

Evaluating Model Mismatch Impacting CACC Controllers in Mixed Traffic using a Driving Simulator

Maytheewat Aramrattana^{1,4}, Raj Haresh Patel²,
Cristofer Englund^{1,3}, Jérôme Härrä², Jonas Jansson⁴ and Christian Bonnet²

Abstract—At early market penetration, automated vehicles will share the road with legacy vehicles. For a safe transportation system, automated vehicle controllers therefore need to estimate the behavior of the legacy vehicles. However, mismatches between the estimated and real human behaviors can lead to inefficient control inputs, and even collisions in the worst case. In this paper, we propose a framework for evaluating the impact of model mismatch by interfacing a controller under test with a driving simulator. As a proof-of-concept, an algorithm based on Model Predictive Control (MPC) is evaluated in a braking scenario. We show how model mismatch between estimated and real human behavior can lead to a decrease in avoided collisions by almost 46%, and an increase in discomfort by almost 91%. Model mismatch is therefore non-negligible and the proposed framework is a unique method to evaluate them.

I. INTRODUCTION

Commuters today often face issues such as traffic jams, stop-and-go scenarios, collision avoidance maneuvers, etc. In such circumstances, the vehicles need to frequently accelerate and brake to follow the traffic rhythm, which not only increases fuel consumption, but also increases driving discomfort [1]. In the context of connected and automated driving, Cooperative Adaptive Cruise Control (CACC) is posed as a solution to assist the drivers, and ease them from a lot of these issues.

Due to the presence of on-board communication modules that support vehicle-to-vehicle (V2V) and/or vehicle-to-infrastructure (V2I) communication, CACC vehicles have the ability to coordinate control actions [2] with other CACC vehicles. There are many applications of coordinated mobility, including, but not limited to cooperative intersection management [3] and merging on highway [4]. Nonetheless, introduction of CACC also comes with challenges such as perception and localization errors [5], communication delays and packet losses [6], control actuator delays and imperfections [7], etc.

Most algorithms robust to the above issues are developed for a stream of identical CACC vehicles. However, in early

deployment phase of future Cooperative Intelligent Transportation Systems (C-ITS), CACC vehicles will need to share roads with other vehicles with little or no automation like manually driven vehicles (MDVs). Unless dedicated lanes are allocated to CACC vehicles, CACC vehicles will need to properly adapt to the behavior of MDVs and vice versa. In order to adapt aptly, controllers of CACC vehicles (on board or remote) will have to make predictions or assumptions on MDVs' behavior. These assumed behavior will not always match the actual behavior of human drivers. In this paper such conflicts are referred to as *model mismatch*.

Mixed vehicle scenarios have previously been analyzed in [8], [9], but model mismatch related issues is yet to receive more attention. To counter model uncertainties, [10] proposes the use of a robust MPC algorithm. Assumed and actual model of MDVs are represented by IDM+ (IDM+ [11] is an extension of Intelligent Driver Model (IDM) [12]) but with different parameter values to illustrate a model mismatch. Simulations in [10] do not involve human participation and the performance of the algorithm is yet to be tested in real-life conditions.

Because human drivers have diverse driving behavior under different conditions, it is difficult to represent their behavior mathematically. It is thus necessary to evaluate and validate novel algorithms and innovations by involving real human drivers. At the same time, safety of the human participants also needs to be ensured. Thus, we use a driving simulator to obtain real-time control behavior of human drivers under different driving conditions, and ensure that they are not exposed to the risk of real-life traffic accidents. For instance, a driving simulator was integrated with Plexe (Platooning Extension for Veins) [13], to study the behavior of a CACC vehicle platoon when a MDV cuts in [14]; however, model mismatch was not considered.

In this paper, we focus on a braking scenario, where a centralized controller intervenes and controls CACC vehicles to avoid collisions. The main contribution is: the evaluation of a centralized controller in presence of model mismatch with real human drivers using the driving simulator from the Swedish National Road and Transport Research Institute (VTI) in a mixed traffic scenario. Moreover, we show the versatility of the controller by simulating braking coordination in a CACC only traffic scenario and compare its performance with two existing CACC controllers.

