Cooperative navigation for efficient trucks platooning

Alessando Assirelli, *231685*

***Abstract*—This project aims to enhance the efficiency of a trucks fleet operating in a highway. The fleet comprises trucks with diverse fuel consumption rates, and the leading vehicle faces additional fuel consumption due to aerodynamic drag. The improved efficiency is obtained by changing the fleet leader in order to smartly split the effort among all the trucks. The project is structured around four key points. Firstly, proposing the implementation of Distributed Extended Kalman Filkter (DEKF) for precise localization. Secondly, conducting the implementation and analysis of Cooperative Adaptive Cruise Control (CACC) to manage longitudinal dynamics. The third point concerns the implementation an optimization strategy designed to maximize overall fleet efficiency by rotating the leader position among the vehicles. Lastly, an overtaking maneuver scheme is proposed to accomplish the optimal schedule.**

I. Introduction

The transportation sector faces growing pressure to reduce its environmental footprint and improve safety. Autonomous vehicles, equipped with advanced sensors, actuators, and communication systems, offer a promising solution to address these challenges. V2V communication, particularly, enables vehicles to exchange information in real-time, enabling coordinated and efficient driving.

Autonomous vehicles employ a suite of advanced technologies to operate independently. Sensors, such as cameras, radar, and Lidar, provide real-time information about the vehicle's surroundings, including surrounding vehicles, road conditions, and traffic signals. Actuators, including brakes, accelerators, and steering systems, enable the vehicle to maneuver accordingly. V2V communication, allows vehicles to exchange information about their position, speed, intentions, and other relevant data.

Trucks, due to their size and weight, are particularly energy-intensive vehicles. Aerodynamic drag plays a significant role in fuel consumption, and the leading vehicle in a convoy experiences increased drag. Distributing the leading position among multiple trucks can significantly reduce the overall fuel consumption of the fleet. A scheduling mechanism can be implemented to assign the leading position to trucks based on their current fuel consumption profiles, traffic conditions, and remaining driving distance. This approach balances the energy burden across the fleet. This approach of splitting the hard work among different agent is the one used for example by cyclist when racing. The leader makes the ride easier for the other riders by experiencing an increased aerodynamics drag and then, at some point the leader gets changed to let him rest.

V2V also allows to exchange information about a vehicle's measurements which makes it attractive for the implementation of a distributed filter that reconstruct the geometry of the trucks fleet. Such knowledge can be exploited when performing overtaking maneuvers. When a vehicle overtakes another it usually relies only on its measurements, but by fusing them with the ones of the other vehicles a more reliable maneuver can be realized. If the vehicles exchange also informations about their speed intentions it is possible to implement CACC to adapt the speeds of the other vehicles to maintain a safe and efficient following distance. This cooperative control can reduce the frequency of accelerations and decelerations, minimizing slinky effects and further improving fuel efficiency. CACC allows trucks to form tight platoons, A coordinated driving formation in which trucks travel closely together which further enhances fuel efficiency by reducing aerodynamic drag.

A. Description of the problem

- **Distributed Filter**: a Distributed Filter is an algorithm that takes all the measurements of all the agents of the fleet and fuses them together to estimate the fleet state. Because the filter is distributed, there is not a central node where all the informations flows, but every node of the network will communicate with the others a local estimate of the fleet state.
- **Longitudinal Control**: Longitudinal control refers to maintaining a safe and efficient following distance between trucks within a convoy. V2V communication enables trucks to share their speed intentions, allowing them to adjust their speeds seamlessly and maintain a consistent gap. This cooperative speed control minimizes frequent accelerations and decelerations, further reducing fuel consumption and enhancing safety.
- **Optimal Scheduling**: Optimal scheduling consists in assigning the leading position among trucks within a convoy. This approach ensures that the leading position, which typically experiences increased aerodynamic drag, is distributed among trucks efficiently. It also considers each vehicle's specific consumption to balance the load of the fleet.
- **Overtaking Maneuver**: Overtaking maneuvers are essential for implementing the optimal scheduling strategy, allowing trucks to rotate through the leading position efficiently. When the optimal scheduling algorithm determines that a truck should take the lead, an overtaking maneuver gets instatiated.

B. Related Work

A Distributed Extended Kalman Filter is a type of Kalman filter that is designed to work in a distributed sensor network.

In a distributed sensor network, there are multiple sensors that are spread out. Each sensor has its own local measurements, but it does not have access to all of the measurements from the other sensors. The DEKF is able to estimate the state of the system using only the local measurements from each sensor, and by exchanging information with its neighboring sensors. The DEKF algorithm has been extensively studied in literature, with numerous research papers demonstrating its effectiveness in various applications, including autonomous vehicle state estimation.

CACC has been extensively researched in the context of autonomous vehicle platooning, with numerous studies investigating various strategies for ensuring coordinated speed adjustments among vehicles. A possible approach has been proposed by [1] which integrates the ego measurements of the location, with state and control informations coming from the preceding vehicle to obtain a safe longitudinal controller.

