A Study of Different Observation Models for Cooperative Localization in Platoons

Elwan Héry, Philippe Xu, Philippe Bonnifait

A Study of Different Observation Models for Cooperative Localization in Platoons

Elwan Héry[1], Philippe Xu[2] and Philippe Bonnifait[2]

***Abstract*— Localization and perception for autonomous vehicles are often studied separately. However, they can be regroup on a dynamic map representing the environment of the vehicle. This dynamic map can be exchanged with other vehicles to be fused with their own dynamic maps to increase their accuracy. This paper presents a decentralized data fusion method for cooperative localization based on both Extended Kalman Filter and Covariance Intersection Filter. Different observation models of the relative measurements from the perception (Cartesian and polar relative poses, distances, bearings and relative yaws) are compared. The approach is tested on data for 10 vehicles generated from a real dataset and completed with a simulated perception.**

I. INTRODUCTION

Localization and perception are among the bigger challenges for autonomous vehicle's navigation. The ego-vehicle needs to be correctly localized, but also understand where the other vehicles around it are. To extend the perception range outside the Field of View (FoV) of the ego-vehicle, wireless communications between vehicles can be used. To improve the localization and perception at the same time, the information coming from localization, perception and communication can be fused to obtain a more accurate representation of the scene surrounding the ego-vehicle.

One of the first methods to solve cooperative localization issues, was to see the robot's formation as a rigid structure [1]. The robots are nodes of the rigid graph and the perception measurements such as bearing [2] or range [3], [4] correspond to the links. Graph optimization methods can then be applied to solve the cooperative localization issue. Another solution is to use filtering techniques on a global state composed of the states of every vehicle. An Extended Kalman Filter (EKF) can be used on each vehicle to obtain a decentralized approach.

When using a decentralized approach with communication, if the states are exchanged several times, these observations and the state of the ego-vehicle have correlated errors. This issue named data incest makes the EKF over-converge and the estimated state not consistent anymore. Different solutions exist, only raw measurements can be communicated, such as GNSS (Global Navigation Satellite System) pseudoranges [5] instead of the already fused states. Two EKF [6], [7] can also be used. One without the information of other vehicles (it gives the estimation of the state for the other vehicles) and one with this information for the ego-vehicle. To avoid the data incest issue, other filters exist, such as the Covariance Intersection Filter (CIF) [8]. The uncertainty of the estimation of this filter is often pessimistic. A solution to obtain a compromise between the EKF and the CIF is to use the Split Covariance Intersection Filter (SCIF) [9]. This filter has been used several times for cooperative localization [10], [11] as well as for cooperative perception [12]. For cooperative perception [13], deep learning fusion method exists with early, intermediate or late fusion [14], where raw data, feature maps or already detected objects are fused.

The cooperative localization data fusion algorithm presented in this work is based on multiple filters: an EKF for GNSS poses, kinematics (longitudinal speeds and yaw rates) and perception sensors observations and CIF for vehicles states received through communication. Both filters update the same state of the dynamic map of the ego-vehicle. In this paper, the main contributions are:

- The comparison of different observation models (Cartesian and polar relative poses, distances, bearings and relative yaws) for the relative measurements of the perception to improve the state of the vehicles both in terms of accuracy and consistency.
- The evaluation of the method on a hybrid simulation generated from a real dataset and completed with simulations to obtain a platoon of 10 vehicles. The perceptions are simulated with a FoV taking into account vehicles obstructions.

Compared to our previous work [15] the goal is, therefore, to study different observation models on a much longer platoon (10 vehicles instead of 2).

In Sec. II, the data fusion algorithm is presented globally and with its different steps: EKF prediction, EKF updates with the different observations models and CIF updates from the states of other vehicles. After this, in Sec. III the experimental and simulated setup is described before analyzing the obtained results.

II. DATA FUSION ARCHITECTURE

A. Global Architecture

The data fusion is organized with a decentralized architecture as shown in Fig. 1. Each vehicle estimates its state (its position, yaw, longitudinal speed and yaw rate) and the state of every vehicle of the platoon (composed of the same elements) as described in Tab. I. Each vehicle has, therefore, a global state of all the dynamics agents (only

[1]Nantes Université, École Centrale de Nantes and CNRS LS2N, 44300 Nantes, France first-name.last-name@ec-nantes.fr

[2]Université de technologie de Compiègne, CNRS, Heudiasyc UMR 7253, France firstname.surname@hds.utc.fr

