Data Incest in Cooperative Localisation with the Common Past-Invariant Ensemble Kalman Filter

Jan Čurn[1], Dan Marinescu[1], Niall O'Hara[1,2], and Vinny Cahill[1,2]
[1]School of Computer Science and Statistics
Trinity College Dublin, Ireland
[2]Lero - The Irish Software Engineering Research Centre
{jan.curn,dan.marinescu,niohara,vinny.cahill}@scss.tcd.ie

***Abstract*—In this paper we consider the problem of cooperative vehicle localisation, in which a group of vehicles are driving in an outdoor environment, each estimating their position using a global positioning system (GPS) and odometry. Additionally, the vehicles can improve their estimates by observing positions of other vehicles using a proximity sensor, such as a radar, and a mutual communication, which is especially helpful to those vehicles operating in areas with no GPS coverage.**

In a distributed fusion system, each vehicle needs to account for the fact that information received from other vehicles might originate in part from the vehicle itself, resulting in a correlation between the state estimate and observation errors. This problem, also known as *data incest*, is amplified by the dynamic and unstructured nature of the communication topology, inherent to a cooperative localisation scenario.

We provide a novel solution to the problem based on the Common Past-Invariant Ensemble Kalman filter (CPI-EnKF) - a generalisation of the Ensemble Kalman filter that can be applied in the presence of common past information shared between the state estimate and the observation, which has been recently proposed by this paper's authors. As we will demonstrate, the CPI-EnKF is simpler to apply, provides better estimates, can be scaled to an arbitrary number of vehicles and is computationally more efficient than other similar methods.

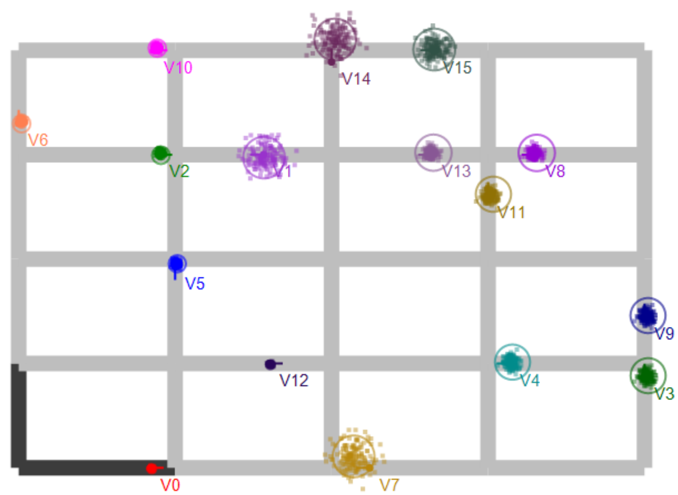

Fig. 1. A configuration of the roads in the evaluation scenario, including an example snapshot of the vehicles from a simulation run. The darker segment in the bottom left section represents the only road with a GPS coverage. The points depict the current vehicle position estimates (*ensembles*) of the respective CPI-EnKF filters, while the solid ellipses depict the 95% confidence regions of the (apparently more conservative) Split CI estimates.

I. Introduction

The Kalman filter, originally introduced in [1], and all its variants represent arguably the most popular and widely applied class of algorithms to estimate the state of a physical system from a sequence of noisy sensor observations. In the traditional definition of the filter, the sensor observations are assumed to be affected by a white Gaussian noise, which is statistically independent of the current state estimate. Unfortunately, in many practical applications, this assumption is not satisfied and consequently, the filter may provide overconfident state estimates and ultimately diverge.

If a correlation between the current state estimate and the observation is present, the traditional Kalman filter update equation can be replaced with a generalised equation [2, Ch. 5]. However, this equation requires a value of the cross-covariance between the state estimate and observation errors, which, bar a few special cases, is rather difficult to compute analytically. For example, if the correlation is caused by a well-modelled sequential correlation of the observation noise process, an algebraic reordering of the update equation might reveal the required cross-covariance term, or eliminate it altogether. This is the base principle behind the *state augmentation* and the *measurement differencing* approaches [3, Ch. 7].

The correlation between the state estimate and observation errors can also be caused by the presence of common past information shared between the two. This so-called *data incest* problem, also known as *double-counting* or a problem of *mutual information*, occurs whenever information derived from a single raw sensor measurement, directly or indirectly, has a chance to affect a state estimate multiple times. This is a well-know problem in decentralised sensor networks, where each node maintains an estimate of the state of some physical phenomenon and communicates it with neighbours. If the communication topology contains no loops, a node can keep track of the information sent to neighbouring nodes and ”subtract” it from information it receives back. Unfortunately, this approach, which is also known as *channel filtering* [4], cannot be used in cooperative localisation, because the communication topology will typically contain loops.

Alternatively, the data incest problem can be addressed from a larger perspective. For example, if the state of all nodes in a wireless sensor network was modelled as one large system, the state of such a system could be estimated by a single Kalman filter and the observations performed by the nodes could be considered independent to the state. Unfortunately, such a centralised approach requires communication of all sensor measurements to a single fusion center, which is not practical in large networks and also, it exposes the network to the risk of a single point of failure. Therefore, numerous approaches have been proposed to distribute the centralised filter to the nodes in the network, which became known as *decentralised Kalman filters* [5], [6], [7]. The approaches in literature vary greatly in their communication patterns, accuracy of the estimates provided, and requirements of knowledge on the network structure. We will describe several such approaches, which are applicable to the problem of cooperative localisation, in greater detail in Section III.

The *covariance intersection* (CI) algorithm, introduced in [8], is a generic data fusion approach that effectively replaces the Kalman filter update rule, assuming the presence of a correlation between the state estimate and the observation of an unknown magnitude. However, the CI algorithm has two principal issues: First, if the exact correlation is known, the estimate provided by the CI is typically overly pessimistic and not optimal. Second, in order to to compute an optimal weight parameter, an additional non-linear optimisation step is necessary [9].

