An Improved Robust Low Cost Approach for Real Time Vehicle Positioning in a Smart City

Ikram Belhajem(✉), Yann Ben Maissa, and Ahmed Tamtaoui

Laboratory of Telecommunications, Networks and Service Systems,
National Institute of Posts and Telecommunications, Rabat, Morocco
{belhajem,benmaissa,tamtaoui}@inpt.ac.ma

Abstract. The Global Positioning System (GPS) aided low cost Dead Reckoning (DR) system can provide without interruption the vehicle position for efficient fleet management solutions in smart cities. The Extended Kalman Filter (EKF) is generally applied for data fusion using the sensor's measures and the GPS position as a helper.

However, the EKF depends on the vehicle dynamic variations and may quickly diverge during periods of GPS signal loss.

In this paper, we present a robust low cost approach using EKF and neural networks (NN) with Particle Swarm Optimization (PSO) to reliably estimate the real time vehicle position. While GPS signals are available, we train the NN with PSO on different dynamics and outage times to learn the position errors so we can correct the future EKF predictions during GPS signal outages. We obtain empirically an improvement of up to 94% over the simple EKF predictions in case of GPS failures.

Keywords: Data fusion · Extended kalman filter · Global positioning system · Intelligent transportation systems · Smart cities · Dead reckoning · Low cost · Neural networks · Particle swarm optimization

1 Introduction

Context. In a smart city, traditional infrastructures are merged with information and communication technologies to face problems resulting from the rapid urban population growth, for a sustainable economic development and a high quality of life [1]. Based on wireless technologies, smart cities can establish intelligent transportation systems (ITS) capable of carrying out effective and energy sufficient transport services at an inexpensive cost. There are several applications of ITS including fleet management solutions that can achieve the route optimization and reduce the fuel consumption by giving a real time situational awareness of traffic conditions and the data for a flow of vehicles in the roads

This work was financially supported by the National Center of Scientific and Technical Research (CNRST), Morocco.

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2017
L.A. Maglaras et al. (Eds.): INISCOM 2016, LNICST 188, pp. 77–89, 2017.
DOI: 10.1007/978-3-319-52569-3_7

(e.g., ambulances and police vehicles). This will certainly require a real time visibility of an accurate vehicle position to improve the driver safety and comfort.

Positioning systems often rely on the Global Positioning System (GPS) which provides, based on satellite signals received, the location, altitude and velocity at a low frequency. Unfortunately, GPS receivers perform badly within tall urban buildings or beneath dense foliage because of signal masking and multipath phenomenon (i.e., multiple copies of GPS signal reach a receiver's antenna by two or more different paths). The combination of GPS and Inertial Navigation Systems (INS) (i.e., autonomous systems that provide positioning, velocity, and attitude information at high update rates) can ensure a continuous position estimate and overcome the limitations of using each sensor individually [7,9,12]. Another possibility is the GPS enhanced with INS based on microelectromechanical systems (MEMS) [2] since the high performance inertial sensors are very expensive. The MEMS technology offers cost reduction coupled to small size and lower power consumption advantages.

Practically, multisensor data fusion is performed using the Kalman filter (KF) which is an optimal state estimator start from noisy and erroneous observations [3]. The application of the KF is restricted to linear systems; therefore the Extended Kalman Filter (EKF) is adopted through a first-order linearization procedure for non-linear systems.

Problem. The INS impose restrictions on the environments where they are implemented because of their computing *complexity* in addition to their high *cost* [2]. Furthermore, MEMS-based INS suffer from a rapid accumulation of errors when operating in a stand-alone mode during GPS outages.

The KF performance depends on how the stochastic models of the sensors are accurate, also it requires a priori information of system noises and measurement errors. For those reasons, the KF predicted position tends to quickly *diverge* when GPS outage occurs.

