# Performance Evaluation of Movement Prediction Techniques for Vehicular Networks

Noura Aljeri
EECS, University of Ottawa, Canada

Azzedine Boukerche
EECS, University of Ottawa , Canada

***Abstract*—Intelligent Transportation Systems have recently received great deal of attention and Vehicular networks and its applications represent a major part of ITS. Many vehicular network applications require accurate location information to improve their performance. Over the past years, many researchers worked on state prediction/estimation techniques in tracking, navigation applications for mobile ad hoc networks and wireless sensor networks. Yet, few were into the field of Vehicular networks. In this paper, We study five different movement prediction models and their efficiency and effectiveness for VANETs. We compare them using both real vehicle mobility traces of taxi cabs and generated mobility traces from SUMO.**

## I. Introduction

Smart Vehicular NETworks (SVNET) is one of the major component of Intelligent transportation systems (ITS). Vehicular network is vital to provide safety, assistance to drivers, and traffic control, etc. In many Vehicular NETworks applications, accurate location information is essential. Wireless network management, routing, mobility management, service discovery, and collision avoidance protocols would enhance their capabilities with a prior knowledge of the vehicle next location. To the best of our knowledge, limited work has been done towards understanding the insight of movement prediction for vehicular networks.

To estimate the position of a vehicle, several geographical localization based systems can be used, such as Global Positioning system (GPS), Global System for Mobile communication (GSM) and Triangulation. Yet, the accuracy of the GPS data is still an issue within VANETs based applications.

Most of the available prediction models are based on mobility patterns or probability modeling such as the Kalman filter and it's extensions. The Kalman filter (KF) require a linear system model and Gaussian noise to preform optimally. On the other hand, Extended Kalman filter (EKF) works on linearisation of the non-linear system using Taylor series expansion. As for the Unscent Kalman filter (UKF), it avoids linearisation by having a set of sigma points and the use of Unscent Transformation. Another model named Particle filter (PF), which uses a large number of particles with recursive Monte Carlo method. With the recent advances in the field of computing, it became easier to use complex mathematical models that led many researchers to use Bayesian models for tracking, navigation and prediction [1].

The main goal of this study is to evaluate the performance of movement prediction techniques in term of their efficiency and effectiveness for VANETs.

The remainder of this paper is organized as follows: In section II, we review existing vehicle movement prediction methods. Section III describes the Kalman filter based prediction technique for vehicular movement along with the general dynamic model of the vehicle. Then we describe the four prediction models in section IV to section VII. We report on the performance of the five models and discuss the results from simulation in Section IX, and Section X concludes our paper.

## II. Releated Work

In the literature, many works have been propsed for localization and tracking in mobile networks and wireless sensor networks [2]. Nevertheless, few of them were toward predicting vehicle movements in VANET. Movement prediction techniques can be divided into three categories: Deterministic, History-based, and Stochastic models [3]. Deterministic prediction, uses the vehicle kinetics to compute the future position of a vehicle. While history-based prediction, the model would learn from repeated movement patterns to predict future position. As for stochastic models, the focus is on correcting the prediction error using probabilities.

A survey on filtering techniques for vehicles tracking was presented by Floudas et.al. [4]. While, Burbey et.al. [5], introduced a location prediction method using prediction-by-partial match data compression PPM. They used IEEE 802.11 wireless access logs to obtain time and positioning data. Their results showed high success rate in predicting locations in first and third order models, given the time. Feng et.al. [6] used stochastic techniques for a future location prediction model using Kalman filter with real traffic traces and GPS information. Jaiswal et.al. [7] proposed a location prediction algorithm for vehicular movement using extended Kalman filter using city and highway mobility traces. They have compared EKF against the Kalman filter, which showed that EKF gives high accuracy in compared to KF location prediction. Wang et.al. [8] introduced a fixed-gain alpha-beta filter for location estimation in wireless sensor networks. They compared between alpha-beta and Kalman filter approach, which showed that alpha-beta method is done with reasonably good performance and lower computational complexity. On the study of particle filters, a survey on Sequential Monte carlo methods was studied in [9] and [10]. While, Gustafsson et.al. described a general framework for particle filters in positioning, navigation and target tracking in [11]. A particle

978-1-4673-8999-0/17/$31.00 ©2017 IEEE

filter based with adaptive markove chain and monte carlo method is introduced to improve the positiong accuracy of GPS receiver in [12].