In this paper, we will address the problem of data incest in the context of cooperative localisation using the Common Past-Invariant Ensemble Kalman filter (CPI-EnKF). The CPI-EnKF is a generalisation of the Ensemble Kalman filter (EnKF), recently introduced by this paper's authors in [10], which provides an optimal Kalman filter update rule even in the presence of a correlation between the state estimate and observation errors. It retains all the fundamental advantages of the EnKF, such as favourable asymptotic performance and high accuracy with non-linear models. We will demonstrate that the CPI-EnKF is equally scalable and as simple to apply as the CI algorithm, while it provides more accurate state estimates.

In Section II we provide a brief overview of the EnKF and the CPI-EnKF algorithms, including references to more detailed literature. Section III discusses the problem of cooperative localisation in detail, and provides an overview of the state-of-the-art approaches that address it. Our new approach to the problem based on the CPI-EnKF is then described in Section IV. In Section V, we present a simulation scenario that was used for the evaluation of the new algorithm, describe the details of its implementation and the implementation of other benchmark algorithms, and present the evaluation results. Finally, the paper is concluded in Section VI.

II. Common Past-Invariant Ensemble Kalman filter

The Ensemble Kalman filter (EnKF), originally proposed in [11], is a Monte Carlo variant of the Kalman filter, which represents the state estimates and observations using a set of random samples. Due to such a representation, the EnKF possesses three fundamental benefits compared to other variants of the Kalman filter [12], [13]:

- The computational and space complexity of both the prediction and update operations scales linearly with the number of the state space dimensions, as opposed to the more popular Extended Kalman filter (EKF) and Unscented Kalman filter (UKF), for which these operations generally scale no better than quadratically.
- The EnKF supports non-linear prediction and observation models with an accuracy only limited by the number of Monte Carlo samples, and is therefore potentially more precise than both the EKF and UKF.
- Unlike the EKF, the EnKF does not need Jacobians of the prediction and observation models, which might be difficult to compute per se.

The features of the EnKF led to its widespread adoption in the Earth sciences, where models are often non-linear and high-dimensional, and thus other Kalman filter variants are not applicable. For example, the EnKF became a de facto standard tool for data assimilation in numerical weather forecasting. In this paper, we aspire to demonstrate that the EnKF is also a useful tool for real-time engineering applications, such as the intelligent transportation systems and robotics.

In the EnKF, the current state estimate is represented using a set of random samples $\mathbf{x}_1, \ldots, \mathbf{x}_N \in \mathbb{R}^n$ (column vectors), called an *ensemble*, which is organised in a matrix $\mathbf{X} \in \mathbb{R}^{n \times N}$ as:

$$\mathbf{X} = [\, \mathbf{x}_1, \ldots, \mathbf{x}_N \,] \tag{1}$$

where n is the number of state-space dimensions and N is the number of random samples. With such a representation of the state estimate, the prediction step of the filter can be facilitated simply by applying the prediction model $f\colon \mathbb{R}^n \to \mathbb{R}^n$ to each ensemble member independently, in order to obtain the *prior ensemble* matrix $\mathbf{X}^- \in \mathbb{R}^{n \times N}$ as:

$$\mathbf{X}^- = [\, f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N) \,] \tag{2}$$

The prediction model f can internally utilise a control input, add random noise to the input state sample, and can be non-linear. Note that the accuracy with which the ensemble represents a probability distribution associated with the estimate is only determined by the number of ensemble members N; the representation is perfect in the limit of an infinite ensemble. For brevity, in the remainder of this paper, the prior ensemble will also be denoted simply as $\mathbf{X}$, without any confusion.

Similarly, an observation of the physical system is represented using a set of random samples $\mathbf{d}_1, \ldots, \mathbf{d}_N \in \mathbb{R}^m$ organised in a matrix $\mathbf{D} \in \mathbb{R}^{m \times N}$:

$$\mathbf{D} = [\, \mathbf{d}_1, \ldots, \mathbf{d}_N \,] \tag{3}$$

where m denotes the number of observation dimensions. Typically, a raw sensor measurement only consists of a single m-dimensional vector, and therefore, the matrix $\mathbf{D}$ needs to be generated from such a vector by adding a random noise with a probability distribution corresponding to the statistical properties of the sensor.

The relation between the state estimate and the observation is given by an observation model $h\colon \mathbb{R}^n \to \mathbb{R}^m$, which is a function that maps a state vector to an expected observation vector. The observation model can be used to compute the expected observation ensemble $h(\mathbf{X}) \in \mathbb{R}^{m\times N}$ given the state estimate ensemble $\mathbf{X}$ as:

$$h(\mathbf{X}) = [\, h(\mathbf{x}_1), \ldots, h(\mathbf{x}_N)\,] \tag{4}$$

Again, the function h can be non-linear.

The update step of the filter refines the prior ensemble $\mathbf{X}$ by accommodating the observation $\mathbf{D}$, which results in the *posterior ensemble* $\mathbf{X}^+ \in \mathbb{R}^{n\times N}$ with a potentially lower uncertainty than $\mathbf{X}$. The update step is performed using the following equation:

$$\mathbf{X}^+ = \mathbf{X} + \mathbf{K}(\mathbf{D} - h(\mathbf{X})) \tag{5}$$

where $\mathbf{K} \in \mathbb{R}^{n\times m}$ is the *Kalman gain* factor computed as:

$$\mathbf{K} = \operatorname{cov}(\mathbf{X}, h(\mathbf{X}))\,[\, \operatorname{cov}(h(\mathbf{X})) + \operatorname{cov}(\mathbf{D})\,]^{-1} \tag{6}$$

Note that $\operatorname{cov}(\mathbf{A}, \mathbf{B})$ denotes a cross-covariance between two ensembles $\mathbf{A}$ and $\mathbf{B}$, which is estimated as:

$$\operatorname{cov}(\mathbf{A}, \mathbf{B}) = \mathrm{E}\left[(\mathbf{A} - \mathrm{E}[\mathbf{A}])(\mathbf{B} - \mathrm{E}[\mathbf{B}])^{\mathrm{T}}\right] \tag{7}$$

and $\mathrm{E}[\mathbf{C}]$ denotes the expected value (arithmetic mean) of an ensemble $\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_N]$, computed as:

$$\mathrm{E}[\mathbf{C}] = \frac{1}{N}\sum_{i=1}^{N} \mathbf{c}_i \tag{8}$$

For brevity, a shortened notation $\operatorname{cov}(\mathbf{A})$ is used instead of $\operatorname{cov}(\mathbf{A}, \mathbf{A})$.

