HAL
open science

Cooperative Lateral Maneuvers Manager for Multi-Autonomous Vehicles

Mohamad Ali Assaad, Reine Talj, Ali Charara

To cite this version:

Mohamad Ali Assaad, Reine Talj, Ali Charara. Cooperative Lateral Maneuvers Manager for Multi-Autonomous Vehicles. IEEE International Conference on Systems, Man, and Cybernetics (SMC 2018), Oct 2018, Miyazaki, Japan. pp.2651-2656. hal-03335568

HAL Id: hal-03335568

https://hal.science/hal-03335568v1

Submitted on 6 Sep 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Cooperative Lateral Maneuvers Manager for Multi-Autonomous Vehicles

Mohamad Ali Assaad*, Reine Talj*, and Ali Charara*
*Sorbonne Universités, Université de technologie de Compiègne, CNRS, Heudiasyc UMR 7253.
57 avenue de Landshut, 60203 Compiègne, France.
Email: mohamad-ali.assaad, reine.talj, ali.charara@hds.utc.fr

***Abstract*—Vehicle automation opened the door for a myriad of new applications that target vehicles. Longitudinal maneuvers are well studied in the literature, whereas lateral maneuvers studies are scarce in comparison. Maneuvers such as lane changing on highways are complex, and require gathering information about the different neighbors in the vicinity of the subject vehicle. Vehicle-to-vehicle (V2V) communications provided new ways for acquiring such information. Cooperation is proven to be useful when performing such maneuvers. Using V2V communication, and Systems of Systems (SoS) principles, this paper presents a Cooperative Lateral Maneuvers Manager (CLMM), an integrated system in autonomous vehicles, that implements the requirements of the Cooperative maneuvers manager for autonomous vehicles (CMMAV), a SoS that enables cooperation between autonomous vehicles to perform their desired lateral maneuvers. CLMM was validated using Anylogic*, a multi-agent simulator, and the results show that it respects the requirements of the system.**

***Index Terms*—Systems of systems, Intelligent transportation systems, autonomous vehicles, lateral maneuvers.**

I. Introduction

Overtaking, lane changing, platooning and merging are all different types of maneuvers constantly performed by vehicles on roads. These maneuvers are very complex in that they require the driver to perform multiple tasks in a short period of time, from the perception of the surrounding vehicles, to the estimation of speeds and distances. The main benefit of driving automation is to perform these tasks in a safer manner, and to eliminate human errors from the process of driving and decision-making. Cooperation using Vehicle-to-X (V2X) communications was proven to be useful in the literature [1], as well as in live demonstrations [2]. Acquiring an accurate knowledge about the states and intentions of the surrounding vehicles enables a better decision-making process about the next action for a vehicle.

Driving automation and V2X communication allowed the development of new applications that aim to improve traffic flow and safety on roads, namely intelligent transportation systems (ITS). ITS regroup different families of systems and applications [3]. From the advanced public transportation systems (APTS), which gathers all applications and systems that target the public transportation domains, to the advanced travelers information systems (ATIS), which provide different types of information about traffic to any type of interested users, all these systems share the objective of improving safety and traffic flow. Advanced vehicles control and safety systems (AVCSS) are the applications that targets vehicles internally. They regroup systems such as adaptive cruise control and lane keeping assistance and any other system that target the internal behavior of vehicles. To contribute to this family of systems, we introduce in this paper the Cooperative Lateral Maneuvers Manager (CLMM), the physical form of the Cooperative Maneuvers Manager for Autonomous Vehicles (CMMAV) framework. CMMAV's objective is to enable cooperation between equipped vehicles before, during, and after a certain maneuver (overtaking, lane changing). CLMM does not control the subject vehicle on a low level, but rather intervenes on the decision-making level to decide whether the desired maneuver is safe to be executed, or to negotiate with neighbors to achieve this.