In order for a vehicle to overtake another, it has to perform a lane change. Among the possible algorithms, a simple, yet effective approach is the Pure Pursuit Kinematic Controller (PPC) [2]. PPC has been widely adopted in autonomous vehicle steering due to its ability to generate smooth and efficient steering commands. The controller relies on the vehichle's position estimate and calculates the steering angle needed to reach a point through simple kinematics consideration. The distance of the tracked point is ruled by the lookahead distance (LD), that rules the responsiveness of the system.

II. Adopted Models

A. System model

Each vehicle is modeled with a car like kinematics model, whose longitudinal dynamics is ruled by the third order model

$$\begin{pmatrix} \dot{s}_i(t) \\ \dot{v}_i(t) \\ \dot{a}_i(t) \end{pmatrix} = \begin{pmatrix} v_i(t) \\ a_i(t) \\ \frac{-1}{\tau} a_i(t) + \frac{1}{\tau} p_i(t) \end{pmatrix} \tag{1}$$

where a_i is the acceleration of vehicle i, while p_i is the input to be interpreted as desired acceleration and τ is a time constant representing engine dynamics. This model is rather popular in cruise control research [3].

The overall state of each vehicle is thus of dimension $n = 6$

$$X_i(t) = \begin{pmatrix} x_i(t) \\ y_i(t) \\ \delta_i(t) \\ \alpha_i(t) \\ v_i(t) \\ a_i(t) \end{pmatrix} \quad , \quad \dot{X}_i(t) = \begin{pmatrix} cos(\delta_i(t))v_i(t) \\ sin(\delta_i(t))v_i(t) \\ tan(\alpha_i(t))/Lv_i(t) \\ \omega_i(t) \\ a_i(t) \\ \frac{-1}{\tau} a_i(t) + \frac{1}{\tau} p_i(t) \end{pmatrix} \tag{2}$$

where the control actions are the steering velocity $\omega_i(t)$ and the throttle $p_i(t)$, so $u_i(t) = \begin{bmatrix} p_i(t) & \omega_i(t) \end{bmatrix}^T$.

The continuous time system is discretized by means of Euler discretization. The model is also affected by the uncertainty $\eta_{i,k}$, so that the final system can be described by the nonlinear relation

$$X_{i,k+1} = f(X_{i,k}, u_{i,k}, \eta_{i,k}) \tag{3}$$

For a fleet composed by m vehicles, the whole state of the is obtained by stacking each vehicle's state. The order in which the state is stacked match the initial configuration of the vehicles in the fleet. Such state is thus of dimension $n * m$.

$$X_k = \begin{pmatrix} x_{1,k} \\ y_{1,k} \\ \delta_{1,k} \\ \vdots \\ x_{m,k} \\ y_{m,k} \\ \delta_{m,k} \end{pmatrix} \tag{4}$$

B. Communication System

The fleet is made up by intelligent vehicles equipped with sensors and actuators to autonomously interact with the environment (i.e. the highway). Each vehicle is also able to communicate. The communication happens only if two vehicles are within the communication range. The probability of a vehicle to communicate with the others is also bounded by $P_{comm} < 1$, to account for non ideality. The communication is bidirectional. When two vehicles are able to communicate they exchange a series of information:

- $F_{i,k}$: Composite information matrix used in the consensus protocol for DEKF
- $a_{i,k}$: Composite information state used in the consensus protocol for DEKF
- $d_{i,k}$: Degree of the $i - th$ vehicle. used to reach the average consensus
- $p_{i,k}$: Control action of vehicle i, used for the prediction step of the DEKF and for the application of CACC algorithm

C. Measurement model

Each vehicle is equipped with a set of sensors that allows it to estimate not only its state, but also the one of the vehicle in front of it. Every sensor has been calibrated, while the noise is assumed normally distributed.

$$\begin{gathered} s_{meas} = s_{true} + \epsilon \\ \epsilon \sim \mathcal{N}(0, \sigma^2) \end{gathered} \tag{5}$$

The sensor measurements are also independent. The measurement model of vehicle i is definded as:

$$z_{i,k} = h(X_k, \epsilon_k) = \begin{pmatrix} x_{i,k} \\ y_{i,k} \\ \delta_{i,k} \\ \alpha_{i,k} \\ \sqrt{(x_{j,k} - x_{i,k})^2 + (y_{j,k} - y_{i,k})^2} \\ atan(\frac{y_{j,k} - y_{i,k}}{x_{j,k} - y_{i,k}}) - \delta_{i,k} \\ \delta_{j,k} - \delta_{i,k} \end{pmatrix} + \epsilon_k \tag{6}$$

Where the first four rows correspond to the ego measurements of the vehicle i, while the last three are the measurements of the preceding vehicle j, if it is within the measurement range. The first two measurements come from the Radar, while the last one is considered a preprocessed information coming from the Lidar about the rotation of the vehicle in front. Table I shows the uncertainty considered for each sensor.

