



RESEARCH ARTICLE

# Comparative evaluation of Kalman filters and motion models in vehicular state estimation and path prediction

Lu Tao,<sup>1</sup> Yousuke Watanabe,<sup>2</sup> Shunya Yamada,<sup>1</sup> and Hiroaki Takada<sup>1,2</sup>

<sup>1</sup> Graduate School of Informatics, Nagoya University, Chikusa, Nagoya, Aichi, Japan.

<sup>2</sup> Institute of Innovation for Future Society, Nagoya University, Chikusa, Nagoya, Aichi, Japan.

Corresponding author. E-mail: [tao.lu@a.mbox.nagoya-u.ac.jp](mailto:tao.lu@a.mbox.nagoya-u.ac.jp)

**Received:** 17 June 2020; **Accepted:** 03 April 2021; **First published online:** 4 June 2021

**Keywords:** estimation, extended Kalman filter, unscented Kalman filter, path

## Abstract

Vehicle state estimation and path prediction, which usually involve Kalman filter and motion model, are critical tasks for intelligent driving. In vehicle state estimation, the comparative performance assessment, regarding accuracy and efficiency, of the unscented Kalman filter (UKF) and the extended Kalman filter (EKF) is rarely discussed. This paper is devoted to empirically evaluating the performance of UKF and EKF incorporating different motion models and investigating the models' properties and the affecting factors in path prediction. Extensive real world experiments have been carried out and the results show that EKF and UKF have roughly identical accuracy in state estimation; however, EKF is faster than UKF generally; the fastest filter is about 2.6 times faster than the slowest. The path prediction experiments reveal that the velocity estimate and the used motion model affect path prediction; the more realistically the model reflects the vehicle's driving status, the more reliable its predictions.

## 1. Introduction

The vehicle's current and future states, such as position and velocity, are the cornerstone for both advanced driver assistance systems (ADAS) and autonomous driving (AD) because they enable the vehicle to understand current and upcoming situations. In order to obtain the state, multiple sensors are mounted on the vehicle, such as global positioning system (GPS), inertial measurement unit (IMU) and light detection and ranging (LiDAR). Unfortunately, these sensors inherently suffer from noise, meaning that the raw sensor data cannot be used directly, in general. The method introduced by Kalman (1960) is the method most commonly used for addressing such noise. There are two models of Kalman filter (KF): the process model, describing the system's temporal evolution; and the observation model, describing the sensors' properties. The standard KF is designed for linear systems. When the system is nonlinear, KF is extended by linearising the nonlinear models; this is the so-called extended KF (EKF) (Gelb, 1974). In EKF, the nonlinearities are approximated analytically through first-order Taylor expansion. For decades, EKF has been the dominant state estimator for nonlinear problems. However, it has two flaws: (1) the linearisation may lead to poor performance and divergence when the nonlinearities are considerable, because the higher order terms of Taylor expansion are ignored; (2) the derivation of the Jacobian matrices may be nontrivial in some applications. In contrast with EKF, the unscented KF (UKF) (Julier et al., 1995) is founded on the intuition that it is easier to approximate a Gaussian distribution than it is to approximate an arbitrary nonlinear function or transformation (Julier and Uhlmann, 1997). In UKF, a minimal set of deterministically sampled sigma points is utilised to capture completely the true mean and covariance of the Gaussian random variables and when propagated through the true

nonlinearity, captures the posterior mean and covariance, accurate to the third order (Taylor expansion) for any nonlinearity; in contrast, EKF only achieves first-order accuracy (Wan and Van Der Merwe, 2000).

There are ample studies that compare the performance of EKF and UKF. Lefebvre et al. (2004) theoretically compared the performance in process and measurement update step of some KFs. In vehicle localisation, the performances of EKF and UKF are compared experimentally and they were found to behave similarly in terms of accuracy and consistency (Ndjeng Ndjeng et al., 2009). In vehicle navigation, the accuracy and computational time of EKF and UKF were evaluated by St-Pierre and Gingras (2004); it was found that UKF had just a slight improvement in accuracy whereas it cost much more computational time than EKF. Yang et al. (2017) investigated the performance of EKF and UKF with different feedback control models and showed the superiority of UKF. In vehicular state estimation, the performance of EKF and UKF is rarely discussed, because UKF is seen as a better choice and sufficient for most current applications where only several vehicles are involved.

Besides the KF form (EKF or UKF), the process model also affects the performance of a filter. An elaborate process model significantly improves the filter's performance (Julier and Durrant-Whyte, 2003). In road safety related applications, kinematic models prevail (Lefèvre et al., 2014), such as the constant velocity/acceleration (CV/CA) model and constant turn rate and velocity/acceleration (CTRV/CTRA) model. A comprehensive comparison and evaluation of these models was carried out by Schubert et al. (2011, 2008); they concluded that the sophisticated model did not demonstrate better performance in any case, and the appropriateness of model heavily depended on the application. In the literature, CTRV and CTRA models are popular choices in various applications (Madhavan et al., 2006; Polychronopoulos et al., 2007; Lytrivis et al., 2011; Houenou et al., 2013).

