Cooperative Node Positioning In Vehicular Networks Using Inter-Node Distance Measurements

Peyman Hadi Mohammadabadi
Department of Electrical and Computer Engineering
University of Toronto
Toronto, Ontario, Canada
Email: hadi@ece.utoronto.ca

Shahrokh Valaee
Department of Electrical and Computer Engineering
University of Toronto
Toronto, Ontario, Canada
Email: valaee@ece.utoronto.ca

***Abstract*—This paper presents a novel cooperative positioning (CP) method to increase localization accuracy in vehicular ad-hoc networks (VANET). The proposed method uses a semi-extended Kalman filter to fuse position data and distance information of nodes in the network. This paper also introduces a new distance measurement method using the time-difference-of-arrival in positive orthogonal codes. Simulation results show that the proposed method outperforms other cooperative positioning systems, both for low and high traffic densities.**

I. Introduction

Most applications in vehicular ad hoc networks (VANETs) need the position of vehicles to be accurately estimated in real-time [1]. For instance, the exact location of an accident should be distributed to vehicles arriving at the accident scene. Conventionally, the location of a vehicle is found by GPS devices. However, GPS needs line-of-sight to at least four positioning satellites, which limits the availability and the accuracy of the technology in covered areas, and downtown cores populated by high-rise buildings.

There has been growing interest in recent years on cooperative positioning (CP) in vehicular networks. Cooperative positioning (CP) methods can mitigate the GPS error by incorporating other information such as distance measurements between neighboring nodes. Drawil *et al.* [2] use a Kalman filter to fuse the information of GPS position and inertial navigation system (INS) of vehicle to increase the accuracy of positioning. Whenever a vehicle enters a multipath environment, the vehicle uses triangulation to estimate its position based on the position of three neighbors in its communication area. Although the presented method in [2] is less complex than [3], this method does not use distance information and position of nodes simultaneously.

In the cooperative inertial navigation (CIN) introduced in [4], [5], vehicles communicate their inertial measurement unit (IMU) and their INS-based positioning data with vehicles travelling on the opposite direction of road. This information is fused with carrier frequency offset (CFO) of received packets by each vehicle to improve localization accuracy. In [6], the CP method uses a Doppler-based range rating. GPS pseudo-range sharing for cooperative localization are used in [7], [8]. GPS pseudo-ranging is used to improve inter-vehicle distance ranging.

VANET Localization Improve (VLOCI) algorithm is proposed in [9]. It is assumed that all vehicles are travelling only on one lane. The VLOCI algorithm attempts to improve the location of each vehicle only on one dimension. The position of each node is estimated based on the position of its neighbours and their corresponding distances. The VLOCI2 algorithm is proposed by the same authors in [10] to solve the localization problem in a 2D space.

The present paper has two contributions. First, we propose a new distance ranging method through the use of *positive orthogonal codes* (POC) [11]. We assume that vehicles use a POC-based medium access control. The POC-based MAC is an enhancement to the IEEE 802.11p WAVE standard, which reduces the probability of packet collision in the wireless channel. The POC codes are synchronized and their time-difference-of-arrival is used to estimate the distance between any two pair of nodes.

As the second contribution, we propose a *semi-extended* Kalman filter to reduce the nonlinear behavior of localization system. Our work is an extension of [12] where Kalman filter was used to reduce the location estimation error in a vehicular environment. Parker and Valaee used an Extended Kalman filter to fuse the position information of vehicles and their relative distances. To estimate the distances between nodes, the received-signal-strength indicator (RSSI) was used in [3], [12], [13]. Our approach is different from the above work in the way our Extended Kalman Filter is designed. We decompose the observation vector into two linear and nonlinear fragments and apply a linear approximation to only the nonlinear component.