Contribution. In this paper, we suggest a new robust low cost approach using EKF and Evolutionary Machine Learning in order to yield an optimal real time vehicle positioning in a smart city. The sensors used are GPS enhanced with low cost Dead Reckoning (DR) system (composed of only an odometer and a gyrometer to measure angular velocity and displacement of a vehicle) which is easy to use and keep the calculations simple. During GPS signal presence, the EKF estimates the position while the neural networks (NN) learn the position errors. The NN compensate the additional EKF position errors when no GPS information exists, allowing the system to correct the EKF estimations and preventing it from divergence. The training phase of NN is achieved through evolutionary learning algorithms notably the Particle Swarm Optimization (PSO).

Contents. The remainder of this paper is organized as follows: In Sect. 2, we discuss some relevant works on this topic. We present the essential background on EKF, NN and PSO in Sect. 3. Then, we deal with the formulation of our proposed approach in Sect. 4. In Sect. 5, we detail the experimental results.

2 Related Works

In literature, several works related to vehicle positioning suggest the integration of GPS and DR sensors like odometers and gyrometers [4,6]. The odometer provides the distance travelled by the vehicle while the gyrometer measures the angular velocity. Both of these sensors are subject to time growing errors (e.g., bias drift and scale factor change) in spite of their autonomy and independency of any signal blockage. The combined system GPS/DR helps then to keep down the odometer and gyrometer errors when GPS is available; conversely, in urban areas where GPS signals are frequently blocked, the DR ensures a continuous positioning.

Domenico et al. [4] propose a hybrid strategy using the EKF for the GPS, odometer and gyrometer data fusion when GPS signal is available; inversely the algorithm switches to an open-loop model-based estimation when it is not available because its loss for long periods reduces the EKF accuracy and even causes its divergence.

In their work, Georgy et al. [5] compare the performance of the KF and the Particle Filter (PF) used as vehicle state estimators combining GPS with the vehicle odometer and low cost MEMS-based inertial sensors; they conclude according to empirical results that the PF outperforms KF in case of GPS masks. Nevertheless, the increased power of PF may require a large number of particles; this comes at the cost of higher computational complexity.

Ismaeel et al. [7] propose the use of multilayer feed-forward NN with Backpropagation Learning Algorithm (BPLA) to integrate data from INS and GPS for vehicular positioning and velocity information. Besides, the wavelet multiresolution analysis is used to compare the INS and GPS position outputs at different resolution levels before processing them by the NN.

In [8], Hasan et al. introduce an adaptive neuro-fuzzy inference system (ANFIS) using PSO to combine data from MEMS grade INS and GPS for a reliable navigation solution. This network is trained during the availability of GPS signal so to estimate the INS position errors during GPS signal blockage.

Malleswaran et al. [9] present a performance analysis of GPS/INS integration based NN trained with weight optimization techniques namely the BPLA, Genetic Algorithm (GA), and PSO. Experimental tests indicate that the PSO training algorithm provides superior learning capability and is well adapted for intelligent navigation systems.

Unluckily, replacing the EKF completely by NN can be a non-optimal solution because their estimation quality can only be guaranteed if the trained data are sufficient. Belhajem et al. claim that the use of NN only [10] or coupled to Autoregressive Integrated Moving Average (ARIMA) models [11] can bridge the gap in the EKF prediction mode based on data from GPS and a low cost DR system. While GPS is available, the NN are trained on different samples to learn the position errors, so they can correct the additional EKF drifts during GPS signal loss.

3 Background

In this section, we cover the background of the EKF that estimates the vehicle position and the NN used for learning the EKF predicted position errors. Also, we describe the principle of the PSO metaheuristic.

3.1 Extended Kalman Filter

The EKF is a non-linear version of the KF that linearizes the process and measurement models about the current mean and covariance. The filter is a set of mathematical equations which uses the process model to estimate the current state of a system, then a correction of this estimate is performed using any available sensor measurements.

3.1.1 Modeling

Let us consider a car-like model of a front-wheel drive vehicle. The origin M of the body frame (rigidly attached to the vehicle) is located midway the rear axle while the x-axis is aligned with the vehicle longitudinal axis (see Fig. 1). For the vehicle dynamics analysis, the North-East-Down frame known also as a navigation frame is used; so any movement related to the body frame have to be converted to the navigation frame.

