Graph Based Vehicle Infrastructure Cooperative Localization

Dhiraj Gulati*§, Feihu Zhang†, Daniel Malovetz*, Daniel Clarke‡, Gereon Hinz* and Alois Knoll§
*fortiss GmbH, München, Germany. {gulati, malovetz, hinz}@fortiss.org
†School of Marine Science and Technology, Northwestern Polytechnical University, 710072, Xi'an, China. feihu.zhang@nwpu.edu.cn
‡ Cogsense Technologies Limited, United Kingdom. daniel.clarke@cogsense.co.uk
§Technische Universität München, Garching bei München, Germany. dhiraj.gulati@tum.de, knoll@in.tum.de

***Abstract*—This paper presents a novel and an improved approach for estimating the position of a vehicle using vehicle-infrastructure cooperative localization. In our previous work we presented a Factor Graph based solution which added the topology (inter-vehicle distance) as a constraint while localizing the vehicle using data from sensors from both inside and outside the vehicle. This paper extends the work by reducing the error in calculating the precision of the position by almost 27% in the best case and lowering the computational time by at least 50% over our previously proposed solution. This is achieved by modifying current topology constraints to be also dependent on the previous state estimate. The proposed solution remains scalable for many vehicles without increasing the execution complexity. Finally, simulations indicate that incorporating the new topology information via Factor Graphs can improve performance over the traditional, state of the art, Kalman Filter approach.**

I. Introduction

Highly assisted and autonomous driving applications rely on the ability of the vehicle to localize itself within a common reference framework. Multi-sensor autonomous systems achieve this by fusing noisy data from various sensors like Odometry, GPS, Cameras and LIDAR. Modeling the stochastic effects of noise is relatively straightforward, however modeling systematic errors is inherently very difficult and can result in imprecise localization.

Solutions using high resolution sensors are generally unacceptable, as they will most likely increase the cost of manufacturing. Another possible solution is Cooperative Localization (CL), which improves the self-localization in cooperation with other sensors outside the vehicle. With advancement in Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication technologies, such CL has become a viable and beneficial solution [1]. This is because exchanging the sensor observations and the state estimates has become possible in real time.

There exist many innovative methods which solve the problem of CL. Although all the solutions provide novel ways of solving the problems, they still lack one or more points. We highlight those points from our previous work [2] :

- Bandwidth Limitations: The increasing number of vehicles for CL contend for the finite available bandwidth. Howard, A. et al. [3] showed the minimum bandwidth usage by using Maximum Likelihood Estimation but it depends on perfect identification of the other participants.
- Data Association Uncertainties: Real environment has multiple targets, hence correct data association task becomes difficult. Montemerlo, M. et al. [4] demonstrated the use of FastSLAM to perform localization in case of unknown data association.
- Coordinate Transformations: Every sensor has its own frame of reference. Therefore to perform the fusion from various sensors, the data has to be converted to one common frame of reference. This is difficult to achieve in a highly dynamic environment, as the location and orientation of external sensors is unknown. Zhang, F. et al. [5] demonstrated the use of symmetric equations to address the issue.
- Scalability: For many solutions, the number of participants is directly proportional to the difficulty in manageability. Hence mostly they do not scale well with an increase in the participants. Distributed Conjugate Gradient (DCG) algorithm in Maximum a Posteriori (MAP) is quite scalable as the algorithm is at most quadratic to the number of robots [6]. But it requires synchronous communication to be maintained with all the participants, which is challenging in a dynamic environment.

Various solutions attempt to address the above challenges but not in a uniform framework. In [2], we presented the use of topology (inter vehicle distance) as constraint between factors of factor graph to provide one uniform framework covering all the above issues simultaneously. Measurements from internal sensors, like GPS, were formulated as factor and the topology constraints (created from an infrastructure sensor, like RADAR) were added between the states for various vehicles. Smoothing algorithm, Levenberg Marquardt Optimizer (LM), was used to optimize the joint probability density function and extract the final fused states.

In this paper we extend the work by a new and improved method to incorporate the topology information in the factor graph. Instead of computing the topology factor from the infrastructure sensor, we compute it from the Odometry sensor readings. And then compare it with the infrastructure sensor measurement.