II. Problem formulation

A. Definition

1) Cluster: Consider *N* mobile nodes in a *2D* space. These nodes can be vehicles, pedal-cyclists, or pedestrians. Each node is equipped with a GPS receiver and it can communicate with other nodes in its communication range. For every node *i*, $1 \leq i \leq N$, cluster C_i is defined as the set of nodes, including node *i*, that can communicate with node *i*, i.e.,

$$C_i = \{j : 1 \leq j \leq N, distance(i, j) \leq \min(R_{c_i}, R_{c_j})\} \quad (1)$$

978-1-4799-4912-0/14/$31.00 ©2014 IEEE

0	0	1	0	1	0	0	1	0	0

Fig. 1. Transmission pattern for an arbitrary node with POC codeword of length $L = 10$ and Hamming weight $w = 3$. L indicates the number of time slots at each time frame and w represents the number of transmissions for each node during one time frame. Each transmission time slot is shown with a 1 and idle (no transmission) time slots are shown with 0.

where R_{c_i} is the communication range of node *i*; and $distance(i, j) \leq \min(R_{c_i}, R_{c_j})$ indicates that both nodes *i* and *j* can communicate with each other.

2) Positive Orthogonal Code (POC): It is assumed that all nodes are using synchronous Positive Orthogonal Codes (POC) MAC protocol to transmit safety packets to other nodes [11], [14]. Binary codewords x and y are synchronous POC of length L and weight w if cross-correlation between them is at most λ [15]:

$$\sum_{i=1}^{L} x_i y_i \leq \lambda. \tag{2}$$

A synchronous POC codeword is assigned to each node. As shown in Fig. 1 at each time frame, nodes obtain their transmission pattern through their POC codeword in a synchronous POC MAC protocol.

It is assumed that all nodes in a cluster are *frame-synchronous* and *slot-synchronous*. Since all nodes are using GPS devices, synchronization can be done by utilizing the clock of the GPS signal.

Each node broadcasts safety packets during its allocated time slots at each time frame. Since all nodes in a cluster are *slot-synchronous*, each node finds its distance from the broadcasting node by using time difference of received packets.

Several approaches have been introduced in the literature to reduce the GPS jitter time. In [16], an average standard deviation of $7nsec$ is reported as the error of GPS timing. This translates to about 2.1 meters of error in ranging through TOA.

B. Semi-Extended Kalman Filter

For the simplicity of notation, in the sequel we will drop the index i from each cluster and its corresponding parameters. For each cluster $\mathcal{C}$, vector $\mathbf{a}_k$ represents the position of nodes in $\mathcal{C}$ at time step *k*,

$$\mathbf{a}_k = [x_{1,k}, \ldots, x_{n,k}, y_{1,k}, \ldots, y_{n,k}]^T \tag{3}$$

where n is the number of nodes in the cluster at the *k*th time step (n may vary at each time step); superscript T denotes transposition; $(x_{i,k}, y_{i,k})$ is the position of node *i* at time step *k*. Position of nodes at each time step can be modeled as:

$$\mathbf{a}_k = \mathbf{a}_{k-1} + T_s \mathbf{v}_{k-1} + T_s \mathbf{\Omega}_{k-1} \tag{4}$$

where $\mathbf{\Omega}_{k-1}$ is the processing noise, which describes the mobility variation; T_s is the sampling time; and vector $\mathbf{v}_{k-1}$ is the velocity vector at time $(k-1)$:

$$\mathbf{v}_{k-1} = [v_{x_1,k-1}, \ldots, v_{x_n,k-1}, v_{y_1,k-1}, \ldots, v_{y_n,k-1}]^T \tag{5}$$

TABLE I
NOTATIONS FOR POSITION ESTIMATES AND COVARIANCE MATRICES

Notation	Mathematical representation	Definition
$\hat{\mathbf{a}}_k^-$	$\hat{\mathbf{a}}_{k\|k-1}$	a *priori* estimate
$\mathbf{P}_k^-$	$\mathbf{P}_{k\|k-1}$	a *priori* covariance
$\hat{\mathbf{a}}_k^+$	$\hat{\mathbf{a}}_{k\|k}$	a *posteriori* estimate
$\mathbf{P}_k^+$	$\mathbf{P}_{k\|k}$	a *posteriori* covariance

where $(v_{x_i,k-1}, v_{x_i,k-1})$ represents the velocity of node *i* at time step $k-1$.

