



Intelligent Lane Change Strategies

Rovetta Alberto - Tamburini Alberto

Abstract

Il presente elaborato rappresenta un primo tentativo di progettare una manovra che emuli il cambio di corsia di un intero platoon in sicurezza. Tale manovra è stata programmata, simulata e testata all'interno di Omnet++, col supporto dei framework Sumo, Veins e Plexe.

La manovra è costituita da una prima fase di handshake, durante la quale tutti i membri del platoon verificano di poter eseguire il cambio di corsia in sicurezza. In caso di esito positivo la manovra si svolge per poi concludersi controllando che tutti i membri abbiano svolto correttamente il cambio di corsia. Altrimenti viene immediatamente interrotta.

Come punto di partenza sono state utilizzate le librerie di funzioni di plexe; da ciò derivano alcune limitazioni, prima fra tutte l'impossibilità di simulare una traslazione continua fra la corsia di partenza e quella di destinazione. Di fatto l'atto di cambiare corsia è atomico e dunque istantaneo.

Il cambio di corsia si svolge traslando tutte le macchine contemporaneamente come fosse una traslazione di un corpo rigido, verso la corsia di destinazione. Tale approccio è definito "Single Body Lane Change" (SBLC), ed è contrapposto all'approccio "Common Trajectory Lane Change" (CTLC) dove le macchine cambiano corsia una per volta.

La manovra è stata testata con successo su un percorso rettilineo sia in presenza che assenza di macchine interferenti, non necessariamente autonome.

Indice

1	Descrizione Progetto	3
2	Progetto Del Protocollo	5
2.1	Implementazione	7
3	Risultati	8
4	Conclusioni E Sviluppi Futuri	9
A	Manuale D'Uso	10

1 Descrizione Progetto

Lo scopo della manovra è quello di far sì che l'intero platoon cambi corsia contemporaneamente, evitando non solo collisioni con altri veicoli presenti nella corsia di destinazione, ma anche cercando di evitare il cambio di corsia se esso dovesse comportare la rottura del platoon stesso.

A tal scopo non è stato possibile implementare una manovra costituita dal solo invio della richiesta agli altri membri del platoon, ma è stato necessario prevedere un protocollo di comunicazione tra colui che inizia la manovra e i membri restanti del platoon. Infatti, nel caso nel quale la corsia di destinazione fosse occupata da un veicolo, e non fosse presente un controllo preliminare da parte di tutti i membri del platoon, solo i membri abilitati ad effettuare il cambio di corsia in sicurezza cambierebbero effettivamente la corsia; il risultato sarebbe quello di avere una parte del platoon sulla nuova corsia, l'altra nella corsia di partenza, spezzando di fatto il platoon.

Come prima scelta progettuale è stato deciso che l'unico membro abilitato ad iniziare la manovra è sempre e solo il leader del platoon. Il codice è stato testato anche facendo partire la manovra da un membro qualunque e nei casi analizzati l'esito della manovra non cambiava. Dunque pare verosimile che la scelta del leader non fosse strettamente necessaria. Tuttavia, si è ritenuto che fosse la scelta migliore, in quanto ci è parso irragionevole che un follower potesse decidere il comportamento dell'intero platoon. Una via di mezzo, poteva essere quella di permettere ad un follower di chiedere al leader di inizializzare la manovra. Non avendo trovato un caso d'uso tale da giustificare lo sforzo implementativo, si è deciso di tralasciare quest'ultima opzione.

Dunque, la manovra viene inizializzata dal leader. Esso decide anzitutto la corsia di destinazione secondo i seguenti criteri:

- la corsia deve esistere;
- la corsia deve essere adiacente a quella corrente;
- nel caso in cui ci fossero più corsie di destinazione ne viene scelta una randomicamente.

Ciò implica che potrebbe essere scelta una corsia che in realtà è occupata. In un caso più realistico la scelta della corsia dovrebbe essere deterministica e dipendere da criteri esterni quali ad esempio:

- la scelta di prendere un'uscita dall'autostrada;
- la presenza di veicoli più lenti nella medesima corsia, di fatto un sorpasso.

I casi citati potrebbero essere implementati sfruttando la presente manovra.