The remainder of the paper is organized as follows. Section II presents our proposed centralized algorithm. The simulation scenario is described in Section III, with the

¹ Maytheewat Aramrattana and Cristofer Englund are with the School of Information Technology (ITE), Halmstad University, Halmstad, Sweden. maytheewat.aramrattana@hh.se

² Raj Haresh Patel, Jérôme Härrä and Christian Bonnet are with the Communication Systems Department, EURECOM, Sophia-Antipolis, France. {patel,haerri,bonnet}@eurecom.fr

³ Cristofer Englund is with RISE Viktoria, Gothenburg, Sweden. cristofer.englund@ri.se

⁴ Jonas Jansson is with the Swedish National Road and Transport Research Institute (VTI), Linköping, Sweden. jonas.jansson@vti.se

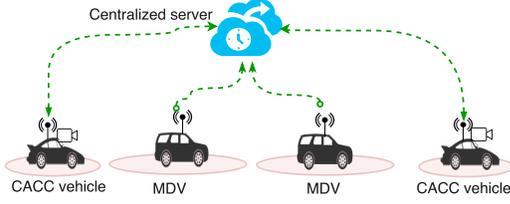


Fig. 1: System model: CACC vehicles have both uplink and downlink; MDVs only have uplink.

results presented in Section IV. Section V concludes the paper with suggestions for future work.

II. CENTRALIZED COLLISION AVOIDANCE CONTROLLER

A. System Model

We focus on a mixed traffic scenario consisting of vehicles with different levels of automation. We assume all vehicles to be connected via wireless communication technologies to a centralized controller as illustrated in Fig. 1. All vehicles send their state parameters (position, velocity, acceleration) to the centralized controller over the “uplink”. We assume that the centralized controller can control CACC vehicles, which are assumed to have the capability to perform the control action as soon as they receive them. MDVs can not be controlled by the centralized controller. Computations take place on this centralized server, considering vehicles’ latest state parameters. Our proposal is to take into account non-automated vehicles such as MDVs, and compute control values for CACC vehicles at each time slot n over a simulation horizon N ($n = 1 \dots N$). The computed controls are then transmitted to CACC vehicles in the “downlink”.

In this work, the centralized controller is optimized for the braking scenario, where combinations of CACC vehicles and MDVs are driving towards an obstacle. An example of this scenario is described in Section III. The centralized controller computes control inputs for CACC vehicles, such that they avoid front and rear-end collisions.

B. CACC vehicle model

CACC vehicles are controlled by the centralized controller, and are modeled in a way that they avoid both front and rear-end collisions. CACC vehicle i ($i \in Z$) ensures distance $d_{i,i-1}$ between itself (i) and the vehicle in front ($i-1$) and distance $d_{i+1,i}$ between the vehicle following it ($i+1$) and itself are positive.¹ Z is the set of all CACC vehicles.

$$d_{i,i-1} > 0 \quad d_{i+1,i} > 0 \quad (1)$$

where

$$\begin{aligned} d_{i,i-1}(n) &= p_i(n) - p_{i-1}(n) - l_{i-1} & n_v \geq i > 1 \\ d_{i+1,i}(n) &= p_{i+1}(n) - p_i(n) - l_i & 1 \leq i < n_v \end{aligned} \quad (2)$$

p_i represents the front end position of the vehicle i , and l_i represents the length of vehicle i . n_v is the total number of vehicles. Note that vehicles $i-1$ and $i+1$ need not be

¹vehicle $i-1$ is followed by i which is followed by $i+1$, and so on, vehicles are moving towards the origin where obstacle is located.

CACC vehicles. We assume perfect localization capability in this work.