In our proposed method, it is assumed that every node uses a GPS device to find its location. Each node measures its distance from the neighboring nodes through the time differences of the POC codewords. Then, nodes broadcast their measured position and ranging information to their neighborhood. Hence, every node would have the reported location of the its neighbors and the relative distance between each pair of nodes in the cluster.

Let $r_{(i,j),k}$ be the distance between node *i* and node *j* measured by node *i* at time step *k*. Since two measurements are available for each link, the measurements can be averaged to get

$$r\prime_{(i,j),k} = \frac{w_{i,j} r_{(i,j),k} + w_{j,i} r_{(j,i),k}}{w_{i,j} + w_{j,i}} \tag{6}$$

where $w_{i,j}$ can be adjusted based on the accuracy of each measurement. Therefore, two vectors of measurements are available

$$\mathbf{z}_{k_1} = \mathbf{a}_k \tag{7}$$

$$\mathbf{z}_{k_2} = [r\prime_{(1,2),k}, \ldots, r\prime_{(1,n),k}, r\prime_{(2,n),k}, \ldots, r\prime_{(n-1,n),k}]^T \tag{8}$$

which can be concatenated to build the measurement vector

$$\mathbf{z}_k = \begin{bmatrix} \mathbf{z}_{k_1} \\ \mathbf{z}_{k_2} \end{bmatrix}. \tag{9}$$

The proposed extended Kalman filter can be modeled as:

$$\hat{\mathbf{a}}_k^- = \hat{\mathbf{a}}_{k-1}^+ + T_s \mathbf{v}_{k-1} \tag{10}$$

$$\mathbf{P}_k^- = T_s^2 \mathbf{\Gamma}_{k-1} + \mathbf{P}_{k-1}^+ \tag{11}$$

$$\hat{\mathbf{a}}_k^+ = \hat{\mathbf{a}}_k^- + \mathbf{K}_k (\mathbf{z}_k - h_k(\hat{\mathbf{a}}_k^-)) \tag{12}$$

$$\mathbf{P}_k^+ = \mathbf{P}_k^- - \mathbf{K}_k \hat{\mathbf{H}}_k \mathbf{P}_k^- \tag{13}$$

where the definition of variables is given in Table I; $\mathbf{\Gamma}_{k-1}$ is the covariance matrix of mobility error; $\mathbf{K}_k$ is the Kalman filter gain at time step k, which is defined as:

$$\mathbf{K}_k = \mathbf{P}_k^- \hat{\mathbf{H}}_k^T (\hat{\mathbf{H}}_k \mathbf{P}_k^- \hat{\mathbf{H}}_k^T + \mathcal{R}_k)^{-1} \tag{14}$$

where $\mathcal{R}_k$ is the covariance matrix of observation vector $\mathbf{z}_k$.

Function $h_k(\hat{\mathbf{a}}_k^-)$ models the relation between the estimated position of nodes $\hat{\mathbf{a}}_{\mathbf{k}}^-$ and the observation vector $\mathbf{z}_k$. Similar to $\mathbf{z}_k$, function $h_k(\hat{\mathbf{a}}_k^-)$ is obtained by concatenating two sub-functions $h_{k_1}(\hat{\mathbf{a}}_k^-)$ and $h_{k_2}(\hat{\mathbf{a}}_k^-)$, where $h_{k_1}(\hat{\mathbf{a}}_k^-)$ models the relation between *a priori* estimate $\hat{\mathbf{a}}_k^-$ and observed GPS position $\mathbf{a}_k$; $h_{k_2}(\hat{\mathbf{a}}_k^-)$ models the relation between *a priori* estimate $\hat{\mathbf{a}}_k^-$ and measured distance between nodes:

$$h_k(\hat{\mathbf{a}}_k^-) \;=\; \begin{bmatrix} h_{k_1}(\hat{\mathbf{a}}_k^-) \\ h_{k_2}(\hat{\mathbf{a}}_k^-) \end{bmatrix} \tag{15}$$

$$h_{k_1}(\hat{\mathbf{a}}_k^-) = \mathbf{I}_{2n\times 2n}\; \hat{\mathbf{a}}_k^- \tag{16}$$

