# D3.3 Distributed SPL Algorithms for node ranking and topology inference-selection

| | |
|---|---|
| *Authors* | ISI, RTC, UPAT, PASEU |
| *Work Package* | WP3 - Model based CP(H)S Layer Design and Development supporting Distributed Assisted, Augmented and Autonomous Intelligence |

## Abstract

The overall aim of this deliverable is to i) define a utility measure for the nodes of the network, ii) develop distributed algorithms able to infer information about network topology, iii) limit the impact of selfish nodes and iv) formulate network wide objectives. To address these challenges, we have developed novel distributed Signal Processing and Learning algorithms in order to increase global and local location awareness as well as platooning ability in the connected and automated vehicles of the traffic network. To do so, network and local wide objectives have been formulated, and achieved by Kalman filtering, least-squares minimization, information diffusion on graphs and convex optimization approaches. Cramer-Rao Lower Bound has also been employed in order to quantify the usefulness of each involved node. Additionally, a simulation framework has been developed, so as to simulate realistic vehicle-to-vehicle communication conditions.

Funded by the Horizon 2020 Framework Programme of the European Union

| | |
|---|---|
| Deliverable Information | |
| *Work Package* | WP3- Model based CP(H)S Layer Design and Development supporting Distributed Assisted, Augmented and Autonomous Intelligence |
| *Task* | T3.3. Distributed and Coalitional AI supporting autonomic intelligence |
| *Deliverable title* | Distributed SPL Algorithms for node ranking and topology inference-selection |
| *Dissemination Level* | PU |
| *Status* | Ongoing |
| *Version Number* | 0.3 |
| *Due date* | M28 |
| Project Information | |
| *Project start and duration* | 01/01/2020 – 31/12/2022, 36 months |
| *Project Coordinator* | |
| | Industrial Systems Institute, ATHENA Research and Innovation Center 26504, Rio-Patras, Greece |
| *Partners* | 1. ATHINA-EREVNITIKO KENTRO KAINOTOMIAS STIS TECHNOLOGIES TIS PLIROFORIAS, TON EPIKOINONION KAI TIS GNOSIS (ISI) the Coordinator<br>2. FUNDACIO PRIVADA I2CAT, INTERNET I INNOVACIO DIGITAL A CATALUNYA (I2CAT)<br>3. IBM ISRAEL - SCIENCE AND TECHNOLOGY LTD (IBM ISRAEL)<br>4. ATOS SPAIN SA (ATOS)<br>5. PANASONIC AUTOMOTIVE SYSTEMS EUROPE GMBH (PASEU)<br>6. EIGHT BELLS LTD (8BELLS)<br>7. UNIVERSITA DELLA SVIZZERA ITALIANA (USI),<br>8. TAMPEREEN KORKEAKOULUSAATIO SR (TAU)<br>9. UNIVERSITY OF PELOPONNESE (UoP)<br>10. CATALINK LIMITED (CATALINK)<br>11. ROBOTEC.AI SPOLKA Z OGRANICZONA ODPOWIEDZIALNOSCIA (RTC)<br>12. CENTRO RICERCHE FIAT SCPA (CRF)<br>13. PANEPISTIMIO PATRON (UPAT) |
| *Website* | www.cpsosaware.eu |

## Control Sheet

| VERSION | DATE | SUMMARY OF CHANGES | AUTHOR |
|---|---|---|---|
| *0.1* | *15/03* | *Initial draft* | *ISI* |
| *0.2* | *01/04* | *Pre-final draft* | *ISI, RTC, UPAT, PASEU* |
| *0.3* | *19/04* | *Finalization* | *ISI* |

|  | NAME |
|---|---|
| Prepared by | ISI |
| Reviewed by | UPAT, PASEU |
| Authorized by | ISI |

| DATE | RECIPIENT |
|---|---|
| *28/04* | Coordinator |
| 30/04 | European Commission |

## Table of Contents

## List of Figures

## List of Algorithms

## List of Tables

## Executive Summary

The goal of this deliverable is to:

- Define a utility measure for the nodes of the network
- Develop distributed algorithms able to infer information about network topology
- Limit the impact of selfish nodes
- Formulate network wide objectives

All these challenges are discussed and addressed in the following Sections of the deliverable. More specifically:

- Section 2 introduces the concept of Cooperative Awareness through the development of a communication simulation framework.
- In Section 3 three distributed localization algorithms are introduced which focus on realizing the so called local cooperative awareness of each vehicle of the traffic network. Kalman filtering and linear-least squares have been employed to derive and formulate the relevant localization algorithms, in which each vehicle relies only to its direct neighborhood in order to estimate self and neighboring locations. To quantify the quality of location estimation as well as the impact of neighboring nodes in algorithms' performances, Cramer-Rao Lower Bound has been utilized.
- In Section 4, three additional distributed localization algorithms are formulated, based on information diffusion in graphs. More specifically, vehicles are using these algorithms in order to achieve the so-called global awareness, in which a network wide objective is addressed in a distributed manner.
- Section 5 introduces a novel cooperative control scheme for platooning of vehicles using graph Laplacian and Alternating Direction Method of Multiplies (ADMM) algorithm.

# 1 Introduction

The main outcomes of this deliverable related to local and global awareness, cooperative control for platooning, as well as vehicular communication simulation, have been identified below:

- Section 2 realizes Cooperative Awareness through a communication simulation framework.
- Section 3 presents three distributed localization algorithms related to local awareness.
- Section 4 presents three distributed localization algorithms based on information diffusion on graphs, so as to address global awareness task.
- Section 5 introduces a novel cooperative control scheme for platooning of vehicles.

The deliverable is placed within the context of CPSoSAware in the following ways: The simulation framework of Section 5 will serve as the baseline in order to evaluate the proposed methos with realistic V2V conditions in the testing setups of the project. Task 6.3. Additionally, the scene analysis and understanding modules of Task 3.1 will be integrated to the simulation framework developed in this deliverable and T4.4, in order to have a complete automotive awareness system comprised of perception and localization layers. The concepts originally developed in this deliverable will contribute towards the design of a more robust backend vehicle odometry module, during the activities of Tasks 4.5 and 3.6. As a last remark, the proposed schemes will be deployed in hardware as part of Task 5.2.

# 2 Realizing Cooperative Awareness through the integration of communication simulator

Cooperative awareness based on vehicle-to-vehicle (V2V) communication, apart from improving ego vehicle's location estimation, can improve safety between vehicles. According to the National Highway Traffic Safety Administration study on the effectiveness of V2V communication [1], V2V safety technology can help drivers avoid or reduce the severity of over 80% of unimpaired vehicle crashes. It also has a great potential to increase safety in autonomous vehicles operation. Passing messages of different items (e.g., other vehicles, pedestrians, roadblocks, etc.) visible to other vehicles can reveal some obstacles that may be invisible or hidden and thus become a reason for the vehicle to slow down or even stop. Such messages can be used in other cases, such as rearranging paths to omit obstacles, or even to save fuel (especially when it comes to electric-powered vehicles, which have a limited range and are sensitive to sudden changes in speed). Moreover, unlike visual and radar-based detection systems, the application using V2V technology can function regardless of lighting and weather conditions, allowing for a more robust system. These are the reasons that Cooperative Awareness, which delivers the position of surrounding vehicles in real-time, is of special interest to build safety applications such as anticipated collision avoidance or cooperative maneuvering.

For testing the effectiveness of Cooperative Awareness based on V2V communication, two separate simulators will be used. Robotec Simulator manages all traffic in the scene and propagates ROS2 messages between the vehicles (agents). These messages contain information about each vehicles' self-aware localization and objects perceived by this vehicle. Communication simulator, based on one of several wave propagation models, simulates message transfer between agents. Robotec Simulator receives ROS2 messages from the Communication simulator, so the information about the objects that each vehicle is aware of is known. Depending on the model used, there can be various visibility. This information is then used in Robotec Simulator for determining vehicle path or speed as efficiently as possible.

Based on the NHTSA research report on V2V readiness [1] and V2V safety-related literature review [2], [3], a few possible scenarios covering the majority of addressed safety applications were proposed. All the proposed scenarios can be executed in simulation with various lighting and weather conditions, as well as challenging conditions resulting from high-speed driving or interfered communications, among others. With the technological advancement of V2V communication sơystems, more complex scenarios can be implemented to explore the achievable performance of Local Cooperative Awareness under realistic transmit power and transmit rate constraints.

## 2.1 Scenario 1 – intersection collision warning

There are two vehicles (1 and 2) and a pedestrian (3).

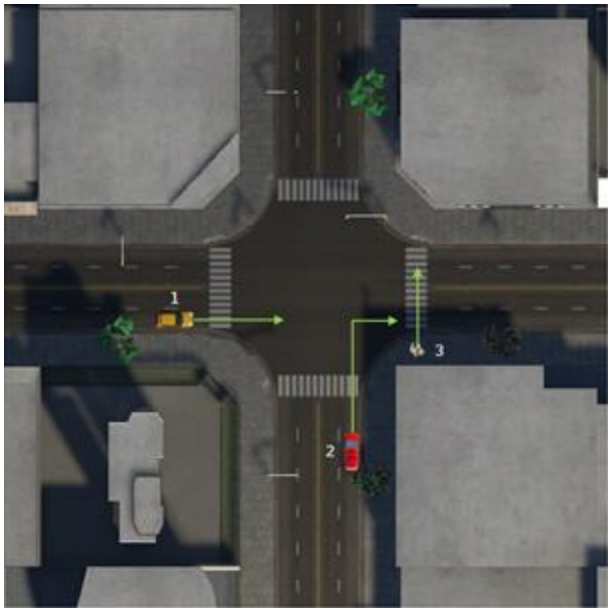


*Figure 1: Cooperative Awareness scenario 1 design*

Vehicle 1 is driving straight through the crossroad, pedestrian 3 is on the pedestrian crossing and vehicle 2 is turning right. Vehicle 2 is aware of vehicle 1 and can turn right before it. Because of the building corner, vehicle 2 is not aware of pedestrian 3.

Without Cooperative Awareness vehicle 2 would turn to cause a potentially hazardous traffic situation with pedestrian 3.

With Cooperative Awareness vehicle 1 distributes messages with objects it sees - sends localization of the pedestrian 3 to vehicle 2. After receiving a message about pedestrian 3, vehicle 2 may adjust its speed or stop to avoid a possibly hazardous event.

## 2.2 Scenario 2 – intersection collision warning

There is a truck (1), a car (2), and a pedestrian (3).

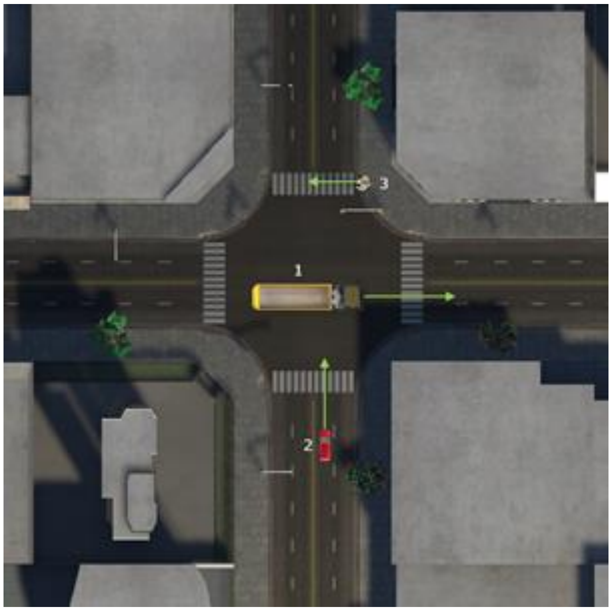


*Figure 2: Cooperative Awareness scenario 2 design*

Truck 1 is driving across the crossroad, pedestrian 3 is on the pedestrian crossing and car 2 is driving straight through the crossroad. Car 2 is aware of truck 1 and will drive on the crossroad after truck 1 leaves it. Because of the sizes of truck 1, car 2 vision is limited, and it is not aware of pedestrian 3.

Without Local Cooperative Awareness car 2 would drive through the crossroad and had to perform an emergency maneuver (evasion or braking) or it would cause a potentially hazardous traffic situation with pedestrian 3.

With Local Cooperative Awareness truck 1 distributes messages with objects it sees - sends localization of the pedestrian 3 to the car 2. After receiving a message about pedestrian 3, car 2 may perform the actions necessary to cross the road safely.

## 2.3 Scenario 3 – intersection collision warning

There are three vehicles (1, 2 and 3).

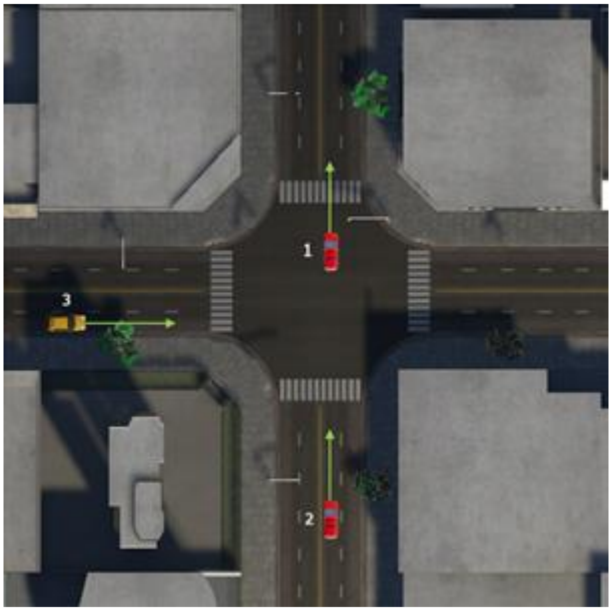


*Figure 3: Cooperative Awareness scenario 3 design*

Vehicle 1 is driving through the crossroad, vehicle 2 is driving straight to the crossroad and vehicle 3 is driving straight to the crossroad. Because of the building corner, vehicles 2 and 3 are not aware of each other.

Without Local Cooperative Awareness vehicle 3 would cause a potentially hazardous traffic situation with vehicle 2.

With Local Cooperative Awareness vehicle 1 distributes messages with objects it perceives – sends localization of vehicle 2 to vehicle 3 and vice versa. After receiving those messages vehicle 2 and vehicle 3 are aware of each other, and thus can adjust their actions to drive through the crossroad safely.

## 2.4 Scenario 4 – forward collision warning/emergency braking

There is a string-like formation of vehicles (e.g., platoon, number of vehicles yet to be determined). The vehicle at the head of the line emergency brakes and all the following vehicles have to automatically react in time to avoid rear-end collisions.

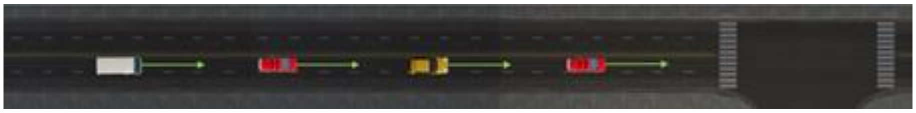


*Figure 4: Cooperative Awareness scenario 4 design*

Without Local Cooperative Awareness, the chances of rear-end collision are higher, due to the response time chain reaction.

With Local Cooperative Awareness, the reaction time can be significantly decreased since the leading vehicle can explicitly inform other platooning members about the critical braking maneuver.

Similar action will be desired in case the leading vehicle faces technical or environmental problems, e.g., drives in on a slippery surface (lack of adhesion) or experiences a wheels-related incident (like a broken tire). In case of a problem detection or a sudden trajectory change, the message should be sent to the following vehicles, so they can adjust their actions to the possible hazard.

## 2.5 Scenario 5 – overtaking/do not pass warning

Two vehicles (1 and 2) move in opposite directions approaching each other, and vehicle 3 limits their visibility.

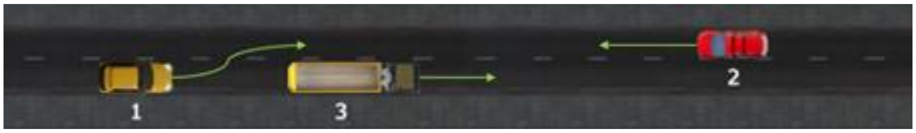


*Figure 5: Cooperative Awareness scenario 5 design*

Without Local Cooperative Awareness vehicle 1 overtaking vehicle 3 would cause a possibly hazardous traffic situation (e.g., driving directly onto a head-on collision with vehicle 2).

Local Cooperative Awareness has the potential to reduce crashes that are not easily addressed by the limited detection range and line of sight capabilities of radar or camera-based systems. V2V communications afford vehicles a rich set of information (e.g., position and trajectory) regarding the other vehicles on the road over a long distance.

## 2.6 Scenario 6 – obstacle avoidance/evasive lane change

Two vehicles (1 and 2) traveling on the right lane of a two-lane road, one vehicle (3) is moving on the left lane. An obstacle appears on the right side of the road.

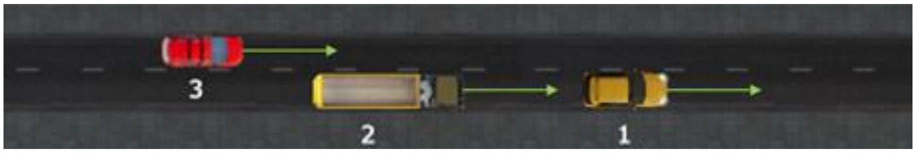


*Figure 6: Cooperative Awareness scenario 6 design*

This is a scenario in which vehicle 1 may react in different ways. Two correct responses can be expected:

- **vehicle 1 evades the obstacle by changing the lane to the left one**

Without Local Cooperative Awareness vehicle 3 may cause a potentially hazardous traffic situation with vehicle 1 as a result of not seeing him (vehicle 1 visibility is limited by vehicle 3).

Using Local Cooperative Awareness communication, vehicle 3 may receive an early warning of an expected emergency maneuver of vehicle 1 and adjust its speed to let vehicle 1 safely overtake the obstacle by changing the lane. It also results in more fluent traffic movement that would positively impact fuel consumption.

- **vehicle 1 applies emergency braking**

Without Local Cooperative Awareness an emergency braking of vehicle 1 may lead to a potentially hazardous traffic situation (e.g., vehicles 1 and 2 rear-end collision).

Using Local Cooperative Awareness communication, vehicle 2 may receive an early warning of an expected emergency maneuver of vehicle 1 and adjust its speed or route (brake immediately or change the lane, if then it is not on a collision course with vehicle 2).

## 2.7 Evaluation

To evaluate Cooperative Awareness communication, the following evaluation metrics will be used:

- time since the event – elapsed time between the event and the time the vehicle first received information about its occurrence,
- speed over time graphs – data visualization will help evaluate speed profile, and potential safe stops will be easily visible. The benefit of such path planning is lower fuel consumption (especially important for electric cars).

Both metrics will be provided with and without the usage of Cooperative Awareness communication and then compared to evaluate Cooperative Awareness performance.

## 2.8 Demo video repository

An indicative demo video demonstrating discussed communication simulator in a Cooperative Awareness scenario is shown in:
https://drive.google.com/file/d/1bKM8LgxO-ZZdVAJ5Z_xJ6yJPmQAbN6fE/view?usp=sharing

# 3 Local Cooperative Awareness via Graph Laplacian Processing[1]

Cooperative Localization and Tracking is a fundamental task for critical automotive applications, which coincides with the so-called Cooperative Situational Awareness. Vehicles have to know their exact locations in order to plan future driving actions, improve their safety, etc. At the same time, the challenging task of situational Cooperative Awareness, i.e., accurate knowledge of neighboring vehicles' locations, is fundamental for improving autonomous driving performance in diverse traffic conditions. Integrated advanced sensors like LIDAR, Cameras and GPS, as well as 5G/V2X communication abilities, enable the close collaboration between the moving vehicles for significantly enhancing the positioning accuracy via multi-modal fusion. In this deliverable, two novel and distributed Cooperative Localization and Tracking algorithms have been formulated, based on least-squares minimization and the celebrated Extended Kalman Filter. They both aim to improve ego vehicle's location estimation, as well as to estimate the position of its neighbors. For that purpose, ego vehicle forms a star like topology with its neighbors, and fuses three types of inter-vehicular measurements via the linear Graph Laplacian operator. An extensive experimental study has been conducted in CARLA simulator, highlighting proposed methods' benefits. The proposed distributed approaches offer high positioning accuracy, outperforming other state-of-the art methods. The key idea of this Section is Local Cooperative Awareness, i.e., distributed multimodal localization able to infer knowledge about ego vehicle and its neighboring vehicles states, without increased communication burden.

## 3.1 Preliminary approaches

Consider a 2-D region where $N$ sensor rich vehicles of a vehicular ad hoc network (VANET) (shown in Figure 7) at an urban environment collect and exchange measurements using V2V communications.

[1] N. Piperigkos, A. S. Lalos and K. Berberidis, "Multi-modal cooperative awareness of connected and automated vehicles in smart cities," *2021 IEEE International Conference on Smart Internet of Things (SmartIoT)*, 2021, pp. 377-382, doi: 10.1109/SmartIoT52359.2021.00070.

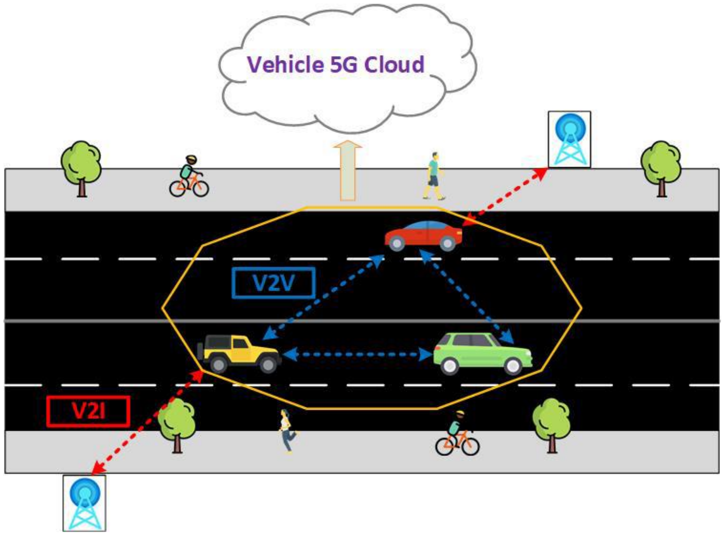


*Figure 7: Vehicular ad hoc network*

Vehicles communicate with the other vehicles of VANET only within a direct communication range $r_c$, according to established V2V communication standards [4]. That means that vehicles receive messages from other vehicles if only their distance is below than $r_c$. Furthermore, we set a maximum number of closest neighbors $N_{max}$ for each vehicle, as in [5]. Thus, a reduced computational load with permissible localization accuracy can be achieved, avoiding also extreme cases like two vehicles of the same VANET are far (even kilometers) away from each other. Vehicles can utilize a multiple access like communication protocol. As it will be shown in Section 3.6, attained localization accuracy is very promising without the need of all-to-all vehicles connections. Hence, scalability is another important aspect of the proposed approaches.

The absolute location of $i$-th vehicle at time instant $t$ is equal to $\boldsymbol{p}_{\boldsymbol{i}}^{(\boldsymbol{t})} = \left[x_i^{(t)} \quad y_i^{(t)}\right]^T \in \mathbb{R}^2$, whereas the distance, angle and azimuth angle between connected vehicles $i$ and $l$ are equal to $z_{d,il}^{(t)} = \left\|\boldsymbol{p}_{\boldsymbol{i}}^{(\boldsymbol{t})} - \boldsymbol{p}_{\boldsymbol{l}}^{(\boldsymbol{t})}\right\|$, $z_{a,il}^{(t)} = \arctan\frac{y_l^{(t)}-y_i^{(t)}}{x_l^{(t)}-x_i^{(t)}}$ and $z_{az,il}^{(t)} = \begin{cases} \epsilon\pi + \arctan\frac{|x_l^{(t)}-x_i^{(t)}|}{|y_l^{(t)}-y_i^{(t)}|}, \epsilon = 0,1 \\ \epsilon\pi + \arctan\frac{|y_l^{(t)}-y_i^{(t)}|}{|x_l^{(t)}-x_i^{(t)}|},\ \epsilon = \frac{1}{2},\frac{3}{2} \end{cases}$, shown in Figure 8 with $\epsilon = 0$.

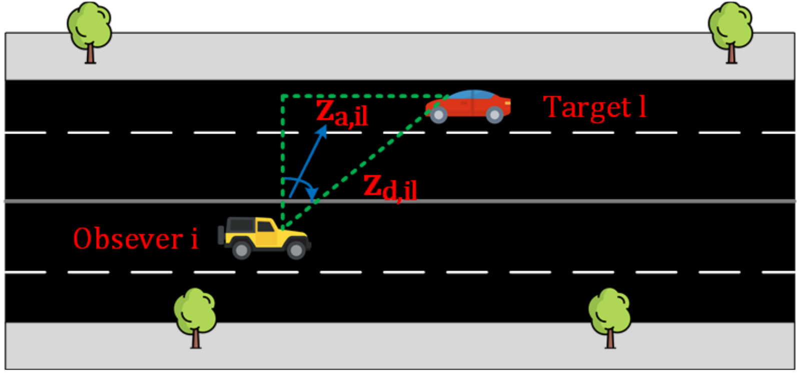


*Figure 8: Range measurements*

These measurements could be provided by sensors like GPS, LIDAR, RADAR or Camera, which are integrated to the vehicles, assuming also additive white Gaussian measurement noise [6]. Hence, we acquire the following three measurement models for each vehicle:

- Absolute position measurement: $\tilde{\boldsymbol{z}}_{\boldsymbol{p},\boldsymbol{i}}^{(\boldsymbol{t})} = \boldsymbol{p}_{\boldsymbol{i}}^{(\boldsymbol{t})} + \boldsymbol{n}_{\boldsymbol{p}}^{(\boldsymbol{t})},\ \boldsymbol{n}_{\boldsymbol{p}}^{(\boldsymbol{t})} \sim \mathcal{G}(0, \boldsymbol{\Sigma}_{\boldsymbol{p}})$

Covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{p}}$ is a diagonal matrix equal to $diag(\sigma_x^2, \sigma_y^2)$.

- Relative distance measurement: $\tilde{z}_{d,il}^{(t)} = z_{d,il}^{(t)} + n_d^{(t)},\ n_d^{(t)} \sim \mathcal{G}(0, \sigma_d^2)$
- Relative angle measurement: $\tilde{z}_{a,ij}^{(t)} = z_{a,ij}^{(t)} + n_a^{(t)},\ n_a^{(t)} \sim \mathcal{G}(0, \sigma_a^2)$
- Relative azimuth Angle measurement: $\tilde{z}_{az,il}^{(t)} = z_{az,il}^{(t)} + n_{az}^{(t)},\ n_{az}^{(t)} \sim \mathcal{G}(0, \sigma_{az}^2)$

We have to point out that an effective data association step is necessary for vehicle $i$ to match range measurement (from LIDAR) $\tilde{z}_{d,ij}^{(t)}$ to correct vehicle id $j$ (e.g. from V2V). This aspect is explicitly discussed in Section 3.5.1.

In the context of CA, ego vehicle $i$ is interested in estimating its position, along with the positions of vehicles belonging to its neighborhood $\mathcal{N}_i^{(t)}$ in a distributed manner. This group of collaborating vehicles makes up a star like V2V topology. Formally speaking, vehicle $i$ wants to estimate its state $\hat{\boldsymbol{x}}_{\boldsymbol{i}}^{(\boldsymbol{t})} \in \mathbb{R}^{3|\mathcal{N}_i^{(t)}|+3}$ which contains the 2D positions and yaw angles of itself and those vehicles which belong to its neighborhood in a cost effective and efficient manner. Although yaw is integrated to the state, in this work we are only interested in estimating 2D position.