The update Equation (5) is based on the traditional Kalman filter assumptions, such as that all the involved probability distributions are Gaussian, the observation model h is linear, and that the state estimate $\mathbf{X}$ and the observation $\mathbf{D}$ are statistically independent. Furthermore, due to its Monte Carlo nature, the equation only provides an optimal state estimate in the limit of an infinite ensemble. Although these assumptions are quite restrictive, like other Kalman filter variants, an EnKF with a moderate number of ensemble members provides reasonable state estimates even in situations where the assumptions are not exactly satisfied, i.e. in most practical applications. Nevertheless, as argued in Section I, the presence of a correlation between $\mathbf{X}$ and $\mathbf{D}$ can cause the filter to provide overconfident estimates and diverge.

The Common Past-Invariant Ensemble Kalman filter (CPI-EnKF), recently proposed in [10], is a generalisation of the EnKF that lifts the assumption of independence between the state estimate $\mathbf{X}$ and the observation $\mathbf{D}$, which also implies independence between $h(\mathbf{X})$ and $\mathbf{D}$. Instead, the CPI-EnKF assumes that both $h(\mathbf{X})$ and $\mathbf{D}$ can be decomposed into two independent additive components: a shared zero-mean error term $\mathbf{\Sigma} \in \mathbb{R}^{m\times N}$, and terms $\Delta^{h(\mathbf{X})} \in \mathbb{R}^{m\times N}$ and $\Delta^{\mathbf{D}} \in \mathbb{R}^{m\times N}$, respectively:

$$\begin{aligned} h(\mathbf{X}) &= \mathbf{\Sigma} + \Delta^{h(\mathbf{X})} \\ \mathbf{D} &= \mathbf{\Sigma} + \Delta^{\mathbf{D}} \end{aligned} \tag{9}$$

such as that $\mathbf{\Sigma}$, $\Delta^{h(\mathbf{X})}$ and $\Delta^{\mathbf{D}}$ are all mutually independent. In this model, only the terms $\Delta^{h(\mathbf{X})}$ and $\Delta^{\mathbf{D}}$ carry a useful information for the update of the state estimate, while the term $\mathbf{\Sigma}$ must not affect it. It has been proven in [10] that such a generalised update can be still facilitated using Equation (5) only by altering the Kalman gain formula in Equation (6) to:

$$\begin{aligned} &\mathbf{K} = \\ &= \operatorname{cov}(\mathbf{X}, h(\mathbf{X}))\operatorname{cov}(h(\mathbf{X}))^{-1} \\ &\quad \times \left(\operatorname{cov}(h(\mathbf{X})) - \tfrac{1}{2}\left[\operatorname{cov}(h(\mathbf{X}), \mathbf{D}) + \operatorname{cov}(\mathbf{D}, h(\mathbf{X}))\right]\right) \\ &\quad \times \operatorname{cov}(\mathbf{D} - h(\mathbf{X}))^{-1} \end{aligned} \tag{10}$$

Note that Equation (10) is a generalisation of Equation (6); if $\operatorname{cov}(h(\mathbf{X}), \mathbf{D}) = 0$ then the former reduces to the latter.

The CPI-EnKF exploits another advantageous feature of the EnKF - the representation of state estimates and observations using ordered random sample sets implicitly retains the correlation information, and hence the cross-covariance $\operatorname{cov}(h(\mathbf{X}), \mathbf{D})$ can be easily estimated from the data. Note that the requirement of a fixed order of ensemble members is extremely important. For example, consider a situation in which the order of samples in ensemble $\mathbf{D}$ was randomly reshuffled. As such, ensemble $\mathbf{D}$ would effectively loose information about its correlation to ensemble $h(\mathbf{X})$, and the CPI-EnKF update rule would produce an overconfident estimate. In other words, the correlation information is encoded in the order of ensemble members.

Also note that the traditional Kalman filter can be generalised in a similar way as the EnKF, using a Kalman gain formula similar to Equation (10). Unfortunately, with the traditional Kalman filter and its representation of estimates using a mean and covariance, it is practically impossible to compute the necessary cross-covariance between the state and observation, except of few special cases.

III. Cooperative localisation

The past two decades witnessed a revolution in the deployment of advanced driver assistance systems. A significant number of vehicles currently available on our roads are equipped with global positioning system (GPS) receivers and on-board navigation computers, which inform the driver about their current position on a map and advise them on navigation decisions. The built-in navigation computers often integrate vehicle odometry information obtained through wheel rotation counters or similar sensors, which helps the navigation computers to maintain the global position estimates even in areas with no GPS coverage, such as in urban canyons or tunnels. Although additional sensors, such as an inertial measurement unit (IMU), magnetic compass or visual odometry, can help the computer to maintain a more accurate global position estimate, without any global reference, the accuracy of the estimate always deteriorates as relative errors accumulate over time.

Various approaches to improve the accuracy of the global position estimates in the case of poor or no GPS signal reception have been proposed. For example, Google's self-driving cars employ highly detailed 3D maps of the environment and advanced on-board sensors in order to fix the global position of a vehicle in the map with a high accuracy [14]. Another class of methods propose extending the road infrastructure with active or passive beacons that broadcast their global position, so that nearby vehicles equipped with an appropriate sensor can improve their global position estimate [15]. Unfortunately, all these approaches are costly, which impedes their wide-scale deployment to our public road network.

Cooperative localisation is based on a simple idea that vehicles with more accurate position estimates can help nearby vehicles improve their potentially poor position estimates. In technical terms, a vehicle can detect a relative location of a nearby vehicle using a proximity sensor such as a radar, a laser range finder (LIDAR) or a video camera. This information can then be communicated between the vehicles over a wireless network link and fused with their current positions estimates, and consequently, improve their accuracy. Such an improvement stems from the fact that the other vehicle might be driving through an area with a better GPS coverage and thus have a better position estimate, and also because the fusion of observations from multiple independent sources generally leads to better estimates. It is important to note that the cooperative localisation does not require any investments in the road infrastructure, while radars or video cameras are increasingly available in new cars, even in the mid-range segment, and a wireless communication capability is generally cheap. Furthermore, cooperative localisation can be seamlessly combined with other positioning systems and sensors.