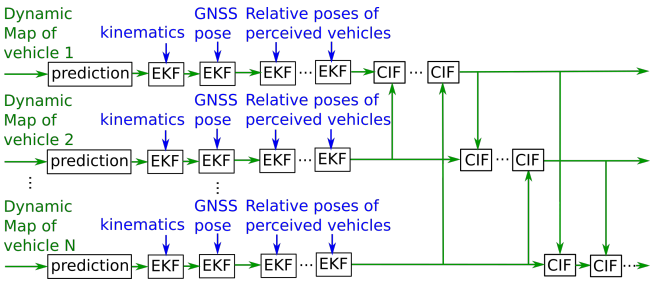


Fig. 1: Global architecture of the dynamic maps updates of N vehicles based on EKF predictions and updates and CIF updates. The dynamic map states are in green and the observations in blue.

TABLE I: State of the vehicle dynamic map

Dynamic map: X					
...	Vehicle i state: X_i				...
	pose: q_i		velocity: u_i		
	position: p_i	yaw: θ_i	speed:v_i	yaw rate: w_i	
	x_i	y_i			

vehicles in our work) surrounding it. We call this global state a dynamic map, as it represents the dynamic environment of the ego-vehicle. By grouping all these elements into one dynamic map, each vehicle is able to obtain a high-level representation of the dynamic agents surrounding it. The localization, perception and communication information are fused to obtain a more accurate representation of this dynamic map. To model the uncertainty of the dynamic map, a covariance matrix will be estimated as well. In this paper, we use the notation Σ_Z for the covariance matrix of any variable Z.

We suppose in this work that all the observations are synchronized. As the data used for the evaluation of our approach will be partially simulated to obtain more vehicles, this assumption greatly simplified the architecture. A more complex asynchronous cooperative localization architecture was studied in [15]. It is not in the scope of this work. This choice was made to obtain a repeatable algorithm that loads the observations one after the other to obtain results that will not change, if the computation time is slightly different. Indeed, if each EKF update is done in a different thread, depending on the computation time of each thread, the different updates would be applied with a different order, which will give different results. This was one of the main issue of [15] (much realistic but also less easy to study the results). A linear architecture, as shown in this work, gives repeatable results that can be easily compared.

In Fig. 1, one can see the different steps to update the state of each vehicle with the different observations. For each time step, the state is first predicted at the new time step with an EKF evolution model. After this, the state is updated with an EKF with the observation model of the kinematics (longitudinal speed and yaw rate) of the GNSS pose and of every perceived vehicle. Each relative measurement of the perception is taken into account one after the other on different EKF updates. We choose to keep sequential updates of the observations instead to fuse all the synchronized observations on one update, as it would be the case on an asynchronous architecture. This choice gives a more modular architecture, able to work even if one observation is missing, without complexification of the architecture, as it would be the case in an asynchronous architecture. Finally, the communication between vehicles is taken into account with a Covariance Intersection Filter (CIF) to avoid data incest.

B. EKF prediction

The first step is to predict the state of the dynamic map at the new time step. The following evolution model is used:

$$X_{t_k} = \text{evo}(X_{t_{k-1}}) = X_{t_{k-1}} + V_{t_{k-1}} \tag{1}$$

For the vehicle i of the dynamic map of the ego-vehicle:

$$\begin{pmatrix} \vdots \\ x_i \\ y_i \\ \theta_i \\ v_i \\ \omega_i \\ \vdots \end{pmatrix}_{t_k} = \begin{pmatrix} \vdots \\ x_i \\ y_i \\ \theta_i \\ v_i \\ \omega_i \\ \vdots \end{pmatrix}_{t_{k-1}} + \begin{pmatrix} \vdots \\ v_i \Delta t \cos(\theta_i + \frac{\Delta t}{2}\omega_i) \\ v_i \Delta t \sin(\theta_i + \frac{\Delta t}{2}\omega_i) \\ \omega_i \Delta t \\ 0 \\ 0 \\ \vdots \end{pmatrix}_{t_{k-1}} \tag{2}$$

All elements of the arrays are at the time of the indices of the arrays.

The dynamic map X_{t_k} of the ego-vehicle at the time t_k of the step k is obtained from the dynamic map $X_{t_{k-1}}$ of the ego-vehicle at the time t_{k-1} from this previous step and the motion $V_{t_{k-1}}$ of every vehicle computed from the longitudinal speeds, yaw rates and yaws at the previous time step t_{k-1} and with the time delay $\Delta t = t_k - t_{k-1}$. To compute the new covariance matrix for the dynamic map, we use the following prediction model:

$$\Sigma_{X_{t_k}} = J_{\text{evo}} \Sigma_{X_{t_{k-1}}} J_{\text{evo}}^T + \Sigma_{\text{evo}} \Delta t \tag{3}$$

where J_{evo} is the Jacobian matrix of the evolution model $\text{evo}(X_{i,t_{k-1}})$ previously defined and Σ_{evo} the covariance matrix of the evolution model noise.