With the new approach, we address all the above mentioned challenges. But in addition to the above advantages we also have: (a) It results in more precise localization and (b) has better computational performance.

II. Problem Description

A simple infrastructure vehicle CL scenario can be seen in the Fig. 1(a). The assumptions are as follows:

- All vehicles have internal GPS sensors which measures their position in an absolute reference of a 2D global coordinate system.
- The infrastructure sensor (RADAR) measures the relative positions of the vehicles in its own local 2D coordinate system. Configuration information is not available, such that its location and orientation is unknown.
- The participating vehicles and the infrastructure sensor can communicate in either direction to exchange data. Also there is no timing delay or data error in communication.
- No mechanism is available, including communication mechanism and/or the protocol to identify individual vehicles. This introduces a challenge from the perspective of data association,
- The environment has zero clutter and as such there is no false detections or missed detections of vehicles.

Then the task of CL is to reduce the uncertainty of the state estimation by fusing the data from all the available sensors.

III. Factor Graphs

A. Overview

Definition: A bipartite graph $G_k = (F_k, V_k, E_k)$ is defined as a Factor Graph when: (1) It has two types of nodes: factor nodes $f_i \in F_k$ and variable nodes $v_j \in V_k$; (2) Edges $e_{ij} \in E_k$ can exist only between factor nodes and variable nodes, and are present if and only if the factor f_i involves a variable v_j [7].

In simpler words, a factor graph explains the connection between the complex functions with many variables and its factors of simpler functions. Fig. 1(b) shows a simple factor graph with variables a, b, c and d and functions f_1 and f_2 with factorization: $h(a, b, c, d) = f_1(a, b, c) * f_2(b, c, d)$.

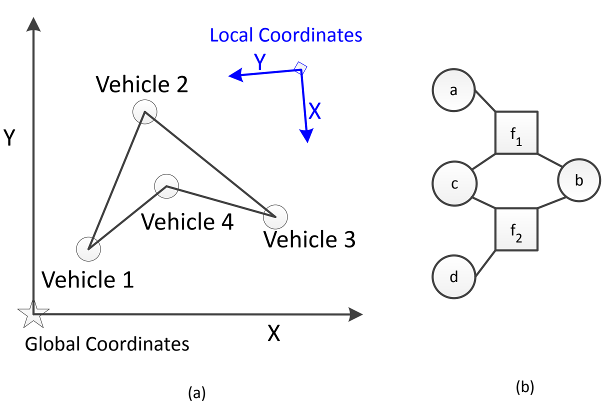

Fig. 1. (a) Topology for multiple vehicle cooperative localization system. RADAR Coordinate system is represented in blue color. (b) Factor graph with variables a, b, c and d and functions $f_1(a, b, c)$ and $f_2(b, c, d)$.

Factor graphs were initially introduced for the calculation of the sum-product algorithm [8]. Recently Makarau, A. et al. [9] applied the concept of factor graph for alphabet-based multi-sensor data fusion and classification. Indelman, V. et al. [7] demonstrated the use of factor graph for multi-sensor information fusion for navigation.

For a successful precise localization of a single target using CL, the infrastructure sensor should be able to observe it. But for tracking multiple targets, additional task of data association is required. Various solutions exist for the same like Joint Probabilistic Data Association (JPDA) [10] by Fortmann et al., Probability Hypothesis Density (PHD) filter [11] by Mahler and Multi Hypothesis Tracking [12] by Reid. And there exist various improvements of these solutions like [13], [14] and [15]. But using these additional data association algorithms not only increases the complexity of the task but at times also can increase the execution time. Also, at times these algorithms are unable to scale with an increasing number of vehicles.

Many solutions for multi-target tracking don't take into account the topology of the vehicle group. We demonstrated the use of factor graphs to avoid data association using the topology as a constraint factor.

While working further on constraints based on the topology, specific contributions of this paper is to:

1) further improve the precision in localization; and,
2) improve the computational performance of the process.

The task of smoothing is still performed through optimizing the resultant graph (using LM) to provide an estimate of the vehicle state.

B. Factor Graph formulations

A factor graph G_k can also be expressed as:

$$g(X) = \prod_i f_i(X_i), \tag{1}$$

where X_i is the set of all variables x_j connected by an edge to factor f_i.

