette anche una migliore gestione del traffico da parte degli addetti ai lavori, perché con una buona percentuale di utilizzo si hanno incrementi alla capacità e alla stabilità della stringa.

Infine, alcuni studi, evidenziano un ulteriore beneficio in termini di consumi, constatando come una riduzione delle interdistanze, dovuta all'applicazione del sistema cooperativo, contribuisca alla riduzione dei consumi di CO₂ e di carburante.

Tuttavia, sono carenti studi più intensi e a più larga scala sugli effetti dei sistemi cooperativi in condizioni di traffico misto.

Capitolo 3

Strumenti della Simulazione

Una simulazione di traffico può essere suddivisa in due grandi categorie: la macro-simulazione e la micro-simulazione.

La macro-simulazione è un'analisi di tipo macroscopico che permette di analizzare e quantificare gli effetti sulla viabilità e circolazione stradale derivanti da modifiche alla rete stradale. Tale analisi si basa su due input fondamentali, la domanda di trasporto, (caratterizzata da una matrice di origine/destinazione e costituita dai veicoli che percorrono la rete stradale) e dall'offerta di trasporto (costituita dalla rete di archi e nodi che rappresentano strade e intersezioni).

La micro-simulazione, utilizzata nell'ambito di questo studio, è invece in grado di analizzare in maniera puntuale il traffico e la sua evoluzione istantanea, elaborando istante per istante il movimento del veicolo, soggetto a preimpostati comportamenti di guida del conducente, ovvero come egli varia la sua velocità, accelerazione, manovre di svolta ecc...

Per effettuare una micro-simulazione di traffico è necessario l'utilizzo del simulatore adeguato e opportunamente calibrato secondo le esigenze richieste dal lavoro.

La calibrazione è un processo di regolazione di alcuni parametri propri del modello per migliorare la capacità simulativa, al fine di ottenere un modello che riesca a riprodurre in maniera accurata le locali condizioni di traffico. Tale operazione, richiede il confronto di alcune misure con alcuni valori di riferimento. Di seguito, sono descritti oltre ai software anche gli strumenti usati per lo sviluppo delle simulazioni d'interesse.

3.1 Background

PTV VISSIM è un microsimulatore di traffico del gruppo PTV GROUP usato per molteplici applicazioni nel campo dell'ingegneria dei trasporti e nella pianificazione urbana.

La micro-simulazione prevede che ogni entità, ogni agente del sistema sia simulato individualmente così come le interazioni tra esse. Al contrario, la macro-simulazione prevede la descrizione dei parametri di simulazione basata sui valori aggregati dei parametri di traffico.

In VISSIM la programmazione è un requisito fondamentale per sfruttare a pieno le potenzialità del software, infatti, limitatamente alla definizione dei modelli di guida, eventuali modifiche possono essere fatte solo ai parametri dei modelli di Wiedemann. Per impostare, ad esempio, le funzioni desiderate di car following o lane change ai veicoli, è necessario ricorrere alla COM Interface o sviluppare delle DLL. La versione del simulatore utilizzata è la 2021.00-01 con licenza concessa per scopi accademici.

Per lo sviluppo di librerie dinamiche per la cooperazione con il simulatore è stato utilizzato l'ambiente di sviluppo integrato di Visual Studio, con il quale sono stati creati i comportamenti di guida automatici ACC e CACC in C++. Un ulteriore software, testato in fase di approccio al lavoro, è CARLA Simulator, un simulatore open-source sviluppato in Unreal Engine. L'architettura è di tipo client-server. Il server è responsabile per tutto ciò che concerne la simulazione stessa come sensoristica, rendering, fisica del veicolo e altro. Il client consiste in una serie di moduli che controllano le operazioni degli agenti e dello scenario tramite API (scritte in Python o C++). Tuttavia, per il sopraggiungersi di diverse limitazioni descritte di seguito, tale simulatore è stato scartato.

3.1.1 CARLA Simulator

CARLA Simulator, CAR Learning to Act, è un simulatore open-source per ricerche e studi nel campo della guida automatica. CARLA è sviluppato in Unreal Engine ed è basato su un'architettura client-server. Qualsiasi controllo, azione deve essere lanciata da apposito terminale mediante API e moduli scritti in Python o C++.

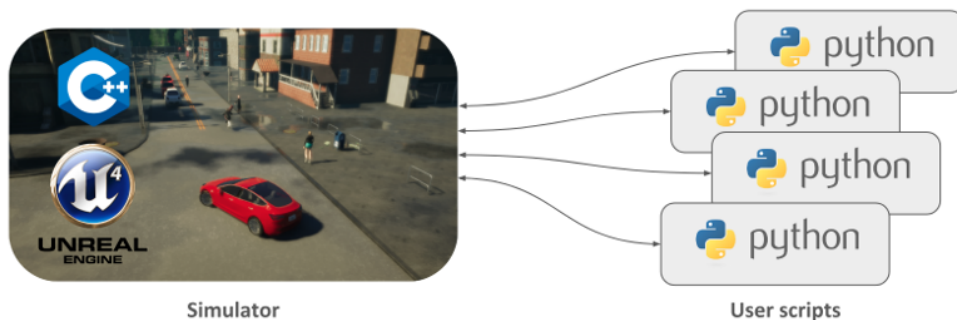


Figura 3.1: Architettura CARLA Simulator

Il software alla versione 0.9.10 è compatibile con i sistemi operativi Linux e Windows e per il suo funzionamento richiede una GPU dedicata o al più una GPU di 4 GB è consigliata dato l'elevato sforzo computazionale richiesto. È, altresì, richiesta una connessione a due porte TCP (2000 e 2001 di default) e una buona connessione internet. In fase di installazione, è necessario disporre di alcuni pacchetti per la gestione degli eseguibili e delle librerie.

I principali moduli di CARLA sono il "client" e il "world"; il "client" è il modulo che l'utente avvia per chiedere informazioni o dare istruzioni al server. Tale client funziona attraverso una connessione a un indirizzo IP e una porta specifici e comunica con il server via terminal. In caso di istruzioni multiple è possibile avviare più client allo stesso tempo.

Il "World" è l'oggetto che rappresenta la simulazione; contiene i metodi principali per generare gli attori della simulazione, per cambiare le condizioni meteorologiche e altro. Per ogni simulazione vi è solo un World possibile.

Altri moduli sono poi rappresentati dagli "Actors" e i "Blueprints"; Gli "Actors" o attori sono tutti quegli elementi che partecipano alla simulazione (Veicoli, pedoni, sensori, segnali, semafori ecc.) mentre i "Blueprints" sono una serie di schemi già predefiniti e personalizzabili necessari per generare gli actor.

Tra i principali punti di forza di CARLA, vi è la possibilità di simulare scenari di guida automatica in condizioni realistiche, data la buona personalizzazione anche delle condizioni meteorologiche e l'elevata qualità di dettaglio grafica, ma tra i principali scopi del simulatore vanno menzionati senz'altro i sensori, indispensabili per il funzionamento di un veicolo a guida automatica. Tali sensori sono essenzialmente actors e sono definiti da una apposita biblioteca di Blueprints.

Tuttavia, a discapito delle evidenti potenzialità del software, CARLA presenta delle limitazioni per il lavoro in oggetto, dato che è pensato per sviluppare essenzialmente algoritmi di machine learning utili a fronteggiare situazioni su piccola scala apprezzando la risposta dell'ego vehicle, ovvero il veicolo che si sta controllando lato client. Alla versione attuale, inoltre, l'utilizzo dei sensori è una prerogativa di Linux e questo ha creato problemi di incompatibilità. Non è stato possibile implementare modelli di guida personalizzati, non gestibili con API, ma possibili solo mediante modifica del server.

La simulazione di sistemi di guida cooperativi su larga scala, alla versione attuale non sembra essere una prerogativa di CARLA, preferibile invece per implementare algoritmi di machine learning su un solo veicolo e registrarne la risposta.

3.1.2 Modifica dei parametri di VISSIM

VISSIM permette all'interno dell'interfaccia la modifica dei parametri di guida d'interesse, ciò permette di calibrare la simulazione secondo le proprie esigenze. Modificare i parametri di guida permette di variare le caratteristiche del veicolo,

come l'headway (la distanza temporale in secondi tra i paraurti anteriori dei veicoli), la distanza di sicurezza, il numero di veicoli che un conducente riesce a vedere e molto altro (Figure 3.2).

I parametri personalizzabili modificano gli algoritmi già esistenti di Wiedemann 74 e Wiedemann 99 per quanto riguarda il car following e le impostazioni predefinite di lane change di VISSIM.

Per sviluppare i comportamenti di guida automatici e/o connessi è opportuno implementare modelli di guida esterni, possibili mediante DLL.

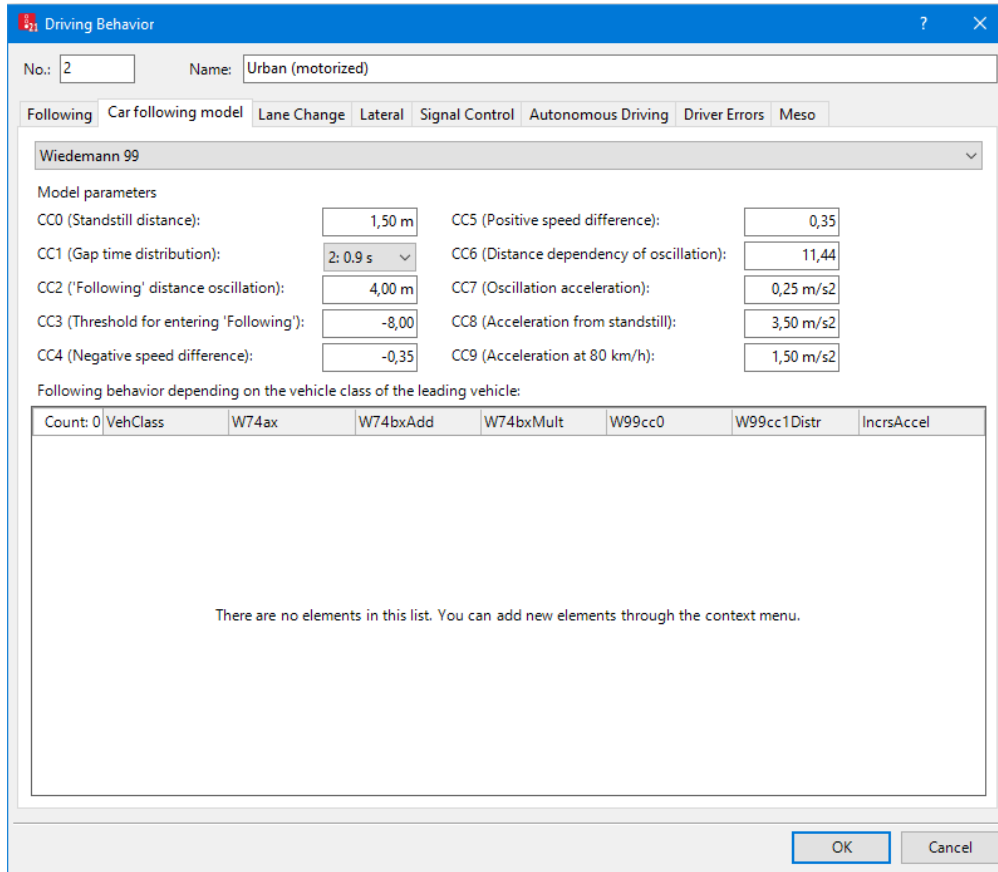


Figura 3.2: Impostazioni del comportamento di guida in VISSIM

3.1.3 Integrazione di un modello di guida esterno in VISSIM

VISSIM consente l'integrazione di Dynamic-Link Libraries per lo sviluppo di strumenti complessi da eseguire all'interno del simulatore. Una DLL è una libreria software che viene caricata, appunto, dinamicamente durante l'esecuzione del

software. Questa libreria contiene un codice che può essere usato in sostituzione di sistemi implementati all'interno di VISSIM. Si possono creare algoritmi più complessi per il calcolo delle emissioni o per il controllo semaforico, ma per questo lavoro di tesi, sono stati implementati delle DLL che introducono dei modelli di guida automatica basata sia su sistemi ACC che CACC.

I nuovi DriverModel scritti in Visual Studio, con linguaggio di programmazione C++, seguono uno schema ben preciso, definito all'interno della guida di PTV VISSIM. Essi contengono le informazioni che il veicolo deve recepire e quelle istruzioni che deve seguire per ogni step di simulazione. Gli algoritmi definiti riguardano, ad esempio, nuove funzioni di accelerazione e/o di velocità.

La struttura della DLL e gli algoritmi implementati sono definiti nei capitoli successivi in forma concettuale e riportati in Appendice in forma completa.

3.2 Sviluppo dei Veicoli Automatici Connessi

Per la riproduzione di Veicoli Automatici Connessi (CAV: Connected Automated Vehicles) all'interno del simulatore è opportuno individuare tutte le fasi necessarie. È, infatti, necessario capire come riprodurre al simulatore quei componenti hardware e software installati a bordo del veicolo, come effettuare le modifiche necessarie e con quale strumento, come configurare la valutazione dei risultati e come avviare la simulazione.

I procedimenti sono descritti concettualmente di seguito.

La tecnologia che rende possibile un sistema cooperativo è quello specifico componente usato per definire l'applicazione e la trasmissione delle informazioni necessarie per far interagire tra loro i veicoli.

Il veicolo è generalmente dotato di un set di sensori e un'unità dedicata di comunicazione a corto raggio (DSRC) con un sistema di GPS incorporato in un'unità wireless sicura (WSU). I controlli sono implementati mediante specifici software inclusi in centraline che ricevono i dati sia dalla WSU che dal Controller Area Network (CAN) del veicolo.

È importante conoscere la tecnologia usata per capire dove apportare le modifiche nel microsimulatore.

Gli strumenti necessari per implementare comportamenti di guida cooperativa in VISSIM sono stati descritti al capitolo 3.1.2 e 3.1.3. I parametri di guida di VISSIM possono essere modificati per essere integrati con modelli di guida esterni. Modificando parti del DriverModel è possibile seguire nuovi algoritmi che modificano, ad esempio, la velocità desiderata, l'accelerazione istante per istante, il comportamento da seguire nei confronti di un determinato veicolo che segue o che precede, la corsia di destinazione per effettuare un cambio corsia e con quali caratteristiche geometriche e molto altro. La DLL può essere impostata per diversi

tipi di veicolo. Per ogni step di simulazione, precedentemente impostato, ciascun veicolo che segue tale DLL riceve le informazioni definite ed esegue le istruzioni richieste.

Una volta individuato lo strumento necessario per implementare nuovi modelli di guida all'interno del simulatore è necessario programmare per modificare e scrivere gli algoritmi all'interno di questo tool.

Per le modifiche dei parametri di Wiedemann si interagisce con la GUI di VISSIM. Per le modifiche al DLL servono software esterni inclusi all'interno dell'OS. Per questo lavoro è stato usato Microsoft Visual Studio 2019 con linguaggio di programmazione C++.

VISSIM consente di configurare le opzioni per registrare le valutazioni d'interesse e fornire gli output necessari. Gli output registrati riguardano le condizioni generali di traffico, le caratteristiche di deflusso per corsia, le traiettorie dei veicoli e le caratteristiche dinamiche dei veicoli per ogni step di simulazione.

Data la natura stocastica del traffico, è necessario eseguire diverse simulazioni per descrivere al meglio le caratteristiche medie di flusso e ogni simulazione deve essere caratterizzata da un diverso seme casuale. È necessario, inoltre, escludere dalla simulazione le sezioni influenzate dall'input dei veicoli ed escludere un periodo di warm-up.

La valutazione delle simulazioni deve essere fatta secondo le intenzioni del lavoro richiesto. Capire che tipo di valutazione si voglia fare è necessario per calibrare al meglio il modello e gestire i parametri di output. Le simulazioni condotte nell'ambito di questo lavoro di tesi hanno come obiettivo la valutazione dei benefici di sistemi cooperativi alle caratteristiche di traffico e il loro impatto sulla sicurezza. La valutazione può essere eseguita con diversi software oltre VISSIM, quali SQLite e Microsoft Excel.

Capitolo 4

Metodologia per la Simulazione dei Sistemi di Guida

In questo capitolo viene descritta la metodologia seguita per l'implementazione dei sistemi di guida automatici e tradizionali, dall'approfondimento concettuale del loro funzionamento all'applicazione integrativa nel simulatore.

Vengono definite le indicazioni seguite per la costruzione di librerie dinamiche da far interagire con il simulatore per l'esecuzione di modelli di guida esterni, e viene altresì descritta la metodologia relativa al cruise control cooperativo e adattivo.

4.1 Struttura dei modelli di guida esterni

Gli algoritmi per i comportamenti di guida sono stati implementati in VISSIM mediante DLL esterne. Le istruzioni necessarie per la definizione della struttura di questa libreria e le informazioni che passeranno all'interno del simulatore sono descritte all'interno della "Interface Documentation" di VISSIM stesso.

Di seguito viene spiegata concettualmente la struttura della DLL, il suo funzionamento e la spiegazione dei codici che, invece, sono riportati in forma completa in Appendice A (CACC Driver Model) e Appendice B (ACC Driver Model).

Durante una simulazione, Vissim chiama il codice DLL per ogni veicolo interessato in ogni time step per determinare il comportamento del veicolo, quindi passa lo stato corrente del veicolo e dei veicoli vicini alla DLL che calcola l'accelerazione / decelerazione del veicolo e il comportamento laterale (principalmente per i cambi di corsia) e restituisce questi valori a Vissim per essere utilizzati nel time step corrente. Il modello di guida esterno può essere attivato per ogni tipo di veicolo

separatamente nella finestra di dialogo "Vehicle Type" spuntando la modalità "Usa External Driver Model" nella scheda "External Driver Model" e selezionando un file DLL. Se questa opzione è selezionata, il comportamento di guida di tutti i veicoli appartenenti alla suddetta tipologia verrà calcolato dalla DLL selezionata. Per fare in modo che la simulazione recepisca più modelli esterni il numero di core non deve essere impostato su 1 nei parametri di simulazione della rete Vissim e la DLL deve confermare di supportare il multithreading imponendo tale funzione nel codice. In caso contrario, l'esecuzione della simulazione viene annullata con un messaggio di errore.

Se per una simulazione vengono utilizzati più core, i veicoli su diversi links possono essere gestiti da thread diversi. L'assegnazione dei link a questi thread può essere diversa in ogni time step. Le funzioni della DLL vengono chiamate durante un time step in una sequenza non deterministica da più thread. Quindi i dati possono essere assegnati correttamente solo se viene utilizzata l'archiviazione locale del thread (invece di variabili globali che vanno bene per l'uso a thread singolo).

Per l'implementazione della DLL, vengono forniti diversi file di codice sorgente:

- DriverModel.h:

File di intestazione per la DLL del modello di guida. Questo file non viene modificato in quanto contiene le definizioni di tutte le costanti "type" e "number" usate da Vissim nel chiamare le funzioni "DriverModel *" della DLL.

- DriverModel.cpp:

File sorgente principale della DLL del modello di guida. Questo file è il luogo in cui sono aggiunti i calcoli o le funzioni dell'algoritmo del modello di comportamento di guida. Il preprocessore define DRIVERMODELEXPORTS deve essere impostato nelle opzioni del compilatore per la compilazione di DriverModel.cpp!

- DriverModel.vcproj:

file di progetto Visual C++ 2010 per la DLL del modello di guida. Questo file può essere utilizzato se la DLL deve essere creata con Microsoft Visual C++.

Una DLL del modello di guida deve contenere ed esportare 3 funzioni chiamate da Vissim: DriverModelSetValue, DriverModelGetValue e DriverModelExecuteCommand. Due funzioni aggiuntive sono opzionali (ma obbligatorie per alcuni dati): DriverModelSetValue3 e DriverModelGetValue3.

Vissim passa il valore corrente dell'elemento di dati indicato da `type` e indicizzato da `index1` e talvolta `index2`, in `DriverModelSetValue3` anche da `index3`. Il valore viene passato in `long_value`, `double_value` o `string_value`, a seconda del tipo. In alcuni casi, i valori vengono passati in più di una di queste variabili. Il codice nella funzione deve, inoltre, assicurarsi di salvare il valore, se è richiesto in seguito per il calcolo del comportamento di guida, perché la prossima chiamata di questa funzione da Vissim sovrascriverà il parametro locale. La funzione deve restituire 1 per tutti i `type` che non sono contrassegnati come opzionali. Per i `type` facoltativi, può restituire 0 per informare Vissim che non gestisce questo `type`.

Le costanti di comando attualmente disponibili sono `DRIVER_COMMAND_INIT`, `DRIVER_COMMAND_CREATE_DRIVER`, `DRIVER_COMMAND_MOVE_DRIVER` e `DRIVER_COMMAND_KILL_DRIVER`. La funzione deve restituire 1 per tutti questi comandi affinché Vissim interrompa l'esecuzione della simulazione. Prima che Vissim richieda l'esecuzione di uno dei comandi disponibili (Init, CreateDriver, MoveDriver, KillDriver) della DLL, ci sono sempre diverse chiamate della funzione DLL `DriverModelSetValue`, una per ogni elemento di dati che potrebbe essere utilizzato dalla DLL durante l'esecuzione del comando. Dopo che il comando `MoveDriver` ha terminato il calcolo, lo stato risultante del veicolo viene recuperato dalla DLL in modo simile nuovamente da diverse chiamate di `DriverModelGetValue`.

Ci sono quattro comandi che la DLL deve implementare: `Init`, `CreateDriver`, `MoveDriver` and `KillDriver`. Di seguito sono riportati quelli scelti per la simulazione.

INIT

Questo comando viene eseguito attraverso una chiamata di *DriverModelExecuteCommand* (`DRIVER_COMMAND_INIT`) all'avvio della simulazione per inizializzare la DLL. Alcuni parametri basilari sono passati alla DLL prima di questa chiamata attraverso *DriverModelSetValue* () (chiamata "Set" di seguito), e altri valori sono ottenuti dalla DLL prima e dopo questa chiamata attraverso *DriverModelGetValue* () ("Get"). La sequenza di chiamate della DLL è la seguente, per ogni tipologia di veicolo che usa la DLL:

Get `DRIVER_DATA_STATUS` (opzionale)

Get `DRIVER_DATA_STATUS_DETAILS` (solo se `STATUS` non è 0; opzionale)

Set `DRIVER_DATA_PARAMETERFILE` (opzionale)

Set `DRIVER_DATA_Timestep`

Set `DRIVER_DATA_TIME`

Set `DRIVER_DATA_VEH_TYPE`

Get `DRIVER_DATA_WANTS_SUGGESTION`

Get `DRIVER_DATA_SIMPLE_LANECHANGE`

Get `DRIVER_DATA_WANTS_ALL_NVEHS` (opzionale)

Poi solo una volta:

Get DRIVER_DATA_ALLOW_MULTITHREADING (opzionale)

DriverModelGetValue deve essere *long_value a 1 e ritornare 1 se il DLL deve gestire più threads.

CREATE DRIVER

DriverModelExecuteCommand (DRIVER_COMMAND_CREATE_DRIVER) viene chiamato da Vissim durante l'esecuzione della simulazione ogni volta che un nuovo veicolo viene inserito nella rete di Vissim (all'inizio di un time step o da un ingresso del veicolo). Nello stesso time step, seguirà un comando *MoveDriver* per lo stesso veicolo. La sequenza delle chiamate della DLL è la seguente:

Set DRIVER_DATA_VEH_TYPE = numero del tipo di veicolo

Set DRIVER_DATA_VEH_DESIRED_VELOCITY = Velocità desiderata

Execute DRIVER_COMMAND_CREATE_DRIVER

MOVE DRIVER

DriverModelExecuteCommand (DRIVER_COMMAND_MOVE_DRIVER) viene chiamato da Vissim durante l'esecuzione della simulazione ogni volta per time step per ogni veicolo che utilizza questa DLL DriverModel. Prima di questa chiamata, ci sono molte chiamate di *DriverModelSetValue()* per passare lo stato corrente del veicolo e dei suoi dintorni alla DLL. Dopo l'esecuzione del comando, ci sono diverse chiamate di *DriverModelGetValue()* per recuperare i nuovi valori per l'accelerazione e il comportamento laterale. La sequenza delle chiamate alla DLL è la seguente:

Set DRIVER_DATA_VEH_LANE = numero della corsia (più a destra = 1)

Set DRIVER_DATA_VEH_VELOCITY = velocità attuale [m/s]

Set DRIVER_DATA_VEH_ACCELERATION = accelerazione attuale [m/s²]

Set DRIVER_DATA_VEH_LENGTH = lunghezza del veicolo [m]

Set DRIVER_DATA_VEH_MAX_ACCELERATION = accelerazione massima [m/s²]

Set DRIVER_DATA_VEH_COLOR = colore del veicolo

Set DRIVER_DATA_VEH_DESIRED_VELOCITY = velocità desiderata [m/s]

Set DRIVER_DATA_VEH_TYPE = numero del tipo di veicolo

Per ogni veicolo vicino (fino a due ciascuno a valle e a monte, su un massimo di 2 corsie ciascuno su entrambi i lati e sulla corsia attuale) vengono passati da Vissim diversi valori: index1 e index2 sono usati come segue per DRIVER_DATA_NVEH_*: index1 = corsia di riferimento: +2 = seconda corsia a sinistra, +1 = corsia successiva a sinistra, 0 = corsia attuale, -1 = corsia successiva a destra, -2 = seconda corsia a destra.

index2 = Posizione relativa: positive = a valle (+1 Primo a valle, +2 Secondo a valle, se ne possono aggiungere altri se DRIVER_DATA_WANTS_ALL_NVEHS

è impostato) negativo = a monte (-1 primo a monte, -2 secondo a monte, se ne possono aggiungere altri se DRIVER_DATA_WANTS_ALL_NVEHS è impostato.)
Set DRIVER_DATA_NVEH_DISTANCE = distanza da paraurti anteriore a paraurti anteriore [m]
Set DRIVER_DATA_NVEH_REL_VELOCITY = differenza di velocità [m/s]
Set DRIVER_DATA_NVEH_ACCELERATION = accelerazione attuale del veicolo vicino [m/s²]
Set DRIVER_DATA_NVEH_LENGTH = lunghezza del veicolo vicino [m]
Set DRIVER_DATA_NVEH_TYPE = numero del Vehicle Type in VISSIM
Get DRIVER_DATA_VEH_DESIRED_VELOCITY = velocità desiderata [m/s]
Get DRIVER_DATA_VEH_COLOR = colore del veicolo
Get DRIVER_DATA_DESIRED_ACCELERATION = nuova accelerazione [m/s²], (opzionale)

KILL DRIVER

DriverModelExecuteCommand (DRIVER_COMMAND_KILL_DRIVER) è chiamato da VISSIM quando un veicolo raggiunge la destinazione e quindi lascia la rete (in modo da lasciare spazio in memoria che la DLL aveva allocato per quel veicolo):

Set DRIVER_DATA_VEH_ID = ID of the vehicle to be killed
Execute DRIVER_COMMAND_KILL_DRIVER

4.2 Cruise Control Adattivo Cooperativo

Il Cruise Control Adattivo Cooperativo (in inglese Cooperative Adaptive Cruise Control o brevemente CACC) è un sistema di assistenza alla guida, basato su una comunicazione di tipo V2V, con lo scopo di controllare automaticamente la marcia del veicolo in totale sicurezza e, altresì, incrementare la capacità della rete stradale. Il suo funzionamento è simile al cruise control adattivo (in inglese Adaptive Cruise Control o ACC) che permette di regolare la velocità del proprio veicolo (ego vehicle) in funzione del veicolo che segue (lead vehicle o leader) basandosi su un riconoscimento della velocità dello stesso attraverso sensori LIDAR, RADAR e telecamere che garantiscono, secondo uno studio [17] un range di funzionamento di circa 120 m. Il sistema ACC permette, dunque, di decelerare o accelerare quando il veicolo si avvicina ad un leader più lento e di accelerare fino a raggiungere la velocità precedentemente settata quando non c'è alcun veicolo che precede all'interno del range di funzionamento, comportandosi come un normale Cruise Control. Essendo un sistema automatico di regolazione della velocità, l'ACC permette, dunque, di ridurre la normale distanza di sicurezza, che per un veicolo a guida manuale, è legata a un fattore di percezione/reazione ovviamente ridotto nei sistemi automatici. La distanza di sicurezza (D_s) viene definita all'art. 5.1.2 del D.M. 6792/2001 come somma dello spazio di frenatura (S_f) e dello spazio percorso durante il tempo impiegato dal momento in cui si percepisce il pericolo fino al momento in cui inizia la frenata, detto appunto, di percezione/reazione (τ).

$$D_s = S_f + \tau * v \quad (4.1)$$

Il tempo di percezione/reazione è fortemente variabile da conducente a conducente oltre che dalla velocità, condizioni di visibilità, ecc. La normativa vigente, il D.M. 6792/2001, calcola tale valore come segue:

$$\tau = (2,8 - 0,01v) \quad (4.2)$$

Assumendo, dunque, una linearità con valori di τ decrescenti all'aumentare della velocità da 2,5 s per 20 km/h fino a 1,5 s per 130 km/h.