In this paper, we focus on Stochastic techniques for Vehicular movement prediction including the Kalman filter, Extended Kalman filter, Unscent Kalman filter, Particle filter, and Alpha Beta Gamma filter.

## III. Kalman Filter based prediction model KF

A Kalman filter [13] is a set of mathematical equations that efficiently estimate the state of a linear system that minimizes the estimated error covariance to reach optimization. In VANET, the vehicle state vector $x$ is defined by the set of data $[p_x, v_x, a_x, p_y, v_y, a_y]$, which describes the vehicle's movement at time $t$. Where $p_x, v_x, a_x$ correspond to the latitude point, velocity and acceleration, and $p_y, v_y, a_y$ correspond to the longitude parameters of a vehicle. To measure the change in the state of the vehicle within $\Delta t$, we use the kinetic equation of motion.

$$x = x_0 + v.\Delta t + 0.5.a^2.\Delta t$$

$$v = v_0 + a.\Delta t$$

Where $x_0$ is the initial state of the vehicle, $v$ for the velocity, and $a$ is the acceleration of the vehicle (assuming constant acceleration at time t).

The Kalman filter works in two steps recursively at each time step $\Delta t$: Time update, which predict the next estimation of a current state, and Measurement update that adjusts the current state estimation with actual measurements at time $t$, they are defined as :

### A. Time update (Predict):

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k + w_{k-1} \tag{1}$$

$$P_k^- = AP_{k-1}A^T + Q \tag{2}$$

Where $w_k$ represent a normal probability distribution of the process white noise, $u$ is the control input vector (assumed zero as acceleration are considered states), $P_k^-$ is the priori estimate error covariance matrix, and $Q$ is the process noise covariance. To produce the priori estimated state $x_k^-$ at time interval $k$, the matrix $x$ is multiplied by the state model $A$.

$$x = \begin{bmatrix} p \\ v \\ a \end{bmatrix} A = \begin{bmatrix} 1 & t & 0.5t^2 \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix} Q = \begin{bmatrix} Q_x & 0 \\ 0 & Q_y \end{bmatrix}$$

### B. Measurement update (Correct):

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \tag{3}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \tag{4}$$

$$P_k = (I - K_k H)P_k^- \tag{5}$$

In the measurements update step, which is also referred to as the correction step, $\hat{x}_k$ is the posteriori state estimation at time step $k$, $P_{k-1}$ is the posteriori estimate error covariance matrix, $H$ is the measurement equation, $R$ is the measurements noise, which is in our paper equal to the GPS measurement noise, and $K_n$ is the Kalman Gain matrix. To update the estimated value of $x_k$, we use the GPS measurement $z_k$.

$$R = \begin{bmatrix} R_x & 0 \\ 0 & R_y \end{bmatrix} H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

The Kalman filter performance is dependent on how accurate is the state dynamics and the measurements of the GPS data, which are described as the error in process, measurements Q and R respectively [14]. In our comparison we treated them in ad hoc manner by tuning their values.