The cooperative intelligent transportation system (ITS) has been regarded as the next important step towards the vision of accident-free driving (Weiß, 2011). The applications involved with cooperative ITS and vehicular networking are of real-time nature (Karagiannis et al., 2011). For instance, recently Dynamic Map 2.0 (DM2.0) has been proposed to manage city-scale traffic data to facilitate safe and comfortable driving (Watanabe et al., 2018). In DM2.0, massive sensor data are uploaded to the server and that poses a compelling real-time need for the system.

The available studies compared either EKF/UKF or motion models separately, focusing on the one-sided aspect of accuracy. In their conclusions, only an appropriate KF form or motion model was recommended. In practice, the best filter is a tradeoff between accuracy and efficiency. In this paper, the accuracy and efficiency of both EKF and UKF, incorporating CTRV and CTRA models, are evaluated extensively. The analyses can guide the reader to make an appropriate filter considering their specific demands. Moreover, in path prediction, the affecting factors and the models' properties are also investigated.

The remainder of this paper is structured as follows. Section 2 presents the methodology of this paper. In Section 3, the experimental results are discussed in details. Finally, the conclusions are drawn in Section 4.

## 2. Methodology for comparative evaluation

### 2.1. System framework

The system framework of this study is illustrated in Figure 1. In this system, LiDAR, GPS and IMU are used. Due to the sensor data inherently containing greater or lesser uncertainties, it is not suggested that the raw sensor measurements should be used directly in application. Especially in path prediction, the uncertainties will be propagated and accumulated. In application, it is necessary initially to employ a state estimator to reduce the uncertainties considering real-time sensor data and vehicular motion model. An accurate vehicular state estimate helps both current and future situation awareness.

To gain awareness of the upcoming situations, path prediction is usually performed in ADAS and AD applications. In this paper, a future path is defined as a moving trajectory between the current time

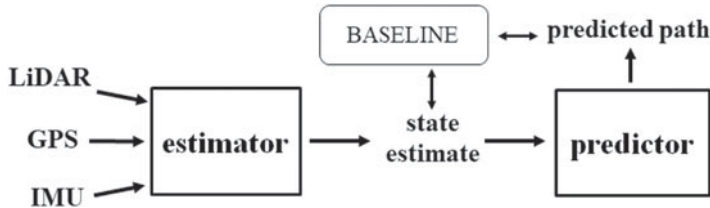


Figure 1. Overview of system.

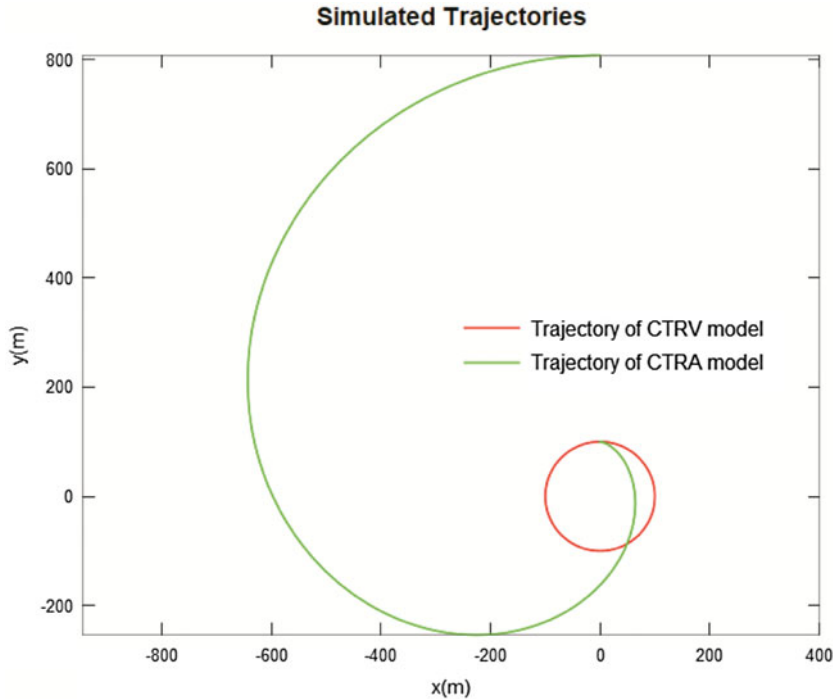


Figure 2. Simulated trajectories of CTRV and CTRA models.

(instant  $k$ ) and few seconds later. In each instant  $(\dots, k-1, k, k+1, \dots)$ , a future path is iteratively predicted by the kinematic model. The obvious advantages of utilising the models to predict path are feasibility and simplicity. Given an initial state, a path can be generated iteratively in real time, as shown in Figure 2. Such predicted path has been used in some vital applications, such as collision warning (Tan and Huang, 2006) and emergency electronic brake lights (Lytrivis et al., 2011). The estimated state and predicted path are compared with the baseline to assess their relative difference.

## 2.2. Vehicular motion models

The vehicle movement is governed by physical laws. Therefore the vehicle's motion can be explicitly expressed by mathematical equations. In this paper, vehicular motion models, specifically the CTRV and CTRA models, play the roles of KF's process model and path predictor. Certainly, more sophisticated models do exist, e.g., that of Julier and Durrant-Whyte (2003). The reasons these two models are chosen are: (1) they are sufficient for most ITS applications; (2) more elaborate models may contain unobservable parameters; (3) they are widely used in intelligent vehicles.