A natural option for the CA estimation task is to employ the well known EKF algorithm. The latter treats the positioning problem as a probabilistic inference task, integrating also a node mobility model. Under such an approach, related algorithms return a set of possible locations leading to a more direct representation of the solution quality. For that reason, the non-linear state transition and measurement functions corresponding to each vehicle $i$ are exploited:

$$\hat{\boldsymbol{x}}_{\boldsymbol{i}}^{(\boldsymbol{t})} = f\left(\hat{\boldsymbol{x}}_{\boldsymbol{i}}^{(\boldsymbol{t-1})}, \boldsymbol{u}_{\boldsymbol{i}}^{(\boldsymbol{t})}\right) + \epsilon_1, \qquad \epsilon_1 \sim \mathcal{G}(0, \boldsymbol{R}_{\boldsymbol{i}}) \qquad \textit{Eq. 1}$$

$$\hat{\boldsymbol{z}}_{\boldsymbol{i}}^{(\boldsymbol{t})} = g\left(\hat{\boldsymbol{x}}_{\boldsymbol{i}}^{(\boldsymbol{t})}\right) + \epsilon_2, \qquad \epsilon_2 \sim \mathcal{G}(0, \boldsymbol{Q}_{\boldsymbol{i}}) \qquad \textit{Eq. 2}$$

State covariance matrix $\boldsymbol{R}_{\boldsymbol{i}} \in \mathbb{R}^{(3|\mathcal{N}_i^{(t)}|+3)\times(3|\mathcal{N}_i^{(t)}|+3)}$. As for the function $f(\cdot)$, we choose the bicycle kinematic model [7]:

$$x_i^{(t)} = x_i^{(t-1)} + \left(-\frac{v_i^{(t)}}{\omega_i^{(t)}}\right)\sin\theta_i^{(t-1)} + (v_i^{(t)}/\omega_i^{(t)})\sin(\theta_i^{(t-1)} + \omega_i^{(t)}\Delta T) \qquad \textit{Eq. 3}$$

$$y_i^{(t)} = y_i^{(t-1)} + \left(\frac{v_i^{(t)}}{\omega_i^{(t)}}\right)\cos\theta_i^{(t-1)} + (-v_i^{(t)}/\omega_i^{(t)})\cos(\theta_i^{(t-1)} + \omega_i^{(t)}\Delta T) \qquad \textit{Eq. 4}$$

$$\theta_i^{(t)} = \theta_i^{(t-1)} + \omega_i^{(t)}\Delta T, \qquad \textit{Eq. 5}$$

where $\Delta T$ is the sampling interval and $v_i^{(t)}, \omega_i^{(t)}$ are the linear velocity and yaw rate, respectively. The last two quantities are known as control inputs and provided by the IMU sensor. Note that IMU noise is modelled to follow the Gaussian distribution [8], with $\sigma_{v_i^{(t)}} = 0.1 v_i^{(t)}$ and $\sigma_{\omega_i^{(t)}} = 2°/\sqrt{hour}$. Thus, control input vector $\boldsymbol{u}_{\boldsymbol{i}}^{(\boldsymbol{t})} \in \mathbb{R}^{2|\mathcal{N}_i^{(t)}|+2}$ shall contain the two control inputs measured by ego vehicle and transmitted by each neighbor. Measurement vector $\hat{\boldsymbol{z}}_{\boldsymbol{i}}^{(\boldsymbol{t})}$ shall contain apart from GPS positions and yaws, not only the range measurements of ego towards neighbors, but also the range measurements of neighbors towards ego. However, neighbors estimate in fact a vector of range measurements since they have their own neighbors. Thus, ego vehicle has to find the "correct" range measurements which match to itself (data association). For example, for the operating vehicles $i$, $j$ and $k$, with $i$ as the ego, measurement vector shall contain in the optimal case: $\hat{\boldsymbol{z}}_{\boldsymbol{i}}^{(\boldsymbol{t})} = \begin{bmatrix} \tilde{z}_{d,ij}^{(t)} & \tilde{z}_{d,ik}^{(t)} & \tilde{z}_{d,ji}^{(t)} & \tilde{z}_{d,ki}^{(t)} & \tilde{z}_{a,ij}^{(t)} & \tilde{z}_{a,ik}^{(t)} & \tilde{z}_{a,ji}^{(t)} & \tilde{z}_{a,ki}^{(t)} & \hat{\boldsymbol{z}}_{\boldsymbol{p,i}}^{(\boldsymbol{t})} \\ \hat{\boldsymbol{z}}_{\boldsymbol{p,j}}^{(\boldsymbol{t})} & \hat{\boldsymbol{z}}_{\boldsymbol{p,k}}^{(\boldsymbol{t})} & \tilde{\theta}_i^{(t)} & \tilde{\theta}_j^{(t)} & \tilde{\theta}_k^{(t)} \end{bmatrix}^T$ As a matter of fact, $\hat{\boldsymbol{z}}_{\boldsymbol{i}}^{(\boldsymbol{t})} \in \mathbb{R}^{6|\mathcal{N}_i^{(t)}|+2}$ and $\boldsymbol{Q}_{\boldsymbol{i}} \in$

$\mathbb{R}^{(6|\mathcal{N}_i^{(t)}|+2)\times(6|\mathcal{N}_i^{(t)}|+2)}$. Note that $\tilde{\theta}_i^{(t)} = \theta_i^{(t)} + n_\theta$ is the yaw measurement, with $n_\theta \sim \mathcal{G}(0, \sigma^2_{\omega_i^{(t)}})$. We have to point out that in realistic traffic conditions, V2V topologies are dynamic since new vehicles may enter or exit them according to their $r_c$. However, in those cases EKF has to be reset since the state vector is no longer the same as before and reinitialized with a low cost one-shot like solution. The main steps of the discussed distributed $\boldsymbol{EKF\ for\ CA}$ or $\boldsymbol{EKF - CA}$ are summarized on **Algorithm 1**. Notice that due to the non-linearity of Eq. 1 and Eq. 2, EKF approximates the two functions via Taylor expansion, using the corresponding Jacobian matrices: $\boldsymbol{F} = \frac{\partial f(\cdot)}{\partial \hat{\boldsymbol{x}}_i^{(t-1)}}\Big|_{\overline{\boldsymbol{x}}_i^{(t)}}$ and $\boldsymbol{H} = \frac{\partial g(\cdot)}{\partial \hat{\boldsymbol{x}}_i^{(t)}}\Big|_{\overline{\boldsymbol{x}}_i^{(t)}}$.

***Algorithm 1:*** $\boldsymbol{EKF - CA}$

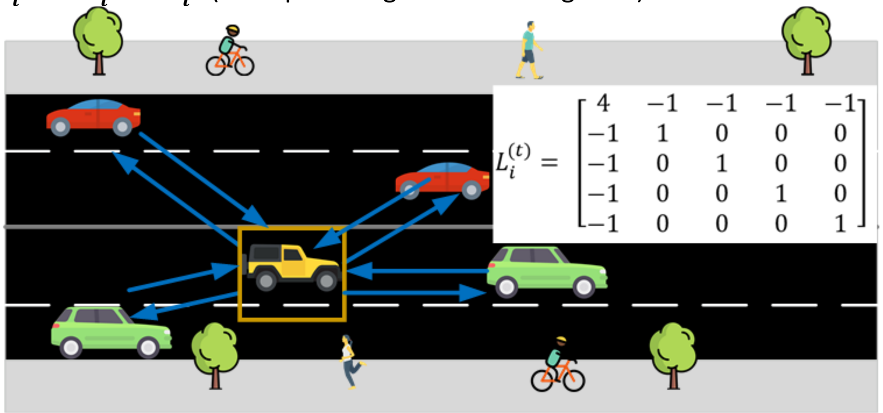
```
Input: $T, \boldsymbol{Q}_i, \boldsymbol{R}_i$
Output: $\hat{\boldsymbol{x}}_i^{(t)} \in \mathbb{R}^{3|\mathcal{N}_i^{(t)}|+3}$
For $t = 1,2,\dots T$ do
    For each ego vehicle $i$ do
        Vehicles transmit to $i$ their 2D GPS positions, vector of range measurements, linear velocity, yaw rate and yaw ;
        Ego matches range measurements to correct vehicles' ids (data association) ;
        If $\mathcal{N}_i^{(t)}$ not identical to $\mathcal{N}_i^{(t-1)}$ then
            Initialize locations with low cost one-shot solution ;
        Else
            $\overline{\boldsymbol{x}}_i^{(t)} = f\left(\hat{\boldsymbol{x}}_i^{(t-1)}, \boldsymbol{u}_i^{(t)}\right);$
            $\overline{\boldsymbol{\Sigma}}_i^{(t)} = \boldsymbol{F}_i\widehat{\boldsymbol{\Sigma}}_i^{(t-1)}\boldsymbol{F}_i^T + \boldsymbol{R}_i;$
            $\boldsymbol{K} = \overline{\boldsymbol{\Sigma}}_i^{(t)}\boldsymbol{H}_i^T\left(\boldsymbol{H}_i\overline{\boldsymbol{\Sigma}}_i^{(t)}\boldsymbol{H}_i^T + \boldsymbol{Q}_i\right)^{-1};$
            $\hat{\boldsymbol{x}}_i^{(t)} = \overline{\boldsymbol{x}}_i^{(t)} + \boldsymbol{K}\left(\hat{\boldsymbol{z}}_i^{(t)} - g\left(\overline{\boldsymbol{x}}_i^{(t)}\right)\right);$
            $\widehat{\boldsymbol{\Sigma}}_i^{(t)} = \left(\mathbb{I}_{3|\mathcal{N}_i^{(t)}|+3} - \boldsymbol{K}\boldsymbol{H}_i\right)\overline{\boldsymbol{\Sigma}}_i^{(t)};$
        End
    End
End
```

Therefore, we have derived a distributed tracking scheme aiming to provide increased CA ability to the vehicles. Ego vehicle receives the necessary information from its neighbors and executes the proposed $\boldsymbol{EKF - CA}$ in order to estimate its position and the positions of neighboring vehicles. Our approach was inspired by the functionalities executed at the traditional Simultaneous Localization and Mapping (SLAM) back-ends [9], [7], where the vehicle/robot navigates in unknown environments and tries to determine its position and the neighboring landmarks using pair-wise measurements.

## 3.2 Least-squares estimation

Vehicle $i$ can utilize the vectors of differential coordinates for $x$ and $y$ positions of star topology's vehicles: $\boldsymbol{\Delta}_i^{(t)} = \left[\boldsymbol{\delta}_i^{(x,t)} \quad \boldsymbol{\delta}_i^{(y,t)}\right] \in \mathbb{R}^{(|\mathcal{N}_i^{(t)}|+1)\times 2}$. For the case of ego vehicle: $\delta_{ii}^{(x,t)} = \sum_{j\in\mathcal{N}_i^{(t)}}\left(-\tilde{z}_{d,ij}^{(t)}\sin\tilde{z}_{az,ij}^{(t)}\right)$ and $\delta_{ii}^{(y,t)} = \sum_{j\in\mathcal{N}_i^{(t)}}\left(-\tilde{z}_{d,ij}^{(t)}\cos\tilde{z}_{az,ij}^{(t)}\right)$. For every neighboring vehicle $j \in \mathcal{N}_i^{(t)}$, ego vehicle has to compute the two following scalar values: $\delta_{ij}^{(x,t)} = -\tilde{z}_{d,ji}^{(t)}\sin\tilde{z}_{az,ji}^{(t)}$ and $\delta_{ij}^{(y,t)} = -\tilde{z}_{d,ji}^{(t)}\cos\tilde{z}_{az,ji}^{(t)}$. Neighbors actually transmit a vector of range measurements extracted by LIDAR and/or Cameras, $i$ receives it and has to match $\left(\tilde{z}_{d,jx}^{(t)}, \tilde{z}_{az,jx}^{(t)}\right)$ to $\left(\tilde{z}_{d,ji}^{(t)}, \tilde{z}_{az,ji}^{(t)}\right)$, since $j$ is not able to correlate LIDAR based detected vehicles with V2V individual ids. This necessary data association pre-processing step is omitted for the time being. Afterwards, $i$ defines the Laplacian matrix [6] $\boldsymbol{L}_i^{(t)} \in$

$\mathbb{R}^{(|\mathcal{N}_i^{(t)}|+1)\times(|\mathcal{N}_i^{(t)}|+1)}$ which captures the connections of star topology and then the extended Laplacian matrix $\boldsymbol{\tilde{L}_i^{(t)}} = \begin{bmatrix} \boldsymbol{L_i^{(t)}} \\ \mathbb{I}_{|\mathcal{N}_i^{(t)}|+1} \end{bmatrix}$, $\boldsymbol{\tilde{L}_i^{(t)}} \in \mathbb{R}^{(2|\mathcal{N}_i^{(t)}|+2)\times(|\mathcal{N}_i^{(t)}|+1)}$. Laplacian matrix is equal to the degree minus the adjacency matrix of the graph topology, i.e., $\boldsymbol{L_i^{(t)}} = \boldsymbol{D_i^{(t)}} - \boldsymbol{A_i^{(t)}}$(example for ego vehicle of Figure 9).



$$L_i^{(t)} = \begin{bmatrix} 4 & -1 & -1 & -1 & -1 \\ -1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

*Figure 9: Star V2V topology and interaction of vehicles*

Neighbors, apart from the vector of range measurements, transmit to ego vehicle their 2D GPS positions, and then $i$ creates the vector $\boldsymbol{b}^{(x,t)} = \begin{bmatrix} \boldsymbol{\delta_i^{(x,t)}} \\ \boldsymbol{\tilde{z}_{p,i}^{(x,t)}} \end{bmatrix}$, with $\boldsymbol{b}^{(x,t)} \in \mathbb{R}^{2|\mathcal{N}_i^{(t)}|+2}$, while $\boldsymbol{\tilde{z}_{p,i}^{(x,t)}} \in \mathbb{R}^{|\mathcal{N}_i^{(t)}|+1}$ are the GPS's $x$ positions. The latter are also known as (noisy) anchor points and are linked to the second half of $\boldsymbol{\tilde{L}_i^{(t)}}$ [6]. Finally, the $x$ positions $\boldsymbol{\tilde{x}_i^{(t)}} \in \mathbb{R}^{|\mathcal{N}_i^{(t)}|+1}$ of vehicles estimated by $i$, follow the linear system of:

$$\boldsymbol{\tilde{L}_i^{(t)}} \boldsymbol{\tilde{x}_i^{(t)}} = \boldsymbol{b}^{(x,t)} \qquad \textit{Eq. 6}$$

From least-squares minimization, we have that:

$$\boldsymbol{\tilde{x}_i^{(t)}} = (\boldsymbol{\tilde{L}_i^{(t)T}} \boldsymbol{\tilde{L}_i^{(t)}})^{-1} \boldsymbol{\tilde{L}_i^{(t)T}} \boldsymbol{b}^{(x,t)}$$

The same goes for estimating $y$ positions $\boldsymbol{\tilde{y}_i^{(t)}} \in \mathbb{R}^{|\mathcal{N}_i^{(t)}|+1}$. The proposed scheme was named $\boldsymbol{Distributed\ Laplacian\ Localization\ for\ CA}$ or $\boldsymbol{DLL-CA}$ while its main steps are summarized in **Algorithm 2**.

***Algorithm 2:*** $\boldsymbol{DLL-CA}$

```
Input: $T$
Output: $\tilde{x}_i^{(t)}, \tilde{y}_i^{(t)} \in \mathbb{R}^{|\mathcal{N}_i^{(t)}|+1}$
For $t = 1,2, \ldots T$ do
    For each ego vehicle $i$ do
        Vehicles transmit to $i$ their 2D GPS positions and vector of range measurements,;
        Ego matches range measurements to correct vehicles' ids (data association) ;
        $i$ defines the extended Laplacian matrix $\tilde{L}_i^{(t)}$ ;
        $i$ computes the differential coordinates $\Delta_i^{(t)}$ ;
        $\tilde{x}_i^{(t)} = (\tilde{L}_i^{(t)T} \tilde{L}_i^{(t)})^{-1} \tilde{L}_i^{(t)T} b^{(x,t)}$;
        $\tilde{y}_i^{(t)} = (\tilde{L}_i^{(t)T} \tilde{L}_i^{(t)})^{-1} \tilde{L}_i^{(t)T} b^{(y,t)}$;
    End
End
```

To sum up, we have derived a distributed CL scheme which focuses on CA. Differential coordinates, which capture the geometry of star topology and combine two non-linear measurement models are linearly fused with the Laplacian matrix, which dictates the connectivity representation of vehicles. Using least-squares minimization, ego vehicle performs the location estimation task. Note that this modelling enables the treatment of measurements and topology in a unified and compact manner which will offer very competitive advantages with respect to $EKF - CA$, as shown in Section 3.6. We have to point out that $\boldsymbol{DLL - CA}$ is inherently different than the distributed method of [10], since the latter focuses explicitly on ego vehicle location estimation.

## 3.3 Graph Laplacian Extended Kalman Filter

Although $\boldsymbol{DLL - CA}$ enables ego and neighbors locations' estimation, it does that in a static and one-shot manner, treating each time instant independently. It would be more efficient for both CL and CA tasks to exploit the motion of vehicles within a tracking scheme. For that purpose, we resort once again to prominent EKF in order to improve the positioning accuracy.

Vehicle $i$ is interested to estimate the states $\hat{\boldsymbol{x}}_{\boldsymbol{i}}^{(\boldsymbol{t})} \in \mathbb{R}^{3|\mathcal{N}_i^{(t)}|+3}$, i.e. 2D positions and yaws of star topology's vehicles given their control inputs $\boldsymbol{u}_{\boldsymbol{i}}^{(\boldsymbol{t})} \in \mathbb{R}^{2|\mathcal{N}_i^{(t)}|+2}$ and measurements $\hat{\boldsymbol{z}}_{\boldsymbol{i}}^{(\boldsymbol{t})}$. As a matter of fact, the kinematic model of (3)-(5) and the non-linear transition function of $\boldsymbol{EKF - CA}$ are used. For that reason, neighboring vehicles to $i$ have to transmit their linear velocities, yaw rates and yaws. The main contribution in deriving the proposed Graph Laplacian EKF is to define a linear measurement model which in addition integrates all three measurement models in a unified manner. On the contrary, non-linear pair-wise measurements of $\boldsymbol{EKF - CA}$ require the Jacobian matrices, which however introduce high positioning errors due to the Taylor first order approximation.

Measurement model is actually based on Eq. 6. Therefore, measurement vector $\hat{\boldsymbol{z}}_{\boldsymbol{i}}^{(\boldsymbol{t})} \in \mathbb{R}^{5|\mathcal{N}_i^{(t)}|+5}$ will be comprised of differential coordinates and yaw measurements: $\hat{\boldsymbol{z}}_{\boldsymbol{i}}^{(\boldsymbol{t})} = \left[\boldsymbol{\delta}_{\boldsymbol{i}}^{(x,t)} \quad \boldsymbol{\delta}_{\boldsymbol{i}}^{(y,t)} \quad \widetilde{\boldsymbol{\theta}}_{\boldsymbol{i}}^{(t)}\right]^T$, with yaw measurements vector $\widetilde{\boldsymbol{\theta}}_{\boldsymbol{i}}^{(\boldsymbol{t})} \in \mathbb{R}^{|\mathcal{N}_i^{(t)}|+1}$ of star topology. Furthermore, the function $g(\hat{\boldsymbol{x}}_{\boldsymbol{i}}^{(\boldsymbol{t})})$ follows the linear system of: $g(\hat{\boldsymbol{x}}_{\boldsymbol{i}}^{(\boldsymbol{t})}) = \boldsymbol{H}_{\boldsymbol{i}}\hat{\boldsymbol{x}}_{\boldsymbol{i}}^{(\boldsymbol{t})}$, with block diagonal matrix $\boldsymbol{H}_{\boldsymbol{i}} \in \mathbb{R}^{(5|\mathcal{N}_i^{(t)}|+5)\times(3|\mathcal{N}_i^{(t)}|+3)}$ equal to:

$$\boldsymbol{H}_{\boldsymbol{i}} = \begin{bmatrix} \widetilde{\boldsymbol{L}}_{\boldsymbol{i}}^{(\boldsymbol{t})} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \widetilde{\boldsymbol{L}}_{\boldsymbol{i}}^{(\boldsymbol{t})} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \mathbb{I}_{|\mathcal{N}_i^{(t)}|+1} \end{bmatrix}$$

Hence, the measurement model which extends Eq. 6 in order to capture 2D positions and yaws using the differential coordinates and the connectivity representation of vehicles, will be given by:

$$\hat{\boldsymbol{z}}_{\boldsymbol{i}}^{(\boldsymbol{t})} = \boldsymbol{H}_{\boldsymbol{i}}\hat{\boldsymbol{x}}_{\boldsymbol{i}}^{(\boldsymbol{t})}$$

Finally, covariance matrices $\boldsymbol{R}_{\boldsymbol{i}} \in \mathbb{R}^{(3|\mathcal{N}_i^{(t)}|+3)\times(3|\mathcal{N}_i^{(t)}|+3)}$ and $\boldsymbol{Q}_{\boldsymbol{i}} \in \mathbb{R}^{(5|\mathcal{N}_i^{(t)}|+5)\times(5|\mathcal{N}_i^{(t)}|+5)}$. The main steps of the proposed $\boldsymbol{Laplacian\ EKF\ for\ CA}$ or $\boldsymbol{LEKF - CA}$ are summarized in **Algorithm 3**.

***Algorithm 3:*** $\boldsymbol{LEKF - CA}$

```
Input: $T, \boldsymbol{Q}_{\boldsymbol{i}}, \boldsymbol{R}_{\boldsymbol{i}}$
Output: $\hat{\boldsymbol{x}}_{\boldsymbol{i}}^{(\boldsymbol{t})} \in \mathbb{R}^{3|\mathcal{N}_i^{(t)}|+3}$
For $t = 1,2, \dots T$ do
    For each ego vehicle $i$ do
        Vehicles transmit to $i$ their 2D GPS positions and vector of range measurements,;
        Ego matches range measurements to correct vehicles' ids (data association) ;
        $i$ defines the extended Laplacian matrix $\widetilde{\boldsymbol{L}}_{\boldsymbol{i}}^{(\boldsymbol{t})}$ ;
        $i$ computes the differential coordinates $\boldsymbol{\Delta}_{\boldsymbol{i}}^{(\boldsymbol{t})}$ ;
        If $\mathcal{N}_i^{(t)}$ not identical to $\mathcal{N}_i^{(t-1)}$ then
            Initialize locations with $\boldsymbol{DLL - CA}$;
        Else
```

```
                $\bar{\boldsymbol{x}}_i^{(t)} = f\left(\hat{\boldsymbol{x}}_i^{(t-1)}, \boldsymbol{u}_i^{(t)}\right);$
                $\bar{\boldsymbol{\Sigma}}_i^{(t)} = \boldsymbol{F}_i \hat{\boldsymbol{\Sigma}}_i^{(t-1)} \boldsymbol{F}_i^T + \boldsymbol{R}_i;$
                $\boldsymbol{K} = \bar{\boldsymbol{\Sigma}}_i^{(t)} \boldsymbol{H}_i^T \left(\boldsymbol{H}_i \bar{\boldsymbol{\Sigma}}_i^{(t)} \boldsymbol{H}_i^T + \boldsymbol{Q}_i\right)^{-1};$
                $\hat{\boldsymbol{x}}_i^{(t)} = \bar{\boldsymbol{x}}_i^{(t)} + \boldsymbol{K}\left(\hat{\boldsymbol{z}}_i^{(t)} - \boldsymbol{H}_i \bar{\boldsymbol{x}}_i^{(t)}\right);$
                $\hat{\boldsymbol{\Sigma}}_i^{(t)} = \left(\mathbb{I}_{3|\mathcal{N}_i^{(t)}|+3} - \boldsymbol{K}\boldsymbol{H}_i\right) \bar{\boldsymbol{\Sigma}}_i^{(t)};$
            End
        End
    End
```

Thus, a distributed tracking scheme aiming to CA has been formulated, based on Graph Laplacian processing and EKF's algorithm. Ego vehicle acts now as the fusion center. It receives from its connected neighbors all the necessary measurements, defines the extended Laplacian matrix of its own star topology, computes and finally fuses the differential coordinates, in order to estimate its position along with those of its neighbors. The case of dynamic neighborhoods for the time being are treated as locations re-initialization with $\boldsymbol{DLL} - \boldsymbol{CA}$ solution.

## 3.4 Utility scores through Crámer-Rao Lower Bound

It is often highly desirable to accurately measure the performance of localization, and in more general, estimation algorithms. For this fundamental task, Cramér-Rao Lower Bound (CRLB) and Fisher Information Matrix (FIM) are necessary. For any unbiased estimation $\hat{\boldsymbol{x}}$ of parameter $\boldsymbol{x}$, FIM $\boldsymbol{J}(\boldsymbol{x})$ indicates the amount of information the observations carry about the unknown parameter $\boldsymbol{x}$ on average [6]. Let $f(\boldsymbol{x};\hat{\boldsymbol{x}})$ be the probability distribution function for $\boldsymbol{x}$ conditioned on $\hat{\boldsymbol{x}}$. Then, FIM is equal to:

$$\boldsymbol{J}(\boldsymbol{x}) = -\mathbb{E}\left[\frac{\partial^2}{\partial \hat{\boldsymbol{x}}^2} f(\boldsymbol{x};\hat{\boldsymbol{x}})\Big|_{\hat{\boldsymbol{x}}}\right]$$

Instead of $f(\boldsymbol{x};\hat{\boldsymbol{x}})$, log-likelihood $\log f(\boldsymbol{x};\hat{\boldsymbol{x}})$ can be used (FIM now known as the variance of score). Then, CRLB is:

$$\boldsymbol{J}^{-1}(\boldsymbol{x}) \preceq \mathbb{E}\{(\hat{\boldsymbol{x}} - \boldsymbol{x})(\hat{\boldsymbol{x}} - \boldsymbol{x})^T\} \qquad \textit{Eq. 7}$$

If, in addition, we take the trace and square root, then (7) transforms to:

$$\sqrt{tr(\boldsymbol{J}^{-1}(\boldsymbol{x}))} \le \sqrt{tr(\mathbb{E}\{(\hat{\boldsymbol{x}} - \boldsymbol{x})(\hat{\boldsymbol{x}} - \boldsymbol{x})^T})\} \qquad \textit{Eq. 8}$$

Eq. 7 and Eq. 8 simply state that the minimum attained root mean square estimation (position) error is equal to $\sqrt{tr(\boldsymbol{J}^{-1}(\boldsymbol{x}))}$ [6]. CRLB can also be used for evaluating whether or not the collaboration of nodes contribute to the improvement of positioning accuracy. As it is mentioned in [11], collaboration indeed improved position estimates. Therefore, we have to determine the FIM for each one of the methods. More specifically, the FIM for $\boldsymbol{DLL} - \boldsymbol{CA}$ is equal to [12] $\begin{bmatrix} \tilde{\boldsymbol{L}}_i^{(t)T} \tilde{\boldsymbol{L}}_i^{(t)} & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{L}}_i^{(t)T} \tilde{\boldsymbol{L}}_i^{(t)} \end{bmatrix}$ in order to capture both $x$ and $y$ locations' estimation. As for the discussed EKF based solutions, it is proven in [13] that inverse FIM is equal to the predicted covariance matrix of EKF algorithm. Since yaw is also estimated by EKF, we have to keep only the parts of covariance matrices which correspond to $x$ and $y$ positions. Thus, the individual FIM are equal to:

$\boldsymbol{EKF} - \boldsymbol{CA}$: $\boldsymbol{J}^{-1}(\boldsymbol{x}_i^{(t)}) = \hat{\boldsymbol{\Sigma}}_i^{(t)}[1{:}\,2|\mathcal{N}_i^{(t)}| + 2, 1{:}\,2|\mathcal{N}_i^{(t)}| + 2]$

$\boldsymbol{LEKF} - \boldsymbol{CA}$: $\boldsymbol{J}^{-1}(\boldsymbol{x}_i^{(t)}) = \hat{\boldsymbol{\Sigma}}_i^{(t)}[1{:}\,2|\mathcal{N}_i^{(t)}| + 2, 1{:}\,2|\mathcal{N}_i^{(t)}| + 2]$

## 3.5 Challenges in Practical Considerations

This Section discusses three main challenges of CA: i) how ego vehicle matches range measurements from LIDAR and/or Cameras with the correct vehicle's id determined by V2V communication, ii) how EKF can be reformulated in order to capture and be adaptive to dynamic and modifying star topologies, and iii) range measurements uncertainty.

### 3.5.1 Data association

It is evident that vehicles which detect nearby objects using their Perception module (LIDAR, Camera, CNN) and measure the relative distances and angles, don't have the ability to directly link the measurements with specific V2V neighbors ids. However, the discussed localization methods rely on the assumption that vehicle $i$ knows in fact that $(\tilde{z}_{d,ij}^{(t)}, \tilde{z}_{az,ij}^{(t)})$ correspond to neighbor $j$. Therefore, a mechanism which is able to provide this kind of matching is of great importance.

Ego vehicle $i$ receives from its connected neighbors their GPS positions, i.e. the former knows exactly which (noisy) positions correspond to individual neighboring vehicles. Since ego is actually interested in computing the differential coordinates of its neighbor $j$ towards itself, it creates the synthetic relative distance and azimuth angle using its own GPS position:

$$d_s^{(t)} = \left\| \tilde{\boldsymbol{z}}_{\boldsymbol{p,i}}^{(\boldsymbol{t})} - \tilde{\boldsymbol{z}}_{\boldsymbol{p,j}}^{(\boldsymbol{t})} \right\|, \qquad a_s^{(t)} = \begin{cases} \lambda\pi + \arctan \frac{|\tilde{z}_{p,i}^{(x,t)} - \tilde{z}_{p,j}^{(x,t)}|}{|\tilde{z}_{p,i}^{(y,t)} - \tilde{z}_{p,j}^{(y,t)}|}, \lambda = 0,1 \\ \lambda\pi + \arctan \frac{|\tilde{z}_{p,i}^{(y,t)} - \tilde{z}_{p,j}^{(y,t)}|}{|\tilde{z}_{p,i}^{(x,t)} - \tilde{z}_{p,j}^{(x,t)}|}, \ \lambda = \frac{1}{2}, \frac{3}{2} \end{cases},$$

and then the vector:

$$\boldsymbol{p}_{\boldsymbol{s}}^{(\boldsymbol{t})} = \left[ -d_s^{(t)} \mathrm{sin} a_s^{(t)} \quad -d_s^{(t)} \mathrm{cos} a_s^{(t)} \right]^T \qquad \textit{Eq. 9}$$

Afterwards, since ego vehicle $i$ receives from $j$ the latter's vector of range measurements, creates the matrix which contain the differential coordinates of $j$ towards its own neighbors:

$$\boldsymbol{P}_{\boldsymbol{s}}^{(\boldsymbol{t})} = \begin{bmatrix} -\tilde{z}_{d,j1}^{(t)} \mathrm{sin} \tilde{z}_{az,j1}^{(t)} & \dots & -\tilde{z}_{d,j|\mathcal{N}_j^{(t)}|}^{(t)} \mathrm{sin} \tilde{z}_{az,j|\mathcal{N}_j^{(t)}|}^{(t)} \\ -\tilde{z}_{d,j1}^{(t)} \mathrm{cos} \tilde{z}_{az,j1}^{(t)} & \dots & -\tilde{z}_{d,j|\mathcal{N}_j^{(t)}|}^{(t)} \mathrm{cos} \tilde{z}_{az,j|\mathcal{N}_j^{(t)}|}^{(t)} \end{bmatrix} \qquad \textit{Eq. 10}$$

Finally, $i$ tries to find which elements of $\boldsymbol{P}_{\boldsymbol{s}}^{(\boldsymbol{t})}$ are more similar to $x$ and $y$ components of $\boldsymbol{p}_{\boldsymbol{s}}^{(\boldsymbol{t})}$. For that reason, it has to find the minimum values of $\left\| \boldsymbol{p}_{\boldsymbol{s}}^{(\boldsymbol{t})} - \boldsymbol{P}_{\boldsymbol{s}}^{(\boldsymbol{t})}[:, k_1] \right\|$, $\forall k_1 \in [1, |\mathcal{N}_j^{(t)}|]$. Should the minimum is determined, column index $k_1$ would coincide (in the optimal case) with the desired $(-\tilde{z}_{d,ji}^{(t)} \mathrm{sin} \tilde{z}_{az,ji}^{(t)}, -\tilde{z}_{d,ji}^{(t)} \mathrm{cos} \tilde{z}_{az,ji}^{(t)})$.

