Modeling Slipstreaming Effects in Vehicle Platoons

Michele Segata*, Andrea Stedile†, Renato Lo Cigno‡
*Faculty of Computer Science, Free University of Bolzano, Italy
†University of Trento, Italy
‡DII – Dept. of Information Engineering, University of Brescia, Italy

***Abstract*—Vehicle platooning can produce significant fuel saving due to reduced aerodynamic drag. In the design of future cooperative driving systems, quantifying such benefits will be of utmost importance, because it will need to be considered in designing, forming and managing platoons. This work aims at developing an Open Source framework for modeling aerodynamics effects. We perform a Computational Fluid Dynamics (CFD) study about the aerodynamics of vehicle platoons, describe a model that exploits the resulting measurements, and implement it inside a road traffic simulator to show how savings can be estimated by means of simulations. Furthermore, we publish the necessary tools required to reproduce the results, enabling further research. Strictly speaking, this work does not deal with vehicular *networks*, but it contributes to tackle the problem of optimizing the management and control of platoons through proper use of communications from the point of view of fuel consumption and battery usage.**

I. Introduction

The cooperative driving research field is notoriously multi-disciplinary. Wireless networks ensure communication, but we should consider vehicle dynamics, control systems, and traffic engineering to see the big picture. Vehicle dynamics and control systems are required for driving autonomous vehicles, and traffic engineering to evaluate the benefits of their application. Research in Cooperative Autonomous Vehicles (CAVs) deals with the improvement of transportation: reducing traffic congestion, pollution, costs, and increasing safety; platooning is one of the applications with such goals.

A platoon is nothing but a road-train of vehicles, where the leading one drives (either manually or autonomously) and the others autonomously follow at a close distance. The longitudinal autonomous following function is implemented by a control system called Cooperative Adaptive Cruise Control (CACC), which exploits locally-measured (e.g., radar-measured distance) and wirelessly-received (e.g., leading vehicle speed or acceleration) data to maintain the target inter-vehicle distance. Wireless communication allows the inter-vehicle distance to be small without compromising safety, which cannot be achieved using local sensing only (e.g., by employing a standard Adaptive Cruise Control (ACC)).

The reasons for reducing the distance are mainly two. The first one is improving the road infrastructure use. Human driving is subject to speed-dependent safety distances. In particular, most European countries mandate the 2 seconds rule [1] which means that, if a vehicle is driving at $130\,\mathrm{km/h}$ ($\sim 36\,\mathrm{m/s}$), the driver should keep a distance larger than $72\,\mathrm{m}$. If we consider a car length to be around $4\,\mathrm{m}$ to $5\,\mathrm{m}$, then more than $93\,\%$ of the road is empty (and thus wasted). Reducing the inter-vehicle distance means increasing the capacity of the road at no cost and reducing traffic congestion. With lower congestion, traffic flows more smoothly and fuel consumption decreases.

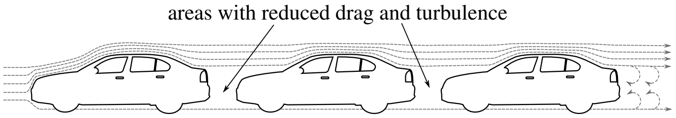


Figure 1: Slipstreaming effect in vehicle platooning.

The second one, which is the focus of this work, is related to slipstreaming, sketched in Fig. 1. At high speed, air drag is the major source of resistance, as air drag grows with the square of speed. Simplistically speaking, the main sources of air drag on a vehicle are two: air pressure, which (with respect to normal air pressure) is higher at the front and lower at the back, and turbulence, which is evident at the back; friction on the surface of the vehicle is marginal. If two vehicles are close to each other, they experience lower air drag, because the first vehicle's turbulence is disrupted by the second, and the second vehicle's air pressure at the front is lowered.

Lower air drag translates into lower fuel consumption and emissions, but one of the still open problems in platooning is quantifying such savings, because air drag phenomena are non linear and extremely complex, thus simplistic solutions as "the closer the better" might prove false. An additional problem is related to the high-level coordination of platoons, which need to be created and maintained according to some rules [2], [3]. As an example, consider a lone vehicle contacting the Traffic Authority (TA) to search for an existing platoon to join to reduce fuel consumption. If the platoon to be joined is several kilometers ahead, the amount of fuel required to speed up and approach the platoon might be more than the fuel saved by the slipstream effect [4]: it is more convenient to continue driving alone or trying to form a new platoon with neighboring vehicles. The TA needs means to estimate such savings to take the proper decision. Last but not least, models for slipstreaming are needed for the simulation of cooperative driving systems, which remains the only viable solution to study the effects before deploying the system.

A. Motivation and Contribution

Studies that analyze slipstreaming of truck or car platoons do not derive a general model that can be used to estimate savings. Furthermore, they do not give access to the tools to build on top of them or reproduce the results. Nonetheless, it may not be evident why having such a model would be of interest for the vehicular network research community and a networking conference.

As already studied by our and other research groups (see [5]–[10] to cite a few works), communication is fundamental: local flow communications pertain to platoon control (for example, longitudinal, lateral and maneuver control), whereas global flow communications to platoon formation, management and so forth. Optimizing platoons for fuel consumption introduces other challenges, where networking and computing (distributed, cloud, edge, ...) play a fundamental role too. As will be shown in the paper, the problem is a function of many parameters, such as the speed of the vehicles and their distances, with a non linear nature. It requires a constant flow of information, which is neither local or global, to coherently transfer information from the platoon to the infrastructure and back, and to compute the compensations necessary to spread the advantages evenly among all platoon members. We need to study and define what this flow of information is: which is the relevant information, how large it is, how often it has to be communicated. Moreover, we need to understand if the problem is best modeled as a dynamic problem, implicitly coupled with the control algorithms, or instead if is better to model it as a step-wise static problem, updating the solution from time to time (seconds? minutes?) with a time-scale that ensures decoupling with the control algorithms, i.e., it ensures that there is enough time between successive updates of the solution to let the control algorithms reach a steady state of the platoon and avoid any possible source of instability.