C. EKF updates

Once the state of the dynamic map of the vehicle is at the new time, we can update it with the different observations. To simplify the notations we remove the time t_k, as the state and the observations are all at this time in the rest of the paper. To update the dynamic map of the ego-vehicle, we first compute the innovation from the observation Z and the observation model $\text{obs}(X)$. The covariance of the innovation is also estimated from the Jacobian matrix of the observation model J_{obs} and the covariance matrix of the observation noise Σ_Z:

$$Y = Z - \text{obs}(X) \tag{4}$$

$$\Sigma_Y = J_{\text{obs}} \Sigma_X J_{\text{obs}}^T + \Sigma_Z \tag{5}$$

Then, the Kalman gain is computed:

$$K = \Sigma_X J_{\text{obs}}^T \Sigma_Y^{-1} \tag{6}$$

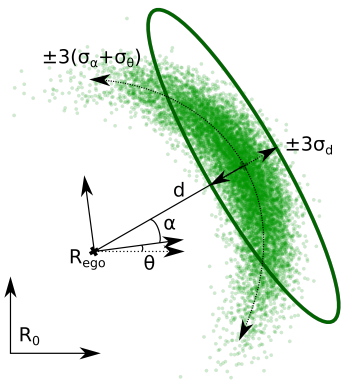


Fig. 2: Polar representation with the distance d, yaw θ and bearing α and with its uncertainty represented by the banana shape and the ellipse of uncertainty approximating this distribution.

Finally, the state and the covariance matrix are updated:

$$\widehat{X} = X + KY \tag{7}$$

$$\widehat{\Sigma}_X = (I - KJ_{\text{obs}})\Sigma_X(I - KJ_{\text{obs}})^T + K\Sigma_Z K^T \tag{8}$$

The update of the EKF is used for all the following observations: we note ${}^{j}z_i$ the variable z of vehicle i (with the notation "ego" for the ego-vehicle) in the frame of vehicle j (this exponent is removed if it is in the global frame).

1) Kinematics: The kinematics containing the longitudinal speed and yaw rate with the observation model are:

$$\text{obs}_{\text{vel}}(X) = \begin{pmatrix} v_{\text{ego}} \\ \omega_{\text{ego}} \end{pmatrix} \tag{9}$$

2) GNSS pose: The pose computed by the GNSS receiver is updated with the observation model:

$$\text{obs}_{\text{pose}}(X) = \begin{pmatrix} x_{\text{ego}} \\ y_{\text{ego}} \\ \theta_{\text{ego}} \end{pmatrix} \tag{10}$$

*3) **Relative measurement of the** perception:* For each perceived vehicle i, detected by the ego-vehicle, we can define a different observation model of the relative pose ${}^{\text{ego}}q_i$ of this vehicle i in the mobile frame of the ego-vehicle:

$$\text{obs}_{\text{pose\_cartesian}}(X) = {}^{\text{ego}}q_i = \begin{pmatrix} {}^{\text{ego}}x_i \\ {}^{\text{ego}}y_i \\ {}^{\text{ego}}\theta_i \end{pmatrix} = R_{\text{ego}}(q_i - q_{\text{ego}}) \tag{11}$$

with the rotation matrix:

$$R_{\text{ego}} = \begin{pmatrix} \cos\theta_{\text{ego}} & \sin\theta_{\text{ego}} & 0 \\ -\sin\theta_{\text{ego}} & \cos\theta_{\text{ego}} & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{12}$$

However, we showed in [16] the issues of propagating a pose with uncertainty. The obtained distribution at the end of the transformation is not a Gaussian distribution anymore but a banana shaped one, illustrated in Fig. 2 (the uncertainties are exaggerated in this figure to better observe this shape). In this filter, the observation model $\text{obs}_{\text{pose\_cartesian}}(X)$ gives an uncertainty distribution like this. To better understand this shape, another solution is to represent the observation by a polar relative pose. With this new observation, we have the following observation model:

$$\begin{aligned} \text{obs}_{\text{pose\_polar}}(X) &= \begin{pmatrix} {}^{\text{ego}}r_i \\ {}^{\text{ego}}\alpha_i \\ {}^{\text{ego}}\theta_i \end{pmatrix} \\ &= \begin{pmatrix} \sqrt{(x_i - x_{\text{ego}})^2 + (y_i - y_{\text{ego}})^2} \\ \arctan 2(y_i - y_{\text{ego}}, x_i - x_{\text{ego}}) - \theta_{\text{ego}} \\ \theta_i - \theta_{\text{ego}} \end{pmatrix} \end{aligned} \tag{13}$$

