Distributed Lane Selection in Autonomous Platoon Coordination

Marc Facerías[1], Vicenç Puig[2] and Alexandru Stancu[1]

***Abstract*— This paper proposes a novel non-linear lane selection algorithm for autonomous vehicles based on the Optimality Condition Decomposition algorithm. An initial algorithm based on mixed integer quadratic programming was studied, which proved untreatable due to its computational complexity, leading to similar formulations that do not require such a nature in their variables. As a result, a cascade strategy combining both distributed local planning and decision-making layers is implemented, leading to promising results. The proposed approach can determine paths that consider individual goals and the environment so that agents can collaborate to establish a formation that re-configures online, allowing vehicles to reach their individual goals.**

I. Introduction

Autonomous driving is becoming a hot topic as many of its sub-problems such as intersection management, assistive driving, and platooning are being studied. In this paper, we aim to study the role of lanes when driving platoons of coordinated agents in a shared environment, where each agent has both the common interest of traversing the highway as fast as possible and an individual goal. While being a vital part of trajectory planning, lane change is usually treated separately as a decision-making sub-problem. In this context, it is usually modelled as whether the agent should move to a neighbouring lane or stay in the current one. Intuitively, this can be represented as a state machine, as seen in [1], where the lane change behaviour is decided by exploring a topological tree with a cost associated with each manoeuvre, to later be interfaced with a local planning based on polynomial candidate generation that through an optimisation problem chooses a suitable candidate for the manoeuvre. Similarly, [2] proposes a safe trajectory generation to overtake neighbouring vehicles as an assistive module, thus without decision-making involved. This line of work proved its applicability in real scenarios, as it can be seen in [3], where the lane change behaviour is solved via an MPC that, aided by a neighbour behavioural estimation module, manages to assist the driver during lane changes in a realistic scenario generated using a data-set containing real car data from a highway. So far, the work presented tackles the problem from the ego vehicle point of view, considering the rest of the agents on the road as moving obstacles to be avoided. However, there is an open line of research seeking to perform lane changes collaboratively. For instance, in [4] a two-level decision-making strategy is performed where individual vehicles decide whether or not a lane change should be performed based on an optimisation problem and then a cooperative algorithm executes the subset of feasible lane changes that is more beneficial for the fleet, running out any potential collision manoeuvres. Similarly, [5] proposes a three-stage lane change strategy, where firstly lane changes are evaluated, secondly, several regions are explored to perform the change and finally, the strategy is refined by imposing traffic safety constraints. Applying this algorithm leads to improvements in both traffic flow and the safety metrics proposed by the authors when comparing it with other state-of-the-art strategies. In another line of work, [6] proposes an MPC to refine the trajectories followed by a set of agents while performing a lane change, improving the expected results from a human-like baseline manoeuvre. Following a centralised optimisation fashion, we can see how [7] proposes two different sets of constraints depending on the role of the agent, whether it is an individual agent or is part of a platoon formation, to be later embedded in a common large-scale non-linear optimisation problem that provides control variables for each of the agents involved. Finally, other approaches rely not on solving a single-shot optimisation problem but on learning certain policies to be applied, being of particular interest [8], where an actor-critic strategy is used to learn a lane change policy for an ego vehicle. Alternatively, it is quite popular in the field to treat lane change from the game theory point of view, as clearly it can be modelled as a set of players choosing whether or not to change their lane depending on an expected reward. In [9], game theory is embedded into an optimisation problem so that agents decide their future manoeuvres based on the expected behaviour of their neighbours, which is later extensively tested in both cooperative and competitive test scenarios with promising results. In this paper, we aim to tighten the ties of both agent coordination and decision-making by exploring strategies in a distributed manner. In particular, a two-stage distributed strategy will be derived by exploiting both the slower dynamics of the lane change and the simplicity of considering vehicles to be traversing a lane-free environment. In this context, we aim to contribute to the field by proposing a MIQP lane change strategy, which will be later rewritten in a non-linear fashion to tackle the inherently prohibitive computational burden of integer-based optimisation problems.

