Accurate Vehicle Position Estimation Using a Kalman Filter and Neural Network-based Approach

Stanley Baek, Chang Liu, Paul Watta, and Yi Lu Murphey
Department of Electrical and Computer Engineering
University of Michigan - Dearborn
Dearborn, MI 48128

{stanbaek, liuchan, watta, yilu}@umich.edu

***Abstract*—Accurate detection of vehicle position plays an important role in many intelligent transportation systems, especially vehicle-to-vehicle applications. In this paper, we propose an Extended Kalman Filter (EKF) based method to detect Global Positioning System (GPS) errors for such vehicle-based applications. A machine learning methodology is presented for Kalman filter parameter tuning with application to GPS error correction in vehicle positioning. We also present a model free neural network that is trained on past vehicle GPS trajectories to predict the current vehicle position. Experimental results on real-world data show that the proposed system is effective for detecting and reducing GPS errors. The machine learning algorithm for EKF parameter tuning can be implemented through in-vehicle learning, and the proposed GPS error detection method can be implemented for in-vehicle applications.**

Keywords—Kalman filter; Global Positioning System (GPS); GPS error；online detection; neural network

I. Introduction

Accurate detection of vehicle position plays an important role in many intelligent transportation systems (ITS), especially in applications involving vehicle-to-vehicle (V2V), vehicle-to-pedestrian (V2P) and vehicle-to-other vulnerable road users (V2X), such as pre-crash detection, collision avoidance and autonomous driving. V2V, V2P and V2X communications all require a vehicle on-board communication unit, such as dedicated short-range radio communication (DSRC) devices or cellular networks, or other means of communication, to transmit messages about a vehicle's position, speed, heading, brake status, and other information to other vehicles (and receive such information, as well). According to the National Highway Traffic Safety Administration, V2V applications have the potential to address 80% of unimpaired crashes. However, most GPS devices used in such V2V systems are low-cost grade, and therefore, prone to errors resulting from a number of sources, such as receiver error, satellite error, multi-path error, and random error. Studies have shown that typical GPS errors range between 1 - 10 meters [1, 17, 18], which is not acceptable in applications related to driving safety systems.

GPS error is caused by a number of factors, including: time delays (troposphere and ionosphere), orbital errors due to inaccuracies in satellite position, multipath signal errors resulting from the GPS signal reflecting off structures in the environment, such as buildings and other vehicles, and timing errors due to the receiver clock inaccuracies [1, 3, 17].

A large amount of research has been conducted in correcting GPS errors. One type of approach is to use other sensors in the vehicle or roadside units as reference to estimate vehicle true position and compare it with the GPS measured position [17, 19]. In [17], Bhawiyuga, Nguyen, and Jeong presented a vehicular positioning accuracy enhancement (VITAE) algorithm that reduces the vehicle position error by taking into account GPS estimates, as well as estimates of nearby vehicles using a vehicular radar sensor. They first used a greedy data-association (GDA) method to find a minimal-weight matching between a GPS estimate and a sensing estimate based on a Euclidian distance. The algorithm then constructs two polygons of position estimates: a GPS polygon and a radar sensing polygon. The GPS polygon is constructed by connecting the GPS measures for remote vehicles provided by V2V communications. The sensing polygon is constructed by connecting the relative locations of remote vehicles provided by the radar sensor mounted on the host vehicle. The position of the host vehicle is adjusted by the difference between the mass center of the GPS polygon and that of the sensing polygon.

The most popular approach to correcting GPS errors is to formulate a system model of the vehicle dynamics, which involves not only GPS position, but also velocity and heading, and then use a Kalman Filter (KF) to smooth the trajectory [1, 6, 8, 9, 11, 12, 15, 16].

Xu et al used an improved Kalman filter approach to correct GPS errors in map-matching a vehicle location to the underlying road map [1]. Their method consists of a Kalman filter with a novel method to minimize the biased error of GPS after the vehicles makes a turn. The authors used real-world driving trips to evaluate their method. The result shows that it handles the biased error and the random error of the GPS signals reasonably well in both the along-road and cross-road directions. The authors showed that the raw GPS data along-track have a mean error of 21 meters, and the corrected GPS data along the same track have a mean error of 1.6 m.

Considerable research has been conducted on selecting proper Kalman Filter tuning parameters. Although the Kalman filter and the Extended Kalman Filter (EKF) are well established techniques for state estimation, the choice of the filter tuning parameters still poses a major challenge [11, 12, 16]. Saha et al. recently presented two new metrics for determining the filter tuning parameters on the basis of the innovation covariance [11]. Among all the tuning parameters, they identified the tuning of the process noise covariance matrix **Q** as being the most critical,

978-1-5386-2726-6/17/$31.00 ©2017 IEEE

since all the model uncertainties and inaccuracies as well as the noise affecting the process are incorporated quantitatively into $\mathbf{Q}$. They presented a metric based offline method usable for predicting the actual filter RMSE performance for a particular application. The metric can be used to find the proper combination of filter tuning parameters.

In this paper we present a machine learning methodology for Kalman filter parameter tuning with application to GPS error correction in vehicle positioning, and a model free neural network that is trained on past vehicle GPS trajectories to predict the current vehicle position. This paper is organized as follows. Section II gives an overview of the EKF approach for detecting vehicle positioning error. Section III presents the machine learning approaches we proposed for vehicle positioning error detection. Section IV presents experimental results conducted on real world driving trips and Section V concludes the paper.