### 2.2.1. CTRV model

The CTRV model hypothesises that the vehicle is moving at a constant turn rate and velocity, which results in a circular trajectory, as shown in Figure 2 (simulated with initial state  $[0 \text{ m}, 100 \text{ m}, 0 \text{ deg}, 3.14 \text{ m/s}, -18 \text{ deg/s}]^T$  and duration 200 s). The state vector of CTRV model at instant  $k$  is

$$\mathbf{x}_k = [x, y, \theta, v, \omega]^T \quad (1)$$

where  $x, y$  represent the position;  $\theta$  is the heading;  $v$  the velocity;  $\omega$  the angular rate. Based on the hypothesis, the next instant's state is derived through

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \int_{t_k}^{t_{k+1}} \dot{\mathbf{x}}_{\tau|t_k} d\tau \quad (2)$$

Consequently, the CTRV model's state transition equation is

$$\mathbf{x}_{k+1} = \begin{bmatrix} x_k + \frac{v_k}{\omega_k} \sin(\theta_k + \omega_k T) - \frac{v_k}{\omega_k} \sin(\theta_k) \\ y_k - \frac{v_k}{\omega_k} \cos(\theta_k + \omega_k T) + \frac{v_k}{\omega_k} \cos(\theta_k) \\ \theta_k + \omega_k T \\ v_k \\ \omega_k \end{bmatrix} \quad (3)$$

where  $T$  is the prediction time interval.

### 2.2.2. CTRA model

Regarding the velocity derivative as constant, the CTRV model is extended to the CTRA model. The trajectory of this model follows a clothoid, as Figure 2 shows (simulated with initial state  $[0 \text{ m}, 100 \text{ m}, 0 \text{ deg}, 3.14 \text{ m/s}, 0.11 \text{ m/s}^2, -1.8 \text{ deg/s}]^T$  and duration 200 s). At instant  $k$ , the CTRA model's state vector is

$$\mathbf{x}_k = [x, y, \theta, v, a, \omega]^T \quad (4)$$

where  $a$  denotes the tangential acceleration. Through Equation (2), the model's state transition equation is derived as follows.

$$\mathbf{x}_{k+1} = \begin{bmatrix} x_k + \frac{(v_k + a_k T) \sin(\theta_k + \omega_k T) - v_k \sin(\theta_k)}{\omega_k} + \frac{a_k [\cos(\theta_k + \omega_k T) - \cos(\theta_k)]}{\omega_k^2} \\ y_k - \frac{(v_k + a_k T) \cos(\theta_k + \omega_k T) - v_k \cos(\theta_k)}{\omega_k} + \frac{a_k [\sin(\theta_k + \omega_k T) - \sin(\theta_k)]}{\omega_k^2} \\ \theta_k + \omega_k T \\ v_k + a_k T \\ a_k \\ \omega_k \end{bmatrix} \quad (5)$$

## 2.3. KFs for vehicular state estimation

### 2.3.1. State space model

The discrete-time nonlinear dynamic system for vehicular state estimation is formed by

$$\mathbf{x}_{k+1} = F(\mathbf{x}_k) + \mathbf{w}_k \quad (6)$$

$$\mathbf{y}_k = H(\mathbf{x}_k) + \mathbf{e}_k \quad (7)$$

where  $\mathbf{x}_{k+1}$  denotes the  $L$ -dimensional state of the system at instant  $k+1$ , and  $\mathbf{y}_k$  is the observation.  $F$  and  $H$  are the process and observation functions.  $\mathbf{w}_k$  and  $\mathbf{e}_k$  are the process and observation noises, which are assumed as independent zero-mean white Gaussian noise of covariance matrices  $\mathbf{Q}_k$  and  $\mathbf{R}_k$ .

### 2.3.2. EKF

EKF follows the same prediction-correction mechanism as the standard KF. The main difference is that EKF has to linearise the nonlinear state space models [Equations (6) and (7)] at each iteration through first-order Taylor expansion around the most recent state estimate, employing the Jacobian matrices in Equation (8).

$$\mathbf{J}_F = \left. \frac{\partial F(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_k} \quad \mathbf{J}_H = \left. \frac{\partial H(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k+1}^-} \quad (8)$$

where  $\mathbf{J}_F$  and  $\mathbf{J}_H$  denote the Jacobian matrices of the process and observation functions (for more details see [Appendix A](#)). EKF is initialised via

$$\hat{\mathbf{x}}_0 = E[\mathbf{x}_0] \quad \mathbf{P}_0 = E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T] \quad (9)$$

Suppose the system state has been estimated at instant  $k$ , the state and covariance at next instant are then predicted through

$$\hat{\mathbf{x}}_{k+1}^- = F(\hat{\mathbf{x}}_k) \quad (10)$$

$$\mathbf{P}_{k+1}^- = \mathbf{J}_F \mathbf{P}_k \mathbf{J}_F^T + \mathbf{Q}_k \quad (11)$$

As long as the new observations are available, the prediction is corrected by

$$\mathbf{G}_{k+1} = \mathbf{P}_{k+1}^- \mathbf{J}_H^T (\mathbf{J}_H \mathbf{P}_{k+1}^- \mathbf{J}_H^T + \mathbf{R}_{k+1})^{-1} \quad (12)$$

$$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_{k+1}^- + \mathbf{G}_{k+1} (\mathbf{y}_{k+1} - H(\hat{\mathbf{x}}_{k+1}^-)) \quad (13)$$

$$\mathbf{P}_{k+1} = (\mathbf{I} - \mathbf{G}_{k+1} \mathbf{J}_H) \mathbf{P}_{k+1}^- \quad (14)$$

where  $\mathbf{G}_{k+1}$  is Kalman gain. The EKF approximates an optimal estimate through Equations (8)–(14), recursively. The first-order linearisation and the Taylor expansion around the recent estimates (not the true state) lead to additional error. UKF, which is derivative free, is proposed to overcome the limitations.