In conclusione, i veicoli a guida manuale (senza l'ausilio di sistemi di cruise control) si mantengono a una distanza dal veicolo che precede di circa 1,5-2,5 secondi. La distanza tra due veicoli misurata in secondi equivale all'intervallo di tempo tra il passaggio di due successivi veicoli misurato in una sezione stradale dal paraurti posteriore del primo veicolo al paraurti anteriore del secondo veicolo ed è chiamato "time-gap".

I veicoli dotati di ACC riescono, dunque, a ridurre il time-gap da 2,5 secondi circa alle alte velocità a 1,4 secondi, riducendo così la distanza di sicurezza e permettendo un incremento della capacità della strada.

I veicoli dotati di CACC, invece, si basano, oltre che su sensori, anche su una comunicazione wireless di tipo V2V, ricevendo, dunque, molte più informazioni dal veicolo leader. Tali informazioni sono, ad esempio, i valori di velocità e accelerazione istante per istante che permettono di ridurre ulteriormente il tempo di risposta delle azioni di marcia del veicolo nei confronti del leader garantendo per il conducente un controllo della marcia molto più stabile e confortevole, oltre che sicuro.

Il valore di accelerazione del veicolo che segue è un requisito importante per il funzionamento di un sistema CACC perché la sua acquisizione è utilizzata per generare un valore proprio di accelerazione tale che il time-gap sia ulteriormente ridotto. I veicoli equipaggiati con CACC, consentono, infatti, di ridurre il time-gap fino a 0,5 secondi, garantendo così in teoria, un notevole incremento della capacità. Il cruise control cooperativo, basandosi su una tecnologia wireless ha, inoltre, un range di funzionamento molto più ampio rispetto a veicoli con ACC, arrivando a coprire una distanza anche di 300 metri [17].

Ovviamente, il CACC funzionerà esclusivamente nel caso in cui il veicolo leader, che si mantiene ad una distanza di sicurezza dal veicolo che lo precede, sia anch'esso dotato di CACC, con la possibilità quindi, in presenza di più veicoli connessi, di formare dei platoon. Se il veicolo che segue non è dotato di CACC, il veicolo si comporterà come un normale ACC, o in assenza di veicoli, come un normale CC.

4.2.1 Metodologia per il CACC

Come descritto nei capitoli precedenti, le simulazioni (descritte nei capitoli successivi) prevedono tre tipi di veicoli, quelli tradizionali o a guida manuale, quelli dotati di ACC e quelli equipaggiati con il CACC. Per distinguere visivamente i veicoli tra loro sono state adottate tre colorazioni differenti. I veicoli manuali in giallo, i veicoli dotati di ACC in verde e i veicoli dotati di CACC sono rossi o neri (in base allo stato di guida nel quale si trovano, vedi cap. 4.2.3). Un esempio è riportato nella figura 4.1.

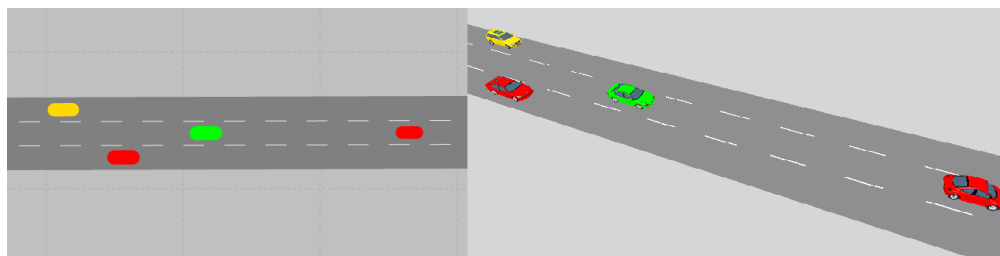


Figura 4.1: Visualizzazione 2D e 3D delle diverse tipologie di veicoli

Per simulare la cooperazione tra veicoli e i comportamenti di guida è stato necessario implementare dei modelli di guida esterni sottoforma di DLL, librerie dinamiche che interagiscono con il simulatore in tempo reale. Sono state create due DLL, sia per l'ACC che per il CACC, mentre la guida manuale è stata affidata ai parametri interni di VISSIM.

Durante la simulazione la DLL riceve determinate informazioni dalla simulazione stessa, tali informazioni permettono di definire quegli algoritmi necessari per i parametri d'interesse, quali accelerazioni, velocità, distanze, formazione di platoon che vengono applicate in VISSIM per ogni step di simulazione.

Tra le potenzialità della cooperazione e quindi della comunicazione wireless, vi è la possibilità di formare platoon in un numero predefinito di veicoli. La DLL, quindi, permette di identificare il veicolo leader del platoon e trasmette al veicolo le informazioni necessarie per aggregarsi (join) al platoon, quali, ad esempio, l'accelerazione o la decelerazione da adottare per unirsi al platoon in sicurezza.

Gli algoritmi usati per il calcolo dell'accelerazione dei veicoli automatici si basano su quelli proposti da Xiao et al. [17] e da Van Arem et al. [15].

I controller ACC-CACC si dividono in due categorie per le diverse condizioni di guida e per ognuna di esse c'è un diverso modello di accelerazione da implementare. In primo luogo, vi è la modalità di controllo della velocità o "Cruising Mode" adottato per mantenere una velocità predefinita in assenza di un veicolo che precede. La modalità di controllo del gap o "Gap Regulation Mode" funziona, invece, per una situazione di car-following ovvero quando si ha necessità di mantenere un gap costante con il veicolo che precede. Il Cruising Mode resta in funzione finché non vi sia alcun veicolo che preceda entro i 100 m, mentre al di sotto di questa soglia si attiva la modalità di Gap Regulation Mode.

- Cruising Mode

Il Cruising Mode è lo stesso per i due sistemi ACC e CACC, dal momento che non ci sono informazioni di tipo V2V, data l'assenza di veicoli che precedono all'interno del range prestabilito. In tal caso l'accelerazione $a_{i,k}$ da adottare per il time step k può essere calcolata come funzione della differenza tra l'attuale velocità $v_{i,k-1}$ e quella desiderata v_{des} (4.3). Il Cruising Mode è, di fatto, equivalente al normale Cruise Control.

$$a_{i,k} = k_1 * (v_{des} - v_{i,k-1}) \quad (4.3)$$

k_1 è un parametro utile a determinare il tasso di variazione della velocità sull'accelerazione, tale valore oscilla tra 0,3 a 0,4 s^{-1} in accordo con gli studi presenti in letteratura [17] e in questo caso è stato scelto il valore di 0,4 s^{-1} .

- Gap Regulation Mode

Nella modalità di controllo del gap l'accelerazione all'istante k è una funzione di trasferimento del secondo ordine, funzione della distanza tra i due veicoli e della differenza di velocità nel caso di veicoli equipaggiati con ACC (4.4) e anche in funzione dell'accelerazione del veicolo che precede nel caso di veicoli dotati di CACC (4.5). La differenza tra k e $k-1$ rappresenta il time step, ovvero la frequenza con la quale il simulatore aggiorna la posizione del veicolo e le relative caratteristiche di guida, tale valore è stato scelto pari a 0,5 secondi, che rappresenta il miglior compromesso tra una condizione ideale di reazione quasi istantanea (0,1 s ad esempio) e una condizione realistica (0,8 s) non realizzabile all'interno del simulatore.

Le funzioni di accelerazione sono le seguenti:

$$a_{i,k} = k_v * (v_{i-1,k-1} - v_{i,k-1}) + (k_s * e_{i,k}) \quad (4.4)$$

$$a_{i,k} = k_a * a_{i-1,k-1} + k_v * (v_{i-1,k-1} - v_{i,k-1}) + (k_s * e_{i,k}) \quad (4.5)$$

$e_{i,k}$ rappresenta il contributo di errore del time-gap e vale:

$$e_{i,k} = (x_{i-1,k-1} - x_{i,k-1} - v_{i,k-1} * t_{des}) \quad (4.6)$$

$-v_{i-1,k-1} - v_{i,k-1}$ è la differenza di velocità tra il veicolo leader e il veicolo equipaggiato con ACC/CACC.

$-x_{i-1,k-1} - x_{i,k-1}$ è la differenza tra la posizione del veicolo leader e il veicolo equipaggiato con ACC/CACC, ovvero la distanza netta tra i due veicoli.

$-t_{des}$ è il time gap desiderato, impostato a 1,4 s per l'ACC e 0,5 secondi per il CACC.

$-a_{i-1,k-1}$ è l'accelerazione del veicolo leader all'istante $k-1$.

$-k_a, k_v, k_s$ sono i parametri del modello. k_a è stato impostato pari a 1,0 in accordo con quanto previsto da VanderWerf et al. (2003) [6], mentre per i valori di k_v e k_s sono stati scelti i valori proposti da K.M.Malone (2004) [18], pari rispettivamente a 0,58 e 0,1.

Le due funzioni differiscono, dunque per il time-gap desiderato e per l'accelerazione del veicolo che precede. Il time-gap è impostato a 1,4 secondi per l'ACC e a 0,5 secondi per il CACC. I veicoli quindi potranno ridurre notevolmente

l'interdistanza grazie proprio alla possibilità di acquisire molte più informazioni tramite comunicazioni V2V, tra queste proprio l'accelerazione del veicolo che precede viene inserita all'interno del calcolo per quanto riguarda il CACC.

I sistemi ACC e CACC sono pensati per assicurare, tra le altre cose, un confort di guida per il conducente, per cui l'accelerazione massima è limitata a 2 m/s^2 e quella minima a -3 m/s^2 al fine di evitare fenomeni di frenate brusche o accelerazioni accidentali.

Per ogni time-step quindi il valore di accelerazione di riferimento sarà:

$$a_{ref} = \max(a_{min}; \min(a_{i,k}; a_{max})) \quad (4.7)$$

Tuttavia, in condizioni emergenziali, il veicolo, comunque dotato di frenata d'emergenza assistita (AEB: Autonomous Emergency Braking), o con pressione manuale del pedale del freno può assicurare valori di decelerazione inferiori al limite minimo imposto sopra, in modo da permettere una frenata tempestiva in situazioni di elevato rischio di collisione.

A tal proposito, è stato condotto un test al simulatore nel quale un veicolo dotato di CACC segue un veicolo leader anch'esso dotato di Cruise Control Cooperativo, il quale è soggetto a ripetute variazioni di velocità, fino ad arrestarsi da una velocità di 70 km/h , attuando l'AEB.

In figura 4.2, 4.3, 4.4, sono riportati i valori di velocità, accelerazione e time gap dei due veicoli.

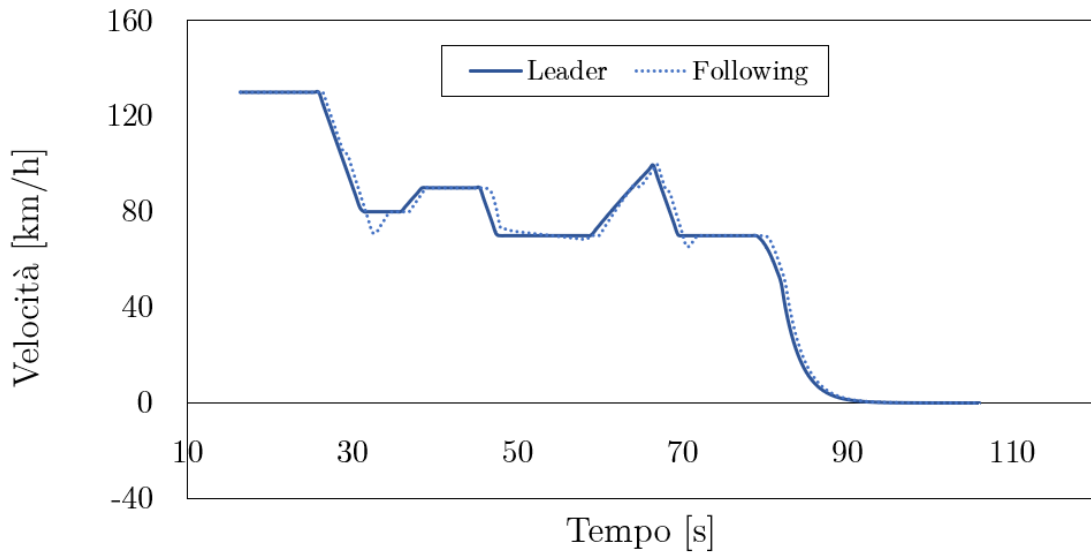


Figura 4.2: Profili di velocità di due veicoli dotati di CACC

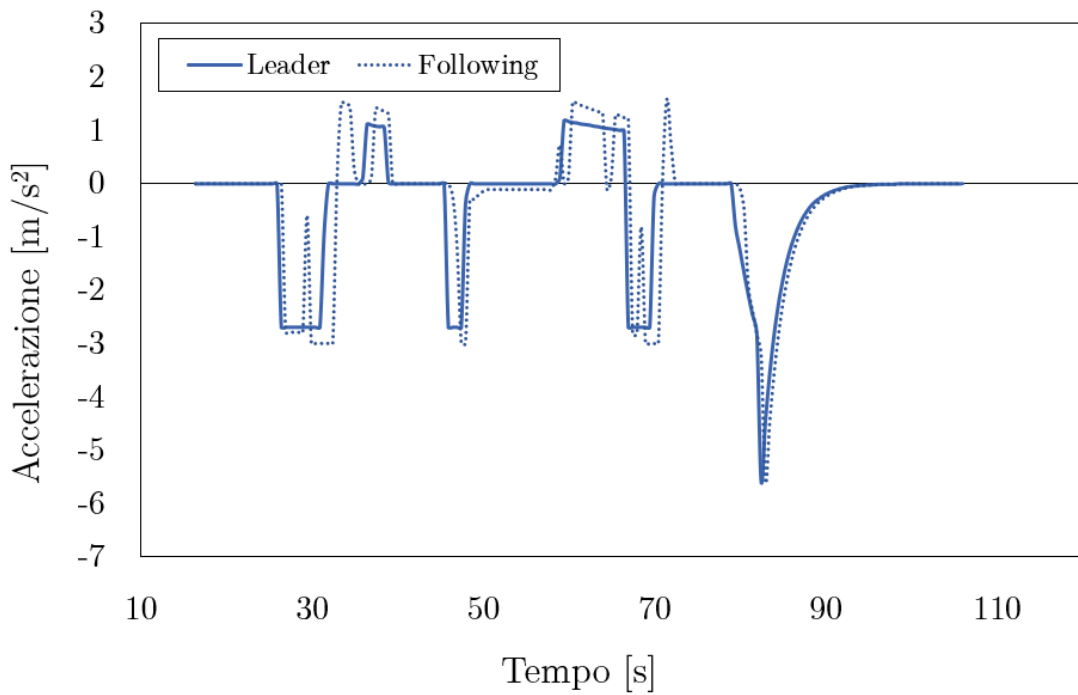


Figura 4.3: Profili di accelerazione di due veicoli dotati di CACC

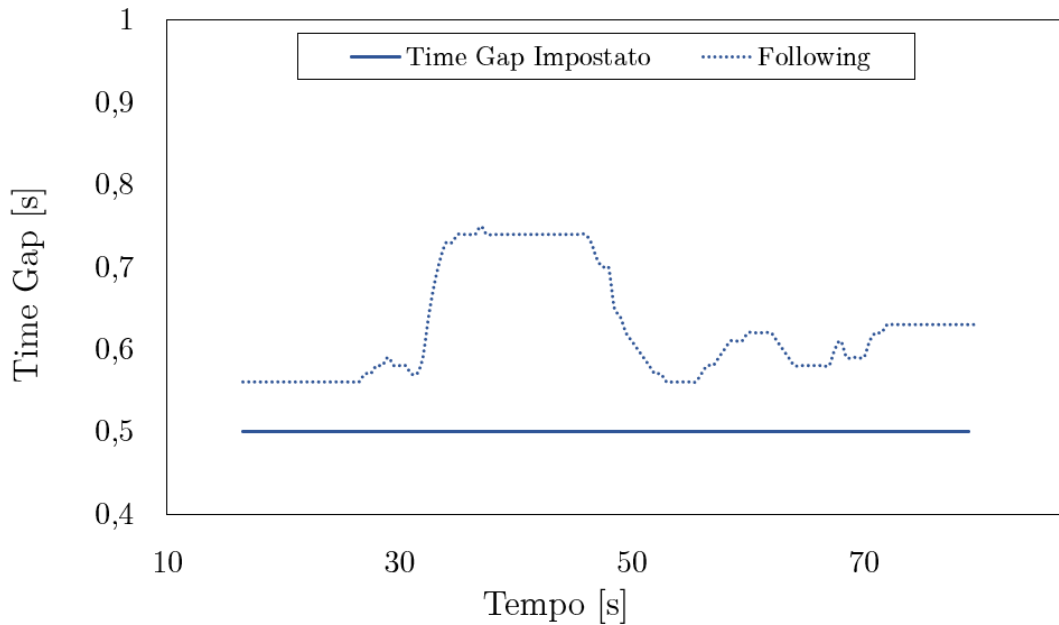


Figura 4.4: Time Gap veicolo dotato di CACC

La frenata avviene in condizioni di sicurezza, senza collisione; inoltre, per evitare fenomeni di incidenti a catena, i platoon di veicoli sono opportunamente distanziati tra loro, per evitare che fenomeni di impatto molto violenti possano propagarsi a più di un platoon.

Per testare, invece, le proprietà dei sistemi cooperativi nei confronti della regolare circolazione e verificare l'efficacia degli algoritmi impostati per le diverse tipologie di veicoli, sono state condotte tre simulazioni differenti in ognuna delle quali un platoon di 5 veicoli si avvicina a un leader più lento (4.5, 4.6). Di seguito sono descritte le caratteristiche di questi test:

| Test | Leader | Velocità Leader | Following | Velocità Following |
|------------|--------|-----------------|-----------|--------------------|
| Scenario 1 | CACC | 80 km/h | Manuali | 100 km/h |
| Scenario 2 | CACC | 80 km/h | ACC | 100 km/h |
| Scenario 3 | CACC | 80 km/h | CACC | 100 km/h |

Tabella 4.1: Input degli scenari di test

Sono stati estratti i profili di velocità e accelerazione dei vari scenari e dall'analisi di tali profili si conclude che i veicoli equipaggiati con il CACC rispondono

più velocemente nei confronti della decelerazione del veicolo che precede rispetto agli altri due scenari. I profili di velocità del diagramma riferito a soli veicoli connessi (4.11) sono quasi sovrapposti ad indicare che il tempo che intercorre tra un azione di accelerazione o decelerazione del veicolo che segue è considerevolmente minore rispetto agli altri due casi, come è visibile dai relativi profili di accelerazione (4.12).

Inoltre, le curve di tale diagramma sono funzioni quasi interamente regolari, ovvero non presentano punti spigolosi o discontinuità a suggerire un comportamento di guida confortevole senza brusche variazioni di velocità.