An error function of each factor f_i represents the error between the predicted measurement and the actual measurement. To obtain the predicted state, the aim for non-linear least square optimizers is to minimize this function. This is done by adjusting the estimates of the variables X. The optimal estimate $\hat{X}$ is then obtained by optimizing the complete graph G as:

$$\hat{X} = \arg\min_X \left(\prod f_i(X_i) \right) \tag{2}$$

The above methodology can be compared with Kalman Filter. $h(\cdot)$ is the measurement model that predicts a sensor measurement from a given state estimate. The factor of the Factor Graphs is then synonym of this measurement model. For a Gaussian noise model, a measurement factor can be written as:

$$f_i(X_i) = d[h_i(X_i) - z_i], \tag{3}$$

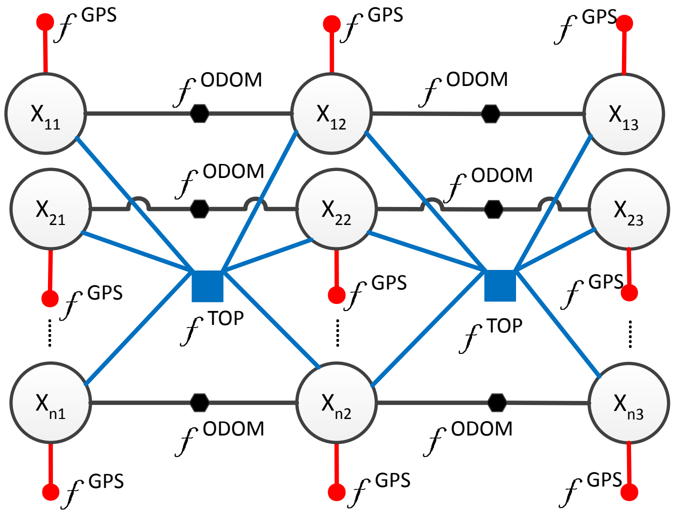

Fig. 2. Factor graph for n vehicles with three state nodes, two odometry factors, three GPS factors and two topology factors each.

where $h_i(X_i)$ is the measurement model as a function of the state variables X_i; z_i is the actual measurement and the operator $d(\cdot)$ represents a cost function.

The process model can be similarly represented as a factor graph (more detail is provided in [16]).

C. Factor formulations

We briefly explain the used factors. Please refer [2] for further details.

1) Odometry Measurements: For a constant velocity model, the equation is:

$$z^o_{t+1} = h^o(z^o_t) + n^o \tag{4}$$

where z^o_t is the Odometry measurement at time t, h^o is the function to calculate the odometry measurement at time $t+1$ and n^o is the measurement noise. Thus the binary factor for states X_{t+1}, X_t becomes:

$$f^{ODOM}(X_{t+1}, X_t) \triangleq d(z^o_{t+1} - h^o(z^o_t)) \tag{5}$$

The covariances provided by the sensor manufacturer are used while formulating the corresponding factors.

2) GPS Measurements: The GPS measurement equation is:

$$z^g_t = h^g(z_t) + n^g, \tag{6}$$

where n^g is the measurement noise and h^g is the measurement function, providing the relation between the measurement z^g_t and the position of the vehicle. Equation (6) gives an unary factor f^{GPS} which is written as:

$$f^{GPS}(X_t) \triangleq d(z^g_t - h^g(z_t)) \tag{7}$$

The covariances provided by the sensor manufacturer are used while formulating the corresponding factors.

3) Topology Measurements: In our previous work [2] we had used measurements from RADAR. The configuration (i.e. orientation and location) of this sensor was unknown. The topology information (the distance between the vehicle) at time t was calculated as follows:

$$(z^T_t)^2 = \sum_{i=1}^{N-1}\sum_{j=i+1}^{N}(p^i_{x,t} - p^j_{x,t})^2 + \sum_{i=1}^{N-1}\sum_{j=i+1}^{N}(p^i_{y,t} - p^j_{y,t})^2 \tag{8}$$

where $p^i_{x,t}, p^i_{y,t}$ represented the x and y position of i^{th} vehicle as observed by RADAR. p^j terms are analogous p^i terms.