The remainder of the paper is divided as follows: Section II describes the related works found in the literature, and introduces our contributions to ITS, it is then followed by a system description, which describes the CLMM in the context of CMMAV in Section III, and the architecture of the CLMM in Section IV. To demonstrate how the CLMM works Section V presents two subsections: Subsection V-A which describes a case study that show how the system reacts in a particular scenario, and Subsection V-B which describes the simulations performed to validate the systems behavior. Finally, we end the paper with a conclusion in Section VI.

II. Related Works

Studies concerning lateral maneuvers such as overtaking and lane changing treat the topic from different points of view. From the modeling aspect, Zheng [4] reviews studies about lane-changing models, and distinguishes two groups of models: models that capture the lane-changing decision making process, and others that study the influence of lane-changing on nearby traffic flow. Always in the modeling aspect, Hidas [5] introduces SITRAS, a multi-agent simulation system that could be used to evaluate ITS applications, and studied models treating lane-changing and merging. Petrov et al. [6] modeled autonomous vehicle overtaking, and proposed a nonlinear control scheme that uses only the relative position and orientation with respect to the overtaken vehicle acquired from onboard sensors. Other studies such as [7] contributed to the subject by proposing a vehicle detection assistance

*www.anylogic.com

system, that uses appearances to detect overtaking and receding vehicles. Concerning the actual maneuvers themselves, different control laws were proposed to govern the maneuvers in autonomous vehicles ([8], [9]).

The former studies deal with the maneuvers from the subject vehicle's perspective, and do not treat the cooperative aspect. Vehicle to X communications (vehicle to infrastructure, vehicle to vehicle) opened new potential to automated driving applications, and notably cooperative applications such as platooning and cooperative merging applications, which benefit from the increased accuracy of information acquired by vehicles. Luo et al. [10] proposed a trajectory planning method that uses vehicle-to-vehicle (V2V) communication to plan a reference maneuver trajectory that avoid collisions, while Nie et al. [11] proposed a decentralized approach to the lane-changing maneuver, where V2V communication is used to make decisions about lane-changing in autonomous vehicles.

In this paper, we introduce the CLMM, a system that may be installed into autonomous vehicles, that integrates the requirements and rules of the Cooperative Maneuvers Manager for Autonomous Vehicles (CMMAV). CMMAV is a framework that uses SoS approach, to enable cooperation between equipped vehicles before, during, and after performing a certain lateral maneuver on highways. Our contributions are the following:

1) CLMM is designed to be deployed on any communicating vehicle, particularly autonomous vehicles. It does not perform low-level control of the equipped vehicle, but rather intervene in specific situations to make decisions about some actions.
2) CMMAV is designed to respect the vehicles independence. Vehicle owners could set their desired level of cooperation, and thus preserving their autonomy.

III. System Description

A. CMMAV

CMMAV is a framework built using SoS principles, that uses communication and cooperation between autonomous vehicles to allow them to perform lateral maneuvers on highway in a safer manner and in shorter times. Fig. 1 describes the architecture of CMMAV. It consists of a framework and a physical system that gets installed on autonomous vehicles. CMMAV's two forms are:

- a framework that gathers all the requirements of the system by looking at the stakeholders involved in it;
- a physical system represented by the cooperative lateral maneuvers manager (CLMM), a system that translates the requirements into concrete actions in an equipped vehicle.

The requirements are extracted from different sources that represent actors and stakeholders in the ecosystem of autonomous vehicles, like the traffic laws in the region of interest, comfort requirements of passengers and different standards that must be applied.