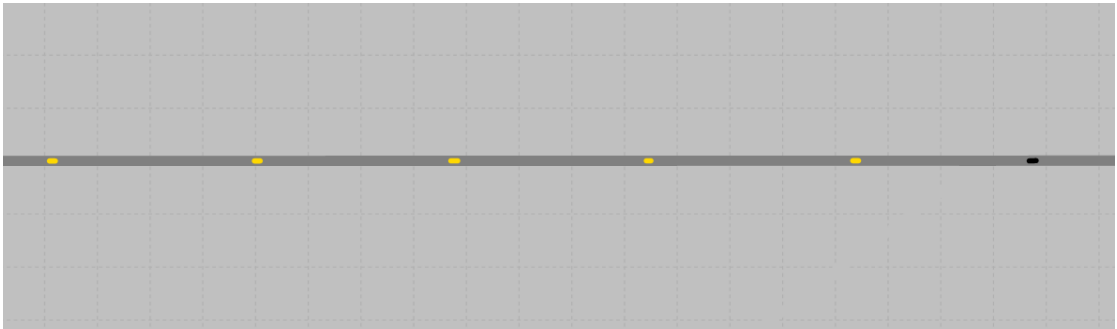


Figura 4.5: Platoon di veicoli a guida manuale e leader più lento

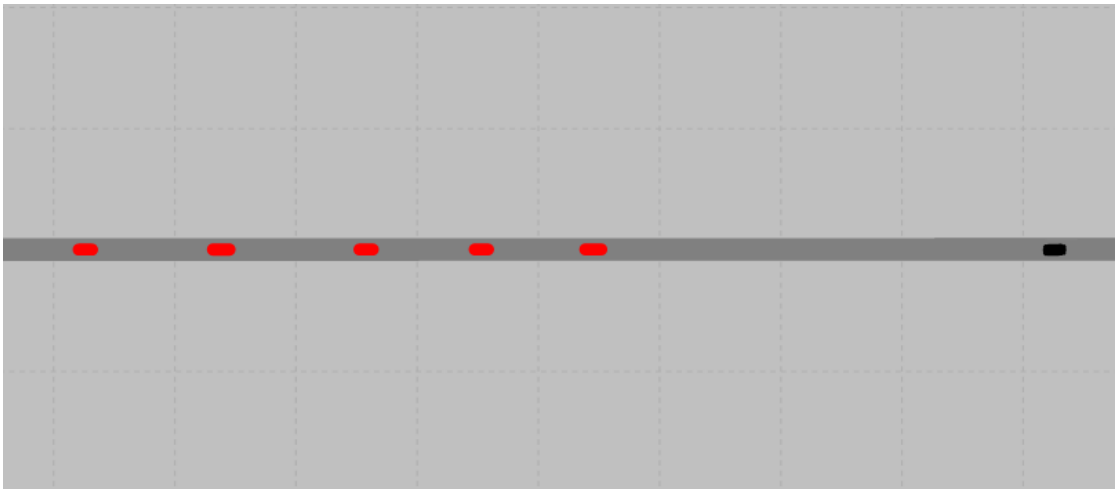


Figura 4.6: Platoon di veicoli CACC e leader più lento

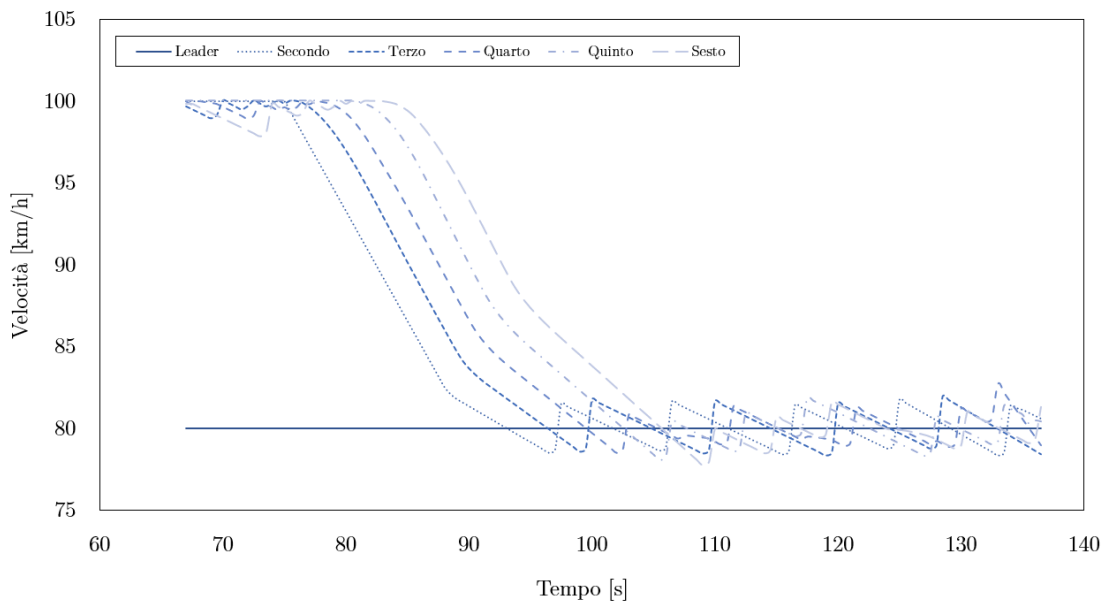


Figura 4.7: Velocità del platoon di veicoli a guida manuale

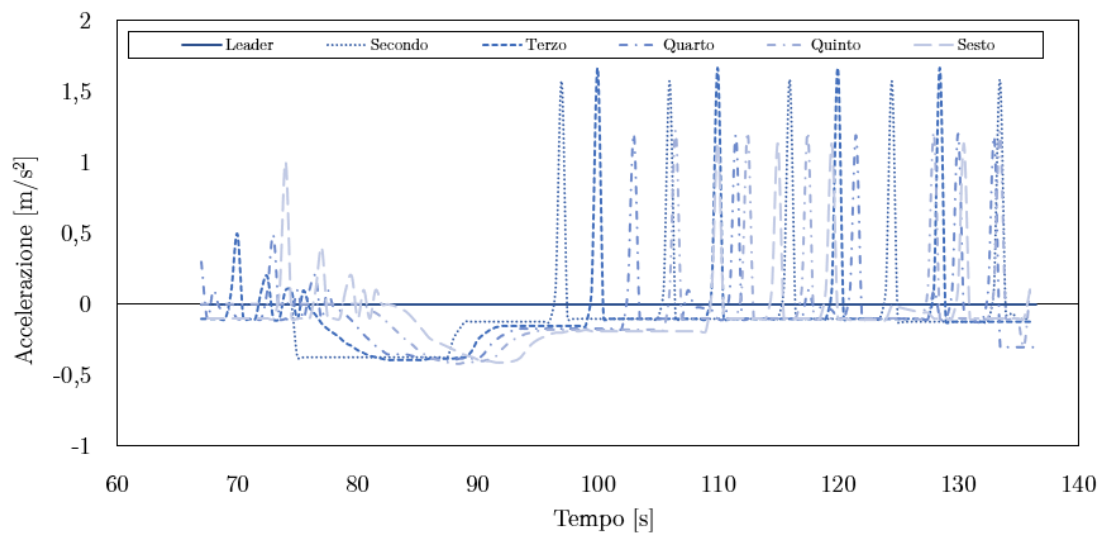


Figura 4.8: Accelerazione del platoon di veicoli a guida manuale

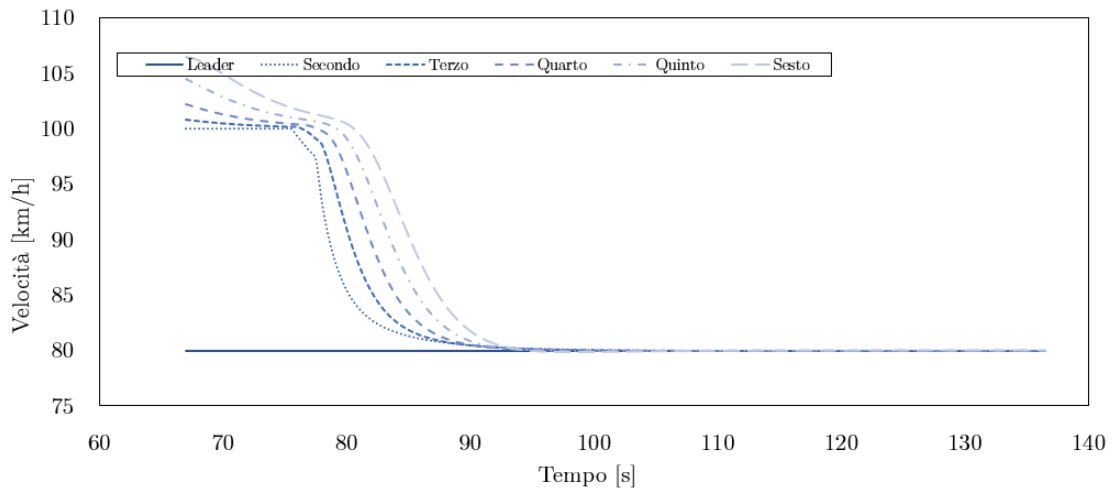


Figura 4.9: Velocità del platoon di veicoli dotati di ACC

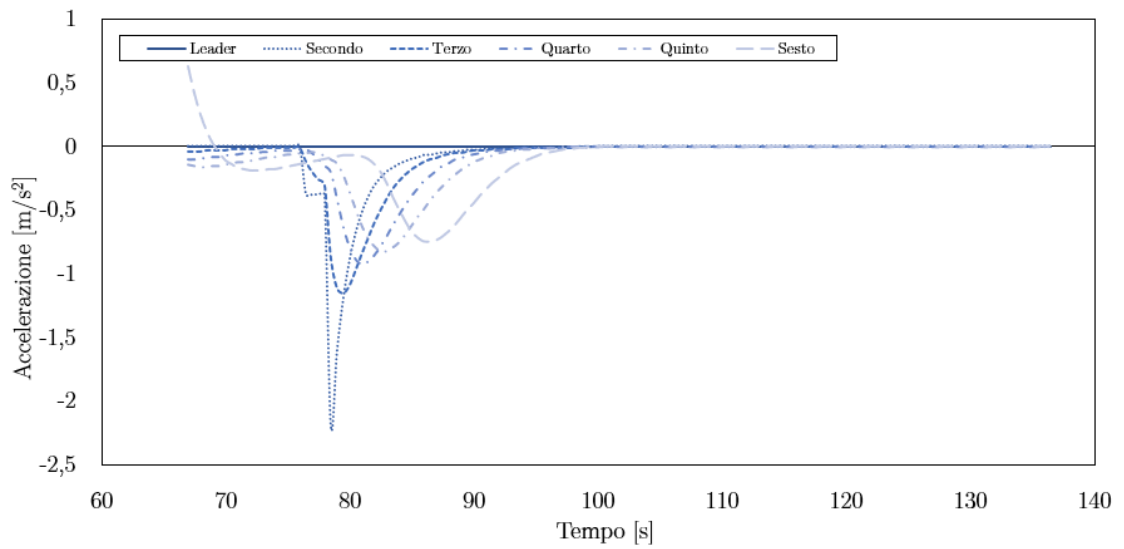


Figura 4.10: Accelerazione del platoon di veicoli dotati di ACC

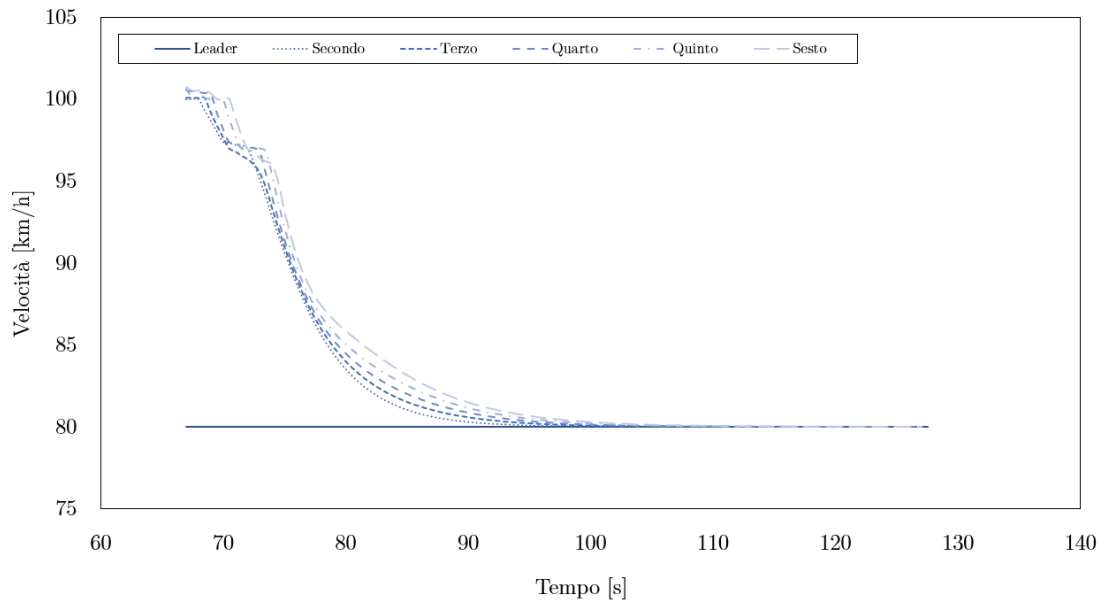


Figura 4.11: Velocità del platoon di veicoli dotati di CACC

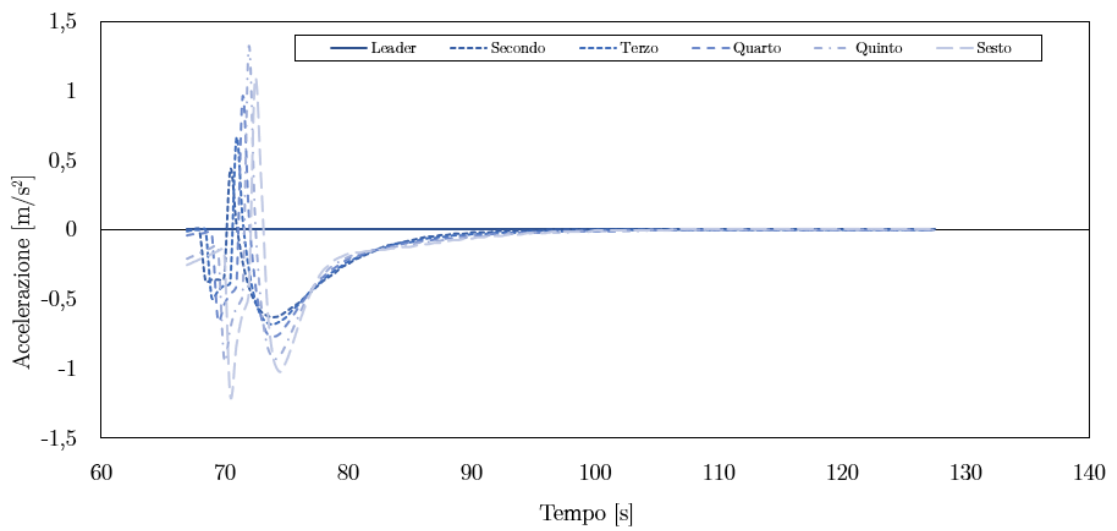


Figura 4.12: Accelerazione del platoon di veicoli dotati di CACC

4.2.2 Test all'interno di un Anello

È stato condotto un ulteriore test per la valutazione dell'efficacia del CACC.

In questo caso una flotta di 10 veicoli è stata inserita in una corsia stradale di forma anulare di 170 m di circonferenza. L'obiettivo è valutare la stabilità del flusso di traffico confrontando due scenari; un primo scenario con 10 veicoli a guida tradizionale e un secondo con 10 veicoli equipaggiati con CACC.

In entrambi i casi i veicoli partono da fermi in un punto prestabilito dell'anello indicato dalla linea rossa in Figura 4.13 per poi partire attestandosi alla velocità desiderata.

In ogni caso è stato imposto un valore di 50 km/h come limite di velocità per il leader.

Per indurre una perturbazione interna, al secondo 37 il veicolo Leader riduce la sua velocità, passando da 50 km/h a 40 km/h. I valori di input relativi all'accelerazione/decelerazione massima sono i medesimi descritti sopra.

Nella figura seguente sono indicati a destra i veicoli a guida manuale e a sinistra quelli dotati di CACC. Tutti i veicoli sono distanziati di 2 metri in condizioni iniziali.

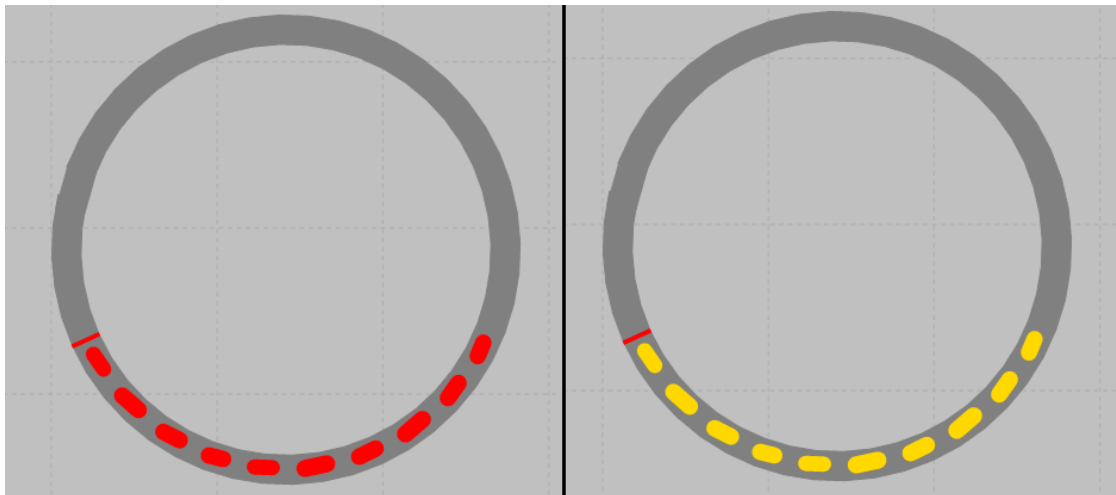


Figura 4.13: Schema iniziale per il test su strada anulare

La simulazione è durata 120 secondi e i dati sono stati ottenuti direttamente dai veicoli ogni 0,5 s. Sono stati acquisiti i valori di velocità, time gap e clearance come indicatori che evidenziano al meglio eventuali fenomeni di perturbazioni nel flusso di traffico.

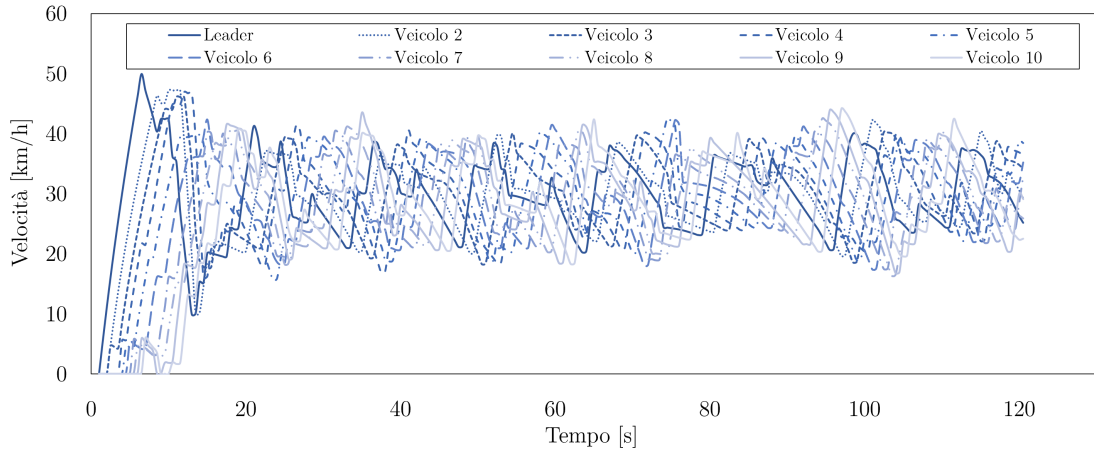


Figura 4.14: Profili di velocità dei veicoli a guida manuale all'interno dell'anello

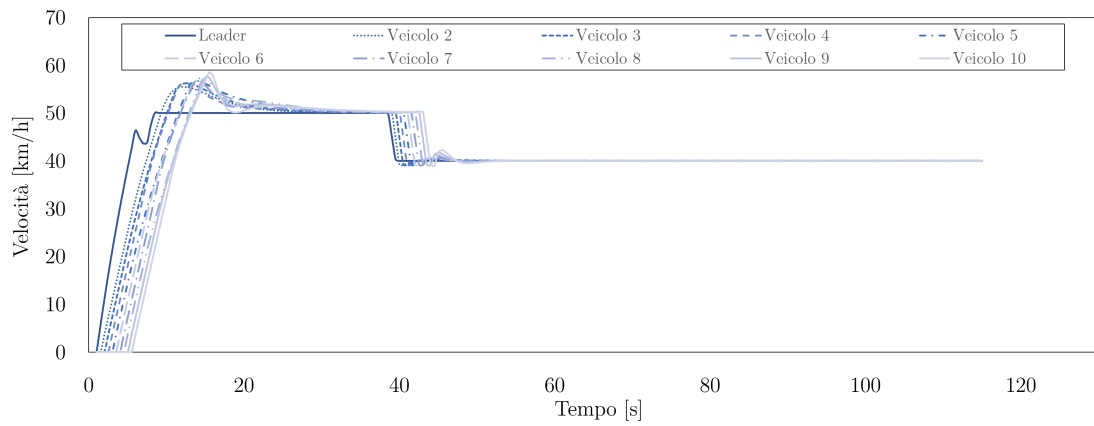


Figura 4.15: Profili di velocità dei veicoli dotati di CACC all'interno dell'anello

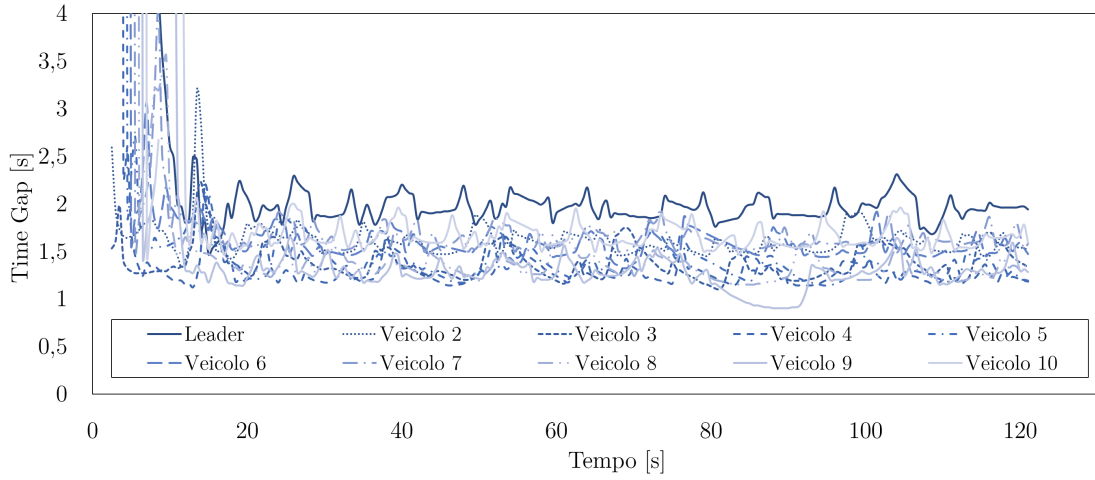


Figura 4.16: Time gap dei veicoli a guida manuale all'interno dell'anello

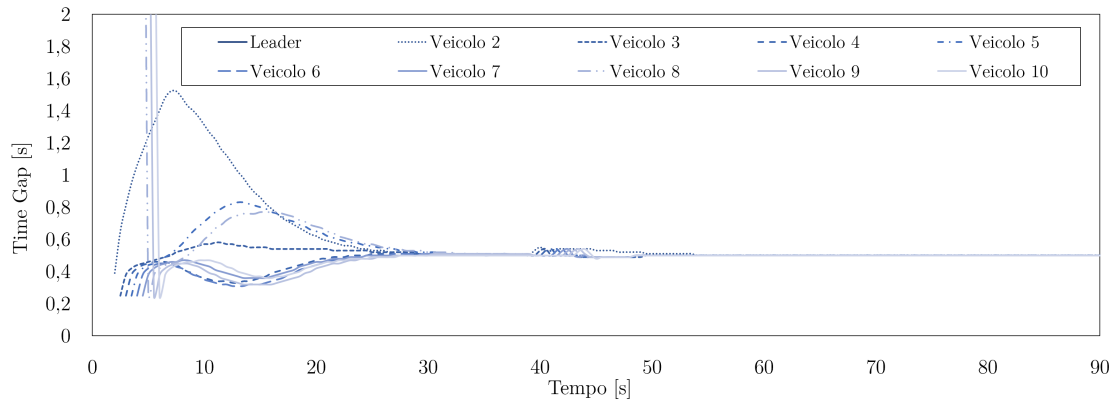


Figura 4.17: Time gap dei veicoli dotati di CACC all'interno dell'anello

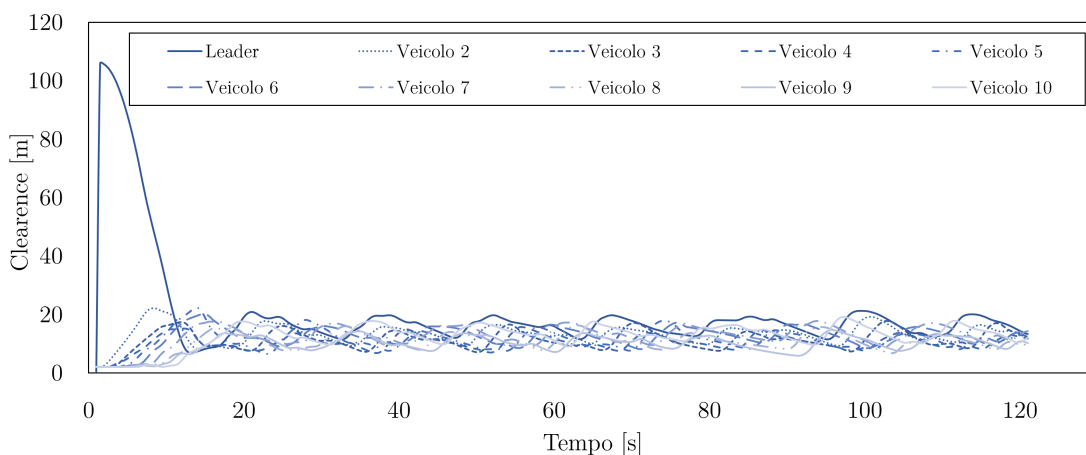


Figura 4.18: Clearance dei veicoli a guida manuale all'interno dell'anello

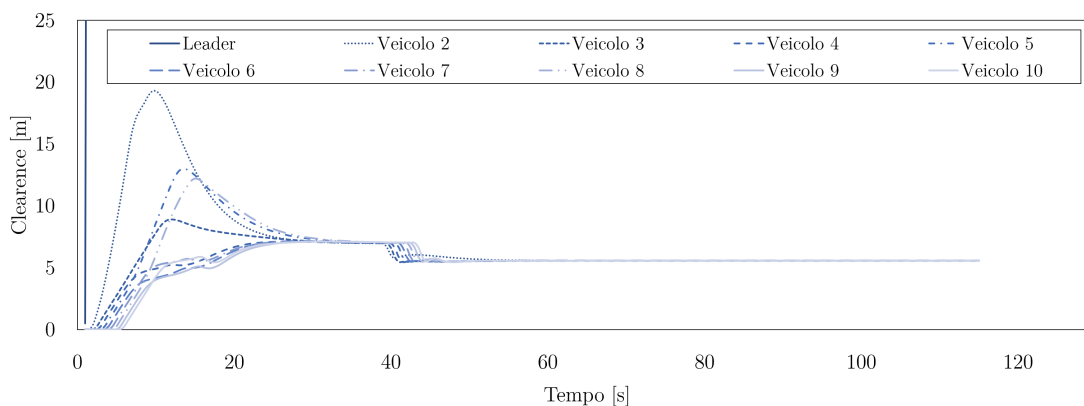


Figura 4.19: Clearance dei veicoli dotati di CACC all'interno dell'anello

Dai risultati delle simulazioni si può notare come nel caso di veicoli a guida manuale, senza indurre alcuna perturbazione dall'esterno, si innesti una fluttuazione delle velocità all'inizio della simulazione (Figura 4.14) che genera un'onda di stop & go che si propaga a monte per tutta la durata della simulazione. I conducenti non riescono a regolare la velocità alla perfezione e sovente tendono a regolare continuamente la velocità in seguito alla decisione di accelerare o decelerare in risposta alle condizioni del traffico. Ciò è dovuto principalmente a errori di percezione e stima. Gli errori di percezione sono errori di valutazione della distanza o della velocità del veicolo che precede. Per stima si intende il processo di valutazione degli effetti delle azioni di altri veicoli, ovvero quando un veicolo leader inizia a frenare, il follower deve determinare la quantità di decelerazione richiesta. Ciò porta a sovrastimare o sottostimare il valore di accelerazione o decelerazione

realmente necessario tale da mantenere un time gap pressoché costante. Tale comportamento di regolazione della velocità continuo è evidente in Figura 4.14 dove i profili di velocità sono assimilabili a equazioni d'onda. Il veicolo tenderà ad agire in base alle soglie descritte al capitolo 4.3.

Nel secondo caso, invece, con tutti i veicoli equipaggiati con il CACC, la perturbazione svanisce rapidamente con un evidente livellamento sulle oscillazioni, per effetto della comunicazione interveicolare.

Ciò è dovuto al fatto che il veicolo riceve i valori di accelerazione, velocità e distanza dal veicolo che lo precede e di conseguenza calcola continuamente il valore di accelerazione effettivamente necessario per mantenere un time gap predefinito ed uguale per tutti i veicoli coinvolti.

Il valore di accelerazione calcolato per ogni time step è indicato dall'equazione 4.5, al capitolo 4.2.1. Dalle figure 4.15 e 4.17 si osserva che i veicoli si assestano sulla velocità limite imposta dal leader in circa 10 secondi. Dalla Figura 4.17 si osserva che la variazione di velocità al secondo 37 non influenza significativamente il time gap dei veicoli in fase iniziale, a testimonianza di una rapida risposta nei confronti delle azioni di guida dei veicoli che precedono.

Il Cooperative Adaptive Cruise Control migliora, dunque, la stabilità del traffico. In conclusione, il sistema di Cruise Control cooperativo rappresenta un'evoluzione del normale cruise control adattivo con l'obiettivo di ridurre l'interdistanza veicolare, migliorando la capacità della strada, incrementare l'efficienza aerodinamica riducendo i consumi e le emissioni, promuovere la formazione di platoon e migliorare la stabilità del flusso di traffico.

Rispetto al sistema di rilevamento della distanza basato su una sensoristica LIDAR presente nei veicoli dotati di ACC, il Cruise Control cooperativo implementa anche una comunicazione wireless di tipo V2V, idonea a condividere informazioni sullo stato di marcia del veicolo in tempo reale a meno di un tempo di latenza (tipicamente dell'ordine di 0,1 s, ma test su strada hanno evidenziato come il ritardo delle comunicazioni wireless sia addirittura trascurabile, verificando una percentuale di perdita di pacchetti dati insignificante [19]). La comunicazione è basata su un sistema dedicato a corto raggio (DSRC) tipicamente a 5.9 GHz, un sistema di posizionamento GPS implementati in un'unità WSU (Wireless Safety Unity). I controlli sono implementati mediante specifici software inclusi in centraline che ricevono i dati sia dalla WSU che dal Controller Area Network (CAN) del veicolo.

4.2.3 Condizioni del Modello di Guida CACC

Una delle principali funzionalità del Cruise Control Cooperativo è la possibilità di formare platoon per mezzo della comunicazione tra veicoli. Per tenere in conto di questa potenzialità, i veicoli equipaggiati con il CACC, rispetto a gli altri tipi di veicoli, sono simulati in tre diversi stati di guida elencati di seguito:

- Stato di Guida Normale

Questo è il caso in cui il veicolo è in modalità di controllo della velocità, ovvero non ha veicoli a valle nel raggio di comunicazione possibile e procede come un normale Cruise Control a una velocità predefinita. In questo caso, il veicolo è di colore nero nella simulazione.

- Stato di Leader del Platoon

In questo caso, il veicolo viene riconosciuto come leader del platoon. Esso procederà con il CC in assenza di veicoli a valle o con l'ACC in presenza di veicoli che lo precedano. Il veicolo leader sarà di colore nero per riconoscere all'interno della simulazione la formazione del platoon.