## IV. Extended Kalman Filter based prediction model EKF

The Kalman filter in the previous section works on estimating the state of a linear-time equation. Whereas the EKF try to approximate a nonlinear equation by linearization through Taylor series expansion [15]. EKF linearize the nonlinear system by partial differentiation using the Jacobian matrix to estimate the state of a system. To predict the next state of a system, a priori estimation $\hat{x}_k^-$ and the error covariance $P_k^-$ are calculated as follows:

$$\hat{x}_k^- = f(\hat{x}_{k-1}^-, u_k, w_k) \tag{6}$$

$$P_k^- = F_k P_{k-1} F_k^T + W_k Q_t W_k^T$$

Where $F_k$, $W_k$ are the Jacobian matrix of the nonlinear system $f(.)$ and $w$ is the partial differentiation of function $f(.)$ to the process noise (7). The covariance matrix of the state noise is represented by $Q_{k-1}$.

$$F_k = \frac{f(\hat{x}_{k-1}^-, 0, 0)}{x}, \quad W_k = \frac{f(\hat{x}_{k-1}^-, 0, 0)}{w} \tag{7}$$

and are initially assumed as

$$F = \begin{bmatrix} 1 & t & t^2/2 \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix} W = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

In the correction step, the Kalman gain $K_k$ is used to calculate the a posteriori estimate of the system dynamics and the error covariance $P_k$ as:

$$K_k = P_k^- H^T (HP_k^- H^T + V_k R V_k^T)^{-1}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - h(\hat{x}_k^-, v_k))$$

$$P_k = (I - K_k H)P_k^-$$

Similarly, the matrix $H$ is the Jacobian matrix of the function $h$ partial differentiation and $V$ represent the Jacobian matrix of the measurement noise $v$. The initial error covariance $P_0$ is set to a large value.

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} P_0 = \begin{bmatrix} 1000 & 0 \\ 0 & 1000 \end{bmatrix}$$

Where $K_k(z_k - h(\hat{x}_k^-, v_k))$ is the innovation equation that calculate the difference between predicted and real measurements.

## V. Unscent Kalman Filter based prediction model UKF

The Unscent Kalman filter [16] avoids to linearize around the mean as the EKF by having a set of sigma points represent the Gaussian random variable of the state vector. The UKF is based on Unscented Transformation (UT). It uses the nonlinear equation $f(.)$ (6) of the vehicle state vector $x$ with $n$ number of states to compute a set of sigma points matrix $X$ as follows:

$$\chi_0 = \hat{x}, \quad \lambda = \alpha^2(n+\kappa) - n \tag{8}$$
$$\chi^i = \hat{x} + (\sqrt{(n+\lambda)P_x})_i \quad i = 1, ...n$$
$$\chi^{i+n} = \hat{x} - (\sqrt{(n+\lambda)P_x})_{i-n} \quad i = n+1, .., 2n$$

Where $i$ is the $i^{th}$ column or row of the matrix square root of $\hat{x}$ and $P$ , which are the mean and covariance of the state vector, respectively. The symbol $\lambda$ is a scaling factor with influencing constants $\alpha$, $\kappa$ of how far the sigma points from the mean and is usually set to a very small value. Then transform each sigma point through the nonlinear function and compute the Gaussian of their transformation and weight. The weights are calculated as follows:

$$w_m^0 = \frac{\lambda}{n+\lambda} \tag{9}$$
$$w_c^0 = w_m^0 + (1 - \alpha^2 + \beta), \quad \beta = 2$$
$$w_m^i = w_c^i = \frac{1}{2(n+\lambda)} \quad i = 1, .., 2n$$

The Unscent Kalman filter model differs from EKF in the prediction by computing sigma point mean and covariance at each time step $t$. Each sigma point is initiated through the process model

$$\hat{\chi_t} = f(\chi_{t-1}, u_t)$$

The predicted mean and covariance is given by

$$\hat{x}_t^- = \sum_{i=0}^{2n} w_m^i \hat{\chi}_t^i$$

$$\hat{P_t}^- = \sum_{i=0}^{2n} w_c^i (\hat{\chi}_t^i - \hat{x_t}^-)(\hat{\chi}_t^i - \hat{x_t}^-)^T + R_t$$

