2020 IEEE Intelligent Vehicles Symposium (IV)
October 20-23, 2020. Las Vegas, USA

Decentralized Cooperative Collision Avoidance for Automated Vehicles: a Real-World Implementation*

Daniele Bellan[1] and Charles Wartnaby[2]

Abstract— **Connected and automated vehicles provide a new opportunity for collision avoidance, in which several cars cooperate to reach an optimal overall outcome. However, this requires solving the challenging real-time problem of global designing of the joint trajectories that a group of cooperating vehicles should follow to avoid the obstacle and mutually avoid each other. The leaderless method demonstrated here uses the notion of "desired" versus "planned" trajectories, allowing vehicles to influence each other for mutual benefit: each vehicle attempts to avoid independently the desired trajectories of other vehicles, and thus cooperative behavior emerges. Each vehicle is equipped with a model predictive controller that optimizes both its trajectories. This flexible, decentralized method has been executed in a novel demonstration, using two full-sized vehicles on a test track, and results of the fully automated cooperative behavior achieved are presented.**

I. INTRODUCTION

The great development in the field of automated vehicles provides systems that improve the driving experience in terms of safety and comfort thanks to environment perception and electronic actuator controls. Connected and Automated Vehicles (CAVs) additionally share information using local radio links, promising various benefits [1]. Furthermore, bringing together these control and communication functions also allows automating collision avoidance in which multiple cars cooperate to avoid an accident; for example, one car may preemptively swerve to allow room for a second car to avoid an obstacle. This cooperation means a group of CAVs working together can potentially avoid accidents that human drivers acting individually could not. This is especially the case in high-speed highway situations in which braking is inadequate to avoid an accident, and automated steering intervention is required [2]. High-speed accidents such as this, often involving multiple vehicles, tend to have particularly severe outcomes, and cause large-scale traffic disruption if highways are brought to a halt. Cooperatively and safely swerving around a static obstacle would potentially avoid or mitigate such accidents. There is thus a strong incentive to introduce an automated cooperative collision avoidance system. This is the focus of the Multi-Car Cooperative Collision Avoidance (MuCCA) project [3].

A. Challenges

The topic of cooperative collision avoidance has been studied especially in specific scenarios, i.e. overtaking or intersections [4]. Negotiations for lane changes is studied in [5]. In terms of avoiding a static obstacle on the road, the authors of [2] presented a solution for achieving obstacle avoidance at high speed, but without addressing what they called the "negotiations" between vehicles; besides, in their simulation experiment with two vehicles, there is only one "active" actor that autonomously steers to avoid the obstacle.

The authors of [6] evaluated different cooperative motion planning algorithms such as tree search, elastic bands, and Mixed-Integer Linear Programming (MILP). The MILP formulation includes obstacle avoidance as hard constraints of the optimization problem. Such an approach, as the work in [7], is expected to ensure safety but at a high price in terms of computational time due to the binary variables to optimize. In addition, it does not cover scenarios in which a collision cannot be avoided; the problem would yield no numerical solution, yet a real vehicle must be controlled regardless. In addition, another work using binary constraints and a centralized approach for cooperative overtaking does not scale with the number of vehicles and is not fail-safe in case of communication failure [8].

A novel alternative, leaderless, protocol for achieving cooperative maneuvers was used in [9] to improve traffic efficiency. There, each vehicle transmits both a planned trajectory (that which it currently intends to follow) and the desired trajectory (that which it would prefer to follow, were it not blocked by other vehicles). A cooperating vehicle updates its trajectories to accommodate, if possible, the desired trajectories of the other vehicles and then broadcasts its updated plan. At no time is a leader required. This concept has been proved in simulation [10] and it is called Desired-versus-Planned (DVP).

B. This work

Here, the aim is to prove that the DVP approach can be successfully applied to achieve effective cooperative collision avoidance among an arbitrary number of real vehicles, with no leader or central authority. In particular, this work is possibly the first demonstration of two real vehicles cooperatively swerving at the same time around a static obstacle. Collision avoidance was achieved via a "soft" cost function, not via "hard" inviolable constraints. This solved the problem of no numerical solution emerging in scenarios where the collision was inevitable. Moreover, this approach guarantees that the vehicle will produce a trajectory in case of loss of communication: each vehicle is independent of the others. A Model Predictive Controller (MPC) was used to optimize the trajectories while minimizing a multi-objective

*Project supported by Innovate UK.

[1]Daniele Bellan is an Engineer with Applus IDIADA, Cambridge, CB24 6AZ, UK daniele.bellan@idiada.com

[2]Charles E. Wartnaby is a Chief Engineer with Applus IDIADA, Cambridge, CB24 6AZ, UK charlie.wartnaby@idiada.com

cost function. The MPC included terms in the cost function to (weakly) encourage keeping to the current lane and maintain speed. The system is free to find the optimal solution as the situation evolves, which in real life would naturally accommodate unpredictably changing information in a near-collision scenario.

The rest of this paper is organized as follows: Section II explains the desired versus planned concept, and the control system structure. Section III details the optimization problem and the cost function components used. Section IV shows the real-world experiments and presents the obtained results. Section V concludes with suggestions for future work.

II. VEHICLE MODEL AND CONTROL SYSTEM ARCHITECTURE

The MuCCA system is an autonomous emergency feature designed to prevent crashes between vehicles in case of any potential collision scenario on a highway. The vehicles collaborate to avoid the collision, i.e. one or more obstacles. How this collaboration is implemented is the scope of this work. The DVP approach presented in this paper has been applied to two real off-the-shelf automated vehicles, which we will call MuCCA Equipped Vehicles (MuCCA Equipped Vehicles (MEVs)). They were controlled using an in-house-built on-board unit and the specific vehicle platform interface.