The proposed data association strategy correlates GPS with range measurements, since GPS positions indicate the individual ids of connected neighbors. Obviously, measurements (either GPS or range) highly contaminated by noise seriously affect the performance of data association. A similar approach is followed in $\boldsymbol{EKF - CA}$, though instead of differential coordinates, relative distances and angles are treated separately. We have to point out that under the assumption that vehicles detect only those vehicles which can communicate, proposed Graph Laplacian methods exhibit a competitive advantage with respect to $\boldsymbol{EKF - CA}$: no data association is required for computing ego vehicle's differential coordinates, since it suffices to take into account the average of range measurements. The main steps of the proposed data association strategy for $\boldsymbol{DLL - CA}$ and $\boldsymbol{LEKF - CA}$ are summarized in **Algorithm 4**

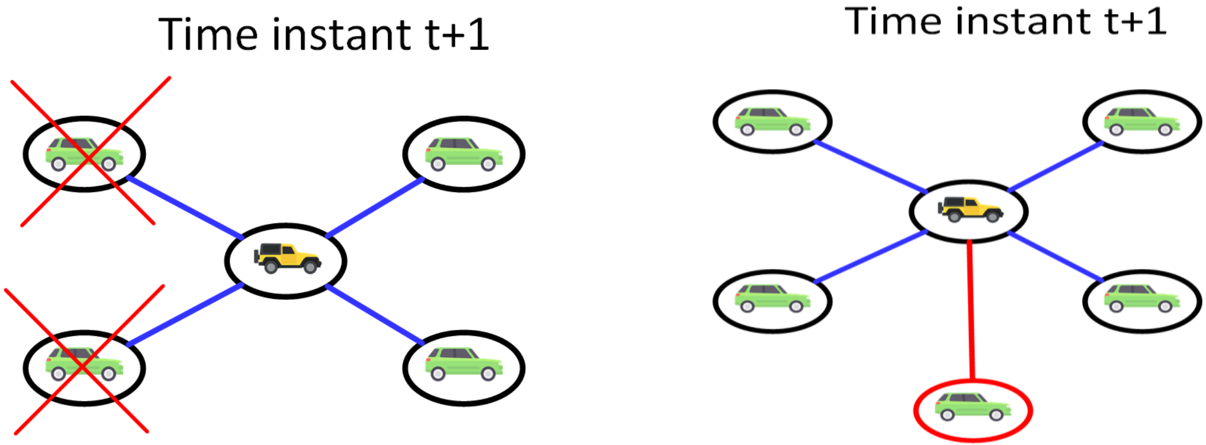
***Algorithm 4: Data association***

```
Input: GPS positions, vector of range measurements
Output: matching of vehicles' ids with the range measurements
For $t = 1,2, \dots T$ do
    For each ego vehicle $i$ do
        Vehicles transmit to $i$ their 2D GPS positions and vector of range measurements ;
        Ego matches range measurements to correct vehicles' ids (data association) ;
        $i$ defines the extended Laplacian matrix $\tilde{\boldsymbol{L}}_{\boldsymbol{i}}^{(\boldsymbol{t})}$ ;
        $i$ computes the differential coordinates $\boldsymbol{\Delta}_{\boldsymbol{i}}^{(\boldsymbol{t})}$ ;
        For every neighbor $j \in \mathcal{N}_i^{(t)}$ in parallel do
```

```
            $i$ computes $d_s^{(t)} = \left\|\tilde{\boldsymbol{z}}_{\boldsymbol{p,i}}^{(\boldsymbol{t})} - \tilde{\boldsymbol{z}}_{\boldsymbol{p,j}}^{(\boldsymbol{t})}\right\|, a_s^{(t)} = \begin{cases} \lambda\pi + \arctan\frac{|\tilde{z}_{p,i}^{(x,t)} - \tilde{z}_{p,j}^{(x,t)}|}{|\tilde{z}_{p,i}^{(y,t)} - \tilde{z}_{p,j}^{(y,t)}|}, \lambda = 0,1 \\ \lambda\pi + \arctan\frac{|\tilde{z}_{p,i}^{(y,t)} - \tilde{z}_{p,j}^{(y,t)}|}{|\tilde{z}_{p,i}^{(x,t)} - \tilde{z}_{p,j}^{(x,t)}|}, \ \lambda = \frac{1}{2}, \frac{3}{2} \end{cases}$;
            $i$ creates vector $\boldsymbol{p}_{\boldsymbol{s}}^{(\boldsymbol{t})}$ from Eq. 9;
            $i$ creates matrix $\boldsymbol{P}_{\boldsymbol{s}}^{(\boldsymbol{t})}$ from Eq. 10;
            find $k_1 \in [1, |\mathcal{N}_j^{(t)}|]$ which minimize $\left\|\boldsymbol{p}_{\boldsymbol{s}}^{(\boldsymbol{t})} - \boldsymbol{P}_{\boldsymbol{s}}^{(\boldsymbol{t})}[:, k_1]\right\|$ ;
            $k_1$ would correspond to $\left(-\tilde{z}_{d,ji}^{(t)}\sin\tilde{z}_{az,ji}^{(t)}, -\tilde{z}_{d,ji}^{(t)}\cos\tilde{z}_{az,ji}^{(t)}\right)$;
        End
    End
End
```

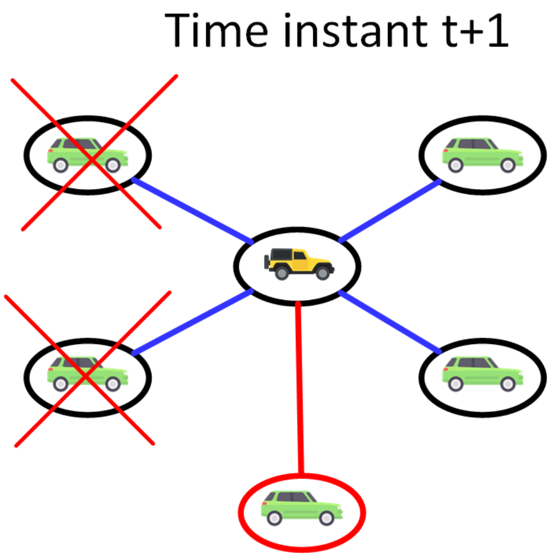
### 3.5.2 Adaptability to modifying CAV’s topologies

Ego vehicle $i$ forms V2V connections with nearby vehicles according to their communication ranges, while they are moving in urban canyons. As long as the related star topology doesn’t change, ego executes iteratively e.g. the $\boldsymbol{LEKF - CA}$, in order to estimate vehicles’ positions. However, when nearby vehicles are moving in the opposite direction or perform turn maneuvers, then it is likely that after a small time period the star topology will be modified. This is due to the fact that vehicles exit, enter or both exit and enter the star topology. These three dynamic cases are highlighted in Figure 10.



(a) *Exit*

(b) *Enter*



(c) $Exit\ and\ enter$

*Figure 10: Cases of dynamic CAV's topologies*

In all three cases, $\boldsymbol{LEKF-CA}$ has to be reset since both the state vector and covariance matrix do not correspond to the current neighbors of $i$. Instead of initialization with $\boldsymbol{DLL-CA}$ scheme, it is more efficient to reformulate the state vector and covariance matrix in order to capture topology's modifications, without resetting EKF's operation. Therefore, a simple yet effective approach is described which offers a kind of adaptability to $\boldsymbol{LEKF-CA}$:

1. Vehicles exit the topology (Figure 10-(a)):

Delete the elements and rows/columns of $\widehat{\boldsymbol{x}}_{\boldsymbol{i}}^{(\boldsymbol{t-1})}$ and $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{i}}^{(\boldsymbol{t-1})}$ which correspond to the outgoing vehicles.

2. Vehicles enter the topology (Figure 10-(b)):

Add to $\widehat{\boldsymbol{x}}_{\boldsymbol{i}}^{(\boldsymbol{t-1})}$ the 2D positions and yaws of incoming vehicles, using $\boldsymbol{DLL-CA}$. Add to $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{i}}^{(\boldsymbol{t-1})}$ new rows and columns with 0's and 1 to the diagonal, corresponding to incoming vehicles.

3. Vehicles exit and enter the topology (Figure 10-(c)):

Combine 1) and 2) cases.

Note that ego vehicle determines outgoing and incoming vehicles according to their individual V2V ids. The described approach offers adaptability to dynamic CAV's topologies to the EKF-based solutions and replaces the locations' initialization of **Algorithm 1** and **Algorithm 3**.

### 3.5.3 Range measurements uncertainties

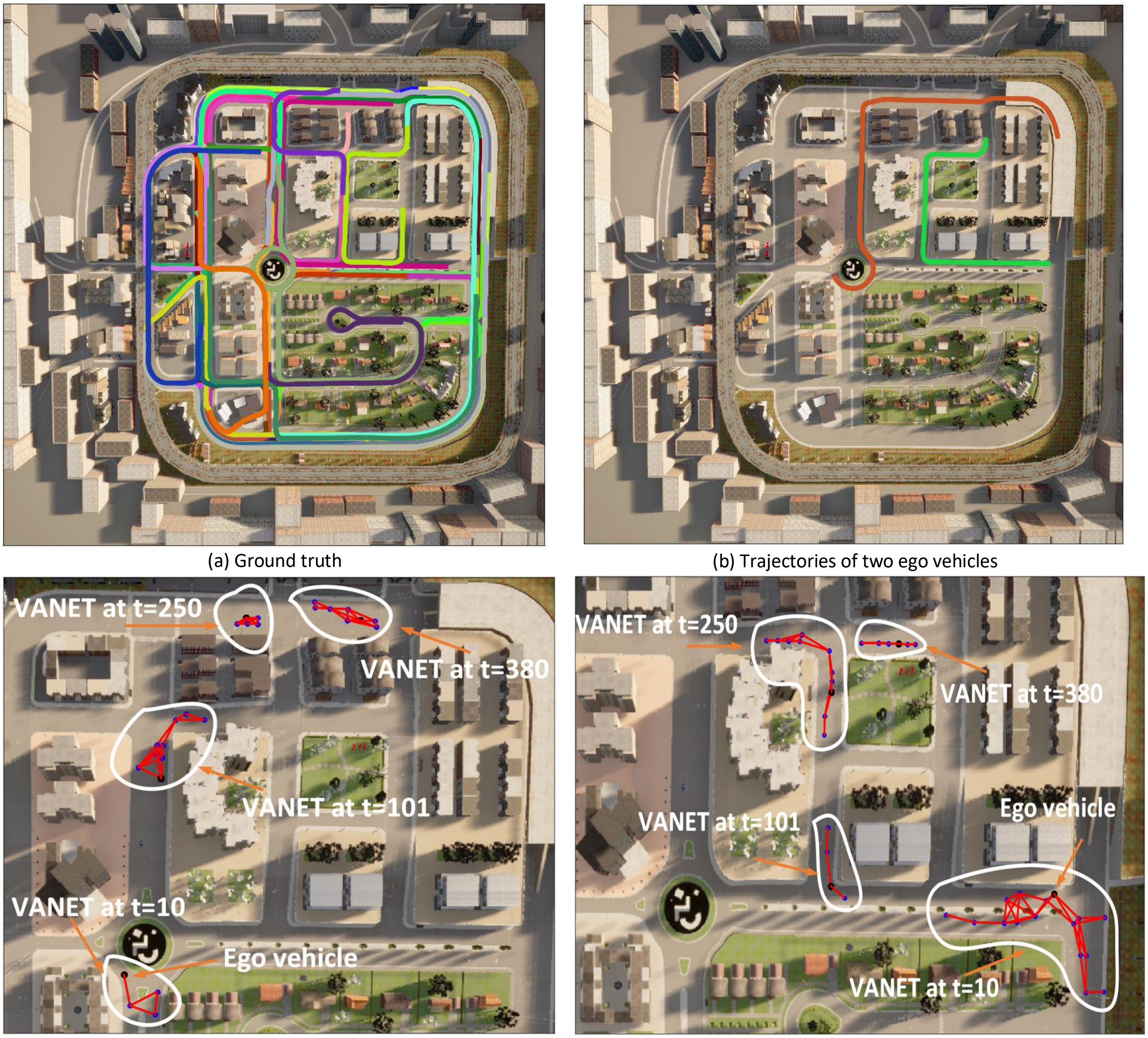
Advanced visual sensors of LIDAR and Cameras capture the traffic scenes and provide fine details of nearby detected vehicles. Nonetheless, due to a variety of reasons like diverse weather conditions (e.g. sunny, rainy, fog, etc.), night time, scattered environments, viewpoint changes of sensors, etc., extracted range measurements may suffer from lower accuracy. Furthermore, in dense urban canyons traffic scenes contain occluded objects, and as a matter of fact not direct LOS link could be established between the vehicles. To explicitly model those NLOS links, the Gaussian measurement model of relative distance and angle/azimuth angle is not sufficient. For this task, and as mentioned in [14], we have to deduce to the two-mode Gaussian mixture:

$$n_d^{(t)} \sim \alpha \cdot \mathcal{G}(0, \sigma_d^2) + (1-\alpha) \cdot \mathcal{G}(\mu_d^{NLOS}, \sigma_d^{NLOS}),$$

with $\mu_d^{NLOS} > 0$ and $\sigma_d^{NLOS} > \sigma_d$. Simply stated, $n_d^{(t)}$ falls in the LOS distribution with probability $\alpha$ and the NLOS distribution with $1-\alpha$. The same goes also for $n_a^{(t)}, n_{az}^{(t)}$.

## 3.6 Experimental setup and results

We extracted using the CARLA simulator the trajectories of $N = 150$ vehicles for $T = 448$ time instances, with sampling interval $\Delta T = 0.4sec$. The latter coincides with GPS updating time. Note that in realistic road conditions, if updating time is higher than $0.4sec$ then GPS's utility is of limited use in real-time applications. Example trajectories of $50$ vehicles moving in the simulated city of CARLA are shown in Figure 11-(a). We have chosen two random ego vehicles (trajectories shown in Figure 11-(b)) to conduct the related experiments. Additionally, the VANETs which belong to at four different time slots are demonstrated in Figure 11-(c),(d).



(a) Ground truth

(b) Trajectories of two ego vehicles

(c) Different VANETs for ego 1

(d) Different VANETs for ego 2

*Figure 11: Trajectories and VANETs*

Since vehicles are moving in urban environment with low mobility, linear velocity and yaw rate range between $0 - 20.09km/h$ and $0 - 0.73rad/sec$ for ego vehicle 1 and $0 - 22.3km/h$ and $0 - 0.62rad/sec$ for ego vehicle 2. Maximum number of connected neighbors $N_{max}$ ranges between 1 and 7 for the two vehicles with $r_c = 20m$. GPS noise has been generated by setting $\sigma_x = 3m$ and $\sigma_y = 2.5m$. Additionally, it is assumed that all methods don't have any knowledge of measurement noise variance, and as a matter of fact covariance matrices were initialized to identity. Conducted experiments study the: i) impact of vehicles' network connections and ii) highly noisy range measurements between the vehicles, and iii) CA performance. Network connections, i.e. number of connected neighbors for each vehicle are related to the communication range parameter $r_c$. As evaluation metrics, individual Localization Error:

$$LE_{(i)}^{(t)} = \left\| \boldsymbol{x}_{\boldsymbol{i}}^{(\boldsymbol{t})} - \left[\hat{x}_i^{(t)} \quad \hat{y}_i^{(t)}\right]^T \right\|$$

overall Localization Mean Square Error:

$$LMSE^{(t)} = \frac{1}{N}\sum_{i=1}^{N} \left\| \boldsymbol{x}_{\boldsymbol{i}}^{(\boldsymbol{t})} - \left[\hat{x}_i^{(t)} \quad \hat{y}_i^{(t)}\right]^T \right\|^2$$

and Average Localization Error for the CA task of ego vehicle $i$:

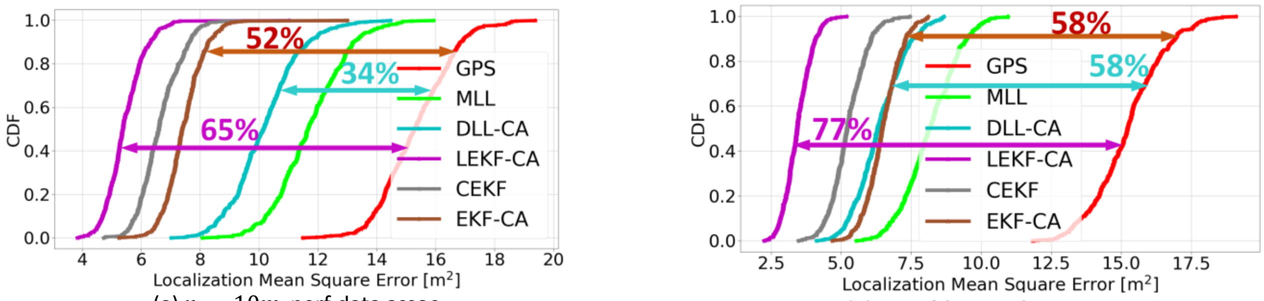
$$ALE_{(i)}^{(t)} = \frac{1}{|\mathcal{N}_i^{(t)}|-1}\sum_{j\in\mathcal{N}_i^{(t)}\setminus i} \left\| \boldsymbol{x}_{\boldsymbol{j}}^{(\boldsymbol{t})} - \left[\hat{x}_j^{(t)} \quad \hat{y}_j^{(t)}\right]^T \right\|$$

have been exploited. Furthermore, we constructed the corresponding Cumulative Distribution Function (CDF).

Two traditional centralized CL techniques have been employed to evaluate the performance of our schemes, assuming that a fusion center has access to the measurements of the vehicles of the cluster $C_i^{(t)}$. The first one is in fact static and relies on Maximum Likelihood Estimation (MLE) critection to define a cost function [15], which integrates the different measurement modalities. MLE is useful due to its consistency, asymptotic optimality and normality properties. The aim is to minimize this cost function in order to estimate the positions. MLE is useful due to its consistency, asymptotic optimality and normality properties. To minimize the cost function, a GD solution [16] can be used due to its simplicity. This approach was named $\boldsymbol{Maximum\ Likelihood\ based\ Localization}$ or $\boldsymbol{MLL}$. The second method is an adaptive or tracking online method discussed in [17]. It is a straightforward EKF centralized implementation, in which the vehicles broadcast the necessary information and control inputs to the fusion center, which in turn estimates their positions. We have extended it by adding to the measurement model: relative distances, 2D GPS positions and yaw measurements, apart from relative angle measurements. As for the state transition function, the bicycle kinematic model can be used. This method was named $\boldsymbol{Centralized\ EKF}$ or $\boldsymbol{CEKF}$. Additionally, a specific subsection in the Evaluation Study is dedicated to the presentation of distributed $\boldsymbol{Multi\ Sensor\ Multi\ Vehicle}$ ($\boldsymbol{MSMV}$) [18] method.

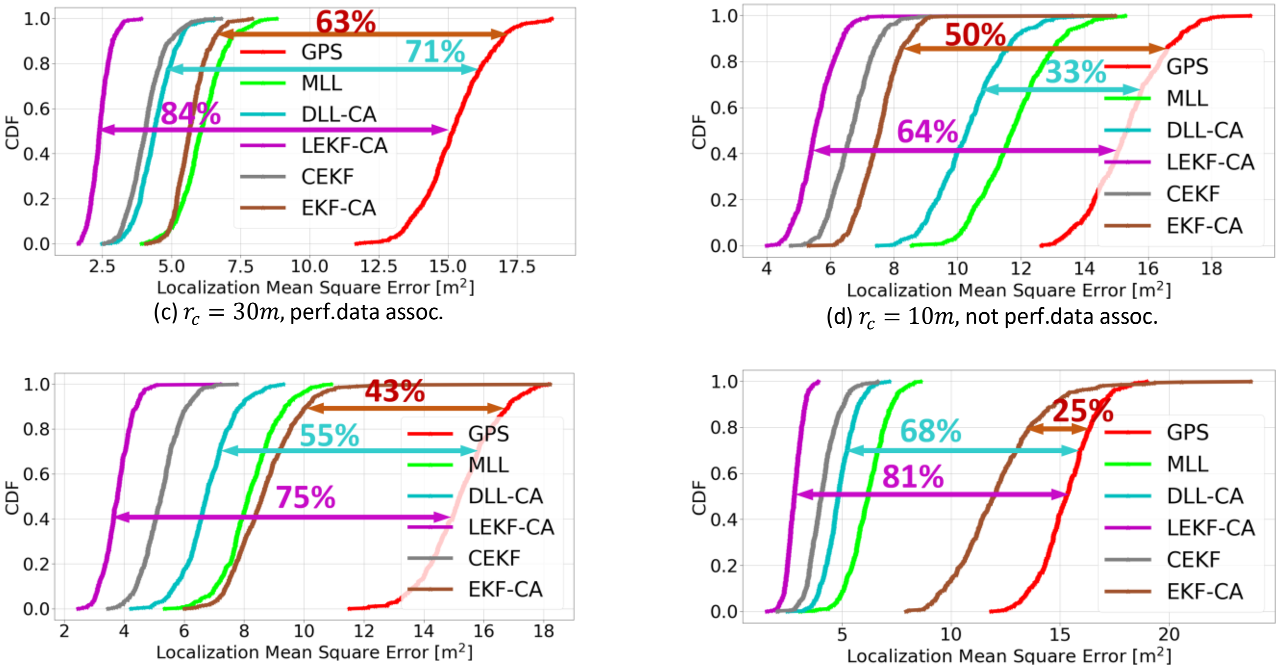
1. Impact of vehicles' connections

Vehicles may communicate and establish V2V links with a different number of vehicles while they are moving. Therefore, the amount of relative range measurements could be modified and as a matter of fact, it is straightforward to study the impact of network connections to the location estimation accuracy. For this task, three different communication ranges have been utilized, i.e. $r_c = \{10m, 20m, 30m\}$, without imposing any restriction to the maximum allowed number of connected neighbors for each vehicle. Moreover, perfect as well as non perfect data association will be discussed. For comparative reasons, centralized methods of $\boldsymbol{CEKF}$ and $\boldsymbol{MLL}$ have been implemented with perfect data association. The impact of communication range is demonstrated in Figure 12, with $\sigma_d = 1m$ and $\sigma_a = 4°$.



(a) $r_c = 10m$, perf.data assoc.

(b) $r_c = 20m$, perf.data assoc.



(c) $r_c = 30m$, perf.data assoc.

(d) $r_c = 10m$, not perf.data assoc.

(e) $r_c = 20m$, not perf.data assoc.

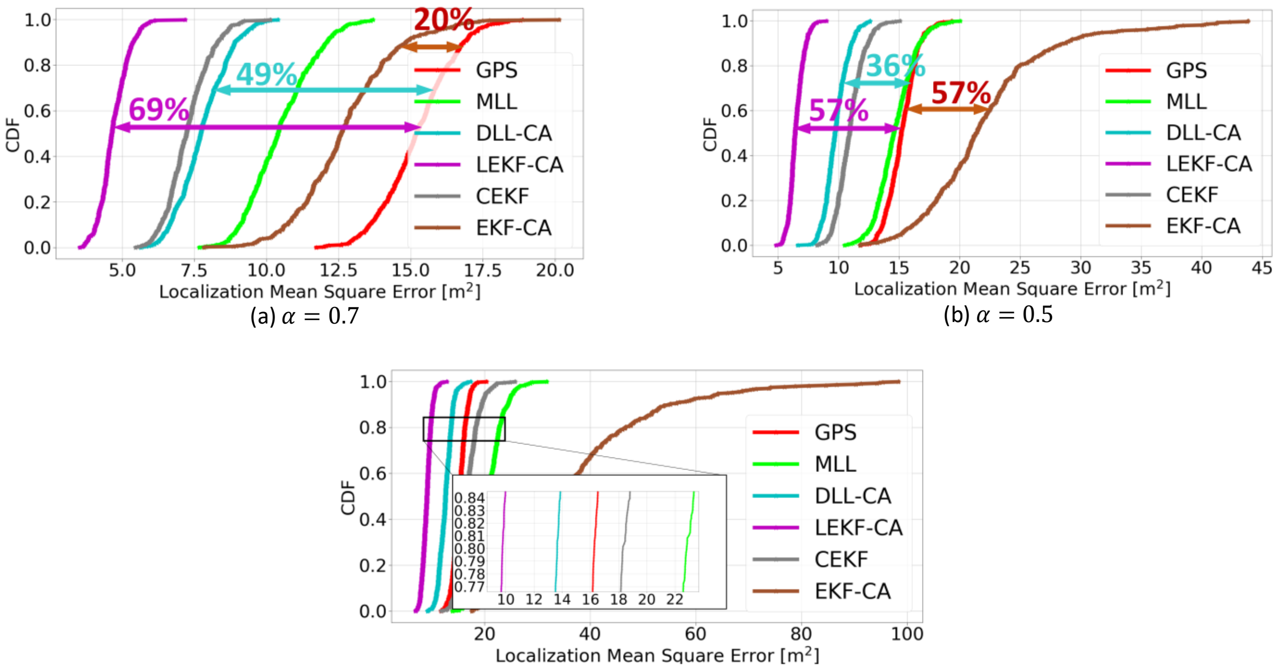
(f) $r_c = 30m$, not perf.data assoc.

*Figure 12: Impact of communication range $r_c$ and data association to the locations' estimation performance*

In all cases, it is clear that as $r_c$ increases so does the reduction of GPS LMSE attained by each one of the methods, since vehicles integrate a larger amount of information from their neighbors. For example, in Figure 12-(a) $\boldsymbol{LEKF-CA}$ achieved **65%** reduction of GPS LMSE, $\boldsymbol{DLL-CA}$ achieved **34%** reduction, $\boldsymbol{EKF-CA}$ achieved 52% reduction, $\boldsymbol{CEKF}$ achieved 57% reduction, while $\boldsymbol{MLL}$ achieved only 24% reduction. In Figure 12-(b), **LEKF-CA** achieved **77%** reduction of GPS LMSE, both $\boldsymbol{DLL-CA}$ and $\boldsymbol{EKF-CA}$ achieved **58%** reduction, $\boldsymbol{CEKF}$ achieved 65% reduction, while $\boldsymbol{MLL}$ achieved 46% reduction. Finally, in Figure 12-(c) $\boldsymbol{LEKF-CA}$ achieved **84%** reduction of GPS LMSE, $\boldsymbol{DLL-CA}$ achieved **71%** reduction, $\boldsymbol{EKF-CA}$ achieved 63% reduction, $\boldsymbol{CEKF}$ achieved 73% reduction, while $\boldsymbol{MLL}$ achieved 60% reduction. With non perfect data association (Figure 12-(d)-(f)), which is in fact a much more realistic scenario, performances of $\boldsymbol{LEKF-CA}$ and $\boldsymbol{DLL-CA}$ have been only slightly reduced. On the contrary, $\boldsymbol{EKF-CA}$ is seriously degraded as $r_c$ increases, due to the fact that data association is performed in two stages: both for the range measurements transmitted by the neighbors, as well as those produced by each ego vehicle. Therefore, the competitive advantage of both $\boldsymbol{LEKF-CA}$ and $\boldsymbol{DLL-CA}$, i.e. data association is required only for the transmitted range measurements, has a direct impact to their performances. It is evident that the proposed distributed tracking $\boldsymbol{LEKF-CA}$ significantly outperforms all the other methods, both the centralized, highlighting the benefits of multi-modal fusion in the context of compact differential coordinates (instead of pair-wise measurements) and the connectivity representation of CAV via the Graph Laplacian operator. We conclude that larger number of connected neighbors positively affects the performance of the two proposed distributed Laplacian schemes. In the following experiments, communication range equals to $r_c = 20m$ and non perfect data association will be assumed, unless mentioned otherwise.

2. <u>Noisy range measurements</u>

Range measurements is a key factor for the success of the proposed schemes since differential coordinates derive directly from them. Measurements from visual sensors like LIDAR and Cameras could be degraded due to occluded environments, illumination and viewpoint changes, diverse weather conditions, etc. To model these natural uncertainties but also to investigate the robustness of the proposed schemes, the relative distance and angle/azimuth angle models have been reformulated in Section 3.5.3 so as to capture both LOS and NLOS links between the involved CAV. We set $\mu_d^{NLOS} = 5m$, $\sigma_d^{NLOS} = 10m$, $\mu_a^{NLOS} = 8°$ and $\sigma_a^{NLOS} = 20°$. Additionally, three different values of the parameter $\alpha$ will be utilized, modelling moderate, high and severe NLOS conditions: $\alpha = \{0.7,0.5,0.3\}$. The effect of the range measurements uncertainties is depicted in Figure 13.



(a) $\alpha = 0.7$

(b) $\alpha = 0.5$

(c) $\alpha = 0.3$

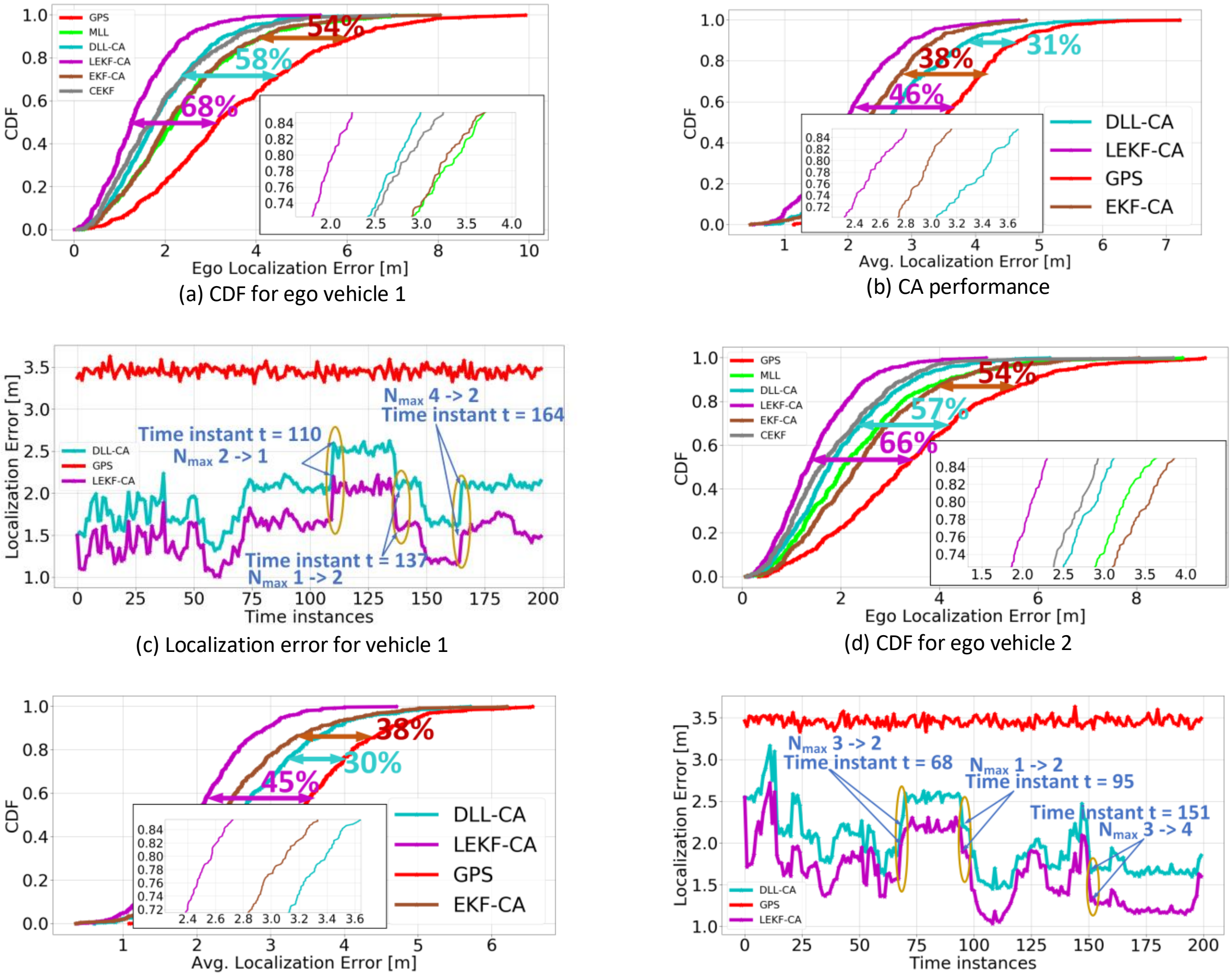
*Figure 13: Effect of LOS and NLOS range measurements links*

Accuracy has been decreased with respect to Figure 12-(e), especially $for\ \boldsymbol{EKF-CA}$. For instance, with moderate NLOS in Figure 13-(a) $\boldsymbol{LEKF-CA}$ achieved **69%** reduction of GPS LMSE, $\boldsymbol{DLL-CA}$ achieved **49%** reduction, while $\boldsymbol{EKF-CA}$ only 20%. Additionally, distributed $\boldsymbol{LEKF-CA}$ outperformed the centralized methods **CEKF** and **MLL**. In Figure 13-(b) with high NLOS, $\boldsymbol{LEKF-CA}$ and $\boldsymbol{DLL-CA}$ achieved **57%** and **36%** reduction, while **EKF-CA** totally failed to perform since it actually increased GPS LMSE by 57%. Both $\boldsymbol{LEKF-CA}$ and $\boldsymbol{DLL-CA}$ outperformed the centralized methods. Finally, with severe NLOS in Figure 13-(c) all three methods of $\boldsymbol{CEKF}$, $\boldsymbol{MLL}$ and $\boldsymbol{EKF-CA}$ significantly increased GPS LMSE, especially $\boldsymbol{EKF-CA}$. The proposed $\boldsymbol{LEKF-CA}$ and $\boldsymbol{DLL-CA}$ seem to be much more effective, since they attained **40%** and **19%** reduction. We conclude that range measurements highly contaminated by noise severely impacts the performance of CL and Tracking, especially for those methods who treat the range measurements independently and pair-wise. On the contrary, the proposed distributed Laplacian schemes have proven their robustness and effectiveness in a high enough level, explicitly due to the differential coordinates. The latter are able to reduce the measurements noise since they compute in fact the average of range measurements, resulting in much more accurate locations' estimation.