Viene quindi effettuata la verifica che la corsia di destinazione sia libera a prescindere dal fatto che il veicolo interferente sia autonomo o guidato da un umano, in quanto si basa sulla rilevazione della presenza di un oggetto fisico. Se la corsia scelta è libera per il leader, allora questo chiederà a tutti i follower di verificare che essa sia libera anche per loro. Costoro dovranno rispondere al leader con un ack se la corsia è libera; altrimenti, colui/coloro che non rilevano la corsia libera, inoltrerà/inoltreranno agli altri membri del platoon immediatamente un messaggio di abort della manovra, senza per questa ragione rompere il platoon. Di fatto la manovra incarna un'architettura master-slave, in cui il leader è il master, mentre i follower sono gli slave.

Per quanto riguarda l'operazione di abort, essa può essere inoltrata in qualunque momento da parte di un qualsiasi membro del platoon durante l'esecuzione della manovra. Due sono le situazioni principali in cui è stato previsto l'abort:

- quando, prima di iniziare lo spostamento laterale, uno dei follower rileva un'auto sulla corsia di destinazione;
- quando il leader non riceve tutti gli ack dai follower entro un tempo prestabilito.

Un veicolo che ritiene necessario effettuare l'abort della manovra dovrà avvisare tutti gli altri membri del platoon dell'evento; in seguito tornerà allo stato idle. Tutti gli altri veicoli, ricevuto il messaggio di abort dovranno interrompere immediatamente la manovra, qualunque sia il loro stato e ruolo all'interno del platoon e tornare anch'essi nello stato di idle.

Altri abort, ovvero abort non derivanti dalle operazioni proprie della manovra corrente, non sono gestiti direttamente all'interno del protocollo, in quanto andrebbero gestiti da parti terze del software più generali rispetto alla specifica manovra.

Per testare la manovra sono state pensate due possibili casistiche:

- un veicolo esterno che percorre la corsia di destinazione ad una distanza tale da non interferire con la manovra stessa;
- un veicolo esterno che percorre la corsia di destinazione ad una distanza tale da interferire con la manovra stessa.

Nel primo caso avviene con successo il cambio di corsia del platoon, mentre nel secondo il cambio di corsia non viene effettuato in quanto non vi sono le ipotesi per effettuare la manovra in sicurezza.

2 Progetto Del Protocollo

LEADER

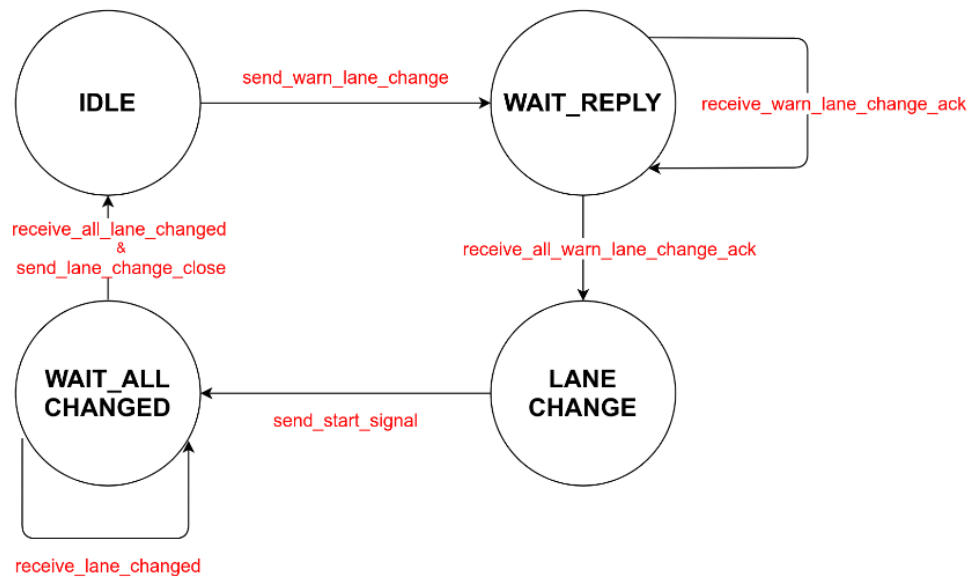


Figura 1: Finite State Machine (FSM) di un generico leader di un platoon.