A. Assumptions

The communication between vehicles is obtained via Vehicle-to-Vehicle (V2V) messaging, and we assume no packet loss. Further details about V2V will be provided in Section IV. Since the scope of the paper is not to demonstrate the functionality of a completely automated vehicle, perception is not considered. Therefore, the position of the lanes is known in advance and the positions of other vehicles (MEVs or obstacle) are broadcast via V2V. The MPC is a well-known method to handle expected latencies, but for simplification we assume no latency in the control loop and in the messaging.

B. Desired versus Planned

Sharing the trajectories is the key to having a decentralized architecture that empowers vehicles to be adaptable to potentially every scenario.

1) The concept: As said before, the two trajectories are called *planned* and *desired*. The former is the trajectory the vehicle intends to follow and that it is *currently* following, the latter is what the vehicles would follow if not blocked by other vehicles. The desired trajectory thus indicates what the vehicle would prefer to do if only all cooperating vehicles were to make way for it. Furthermore, when forming the desired trajectory, it is fundamental to include other MEVs' desired trajectories. That is necessary to achieve cooperation between more than two MEVs, in some scenarios, e.g. a 3-vehicle cooperative swerve [10].

2) Importance of desired trajectory: Consider a 3-lane motorway with two MEVs moving at the same speed side by side, one in the leftmost lane, the second in the central lane. The leftmost MEV is approaching an obstacle in its lane: its planned trajectory may be to brake hard, but nevertheless suffer a severe collision; its desired trajectory may be to harmlessly swerve, but that maneuver is blocked unless the other MEV makes room for it. The difference in cost between its planned and desired trajectories will be considerable in this situation. Conversely, the second MEV may find its optimal planned trajectory is to continue in its lane. However, when having received the desired trajectory from the first MEV, it updates its planned trajectory to swerve to the right, while its desire would still be continuing in its lane. It this way, space becomes available to the first MEV to swerve around the obstacle. The second MEV decided to swerve to the right because the first one broadcast a *signal* of urgency of its desired trajectory to be accommodated. Hence the desired trajectory costs for the second MEV were scaled by a multiplicative factor, the *importance*, broadcast by the originator, reflecting the priority with which its desired trajectory should be respected. A third MEV present may have divergent desired and planned trajectories, yet with little cost difference. In that case, the *importance* of accommodating its desire is small, and priority should be given to the first MEV whose need is much greater. More information and examples can be found in [10].

C. Mathematical model

The work presented here is for proof of principle only, to explore the qualitative behavior of the DVP approach. Therefore, no attempt was made to implement a high-fidelity vehicle model. Instead, a kinematic model was assumed. Each vehicle was represented by a center point position with orientation, velocity, and yaw rate, while collision overlaps did consider the length and width of the vehicle. The kinematic model of the vehicle is shown in Fig. 1. The control inputs to the kinematic model are the longitudinal acceleration a and steering angle δ. The total acceleration a is obtained by summing the forward acceleration a_x and the negative acceleration (i.e brake) a_{-x}; both are control inputs and supposed positive, therefore the total longitudinal acceleration is $a = a_x - a_{-x}$. From Fig. 1 the longitudinal and lateral velocities of the Center of Gravity (CoG) are given by:

$$\begin{aligned} \dot{x} &= v\cos(\psi+\beta) \\ \dot{y} &= v\sin(\psi+\beta), \end{aligned} \tag{1}$$

while the forward velocity of the vehicle is v. The rotation rate of the vehicle Ω is given by:

$$\Omega = \dot{\psi} = \frac{v}{R} = \frac{v}{l_r}\sin\beta, \tag{2}$$

where R is the radius of the instantaneous center of rotation and l_r the distance between the rear axle and the CoG. The relation between steering angle δ and vehicle sideslip angle

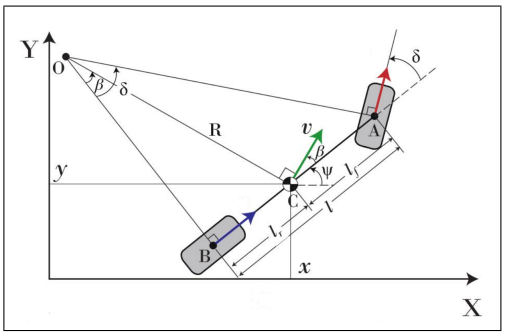

Fig. 1. Kinematic model of a vehicle rotating about the instantaneous center of rotation O.

β can be obtained with simple trigonometric relations:

$$\beta = \tan^{-1}\left(\frac{l_r}{l}\right)\tan\delta, \tag{3}$$

with l being the wheelbase. Finally, the kinematic model is:

$$\begin{aligned}\dot{x} &= v\cos(\psi+\beta)\\ \dot{y} &= v\sin(\psi+\beta)\\ \dot{\psi} &= \tfrac{v}{l_r}\sin\beta\\ \dot{v} &= a = a_x - a_{-x}.\end{aligned} \tag{4}$$

D. Controller

The controller architecture for each vehicle is formed of a high-level MPC that implicitly includes path planning and path following, and a low-level controller (see Fig. 2). However, the implementation of the latter is vehicle dependent, and therefore out of the scope of this paper. Each vehicle solves its own optimization problem, giving as output the trajectories that would be broadcast and the actuation commands to obtain the planned trajectory. Such commands are actuated by the low-level controller.