The state-of-the-art approaches to the problem of cooperative localisation can be divided into two principal classes, based on the type of information the vehicles communicate: either they only communicate local information inferred exclusively through vehicle's own sensors, or they communicate information potentially inferred from other vehicle's sensor.

In the first class of cooperative localisation approaches, each vehicle only communicates local (pre-processed) sensor measurements obtained exclusively using vehicle's own sensors. Such measurements are typically statistically independent of other vehicles' measurements, and therefore the data incest problem is avoided. The measurements can be communicated to a single central authority that estimates the position of all the vehicles in the environment as one large system, using fusion algorithms such as the Extended Kalman filter [16], [17], [18], particle filter [19] or Maximum Likelihood estimation [20]. Although the centralised methods typically provide optimal global position estimates, they are susceptible to a single point of failure and are not scalable to a larger number of vehicles. In order to eliminate the single point of failure problem, the central filter can be decomposed into a set of communicating filters distributed among all the vehicles (i.e. decentralised), either in an optimal fashion, at the expense of a higher computation and communication cost [21], [22], or in an approximate fashion, with more efficient communication and computation [23]. Another approach, described in [24], assumes the vehicles only communicate sporadically and it proposes an algorithm that allows the vehicles to only store and communicate the smallest necessary set of sensor measurements that still guarantees optimal position estimation. Unfortunately, this is only possible for an a priori known and fixed number of vehicles. Alternatively, each vehicle can maintain an estimate of the state of all neighbouring vehicles based solely on the vehicle's own sensors, and broadcast this so-called *group state* to nearby vehicles, thus helping them to improve their own global position estimates [25]; although the communication and computation is quite efficient in such a system, the information does not flow transitively between non-neighbouring vehicles and therefore, the position estimates might be of mediocre quality.

In the second class of approaches, the vehicles communicate data that is potentially inferred from other vehicles' sensor measurements - typically, the actual global position estimates. In principle, when a vehicle receives a global position estimate which was broadcast by a nearby vehicle, it combines it with information about the relative location of that vehicle obtained from an on-board sensor, and uses this combined information as an observation to improve it's own global position estimate, using a fusion algorithm such as the Kalman filter. Unfortunately, such an approach is susceptible to the data incest problem, because the global position estimate of the other vehicle might depend on the global position estimate of the local vehicle that had been broadcast earlier. In effect, this dependence causes a correlation between the local estimate error and the observation error and it violates the assumption of independence, inherent to the Kalman filter and many other data fusion algorithms. The state of the art offers various approaches to the data incest problem, such as to ignore it [26], to maintain an (inherently incomplete) dependency tree to limit the extent of so-called *circular updates* [27], or to use a sub-optimal but consistent algorithm such as the covariance intersection (CI) to fuse the data [28].

Intuitively, in order to make a cooperative localisation system applicable to the existing public road network, the system must fulfill the following requirements: There can be an arbitrary large and a priori unknown number of vehicles, which can join and leave the road network at any time. The vehicles can have different shapes, move erratically in the environment rather than in a particular formation, can employ different types of sensors, and they can be equipped with maps of varying precision, or no maps at all. Furthermore, a wireless communication is generally not reliable and therefore, the cooperative localisation system must not depend on any well-defined communication pattern; instead, it should operate opportunistically on a best-effort basis whenever a communication channel between vehicles can be established. In the next section, we propose a cooperative localisation system based on the CPI-EnKF filter, which meets all the above requirements.

IV. Cooperative localisation using Common Past-Invariant Ensemble Kalman filter

Let's assume each vehicle maintains an individual Ensemble Kalman filter (EnKF) in order to estimate its own global position using data from on-board sensors, such as the GPS, odometry, or any other suitable sensor. The design and implementation of such an estimation system is a standard task, as mathematical models for many vehicle and sensor types are well understood and generally available. Additionally, the vehicles are equipped with one or more proximity sensors, such as a radar or video camera, that enable them to detect relative locations of nearby vehicles. When a vehicle detects another vehicle in its vicinity, it can establish a vehicle-to-vehicle (V2V) communication channel, enabling the two to exchange their current global position estimates and measured relative displacement, and use that information to update their local position estimates. In the following paragraph, we describe how to perform such an update using the CPI-EnKF.

Formally, at a discrete time step $t-1$, a vehicle represents the current state estimate using an ensemble $\mathbf{X}_{t-1} \in \mathbb{R}^{n \times N}$. The state is composed of variables that describe its global position in the world (e.g. latitude, longitude, direction), as well as other variables potentially needed to model the vehicle dynamics (e.g. steering angle, speed, acceleration). The cooperative localisation algorithm repeats the following steps:

1) Given the last vehicle's state estimate represented as an ensemble $\mathbf{X}_{t-1}$, predict the state at the next discrete time step t using Equation (2), with a prediction model based either on ego-motion sensors (e.g. odometry) or a vehicle dynamics model. The resulting state estimate is represented as a *prior ensemble* $\mathbf{X}_t^- \in \mathbb{R}^{n \times N}$.
2) Update the prior ensemble $\mathbf{X}_t^-$ using Equations (5) and (6) in order to assimilate the current positional sensors measurements (e.g. GPS, IMU, compass), and thus obtain a more accurate state estimate represented as a *communication ensemble* $\mathbf{X}_t^* \in \mathbb{R}^{n \times N}$. Such sensor measurements can be assumed to be affected by a white Gaussian noise, and therefore, the traditional Kalman gain formula can be applied.
3) Measure the relative displacement $\mathbf{r}_t \in \mathbb{R}^m$ of a neighbour vehicle using a proximity sensor (e.g. radar, LIDAR, video camera). Let $\mathbf{R}_t \in \mathbb{R}^{m \times m}$ denote a covariance matrix characterising the error of the measurement.
4) Send a message to the neighbour vehicle containing the relative displacement $\mathbf{r}_t$, covariance $\mathbf{R}_t$, and the communication ensemble $\mathbf{X}_t^*$ (or only a subset of thereof limited to the global position-related state variables).
5) Receive a message submitted by the neighbour vehicle containing its own communication ensemble, herein denoted as $\mathbf{Y}_t \in \mathbb{R}^{n \times N}$.
6) Update the prior ensemble $\mathbf{X}_t^-$ using Equations (5) and (10) to accommodate the "observation" indicating the vehicle's location, which is an ensemble constructed from the received ensemble $\mathbf{Y}_t$, the relative displacement measurement $\mathbf{r}_t$ and randomly generated samples with covariance $\mathbf{R}_t$. The generalised Kalman gain formula needs to be applied in this step because $\mathbf{X}_t^-$ and $\mathbf{Y}_t$ are potentially correlated as they might share a common past information. The result of such an update leads to a more accurate state estimate represented by a *provisional posterior ensemble* $\mathbf{X}_t^+ \in \mathbb{R}^{n \times N}$.
7) Update the provisional posterior ensemble $\mathbf{X}_t^+$ using Equations (5) and (6) to accommodate the current positional sensors measurements (similarly as in Step 2), leading to the *final posterior ensemble* $\mathbf{X}_t$ that represents the current best state estimate.