II. Problem statement

Navigation in lane-constrained environments is a dual problem involving both spatial planning and decision-making, as vehicles need to select a subspace of the region

[1] M. Facerías and A. Stancu belong to the Faculty of Electrical Engineering, University of Manchester, The United Kingdom marc.faceriaspelegri@manchester.ac.uk

[2] V. Puig belongs to the Institute of Robotics, CSIC-UPC, Barcelona Spain vicenc.puig@upc.edu

they are navigating, e.g. a lane, while generating suitable trajectories to reallocate themselves and operate safely. In this context, lane-based autonomous navigation can be modelled as teams of N robots, each having their individual reference r_n, traversing a common environment W divided into L subregions $W = [l_0...l_N]$. Both references and world are modelled after Frenet curves, defined as a collection of segments of constant curvature. Note that the experimental environment is reduced to a local expression W_k, containing the set of reachable lanes within one planning step. In general, this scenario can be solved by a non-linear optimisation problem of the following shape

$$\begin{aligned}
\min_{\Delta u_n^k} \quad & \sum_{k=0}^{H}\sum_{n=0}^{N}(J_n^k(x_n^k, r_n^k) + J_g(x_n^k)) \\
\text{s.t.} \quad & \forall k \in [0,H], \forall n \in [1,N] \\
& x_n^{k+1} = f(x_n^k, u_n^k) \\
& f_l = (x_n^{k+1} \in l_i) \wedge (x_n^{k+1} \notin l_v) \quad \forall v \in L \setminus i \quad (1) \\
& u_n^k = u_n^{k-1} + \Delta u_n^k \\
& u_n^k \in [\underline{u_n^k}, \overline{u_n^k}] \\
\text{s.t.} \quad & \forall k \in [0,H], \forall n \in [1,N], \forall j \in [1,N]\ j \neq n \\
& f_c(x_n^k, x_j^k) < 0
\end{aligned}$$

where both $J_n^k(x_n^k, r_n^k)$ and $J_g(x_n^k)$ are arbitrary cost functions, $f(x_n^k, u_n^k)$ is the model of each agent, $f_c(x_n^k, x_j^k)$ models inter-vehicular interactions, f_l models lane-belonging constraints, u_n^k are the control actions applied to the vehicle n at time k and x_n^k is the state of vehicle n at time k.

III. Proposed solutions

A. Binary Lane Scheduler

Inspired by [10], a relaxation can be proposed over the nonlinear problem presented in Eq. (1) by using binary variables for both the collision constraints and the lane selection. Further relaxations can be imposed by limiting the shapes of both constraints and cost functions. For instance, if the term $f(x_n^k, u_n^k)$ is linearised then the problem becomes either a Mixed Integer Linear Programming (MILP) or a Mixed Integer Quadratic Programming (MIQP), depending on the shape of $J_n^k(x_n^k, r_n^k)$ and $J_g(x_n^k)$. Overall, such design considerations allow solving Eq. (1) in treatable times. This motivated the development of a case study in which we aim to propose an MIQP collaborative planner applicable to a highway-like environment. In particular, $f_c(x_n^k, x_j^k)$ and f_l are expressed in Eqs. (3) and (4), respectively. Note that the collision check is done in a Manhattan fashion, thus allowing using an ∞-norm. Agents are modelled through a set of linear discrete equations of the shape $x_n^{k+1} = A_n^k x_n^k + B_n^k u_n^k$ and referenced to a common global lane. Collision avoidance between an agent i and j can be expressed as

$$\|x_j - x_i\| \geq d_{saf} \quad OR \quad \|y_j - y_i\| \geq d_{saf} \qquad (2)$$

where x_j and y_j refer to the x, y spatial coordinates of a given vehicle j.