C. MDV braking model assumed by centralized controller

At each time slot n in the simulation horizon, when a computation takes place, the centralized controller uses a model to predict braking control values for each MDV over the entire prediction horizon. Different models can be used, we choose a simple but realistic prediction model:

- Before perception response time (Eq. 4 where $\zeta = c \cdot t_{i,1}$):
the centralized controller assumes that the human driver in vehicle i will start braking after a certain assumed perception response time $t_{i,1}$. The driver then increases the braking strength until it reaches a maximum. At maximum braking strength, the vehicle continues to brake until it comes to a halt.
- After perception response time:
 - If braking magnitude is zero (Eq. 4 where $\zeta = 0$) - if vehicle has not started braking, controller assumes that the vehicle will start braking now and continue to increase its braking strength until the vehicle attains maximum braking strength. At maximum braking strength, the vehicle will continue to brake until it comes to a halt.
 - If braking magnitude is increasing (Eq. 5) - the centralized controller assumes that the vehicle will continue to increase its braking strength until the vehicle attains maximum braking strength. At maximum braking strength, the vehicle will continue to brake until it comes to a halt.
 - If braking magnitude is decreasing or constant (Eq. 6) - the model assumes, the vehicle will continue to brake at the previous braking magnitude until halt.

As soon as the velocity reaches zero each vehicle stops braking, regardless of its braking strength. We let prediction horizon and the simulation horizon be of equal duration. Let η represent the time slot in the prediction horizon, $\eta = 1 \dots N$. The above mentioned braking model can be presented mathematically as:

$$u_{i,n} = \begin{cases} \psi_1 |_{\zeta=c \cdot t_{i,1}} & n \leq c \cdot t_{i,1} \\ \psi_1 |_{\zeta=0} & u(n) = 0 \ \& \ n > c \cdot t_{i,1} \\ \psi_2 & \Delta u(n) < 0 \ \& \ n > c \cdot t_{i,1} \\ \psi_3 & \Delta u(n) \geq 0 \ \& \ n > c \cdot t_{i,1} \end{cases} \quad \forall i \in Z^c \quad (3)$$

where ψ_1, ψ_2, ψ_3 are predicted acceleration values, they are explained in Eq. (4), (5), and (6) respectively.

$$\psi_1 = \begin{cases} \eta \cdot \Delta u_i^{min} & \zeta < \eta \leq c \cdot t_{u_i^{min}} \\ u_i^{min} & c \cdot t_{u_i^{min}} < \eta < c \cdot t_{v_i=0} \\ 0 & otherwise \end{cases} \quad (4)$$

$$\psi_2 = \begin{cases} u_i(n-1) + \eta \cdot \Delta u(n) & 1 \leq \eta \leq c \cdot t_{u_i^{min}} \\ u_i^{min} & c \cdot t_{u_i^{min}} < \eta < c \cdot t_{v_i=0} \\ 0 & otherwise \end{cases} \quad (5)$$

$$\psi_3 = \begin{cases} u_i(n-1) & 1 \leq \eta \leq c \cdot t_{v_i=0} \\ 0 & \eta > c \cdot t_{v_i=0} \end{cases} \quad (6)$$

where complementary set of Z which has all MDVs is Z^c . For vehicle i at time slot n , $u_{i,n}$ is the predicted acceleration vector (as predicted horizon is equal to simulation horizon, $u_{i,n}$ would have N values). $u_i(n)$ is the actual value of acceleration; change in acceleration between two time slots $\Delta u(n) = u(n) - u(n-1)$ is the value of jerk; Δu_i^{min} is the maximum permitted decrease in acceleration between two time slots. $t_{i,1}$ is the perception response time in seconds. $t_{v_i=0}$ is the time at which velocity reaches zero; $t_{u_i^{min}}$ is the time when the acceleration of the vehicle reaches maximum braking magnitude. If $n = 1$, $u_i(n-1) = 0$; Values in seconds are multiplied with constant $c = 10$ and converted to time slots (1 second = 10 time slots).

D. Model Mismatch

Model mismatch arises when the actual model is different from the assumed model (described in Section II-C).

In this work, we used two actual models of the MDV, which are derived from either: *i*) the driving simulator; or *ii*) a modified version of intelligent driver model (IDM) [12]. This modified version of IDM involves MDVs implementing controls based on IDM after a particular *perception response time*. Within the perception response time, acceleration of MDV is zero. This *perception response time* is added to imitate response delay of a human driver.²

Model mismatches can only be eliminated if the assumed and the actual model is the same. If there is no model mismatch (and no source of error) then the control inputs computed once should be valid and ensure a collision free maneuver. If predicted control values computed at any particular time slot is used in time slots other than the one in which it was computed it might result into collisions. This is because the error introduced by *model mismatch* would keep accumulating. Thus re-computations of controls is necessary which is the basis of a MPC system.