One can see in Fig. 2, that the non-linearity creating a banana shaped distribution instead of Gaussian distribution, are linked to the yaw and relative yaw angle uncertainties. It is also possible to define observation models with only the distance ${}^{\text{ego}}d_i$ to be independent of these angles or with only the bearing angle ${}^{\text{ego}}\alpha_i$ or the relative yaw ${}^{\text{ego}}\theta_i$ if we want to observe their effects. To obtain the associated observation model, we just have to keep the correct line of the polar observation model $\text{obs}_{\text{pose\_polar}}(X)$.

D. CIF updates

We suppose in this work that every vehicle is able to communicate its dynamic map to all the other vehicles. To use these new observations while avoiding data incest, we use a Covariance Intersection Filter (CIF) instead of an EKF update. As we supposed that each vehicle is able to communicate with every vehicle and that all vehicles are present in the same order inside each dynamic map, the observation model is the identity.

To update the dynamic map with the CIF, we first compute the innovation from the state X (the dynamic map of the ego-vehicle) and the observation Z (the dynamic map communicated by another vehicle):

$$Y = Z - X \tag{14}$$

Then, we search the weight $w \in [0, 1]$ that minimizes the following cost function:

$$\det\left(\widehat{\Sigma}_X\right) \tag{15}$$

with

$$\widehat{\Sigma}_X^{-1} = w\Sigma_X^{-1} + (1 - w)\Sigma_Z^{-1} \tag{16}$$

Once the optimal weight $\hat{w}$ is estimated, we can update the state and its covariance matrix:

$$\widehat{\Sigma} = (\hat{w}\Sigma_X^{-1} + (1 - \hat{w})\Sigma_Z^{-1})^{-1} \tag{17}$$

$$\widehat{X} = \widehat{\Sigma}_X(\hat{w}\Sigma_X^{-1}X + (1 - \hat{w})\Sigma_Z^{-1}Z) \tag{18}$$

As this data fusion method includes an optimization, this is the step that takes the longest time in the complete data fusion architecture. Our implementation is coded in python and is not optimized. The computation time is, therefore, not the one to expect from a real-time application. However, one can see in this architecture, when $N - 1$ vehicles can communicate with the ego-vehicle, $N - 1$ CIF updates are required with a state and measurement of size $5 \times N$ (as one can see on Tab. I). This means that the computation time will

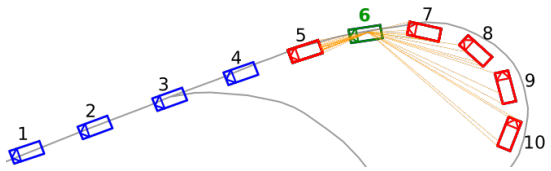


Fig. 3: The simulated field of view of vehicle 6 (with the green bounding box) detects the red bounding boxes of the other vehicles by ray casting (in orange).

greatly increase when the number of communicating vehicles increases.

III. Experimental results

A. Experimental scenarios

The experimentation is based on real data from the dataset that we published[1] with [15]. This dataset is composed of the sensors data of a Renault Zoe car recorded on 10 laps of a test road of the Heudiasyc lab in Compiègne, France. This road is composed of two roundabouts with a straight line between them. In this dataset, the following sensor data are used for this work:

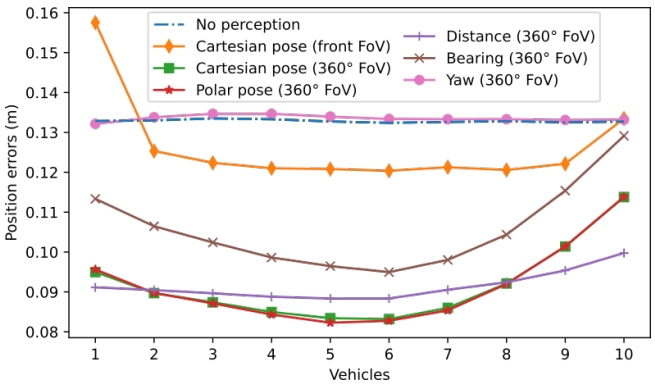
- GNSS poses from an Ublox M8T (unbiased with a mean filter with a 30 s rolling window).
- Longitudinal speeds and yaw rates computed from the odometry and IMU coming from the CAN bus of the vehicle.
- Ground truth of all the previous information from a SPAN CPT an IMU / GNSS receiver with RTK corrections.