II. Overview of the extended Kalman Filter

The displacement of a ground vehicle during the duration of time, ΔT, can be written as

$$s_x = \Delta T\, v\cos\theta + \frac{1}{2}\Delta T^2 a\cos\theta$$

$$s_y = \Delta T\, v\sin\theta + \frac{1}{2}\Delta T^2 a\sin\theta$$

where s_x and s_y are, respectively, the lateral and longitudinal displacement of the vehicle, v is the speed of the vehicle, θ is the heading of the vehicle, and a is the constant acceleration of the vehicle during ΔT, the sampling step size. In the discrete-time domain with $s_x = x_{k+1} - x_k$ and $s_y = y_{k+1} - y_k$, we can rewrite these equations as

$$x_{k+1} = x_k + \Delta T\, v_k\cos\theta_k + \frac{1}{2}\Delta T^2 a_k\cos\theta_k \tag{1}$$

$$y_{k+1} = y_k + \Delta T\, v_k\sin\theta_k + \frac{1}{2}\Delta T^2 a_k\sin\theta_k \tag{2}$$

$$v_{k+1} = v_k + \Delta T a_k \tag{3}$$

$$\theta_{k+1} = \theta_k + \Delta T\omega_k \tag{4}$$

where x_k and y_k are, respectively, the lateral and longitudinal position of the vehicle at time k, x_{k+1} and y_{k+1} are, respectively, the lateral and longitudinal *predicted* position of the vehicle at time $k+1$ based on the position (x_k, y_k), speed v_k, acceleration a_k, heading θ_k, and angular velocity ω_k of the vehicle at time k.

The state variables of interest in this work are the position, heading, and speed of the vehicle. Thus, the state vector at time k is defined by

$$\boldsymbol{x}_k = [x_k \quad y_k \quad v_k \quad \theta_k]^{\mathsf{T}}$$

For the optimal estimate of the states, we have formulated an extended Kalman filter as follows

$$\mathbf{x}_{k+1} = \mathbf{F}_k\mathbf{x}_k + \boldsymbol{\omega}_k$$
$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k$$

where $\mathbf{F}_k$ is the state transition matrix at time k, $\mathbf{H}_k$ is the observation matrix, $\mathbf{z}_k$ is the sensor measurements, $\boldsymbol{\omega}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$ is a zero mean Gaussian random vector with covariance $\mathbf{Q}_k$, representing system uncertainty, and $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ is a zero mean Gaussian random vector with covariance $\mathbf{R}$, and which represents sensor noise The state transition matrix $\mathbf{F}_k$ can be obtained from (1) - (4) and is given by

$$\mathbf{F}_k = \begin{bmatrix} 1 & 0 & \Delta T\cos\theta_k & 0 \\ 0 & 1 & \Delta T\sin\theta_k & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The acceleration a_k and angular velocity ω_k of the system in (1) - (4) are unknown, and therefore we consider them as unmodeled system parameters (or model uncertainty). Thus, the system uncertainty $\boldsymbol{\omega}_k$ can be defined by

$$\boldsymbol{\omega}_k = \begin{bmatrix} \frac{1}{2}\Delta T^2\bar{a}_k\cos\theta_k \\ \frac{1}{2}\Delta T^2\bar{a}_k\sin\theta_k \\ \Delta T\bar{a}_k \\ \Delta T\bar{\omega}_k \end{bmatrix}$$

where $\bar{a}_k$ and $\bar{\omega}_k$ are uncorrelated zero mean Gaussian random variables with variance of σ_a^2 and σ_ω^2, respectively. The system uncertainty covariance $\mathbf{Q}_k \in \mathbb{R}^{\mathbf{4\times4}}$ can be formulated by

$$\mathbf{Q}_k = \mathrm{E}[\boldsymbol{\omega}_k{\boldsymbol{\omega}_k}^T] = [q_{ij}]$$

where $\mathrm{E}[\cdot]$ denotes expectation of a random variable, and the elements of $\mathbf{Q}_k$ are shown below:

$$q_{11} = \frac{1}{4}\sigma_a^2\cdot\Delta T^4\cdot\cos^2\theta_k$$

$$q_{12} = q_{21} = \frac{1}{4}\sigma_a^2\cdot\Delta T^4\cdot\cos\theta_k\cdot\sin\theta_k$$

$$q_{13} = q_{31} = \frac{1}{2}\sigma_a^2\cdot\Delta T^3\cdot\cos\theta_k$$

$$q_{22} = \frac{1}{4}\sigma_a^2\cdot\Delta T^4\cdot\sin^2\theta_k$$

$$q_{23} = q_{32} = \frac{1}{2}\sigma_a^2\cdot\Delta T^3\cdot\sin\theta_k$$

$$q_{33} = \sigma_a^2\cdot\Delta T^2$$

$$q_{44} = \sigma_\omega^2\cdot\Delta T^2$$

$$q_{14} = q_{24} = q_{34} = q_{41} = q_{42} = q_{43} = 0$$

Since we directly measure the state $\mathbf{x}_k$ contaminated with sensor noise, the observation matrix $\mathbf{H}_k$ becomes the identity matrix independent of time, i.e., $\mathbf{H}_k = \mathbf{H} = \mathbf{I}^{4\times4}$. The measurement noise covariance matrix is defined by

$$\mathbf{R} = \begin{bmatrix} \rho_x^2 & 0 & 0 & 0 \\ 0 & \rho_y^2 & 0 & 0 \\ 0 & 0 & \rho_v^2 & 0 \\ 0 & 0 & 0 & \rho_\theta^2 \end{bmatrix} \qquad (5)$$

where ρ_x^2, ρ_y^2, ρ_v^2, ρ_θ^2 are the variances of sensor noise of the lateral position, longitudinal position, speed, and heading, respectively.