In principle, this opportunistic distributed cooperative localisation algorithm is very similar to the decentralised covariance intersection data fusion algorithm described in [9], the main difference being that our algorithm operates with ensembles instead of means and covariances. The efficiency of the algorithm stems from that fact that it exploits the locality in the communication at the expense of sub-optimality of the position estimates compared to a theoretical central estimator - the sensor information is only propagated locally between neighbouring vehicles and therefore, some vehicles will not receive it and will not update their position estimates.

Note that the algorithm assumes that the ensembles maintained by all the vehicles have the same number of samples. Although the algorithm is described from the perspective of an observing vehicle, the steps taken by the vehicle being observed are very similar as it practically makes no difference which of the two vehicles performs the actual relative displacement measurement and initiates the communication. The steps of the algorithm are idealised in the sense they assume a single neighbour vehicle is detected at every time step; in practice, if two or more vehicle were detected, Steps 3-6) need to be repeated separately for each of the vehicles detected, or not performed at all if no vehicle was detected. Furthermore, we assume that all the vehicles have some unspecified means to associate an observed neighbour vehicle to the vehicle communicated to, and also that all the vehicles operate at synchronous time steps. In practice, additional provisions need to be made to ensure a correct data association and timing.

V. Evaluation

In this section, we describe an experimental evaluation of the accuracy of the CPI-EnKF cooperative localisation algorithm proposed in Section IV, performed using computer simulation. The performance of the algorithm is compared to the following algorithms: the covariance intersection (CI), the split covariance intersection (Split CI), and the local Kalman filter (Local KF). These reference algorithms, all of whom will also be described in detail in this section, have been chosen because they all meet the criteria outlined in Section III: they are fully decentralised, cost a constant space and time per update, can be scaled up to an arbitrary number of vehicles, support diverse vehicle types and sensors, only assume opportunistic communication and, apart from the Local KF algorithm, they all provide consistent estimates, given certain assumptions. The Local KF algorithm is only included to illustrate the problem of overconfidence and divergence should the data incest be neglected. Additionally, we also include a centralised Kalman filter (Central KF) algorithm to show the optimal position estimates for a reference.

A. Scenario

The world in the evaluation scenario is best imagined as a two-dimensional city consisting of 16 rectangular city blocks, organised in a 4-by-4 grid, with each block of dimensions 150 by 100 m, as depicted in Figure 1. There are in total 16 vehicles driving around the blocks on "roads" that are, for simplicity, assumed to be straight lines of zero width. The vehicles, which themselves are modelled as simple points, only drive on these straight lines and collisions between the vehicles are not considered. The speed of the vehicles varies from 1 to 20 m/s, and it changes with an acceleration that evolves randomly over time as a Wiener process bounded between -4 and 4 m/s^2, rescaled to have an absolute standard deviation of 0.632 m/s^2 per second. Whenever a vehicle reaches the end of a city block, it randomly decides whether to take a turn (not a U-turn though) or to continue. This action affects neither the vehicle's speed nor its acceleration. The evolution of the world is simulated in discrete time-steps of 0.1 s. The vehicles start off with an initial global position estimate that is consistent with their true position and has an error with a standard deviation of 10 m.

Each vehicle is equipped with a (simulated) odometry sensor, that reports a relative two-dimensional displacement of the vehicle since the previous reading, affected independently in both axes by a zero-mean white Gaussian noise with a standard deviation of 0.05 m for every meter of the true distance travelled. The odometry has no angular error, in order to avoid a bias in the evaluation results caused by non-linear effects. Each vehicle is also equipped with a GPS receiver, that enables them to detect their global position, affected by a zero-mean white Gaussian noise with a standard deviation depending on the vehicle's location in the city. The odometry sensor provides readings with a period of 0.1 s (i.e. every simulation time-step), while the GPS receiver reports the global position with a period of 1 s (i.e. every 10th simulation time-step).

Additionally, each vehicle is equipped with a proximity sensor that can detect the relative distance in two dimensions to nearby vehicles driving on the same straight road segment, with a probability of making an actual observation (at any given simulation time-step) that decreases linearly from 15% to 0% as the distance to the other vehicle increases from 5 m to 100 m. The proximity sensor readings are affected independently in both axes by a zero-mean white Gaussian noise with a standard deviation of 0.05 m for every meter of the true distance between the vehicles. One vehicle can observe multiple vehicles during a single time-step, and it can always establish a communication channel with all the vehicles observed. The communication is instant. Note that it would be counterproductive to perform the cooperative localisation data exchanges too often, because the chance that subsequent information exchanges in a close group of vehicles bring any new information is quite low, while these exchanges still contribute to the accumulation of numerical errors, and in real-world applications they congest the wireless communication medium.

As argued in Section III, cooperative localisation has the most significant impact on vehicles that have very inaccurate global position estimates, such as vehicles operating in areas with a poor GPS coverage. In order to introduce this effect in our evaluation scenario, a GPS signal is only available on a single road segment of the city (see the darker bottom left region in Figure 1). In this area, the GPS signal has such a quality that enables the receiver to compute the global position with an error equivalent to a zero-mean white Gaussian noise with a standard deviation of 3 m. All the other areas in the city have no GPS coverage at all, and therefore, for vehicles driving in these areas the cooperative localisation represents the only available means to maintain position estimates with a reasonably accuracy.