FOLLOWER

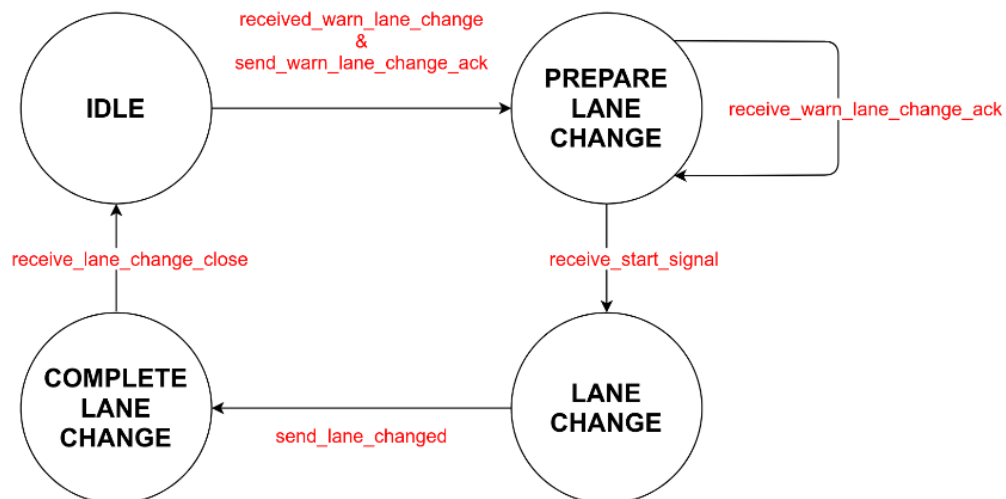


Figura 2: FSM di un generico follower di un platoon.

Segue una descrizione di come avviene la manovra:

1. Il leader sceglie la corsia di destinazione, verifica che questa sia libera. Se libera allora si invia *"warn-lane-change"*, contenente la corsia di destinazione, ci si sposta nello stato *"WAIT-REPLY"* e si procede al punto 2, altrimenti viene annullata la manovra.
2. Ogni follower riceve *"warn-lane-change"*, verifica che la corsia di destinazione sia libera. Se libera invia al leader *"warn-lane-change-ack"* e ci si sposta nello stato *"PREPARE-LANE-CHANGE"*, altrimenti abort.
3. Se il leader riceve tutti gli ack entro un tempo limite di 0.2 secondi, allora passa allo stato *"LANE-CHANGE"* invia il messaggio *"start-signal"* a tutti i follower, cambia corsia e infine passa allo *"WAIT-ALL-CHANGED"*. Altrimenti invia abort a tutti e torna in IDLE.
4. Ogni follower riceve *"start-signal"*, passa allo stato *"LANE-CHANGE"* effettua il cambio di corsia, informa il leader del cambio attraverso il messaggio *"lane-changed"* e passa allo stato *"COMPLETE-LANE-CHANGE"*.
5. Se il leader riceve *"lane-changed"* da tutti i follower entro 0.2 secondi allora invia *"lane-change-close"* a tutti i follower e ritorna allo stato *"IDLE"*. Altrimenti invia abort a tutti e torna in *"IDLE"*.
6. Ogni follower riceve *"lane-change-close"* e ritorna allo stato *"IDLE"*.

Affinché la manovra possa iniziare è necessario che tutti i veicoli siano nello stato *"IDLE"*.

Non sono stati testati scenari in cui vi è una concorrenza nei cambi di corsia di diversi platoon e/o veicoli. Infatti, potrebbero verificarsi collisioni nel caso in cui più platoon decidessero di effettuare nello stesso momento la manovra verso la stessa corsia.

Non sono stati presi in considerazione i casi nei quali fallisce la trasmissione dei messaggi. Questo può causare problemi ad un follower nello stato *"COMPLETE-LANE-CHANGE"* che, a causa del mancato arrivo del messaggio *"lane-change-close"* non potrà tornare allo stato *"IDLE"*. In tal caso il simulatore esce, come avviene nelle manovre di esempio di plexe.

2.1 Implementazione

All'interno del codice, la manovra si presenta come un esempio dal nome *"laneChangeManeuver"*. L'implementazione è stata ottenuta prendendo ispirazione dalla struttura della manovra *"JoinManeuver"* presente in plexe.