### 2.3.3. UKF

Unlike EKF, UKF copes with the nonlinearities by deterministically sampling around the optimal estimate  $\hat{\mathbf{x}}_k$ . Firstly,  $2L+1$  sigma points  $\mathcal{X}_{k,i}$  (with the weights  $W_i$ ) are selected through

$$\begin{aligned} \mathcal{X}_{k,0} &= \hat{\mathbf{x}}_k \\ \mathcal{X}_{k,i} &= \hat{\mathbf{x}}_k + (\sqrt{(L+\lambda)\mathbf{P}_k})_i, \quad i = 1, \dots, L \\ \mathcal{X}_{k,i} &= \hat{\mathbf{x}}_k - (\sqrt{(L+\lambda)\mathbf{P}_k})_i, \quad i = L+1, \dots, 2L \\ W_0^{(m)} &= \frac{\lambda}{L+\lambda}, \quad W_0^{(c)} = \frac{\lambda}{L+\lambda} + (1 - \alpha^2 + \beta) \\ W_i^{(m)} &= W_i^{(c)} = \frac{1}{2(L+\lambda)}, \quad i = 1, \dots, 2L \end{aligned} \quad (15)$$

where  $(\sqrt{(L+\lambda)\mathbf{P}_k})_i$  is the  $i$ -th column of the matrix square root.  $\lambda = \alpha^2(L+\kappa) - L$  is the scaling parameter;  $\alpha$  determines the spread of the sigma points around  $\hat{\mathbf{x}}_k$  and is usually set to a positive value not more than 1;  $\kappa$  is the second scaling parameter usually set to 0;  $\beta$  is used to incorporate prior knowledge of the distribution of  $\mathbf{x}$ ; for Gaussian distribution,  $\beta = 2$  is optimal (Wan and Van Der Merwe, 2000). In

the prediction step, each sigma point is propagated through

$$\mathcal{X}_{k+1,i} = F(\mathcal{X}_{k,i}) \quad (16)$$

The state and covariance prediction is derived from the weighted mean and covariance of the transformed sigma points

$$\hat{\mathbf{x}}_{k+1}^- = \sum_{i=0}^{2L} W_i^{(m)} \mathcal{X}_{k+1,i} \quad (17)$$

$$\mathbf{P}_{k+1}^- = \sum_{i=0}^{2L} W_i^{(c)} [\mathcal{X}_{k+1,i} - \hat{\mathbf{x}}_{k+1}^-] [\mathcal{X}_{k+1,i} - \hat{\mathbf{x}}_{k+1}^-]^T + \mathbf{Q}_k \quad (18)$$

In the correction step, the observations are predicted through

$$\mathcal{Y}_{k+1,i} = H(\mathcal{X}_{k+1,i}), i = 0, \dots, 2L \quad (19)$$

$$\hat{\mathbf{y}}_{k+1}^- = \sum_{i=0}^{2L} W_i^{(m)} \mathcal{Y}_{k+1,i} \quad (20)$$

Then the predicted state and covariance are corrected via

$$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_{k+1}^- + \mathbf{G}_{k+1} (\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1}^-) \quad (21)$$

$$\mathbf{P}_{k+1} = \mathbf{P}_{k+1}^- - \mathbf{G}_{k+1} \mathbf{P}_{\mathcal{Y}_k \mathcal{Y}_k} \mathbf{G}_{k+1}^T \quad (22)$$

where Kalman gain  $\mathbf{G}_{k+1}$  is

$$\mathbf{G}_{k+1} = \mathbf{P}_{\mathcal{X}_{k+1} \mathcal{Y}_{k+1}} \mathbf{P}_{\mathcal{Y}_{k+1} \mathcal{Y}_{k+1}}^{-1} \quad (23)$$

where

$$\mathbf{P}_{\mathcal{Y}_{k+1} \mathcal{Y}_{k+1}} = \sum_{i=0}^{2L} W_i^{(c)} [\mathcal{Y}_{k+1,i} - \hat{\mathbf{y}}_{k+1}^-] [\mathcal{Y}_{k+1,i} - \hat{\mathbf{y}}_{k+1}^-]^T + \mathbf{R}_{k+1} \quad (24)$$

$$\mathbf{P}_{\mathcal{X}_{k+1} \mathcal{Y}_{k+1}} = \sum_{i=0}^{2L} W_i^{(c)} [\mathcal{X}_{k+1,i} - \hat{\mathbf{x}}_{k+1}^-] [\mathcal{Y}_{k+1,i} - \hat{\mathbf{y}}_{k+1}^-]^T \quad (25)$$

Both UKF and EKF are based on the standard KF framework; they just differ in addressing the nonlinearities: UKF approximates the state probability distribution, while EKF approximates the nonlinear state space models. Theoretically, UKF outperforms EKF and they have equal computational complexity of  $O(L^3)$ . However, UKF needs to transform each sigma point through the process and observation functions at each prediction and update process, which may cost more time for high dimensional problems.