As for the measurements update equations in the UKF, it is computed as follows:

$$\hat{Z}_t = h(\hat{X}_t), \quad \hat{z}_t = \sum_{i=0}^{2n} w_m^i \hat{Z}_t^i \tag{10}$$

$$S_t = \sum_{i=0}^{2n} w_c^i (\hat{Z}_t^i - \hat{z}_t)(\hat{Z}_t^i - \hat{z}_t)^T + Q_i$$

$$\hat{P}_t^{x,z} = \sum_{i=0}^{2n} w_c^i (\hat{X}_t^i - \hat{x}_t)(\hat{Z}_t^i - \hat{z}_t)^T$$

Where each point $\hat{Z}_t$ is instantiate through the observation model $h(.)$, $\hat{z}_t$ is the predicted observation, $S_t$ is the innovation covariance and the cross covariance matrix is $\hat{P}_t^{x,z}$. In the final step of updating and correcting the estimated mean and covariance, using the following equations

$$K_t = \hat{P}_t^{x,z} S_t^{-1}$$
$$\hat{x}_t = \hat{x_t}^- + K_t(z_t - \hat{z}_t)$$
$$P_t = \hat{P}_t - K_t S_t K_t^T$$

## VI. Alpha Beta Gamma Filter based prediction model $\alpha$-$\beta$-$\gamma$

All previous models have high computational costs, which could be an issue for some applications. As for fixed-gain Alpha beta gamma filters, this is not an issue since they only involve small computational cost [17]. The prediction equations for the alpha beta gamma filter with respect to position and velocity is defined as follows:

$$x_{t+1} = x_t + Tv_t + \frac{1}{2}Ta_t \tag{11}$$
$$v_{t+1} = v_t + Ta_t$$

and the kinematic variables are updated by weighting the innovation at each time step $t$

$$x_{t+1} = x_{t+1} + \alpha(z_t - x_{t+1}) \tag{12}$$
$$v_{t+1} = v_{t+1} + \frac{\beta}{T}(z_t - x_{t+1})$$
$$a_{t+1} = a_t + \frac{\gamma}{2T^2}(z_t - x_{t+1})$$

Where $z_t$ is the observation (GPS data) at time $t$. The bounds of $\alpha$, $\beta$ and $\gamma$ is constraint by

$$0 < \alpha < 2 \tag{13}$$
$$0 < \beta < 4 - 2\alpha$$
$$0 < \gamma < \frac{4\alpha\beta}{2-\alpha}$$

## VII. Particle Filter based prediction model PF

The evolution of particle filters in the research area started with a paper by Gordon et.al. [18]. Particle filters are sequential Monte Carlo methods based on the weight representation of probability densities of any given state model. Monte Carlo methods is a general class which convert closed form statistical quantities to distributed samples and using their average for estimation [19]. The distributed samples are referred to as particles. In the following we generally describe how the particle filter algorithm works.

Initially, we generate a set of random samples $(x_1^i : i = 1, .., N)$ from the PDF $p(x_0)$ and set the weight of each sample to $w_1^i = 1/N$. Where the equation of the PDF construction is obtained recursively as follows:

$$p(x_k|z_{k-1}) = \int \underbrace{p(x_k|x_{k-1})}_{\text{dynamic model}} \underbrace{p(x_{k-1}|z_{k-1})}_{\text{prior}} dx_{k-1}$$

At each time step $k + 1$, for each particle we sample the prior $x_k^i$ of PDF using the system model $x_k^i = f_{k-1}(x_{k-1}^i, wk - 1^i)$, Where $w^i$ is the PDF of the system noise $p(w)$.

$$p(x_k^i) = p(x_k^i | x_{k-1})$$

Using the measurement $y_k$, we can form the weight of particles based on a likelihood equation of each prior sample.