E. Controller Model

In this work, we propose the use of a centralized controller based on a Model Predictive Control (MPC) system. A MPC controller is able to consider predicted models of different vehicles, different state parameters, various constraints and is able to generate control inputs. In an ideal scenario, at every time slot, fresh state parameters are used to generate fresh control values and are transmitted back to the CACC vehicles. These computations thus take place in a receding horizon method.

The benefits of MPC are as follows: first, at each time slot when control computation takes place, error between predicted model and the actual model can be rectified. Second, MPC based controller generates control inputs over a horizon. In case of computational infeasibility, control inputs generated the previous time slot are to be used.

²perception response time values are drawn from a normal distribution $\mathcal{N}(1.33, (0.27)^2)$ and are capped between 0.8 s and 1.8 s. Please refer to [15] for details on computation of perception response time for MDVs.

Next, we formulate a generic version of the above described centralized controller. The state variable x_i of a vehicle i ($i \in 1 \dots n_v$) is defined as the position p_i , velocity v_i tuple in Eq. (7).

$$x_i = [p_i \ v_i]^T \quad (7)$$

The relation between position, velocity and acceleration is given by Eq. (8). We assume acceleration to remain constant between two time slots.

$$\begin{aligned} \Delta u_i(n+1) &= u_i(n+1) - u_i(n) \\ v_i(n+1) &= v_i(n) + u_i(n)\Delta t \\ p_i(n+1) &= p_i(n) + v_i(n)\Delta t + 0.5 * u_i(n)(\Delta t)^2 \end{aligned} \quad (8)$$

A discrete time linear control system represented by Eq. (9) is used, where values for constants are given by Eq. (10),

$$x_i(n+1) = Ax_i(n) + Bu_i(n) \quad (9)$$

$$A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} (\Delta t)^2/2 \\ \Delta t \end{bmatrix} \quad (10)$$

where Δt is the time between two consecutive time slots n and $n+1$. Vehicle and road constraints in terms of minimum and maximum values of position, velocity, acceleration are accounted for in Eq. (11a), and Eq. (11b),

$$\begin{bmatrix} p_i^{min} \\ v_i^{min} \end{bmatrix} \leq x_i(n) \leq \begin{bmatrix} p_i^{max} \\ v_i^{max} \end{bmatrix} \quad (11a)$$

$$u_i^{min} \leq u_i(n) \leq u_i^{max} \quad (11b)$$

Maximum acceleration and maximum braking is represented by u_i^{max} and u_i^{min} respectively. To ensure a smooth braking, jerk $\Delta u(n)$ is bounded. Note: MDVs are controlled by a human driver, their control values can not be optimized and jerk can not be controlled

$$\Delta u_i^{min} \leq \Delta u_i(n) \leq \Delta u_i^{max} \quad (12)$$

To ensure collision avoidance, the system ensures the distance between vehicles is always positive (Eq. (1)). Eq. (13) ensures the terminal velocity of CACC vehicles reach zero.

$$v_i(N) = 0 \quad (13)$$

Due to the presence of CACCs among MDVs, the behavior of one influences the other. MDV model in this work does not consider the impact of CACC vehicles on MDV. If the actual interdependence can be modeled, the true potential of a centralized controller with CACC vehicles can be evaluated.

The cost function (J) is set to penalize strong deviations in acceleration. We can thus set the MPC based system as:

$$\text{minimize } J = \sum_{i=1}^{n_v} \sum_{n=2}^N (u_i(n) - u_i(n-1))^2 \quad (14)$$

subject to

$$\text{Eq. (1), (3), (7), (8), (9), (10), (11a), (11b), (12), (13)}$$

The above quadratic optimization problem is to be solved on a centralized controller. We use QUADPROG toolbox on MATLAB to solve Eq. (14).