The general form of the MPC for achieving a cooperative obstacle avoidance is:

$$\min_{\mathcal{X}} \quad \mathcal{J}_{\mathcal{X}} \tag{5}$$

$$\text{s.t.} \quad \mathcal{B}, \tag{6}$$

where $\mathcal{J}_{\mathcal{X}}$ is the cost to minimize, $\mathcal{X}$ is the vector of optimization variables, and $\mathcal{B}$ is the set of constraints. This flexible formulation allows the inclusion of different objectives, e.g. being between the edges of the road, lane-centering feature, including other vehicles trajectories, limits on actuators, etc. The optimization variables are the states along the prediction horizon, and the control variables, i.e. the steered wheel angle δ, the forward acceleration a_x and the brake a_{-x}. The length of the MPC horizon is N representing the number of points, while the timestep is ΔT. Given the structure, it is convenient to now derive the discrete version of the kinematic model described in (4):

$$\begin{aligned}x[k+1] &= x[k] + v[k]\cos(\psi[k]+\beta[k])\Delta T\\ y[k+1] &= y[k] + v[k]\sin(\psi[k]+\beta[k])\Delta T\\ \psi[k+1] &= \psi[k] + v[k]\tfrac{\sin\beta[k]}{l_r}\Delta T\\ v[k+1] &= v[k] + (a_x[k] - a_{-x}[k])\Delta T,\end{aligned} \tag{7}$$

and $\beta[k] = \tan^{-1}(l_r/l)\tan\delta[k]$.

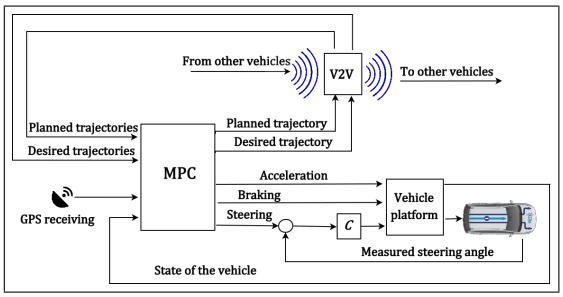

Fig. 2. Controller architecture for cooperative obstacle avoidance. The steering command is the reference for a low-level feedback controller C that delivers the actuation. Acceleration and braking commands are implemented by the vehicle platform. The inputs for the MPC are the state of the vehicle obtained via the vehicle platform and the GNSS antenna, and the trajectories from other vehicles.

III. OPTIMIZATION PROBLEM

The main goal of this paper is to demonstrate the effectiveness of the DVP concept in cooperation between MEVs to avoid static obstacles on the motorway. In order to be successful, all the MEVs have to broadcast their trajectories. The entire optimization of the trajectory was done twice at each update, for each vehicle: once to determine the planned trajectory, and once for the desired trajectory. The latter omitted the calculation of the trajectory overlap and collision cost for the planned trajectories of cooperating vehicles. When computing the planned trajectory, the cost function is designed in order to strongly prefer to avoid trajectory overlap with the planned trajectories of other MEVs, and weakly preferred to avoid the desired trajectories. This led to cooperative behavior, and demonstrating this was the crux of this work. Thus, there are two very similar optimization problems with objective functions $\mathcal{J}_{pl}$ and $\mathcal{J}_{des}$ for finding planned and desired trajectories respectively.

A. Cost function design

The cost function describes the overall objective we aim to achieve. The cost function used was a weighted sum of the components highlighted in the following subsections.

1) Vehicle avoidance and trajectory overlapping: The idea is having an "activation function" that enables such part of the cost when there is a chance or an actuality of a collision. A collision is possible when two vehicles are relatively close longitudinally or laterally. When solving the optimization problem, each vehicle has its own local frame, called *ego* frame. Consider an orthogonal frame and two MEVs, as shown in Fig. 3. The origin of the frame is set at the CoG of MEV1, which is the *ego* vehicle solving the optimization problem. The x-axis is orientated as MEV1 and the y-axis is orthogonal to MEV1's direction. It is worth emphasizing that MEV2 has its own ego frame and solves its own optimization problem, in a completely symmetric way. In a more general case of V vehicles cooperating, each of these V vehicles will use its own local frame to solve the optimization problem.

Let Δx be the Euclidean distance between the x-

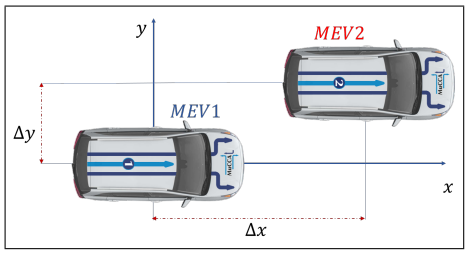

Fig. 3. Schematic representation of a local frame of one of the vehicle.