$$w_k^{*i} = w_k - 1^i p(z_t | x_t^i)$$

Then we normalize the weight by

$$w_k^i = \frac{w_k^{*i}}{\sum_{j=1}^{N} w_k^{*j}}$$

The posterior probability density is then calculated by

$$p(x_{0:k} | z_{1:k}) = \sum_{i=1}^{N} w_k^i \delta(x_{0:k} - x_{0:k}^i)$$

where $\delta(.)$ is the Dirac delta function. After a while the basic particle filter will suffer from $degeneracy$ or $sampleimpoverishment$. Meaning that all but few particles will have negligible weights. To solve this problem, researchers introduced resampling methods such as Sampling Importance Resampling(SIR). The basic idea of resampling technique is to replace the light-weighted sample with the high-weighted ones. This step can be done when needed either in predefined time step or by finding the effective number of particles [20] as

$$N_e f f = \frac{1}{\sum_{i=1}^{N} (w_k^i)^2}$$

In this case, the resampling is preformed when the number of effective particles is below the total number of particles. Some of the most commonly used resampling methods is Systematic resampling [21] among others.

## VIII. Performance Evaluation

The performance of the five different prediction techniques have been implemented in MATLAB on a Mac machine with 2.4 GHz Intel Core i7. In our simulation, each vehicle is equipped with a prediction model and a GPS measurement sensor. For the prediction of vehicle movement, we set the state of the vehicle dynamics to include position $(x_k, y_x)$, velocity and acceleration.

Using two different sets of mobility traces. The first set is a real vehicle mobility trace from Roma, Italy and the second set is generated mobility trace using SUMO simulator and OpenStreet Map (OSM).

- *SUMO Traces* Comparing the different prediction methods using generated mobility traces from SUMO and OpenStreetMap. A mobility trace of 100 vehicle randomly moving with speed between 0-20m/s , following the roads constraints of Ottawa city. Readings were taken every 1 seconds and includes the vehicle position $x$ and $y$ as well as its speed $s$ at time $t$.
- *real mobility traces* Simulation have been done in more challenging scenario, using real mobility traces from taxi cabs in Rome. Comparing the five different filters performance using dataset of mobility traces of taxi cabs in Rome, Italy [22].The dataset include the taxi GPS latitude and longitude, readings were taken every 15 seconds.

We evaluated the performance of the prediction models using the following metrics:

- *DE* compute the Euclidean distance error between measured and predicted location

$$D_{error} = \sqrt{(x_k - \hat{x_k}) + (y_k - \hat{y_k})} \tag{14}$$

- *RMSE* compute the square root of the mean square error between measured and predicted location.

$$RMSE = \sqrt{\frac{1}{N} \sum_{k=1}^{N} (x_k - \hat{x_k})} \tag{15}$$

In the following we illustrate our comparison results between KF, EKF, UKF, PF and ABG filter. The results show accuracy and robustness of the different methods.

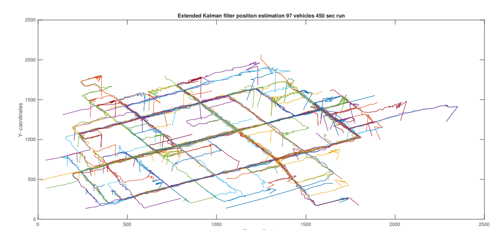


Fig. 1: Complete movement prediction of 100 vehicle in duration of 500 sec. for SUMO generated mobility trace

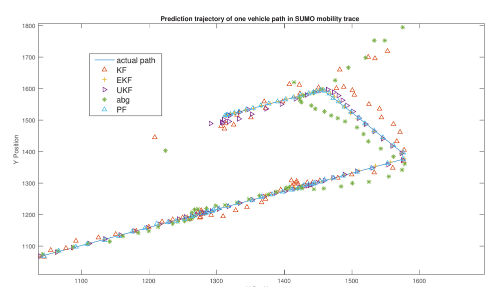


Fig. 2: True trajectory comparison with estimated trajectory of KF, EKF, UKF, abg , PF for SUMO trace