### 3. Experiments on vehicular state estimation and path prediction

#### 3.1. Experimental setup

The setup of the experiments is described in Figure 3. The experiments were implemented in C++. GPS, IMU and LiDAR are installed on the *Autoware* (Kato et al., 2015) platform to provide vehicle state measurements, including position, heading, velocity, acceleration and angular rate. The measurements are fused by the four estimators considering the kinematic models to generate vehicle state estimates. In the experiments, all the combinations of the most popular KF derivatives (EKF and UKF) and motion models (CTRV and CTRA) are investigated. In both theory and practice, UKF outperforms EKF (Wan

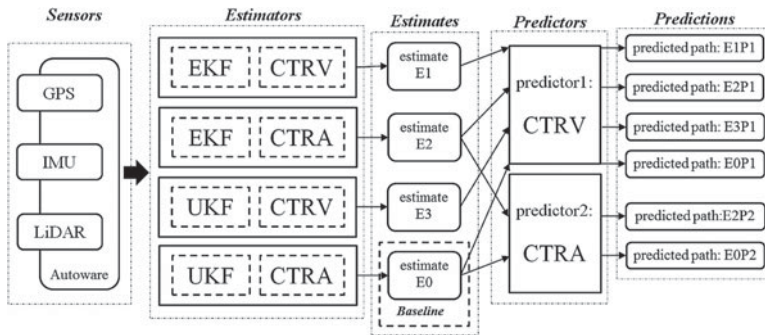


Figure 3. Experimental setup.

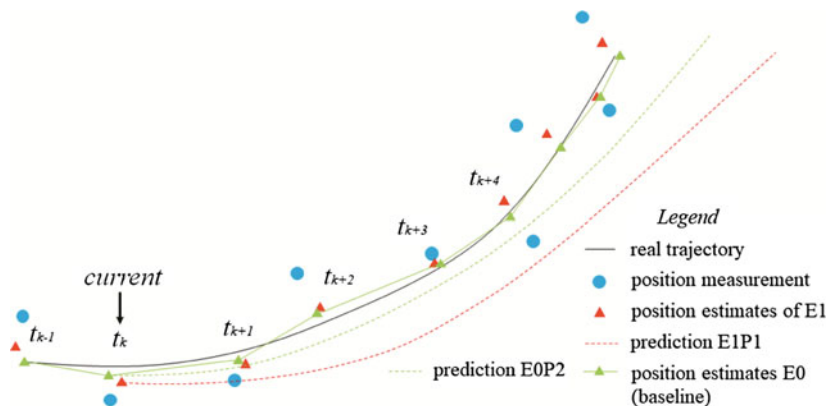


Figure 4. Relationship of measurement, estimate and prediction.

and Van Der Merwe, 2000; Julier, 2002; Yang et al., 2017); and the CTRA model outperforms other kinematic models (Schubert et al., 2008). Therefore, in this paper, the estimate (E0) of the UKF-CTRA filter is regarded as the baseline with which the other filters' estimates (E1, E2 and E3) are compared to find out the relative difference.

The predictors, CTRV and CTRA, generate 3 s-ahead paths using the estimated states and the paths' reliabilities are checked through comparison with the baseline. The notation of the path indicates the used state estimate and predictor. For example, E0P2 is generated by predictor2 based on estimate E0. The relationship of measurement, estimate and prediction is illustrated by Figure 4, where, for the sake of brevity, only some of them are drawn. The blue point represents the raw position measurement that inevitably contains noise. The KF is used to reduce the noise. As the figure illustrates, the position estimates (the triangles) are closer to the real trajectory. In practice, the real trajectory cannot be completely known but approximated, since the uncertainties cannot be removed completely. Based on different state estimates, different paths are predicted by the motion models solely and their reliabilities decrease over time. The efficiency of the estimator is also investigated through the mean computational time of each iteration.

### 3.1.1. Sensors of experimental vehicle

The experiments were carried out via the vehicle shown in Figure 5. Several kinds of sensors are mounted on the vehicle. In the experiments, LiDAR (Velodyne HDL-64ES3), IMU (Xsens MTi-300) and GNSS (global navigation satellite system) receiver (Trimble NetR9) were used. Only fixed GPS data were received in the test fields.