coordinates of the CoGs of both MEVs:

$$\Delta x = x_{MEV2} - x_{MEV1},$$

and let C_c be a general cost of collision between two vehicles. Such cost would be:

$$C_c = \begin{cases} w_c f_{xy} & \text{if } |\Delta x| < d_x \text{ and } |\Delta y| < d_y \\ 0 & \text{otherwise} \end{cases}, \quad (8)$$

where w_c is the weight, f_{xy} is the collision avoidance function depending on Δx and Δy, and d_x, d_y are safe distances (longitudinal and lateral respectively) that account for the size of the vehicles. From the equation shown above, it is easy to infer that the cost is activated when the distance between the two vehicles is smaller than a certain threshold considering both lateral and longitudinal distance. A function defining the "activation" mechanism is here designed to be a smooth windows function that avoids discrete cases and thus prevents the need to implement binary constraints. Such a function has to be smooth to avoid discontinuity in the cost function. We define it by multiplying two logistic functions:

$$\begin{aligned} \sigma_1(\Delta x) &= \frac{1}{1 + e^{-a_{\sigma x}(d_x - \Delta x)}} \\ \sigma_2(\Delta x) &= \frac{1}{1 + e^{-a_{\sigma x}(d_x - (-\Delta x))}}, \end{aligned} \quad (9)$$

where $a_{\sigma x}$ is a parameter defining the steepness of the function, and d_x represents the longitudinal safe distance. The two functions are symmetric about $\Delta x = 0$, i.e. $\sigma_2(\Delta x) = \sigma_1(-\Delta x)$. The overall activation function is shown in Fig. 4 and defined as:

$$\Sigma(\Delta x) = \sigma_1(\Delta x) \cdot \sigma_2(\Delta x), \quad (10)$$

The same concept is applied for the inter-vehicles lateral distance:

$$\Delta y = y_{MEV2} - y_{MEV1}, \quad (11)$$

$$\sigma_1(\Delta y) = \frac{1}{1 + e^{(-a_{\sigma y}(d_y - \Delta y))}}, \quad (12)$$

$$\sigma_2(\Delta y) = \frac{1}{1 + e^{(-a_{\sigma y}(d_y - (-\Delta y)))}}, \quad (13)$$

$$\Sigma(\Delta y) = \sigma_1(\Delta y) \cdot \sigma_2(\Delta y). \quad (14)$$

Starting from (14):

$$\Sigma(\Delta y) = \frac{1}{1 + e^{(-a_{\sigma y}(d_y - \Delta y))}} \cdot \frac{1}{1 + e^{(-a_{\sigma y}(d_y + \Delta y))}},$$

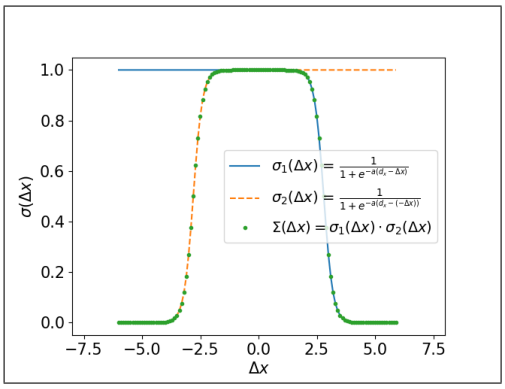

Fig. 4. Activation function for longitudinal collision avoidance.

let us write the function in a simpler way. By multiplying the denominators, we have

$$\Sigma(\Delta y) = \frac{1}{1 + e^{(-a_{\sigma y}(d_y - \Delta y))} + e^{(-a_{\sigma y}(d_y + \Delta y))} + e^{-2a_{\sigma y} d_y}}. \quad (15)$$

Recalling the definition of hyperbolic cosine:

$$\cosh(x) = \frac{e^x + e^{-x}}{2},$$

we can write the term in (15) as:

$$e^{(-a_{\sigma y}(d_y - \Delta y))} + e^{(-a_{\sigma y}(d_y + \Delta y))} = e^{-a_{\sigma y} d_y}(e^{a_{\sigma y}\Delta y} + e^{-a_{\sigma y}\Delta y}),$$

and therefore:

$$e^{(-a_{\sigma y}(d_y - \Delta y))} + e^{(-a_{\sigma y}(d_y + \Delta y))} = e^{-a_{\sigma y} d_y} 2\cosh(a_{\sigma y}\Delta y). \quad (16)$$

Substituting (16) in (15), we have

$$\Sigma(\Delta y) = \frac{1}{1 + e^{-a_{\sigma y} d_y} 2\cosh(a_{\sigma y}\Delta y) + e^{-2a_{\sigma y} d_y}} \quad (17)$$

and finally, by some rearranging of the equation, we have:

$$\Sigma(\Delta y) = \frac{e^{2a_{\sigma y} d_y}}{e^{2a_{\sigma y} d_y} + 2e^{2a_{\sigma y} d_y}\cosh(a_{\sigma y}\Delta y) + 1}. \quad (18)$$

In this way, the non-linearity of the function is only in the term $\cosh(a_{\sigma y}\Delta y)$. That will help the numerical solver. We can do the same for the x-axis cost function and obtain:

$$\Sigma(\Delta x) = \frac{e^{2a_{\sigma x} d_x}}{e^{2a_{\sigma x} d_x} + 2e^{2a_{\sigma x} d_x}\cosh(a_{\sigma x}\Delta x) + 1}. \quad (19)$$

Some considerations to make: d_x and d_y are different; $\Sigma(\Delta y)$ and $\Sigma(\Delta x)$ are independent of each other, but the collision cost shall be different from zero if both are activated. Therefore, the collision cost can be defined as

$$C_c = w_c \Sigma(\Delta x, \Delta y), \quad \Sigma(\Delta x, \Delta y) = \Sigma(\Delta x)\Sigma(\Delta y). \quad (20)$$

This function is shown in Fig. 5. The equation highlighted above holds for a control horizon of length equal to one. For our scenario, a model predictive controller with horizon equal to N, the cost of collision is computed along the trajectories