Il veicolo viene riconosciuto come leader quando chi lo precede non è equipaggiato con il CACC e quando riconosce un veicolo che lo segue dotato di CACC entro i 100 metri di distanza, in tal caso il veicolo notificherà a monte di essere il leader del platoon, ciò permetterà a chi lo segue di agganciarsi allo stesso.

- Stato di Unione al Platoon

Tale stato si verifica quando il veicolo si trova nel processo di unione al platoon o è già all'interno di esso. Il colore sarà rosso. Quando sulla stessa corsia viene riconosciuto un veicolo a valle equipaggiato con il CACC e segnalato come leader, comincia la manovra di aggregazione al platoon, imponendo l'accelerazione (4.5), riducendo quindi il time-gap a 0,5 s.

4.2.4 Algoritmi per i comportamenti di guida automatica

All'interno di un simulatore, gli algoritmi di guida che è possibile implementare fanno riferimento a due principali categorie: Il *Car-following model*, ovvero il modello di inseguimento e il *Lane-change model*, ovvero il modello di scelta di cambio corsia. Eventualmente, si possono implementare altri modelli, come quello necessario alla formazione del platoon.

Il modello di *Car-following* è responsabile del controllo longitudinale dei veicoli. Fornisce un nuovo valore di accelerazione/decelerazione per ogni time step. Il modello di guida utilizzato è stato ampiamente descritto al capitolo 4.2.1.

Questo modello ha fondamentalmente due modalità. Una è la modalità di controllo della velocità (Cruising Mode) e l'altra è la modalità di controllo del gap (Gap Regulation Mode). La modalità di controllo della velocità si attiva quando l'auto è a flusso libero e non c'è un leader o, in alternativa, quando il leader è a più di 100 m di distanza. In questa modalità l'obiettivo del veicolo è raggiungere la velocità desiderata. La modalità di regolazione del gap, invece, viene attivata quando la distanza tra il veicolo ego e il leader è inferiore a 100 metri. Il veicolo tiene conto della distanza tra esso e quello che precede, la loro velocità relativa e formula un valore di accelerazione in accordo alla 4.5.

La Figura 4.20 rappresenta il diagramma di flusso dell'algoritmo utilizzato nella DLL del modello di guida per i veicoli equipaggiati con il CACC. Questo algoritmo è richiamato in ogni time step per ogni veicolo dotato di CACC nella rete per calcolare l'accelerazione per quel time-step.

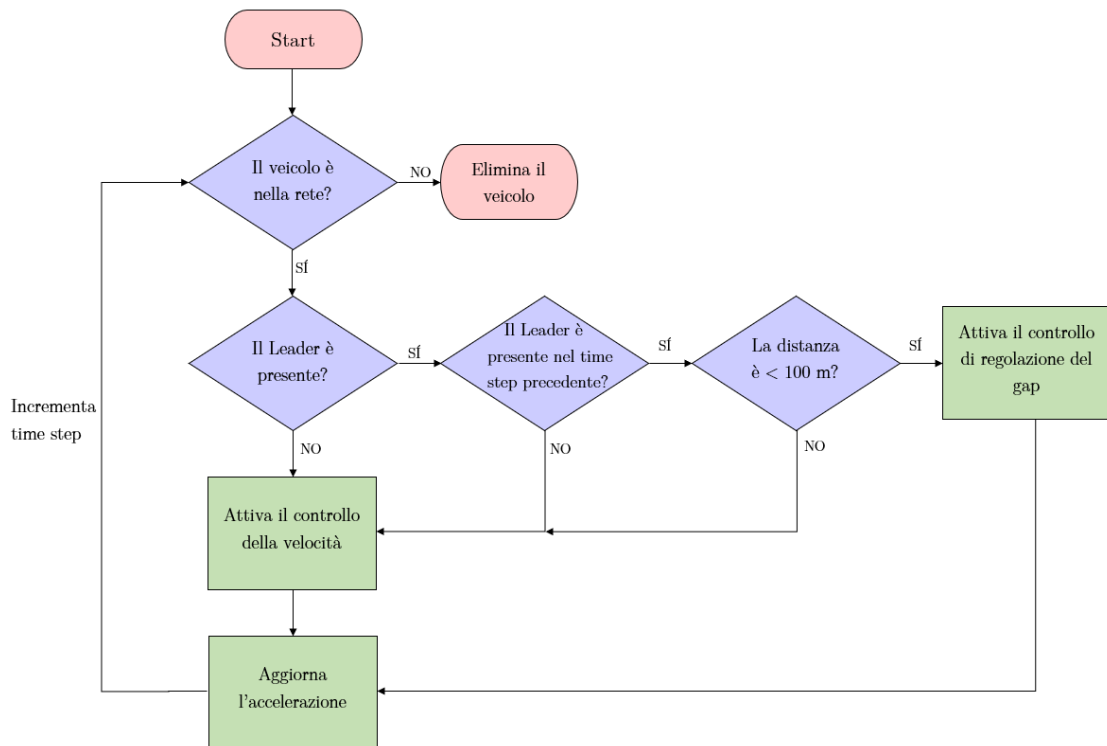


Figura 4.20: Diagramma di flusso per CACC

Sebbene si possa implementare un modello alternativo di cambio corsia, questo algoritmo è stato lasciato alle impostazioni interne del simulatore. I veicoli dotati di CACC, invece, seguono delle istruzioni precise per il cambio corsia al fine di incentivare la formazione di platoon.

Se un veicolo CACC non è nella condizione di Leader e/o non ha veicoli a valle sulla stessa corsia entro il range di funzionamento, controlla il tipo di veicolo a sinistra di esso e davanti e dietro.

- Se il veicolo a sinistra e davanti ad esso non è un veicolo CACC, continua a guidare con il modello di comportamento di guida definito dal sistema e attivo fino a quel momento.

- Se il veicolo a sinistra e davanti ad esso è un veicolo CACC, vorrebbe unirsi a un platoon:

Se il tipo di veicolo a sinistra e dietro di esso non è un veicolo CACC, cambia corsia e si unisce al platoon dietro il veicolo CACC. Se il tipo di veicolo a sinistra e dietro è un veicolo CACC, attende fino alla fine del platoon per cambiare corsia e unirsi.

Se un platoon è già formato e un veicolo vorrebbe uscire dal plotone, quel veicolo avvisa il veicolo in testa e poi inizia la manovra di uscita. Per la formazione del platoon la sequenza di comandi utili per la sua formazione è definita di seguito.

Tale algoritmo segue una impostazione semplice definita nella figura 4.21 e basata su alcuni concetti basilari:

- Il platoon può essere formato solo da veicoli CACC per un massimo di cinque veicoli a valle del Leader.
- La distanza tra due veicoli deve essere almeno 100 metri per permettere il riconoscimento del veicolo CACC e dunque la formazione del platoon.
- Se il numero di veicoli all'interno del platoon è maggiore di 6 (leader compreso) il platoon si interrompe al sesto veicolo che dunque diventa leader procedendo come un normale ACC.

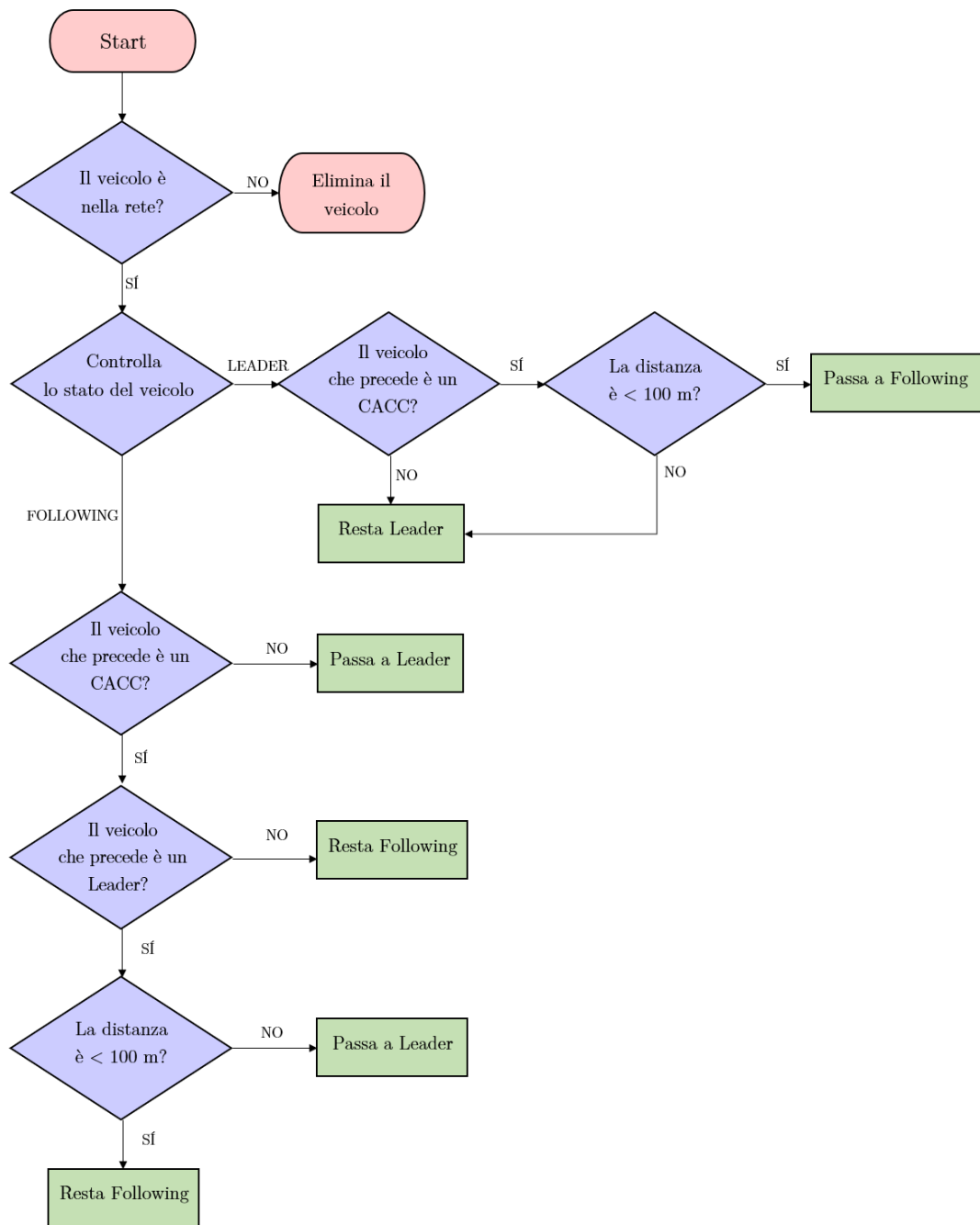


Figura 4.21: Diagramma di flusso per la formazione del platoon

4.3 Parametri del modello di guida interno

I veicoli a guida tradizionale sono stati simulati mediante una configurazione dei parametri interni del simulatore, senza l'ausilio di librerie esterne.

VISSIM utilizza un modello psicofisico di car-following basato sul modello sviluppato da Wiedemann (1974), che definisce le soglie di percezione del guidatore. C'è un altro modello di car-following, il Wiedemann 99, che è per molti versi simile al Wiedemann 74, ma pensato per modellare meglio il traffico autostradale. Le soglie di questi modelli sono regolabili dall'utente.

Il modello Wiedemann presuppone che un conducente possa essere in quattro diversi regimi di guida: following, guida libera, avvicinamento o frenata. Questi regimi sono definiti da soglie (o punti di azione) che rappresentano i punti in cui un conducente cambia il suo comportamento di guida. Le soglie e i regimi per il modello Wiedemann 74 sono spiegati di seguito con l'ausilio della Figura 4.22 e della Tabella 4.2

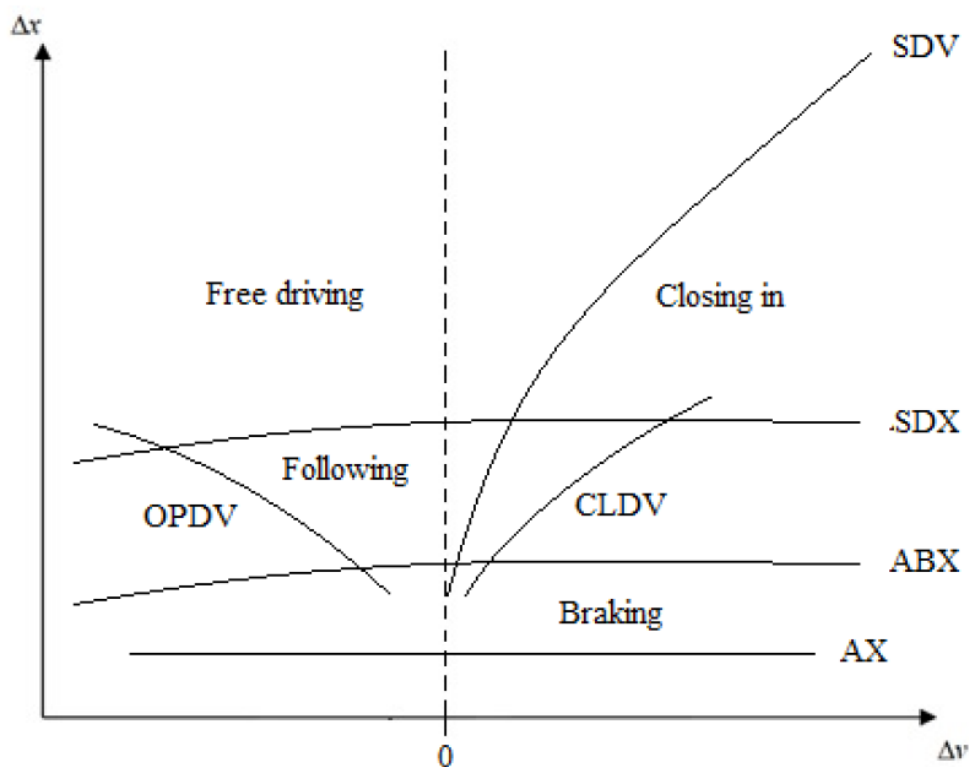


Figura 4.22: Rappresentazione grafica del modello di Wiedemann

| | |
|------|--|
| AX | Distanza desiderata tra due veicoli fermi |
| ABX | Distanza minima di inseguimento tra due veicoli che viaggiano circa alla stessa velocità |
| SDX | Distanza massima di inseguimento tra due veicoli che viaggiano circa alla stessa velocità |
| SDV | Distanza rispetto alla quale il conducente realizza di aver raggiunto la distanza desiderata rispetto al veicolo che precede |
| CLDV | Soglia della percezione per riconoscere le piccole differenze di velocità in breve tempo ma con distanze che diminuiscono |
| OPDV | Soglia della percezione per riconoscere le grandi differenze di velocità in breve tempo ma con distanze che aumentano |

Tabella 4.2: Parametri del modello di Wiedemann

La descrizione dei regimi di guida, definiti dalle soglie della tabella 4.2, può essere riassunta nel seguente modo:

- Following

Un conducente in questo regime è attento a mantenere una distanza di sicurezza relativamente costante rispetto al veicolo che precede. Quando un veicolo entra nel regime di following, attraversando la soglia OPDV o SDX, gli viene assegnato un valore positivo di accelerazione. Se l'SDV o l'ABX viene superato, al conducente viene, invece, assegnato un valore negativo di accelerazione.

- Guida libera

Nel regime di guida libera, il conducente non è limitato da alcun veicolo leader. Quindi, utilizza il valore di accelerazione massima per raggiungere la velocità desiderata.

- Avvicinamento

Il regime di avvicinamento (closing in) descrive lo scenario nel quale il conducente deve decelerare per evitare la collisione con un veicolo più lento che precede, cioè quando la soglia SDV è stata superata. La decelerazione del veicolo viene regolata in modo da essere pari alla velocità del leader nel momento in cui viene raggiunta la distanza di sicurezza desiderata.

- Frenata

Se il veicolo che segue è più vicino al veicolo leader rispetto alla distanza di sicurezza desiderata, il conducente sarà in regime di frenata. Poiché la distanza tra i veicoli è troppo bassa, il conducente del veicolo che segue decelera per evitare la collisione.

I parametri disponibili per modellare il comportamento di car-following in VISSIM sono esposti in Figura 4.23 e spiegati in tabella 4.3.

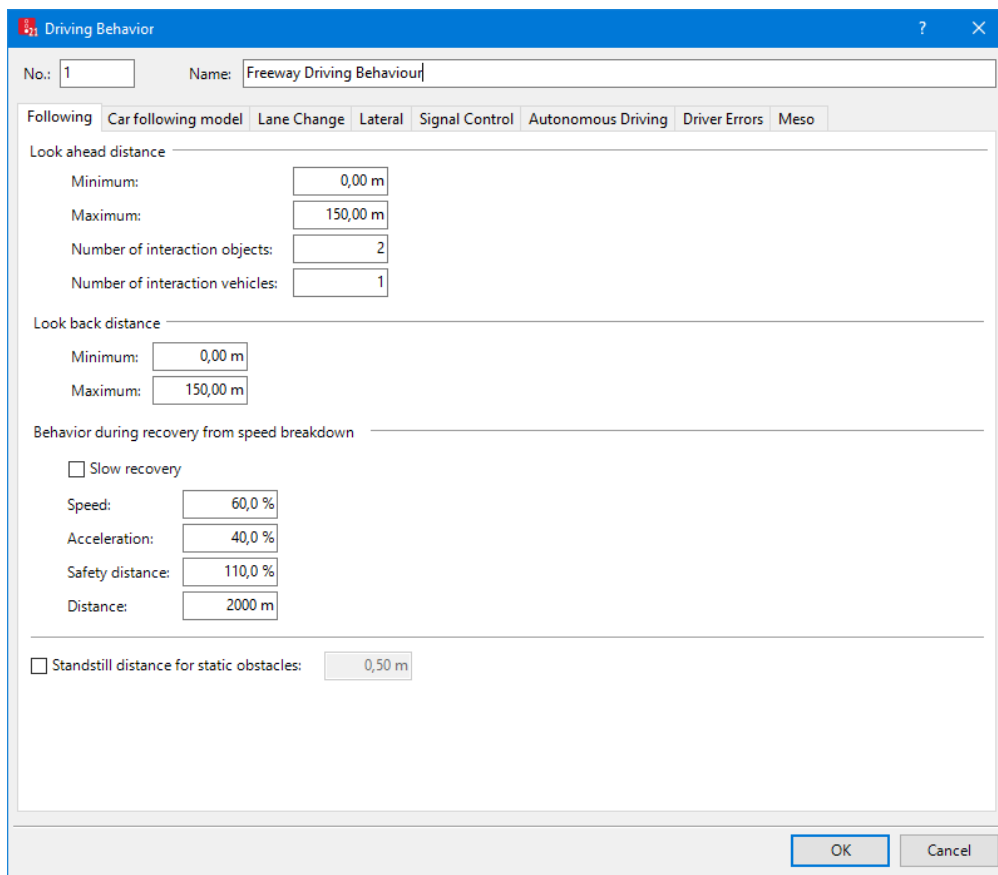


Figura 4.23: Schermata dei parametri del comportamento di guida di VISSIM

| | |
|--|---|
| Look Ahead Distance | La distanza di visibilità entro la quale il conducente può vedere ed essere ancora in grado di reagire ad azioni compiute dai conducenti circostanti. Number of interaction objects: Il numero degli oggetti della rete stradale (veicoli compresi, number of interaction vehicles) che il conducente può percepire e rispondere ad eventuali azioni. |
| Look back distance | Equivalente alla Look Ahead Distance, ma riferito ai veicoli che seguono |
| Behaviour during recovery from speed breakdown | Condizioni per la ripartenza dopo una situazione di veicolo fermo |
| Standstill distance for static obstacles | Distanza che il conducente mantiene, da fermo, rispetto ad oggetti della rete, quali semafori, segnali stradali orizzontali ecc. |

Tabella 4.3: Parametri di Car-Following in VISSIM

Come detto, ci sono due modelli di guida disponibili in VISSIM: Wiedemann 74 e Wiedemann 99. I modelli implementati differiscono leggermente dal modello Wiedemann presentato finora. La differenza principale è che i modelli in VISSIM cercano di creare una popolazione di conducenti quanto più diversificata possibile, dove ad esempio la stima della distanza o della velocità desiderata varia tra i singoli conducenti. Per creare un modello che rifletta un comportamento così eterogeneo, Higgs (2011) [20] spiega che la capacità di percezione di un conducente e il comportamento in situazioni di rischio sono simulate in VISSIM aggiungendo valori casuali a ciascuna delle soglie presentate nella tabella 4.2.

Il modello di Wiedemann 74 tiene conto di tre parametri fondamentali:

- Average standstill distance

(ax) Rappresenta la distanza media desiderata di arresto. Corrisponde al valore di AX di Tabella 4.2. Il valore standard è 2 m.

- Additive part of safety distance

(bx_{add}) Incluso nel calcolo della distanza di sicurezza d , rappresenta la parte costante. Il valore standard è 2 m.

- Multiplicative part of desired safety distance

(bx_{mult}) Incluso nel calcolo della distanza di sicurezza d , rappresenta la parte moltiplicativa. Il valore standard è 3.

La distanza di sicurezza minima desiderata d viene calcolata come:

$$d = ax + bx \quad (4.8)$$

dove:

$$bx = bx_{add} + bx_{mult} * z * \sqrt{v} \quad (4.9)$$

v è la velocità del veicolo in m/s e z è un valore compreso tra 0 e 1, normalmente distribuito intorno a 0,5 con deviazione standard di 0,15.

Di seguito, sono proposti alcuni valori di d in funzione della velocità, come riportato da B.Higgs et al. [21]:

| | | | | | | | | | |
|---|---------------|---------------|---------------|---------------|---------------|---------------|---------------|--------------------|---------------------|
| | 20-30 km/h | 30-40 km/h | 40-50 km/h | 50-60 km/h | 60-70 km/h | 70-80 km/h | 80-90 km/h | 90- 100 km/h | 100- 110 km/h |
| d | 13,90 | 16,58 | 18,71 | 19,20 | 14,92 | 15,04 | 15,89 | 17,07 | 20,11 |

Il modello Wiedemann 99, invece, si basa sulla definizione di 10 parametri descritti in tabella 4.4:

| Parametro | Unità | Descrizione |
|-----------|-------|--|
| CC0 | m | Distanza di arresto: Distanza di arresto media desiderata fra due veicoli. |
| CC1 | s | Distribuzione del Time gap. Distribuzione di tempo dalla quale è estratto il time gap in secondi che un conducente vuole mantenere rispetto a un veicolo che lo precede in aggiunta alla normale distanza di arresto. Alla velocità v , la distanza di sicurezza desiderata è calcolata come: $dx_{safe} = CC0 + CC1 * v$ |
| CC2 | m | Limita la variazione di distanza (oscillazione longitudinale), ovvero la differenza fra la distanza di sicurezza reale e la distanza di sicurezza desiderata. Se ad esempio questo valore è impostato a 10 m, l'accodamento produce distanze da dx_{safe} e $dx_{safe} + 10m$. Il valore standard è 4 m, da cui deriva un comportamento di accodamento piuttosto stabile. |

| | | |
|-----|---------|--|
| CC3 | s | Controlla l'inizio del processo di decelerazione, ovvero il numero di secondi trascorsi prima del raggiungimento della distanza di sicurezza. In questa fase, il guidatore realizza la presenza di un veicolo che lo precede a velocità inferiore. |
| CC4 | m/s | Definisce il valore negativo della differenza di velocità, durante il processo di accodamento. Se i valori si mantengono bassi, ne deriva una reazione più sensibile del guidatore all'accelerazione o decelerazione del veicolo che precede. |
| CC5 | m/s | Definisce il valore positivo della differenza di velocità, durante il processo di accodamento. Anche in questo caso, se i valori si mantengono bassi, ne deriva una reazione più sensibile del guidatore all'accelerazione o decelerazione del veicolo che precede. |
| CC6 | 1/(m/s) | Influenza della distanza sull'oscillazione della velocità durante il processo di accodamento: Valore 0: L'oscillazione della velocità è indipendente dalla distanza. Valori > 0: Portano ad una maggiore oscillazione della velocità con aumento della distanza. |
| CC7 | m/s^2 | Oscillazione dell'accelerazione durante il processo di following. |
| CC8 | m/s^2 | Accelerazione desiderata quando il veicolo parte da fermo (è limitata dall'accelerazione massima definita all'interno delle curve di accelerazione). |
| CC9 | m/s^2 | Accelerazione desiderata a 80 km/h (è limitata dall'accelerazione massima delle curve di accelerazione). |

Tabella 4.4: Parametri del modello Wiedemann 99 in VISSIM

Capitolo 5

Simulazione degli scenari di Traffico

In questo capitolo sono descritte le micro-simulazioni dell'ambiente di traffico connesso e le procedure seguite per la costruzione del modello di guida cooperativo e adattivo.

Quando si parla di micro-simulazioni si parla di simulazioni dei movimenti individuali dei veicoli per ogni secondo o frazione di secondo con lo scopo di valutare le caratteristiche del traffico [22]. I metodi e le procedure adottate sono descritti di seguito.

5.1 Scenario 1

Il primo scenario creato per valutare l'effetto del CACC sulle prestazioni del traffico e sulle caratteristiche della rete si riferisce ad una sezione autostradale in condizioni di flusso libero; priva, cioè, di interferenze esterne, quali rampe di ingresso o uscita, caselli, bottleneck. Sono stati valutati vari fattori come il flusso veicolare e il livello di penetrazione nel mercato (MPR) dei veicoli connessi. I fattori sono stati combinati tra loro per ottenere diversi scenari. Questi scenari sono stati utilizzati per valutare l'impatto dei veicoli cooperativi nei confronti di altri veicoli nella rete e su diverse misure delle prestazioni della strada come la stabilità del traffico e la sicurezza. In questo capitolo viene descritta la procedura di settaggio del simulatore, gli input della simulazione, il contesto in cui viene riprodotta e la procedura di raccolta dei dati. Nel capitolo successivo sono analizzati i risultati.

5.1.1 Input della Simulazione

Questa simulazione è stata effettuata su una sezione autostradale di 4 km. Tutte le rampe di entrata e di uscita lungo l'autostrada non vengono utilizzate per valutare la variazione delle caratteristiche del traffico in assenza di interferenze esterne o interne e capire come i veicoli cooperativi possano coadiuvare i veicoli non connessi per migliorare la qualità della circolazione.

Specialmente in casi di flusso elevato, anche in condizioni di sezione libera, possono presentarsi degli ingorghi. Tale fenomeno, noto come Phantom Traffic Jam o Ingorgo Stradale fantasma, è spiegato dal fatto che in presenza di molti veicoli sulla strada che viaggiano a velocità costanti e a distanze uguali e relativamente basse, la minima perturbazione di un veicolo nella velocità, ad esempio, si ripercuote sui veicoli che seguono, generando un'onda di stop and go che si propaga anche per molti km a monte. L'autostrada ha 3 corsie unidirezionate e ha come limite di velocità 130 km/h. Il tempo di viaggio viene raccolto ogni qual volta il veicolo lascia la rete, mentre i punti di raccolta dati sono collocati ogni 500 m.