Sensor	σ
GPS (x)	1e-2
GPS (y)	1e-2
Magnetometer	1e-1
Encoder on steering wheel	1e-2
Radar (ρ)	5 1e-2
Radar (ϕ)	5 1e-2
Preprocessed Lidar	1e-2

TABLE I: Sensors uncertainty

D. Consumption model

In order to describe the additional effort needed for the fleet leader a simple linear model has been used and it is described by the relation:

$$c_i = \begin{cases} (\beta_i + \gamma_i) s_{i,L} & \text{if } i \text{ is leader} \\ \beta_i s_{i,\bar{L}} & \text{if } i \text{ is not leader} \end{cases} \tag{7}$$

Where c is the total consumption in L/km, the term β_i is the regular consumption due to loading conditions, engine specification and aerodynamics drag, while γ_i is the additional effort required by the leader of the fleet. The terms $s_{i,L}$ and $s_{i,\bar{L}}$ are the distances traveled in the first, or in any other position. Clearly the sum $s_{i,L} + s_{i,\bar{L}}$ must be equal to the overall traveling distance S of the vehicle i

III. Solution

A. Schedule Optimization

The problem of finding an optimal schedule for autonomous trucks platoon can be formulated as a Quadratic Program (QP). To exploit this optimization strategy it is assumed that the solution match the template shown in Figure 1.

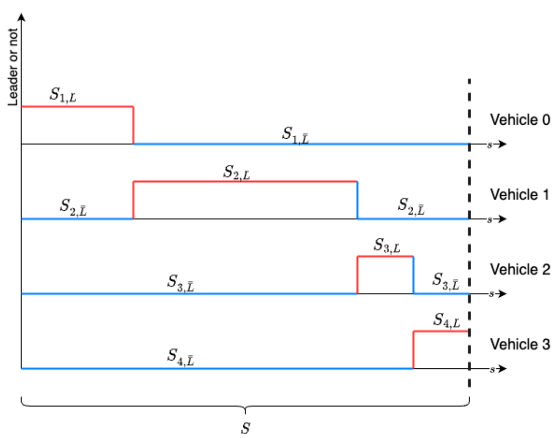


Fig. 1: Template solution for QP

Quadratic programming finds the global optimal solution to the problem

$$\begin{aligned} \min_{Y} \quad & \frac{1}{2} Y O Y^T + cY \\ \text{s.t.} \quad & \\ & AY = b \\ & GY \leq h \end{aligned} \tag{8}$$

The job of the QP solver is to stretch and shrink the graphs in order to obtain the minimum cost. In this case the optimization variables Y are the distances traveled by the vehicles in the first or in any of the other positions. O is instead the consumption matrix, which collects all the terms β_i and γ_i.

$$Y = \begin{bmatrix} s_{1,L} \\ s_{1,\bar{L}} \\ s_{2,L} \\ s_{2,\bar{L}} \\ \vdots \\ s_{m,L} \\ s_{,\bar{L}} \end{bmatrix}_{2m} \tag{9}$$

$$O = \begin{bmatrix} \beta_1 + \gamma_1 & \beta_1 & 0 & 0 & \dots \\ 0 & 0 & \beta_2 + \gamma_2 & \beta_2 & \dots \\ \vdots & \vdots & \ddots & & \\ 0 & 0 & \dots & \beta_m + \gamma_m & \beta_m \end{bmatrix}_{mX2m} \tag{10}$$

The linear constraint $AX = b$ is needed to guarantee that each vehicle travels the distance S. In this application the vectors c, h and the matrix G are note used.

IV. Control Design

A. Distributed Extended Kalman Filter

In order to reconstruct the state of the fleet of dimension $n * m$ DEKF has been implemented. In DEKF the nodes of the network solve two consensus problems that allow them to calculate average information matrix and average information state at every iteration k. Every node of the network i then, can calculate the state estimate $\hat{X}$ at iteration k using the update equations of the filter [4]

$$\begin{aligned} M_i &= (P_{i,k}^{-1} + F_i)^{-1} \\ \hat{X}_{i,k+1}^{-} &= f(X_{i,k}, U_{i,k}, \bar{0}) \\ \hat{X}_{i,k+1} &= \hat{X}_{i,k+1}^{-} + M_i(a_i - F_i \hat{X}_{i,k+1}^{-}) \\ P_{i,k+1} &= A_i M_i A_i^T + G_i Q_i G_i^T \end{aligned} \tag{11}$$

$$A_i = \left. \frac{\delta f(X,U,\eta)}{\delta X} \right|_{X=\hat{X}_{i,k+1}^{-}}^{\eta=0} \qquad G = \left. \frac{\delta f(X,U,\eta)}{\delta \eta} \right|_{X=\hat{X}_{i,k+1}^{-}}^{\eta=0} \tag{12}$$