$$\begin{aligned} h_{k_2}(\hat{\mathbf{a}}_k^-) = [&f_{1,2}(\hat{\mathbf{a}}_k^-),\ldots,f_{1,n}(\hat{\mathbf{a}}_k^-), \\ &f_{2,n}(\hat{\mathbf{a}}_k^-),\ldots,f_{n-1,n}(\hat{\mathbf{a}}_k^-)]^T \end{aligned} \tag{17}$$

where

$$f_{i,j}(\mathbf{a}_k) = \sqrt{(x_{i,k}-x_{j,k})^2+(y_{i,k}-y_{j,k})^2} \tag{18}$$

is the Euclidean distance between a pair of nodes and $\mathbf{I}_{2n\times 2n}$ is the unitary matrix.

The Kalman filter is optimal when: (1) noise is Gaussian and (2) there is a linear model between the state variables $\mathbf{a}_k$ and the observation vector $\mathbf{z}_k$. Here, a part of the observation vector $\mathbf{z}_{k_2}$ is not a linear function of the state variables $\mathbf{a}_k^-$. Therefore, an extended Kalman filter should be used. A linear estimate of function $h(\mathbf{a}_k)$ is needed for the extended Kalman filter. This can be achieved by using the *Jacobian* matrix of $h(\mathbf{a}_k)$

$$\hat{\mathbf{H}}_k = \frac{\partial h_k(\mathbf{a})}{\partial \mathbf{a}_i}\Big|_{\mathbf{a}=\hat{\mathbf{a}}_k^-}. \tag{19}$$

The extended Kalman filter in [3], [12] treats the GPS data position as the distance from the origin (i.e. distance from point $(0,0)$). As discussed above, the Kalman filter is optimal for linear systems. Therefore, treating the position data as distance reduces the accuracy of the Kalman filter. Hence, we introduce our *Semi-Extended Kalman filter*. This filter treats data based on its type. In other words, this filter uses the regular Kalman filter for linear data (position data) and it uses the extended Kalman filter for non-linear data (distances).

As illustrated earlier, $h_{k_2}(\mathbf{a}_k^-)$ is not a linear function of $\mathbf{a}_k^-$. Therefore, (19) is used to find the linear estimate of $h_{k_2}(\mathbf{a}_k^-)$. Hence, matrix $\hat{\mathbf{H}}_k$ can be written as:

$$\hat{\mathbf{H}}_k \;=\; \begin{bmatrix} \hat{\mathbf{H}}_{k_1} \\ \hat{\mathbf{H}}_{k_2} \end{bmatrix} \tag{20}$$

$$\hat{\mathbf{H}}_{k_1} = \mathbf{I}_{2n\times 2n} \tag{21}$$

$$\begin{aligned} \hat{\mathbf{H}}_{k_2} &= \frac{\partial}{\partial a_i} h_{k_1}(\mathbf{a})\big|_{\mathbf{a}=\hat{\mathbf{a}}_k^-} \\ &\triangleq \begin{bmatrix} \frac{\partial}{\partial x_1} f_{1,2}(\mathbf{a})\big|_{\mathbf{a}=\hat{\mathbf{a}}_k^-} & \cdots & \frac{\partial}{\partial y_n} f_{1,2}(\mathbf{a})\big|_{\mathbf{a}=\hat{\mathbf{a}}_k^-} \\ \vdots & \ddots & \vdots \\ \frac{\partial}{\partial x_1} f_{1,n}(\mathbf{a})\big|_{\mathbf{a}=\hat{\mathbf{a}}_k^-} & \cdots & \frac{\partial}{\partial y_n} f_{1,n}(\mathbf{a})\big|_{\mathbf{a}=\hat{\mathbf{a}}_k^-} \\ \frac{\partial}{\partial x_1} f_{2,3}(\mathbf{a})\big|_{\mathbf{a}=\hat{\mathbf{a}}_k^-} & \cdots & \frac{\partial}{\partial y_n} f_{2,3}(\mathbf{a})\big|_{\mathbf{a}=\hat{\mathbf{a}}_k^-} \\ \vdots & \ddots & \vdots \\ \frac{\partial}{\partial x_1} f_{n-1,n}(\mathbf{a})\big|_{\mathbf{a}=\hat{\mathbf{a}}_k^-} & \cdots & \frac{\partial}{\partial y_n} f_{n-1,n}(\mathbf{a})\big|_{\mathbf{a}=\hat{\mathbf{a}}_k^-} \end{bmatrix} \end{aligned} \tag{22}$$