Data la natura stocastica del traffico, per ogni simulazione, sono state effettuate 5 ripetizioni o simulation runs di 60 minuti ciascuna con incremento del seme casuale di 20. Lo step di simulazione, corrispondente con il reaction time è stato scelto pari a 0,5 s. Dalla valutazione dei dati sono stati esclusi i primi 1000 m influenzati dall'input dei veicoli e i primi 5 minuti, inteso come periodo di warm-up.

Il numero di cores, necessario per allocare e permettere il funzionamento simultaneo di più librerie dinamiche, è stato impostato a 3. In questo scenario, dalla combinazione dei diversi fattori descritti in seguito, sono state effettuate 54 simulazioni (270 simulation runs) per un totale di 14.850 minuti simulati, corrispondenti a circa 3.780 minuti al simulatore.

Per quanto concerne gli input della simulazione sono stati considerati diversi fattori. I volumi di traffico investigati vanno da 3000 veh/h (condizioni di traffico medio) a 8000 veh/h (possibili fenomeni di congestione) e per ogni flusso è stata inserita nella rete una diversa composizione di veicoli, con diversi tassi di penetrazione (MPR). Sono state considerate, infatti, nove distribuzioni di veicoli dal 10% al 90% di veicoli connessi, considerando le tre categorie: Veicoli a guida tradizionale(Manual), ACC e CACC.

| Parametri della simulazione | | |
|-----------------------------------|------------------------|--|
| Lunghezza Rete | 3500 m + 500 m esclusi | La lunghezza del tronco autostradale è di 4 km ma i primi 500 m sono stati esclusi dalla simulazione |
| Periodo di simulazione | 3000 s + 600 s esclusi | Il periodo di ogni sim run è di 60 min, ma i primi 5 min sono esclusi dalla valutazione |
| Reaction Time/ Simulation Step | 0,5 | Frequenza con cui i veicoli vengono aggiornati |
| Seed iniziale | 65 | Valore per la riproduzione di una sequenza di numeri casuali |
| Incremento del seed | 20 | Valore di incremento del seed iniziale per ogni ripetizione |
| Simulation Run | 5 | Numero di ripetizioni della simulazione |
| N. Cores | 3 | Numero di nuclei operativi della CPU per lo sviluppo di più DLL |

Tabella 5.1: Parametri della simulazione

| Composizione dei veicoli | | | |
|--------------------------|-------------|----------|-----------|
| Nome | % di Manual | % di ACC | % di CACC |
| 10% CV | 80% | 10% | 10% |
| 20% CV | 70% | 10% | 20% |
| 30% CV | 60% | 10% | 30% |
| 40% CV | 50% | 10% | 40% |
| 50% CV | 40% | 10% | 50% |
| 60% CV | 30% | 10% | 60% |
| 70% CV | 20% | 10% | 70% |
| 80% CV | 10% | 10% | 80% |
| 90% CV | 0% | 10% | 90% |

Tabella 5.2: Composizioni degli input dei veicoli

Per ogni tipologia di veicolo (a guida manuale, ACC e CACC) ampiamente

discussi nei capitoli precedenti, sono state considerate delle relative distribuzioni di velocità. Si tratta di distribuzioni di probabilità cumulata di velocità desiderata (in condizioni di flusso libero) diverse per le diverse tipologie. È stata considerata una distribuzione con valori tra 80 e 140 km/h per veicoli a guida manuale (Figura 5.1), tenendo conto dei diversi stili di guida del conducente tradizionale e della possibilità di eccedere i limiti di velocità in alcuni casi. Per i veicoli dotati di ACC e CACC, venendo meno il fattore umano, sono stati imposti dei valori più stringenti compresi tra 125 e 130 km/h, escludendo, dunque, la possibilità di velocità superiori al limite (Figura 5.2).

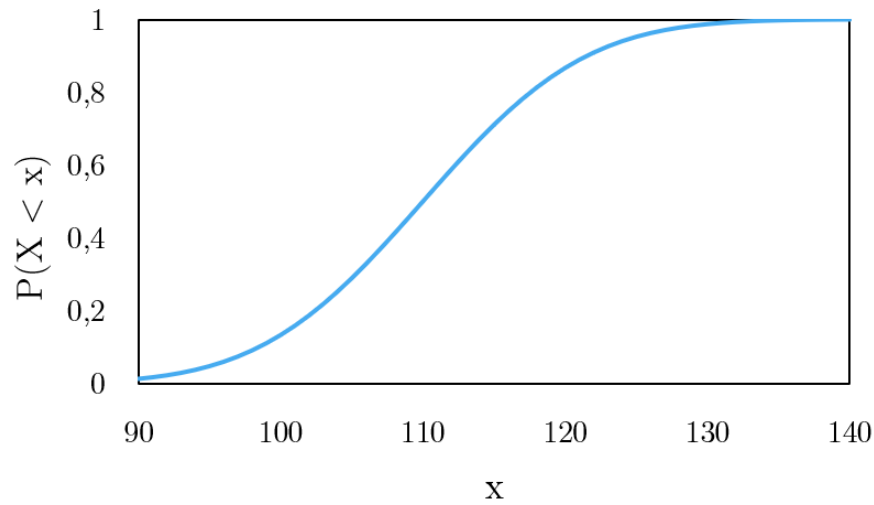


Figura 5.1: Probabilità cumulata per velocità a guida manuale

Per quanto riguarda i valori di accelerazione/decelerazione, i veicoli a guida tradizionale, conservano le impostazioni predefinite, mentre i veicoli dotati di ACC e CACC hanno un limite all'accelerazione massima di 2 m/s^2 e un limite a quella minima di -3 m/s^2 per assicurare un adeguato confort di guida per il conducente ed evitare, quindi, fenomeni di frenate brusche o accelerazioni accidentali.

I Time gap considerati sono implementati all'interno delle librerie dinamiche per l'ACC e il CACC e impostati rispettivamente pari a 1,4 s e 0,5 s, mentre per i veicoli a guida tradizionali sono stati impostati all'interno dei parametri del simulatore con una variabilità tra 1,5 s e 2,5 s.

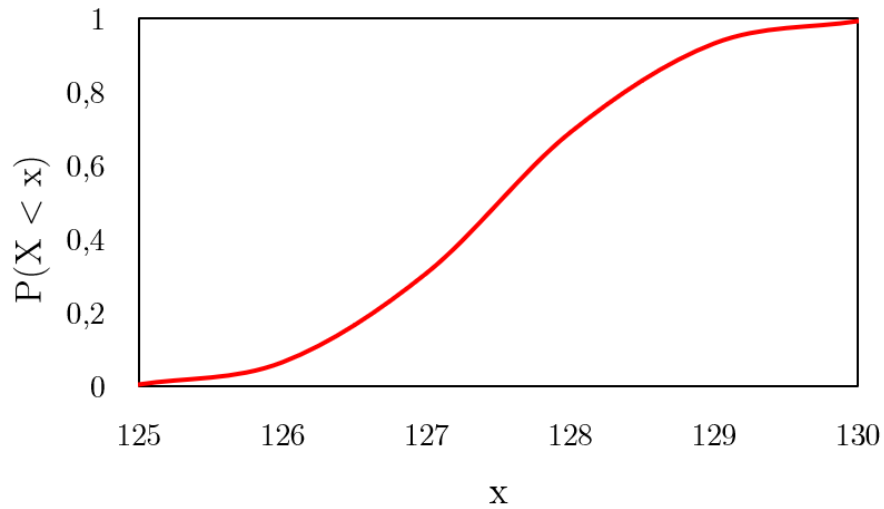


Figura 5.2: Probabilità cumulata per velocità ACC, CACC

| Input della simulazione | |
|-------------------------------|--|
| Volumi di Traffico | 3000 - 8000 veh/h |
| MPR | 10-90 % CV |
| Time Gap | Manual: 1,5-2,5 s ACC: 1,4 s CACC: 0,5 s |
| Lunghezza massima del platoon | 5 veicoli |

Tabella 5.3: Input della simulazione

5.2 Scenario 2

Il secondo scenario fa riferimento alla stessa sezione autostradale del primo scenario, ma presenta un restringimento di corsia al km 2,500, passando così da 3 corsie a 2 nei restanti 1,5 km del tronco autostradale.

Tale modifica della geometria della strada, noto anche come bottleneck, può essere causata da vari fattori, come lavori di manutenzione o incidenti. I bottleneck sono i punti più vulnerabili della rete stradale, riducono la capacità della strada, comportano una riduzione della velocità in prossimità del restringimento, generano code a monte e possono essere causa di incidenti. Sono state condotte diverse simulazioni con differenti MPR, per testare il comportamento dei veicoli equipaggiati con CACC nei confronti del restringimento e se questi siano in grado di interagire tra essi per cooperare il cambio di corsia al fine di migliorare la qualità della circolazione ed evitare fenomeni di collisione. In questo capitolo viene descritta la

procedura di settaggio del simulatore, gli input della simulazione, il contesto in cui viene riprodotta e la procedura di raccolta dei dati. Nel capitolo successivo sono analizzati i risultati.

5.2.1 Input della Simulazione

Questa simulazione è stata effettuata sulla stessa sezione autostradale di 4 km, ma dopo 2500 m è presente un bottleneck della lunghezza di 50 m, provocando un restringimento da 3 a 2 corsie nei successivi 1500 m.

L'autostrada ha come limite di velocità 130 km/h. I punti di raccolta dati sono collocati ogni 500 m.

Data la natura stocastica del traffico, per ogni simulazione, sono state effettuate 5 ripetizioni o simulation runs di 60 minuti ciascuna con incremento del seme casuale di 20. Lo step di simulazione, corrispondente con il reaction time è stato scelto pari a 0,5 s. Dalla valutazione dei dati sono stati esclusi i primi 1000 m influenzati dall'input dei veicoli e i primi 5 minuti, inteso come periodo di warm-up.

Il numero di cores, necessario per allocare e permettere il funzionamento simultaneo di più librerie dinamiche, è stato impostato a 3. Dalla combinazione dei diversi fattori descritti in seguito, sono state effettuate 9 simulazioni (45 simulation runs) per un totale di 2.475 minuti simulati, corrispondenti a circa 630 minuti al simulatore.

Per quanto concerne gli input della simulazione sono stati considerati diversi fattori. Per questo scenario è stato adottato un unico flusso veicolare pari a 3000 veh/h e per tale flusso è stata inserita nella rete una diversa composizione di veicoli, con diversi tassi di penetrazione (MPR). Sono state considerate, infatti, nove distribuzioni di veicoli dal 10% al 90% di veicoli connessi, considerando le tre categorie: Veicoli a guida tradizionale(Manual), ACC e CACC.

| Parametri della simulazione | | |
|-----------------------------------|------------------------|--|
| Lunghezza Rete | 3500 m + 500 m esclusi | La lunghezza del tronco autostradale è di 4 km ma i primi 500 m sono stati esclusi dalla simulazione |
| Periodo di simulazione | 3000 s + 600 s esclusi | Il periodo di ogni sim run è di 60 min, ma i primi 5 min sono esclusi dalla valutazione |
| Reaction Time/ Simulation Step | 0,5 | Frequenza con cui i veicoli vengono aggiornati |
| Seed iniziale | 65 | Valore per la riproduzione di una sequenza di numeri casuali |
| Incremento del seed | 20 | Valore di incremento del seed iniziale per ogni ripetizione |
| Simulation Run | 5 | Numero di ripetizioni della simulazione |
| N. Cores | 3 | Numero di nuclei operativi della CPU per lo sviluppo di più DLL |

Tabella 5.4: Parametri della simulazione

| Composizione dei veicoli | | | |
|--------------------------|-------------|----------|-----------|
| Nome | % di Manual | % di ACC | % di CACC |
| 10% CV | 80% | 10% | 10% |
| 20% CV | 70% | 10% | 20% |
| 30% CV | 60% | 10% | 30% |
| 40% CV | 50% | 10% | 40% |
| 50% CV | 40% | 10% | 50% |
| 60% CV | 30% | 10% | 60% |
| 70% CV | 20% | 10% | 70% |
| 80% CV | 10% | 10% | 80% |
| 90% CV | 0% | 10% | 90% |

Tabella 5.5: Composizioni degli input dei veicoli

Per ogni tipologia di veicolo (a guida manuale, ACC e CACC) sono state

considerate le relative distribuzioni di velocità già discusse al cap. 5.1.1, vedi Figura 5.1 e Figura 5.2.

Per quanto riguarda i valori di accelerazione/decelerazione si rimanda a quanto detto al cap. 5.1.1.

| Input della simulazione | |
|-------------------------------|--|
| Volume di Traffico | 3000 veh/h |
| MPR | 10-90 % CV |
| Time Gap | Manual: 1,5-2,5 s ACC: 1,4 s CACC: 0,5 s |
| Lunghezza massima del platoon | 5 veicoli |

Tabella 5.6: Input della simulazione

Capitolo 6

Risultati e Considerazioni

6.1 Scenario 1

Per lo scenario 1, sono state effettuate un totale di 54 simulazioni date dalla combinazione di 6 flussi veicolari e 9 MPR, e ciascuna simulazione è stata ripetuta cinque volte con un seed diverso, per un totale di 270 simulation runs.

La situazione di traffico misto 10% CV rappresenta una condizione realizzabile nel prossimo futuro, ma che al tempo stesso, data la scarsa percentuale di cooperazione risulta improbabile avere due veicoli consecutivi dotati di CACC; il 10 % dei veicoli dotati di questo servizio si comporteranno, dunque, per la quasi totalità della simulazione come veicoli ACC. Ciò fa sì che tale scenario possa essere considerato uno scenario base, rispetto al quale, cioè, calcolare le variazioni percentuali dei valori di traffico.

Lo scenario con il 90 % dei veicoli connessi rappresenta, invece, uno scenario ideale, in cui non esistano veicoli a guida tradizionale e la marcia è, dunque, quasi interamente automatizzata.

Nella valutazione dei risultati sono stati estratti delle caratteristiche principali di traffico, quali velocità, densità e tempo di viaggio medio.

I risultati sono riportati nelle tabelle e le figure seguenti.

Ogni cella della tabella rappresenta il valore medio delle cinque ripetizioni a cui è stata sottoposta ogni simulazione.

Oltre ai valori assoluti, viene riportata anche una tabella con le variazioni percentuali rispetto allo scenario con MPR 10 % CV

| Flusso | 10% CV | 20% CV | 30% CV | 40% CV | 50% CV | 60% CV | 70% CV | 80% CV | 90% CV |
|---------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 3000 veh/h | 95,9 | 97,3 | 98,3 | 99,5 | 100,2 | 101,1 | 101,3 | 102,6 | 108,1 |
| 4000 veh/h | 76,8 | 85,9 | 92,7 | 96,2 | 97,5 | 99,2 | 100,4 | 101,1 | 107,5 |
| 5000 veh/h | 72,1 | 69,7 | 72,2 | 84,6 | 94,6 | 97,4 | 99,2 | 100,8 | 105,9 |
| 6000 veh/h | 71,0 | 70,0 | 70,4 | 72,4 | 79,2 | 93,7 | 96,8 | 99,7 | 105,5 |
| 7000 veh/h | 71,9 | 69,7 | 70,5 | 72,9 | 76,1 | 84,0 | 94,8 | 98,7 | 104,0 |
| 8000 veh/h | 69,3 | 69,7 | 70,6 | 73,0 | 76,2 | 84,1 | 89,2 | 96,4 | 102,6 |

Tabella 6.1: Velocità medie in km/h per i differenti livelli di MPR dei CACC

| Flusso | 20% CV | 30% CV | 40% CV | 50% CV | 60% CV | 70% CV | 80% CV | 90% CV |
|---------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 3000 veh/h | 1,4% | 2,5% | 3,6% | 4,3% | 5,1% | 5,3% | 6,5% | 11,3% |
| 4000 veh/h | 10,7% | 17,1% | 20,2% | 21,2% | 22,6% | 23,6% | 24,1% | 28,6% |
| 5000 veh/h | -3,4% | 0,1% | 14,8% | 23,8% | 26,0% | 27,3% | 28,5% | 31,9% |
| 6000 veh/h | -1,4% | -1,0% | 1,9% | 10,4% | 24,2% | 26,6% | 28,8% | 32,7% |
| 7000 veh/h | -3,2% | -2,0% | 1,4% | 5,4% | 14,4% | 24,1% | 27,1% | 30,8% |
| 8000 veh/h | 0,5% | 1,9% | 5,1% | 9,1% | 17,6% | 22,3% | 28,1% | 32,5% |

Tabella 6.2: Variazione % delle velocità medie per i differenti livelli di MPR dei CACC rispetto allo scenario "10% CV"

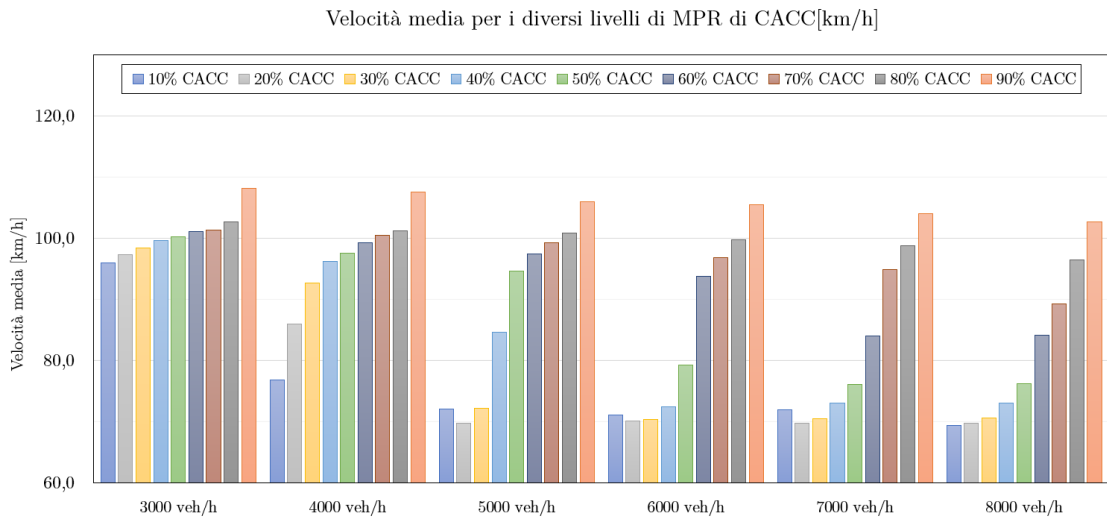


Figura 6.1: Velocità medie in km/h per i differenti livelli di MPR dei CACC

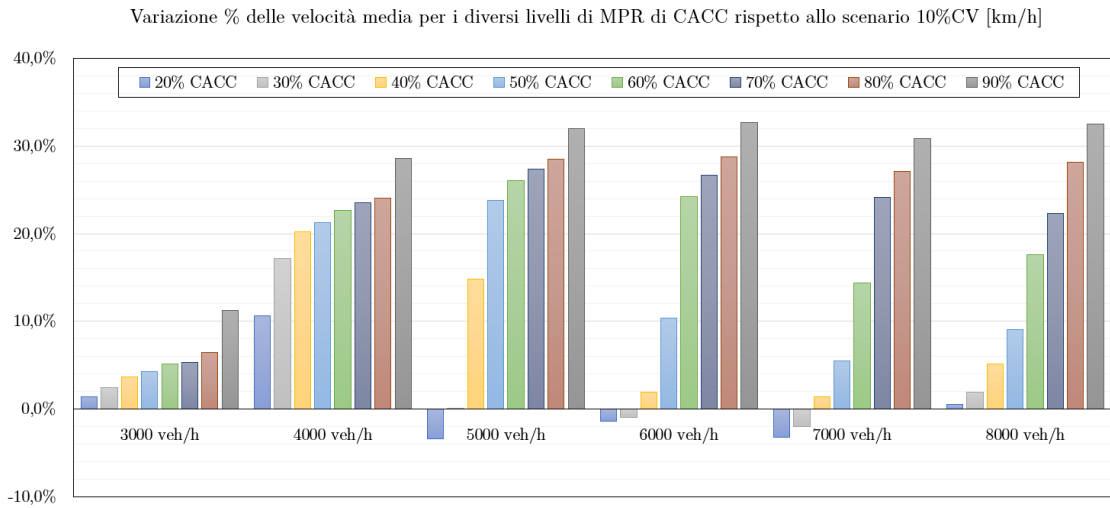


Figura 6.2: Variazione % delle velocità medie per i differenti livelli di MPR dei CACC rispetto allo scenario 10% CV

| Flusso | 10% CV | 20% CV | 30% CV | 40% CV | 50% CV | 60% CV | 70% CV | 80% CV | 90% CV |
|---------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 3000 veh/h | 31,3 | 30,9 | 30,6 | 30,2 | 30,0 | 29,8 | 29,7 | 29,3 | 27,8 |
| 4000 veh/h | 52,0 | 46,4 | 43,0 | 41,5 | 41,0 | 40,2 | 39,8 | 39,5 | 37,2 |
| 5000 veh/h | 60,2 | 67,1 | 68,9 | 58,9 | 52,8 | 51,2 | 50,3 | 49,5 | 47,2 |
| 6000 veh/h | 61,1 | 66,8 | 72,1 | 76,0 | 75,1 | 63,7 | 61,7 | 59,9 | 56,7 |
| 7000 veh/h | 60,3 | 67,1 | 72,0 | 75,7 | 79,4 | 80,6 | 73,6 | 70,7 | 67,2 |
| 8000 veh/h | 62,7 | 67,2 | 71,9 | 75,3 | 79,5 | 80,3 | 86,5 | 83,0 | 78,1 |

Tabella 6.3: Densità medie in veh/km per i differenti livelli di MPR dei CACC

| Flusso | 20% CV | 30% CV | 40% CV | 50% CV | 60% CV | 70% CV | 80% CV | 90% CV |
|---------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 3000 veh/h | -1,4% | -2,4% | -3,7% | -4,3% | -5,3% | -5,5% | -6,8% | -12,5% |
| 4000 veh/h | -12,0% | -20,8% | -25,4% | -26,9% | -29,2% | -30,8% | -31,7% | -39,6% |
| 5000 veh/h | 10,3% | 12,6% | -2,1% | -14,1% | -17,5% | -19,7% | -21,6% | -27,6% |
| 6000 veh/h | 8,6% | 15,3% | 19,6% | 18,7% | 4,1% | 1,0% | -1,9% | -7,7% |
| 7000 veh/h | 10,1% | 16,2% | 20,3% | 24,0% | 25,2% | 18,0% | 14,6% | 10,2% |
| 8000 veh/h | 6,7% | 12,9% | 16,8% | 21,1% | 22,0% | 27,5% | 24,5% | 19,8% |

Tabella 6.4: Variazione % delle densità medie per i differenti livelli di MPR dei CACC rispetto allo scenario "10% CV"

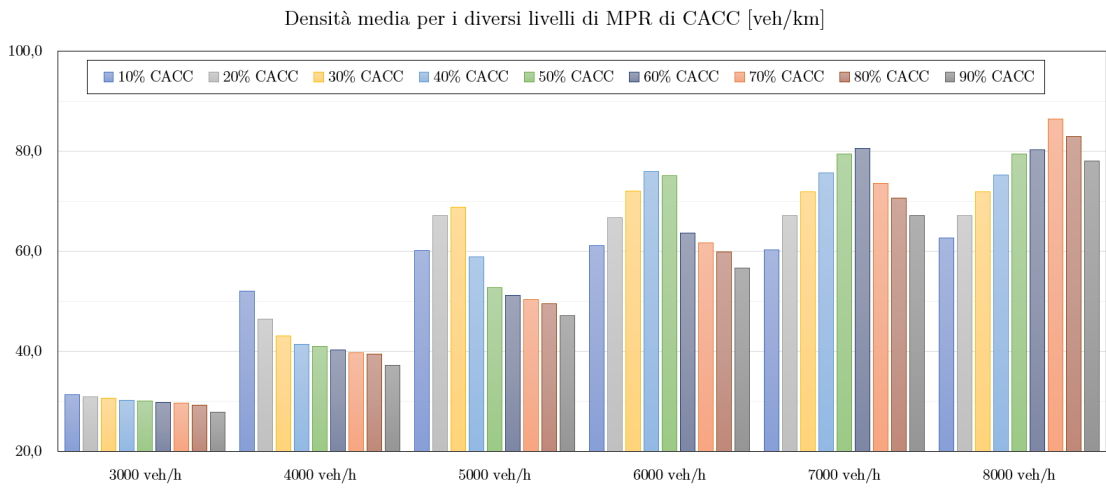


Figura 6.3: Densità medie in veh/km per i differenti livelli di MPR dei CACC

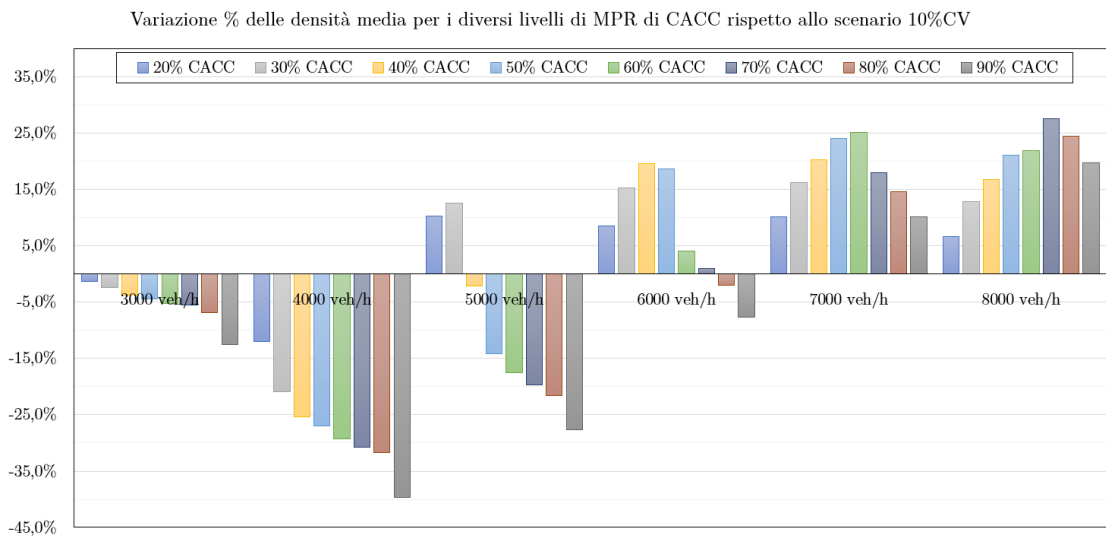


Figura 6.4: Variazione % delle densità medie per i differenti livelli di MPR dei CACC rispetto allo scenario "10% CV"

| Flusso | 10% CV | 20% CV | 30% CV | 40% CV | 50% CV | 60% CV | 70% CV | 80% CV | 90% CV |
|---------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 3000 veh/h | 149,8 | 147,4 | 145,8 | 143,7 | 142,5 | 140,6 | 139,5 | 136,7 | 128,4 |
| 4000 veh/h | 186,5 | 166,0 | 154,2 | 148,9 | 146,5 | 143,6 | 140,7 | 137,9 | 128,6 |
| 5000 veh/h | 206,2 | 209,8 | 199,4 | 167,7 | 151,2 | 146,1 | 142,6 | 139,0 | 130,7 |
| 6000 veh/h | 208,8 | 208,9 | 205,9 | 198,2 | 180,6 | 152,3 | 146,4 | 140,9 | 131,7 |
| 7000 veh/h | 206,6 | 209,6 | 205,7 | 197,3 | 187,9 | 169,7 | 150,4 | 143,0 | 133,7 |
| 8000 veh/h | 213,8 | 209,8 | 205,7 | 196,9 | 187,3 | 170,4 | 160,7 | 147,1 | 135,4 |