Where F_i and a_i are respectively the local composite information matrix and state, obtained by computing the average consensus. Q_i is the covariance matrix of the model uncertainty. The consensus protocol is initialized at every iteration k with:

$$\begin{aligned} F_{i,0} &= H_i^T R_i^{-1} H_i \\ a_{i,0} &= H_i^T R_i^{-1} z_{i,k} \end{aligned} \tag{13}$$

Where R_i is the covariance matrix of the sensor noise and H_i is the linearized model measurement. Note that the set of available sensors can change with time, for example if no ranging measurement can be made.

$$H = \left. \frac{\delta h(X,\epsilon)}{\delta X} \right|_{X=\hat{X}_{i,k+1}^{-}}^{\epsilon=0} \tag{14}$$

To reach the average consensus at every iteration k, 5 kk messaging rounds are performed to reach consensus. In doing so the Metropolis Hastings weighting has been used:

$$q_{ij,k} = \begin{cases} \frac{1}{max(d_{i,k},d_{j,k})+1} & \text{if } i \neq j \\ 1 - \sum_{j=1,i\neq j}^{m} q_{ij,k} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

$$F_{i,kk+1} = q_{ii,k}F_{i,kk} + \sum_{j=1,i\neq j}^{m_r} q_{ij,k}F_{j,kk} \tag{15}$$

$$a_{i,kk+1} = q_{ii,k}a_{i,kk} + \sum_{j=1,i\neq j}^{m_r} q_{ij,k}a_{j,kk}$$

m_r is the number of agents reachable by communication

B. Cooperative Adaptive Cruise Control

In order to develop a controller that stabilize the vehicles along the string a linear feedback controller with feedforward term has been implemented.

Consider a string of m vehicles, with d_i being the distance between vehicle i and its preceding vehicle $i-1$, and v_i the velocity of vehicle i. The main objective of each vehicle is to follow its preceding vehicle at a desired distance $d_{r,i}$. Here, a constant time headway spacing policy is adopted, formulated as

$$d_{r,i} = r_i + hv_i(t) \tag{16}$$

where h is time headway and r_i a constant spacing accounting for vehicle dimensions and stanstill safe spacing. Such a policy is known to imporve string stability [5]. The error is thus defined as:

$$e_i(t) = d_i(t) - d_{r,i}(t) = (s_{i-1}(t) - s_i(t)) - r_i - hv_i(t) \tag{17}$$

With $s_i(t)$ the position along the road of vehicle i at time t. The error state can be designed as

$$\begin{pmatrix} \dot{e}_{1,i}(t) \\ \dot{e}_{2,i}(t) \\ \dot{e}_{3,i}(t) \end{pmatrix} = \begin{pmatrix} e_i(t) \\ \dot{e}_i(t) \\ \ddot{e}_i(t) \end{pmatrix} \tag{18}$$

From the error definition, using the third order model it follows that:

$$\dot{e}_{3,i} = -\frac{1}{\tau}e_{3,i} - \frac{1}{\tau}\lambda_i + \frac{1}{\tau}p_{i-1} \tag{19}$$

where λ_i is the new input defined as: $\lambda_i = h\dot{p}_i + p_i$. With the knowledge of p_{i-1}, obtained via wireless communication between vehicles, one could choose the new control input λ_i in such a way that the error converges to zero. A linear feedback has been implemented:

$$\lambda_i = K \begin{pmatrix} \dot{e}_{1,i}(t) \\ \dot{e}_{2,i}(t) \\ \dot{e}_{3,i}(t) \end{pmatrix} + p_{i-1} \tag{20}$$

with $K = \begin{pmatrix} k_p & k_d & k_{dd} \end{pmatrix}$

By choosing the control variable p_i as

$$\dot{p}_i = \frac{1}{h}(p_i + k_p e_{1,i} + k_d e_{2,i} + k_{dd} e_{3,i} + p_{i-1}) \tag{21}$$

the error converges to zero

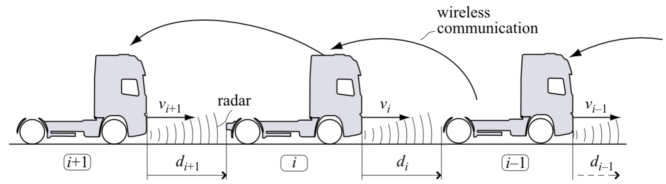


Fig. 2: Conceptual scheme for CACC

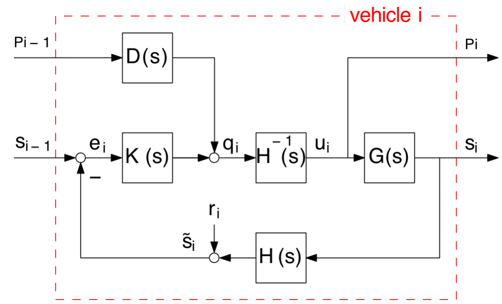


Fig. 3: Block Diagram

C. String stability analysis

String stability in platoon control is a crucial property that determines the overall safety and performance of autonomous vehicle platoons. It refers to the ability of a string of vehicles (followers) to maintain a stable and bounded spacing with respect to a lead vehicle even in presence of disturbances on the leading vehicle. The controller stated before, while guaranteeing convergence of the error, does not proove string stability. To do so it is possible to consider $\Gamma_i(s)$ as the Transfer function from v_{i-1} to v_i.