3. <u>Cooperative Awareness evaluation</u>

The accuracy of the discussed methods for ego vehicle case is shown in Figure 14-(a) and (d). For vehicle 1, once again **LEKF-CA** outperformed all others, both the centralized ones. In particular, **LEKF-CA** reduced GPS LE by **68%**, **DLL-CA** by **58%** and **EKF-CA** by 54%. The same are more or less apparent for ego vehicle 2. In terms of CA performance evaluation, we plotted the ALE of the three CA schemes in Figure 14-(b) and (e). For ego vehicle 2, **LEKF-CA** proved to be the most effective for this task. It actually reduced GPS ALE by **45%**, **DLL-CA** by **30%** and **ELF-CA** by 38%. The same behavior is also noticed for ego vehicle 1. For the major task of CA, upon which three methods were built, **LEKF-CA** is verified to be the most superior since it succeeds in significantly reducing both LE and ALE. Finally, in Figure 14-(c) and (f) we plotted for the first 200 time instances the averaged over 1000 iterations LE for the two vehicles. It is obvious that **LEKF-CA** captures the modifications of star topology, without the need of total re-initialization with **DLL-CA**. The proposed strategy of EKF's modification without reset is proved to be effective, whereas satisfies the property of scalability. For example, in Figure 14-(c) vehicle 1 has only one neighbor at $t = 136$. Neighbors were increased to two at $t = 137$, while **LEKF-CA** didn't need to be reset. Additionally, ego LE was reduced since larger amount of information was fused. By the same reasoning, ego LE was increased when the $N_{max}$ was reduced. Similar behavior is also noticed for vehicle 2. CARLA visualizations which contain the results of CA are shown in Figure 15-(c) and (f).

Clearly, **LEKF-CA** offers increased situational awareness to the vehicles, much better than simple GPS sensor.



(a) CDF for ego vehicle 1

(b) CA performance

(c) Localization error for vehicle 1

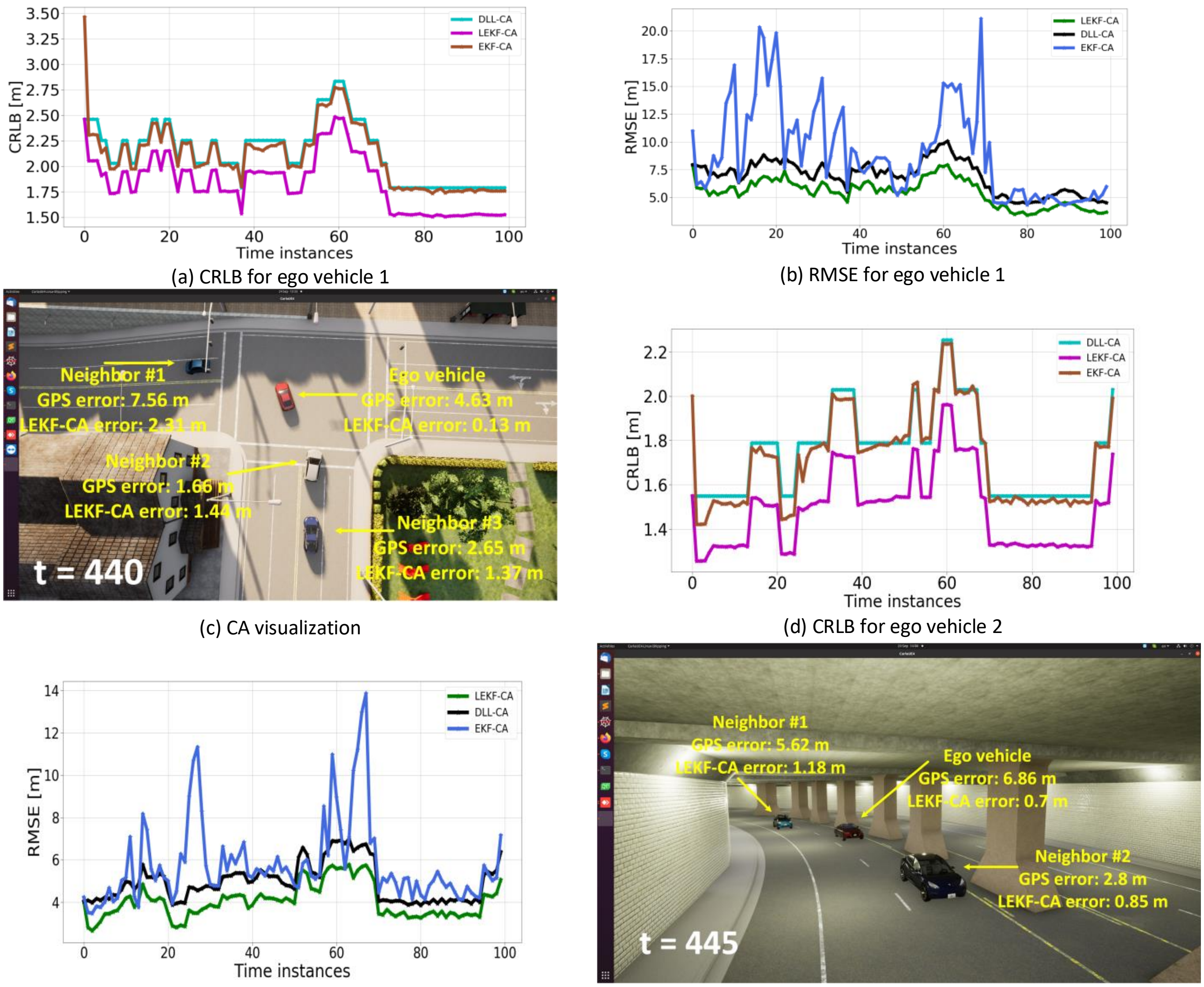
(d) CDF for ego vehicle 2

(e) CA performance

(f) Localization error for vehicle 2

*Figure 14: Location estimation accuracy for self and CA case*

It is evident from Figure 15-(a) and (d), that the minimum value of CRLB, following Eq. 8, has been attained by **LEKF-CA**, proving in a theoretical base its supremacy and effectiveness in terms of accuracy with respect to **EKF-CA** and **DLL-CA**. Variations of CRLB values are mainly due to the modifications of individual star topology. For a network of $M$ 2D nodes and as pointed out in [11], CRLB decreases (i.e. lower uncertainty about estimated positions) when the entering node/vehicle introduces at least $2M+1$ measurements. In our case, each vehicle needs to estimate $2(|\mathcal{N}_i^{(t)}|+1)$ positions. When a new vehicle enters the star topology, it introduces 2 range measurements and its 2D GPS position: a total of 4 values. Therefore, CRLB will be decreased if only $4 \geq 2(|\mathcal{N}_i^{(t)}|+1)+1$. Simply stated, it needs to be $|\mathcal{N}_i^{(t)}| \leq 0.5$ which is unfeasible. As a matter of fact, exactly due to the formulation of the problem, positioning estimation uncertainty increases/decreases when new vehicles enter/exit the star topology of ego vehicle. Finally, we plotted the RMSE over 500 iterations in Figure 15-(b) and (e). Clearly, **LEKF-CA** achieved once again the greatest performance, reaching very close to the theoretical bound. On the other hand, with **EKF-CA** the RMSE even exceeds $20m$, far away from its lower bound. The excess between theoretical CRLBs and practical RMSEs were caused by the measurements noise.



(a) CRLB for ego vehicle 1

(b) RMSE for ego vehicle 1

(c) CA visualization

(d) CRLB for ego vehicle 2

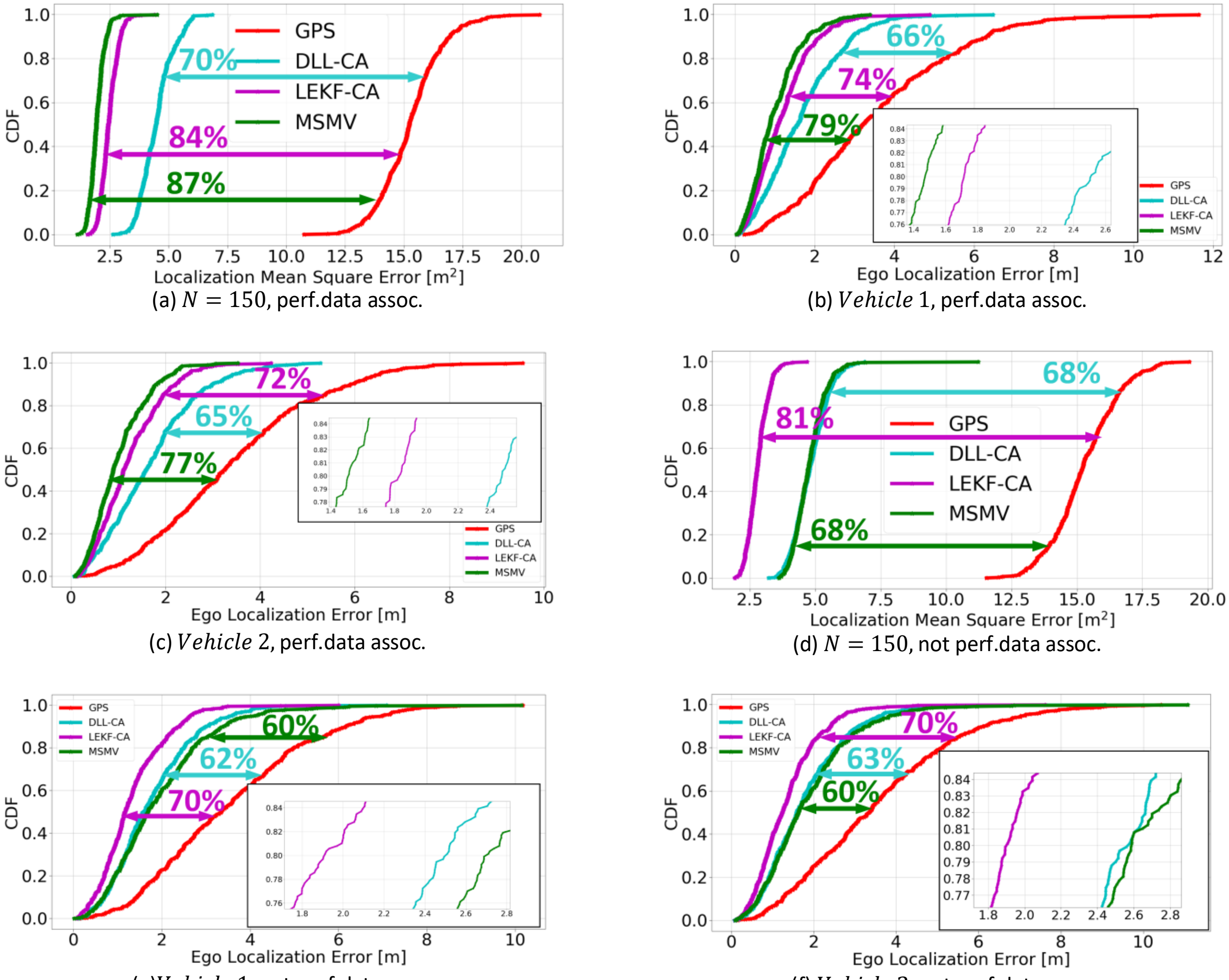
(e) RMSE for ego vehicle 2

(f) CA visualization

*Figure 15: CRLB, RMSE and CARLA visualization*

## 4. Comparison with MSMV method

To further assess proposed schemes' performances, state-of-the-art **MSMV** method [18] will act comparatively. This approach constructs a distributed framework for self-position estimation comprising actually of two types of measurements: i) self GPS and IMU sensors outputs, ii) GPS and range measurements transmitted by the connected neighboring vehicles. These two types of measurements represent in fact the position of ego vehicle, measured by i) itself and ii) its connected neighbors. For instance, consider ego vehicle $i$ and neighbor $j$. Position of $i$ measured and transmitted by $j$ is in the form of: $(\tilde{z}_{p,j}^{(x,t)} + \tilde{z}_{d,ji}^{(t)} \sin \tilde{z}_{a,ji}^{(t)}, \tilde{z}_{p,j}^{(y,t)} + \tilde{z}_{d,ji}^{(t)} \cos \tilde{z}_{a,ji}^{(t)})$. For each type of measurement, a local Kalman Filter is employed to provide a rough estimation of $i$-th position. Eventually, by combining the different positions estimations along with their associated covariance matrices, a global filter is formulated which performs the fusion in optimal manner due to the Gaussian measurement noise. A total number of $|\mathcal{N}_i^{(t)}| + 1$ KFs are utilized by the ego vehicle so as to provide an accurate position estimation. We have to point out that the discussed framework doesn't focus on CA task at all, as we do. Additionally, data association step is omitted and considered optimal when $j$ transmits to $i$ its own estimation, although using only LIDAR sensor the individual vehicles ids aren't distinguishable as we have detailed elaborated in Section 3.5.1.



(a) $N = 150$, perf.data assoc.

(b) $Vehicle$ 1, perf.data assoc.

(c) $Vehicle$ 2, perf.data assoc.

(d) $N = 150$, not perf.data assoc.

(e)$Vehicle$ 1, not perf.data assoc.

(f) $Vehicle$ 2, not perf.data assoc.

*Figure 16: MSMV evaluation*

Figure 16 presents the numerical evaluation with $\sigma_d = 1m, \sigma_a = 4°$ and $r_c = 30m$. The first row corresponds to the optimal data association. For instance, in Figure 16-(a) the overall LMSE for $N = 150$ vehicles is depicted. **MSMV** reduced GPS LMSE by 87%, while **LEKF-CA** and **DLL-CA** achieved **84%** and **70%** reduction, respectively. Additionally, the same behavior can be noticed for ego LE case. For ego vehicle 1 in Figure 16-(b), **MSMV** reduced GPS LE by 79%, while **LEKF-CA** and **DLL-CA** by **74%** and **66%**. The same goes also for vehicle 2 in Figure 16-(c). Clearly, **MSMV** exhibits slightly better performance than ours **LEKF-CA**. Nonetheless, in the much more realistic scenario of non optimal data association, the performance of **MSMV** is severely degraded (second row of Figure 16). For the overall LMSE in Figure 16-(d), **MSMV** exhibits 68% reduction of GPS LMSE, instead of **81%** with **LEKF-CA**. Furthermore, for vehicle 2 in Figure 16-(f) **MSMV** attains 60% reduction of GPS LE, instead of **70%** with **LEKF-CA**. By inspecting the two rows of Figure 16, we deduce that the impact of non optimal data association is much stronger to **MSMV**. This is due to the fact that the local KFs are employed independently for each type of measurements, thus increasing the potential footprint of range measurements uncertainty. It is proven once again that the compact and unified treatment of measurements performed by the Graph Laplacian schemes, offers high enough position estimation accuracy.

## 3.7 Demo video repository

An indicative demo video shown the accuracy of Cooperative Awareness for a random ego vehicle along with its neighbors in CARLA environment, is shown in:
https://drive.google.com/file/d/1wxy-41LTaBoWYj-KKPXchjn0lJyBiiRM/view?usp=sharing

The Python code required for producing the associated results is available in:
https://gitlab.com/isi_athena_rc/cpsosaware/cooperative-localization-and-tracking/ros_ekf

# 4 Global Cooperative Awareness through information diffusion[2]

Global Cooperative Awareness of CAVs, i.e., vehicles of a VANET to be aware of its own but also of its multi-hop (not just direct) neighboring vehicles states is explicitly discussed in this Section. To do so, we design distributed multi-modal localization approaches. We utilize information diffusion on graphs formed by moving vehicles, based on Adapt-then-Combine (ATC) strategies coupled with the Least-Mean-Squares (LMS) and the Conjugate Gradient (CG) algorithms. We treat the vehicular network as an undirected graph, where vehicles communicate with each other by means of Vehicle-to- Vehicle communication protocols. Connected vehicles perform cooperative fusion of different measurement modalities, including location and range measurements, in order to estimate both their positions and the positions of all other networked vehicles, by interacting only with their local neighborhood. The trajectories of vehicles were generated either by a well-known kinematic model, or by using the CARLA autonomous driving simulator. The various proposed distributed and diffusion localization schemes significantly reduce the GPS error and do not only converge to the global solution, but they even outperformed it. Extensive simulation studies highlight the benefits of the various methods, which in turn outperform other state of the art approaches. The impact of the network connections and the network latency are also investigated.

## 4.1 Preliminaries

Consider a network graph of $N$ nodes/vehicles (example depicted on Figure 17*), where each node $i$ ($i = 1, \dots N$) has access to a scalar measurement $d_i^{(t)} \in \mathbb{R}$ and a regression vector $\boldsymbol{u}_{\boldsymbol{i}}^{(\boldsymbol{t})} \in \mathbb{R}^N$ at time instant $t$ ($t = 1, \dots T$). Edges between nodes imply communication connection. It is assumed that each node follows a linear measurement model according to:*

$$d_i^{(t)} = \boldsymbol{u}_{\boldsymbol{i}}^{(\boldsymbol{t})\boldsymbol{T}} \boldsymbol{w}^{(\boldsymbol{t})} + n_i^{(t)} \qquad \text{Eq. 11}$$

As it will be thoroughly explained in the following Sections, the general model of Eq. 11 lends itself for cooperative localization applications, since $d_i^{(t)}, \boldsymbol{u}_{\boldsymbol{i}}^{(\boldsymbol{t})}$ and $\boldsymbol{w}^{(\boldsymbol{t})}$ refer to differential coordinate and V2V neighbors of vehicle $i$, as well as the positions of all VANET's vehicles, respectively. The network's objective is to optimally estimate the, common across the nodes, parameter vector $\boldsymbol{w}^{(\boldsymbol{t})} \in \mathbb{R}^N$. Note that $n_i^{(t)}$ is a zero mean spatially independent measurement noise

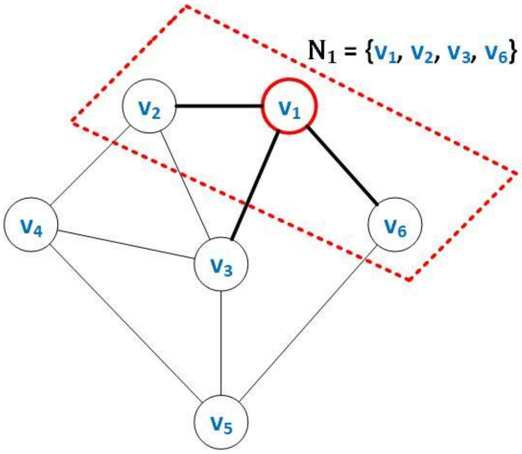


*Figure 17: Network of N nodes*

[2] N. Piperigkos, A. S. Lalos and K. Berberidis, "Graph Laplacian Diffusion Localization of Connected and Automated Vehicles," in *IEEE Transactions on Intelligent Transportation Systems*, doi: 10.1109/TITS.2021.3110650.

, with variance $\sigma_n^2$. To this end, the following global optimization problem can be defined:

$$argmin_{\boldsymbol{w}^{(t)}} J\left(\boldsymbol{w}^{(t)}\right) = argmin_{\boldsymbol{w}^{(t)}} \sum_{i=1}^{N} \left(d_i^{(t)} - \boldsymbol{u}_{\boldsymbol{i}}^{(t)\boldsymbol{T}} \boldsymbol{w}^{(t)}\right)^2 \tag*{\textit{Eq. 12}}$$

It is proven that the optimal vector at time instant $t$ is equal to:

$$\boldsymbol{w}^{(\boldsymbol{t})} = \left(\sum_{i=1}^{N} \boldsymbol{R}_{\boldsymbol{i}}^{(\boldsymbol{t})}\right)^{-1} \sum_{i=1}^{N} \boldsymbol{r}_{\boldsymbol{i}}^{(\boldsymbol{t})}, \tag*{\textit{Eq. 13}}$$

with positive semi-definite covariance matrix $\boldsymbol{R}_{\boldsymbol{i}}^{(\boldsymbol{t})} = \boldsymbol{u}_{\boldsymbol{i}}^{(\boldsymbol{t})} \boldsymbol{u}_{\boldsymbol{i}}^{(\boldsymbol{t})\boldsymbol{T}}$ and cross correlation vector $\boldsymbol{r}_{\boldsymbol{i}}^{(\boldsymbol{t})} = \boldsymbol{u}_{\boldsymbol{i}}^{(\boldsymbol{t})\boldsymbol{T}} \boldsymbol{d}_{\boldsymbol{i}}^{(\boldsymbol{t})}$. As it may be obvious, Eq. 13 is actually a centralized (least-squares) implementation since all nodes have to broadcast their measurements and regression vectors to an overall fusion center, which in turn will estimate parameter $\boldsymbol{w}^{(\boldsymbol{t})}$ and inform nodes about it. However, it is more robust and cost effective for each node to estimate $\boldsymbol{w}^{(\boldsymbol{t})}$ in a distributed manner and on its own, relying only to its connected neighbors. Distributed implementation of Eq. 12, which the proposed Graph Laplacian methods will be derived from, is discussed in the following subsection.

A gradient descent based solution will be employed to address the distributed solution of Eq. 12. For a total number of iterations, say $K$, the optimal vector at each iteration $k$ is given by:

$$\boldsymbol{w}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{k}+\boldsymbol{1})} = \boldsymbol{w}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{k})} - \mu^{(t)} \frac{\partial J\left(\boldsymbol{w}^{(\boldsymbol{t},\boldsymbol{k})}\right)}{\partial \boldsymbol{w}^{(\boldsymbol{t},\boldsymbol{k})}}, \tag*{\textit{Eq. 14}}$$

where $\frac{\partial J\left(\boldsymbol{w}^{(\boldsymbol{t},\boldsymbol{k})}\right)}{\partial \boldsymbol{w}^{(\boldsymbol{t},\boldsymbol{k})}} = \sum_{i=1}^{N} \left(\boldsymbol{R}_{\boldsymbol{i}}^{(\boldsymbol{t})} \boldsymbol{w}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{k})} - \boldsymbol{r}_{\boldsymbol{i}}^{(\boldsymbol{t})}\right)$ and small scalar step size $\mu^{(t)} > 0$. Thus,Eq. 14 is given by:

$$\boldsymbol{w}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{k}+\boldsymbol{1})} = \boldsymbol{w}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{k})} - \mu^{(t)} \sum_{i=1}^{N} \boldsymbol{u}_{\boldsymbol{i}}^{(\boldsymbol{t})\boldsymbol{T}} \left(d_i^{(t)} - \boldsymbol{u}_{\boldsymbol{i}}^{(\boldsymbol{t})} \boldsymbol{w}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{k})}\right) \tag*{\textit{Eq. 15}}$$

Although Eq. 15 is not a distributed implementation since data across the whole network are required, it motivated the development of distributed ATC diffusion LMS algorithm [19]:

$$\boldsymbol{\psi}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{k}+\boldsymbol{1})} = \boldsymbol{w}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{k})} - \mu_i^{(t)} \boldsymbol{u}_{\boldsymbol{i}}^{(\boldsymbol{t})\boldsymbol{T}} \left(d_i^{(t)} - \boldsymbol{u}_{\boldsymbol{i}}^{(\boldsymbol{t})} \boldsymbol{w}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{k})}\right) \tag*{\textit{Eq. 16}}$$

$$\boldsymbol{w}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{k}+\boldsymbol{1})} = \sum_{l \in \mathcal{N}_i^{(t)}} c_{il}^{(t)} \boldsymbol{\psi}_{\boldsymbol{l}}^{(\boldsymbol{t},\boldsymbol{k}+\boldsymbol{1})} \tag*{\textit{Eq. 17}}$$

The neighborhood of node $i$ is the set $\mathcal{N}_i^{(t)}$ with cardinality $|\mathcal{N}_i^{(t)}|$, consisting of self and neighbouring nodes. During the adaptation step of (6), each node $i$ estimates in parallel the intermediate vector $\boldsymbol{\psi}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{k}+\boldsymbol{1})}$, based on the previously estimated vector $\boldsymbol{w}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{k})}$ and the pair $\{d_i^{(t)}, \boldsymbol{u}_{\boldsymbol{i}}^{(\boldsymbol{t})}\}$. During the combination step of Eq. 17, each node $i$ again in parallel, receives the intermediate vectors from its neighbourhood and convexly combines them, in order to estimate the common parameter vector. That last step is critical for feasible estimation of the parameter vector. Actually without it, the node is unable to estimate accurately the desired vector. Combination weights $c_{il}^{(t)}$ define the combination matrix $\boldsymbol{C}^{(\boldsymbol{t})} \in \mathbb{R}^{N \times N}$. A typical choice for combination weights is based on the Metropolis rule [20].

A variant of ATC has also been proposed, where a convex combination operation is added to the adaptation step, leading to ATC with measurements exchanges diffusion LMS:

$$\boldsymbol{\psi}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{k}+\boldsymbol{1})} = \boldsymbol{w}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{k})} - \mu_i^{(t)} \sum_{l \in \mathcal{N}_i^{(t)}} c_{il}^{(t)} \boldsymbol{u}_{\boldsymbol{l}}^{(\boldsymbol{t})\boldsymbol{T}} \left(d_l^{(t)} - \boldsymbol{u}_{\boldsymbol{l}}^{(\boldsymbol{t})} \boldsymbol{w}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{k})}\right) \tag*{\textit{Eq. 18}}$$

$$\boldsymbol{w}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{k}+\boldsymbol{1})} = \sum_{l\in\mathcal{N}_i^{(t)}} c_{il}^{(t)} \boldsymbol{\psi}_{\boldsymbol{l}}^{(\boldsymbol{t},\boldsymbol{k}+\boldsymbol{1})} \qquad \textit{Eq. 19}$$

Notice from Eq. 18 that each node receives now the pair $\{d_l^{(t)}, \boldsymbol{u}_{\boldsymbol{l}}^{(\boldsymbol{t})}\}$ from its neighbours, adding one more communication and exchange step to the diffusion algorithm. The LMS diffusion strategy of ATC with measurements exchanges for node v $_1$ is describing on Figure 18. Consequently, each node estimates the common parameter vector in a distributed manner, hence avoiding the heavy computational burden of a centralized processing architecture.

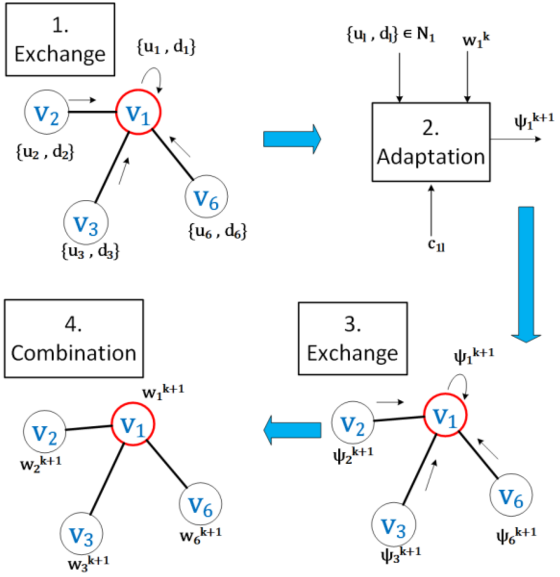


*Figure 18: ATC with measurements exchange*

Before we proceed with describing the proposed distributed and diffusion localization strategies, we will shortly review our previous work on ***Centralized Laplacian Localization*** (***CLL***) [10], [21], [22]. The latter, a graph based approach for Cooperative Localization (CL), performs cooperative multi-modal fusion, exploiting the spatial coherences and connectivity properties of the vehicles of a vehicular network. Furthermore, a measurement model fusing GPS positions, range measurements and the topology of VANET using the linear Graph Laplacian operator, is utilized. The main assumption of that approach relies on the feasible estimation of the so-called differential coordinates, by encoding each vehicle's position relative to its neighbouring.