**Figure 5.** Test vehicle.

**Table 1.** Standard deviation of process noise.

Parameter	Value	Description
$w_x, w_y$	0.10 m	Position process noise
$w_\theta$	$3.16 \times 10^{-4}$ rad	Heading process noise
$w_v$	$3.16 \times 10^{-3}$ m/s	Velocity process noise
$w_a$	$3.16 \times 10^{-3}$ m/s <sup>2</sup>	Acceleration process noise
$w_\omega$	$3.16 \times 10^{-4}$ rad/s	Turn rate process noise

In these experiments, *Autoware* just generates raw vehicle state measurements within a unified coordinate system (some sample data, see [Appendix B](#)). The LiDAR measurements were generated at 10 Hz, the GPS at 20 Hz (in two drives, GPS worked at 10 Hz) and the IMU 100 Hz. In urban areas, the measurements of LiDAR and IMU were fused by the estimators; on the highway, GPS and IMU were used instead. In KF, the prediction process is performed at 100 Hz and the update process is performed as soon as a new measurement is available. The observation functions are presented in Equation (26), where the bold subscript  $\mathbf{x}$  and  $\mathbf{y}$  respectively indicate the system state and observation vector. As the equation expresses, all the states are observed directly, except the velocity and acceleration.

$$\begin{aligned} x_y = x_x \quad \theta_y = \theta_x \quad v_y^x = v_x \cos \theta_x \quad a_y^x = a_x \cos \theta_x \\ y_y = y_x \quad \omega_y = \omega_x \quad v_y^y = v_x \sin \theta_x \quad a_y^y = a_x \sin \theta_x \end{aligned} \quad (26)$$

### 3.1.2. Configurations of EKF and UKF

For EKF and UKF, it is necessary to select the appropriate process and observation noise covariance matrix  $\mathbf{Q}$  and  $\mathbf{R}$  and to start the algorithm with initial  $\hat{\mathbf{x}}_0$  and  $\mathbf{P}_0$ . In this paper,  $\mathbf{Q}$  and  $\mathbf{R}$  are considered to be constant. The setting of these parameters follows the approach proposed by Schneider and Georgakis (2013), and some of them are tuned empirically. Since  $\mathbf{Q}$  and  $\mathbf{R}$  are system and sensor dependent, they should be determined based on the user's system and sensors. The parameters used in the experiments are listed in [Tables 1](#) and [2](#). In the experiments, CTRV and CTRA use identical process noise parameters. The first aligned observation is assigned to  $\hat{\mathbf{x}}_0$  and  $\mathbf{P}_0$  is set by

$$\mathbf{P}_0 = \sigma^2 \mathbf{I} \quad (27)$$

In this paper  $\sigma = 10$ . In addition, the parameters  $\alpha$ ,  $\beta$  and  $\kappa$  of UKF are set to 0.1, 2 and 0, respectively.



Table 2. Standard deviation of observation noise.

Parameter	Value	Description
$e_{x\_lidar}, e_{y\_lidar}$	0.50 m	Position observation noise of LiDAR
$e_{x\_gps}, e_{y\_gps}$	3.00 m	Position observation noise of GPS
$e_{\theta\_lidar}$	$7.07 \times 10^{-3}$ rad	Heading observation noise of LiDAR
$e_{\theta\_gps}$	$4.47 \times 10^{-2}$ rad	Heading observation noise of GPS
$e_{vx\_lidar}, e_{vy\_lidar}$	$7.07 \times 10^{-2}$ m/s	Velocity observation noise of LiDAR
$e_{vx\_gps}, e_{vy\_gps}$	0.22 m/s	Velocity observation noise of GPS
$e_{ax}, e_{ay}$	0.80 m/s <sup>2</sup>	Acceleration observation noise
$e_{\omega}$	0.04 rad/s	Turn rate observation noise

Fig. 6 - Colour online. Colour in print

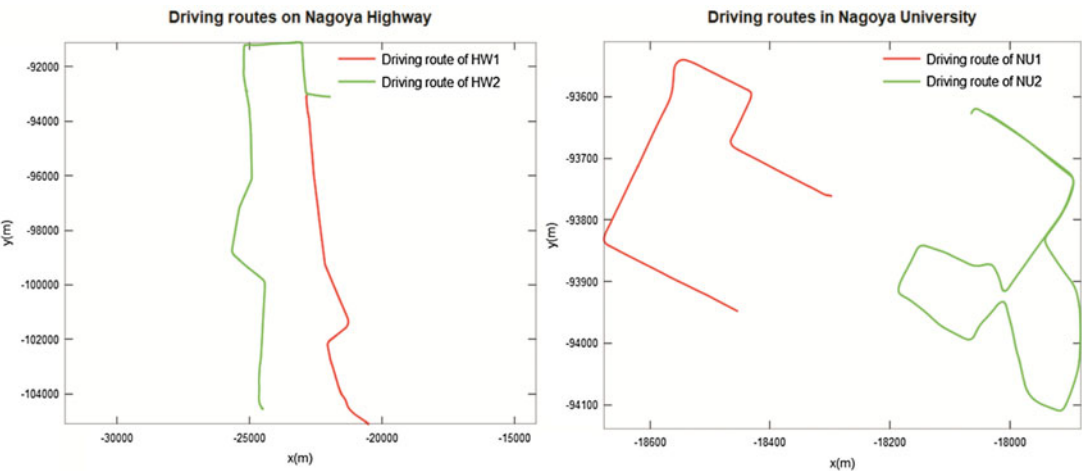


Figure 6. Driving routes of experiments.

3.2. Collected sensor data

Four typical routes in urban and highway scenarios were selected for the experiments presented in this paper. Figure 6 (left) shows the routes on Nagoya highway (HW) and Figure 6 (right) the routes in the Nagoya University campus (NU). In both the HW and NU experiments, drives were repeated three and four times, respectively. The statistics of all these drives are summarised in Tables 3 and 4. The total length of the driving routes was more than 105 km and the total driving time was nearly 2 h. In these drives, most of the commonest driving manoeuvres were performed: car following, overtaking and lane change on Nagoya highway; braking, stop, accelerating and turning on the campus of Nagoya University.