All of these challenges, typical of vehicular networks, require a proper slipstreaming model. This motivates our work, whose contributions can be summarized as follows:

- We perform a Computational Fluid Dynamics (CFD) study using the Open-source Field Operation And Manipulation (OpenFOAM) software: we analyze slipstreaming of simplified vehicles called Ahmed bodies and derive general results;
- We publish the tools to enable further research and reproduce the results;
- We provide a simple model to interpolate values missing from CFD analyses; and
- We implement the model inside the SUMO traffic simulator to show how fuel saving can be estimated in a simulation environment.

II. Related Work

Before reviewing literature, we need to introduce the definition of drag force F_D [11, Eq. 1.2], or simply drag:

$$F_D = \frac{1}{2}\rho v^2 C_D A. \quad (1)$$

Eq. (1) computes the resisting drag force [N] an object experiences when traveling at a speed v [m/s] through a fluid with a density ρ [kg/m^3]. A [m^2] represents the cross-section area of the object with respect to the motion direction, while C_D is a constant called drag coefficient that depends on the shape of the object. The lower the C_D value, the better the aerodynamic properties. The C_D is measured inverting Eq. (1) in controlled conditions (e.g., inside a wind tunnel). Measurements that study slipstreaming compute the drag force when objects are placed close to each other and compute the change in C_D. Studying slipstreaming thus translates into measuring the change of C_D. A reduction of C_D by an amount x does not translate in the same reduction of consumption, as other resisting forces, such as rolling resistance, exist. Still, considering that at high speeds drag contributes to a large fraction of the consumption (e.g., around $70\,\%$ at $120\,\mathrm{km/h}$ [12, Fig. 22]), the impact is non-negligible.

In the literature we find different studies on the effect of vehicle platooning on drag, starting from fundamental books on aerodynamics of vehicles [11, Sect. 8.6.1]. The book shows the results for small convoys of buses. In a platoon of two buses driving at $80\,\mathrm{km/h}$ at an inter-vehicle distance of $5\,\mathrm{m}$, the first vehicle experiences a C_D reduction around $7\,\%$ to $8\,\%$ and the second one around $40\,\%$. In a convoy of three, at the same speed and distance, the third vehicle experiences a C_D reduction of $50\,\%$. A drag reduction is not only experienced at small distances, but also at moderate ones (e.g., at $50\,\mathrm{m}$, $20\,\%$ for the second vehicle and $30\,\%$ for the third). As the book focuses on aerodynamics of single vehicles, it presents these results to highlight the potential of driving in convoys of vehicles and does not investigate the question further. In [13] the authors derive a simple model based on this data. The model does not consider vehicle types and positions for mathematical tractability, but it testifies the need for a generic model for cooperative platooning applications.

More in-depth results have been obtained in the scope of the California PATH project [14]. In one of the project reports [12], the authors quantify vehicle drag as a function of vehicle spacing through wind tunnel measurements, for platoons of up to four vehicles. The authors made a huge effort in publishing all the measurement results in the report, which are otherwise challenging to reproduce. The results clearly show the benefit of platooning, but also its limits, by deriving some general rules. For example, for the leading vehicle, there is no reduction in C_D for inter-vehicle distances larger than one vehicle length, independently of the number of vehicles in the platoon, and that the last vehicle is the one that experiences the least reduction. In addition, the paper derives a simple generic model, assuming that data for four vehicles is enough to derive such a model. Yet, the model considers homogeneous vehicles and inter-vehicle distances, and it is not possible to compute the reduction in drag coefficient for different vehicles or different distances.

Another project focusing on platooning was SARTRE [15]. The final report [16, Sect. 13] includes measurements of a platoon composed by two trucks and three cars. Computer simulations performed with the open-source software OpenFOAM are used to obtain drag coefficients and field operational tests to obtain fuel saving percentages. The results are very encouraging, but are limited to the platoon formation under consideration. Furthermore, no general model is derived, and OpenFOAM configuration files are not available.

Other studies investigate the relationship of vehicle ordering and slipstreaming [17]. Aerodynamically speaking, it might seem evident that having a truck in front of a car is more favorable than the other way round, but in this study the authors show something not so obvious, i.e., that fuel saving when a heavier truck is in front of a lighter differs from the saving of two identically weighting trucks. In the latter case, fuel saving is larger. This is of fundamental importance, as such details need to be considered by the Traffic Authority to organize platoons.

The authors of [18] develop a model to simulate the aerodynamic characteristics of bodies in a platoon formation using simplified shapes called Ahmed bodies [19] and simplified truck geometries organized in platoons of up to four vehicles. As in [12], the results are obtained using inter-vehicle distances relative to vehicle lengths (0.2, 0.5, and 1 vehicle length). The work is particularly interesting for the insights it provides, showing that, due to the complexity of fluid dynamics, a smaller inter-vehicle gap might not always translate into lower air drag for a certain vehicle. In certain conditions, the C_D might even increase with respect to the nominal value [20, Fig. 4.1].

All the works in the literature provide valuable insights on the properties of slipstreaming in platoons: however, they lack to provide a general model and the tools required to reproduce the results.

III. Modeling and Analyzing Slipstreaming

To analyze the slipstream effect we exploit OpenFOAM[1], an Open Source CFD software that enables the analysis of fluid flow [21]. Vehicles traveling through air represent one such case. Here we describe the essentials; the interested reader should refer to the official OpenFOAM documentation.

OpenFOAM simulations are referred to as *cases*. Inside a case directory we typically find three subdirectories:

- "constant": it contains the geometries of the objects and the physical properties of the fluid;
- "system": it includes the configuration files of the simulation, such as the time steps, algorithms to be used, etc.; and
- "times": not a single directory, but rather a set of directories that represent time steps. Users must mandatorily specify the "0" directory, including – among other things – the initial conditions (i.e., at time $t = 0$). Additional directories are created by OpenFOAM to store the results of the simulation at different time steps, with the names of the directories being the time step values.