To proove string stability one needs to verify that

$$||\Gamma(j\omega)||_\infty \leq 1 \tag{22}$$

which is conceptually the same as requiring energy dissipation along the queue of vehicles. The Transfer function of the system can be obtained from the block scheme of Figure 3. Here each block is described:

$$G(s) = \frac{s_i(s)}{p_i(s)} = \frac{1}{s^2(\tau s+1)}$$
$$H(s) = \frac{\lambda_i(s)}{p_i(s)} = hs + 1 \tag{23}$$
$$K(s) = \frac{\lambda_i(s)}{e_i(s)} = k_p + k_d s + k_{dd} s^2$$

The scheme of Figure 3 considers also a time delay through the block $D(s)$, which can be set to 1 if no delay is considered. Another possibility is to set $D(s)$ to zero to obtain the standard adaptive cruise control, without knowledge of the action of the preceding vehicle.

$$\Gamma(s) = \frac{1}{H(s)} + \frac{D(s) + G(s)K(s)}{1 + G(s)K(s)} \tag{24}$$

The communication delay plays an important role. For $D(s) = 1$ the system is stable by definition, being $||\Gamma(j\omega)||_\infty = \sup_\omega |H^{-1}(s)| = 1$ By removing the information about the forward vehicle's action, such guarantee does not hold. Figure 4 shows a comparison of $|\Gamma(j\omega)|$, when $D(s)$

is set to one and when it is set to zero. In order to exagerate the phenomenon the following parameters has been chosen: $\tau = 0.1s$, $k_p = 0.2$, $k_d = 0.7$, $k_{dd} = 0$, $h = 0.1$

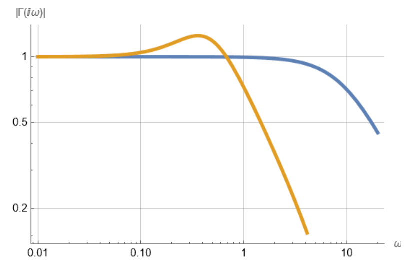


Fig. 4: Frequency response of vehicle i, in presence and in absence of feedforward term p_{i-1}

As can be seen, when using the feed forward term $|\Gamma(j\omega)|$ never exceeds 1, while this doesn't hold for the second case. It is also important to note that for larger values of h this phenomenon is reduced.

D. Pure Pursuit Kinematic Controller

The algorithm works by selecting a target point (TP) on the vehicle's path that is a fixed distance ahead of the vehicle's current position. The vehicle then steers towards the TP. To calculate the steering angle it relies on simple kinematics considerations. In Figure 5 the PPC reference quantities are shown. Since the distance from ICR to TP is the same as the one from ICR to the robot center, it follows that $\gamma_2 = \gamma_3$, while $\gamma_2 = 90 - \alpha$, then by the law of sines

$$\frac{LD}{sin(\gamma_1)} = \frac{R}{sin(\gamma_2)} \tag{25}$$

and since the steering angle α is linked to the steering radius R by the relation $\alpha = atan(L/R)$, the desired steering angle can be easily obtained

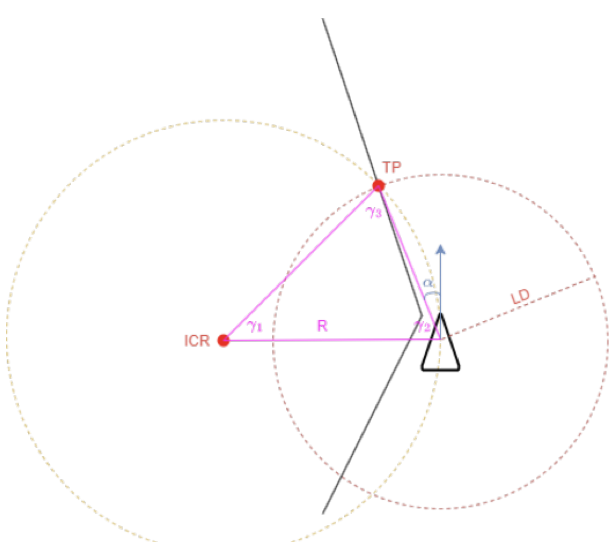


Fig. 5: Pure Pursuit Controller reference diagram

V. Implementation Details

The whole project has been carried out in simulation. The simulator runs in python and the following details are considered to be important for the understanding of the results.