where $f_{i,j}(\mathbf{a}), \forall i,j \in \mathcal{C}$ is defined in (18). We note that $\hat{\mathbf{H}}_{k_2}$ is a $n(n-1)/2 \times 2n$ matrix. Each row is related to a distance measurement between two nodes.

As mentioned in (8), each element of vector $\mathbf{z}_{k_2}$ represents weighted mean of reported values for distance between two specific nodes. If one of these two measurements is missing, the other reported measurement should be used instead of the weighted mean. For example, for nodes *i* and *j*, if $r_{(i,j),k}$ is missing, $r\prime_{(i,j),k}$ is replaced by $r_{(j,i),k}$. Moreover, if both $r_{(i,j),k}$ and $r_{(j,i),k}$ are missing, then $r\prime_{(i,j),k}$ should be deleted from $\mathbf{z}_{k_2}$.

As it can be observed from (10-11), at time step *k*, the Kalman filter uses the estimated values of the previous time step $k-1$. Therefore, it is necessary to initialize the system for time step $k=1$. The following equations illustrate how to initialize the system:

$$\hat{\mathbf{a}}_0^+ = E(\mathbf{a}_0) \tag{23}$$

$$\mathbf{P}_0^+ = E[(\hat{\mathbf{a}}_0^+ - \mathbf{a}_0)(\hat{\mathbf{a}}_0^+ - \mathbf{a}_0)^T] \tag{24}$$

If the exact position of each node is used to initialize the Kalman filter, then $\mathbf{P}_0^+ = \mathbf{0}$. If there is no information about $\hat{\mathbf{a}}_0^+$, then $\mathbf{P}_0^+ = \infty\mathbf{I}_{2n\times 2n}$ [3].

C. Map matching

Map information can be used to improve the localization accuracy. Each node can be localized only on specific locations. For instance, a vehicle is expected to be localized on the roads and pedestrians are normally localized on sidewalks. In [12], authors used a projection method to enforce map constraint on the position estimate of Kalman filter.

Projection method is not useful when nodes are close to intersections. In such situations, nodes might be projected onto wrong roads. To overcome this issue, we define the following optimization problem to enforce map constraints:

$$\mathcal{L}_j = \arg\max_i \cos(\theta_{i,j})/d_{i,j} \tag{25}$$

where $\theta_{i,j}$ is the angle between the direction of lane *i* and movement direction of node *j*; $d_{i,j}$ is the distance between estimated position of node *j* and lane *i*; and $\mathcal{L}_j$ is the lane selected for node *i*.

The optimization problem in (25) finds the lane, which is close to the estimated position for the node, while it considers the direction of road and node movement. This method prevents choosing a wrong lane especially when a node is close to an intersection. Furthermore, it avoids choosing lanes with nodes moving in the opposite direction.

III. SIMULATIONS

Two different simulation scenarios are used to study the performance of our method. The first scenario is a single road with light traffic. The study area of the second scenario has several roads with different traffic densities on each road (Fig. 2).

In both scenarios, it is assumed that each lane is 3.66 meters (12 feet) [17]. The communication range for each node is set to 50 meters. When a vehicle exits a road, a new vehicle enters the road on the same lane.