We define the Laplacian matrix of VANET graph $\boldsymbol{L}^{(\boldsymbol{t})} \in \mathbb{R}^{N\times N}$ as $\boldsymbol{L}^{(\boldsymbol{t})} = \boldsymbol{D}^{(\boldsymbol{t})} - \boldsymbol{V}^{(\boldsymbol{t})}$, where $\boldsymbol{D}^{(\boldsymbol{t})}, \boldsymbol{V}^{(\boldsymbol{t})} \in \mathbb{R}^{N\times N}$ are the degree and adjacency matrices of VANET graph. The differential coordinates per vehicle $\boldsymbol{\delta}_{\boldsymbol{i}}^{(\boldsymbol{t})} = \begin{bmatrix}\delta_i^{(t,x)} & \delta_i^{(t,y)}\end{bmatrix}^T \in \mathbb{R}^2$ are equal to:

$$\delta_i^{(t,\mathrm{x})} = \frac{1}{|\mathcal{N}_i^{(t)}|-1} \sum_{l\in\mathcal{N}_i^{(t)}} \left(x_i^{(t)} - x_l^{(t)}\right) = \frac{1}{|\mathcal{N}_i^{(t)}|-1} \sum_{l\in\mathcal{N}_i^{(t)}} \left(-\tilde{z}_{d,il}^{(t)} sin\tilde{z}_{az,il}^{(t)}\right)$$

and

$$\delta_i^{(t,y)} = \frac{1}{|\mathcal{N}_i^{(t)}|-1} \sum_{l\in\mathcal{N}_i^{(t)}} \left(y_i^{(t)} - y_l^{(t)}\right) = \frac{1}{|\mathcal{N}_i^{(t)}|-1} \sum_{l\in\mathcal{N}_i^{(t)}} \left(-\tilde{z}_{d,il}^{(t)} \cos\tilde{z}_{az,il}^{(t)}\right)$$

By creating the vector $\boldsymbol{\delta}^{(\boldsymbol{t},\boldsymbol{x})} \in \mathbb{R}^N$ ($x$-differentials), the $x$-coordinates position of vehicles (as vector $\boldsymbol{x}^{(\boldsymbol{t})} \in \mathbb{R}^N$) follow the linear measurement model:

$$\boldsymbol{L}^{(t)}\boldsymbol{x}^{(t)} = \boldsymbol{D}^{(t)}\boldsymbol{\delta}^{(t,x)} \qquad \textit{Eq. 20}$$

However, $\boldsymbol{L}^{(t)}$ is positive semi-definite and non-invertible [23], and as such Eq. 20 is reformulated by creating the extended Laplacian matrix $\tilde{\boldsymbol{L}}^{(t)} \in \mathbb{R}^{2N\times N}$ as $\tilde{\boldsymbol{L}}^{(t)} = \begin{bmatrix} \boldsymbol{L}^{(t)} \\ \mathbb{I}_N \end{bmatrix}$ and vector $\boldsymbol{s}^{(t,x)} \in \mathbb{R}^{2N}$ as $\boldsymbol{s}^{(t,x)} = \begin{bmatrix} \boldsymbol{D}^{(t)}\boldsymbol{\delta}^{(t,x)} \\ \tilde{\boldsymbol{z}}_{\boldsymbol{p}}^{(t,x)} \end{bmatrix}$, where $\tilde{\boldsymbol{z}}_{\boldsymbol{p}}^{(t,x)} \in \mathbb{R}^N$ are the GPS $x$-positions.The latter act as anchor points [6]. Therefore, Eq. 20 transforms to:

$$\tilde{\boldsymbol{L}}^{(t)}\boldsymbol{x}^{(t)} = \boldsymbol{s}^{(t,x)} \qquad \textit{Eq. 21}$$

which can be solved in the least-squares sense and estimate the $x$-positions of vehicles. Obviously, the same approach is followed for $y$-positions $\boldsymbol{y}^{(t)} \in \mathbb{R}^N$. Note that we make an explicit statement about data association, i.e. for every range measurement ($\tilde{z}_{d,il}^{(t)}$ or $\tilde{z}_{az,il}^{(t)}$) of vehicle $i$, the latter knows which vehicle $l$ is associated with. Although it is possible to model the associations by comparing range measurements with GPS positions of neighbors, we assume that data association is given to us as part of a pre-processing step. Thus, **CLL** assumes that communication and sensing graph are identical.

However, Eq. 21 is a centralized implementation, since the vehicles are required to broadcast and send their ids and measurements to a central node/fusion center (e.g. 5G cloud), which in turn will estimate and inform them about their positions. This is exactly what should be avoided due to serious limitations and drawbacks of central node operation like malfunctioning or cyber-attack. **CLL** will act as a baseline to the development of fully distributed solutions for CL.

## 4.2 Graph Laplacian LMS

It is important to notice the resemblance between Eq. 20 and Eq. 11. In both cases, the vector to be estimated is linearly correlated with measurements and regression vectors. In the localization framework the differential coordinates, divided by nodes' degrees, act as the measurements, while the rows of Laplacian matrix as the corresponding regression vectors. Subsequently, for each vehicle the following linear measurement model (for $x$-coordinates) is defined:

$$\delta_i^{(t,x)} = \boldsymbol{L}_{\boldsymbol{i}:}^{(t)\boldsymbol{T}}\boldsymbol{w}_{\boldsymbol{i},\boldsymbol{x}}^{(t)} \qquad \textit{Eq. 22}$$

where the unknown vector $\boldsymbol{w}_{\boldsymbol{i},\boldsymbol{x}}^{(t)} \in \mathbb{R}^N$ corresponds to the $x$-coordinates position of vehicles, estimated by vehicle $i$. Apparently, the optimal vector is common across the vehicles of the network. Based on that local model, each vehicle is totally unable to estimate its position and all others on its own. Only by means of cooperation and information diffusion through V2V and 5G related protocols, the vehicles can learn the desired vector. As a natural consequence of Eq. 16 and Eq. 17, the two steps of the proposed $\boldsymbol{Graph\ Laplacian\ LMS}$ or $\boldsymbol{GLLMS}$ approach are derived:

$$\boldsymbol{\psi}_{\boldsymbol{i},\boldsymbol{x}}^{(\boldsymbol{t},\boldsymbol{k}+\boldsymbol{1})} = \boldsymbol{w}_{\boldsymbol{i},\boldsymbol{x}}^{(\boldsymbol{t},\boldsymbol{k})} - \mu_{1,i}^{(t)}\boldsymbol{L}_{\boldsymbol{i}:}^{(t)\boldsymbol{T}}\left(\delta_i^{(t,x)} - \boldsymbol{L}_{\boldsymbol{i}:}^{(t)}\boldsymbol{w}_{\boldsymbol{i},\boldsymbol{x}}^{(\boldsymbol{t},\boldsymbol{k})}\right) \qquad \textit{Eq. 23}$$

$$\boldsymbol{w}_{\boldsymbol{i},\boldsymbol{x}}^{(\boldsymbol{t},\boldsymbol{k}+\boldsymbol{1})} = \sum_{l\in\mathcal{N}_i^{(t)}} c_{il}^{(t)}\boldsymbol{\psi}_{\boldsymbol{l},\boldsymbol{x}}^{(\boldsymbol{t},\boldsymbol{k}+\boldsymbol{1})} \qquad \textit{Eq. 24}$$

At the adaptation step of Eq. 23, each vehicle estimates in parallel the intermediate vectors, using the pair $\{\delta_i^{(t,x)}, \boldsymbol{L}_{\boldsymbol{i}:}^{(t)\boldsymbol{T}}\}$, while in the combination step of Eq. 24, receives from and sends to the neighborhood the intermediate vectors, in order to estimate the desired vectors. At the end of that procedure, each vehicle will estimate and converge to the same location vector. During the initialization stage at time instant $t$, the desired vector is set to the GPS positions of the vehicles, as a rough estimation of the solution vector. Vehicles via, e.g. TDMA protocol, are in fact informed about the noisy locations of all VANET's vehicles (not just neighbors), in order to successively estimate their positions. The $\boldsymbol{GLLMS}$ approach for $x$-coordinates is summarized on **Algorithm 5**.

***Algorithm 5: Graph Laplacian LMS or GLLMS***

```
Input: $N, T, K, \tilde{\boldsymbol{z}}_{\boldsymbol{p}}^{(\boldsymbol{t},\boldsymbol{x})}, \delta_i^{(t,x)}, \boldsymbol{L}_{\boldsymbol{i}:}^{(\boldsymbol{t})}, \boldsymbol{C}^{(\boldsymbol{t})}$
Output: $\boldsymbol{w}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{x})} \in \mathbb{R}^N, i \in N$
For $t = 1, \dots T$ do
        For each vehicle $i\ in\ parallel$ do
                $\boldsymbol{w}_{\boldsymbol{i},\boldsymbol{x}}^{(\boldsymbol{t},\boldsymbol{1})} = \tilde{\boldsymbol{z}}_{\boldsymbol{p}}^{(\boldsymbol{t},\boldsymbol{x})}$ ;
                For $k = 1, \dots K$ do
                        $\mu_{1,i}^{(t)}$ from Eq. 29;
                        $\boldsymbol{\psi}_{\boldsymbol{i},\boldsymbol{x}}^{(\boldsymbol{t},\boldsymbol{k}+\boldsymbol{1})} = \boldsymbol{w}_{\boldsymbol{i},\boldsymbol{x}}^{(\boldsymbol{t},\boldsymbol{k})} - \mu_{1,i}^{(t)} \boldsymbol{L}_{\boldsymbol{i}:}^{(\boldsymbol{t})} \left(\delta_i^{(t,x)} - \boldsymbol{L}_{\boldsymbol{i}:}^{(\boldsymbol{t})} \boldsymbol{w}_{\boldsymbol{i},\boldsymbol{x}}^{(\boldsymbol{t},\boldsymbol{k})}\right)$ ;
                        $\boldsymbol{w}_{\boldsymbol{i},\boldsymbol{x}}^{(\boldsymbol{t},\boldsymbol{k}+\boldsymbol{1})} = \sum_{i \in \mathcal{N}_i^{(t)}} c_{il}^{(t)} \boldsymbol{\psi}_{\boldsymbol{l},\boldsymbol{x}}^{\boldsymbol{t},\boldsymbol{k}+\boldsymbol{1}}$;
                End
        End
End
```

It is worth mentioning that each vehicle is interested enough to not only estimate accurately its own position, but also its direct neighborhood and all others vehicles of the network. Since individual vehicles would estimate on its own the entire location vector, the distributed and diffusion localization will facilitate the design of an efficient individual path planning and control mechanism. The latter will determine the best possible future driving actions, improving the overall performance of VANET in terms of e.g. reduced traffic accidents and fuel consumption.

## 4.3 Graph Laplacian Conjugate Gradient for improved cooperative awareness

The previous distributed and diffusion localization method converges each time instant to the global solution at the cost of high enough total number of iterations. This is a serious limitation towards real-time implementation, since the vehicles are required to estimate their positions before the new GPS measurement arrive, namely between 100-300 ms. Since vehicles are connected through V2V, it is expected that an additional measurement exchange step, will speed up the convergence of diffusion. Usually, CG algorithm is employed to accelerate LMS type approaches. The acceleration is attained due to the fact that CG selects the successive direction vectors towards solution as a conjugate version of the successive gradients obtained at each step. The conjugate directions are not specified beforehand, but rather are determined sequentially at each step of the iteration. The directions are based on the gradients, therefore the process makes good uniform progress toward the solution at every step. Furthermore, no line searching is required at any stage as in LMS.

We define matrix $\boldsymbol{U}_{\boldsymbol{i}}^{(\boldsymbol{t})} \in \mathbb{R}^{|\mathcal{N}_i^{(t)}| \times N}$ and vector $\boldsymbol{q}_{\boldsymbol{i}} \in \mathbb{R}^{|\mathcal{N}_i^{(t)}|}$ by putting together the rows of Laplacian matrix and differential coordinates, respectively, which belong to the neighborhood $\mathcal{N}_i^{(t)}$. As a matter of fact, neighboring vehicles broadcast the pair $\{\boldsymbol{L}_{\boldsymbol{l}:}^{(\boldsymbol{t})\boldsymbol{T}}, \delta_l^{(t,x)}\}$, vehicle $i$ receives them and together with its own data, defines $\{\boldsymbol{U}_{\boldsymbol{i}}^{(\boldsymbol{t})}, \boldsymbol{q}_{\boldsymbol{i}}^{(\boldsymbol{t})}\}$. The first row/element of the pair corresponds to $i$-th vehicle. Thus, the following linear problem can be defined:

$$\boldsymbol{U}_{\boldsymbol{i}}^{(\boldsymbol{t})} \boldsymbol{w}_{\boldsymbol{i},\boldsymbol{x}}^{(\boldsymbol{t})} = \boldsymbol{q}_{\boldsymbol{i}}^{(\boldsymbol{t})} \qquad \textit{Eq. 25}$$

Furthermore, if we utilize the instantaneous positive semi-definite covariance matrix $\boldsymbol{A}_{\boldsymbol{i}}^{(\boldsymbol{t})} = \boldsymbol{U}_{\boldsymbol{i}}^{(\boldsymbol{t})\boldsymbol{T}} \boldsymbol{U}_{\boldsymbol{i}}^{(\boldsymbol{t})}$, $\boldsymbol{A}_{\boldsymbol{i}}^{(\boldsymbol{t})} \in \mathbb{R}^{N \times N}$ and covariance vector $\boldsymbol{b}_{\boldsymbol{i}}^{(\boldsymbol{t})} = \boldsymbol{U}_{\boldsymbol{i}}^{(\boldsymbol{t})\boldsymbol{T}} \boldsymbol{q}_{\boldsymbol{i}}^{(\boldsymbol{t})}$, $\boldsymbol{b}_{\boldsymbol{i}}^{(\boldsymbol{t})} \in \mathbb{R}^N$, we end up (as a direct sequence of Eq. 25) to the following linear problem:

$$\boldsymbol{A}_{\boldsymbol{i}}^{(\boldsymbol{t})} \boldsymbol{w}_{\boldsymbol{i},\boldsymbol{x}}^{(\boldsymbol{t})} = \boldsymbol{b}_{\boldsymbol{i}}^{(\boldsymbol{t})} \qquad \textit{Eq. 26}$$

The CG optimization method is used to tackle Eq. 26, avoiding the high complexity and unstable performance of RLStype methods. Its main idea lies on the fact that a set of direction vectors conjugate to $\boldsymbol{A}_i^{(t)}$ are exploited to estimate the desired optimal vector $\boldsymbol{w}_{i,x}^{(t)}$.

However, Eq. 26 corresponds also to the minimization of cost function [24] $\mathcal{V}\left(\boldsymbol{w}_{i,x}^{(t)}\right) = \mathbb{E}\{(\boldsymbol{q}_i^{(t)})^2\} - \boldsymbol{b}_i^{(t)T}\boldsymbol{w}_{i,x}^{(t)} - \boldsymbol{w}_{i,x}^{(t)T}\boldsymbol{b}_i^{(t)} + \boldsymbol{w}_{i,x}^{(t)T}\boldsymbol{A}_i^{(t)}\boldsymbol{w}_{i,x}^{(t)}$. The main steps of the proposed $\boldsymbol{Graph\ Laplacian\ CG}$ or $\boldsymbol{GLCG}$ are summarized on **Algorithm 6**.

***Algorithm 6:*** $\boldsymbol{Graph\ Laplacian\ CG\ or\ GLCG}$

```
Input: $N, T, K, \tilde{\boldsymbol{z}}_{\boldsymbol{p}}^{(t,x)}, \delta_i^{(t,x)}, \boldsymbol{U}_i^{(t)}, \boldsymbol{q}_i^{(t)}, \boldsymbol{C}^{(t)}, \lambda$
Output: $\boldsymbol{w}_i^{(t,x)} \in \mathbb{R}^N, i \in N$
For $t = 1, \dots T$ do
        For each vehicle $i\ in\ parallel$ do
                $\boldsymbol{w}_{i,x}^{(t,1)} = \tilde{\boldsymbol{z}}_{\boldsymbol{p}}^{(t,x)}$ ;
                $\boldsymbol{A}_i^{(t)} = \boldsymbol{U}_i^{(t)T}\boldsymbol{U}_i^{(t)} + 10^{-7}\mathbb{I}_N$;
                $\boldsymbol{b}_i^{(t)} = \boldsymbol{U}_i^{(t)T}\boldsymbol{q}_i^{(t)}$;
                $\boldsymbol{g}_i^{(1)} = \boldsymbol{b}_i^{(t)} - \boldsymbol{A}_i^{(t)}\boldsymbol{w}_{i,x}^{(t,1)}$;
                $\boldsymbol{r}_i^{(2)} = \boldsymbol{g}_i^{(1)}$
                For $k = 1, \dots K$ do
                        $a_i = \dfrac{\boldsymbol{r}_i^{(k+1)T}\boldsymbol{g}_i^{(k)}}{\boldsymbol{r}_i^{(k+1)T}\boldsymbol{A}_i^{(t)}\boldsymbol{r}_i^{(k+1)}}$;
                        $\boldsymbol{\psi}_{i,x}^{(t,k+1)} = \boldsymbol{w}_{i,x}^{(t,k)} + a_i\boldsymbol{r}_i^{(k+1)}$; ;
                        $\widetilde{\boldsymbol{g}}_i^{(k)} = \lambda\boldsymbol{g}_i^{(k)} + \boldsymbol{b}_i^{(k)} - \boldsymbol{A}_i^{(t)}\boldsymbol{w}_{i,x}^{t,k}$;
                        $\boldsymbol{g}_i^{(k+1)} = \widetilde{\boldsymbol{g}}_i^{(k)} + a_i\boldsymbol{r}_i^{(k+1)}$;
                        $\beta_i = \min\left(\dfrac{\left(\boldsymbol{g}_i^{(k+1)} - \boldsymbol{g}_i^{(k)}\right)^T\boldsymbol{g}_i^{(k+1)}}{\boldsymbol{g}_i^{(k)T}\boldsymbol{g}_i^{(k)}}, \dfrac{\boldsymbol{g}_i^{(k+1)T}\boldsymbol{g}_i^{(k+1)}}{\boldsymbol{g}_i^{(k)T}\boldsymbol{g}_i^{(k)}}\right)$;
                        $\boldsymbol{r}_i^{(k+2)} = \boldsymbol{g}_i^{(k+1)} + \beta_i\boldsymbol{r}_i^{(k+1)}$;
                        $\boldsymbol{w}_{i,x}^{(t,k+1)} = \sum_{j\in\mathcal{N}_i^{(t)}} c_{il}^{(t)}\boldsymbol{\psi}_{i,x}^{(t,k+1)}$ ;
                End
        End
End
```

Optimal step sizes $\alpha_i$ are computed as the minimizing arguments of $\mathcal{V}$, factors $\beta_i$ are to be ensure $\boldsymbol{A}_i^{(t)}$-orthogonality for the direction vectors $\boldsymbol{r}_i$, while $\boldsymbol{g}_i$ is the negative gradient of $\mathcal{V}$. Factors $\beta_i$ are chosen as the minimum between the Polak-Ribiere formula $\beta_i^{PR} = \frac{(\boldsymbol{g}_i^{(k+1)} - \boldsymbol{g}_i^{(k)})^T\boldsymbol{g}_i^{(k+1)}}{\boldsymbol{g}_i^{(k)T}\boldsymbol{g}_i^{(k)}}$ and Fletcher-Reeves $\beta_i^{FR} = \frac{\boldsymbol{g}_i^{(k+1)T}\boldsymbol{g}_i^{(k+1)}}{\boldsymbol{g}_i^{(k)T}\boldsymbol{g}_i^{(k)}}$, in order to avoid anomalous behaviour and numerical instability. A forgetting factor $0 < \lambda < 1$ is employed to update the instantaneous covariance vector. We choose $\lambda = 0.2$. Note that small factor $10^{-7} \cdot \mathbb{I}_N$ has been added to $\boldsymbol{A}_i^{(t)}$, since the latter is in fact an ill-conditioned and low-rank matrix, as a product of rows of singular Laplacian matrix. This a serious limitation of the optimization method, since the convergence speed is determined by the condition number $\kappa(\boldsymbol{A}_i^{(t)}) = \kappa(\boldsymbol{U}_i^{(t)T}\boldsymbol{U}_i^{(t)}) = \kappa(\boldsymbol{U}_i^{(t)})^2$: the larger $\kappa(\boldsymbol{A}_i^{(t)})$ is, the slower the improvement towards solution [25]. However, by means of information diffusion, GLCG finally succeeds to converge to the optimal solution vector. Note that if ,$\boldsymbol{U}_i^{(t)}$ were chosen to be ,$\boldsymbol{L}_{i:}^{(t)}$, i.e. avoiding to use measurements exchanges, then ,$\boldsymbol{A}_i^{(t)}$ would be rank-one matrix, almost prohibitive to be used in estimating the location vector.

Therefore, a novel distributed localization scheme based on ATC framework and CG optimization with measurements exchanges has been developed. Each vehicle creates the matrix/vector $\{\boldsymbol{U}_{\boldsymbol{i}}^{(\boldsymbol{t})}, \boldsymbol{q}_{\boldsymbol{i}}^{(\boldsymbol{t})}\}$, by receiving the transmitted $\{\boldsymbol{L}_{\boldsymbol{l}:}^{(\boldsymbol{t})\boldsymbol{T}}, \delta_l^{(t,x)}\}$ from its neighborhood. Afterwards, it estimates the intermediate vectors using CG method and exploits its benefits. Finally, it estimates the desired location vector by a convex combination of neighboring intermediate vectors.

Although the communication burden is now increased, it can be performed in efficient manner, since each vehicle has to broadcast only a scalar value and a sparse vector with non-zero integer entries equal to $|\mathcal{N}_i^{(t)}|$. Laplacian matrix is actually sparse, since vehicles are connected only to a subset of operating vehicles. The measurements exchanges step has a significant impact on the convergence speed of the proposed schemes.

## 4.4 Graph Laplacian LMS for improved cooperative awareness

The main limitation of **GLCG** is related to the ill-conditioned $\boldsymbol{A}_{\boldsymbol{i}}^{(\boldsymbol{t})}$, which may deteriorate the performance of diffusion, as well as optimal step sizes $\alpha_i$ and $\beta_i$. Since the latter are estimated directly from available data, they are also vulnerable to increased noisy data, coming either from uncertain measurements or network latency (past estimations treated as current). As such, a variant of $\boldsymbol{GLCG}$ countering those drawbacks, which in addition will act comparatively, should be developed. Diffusion LMS solution of Eq. 25 derives exactly from Eq. 18 and Eq. 19, formulating the proposed method of $\boldsymbol{Graph\ Laplacian\ LMS\ with\ measurements\ exchanges}$ or $\boldsymbol{GLLME}$:

$$\boldsymbol{\psi}_{\boldsymbol{i,x}}^{(\boldsymbol{t,k+1})} = \boldsymbol{w}_{\boldsymbol{i,x}}^{(\boldsymbol{t,k})} - \mu_{2,i}^{(t)} \sum_{l\in\mathcal{N}_i^{(t)}} c_{il}^{(t)} \boldsymbol{L}_{\boldsymbol{l}:}^{(\boldsymbol{t})\boldsymbol{T}} \left(\delta_l^{(t,x)} - \boldsymbol{L}_{\boldsymbol{l}:}^{(\boldsymbol{t})} \boldsymbol{w}_{\boldsymbol{i,x}}^{(\boldsymbol{t,k})}\right) \qquad \textit{Eq. 27}$$

$$\boldsymbol{w}_{\boldsymbol{i,x}}^{(\boldsymbol{t,k+1})} = \sum_{l\in\mathcal{N}_i^{(t)}} c_{il}^{(t)} \boldsymbol{\psi}_{\boldsymbol{l,x}}^{(\boldsymbol{t,k+1})} \qquad \textit{Eq. 28}$$

During the adaptation step of Eq. 27, an extra communication step has been added, since each vehicle receives from connected neighbors the pair of $l$-th row of Laplacian matrix and the $l$-th differential coordinate, in the form of $\{\delta_i^{(t,x)}, \boldsymbol{L}_{\boldsymbol{l}:}^{(\boldsymbol{t})\boldsymbol{T}}\}$. At the initialization stage, once again the noisy GPS locations are utilized as a rough estimation. The $\boldsymbol{GLLME}$ approach is summarized on **Algorithm 7**.

***Algorithm 7: Graph Laplacian LMS with measurements exchanges or GLLME***

```
Input: $N, T, K, \tilde{\boldsymbol{z}}_{\boldsymbol{p}}^{(\boldsymbol{t,x})}, \delta_i^{(t,x)}, \boldsymbol{L}_{\boldsymbol{i}:}^{(\boldsymbol{t})}, \boldsymbol{C}^{(\boldsymbol{t})}$
Output: $\boldsymbol{w}_{\boldsymbol{i}}^{(\boldsymbol{t,x})} \in \mathbb{R}^N, i \in N$
For $t = 1, \dots T$ do
    For each vehicle $i\ in\ parallel$ do
        $\boldsymbol{w}_{\boldsymbol{i,x}}^{(\boldsymbol{t,1})} = \tilde{\boldsymbol{z}}_{\boldsymbol{p}}^{(\boldsymbol{t,x})}$ ;
        $\mu_{1,i}^{(t)}$ from Eq. 30;
        For $k = 1, \dots K$ do
            $\boldsymbol{\psi}_{\boldsymbol{i,x}}^{(\boldsymbol{t,k+1})} = \boldsymbol{w}_{\boldsymbol{i,x}}^{(\boldsymbol{t,k})} - \mu_{2,i}^{(t)} \sum_{l\in\mathcal{N}_i^{(t)}} \boldsymbol{L}_{\boldsymbol{l}:}^{(\boldsymbol{t})} \left(\delta_l^{(t)} - \boldsymbol{L}_{\boldsymbol{l}:}^{(\boldsymbol{t})} \boldsymbol{w}_{\boldsymbol{i,x}}^{(\boldsymbol{t,k})}\right)$;
            $\boldsymbol{w}_{\boldsymbol{i,x}}^{(\boldsymbol{t,k+1})} = \sum_{i\in\mathcal{N}_i^{(t)}} c_{il}^{(t)} \boldsymbol{\psi}_{\boldsymbol{l,x}}^{\boldsymbol{t,k+1}}$;
        End
    End
End
```

Therefore, both $\boldsymbol{GLCG}$ and $\boldsymbol{GLLME}$ are actually addressing the same optimization problem of Eq. 25, employing CG and LMS with measurements exchanges algorithms, respectively. They mainly focus on improving cooperative awareness ability via the integration of additional information, with respect to $\boldsymbol{GLLMS}$. However, $\boldsymbol{GLLME}$ doesn't utilize the ill-conditioned $\boldsymbol{A}_{\boldsymbol{i}}^{(\boldsymbol{t})}$. At the same time, the optimal step size of the proposed scheme, as well as that of $\boldsymbol{GLLMS}$, can be optimally determined according to the best practices, i.e. exploiting the connectivity topology of vehicles (shown in the next subsection). Thus, avoiding the impact of highly contaminated by noise heterogeneous data to the performance of Laplacian diffusion. Finally, note that the communication overhead of $\boldsymbol{GLLME}$ is the same as $\boldsymbol{GLCG}$, since the same data pair has to be broadcast.

The regression vector $\boldsymbol{L}_{\boldsymbol{i}:}^{(\boldsymbol{t})}$ is a deterministic quantity, since is referring to $i$-th row of Laplacian matrix. This property facilitates the optimal selection of step sizes $\mu_{1,i}^{t)}$ and $\mu_{2,i}^{(t)}$, respectively, for convergence in the mean sense [26]: $0 < \mu_{1,i}^{(t)} < \frac{2}{\lambda_1^{max}}$ and $0 < \mu_{2,i}^{(t)} < \frac{2}{\lambda_2^{max}}$. The $\lambda_1^{max}$corresponds to the maximum eigenvalue of instantaneous covariance matrix $\boldsymbol{L}_{\boldsymbol{i}:}^{(\boldsymbol{t})\boldsymbol{T}}\boldsymbol{L}_{\boldsymbol{i}:}^{(\boldsymbol{t})}$, while $\lambda_2^{max}$corresponds to [19] the maximum eigenvalue of instantaneous covariance positive semi-definite matrix $\sum_{l\in\mathcal{N}_i^{(t)}} c_{il}^{(t)}\left(\boldsymbol{L}_{\boldsymbol{l}:}^{(\boldsymbol{t})\boldsymbol{T}}\boldsymbol{L}_{\boldsymbol{l}:}^{(\boldsymbol{t})}\right)$. At the same time, maximum convergence speed is attained when: $\mu_{1,i}^{(t)} = \frac{2}{\lambda_1^{max}+\lambda_1^{min}}$ and $\mu_{2,i}^{(t)} = \frac{2}{\lambda_2^{max}+\lambda_2^{min}}$, where $\lambda_1^{min}, \lambda_2^{min}$.are the minimum eigenvalues of corresponding covariance matrices.