A platoon of 10 vehicles is generated from this vehicle, by replaying the same data with a different time delay for each vehicle. Finally, the relative poses from the perception between two vehicles (which cannot be generated with this method) are simulated from the ground truth using the same transformation as Eq. (11). Gaussian noises with a standard deviation of 5 cm for the relative positions coordinates and 0.05 rad for the relative yaws are added to these relative poses. The 360° field of view (FoV) of the perception sensor of each vehicle is simulated by ray casting to take into account the obstructions of the other vehicles, as shown in Fig. 3. One can see on this figure, when the vehicles are on the roundabout, more vehicles can be detected than when the vehicles are on the straight lane. This 360° FoV is compared to a limited FoV where only the vehicle in front of the ego-vehicle is detected. In practice, these relative measurements could be obtained with perception sensors more accurate to estimate bearings, such as cameras, or distances, such as LiDAR. We chose a noise model of what can be expected from a LiDAR, similarly to what we observed in [17].

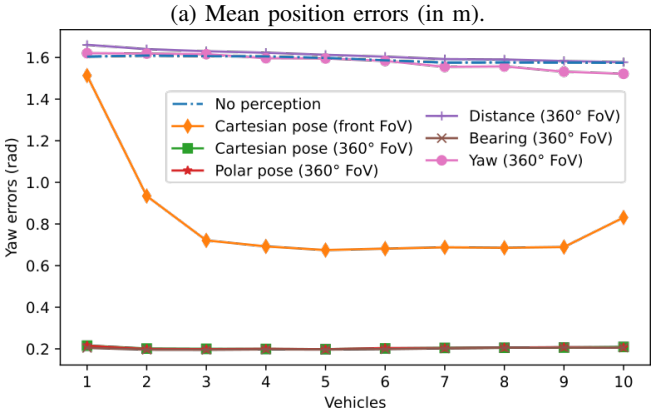
B. Global localization

The different results of this paper are computed with the observation models of Cartesian relative pose "Cartesian pose (front FoV)" with the limited front FoV and Cartesian

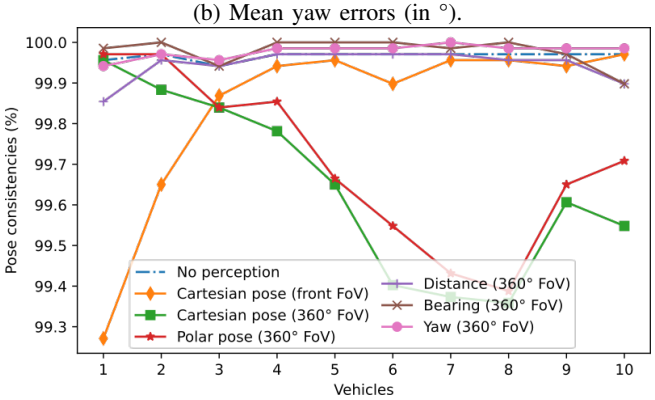
[1] https://datasets.hds.utc.fr/project/2



(a) Mean position errors (in m).



(b) Mean yaw errors (in °).



(c) Consistency of the poses (in %).

Fig. 4: Global localization of the ego-vehicle from each dynamic map for the different observation models of the perception.

relative pose "Cartesian pose (360° FoV)", polar relative pose "polar pose (360° FoV)", distance "Distance (360° FoV)", bearing "Bearing (360° FoV)" and relative yaw "Yaw (360° FoV)" for the 360° FoV. Fig. 4a shows the mean of the norms of the positions errors ϵ_p and Fig. 4b, the mean of the absolute value of the yaw errors ϵ_θ computed from the estimates ($\hat{p}$ and $\hat{\theta}$) with respect to the ground truth (p_{ref} and θ_{ref}):

$$\epsilon_p = \overline{||\hat{p} - p_{\text{ref}}||}\ , \quad \epsilon_\theta = \overline{|\hat{\theta} - \theta_{\text{ref}}|} \tag{19}$$