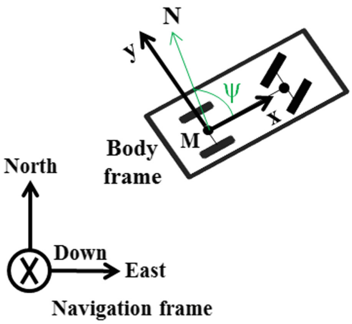

Fig. 1. Vehicle kinematic model

The kinematic equations mentioned below describe the vehicle position denoted by (N,E,ψ) where N and E denote the north and east components and ψ represents the heading [6]:

$$\begin{cases} N_{k+1} = N_k + ds_{k+1}.sinc(\frac{d\psi_{k+1}}{2}).cos(\psi_{k+1} + \frac{d\psi_{k+1}}{2}) \\ \qquad\quad -d\psi_{k+1}.(D_x.sin(\psi_k) + D_y.cos(\psi_k)). \\ E_{k+1} = E_k + ds_{k+1}.sinc(\frac{d\psi_{k+1}}{2}).sin(\psi_{k+1} + \frac{d\psi_{k+1}}{2}) \\ \qquad\quad +d\psi_{k+1}.(D_x.cos(\psi_k) - D_y.sin(\psi_k)). \\ \psi_{k+1} = \psi_k + d\psi_{k+1}. \end{cases} \tag{1}$$

where

- ds_{k+1} is the distance traveled by the vehicle between k and k + 1;
- $d\psi_{k+1}$ represents the heading variation corresponding to the angular velocity between k and k + 1;
- D_x and D_y are the distances in the body frame between the GPS antenna and the middle of the rear axle.

3.1.2 Prediction and Correction Phases

Figure 2 shows a scheme for the KF model. In our case, the state vector at time epoch k is $X_k = (N_k, E_k, \psi_k)$ and the measurement vector is $Z_k = (N_{GPS_k}, E_{GPS_k})$. Before the estimation process starts, values for the initial state $\hat{X}_0^+$ and the corresponding error covariance P_0^+ are assumed to be known. The prediction mode starts by projecting the state and error covariance ahead to estimate $\hat{X}_k^-$ and P_k^-. When new GPS measurements are available at time epoch k, the filter starts the update mode. At this stage, the Kalman gain K_k, updated state $\hat{X}_k^+$ and error covariance P_k^+ are computed.

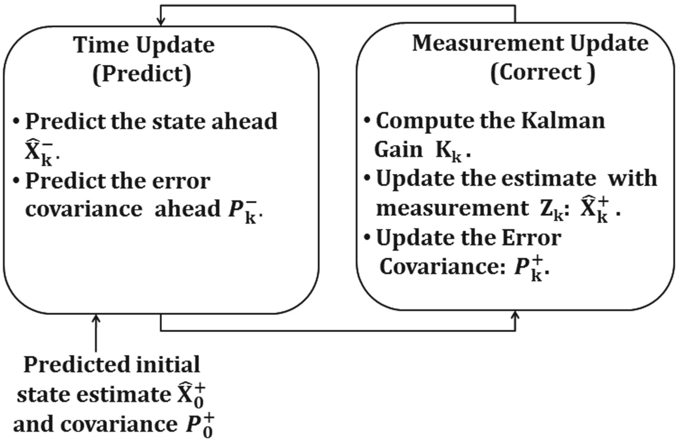

Fig. 2. Kalman filter model

3.2 Neural Networks

NN are a subset of Machine Learning methods acting as massively parallel distributed processors that have a natural propensity for storing experiential knowledge and making it available for use [13]. Inspired by the structure and functions of the human brain, NN are adaptive models that can map input patterns to output patterns without knowing the mathematical process involved. NN are composed of smaller units called neurons interconnected through synaptic weights.