***Proposition 1****: To ensure fast convergence to the optimal solution and asymptotic unbiasedness of* $\boldsymbol{GLLMS}$ *and* $\boldsymbol{GLLME}$*, the following must hold:*

$$\mu_{1,i}^{(t)} = 2/(|\mathcal{N}_i^{(t)}| - 1)^2 + |\mathcal{N}_i^{(t)}| - 1)$$
$$\mu_{2,i}^{(t)} = 2/\lambda_2^{max}$$

*Proof:*

Covariance matrix $\boldsymbol{L}_{\boldsymbol{i}:}^{(\boldsymbol{t})\boldsymbol{T}}\boldsymbol{L}_{\boldsymbol{i}:}^{(\boldsymbol{t})}$ is in fact a rank-one matrix, as a product of vector and its transpose. However, rank-one matrices have only one non-zero eigenvalue. Thus, $\lambda_1^{min} = 0$. Furthermore, the trace of a matrix is equal to the sum of its eigenvalues, i.e. to the largest eigenvalue in case of rank-one matrix. Therefore:

$$\lambda_1^{max} = tr(\boldsymbol{L}_{\boldsymbol{i}:}^{(\boldsymbol{t})\boldsymbol{T}}\boldsymbol{L}_{\boldsymbol{i}:}^{(\boldsymbol{t})}) = \left\|\boldsymbol{L}_{\boldsymbol{i}:}^{(\boldsymbol{t})}\right\|^2 = (|\mathcal{N}_i^{(t)}| - 1)^2 + |\mathcal{N}_i^{(t)}| - 1.$$

Furthermore, we set:

$$\mu_{1,i}^{(t)} = min(0.1, \qquad 2/((|\mathcal{N}_i^{(t)}| - 1)^2 + |\mathcal{N}_i^{(t)}| - 1)) \qquad \textit{Eq. 29}$$

$$\mu_{2,i}^{(t)} = min(0.1, \qquad 2/\lambda_2^{max}) \qquad \textit{Eq. 30}$$

in order to avoid large step sizes which may slower down the convergence. Finally, the convergence in the mean sense of $\boldsymbol{GLLME}$ is guaranteed as follows:

***Proposition 2****: A sufficient condition of asymptotic unbiasedness of* $\boldsymbol{GLLME}$ *is provided by:*

$$0 < \mu_{2,i}^{(t)} < \frac{2}{{}_{l\in\mathcal{N}_i^{(t)}}((|\mathcal{N}_l^{(t)}| - 1)^2 + |\mathcal{N}_l^{(t)}| - 1)}$$

*Proof:*

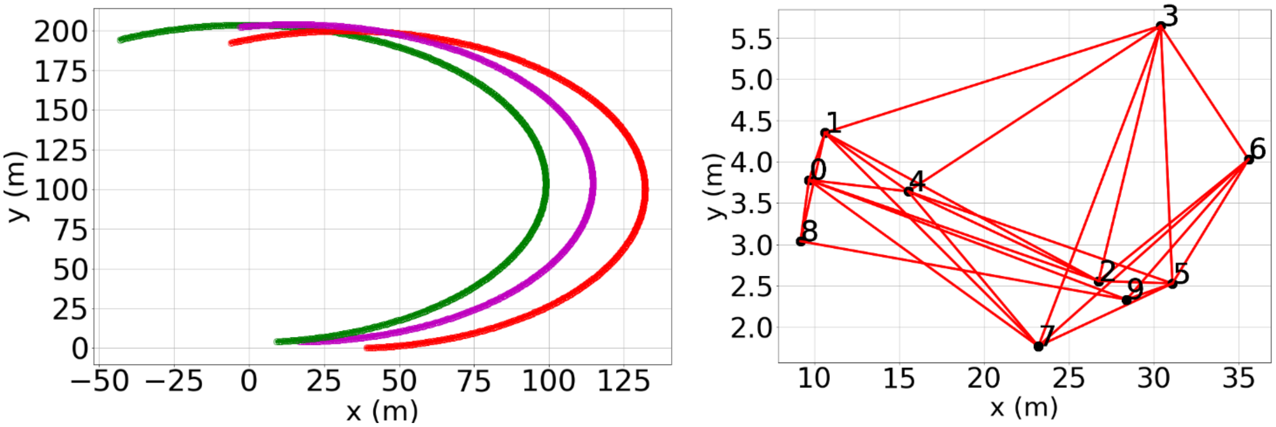
By the properties of positive semi-definite matrix (non-negative eigenvalues and at least one zero eigenvalue), we derive that $\lambda_2^{min} = 0$. As mentioned in [26], the largest eigenvalue of a real symmetric matrix is convex in the elements of that matrix. Thus, and due to the convexity of $c_{il}^{(t)}$, we have:

$$\lambda_{max}\left(\sum_{l\in\mathcal{N}_i^{(t)}} c_{il}^{(t)}\left(\boldsymbol{L}_{\boldsymbol{l}:}^{(\boldsymbol{t})\boldsymbol{T}}\boldsymbol{L}_{\boldsymbol{l}:}^{(\boldsymbol{t})}\right)\right) \le \sum_{l\in\mathcal{N}_i^{(t)}} c_{il}^{(t)}\lambda_{max}\left(\boldsymbol{L}_{\boldsymbol{l}:}^{(\boldsymbol{t})\boldsymbol{T}}\boldsymbol{L}_{\boldsymbol{l}:}^{(\boldsymbol{t})}\right)$$
$$\le argmax_{l\in\mathcal{N}_i^{(t)}}\lambda_{max}\left(\boldsymbol{L}_{\boldsymbol{l}:}^{(\boldsymbol{t})\boldsymbol{T}}\boldsymbol{L}_{\boldsymbol{l}:}^{(\boldsymbol{t})}\right) = argmax_{l\in\mathcal{N}_i^{(t)}}tr(\boldsymbol{L}_{\boldsymbol{l}:}^{(\boldsymbol{t})\boldsymbol{T}}\boldsymbol{L}_{\boldsymbol{l}:}^{(\boldsymbol{t})})$$

$$= \ argmax_{l\in\mathcal{N}_i^{(t)}} \left\|\boldsymbol{L}_{l:}^{(t)}\right\|^2 = \ argmax_{l\in\mathcal{N}_i^{(t)}}((|\mathcal{N}_l^{(t)}|-1)^2+|\mathcal{N}_l^{(t)}|-1)$$
$$\Leftrightarrow \lambda_2^{max} \le \ argmax_{l\in\mathcal{N}_i^{(t)}}((|\mathcal{N}_l^{(t)}|-1)^2+|\mathcal{N}_l^{(t)}|-1).$$

## 4.5 Experimental setup and results

CARLA simulator has been employed to extract different traffic patterns of vehicles moving in an urban city (example shown in Figure 24).



(a) $Ground\ truth$ (b) $VANET\ graph\ example$

*Figure 19: Reference trajectories and VANET graph*

We generated also the positions of $N$ vehicles using the bicycle kinematic model of Eq. 3-Eq. 5. To evaluate the effectiveness of the introduced approaches, we assume that vehicles are always members of VANET, with $\sigma_x = 3m$ and $\sigma_y = 2.5m$ resulting in average GPS localization error of $3.4m$. Ground truth for $N = 3$ vehicles and a VANET graph is shown in Figure 19. The simulation horizon was set to $T = 500$ time instances, with sampling interval $\Delta T = 0.1s$. GPS updating time was chosen to coincide with the sampling interval $\Delta T$ of simulation. However, in realistic conditions updating time may be much higher, and as a matter of fact GPS's availability for real-time applications is reduced. For that purpose, the prediction step of Kalman filter can be utilized in order to provide GPS-like position [27], until the next true GPS measurement is provided by the sensor. In any case, instead of GPS, a visual odometry like solution which meets real-time constraints can also be used [28]. The communication range was initialized to $r_c = 20m$. We create matrices $\boldsymbol{P}^{(t)} = [\boldsymbol{x}^{(t)} \quad \boldsymbol{y}^{(t)}] \in \mathbb{R}^{N\times 2}$ and $\boldsymbol{W}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{k})} = \left[\boldsymbol{w}_{\boldsymbol{i},\boldsymbol{x}}^{(\boldsymbol{t},\boldsymbol{k})} \quad \boldsymbol{w}_{\boldsymbol{i},\boldsymbol{y}}^{(\boldsymbol{t},\boldsymbol{k})}\right] \in \mathbb{R}^{N\times 2}, \forall i, k$. We measured the normalized Average Mean Square Deviation (AMSD) [18] of VANET over $T$ for each iteration:

$$AMSD(k) = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{N}\sum_{i=1}^{N}\frac{\left\|\boldsymbol{P}^{(t)} - \boldsymbol{W}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{k})}\right\|^2}{\|\boldsymbol{P}^{(t)}\|^2},$$

and the Localization Mean Square Error (LMSE) at each time instant:

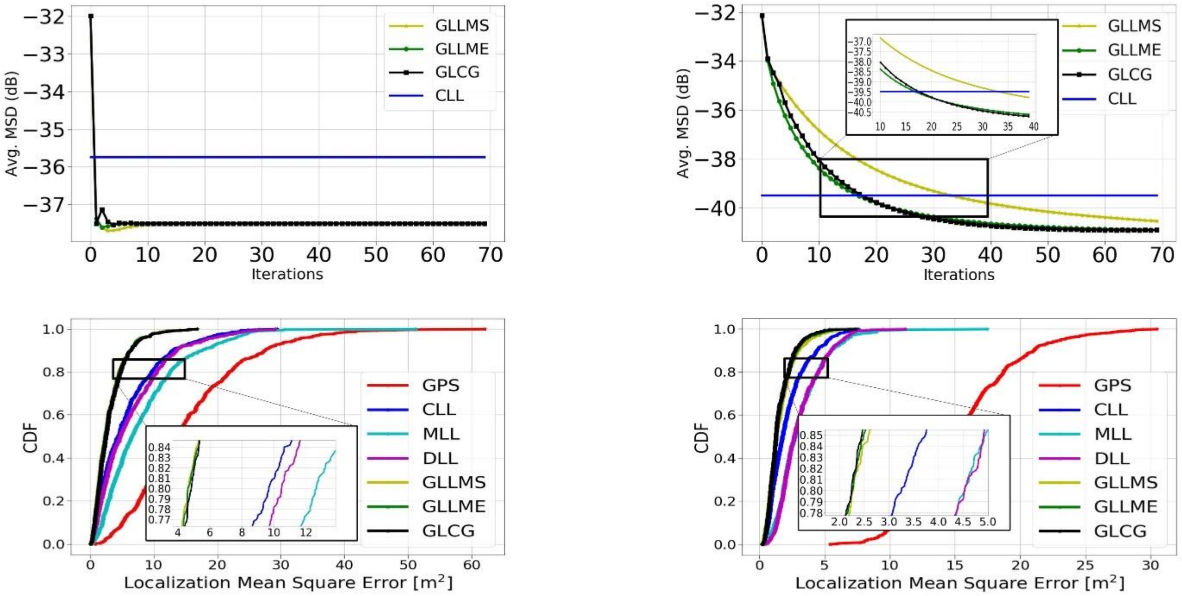
$$LMSE(t) = \frac{1}{N}\sum_{i=1}^{N}\left\|\boldsymbol{p}_{\boldsymbol{i}}^{(\boldsymbol{t})} - \boldsymbol{W}_{\boldsymbol{i}:}^{(\boldsymbol{t},\boldsymbol{K})}\right\|^2$$

Since vehicles utilizing the proposed distributed and diffusion approaches converge to the same location vector, LMSE has been computed exploiting the estimation of a random vehicle for each associated experiment. The conducted experiments were based on: i) the impact of connectivity topology of involved vehicles, i.e. VANET size $N$ and maximum number of neighbors $N_{max}$, ii) network delay effect, and iii) range measurements uncertainty. The latter dictates the feasible estimation of differential coordinates even in occluded and highly complex environments, which is considered vital for the convergence speed and location estimation accuracy of the proposed methods. Finally, we constructed the Cumulative Distribution Function (CDF) of LMSE. As it will be shown, all three proposed approaches outperform in terms of location estimation accuracy, the distributed low cost variant

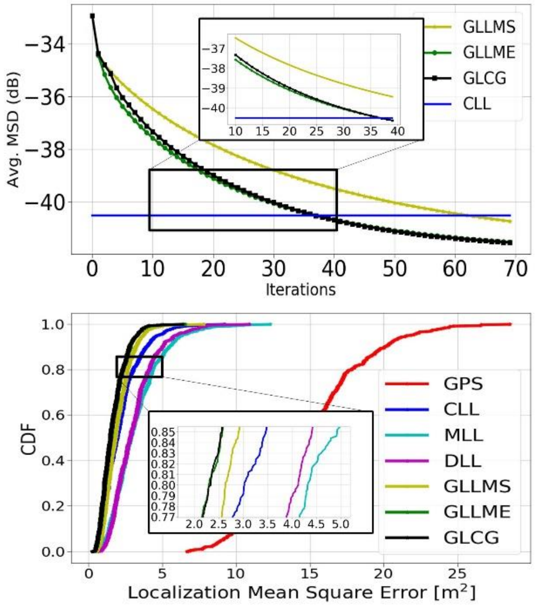
$\boldsymbol{Distributed\ Laplacian\ Localization}$ $(\boldsymbol{DLL})$ [10] of **CLL** and the method of $\boldsymbol{MLL}$. The former estimates only the ego-vehicle location, using the noisy positions of local neighborhood and the so-called local Graph Laplacian operator. The latter is chosen since it exploits exactly the same multi-modal data as we do, while it utilizes for fusion the, prominent in CL literature [6], maximum likelihood estimation criterion.

1. Impact of VANET size:

VANET size is a critical factor for successive Graph Laplacian diffusion localization, due to the growing size of the common location vector that needs to be estimated by the vehicles. In Figure 20, we depict the AMSD and corresponding CDF's for $N = 3, 13$ and $15$, with $K = 70$, $N_{max} = 6$ and $\sigma_d = 1m, \sigma_{az} = 4°$. More specifically, in Figure 20-(a) all three diffusion schemes significantly outperformed $\boldsymbol{CLL}$, requiring only 1 iteration, avoiding constantly receiving and broadcasting measurements. That superior performance is also present on the CDF of LMSE. The reduction of GPS LMSE is **84%** with $\boldsymbol{GLLMS}$, $\boldsymbol{GLLME}$ and $\boldsymbol{GLCG}$, 65% with $\boldsymbol{CLL}$, 62% with $\boldsymbol{DLL}$ and 53% with $\boldsymbol{MLL}$. The significant location estimation performance of the proposed approaches has been achieved since vehicles are more likely to be all-to-all V2V connected. With a larger VANET size, more iteration are required for convergence. In Figure 20-(b) with $N = 13$, $\boldsymbol{GLLME}$ and $\boldsymbol{GLCG}$ converged faster than $\boldsymbol{GLLMS}$, i.e. in around 20 iterations instead of 35. Clearly, measurements exchanges step enhanced the convergence speed of diffusion, since additional knowledge of global solution is utilized. Moreover, all three approaches outperformed all others in terms of location estimation. The reduction of GPS LMSE is **90%** with $\boldsymbol{GLLMS}$, $\boldsymbol{GLLME}$ and $\boldsymbol{GLCG}$, 87% with $\boldsymbol{CLL}$ and 80% with $\boldsymbol{DLL}$ and $\boldsymbol{MLL}$. Finally, in Figure 20-(c) with $N = 15$, $\boldsymbol{GLLME}$ and $\boldsymbol{GLCG}$ converged in around 40 iterations, instead of 60 required by $\boldsymbol{GLLMS}$. Moreover, the reduction of GPS LMSE is **87%** with $\boldsymbol{GLLMS}$, **89%** with $\boldsymbol{GLLME}$ and $\boldsymbol{GLCG}$, 86% with $\boldsymbol{CLL}$ and 80% with $\boldsymbol{DLL}$ and $\boldsymbol{MLL}$. Evidently, $\boldsymbol{GLLME}$ and $\boldsymbol{GLCG}$ are much more efficient in terms of convergence speed than $\boldsymbol{GLLMS}$, when VANET size grows. At the same time, all three methods exhibited superior location estimation performance over $\boldsymbol{CLL}$, $\boldsymbol{DLL}$ and $\boldsymbol{MLL}$.



(a) $VANET\ size\ N\ =\ 3$ (b) $VANET\ size\ N\ =\ 13$

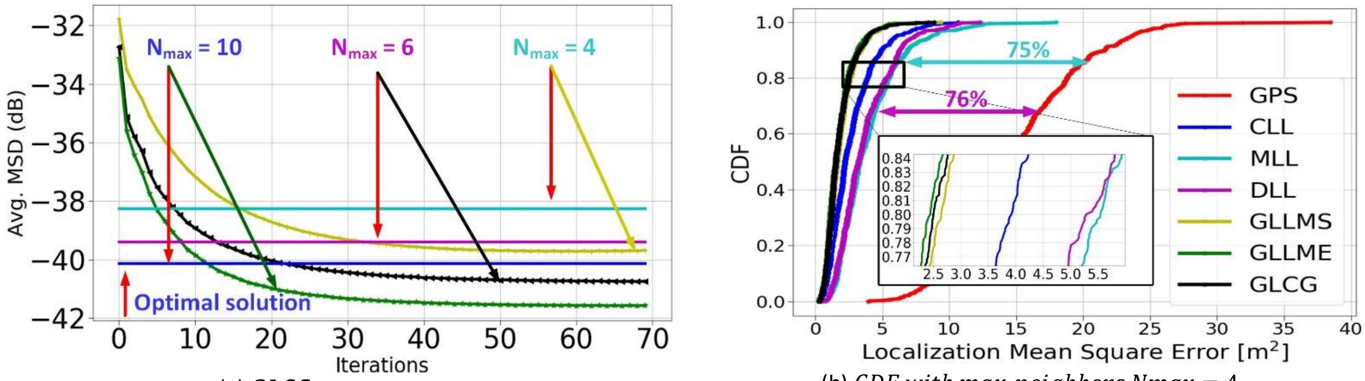


(c) $VANET\ size\ N\ =\ 15$

*Figure 20: Learning curves and CDF with max. neighbors Nmax = 6*

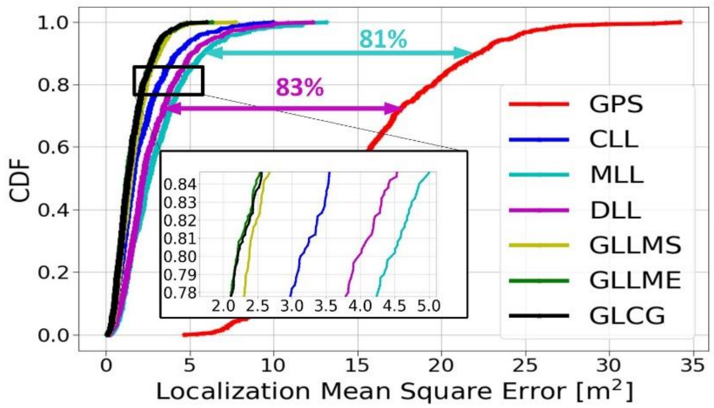
2. <u>Impact of vehicle connections:</u>

Vehicles may communicate with a different number of neighbors while they are moving. Since the Graph Laplacian regression vectors represent in fact those possible V2V connections, it is straightforward to study the impact of $N_{max}$ to the diffusion schemes. Therefore, the effect of that parameter in convergence speed and location estimation accuracy is demonstrated in Figure 21, with $N = 10$, $K = 70$ and $\sigma_d = 1m, \sigma_{az} = 4°$. For example, in Figure 21-(a) we depict the AMSD of $\boldsymbol{GLCG}$ for $N_{max} = 4{,}6$ and $10$. Clearly, the optimal **CLL** solution is achieved when $N_{max} = 10$, since vehicles integrate greater amount of information. Accordingly, $\boldsymbol{GLCG}$ with $N_{max} = 10$ converged much faster to the optimal solution, i.e. in around 10 iterations. Each vehicle is connected to a large number of neighbors, even to the overall number of VANET's vehicles, and thus it is much for efficient to estimate the entire location vector in a fully distributed manner. Additionally, **GLCG** with $N_{max} = 6$ converges in around 20 iterations, while $N_{max} = 4$ requires much more iterations than $K = 70$. In Figure 21-(b), the reduction of GPS LMSE is **88%** with $\boldsymbol{GLLMS}$, $\boldsymbol{GLLME}$ and $\boldsymbol{GLCG}$ and 83% with $\boldsymbol{CLL}$. In Figure 21-(c), with $N_{max} = 10$, performances have been improved. For example, the reduction of GPS LMSE is **90%** with $\boldsymbol{GLLMS}$, $\boldsymbol{GLLME}$ and $\boldsymbol{GLCG}$ and 87% with $\boldsymbol{CLL}$. Once again, the proposed distributed and diffusion schemes outperformed not only the global solution, but also $\boldsymbol{DLL}$ and $\boldsymbol{MLL}$. Consequently, the impact of V2V connections, in the form of Graph Laplacian regression vectors, is crucial for the convergence speed.



(a) $\boldsymbol{GLCG}\ convergence$

(b) $CDF\ with\ max.neighbors\ Nmax = 4$



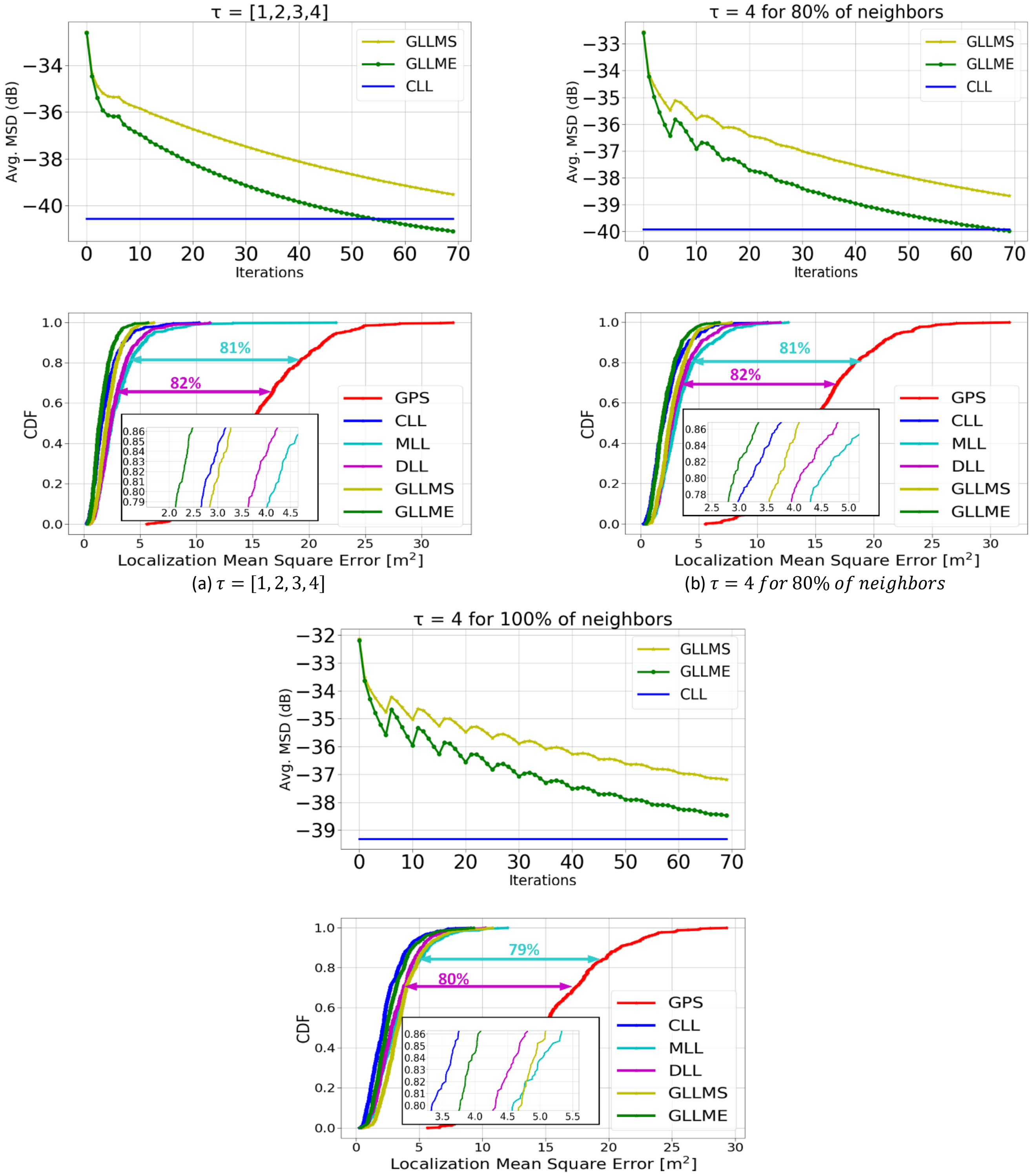
(c) $CDF\ with\ max.neighbors\ Nmax = 10$

*Figure 21: Impact of connections with VANET size N = 10*

3. Impact of network delay:

The upcoming 5G standard will include Ultra Reliable Low Latency Communication (URLLC) services. URLLC is designed for applications that require stringent latency and reliability requirements in vehicular communications [29]. Clearly, the network of vehicles must be time synchronized, which means that each vehicle transmits to neighbors the intermediate vectors before the next iteration $k+1$. However, according to URLLC specifications, communication delay can be regarded around $10ms$ for a network of approximately 10 vehicles, with velocities lower than $10m/s$. To this end, and based on [30], the location estimation vector at every iteration is now provided by: $\boldsymbol{w}_{\boldsymbol{i},\boldsymbol{x}}^{(\boldsymbol{t},\boldsymbol{k+1})} = \sum_{l\in\mathcal{N}_i} c_{il}^{(t)} \boldsymbol{\psi}_{\boldsymbol{l},\boldsymbol{x}}^{(\boldsymbol{t},\boldsymbol{k+1-\tau})}$. Small integer $\tau > 0$ indicates the delayed version of intermediate vectors received by vehicle $i$. Each vehicle broadcasts CAM messages at least every $100ms$, while the maximum delay introduced by V2V communication can reach 300$ms$ [31] at heavy traffic density 0.1 vehicles/meter. Therefore, for every iteration of proposed algorithms we have at most 400 $ms$ delay, which imply that $\tau$ can reach 4, i.e. vehicle $i$ receives vectors $\boldsymbol{\psi}_{\boldsymbol{l},\boldsymbol{x}}^{(\boldsymbol{t},\boldsymbol{k+1-4})}$ by its neighbors, estimated 4 iterations before.

Network delay effect in Graph Laplacian diffusion is demonstrated in Figure 22, with $N = 13$, $K = 70$ and $\sigma_d = 1m, \sigma_{az} = 4°$. The $\boldsymbol{GLCG}$ has been omitted from evaluation, since it was verified during the conducted experiments that totally failed to operate. That drawback is related to computing the optimal step size $a_i$ and factor $\beta_i$ directly from available data. Due to the fact that past estimations are actually used, the two main parameters are significantly deviate from their expected values and negatively influence the convergence of $\boldsymbol{GLCG}$. The bad condition number $\kappa(\boldsymbol{U}_{\boldsymbol{i}}^{(\boldsymbol{t})})^2$ additionally affects $\boldsymbol{GLCG}$. In Figure 22-(a), the network delay parameter, for each vehicle and iteration number, is randomly chosen (with probability 0.25) to take values from the set $\tau = [1,\quad 2,\quad 3,\quad 4]$ in order to simulate random time-varying delay for each vehicle. Clearly, $\boldsymbol{GLLME}$ converges much faster than $\boldsymbol{GLLMS}$. The latter actually requires more iterations than 70 in order to converge effectively. However, the convergence speed in general, has been significantly reduced compared to Figure 22-(b). $\boldsymbol{GLLME}$ achieved **90%** reduction of LMSE (greater than **CLL**), while $\boldsymbol{GLLMS}$ **86%**. To further investigate the impact of delay, we set that each vehicle receives from 80% of its neighbors the intermediate vectors estimated $\tau = 4$ iterations before. From Figure 22-(b), we derive that $\boldsymbol{GLLME}$ attains much higher convergence speed than $\boldsymbol{GLLMS}$ once again, though significantly increased with respect to previous case. **GLLME** achieved **86%** reduction of LMSE, while $\boldsymbol{GLLMS}$ **82%**. Finally, with $\tau = 4$ for all vehicles and their neighbors, network delay has a much stronger footprint, as demonstrated in Figure 22-(c). Once again $\boldsymbol{GLLME}$ exhibits higher convergence speed, though it can't converge during $K = 70$ iterations. Network delay impacts also on location estimation accuracy, since $\boldsymbol{GLLME}$ achieved **83%** reduction of LMSE, while $\boldsymbol{GLLMS}$ **77%**, both lower than $\boldsymbol{CLL}$. Therefore, we conclude that $\boldsymbol{GLLME}$ seems to be more robust to network delay effect, i.e. time-varying delay parameters and constant parameter with stronger footprint, than $\boldsymbol{GLLMS}$ both for the case of convergence and location accuracy.



(a) $\tau = [1, 2, 3, 4]$

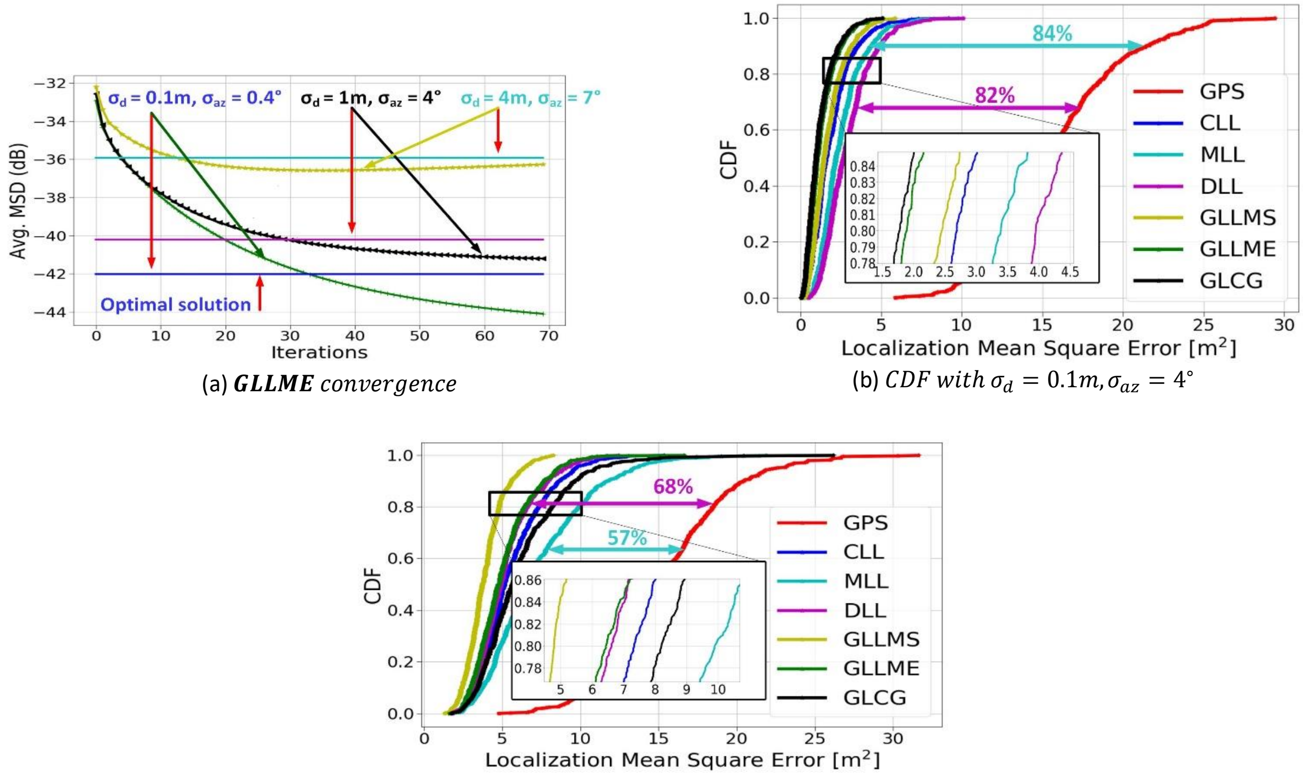
(b) $\tau = 4\ for\ 80\%\ of\ neighbors$

(c) $\tau = 4\ for\ 100\%\ of\ neighbors$

*Figure 22: Network delay study with VANET size N = 13, max. neighbors Nmax = 6*

4. Impact of range measurements uncertainty:

In highly complex urban environments, vehicles may exhibit non-line-of-sight conditions, limiting their ability of accurate relative measurements. The impact of uncertainty in range measurements is depicted in Figure 23, with $N = 15$ and $K = 70$. Differential coordinates with increased measurement noise may deteriorate the performance of diffusion schemes. For example, in Figure 23-(a) we present the AMSD for $\boldsymbol{GLLME}$. The optimal $\boldsymbol{CLL}$ solution has been achieved with the lowest range noise, i.e. $\sigma_d = 0.2m$ and $\sigma_{az} = 0.4^\circ$, since it facilitates the feasible estimation of differential coordinates. The $\boldsymbol{GLLME}$ converges in around 30 iterations. However, when range noise increases, $\boldsymbol{GLLME}$ fails to converge during $K = 70$ iterations. Especially in the case of $\sigma_d = 4m$ and $\sigma_{az} = 7^\circ$, $\boldsymbol{GLLME}$ is far from the optimal global solution. In Figure 23-(b), the reduction of GPS LMSE is **89%** with $\boldsymbol{GLLMS}$, **92%** with $\boldsymbol{GLLME}$ and $\boldsymbol{GLCG}$ and 88% with $\boldsymbol{CLL}$. Evidently, $\boldsymbol{GLLME}$ and **GLCG** outperformed all others, including $\boldsymbol{DLL}$ and $\boldsymbol{MLL}$. However, in Figure 23-(c), the reduction of GPS LMSE is **76%** with **GLLMS**, **69%** with $\boldsymbol{GLLME}$, **64%** with $\boldsymbol{GLCG}$ and 66% with $\boldsymbol{CLL}$. The $\boldsymbol{GLCG}$ approach exhibits lower performance than all others, except $\boldsymbol{MLL}$. The $\boldsymbol{GLLMS}$ approach proves its robustness, since each vehicle utilizes only its own differential coordinate, and thus limiting their noisy impact. The two other diffusion schemes are severely degraded, due to their main feature: receiving the noisy differentials of their connected neighbors. Thus, range measurements uncertainty is a critical factor of both convergence and location estimation of Graph Laplacian diffusion.