OpenFOAM includes utilities that perform pre- and post-processing, such as tools to create meshes (the discretization of the space around the objects), and solvers, that solve specific CFD problems. It also offers a GUI software (ParaView) that is used for post-processing and visualization of the results.

In the analyses we use Ahmed bodies [19]. The Ahmed body, depicted in Fig. 2, is a benchmark model that captures some of the essential features of air flow around real vehicles. Being much simpler than precise car models, it reduces the computational burden. The shape has a length of $1.044\,\mathrm{m}$, a width of $0.389\,\mathrm{m}$, and a height of $0.388\,\mathrm{m}$ including the legs. A property of the Ahmed body is represented by the rear slant angle which, in our simulations, is set to $25°$. For what concerns aerodynamics, the Ahmed body has a C_D of roughly 0.29 [19, Fig. 4][2]. In [20, Tab. 4.1] the C_D is equal to 0.30, while online experiments report a C_D between 0.28[3] and 0.306[4].

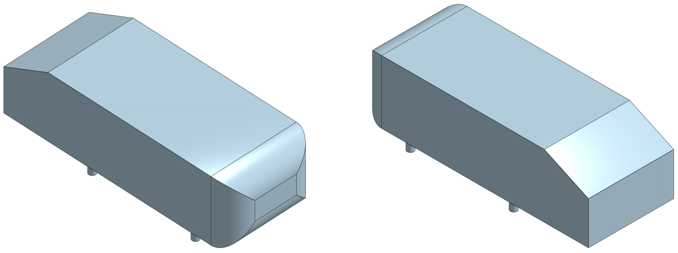

(a) front view (b) rear view

Figure 2: Three-dimensional view of the Ahmed body.

Parameter	Value
Free stream velocity	$63.7\,\mathrm{m/s}$
Air kinematic viscosity	$1.53 \times 10^{-5}\,\mathrm{kg/(m\,s)}$
Air density	$1.196\,\mathrm{kg/m^3}$

Table I: Single Ahmed body simulation parameters.

A. Ahmed body in isolation

The rationale of the CFD simulations is to position multiple Ahmed bodies at different distances and measure the C_D of each Ahmed body. We first perform a simulation with a single Ahmed body in isolation and verify that the resulting C_D is about in the same range of those found in the literature. Tab. I lists the most relevant simulation parameters. The complete simulation setup is available in our repository[5], so results can be reproduced. We use the same parameters of validation tests found online and reproduce their conditions, including a very high, non vehicular speed. Air properties (viscosity and density) correspond to 22° at sea level.

Fig. 3 shows the value of C_D as function of the simulation step, the initial wide variations are due to the convergence algorithm that starts with large correction coefficients forcing oscillations before reducing them and converge; we compute C_D as the average over the last 100 iterations. The result is a C_D of 0.303, which differs by roughly $4\,\%$ from the value extrapolated from the original paper [19] and lies within confidence intervals. Following the notation in [20], we denote the C_D of the Ahmed body in isolation with C_{D_∞}.

[1] https://www.openfoam.com/

[2] The exact value is not reported in the paper but can be extrapolate from the cited figure.

[3] https://www.simscale.com/forum/t/ahmed-body/64235, visited Aug 13, 2020.

[4] https://www.simscale.com/docs/validation-cases/aerodynamics-flow-around-the-ahmed-body/, visited Aug 13, 2020.

[5] https://github.com/AdvancedNetworkingSystems/openfoam-platoons.git

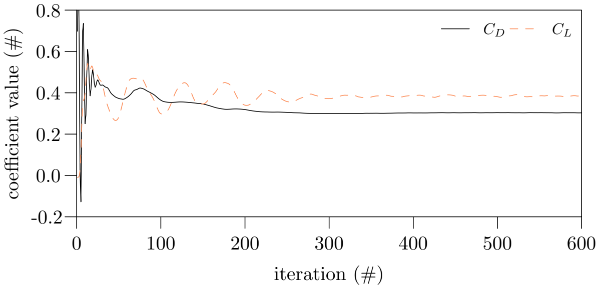
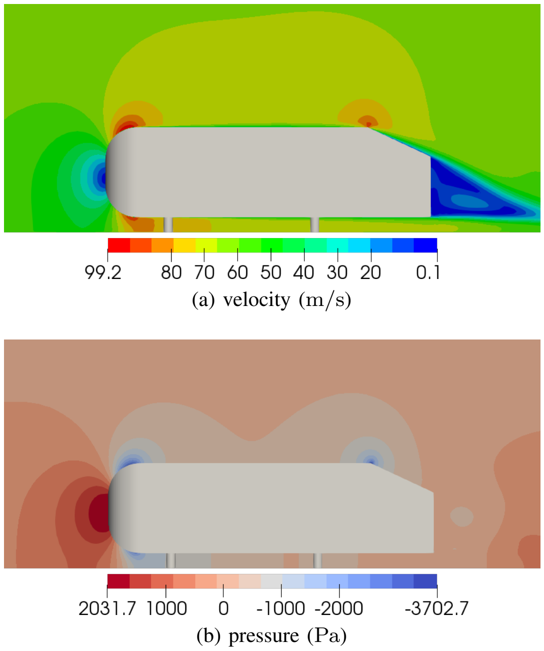


Figure 3: Results of the simulations for the Ahmed body in isolation.



(a) velocity (m/s)

(b) pressure (Pa)

Figure 4: Air velocity magnitude and pressure plots for the single Ahmed body simulation.

Fig. 3 also shows the lift coefficient C_L, a quantity of interest that OpenFOAM calculates. It indicates how much an object is pushed upward by lift forces due to the air stream. We do not consider it in this work, but future works might investigate the impact of platooning on lift and thus on vehicle stability.