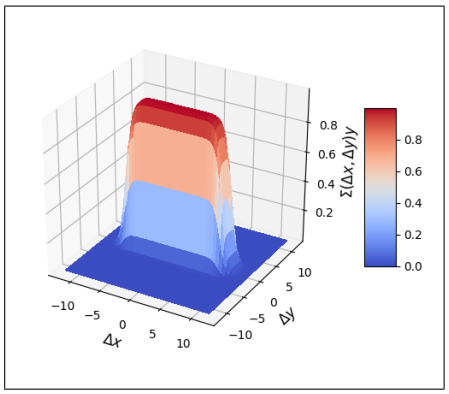

Fig. 5. Activation function for collision avoidance.

of the ego vehicle, of the M obstacles and *all* the other V MEVs broadcasting their messages. Therefore, we define the term $\Delta x_{ij} = x_{ego_i} - x_{ij}$ as the difference of longitudinal positions between ego vehicle and MEV $j \in [1, V]$ in x_{ego} frame at time $i \in [1, N]$. We reinforce the concept that the origin of the ego frame is the position of the ego vehicle at time $i = 0$. Recalling the distinction made about planned and desired trajectory, we defined the collision cost when computing the planned trajectory:

$$C_{cpl} = \sum_{j=1}^{M}\sum_{i=1}^{N} w_{obs}\Sigma(\Delta x_{ij}, \Delta y_{ij}) + \sum_{j=1}^{V}\left(\sum_{i=1}^{N} w_{pl}\Sigma(\Delta x_{ij}, \Delta y_{ij}) + \eta_j w_{des}\Sigma(\Delta x_{ij}, \Delta y_{ij})\right), \tag{21}$$

where the subscript i refers to the x, y position at prediction i; w_{obs}, w_{pl}, and w_{des} are the weights for respectively obstacle, planned, and desired trajectory. The trajectories of the obstacles are constant, i.e. $x_{0j} = x_{ij}, \forall i \in [1, N] \wedge \forall j \in [1, M]$. The value η_j represents the *importance* of accommodating the desired trajectory of vehicle j. It is updated by each vehicle at the end of each iteration of the optimization problems and broadcast with the trajectories and computed as follows:

$$\eta = \begin{cases} \log(\mathcal{J}_{pl} - \mathcal{J}_{des}) & \text{if } \mathcal{J}_{pl} - \mathcal{J}_{des} > 1 \\ 0 & \text{otherwise} \end{cases}.$$

The cost of collision when computing desired trajectory is:

$$C_{cdes} = \sum_{j=1}^{M}\sum_{i=1}^{N} w_{obs}\Sigma(\Delta x_i, \Delta y_i) + \sum_{j=1}^{V}\sum_{i=1}^{N} \eta_j w_{des}\Sigma(\Delta x_i, \Delta y_i). \tag{22}$$

2) Geofencing: One of the assumptions was that in the scenarios considered the road is formed of three lanes. An objective of the vehicle is also remaining on the road, and enabling for experimental purposes a geofencing feature that gives back control to the driver when the vehicle goes off-road. Therefore, being really close to the edge of the road has a high cost, that quickly drops to zero when we are inside the lanes. We follow the same approach as above. Consider for instance the left edge of the road; consider a frame local to such edge, where the x-axis is along the edge and the y-axis is perpendicular to the edge, with y value increasing to the right. A schematic representation is shown in the Fig. 6: the brown frame is the left edge local frame.

The cost term computed in the edge frame of reference will have the following form:

$$C_{edge} = \sum_{i=1}^{N} w_{edge}\sigma(y_{ei}), \tag{23}$$

with

$$\sigma(y_{ei}) = \frac{1}{1 + e^{-a_{edge}(\epsilon - y_{ei})}},$$

where the y_e is the y-coordinate of the CoG of the vehicle computed in the edge frame via simple trigonometric relations, a_{edge} is a parameter defining the steepness of the cost function, w_{edge} is the weight, and ϵ is a parameter that defines the safe distance from the edge, used to account for the width of the car. The same concept is applied symmetrically for the right edge.

3) Lane centering: MuCCA is an emergency system. When MuCCA is engaged, but there are no obstacles to avoid, the vehicles are encouraged to stay in the center of the lane. It is worth emphasizing that there is no target lane. The system will encourage the vehicle to self-center in whatever lane it finds itself. To achieve that, we define an orthogonal frame, with the x-axis set on the left line of the current lane, and the y-axis passing through the CoG of the car. We consider a parabola with the vertex on the y-axis, of parameter a_{lk} and having as variable the y-coordinate of the car CoG in this frame of reference, that is y_l. The cost term for lane keeping is:

$$C_{lk} = \sum_{i=1}^{N} w_{lk} a_{lk}\left(y_{li} - \frac{l_w}{2}\right)^2, \tag{24}$$

with l_w being the width of the lane. The steepness parameter of the parabola has been designed to be $a_{lk} = 0.1$. A higher value of a_{lk} means a steeper curve, that will lead the solver to further penalize being far from the center.

In addition, the vehicle is encouraged to have the same heading as the pre-mapped lanes,

$$C_h = \sum_{i=1}^{N} w_h(h_l - \psi_{li})^2, \tag{25}$$

where w_h is the weight for this objective, h_l is the pre-known heading of the mapped lanes in ego-frame, and ψ_{li} is the heading of the vehicle in ego-frame. Finally, the MEV is encouraged to keep the velocity $\bar{v}$ it had when MuCCA was engaged:

$$C_h = \sum_{i=1}^{N} w_v(\bar{v} - v_i)^2, \tag{26}$$

where v_i is the velocity at timestep i.