Although our evaluation scenario involves a simplistic model of the world, it suffices as a tool to compare the various cooperative localisation algorithms. Importantly, the simulated world meets all the theoretical assumptions of the algorithms being evaluated, such as that the system dynamics are linear and all the errors involved are Gaussian, and therefore, all the algorithms have perfect conditions for their operation. Also, note that all the constants in the presented scenario are chosen in order to model a real-world city traffic as closely as possible, but similar results can also be achieved with a different constant selection.

B. Implementation

For the purpose of this evaluation, we implemented several variants of the opportunistic distributed cooperative localisation algorithm described in Section IV. These variants, which will be discussed in this section, differ mainly in the way they represent the state estimates and how they accommodate the potentially correlated observations. All of them model the vehicle's system state as a two-dimensional vector that represents the absolute coordinates of the vehicle in the city grid, measured in meters, with the coordinate origin in the bottom-left corner of the map.

All the evaluated variants of the cooperative localisation algorithm facilitate the prediction in Step 1) simply by altering the state estimate using a two-dimensional odometry reading and an associated covariance matrix, and the update in Steps 2) and 7) using a two-dimensional GPS reading and an associated covariance matrix, whenever available. The relative displacement of a neighbour vehicle measured in Step 3) is represented as a two-dimensional vector, i.e. $\mathbf{r}_t \in \mathbb{R}^2$ and $\mathbf{R}_t \in \mathbb{R}^{2\times 2}$. Steps 4) and 5) do not require any implementation, as all the necessary information is implicitly available to the simulation process. Therefore, in the following text, we will only focus on the description of the implementation of Step 6) by the particular algorithms, which is arguably the only non-trivial part.

In the implementation of the base **CPI-EnKF** variant of the cooperative localisation algorithm, the current state estimate is represented using an ensemble with the number of samples $N = 1000$. As described in Step 6), the provisional posterior ensemble $\mathbf{X}_t^+ \in \mathbb{R}^{2\times 1000}$ is computed using Equations (5) and (10). The inputs to these equations are defined as follows:

$$\begin{aligned} \mathbf{X} &= \mathbf{X}_t^- \\ \mathbf{D} &= \mathbf{Y}_t - \mathcal{N}(\mathbf{r}_t, \mathbf{R}_t) \end{aligned} \tag{11}$$

where $\mathbf{X}, \mathbf{D} \in \mathbb{R}^{2\times 1000}$ and the observation model $h\colon \mathbb{R}^2 \to \mathbb{R}^2$ is simply an identity function, and therefore:

$$h(\mathbf{X}) = \mathbf{X}_t^- \tag{12}$$

Note that $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ above denotes a matrix of a suitable dimension, whose columns are randomly generated vectors from a (multivariate) Gaussian distribution with a mean vector $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Sigma}$.

In the covariance intersection (**CI**) variant of the cooperative localisation algorithm, the state estimate is represented using a two-dimensional mean vector and a covariance matrix. These values are communicated between vehicles instead of the ensembles. As such, the whole algorithm is de facto equivalent to the distributed data fusion algorithm described in [9].

Let's assume the prior state estimate is denoted $\hat{\mathbf{x}} \in \mathbb{R}^2$ and the associated error covariance $\mathcal{X} \in \mathbb{R}^{2\times 2}$, and the state estimate received from the other vehicle is denoted $\hat{\mathbf{y}} \in \mathbb{R}^2$ and the associated error covariance $\mathcal{Y} \in \mathbb{R}^{2\times 2}$. Then in Step 6), the provisional posterior estimate $\hat{\mathbf{x}}^+ \in \mathbb{R}^2$ and the associated error covariance $\mathcal{X}^+ \in \mathbb{R}^{2\times 2}$ is computed using the CI update rule as:

$$\begin{aligned} \mathcal{X}^+ &= \left[\omega\mathcal{X}^{-1} + (1-\omega)\mathcal{Y}^{-1}\right]^{-1} \\ \hat{\mathbf{x}}^+ &= \mathcal{X}^+ \left(\omega\mathcal{X}^{-1}\hat{\mathbf{x}} + (1-\omega)\mathcal{Y}^{-1}\hat{\mathbf{y}}\right) \end{aligned} \tag{13}$$

with the coefficient $\omega \in \mathbb{R}$ optimised on the fly so that the trace of the covariance matrix $\mathcal{X}^+$ is minimal. Note that the CI update rule is immune to the correlation between $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$, and therefore, it provides a consistent estimate in this application [8].

In the split covariance intersection (**Split CI**) variant of the cooperative localisation algorithm, the state estimate is also represented using a two-dimensional mean vector. However, as opposed to the CI, the error covariance matrix is split into two additive component covariance matrices: the first covariance matrix represents the potentially correlated error component, and the second covariance matrix represents the known-independent error component. Such a splitting of the covariance leads to a higher accuracy of the estimation, compared to the CI [9]. Due to the fact that the vehicle's on-board sensors produce independent observations, the prediction step using the odometry readings will only affect the known-independent error component of the state estimate, and the update using the GPS readings will be optimal because the correlated component of the GPS error is zero. Note that such an algorithm is similar to the cooperative localisation algorithm presented in [28].