Fig. 4 shows magnitude of air velocity and pressure around the Ahmed body. Far from the Ahmed body, the stream velocity is 63.7 m/s (light green), meaning that air there is unaltered. Air at the front and at the rear of the Ahmed body is instead stationary (blue), meaning that air is traveling at the same speed of the vehicle. With respect to the front, we observe an increase of pressure (Fig. 4b) which causes frontal drag. The wake at the back is what followers exploit to reduce their frontal drag.

B. Platoons of Ahmed bodies

After verifying that our simulation setup is correct, we proceed by placing multiple Ahmed bodies in platoon arrangements, with a number of vehicles ranging from 2 to 5. We perform the simulations at homogeneous distances (using the same distance between each pair of vehicles). We choose the set of distances listed in Tab. II (“Real” columns); as the Ahmed body is smaller than a real vehicle, the distance in the simulation is smaller than the real one, scaled by a factor of 3.5, which is the same scale between the real vehicle length of 3.653 m (a small-sized car) and the Ahmed body length of 1.044 m. Even though in this work we consider platoons with same sized cars only, we plan to investigate the implications of having different vehicle types and our framework already allows doing it.

Simulation	Real	Simulation	Real
0.71 m	2.5 m	2.14 m	7.5 m
1 m	3.5 m	2.86 m	10 m
1.43 m	5 m	4.29 m	15 m
1.72 m	6 m		

Table II: Inter-vehicle distances in the simulation and the actual distance according to the scaling factor, to account for the fact that Ahmed bodies are smaller than real vehicles.

Fig. 5 summarizes the results of the simulations. To ease the interpretation of the results, the C_D of each vehicle is normalized against C_{D_∞} (0.303), which was computed for the Ahmed body in isolation. The trends of the experiments are consistent with the results in [20], [22]. In particular, the leading vehicle’s C_D is always reduced, whereas at small distances the last vehicle’s C_D is always increased. For the subset of distances (5 m, 6 m, and 7.5 m), the C_D are reduced or, at least, left unaltered. These preliminary results suggest that fuel savings from platooning should not be given for granted, because shortening the distance might not translate in reducing the C_D of every vehicle.

In general, the overall efficiency is increased, but gains are very different, highlighting the need of effective compensation mechanisms. Fig. 6 reports the average C_D/C_{D_∞} over all vehicles as function of distance. Filled points are used to indicate when $C_D/C_{D_\infty} < 1$ for all vehicles and empty points otherwise. The average C_D/C_{D_∞} is always smaller than 1, which indicates an overall increased efficiency. C_D/C_{D_∞} is smaller than 1 for all vehicles and all configurations only for distances of 6 m, 8 m, and 10 m and there is no much difference if the number of vehicles in the platoon is increased.

IV. Proof of Concept Implementation

We present here a first implementation of the slipstream effect in SUMO. Recall that the computational complexity of CFD makes it impossible computing the slipstream effect during SUMO simulations: to obtain the results presented in Sect. III-B, a high-end PC required a few tens of hours. The only viable solution is to pre-compute slipstream effects and import them later in SUMO. Furthermore, even with pre-computations, it is impossible to obtain results for all vehicle arrangements and inter-vehicle distances, thus an interpolation methodology is needed. An additional challenge which we leave as future work is understanding the bounds within which effects are worth being considered. For example, we conjecture that vehicles having the largest impact are the closest ones,

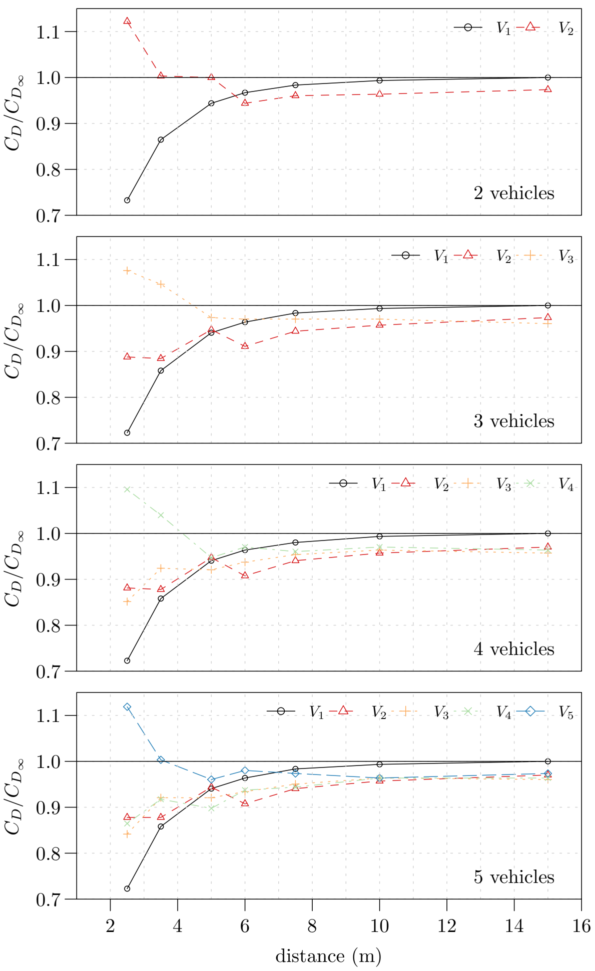


Figure 5: C_D/C_{D_∞} for platoon of Ahmed bodies with different number of vehicles.