In this paper we extend the above by proposing a new way of constructing the topology factor which performs better than the previously proposed solution. This is achieved by formulating the topology factor using the current and the past state estimate. At any given time t the position of the vehicle can be calculated as:

$$x_t = x_{t-1} + \delta \tag{9}$$

where x_{t-1} is the position calculated at $t-1$ and δ is the the distance traveled under constant velocity.

Using (9), current $p^i_{x,t}$ from (8) can be expressed in terms of the previous state and the odometry measurement as:

$$p^i_{x,t} = p^i_{x,t-1} + \delta x^i_{[t-1][t]} \tag{10}$$

where $p^i_{x,t}$ is x position of i^{th} vehicle to be estimated, $p^i_{x,t-1}$ is x position of i^{th} vehicle at time $t-1$ as calculated by the system and $\delta x^i_{[t-1][t]}$ is x odometry reading for i^{th} vehicle between the time $t-1$ and t. $p^i_{y,t}$ terms can be expanded similarly.

Assuming the configuration (i.e. orientation and location) of the RADAR sensor still remains unknown, using (10), topology (8) can be transformed as follows:

$$\begin{aligned}(z^T_t)^2 &= \sum_{i=1}^{N-1}\sum_{j=i+1}^{N}((p^i_{x,t-1} + \delta x^i_{[t-1][t]}) - (p^j_{x,t-1} + \delta x^j_{[t-1][t]}))^2 \\ &+ \sum_{i=1}^{N-1}\sum_{j=i+1}^{N}((p^i_{y,t-1} + \delta y^i_{[t-1][t]}) - (p^j_{y,t-1} + \delta y^j_{[t-1][t]}))^2\end{aligned} \tag{11}$$

where the terms have the same meaning as explained in (8) and (10). And p^j terms are analogous p^i terms.

Like our previous topology factor, it remains independent of the coordinate system. Eq. (11) can be summarized as:

$$(z^T_t)^2 = h^{top}(p^0_t, \cdots, p^N_t, p^0_{t-1}, \cdots, p^N_{t-1}, \delta^0_{[t-1][t]}, \cdots, \delta^N_{[t-1][t]}) + n^{top} \tag{12}$$

where n^{top} is the measurement noise and h^{top} is the new measurement function, that forms a relationship between the current states X^k_t, the previous state X^k_{t-1} and the odometry $\delta^k_{[t-1][t]}$ for participants $k = 1 \cdots N$. A point to be noted here is that δ is a scalar value. So effectively the new function is composed of $2 * N$ states i.e. is proportional to double the number of participants in the system. Hence this topology is a $(2 * N) - ary$ factor, while topology factor in [2] was an

$N-ary$ factor, where N is the number of vehicles in the system. The resultant factor f^{TOP} can be written as:

$$f^{TOP}(X_{1t},\cdots,X_{Nt}) \triangleq d((z_t^T)^2 - h^{top}(p_t^0,\cdots,p_t^N,p_{t-1}^0,\cdots,p_{t-1}^N,\delta_{[t-1][t]}^0,\cdots,\delta_{[t-1][t]}^N)) \tag{13}$$

This factor is connected to all the states X_i at time t and $t-1$. Fig. 2(c) illustrates a factor graph for n vehicles with three state nodes, two odometry factors and three GPS factors each. The corresponding state nodes for the vehicles are also connected to each other with derived topology factors.

Now the topology measurement is a derived measurement, hence we also need to calculate the covariance for it. If σ_x^2 and σ_y^2 are the x and y variances respectively for the infrastructure sensor (because we use the measurement from the infrastructure sensor), then the corresponding matrix for the measurements from the sensor is a diagonal matrix which can be written as (see [17] for more details):

$$Cov(x,y) = \text{diag}\left[\sigma_{x_1}^2,\cdots,\sigma_{x_n}^2,\sigma_{y_1}^2,\cdots,\sigma_{y_n}^2\right] \tag{14}$$