3.3. Comparative evaluation of the filters' performance in vehicular state estimation.

3.3.1. Relative difference

The most popular indicator for estimation accuracy is root-mean-square error (RMSE). In this paper, the relative difference evaluation was provided, instead of the absolute error evaluation. In the NU and HW experiments, the state estimates E1, E2 and E3 are respectively compared with the corresponding baseline E0 to assess their relative difference. (Both the baselines of NU and HW were generated by UKF-CTRA, integrating LiDAR-IMU and GPS-IMU respectively. The configurations are presented in

**Table 3.** Statistics of each drive at Nagoya University.

Drive	Length(m)	Duration	Mean(v) (m/s)	Std(v) (m/s)	Mean(a) (m/s <sup>2</sup> )	Std(a) (m/s <sup>2</sup> )
NU1-1	981.74	2 min 44 s	6.1264	2.2486	−0.053812	0.69399
NU1-2	974.05	2 min 50 s	5.7021	1.7541	−0.066964	0.49773
NU1-3	992.68	2 min 53 s	5.7723	2.0001	0.034385	0.62067
NU1-4	966.11	3 min 01 s	5.341	1.8286	−0.074118	0.48275
NU2-1	1,733.3	5 min 19 s	5.4971	2.0158	0.058261	0.47464
NU2-2	1,727.5	4 min 30 s	6.4593	2.4218	0.034729	0.58905
NU2-3	1,698.7	5 min 26 s	5.2544	2.2391	0.046688	0.51144
NU2-4	1,746.3	5 min 01 s	5.8214	2.6551	0.046192	0.62631
Total	10,820.38	31 min 44 s	—	—	—	—

**Table 4.** Statistics of each drive on Nagoya highway.

Drive	Length(m)	Duration	Mean(v) (m/s)	Std(v) (m/s)	Mean(a) (m/s <sup>2</sup> )	Std(a) (m/s <sup>2</sup> )
HW1-1	13,138.77	10 min 42 s	20.455	3.2057	−0.016844	0.22684
HW1-2	13,124.09	12 min 57 ses	16.841	2.5291	−0.07341	0.22356
HW1-3	13,122.98	12 min 24 s	17.67	2.2747	0.0094033	0.181
HW2-1	19,010.99	16 min 44 s	18.942	2.2481	0.028385	0.2221
HW2-2	17,794.26	15 min 35 s	19.025	2.1233	0.059228	0.22881
HW2-3	19,253.90	17 min 42 s	18.121	2.428	0.033483	0.257
Total	95,444.98	86 min 04 s	—	—	—	—

Section 3.1.2.) The relative RMSE (RRMSE) is formulised as

$$\text{RRMSE}(x) = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - x_i^{\text{baseline}})^2} \quad (28)$$

The mean RRMSEs of all the state estimates of each route are presented in Tables 5–8. From the tables, it can be found that the four filters have not made a significant difference to state estimation (not positioning). UKF and the EKF have roughly identical performance in contrast to the simulation-based experiments where UKF shows a distinct improvement of accuracy (Tsogas et al., 2005; Yang et al., 2017).

A trend can be found in position estimation that E2 has the smallest relative difference, then E3 and E1 has the largest relative difference, with respect to E0. This matches the conclusions of the existing literature that UKF outperforms EKF (Yang et al., 2017) and CTRA based filters make slightly better position estimates than CTRV based filters (Schubert et al., 2008, 2011). The heading and turn rate estimates of the four filters appear to be exactly the same, which is due to the CTRV and CTRA models having the same constant turn rate hypotheses.

Compared with other state estimates, the velocity estimates of CTRA and CTRV based filters produced relatively obvious differences, in both highway and urban scenarios. The reason is the different velocity assumptions of the two models. The CTRA model takes acceleration into account and the velocity is driven by acceleration; however, the CTRV model assumes the velocity remains constant. In velocity alone, it can be observed that the relative difference of the velocity estimate in urban driving is larger than that of highway driving, which is caused by different driving styles on highway

Table 5. The mean RRMSE of HW1.

Model	CTRV		CTRA	
	EKF	UKF	EKF	UKF
Estimate	E1	E3	E2	E0
$x$ (m)	0.02094	0.01884	0.03841	0
$y$ (m)	0.03227	0.03002	0.01225	0
$\theta$ (deg)	0.00289	0.00108	0.00523	0
$v$ (m/s)	0.11288	0.11288	0.00044	0
$a$ (m/s <sup>2</sup> )	–	–	0.00041	0
$\omega$ (deg/s)	0.00177	0.00010	0.00278	0

Table 6. Mean RRMSE of HW2.

Model	CTRV		CTRA	
	EKF	UKF	EKF	UKF
Estimate	E1	E3	E2	E0
$x$ (m)	0.05695	0.05487	0.03683	0
$y$ (m)	0.05904	0.05671	0.05054	0
$\theta$ (deg)	0.00369	0.00148	0.00532	0
$v$ (m/s)	0.15800	0.15799	0.00157	0
$a$ (m/s <sup>2</sup> )	–	–	0.00082	0
$\omega$ (deg/s)	0.00291	0.00017	0.00339	0

Table 7. Mean RRMSE of NU1.