(a) $\boldsymbol{GLLME}$ $convergence$

(b) $CDF\ with\ \sigma_d = 0.1m, \sigma_{az} = 4^\circ$

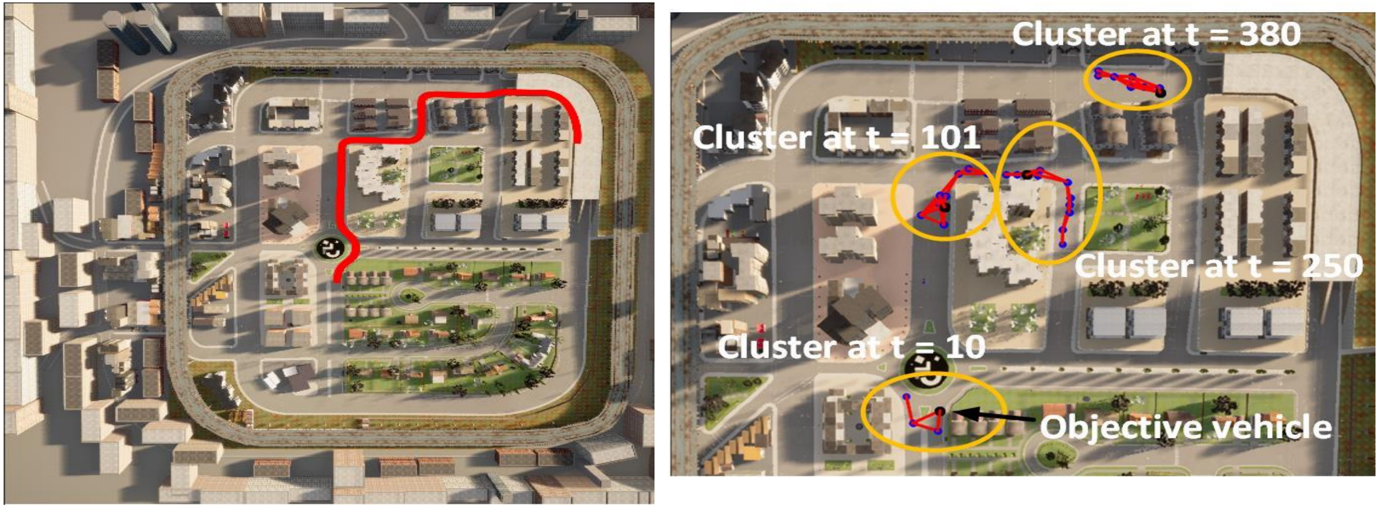
(c) $CDF\ with\ \sigma_d = 4m, \sigma_{az} = 7^\circ$

*Figure 23: Range measurements uncertainty study, assuming a VANET with N = 15 vehicles, while the maximum number of neighbors is Nmax = 6*

Therefore, we conclude that measurements exchanges step significantly increases the convergence speed towards the global solution of $\boldsymbol{CLL}$, by broadcasting only a scalar value and a sparse vector. At the same time, LMS with measurements exchanges exhibits almost the same convergence speed and location estimation accuracy with CG, and even outperform it in the case of network delay and increased range measurements noise. Thus, it isn't required to optimally select the step size as the optimizing argument of the gradient, or to perform conjugate steps towards the solution as $\boldsymbol{GLCG}$ does. An LMS based solution, exploiting step sizes according to **Proposition 1** and **2**, would be suffices for effective Graph Laplacian diffusion localization.

5. <u>Evaluation in CARLA simulator:</u>

The effectiveness of vehicular diffusion localization has been further validated using random realistic trajectories generated by CARLA. The latter is a renowned autonomous driving simulator, extensively used in various automotive applications, especially in computer vision based CAV perception. Therefore, we extracted the trajectory of a random objective vehicle (id 131) and those which belong to the same VANET or cluster as the objective, for $T = 448$ instances. We remind that clusters are formed by imposing a fixed communication range $r_c = 20m$ and maximum number of neighbors $N_{max} = 6$. Objective's trajectory and associated clusters at four distinct time instances are depicted in Figure 24. Black dot represent the objective vehicle. The size of clusters ranges between 2 and 26 vehicles. Total number of iterations is set to $K = 70$. Learning curves of AMSD have been omitted from evaluation, due to the fact that clusters don't contain the same vehicles as time evolves. Actually, different vehicles enter or exit the associated cluster during the simulation horizon.

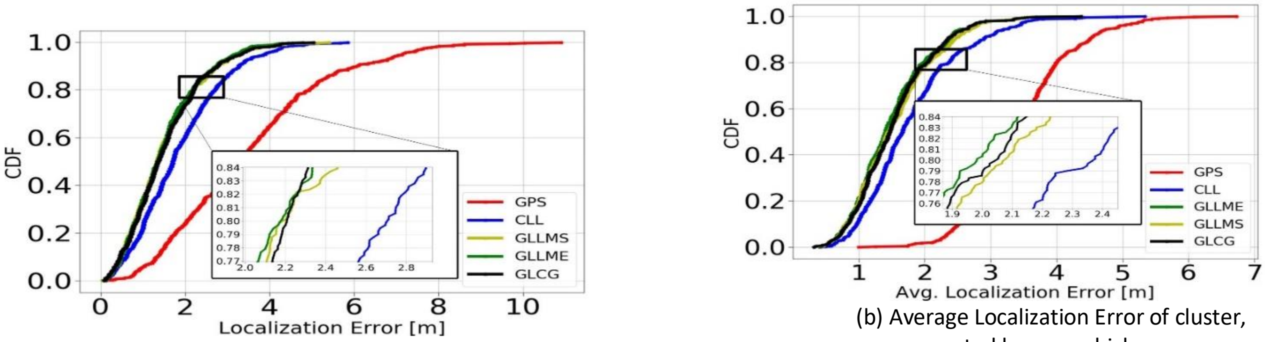


(a) Ground truth

(b) Clusters

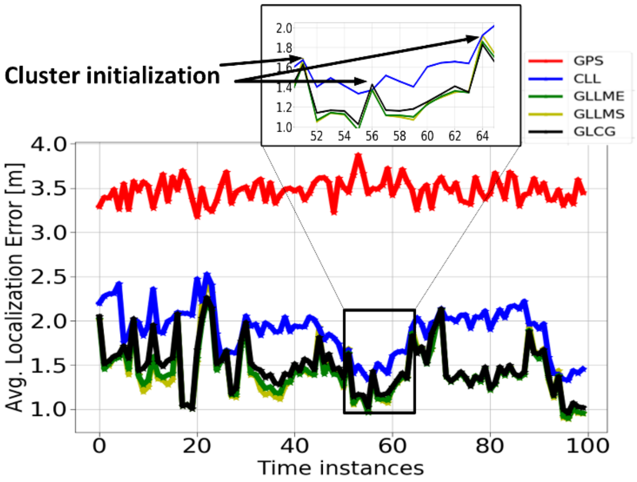
*Figure 24: Reference trajectory and clusters at four time instances in CARLA*

In Figure 25-(a), we demonstrate the Localization Error of objective vehicle, with $\sigma_d = 1m$ and $\sigma_{az} = 4°$. The reduction of GPS Localization Error is 72% with $\boldsymbol{GLLMS}$, $\boldsymbol{GLLME}$ and $\boldsymbol{GLCG}$ and 59% with $\boldsymbol{CLL}$. The overall location accuracy achieved by the objective vehicle, in terms of estimating the location vector of its cluster using the three proposed approaches, is depicted in Figure 25-(b). We measured the Average Localization Error of the associated cluster using the entire location vector estimated by the objective vehicle each time instant. The reduction of GPS Average Localization Error is 59% with $\boldsymbol{GLLMS}$, 60% with $\boldsymbol{GLLME}$, 59% with $\boldsymbol{GLCG}$ and 51% with $\boldsymbol{CLL}$. Finally, in Figure 25-(c) we plotted the Average Localization Error of objective vehicle over 200 iterations, for the first 100 time instances. Clearly, all three proposed approaches significantly outperformed $\boldsymbol{CLL}$. Moreover, the peaks of diffusion towards the global solution of $\boldsymbol{CLL}$ are due to the fact that at those time instances the associated cluster of objective vehicle is modified, since vehicles do constantly enter or exit.



(a) Localization Error of objective vehicle

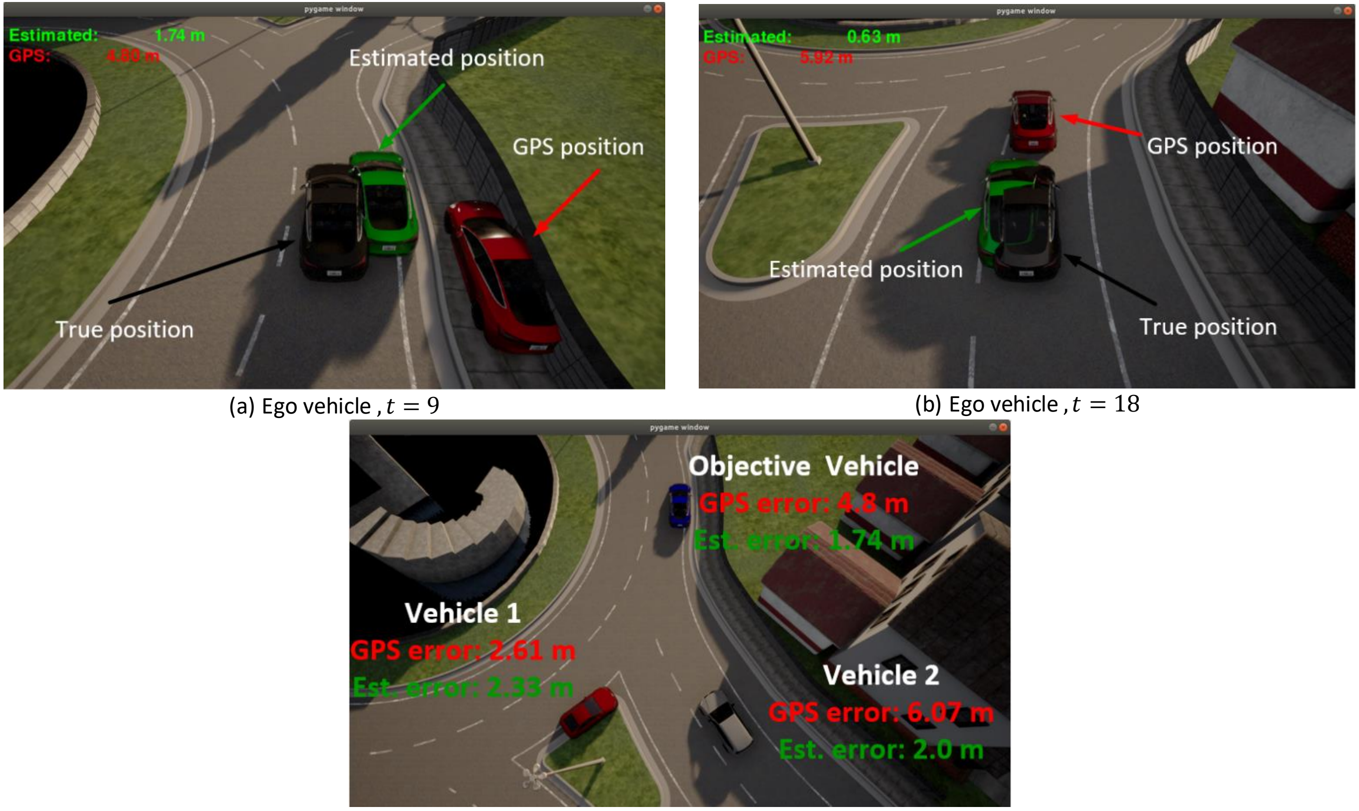
(b) Average Localization Error of cluster, computed by ego vehicle



(c) Average Localization Error of objective vehicle

*Figure 25: Indicative statistical results in CARLA simulator*

Furthermore, we provide in Figure 26 some indicative CARLA based results at time instances $t = 9$ and 18, utilizing $\boldsymbol{GLLME}$. In Figure 26-(a),(b), black vehicle corresponds to the true position, green one to the estimated by $\boldsymbol{GLLME}$, while red vehicle is the actual GPS position. As you may see in both cases, objective's location accuracy is much higher than GPS ($1.74m$ vs $4.8m$ and $0.63m$ vs $5.92m$, respectively). The effectiveness of accurately estimating neighbor's location is also apparent in Figure 26-(c). At time instant $t = 9$, objective vehicle and its two connected neighbors constitute the associated cluster. Obviously, $\boldsymbol{GLLME}$ estimates not only ego's ($1.74m$ vs $4.8m$) but also neighbor's location ($2.33m$ vs $2.61m$ and $2.0m$ vs $6.07m$) much more accurate than GPS. Consequently, all three proposed vehicular diffusion schemes achieved greater performance than GPS and global centralized solution of $\boldsymbol{CLL}$, even in realistic urban traffic conditions generated by CARLA simulator.



(a) Ego vehicle , $t = 9$

(b) Ego vehicle , $t = 18$

(c) Ego and neighbors location estimation , $t = 9$

*Figure 26: CARLA visualization*

As a final remark from the conducted experiments, we have to point out that $\boldsymbol{GLCG}$ exhibits serious limitations and drawbacks as a cooperative localization approach, both in the presence of sensing (e.g., range measurements uncertainties) and communication-network failures, due to its highly vulnerability to noisy data. Hence, LMS based solutions seem to have the potential for efficient localization. The $\boldsymbol{GLLME}$ achieved higher convergence speed and accuracy, especially in the presence of a common network delay, while $\boldsymbol{GLLMS}$ exhibits low execution time and robustness to increased range measurements noise. The tradeoff between the two LMS methods is explicitly related to convergence and accuracy: an increased convergence speed of $\boldsymbol{GLLME}$ implies that a higher accuracy can be achieved earlier. Although the execution time of $\boldsymbol{GLLME}$ is higher, it is related to the required number of iterations $K$. For instance, if $K$ in the learning curve of Figure 20-(c) is reduced almost by half, then $\boldsymbol{GLLME}$ converges (desired localization accuracy is attained) during $K$ iterations with total execution time almost $\sim 100$msec (from Table 2), while $\boldsymbol{GLLMS}$ is far from the optimal solution.

## 4.6 Code repository

The code for the related algorithms is given in:
https://gitlab.com/isi_athena_rc/cpsosaware/cooperative-localization-and-tracking/code/-/blob/master/Diffusion.py

# 5 Graph Laplacian Cooperative Control for Platooning of CAVs[3]

Distributed model-predictive controllers (MPCs) provide a robust way to adjust the acceleration of each platoon vehicle and avoid collisions. This is achieved by transforming the control problem into an iterative, finite-horizon optimization with local constraints. However, the derivation of the global optimal solution is not straightforward. In this Section, first, the consensus cost function is formulated, constrained by minimum distance requirements between the vehicles. Then, the solution is derived via the alternating direction method of multipliers (ADMM), an iterative and robust solver with minimal communication demands. A low-complexity solution is proposed by casting the problem as stochastic control optimization. The developed techniques are evaluated via simulations, where the trajectory of the leading vehicle is generated by CARLA open-source software for autonomous driving.

## 5.1 Controller Design for Cooperative Platooning

Cooperative control aims to design appropriate distributed algorithms such that the group of vehicles can reach consensus on the shared information in the presence of limited and unreliable information exchange and dynamically changing interaction topologies [32]. We use an information graph $\mathcal{G} = (V, E)$, to model the interaction topology between the vehicles. The set $V$ denotes the nodes, i.e., the vehicles, and the set $E \subseteq V \times V$ the edges, which specifies the information flow between neighboring vehicles. An edge $(i, j)$ exists if the $i$-th vehicle has information exchange with the $j$-th vehicle. The set of neighbors of the $i$-th vehicle is defined as $\mathcal{N}_i \triangleq \{j \in V: (i, j) \in E\}$.

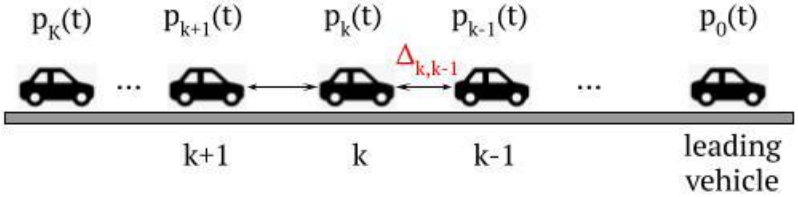


*Figure 27: Illustration of the 1D platooning with a leading and following vehicles.*

Let us consider a network of vehicles moving in line, as shown in Figure 27. For ease of exposition, we only consider one dimension (1D) of the translation motion. Note that, the following analysis is also applicable to two and three dimensions, as long as the dynamics of the vehicle in each coordinate of the Euclidean space is decoupled. The position of the $k$-th vehicle is denoted by $p_k$, its velocity by $\dot{p}_k$, and its acceleration $\ddot{p}_k(t)$. Double-integrator dynamic model captures the dynamics of physical agents such as CAVs in a platooning system. The information states with double-integrator dynamics for the $k$-th vehicle is given by:

[3] E. Vlachos, and Aris S. Lalos, "ADMM-based Cooperative Control for Platooning of Connected and Autonomous Vehicles", IEEE International Conference on Communications (ICC) 2022, *to appear*.

$$\ddot{p}_k = u_k + n_k, \tag{Eq. 31}$$

where $u_k$ is the control input, $n_k$ is the external disturbance, and $k = 1, \dots, K$. The control objective is to make the CAVs maintain a rigid formation geometry by following a desired trajectory.

We consider the following distributed linear control law, where the control action $u_k$ only depends on the relative position and relative velocity information from its neighbors [32], i.e.,

$$u_k = -\sum_{j \in \mathcal{N}_k} \left(c_1(p_k - p_j + \Delta p_{j,k}) + c_2(\dot{p}_k - \dot{p}_j)\right), \tag{Eq. 32}$$

where $\Delta p_{j,k}$ is the desired inter-vehicle gap between vehicles $k$ and $j$, while $c_1$ and $c_2$ are positive constants.

For the network of CAVs, the control objective is to make the CAVs maintain a rigid formation geometry by following a desired trajectory. The desired geometry of the formation is specified by the desired gap, as shown in Figure 27. For each vehicle $k$, with $k = 1,2, \dots, K$, we assume that the underlying physical modeling (plant) at each time instance $t$, is described by the state and input/output equations, i.e.,

$$\boldsymbol{\xi}_k(t+1) = \widehat{\mathbf{A}}_k \boldsymbol{\xi}_k(t) + \hat{\mathbf{b}}_k u_k(t), \tag{Eq. 33}$$

$$y_k(t) = \hat{\mathbf{c}}_k^T \boldsymbol{\xi}_k(t), \tag{Eq. 34}$$

where $\boldsymbol{\xi}_k(t) \in \mathbb{R}^{2\times1}$ is the state vector of the $k$-th vehicle, $u_k(t) \in \mathbb{R}$ is the control input, $y_k(t) \in \mathbb{R}$ is the measured output, and the system matrices/vectors are expressed as:

$$\widehat{\mathbf{A}}_k = \begin{bmatrix} 0 & 1 \\ -c_{1,k} & -c_{2,k} \end{bmatrix}, \hat{\mathbf{b}}_k = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \hat{\mathbf{c}}_k = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

For simplicity, we assume identical vehicles, thus $c_{1,1} = \cdots, c_{1,K} \triangleq c_1$, and $c_{2,1} = \cdots = c_{2,K} \triangleq c_2$. Moreover, we adopt a common sub-index to denote the per-vehicle quantities, i.e.,

$$\widehat{\mathbf{A}}_s \triangleq \widehat{\mathbf{A}}_k, \hat{\mathbf{b}}_s \triangleq \hat{\mathbf{b}}_k, \hat{\mathbf{c}}_s \triangleq \hat{\mathbf{c}}_k \text{ for } k = 1, \dots, K. \tag{Eq. 35}$$

### 5.1.1 Unconstrained MPC

For the single-directional architecture of the platoon (Figure 27), the generalized Laplacian $\mathbf{L} \in \mathbb{R}^{K\times K}$ is a non-symmetric matrix with the main diagonal and an upper diagonal, namely:

$$[\mathbf{L}]_{ij} = \begin{cases} 1 & \text{for} i = j, i = 1, \dots, K, \\ -1/K & \text{for} j = i+1, \\ 0 & \text{elsewhere.} \end{cases} \tag{Eq. 36}$$

The Laplacian matrix is used to extend the plant model of Eq. 33-Eq. 34 for the entire fleet of vehicles, where the network-wide plant is expressed as [32]:

$$\boldsymbol{\xi}(t+1) = \widehat{\mathbf{A}}\boldsymbol{\xi}(t) + \widehat{\mathbf{B}}\mathbf{u}(t), \tag{Eq. 37}$$

$$\mathbf{y}(t) = \widehat{\mathbf{C}}^T \boldsymbol{\xi}(t), \tag{Eq. 38}$$

where $\boldsymbol{\xi}(t) \triangleq [\boldsymbol{\xi}_1^T(t) \cdots \boldsymbol{\xi}_K^T(t) \quad ]^T \in \mathbb{R}^{2K\times1}$ is the state vector for all vehicles, $\mathbf{u}(t) \triangleq [u_1(t) \cdots u_K(t) \quad ]^T \in \mathbb{R}^{K\times1}$ is the control input vector for the $K$ vehicles, $\mathbf{y}(t) \in \mathbb{R}^{K\times1}$ is the measured output vector of all the $K$ vehicles, while

$$\widehat{\mathbf{A}} \triangleq \mathbf{L}(t) \otimes \widehat{\mathbf{A}}_s \in \mathbb{R}^{2K\times2K}, \widehat{\mathbf{B}} \triangleq \mathbf{L}(t) \otimes \hat{\mathbf{b}}_s \in \mathbb{R}^{2K\times K} \tag{Eq. 39}$$

and

$$\widehat{\mathbf{C}} \triangleq \mathbf{L}(t) \otimes \hat{\mathbf{c}}_s^T \in \mathbb{R}^{K\times2K},$$

where $\mathbf{L}(t)$ is the Laplacian matrix for the $t$-th time instance. Following the standard MPC problem formulation [33], we take the difference on both sides of Eq. 37, i.e.,

$$\boldsymbol{\xi}(t+1) - \boldsymbol{\xi}(t) = \widehat{\mathbf{A}}(\boldsymbol{\xi}(t) - \boldsymbol{\xi}(t-1)) + \widehat{\mathbf{B}}(\mathbf{u}(t) - \mathbf{u}(t-1)) \qquad \textit{Eq. 40}$$

or equivalently, the new state equation is expressed as:

$$\boldsymbol{\Delta\xi}(t+1) = \widehat{\mathbf{A}}\boldsymbol{\Delta\xi}(t) + \widehat{\mathbf{B}}\boldsymbol{\Delta}\mathbf{u}(t), \qquad \textit{Eq. 41}$$

where $\boldsymbol{\Delta\xi}(t+1) \triangleq \boldsymbol{\xi}(t+1) - \boldsymbol{\xi}(t)$, and

$$\boldsymbol{\Delta}\mathbf{u}(t) \triangleq \mathbf{u}(t) - \mathbf{u}(t-1).$$

Furthermore, the new input/output model is expressed as:

$$\begin{aligned}\mathbf{y}(t+1) = \mathbf{y}(t) + \widehat{\mathbf{C}}^T(\boldsymbol{\xi}(t+1) - \boldsymbol{\xi}(t)) &= \mathbf{y}(t) + \widehat{\mathbf{C}}^T\boldsymbol{\Delta\xi}(t+1) \\ &= \mathbf{y}(t) + \widehat{\mathbf{C}}^T\widehat{\mathbf{A}}\boldsymbol{\Delta\xi}(t) + \widehat{\mathbf{C}}^T\widehat{\mathbf{B}}\boldsymbol{\Delta}\mathbf{u}(t).\end{aligned} \qquad \textit{Eq. 42}$$

The next step is to define a new state variable vector which connects the state $\boldsymbol{\Delta\xi}(t)$ with the output $\mathbf{y}(t)$. Thus, putting together Eq. 41 and Eq. 42 leads to the following augmented state-space model:

$$\underbrace{\begin{bmatrix}\boldsymbol{\Delta\xi}(t+1) \\ \mathbf{y}(t+1)\end{bmatrix}}_{\mathbf{x}(t+1)} = \underbrace{\begin{bmatrix}\widehat{\mathbf{A}} & \mathbf{O}_{K,K} \\ \widehat{\mathbf{C}}^T\widehat{\mathbf{A}} & \mathbf{E}_{K,K}\end{bmatrix}}_{\triangleq\mathbf{A}}\underbrace{\begin{bmatrix}\boldsymbol{\Delta\xi}(t) \\ \mathbf{y}(t)\end{bmatrix}}_{\mathbf{x}(t)} + \underbrace{\begin{bmatrix}\widehat{\mathbf{B}} \\ \widehat{\mathbf{C}}^T\widehat{\mathbf{B}}\end{bmatrix}}_{\triangleq\mathbf{B}}\boldsymbol{\Delta}\mathbf{u}(t) \qquad \textit{Eq. 43}$$

$$\mathbf{y}(t+1) = \underbrace{[\mathbf{O}_{K,K} \quad \mathbf{E}_{K,K}]}_{\triangleq\mathbf{C}}\begin{bmatrix}\boldsymbol{\Delta\xi}(t) \\ \mathbf{y}(t)\end{bmatrix}, \qquad \textit{Eq. 44}$$

where $\mathbf{O}_{K,K} \in \{0\}^{K\times K}$ is a matrix filled with zeros, and $\mathbf{E}_{K,K} \in \{1\}^{K\times K}$ is a matrix filled with ones.
The MPC for prediction horizon of $N_p$ and control horizon of $N_c$ is expressed compactly as:

$$\mathbf{z}(t+1) = \mathbf{F}\mathbf{x}(t) + \mathbf{G}\mathbf{v}(t), \qquad \textit{Eq. 45}$$

where

$$\mathbf{z}(t+1) \triangleq \left[\mathbf{x}^T(t+1) \quad \cdots \quad \mathbf{x}^T(t+N_p)\right]^T \in \mathbb{R}^{2N_pK\times 1},$$

$$\mathbf{v}(t) \triangleq [\boldsymbol{\Delta}\mathbf{u}^T(t) \quad \cdots \quad \boldsymbol{\Delta}\mathbf{u}^T(t+N_c-1)]^T \in \mathbb{R}^{N_cK\times 1},$$

$$\mathbf{F} = [(\mathbf{CA})^T \qquad\qquad (\mathbf{CA}^{N_p})^T]^T \in \mathbb{R}^{2N_pK\times 2K}, \qquad \textit{Eq. 46}$$

$$\mathbf{G} = \begin{bmatrix}\mathbf{CB} & & \\ \vdots & \ddots & \\ \mathbf{CA}^{N_p-1}\mathbf{B} & \cdots & \mathbf{CA}^{N_c-1}\mathbf{B}\end{bmatrix} \in \mathbb{R}^{2N_pK\times N_cK}. \qquad \textit{Eq. 47}$$

Let us define the centralized cost function

$$\mathcal{J}(\mathbf{v}(t)) =\parallel \boldsymbol{\beta}(t) - \mathbf{F}\mathbf{x}(t) - \mathbf{G}\mathbf{v}(t) \parallel^2, \qquad \textit{Eq. 48}$$

where

$$\boldsymbol{\beta}(t) \triangleq \mathbf{E}_{N_p,1} \otimes [\Delta p_1^*(t) \quad \cdots \quad \Delta p_K^*(t)]^T \in \mathbb{R}^{N_pK\times 1},$$

is the set-point information, defined as:

$$\Delta p_k^*(t) = p_k^*(t) - p_k^*(t-1), \qquad \textit{Eq. 49}$$

and $p_k^*(t)$ is the provided waypoint for the $k$-th vehicle at the $t$-th time instance, and $\mathbf{E}_{N_p,1} \in \{1\}^{N_p\times 1}$. Note that, the waypoints for the leading vehicle (e.g., $k=1$) are provided externally. For $k>1$, we use $p_k^*(t) \triangleq p_{k-1}(t-1)$, which expressed that the $k$-th vehicle has to follow the preceding vehicle $k-1$ and its trajectory.
Then, the unconstrained optimization problem that provides the MPC solution is expressed as:

$$\min_{\mathbf{v}(t)} \mathcal{J}(\mathbf{v}(t)), \qquad \textit{Eq. 50}$$

which has a closed-form solution given by solving of the following system of equations, $\mathbf{Gv}(t) = \boldsymbol{\beta}(t) - \mathbf{Fx}(t)$.