Tabella 6.5: Tempi di viaggio medi in secondi per i differenti livelli di MPR dei CACC

| Flusso | 20% CV | 30% CV | 40% CV | 50% CV | 60% CV | 70% CV | 80% CV | 90% CV |
|---------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 3000 veh/h | -1,6% | -2,7% | -4,2% | -5,2% | -6,6% | -7,4% | -9,6% | -16,7% |
| 4000 veh/h | -12,3% | -21,0% | -25,2% | -27,3% | -29,8% | -32,5% | -35,3% | -45,0% |
| 5000 veh/h | 1,7% | -3,4% | -22,9% | -36,4% | -41,1% | -44,6% | -48,3% | -57,8% |
| 6000 veh/h | 0,1% | -1,4% | -5,3% | -15,6% | -37,1% | -42,6% | -48,2% | -58,5% |
| 7000 veh/h | 1,4% | -0,5% | -4,7% | -10,0% | -21,7% | -37,3% | -44,5% | -54,5% |
| 8000 veh/h | -1,9% | -3,9% | -8,6% | -14,2% | -25,5% | -33,1% | -45,3% | -57,9% |

Tabella 6.6: Variazione % dei tempi di viaggio medi per i differenti livelli di MPR dei CACC rispetto allo scenario "10% CV"

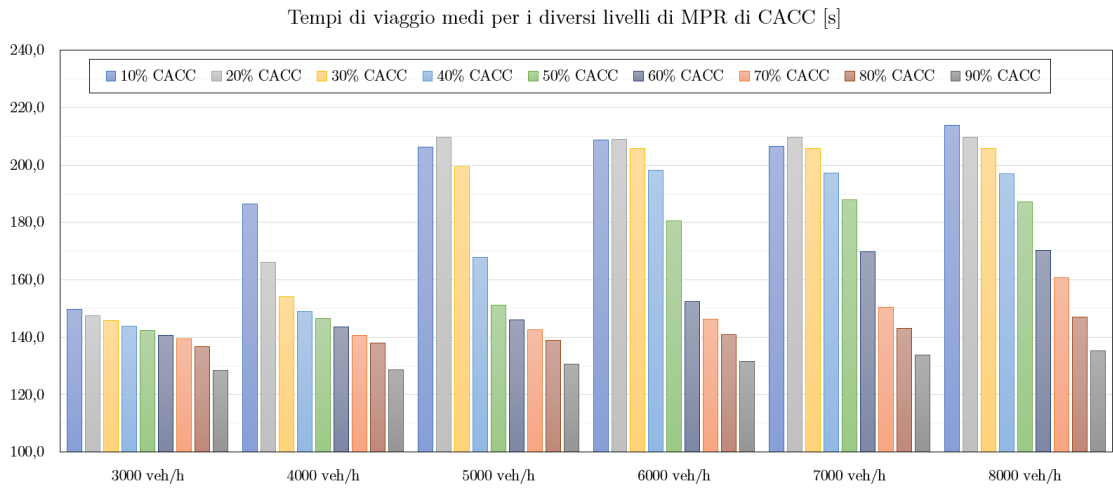


Figura 6.5: Tempi di viaggio medi in s per i differenti livelli di MPR dei CACC

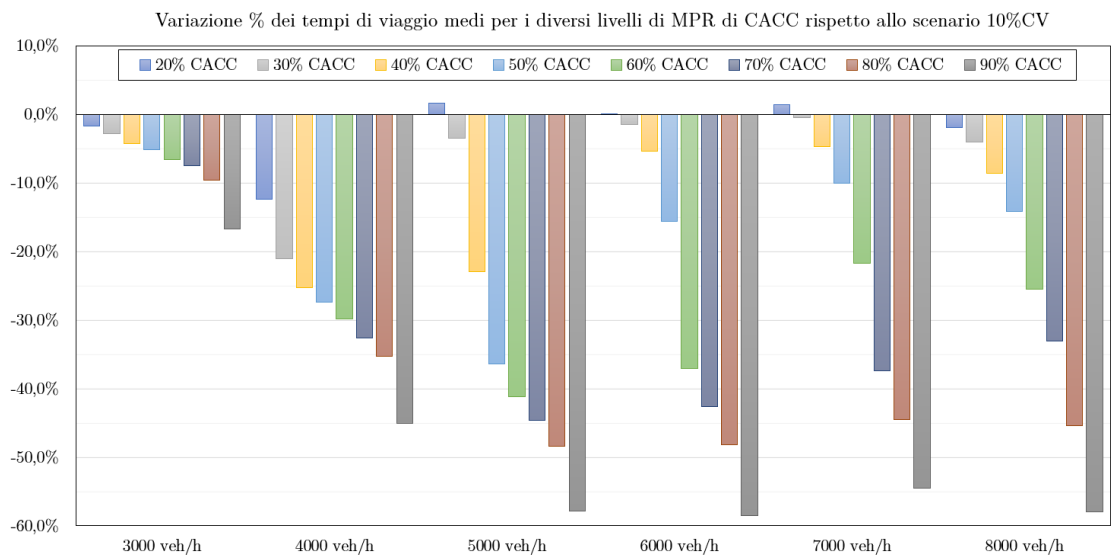


Figura 6.6: Variazione % dei tempi di viaggio per i differenti livelli di MPR dei CACC rispetto allo scenario "10% CV"

Come era lecito aspettarsi, per un dato volume di traffico le velocità medie aumentano all'aumentare del livello di penetrazione dei veicoli dotati di CACC nella rete. I valori indicati in tabella 6.2 attestano un incremento della velocità media intorno al 30 % per percentuali alte di CACC rispetto allo scenario iniziale. Per quanto riguarda la densità, ci si attende che una maggiore cooperazione dei veicoli che, garantendo degli headway ridotti, possano portare a un incremento della densità. Dalla valutazione dei risultati ciò risulta essere vero in parte. Per la stessa percentuale di veicoli CACC la densità aumenta all'aumentare del flusso, ma per un dato flusso veicolare, tende a diminuire per valori MPR crescenti e questo è spiegato dal fatto che con più veicoli automatici le velocità tendono ad aumentare per cui a parità di flusso i valori di densità diminuiscono. La mobilità dei diversi livelli di penetrazione viene valutata, inoltre, anche in base ai tempi di percorrenza della rete.

La Figura 6.5 mostra una rappresentazione grafica dei tempi di viaggio medi per i vari MPR e per i diversi flussi di traffico. La Figura 6.6 mostra, invece, una rappresentazione delle variazioni percentuali del tempo di viaggio rispetto al tempo medio di viaggio totale nello scenario iniziale (10% CV). I risultati mostrano che all'aumentare del livello di penetrazione dei veicoli equipaggiati con CACC, il tempo medio di viaggio si riduce, migliorando così la mobilità e in generale la qualità della circolazione. Dalla Tabella 6.6 e dalla Figura 6.6 si evince come il 90 % di veicoli dotati di CACC porti a una riduzione del tempo di viaggio di circa il 60 % con più di 5000 veh/h, dato che si mantiene quasi costante anche a 8000 veh/h a dimostrazione del fatto che la totale cooperazione dei veicoli sulla strada non è compromessa nonostante l'aumento dei veicoli e i possibili fenomeni di congestione. Non si riscontrano, invece, apprezzabili miglioramenti per livelli di penetrazione bassi (20-30 %), mostrando risultati simili allo scenario base.

In generale l'efficacia della cooperazione sulle caratteristiche di traffico, relativamente a questo tipo di scenario, risulta evidente con percentuali di penetrazione di veicoli CACC maggiori del 40 %.

6.2 Scenario 2

Per lo scenario 2, sono state effettuate un totale di 9 simulazioni e ciascuna simulazione è stata ripetuta cinque volte con un seed diverso, per un totale di 45 simulation runs.

Come per lo scenario 1 la situazione di traffico misto 10% CV rappresenta una condizione realizzabile nel prossimo futuro, ma che può essere considerata base, rispetto alla quale, cioè, calcolare le variazioni percentuali dei valori di traffico, come detto al paragrafo 6.1.

L'obiettivo di questa simulazione è capire come i sistemi cooperativi possano migliorare le caratteristiche di flusso in presenza di un restringimento di corsia. Nella valutazione dei risultati sono state analizzate oltre ai valori di velocità e densità media lungo la sezione, anche il ritardo medio e il numero di arresti per veicolo ed è stata condotta anche un'analisi sulla sicurezza tramite software SSAM.

SSAM (Surrogate Safety Assessment Model) è un software realizzato dalla Federal Highway Administration e combina la micro-simulazione con l'analisi automatizzata dei conflitti, analizzando la frequenza e il carattere delle collisioni da veicolo a veicolo nel traffico, per valutare la sicurezza della mobilità stradale.

Nei pressi di un bottleneck stazionario, ciò che generalmente si osserva è un incremento della densità. Un restringimento di corsia aumenta dunque la densità del flusso di traffico, ma se tale densità indotta è bassa, i veicoli procedono senza particolari complicazioni.

L'aumento della densità dovuto all'effetto di un bottleneck è la chiave per comprendere l'emergere di un fenomeno di congestione e lo sviluppo di code a monte del restringimento.

Di seguito, sono riportati alcuni dettagli grafici dell'ambiente di simulazione in alcuni momenti chiave. Per ogni cattura in basso a sinistra è indicata la scala e il secondo di simulazione.

In Figura 6.7 relativa allo scenario 10% CV si nota come in prossimità della fine del restringimento vi sia un possibile fenomeno di collisione, inoltre, quasi tutti i veicoli a monte del restringimento sono prossimi tra loro a suggerire il fatto che essi siano fermi o comunque procedono a basse velocità ad eccezione della corsia all'estrema destra. In questo caso risulta evidente come la coda si sia già formata dopo circa 12 minuti.

Procedendo con la simulazione (Figura 6.8) la coda tende ad aumentare; è evidente che la circolazione sia ormai compromessa, la propagazione di stop & go è in aumento e il sistema non è più in grado di smaltire la coda.

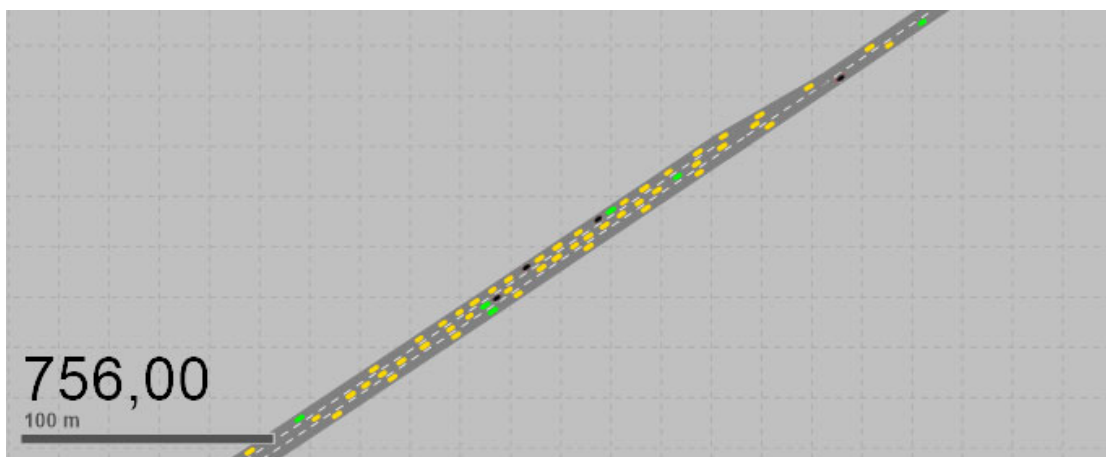


Figura 6.7: Dettaglio n. 1 sul restringimento di corsia nello scenario "10% CV"

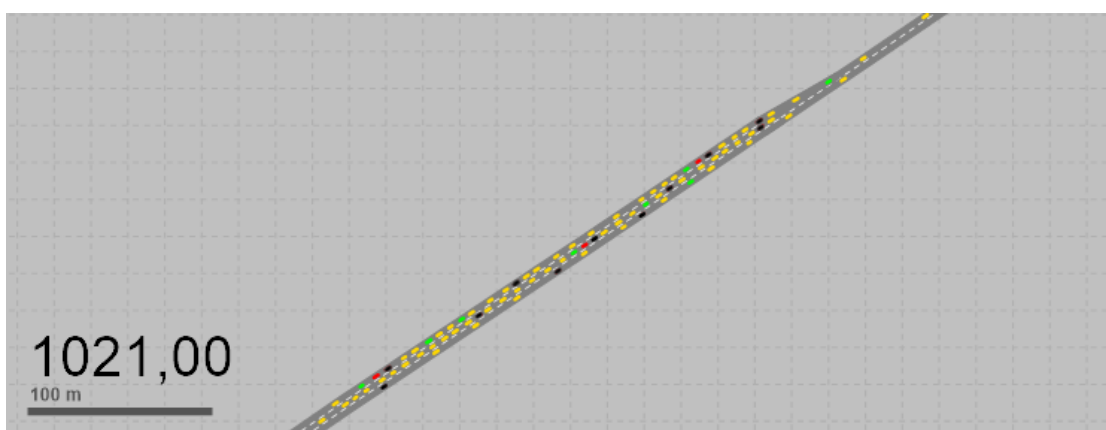


Figura 6.8: Dettaglio n. 2 sul restringimento di corsia nello scenario "10% CV"

Nello scenario intermedio (50 % CV), come si può vedere in Figura 6.9, in corrispondenza del restringimento nella corsia centrale, si è aperto un gap tra due veicoli equipaggiati con il CACC e facenti parte di un platoon (veicolo Leader in nero che li precede). Quest'apertura è resa possibile dalla comunicazione tra i veicoli su corsie laterali e permette l'immissione in sicurezza dell'automobile presente sulla corsia di sinistra e senza la necessità di arrestarsi. In Figura 6.10 è evidente come dopo circa 20 minuti di simulazione non vi siano rallentamenti significativi a monte del restringimento. In Figura 6.11 si riporta un ulteriore caso di apertura di un gap in un platoon, vedi corsia centrale, per permettere l'immissione, in questo caso, di due veicoli adiacenti.

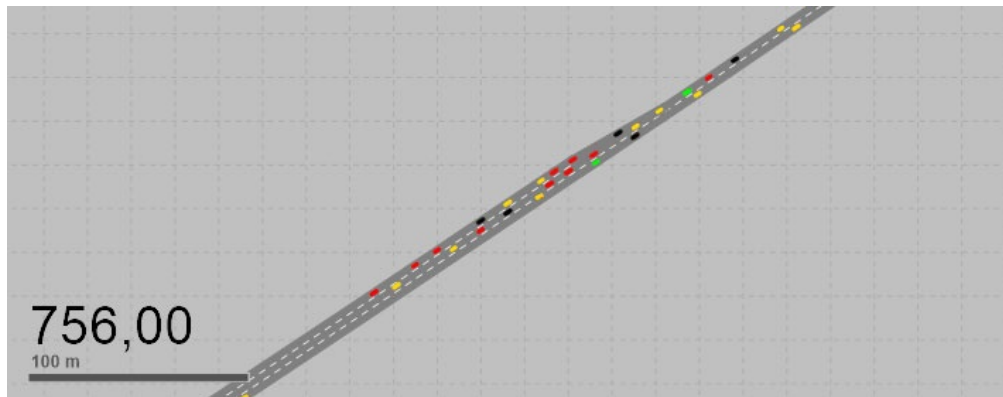


Figura 6.9: Dettaglio n. 3 sul restringimento di corsia nello scenario "50% CV"

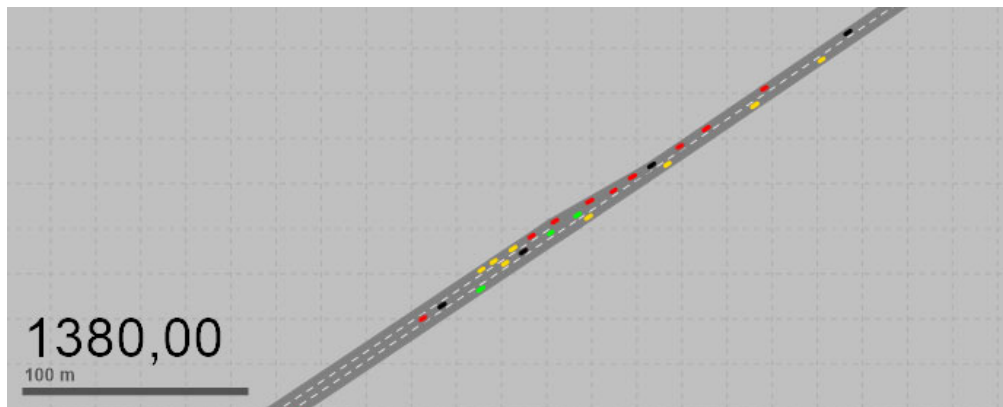


Figura 6.10: Dettaglio n. 4 sul restringimento di corsia nello scenario "50% CV"

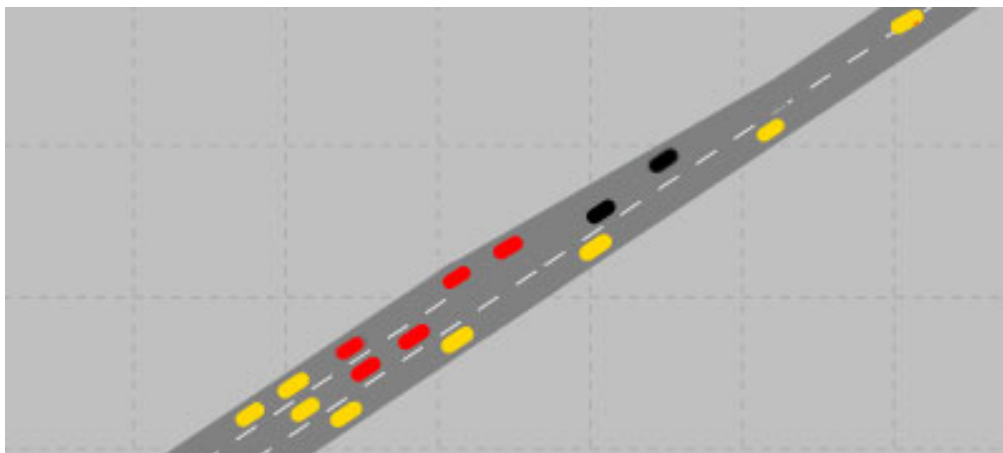


Figura 6.11: Dettaglio n. 5 sul restringimento di corsia nello scenario "50% CV"

I risultati delle simulazioni sono riportati graficamente nelle figure seguenti.

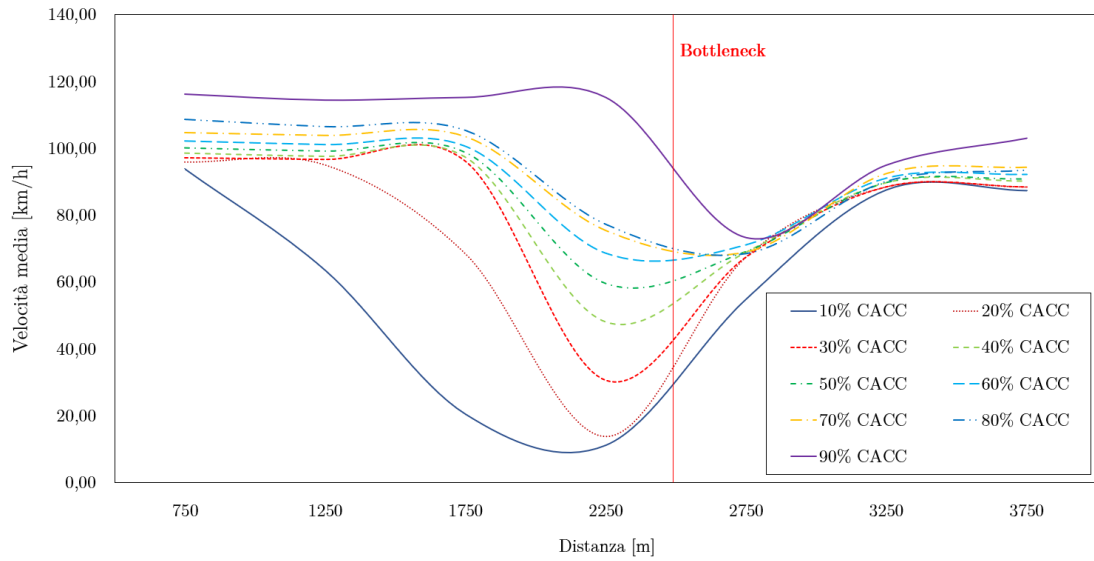


Figura 6.12: Velocità media dei vari MPR lungo la sezione

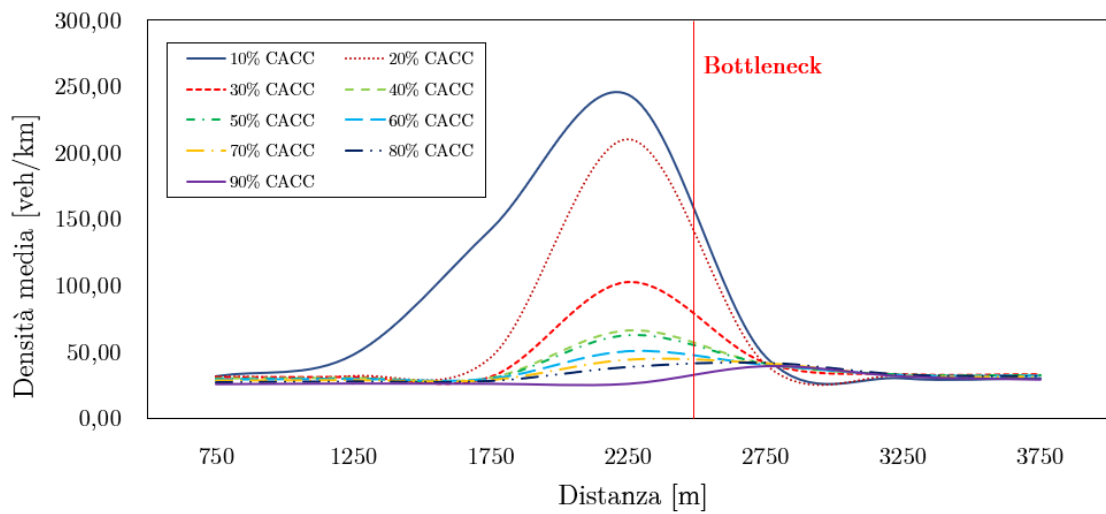


Figura 6.13: Densità media dei vari MPR lungo la sezione

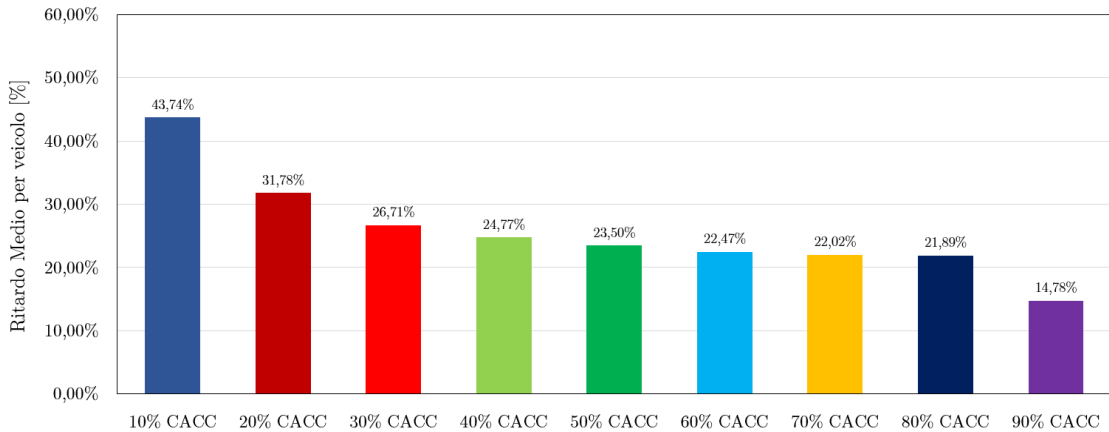


Figura 6.14: Ritardo medio in % per veicolo

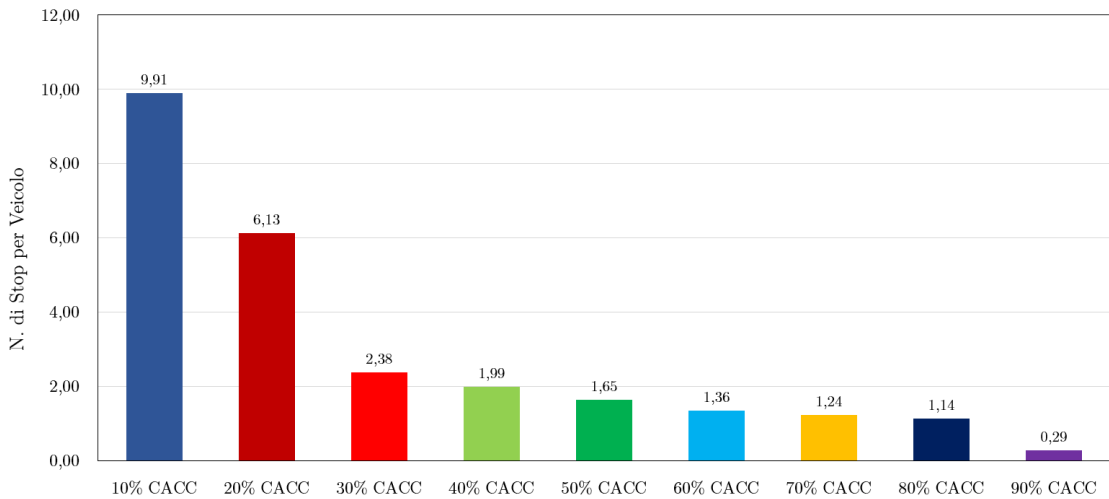


Figura 6.15: Numero medio di Stop per veicolo

I risultati evidenziano come per una percentuale di veicoli cooperativi crescente i valori di densità nei pressi del bottleneck diminuiscono, il ritardo medio si riduce con una deviazione massima di circa il 30 % così come diminuiscono considerevolmente il numero di stop per veicolo, cioè, con una cooperazione del 90 % i veicoli quasi mai sono costretti a fermarsi, a riprova del fatto che i fenomeni di stop & go vengono ridotti.