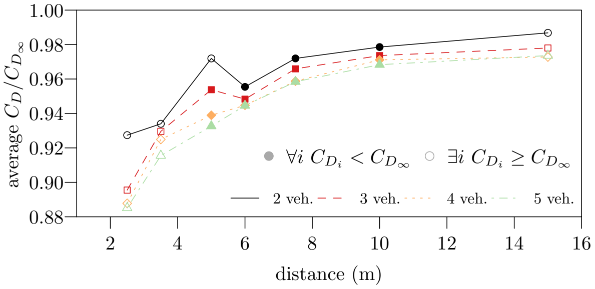


Figure 6: Average C_D/C_{D_∞} over all vehicles as function of the distance for multiple Ahmed bodies in a platoon formation. Filled points indicate that drag coefficients of all vehicles are smaller than C_{D_∞} while empty points indicate that, for at least one vehicle, the drag coefficient is larger than C_{D_∞}.

while further ones might simply be neglected. This needs to be verified, but it would limit the size of the CFD simulation campaign.

A. A New SUMO Device

The slipstream model is implemented as a SUMO *device*. A device is a software component that can be added to individual vehicles SUMO simulations. At each simulation step, a device carries out some functionality, such as computing a metric and update the vehicle's state. One example is the HBEFA device (the name comes from the HandBook Emission FActors for road transport [23]), which computes the amount of polluting emissions a vehicle produces. As devices can be enabled for specific vehicles, it is possible to obtain information from a fraction of the fleet.

The name of our prototype device is Slipstream (implemented within the MSDevice_Slipstream class). It works in symbiosis with the Battery device, which simulates the battery consumption of electric vehicles [24]. To compute the consumption, the Battery device considers numerous sources of energy loss, including wheels rolling resistance, road curvature and, of course, air resistance. The air drag loss component is computed as in Eq. (1), so we modified the Battery device to invoke the Slipstream device and obtain a (potentially different) C_D value at each simulation step. The reason why we opt for the Battery device is because it explicitly considers drag force (Eq. (1)) in the computation of the instantaneous power consumption. Considering the effects of drag simply requires to modify C_D according to the model, while for combustion models such as HBEFA this is not as simple because the effect of air drag is embedded in the model and would be hard to isolate. We believe this is possible, but outside the scope of this work.

During the initialization, the device loads a dataset that includes the measurement values obtained through CFD. The dataset is a collection of records, each of which describes a platoon: the vehicle types, the inter-vehicle distances, and the C_D/C_{D_∞} values for each vehicle. Each record has the following form:

$$\begin{array}{ccccc} t_1 & t_2 & \dots & t_{P-1} & t_P \\ d_1 & d_2 & \dots & d_{P-1} & - \\ c_1 & c_2 & \dots & c_{P-1} & c_P \end{array}$$

t_i are strings indicating the vehicle type and should match the vehicle types in the SUMO simulation. Custom vehicle types with specific properties (length, width, maximum speed...) can be added to SUMO if needed. d_i are real values indicating the distance from vehicle i to vehicle $i+1$. They might differ one another because the dataset can include values for non-homogeneous distances. Finally, c_i are the C_D/C_{D_∞} ratios for that specific platoon configuration. The dataset includes multiple records, one for each specific configuration. After the dataset is loaded, SUMO invokes the main logic of the slipstream device at each time step.

The device obtains from SUMO the list of neighboring vehicles. Assume that the device needs to find the air drag for a vehicle v of type t in position p in a platoon of P vehicles. We define a platoon as a list $((t_1, d_1), (t_2, d_2), \dots, (t_{P-1}, d_{P-1}), (t_P, -))$, where 1 and P are the first and the last vehicle in the platoon, and t_i, d_i have the meaning defined above. The distance d_P is undefined, so we simply set it to a dash symbol for coherence in the notation. Fig. 7 shows a graphical representation of a platoon.

The device searches for compatible records inside the dataset. A record R of size M is defined as a list $((t^R_1, d^R_1, c^R_1), (t^R_2, d^R_2, c^R_2), \ldots, (t^R_M, -, c^R_M))$. As for the platoon in the simulation, the distance d^R_M is undefined. A record is compatible if:

1) $M = P$: the record must include the same number of vehicles that there are inside the platoon; and
2) $t_i = t^R_i, i \in \{1, \ldots, P\}$: the types of vehicles in the platoon match the ones in the record.

If no compatible record is found, the device issues a warning and returns a C_D/C_{D_∞} value of 1, leaving the drag coefficient unaltered. Since a simulation is dynamic, distances between vehicles change continuously and are arbitrary. Thus, we cannot assume that there is "the right record". For sake of simplicity, we assume that, if compatible records exist, then they must be more than one. We need an interpolation technique that enables to derive the slipstream effect for the considered vehicle, i.e., perform the interpolation of the C_D/C_{D_∞} values of the two most similar records. To do so, we split the records into two sets: one for the records for which the distance between v and the predecessor and the successor are shorter than the actual distances (set $\mathcal{S}$), one for which it is longer (set $\mathcal{L}$). This sub-division has three different cases, i.e., v being the first vehicle, the last, or any vehicle in between. Formally, let $\mathcal{R}$ be the set of compatible records. If v is the first

$$\mathcal{S} = \{R : d^R_1 < d_1 \text{ and } R \in \mathcal{R}\} \tag{2}$$

$$\mathcal{L} = \{R : d^R_1 \geq d_1 \text{ and } R \in \mathcal{R}\}, \tag{3}$$

if v is the last

$$\mathcal{S} = \{R : d^R_{P-1} < d_{P-1} \text{ and } R \in \mathcal{R}\} \tag{4}$$

$$\mathcal{L} = \{R : d^R_{P-1} \geq d_{P-1} \text{ and } R \in \mathcal{R}\}, \tag{5}$$

and, if v is any vehicle in the middle

$$\mathcal{S} = \{R : d^R_{p-1} < d_{p-1} \text{ and } d^R_p < d_p \text{ and } R \in \mathcal{R}\} \tag{6}$$

$$\mathcal{L} = \{R : d^R_{p-1} \geq d_{p-1} \text{ and } d^R_p \geq d_p \text{ and } R \in \mathcal{R}\}. \tag{7}$$

The partitioning in Eq. (6) and (7) automatically discards some records that might be useful for computing the interpolation, i.e., the following two sets of records:

$$\{R : d^R_{p-1} < d_{p-1} \text{ and } d^R_p \geq d_p \text{ and } R \in \mathcal{R}\} \tag{8}$$

$$\{R : d^R_{p-1} \geq d_{p-1} \text{ and } d^R_p < d_p \text{ and } R \in \mathcal{R}\}. \tag{9}$$

We are aware that this proof-of-concept implementation is sub-optimal, and we only consider two records for computing the drag coefficient with a linear interpolation, while more complex approaches may yield better results. Developing smarter solutions is part of our future work as well as the estimation of errors introduced by simple linear interpolation.