La manovra è costituita da una *"app"*, implementata all'interno di *LaneChangePlatooningApp*. Essa permette di inizializzare una simulazione, selezionando uno scenario appropriato, e di mantenere il platoon grazie ai metodi ereditati dalla classe *"GeneralPlatooningApp"*. Qui vengono schedulati i timeout della manovra citati nelle sezioni precedenti.

Lo scenario, implementato all'interno di *"LaneChangeManeuverScenario"*, si occupa di inizializzare il platoon e la macchina interferente. Inoltre è qui che viene schedulato l'avvio della manovra (al secondo 3) tramite l'invio di un self message da parte del leader. Inoltre, tale classe è un'estensione di *"BaseScenario"* e dunque ne eredita le funzionalità.

Come traffic manager è stato utilizzato *"PlatoonsTrafficManager"*, già presente all'interno delle librerie di plexe il quale ci consente di inizializzare correttamente il platoon.

All'interno della classe *"LaneChange"* è implementato il protocollo descritto nelle sezioni precedenti. Qui sono presenti gli stati, i passaggi di stato e gli invii di messaggi necessari all'esecuzione della manovra. La logica si basa sul metodo *onManeuverMessage*, che associa ad ogni messaggio le operazioni relative allo stato corrispondente. Tali operazioni sono divise in due parti essenziali:

- la verifica delle precondizioni, ad esempio che il membro ricevente sia nello stato corretto e abbia all'interno del platoon un ruolo compatibile con l'operazione;
- effettuare le azioni proprie dello stato se le precondizioni sono soddisfatte.

Essa è un'estensione della superclasse *"Maneuver"*.

Oltre ad esse sono state definite le classi per i messaggi. Tutte queste estendono la classe *"ManeuverMessage"*. L'unico messaggio che contiene informazioni aggiuntive è *"warn-change-lane"*, contenente la corsia di destinazione.

3 Risultati

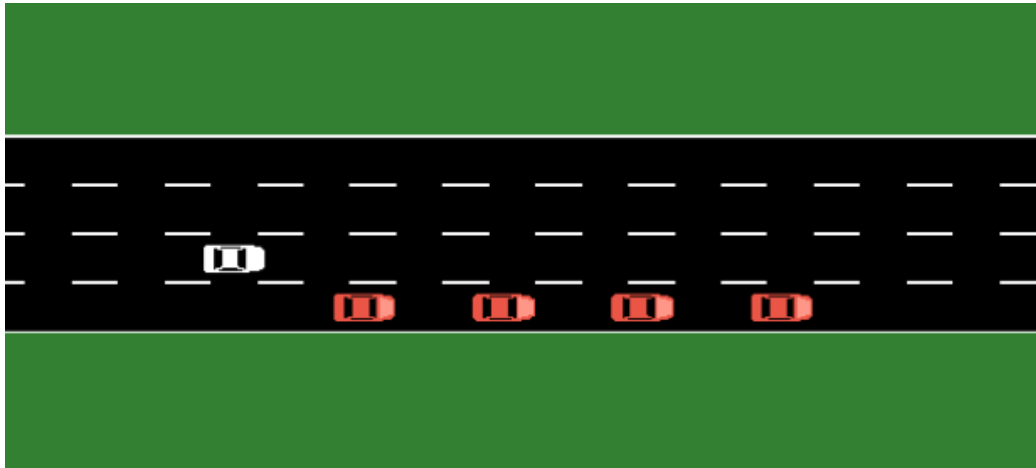


Figura 3: Inizio della simulazione della manovra.