With this formulation, the Kalman prediction and update steps are

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1|k-1}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}^T(\mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R})^{-1}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1})$$

$$\mathbf{P}_{k|k} = (\mathbf{H} - \mathbf{K}_k\mathbf{H}\mathbf{P}_{k|k-1})$$

where $\hat{\mathbf{x}}_{k|k-1} = \mathbf{E}[\mathbf{x}_k|\mathbf{z}_{k-1}, \mathbf{z}_{k-2}, \dots, \mathbf{z}_1]$ is the estimate of $\mathbf{x}_k$ given the observations $\mathbf{z}_{k-1}, \mathbf{z}_{k-2}, \dots, \mathbf{z}_1$, the quantity $\hat{\mathbf{x}}_{k|k} = \mathbf{E}[\mathbf{x}_k|\mathbf{z}_k, \mathbf{z}_{k-1}, \mathbf{z}_{k-2}, \dots, \mathbf{z}_1]$ is the estimate of $\mathbf{x}_k$ given the observations $\mathbf{z}_k, \mathbf{z}_{k-1}, \mathbf{z}_{k-2}, \dots, \mathbf{z}_1$, the matrix $\mathbf{P}_{k|k-1} = \mathbf{E}[\left(\hat{\mathbf{x}}_{k|k-1} - \mathbf{x}_k\right)\left(\hat{\mathbf{x}}_{k|k-1} - \mathbf{x}_k\right)^T]$ is the *a posteriori* covariance matrix (a measure of uncertainty of state estimate $\hat{\mathbf{x}}_{k|k-1}$), $\mathbf{P}_{k|k} = \mathbf{E}[\left(\hat{\mathbf{x}}_{k|k} - \mathbf{x}_k\right)\left(\hat{\mathbf{x}}_{k|k} - \mathbf{x}_k\right)^T]$ is the *a priori* covariance matrix (a measure of uncertainty of state estimate $\hat{\mathbf{x}}_{k|k}$), and $\mathbf{K}_k$ is the optimal Kalman gain.

The most arduous step in the implementation of a Kalman filter is finding the elements in the **Q** and **R** matrices, which are the most influential in the estimation performance. The variances of sensor noise in (5) can be measured from GPS and an inertial measurement unit (IMU) whereas the variances of system noise (σ_a^2 and σ_ω^2) in the **Q** matrix are not easy to determine. In the next section, we develop a robust method to compute these tuning parameters to correct GPS errors.

III. Machine learning approaches for GPS error detection

In this paper, we assume that the following vehicle data are available:

λ : Latitude φ: Longitude v: Vehicle speed

a: Vehicle acceleration θ: Vehicle heading

These measures are commonly available on GPS and On-board Diagnostics (OBD) devices. Note that these are time-based signals with a sampling period of ΔT. Angular velocity can be computed from the vehicle heading as follows:

$$\omega(t) = \frac{\theta(t) - \theta(t - \Delta T)}{\Delta T}$$

The UTM coordinate system was used to covert latitude and longitude (in degrees) to Cartesian coordinates: $x(t)$ and $y(t)$ (in meters).

The EKF outlined in the previous section requires that we choose two important parameters: angular velocity variance σ_ω^2, and acceleration variance: σ_a^2 . The most straightforward approach to computing the needed parameters is to use an available data set of recorded trips and simply compute each using the standard variance formula; for example:

$$\sigma_\omega^2 = \frac{1}{n-1}\sum_{i=1}^{n}(\omega_i - \overline{\omega})^2 \qquad (6)$$

where $\omega_1, \omega_2, \cdots, \omega_n$ are the recorded angular velocities. Note that angular velocity is a zero mean signal, so the $\overline{\omega}$ term can be dropped.

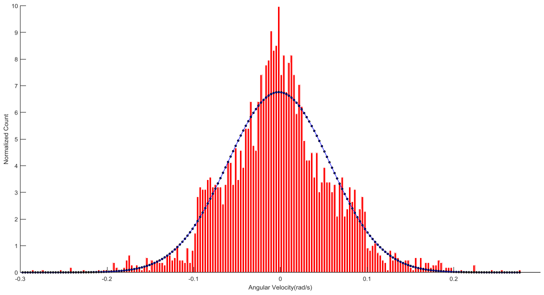


Fig. 1. Calculation of σ_ω^2. The histogram shown in red represents normalized count values: $h(\omega_i)$ of the measured angular velocity, while the blue curve shows the best fit Gassian, with samples: $f(\omega_i)$. To determine the best fit Gaussian, we minimize E_ω or $E_\omega^{Weighted}$ over σ_ω. A similar historam and optimization problem are formulated to determine the variance of acceleration: σ_a , as well.

We propose a machine learning approach that is based on tuning parameters to get a best-fit Gaussian curve to the data. The algorithm consists of the following steps. First, we compute a histogram of the tuning parameter values from training trip data. Figure 1 shows such a (normalized) histogram for the angular velocity (in red). Let $h(\omega_i)$ represent the normalized count (y-axis) from the histogram for the given value of ω_i. Next, we want to find the best fitting Gaussian curve, of the form:

$$f(\omega) = \frac{1}{\sqrt{2\pi}\sigma_\omega} e^{-\frac{1}{2}\frac{\omega^2}{\sigma_\omega^2}} \qquad (7)$$

For a given value of σ_ω and ω_i, we can compute the corresponding value of the Gaussian curve: $f(\omega_i)$. Hence, we can formulate the following optimization problem:

$$\text{Minimize } E_\omega$$
$$\text{s.t. } \sigma_\omega \in R^+$$

where the objective function E_ω is the sum-squared error between the f and h samples:

$$E_\omega = \sum_{\omega_i}[f(\omega_i) - h(\omega_i)]^2 \qquad (8)$$

In the next section, we will show that the following weighted objective function yields slightly better results:

$$E_{\omega}^{Weighted} = \sum_{\omega_i} \omega_i [f(\omega_i) - h(\omega_i)]^2 \quad (9)$$

This objective function puts more weight on fitting the tails of the distribution.

Similar fitting functions: $h(a_i)$, $f(a_i)$, and objective functions: E_a, and $E_a^{Weighted}$ were formulated to compute the variance of acceleration σ_a^2, as well.

We propose a model free neural network-based method for vehicle position error detection based on the past trajectory of vehicle positions (measured GPS coordinates). We used a multi-layer perceptron (MLP) with architecture shown in Figure 2. We found that this simple structure with one hidden layer was sufficiently powerful for the present error detection problem. The inputs to the neural network are measurements of four variables: vehicle position, speed, and heading, over the time period: $t-1, t-2, \cdots, t-w$, where w is a parameter which specifies the size of the temporal *sliding window*. The outputs of the network: $E_x(t)$ and $E_y(t)$ occur at time t and represent the amount of error in position. A data set of inputs and target outputs is needed to train the neural network.