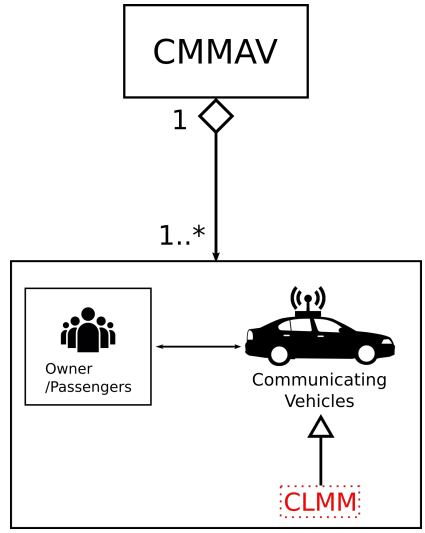

Fig. 1. The architecture of CMMAV

B. CLMM: The physical form

CLMM is designed to be deployed into any vehicle that supports V2X communication. In human-driven communicating vehicles, it serves as a driver assistance system, which informs the driver whether it is safe to perform a certain maneuver or not. In addition, it cooperates with neighbor vehicles that are equipped with the system, to help the driver perform the maneuver in the shortest time possible. In autonomous vehicles, however, it takes control over the vehicle throughout the maneuver to ensure that the execution of the maneuver respects the requirements of the system. The ideal environment in which CMMAV would be deployed is an environment where every vehicle is autonomous, and equipped with CLMM. Realistically, this scenario is far from now, and even when we get to the point where only autonomous vehicles exist, it is unrealistic to assume that all vehicle owners (citizens, makers, service providers) will adopt the same system. Since CMMAV is an SoS framework, it should respect some design principles and heuristics [12]. One particular heuristic is the "Stable intermediate forms", which states that a SoS's framework must ensure the operation of the SoS in any stage of development. To respect this heuristic, CMMAV must take into consideration the differences in its environment. A vehicle that belongs to the CMMAV might encounter 3 types of vehicles during its trajectory: another vehicle that belongs to the CMMAV, a communicating vehicle that does not belong to the CMMAV, and a non-communicating vehicle (that obviously does not belong to the CMMAV). These three encounters are treated by the CMMAV as follows:

1) If the neighbor vehicle belongs to the CMMAV, the subject vehicle cooperates with it to achieve its objectives as explained in the remainder of this paper.
2) If the neighbor vehicle is communicating but does not belong to the CMMAV, the subject vehicle uses the communicated information to perform the maneuver in an egocentric manner, and it communicates its intention

to the neighbor vehicle by sending the appropriate type of messages.

3) If the neighbor vehicle does not communicate, CMMAV will use the information gathered from the subject vehicle's sensors to perform the maneuver, and will communicate its intentions by using light signals.

IV. System Architecture

The CLMM is composed of 3 main modules each grouping a number of blocks that have related functionalities: environment assessment, communication, and decision-making blocks. Fig. 2 shows the different modules and blocks of the CLMM, and the information flow between them.

A. Environment Assessment Module

"Environment Assessment Module" is responsible of gathering and/or processing any data related to the environment, such as neighbor vehicles, lane information, and ego data on the subject vehicle. It contains 3 blocks: Detect Neighbors, Detect Lane, and Lane Status blocks.

1) Detect Neighbors: The objective of this block is to construct a local map representing the relative position of each neighbor that exists in the vicinity of the subject vehicle. The different possible positions are shown in Figure 3. Where L is the Length of the subject vehicle (EV), and d_h is the minimal allowed gap between EV and its front neighbor. This distance is computed using $d_h = 5/9 \times v_e$ [†] where d_h is the minimum allowed gap in meters, and v_e is the EV's velocity in km/h. This distance is the limit at which the relative position of a neighbor vehicle changes from bottom neighbor to bottom-front neighbor (or other possibilities).

A vehicle is considered as a direct neighbor vehicle if it satisfies the following conditions:

- It is in the communication range of the subject vehicle (detected by communication);
- It is either on the same lane, or on an immediate adjacent lane to the subject vehicle's;
- It is the closest among other vehicles that share the same relative position (if two vehicles are in front of the subject vehicle, only the closest one is considered a neighbor).

Algorithm 1 describes the process of assigning the front position to a newly detected neighbor.