La principale differenza nei risultati risiede nella cooperazione adottata al momento del cambio corsia forzato. I veicoli dotati di CACC hanno, infatti, la possibilità di coordinare il cambio corsia riducendo la velocità per permettere al veicolo presente nella strozzatura di individuare il gap necessario per effettuare la svolta.

Questa collaborazione permette, chiaramente, di ottimizzare il sistema, evitando una propagazione di stop & go a monte del restringimento.

Un'ulteriore indicazione dell'efficacia dei sistemi cooperativi sulla viabilità generale in presenza di un bottleneck è data dai diagrammi di spazio-tempo seguenti.

Tali diagrammi, riferiti allo scenario 10% CV (Figura 6.16) e allo scenario 90% CV (Figura 6.17), riportano le traiettorie di una parte dei veicoli che attraversano il restringimento. Si vede come nel primo caso, i veicoli siano soggetti a un arresto prolungato nel tempo (pendenza delle curve nulla in prossimità del bottleneck) prima di procedere la marcia regolarmente, cosa che invece è fortemente limitata nel secondo caso, a riprova di quanto detto sopra.

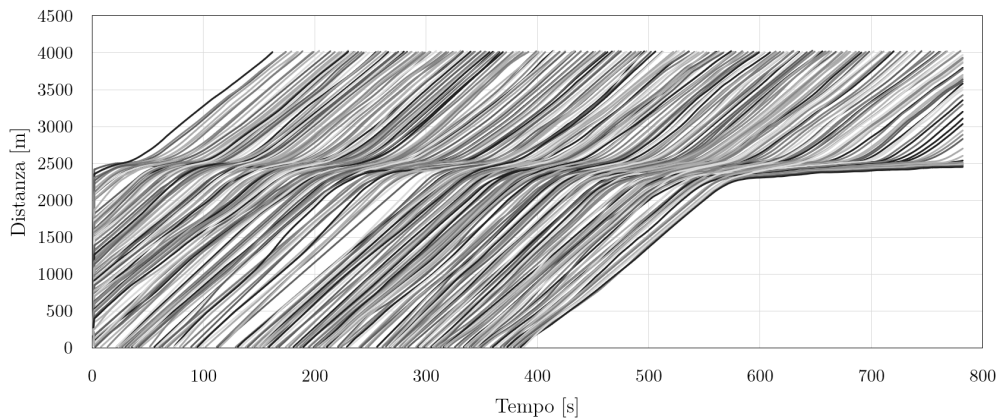


Figura 6.16: Diagramma spazio-tempo dei veicoli nello scenario "10%CV"

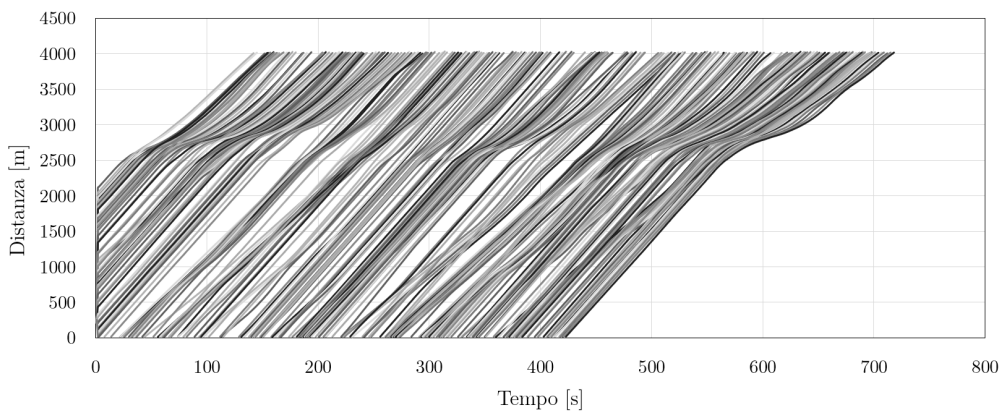


Figura 6.17: Diagramma spazio-tempo dei veicoli nello scenario "90%CV"

6.2.1 Analisi sulla Sicurezza in un Bottleneck

Il Surrogate Safety Assessment Model (SSAM) è un software che analizza le traiettorie dei veicoli forniti come output dal micro-simulatore, al fine di valutare i punti di conflitto osservati nella simulazione. SSAM processa un file di output del micro-simulatore contenente le traiettorie (.trj) e analizza le interazioni da veicolo a veicolo all'interno del file. Insieme al rilevamento dei conflitti, SSAM fornisce misure di sicurezza come output, nonché la classificazione del tipo di conflitto come, ad esempio, collisione laterale, tamponamento o scontro frontale.

Un conflitto è definito come una situazione osservabile in cui due o più utenti della strada si avvicinano l'un l'altro nel tempo e nello spazio in misura tale che esiste il rischio di collisione se i loro movimenti rimangono invariati.

La valutazione dei conflitti è stata effettuata per i livelli di composizione veicolare di cui sopra e sono stati ottenuti i risultati seguenti:

| 10% CV | 20% CV | 30% CV | 40% CV | 50% CV | 60% CV | 70% CV | 80% CV | 90% CV |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 21844 | 17634 | 6313 | 4347 | 4174 | 2941 | 2053 | 1582 | 744 |

Tabella 6.7: Numero di possibili fenomeni di conflitto per le diverse composizioni veicolari

| 20% CV | 30% CV | 40% CV | 50% CV | 60% CV | 70% CV | 80% CV | 90% CV |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| -19% | -71% | -80% | -81% | -87% | -91% | -93% | -97% |

Tabella 6.8: Variazione percentuale del numero di possibili fenomeni di collisioni per le diverse composizioni veicolari rispetto allo scenario "10% CV"

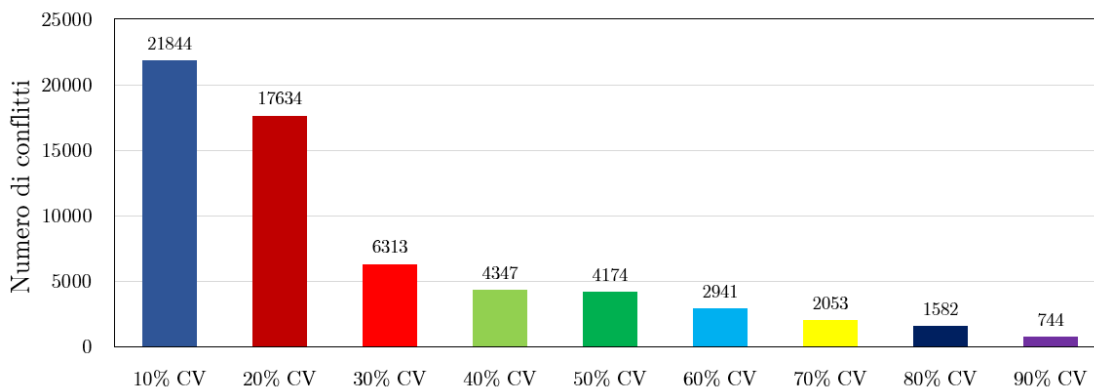


Figura 6.18: Numero di possibili fenomeni di conflitto per le diverse composizioni veicolari

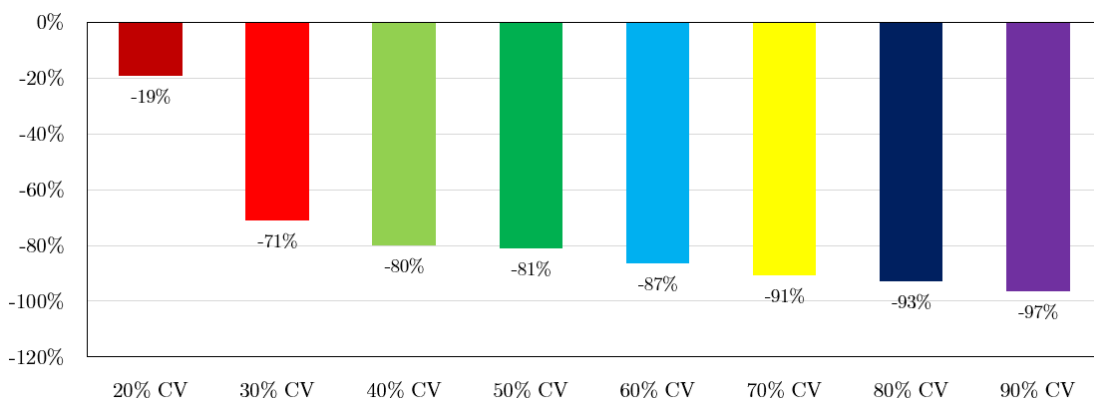


Figura 6.19: Variazione percentuale del numero di possibili fenomeni di conflitto per le diverse composizioni veicolari rispetto allo scenario "10% CV"

La valutazione dell'analisi di sicurezza evidenzia come a un aumento di veicoli connessi nella rete corrisponda una evidente riduzione nel numero di conflitti fino al 97 % nel caso ideale, ma con riduzioni comunque consistenti anche con una percentuale di veicoli dotati di CACC prossima al 40%.

Si precisa che tale valutazione è stata eseguita assumendo una soglia di Time to Collision (TTC) pari a 1,5 secondi, ma la variazione di tale soglia non comporta differenze significative negli scarti percentuali. Per TTC s'intende quel valore di tempo, in un dato istante di tempo t, necessario affinché avvenga la collisione se i due veicoli procedono alla stessa velocità.

Capitolo 7

Conclusioni

La mobilità del prossimo futuro si baserà inevitabilmente su sistemi di guida cooperativi che sono non solo prodromici per i futuri veicoli a guida autonoma, ma permettono di realizzare un vero e proprio ambiente interattivo tra veicoli e infrastruttura.

In questa tesi sono stati presentati e descritti i sistemi di Cruise Control Cooperativo (CACC), intesi come sistemi di guida avanzati, dotati della possibilità di scambiare informazioni con altri veicoli e con altri elementi dell'infrastruttura e atti a migliorare le caratteristiche di traffico, il confort di guida e la sicurezza stradale.

Il cruise control adattivo basato su sensori (ACC) attualmente in uso presenta notevoli limitazioni delle prestazioni, tra le quali un ritardo nella comunicazione fino a 0,5 secondi che comporta un contestuale ritardo nella risposta a cambi di movimento del veicolo che precede e inoltre prevede un time gap minimo di circa 1,5 secondi. Il CACC permette di ridurre le distanze temporali e spaziali tra i veicoli, migliorando così la qualità della circolazione e la stabilità del flusso di traffico. Questi time gap più brevi possono portare a raddoppiare la capacità di una corsia autostradale, ottimizzando dunque l'occupazione dello spazio stradale. All'interno dell'elaborato viene descritto dapprima il funzionamento del sistema di cruise control cooperativo. L'analisi delle caratteristiche di questi sistemi e il loro apporto al miglioramento della circolazione e della sicurezza stradale è stata testata all'interno di un simulatore di traffico. Per simulare la cooperazione tra veicoli è stato necessario implementare dei modelli di guida esterni al simulatore sottoforma di DLL, librerie dinamiche che interagiscono con il simulatore in tempo reale. Sono state create due DLL, sia per l'ACC che per il CACC, mentre la guida manuale è stata affidata ai parametri interni del simulatore.

Durante la simulazione la DLL riceve determinate informazioni dalla simulazione stessa, tali informazioni permettono di definire quegli algoritmi necessari per aggiornare i valori di accelerazioni, velocità, distanze, formazione di platoon che vengono applicate nel simulatore per ogni step di simulazione.

Il comportamento cooperativo è stato, dunque, simulato come aggiornamento del valore di accelerazione. Gli algoritmi usati per il calcolo dell'accelerazione dei veicoli automatici si basano su quelli proposti da Xiao et al. [17] e da Van Arem et al. [15] espresse in 4.4 e 4.5 al capitolo 4.2.1. In seguito, sono stati condotti test per la verifica dell'efficacia dei sistemi di CACC nel miglioramento delle condizioni di circolazione e della loro sicurezza. Un primo test è stato effettuato per evidenziare la possibilità di formazione di un platoon e per vedere come i parametri dinamici si adeguassero meglio e più velocemente rispetto ai veicoli dotati di ACC e ai veicoli a guida manuale. In questo caso un platoon di 5 veicoli si avvicina a un veicolo più lento (Tab. 4.1). I profili di velocità (Figura 4.11) e di accelerazione (Figura 4.12) sono prossimi tra loro a testimoniare la simulazione di una risposta nell'azione di guida quasi immediata rispetto ai veicoli dotati di ACC e ai veicoli a guida manuale. Inoltre, le curve di tale diagramma sono funzioni quasi interamente regolari, ovvero non presentano punti spigolosi o discontinuità a suggerire un comportamento di guida confortevole senza brusche variazioni di velocità.

Un secondo test è stato effettuato per valutare la sicurezza di tali sistemi nei confronti delle distanze ravvicinate che riescono a tenere. Ridurre i time gap tra veicoli vuol dire ridurre le distanze di sicurezza, per cui viene analizzato il comportamento di questi platoon di veicoli in caso di frenate d'emergenza.

In condizioni emergenziali, il veicolo, comunque dotato di frenata d'emergenza assistita (AEB: Autonomous Emergency Braking), o con pressione manuale del pedale del freno può assicurare valori di decelerazione inferiori al limite minimo imposto (-3 m/s^2), in modo da permettere una frenata tempestiva in situazioni di elevato rischio di collisione.

Nel test condotto al simulatore un veicolo dotato di CACC segue un veicolo leader anch'esso dotato di Cruise Control Cooperativo, il quale è soggetto a ripetute variazioni di velocità, fino ad arrestarsi da una velocità di 70 km/h, attuando l'AEB.

Dai risultati delle simulazioni (Fig. 4.2, 4.3, 4.4) si evince che i veicoli rispondono velocemente escludendo collisioni.

Un terzo test, infine, è stato condotto all'interno di un anello per verificare la stabilità del flusso di traffico nei confronti di perturbazioni delle velocità e generazione di onde di stop & go. Nell'anello di 170 m di circonferenza sono stati inseriti dieci veicoli, inizialmente fermi a distanza ravvicinata. Il test è stato eseguito per due scenari, uno in cui i veicoli erano dotati di CACC e l'altro in cui i veicoli erano a guida manuale. Il limite massimo di velocità è stato imposto a 50 km/h, ma ad un certo punto della simulazione il veicolo leader decelera, in modo da indurre una perturbazione nella stabilità del flusso di traffico. I risultati hanno dimostrato che la cooperazione indotta dai sistemi di CACC migliora sensibilmente la stabilità rispetto a uno scenario con soli veicoli a guida manuale. In quest'ultimo scenario, infatti, anche prima di indurre la perturbazione si genera un'onda di stop & go

che si propaga a monte per tutta la durata della simulazione, cosa che invece non avviene nello scenario con soli veicoli cooperativi.

Sono state eseguite, poi, le simulazioni in diversi scenari di traffico misto. Un primo scenario (Capitolo 5.1) è stato simulato su un ramo autostradale privo di interferenze interne ed esterne, quindi in condizioni di flusso libero. Per quanto concerne gli input della simulazione sono stati considerati diversi fattori. Un volume di traffico variabile da 3000 veh/h a 8000 veh/h e per ogni flusso è stata inserita nella rete una diversa composizione di veicoli, per un totale di nove, con percentuale crescente di veicoli connessi.

Dall'analisi dei risultati (Capitolo 6.1) si è osservato che in condizioni di flusso libero e in assenza di interferenze interne ed esterne alla circolazione i parametri di velocità, densità e tempo di viaggio migliorano sensibilmente all'aumentare del tasso di penetrazione di veicoli dotati di CACC, così come aumenta la loro sicurezza.

Per i diversi scenari si ha un incremento dei valori di velocità media fino al 30% nello scenario ideale (90% CV), un dimezzamento dei tempi di viaggio. Per quanto riguarda la densità i risultati delle simulazioni hanno evidenziato alcune divergenze con i risultati attesi. Era lecito aspettarsi che una maggiore cooperazione dei veicoli avrebbe potuto portare a un incremento della densità, poiché i veicoli dotati di CACC sono in grado di garantire degli headway ridotti, ma dalla valutazione dei risultati ciò risulta essere vero in parte. Infatti, per la stessa percentuale di veicoli CACC la densità aumenta all'aumentare del flusso, ma per un dato flusso veicolare, tende a diminuire per percentuali di veicoli connessi crescenti e questo può essere spiegato dal fatto che con più veicoli automatici le velocità tendono ad aumentare per cui a parità di flusso i valori di densità diminuiscono. Infine, un secondo scenario (Capitolo 6.2) è stato simulato sullo stesso ramo autostradale del primo scenario, ma con la presenza di un restringimento di corsia. Anche in questo caso sono state valutate le nove composizioni veicolari di cui sopra, ma per un unico flusso di 3000 veh/h. I risultati hanno evidenziato come per una percentuale di veicoli cooperativi crescente i valori di densità nei pressi del bottleneck diminuiscono, il ritardo medio si riduce con una deviazione massima di circa il 30 % così come diminuiscono considerevolmente il numero di stop per veicolo. Infatti con una cooperazione del 90 %, i veicoli quasi mai sono costretti a fermarsi, a riprova del fatto che i fenomeni di stop & go vengono ridotti.

Un'ulteriore valutazione sulla sicurezza ha evidenziato come per una percentuale crescente di veicoli connessi si riduca notevolmente il rischio di impatto, con percentuali di riduzione superiori al 90%. In conclusione, si può affermare che dall'analisi delle simulazioni condotte, la cooperazione veicolare in uno scenario di traffico misto aiuta a prevenire fenomeni di congestione, nel caso di volumi di traffico elevati per l'intrinseca capacità di ottimizzare lo spazio a disposizione e per la capacità di comunicazione wireless che regola automaticamente la marcia

in modo sicuro e riducendo di fatto l'errore umano. Anche in presenza di una criticità puntuale della strada, come un restringimento di corsia, si è osservato che la cooperazione migliora la qualità della circolazione, previene fenomeni di code a monte del bottleneck e migliora la sicurezza stradale.

Appendice A

CACC Driver Model

```
1
2 #include "DriverModel.h"
3
4 /*=====*/
5
6 double current_speed = 0.0;
7 double desired_acceleration = 0.0;
8 double original_desired_velocity = 0.0;
9 double desired_lane_angle = 0.0;
10 long active_lane_change = 0;
11 long original_active_lane_change = 0;
12 long rel_target_lane = 0;
13 double desired_velocity = 0.0;
14 long turning_indicator = 0;
15 long vehicle_color = CMYK(255, 255, 255, 255);
16 double lead_vehicle_distance = 999.0;
17 double tail_vehicle_distance = 999.0;
18 double left_lead_vehicle_distance = 999.0;
19 double left_tail_vehicle_distance = 999.0;
20 double right_lead_vehicle_distance = 999.0;
21 double right_tail_vehicle_distance = 999.0;
22 double lead_vehicle_distance2 = 999.0;
23 double tail_vehicle_distance2 = 999.0;
24 double left_lead_vehicle_distance2 = 999.0;
25 double left_tail_vehicle_distance2 = 999.0;
26 double right_lead_vehicle_distance2 = 999.0;
27 double right_tail_vehicle_distance2 = 999.0;
28 double lead_vehicle_speed_difference = -99.0;
29 double lead_vehicle_length = 0.0;
30 double ka = 1.0;
31 double kv = 0.58;
```

```

32 double ks = 0.1;
33 double ac = 0.0;
34 double ap = 0.0;
35 double vp = 0.0;
36 double vf = 0.0;
37 double s = 0.0;
38 double v = 0.0;
39 double td = 0.5; /* 0.5 s of time gap */
40 double amin = -3.0;
41 double amax = 2.0;
42 long lead_platoon = 0;
43 long join_platoon = 0;
44 long vehicle_category = 0;
45 long vehicle_type = 0;
46 long lead_vehicle_category = 0;
47 long tail_vehicle_category = 0;
48 long right_lead_vehicle_category = 0;
49 long right_tail_vehicle_category = 0;
50 long left_lead_vehicle_category = 0;
51 long left_tail_vehicle_category = 0;
52 long lead_vehicle_category2 = 0;
53 long tail_vehicle_category2 = 0;
54 long right_lead_vehicle_category2 = 0;
55 long right_tail_vehicle_category2 = 0;
56 long left_lead_vehicle_category2 = 0;
57 long left_tail_vehicle_category2 = 0;
58 long vehicle_lane = 0;
59
60 /*=====*/
61
62 BOOL APIENTRY DllMain(HANDLE hModule,
63     DWORD ul_reason_for_call,
64     LPVOID lpReserved)
65 {
66     switch (ul_reason_for_call) {
67     case DLL_PROCESS_ATTACH:
68     case DLL_THREAD_ATTACH:
69     case DLL_THREAD_DETACH:
70     case DLL_PROCESS_DETACH:
71         break;
72     }
73     return TRUE;
74 }
75
76 /*=====*/
77
78 DRIVERMODEL_API int DriverModelSetValue(long type,
79     long index1,
80     long index2,

```

```

81 | long    long_value ,
82 | double  double_value ,
83 | char    *string_value)
84 | {
85 |     /* Sets the value of a data object of type <type>, selected by <
index1> */
86 |     /* and possibly <index2>, to <int_value>, <double_value> or
*/
87 |     /* <*string_value> (object and value selection depending on <type
>). */
88 |     /* Return value is 1 on success , otherwise 0.
*/
89 |
90 |     switch (type) {
91 |     case DRIVER_DATA_Timestep:
92 |     case DRIVER_DATA_TIME:
93 |         return 1;
94 |     case DRIVER_DATA_VEH_ID:
95 |         /* reset leading vehicle's data for this new vehicle */
96 |         lead_vehicle_distance = 999.0;
97 |         lead_vehicle_speed_difference = -99.0;
98 |         lead_vehicle_length = 0.0;
99 |         return 1;
100 |     case DRIVER_DATA_VEH_LANE:
101 |         vehicle_lane = long_value;
102 |         return 1;
103 |     case DRIVER_DATA_VEH_ODOMETER:
104 |     case DRIVER_DATA_VEH_LANE_ANGLE:
105 |     case DRIVER_DATA_VEH_LATERAL_POSITION:
106 |         return 1;
107 |     case DRIVER_DATA_VEH_VELOCITY:
108 |         current_speed = double_value;
109 |         vf = current_speed;
110 |         return 1;
111 |     case DRIVER_DATA_VEH_ACCELERATION:
112 |     case DRIVER_DATA_VEH_LENGTH:
113 |     case DRIVER_DATA_VEH_WIDTH:
114 |     case DRIVER_DATA_VEH_WEIGHT:
115 |     case DRIVER_DATA_VEH_MAX_ACCELERATION:
116 |         return 1;
117 |     case DRIVER_DATA_VEH_TURNING_INDICATOR:
118 |         turning_indicator = long_value;
119 |         return 1;
120 |     case DRIVER_DATA_VEH_CATEGORY:
121 |     case DRIVER_DATA_VEH_PREFERRED_REL_LANE:
122 |     case DRIVER_DATA_VEH_USE_PREFERRED_LANE:
123 |         return 1;
124 |     case DRIVER_DATA_VEH_DESIRED_VELOCITY:
125 |         original_desired_velocity = double_value;

```



```

126     return 1;
127 case DRIVER_DATA_VEH_X_COORDINATE:
128 case DRIVER_DATA_VEH_Y_COORDINATE:
129 case DRIVER_DATA_VEH_TYPE:
130     vehicle_type = long_value;
131     return 1;
132 case DRIVER_DATA_VEH_COLOR:
133     /*vehicle_color = long_value;*/
134     vehicle_color = CMYK(255, 255, 0, 0);
135     return 1;
136 case DRIVER_DATA_VEH_CURRENT_LINK:
137     return 0; /* (To avoid getting sent lots of
DRIVER_DATA_VEH_NEXT_LINKS messages) */
138     /* Must return 1 if these messages are to be sent
from VISSIM! */
139 case DRIVER_DATA_VEH_NEXT_LINKS:
140 case DRIVER_DATA_VEH_ACTIVE_LANE_CHANGE:
141 case DRIVER_DATA_VEH_REL_TARGET_LANE:
142 case DRIVER_DATA_NVEH_ID:
143 case DRIVER_DATA_NVEH_LANE_ANGLE:
144 case DRIVER_DATA_NVEH_LATERAL_POSITION:
145     return 1;
146 case DRIVER_DATA_NVEH_DISTANCE:
147     if (index1 == 0 && index2 == 1) { /* leading vehicle on own
lane */
148         lead_vehicle_distance = double_value;
149     }
150     else if (index1 == 0 && index2 == -1) {
151         tail_vehicle_distance = double_value;
152     }
153     else if (index1 == 0 && index2 == 2) {
154         lead_vehicle_distance2 = double_value;
155     }
156     else if (index1 == 0 && index2 == -2) {
157         tail_vehicle_distance2 = double_value;
158     }
159     if (vehicle_lane == 1) {
160         if (index1 == 1 && index2 == 1) {
161             left_lead_vehicle_distance = double_value;
162         }
163         else if (index1 == 1 && index2 == -1) {
164             left_tail_vehicle_distance = double_value;
165         }
166         else if (index1 == 1 && index2 == 2) {
167             left_lead_vehicle_distance2 = double_value;
168         }
169         else if (index1 == 1 && index2 == -2) {
170             left_tail_vehicle_distance2 = double_value;
171         }

```

```

172     }
173     else if (vehicle_lane == 2) {
174         if (index1 == 1 && index2 == 1) {
175             right_lead_vehicle_distance = double_value;
176         }
177         else if (index1 == 1 && index2 == -1) {
178             right_tail_vehicle_distance = double_value;
179         }
180         else if (index1 == 1 && index2 == 2) {
181             right_lead_vehicle_distance2 = double_value;
182         }
183         else if (index1 == 1 && index2 == -2) {
184             right_tail_vehicle_distance2 = double_value;
185         }
186     }
187     return 1;
188 case DRIVER_DATA_NVEH_REL_VELOCITY:
189     if (index1 == 0 && index2 == 1) { /* leading vehicle on own
lane */
190         lead_vehicle_speed_difference = double_value;
191         vp = current_speed - lead_vehicle_speed_difference;
192     }
193     return 1;
194 case DRIVER_DATA_NVEH_ACCELERATION:
195     if (index1 == 0 && index2 == 1) {
196         ap = double_value;
197     }
198     return 1;
199 case DRIVER_DATA_NVEH_LENGTH:
200     if (index1 == 0 && index2 == 1) { /* leading vehicle on own
lane */
201         lead_vehicle_length = double_value;
202         s = lead_vehicle_distance - lead_vehicle_length;
203     }
204     return 1;
205 case DRIVER_DATA_NVEH_WIDTH:
206 case DRIVER_DATA_NVEH_WEIGHT:
207 case DRIVER_DATA_NVEH_TURNING_INDICATOR:
208 case DRIVER_DATA_NVEH_CATEGORY:
209 case DRIVER_DATA_NVEH_TYPE:
210     if (index1 == 0 && index2 == 1) { /* Next vehicle on own lane
*/
211         lead_vehicle_category = long_value;
212     }
213     if (index1 == 0 && index2 == -1) { /* Following vehicle on
own lane */
214         tail_vehicle_category = long_value;
215     }