The way a neural network links the input data into output data is defined by its architecture that refers to the arrangement of neurons into layers and the connection strengths within and between these layers [14]. In many practical applications, the most used model is the multilayer feed-forward NN in which all signals flow in one direction; from the input layer to the output layer passing by one or more hidden layers. In order to train the NN, weights of each unit are adjusted according to learning rules defined by weight optimization algorithms like PSO.

3.3 Particle Swarm Optimization

PSO is a robust stochastic evolutionary optimization technique developed by Kennedy and Eberhart [15]. The fundamental PSO inspiration is the social behavior of animals namely the movement and intelligence of swarms looking for the most fertile feeding location. In PSO, each problem to address is represented by a swarm of particles considered as candidate solutions which explore the search space looking for the best solution.

Since the strengths of the PSO include ease of implementation, computational efficiency, and fast convergence [16], it is used to optimize the NN connection weights during the training phase. For this, an initial swarm is generated randomly and the particles are the NN weights to learn. Each particle i moving around the search space is charcterized by a position vector x_i, a velocity vector v_i, and a position at which the best fitness $pbest_i$ is achieved by the particle. Besides, the global best position $gbest_i$ represents the position yielding the lowest error among all the $pbest_i$. At each iteration, the particles of the swarm are updated according to the following equations:

$$v_i(k+1) = wv_i(k) + c_1 r_1(k)(pbest_i - x_i(k)) + c_2 r_2(k)(gbest_i - x_i(k)). \quad (2)$$

$$x_i(k+1) = x_k(t) + v_i(k+1). \quad (3)$$

where c_1 and c_2 are the acceleration constants, $r_1(k)$ and $r_2(k)$ are random numbers uniformly distributed in [0,1], and w is the inertia weight employed to control the impact of the previous history of velocities on the current one. The update process is repeated until a maximum number of iterations is reached or an acceptable $gbest$ is achieved [16].

4 Formulation of the Hybrid EKF/NN Approach

In this section, we present a possible vehicle prototype and the formulation of our suggested approach in both the training and prediction phases. The prototype is not yet implemented, however we conducted extensive simulations on the Institut Pascal Data Sets [17] that were collected using VIPALAB, a platform equiped with multiple sensors. We have improvements over the EKF solution between 54 % and 94 %.

4.1 Possible Vehicle Prototype

Figure 3 illustrates the positions of the GPS and the odometer/gyrometer sensors in our possible vehicle prototype; each one of them is coupled to an Arduino nano and a Xbee module. The Arduino nano is dedicated for the treatement while Xbee module ensures a Zigbee communication for the wireless sensor networks. For the data treatment, a Raspberry with Xbee communication module is mounted on the car's dashboard. Our vision then is to support a real implementation of this prototype.

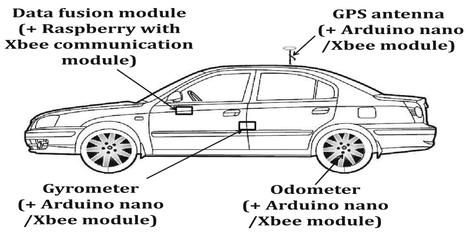

Fig. 3. Wireless sensor network

4.2 EKF/NN Combination

Practically, the EKF receives the speed V_{odom} and the heading ψ_{gyro}; then computes the vehicle predicted position (N_{pred},E_{pred}). When new GPS measurements arrive, the EKF updates the predicted position (N_{cor},E_{cor}) as presented in Fig. 4. However, when no GPS signal exists, the NN trained with PSO compensate the additional EKF predicted position errors.