2) Detect Lane: The objective of this block is to update regularly the information concerning the lane occupied by the subject vehicle, as well as the adjacent lanes. Table I summarizes the variables returned by the "Detect Lane" block. These variables will then be accessible by any other block (for example the Detect Neighbor block, to check whether a specific neighbor shares the same lane as the subject vehicle's or not).

[†]Article R.412-12 of the French Highway Code.

Algorithm 1 Relative position map construction (front neighbor example)

```
read egoMetaData
read neighbsList
read newNeighborEntry
compute relativeDistance
compute laneNumberDifference
if laneNumberDifference == 0 then
    newNeighborEntry is on the same lane
if relativeDistance < 0 then
    newNeighborEntry is in front
if no front neighbors exists in neighbsList then
    add newNeighborEntry as frontNeighbor
else if frontNeighbor is newNeighborEntry then
    return
else
    compare relativeDistance of
  frontNeighbor & newNeighborEntry
    assign frontNeighbor to the closest
```

TABLE I
Lane's information returned by the Detect Lane block.

Variable Name	Variable Type	Description
Current LinkID	Integer	Unique identifier for the link occupied by the vehicle
LeftLink ID	Integer	ID of the left link, 0 if no left link
RightLink ID	Integer	ID of the right link, 0 if no right link
Current SectionID	Integer	ID of the section containing the link
MaxSpeed	Real	The maximum allowed speed on the link, in km/h
LinkWidth	Real	The width of the link
LinkNum	Integer	If more than one lane in the section, this indicates to which lane this link belongs. Starts at 1 for the far right lane, and increments going to the left

3) Lane Status: To ensure the safety of the vehicle during any lateral maneuver, "Lane Status" block is responsible of computing the time-to-collision with neighbor vehicles. It updates two flags regularly: `left_lane_status` and `right_lane_status`, 2 flags that indicate whether it is safe to change to a certain lane or not. Algorithm 2 describes how we set the `right_lane_status` flag.

Time-To-Collision (TTC) for rear neighbors is computed using the following equation [13]:

$$TTC = \frac{X_e - X_n - Ł}{V_n - V_e} \tag{1}$$

where X_e and X_n are the x coordinates of the subject vehicle and its neighbor respectively measured at the front of the vehicle, in its local frame of reference, and V_e and V_n are their respective speeds, and Ł is the subject vehicle's length.