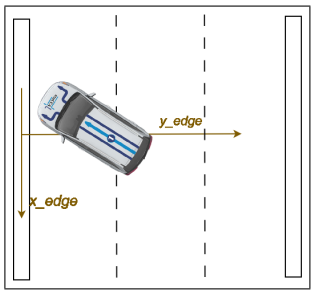

Fig. 6. Schematic representation of the edge-frame, used to define the cost for the geofencing.

4) Control variables smoothness: To avoid abrupt changes of the control variables that could lead to unnecessarily aggressive maneuvers, two cost terms were included for each of the three control variables. The first one limits the actuations as shown:

$$C_{act} = \sum_{i=1}^{N} w_\delta \delta_i^2 + w_{acc} a_{y_i}^2 + w_{br} a_{-y_i}^2. \tag{27}$$

The gap between sequential actuations is minimized as well:

$$C_{diff} = \sum_{i=2}^{N-1} w_{\Delta\delta}(\delta_i - \delta_{i-1})^2 + w_{\Delta acc}(a_{x_i} - a_{x_{i-1}})^2 + w_{\Delta br}(a_{-x_i} - a_{-x_{i-1}})^2. \tag{28}$$

5) Dynamic compensation: As stated in Section II, the vehicle model is kinematic. Therefore, there is no consideration of dynamic components. However, we added another term in the cost function to empirically avoid dynamic instability by limiting steering actuation progressively with vehicle speed:

$$C_{dyn} = \sum_{i=1}^{N} w_{v,\delta} v_i^2 \delta_i^2 + \sum_{i=1}^{N-1} w_{v,\dot{\delta}} v_i^2 \dot{\delta}_i^2. \tag{29}$$

Finally, the total cost to minimize is obtained by adding all the terms from (21) (or (22)) to (29):

$$\mathcal{J} = C_c + C_{edge} + C_{lk} + C_h + C_{act} + C_{diff} + C_{dyn}. \tag{30}$$

Based on which trajectory will be computed, the first term of the cost function, i.e. the collision cost C_c, will be C_{cpl} or C_{cdes}, as in (21) and (22) respectively. The values of all the weights highlighted in this section and used in the experiments of Section IV are in Table I. From Table I, one can notice that the weights were designed to discourage braking actions, in order to let cooperative steering behavior emerge by preference. In Table II the reader can find the parameters of the activation function shown in the previous subsection.

B. Constraints

Having defined the cost to minimize, we here define the set of constraints. First, the actuators commands are limited. At every iteration, there is a routine that computes the upper bounds of the control commands. Such bounds are derived by empirical experiments on the vehicle and safety analysis in order to guarantee the controllability of the vehicle and to allocate one second for the safety drivers to intervene in case of danger. Forward acceleration and braking commands are defined positive, therefore their lower bound is always zero. We want to constrain the difference between our model predicted states using (7) and the *real* state values to be equal to zero. In this way, the optimizer is forced to use our prediction of future state values.

Defining $\mathcal{X} = \begin{bmatrix} x & y & \psi & v \end{bmatrix}$ as the state to optimize, then $\bar{\mathcal{X}} = \begin{bmatrix} \bar{x} & \bar{y} & \bar{\psi} & \bar{v} \end{bmatrix}$ the vector of the state evolution computed using the model defined in (7), and finally let $\mathcal{A} = \begin{bmatrix} \delta & a_x & a_{-x} \end{bmatrix}$ be the vector of actuator commands, the optimization problem is:

$$\min_{\mathcal{X}_i, \mathcal{A}_i \forall i \in [1,N]} \quad \mathcal{J} \tag{31}$$

$$\text{s.t.} \quad -\bar{\delta_k} \leq \delta_i \leq \bar{\delta_k}, \quad \forall i \in [1, N] \tag{32}$$

$$0 \leq a_{x_i} \leq \bar{a}_{xk}, \quad \forall i \in [1, N] \tag{33}$$

$$0 \leq a_{-x_i} \leq \bar{a}_{-xk}, \quad \forall i \in [1, N] \tag{34}$$

$$x_i - \bar{x}_i = 0 \quad \forall i \in [1, N] \tag{35}$$

$$y_i - \bar{y}_i = 0 \quad \forall i \in [1, N] \tag{36}$$

$$\psi_i - \bar{\psi}_i = 0 \quad \forall i \in [1, N] \tag{37}$$

$$v_i - \bar{v}_i = 0 \quad \forall i \in [1, N] \tag{38}$$

where (31) is the cost to minimize as defined in (30); constraints (32)-(34) are the limits of the actuators, with $\bar{\delta_k}, \bar{a}_{xk}, \bar{a}_{-xk}$ bounds computed at time k as explained before; constraints (35)-(38) are the minimization between the evolution of the states and the predicted states. As a clarification, just consider (35), with respect to (7). The solver will force the following differences to be zero:

$$\begin{aligned} x_1 - \bar{x}[k+1] &= 0 \\ x_2 - \bar{x}[k+2] &= 0 \\ &\vdots \\ x_i - \bar{x}[k+i] &= 0 \\ &\vdots \\ x_N - \bar{x}[k+N] &= 0 \end{aligned}$$

given the model (7)