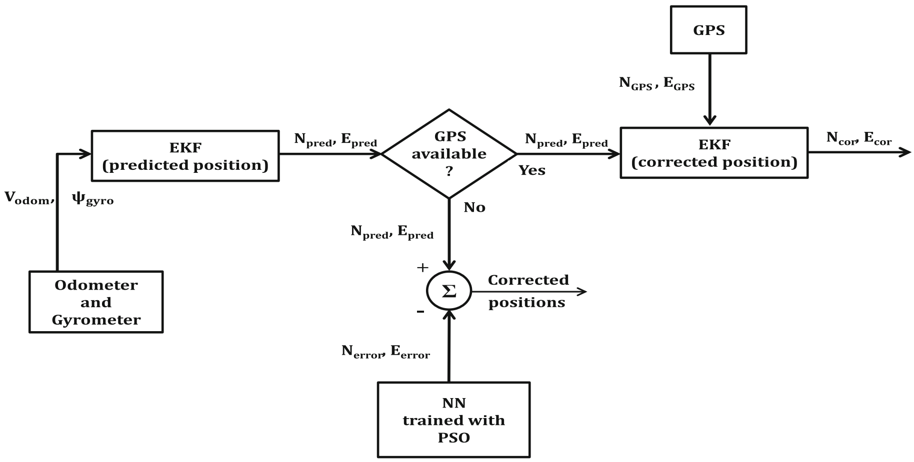

Fig. 4. EKF/NN combination

4.2.1 Training Phase

In general, the odometer bias or gyrometer drift consists of deterministic and stochastic parts. The former can be removed by calibration procedure while the latter is not easy to handle due to its random nature. Accordingly, the EKF performance depends on how the sensor components are correctly modeled; though a perfect tuning of the filter is rarely achieved since vehicle dynamic variations and environment changes occur oftenly. As a consequence, the EKF performs badly during GPS signal blockage which may result in its divergence.

To circumvent the EKF deficiencies, NN are a natural choice that require no calibration or modeling procedures. The networks are used to estimate the time-correlated position errors during the EKF prediction phase when no GPS signal is available. Two three-layer feed-forward NN are trained on different dynamics using PSO, so they can help to predict the north and east error drifts.

To fully represent the vehicle dynamics, the NN inputs consist of vehicle velocity, heading angle and time elapsed since last GPS measurement. The networks outputs are north and east errors which are compared to desired position errors. Figure 5 shows the north and east networks architecture.

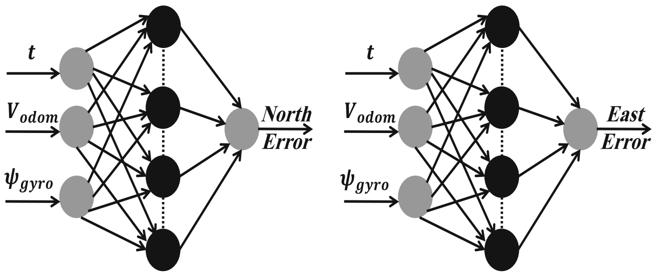

Fig. 5. North and east networks architecture

For training the NN, the target values used are computed as a difference between positions provided by two parallel EKFs. One filter provides a reference vehicle position while the other gives a predicted one by removing intentionally the GPS signals [12]. It should be noted that the training procedure presented in Fig. 6 is executed at the GPS sampling rate.

4.2.2 Prediction Stage

After training on different dynamics and outage times, the networks are used in prediction mode to help compensate for real GPS outages. The inputs are sent to the networks which provide estimates for the position errors along the north and the east directions. Since the EKF predictions without GPS measurements update contain errors, they are compensated by the networks outputs to form the corrected positions. The testing procedure is shown in Fig. 7.