Let's denote again the prior state estimate and the state estimate received from the other vehicle as $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$, respectively, and the corresponding error covariance matrix components as $\mathcal{X}_1+\mathcal{X}_2$ and $\mathcal{Y}_1+\mathcal{Y}_2$, respectively. Step 6) of the cooperative localisation algorithm computes the provisional posterior estimate $\hat{\mathbf{x}}^+$ and the associated covariance matrix components $\mathcal{X}_1^+ + \mathcal{X}_2^+$ as:

$$\begin{aligned} \mathcal{X}^{-1} &= \omega\left(\mathcal{X}_1 + \omega\mathcal{X}_2\right)^{-1} \\ \mathcal{Y}^{-1} &= (1-\omega)\left(\mathcal{Y}_1 + (1-\omega)\mathcal{Y}_2\right)^{-1} \\ \mathcal{X}_1^+ &= \left(\mathcal{X}^{-1} + \mathcal{Y}^{-1}\right)^{-1} \\ \mathcal{X}_2^+ &= 0 \\ \hat{\mathbf{x}}^+ &= \hat{\mathbf{x}} + \mathcal{X}_1^+\mathcal{Y}^{-1}\left(\hat{\mathbf{y}} - \hat{\mathbf{x}}\right) \end{aligned} \tag{14}$$

Again, $\omega \in \mathbb{R}$ is optimised on the fly so that the trace of the covariance matrix $\mathcal{X}_1^+$ is minimal. Note that Equation (14) differs slightly from the standard split covariance intersection algorithm described in [9], because the know-independent covariance component is directly added to the correlated component. This is necessary to ensure a consistency of the cooperative localisation algorithm, because the resulting position estimate might be correlated fully to subsequent position estimates received from other vehicles via the communication.

The localised Kalman filter (**Local KF**) variant of the cooperative localisation algorithm represents, similarly as the CI, the state estimate using a mean vector and a covariance matrix, and the update in Step 6) is performed using a simple Kalman filter update rule. Assuming the same notation as in the CI variant above, the provisional posterior state estimate is computed as follows:

$$\begin{aligned} \mathbf{K} &= \mathcal{X}[\mathcal{X} + \mathcal{Y}]^{-1} \\ \mathcal{X}^+ &= \mathcal{X} - \mathbf{K}\mathcal{X} \\ \hat{\mathbf{x}}^+ &= \hat{\mathbf{x}} + \mathbf{K}(\hat{\mathbf{y}} - \hat{\mathbf{x}}) \end{aligned} \tag{15}$$

As discussed in Section I, the Kalman filter update rule assumes the prior state estimate $\hat{\mathbf{x}}$ is independent of the observation $\hat{\mathbf{y}}$, which in this case does not hold. Therefore, the resulting provisional posterior state estimate $\hat{\mathbf{x}}^+$ with the error covariance $\mathcal{X}^+$ might be inconsistent with the true error, the filter might become overconfident over time and diverge. We include the Local KF algorithm merely to illustrate the consequences of this problem.

Finally, we also compute the position estimates of all the vehicles in the system using a centralised Kalman filter (**Central KF**) algorithm. The system state vector includes the positions of all the vehicles in the simulation. In such a model, all sensor observations are independent of the state. Because all the Kalman filter assumptions are satisfied, the resulting state estimates are guaranteed to be consistent and optimal, given all the available sensor observations [3]. Therefore, the Central KF represents an upper limit of the quality of the other cooperative localisation algorithms and as such it serves as a useful reference in the evaluation.

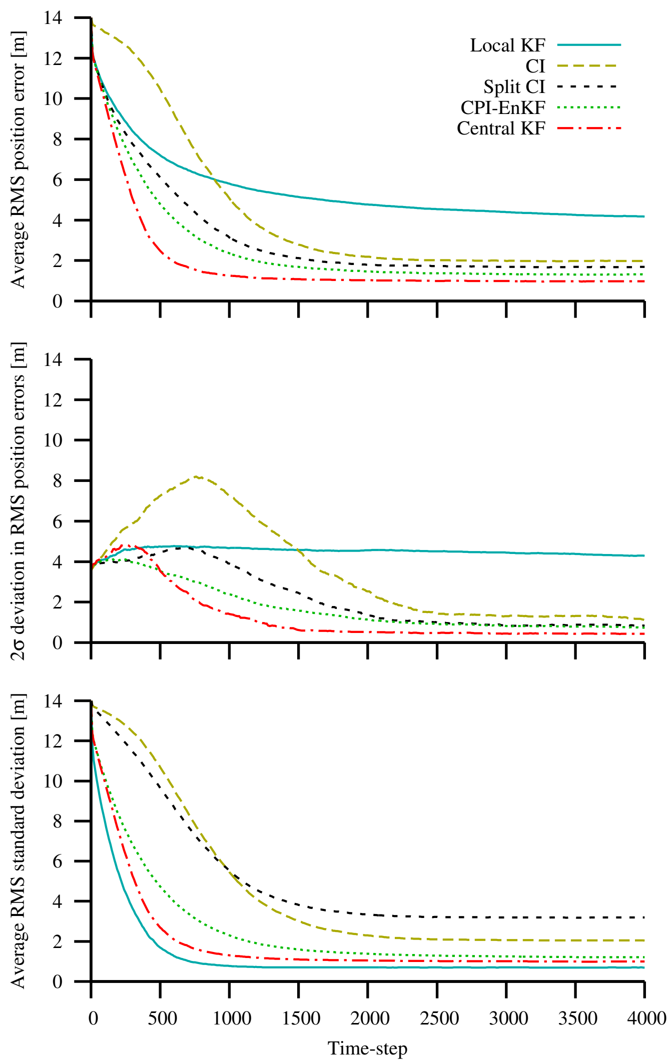

Fig. 2. The quality of position estimates provided by the various cooperative localisation algorithms, averaged over 1000 random simulations. The first graph shows an average position error of the estimates (lower value means smaller errors). The second graphs complements the first graph by showing the deviation in the position errors from the average among all the simulations (lower value means the algorithm is more predictable). The third graph shows an average variance of the errors reported by each of the estimation algorithms (for consistent algorithms, this value should be greater than or equal to the average true position error in the first graph; the closer the better)

C. Results

We executed the simulation 1000 times with a different random seed for every run. Figure 1 depicts a sample snapshot of one of the simulation runs, and Figure 2 illustrates the quality of position estimates provided by the particular cooperative localisation algorithms, averaged over all the simulation runs. The most important results from this evaluation are the following:

- In every aspect evaluated, the CPI-EnKF algorithm is superior to the CI, Split CI and Local KF algorithms.
- The evaluation confirms that the Local KF is inconsistent because it neglects the correlations.
- While the Split CI algorithm provides reasonable position estimates, the associated covariances are overly pessimistic - see also Figure 1.
- After a long enough time, all the algorithms reach an equilibrium where position errors stemming from the inaccurate odometry are, on average, compensated for by the gains in the accuracy due to the GPS and the cooperative localisation. The average magnitude of the position errors in this equilibrium depends on the quality of the cooperative localisation algorithm.