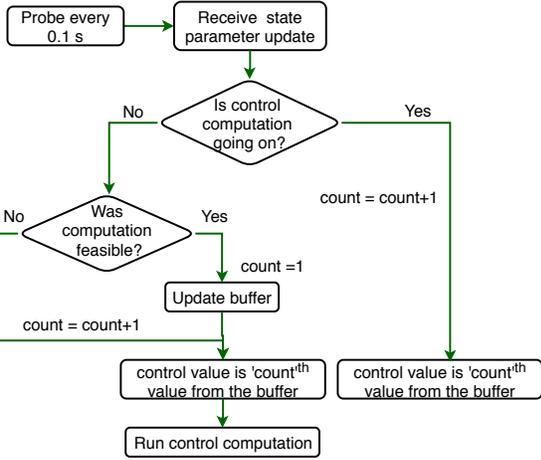


Fig. 2: Flow chart of the control buffer implementation. Initial value of count is one.

F. Control Buffer

The vector of control inputs over the horizon N generated from Eq. (14) is transmitted to CACC vehicles in the downlink. This vector is stored in a buffer. The first value from this buffer is applied by each CACC vehicle.

These computations ideally, should take place with a particular frequency known as the controller update rate. But computations might take longer and controls may not be generated at the desired update rate. Alternately, computations might have terminated but not successful (the optimization problem was infeasible). In such scenario where control inputs for a certain time slot are not computed (due to processing time or infeasibility), we assume CACC vehicles continue to apply next control inputs from the previously received control values stored in the buffer.³ This concept of the buffer implementation with the control algorithm is explained in Fig. 2.

III. SIMULATION FRAMEWORK

Although a generalized multi-vehicle braking system is formulated, in this work we consider a two vehicle braking scenario where a MDV is following a CACC vehicle. We assume the vehicles are on a single lane road, lane change is prohibited. The CACC vehicle implements control inputs generated by Eq. 14, whereas the MDV is driven by either a human driver, or a modified version of IDM. Following are the parameters used with IDM (parameters follow standard notation as in [12]): v_0 , s_0 , T , a , δ , b are 25 m/s, 3 m, 1 s, 1 m/s², 4 and -2 m/s² respectively. Length of all vehicles l is set to 4 m. Initial distance between vehicles is 3 m. The frequency of control computation is defined by the controller's update frequency (set to 10 Hz in this paper; $\Delta t = 0.1s$). $u^{min} = 5.88$ m/s², Δu^{min} , Δu^{max} are set to -0.25 and 0.25 respectively. Control and prediction horizon is equal to simulation horizon N . $N = 100$ unless specified.

³If the control computation is infeasible and the control buffer is empty, CACC vehicles applies brakes with maximum jerk added to previous acceleration value.

Both MDV and CACC vehicles are at halt initially. The first vehicle starts to accelerate with a fixed acceleration of 1 m/s², from 800 meters away from the obstacle. Once the desired speed of 90 km/h (25 m/s) is reached, it cruises until the "notification distance". The notification distance is defined as a distance from the obstacle, where the CACC vehicle detects the obstacle using its sensors, or is notified about the obstacle over V2V/V2I communications. Overview of the scenario is illustrated in Fig. 3. Three different notification distances are used in our study: *i*) 95.9 meters⁴; *ii*) 120 meters; and *iii*) 150 meters. We assume the

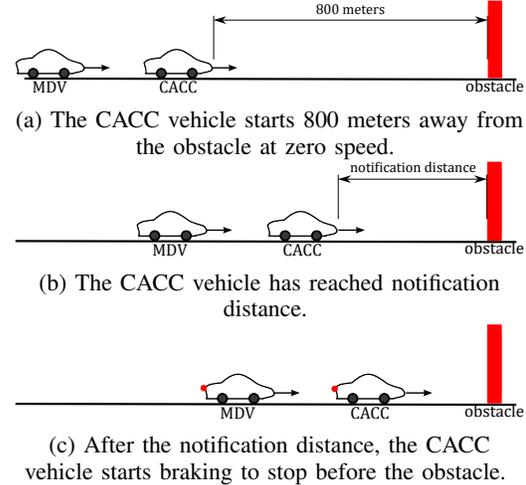


Fig. 3: Overview of the simulated scenario.

proposed algorithm intervenes and helps coordinate a braking procedure as soon as the CACC vehicle is 'notified' about a potential obstacle.