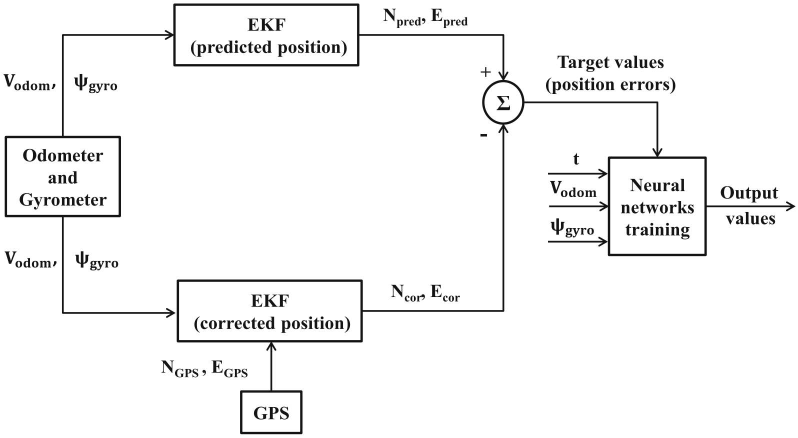

Fig. 6. NN training phase

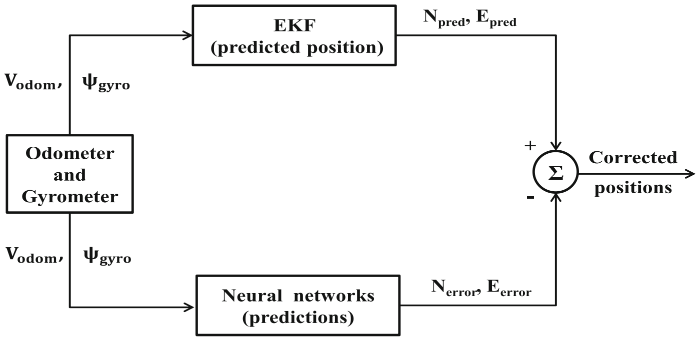

Fig. 7. NN testing phase

5 Experimental Tests and Results

In this section, we present the test vehicle prototype and the simulation results of our suggested hybrid approach.

5.1 Test Vehicle Prototype

The performance of the proposed hybrid technique was examined with the Institut Pascal Data Sets [17]. The field test data were collected using VIPALAB, a platform equiped with multiple sensors. In our case, the test system comprises three sets of an uBlox-6T-0-001 GPS receiver, an odometer and a Melexis MLX90609-N2 gyrometer. GPS values were collected at the frequency rate of 1 Hz while the odometer/gyrometer data at 50 Hz. The road test trajectory used is CEZEAUX-Heko (given in Fig. 8) which spans over a distance of 4.2 km during 28 min.

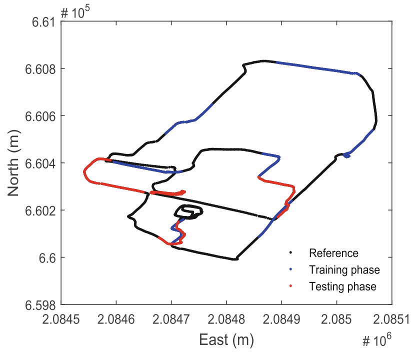

Fig. 8. Field test trajectory

5.2 Simulation Results

To examine the performance of the proposed approach, the field test data were divided into two parts. During the first 19 min, a total of seven GPS simulated outages (given in Fig. 8) are used to train the networks. Each outage lasts 60 s to leave the EKF position errors enough time to diverge. For the last 9 min, the NN run in the testing mode to generate predictions of the position drifts.

The NN are trained using batch-incremental approach. For every outage, a set of new inputs/targets are presented to train the NN using PSO to learn the network weight parameters. The objective is to reduce the mean squared error (MSE) between the networks outputs and the desired values. In our case study, the north and east networks architecture chosen emprically consist of 3 inputs, 5 hidden neurons and one output while the training goal is to reach MSE less than 2.10^{-4} m^2 given the real time constraints. Figure 9 shows the results of the training outages while the time intervals between them are masked. It appears clearly that the outputs of north and east networks are very close to the target values.