La manovra è stata testata su due scenari selezionabili dall'ini file tramite il campo `*.node[*].scenario.otherCarController` che definisce il tipo di controllore di un veicolo che viaggia nella corsia di destinazione del platoon:

- selezionando "CACC" il veicolo interferente viaggia alla stessa velocità del platoon, ed a una distanza tale da non intralciare la manovra. Questo consente di effettuare il cambio di corsia in sicurezza;

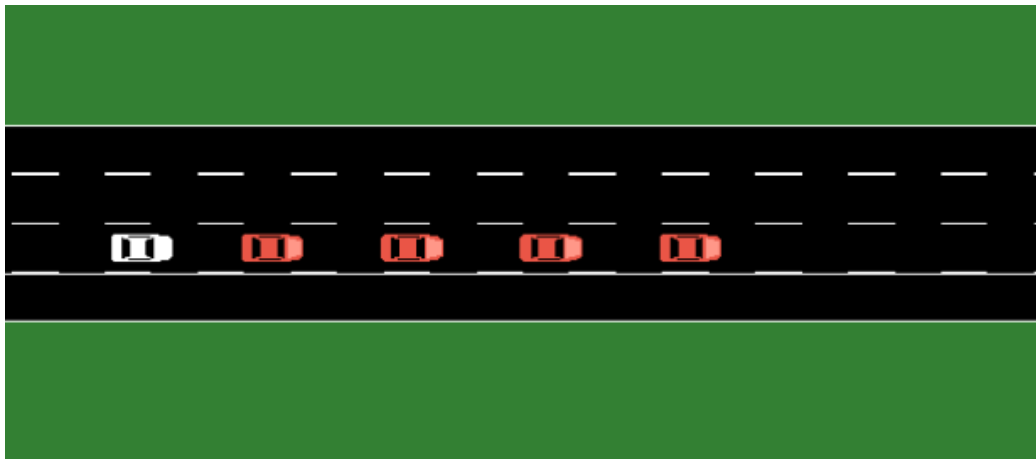


Figura 4: Completamento manovra con "CACC".

- selezionando *"DRIVER"* il veicolo interferente viaggia ad una velocità maggiore rispetto al platoon, occupando di fatto la corsia di destinazione della manovra nel momento in cui andrebbe effettuata. Questo porta all'abort della manovra, che dunque viene conclusa fallendo in sicurezza.

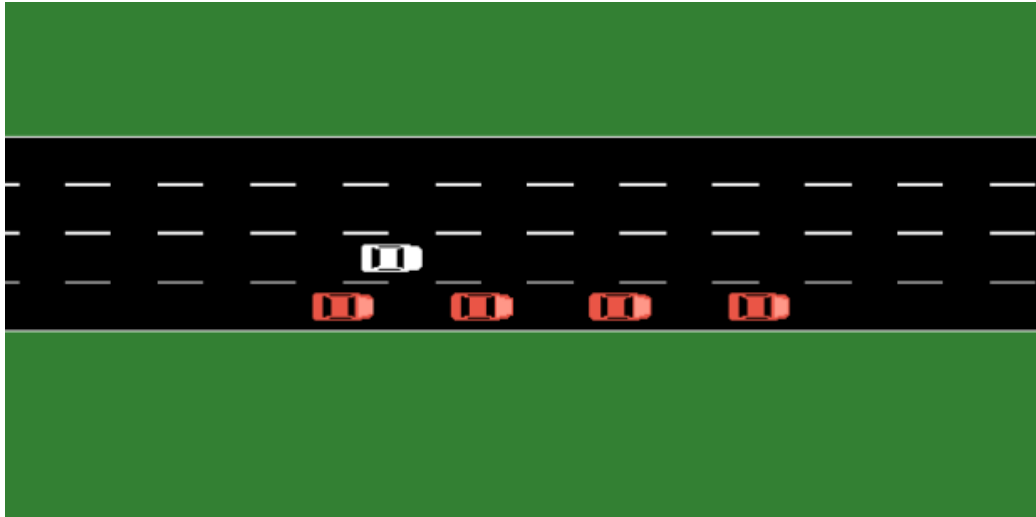


Figura 5: Abort della manovra con *"DRIVER"*.

Avendo utilizzato come base il codice di *"JoinManeuver"*, ne abbiamo ereditato anche alcuni aspetti negativi. Infatti, il numero di veicoli nel platoon rimane fisso a 4 e non può essere modificato dall'ini file.

L'implementazione del protocollo e la documentazione dei messaggi è disponibile al seguente [link](#).