The velocity measurement noise is assumed to be AWGN. The standard deviation for the velocity measurement error is

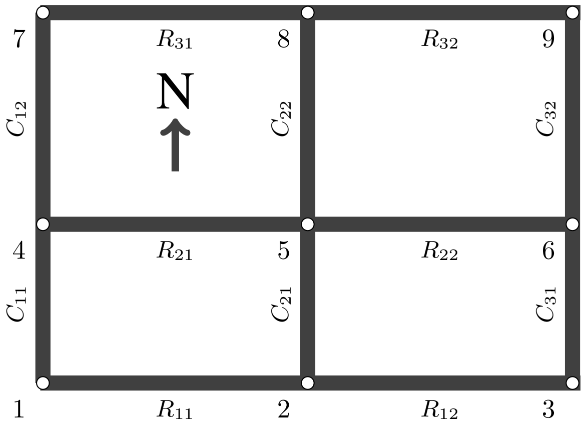

Fig. 2. Multi-road scenario, there are 6 different road in this scenario and vehicles are traveling through these roads. The details of this scenario is presented in table II

set to 1 meter in the direction of the road where the node is traveling on, and 0.2 meter on the orthogonal direction of the road. The distance measurement noise is assumed to be AWGN with the standard deviation $\sigma_d = 3$ meters. In both scenarios, $w_{(i,j),k}$ is set to0.5.

1) Single-Road Scenario: Consider a 3-lane road of length 1 $kilometer$. There are 8 vehicles on each lane (24 vehicles in total). At time $k = 1$, the vehicles on the first lane are located at random on the first 200 meters of the road. The vehicles on the second lane of the road are located at random on the interval between 400 and 600] meters from the start of road. The vehicles on the Third lane of the road are located at random on the interval between 200 and 400] meters from the start of road.

The average velocity for each vehicle is 80 Km/h. Simulation time is 45 seconds, which is the time needed for every node to completely pass through the road. For this scenario, the sampling time T_s is set to 1 second.

2) Multi-Road Scenario: At this scenario, vehicles are traveling on the map shown in Fig. 2. Table II describes details of this scenario. It is assumed that the initial spacing between cars on each lane is an exponentially distributed random variable with the average distance λ_r, where $r \in \{R_{11}, R_{12}, R_{21}, R_{22}, R_{31}, R_{32}, C_{11}, C_{12}, C_{21}, C_{22}, C_{31}, C_{32}\}$. The length of each segment of the road is given as below:

$$\begin{aligned} C_{11} &= C_{21} = C_{31} = 400\ m \\ C_{12} &= C_{22} = C_{32} = 600\ m \\ R_{11} &= R_{21} = R_{31} = 500\ m \\ R_{12} &= R_{22} = R_{32} = 500\ m \end{aligned}$$

The simulation time for this scenario is 90 seconds. During this time, each node travels at least 1000 meters. Similar to the Single-Road scenario, T_s is set to be 1 second.

A. Results

The performance of our method in this paper is compared with VLOCI2 algorithm [9] [10] and the extended Kalman filter presented in [3], [12], [13]. The following performance merit is used:

TABLE II
SIMULATION DETAILS FOR MULTI-ROAD SCENARIO

Road Name	Direction	Lanes	Average Velocity	Average Distance	Vehicles
R_{11}, R_{12}	West East	2 2	50 Km/h	20 m	50×4
R_{21}, R_{22}	West East	2 3	80 Km/h	35 m	29×5
R_{31}, R_{32}	West East	2 1	60 Km/h	20 m	50×3
C_{11}, C_{12}	South	1	40 Km/h	15 m	67×1
C_{21}, C_{22}	South North	3 3	80 Km/h	40 m	25×6
C_{31}, C_{32}	North	2	60 Km/h	30 m	33×2

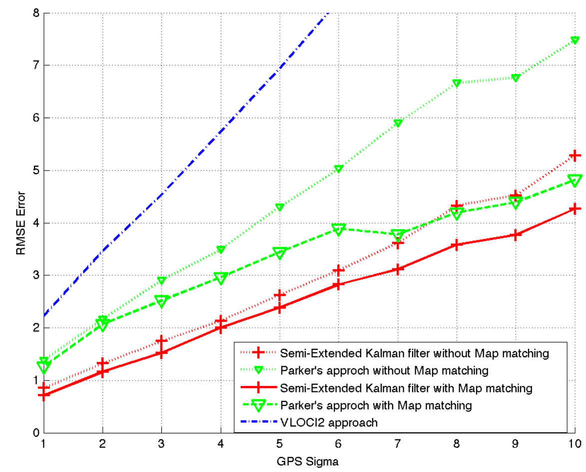

Fig. 3. Mean error Vs. GPS error for single road Scenario

$$\sigma_{pos} = \sum_{i=1}^{N} \sqrt{\frac{(\hat{x}_{i,k} - x_{i,k})^2 + (\hat{y}_{i,k} - y_{i,k})^2}{N}} \tag{26}$$

where $(\hat{x}_{i,k}, \hat{y}_{i,k})$ is the estimated position of vehicle i at time step k and $(x_{i,k}, y_{i,k})$ is the actual position of vehicle i at time step k [13].