As seen in Figure 1 to 4, we evaluate the different models using a generated mobility trace from SUMO. We set the number of particles to be 300 for this comparison and the alpha beta gamma $\lambda$ value to be 0.01. In Figure 1, we show the complete EKF prediction path of 97 vehicles in Urban scenario from a generated mobility trace using SUMO. To clearly see the comparison of the different models in the previous scenario, a close comparison is illustrated in Figure 2 of a vehicle path. In which, we show the performance of the UKF and PF is better than both EKF, KF, ABG. This is due to the fact that PF and UKF uses set of samples and sigma points to estimate the movement of a vehicle. Even though, PF and UKF are computationally expansive, the accuracy of those models are higher than the others in nonlinear dynamic systems such as VANET. Another observation is that the prediction of KF and ABG report badly in specific points when the vehicle turn on path. This is due to that both KF and ABG estimation is mainly done by propagation a linear state using the motion model and to recuperate from it takes some time after couple iterations.

In Figure 3, a comparison on the Distance Error DE is presented, where again UKF and PF showed better result in compared to KF, EKF, ABG. It is noticed that PF has high distance error after almost half the way of the prediction. This is due to the problem introduced in Section VII, which is the degeneracy or sample impoverishment. In which all but few particles will have negligible weights and effect the performance of the filter. Thus, an efficient resampling strategy is required to better estimate the vehicle movement.

TABLE I: Average RMSE for sumo mobility trace

| KF | EKF | UKF | abg | PF |
|---|---|---|---|---|
| 83.8178 | 34.0501 | 34.0479 | 90.0924 | 2.2789 |

As for the average RMSE of all models is given by Table I, showing that EKF and KF perform worse than UKF and Particle filters. As for alpha beta gamma filter, it reports worse than all other filters. Nevertheless, Particle filters still require relatively higher computation cost than the other filters to estimate the movement of vehicle, which could be an issue for some VANET applications where speed and time is essential.

In Figure 4, we compared the total number of distance errors with a distance threshold $\theta$. As it shows that most of the prediction error is less that 10 meters. Which is still very high for a vehicle movement prediction and should be further studied to reduce the error.

TABLE II: Average RMSE for real taxi cabs mobility trace

| KF | EKF | UKF | abg | PF |
|---|---|---|---|---|
| 396.2993 | 133.0352 | 133.0331 | 2821.1033 | 12.1653 |

The result of the different prediction models with a more challenging scenario of real mobility traces in Figures 5 to 6. We ran the simulation multiple times over 200 taxi cabs. In Figure 5, the movement prediction of all five models for one taxi cab is presented, which shows that the PF preformed with high accuracy in compared to the others prediction. In which, Kalman filter reported the wors, because KF fail predict the quick changes in the taxi movement. To better show the difference of the five models, we show the distance error measurement in Figure 6.

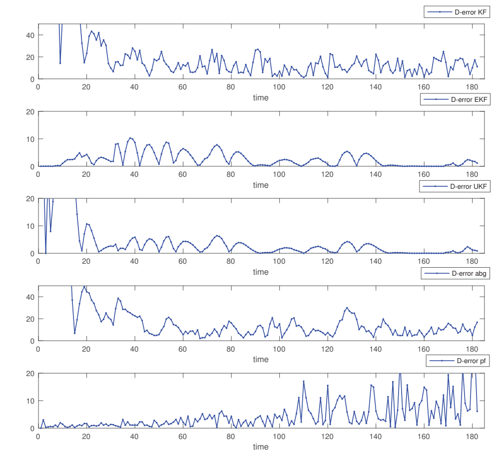


Fig. 3: Distance Error comparison of KF, EKF, UKF, abg, and PF for SUMO trace

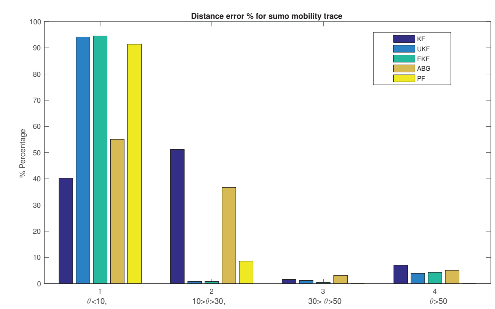


Fig. 4: Distance Error comparison for SUMO trace with respect to threshold $\theta$