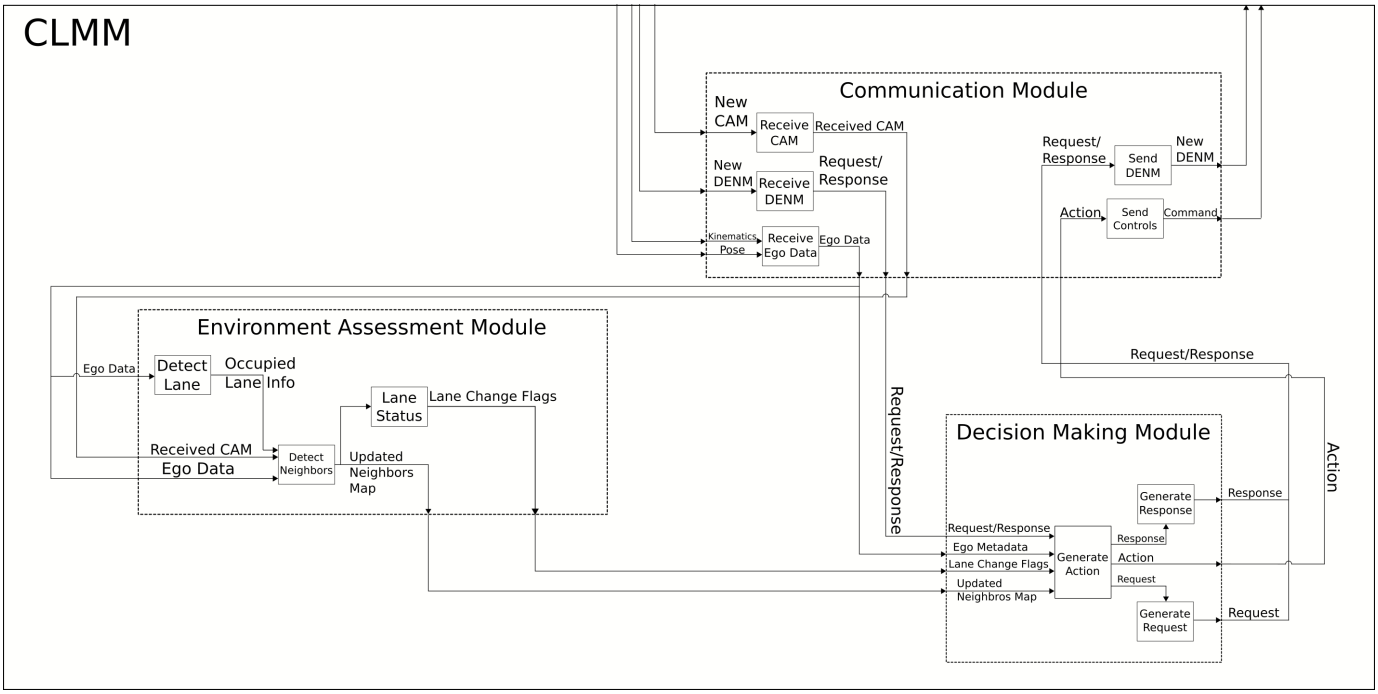

Fig. 2. CLMM's architecture: information flow between different modules and blocks.

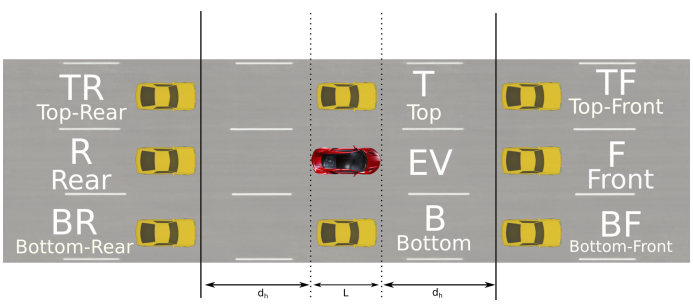

Fig. 3. Local map showing relative positioning with respect to the subject vehicle (EV).

Algorithm 2 Set `right_lane_status` flag

```
read egoMetaData
read neighbsList
if currentLane is first lane then
    return false
else if no neighbors to the right then
    return true
else
    for each neighbor to the right do
        compute TTC
        if TTC ≤ ttcThreshold then
            return false
```

B. Communication Module

The "Communication Module" is responsible of sending and receiving all communications between the CLMM and other modules in the subject vehicles. Since CLMM is an internal module in the subject vehicle, it is not responsible of the actual communications between the subject vehicle and its neighbors, which are done via the main communication module of the vehicle. However, this block sends outgoing messages to the main communication module in the subject vehicle, and receives the appropriate messages from that module. The importance of this module is that it represents the interface between CLMM and the vehicle in which it is installed in. Using this approach, to make CLMM compatible with a particular model of vehicles, it is enough to modify the necessary blocks in this module, and CLMM is ready to work on this vehicle [14]. Communication block contains 5 blocks as shown in Figure 2, each explained below.

1) Send Controls: This block is responsible of sending requests from the CLMM to the different vehicle's module. For example, in the case where the subject vehicle must follow a leading vehicle based upon the request of the CLMM. In this case, the "Send Controls" module sends an activation request to the Adaptive Cruise Control (ACC) module, and specifies the vehicle to follow.