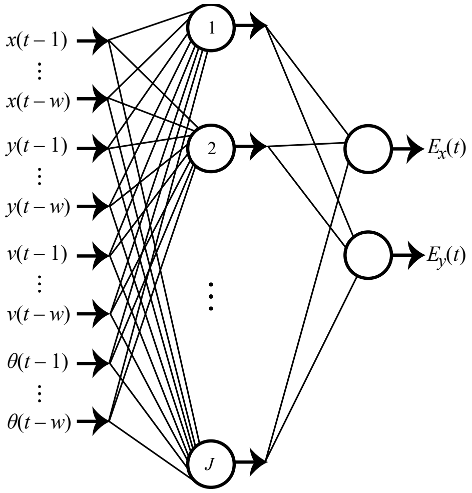


Fig. 2. Architecture of GPS error-correcting neural network.

IV. Experiments

To evaluate the effectiveness of the proposed approaches, we used real GPS data collected over several trips, each about 50 km long in Dearborn, MI, and contains both city and highway driving. Figure 3 shows a map of the route (a roundtrip) taken by the driver. GPS data was collected from 17 such trips with 5 different drivers. The vehicle position, velocity, heading, and acceleration were collected at a sampling rate of 100 Hz.

We are interested in how the parameter values and prediction accuracy are affected by road/traffic conditions. From the entire trip, 5 different segments: S_1 through S_5 were identified as having interesting traffic characteristics. The 5 segments are also shown in Figure 3, and the associated traffic characteristics are listed below:

S_1: Intersection and left turn

S_2: Straight section

S_3: Curve

S_4: Stop sign and U-turn

S_5: Slight turn

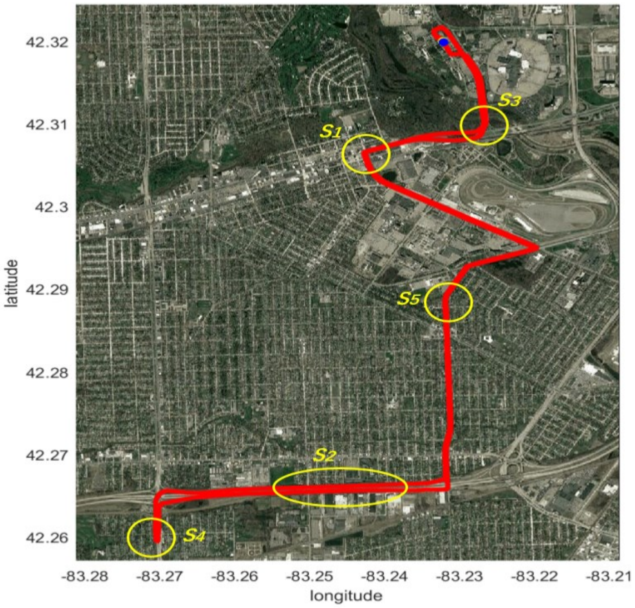


Fig. 3. The 50km route (roundtrip) taken by each driver. Five different segments of the trip are indicated.

In the following subsections, we will corrupt the recorded GPS data with various amounts of additive white Gaussian noise. Figure 4 shows a short segment of the trip along with the original GPS data (in red) and then with added noise set to 0.5m, 1.5m, and 2.5m.

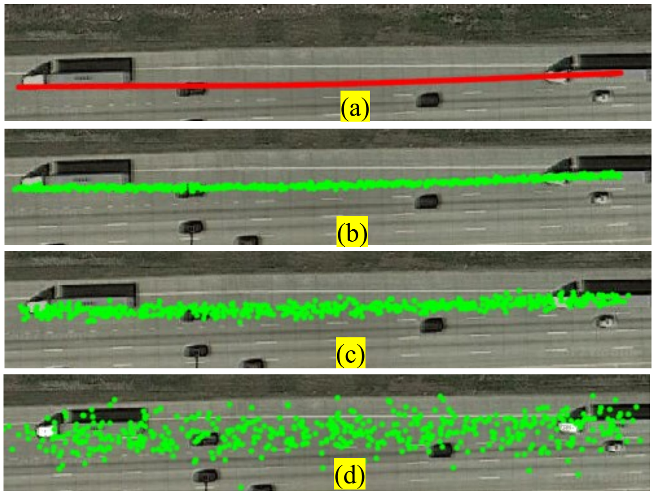


Fig. 4. Illustration of the trip data along with added noise. (a) Original GPS data. (b) The trip with 0.5m additive white Gassuian noise (c) 1.5m noise and (d) 2.5 m noise.

A. Computation of Kalman Filter Parameters

To study the computation of the parameters σ_ω and σ_a needed by the Kalman filter, we performed a preliminary experiment of computing both parameters using the data from Trip #1. Three different methods were used to compute the variances: equation 6, and the two curve fitting methods outlined in (7) – (9). To test the performance of each variance setting, Trips #1 - #3 were corrupted with 1.5m GPS noise, and the Kalman filter was applied to the noisy trip data. A mean-squared error measure was used to assess performance:

$$E = \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{pos}_i - \mathbf{pos}_i^{KF}\|^2$$

where $\mathbf{pos}_i = [x(t_i) \quad y(t_i)]^T$ is a ground truth position vector taken from the original (noiseless) positions and $\mathbf{pos}_i^{KF}$ is the corresponsing position vector after the Kalman filter has been applied to the noisy signal.

Figure 5 shows a plot of the two curve fitting error functions: E_ω or $E_\omega^{Weighted}$. Also shown are the minimizers of the respective functions: σ_ω^* and $\sigma_\omega^{*Weighted}$ which give the best Gaussian fit.