We note $\overline{\cdot}$ the mean value on all the dataset time steps. Fig. 4c shows the consistence C for a 95% probability of

success of a χ^2 test:

$$C = \overline{(\hat{q} - q_{\text{ref}})^T (\widehat{\Sigma}_q + \Sigma_{q_{\text{ref}}})^{-1} (\hat{q} - q_{\text{ref}}) < \chi^2_{3;0.05}} \quad (20)$$

One can see in Fig. 4a and Fig. 4b, that the perception with relative poses observations (Cartesian or polar) increase the accuracy of the position and yaw of the vehicle. This accuracy already starts to increase when only one vehicle is detected by the ego-vehicle (with the front FoV) and is even better when many vehicles can be perceived all around the ego-vehicle (with the 360° FoV). These results can be seen in particular for vehicles in the center of the platoon, as they are able to perceive more vehicles than the one in front or at the back of the platoon. One can see when using only the distances, that the position accuracy increases compare to no perception but not the yaw accuracy. When using the bearings, both the position and yaw accuracy increase, even if the position accuracy increases less than when using distances. The results for these two sub-observations could be expected from Eq. 13, as the distance component depends on the positions and the bearing depends on the positions and ego yaw. The yaw observations, however, do not have any influence on the position or yaw accuracies. One can also see that, to be able to increase correctly the position accuracy with only a limited perception like distances or bearings, it is necessary to observed multiple vehicles (like the vehicles in the center of the platoon). For every observation models, one can see on Fig. 4c that the pose is always consistent and even pessimistic. Fig. 5 also illustrates these results for the vehicle 6 on the total duration of the dataset. One can see that for every observation models, the position coordinates and yaw errors are able to stay inside their uncertainty intervals.

C. Relative localization

Fig. 6a shows the mean of the norms of the relative positions errors and Fig. 6b shows the mean of the absolute value of the relative yaw errors with respect to the ground truth. Finally, Fig. 6c shows the relative consistency for a 95% probability. They are computed similarly to the global localization for each relative pose in the frame of the vehicle 6 from its dynamic map instead of the global frame. No result is displayed for vehicle 6 as it is the ego-vehicle used to compute the relative poses. One can see that, when using relative poses observations, the relative localization is greatly improved, for the position and yaw errors. We can also see that the reduced models, such as distance or yaw models, with less information are less accurate. However, the bearing increase significantly the relative position and yaw accuracies. Even if the global localization is more accurate globally with the distance model compare to no perception, its relative localization is not improved. One can see in these results that the vehicles in the middle of the platoon are also more accurate than the one on the extremities, similarly to the global localization. This result is also present when no perception is available, as the relative poses with the vehicles near the ego-vehicle are more accurate when they are computed with the same transformation as Eq. (11).

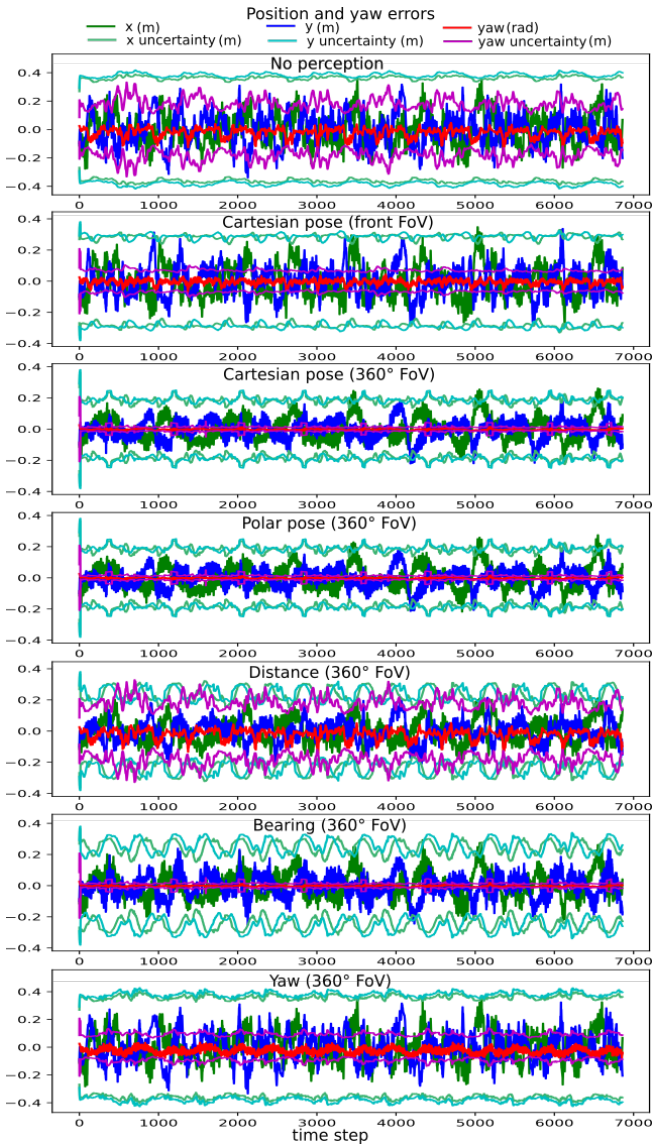
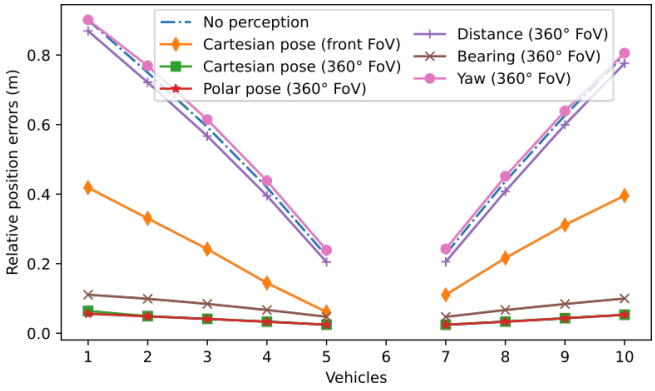