B. Interpolation

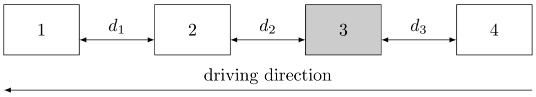


Figure 7: In this example the platoon is made of $P = 4$ vehicles and the vehicle for which the device needs to compute the slipstream effect is highlighted in gray ($p = 3$).

After defining $\mathcal{S}$ and $\mathcal{L}$, we refine the two sets using a selection algorithm to find the two records (one in $\mathcal{S}$ and one in $\mathcal{L}$) that are the most similar to the platoon considered as seen from vehicle v. From the perspective of v, the vehicles that have more influence are the two directly in front and behind it, then those preceding and following these respectively, and so forth. Without claiming to derive an optimal algorithm, given the non-linearity of the problem, we can use a greedy algorithm that extracts, at each iteration k, the subset of all records that minimize the sum of the squared distance errors of the two vehicles at distance (measured as position in the platoon) k from vehicle v from the two vehicles at distance $k - 1$. Formally, define for every iteration k and for every record R

$$\epsilon_f = d^R_{p-k} - d_{p-k}; \quad \epsilon_b = d^R_{p+k-1} - d_{p+k-1} \tag{10}$$

$$\Delta_R = \epsilon_f^2 + \epsilon_b^2 \tag{11}$$

ϵ_f represents the difference in distance between a record R and the actual distance in the platoon for the vehicles in position $p-k$ and $p-k+1$ (so looking towards the head of the platoon). ϵ_b is conceptually the same, but towards the tail. Δ_R is the sum of the squared distance errors that we want to minimize. The reason to use Δ_R instead of the sum of the distance errors in absolute value is that we consider a record to be more similar to the platoon when the distance error is roughly equal in front and rear, rather than when one of the two is very small and the other large.

For both $\mathcal{S}$ and $\mathcal{L}$, we invoke Algorithm 1. The algorithm takes in input the set to be refined and the position p of the vehicle v. It iterates on the distance k from vehicle v until either the beginning or the end of the platoon is hit or until there is only one or no records left. The latter case ($\mathcal{X} = \emptyset$) can only happen if $\mathcal{X}$ is empty since the beginning. If k hits the end of the platoon and we are left with more than a record in $\mathcal{X}$, then we continue to search records by only

Algorithm 1: Algorithm for progressively refining the set of records in $\mathcal{X}$.

```
procedure searchRecords($\mathcal{X}$, p)
    $k \leftarrow 1$
    while $|\mathcal{X}| > 1$ and $p - k \geq 1$ and $p + k \leq P$ do
        $\mathcal{X} \leftarrow \arg\min_{R \in \mathcal{X}} \Delta_R$
        $k \leftarrow k + 1$
    if $\mathcal{X} = \emptyset$ then return $\emptyset$
    if $|\mathcal{X}| = 1$ then return set $\mathcal{X}$
    if $p - k = 0$ then
        return searchToTail($\mathcal{X}$, p)
    else
        return searchToHead($\mathcal{X}$, p)
```

considering vehicles towards the head of the platoon, invoking the *searchToHead* method. If, instead, k hits the head of the platoon, we invoke *searchToTail* to continue searching towards the end of the platoon. The *searchToHead* and *searchToTail* procedures are defined in Algorithm 2. The procedure are similar to the previous case, but now we only have a distance error that depends on a single distance rather than two and we can use the simple distance and not Δ_R for the selection.

Algorithm 2: Search procedures towards the head and the tail of the platoon, starting from position k.

```
procedure searchToHead($\mathcal{X}$, $k$)
    while $|\mathcal{X}| > 1$ do
        $\mathcal{X} = \arg\min_{R \in \mathcal{X}} |d^R_{k-1} - d_{k-1}|$
        $k \leftarrow k - 1$
    return set $\mathcal{X}$
procedure searchToTail($\mathcal{X}$, $k$)
    while $|\mathcal{X}| > 1$ do
        $\mathcal{X} = \arg\min_{R \in \mathcal{X}} |d^R_{k} - d_{k}|$
        $k \leftarrow k + 1$
    return set $\mathcal{X}$
```

After invoking *searchRecords* on both $\mathcal{S}$ and $\mathcal{L}$ we have three possible outcomes:

- $|\mathcal{S}| = 1$ and $|\mathcal{L}| = 0$: in this case the distance of the vehicle v from its direct neighbors is larger than the largest distance in the dataset. In such case we simply return a C_D/C_{D_∞} value of 1 assuming that the vehicle is too far to experience any reduction in air drag;
- $|\mathcal{S}| = 0$ and $|\mathcal{L}| = 1$: in this case the distance of the vehicle v from its direct neighbors is shorter than the shortest distance in the dataset. In such case we return the C_D/C_{D_∞} value c^L_p of the only record in $\mathcal{L}$, assuming that the influence is the same;
- $|\mathcal{S}| = 1$ and $|\mathcal{L}| = 1$: in this case we have two records and we can perform a linear interpolation of two C_D/C_{D_∞} values as in the following description.