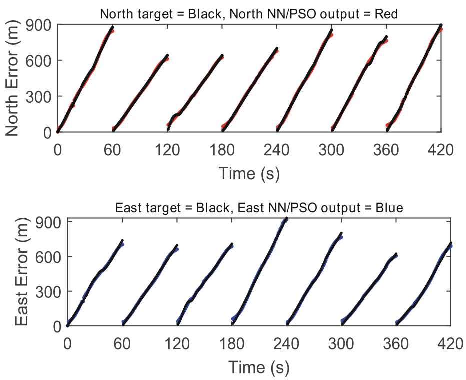

Fig. 9. North and east networks training results using PSO

To investigate the performance of this hybrid method, the GPS data were intentionally removed since there were no natural GPS outages in the field test. The networks can then be used to provide predictions of position drifts to correct the EKF predictions. For this purpose, they generate outputs based on dynamic inputs and latest estimated weight parameters. Relatively, results of the GPS test outages (presented in Fig. 8), with period lengths of 90 and 60 s, are given in Fig. 10.

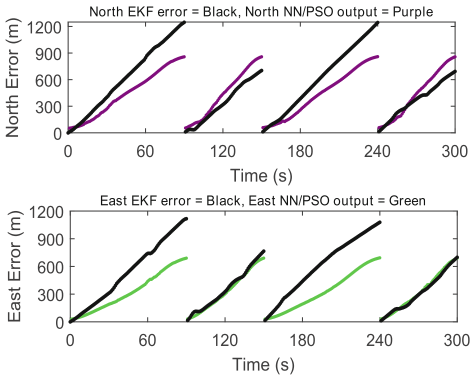

Fig. 10. North and east networks testing results using PSO

To compare the estimated position by our proposed approach and the one of EKF, two different evaluation indicators are calculated for each outage: the root mean square error (RMSE) and the mean absolute error (MAE). They are expressed by the following formulas:

$$RMSE = \sqrt{\frac{1}{n}\sum_{k=1}^{n}(A_k - F_k)^2}. \tag{4}$$

$$MAE = \frac{1}{n}\sum_{k=1}^{n}\left|A_k - F_k\right|. \tag{5}$$

where A_k is the actual EKF updated position by GPS measurements at time epoch k, F_k is either the EKF predicted position or the EKF corrected position by NN while n is the total number of predictions. These results are listed in Tables 1 and 2 for north and east position components. By combining EKF and NN together, the results show a significant decrease in RMSE and MAE over the EKF method. This hybrid approach enhances the vehicle position accuracy over the EKF predictions during GPS outages.

Table 1. GPS test outages north improvement

Outages	EKF		Our approach (EKF/NN)		Improvement (%)	
	RMSE (m)	MAE (m)	RMSE (m)	MAE (m)	RMSE (m)	MAE (m)
Outage 1	732.47	630.57	218.32	190.20	70	69
Outage 2	408.82	343.18	106.51	96.38	73	71
Outage 3	715.54	613.58	203.83	176.78	71	71
Outage 4	418.99	372.16	104.37	85.69	75	76

Table 2. GPS test outages east improvement

Outages	EKF		Our approach (EKF/NN)		Improvement (%)	
	RMSE (m)	MAE (m)	RMSE (m)	MAE (m)	RMSE (m)	MAE (m)
Outage 1	640.44	545.17	251.99	217.47	60	60
Outage 2	419.84	359.72	29.88	25.63	92	92
Outage 3	681.06	606.22	296.68	275.06	56	54
Outage 4	377.17	319.96	21.83	18.91	94	94

6 Conclusion

In this paper, we present an improved *robust* approach to estimate the real time vehicle positioning required by various fleet management applications in the context of a *smart city*. We propose the combination of EKF and NN based on PSO to fuse data coming from a GPS and *low cost* DR integrated sensors. After being trained on different dynamics and outage times, the NN are used to correct the EKF position errors when no GPS signal is detected. Experimental results with field test data demonstrate the ability of feed-forward NN trained with PSO to learn and make reasonable predictions of EKF drifts during different GPS blockage periods.

Future Work
Empirical results with simulated GPS outages showed very promising progress. Nonetheless, the GPS quality degradation in real GPS outages is more complex to handle. Further investigation is then needed to test and improve this hybrid solution during real GPS outages, that is why we intend to implement our vehicle prototype in future related works.