1) Results for Single-Road Scenario: Fig. 3 depicts the root mean squared error (RMSE) σ_{pos} versus the standard deviation of GPS error σ_{GPS}.

As illustrated in Fig. 3, our localization method (without using map matching) performs upto 50% better than the method proposed in [12]. Furthermore, it can be obtained that the map matching increases localization accuracy.

As seen in Fig. 3, our method performs much better than other methods when the cluster size is small. The study of small cluster size is useful for rural areas where traffic density is expected to be low.

2) Results for Multi-Road Scenario: Fig. 4 depicts RMSE error σ_{pos} versus the standard deviation of GPS error, σ_{GPS}. As seen in Fig. 4, our method always performs better than those in [12] and [10]. By means of the map matching, the localization error is reduced and our algorithm can reach a reliable localization accuracy of 2.6 meters while the GPS error has the standard deviation $\sigma_{GPS} = 10$ meters.

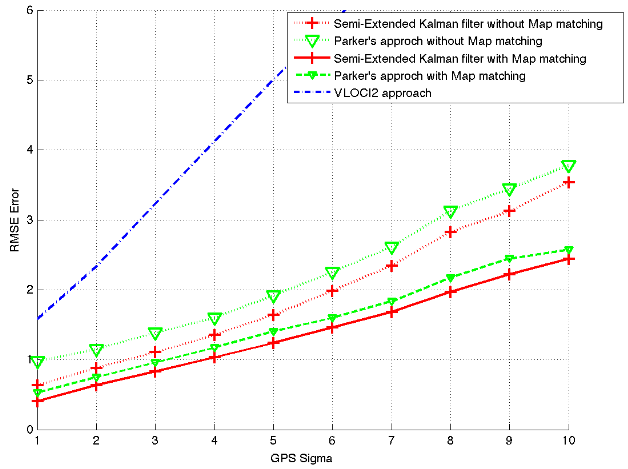

Fig. 4. Mean error Vs. GPS error for multi-road Scenario

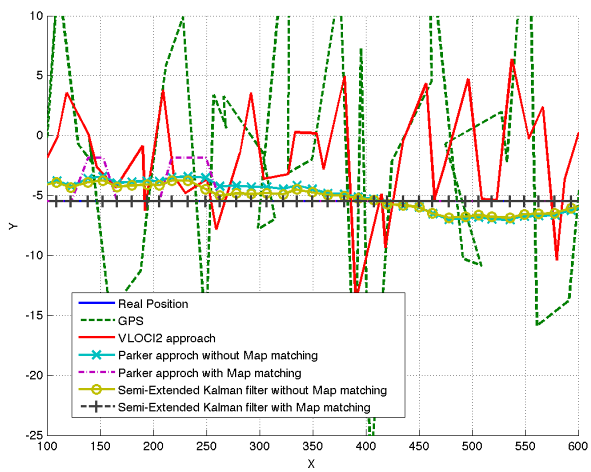

Fig. 5. Real position, GPS reading, and estimated position obtained by different method for an arbitrary vehicle travels on the second lane of road R1 toward East ($y = -5.49m$).

Fig. 5 depicts the real position, the GPS reading and the estimated path of an arbitrary vehicle obtained by different positioning methods in the multi road scenario. The standard deviation of the GPS error is set to $\sigma_{GPS} = 10m$ for this simulation. It can be seen that for this vehicle, the error of our method is smaller than other methods. The map matching improves the accuracy of the positioning method and the vehicle is localized on correct lane.