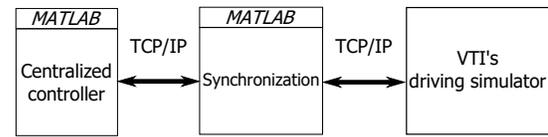


Fig. 4: Connections between MATLAB and the driving simulator.

To involve human drivers in this work, VTI's driving simulation software is used in combination with MATLAB, as illustrated in Fig. 4. All three software blocks in Fig. 4 are running on the same desktop computer. The first MATLAB block, *Centralized Controller*, is executing the control strategy presented in the Section II. The middle block, *Synchronization*, is for synchronizing data exchanges between the driving simulator and the centralized controller over a TCP connection. The data exchange happens every 0.1 second, which is also the simulation time step in MATLAB. The last block, *VTI's driving simulator*, handles the task of fetching inputs from the human driver and displaying positions of the CACC vehicle controlled by MATLAB. The human driver

⁴This is the distance at which at least one DSRC/ITS-G5 safety message would be received with 99.5% probability [16]

uses a gaming steering wheel and pedals to control the MDV. In the scenario where the MDV was controlled by an actual human, we have six participants (2 women and 4 men) involved in the experiments. For each notification distance, each driver is driving the scenario at least two times, in no particular order. The driver was given an instruction to follow the CACC vehicle in front without making any lane changes.

IV. RESULTS AND ANALYSIS

Results from the simulation scenarios described in Section III are presented in this section. The proposed algorithm is evaluated with the following two metrics: *a*) percentage of collision avoidance (whether vehicles halted before the obstacle without collisions); and *b*) discomfort (2 norm of change in acceleration per time slot).

A. Mixed Traffic

The mixed vehicle scenario consists of a MDV following a CACC vehicle, as shown in Fig. 3. We analyze the following *model mismatch* in the braking scenario under two cases:

- Case A: the assumed driving model is the model introduced in section II-C and the actual driving is based on a modified version of IDM
- Case B: the assumed driving model is the model introduced in section II-C and the actual driving inputs are obtained from the driving simulator

Results of the above simulation scenarios are plotted in Fig. 5. We observe in Case A, acceleration and velocity profiles are notably smoother compared to the scenario where the real human is driving the MDV. As one of the evaluation metric is to evaluate discomfort while braking, we compute discomfort values after vehicles are notified at the notification distance. The average discomfort value of Case B is bigger than that of Case A. Moreover, collisions take place when an actual human is driving compared to no collisions when a mathematical model is used. These results are summarized in Table I.

As expected, an increase in notification distance results in higher percentage of collision avoidance. For Case B, the percentage of collisions avoided at 95.9 m, 120 m and 150 m is 35.72%, 57.89% and 60.87% respectively. Overall, 53.57% of collisions were avoided. Whereas in case A, when IDM controls MDV, 100% collisions were avoided.

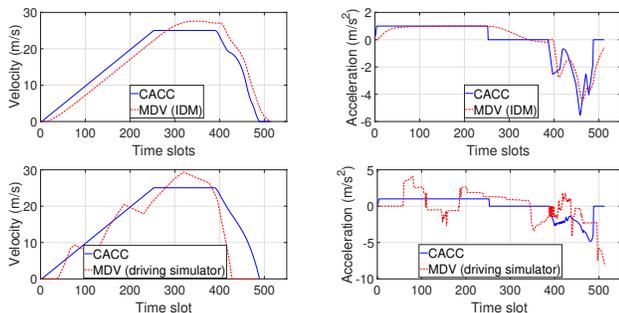


Fig. 5: Mixed vehicle scenario results: top plots show results from case A; bottom plots show results from case B.

TABLE I: Mixed traffic simulation results

	Case A	Case B
Discomfort	6.66	12.76
Collisions avoided (%)	100	53.57

There are several reasons for collisions, due to the differences between assumed value in the centralized controller and the actual value in the driving simulator, e.g. the difference in perception response time, the difference in braking capacity and maximum value of jerk sustainability. In such cases, control optimization computations are infeasible for the assumed values of jerk and the braking capacity. But it may actually be feasible for MDV to brake using the driving simulator.