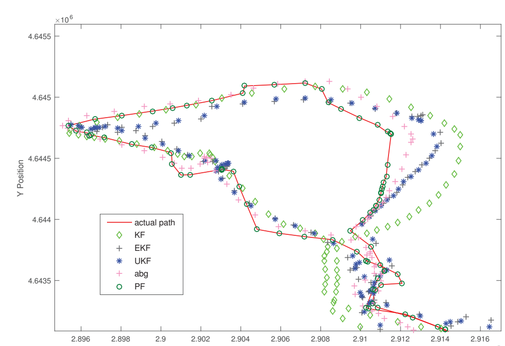


Fig. 5: Prediction of the vehicle's next position for mobility traces of taxi cabs in Rome, Italy

As seen Particle filter illustrate the lowest distance error because of the sampling technique, which randomly selected around the prior distribution. This technique will overcome the quick changes in any vehicle movement. As for the average RMSE of all models is given by Table II. One can also notice that the number of particles or samples used in the prediction can effect the outcome accuracy as shown in Figure 7. We used different number of particles (100, 300, 500) in predicting the movement of a vehicle in real mobility trace in the duration

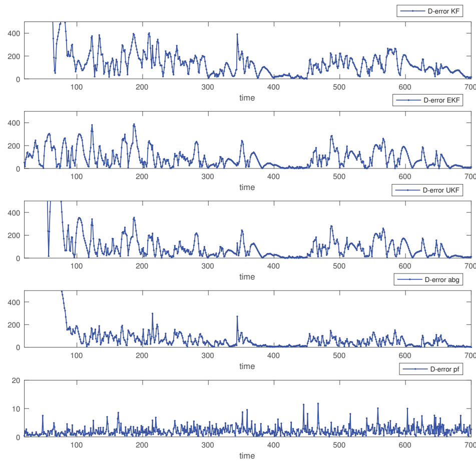


Fig. 6: Distance error DE for mobility traces of taxi cab in Rome, Italy

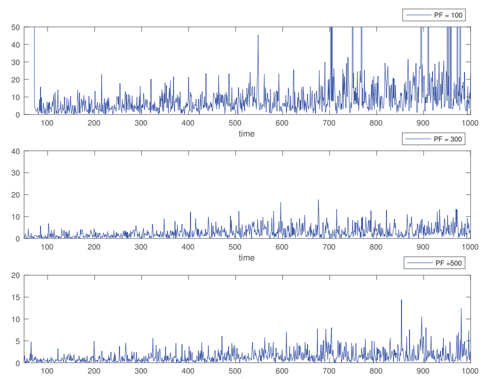


Fig. 7: Particle filter distance error DE comparison with different number of particles

of 16 minutes. In which we derive that the more number of particles assigned to the prediction, the more accurate results is produced.

## IX. Conclusion

This paper presents a performance evaluation of prediction techniques using KF, EKF, UKF, PF, and ABG for VANET. It has been shown through simulation results that the Particle filter outperforms Kalman filter, and Extended Kalman filter. Also, Unscent Kalman filter showed almost equal performance to the Particle filter. As for alpha-beta-gamma, it failed to accurately predict the vehicle movement in comparison to the other methods. Our comparison was based on different mobility traces of real taxi cabs and generated mobility from SUMO. It is also noteworthy that even though particle filters showed high accuracy in predicting the vehicle movement, it still requires higher computational cost that should be investigated further.