Then using (11) and (14), we obtain the covariance for the topology estimate at any time t as:

$$\sigma_{top_{x,y}}^2 = M * Cov(x,y) * M' \tag{15}$$

where M is a $1X2N$ matrix as follows:

$$M = \left[\frac{\partial}{\partial x_1}(z_t^T),\cdots,\frac{\partial}{\partial x_n}(z_t^T),\frac{\partial}{\partial y_1}(z_t^T),\cdots,\frac{\partial}{\partial y_n}(z_t^T)\right] \tag{16}$$

D. Smoothing

We use the LM linearization algorithm to solve the factor graph. Using an initial estimate x_0 it iteratively finds an update Δ from the linearized system:

$$\arg\min_{\Delta} J(x_0)\Delta - b(x_0) \tag{17}$$

where $J(x_0)$ is the sparse Jacobian Matrix at the current linearization point x_0 and $b(x_0) = f(x_0) - z$ is the residual for given the measurement z. The Jacobian matrix is equivalent to a linearized version of the factor graph, and its block structure reflects the structure of the factor graph. After solving (17), the linearization point is updated to the new estimate $x_0 + \Delta$. Further detail on this process is presented within [7].

The Jacobian for the Odometry is calculated from (4) as:

$$\frac{\partial(h^o(x_t^o))}{\partial x \partial y} = \begin{bmatrix} \frac{\partial(h^o(x_t,y_t))}{\partial x} & 0 \\ 0 & \frac{\partial(h^o(x_t,y_t))}{\partial y} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{18}$$

Further detail on this process is presented in [2]. The Jacobian for GPS from (6) is same as that of odometry.

The Jacobian for topology measurement from (11) with ∂x and ∂y is:

$$\text{diag}\Bigg[\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} 2*((p_{x,t-1}^i + \delta x_{[t-1][t]}^i) - (p_{x,t-1}^j + \delta x_{[t-1][t]}^j)), \sum_{i=1}^{N-1}\sum_{j=i+1}^{N} 2*((p_{y,t-1}^i + \delta y_{[t-1][t]}^i) - (p_{y,t-1}^j + \delta y_{[t-1][t]}^j))\Bigg] \tag{19}$$

IV. Evaluation

A. System Setup

Our simulation system setup is divided in two parts. In one we simulate random trajectory with 2 vehicles and the second is implemented with 2, 3 and 4 vehicles on a ground plane using 250 steps. To implement the factor graph and factors we utilize the Georgia Tech smoothing and Mapping (GTSAM) open source library (version 3.2.1) [18]. The tests were coded in C++ and run on an Ubuntu 14.04.4 LTS 64-bit machine with 16 GB RAM and Intel(R) Core(TM) i7-4710MQ CPU @ 2.50GHz processor.

In the simulated vehicles, GPS sensor gives location in global coordinates and Odometry provides relative measurements. RADAR is simulated as the infrastructure sensor and provides location in its local coordinate system. No configuration information for RADAR is available and hence transformation between the global coordinate system of GPS and local coordinate system of RADAR is unknown.

All sensors are assumed to have zero mean Gaussian noise. The covariances are assumed as diag[1.0, 1.0], diag[9.0, 9.0] and diag[0.1, 0.1] for the Odometry, the GPS and the RADAR sensor respectively. We assume the step interval, T as 1. It is also assumed that there is no false or missed detections during the whole process.

For Kalman Filter, using linear Gaussian Dynamics for constant velocity , the process model is represented as:

$$F = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, Q = \alpha^2 \begin{bmatrix} T^2/4 & T^2/2 & 0 & 0 \\ T^2/2 & T & 0 & 0 \\ 0 & 0 & T^2/4 & T^2/2 \\ 0 & 0 & T^2/2 & T \end{bmatrix} \tag{20}$$

where Q is the covariance of the noise w_k and α is the standard deviation of the process noise.

B. Results

For the first set of tests we compare fused results between: (1) the Odometry, the GPS and the old Topology Factor (as implemented in our previous work) [2]; and (2) the Odometry, the GPS and the new Topology Factor.

The second set of tests use Monte Carlo methodology. We compare and contrast our results three ways between: (1) the fused result of the measurements from the Odometry and the GPS using Kalman Filter; (2) the fused result of the measurements from the Odometry, the GPS and the old Topology Factor (as implemented in our previous work) [2]; and (3) the fused result of the measurements from the Odometry, the GPS and the new Topology Factor.