A. Sensors Readings

As explained in section II-C each vehicle measures the preceding one with a Radar and a Lidar. The assumption of measuring just the one in front has been made just to consider the line of sight of the vehicle, but in principle there would be no problem in measuring the others. Indeed if a vehicle is overtaking, thus occupying the other lane, it will be considered to measure every vehicle within the Lidar range.

B. Reference Path

With regards to the lateral controller, implemented via PPC, it is important to point out that each vehicle has an associated path. That path is the one feeded to the PPC to calculate the desired steering. The initial reference path is given at the beginning of the simulation and it corresponds to the geometry of the street. When a vehicle needs to perform a lane change, the path is shifted by an amount equal to the street width on the on the local y axis of the vehicle.

C. PID Controllers

As discussed, PPC provides reference steering angle, but the kinematics model needs steering velocity. A PID controller is thus used to find the steering velocity that tracks the reference steering angle. The same happens for the longitudinal velocity of the leading vehicle. The control action for longitudinal dynamics is p, not v, this requires some sort of map from the desired v to the control action p, to accoplish that another PID controller has been applied

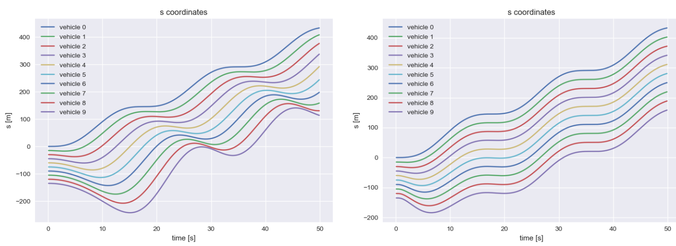
VI. Results

A. CACC

Figure 6 illustrates a simulation with ten vehicles, each with a following distance r_i of 30 meters and time headaway h_i of 0.1 seconds. The vehicles travel on a road aligned with the world x axis. The control strategy of the leading vehicles is designed to oscillate at a frequency close to the platoon resonance frequency, resulting in a sinusoidal control input $p_0 = sin(0.36t)$.

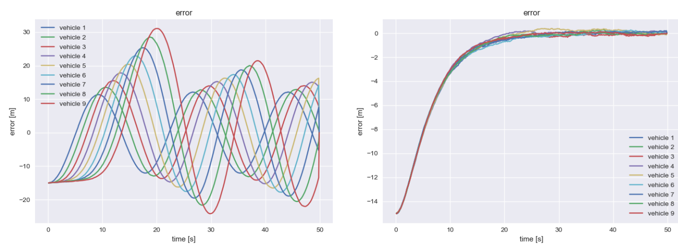
Figure 7a depicts the vehicle-to-vehicle distance error in the conventional ACC system. As observed, the error exhibits significant oscillations, indicating instability and susceptibility to slinky effects. This behavior poses a safety risk, as it could lead to collisions among vehicles in the rearer positions. In contrast, Figure 7b reveals a more stable and consistent vehicle-to-vehicle distance error in the CACC system. This improvement can be attributed to the incorporation of the feedforward term p_{i-1}, which effectively mitigates the slinky effects and ensures smooth and coordinated vehicle operation.

Notably, the control law in Equation (20) does not directly utilize the ranging measurements of the preceding vehicle,



(a) s in ACC (b) s in CACC

Fig. 6: Comparison between ACC and CACC



(a) Error in ACC (b) Error in CACC

Fig. 7: Comparison between ACC and CACC errors

instead, it relies on the estimated values obtained from the DEKF algorithm. This approach enhances the accuracy of the measurements and contributes to the overall stability of the platoon. Figure 8 presents the error distribution of the estimate available to vehicle 1 about vehicle 0, with respect to the x coordinate. Despite the nonlinear nature of the platoon dynamics, the DEKF algorithm effectively estimates the vehicle position, as evident from the normal distribution of the error around zero. This indicates that the filter provides reliable information for the CACC control algorithm. In summary, the

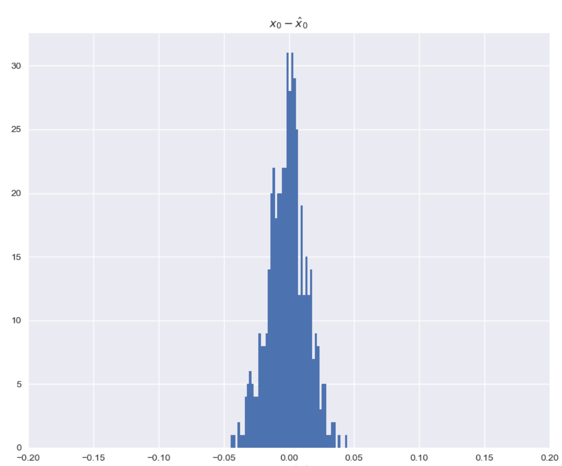
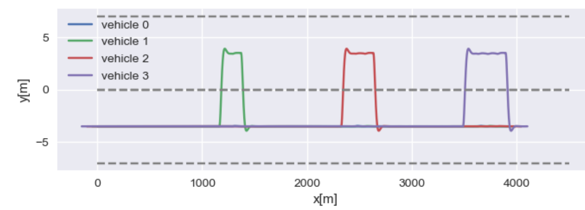