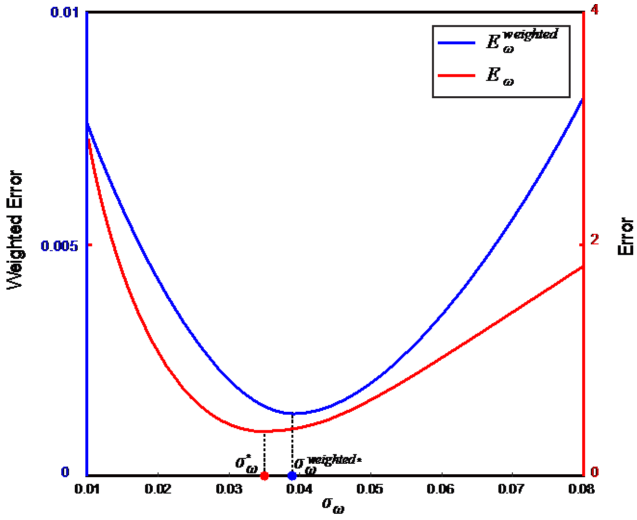


Fig. 5. A plot of the curve fitting errors: E_ω, and $E_\omega^{Weighted}$ as a function of σ_ω. In each case, we choose the value of σ_ω which minimizes the error.

The three values of σ_ω: (i) the one computed from Equation 5, (ii) σ_ω^* and (iii) $\sigma_a^{*Weighted}$ were used to filter the noisy signal. Table I shows the mean-squared error tested over Trips #1 - #3. The best performance was with the curve fitting approach using the weighted objective function. In the experiments that follow, this is the method that will be used to compute σ_ω and σ_a.

The variance values, of course, depend on the data set used to compute them. Figure 6 shows a scatter plot of σ_ω vs. σ_a computed over all 12 trips (shown in blue).

TABLE I. ERROR OF KALMAN FILTERED SIGNAL TESTED ON 3 TRIPS AND WITH 3 DIFFERENT SETTINGS OF THE VARIANCE PARAMETERS.

Kalman Filter Error (m)	**Trip 1**	**Trip 2**	**Trip 3**
Variance Formula	1.4003	1.2195	1.2387
Curve Fitting	1.3370	1.1598	1.2047
Weighted Curve Fit	1.3115	0.9942	1.1108

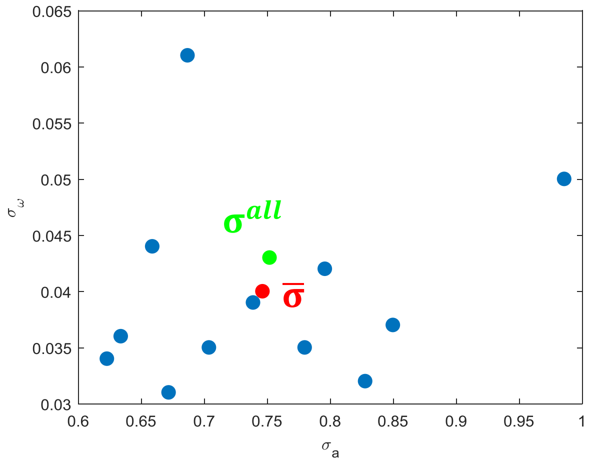


Fig. 6. A scatter pot of σ_ω vs. σ_a from the first 7 trips (shown in blue). The average values: $\bar{\sigma}_\omega$ and $\bar{\sigma}_a$ (averaged sigmas over the first 7 trips) are shown in red. The sigmas resulting from using the data from all 7 trips: σ_ω^{All} and σ_a^{All} are shown in green.

In light of Figure 6, there are two different ways to obtain a reasonable estimate of the variance. The first is to compute the center of mass of individual σ_ω^i and σ_a^i from the various trips This will be denoted by $\bar{\sigma}_\omega = \sum_{i=1}^{K}\sigma_\omega^i$ and $\bar{\sigma}_a = \sum_{i=1}^{K}\sigma_a^i$ and shown in red in Figure 6. The second method is to compute the variance of measured data (including all trips) which will be denoted by σ_ω^{All} and σ_a^{All} and they are shown in green in Figure 6.

B. Evaluation of Performance on Fixed Route

In the following experiments, data from the first 12 trips were used to compute the parameters: σ_ω and σ_a. Several different ways of computing these statistical measures were explored. In the first approach, the standard deviations were computed over each trip separately: σ_ω^i and σ_a^i , $i = 1, 2, \dots, 12$. With these values, the Kalman filter was used to filter the noisy data on the corresponding trip data with 3 levels of additive white noise: 0.5m, 1.5m, and 2.5m. So, for example, σ_ω^1 and σ_a^1 were used to filter the noisy Trip #1 data. The results are shown in Figure 7.

Next, to determine how robust the standard deviation is, we used σ_ω^1 and σ_a^1 and evaluated the Kalman filter on all the *other* 11 noisy trips (2 through 12). The process was repeated for σ_ω^2 and σ_a^2 through σ_ω^{12} and σ_a^{12}. The results are shown in Figure 8.

Note that the results here represent the average error over the 11 other trips.

Next, we tested system performance using the average parameters: $\bar{\sigma}_\omega$ and $\bar{\sigma}_a$. The results of this experiment are shown in Figure 9.

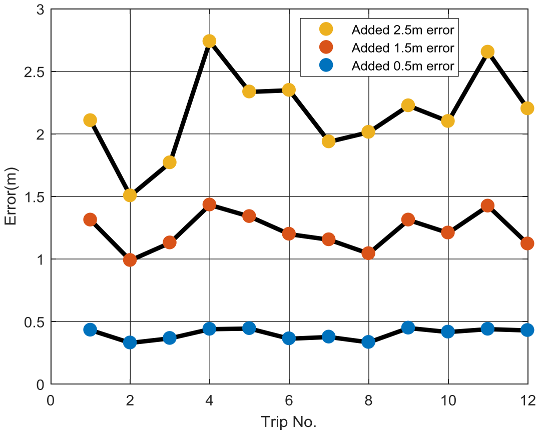


Fig. 7. σ_ω^i and σ_a^i were computed on trip i and evelauted on trip i.