Furthermore, we observe that the optimization computations takes more time than expected and thus control optimization is not computed at the controller frequency. Computations ideally should take less than 0.1 second to keep real-time performance. However, in our experiments, it takes maximum up to 2 seconds (20 times more) to compute. On an average, we have an optimization computation completed every 0.3 seconds. The previously computed control inputs in the buffer are old and may be from a few seconds ago. These control inputs from the buffer are sometimes not able to avoid collisions.

B. Automated vehicles traffic

Although the proposed controller is designed for mixed vehicle scenarios, it can perform in a CACC-only scenario as well. In this section, the scenario illustrated in Fig. 3, where both vehicles are CACC vehicles, is simulated. To evaluate the performance of our proposed controller, we compare its performance with existing CACC controllers in Plexe such as the one proposed by “*Rajamani*” [17, Chapter 7], and “*Ploeg*” [18].

Simulation results show that all controllers are able to avoid the collision with the standstill obstacle and come to a halt. Figure 6 plots the acceleration profile of different controllers, when the notification distance is 120 meters. From the plot, we observe that *Rajamani* controller tends to apply brakes harder than the *Ploeg* controller. Table II

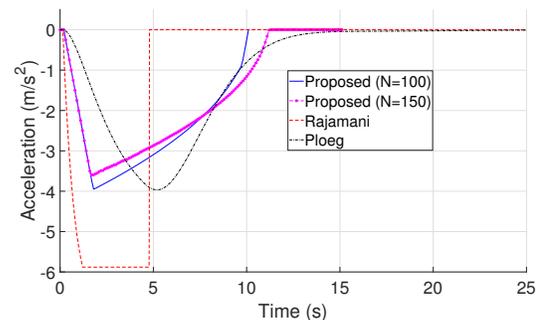


Fig. 6: Acceleration profile of the first vehicle in the automated vehicles traffic scenario, at the notification distance of 120 meters.

summarizes the discomfort values. The discomfort value of the proposed controller is lower (better) than that of the *Rajamani* controller and is similar to the value of the *Ploeg* controller. CACC vehicles based on the *Ploeg* controller take much longer to come to a halt compared to the proposed controller, with approximately the same cost (discomfort). Acceleration profiles can be influenced by changing the simulation horizon N (refer Fig. 6), and the value of discomfort can be reduced (Table II).

TABLE II: Average value of discomfort (2-norm of change in acceleration computed at 10Hz).

	Notification distance (m)		
	95.9	120	150
<i>Rajamani</i> [17]	6.18	6.18	6.18
<i>Ploeg</i> [18]	1.36	0.90	0.80
Proposed ($N = 100$)	1.25	1.15	1.15
Proposed ($N = 150$)	1.24	0.99	0.85

V. CONCLUSIONS AND FUTURE WORK

Difference in the predicted and the actual behaviour of manually driven vehicles (MDVs) gives rise to a model mismatch. In this work, we evaluate the impact of model mismatch on a centralized model predictive control algorithm for braking in mixed traffic. In order to do the evaluation, we propose a simulation framework, which interfaces MATLAB with the VTI's driving simulator.

To simulate a mixed traffic scenario with a CACC and a MDV, first, a mathematical model is used to control the MDV. Second, the controller is interfaced with a driving simulator and a human driver controls the MDV. In the second case, 46% more collisions were recorded and compared to the first case. Moreover, the discomfort of the driver increases by 91%. Based on the significant difference in the results, we conclude that it is necessary to use humans for experimental validation of algorithms. Furthermore, the proposed controller is also compared against *Rajamani* and *Ploeg* controllers in a CACC-only scenario. Each controller is able to avoid collisions although the discomfort values are different. The proposed controller thus adapts to both homogeneous and heterogeneous traffic.

An indepth study of the impact of model mismatch can be carried out by using different assumed and actual models. As future work, tools like the one proposed in [19] can be used to further add other factors such as communication disturbances and extend the scenario to a more diverse mixed traffic scenario. Also, simulations with more than 2 vehicles shall be carried out in the near future. Moreover, the algorithm needs to be made more robust to failures in mixed traffic scenario. Last but not least, to ensure real time simulation of more than two vehicles, the algorithm needs to be improved and optimized.