Fig. 8: Error distribution of estimate available to vehicle 1

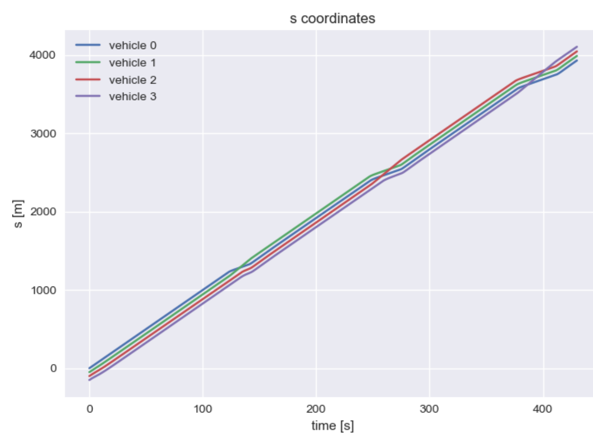
simulation results demonstrate the effectiveness of the CACC system in mitigating slinky effects and maintaining a stable and coordinated platoon formation. The incorporation of the feedforward term and the utilization of the DEKF algorithm for vehicle position estimation play a crucial role in achieving these improvements.

B. *Vehicles estimates*

Figure 9 shows a the simulation of four vehicles traveling on a 4.5 Km road. The vehicles start with the same autonomy level and have the same fuel consumption.



(a) Path drawn by vehicles



(b) Positions along the road

Fig. 9: Simulation of four vehicles traversing a straight road

Initially, vehicle 0 leads the fleet, and the leadership position is gradually distributed among the other vehicles. As vehicle 1 initiates an overtaking maneuver at around 120 seconds, it switches lanes and begins to evaluate the location estimates of the other vehicles. If the uncertainty in the predicted position of another vehicle falls below a certain threshold, the vehicle compares that estimate to its own. Once the vehicle's own position surpasses the estimated positions of the other vehicles plus a safety margin, it initiates a lane change to become the leader.

in doing so the vehicle relies not only on its own sensor measurements but on the estimate obtained by fusing the informations of all the reachable sensors. This implies that the uncertainty on the other's location is smaller than the one possible by using just own measurements making the overtaking safer

Figure 10 shows the estimates of vehicle 1, 2 and 3 available to vehicle 0. Notably, vehicle 0 never directly follows vehicle 2, so it cannot directly measure its position. Vehicle 2's information reaches vehicle 0 via communication. This

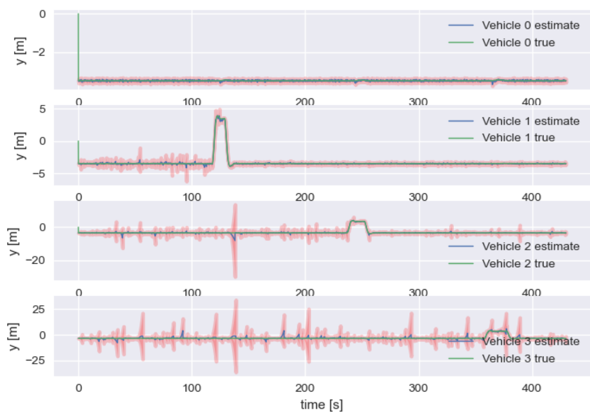


Fig. 10: Vehicle 0's estimates

explains the spikes in the uncertainty of these estimates. As the vehicles are within communication range, the probability of exchanging messages is ruled by P_{comm}, set to 0.3 in this simulation. This means that for most of the time, vehicle 0 relies on predictions for vehicle 2's position, causing the uncertainty to increase. When communication is re-established, the uncertainty decreases. This explains the pattern of the graph.

The estimates of vehicle 1 are more stable. When vehicle 1 overtakes vehicle 0, it can be directly measured by vehicle 0, causing the uncertainty to decrease. As can be seen in Figure 11 the uncertainty decreases after the overtake. However, there

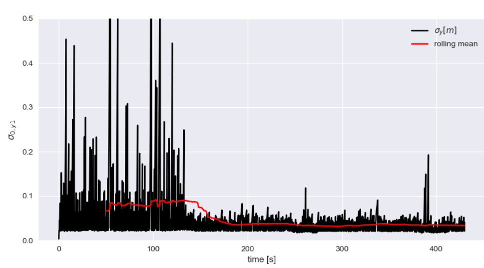


Fig. 11: Uncertainty in vehicle 1 y estimate from vehicle 0

are still spikes in the estimates of vehicle 1, which can be attributed to the increased confidence in the estimates when the two vehicles communicate, as the DEKF algorithm fuses data from multiple sensors.