The performance for n steps of N vehicles is analyzed by calculating Root Mean Square Error (RMSE) values for the total system as:

$$Error = \sqrt{\frac{\sum_{j=1}^{n}\sum_{i=1}^{N}[(x_{i_{est}} - x_{i_{real}})^2 + (y_{i_{est}} - y_{i_{real}})^2]^j}{n}} \tag{21}$$

Fig. 3 shows the ground truth; and the trajectories from the two methods for 2 vehicles. As can be seen the new proposed

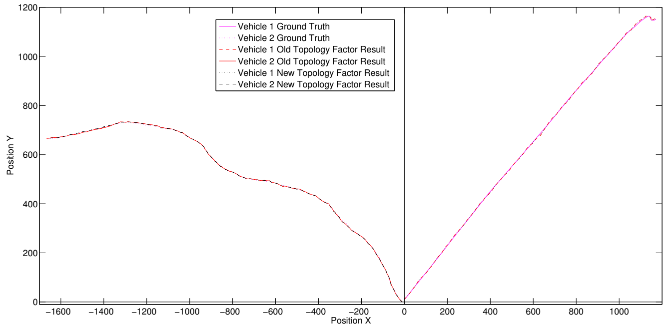

Fig. 3. Graph showing Ground Truth, fusion results for the old topology factor and the new topology factor for 2 Vehicles.

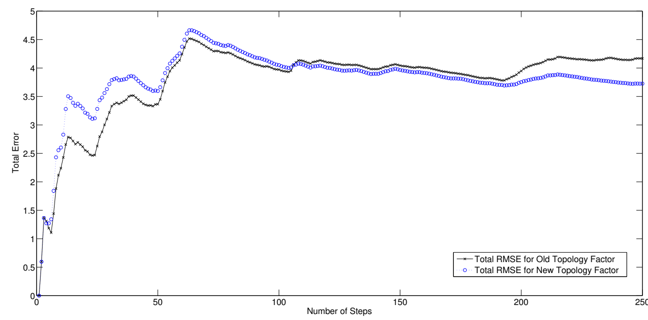

Fig. 4. Graph showing combined RMSE results for the old topology factor and the new topology factor for 2 vehicles.

Topology Factor results in trajectory which is closer to the Ground Truth than the old Topology Factor. Fig. 4 shows the performance by calculating RMSE values at each step for the total system. Although both the methods stabilize almost at the same rate, the method involving the new topology factors has the least total error.

For further result analysis we refer to the second set of tests using Monte Carlo methodology. Table I shows the average final RMSE values for 1000 iterations of 250 steps each, for systems having 2, 3 and 4 vehicles for each of the three methods. As seen the average RMSE value for the new Topology Factor is better than the other two. The last column shows the percentage decrease in the error for the new factor against the old factor. As can be seen the new topology factor performs at times about 27.5% better than the old. In case of three vehicles there is not a significant gain. This highlights another finding that at its worst the new topology factor cannot perform lower than the old factor.

TABLE I
AVERAGE FINAL RMSE VALUES FOR 1000 ITERATIONS

Total Vehicles	New Factor	Old Factor	Kalman Filter	% decrease of RMSE for the new factor
2	5.211	7.187	8.423	27.49
3	7.065	7.076	10.310	0.15
4	7.321	8.184	11.908	10.54

Average execution time is also analyzed using Monte Carlo methodology. Table II shows the average execution time (in seconds) for 1000 iterations of 250 steps each, for systems having 2, 3 and 4 vehicles each for all the three methods. The last column shows the percentage decrease in the execution time for the new factor against the old factor. Although the Kalman Filter performs better than the other two, but the new topology factor performs better than the old topology by at least 50% . This is because Kalman is a filter and calculates the current state while factor graph is smoothed using LM in a batch process.