$$\begin{aligned} \bar{x}[k+1] &= \bar{x}[k] + \bar{v}[k]\cos(\bar{\psi}[k] + \bar{\beta}[k])\Delta T \\ \bar{x}[k+2] &= \bar{x}[k+1] + \bar{v}[k+1]\cos(\bar{\psi}[k+1] \\ &\quad + \bar{\beta}[k+1])\Delta T \\ &\vdots \\ \bar{x}[k+1+i] &= \bar{x}[k+i] + \bar{v}[k+i]\cos(\bar{\psi}[k+i] \\ &\quad + \bar{\beta}[k+i])\Delta T \\ &\vdots \\ \bar{x}[k+1+N] &= \bar{x}[k+N] + \bar{v}[k+N]\cos(\bar{\psi}[k+N] \\ &\quad + \bar{\beta}[k+N])\Delta T, \end{aligned}$$

and so on for the other states. The initial state to optimize is at index $i = 1$ since $i = 0$ is the initial state of the vehicle at time k. Bear in mind that the control inputs that will be actuated by the vehicle are the set computed during the solving of the planned trajectory. It is worth mentioning that

TABLE I
WEIGHTS

w_{obs}	w_{pl}	w_{des}	w_{edge}	w_{lk}	w_h	w_v	w_δ	w_{acc}	w_{br}	$w_{\Delta\delta}$	$w_{\Delta acc}$	$w_{\Delta br}$	$w_{v,\delta}$	$w_{v,\dot{\delta}}$
12.0	6.0	5.0	20.0	0.15	1.0	2.0	1.0	10.0	50.0	6.0	50.0	50.0	6.0	5.0

TABLE II
ACTIVATION FUNCTION PARAMETERS

$a_{\sigma x}$	$a_{\sigma y}$	a_{edge}	d_x	d_y	ϵ
2.0	5.0	5.0	2.9	6.0	0.0

TABLE III
MODEL PREDICTIVE CONTROL PARAMETERS

Horizon	Time interval	Time limit for the solver
N = 6	$\Delta T = 0.800$ [s]	0.250 [s]

the points in the broadcast trajectories are absolute GNSS coordinates, with an associated timestamp.

IV. REAL WORLD IMPLEMENTATION AND RESULTS

A. Hardware and software architecture

1) Autonomous Vehicle Platform: For these experiments, an off-the-shelf robot car platform was used: two Kia Niro hybrid vehicles were fitted with PolySync DriveKit units [11] to render them computer-controllable via CAN bus commands. Brake and accelerator commands were issued as relative proportions, and closed-loop steering wheel control obtained by commanding torque and monitoring angular position. The vehicles were 4.36 meters long, 1.8 meters high and 1.8 meters wide. The wheelbase l was 2.7 meters, and l_r, that is the distance from the rear axle to the CoG, was 1.67 meters.

2) Localization: As the focus of this work was the cooperative control problem, localization was handled simply. Each vehicle used Real-Time Kinematic (RTK) corrected GNSS positioning to obtain absolute antenna location, and (from movement) the velocity vector. The antenna was located above the CoG. A software Local Dynamic Map (LDM) was pre-populated with the coordinates of the road boundaries and lane line positions on the test track.

3) Vehicle to Vehicle communication: V2V communications were implemented using customized versions of draft ETSI-G5 Maneuver Coordination Message (MCM) messages, with 802.11p as the underlying protocol. Each vehicle shares its location, its state, and its intended trajectories plans. The messages were broadcast by each vehicle every 20 ms.

4) On-board unit: The host computer was a proprietary IDIADA ADAS Platform Tool (IDAPT) unit [12], containing an NVIDIA Jetson TX2 main microcontroller, differential GNSS receiver, V2V transceiver and other I/O and safety components. The software was written in C++.

5) Optimization solver: The MPC design parameters are shown in Table III. As shown in Section III, the multi-objective function is non-linear, in order to avoid using binary constraints. Therefore, the chosen solver was Ipopt [13], a non-linear, free optimization solver. It is worth to further emphasize that the approach shown in this paper should never lead to a 'freezing robot' state: by avoiding the use of hard constraints to forbid collisions, we should always get a minimum cost solution with related commands for the actuators, even if that solution involves collision. As an additional safeguard, we added via software an extra safety measure that gives back control to the driver if some consecutive solutions are sub-optimal or not found.

B. Experiment scenario and performance evaluation

The experiments were performed in the 3-lane motorway-like section of Bruntingthorpe Proving Ground, UK (see Fig. 7). The width of the lane l_w is automatically computed at each iteration using the knowledge of the pre-mapped lanes. This is done for compatibility with future work when the lanes could be detected. The scenario tested included two MEVs, i.e. $V = 2$, one in the leftmost lane, one in the central lane, and two obstacles. The obstacles had the same dimensions as each MEV. The first obstacle was in the leftmost lane at 100 meters from the activation point of the MuCCA system; the second obstacle was in the central lane, fifty meters after the first obstacle. Both obstacles locations are known in advance and fitted into the LDM.

Both MEVs were driven by fully-trained safety drivers, whose task was to activate MuCCA and to intervene in case of any danger of collision. The MuCCA system was activated at 100 meters from the first obstacle, at a cruise speed of 30 kilometers per hour. This was a measure to ensure the safety of the drivers. After this point, both cars ran in fully automated mode, implementing cooperative avoidance as a pair. From Fig. 8 and Fig. 9, it is possible to appreciate the quality of the DVP in safely avoiding both obstacles. As it can be observed in Fig. 8, the vehicle in the center lane steers to the right, leaving to the vehicle in the leftmost lane space to safely steer to the right as well. After that, the leftmost MEV steers left of the obstacle to avoid collision with the rightmost MEV, while the rightmost one is free to continue in its lane. The experiment is documented on video[1].