However, to embed this expression into an optimisation problem it needs to be rewritten as the following set of constraints using the well-known big M method, being $c_{i,j}^l$ a binary variable, M a sufficiently large constant and $l = 1, 2$.

$$\begin{aligned}
C_{cl}: \\
& (x_j - x_i) \geq d_{saf} - M(c_{i,j}^1 + c_{i,j}^2) \\
& (x_i - x_j) \geq d_{saf} - M(1 - c_{i,j}^1 + c_{i,j}^2) \\
& (y_j - y_i) \geq d_{saf} - M(1 + c_{i,j}^1 - c_{i,j}^2) \\
& (y_i - y_j) \geq d_{saf} - M(2 - c_{i,j}^1 - c_{i,j}^2)
\end{aligned} \qquad (3)$$

Similarly, we can write the lane selection as:

$$\begin{aligned}
(l_y - l_y^{ref}(l_0)) \quad &= 0 \quad OR \\
(l_y - l_y^{ref}(l_1)) \quad &= 0 \quad OR \\
&\vdots \quad OR \\
(l_y - l_y^{ref}(l_N)) \quad &= 0
\end{aligned} \qquad (4)$$

where $l_y^{ref}(l_0)$ is a reference particularised in lane l_0. In plain words, each agent needs to follow the reference associated with one lane.

In a particular case with four lanes, this can be rewritten as a set of AND constraints to be included in the optimisation problem

$$\begin{aligned}
& C_{cl}: \\
& (l_y^{ref} - l_y^{ref}(l_0)) \in [-M(c_{i,j}^1 + c_{i,j}^2), M(c_{i,j}^1 + c_{i,j}^2)] \\
& (l_y^{ref} - l_y^{ref}(l_1)) \in [-M(1 - c_{i,j}^1 + c_{i,j}^2), M(1 - c_{i,j}^1 + c_{i,j}^2)] \\
& (l_y^{ref} - l_y^{ref}(l_2)) \in [-M(1 + c_{i,j}^1 - c_{i,j}^2), M(1 + c_{i,j}^1 - c_{i,j}^2)] \\
& (l_y^{ref} - l_y^{ref}(l_3)) \in [-M(2 - c_{i,j}^1 - c_{i,j}^2), M(2 - c_{i,j}^1 - c_{i,j}^2)]
\end{aligned} \qquad (5)$$

Regarding the definition of the cost function, we will only enforce a local behaviour through the term $J_n^k(x_n^k, r_n^k)$, which is defined in a quadratic fashion. The structure of the results optimazation problem is a distributed MIQP. Consequently, it can be treated in a distributed manner through the application of the Benders Algorithm [11].

B. Pseudo-binary Distributed Lane Scheduler

Using binary variables proved to be an unsuitable solution due to the inability to solve the problem within tractable computational times. This motivated splitting Eq. (1) in a cascade fashion. A high-level scheduler plans the overlay of the platoon formation considering both global goals while a distributed local planner executes, if possible, those manoeuvres. With appropriate planning rates, this strategy allows solving both sub-problems in a fully distributed fashion, thus removing the need for a central unit and allowing both complex models and better control of their global behaviour. In this section, only the lane selection is going to be derived, as it will be interfaced with a state-of-the-art distributed planner in further sections. From a high-level point of view,

lane management can be seen as finding the formation that maximises traffic flow and minimises agent deviation from their goals. To avoid embedding complex dynamics each agent will be modelled as a point along the lane traversed. Each agent is represented as a set of L virtual agents, one per lane, traversing lanes with a common velocity v_k, redefining the problem as finding the set of virtual agents allowing the fastest individual velocity while keeping inter-vehicular distance constraints. The variables used per agent are v and a corresponding to the velocity and its change rate associated with each punctual agent. Terms l^c and u^c represent the occupancy of a given lane c and its change rate, respectively. Finally, s_c represents the arc length at which each virtual agent is located. Note that the set of variables x_i is initialised with the latest available information obtained from the local distributed planner. Each virtual agent x_i is initialised by projecting the current position of its physical counterpart into all the lanes, providing a reasonable estimation, even with a naive agent model. Furthermore, a soft collision check is performed so that for a virtual agent to be selected as active it must respect a safety distance from any other active agent. Individual goals are introduced to the problem via a rewarding factor d_l, which is active when the desired goal is within a certain range. In particular, whenever an agent gets close to a divergent situation critical for its individual goal d_l will increase accordingly to prioritise a formation that avoids penalising the individual goal of that agent, thus accounting for its need to follow the divergent lane. The term d_l is computed offline by each agent and its expression can be found in Eq. (6). Being s the current arc length of the reference, $d_{c,r}$ the distance between the reference and the current lane, d_{crit} a threshold representing the urgency of the lane change and w an appropriate weighting factor. The operator $d_{c,r}(s)$ is a projection in the interval $[s, s_{lim}]$ evaluating the lateral distance between the same point projected with respect to both lanes, allowing to evaluate whether or not those lanes are diverging