2) Receive CAM: Cooperative awareness messages (CAM) are messages that must be exchanged between all communicating vehicles. These are messages that contain useful information about the sender vehicle, i.e vehicle's type, speed, position, etc. In order for the CLMM to account for neighbor vehicles, it must get each CAM message received by the subject vehicle. Therefore the "receive CAM" block is responsible of transmitting these received messages from the communication module to the CLMM in order to process them.

It is important to note that since every communicating vehicle must send these messages, the CLMM does not contain a

"Send CAM" block, since this functionality is already part of the communicating vehicle itself.

3) Send DENM: Decentralized Environmental Notification Message (DENM) contains information about road alerts or detected events, transmitted to road users in order to inform them about what is going on near. The CLMM uses this type of messages when it requests an action from a neighbor vehicle, and thus, the "Send DENM" block is responsible of sending the relevant information from the CLMM to the main communication module of the vehicle.

4) Receive DENM: This block transmits DENM from the main communication module to the CLMM. These are received in two cases: as a response to a previously sent DENM from the subject vehicle to a neighbor, or as a request to cooperate from a neighbor vehicle.

5) Receive Ego Data: This block communicates with different modules in the vehicle, and gathers the different information related to the state of the subject vehicle (Table II). This information is then transmitted to the "Ego Metadata Block" of the "Environment Assessment Module" for further use.

TABLE II
STATES RELATED TO THE SUBJECT VEHICLE GATHERED BY THE "EGO DATA" BLOCK.

Variable Name	Variable Type
Speed (m/s)	real
Acceleration (m/s^2)	real
Current Lane	integer
Wheel Angle (rad)	real
Vehicle width (m)	real
Vehicle height (m)	real
Vehicle length (m)	real

C. Decision-Making Module

After assessing the environment, and gathering all the necessary information required by the CLMM, the "Decision-Making Module" must decide when it is appropriate on what are the actions that must be performed by the subject vehicle. To do so, this module contains 3 blocks, 2 of them are related to the request/response process between the subject vehicle and neighbors, and the last one is related to the requests from the CLMM to the subject vehicle itself. Decision-making in this module could be done using any decision-making method (such as methods presented in [15]).

1) Generate Request: This block is responsible of generating the requests to the neighbor vehicles when needed. For example, when the subject vehicle initiates an overtaking maneuver, it must inform neighbor vehicles about its intention, therefore this block will generate a request to the "Send DENM" block, which in turn creates a DENM message containing the action intended by the subject vehicle that would be diffused to neighbors.

2) Generate Response: This block is activated whenever the subject vehicle receives a cooperation request from a neighbor vehicle ("keep speed" request for example). It evaluates in this case a utility function based on the deviation from the current speed and steering angle, and returns based on the results two responses: one for the neighbor that requested the cooperation, informing them whether it will cooperate or not, and a second one to the "Generate Action" block in case cooperation was accepted, to perform the requested action in the subject vehicle.

3) Generate Action: This block is responsible of the communications between the "Decision-Making Module" and the "Send Controls" block in the "Communication Module" of the CLMM. It takes actions from the "Decision-Making Module" and transforms them into commands to the subject vehicle. For example, if the subject vehicle requires overtaking, this block will send a "change lane" command to the "Send Controls" block, which in turn sends a command to the "Lane changing system" of the subject vehicle specifying the lane to change to.

V. SIMULATION

A. Case Study

To demonstrate how CLMM works, consider the following scenario (Fig. 4). Vehicle B (subject vehicle) encounters vehicle A, with the respective speeds of $75km/h$ and $65km/h$. If B decides not to overtake A, then it must slow down to A's speed, and wait until A speeds up or changes its lane. When evaluating this case, CLMM decides that B must overtake A. While checking lane change safety flags, it computes the TTC to vehicle C, since it is a back-top neighbor. Assuming a speed of $100km/h$ for C, length of $4.5m$ of B, and a distance of $35m$ between B and C, we get $TTC_{B-C} = 4.3s$. Assuming $TTC_{Threshold} = 12s$, vehicle B could not overtake A until C slows down or changes its lane. To avoid waiting either way, B sends a "Slow down" request to C via CLMM. The speed to which C must slow down would be $92.25km/h$, a $7.75km/h$ difference from the reference speed of C. Depending on the neighbors of C, if it could slow down it will reply with a positive response to B, after which, B initiates the overtaking maneuver. Otherwise, B goes into distance following mode behind A, until a positive response is received from one of its neighbors, or the situation changes in a way that permits B to perform the maneuver.