### 5.1.2 Constrained MPC

The unconstrained MPC design provides a simple solution to the problem, with minimal computational cost, that exhibits desirable performance in the case where the leading vehicle moves with constant speed. In realistic scenarios, where the speed of the leading vehicle is variable, it fails to guarantee a safe distance between the vehicles. To overcome this problem, we formulate a constrained MPC problem, which ensures that the $k-1$-th vehicle will be in front of the $k$ vehicle. Let the position of the leading vehicle is $p_0$ on the x-axe, as shown in Figure 27, while for the $K$ following vehicles we assume that:

$$p_0 < p_1 < \cdots < p_{k-1} < p_k < \cdots < p_K.$$

Then, the constrained MPC problem formulation for all $K$ vehicles is expressed as:

$$\min_{\mathbf{v}(t)} \mathcal{J}\big(\mathbf{v}(t)\big) \text{ s.t.} \, p_k(t) \geq p_{k-1}(t) \text{ for } k = 1, \dots, K, \quad \text{Eq. 51}$$

where, recall that, the vector $\mathbf{v}(t) \in \mathbb{R}^{N_c K \times 1}$ is the control input of the MPC.

To solve Eq. 21, first we have to treat the inequality constraints. Thus, we introduce $K$ auxiliary variables $y_k$, with $k = 1, \dots, K$, and the penalty function $I(y_k)$ which is defined as:

$$I(y_k) = \begin{cases} \infty & y_k < 0 \\ 0 & y_k \geq 0. \end{cases} \quad \text{Eq. 52}$$

Then, the problem Eq. 51 becomes [11], for $k = 1, \dots, K$:

$$\min_{\substack{\mathbf{v}(t) \\ \{y_k\}_{k=1}^K}} \mathcal{J}(\mathbf{v}(t)) + \sum_{k=1}^{K} I(y_k(t)) \text{s.t.} \, p_{k-1}(t) - p_k(t) - y_k(t) = 0. \quad \text{Eq. 53}$$

As it will be seen next, the problem (23) can be solved using the ADMM. To proceed, let us first express the constraints with respect to the unknown variable $\mathbf{v}(t)$. Specifically, the estimated position of the $k$-th vehicle can be expressed by using the state vector as $p_k(t) = [\boldsymbol{\xi}(t)]_k$, where $[\mathbf{q}]_k$ denotes the $k$-th element of the vector $\mathbf{q}$. Thus,

$$[\boldsymbol{\xi}(t)]_k - [\boldsymbol{\xi}(t)]_{k-1} = [\widehat{\mathbf{A}}\boldsymbol{\xi}(t-1) + \widehat{\mathbf{B}}\mathbf{u}(t-1)]_k - [\widehat{\mathbf{A}}\boldsymbol{\xi}(t-1) + \widehat{\mathbf{B}}\mathbf{u}(t-1)]_{k-1} \quad \text{Eq. 54}$$

$$= ([\widehat{\mathbf{A}}]_k - [\widehat{\mathbf{A}}]_{k-1})\boldsymbol{\xi}(t-1) + ([\widehat{\mathbf{B}}]_k - [\widehat{\mathbf{B}}]_{k-1})\mathbf{u}(t-1)$$

$$= \gamma_k(t) + [\mathbf{R}]_k \mathbf{u}(t), \quad \text{Eq. 55}$$

where

$$\gamma_k(t) \triangleq ([\widehat{\mathbf{A}}]_k - [\widehat{\mathbf{A}}]_{k-1})\boldsymbol{\xi}(t-1),$$

and $[\mathbf{R}]_k \triangleq [\widehat{\mathbf{B}}]_k - [\widehat{\mathbf{B}}]_{k-1}$.

Recall that, $\mathbf{u}(t) = \mathbf{u}(t-1) + \boldsymbol{\Delta}\mathbf{u}(t)$, while $\boldsymbol{\Delta}\mathbf{u}(t)$ represents the first $K$ entries of the vector $\mathbf{v}(t)$. In order express Eq. 55 with respect to $\mathbf{v}(t)$, let us employ the following mathematical notation,

$$\mathbf{u}(t) = \operatorname{diag}(\mathbf{e}_K)\mathbf{v}(t), \quad \text{Eq. 56}$$

where $\mathbf{e}_K \in \{0,1\}^{KN_c \times 1}$ with $[\mathbf{e}_K]_i = 1$ for $i = 1, \dots, K$ and $[\mathbf{e}_K]_i = 0$ for $i = K+1, \dots, KN_c$. Then,

$$[\boldsymbol{\xi}(t)]_k - [\boldsymbol{\xi}(t)]_{k-1} = \gamma_k(t) + [\mathbf{R}]_k \operatorname{diag}(\mathbf{e}_K)\mathbf{v}(t), \quad \text{Eq. 57}$$

Therefore, by dropping the time index for simplicity, the problem Eq. 53 becomes, for $k = 1, \dots, K$:

$$\min_{\mathbf{v},\mathbf{y}} \mathcal{J}(\mathbf{v}) + \sum_{k=1}^{K} I(y_k) \text{s.t.} \, \gamma_k + [\mathbf{R}]_k \operatorname{diag}(\mathbf{e}_K)\mathbf{v} - y_k = 0 \quad \text{Eq. 58}$$

Let $q_k$, for $k = 1, \dots, K$, denote the $K$ dual variables, then the augmented Lagrangian function is expressed as:

$$\mathcal{C}(\mathbf{v}, \{y_k\}, \{q_k\}) = \mathcal{J}(\mathbf{v}) + \sum_{k=1}^{K} I(y_k)$$

$$+\frac{\rho}{2}(\| \gamma_k + [\mathbf{R}]_k \mathrm{diag}(\mathbf{e}_K)\mathbf{v} - y_k + q_k \|^2 + \| q_k \|^2). \tag{Eq. 59}$$

Next, we employ the ADMM steps to solve Eq. 59. These steps are executed for a number of iterations, $i = 1, \ldots, I_{max}$. The first step is to minimize $\mathcal{C}(\mathbf{v}, \{y_k\}, \{q_k\})$ over $\mathbf{v}$, e.g.,

$$\mathbf{v}^{(i+1)} = arg\min_{\mathbf{v}} \| \boldsymbol{\beta} - \mathbf{F}\mathbf{x} - \mathbf{G}\mathbf{v} \|^2$$

$$+\frac{\rho}{2}(\| \gamma_k + [\mathbf{R}]_k \mathrm{diag}(\mathbf{e}_K)\mathbf{v} - y_k^{(i)} + q_k^{(i)} \|^2). \tag{Eq. 60}$$

The minimization problem of Eq. 60 is equivalent to the solution of the following system of equations [34]

$$(\mathbf{G}^T\mathbf{G} + \rho \mathrm{diag}(\mathbf{e}_K)^T \sum_{k=1}^{K} [\mathbf{R}]_k^T [\mathbf{R}]_k \mathrm{diag}(\mathbf{e}_K)) \ \mathbf{v}^{(i+1)}$$

$$= \mathbf{G}^T\boldsymbol{\beta} + \rho \sum_{k=1}^{K} (\gamma_k - y_k^{(i)} + q_k^{(i)}). \tag{Eq. 61}$$

Note that, the quantities $\mathbf{G}, \mathbf{R}, \boldsymbol{\beta}$ are known to all vehicles and assumed static over the interval of interest. Hence, the solution of Eq. 61 requires only the scalar quantities $y_k$ and $u_k$ from the other vehicles for consensus.

At the next step, we use the estimated value $\mathbf{v}^{(i+1)}$, and solve over $y_k$, for $k = 1, \ldots, K$ [33]:

$$y_k^{(i+1)} = \max(0, \gamma_k + [\mathbf{B}]_k \mathrm{diag}(\mathbf{e}_K)\mathbf{v}^{(i+1)} + q_k^{(i)}) \tag{Eq. 62}$$

At the third step, we update the dual variable for $k = 1, \ldots, K$:

$$q_k^{(i+1)} = q_k^{(i)} + \gamma_k + [\mathbf{B}]_k \mathrm{diag}(\mathbf{e}_K)\mathbf{v}^{(i+1)} - y_k^{(i+1)} \tag{Eq. 63}$$

The described iterative steps of the ADMM method that solve Eq. 63 are provided in **Algorithm 8**.

***Algorithm 8: ADMM algorithm***

```
Input: $\rho$, $\boldsymbol{\beta}(t)$, $\mathbf{G}$, $\boldsymbol{\gamma}(t)$
Output: $\mathbf{v}(t)$
For $i = 1, \ldots, I_{\max}$ do
    Each vehicle solves its own version of Eq. 61 ;
    $y_k^{(i+1)} = \max(0, \gamma_k + [\mathbf{B}]_k \mathrm{diag}(\mathbf{e}_K)\mathbf{v} + q_k^{(i)})$, for all $k$ ;
    Each vehicle broadcasts $y_k^{(i+1)}$;
    $q_k^{(i+1)} = q_k^{(i)} + \gamma_k + [\mathbf{B}]_k \mathrm{diag}(\mathbf{e}_K)\mathbf{v} - y_k^{(i+1)}$, for all $k$ ;
     Each vehicle broadcasts $q_k^{(i+1)}$ ;
End
```

## 5.2 Efficient Implementation

The computational complexity of the **Algorithm 8** depends on the parameters of the problem, such as the platoon size, the prediction horizon of $N_p$, the control horizon of $N_c$, and the number of iterations, $I_{\max}$, that are required for the convergence of the ADMM algorithm. At each iteration $i$, the most costly step is at line 3, which requires the solution of Eq. 61. Moreover, at each iteration $i$, the updated quantities have to broadcasted to the other vehicles, increasing the communication load. Thus, to reduce the overall complexity of the **Algorithm 8**, either we have to reduce the cost of step 3, or we have to minimize the number of iterations $I_{\max}$. To reduce the cost of step 3, the direct inversion method could be replaced with an iterative solver, e.g., in [35].

In this subsection, we focus on the case where the number of iterations is reduced to $I_{\max} = 1$. This will result into a special case of the algorithm, listed in **Algorithm 9**. Indeed, its convergence is achieved over the time instances, thus, the problem cost function $\mathcal{J}(\mathbf{v}(t))$ will be different at each time iteration. Given that the cost function is not varying significantly at each time update, the cost function can be seen as *stochastic*, i.e., it also depends on a random variation. In this case, it can be seen that the proposed algorithm converges to the minimum value of the constrained stochastic optimization,

$$\min_{\mathbf{v}(t),\{y_k\}_{k=1}^{K}} \quad \mathcal{E}\{\mathcal{J}(\mathbf{v}(t),\xi)\} + \sum_{k=1}^{K} I(y_k(t))$$
$$\text{subject to } p_{k-1}(t) - p_k(t) - y_k(t) = 0 \text{ for } k = 1, \ldots, K, \qquad \textit{Eq. 64}$$

where $\xi$ is an unknown random variable with zero mean and variance $\sigma_\xi^2$, which depends on the variability of the cost function. Next, due to high sampling frequency, we assume that $\sigma_\xi$ is negligible. Thus, to formulate a solution for Eq. 64, first we drop the iteration indexing $(i)$ and replace with the time instances index, i.e., $\mathbf{v}^{(i+1)}(t) = \mathbf{v}(t+1)$, and for $k = 1, \ldots, K$: $q_k^{(i+1)}(t) = q_k(t+1)$, and $y_k^{(i+1)}(t) = y_k(t+1)$. Then, the steps can be similarly obtained as in Eq. 60- Eq. 63, as shown in **Algorithm 9**.

Although **Algorithm 9** reduces significantly the computational complexity, it may also introduce an error due to the unknown random variable. However, ADMM is able to converge under relaxed conditions for the introduced error at each step, while a common measure to ensure the convergence, is to adopt a time-varying step-length $\rho(t)$ [34], such as:

$$\rho(t) = \epsilon\rho(t-1),$$

where $\epsilon \in [0,1]$. Then, it can be shown that: $\sum_t \rho(t) < \infty$.

***Algorithm 9:*** $\boldsymbol{ADMM - L\ algorithm}$

```
**Input:** $\rho$, $\boldsymbol{\beta}(t)$, $\mathbf{G}$, $\boldsymbol{\gamma}(t), \mathbf{v}(t), \{y_k(t)\}_{k=1}^{K}, \{u_k(t)\}_{k=1}^{K}$
**Output:** $\mathbf{v}(t+1), \{y_k(t+1)\}_{k=1}^{K}, \{u_k(t+1)\}_{k=1}^{K}$
**For** $i = 1, \ldots, I_{\max}$ **do**
        Each vehicle solves its own version of
                $(\mathbf{G}^T\mathbf{G} + \rho(t)\mathrm{diag}(\mathbf{e}_K)^T \sum_{k=1}^{K} [\mathbf{R}]_k^T[\mathbf{R}]_k\mathrm{diag}(\mathbf{e}_K))\ \mathbf{v}(t+1)$
                $= \mathbf{G}^T\boldsymbol{\beta} + \rho(t)\sum_{k=1}^{K} \left(\gamma_k - y_k(t) + q_k(t)\right)$ ;
        $y_k(t+1) = \max(0, \gamma_k + [\mathbf{B}]_k\mathrm{diag}(\mathbf{e}_K)\mathbf{v} + q_k(t), \forall k$ ;
        Each vehicle broadcasts $y_k(t+1)$ ;
        Each vehicle broadcasts $q_k(t+1)$ ;
        $\rho(t) = \epsilon\rho(t-1)$ ;
**End**
```

*Table 1: Order of computational and communication costs*

| | **MPC** | **ADMM** | **ADMM-L** |
|---|---|---|---|
| Computation | $\mathcal{O}(N_p^2 N_c K^3)$ | $I_{max}\mathcal{O}(N_p^2 N_c K^3)$ | $\mathcal{O}(N_p^2 N_c K^3)$ |
| Communication | $\mathcal{O}(N_c K)$ | $I_{max}\mathcal{O}(N_c K)$ | $\mathcal{O}(N_c K)$ |

In Table 1, we compare the computational and communication cost for each of the considered techniques using Big-O notation. The computation cost measures the number of multiplications between matrices. The communication cost measures the quantity of scalars that have to be the broadcast of the Laplacian matrix and the $N_c \times 1$ state vector of the $K-1$ vehicles.

## 5.3 Results

In this section, we validate the introduced theoretical framework, by performing computer based simulations, using the MATLAB $^{TM}$ software and the Carla simulator for generating vehicle trajectories.

Initially, all the vehicles are stopped, placed with a $\Delta$ meters gap between them. We consider several variable speed scenarios for the leading vehicle, generated using the Carla open-source simulator for autonomous driving [36]. In both cases, the goal is to end up with a platoon formation where the cars follow the leading vehicle, avoiding any collisions between them. To avoid collisions, the vehicles have to keep a safe distance, while the controller has to update its state at each $T_c$ second. More details about the parameters of the simulation results are shown in the Table of Figure 28.

| Setting | Value |
|---|---|
| Number of vehicles | 5 |
| Update time $T_c$ | 1s |
| Inter-vehicle gap | 1m |
| Max. speed | 10 m/s |
| Control horizon $N_c$ | 5 |
| Samples number $N_p$ | 10 |

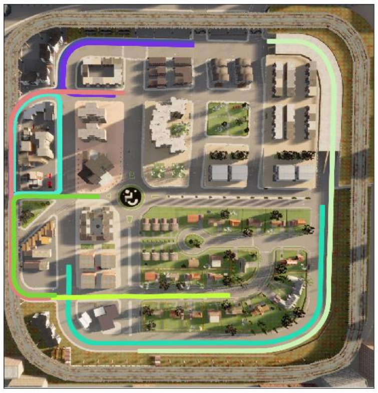


*Figure 28: Simulation Parameters and Road network instance, taken from Carla*

We consider three open-loop controller designs:

- MPC: The conventional closed-form solution of the uncosntrained optimization problem of Eq. 50.
- ADMM: The proposed solution of the constrained optimization problem of Eq. 58 provided by **Algorithm 8**.
- ADMM-L: The proposed low-complexity solution of Eq. 58 provided by **Algorithm 9**.

Concerning the computation and communication overhead, it is important to note that, in our simulations, we have also employed the standard interior-point method, using the CVX public library. However, in all cases, the output of CVX was identical to the ADMM results, and for that reason are not displayed in the figures. As a figure of merit, the proposed ADMM solution was 35x faster than the CVX interior-point technique, while the proposed ADMM-L was 35x faster than the ADMM. As for the communcation cost, the proposed ADMM requires only the transmission of scalar quantities between the vehicles, for each iteration $i$ in **Algorithm 8**, avoiding the transmission of whole matrices as for the standard interior-point methods (e.g., Newton-based methods). Moreover, the ADMM-L futher reduces the communication overhead, by minimizing the number of iterations $I_{\max}$.

In Figure 29 we plot the vehicle trajectories on the x-axe for the considered controller techniques over time. The leading vehicle is not moving till $t = 45s$, and then it starts to move over x-axe with relative constant speed. The MPC technique results into trajectories that collide after $t = 50s$. The constrained MPC techniques, represented by the ADMM and ADMM-L techniques, are able to prevent the collisions. The ADMM exhibits the best performance by keeping the vehicles at the desired safe distance. The ADMM-L, which represents the low-complexity version of the ADMM, avoids the collisions, however the desired distance is not always achieved.

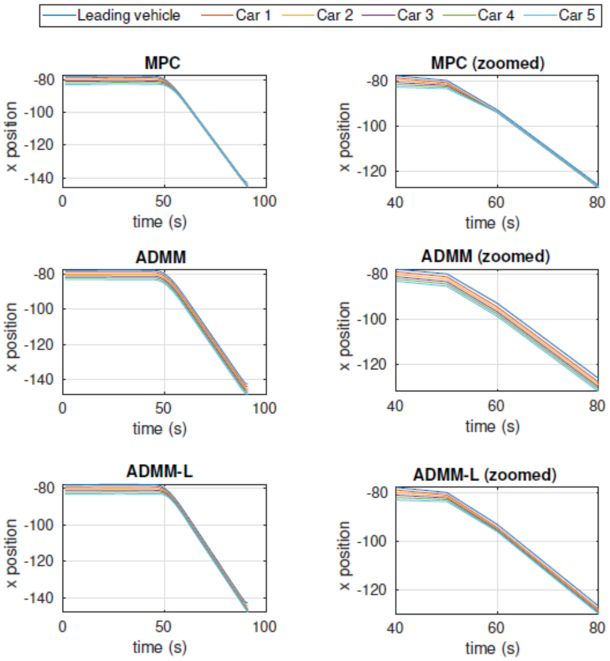


*Figure 29: Vehicles trajectory on the x-axe over time for the case where the leading vehicle follows the trajectory of ID 86 for the Carla generated scenario.*

Let us evaluate the Distance Error for each platoon vehicle $k$ over time in seconds (s). This is defined as, the difference between the estimated position $p_k(t)$ and the desired position $p_k^*(t) = p_0(t) - k\Delta_{k,k-1}$, and it is measured in meters (m), i.e.,

$$\text{DistanceError} \triangleq |p_k(t) - (p_0(t) - k\Delta_{k,k-1})| \ (m), \quad \text{Eq. 65}$$

where the technique with the smaller distance error for each vehicle indicates that the platoon follows the desired waypoints more accurately. We have considered the trajectory case for the leading vehicle with identity (ID) 86, from the Carla generated scenario of Figure 28.

In Figure 30 we present the results for the distance error and the vehicles speed. The error of the unconstrained MPC can be as large as 4.2 (m), while the error is much smaller for the constrained MPC cases. The low-complexity implementation, ADMM-L, reduces the error of the MPC to 2.1 (m), while the ADMM to 0.85 (m). In Figure 31, we plot the CDF of the distance between two successive vehicles, when it is computed over the observed time frame. So for the ADMM case, the distance between all successive vehicles is over 1(m) with probability 1. For the unconstrained MPC and the low-complexity implementation ADMM-L, the probability for the distance between the successive vehicles to be smaller than 1 (m) is not zero. Specifically, it is 50% probable to have distance around 0.4 (m).

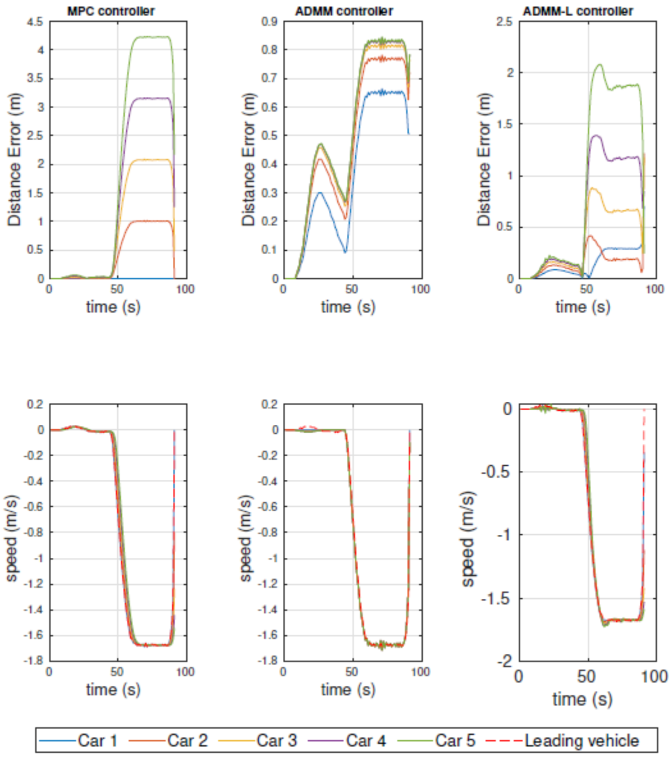


*Figure 30: Controllers' performance evaluation for vehicle with ID 86.*

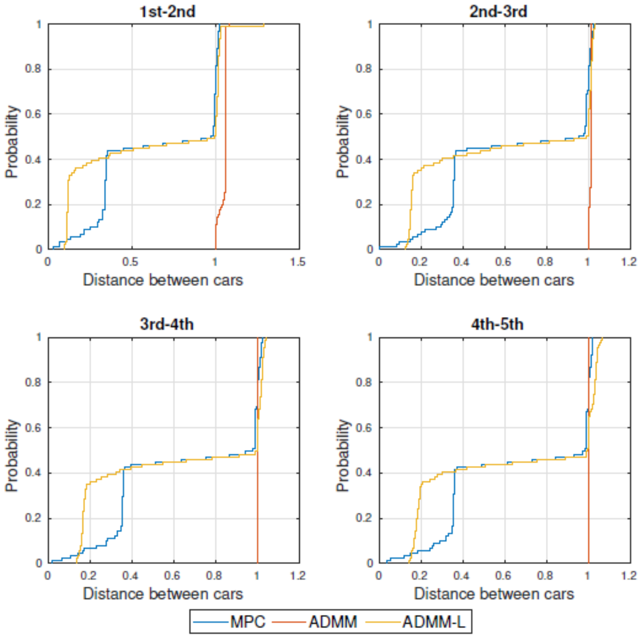


*Figure 31: CDF for the distance between successive vehicles*

So far, we have considered that the controller update time is $T_c = 1s$. However, the update interval has significant impact on the performance of the controller. In particular, by decreasing the update interval, the unconstrained MPC can perform similarly to ADMM, while the performance of the ADMM-L also improves. However, this performance improvement is not always possible, due to the network latency and computational complexity issues.

## 5.4 Code repository

The code for the related cooperative platooning algorithms is given in:
https://gitlab.com/isi_athena_rc/cpsosaware/cooperative_path_planning/cooperative-path-planning

# 6 Conclusion

To summarize, in this deliverable novel distributed localization algorithms have been derived and formulated, based on different estimation approaches, i.e., tracking, least-squares, information diffusion, etc. The indicative results in Sections 3 and 4 have proven the feasibility of the proposed schemes in terms of location accuracy, and thus effectively realizing the vision of global and local cooperative awareness of vehicles. Proposed ADMM based cooperative control platooning scheme of CAVs has also shown very promising results in Section 2. Finally, the simulation framework of Section 6 will serve as the baseline in order to evaluate the proposed methos with realistic V2V conditions in the testing setups of the project.

# References

[1] J. Harding, G. Powell, R. Yoon, J. Fikentscher, C. Doyle, D. Sade, M. Lukuc and J. S. &. J. Wang, "Vehicle-to-vehicle communications: Readiness of V2V technology for application.," *Report No. DOT HS 812 014, Washington, DC: National Highway Traffic Safety Administration,* 2014.

[2] M. Sepulcre and J. Gozalvez, "Experimental evaluation of cooperative active safety applications based on V2V communications," in *Proceedings of the ninth ACM international workshop on Vehicular inter-networking, systems, and applications - VANET \textquotesingle12*, 2012.

[3] M. Sepulcre, J. Gozalvez and J. Hernandez, "Cooperative vehicle-to-vehicle active safety testing under challenging conditions," *Transportation Research Part C: Emerging Technologies,* vol. 26, p. 233–255, 1 2013.

[4] *ETSI EN 302 637-2 V1.4.0 (2018-08),* 2018.

[5] G. Soatti, M. Nicoli, N. Garcia, B. Denis, R. Raulefs and H. Wymeersch, "Implicit Cooperative Positioning in Vehicular Networks," *IEEE Transactions on Intelligent Transportation Systems,* vol. 19, p. 3964–3980, 2018.

[6] R. M. Buehrer, H. Wymeersch and R. M. Vaghefi, "Collaborative Sensor Network Localization: Algorithms and Practical Issues," *Proceedings of the IEEE,* vol. 106, p. 1089–1114, 2018.

[7] S. Thrun, Probabilistic robotics, Cambridge, Massachusetts: The MIT Press, 2006.

[8] M. Elazab, A. Noureldin and H. S. Hassanein, "Integrated cooperative localization for Vehicular networks with partial GPS access in Urban Canyons," *Vehicular Communications,* vol. 9, p. 242–253, 2017.

[9] F. Dellaert and M. Kaess, "Factor Graphs for Robot Perception," *Foundations and Trends in Robotics,* vol. 6, p. 1–139, 2017.

[10] N. Piperigkos, A. S. Lalos and K. Berberidis, "Graph based Cooperative Localization for Connected and Semi-Autonomous Vehicles," in *2020 IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2020.

[11] J. Schloemann and R. M. Buehrer, "On the Value of Collaboration in Location Estimation," *IEEE Transactions on Vehicular Technology,* vol. 65, p. 3585–3596, 2016.

[12] M. A. Spirito, "On the accuracy of cellular mobile station location estimation," *IEEE Transactions on Vehicular Technology,* vol. 50, p. 674–685, 2001.

[13] P. Tichavsky, C. H. Muravchik and A. Nehorai, "Posterior Cramer-Rao bounds for discrete-time nonlinear filtering," *IEEE Transactions on Signal Processing,* vol. 46, p. 1386–1396, 1998.

[14] F. Gustafsson and F. Gunnarsson, "Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements," *IEEE Signal Processing Magazine,* vol. 22, p. 41–53, 2005.

[15] H. Kim, S. H. Lee and S. Kim, "Cooperative localization with distributed ADMM over 5G-based VANETs," in *2018*

*IEEE Wireless Communications and Networking Conference (WCNC)*, 2018.

[16] H. Wymeersch, J. Lien and M. Z. Win, "Cooperative Localization in Wireless Networks," *Proceedings of the IEEE,* vol. 97, p. 427–450, 2009.

[17] C. Gao, Cooperative localization and navigation : theory, research, and practice, Boca Raton: Taylor & Francis, CRC Press, 2019.

[18] P. Yang, D. Duan, C. Chen, X. Cheng and L. Yang, "Multi-Sensor Multi-Vehicle (MSMV) Localization and Mobility Tracking for Autonomous Driving," *IEEE Transactions on Vehicular Technology,* vol. 69, p. 14355–14364, 2020.

[19] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS Strategies for Distributed Estimation," *IEEE Transactions on Signal Processing,* vol. 58, p. 1035–1048, 2010.

[20] S. Chouvardas, K. Slavakis and S. Theodoridis, "Adaptive Robust Distributed Learning in Diffusion Sensor Networks," *IEEE Transactions on Signal Processing,* vol. 59, p. 4692–4707, 2011.

[21] N. Piperigkos, A. S. Lalos, K. Berberidis, C. Laoudias and K. Moustakas, "5G Enabled Cooperative Localization of Connected and Semi-Autonomous Vehicles via Sparse Laplacian Processing," in *2020 22nd International Conference on Transparent Optical Networks (ICTON)*, 2020.

[22] N. Piperigkos, A. S. Lalos, K. Berberidis and C. Anagnostopoulos, "Cooperative Multi-Modal Localization in Connected and Autonomous Vehicles," in *2020 IEEE 3rd Connected and Automated Vehicles Symposium (CAVS)*, 2020.

[23] O. Sorkine, *Laplacian Mesh Processing,* The Eurographics Association, 2005.

[24] A. S. Lalos and K. Berberidis, "An efficient conjugate gradient method in the frequency domain: Application to channel equalization," *Signal Processing,* vol. 88, p. 99–116, 2008.

[25] Y. Saad, Iterative Methods for Sparse Linear Systems, CAMBRIDGE, 2003.

[26] A. H. Sayed, Adaptive Filters, John Wiley & Sons, Inc., 2008.

[27] A. Noureldin, T. B. Karamat and J. Georgy, Fundamentals of Inertial Navigation, Satellite-based Positioning and their Integration, Springer Berlin Heidelberg, 2013.

[28] J. Engel, V. Koltun and D. Cremers, "Direct Sparse Odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 40, p. 611–625, 2018.

[29] X. Ge, "Ultra-Reliable Low-Latency Communications in Autonomous Vehicular Networks," *IEEE Transactions on Vehicular Technology,* vol. 68, p. 5005–5016, 2019.

[30] F. Hua, R. Nassif, C. Richard, H. Wang and A. H. Sayed, "Diffusion LMS With Communication Delays: Stability and Performance Analysis," *IEEE Signal Processing Letters,* vol. 27, p. 730–734, 2020.

[31] Y. Yao, L. Rao, X. Liu and X. Zhou, "Delay analysis and study of IEEE 802.11p based DSRC safety communication in a highway environment," in *2013 Proceedings IEEE INFOCOM*, 2013.

[32] W. Ren and R. W. Beard, Distributed Consensus in Multi-vehicle Cooperative Control, Springer London, 2008.

[33] L. Wang, Model Predictive Control System Design and Implementation Using MATLAB®, Springer London, 2009.

[34] S. Boyd, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends® in Machine Learning,* vol. 3, p. 1–122, 2010.

[35] A. Kozma, J. V. Frasch and M. Diehl, "A distributed method for convex quadratic programming problems arising in optimal control of distributed systems," in *52nd IEEE Conference on Decision and Control*, 2013.

[36] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez and V. Koltun, "CARLA: An Open Urban Driving Simulator," in *Conference on Robot Learning*, 2017.