$$d_l(s, lt) = \begin{cases} 0 & \text{if} \quad (d_{c,r}(s) < d_{crit}) \\ w(d_{c,r}(s)) & else \end{cases} \tag{6}$$

It can be seen that two different dynamics appear within this optimisation problem. On one hand, there is the naive estimation of the position and collision check while on the other hand, we aim to decide the target lane for each vehicle. This nature motivated the definition of two iteration horizons, named H_l and H_a, where the relation $H_l = kH_a$ holds. In this way, for each lane change, we computed k pose estimations and collision checks preventing unfeasible manoeuvres such as vehicle overtaking within the same lane, the resulting optimisation problem can be found

$$\begin{aligned} \min_{u,l} \quad & \sum_{n=0}^{N} \left(\sum_{k=0}^{H_l} \left(J_{ml} + J_{pg} + J_u + \sum_{t=0}^{H_a} (J_v + J_\sigma) \right) \right) \\ \text{s.t.} \quad & \\ & l^n_{c,k} \in [0, 1] \\ & v^n_{k,t} \in [\underline{v}, \overline{v}] \\ & a^n_{k,t} \in [\underline{a}, \overline{a}] \\ & \Delta u^n_{l,k} \in [\underline{\Delta u}, \overline{\Delta u}] \\ & C_l; C_a; \end{aligned} \tag{7}$$

where the constraint sets C_l and C_a associated with the lane and agent dynamics can be found in Eq. (8) and Eq. (9), respectively. Note that the term C_a models the agents as points along a given lane n described by their position s^n and velocity v^n. For clarity purposes, subindexes a_+ and a_- have been used to represent variables of consecutive time instants, e.g v_+ is temporally one step ahead of v_-

$$\begin{aligned} & C_{cl}: \\ & \quad \text{for:} \quad \forall n \in N \\ & \qquad \sum_{k=0}^{N} \sum_{c=0}^{L} (u^n_{c,k}) = 0 \\ & \quad \text{for:} \quad \forall k \in [0, H_l], \forall n \in N, \forall c \in L \\ & \qquad l^n_{c,k} = l^n_{c,k-1} + u^n_{c,k} \\ & \qquad u^n_{c,k} = u^n_{c,k-1} + \Delta u^n_{l,k} \end{aligned} \tag{8}$$

$$\begin{aligned} & C_a: \\ & \quad \text{for:} \quad \forall k \in [0, H_l], \forall t \in [0, H_a], \forall n \in N, \forall c \in L \\ & \qquad s^n_{c,+} = s^n_{c,-} + v^n_{k,t} dt \\ & \qquad v^n_+ = v^n_- + a^n_{k,t} dt \\ & \qquad l^n_{c,k,t} l^j_{c,k,t} (\| s^n_{c,k,t} - s^j_{c,k,t} \| - d_{min}) \geq 0 \quad \forall j \in N \end{aligned} \tag{9}$$

The family of terms J_i, defined in Eq.(10), refers to different system variables penalised in the cost function, ultimately defining the system behaviour. In particular, J_{ml} penalises formations with agents between two lanes, J_{pg} penalises goal-diverging paths, J_u penalises lane changes with low reward and finally J_s paths that traverse a wider region. Note that each sub-cost is weighted to adjust its role in the optimisation

$$J_{ml} = w_{ml} \sum_{m=0}^{k-1} \sum_{i=0}^{L} (4 l^n_{i,m} - 4 (l^n_{i,m})^2) \tag{10a}$$

$$J_{pg}(n) = w_{pg} \sum_{k=0}^{H} \sum_{c=0}^{L} d_l(c) l^n_{c,k} \tag{10b}$$

$$J_u(n, k) = w_u (u^n_{n,k})^2 \tag{10c}$$

$$J_v(n, k) = -w_s l^n_{n,k,t} l^j_{c,k,t} (s^n_{c,k,t} - s^n_{c,0,0}) \tag{10d}$$