IV. Conclusion

This paper presents a novel distributed positioning algorithm for location estimation in vehicular networks. An accurate distance ranging method has been proposed that uses synchronous positive orthogonal codes (POC). In POC codes, nodes use the GPS clock to synchronize and transmit their packets on specific frames. The time-difference-of-arrival can be used to estimate the distance between any transmitter-receiver pairs. This approach provides a convenient ranging method, hence removes the need for expensive high frequency ranging radars. The ranging information is then used, along with the GPS estimates, in a semi-extended Kalman filter where linear approximation is only applied to the nonlinear segment of the state vector. The estimated vehicle position is next compared to the local map to further remove outliers. The proposed map-matching method considers both road boundaries and the direction of move. This method prevents nodes to be localized on roads with wrong direction, specially when nodes are close to an intersection. Simulation results show that our localization method outperforms other cooperative positioning methods.

References

[1] A. Boukerche, H. A. Oliveira, E. F. Nakamura, and A. A. Loureiro, "Vehicular ad hoc networks: A new challenge for localization-based systems," *Computer communications*, vol. 31, no. 12, pp. 2838–2849, 2008.

[2] N. M. Drawil and O. Basir, "Intervehicle-communication-assisted localization," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, no. 3, pp. 678–691, 2010.

[3] R. parker, "Cooperative vehicle position estiomation," Master's thesis, University of Toronto, 2007.

[4] N. Alam, A. Kealy, and A. G. Dempster, "Cooperative inertial navigation for gnss-challenged vehicular environments," 2013.

[5] ——, "An ins-aided tight integration approach for relative positioning enhancement in vanets," 2013.

[6] N. Alam and A. G. Dempster, "Cooperative positioning for vehicular networks: Facts and future," 2013.

[7] N. Alam, A. Tabatabaei Balaei, and A. G. Dempster, "Relative positioning enhancement in vanets: A tight integration approach," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 14, no. 1, pp. 47–55, 2013.

[8] D. Yang, F. Zhao, K. Liu, H. B. Lim, E. Frazzoli, and D. Rus, "A gps pseudorange based cooperative vehicular distance measurement technique," in *Vehicular Technology Conference (VTC Spring), 2012 IEEE 75th*. IEEE, 2012, pp. 1–5.

[9] F. Ahammed, J. Taheri, A. Y. Zomaya, and M. Ott, "Vloci: Using distance measurements to improve the accuracy of location coordinates in gps-equipped vanets," in *Mobile and Ubiquitous Systems: Computing, Networking, and Services*. Springer, 2012, pp. 149–161.

[10] ——, "Vloci2: improving 2d location coordinates using distance measurements in gps-equipped vanets," in *Proceedings of the 14th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*. ACM, 2011, pp. 317–322.

[11] F. Farnoud and S. Valaee, "Reliable broadcast of safety messages in vehicular ad hoc networks," in *INFOCOM 2009, IEEE*. IEEE, 2009, pp. 226–234.

[12] R. Parker and S. Valaee, "Vehicular node localization using received-signal-strength indicator," *Vehicular Technology, IEEE Transactions on*, vol. 56, no. 6, pp. 3371–3380, 2007.

[13] ——, "Cooperative vehicle position estimation," in *Communications, 2007. ICC'07. IEEE International Conference on*. IEEE, 2007, pp. 5837–5842.

[14] F. Farnoud, B. Hassanabadi, and S. Valaee, "Message broadcast using optical orthogonal codes in vehicular communication systems," in *The First International Workshop on Wireless Networking for Intelligent Transportation Systems*. ACM, 2007, p. 4.

[15] L. Zhang, "Positive orthogonal code-based cooperative forwarding for vanets," Master's thesis, University of Toronto, 2010.

[16] Z. Deng, G. Zheng, and J. Ni, "Time synchronization using single gps-satellite and multiple measurement stations," in *Internet Computing for Science and Engineering (ICICSE), 2012 Sixth International Conference on*. IEEE, 2012, pp. 227–234.

[17] U. D. of Transportation Federal Highway Administration. (2007) Mitigation strategies for design exceptions. [Online]. Available: http://safety.fhwa.dot.gov/geometric/pubs/mitigationstrategies/chapter3/3_lanewidth.htm#content