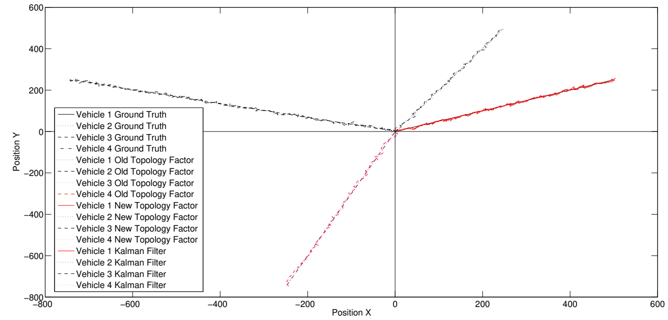

Fig. 5. Graph showing Ground Truth, fusion results for old topology factor, new topology factor and Kalman Filter for 4 Vehicles

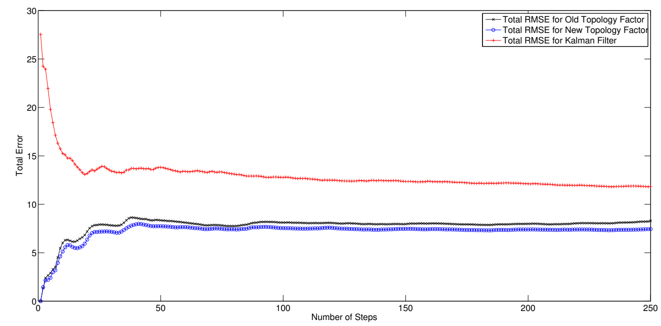

Fig. 6. Graph showing combined RMSE results for for old topology factor, new topology factor and Kalman Filter for 4 vehicles

TABLE II
AVERAGE COMPUTATION TIME (IN SECONDS) FOR 1000 ITERATIONS

Total Vehicles	New Factor	Old Factor	Kalman Filter	% decrease in execution time for the new factor
2	0.153	0.692	0.010	77.89
3	0.181	0.468	0.010	61.32
4	0.273	0.583	0.017	53.17

As seen with the above results, the new topology factor outperforms in cooperative localization for both the methods. This is possible because: (1) Like old topology, the new topology factor adds additional constraint of the topology to reduce the uncertainty in the position; (2) The current state in new topology factor is dependent on the previous state estimate, which not only further reduces the uncertainty, but also helps the LM algorithm to converge to the solution faster, thereby providing better performance than the old.

The solution is scalable for any number of vehicles because the new topology factor (13) remains quadratic for any number of vehicles. Fig. 5 and Fig. 6 show a sample result from the second set of tests for 4 vehicles. The bandwidth requirement remains minimal as only the observations, and not the full covariance matrices, of the sensors are communicated on the network. If a measurement needs 1 byte for x and y each, then for N vehicles require only $2 * N$ bytes at any step

The results presented here support a solution which is superior both in terms of precision and computation time than [2]. The new implementation remains low bandwidth and scalable. The new topology factor continues to avoid the computational and informational burden of data association. The coordinate transformation of nodes within the system are still derived as part of the state estimate. Han-Pang C. et al. demonstrated the plug-and-play feature using Factor Graphs [19]. Our solution also has a similar potential where the constraint factors can be added/removed depending on the measurements from the RADAR.

Although the methodology indicates improved performance in terms of accuracy of localization over the industry wide Kalman Filter implementation but it cannot match the execution performance. A point to be noted is that the better accuracy is achieved by the introduction of the topology factors between the the state nodes. From a Bayesian perspective this results in introduction of new common information between nodes and results in the reduction of the uncertainty estimate.

Presently, the solution is implemented as a batch process and all factors influence the joint state estimate. This is not a viable solution at run-time and some form of local smoothing estimate should be used.

Finally, the results presented here assume an ideal environment without clutter, obscuration, false or missed detection and without the introduction of new vehicles to the system. Further work should evaluate the robustness of the solution in live scenarios including all of the above challenges.

V. Conclusion

In this paper, an improvement over our previous solution based on factor graphs for vehicle-infrastructure cooperative localization is presented. The solution continues to addresses various challenges for this problem, namely, the bandwidth issue, data association uncertainties, coordinate transformation overheads, and scalability, but also shows reduced error in localization by almost 27% in the best case and decreased computational time by at least 50%. This improvement increases the potential to solve a number of challenges in the highly assisted and autonomous driving communities in real time scenarios. The proposed solution is evaluated and discussed using simulated data. Our simulations indicate improved RMSE performance over both the old solution and the traditional Kalman Filter approach. This is achieved through the introduction of improved topology factor, interconnecting all of the nodes to their previous states within the system.