This problem does not naturally distribute into N sub-problems due to the collision avoidance constraints, which depend on neighbouring $l^n_{c,k}$ values, otherwise, the problem

would be directly separable. This structure obeys a non-linear problem with complicating constraints, which makes it solvable through the OCD algorithm presented in [11]. As mentioned, the constraint set that couples each subsystem is related to the collision avoidance, from now on C_{cc}. Decoupling them via OCD generates N sub-problems, one per agent, as seen in Eq. (11) for an arbitrary agent i where $\hat{X}_j$ are the set of variables from other sub-problems. Note that both C^i_{cl}, C^i_a refer to the local subset of constraints that belong to agent i and $f_{lc}(\cdot)$ refers to the appropriate collision avoidance constraint in the local set C^i_{cc}

$$\begin{aligned}\min_{u_a, l_a} \quad & \sum_{k=0}^{H_l}\left(J_{ml}+J_{pg}+J_u+\sum_{t=0}^{H_a}(J_v+J_\sigma)\right)+ \\ & \sum_{c=0}^{L}\sum_{k=0}^{H_l}\sum_{t=0}^{H_a}\left(\lambda^n_{k,t} f_{lc}(s^i_{c,k,t}, \hat{s}^j_{c,k,t})\right) \\ \text{s.t.} \quad & \\ & l^i_{c,k} \in [0,1] \\ & v^i_{k,t} \in [\underline{v}, \overline{v}] \\ & a^i_{k,t} \in [\underline{a}, \overline{a}] \\ & \Delta u^n_{l,k} \in [\underline{\Delta u}, \overline{\Delta u}] \\ & C^i_{cl}; C^i_a; C^i_{cc};\end{aligned} \tag{11}$$

An approximation to the global solution can be found by iteratively solving the N subproblems conforming to the lane scheduler, following the distributed optimisation strategy in Algorithm 1.

Algorithm 1 Optimal Condition Decomposition

1: **while** not convergence **do**
2: Collect current x_i and set $\hat{x}_i = x_i$
3: Update $\lambda_n = \lambda_n + \alpha h(x^k_i, x^k_n)$
4: Compute a step of i_{th} optimisation problem
5: Forward x_n through the system

IV. Case Study: Highway Collaborative Navigation

In this case study, we aim to replicate an environment where a group of N agents enter a highway and rearrange themselves into a platooning formation while traversing the environment as close as possible to the maximum allowed velocity, keeping a safety distance between agents and respecting both the physical limits of the vehicle and the highway. By the end of the track, the road diverges into different destinations, and the platoon should dismantle itself so that individual goals are respected. Each vehicle is modelled as a scale car and with a unique goal path G_p, defined by a set of arc segments of length l and curvature k, being $G_p = [l_0, k_0, l_1, k_1, ..., l_N, k_N]$. Global references, e.g. the sequence of roads that the agent must follow to reach its destination, are considered to be given in a global planner. Additionally, we consider that the environment presented in Fig. 7 is divided by a certain number of lanes, again modelled by curvilinear segments. Thus, agents can position themselves in any formation as long as they remain within the lane limits and keep inter-vehicular safety distance. In this particular case, global references were modelled as a target lane to be followed which, as explained in previous sections, was accounted for via the divergence factor $d_l(s, lt)$. Agents are considered to have a low-level controller that ensures that the local plans are followed perfectly. As presented previously in this paper, lane changes are proposed via a relaxed decision-making layer to be forwarded to a lane-free local planner that adjusts road limits to represent the available space for the lane change. In this manner, for each lane change scheduled, vehicles have a time window to try to safely rearrange themselves ultimately converging into the new formation designed by the scheduler, or at least a feasible subset of manoeuvres.