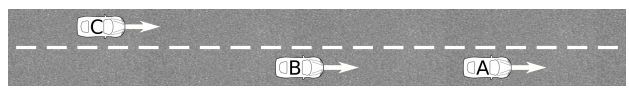

Fig. 4. Vehicle B (subject vehicle) intends to overtake vehicle A.

B. Simulation Results

To validate our framework we used Anylogic, a simulation software that support multi-agent modeling. We chose to use a multi-agent modeling tool to model a SoS because in such simulators, one could build independent agents that simulate the independence of components in SoS. Another reason is its graphical interface, that is useful when one is interested in the global behavior of the system. The first question to answer by simulation was: does the framework respect the basic requirements of the system? In other words, do equipped

vehicles respect speed limits and security gaps? And more importantly, do they perform overtaking when needed while avoiding collisions with other vehicles?
The simulation environment consists of a three-lane, unidirectional freeway. Each lane is defined by an identifier and upper and lower speed limits. The environment randomly generates a vehicle once in a period of $2.5 - 5$ seconds. Vehicles are considered as independent agents, where upon generation, each has its own initial speed and a trajectory defined by a lane's identifier.

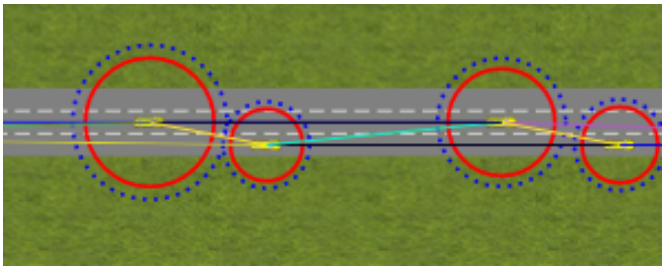

Fig. 5. Safety distances and communication links between neighbors.

We launched the simulation and observed the behavior of the system: for 142 vehicles generated, 32 overtaking maneuvers took place, 3 of which were double overtaking maneuvers. All maneuvers performed respected security requirements, and no collisions were observed. Further simulations will be performed to make sure that the system respects the comfort requirements, and to tune different parameters (such as TTC threshold).

VI. Conclusion and Future Works

In this paper we presented the cooperative lateral maneuvers manager, a system that manages cooperation between equipped communicating vehicles in order to perform lateral maneuvers on highways. The core of the CLMM is the decision-making module, which uses different information from the subject vehicle and its neighbors, to decide the best time to perform a desired maneuver. When a desired maneuver could not be executed because of a conflict with a neighbor, the subject vehicle negotiates with the vehicle in which they are in conflict via the CLMM, in order to eliminate the conflict and perform the maneuver without further delays. CLMM was validated through simulation using Anylogic, a multi-agent simulator. The results show that CLMM respects the safety and comfort requirements of the CMMAV.

In future work, CLMM will be tested using autonomous vehicles of the Heudiasyc laboratory[‡] [16] to validate its operation in a fully autonomous environment (all vehicles are autonomous and equipped with the CLMM), and to tune important variables (such as TTC threshold and overtaking threshold), and in a partially autonomous environment (mix of autonomous and human-driven vehicles) to ensure its operation in a more realistic environment, and to study the effects of communication failure on a such system.

[‡]www.hds.utc.fr

Acknowledgment

This work was carried out and funded in the framework of the Labex MS2T. It was supported by the French Government, through the program "Investments for the future" managed by the National Agency for Research (Reference ANR-11-IDEX-0004-02).