Future work will focus on the implementation of the presented approach with effects like clutter and obscuration with incremental smoothing. This will also demonstrate the plug and play feature for the dynamic environment.

Acknowledgment

This work was supported by fortiss GmbH and BMWi IKT III SADA Project http://www.projekt-sada.de/.

References

[1] R. Kurazume, S. Nagata, and S. Hirose, "Cooperative positioning with multiple robots," in *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, May 1994, pp. 1250–1257 vol.2.

[2] D. Gulati, F. Zhang, D. Clarke, and A. Knoll, "Vehicle infrastructure cooperative localization using factor graphs," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, June 2016, pp. 1085–1090.

[3] A. Howard, M. J. Matark, and G. S. Sukhatme, "Localization for mobile robot teams using maximum likelihood estimation," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 1, 2002, pp. 434–439 vol.1.

[4] M. Montemerlo and S. Thrun, "Simultaneous localization and mapping with unknown data association using fastslam," in *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, vol. 2, Sept 2003, pp. 1985–1991 vol.2.

[5] F. Zhang, G. Hinz, D. Gulati, D. Clarke, and A. Knoll, "Cooperative vehicle-infrastructure localization based on the symmetric measurement equation filter," *Geoinformatica*, vol. 20, no. 2, pp. 159–178, Apr. 2016. [Online]. Available: http://dx.doi.org/10.1007/s10707-016-0244-3

[6] E. D. Nerurkar, S. I. Roumeliotis, and A. Martinelli, "Distributed maximum a posteriori estimation for multi-robot cooperative localization," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, May 2009, pp. 1402–1409.

[7] V. Indelman, S. Williams, M. Kaess, and F. Dellaert, "Factor graph based incremental smoothing in inertial navigation systems," in *Information Fusion (FUSION), 2012 15th International Conference on*, July 2012, pp. 2154–2161.

[8] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, Feb 2001.

[9] A. Makarau, G. Palubinskas, and P. Reinartz, "Alphabet-based multisensory data fusion and classification using factor graphs," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 6, no. 2, pp. 969–990, April 2013.

[10] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Multi-target tracking using joint probabilistic data association," in *Decision and Control including the Symposium on Adaptive Processes, 1980 19th IEEE Conference on*, Dec 1980, pp. 807–812.

[11] R. P. S. Mahler, "Multitarget bayes filtering via first-order multitarget moments," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1152–1178, Oct 2003.

[12] D. Reid, "An algorithm for tracking multiple targets," *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, Dec 1979.

[13] S. H. Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. Dick, and I. Reid, "Joint probabilistic data association revisited," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 3047–3055.

[14] R. L. Streit and T. E. Luginbuhl, "A probabilistic multi-hypothesis tracking algorithm without enumeration and pruning," in *Proceedings of the Sixth Joint Service Data Fusion Symposium*, 1993, pp. 1015–1024.

[15] E. Giannopoulos, R. Streit, and P. Swaszek, "Probabilistic multi-hypothesis tracking in a multi-sensor, multi-target environment," in *Data Fusion Symposium, 1996. ADFS '96., First Australian*, Nov 1996, pp. 184–189.

[16] F. Dellaert and M. Kaess, "Square root sam: Simultaneous localization and mapping via square root information smoothing," *Int. J. Rob. Res.*, vol. 25, no. 12, pp. 1181–1203, Dec. 2006. [Online]. Available: http://dx.doi.org/10.1177/0278364906072768

[17] K. O. Arras, "An introduction to error propagation: Derivation, meaning and examples of cy= fx cx fx," Tech. Rep., 1998.

[18] "GTSAM, Georgia Tech smoothing and Mapping," https://collab.cc.gatech.edu/borg/gtsam/.

[19] H. P. Chiu, X. S. Zhou, L. Carlone, F. Dellaert, S. Samarasekera, and R. Kumar, "Constrained optimal selection for multi-sensor robot navigation using plug-and-play factor graphs," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 663–670.