A. Local planning

In this section, the interfacing procedure between the proposed lane scheduler and a low-level planner will be presented. In particular, it was already covered how both security and accuracy are compromised in the design of the scheduler so we aim to enforce them in the local planner. Following the collaborative spirit of this work, the algorithm used as a local planner is a lane-free distributed quadratic planner, originally presented in [12], with minor modifications. For clarity purposes, a concise formulation is recalled here

$$\begin{aligned}\min_{\Delta u^k_n} \quad & \sum_{k=0}^{H}(J^k_n(x^k_n)) \\ \text{s.t.} \quad & \forall k \in [0, H] \\ & x^{k+1}_n = x^k_n + (A_n(\zeta^k_n)x^k_n + B_n(\zeta^k_n)u^k_n)T_s \\ & u^k_n = u^{k-1}_n + \Delta u^k_n \\ & u^k_n \in [\underline{u^k_n}, \overline{u^k_n}] \\ & e^k_{y_n} \in [\underline{e^k_{y_n}}, \overline{e^k_{y_n}}] \\ \text{s.t.} \quad & \forall k \in [0, H], \forall j \in N \setminus n \\ & \hat{\vartheta}^{k,n,j} p^{k,n} \leq \frac{-D_{sf}}{2}\end{aligned} \tag{12}$$

where x^k_n are vehicle states, u^k_n vehicle inputs, $e^k_{y_i}$ the lateral error and σ a set of slack variables. Finally, $\hat{\vartheta}^{k,n,j} p^{k,n}$ refers to a hyperplane collision constraint and D_{sf} is the minimum security distance. When dealing with the lane reconfiguration problem, two minor modifications need to be added to this structure. Firstly, the bounds of e^k_y are adjusted online to allow lane change. Additionally, fast convergence to the target lane is enforced via the cost function so that the manoeuvre is done swiftly. Secondly, security distance is relaxed for the agent that is changing lanes, forcing neighbouring agents to give way. Agents have a one-second time window to perform the lane change, which proved to be enough in this experimental setup. After that time, if a suitable path to the new lane cannot be planned the manoeuvre is aborted, returning the agent to its original state.

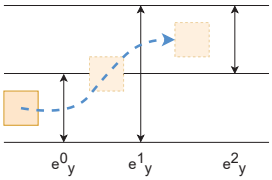

Fig. 1: Road limit modification

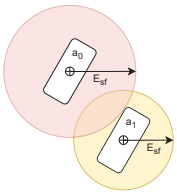

Fig. 2: D_{sf} relaxation with a_1 performing a lane change

B. Results

The effectiveness of the algorithm proposed in Section III-A is going to be assessed by applying it under the conditions presented in Section IV. The results of this section were generated using CASADI [13] and IPOPT [14] as a non-linear solver. Finally, all tests were carried out in a laptop with 16Gb of RAM, i7-13700H CPU and no specific hardware acceleration. It is worth noting that solving Eq. (7) is an NP-Hard problem [15]. Thus, results are susceptible to the experimental conditions. To assess performance, we will consider how well can the system track both the lane changes and individual goals. Additionally, the agent velocities will be studied.

1) NL vs Binary approach: As an initial comparison, we can see in Fig. 3 how the proposed scheduler behaves in the same environment traversed by the MIQP planner, presented in Section III-A. As it can be seen, the trajectories generated by its non-linear counterpart are more logical from a human point of view since once an agent locates itself in an empty lane there is no incentive to change anymore. Moreover, in terms of computing time we can see how adding agents increases the overall complexity of each subproblem.