4 Conclusioni E Sviluppi Futuri

In conclusione la manovra si comporta come previsto all'interno degli scenari testati. Alcune migliorie che potrebbero essere implementate nella manovra sono:

- scambio di messaggi con veicoli esterni al platoon in modo da evitare possibili collisioni nel caso di manovre simultanee;
- effettuare la manovra autonomamente secondo certi criteri come per esempio la velocità del veicolo esterno che precede il platoon sotto una soglia fissata;
- scelta della corsia di destinazione secondo opportuni criteri e non in maniera casuale.

A Manuale D'Uso

Per utilizzare la manovra da noi implementata effettuare il clone del repository di github fornito al link precedente, oppure facendo

```
> git clone https://github.com/Tambup/platoon-lane-change
```

Importare quindi il progetto "platoon-lane-change" e "veins" all'interno di "Omnet++" in un nuovo workspace.

Andare quindi nell' *.ini file che si trova all'interno del percorso

plexexamples/laneChangeManeuver/ ed effettuare quindi la simulazione del progetto.

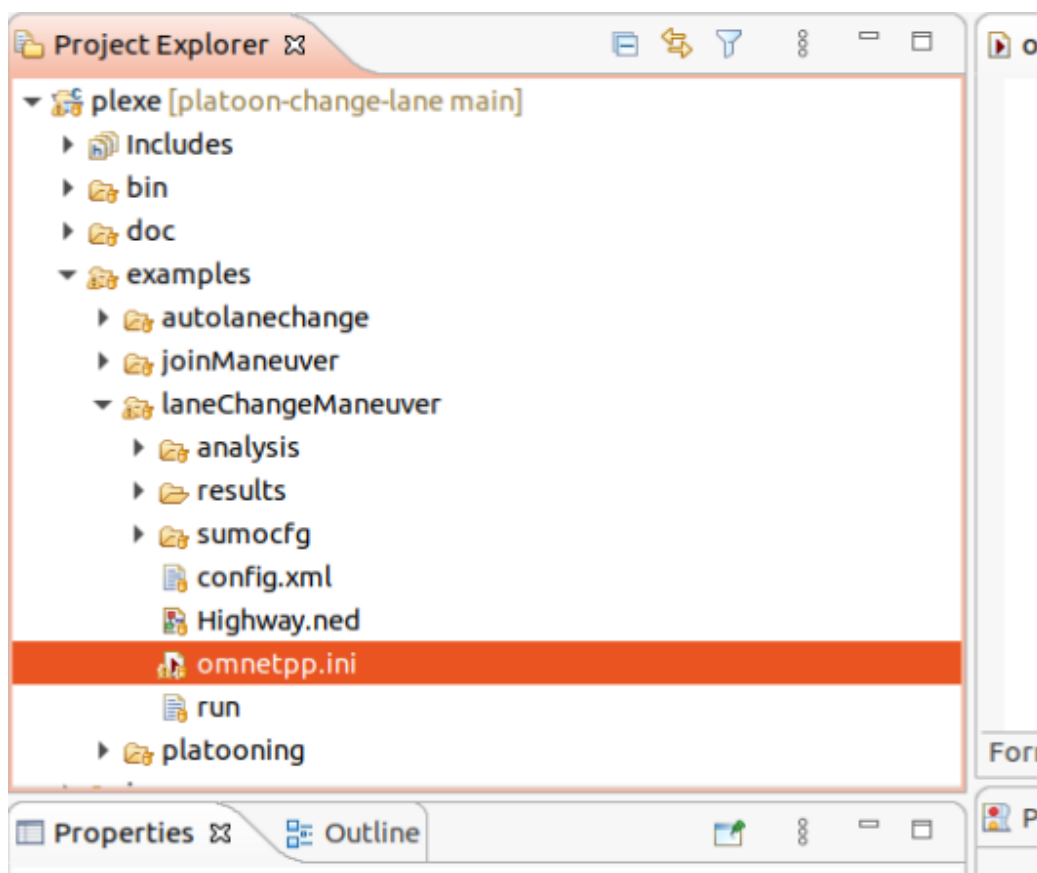


Figura 6: Percorso dove è situato il file ini.

CopyRight



Quest'opera è licenziata dagli autori con una licenza Creative Commons 4.0 CC BY-NC-SA Attribuzione - Non commerciale - Condividi allo stesso modo (<https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode.it>).

È possibile riusare liberamente il materiale per opere derivate nei limiti della licenza e con l'attribuzione dovuta.