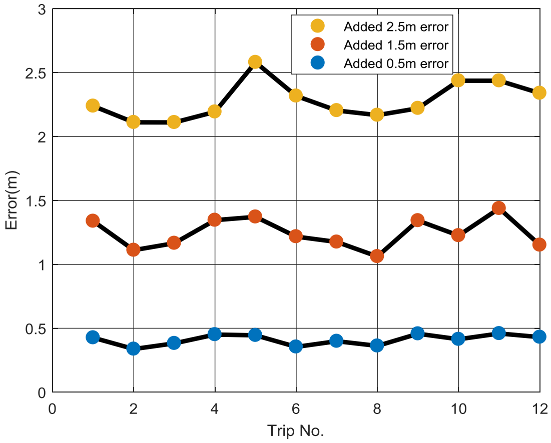


Fig. 8. σ_ω^i and σ_a^i were computed on trip i and evelauted against all 11 other trips.

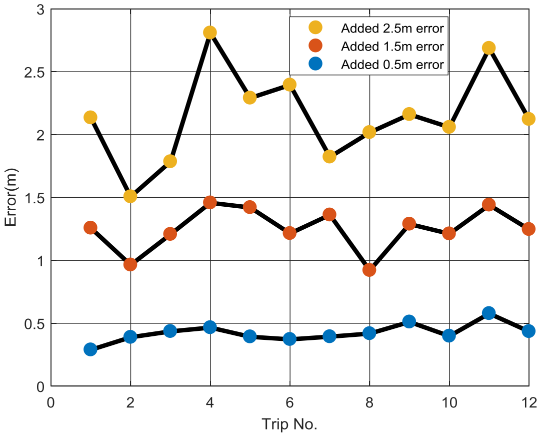


Fig. 9. $\bar{\sigma}_\omega$ and $\bar{\sigma}_a$ were computed by averaging the 12 individual σ_{ω_i} and σ_{a_i} and evaluated on all 12 trips.

Finally, we tested system performance using the parameters fitted by all 12 trip data: σ_ω^{All} and σ_a^{All}. The results are shown in Figure 10. A summary of the average errors (averaged over all the test trips) for each approach is given in Table II.

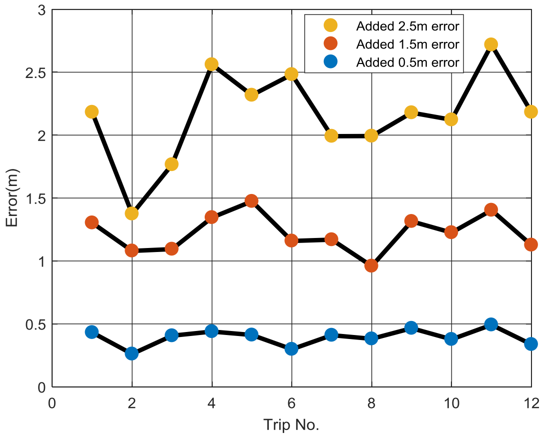


Fig. 10. σ_ω^{All} and σ_a^{All} were computed by using all of the data in the 12 trips. and evaluated on all 12 trips.

TABLE II. SUMMARY OF KALMAN FILTER ERROR.

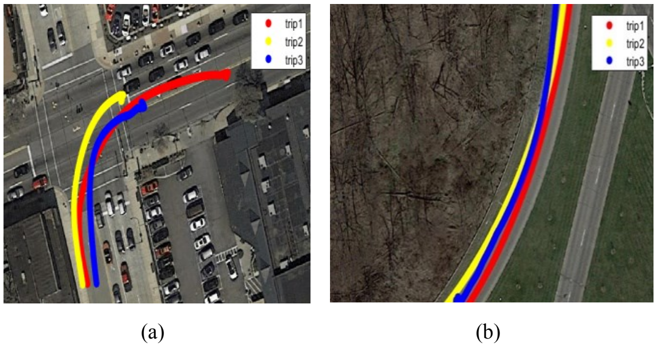
Average Kalman Filter Error (m)	Input Noise **0.5m**	Input Noise **1.5m**	Input Noise **2.5m**
σ_ω and σ_a computed and tested on same trip	0.4012	1.2218	2.1612
σ_ω and σ_a computed on one trip and tested on all 11 others	0.4097	1.2452	2.2739
$\bar{\sigma}_\omega$ $\bar{\sigma}_a$	0.4090	1.2301	2.1640
σ_ω^{All} σ_a^{All}	0.4088	1.2293	2.1627

From the results of this set of experiments we observe the following properties:

(1) the EKF using the parameters fitted on the same trip data as the test data gives the best performance. However, this can only serve as a measure of performance upper bound. It cannot be used for online error prediction.

(2) The performances given by the EKF that uses σ_ω^{All} and σ_a^{All} and the EKF that uses $\bar{\sigma}_\omega$ and $\bar{\sigma}_a$ are very close. However, unlike σ_ω^{All} and σ_a^{All}, $\bar{\sigma}_\omega$ and $\bar{\sigma}_a$ can be updated when new trip data are available for learning without referring to the previous training data. Therefore, we recommend use of $\bar{\sigma}_\omega$ and $\bar{\sigma}_a$.

Next, we investigate how road/traffic features affect system performance. Figure 11 shows GPS data for several trips (with different drivers) over segments S_1 and S_3. The variance parameters were computed on each segment of the 12 trips and then tested on the same 12 trips. Table III shows a comparison of the average Kalman filter error. The results in the table are in the format: Error from using σ_ω^{All} and σ_a^{All} / error from using $\bar{\sigma}_\omega$ and $\bar{\sigma}_a$. From the table, we see that the two methods give very similar values. This is important because it indicates that it is possible to use our algorithm in an online learning mode. That is, $\bar{\sigma}_\omega$ and $\bar{\sigma}_a$ can be continually updated online, as more trips are made by the driver.