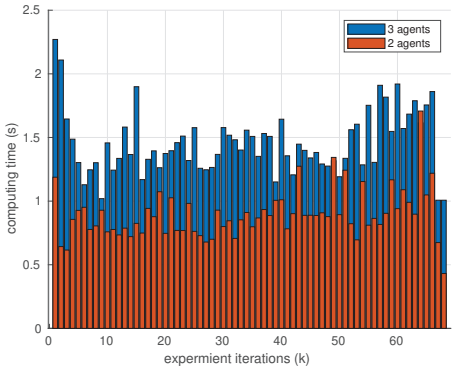

Fig. 3: Computational times of the straight road experiment

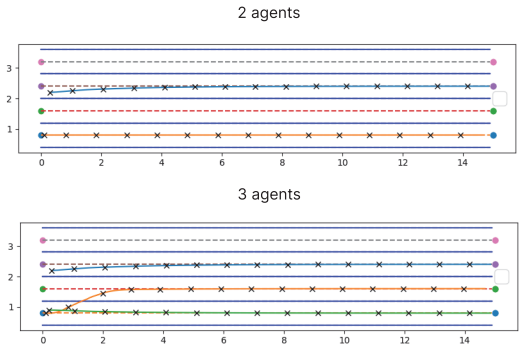

Fig. 4: Generated trajectories in a straight road

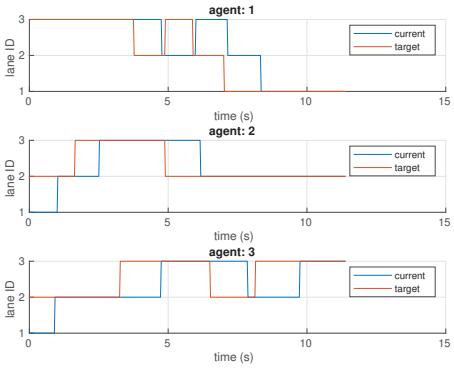

Fig. 5: Lane switching tracking of the local planner

2) Highway-like scenario: In a complex scenario, we can see in Fig. 5 how the proposed lanes are tracked within a one-second time window, which is the requirement enforced in the local planner by design. In terms of the computing time, in Fig. 6, it can be observed a remarkable variance within iterations due to the nature of the problem, being in all cases solved under $3s$. Conveniently, in this particular experiment, the scheduler provided lane changes within $2.5s$ thus we were able to sequentially run the algorithm and provide references promptly. Note that the computing time could be reduced by adjusting the terms H_a and H_l, which refer to both agent collision checks and the lane change horizon respectively. In case of failure, there is no critical risk involved for the agents, as tight safety measures and enforced through both the local planner and a control layer.

Finally, the overall formation can be seen in Fig. 7, where the agents move according to their target lane until they get close to divergent lanes. Then, the system rewards moving agents towards their individual goals, successfully allowing each agent to reach their destination.

V. Conclusions and Future Work

In this work, a novel non-linear distributed scheduler for autonomous vehicles based on the Optimality Condition Decomposition algorithm is proposed and tested its viability in a shared environment. Furthermore, the decision-