Let S and L be the only records in $\mathcal{S}$ and $\mathcal{L}$; C_D/C_{D_∞} for vehicle v is computed by the following interpolation:

$$c^*_p = \frac{c^S_p(d^L_{p-1} - d_{p-1}) + c^L_p(d_{p-1} - d^S_{p-1})}{d^L_{p-1} - d^S_{p-1}}. \qquad (12)$$

Eq. (12) cannot be used when the vehicle v is the first in the platoon because the distance to the front vehicle is not defined. In such case, to keep the model simple, we consider the distance to the vehicle behind, so the equation becomes

$$c^*_1 = \frac{c^S_1(d^L_1 - d_1) + c^L_1(d_1 - d^S_1)}{d^L_1 - d^S_1}. \qquad (13)$$

Part of our future work includes the validation of the proposed interpolation method, which requires additional computationally intensive CFD simulations to test whether the outcome of Eq. (13) matches the real value within an acceptable error bound.

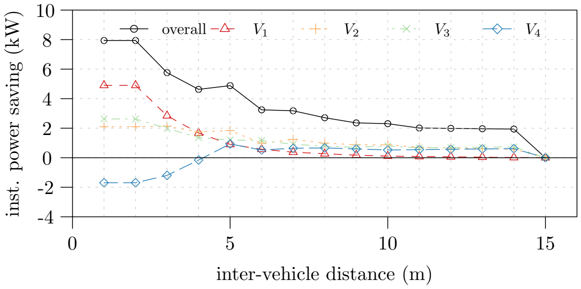


Figure 8: Instantaneous power saving for a platoon of 4 electric cars running at a constant speed of 130 km/h as function of different inter-vehicle distances.

C. Example Results

To showcase the potential of the model, we run a set of simulations using PLEXE [25], which enables the use of CACC algorithms within SUMO. We model a platoon of 4 electric vehicles traveling at a constant speed of 130 km/h at different inter-vehicle distances. The simulation parameters are PLEXE's default ones. We let the simulations reach steady-state with stable inter-vehicle distance and measure the difference in power usage with respect to an isolated vehicle.

It is important to mention is that Ahmed bodies cannot be used to estimate fuel reduction because, being idealized vehicles, they do not have a fuel consumption model. Instead, we use a Tesla Model S P85 with a battery capacity of 85 kW h, a maximum power of 350 kW, a drag coefficient of 0.24 and a cross-section area of 2.34 m^2. We assume that the C_D/C_{D_∞} are the same as those of the Ahmed bodies. We are aware that this is not realistic, but we reiterate that the simulations are meant to prove the feasibility of the method, thus should not be taken as valid from a quantitative point of view. Computing the CFD of a Tesla, or any other real vehicle for what it matters, is computationally too expensive to do without proper support, furthermore it requires having detailed description of the vehicle body to build the mesh, description that is normally not available, thus such advanced steps should be done in cooperation with the car makers.

Fig. 8 shows the results for the four vehicles in terms of instantaneous power saving. Positive values indicate energy being saved. In addition to the per-car power saving, we plot the overall power saving by summing the values of all the vehicles. By observing Fig. 8, we can notice that the model succeeds in computing the impact of slipstreaming even for distances that were not pre-computed (see Fig. 5). This is a key step towards the formulation of a general model. The overall power saving is maximized at the lowest distance, indicating that the vehicles all together require 8 kW less than when driving separately. If we want each of the vehicles to reduce their power consumption, the graph shows that we need to increase the distance to 5 m. With such a distance, the vehicles to save about 1 kW to 2 kW each, leading to an overall power saving of about 5 kW. As reference, the consumption model computes a required power of around 58 kW per vehicle so the power saved is in the order of 2 % to 4 % overall.

V. Conclusions and Future Work

Fuel reduction through the slipstream effect is often mentioned as one of the main drivers to adopt platooning; however, apart from a few small-scale experimental studies, there is no literature that quantifies this effect as a function of the platoon characteristics, nor tools available to the community that empower such studies. The ambition of this paper is to start filling this huge void with some building blocks that open new and interesting research areas, rather than writing "The End" on the problem.

The first one is outside the scope of this paper, but it is worth mentioning. We have worked with Ahmed bodies, which are considered good enough models of real vehicles, but this tells nothing about the possibility of optimizing aerodynamics of vehicles for platooning. Can in the future automotive industries design vehicle bodies that minimize air drag when fully autonomous, self driving vehicles platoon to optimize infrastructure usage and minimize incidents?

A second area of research that is more directly influenced by our work is platoon optimization. CACC algorithms today normally assume identical distance between the platoon vehicles, but it is not difficult to imagine that new algorithms can be designed, or the available one adapted, to control the position of vehicles in a platoon in such a way to minimize the overall fuel consumption, while at the same time computing the compensation incentives that balance the different savings obtained as a function of the position.

A third area of research has already been mentioned, but it is worth recalling. In a fully automated scenario, a Traffic Authority could coordinate formation, re-configuration and, in general, maneuvering of platoons. Optimizing traffic would be a multi-objective function, one of these objectives being definitely reduction of fuel consumption. To achieve this goal, the Traffic Authority needs proper slipstream models, but also tools allowing for fast simulations, as the complexity of the problem rules out the possibility of analytic models.

The lessons learned through this work are multiple: i) It is very difficult to include slipstream effects in dynamic platoon simulations, but it is feasible; ii) Fuel reduction through platooning may not be as large as some people believe, but it occurs in all the platoons we considered; iii) Interdisciplinary work between automotive engineers and computer scientist is still needed to achieve better insights in the problem and produce more effective models; and iv) Last but not least, communications and coordination functions are more important than ever to exploit the socio-economic benefits of autonomous vehicles.

References

[1] "Safe distance between vehicles," TG Road Safety, Conference of European Directors of Roads, Paris, France, Technical report 2009/10.1, Apr. 2010.

[2] J. Heinovski and F. Dressler, "Platoon Formation: Optimized Car to Platoon Assignment Strategies and Protocols," in *10th IEEE Vehicular Networking Conference (VNC 2018)*, Taipei, Taiwan, Dec. 2018.