References

[1] M. Guériau, R. Billot, N. El Faouzi, J. Monteil, F. Armetta, S. Hassas, "How to assess the benefits of connected vehicles? A simulation framework for the design of cooperative traffic management strategies," Transportation Research Part C: Emerging Technologies, vol. 67, pp. 266-279, Jun. 2016.
[2] C. Englund et al., "The Grand Cooperative Driving Challenge 2016: boosting the introduction of cooperative automated vehicles," in IEEE Wireless Communications, vol. 23, no. 4, pp. 146-152, Aug. 2016.
[3] Y. J. Wu, P. Lee, "The use of patent analysis in assessing ITS innovations: US, Europe and Japan," Transportation Research Part A: Policy and Practice, vol. 41, no. 6, pp. 568-586, Jul. 2007.
[4] Z. Zheng, "Recent developments and research needs in modeling lane changing," Transportation Research Part B: Methodological, vol. 60, pp. 16-32, Feb. 2014.
[5] P. Hidas, "Modelling lane changing and merging in microscopic traffic simulation,", in Transportation Research Part C: Emerging Technologies, vol. 10, pp. 351-371, Oct. 2002.
[6] P. Petrov and F. Nashashibi, "Modeling and Nonlinear Adaptive Control for Autonomous Vehicle Overtaking," in IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 4, pp. 1643-1656, Aug. 2014.
[7] R. K. Satzoda and M. M. Trivedi, "Overtaking & receding vehicle detection for driver assistance and naturalistic driving studies," 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), Qingdao, pp. 697-702, Oct. 2014.
[8] N. Murgovski and J. Sjöberg, "Predictive cruise control with autonomous overtaking," 54th IEEE Conference on Decision and Control (CDC), Osaka, pp. 644-649, Feb. 2016.
[9] N. A. Nguyen, D. Moser, P. Schrangl, L. del Re and S. Jones, "Autonomous overtaking using stochastic model predictive control," 11th Asian Control Conference (ASCC), Gold Coast, QLD, pp. 1005-1010, Dec. 2017.
[10] Y. Luo, Y. Xiang, K. Cao, K. Li, "A dynamic automated lane change maneuver based on vehicle-to-vehicle communication," in Transportation Research Part C: Emerging Technologies, vol. 62, pp. 87-102, Jan. 2016.
[11] J. Nie, J. Zhang, W. Ding, X. Wan, X. Chen and B. Ran, "Decentralized Cooperative Lane-Changing Decision-Making for Connected Autonomous Vehicles*," in IEEE Access, vol. 4, pp. 9413-9420, 2016.
[12] Rechtin, E., "System Architecting: Creating and Building Complex Systems," Prentice Hall, Englewood Cliffs, NJ, 1991.
[13] A. Laureshyn, A. Svensson, C. Hydén, "Evaluation of traffic safety, based on micro-level behavioural data: Theoretical framework and first implementation," Accident Analysis & Prevention, vol. 42, Issue 6, pp. 1637-1646, Nov. 2010.
[14] G. M. P. Wanderley, M. H. Abel, J. P. Barthès and E. C. Paraiso, "A core architecture for developing systems of systems," IEEE International Conference on Systems, Man, and Cybernetics (SMC), Banff, AB, pp. 141-146, 2017.
[15] L. Claussmann, M. Revilloud, S. Glaser and D. Gruyer, "A study on al-based approaches for high-level decision making in highway autonomous driving," IEEE International Conference on Systems, Man, and Cybernetics (SMC), Banff, AB, pp. 3671-3676, 2017.
[16] P. Xu, G. Dherbomez, E. Hery, A. Abidli and P. Bonnifait, "System Architecture of a Driverless Electric Car in the Grand Cooperative Driving Challenge," in IEEE Intelligent Transportation Systems Magazine, vol. 10, no. 1, pp. 47-59, 2018.