References

1. Caragliu, A., Del Bo, C., Nijkamp, P.: Smart cities in Europe. J. Urban Technol. **18**, 65–82 (2011). Routledge, England
2. Noureldin, A., Karamat, T.B., Eberts, M.D., El-Shafie, A.: Performance enhancement of MEMS-based INS/GPS integration for low-cost navigation applications. IEEE Trans. Veh. Technol. **58**, 1077–1096 (2009). IEEE

3. Grewal, M.S.: Kalman filtering. In: Lovric, M. (ed.) International Encyclopedia of Statistical Science, pp. 705–708. Springer, Heidelberg (2011)
4. Domenico, D.D., Fiengo, G., Glielmo, L.: On-board prototype of a vehicle localization system. In: 2007 IEEE International Conference on Control Applications, pp. 438–443. IEEE, USA (2007)
5. Georgy, J., Iqbal, U., Noureldin, A.: Quantitative comparison between Kalman filter and Particle filter for low cost INS/GPS integration. In: Proceedings of the 6th International Symposium on Mechatronics and Its Applications (ISMA 2009), pp. 1–7. IEEE, USA (2009)
6. Lucet, E., Betaille, D., Donnay Fleury, N., Ortiz, M., Salle, D., Canou, J.: Real-time 2D localization of a car-like mobile robot using dead reckoning and GPS, with satellite masking prediction. In: Accurate localization for land transportation Workshop, pp. 15–18. France (2009)
7. Ismaeel, S.A., Hasan, A.M.: GPS/INS system integratin based on neuro-wavelet techniques. In: Proceedings of the 2006 International Conference on Artificial Intelligence (ICAI 2006), vol. 2, pp. 446–452. CSREA Press, Las Vegas (2006)
8. Hasan, A.M., Samsudin, K., Ramli, A.R.: Optimizing of ANFIS for estimating INS error during GPS outages. J. Chin. Inst. Eng. **34**, 967–982 (2011). Taylor & Francis, London
9. Malleswaran, M., Vaidehi, V., Sivasankari, N.: A novel approach to the integration of GPS and INS using recurrent neural networks with evolutionary optimization techniques. In: Aerospace Science and Technology, vol. 32, pp. 169–179. Elsevier B.V., Netherlands (2014)
10. Belhajem, I., Ben Maissa, Y., Tamtaoui, A.: A hybrid low cost approach using Extended Kalman filter and neural networks for real time positioning. In: 2016 International Conference on Information Technology for Organizations Development (IT4OD), pp. 1–5. IEEE (2016)
11. Belhajem, I., Ben Maissa, Y., Tamtaoui, A.: A hybrid machine learning based low cost approach for real time vehicle position estimation in a smart city. In: El-Azouzi, R., Menasché, D.S., Sabir, E., Pellegrini, F.D., Benjillali, M. (eds.) Advances in Ubiquitous Networking 2. LNEE, vol. 397, pp. 559–572. Springer, Heidelberg (2017). doi:10.1007/978-981-10-1627-1_44
12. Goodall, C., Syed, Z., El-Sheimy, N.: Improving INS/GPS navigation accuracy through compensation of Kalman filter errors. In: Proceedings of the 64th IEEE Vehicular Technology Conference, VTC Fall 2006, pp. 1–5. IEEE (2006)
13. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall PTR, USA (1994)
14. Yegnanarayana, B.: Artificial Neural Networks. Prentice-Hall of India Pvt. Ltd., New Delhi (2004)
15. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks, pp. 1942–1948. IEEE, USA (1995)
16. Kulkarni, R.V., Venayagamoorthy, G.K.: Particle swarm optimization in wireless-sensor networks a brief survey. IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.) **41**, 262–267 (2011). IEEE Press, Piscataway, NJ, USA
17. Korrapati, H., Courbon, J., Alizon, S., Marmoiton, F.: The Institut Pascal data sets: un jeu de données en extérieur, multicapteurs et datées avec réalité terrain, données d'étalonnage et outils logiciels. In: ORASIS 2013, Congrès des jeunes chercheurs en vision par ordinateur. France (2013)