Because the vehicles are able to communicate, when a vehicle starts the overtaking maneuver, it informs the other vehicles, so they can reduce their cruise velocity, allowing for a faster maneuver. This is an important feature in real life scenarios, where long overtaking maneuver and strong accelerations are both undesired, especially when the platoon is particularly long.

C. Consumption

To validate the efficacy of the proposed scheme in optimizing fuel consumption, a simulated scenario was devised, encompassing four autonomous trucks traversing a 500-

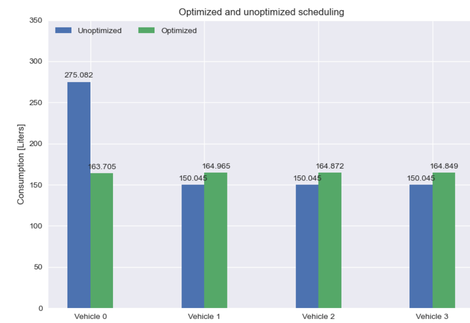


Fig. 12: Comparison of fuel consumption with and without rotating the leader

kilometer route, The vehicles have different fuel consumption and experience different aerodynamics drag increase when leading. The values of β and γ can be chosen randomly, but for a clearer description in this simulation they are set to γ_0 = 0.2 L/km, while γ of all the other vehilces is set to 0.1, β is instead 0.3 for all the vehicles. The simulation outcomes are shown in Figure 12 and clearly show the benefit of a strategy that rotates the leader of the fleet in reducing the fuel consumption. The reduction for this route is of about 10%. Another advantage which was not directly addressed, but results from this simulation, is the similar consumption of every vehicle. Since in this simulation they all start with the same range, this is a desired characteristic as it reduces the stops that the fleet needs to perform to maintain a fixed number of trucks. This behaviour can be prioritized by adding a cost term that minimizes the differences in the autonomy levels in the cost function of Equation (8).

VII. Conclusion

The proposed scheme for autonomous truck platoons has demonstrated its effectiveness in improving fuel efficiency and reducing emissions. This result was expected, as there is an obvious advantage in splitting the job among all the trucks. The focus of this work regards how such result is achieved. By utilizing a Distributed Extended Kalman Filter to estimate the state of neighboring vehicles each vehicle is able to reconstruct the geometry of the fleet, which results in safer operations when a vehicle needs to overtake. The use of CACC dramatically increase the safety as it eliminates the slinky effects that may result in the collision of the last vehicles of the fleet. CACC might also improves fuel efficiency and road throughput by allowing a shorter spacing between the vehicles that is possible because no slinky effect is present. In order to rotate the leader of the fleet the vehicles need to perform overtaking maneuver, so a lateral controller has been implemented with PPC. Overall this project wants to show how a fleet of vehicles could operate to reduce fuel consumption.

However, the proposed scheme has some limitations that need to be addressed in future work. The current consumption model is too simple and does not fully capture the complexities of fuel consumption, such as the impact of velocity, acceleration, and nonlinear effects such as the spacing between vehicles. Additionally, the scheme assumes a static fleet size, which may not reflect real-world scenarios where vehicles may join or leave the platoon. Furthermore, the lateral controller based on pure pursuit with path shifting is not the most optimal solution for lane changes. More advanced path planning techniques could be employed to generate smoother and safer paths during lane transitions. Further research is needed to refine the consumption model, handle dynamic fleet sizes, and optimize lane change maneuvers.

REFERENCES

[1] J. Ploeg, B. T. Scheepers, E. Van Nunen, N. Van de Wouw, and H. Nijmeijer, "Design and experimental evaluation of cooperative adaptive cruise control," in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2011, pp. 260–265.

[2] R. C. Coulter *et al.*, *Implementation of the pure pursuit path tracking algorithm*. Carnegie Mellon University, The Robotics Institute, 1992.

[3] O. Gehring and H. Fritz, "Practical results of a longitudinal control concept for truck platooning with vehicle to vehicle communication," in *Proceedings of Conference on Intelligent Transportation Systems*, 1997, pp. 117–122.

[4] R. Olfati-Saber, "Distributed kalman filter with embedded consensus filters," in *Proceedings of the 44th IEEE Conference on Decision and Control*, 2005, pp. 8179–8184.

[5] G. J. Naus, R. P. Vugts, J. Ploeg, M. J. van De Molengraft, and M. Steinbuch, "String-stable cacc design and experimental validation: A frequency-domain approach," *IEEE Transactions on vehicular technology*, vol. 59, no. 9, pp. 4268–4279, 2010.