Fig. 5: Position and yaw errors of the estimated poses of vehicle 6 in its own dynamic map with their $\pm 2\sigma$ uncertainty intervals in function of the time steps for every observation models of the perception.

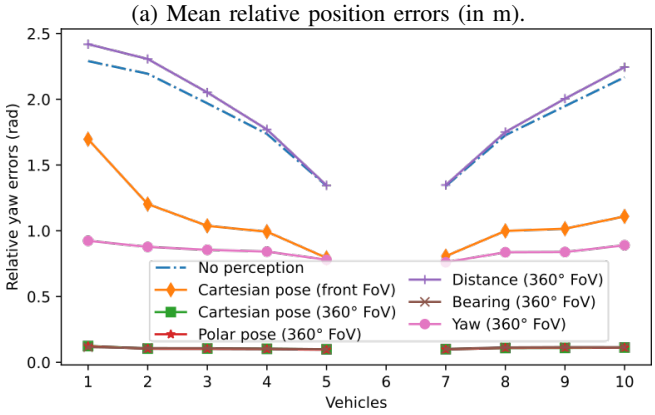
Similarly to the global localization, the relative localization is consistent and even pessimistic for every observation models.

IV. Conclusion

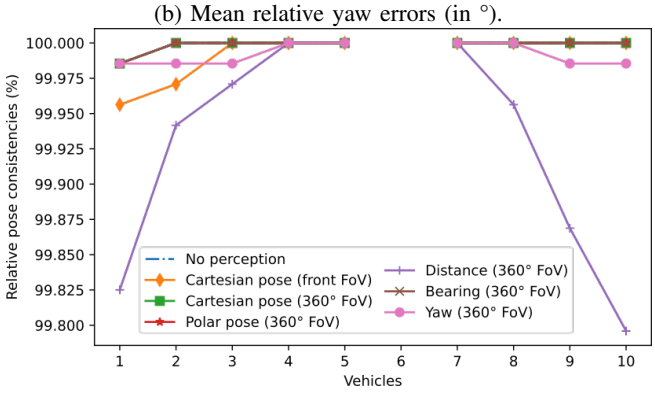
This paper described a distributed data fusion method for cooperative localization and perception. This approach was tested on a platoon composed of 10 vehicles not always able to see every other vehicles. The results of this work presents a comparison of different observation models (Cartesian and polar relative poses, distances, bearings and relative yaws). One can see in these results that the best models are the Cartesian or polar relative pose perception models, as they can greatly increase the global and relative localization while staying consistent. This could be expected, as these models contain more information than the others (three parameters



(a) Mean relative position errors (in m).



(b) Mean relative yaw errors (in °).



(c) Consistency of the relative poses (in %).

Fig. 6: Relative localization of each vehicle in the frame of vehicle 6 from the dynamic map of vehicle 6 for the different observation models of the perception.

instead of one). If we decompose the relative pose model into distance and bearing models, the distances improve the global position only while the bearings improve both the global and relative positions. This study enables to better understand the influence of the different parts of the relative pose on the global and relative localization for our filter.