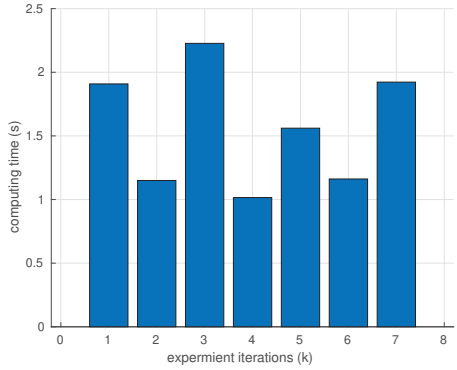

Fig. 6: Computational time of the NL lane scheduler

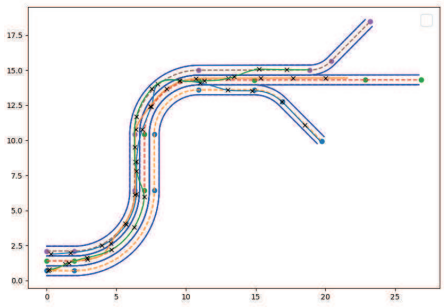

Fig. 7: Path traversed by the agents

making algorithm was coordinated with a lower-level planner, providing a fully decentralised solution to navigated highway-like environments divided by lanes. Additionally, the viability of an integer-based approach was ruled out as the computational costs involved in solving this family of problems are prohibitive for real-time applications, even after solving it in a distributed manner. During this work, a step forward was made towards studying coordination problems in autonomous driving, with a special focus on agent interaction and individual goal satisfaction. A novel fully distributed non-linear scheduler has proved to be able to work in a cascade fashion with state-of-the-art local planning to provide a complete pipeline in terms of distributed autonomous coordination. Furthermore, the system not only accounts for the steady platoon navigation but considers individual agent goals when rearranging the formation, which thanks to the local planner can be done while ensuring the safety of the agents. All in all, we consider the current research line has provided appealing results that need to be explored in both realistic environments and a wide variety of scenarios to assess to what extent the results of this research are extendable to a wider context. There is a vast number of potential deadlocks, such as intersections, and critical scenarios where it is still uncertain if algorithms of this nature will be able to solve. Moreover, modelling human behaviours such as risk evaluation and selflessness, which are key when driving, is not a trivial behaviour. In future work, we aim to study how this human perception can be represented within the algorithm to tackle situations where a higher degree of interaction is required.

ACKNOWLEDGMENT

This work has been co-financed by the Spanish State Research Agency (AEI) and the European Regional Development Fund (ERFD) through the project SaCoAV (ref. MINECO PID2020-114244RB-I00).

REFERENCES

[1] Z. Feng, W. Song, M. Fu, Y. Yang, and M. Wang, "Decision-making and path planning for highway autonomous driving based on spatio-temporal lane-change gaps," *IEEE Systems Journal*, vol. 16, no. 2, pp. 3249–3259, 2021.

[2] S. Zhang, W. Deng, Q. Zhao, H. Sun, and B. Litkouhi, "Dynamic trajectory planning for vehicle autonomous driving," in *2013 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, 2013, pp. 4161–4166.

[3] G. Cesari, G. Schildbach, A. Carvalho, and F. Borrelli, "Scenario model predictive control for lane change assistance and autonomous driving on highways," *IEEE Intelligent transportation systems magazine*, vol. 9, no. 3, pp. 23–35, 2017.

[4] J. Nie, J. Zhang, W. Ding, X. Wan, X. Chen, and B. Ran, "Decentralized cooperative lane-changing decision-making for connected autonomous vehicles," *IEEE access*, vol. 4, pp. 9413–9420, 2016.

[5] Y. Zheng, B. Ran, X. Qu, J. Zhang, and Y. Lin, "Cooperative lane changing strategies to improve traffic operation and safety nearby freeway off-ramps in a connected and automated vehicles environment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4605–4614, 2019.

[6] D. Wang, M. Hu, Y. Wang, J. Wang, H. Qin, and Y. Bian, "Model predictive control–based cooperative lane change strategy for improving traffic flow," *Advances in mechanical engineering*, vol. 8, no. 2, p. 1687814016632992, 2016.

[7] X. Duan, C. Sun, D. Tian, J. Zhou, and D. Cao, "Cooperative lane-change motion planning for connected and automated vehicle platoons in multi-lane scenarios," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[8] Y. Chen, C. Dong, P. Palanisamy, P. Mudalige, K. Muelling, and J. M. Dolan, "Attention-based hierarchical deep reinforcement learning for lane change behaviors in autonomous driving."

[9] M. Wang, S. P. Hoogendoorn, W. Daamen, B. van Arem, and R. Happee, "Game theoretic approach for predictive lane-changing and car-following control," *Transportation Research Part C: Emerging Technologies*, vol. 58, pp. 73–92, 2015.

[10] B. Alrifaee, M. G. Mamaghani, and D. Abel, "Centralized non-convex model predictive control for cooperative collision avoidance of networked vehicles," in *2014 IEEE International Symposium on Intelligent Control (ISIC)*, 2014, pp. 1583–1588.

[11] A. J. Conejo, E. Castillo, R. Minguez, and R. Garcia-Bertrand, *Decomposition techniques in mathematical programming: engineering and science applications*. Springer Science & Business Media, 2006.

[12] M. Facerias, V. Puig, and A. Stancu, "Non-linear distributed mpc coordination of autonomous vehicles using optimality condition decomposition," in *European Control Conference 2024*, 2024.

[13] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, 2018.

[14] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, pp. 25–57, 2006.

[15] S. Sahni, "Computationally related problems," *SIAM Journal on Computing*, vol. 3, no. 4, pp. 262–279, 1974.