ACKNOWLEDGEMENT

Raj Haresh Patel is a recipient of a PhD Grant from the Graduate School of the University Pierre Marie Curie (UPMC), Paris. EURECOM acknowledges the support of its

industrial members, namely BMW Group, IABG, Monaco Telecom, Orange, SAP and Symantec.

REFERENCES

- [1] J. Rios-Torres, A. Malikopoulos, and P. Pisu, "Online optimal control of connected vehicles for efficient traffic flow at merging roads," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE, 2015, pp. 2432–2437.
- [2] S. E. Shladover, C. Nowakowski, X.-Y. Lu, and R. Ferlis, "Cooperative Adaptive Cruise Control: Definitions and Operating Concepts," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2489, pp. 145–152, 2015.
- [3] L. Chen and C. Englund, "Cooperative intersection management: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 570–586, Feb 2016.
- [4] J. Rios-Torres and A. A. Malikopoulos, "A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps," *IEEE Transactions on Intelligent Transportation Systems*, 2016.
- [5] R. H. Patel, J. Härrri, and C. Bonnet, "Impact of localization errors on automated vehicle control strategies," *Vehicular Networking Conference (VNC)*, 2017.
- [6] E. van Nunen and J. Verhaegh, "Robust model predictive cooperative adaptive cruise control subject to V2V impairments," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017.
- [7] X. Liu, A. Goldsmith, S. Mahal, and J. K. Hedrick, "Effects of communication delay on string stability in vehicle platoons," in *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*. IEEE, 2001, pp. 625–630.
- [8] G. Schildbach, M. Soppert, and F. Borrelli, "A collision avoidance system at intersections using robust model predictive control," in *Intelligent Vehicles Symposium (IV), 2016 IEEE*. IEEE, 2016, pp. 233–238.
- [9] H. Jiang, J. Hu, S. An, M. Wang, and B. B. Park, "Eco approaching at an isolated signalized intersection under partially connected and automated vehicles environment," *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 290–307, 2017.
- [10] N. Chen, M. Wang, T. Alkim, and B. van Arem, "A robust longitudinal control strategy of platoons under model uncertainties and time delays," in *TRB Annual Meeting 2018*, 2018.
- [11] W. J. Schakel, B. van Arem, and B. D. Netten, "Effects of cooperative adaptive cruise control on traffic flow stability," in *13th International IEEE Conference on Intelligent Transportation Systems*, Sept 2010, pp. 759–764.
- [12] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical review E*, vol. 62, no. 2, p. 1805, 2000.
- [13] M. Segata, S. Joerer, B. Bloessl, C. Sommer, F. Dressler, and R. L. Cigno, "Plexe: A platooning extension for Veins," in *2014 IEEE Vehicular Networking Conference (VNC)*, Dec 2014, pp. 53–60.
- [14] M. Aramrattana, T. Larsson, C. Englund, J. Jansson, and A. Näbo, "Simulation of cut-in by manually driven vehicles in platooning scenarios," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 315–320.
- [15] R. H. Patel, J. Härrri, and C. Bonnet, "A collision mitigation strategy for intelligent vehicles to compensate for human factors affecting manually driven vehicles," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017.
- [16] N. An, T. Gangel, and H. Hartenstein, "VANET: Is 95% probability of packet reception safe?" in *2011 11th International Conference on ITS Telecommunications*, Aug 2011, pp. 113–119.
- [17] R. Rajamani, *Vehicle dynamics and control*, 2nd ed. Springer US, 2012.
- [18] J. Ploeg, B. T. M. Scheepers, E. van Nunen, N. van de Wouw, and H. Nijmeijer, "Design and experimental evaluation of cooperative adaptive cruise control," in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Oct 2011, pp. 260–265.
- [19] M. Aramrattana, T. Larsson, J. Jansson, and A. Näbo, "A simulation framework for cooperative intelligent transport systems testing and evaluation," *Transportation Research Part F: Traffic Psychology and Behaviour*, 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1369847816306635>