In future works, we would like to extend the dynamic map with diverse dynamic agents (e.g., vehicles but also pedestrians or bicycles) and complete it with static information to obtain a global representation of the static and dynamic environment of the autonomous vehicle.

Acknowledgment

This work was carried out in the framework of the NExT Senior Talent Chair DeepCoSLAM, which weas funded by the National Agency for Research (ANR-16-IDEX-0007), and with the support of Région Pays de la Loire and Nantes Métropole.

References

[1] A. Ahmad, G. D. Tipaldi, P. Lima, and W. Burgard, "Cooperative robot localization and target tracking based on least squares minimization," in *IEEE International Conference on Robotics and Automation*, May 2013, pp. 5696–5701.

[2] D. Zelazo, A. Franchi, and P. R. Giordano, "Rigidity Theory in SE(2) for Unscaled Relative Position Estimation using only Bearing Measurements," in *European Control Conference*, 2014, pp. 2703–2708.

[3] D. Zelazo, A. Franchi, H. H. Bülthoff, and P. R. Giordano, "Decentralized Rigidity Maintenance Control with Range Measurements for Multi-Robot Systems," *The International Journal of Robotics Research*, vol. 34, no. 1, pp. 105–128, 2015.

[4] G. M. Hoangt, B. Denis, J. Härri, and D. Slock, "Cooperative Localization in VANETs: An Experimental Proof-of-Concept Combining GPS, IR-UWB Ranging and V2V Communications," in *Workshop on Positioning, Navigation and Communications*, 2018, pp. 1–6.

[5] K. Lassoued, I. Fantoni, and P. Bonnifait, "Mutual Localization and Positioning of Vehicles Sharing GNSS Pseudoranges: Sequential Bayesian Approach and Experiments," in *IEEE International Conference on Intelligent Transportation Systems*, 2015, pp. 1896–1901.

[6] R. Aufrère, N. Karam, F. Chausse, and R. Chapuis, "A state exchange approach in real conditions for multi-robot cooperative localization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 4346–4351.

[7] N. Karam, F. Chausse, R. Aufrere, and R. Chapuis, "Localization of a Group of Communicating Vehicles by State Exchange," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 519–524.

[8] S. Julier and J. Uhlmann, "A non-divergent estimation algorithm in the presence of unknown correlations," in *American Control Conference*, vol. 4, 1997, pp. 2369–2373.

[9] ——, "General Decentralized Data Fusion with Covariance Intersection (CI)," in *Handbook of Multisensor Data Fusion, Theroy and Practice*, 2001.

[10] H. Li and F. Nashashibi, "Cooperative multi-vehicle localization using split covariance intersection filter," in *IEEE Intelligent Vehicles Symposium*, 2012, pp. 211–216.

[11] T. R. Wanasinghe, G. K. Mann, and R. G. Gosine, "Decentralized Cooperative Localization for Heterogeneous Multi-robot System Using Split Covariance Intersection Filter," in *Canadian Conference on Computer and Robot Vision*, May 2014, pp. 167–174.

[12] A. Lima, P. Bonnifait, V. Cherfaoui, and J. A. Hage, "Data Fusion with Split Covariance Intersection for Cooperative Perception," in *IEEE International Intelligent Transportation Systems Conference*, 2021, pp. 1112–1118.

[13] A. Caillot, S. Ouerghi, P. Vasseur, R. Boutteau, and Y. Dupuis, "Survey on Cooperative Perception in an Automotive Context," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 14 204–14 223, 2022.

[14] S. Ren, S. Chen, and W. Zhang, "Collaborative Perception for Autonomous Driving: Current Status and Future Trend," in *Chinese Conference on Swarm Intelligence and Cooperative Control*, 2023, pp. 682–692.

[15] E. Héry, P. Xu, and P. Bonnifait, "Consistent decentralized cooperative localization for autonomous vehicles using LiDAR, GNSS, and HD maps," *Journal of Field Robotics*, vol. 38, no. 4, pp. 552–571, 2021.

[16] ——, "Pose and covariance matrix propagation issues in cooperative localization with LiDAR perception," in *IEEE Intelligent Vehicles Symposium*, 2019, pp. 1219–1224.

[17] ——, "LiDAR based relative pose and covariance estimation for communicating vehicles exchanging a polygonal model of their shape," in *10th Workshop on Planning, Perception and Navigation for Intelligent Vehicles*, 2018, pp. 65–70.