```

```

216     if (index1 == 0 && index2 == 2) { /* Second next vehicle
downstream on own lane */
217         lead_vehicle_category2 = long_value;
218     }
219     if (index1 == 0 && index2 == -2) { /* Second next vehicle
upstream on own lane */
220         tail_vehicle_category2 = long_value;
221     }
222     if (vehicle_lane == 2) { /* Middle lane for three-lane road
*/
223         lead_platoon = 0;
224         join_platoon = 0;
225         if (index1 == -1 && index2 == 1) { /* Next vehicle
downstream on the right lane */
226             right_lead_vehicle_category = long_value;
227         }
228         if (index1 == -1 && index2 == -1) { /* Next vehicle
upstream on the right lane */
229             right_tail_vehicle_category = long_value;
230         }
231         if (index1 == -1 && index2 == 2) { /* Second next vehicle
downstream on the right lane */
232             right_lead_vehicle_category2 = long_value;
233         }
234         if (index1 == -1 && index2 == -2) { /* Second next
vehicle upstream on the right lane */
235             right_tail_vehicle_category2 = long_value;
236         }
237         if (lead_vehicle_category != vehicle_type &&
tail_vehicle_category == vehicle_type && tail_vehicle_distance >=
-50) {
238             lead_platoon = 1;
239             join_platoon = 0;
240         }
241         else if (lead_vehicle_category == vehicle_type &&
tail_vehicle_category == vehicle_type && tail_vehicle_distance >=
-50 && lead_vehicle_distance <= 50) {
242             lead_platoon = 1;
243             join_platoon = 0;
244         }
245         if (lead_vehicle_category == vehicle_type &&
lead_vehicle_distance <= 50) {
246             if (lead_platoon == 0) {
247                 lead_platoon = 0;
248                 join_platoon = 0;
249             }
250             else if (lead_platoon == 1) {
251                 lead_platoon = 1;
252                 join_platoon = 0;

```

```

253     }
254   }
255   if (lead_vehicle_category != vehicle_type &&
tail_vehicle_category != vehicle_type) {
256     join_platoon = 0;
257     lead_platoon = 0;
258   }
259 }
260 else if (vehicle_lane == 1) {
261   lead_platoon = 0;
262   join_platoon = 0;
263   if (index1 == 1 && index2 == 1) {
264     left_lead_vehicle_category = long_value;
265   }
266   if (index1 == 1 && index2 == -1) {
267     left_tail_vehicle_category = long_value;
268   }
269   if (index1 == 1 && index2 == 2) {
270     left_lead_vehicle_category2 = long_value;
271   }
272   if (index1 == 1 && index2 == -2) {
273     left_tail_vehicle_category2 = long_value;
274   }
275 }
276 return 1;
277 case DRIVER_DATA_NVEH_LANE_CHANGE:
278 case DRIVER_DATA_NO_OF_LANES:
279 case DRIVER_DATA_LANE_WIDTH:
280 case DRIVER_DATA_LANE_END_DISTANCE:
281 case DRIVER_DATA_RADIUS:
282 case DRIVER_DATA_MIN_RADIUS:
283 case DRIVER_DATA_DIST_TO_MIN_RADIUS:
284 case DRIVER_DATA_SLOPE:
285 case DRIVER_DATA_SLOPE_AHEAD:
286 case DRIVER_DATA_SIGNAL_DISTANCE:
287 case DRIVER_DATA_SIGNAL_STATE:
288 case DRIVER_DATA_SIGNAL_STATE_START:
289 case DRIVER_DATA_SPEED_LIMIT_DISTANCE:
290 case DRIVER_DATA_SPEED_LIMIT_VALUE:
291   return 1;
292 case DRIVER_DATA_DESIRED_ACCELERATION:
293   desired_acceleration = double_value;
294   return 1;
295 case DRIVER_DATA_DESIRED_LANE_ANGLE:
296   desired_lane_angle = double_value;
297   return 1;
298 case DRIVER_DATA_ACTIVE_LANE_CHANGE:
299   active_lane_change = long_value;
300   return 1;

```

```

301     case DRIVER_DATA_REL_TARGET_LANE:
302         rel_target_lane = long_value;
303         return 1;
304     default:
305         return 0;
306     }
307 }
308
309 /*=====*/
310
311 DRIVERMODEL_API int DriverModelGetValue(long type,
312     long index1,
313     long index2,
314     long *long_value,
315     double *double_value,
316     char **string_value)
317 {
318     /* Gets the value of a data object of type <type>, selected by <
319     index1> */
320     /* and possibly <index2>, and writes that value to <*int_value>,
321     */
322     /* <*double_value> or <**string_value> (object and value
323     selection */
324     /* depending on <type>).
325     */
326     /* Return value is 1 on success, otherwise 0.
327     */
328
329     switch (type) {
330     case DRIVER_DATA_STATUS:
331         *long_value = 0;
332         return 1;
333     case DRIVER_DATA_VEH_TURNING_INDICATOR:
334         *long_value = turning_indicator;
335         return 1;
336     case DRIVER_DATA_VEH_DESIRED_VELOCITY:
337         if (lead_platoon == 1) {
338             if (lead_vehicle_distance <= 50) {
339                 desired_velocity = vp;
340             }
341             else {
342                 desired_velocity = original_desired_velocity;
343             }
344         }
345         else if (join_platoon == 1) {
346             desired_velocity = vp;
347         }
348         else {

```

```

345     desired_velocity = original_desired_velocity;
346 }
347 *double_value = desired_velocity;
348 return 1;
349 case DRIVER_DATA_VEH_COLOR:
350     if (join_platoon == 1) {
351         vehicle_color = CMYK(255, 255, 255, 255);
352     }
353     else if (lead_platoon == 1) {
354         vehicle_color = CMYK(255, 0, 0, 0);
355     }
356     *long_value = vehicle_color;
357     return 1;
358 case DRIVER_DATA_WANTS_SUGGESTION:
359     *long_value = 1;
360     return 1;
361 case DRIVER_DATA_DESIRED_ACCELERATION: {
362     if (join_platoon == 1) {
363         ac = (ka*ap) + (kv*(vp - vf)) + (ks*(s - vf * td));
364         desired_acceleration = max(amin, min(ac, amax));
365     }
366     else if (lead_platoon == 1 || lead_vehicle_category != 660) {
367         /*If the lead vehicle is faster than the desired velocity then
368         this need to not speed up to the lead vehicle for a lead_platton
369         vehicle*/
370         if (lead_vehicle_distance <= 50) {
371             ac = (kv*(vp - vf)) + (ks*(s - vf * 1.4));
372             desired_acceleration = max(amin, min(ac, amax));
373         }
374     }
375     else {
376         if (lead_vehicle_distance <= 50 || lead_vehicle_category
377         != 660) { /*Added a section to cause the CACC vehicle to act like
378         ACC vehicle regardless of it they are leading a platoon or not */
379             ac = (ka*ap) + (kv*(vp - vf)) + (ks*(s - vf * td));
380             desired_acceleration = max(amin, min(ac, amax));
381         }
382     }
383     *double_value = desired_acceleration;
384     return 1;
385 }
386 case DRIVER_DATA_DESIRED_LANE_ANGLE:
387     /*double_value = desired_lane_angle;
388     return 0;
389 case DRIVER_DATA_ACTIVE_LANE_CHANGE:

```

```

385     if (vehicle_lane == 1 && left_lead_vehicle_category ==
vehicle_type && (left_tail_vehicle_category != vehicle_type && (
left_lead_vehicle_category == vehicle_type && (
left_tail_vehicle_distance < -50 && (left_lead_vehicle_distance -
left_tail_vehicle_distance) > 50)))) {
386         if (left_tail_vehicle_distance <= -15 &&
left_lead_vehicle_distance >= 5 && left_lead_vehicle_distance <
50) {
387             active_lane_change = 1;
388         }
389         else {
390             active_lane_change = original_active_lane_change;
391         }
392     }
393     else if (vehicle_lane == 1 && left_lead_vehicle_category !=
vehicle_type && (left_tail_vehicle_category == vehicle_type && (
left_tail_vehicle_category2 != vehicle_type && (
left_tail_vehicle_category2 == vehicle_type && (
left_tail_vehicle_distance - left_tail_vehicle_distance2) < -50)))
) {
394         if (left_tail_vehicle_distance <= -5 &&
left_tail_vehicle_distance > -50 && left_lead_vehicle_distance >=
15) {
395             active_lane_change = 1;
396         }
397         else {
398             active_lane_change = original_active_lane_change;
399         }
400     }
401     else {
402         active_lane_change = original_active_lane_change;
403     }
404     *long_value = active_lane_change;
405     return 1;
406 case DRIVER_DATA_REL_TARGET_LANE:
407     *long_value = rel_target_lane;
408     return 1;
409 case DRIVER_DATA_SIMPLE_LANECHANGE:
410     *long_value = 1;
411     return 1;
412 case DRIVER_DATA_ALLOW_MULTITHREADING:
413     *long_value = 1;
414     return 1;
415 default:
416     return 0;
417 }
418 }
419
420 /*=====*/

```

```
421 |
422 | DRIVERMODEL_API int DriverModelExecuteCommand(long number)
423 | {
424 |     /* Executes the command <number> if that is available in the
425 |     driver */
426 |     /* module. Return value is 1 on success, otherwise 0.
427 |     */
428 |     switch (number) {
429 |     case DRIVER_COMMAND_INIT:
430 |         return 1;
431 |     case DRIVER_COMMAND_CREATE_DRIVER:
432 |         return 1;
433 |     case DRIVER_COMMAND_KILL_DRIVER:
434 |         return 1;
435 |     case DRIVER_COMMAND_MOVE_DRIVER:
436 |         return 1;
437 |     default:
438 |         return 0;
439 |     }
440 | }
441 | /*=====*/
442 | /* End of CACC_Driver_Model.cpp
443 |     */
444 | /*=====*/
```


Appendice B

ACC Driver Model

```
1
2 #include "DriverModel.h"
3
4 /*=====*/
5
6 double current_speed = 0.0;
7 double desired_acceleration = 0.0;
8 double original_desired_velocity = 0.0;
9 double desired_lane_angle = 0.0;
10 long active_lane_change = 0;
11 long rel_target_lane = 0;
12 double desired_velocity = 0.0;
13 long turning_indicator = 0;
14 long vehicle_color = CMYK(255, 255, 255, 255);
15 double lead_vehicle_distance = 999.0;
16 double lead_vehicle_speed_difference = -99.0;
17 double lead_vehicle_length = 0.0;
18 double ka = 0.0;
19 double kv = 0.58;
20 double ks = 0.1;
21 double ac = 0.0;
22 double ap = 0.0;
23 double vp = 0.0;
24 double vf = 0.0;
25 double s = 0.0;
26 double v = 0.0;
27 double td = 1.4; /* 1.4 s gap */
28 double amin = -3.0;
29 double amax = 2.0;
30
31
```

```
32 /*=====*/
33
34 BOOL APIENTRY DllMain(HANDLE hModule,
35     DWORD ul_reason_for_call,
36     LPVOID lpReserved)
37 {
38     switch (ul_reason_for_call) {
39     case DLL_PROCESS_ATTACH:
40     case DLL_THREAD_ATTACH:
41     case DLL_THREAD_DETACH:
42     case DLL_PROCESS_DETACH:
43         break;
44     }
45     return TRUE;
46 }
47
48 /*=====*/
49
50 DRIVERMODEL_API int DriverModelSetValue(long type,
51     long index1,
52     long index2,
53     long long_value,
54     double double_value,
55     char *string_value)
56 {
57     /* Sets the value of a data object of type <type>, selected by <
58     index1> */
59     /* and possibly <index2>, to <int_value>, <double_value> or
60     */
61     /* <*string_value> (object and value selection depending on <type
62     >). */
63     /* Return value is 1 on success, otherwise 0.
64     */
65
66     switch (type) {
67     case DRIVER_DATA_TIMESTEP:
68     case DRIVER_DATA_TIME:
69         return 1;
70     case DRIVER_DATA_VEH_ID:
71         /* reset leading vehicle's data for this new vehicle */
72         lead_vehicle_distance = 999.0;
73         lead_vehicle_speed_difference = -99.0;
74         lead_vehicle_length = 0.0;
75         return 1;
76     case DRIVER_DATA_VEH_LANE:
77     case DRIVER_DATA_VEH_ODOMETER:
78     case DRIVER_DATA_VEH_LANE_ANGLE:
79     case DRIVER_DATA_VEH_LATERAL_POSITION:
80         return 1;
```

```

77 | case DRIVER_DATA_VEH_VELOCITY:
78 |     current_speed = double_value;
79 |     vf = current_speed;
80 |     return 1;
81 | case DRIVER_DATA_VEH_ACCELERATION:
82 | case DRIVER_DATA_VEH_LENGTH:
83 | case DRIVER_DATA_VEH_WIDTH:
84 | case DRIVER_DATA_VEH_WEIGHT:
85 | case DRIVER_DATA_VEH_MAX_ACCELERATION:
86 |     return 1;
87 | case DRIVER_DATA_VEH_TURNING_INDICATOR:
88 |     turning_indicator = long_value;
89 |     return 1;
90 | case DRIVER_DATA_VEH_CATEGORY:
91 | case DRIVER_DATA_VEH_PREFERRED_REL_LANE:
92 | case DRIVER_DATA_VEH_USE_PREFERRED_LANE:
93 |     return 1;
94 | case DRIVER_DATA_VEH_DESIRED_VELOCITY:
95 |     original_desired_velocity = double_value;
96 |     return 1;
97 | case DRIVER_DATA_VEH_X_COORDINATE:
98 | case DRIVER_DATA_VEH_Y_COORDINATE:
99 | case DRIVER_DATA_VEH_TYPE:
100 |     return 1;
101 | case DRIVER_DATA_VEH_COLOR:
102 |     /*vehicle_color = long_value;*/
103 |     vehicle_color = CMYK(255, 0, 255, 0);
104 |     return 1;
105 | case DRIVER_DATA_VEH_CURRENT_LINK:
106 |     return 0; /* (To avoid getting sent lots of
107 |     DRIVER_DATA_VEH_NEXT_LINKS messages) */
108 |     /* Must return 1 if these messages are to be sent
109 |     from VISSIM! */
110 | case DRIVER_DATA_VEH_NEXT_LINKS:
111 | case DRIVER_DATA_VEH_ACTIVE_LANE_CHANGE:
112 |     return 1;
113 | case DRIVER_DATA_VEH_REL_TARGET_LANE:
114 |     return 1;
115 | case DRIVER_DATA_NVEH_ID:
116 | case DRIVER_DATA_NVEH_LANE_ANGLE:
117 | case DRIVER_DATA_NVEH_LATERAL_POSITION:
118 |     return 1;
119 | case DRIVER_DATA_NVEH_DISTANCE:
120 |     if (index1 == 0 && index2 == 1) { /* leading vehicle on own
121 |     lane */
122 |         lead_vehicle_distance = double_value;
123 |     }
124 |     return 1;
125 | case DRIVER_DATA_NVEH_REL_VELOCITY:

```

```
123     if (index1 == 0 && index2 == 1) { /* leading vehicle on own
lane */
124         lead_vehicle_speed_difference = double_value;
125         vp = current_speed - lead_vehicle_speed_difference;
126     }
127     return 1;
128 case DRIVER_DATA_NVEH_ACCELERATION:
129     if (index1 == 0 && index2 == 1) {
130         ap = double_value;
131     }
132     return 1;
133 case DRIVER_DATA_NVEH_LENGTH:
134     if (index1 == 0 && index2 == 1) { /* leading vehicle on own
lane */
135         lead_vehicle_length = double_value;
136         s = lead_vehicle_distance - lead_vehicle_length;
137     }
138     return 1;
139 case DRIVER_DATA_NVEH_WIDTH:
140 case DRIVER_DATA_NVEH_WEIGHT:
141 case DRIVER_DATA_NVEH_TURNING_INDICATOR:
142 case DRIVER_DATA_NVEH_CATEGORY:
143 case DRIVER_DATA_NVEH_LANE_CHANGE:
144     return 1;
145 case DRIVER_DATA_NO_OF_LANES:
146 case DRIVER_DATA_LANE_WIDTH:
147 case DRIVER_DATA_LANE_END_DISTANCE:
148 case DRIVER_DATA_RADIUS:
149 case DRIVER_DATA_MIN_RADIUS:
150 case DRIVER_DATA_DIST_TO_MIN_RADIUS:
151 case DRIVER_DATA_SLOPE:
152 case DRIVER_DATA_SLOPE_AHEAD:
153 case DRIVER_DATA_SIGNAL_DISTANCE:
154 case DRIVER_DATA_SIGNAL_STATE:
155 case DRIVER_DATA_SIGNAL_STATE_START:
156 case DRIVER_DATA_SPEED_LIMIT_DISTANCE:
157 case DRIVER_DATA_SPEED_LIMIT_VALUE:
158     return 1;
159 case DRIVER_DATA_DESIRED_ACCELERATION:
160     desired_acceleration = double_value;
161     return 1;
162 case DRIVER_DATA_DESIRED_LANE_ANGLE:
163     desired_lane_angle = double_value;
164     return 1;
165 case DRIVER_DATA_ACTIVE_LANE_CHANGE:
166     active_lane_change = long_value;
167     return 1;
168 case DRIVER_DATA_REL_TARGET_LANE:
169     rel_target_lane = long_value;
```

```

170     return 1;
171 default:
172     return 0;
173 }
174 }
175
176 /*=====*/
177
178 DRIVERMODEL_API int DriverModelGetValue(long type,
179 long index1,
180 long index2,
181 long *long_value,
182 double *double_value,
183 char **string_value)
184 {
185     /* Gets the value of a data object of type <type>, selected by <
186     index1> */
187     /* and possibly <index2>, and writes that value to <*int_value>,
188     */
189     /* <*double_value> or <**string_value> (object and value
190     selection */
191     /* depending on <type>).
192     */
193     /* Return value is 1 on success, otherwise 0.
194     */
195
196     switch (type) {
197     case DRIVER_DATA_STATUS:
198         *long_value = 0;
199         return 1;
200     case DRIVER_DATA_VEH_TURNING_INDICATOR:
201         *long_value = turning_indicator;
202         return 1;
203     case DRIVER_DATA_VEH_DESIRED_VELOCITY:
204         if (lead_vehicle_distance <= 50) {
205             desired_velocity = vp;
206         }
207         else {
208             desired_velocity = original_desired_velocity;
209         }
210         *double_value = desired_velocity;
211         return 1;
212     case DRIVER_DATA_VEH_COLOR:
213         *long_value = vehicle_color;
214         return 1;
215     case DRIVER_DATA_WANTS_SUGGESTION:
216         *long_value = 1;
217         return 1;

```

```

214     case DRIVER_DATA_DESIRED_ACCELERATION: {
215         if (lead_vehicle_distance <=50) {
216             ac = (ka*ap) + (kv*(vp - vf)) + (ks*(s - vf * td));
217             desired_acceleration = max(amin, min(ac, amax));
218         }
219         *double_value = desired_acceleration;
220         return 1;
221     }
222     case DRIVER_DATA_DESIRED_LANE_ANGLE:
223         /*double_value = desired_lane_angle;
224         return 0;
225     case DRIVER_DATA_ACTIVE_LANE_CHANGE:
226         *long_value = active_lane_change;
227         return 1;
228     case DRIVER_DATA_REL_TARGET_LANE:
229         *long_value = rel_target_lane;
230         return 1;
231     case DRIVER_DATA_SIMPLE_LANECHANGE:
232         *long_value = 1;
233         return 1;
234     case DRIVER_DATA_ALLOW_MULTITHREADING:
235         *long_value = 1;
236         return 1;
237     default:
238         return 0;
239 }
240 }
241
242 /*=====*/
243
244 DRIVERMODEL_API int DriverModelExecuteCommand(long number)
245 {
246     /* Executes the command <number> if that is available in the
247     driver */
248     /* module. Return value is 1 on success, otherwise 0.
249     */
250
251     switch (number) {
252     case DRIVER_COMMAND_INIT:
253         return 1;
254     case DRIVER_COMMAND_CREATE_DRIVER:
255         return 1;
256     case DRIVER_COMMAND_KILL_DRIVER:
257         return 1;
258     case DRIVER_COMMAND_MOVE_DRIVER:
259         return 1;
260     default:
261         return 0;
262     }

```

ACC Driver Model

```
261 }  
262  
263 /*=====*/  
264 /* End of ACC_Driver_Model.cpp  
    */  
265 /*=====*/
```

Bibliografia

- [1] World Health Organization. *The top 10 causes of death*. Dic. 2020. URL: <https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death> (cit. a p. 1).
- [2] European Parliament. *Road fatality statistics in the EU (infographic)*. Apr. 2019. URL: <https://www.europarl.europa.eu/news/en/headlines/society/20190410ST036615/road-fatality-statistics-in-the-eu-infographic> (cit. a p. 1).
- [3] Yao-Ming Yuan, Rui Jiang, Mao-Bin Hu, Qing-Song Wu e Ruili Wang. «Traffic flow characteristics in a mixed traffic system consisting of ACC vehicles and manual vehicles: A hybrid modelling approach». In: *ELSEVIER* 388 (giu. 2009), pp. 2483–2491 (cit. a p. 4).
- [4] P.A. Ioannou e M. Stefanovic. «Evaluation of ACC vehicles in mixed traffic: lane change effects and sensitivity analysis». In: *IEEE (Transactions on Intelligent Transportation Systems)* 6 (2005), pp. 79–89 (cit. a p. 4).
- [5] Bart Van Arem, Jeroen Hogema e Marianne Vanderschuren. «An assessment of the impact of Autonomous Intelligent Cruise Control». In: *Institute for Road Safety Research, SWOV* (mag. 1995) (cit. a p. 4).
- [6] Joel Werf, Steven Shladover, Mark Miller e Natalia Kourjanskaia. «Effects of Adaptive Cruise Control Systems on Highway Traffic Flow Capacity». In: *Transportation Research Record* 1800 (gen. 2002), pp. 78–84 (cit. alle pp. 4, 23).
- [7] Kristoffer Lidström, Katrin Sjöberg, Ulf Holmberg, Johan Andersson, Fredrik Bergh, Mattias Bjäde e Spencer Mak. «A Modular CACC System Integration and Design». In: *IEEE Transactions on Intelligent Transportation Systems* 13.3 (2012), pp. 1050–1061 (cit. a p. 4).
- [8] Steven Shladover. «Automated vehicles for highway operations (automated highway systems)». In: *Proceedings of The Institution of Mechanical Engineers Part I-journal of Systems and Control Engineering - PROC INST MECH*

- ENG I-J SYST C* 219 (feb. 2005), pp. 53–75. DOI: 10.1243/095440705X9407 (cit. a p. 4).
- [9] T. Benz et al. *CHAUFFEUR 2: Final Rep. Deliverable D24*. Rapp. tecn. Stuttgart, Germany: Daimler-Chrysler, 2003 (cit. a p. 4).
- [10] Assad Alam, Ather Gattami e Karl Johansson. «An experimental study on the fuel reduction potential of heavy duty vehicle platooning». In: ott. 2010, pp. 306–311. DOI: 10.1109/ITSC.2010.5625054 (cit. a p. 5).
- [11] Christophe Bonnet e Hans Fritz. «Fuel Consumption Reduction in a Platoon: Experimental Results with two Electronically Coupled Trucks at Close Spacing». In: ago. 2000. DOI: 10.4271/2000-01-3056 (cit. a p. 5).
- [12] S. Tsugawa, S. Kato e K. Aoki. «An automated truck platoon for energy saving». In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2011, pp. 4109–4114. DOI: 10.1109/IR0S.2011.6094549 (cit. a p. 5).
- [13] Martin Treiber, Ansgar Hennecke e Dirk Helbing. «Congested Traffic States in Empirical Observations and Microscopic Simulations». In: *Physical Review E* 62 (feb. 2000), pp. 1805–1824. DOI: 10.1103/PhysRevE.62.1805 (cit. a p. 6).
- [14] Arne Kesting, Martin Treiber e Dirk Helbing. «Enhanced Intelligent Driver Model to Access the Impact of Driving Strategies on Traffic Capacity». In: *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences* 368 (ott. 2010), pp. 4585–605. DOI: 10.1098/rsta.2010.0084 (cit. alle pp. 6, 7).
- [15] B. van Arem, C. J. G. van Driel e R. Visser. «The Impact of Cooperative Adaptive Cruise Control on Traffic-Flow Characteristics». In: *IEEE Transactions on Intelligent Transportation Systems* 7.4 (2006), pp. 429–436. DOI: 10.1109/TITS.2006.884615 (cit. alle pp. 6, 7, 22, 71).
- [16] B. Arem, AP Vos e Marianne Vanderschuren. *THE MICROSCOPIC TRAFFIC SIMULATION MODEL MIXIC 1.3*. Gen. 1997 (cit. a p. 7).
- [17] Lin Xiao, Meng Wang e B. Arem. «Realistic Car-Following Models for Microscopic Simulation of Adaptive and Cooperative Adaptive Cruise Control Vehicles». In: *Transportation Research Record Journal of the Transportation Research Board* 2623 (dic. 2017), p. 5. DOI: 10.3141/2623-01 (cit. alle pp. 20–22, 71).
- [18] K.M. Malone. *CarTALK*. Rapp. tecn. Fin. Rep. 04-7N*141-72541. Delft, The Netherlands: TNO Inro, 2004 (cit. a p. 23).

- [19] V. Milanés, S. E. Shladover, J. Spring, C. Nowakowski, H. Kawazoe e M. Nakamura. «Cooperative Adaptive Cruise Control in Real Traffic Situations». In: *IEEE Transactions on Intelligent Transportation Systems* 15.1 (2014), pp. 296–305. DOI: 10.1109/TITS.2013.2278494 (cit. a p. 35).
- [20] Bryan Higgs, Montasir Abbas e Alejandra Medina. «Analysis of the Wiedemann Car Following Model over Different Speeds using Naturalistic Data». In: set. 2011 (cit. a p. 43).
- [21] Bryan Higgs, Montasir Abbas e Alejandra Medina. «Analysis of the Wiedemann Car Following Model over Different Speeds using Naturalistic Data». In: set. 2011 (cit. a p. 44).
- [22] Richard Dowling, Alexander Skabardonis e Vassili Alexiadis. *Traffic Analysis Toolbox Volume III: Guidelines for Applying Traffic Microsimulation Modeling Software*. Rapp. tecn. FHWA-HRT-04-040. U.S. Department of Transportation, Federal Highway Administration, giu. 2004 (cit. a p. 46).