[3] K. Y. Liang, J. Mårtensson, and K. H. Johansson, "Heavy-Duty Vehicle Platoon Formation for Fuel Efficiency," *IEEE Transactions on Intelligent Transportation Systems (TITS)*, vol. 17, no. 4, pp. 1051–1061, Apr. 2016.

[4] K.-Y. Liang, J. Mårtensson, and K. H. Johansson, "When is it Fuel Efficient for a Heavy Duty Vehicle to Catch Up With a Platoon?" In *7th IFAC Symposium on Advances in Automotive Control*, Tokyo, Japan, Sep. 2013, pp. 738–743.

[5] G. Giordano, M. Segata, F. Blanchini, and R. Lo Cigno, "The joint network/control design of platooning algorithms can enforce guaranteed safety constraints," *Ad Hoc Networks, Elsevier*, vol. 94, Nov. 2019.

[6] S. Santini, A. Salvi, A. S. Valente, A. Pescapè, M. Segata, and R. Lo Cigno, "Platooning Maneuvers in Vehicular Networks: a Distributed and Consensus-Based Approach," *IEEE Transactions on Intelligent Vehicles (T-IV)*, vol. 4, no. 1, pp. 59–72, Mar. 2019.

[7] T. Sturm, C. Krupitzer, M. Segata, and C. Becker, "A Taxonomy of Optimization Factors for Platooning," *IEEE Transactions on Intelligent Transportation Systems (TITS)*, Apr. 2020.

[8] I. Rubin, A. Baiocchi, Y. Sunyoto, and I. Turcanu, "Traffic management and networking for autonomous vehicular highway systems," *Elsevier Ad Hoc Networks*, vol. 83, pp. 125–148, Feb. 2019.

[9] F. Dressler, F. Klingler, M. Segata, and R. Lo Cigno, "Cooperative Driving and the Tactile Internet," *Proceedings of the IEEE*, vol. 107, no. 2, pp. 436–446, Feb. 2019.

[10] Z. Huang, D. Chu, C. Wu, and Y. He, "Path Planning and Cooperative Control for Automated Vehicle Platoon Using Hybrid Automata," *IEEE Transactions on Intelligent Transportation Systems (TITS)*, vol. 20, no. 3, pp. 959–974, Mar. 2019.

[11] W.-H. Hucho, *Aerodynamics of Road Vehicles: From Fluid Mechanics to Vehicle Engineering*. Elsevier, 1987.

[12] M. Zabat, N. Stabile, S. Farascaroli, and F. Browand, "The Aerodynamic Performance Of Platoons: A Final Report," University of California, Berkeley, Berkeley, CA, Research Report UCB-ITS-PRR-95-35, Oct. 1995.

[13] V. Turri, B. Besselink, and K. H. Johansson, "Cooperative Look-Ahead Control for Fuel-Efficient and Safe Heavy-Duty Vehicle Platooning," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 1, pp. 12–28, Jan. 2017.

[14] S. E. Shladover, "PATH at 20 – History and Major Milestones," in *IEEE Intelligent Transportation Systems Conference (ITSC 2006)*, Toronto, Canada, Sep. 2006, pp. 22–29.

[15] T. Robinson, E. Chan, and E. Coelingh, "Operating Platoons On Public Motorways: An Introduction To The SARTRE Platooning Programme," in *17th World Congress on Intelligent Transport Systems (ITS 2010)*, Busan, South Korea, Oct. 2010.

[16] P. S. Jootel, "SAfe Road TRains for the Environment," SARTRE Project, Final Project Report, Oct. 2012.

[17] A. A. Alam, A. Gattami, and K. H. Johansson, "An Experimental Study on the Fuel Reduction Potential of Heavy Duty Vehicle Platooning," in *13th IEEE International Conference on Intelligent Transportation Systems (ITSC 2010)*, Funchal, Portugal, Sep. 2010, pp. 306–311.

[18] C.-H. Bruneau, K. Khadra, and I. Mortazavi, "Flow analysis of square-back simplified vehicles in platoon," *International Journal of Heat and Fluid Flow, Elsevier*, vol. 66, pp. 43–59, Aug. 2017.

[19] S. R. Ahmed, G. Ramm, and G. Faltin, "Some Salient Features of the Time-Averaged Ground Vehicle Wake," in *SAE International Congress and Exposition*, Detroit, MI, Feb. 1984.

[20] R. M. Pagliarella, "On the Aerodynamic Performance of Automotive Vehicle Platoons Featuring Pre and Post-Critical Leading Forms," PhD Thesis, School of Aerospace, Mechanical and Manufacturing Engineering, RMIT University, Melbourne, Australia, Sep. 2009.

[21] H. K. Versteeg and W. Malalasekera, *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*, 2nd ed. Harlow, United Kingdom: Pearson Education, 2007.

[22] E. Hesham and R. Dominy, "Wake and surface pressure analysis of vehicles in platoon," *Journal of Wind Engineering and Industrial Aerodynamics, Elsevier*, vol. 201, Jun. 2020.

[23] P. De Haan and M. Keller, "Modelling fuel consumption and pollutant emissions based on real-world driving patterns: the HBEFA approach," *International Journal of Environment and Pollution, Inderscience*, vol. 22, no. 3, pp. 240–258, 2004.

[24] T. Kurczveil and E. Schnieder, "Implementation of an Energy Model and a Charging Infrastructure in SUMO," in *SUMO User Conference 2013*, Berlin, Germany, May 2013, pp. 88–94.

[25] M. Segata, S. Joerer, B. Bloessl, C. Sommer, F. Dressler, and R. Lo Cigno, "PLEXE: A Platooning Extension for Veins," in *6th IEEE Vehicular Networking Conference (VNC 2014)*, Paderborn, Germany, Dec. 2014, pp. 53–60.