The contribution of the DVP can be observed in another repetition. Consider the experiment in Fig. 9. The reader can appreciate how the desired trajectory (black scattered points) of the leftmost MEV at around 14 seconds is different from the planned trajectory (solid blue line): planned trajectory

[1]Video documentation: https://youtu.be/fjBzOkyDt5M

Fig. 7. The two automated vehicles at Bruntingthorpe Proving Ground.

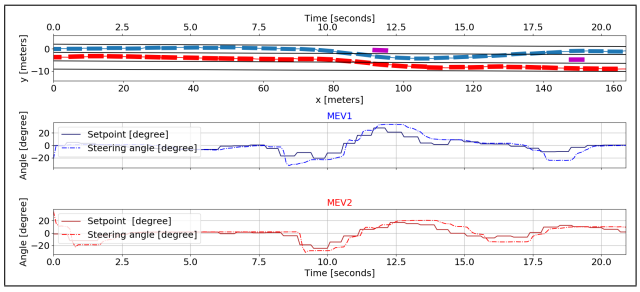

Fig. 8. Double obstacle avoidance of two cooperative autonomous vehicles implementing the DVP concept. The purple boxes represent the obstacles.

was to brake but collide with the obstacle, whereas the desired was to steer around it. The red vehicle accommodated the desired trajectory and gave space by steering to the right.

V. CONCLUSIONS

The DVP approach, using a simple MPC with no hard constraints, was able to produce cooperative behavior among two real vehicles. This simple yet powerful approach promises robustness to unknown or unexpected situations as each vehicle always plans a trajectory based on current knowledge, with no need to compare alternative joint plans or invoke a leader or outside authority. Further work would include implementing a Kalman filter along with a perception system to better estimate the state of the ego and the other vehicles. The control input can be decimated to reduce the space of solutions, and thus reducing the time-limit of the optimization solver, leading to faster exchange of messages between vehicles and therefore better handling of faster speeds. Moreover, a robustness analysis of the MPC is advised for investigating the rate of optimal and sub-optimal solutions for different scenarios. The weights, the horizon, and the timestep duration have been tuned to work in the demonstrated scenario, but finer tuning is required to have more robustness at different speeds and in a wider range of scenarios. Finally, a bicycle model is expected to improve the performance and to bring this solution to a real project fleet at a higher speed in different scenarios.

ACKNOWLEDGMENT

IDIADA would like to thank the MuCCA consortium partners (Cranfield University, Connected Places Catapult, Cosworth Electronics, Westfield Sports Cars and SBD Automotive) for their cooperation.

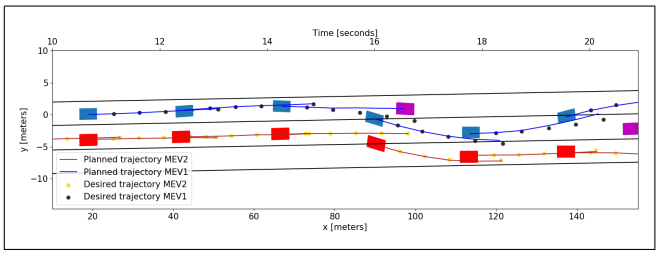

Fig. 9. Instances of the planned and desired trajectories of both vehicles.

REFERENCES

[1] D. J. Fagnant and K. Kockelman, "Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations," *Transportation Research Part A: Policy and Practice*, vol. 77, pp. 167–181, 2015.

[2] J. Tomas-Gabarron, E. Egea-Lopez, and J. Garcia-Haro, "Vehicular trajectory optimization for cooperative collision avoidance at high speeds," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1930–1941, Dec 2013.

[3] The MuCCA project website. [Online]. Available: https://mucca-project.co.uk/

[4] 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.

[5] D. Heß, R. Lattarulo, J. Pérez, T. Hesse, and F. Köster, "Negotiation of cooperative maneuvers for automated vehicles: Experimental results," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Oct 2019, pp. 1545–1551.

[6] C. Frese and J. Beyerer, "A comparison of motion planning algorithms for cooperative collision avoidance of multiple cognitive automobiles," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, June 2011, pp. 1156–1162.

[7] I. B. Viana and N. Aouf, "Distributed cooperative path-planning for autonomous vehicles integrating human driver trajectories," in *2018 International Conference on Intelligent Systems (IS)*, Sep. 2018, pp. 655–661.

[8] C. Burger and M. Lauer, "Cooperative multiple vehicle trajectory planning using miqp," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2018, pp. 602–607.

[9] B. Lehmann, H. Günther, and L. Wolf, "A generic approach towards maneuver coordination for automated vehicles," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2018, pp. 3333–3339.

[10] C. Wartnaby and D. Bellan, "Decentralised cooperative collision avoidance with reference-free model predictive control and desired versus planned trajectories," *CoRR*, vol. abs/1904.07053, 2019. [Online]. Available: http://arxiv.org/abs/1904.07053

[11] PolySync DriveKit website. [Online]. Available: https://polysync-xrcc.squarespace.com/drivekit

[12] IDAPT multi-purpose on-board unit. [Online]. Available: https://blog.applus.com/applus-idiada-enabling-connected-and-automated-vehicle-innovations/

[13] Ipopt website. [Online]. Available: https://coin-or.github.io/Ipopt/