VI. Conclusion & Future Work

In this paper, we provided an overview of the Ensemble Kalman filter (EnKF) and its generalisation recently developed by this paper's authors - the Common Past-Invariant Ensemble Kalman filter (CPI-EnKF) - that can be applied even in the presence of a correlation between the state estimate and observation errors. We discussed the problem of a cooperative localisation in a group of communicating vehicles, described the state of the art approaches that address it, and then developed an entirely new approach based on the CPI-EnKF. The paper argues that the new method, unlike majority of the existing methods, provides consistent estimates given certain assumptions, it is simple to tune and implement, supports arbitrary vehicle types and sensors, and it can be scaled to an unlimited number of vehicles. The performance of the new CPI-EnKF cooperative localisation algorithm has been compared to existing similar algorithms in a comprehensive evaluation. Additionally, this paper indicates that the CPI-EnKF is a very generic algorithm that can be applied to various other problems in data fusion.

In order to apply the new cooperative localisation algorithm in practice, several issues need to be resolved, such as a correct association between the vehicles observed and the vehicles communicated with, an optimal planning of the information exchanges and an associated efficient allocation of the communication bandwidth, in particular in areas with a large number of vehicles, and safety concerns stemming from the ability of vehicles to forge the communicated data.

References

[1] R. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[2] R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering with Matlab Exercises, 4th Edition*. Wiley, Feb 2012.

[3] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley, 2006.

[4] H. Durrant-Whyte, M. Stevens, and E. Nettleton, "Data fusion in decentralised sensing networks," in *Proceedings of the 4th International Conference on Information Fusion (FUSION 2001)*, 2001, pp. 302–307.

[5] H. F. Durrant-Whyte, B. Rao, and H. Hu, "Toward a fully decentralized architecture for multi-sensor data fusion," in *Proceedings of the 1990 IEEE International Conference on Robotics and Automation (ICRA 1990)*. IEEE, 1990, pp. 1331–1336.

[6] P. Alriksson and A. Rantzer, "Distributed Kalman filtering using weighted averaging," in *Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems*, 2006.

[7] R. Olfati-Saber, “Kalman-consensus filter: Optimality, stability, and performance,” in *Proceedings of the 48th IEEE Conference on Decision and Control, held jointly with the 28th Chinese Control Conference (CDC/CCC 2009)*. IEEE, 2009, pp. 7036–7042.

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

[9] ——, *General decentralized data fusion with covariance intersection (CI)*. CRC Press, 2001, ch. 12.

[10] J. Čurn, D. Marinescu, G. Lacey, and V. Cahill, “Estimation with non-white Gaussian observation noise using a Generalised Ensemble Kalman filter,” in *Proceedings of the IEEE International Symposium on Robotic and Sensors Environments (ROSE 2012)*. IEEE, Nov 2012, pp. 85–90.

[11] G. Evensen, “Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics,” *Journal of Geophysical Research*, vol. 99, no. C5, pp. 10 143–10 162, 1994.

[12] ——, *Data Assimilation: The Ensemble Kalman Filter, 2nd Edition*. Springer, 2009.

[13] J. Mandel, “Efficient implementation of the Ensemble Kalman filter,” University of Colorado at Denver and Health Sciences Center, Tech. Rep., 2006.

[14] S. Thrun and C. Urmson, “Self-driving cars,” Keynote Address at IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2011), September 2011.

[15] E. Lee, S. Yang, S. Oh, and M. Gerla, “RF-GPS: RFID assisted localization in VANETs,” in *IEEE 6th International Conference on Mobile Adhoc and Sensor Systems (MASS 2009)*, 2009, pp. 621–626.

[16] A. Mourikis and S. Roumeliotis, “Performance analysis of multirobot cooperative localization,” *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 666–681, 2006.

[17] G. Huang, N. Trawny, A. Mourikis, and S. Roumeliotis, “On the consistency of multi-robot cooperative localization,” in *Proceedings of Robotics: Science and Systems (RSS 2009)*, 2009, p. 25.

[18] S. Roumeliotis and I. Rekleitis, “Propagation of uncertainty in cooperative multirobot localization: Analysis and experimental results,” *Autonomous Robots*, vol. 17, no. 1, pp. 41–54, 2004.

[19] D. Fox, W. Burgard, H. Kruppa, and S. Thrun, “A probabilistic approach to collaborative multi-robot localization,” *Autonomous Robots*, vol. 8, no. 3, pp. 325–344, 2000.

[20] A. Howard, M. Matark, and G. Sukhatme, “Localization for mobile robot teams using maximum likelihood estimation,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2002)*, 2002, pp. 434–439.

[21] S. Roumeliotis and G. Bekey, “Distributed multirobot localization,” *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 781–795, 2002.

[22] E. Nerurkar, S. Roumeliotis, and A. Martinelli, “Distributed maximum a posteriori estimation for multi-robot cooperative localization,” in *IEEE International Conference on Robotics and Automation (ICRA 2009)*, 2009, pp. 1402–1409.

[23] P. Barooah, W. Russell, and J. Hespanha, “Approximate distributed Kalman filtering for cooperative multi-agent localization,” *Distributed Computing in Sensor Systems*, pp. 102–115, 2010.

[24] K. Leung, T. Barfoot, and H. Liu, “Decentralized localization for dynamic and sparse robot networks,” in *IEEE International Conference on Robotics and Automation (ICRA 2009)*, 2009, pp. 3135–3141.

[25] 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 (IROS 2006)*, 2006, pp. 519–524.

[26] A. Martinelli, “Improving the precision on multi robot localization by using a series of filters hierarchically distributed,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007)*, 2007, pp. 1053–1058.

[27] A. Howard, M. Mataric, and G. Sukhatme, “Putting the ’I’ in ’team’: an ego-centric approach to cooperative localization,” in *IEEE International Conference on Robotics and Automation (ICRA 2003)*, 2003, pp. 868–874.

[28] H. Li and F. Nashashibi, “Cooperative multi-vehicle localization using split covariance intersection filter,” in *IEEE Intelligent Vehicles Symposium (IV 2012)*, 2012, pp. 211–216.