## Acknowledgment

This work is partially supported by NSERC, Canada Research Chairs Program, DIVA Strategic Research Network, NSERC-CREATE TRANSIT Program

## References

[1] Z. Chen. Bayesian filtering: from Kalman filters to particle filters, and beyond. *Statistics* 182 (1), 1-69

[2] D. Gavalas, C. Konstantopoulos, B. Mamalis, and G. Pantziou, (2010). Mobility prediction in mobile ad hoc networks. *In S. Pierre (Ed.), Next generation mobile networks and ubiquitous computing* (pp. 226—240). Hershey: IGI Global.

[3] J. Harri and C. Bonnet and F. Filali, The Challenges of Predicting Mobility, *Research Report RR-06-171*. (2007)

[4] N. Floudas, A. Polychronopoulos and A. Amditis, "A survey of filtering techniques for vehicle tracking by radar equipped automotive platforms," 2005 *7th International Conf. on Information Fusion*, 2005, pp. 8

[5] I. Burbey and T. L. Martin, Predicting future locations using prediction-by-partial-match. *In: Proc. MELT*, pp.1-6 ACM Press, (2008).

[6] H. Feng and C. Liu and Y. Shu and O. W. Yang, Location Prediction of Vehicles in VANETs Using A Kalman Filter, *Wirel. Pers. Commun.* 80, 2 2015), 543-559.

[7] R. K. Jaiswal and C. D. Jaidhar, Location prediction algorithm for a nonlinear vehicular movement in VANET using extended Kalman filter, *Wireless Netw* (2016). 1-16

[8] C. L. Wang, Y. S. Chiou and Y. S. Dai, An Adaptive Location Estimator Based on Alpha-Beta Filtering for Wireless Sensor Networks, *IEEE Wireless Communications and Networking Conf.*, 2007, pp. 3285-3290.

[9] J. Liu and R. Chen, Sequential Monte Carlo methods for dynamic systems, *Journal of the American Statistical Association*, vol. 93,1998.

[10] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, *IEEE Trans. on Signal Processing*, vol. 50, no. 2, pp. 174188, 2002.

[11] F. Gustafsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.J. Nordlund. Particle Filters for positioning, navigation, and tracking. *IEEE trans. on signal processing*. 2002. 50,2, 425-437.

[12] E. Wang, M. Cai and T. Pang, Improving GPS Positioning Accuracy Based on Particle Filter Algorithm. *Springer Berlin Heidelberg*, 139-148, 2013.

[13] R. E. Kalman, (1960). A new approach to linear filtering and prediction problems. *Trans. of the ASME Journal of Basic Engineering*, 82, 3545.

[14] V.A. Bavdekar and A.P. Deshpande and S.C Patwardhan. Identification of process and measurement noise covariance for state and parameter estimation using extended Kalman filter. *Journal of Process Control*, 21(4), 585601.(2011)

[15] Simon J. Julier ; Jeffrey K. Uhlmann; New extension of the Kalman filter to nonlinear systems. *Proc. SPIE 3068, Signal Processing, Sensor Fusion, and Target Recognition VI*, 182 (1997)

[16] SJ. Julier and J.K. Uhlmann, Unscented Filtering and Nonlinear Estimation, *Proc. of the IEEE* , Vol. 92, No. 3, pp.401-422, 2004.

[17] D. Tenne and T. Singh, Characterizing performance of Apla Beta Gamma filters. *IEEE Trans Aero Elec Sys.* 2002.

[18] N. J. Gordon, D. J. Salmond and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," in *IEE Proc. F - Radar and Signal Processing,* vol. 140, no. 2, pp. 107-113, 1993.

[19] S. Srkk. Bayesian Filtering and Smoothing. *Cambridge Univ. Press*, 2013.

[20] J. Liu and R. Chen. Blind deconvolution via sequential imputation *American Statistical Association* 90, 567-76.

[21] G. Kitagawa, Monte Carlo filter and smoother for non-Gaussian nonlinear state space models, J. Comput. Graph. Statist., 5(1), pp. 125, 1996.

[22] L. Bracciale, et. al. CRAWDAD dataset (v. 2014.07.17), downloaded from http://crawdad.org/roma/taxi/20140717 (2014).