(a) (b)

Fig. 11. Illustration of a few trips over segments: (a) S_1 (b) S_3.

TABLE III. KALMAN FILTER ERROR USING TWO DIFFERENT SETTINGS OF VARIANCE PARAMETERS: σ_ω^{All} AND σ_a^{All} / $\bar{\sigma}_\omega$ AND $\bar{\sigma}_a$

KF Error (m)	Input Noise **0.5m**	Input Noise **1.5m**	Input Noise **2.5m**
S_1	0.2536 / 0.2587	0.6281 / 0.6456	1.7809 / 1.7845
S_2	0.2190 / 0.2165	0.5892 / 0.5859	1.3315 / 1.3278
S_3	0.3981 / 0.4007	1.1244 / 1.1287	2.0993 / 2.1019
S_4	0.4136 / 0.4029	1.2331 / 1.2245	2.1898 / 2.1697
S_5	0.5009 / 0.4891	1.4895 / 1.4710	2.4076 / 2.4031

C. Comparison of EKF and Neural Network

A neural network with 10 hidden nodes and architecture shown in Figure 2 was trained using data from the first 12 trips and tested the other 5 trips. Figure 12 shows the error of the neural network approach, as well as the two EKF methods with two parameters settings: σ_ω^{All} and σ_a^{All} , and $\bar{\sigma}_\omega$ and $\bar{\sigma}_a$.

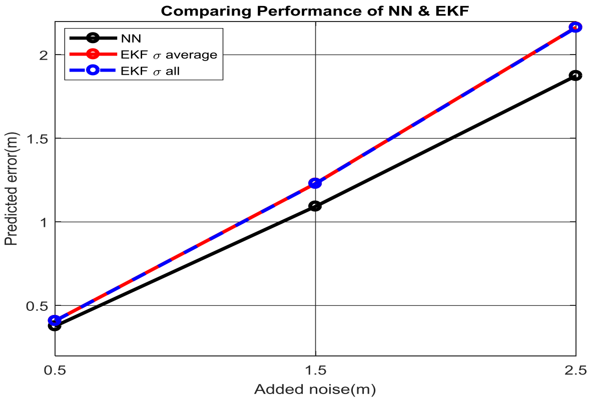


Fig. 12. Performance comparisons between the EKF-based approach and the neural network-based approach using five test trips.

The neural network gives the best performance. The two EKF based methods show almost identical performance. However, we want to point out that the neural network requires supervised learning (with known target values), which is not easy to implement for in-vehicle applications.

D. Testing on a New Route

In the above experiments, the system was trained and tested on data collected over a fixed route, but with multiple drivers. In this section, we utilize the same Kalman filter parameters as computed in Section IV.B, but we test on an entirely new route. The route was local driving for approximately 9 miles and 12 minutes as shown in Figure 13. Two data acquisition systems were used. The first provided high quality GPS measurements with 100 Hz sampling rate. The signals captured with this device were considered ground truth for the trip. The second was a low quality device with a sampling rate of 1 Hz. The signals in this device were considered the noisy input signal that was sent to the Kalman filter.

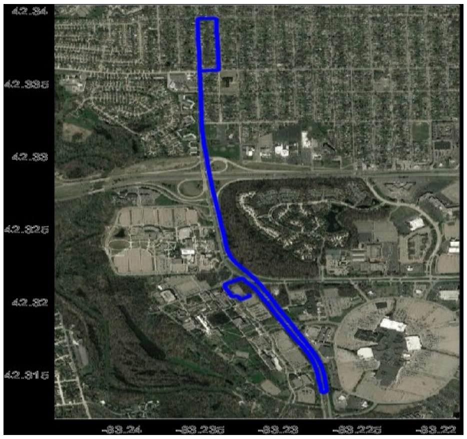


Fig. 13. A new route consisting of 9 miles of city driving and approximately 12 minutes.

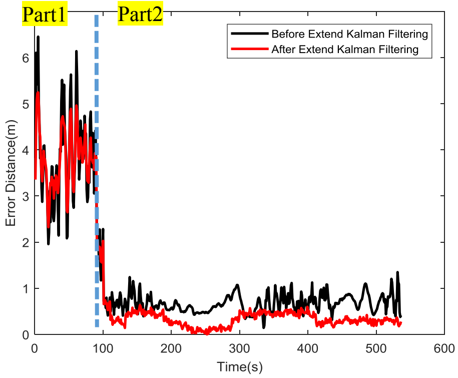


Fig. 14. A plot of Kalman filter error vs. time when tested on the new route.

A plot of the Kalman filter estimation error as a function of time is shown in Figure 14. Notice that there is a large amount of error at the beginning of the trip, but then after some time, the Kalman filter is very effective in removing noise. Figure 15 shows what was occurring at the beginning of the trip: the driver was exiting a parking lot and making several starts and stops.

Figure 16 show a later portion of the trip where the Kalman filter provides effective error correction even with a low cost sensor.

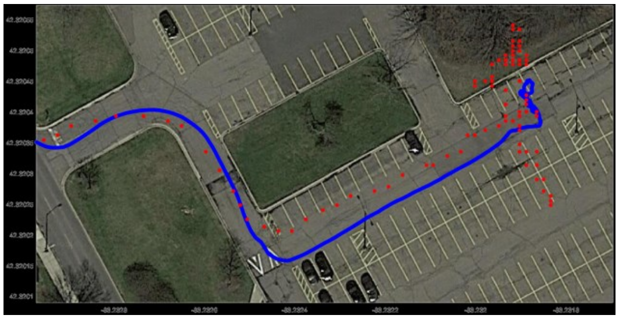


Fig. 15. At the beginning of the trip, the driver was in a parking lot and the input signal (red dots) is very noisy. The blue line is the ground truth.

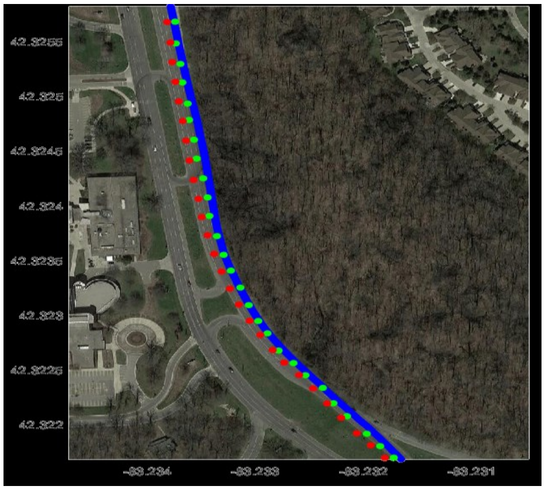


Fig. 16. A section of the trip which shows the the effect of Kalman filtering. The red dots are the noisy signal, the green dots are the Kalman-filtered signal, and the blue line is the ground truth.

V. CONCLUSION

In this paper, we proposed several methods for improving vehicle position estimation. A curve fitting approach was proposed for tuning important Kalman filter parameters. A neural network was proposed for estimating position errors. The results show that the neural network can give slightly better results that the Kalman filter. However, the neural network requires a training set containing ground truth, which makes it less appealing for in-vehicle applications. The Kalman filter approach is more amenable to an online and in-vehicle implementation, where the tuning parameters can be continually updated as new data is obtained. In future work, we will apply these approaches to larger data sets, consisting of many drivers and different driving scenarios. In addition, we will explore its use as a component in other intelligent vehicle systems design problems, such as autonomous vehicle design.

ACKNOWLEDGMENT

This work is support in part by grants from Mobility Transformation Center and Michigan Institution for Data Science at the University of Michigan.

REFERENCES

[1] Xu, H., H. Liu, C-W.Tan, and Y. Bao (2010). "Development and Application of a Kalman Filter and GPS Error Correction Approach for Improved Map-matching."

[2] Bao, Y. and Liu, Z. (2006). GPS, traffic monitoring and digital map. National Defense Industry Press, Beijing.

[3] Grewal, Mohinder S.; Weill, Lawrence Randolph; Andrews, Angus P. (2001). Global positioning systems, inertial navigation, and integration. John Wiley and Sons. ISBN 978-0-47135-032-3.

[4] Kalman, R. E. (1960). "A New Approach to Linear Filtering and Prediction Problems". Journal of Basic Engineering 82: 35.doi:10.1115/1.3662552.

[5] Brown, R. G. and Hwang, P. Y. C. (1992). Introduction to random signals and applied Kalman filtering, Second Edition, John Wiley& Sons, Inc.

[6] Slaughter, D.C.; Giles, D.K.; Downey, D. Autonomous robotic weed control systems: A review. Comput. Electron. Agric. 2008, 61, 63–78.

[7] Brown, R.G.; Hwang, P.Y.C. The Discrete Kalman Filter, State-Space Modeling and Simulation. In Introduction to Random Signals and Applied Kalman Filtering, 3th ed.; John Wiley and Sons: Toronto, ON, Canada, 1997; pp. 190–241

[8] Jo, T., Haseyamai and M., Kitajima, H. (1996). A map matching method with the innovation of the Kalman filtering, IEICE Trans. Fund. Electron. Comm. Comput. Sci. E79-A, 1853-1855.

[9] August, P.; Michaud, J.; Lavash, C.; Smith, C. GPS for environmental applications: Accuracy and precision of locational data. Photogramm. Eng. Remote Sens. 1994, 60, 41–45.

[10] Saha, M.; Goswami, B.; Ghosh, R. Two Novel Costs for Determining the Tuning Parameters of the Kalman Filter. In Proceedings of Advances in Control and Optimization of Dynamic Systems (ACODS-2012), Bangalore, India, 2012; pp. 1–8

[11] Goodall, C. and El-Sheimy, N. Intelligent Tuning of a Kalman Filter Using Low-Cost MEMS Inertial Sensors. In Proceedings of 5th International Symposium on Mobile Mapping Technology (MMT'07), Padua, Italy, 2007; pp. 1–8.

[12] Alonso-Garcia, S. Gomez-Gil, and J. Arribas, J.I. Evaluation of the use of low-cost GPS receivers in the autonomous guidance of agricultural tractors. Span. J. Agric. Res. 2011, 9, 377–388.

[13] Keicher, R. and Seufert, H. Automatic guidance for agricultural vehicles in Europe. Comput. Electron. Agric. 2000, 25, 169–194

[14] Jaime Gomez-Gil, Ruben Ruiz-Gonzalez, Sergio Alonso-Garcia, and Francisco Javier Gomez-Gil, "2GA Kalman Filter Implementation for Precision Improvement in Low-Cost GPS Positioning of Tractors", Sensors 2013, 13, 15307-15323; doi:10.3390/s131115307 Journal of Intelligent Transportation Systems, 14(1), 27-36.

[15] M. Ananthasayanam, "Kalman Filter Design by Tuning its Statistics or Gains?" in International Conference on Data Assimilation, IISc, Bangalore, July 2011, [Online; accessed 18-Oct-2012].

[16] Adhitya Bhawiyuga, Hoa-Hung Nguyen, Han-You Jeong, "An Accurate Vehicle Positioning Algorithm based on Vehicle-to-Vehicle (V2V) Communications," 2012 IEEE International Conference on Vehicular Electronics and Safety, July 24-27, 2012. Istanbul, Turkey

[17] Chen and H. Hu, "IMU/GPS based pedestrian localization," 2012 4th Computer Science and Electronic Engineering Conference (CEEC), Colchester, 2012, pp. 23-28.

[18] Guiqiang Mao, Baris Fidan, Brian D.O. Anderson, "Wireless sensornetwork localization techniques," Elsevier Computer Networks, vol. 51, pp. 2529-2553, May 2009.