Model	CTRV		CTRA	
	EKF	UKF	EKF	UKF
Estimate	E1	E3	E2	E0
$x$ (m)	0.02355	0.01850	0.01011	0
$y$ (m)	0.02975	0.02224	0.01493	0
$\theta$ (deg)	0.00883	0.00027	0.00877	0
$v$ (m/s)	0.34869	0.34869	0.00045	0
$a$ (m/s <sup>2</sup> )	–	–	0.00192	0
$\omega$ (deg/s)	0.00749	0.00011	0.00751	0

and urban routes. On the highway, drivers prefer to drive the car at an even speed most of the time; whereas, in urban areas, many accelerating/decelerating manoeuvres have to be performed, especially at road intersections. That is why acceleration in urban driving shows higher standard deviations than highway driving (Tables 3 and 4). The changes in acceleration result in the greater relative difference in velocity estimation. In low dynamic situations (Tables 5 and 6), the performance of the two models is approximately the same.

The overall results indicate that the motion model has more effect on the filter’s performance than the KF form; with the same process model, the accuracy performance of EKF and UKF is almost identical.

**Table 8.** Mean RRMSE of NU2.

Model	CTRV		CTRA	
	EKF	UKF	EKF	UKF
Estimate	E1	E3	E2	E0
$x$ (m)	0.02471	0.01833	0.01794	0
$y$ (m)	0.02887	0.02144	0.01475	0
$\theta$ (deg)	0.01280	0.00042	0.01281	0
$v$ (m/s)	0.37153	0.37152	0.00053	0
$a$ (m/s <sup>2</sup> )	–	–	0.00234	0
$\omega$ (deg/s)	0.00529	0.00018	0.00549	0

**Table 9.** Average time cost per iteration of HW1 (GPS@20 Hz).

HW1	EKF	UKF	UKF/EKF
CTRV	9.56097	13.11299	1.37151
CTRA	12.86629	24.83430	1.93018
CTRA/CTRV	1.34571	1.89387	2.59747

**Table 10.** Average time cost per iteration of HW2 (GPS@20 Hz).

HW2	EKF	UKF	UKF/EKF
CTRV	9.66745	12.99528	1.34423
CTRA	13.21840	25.25782	1.91081
CTRA/CTRV	1.36731	1.94362	2.61267

In high dynamic situations and where the speed estimate is critical, the CTRA model based filter is recommended. Otherwise, the performance of the CTRV model based filter is also acceptable.

The negligible estimation differences among the four filters are explained by the non-severe nonlinearities of the CTRV and CTRA models, which mean that the linearisation in EKF does not cause much information loss. Besides, the low dynamics of the vehicle and the high sampling rate of the sensor make the vehicular motion quasi-linear. UKF thus does not have obvious superiority.

### 3.3.2. Time efficiency

Limited by the space of this paper, in this section, the results of HW1-1 and HW2-1 drives, where GPS worked at 10 Hz, are not presented. In the experiments, when using LiDAR and IMU, a complete iteration consists of 10 predictions, 10 IMU updates and one LiDAR update; for GPS and IMU, a complete iteration consists of five predictions, five IMU updates and one GPS update. This is done by using an ROS (Robot Operating System) time synchroniser (for more information, please see [http://wiki.ros.org/message\\_filters](http://wiki.ros.org/message_filters)). The experiments were run on a normal personal computer (Inter Core i7-4790 CPU @ 3.6 GHz, RAM 8G, Windows 10). Each drive's experiment was repeated 10 times. The running time of each estimator was recorded and the average time cost (microsecond) per iteration was calculated and shown in Tables 9–12.

In Tables 9–12, each row records the time cost of EKF and UKF, using the same model; and each column is the time cost of the filter using identical KF form and different models. The UKF/EKF

**Table 11.** Average time cost per iteration of NU1 (LiDAR@10 Hz).

NU1	EKF	UKF	UKF/EKF
CTRV	16.14401	21.36358	1.32331
CTRA	22.95781	42.59849	1.85551
CTRA/CTRV	1.42206	1.99398	2.63866

**Table 12.** Average time cost per iteration of NU2 (LiDAR@10 Hz).

NU2	EKF	UKF	UKF/EKF
CTRV	16.84575	22.54865	1.33854
CTRA	23.85105	44.92495	1.88356
CTRA/CTRV	1.41585	1.99236	2.66684

column records the time cost rate of UKF and EKF with the same model, and the CTRA/CTRV row records the time cost rate of the filters using identical KF form and different models. In particular, the bottom-right cell is the time cost rate of the slowest (UKF-CTRA) and the fastest filters (EKF-CTRV). A clear conclusion can be drawn that, when utilising the same model, EKF is faster than UKF; and when utilising the same KF form, the filter using CTRV is faster than the filter using CTRA. For a higher dimension model, EKF is obviously faster than UKF since UKF has to calculate  $2L + 1$  sigma points by taking a matrix square root of the state covariance matrix at each iteration and propagating these sigma points at the prediction and update steps.

In Tables 9 and 10, the average time cost of each filter is shorter than that in Tables 11 and 12. This is caused by the higher update frequency of GPS (20 Hz) compared with LiDAR (10 Hz). Despite that, the UKF/EKF and CTRA/CTRV rates are approximately equal.

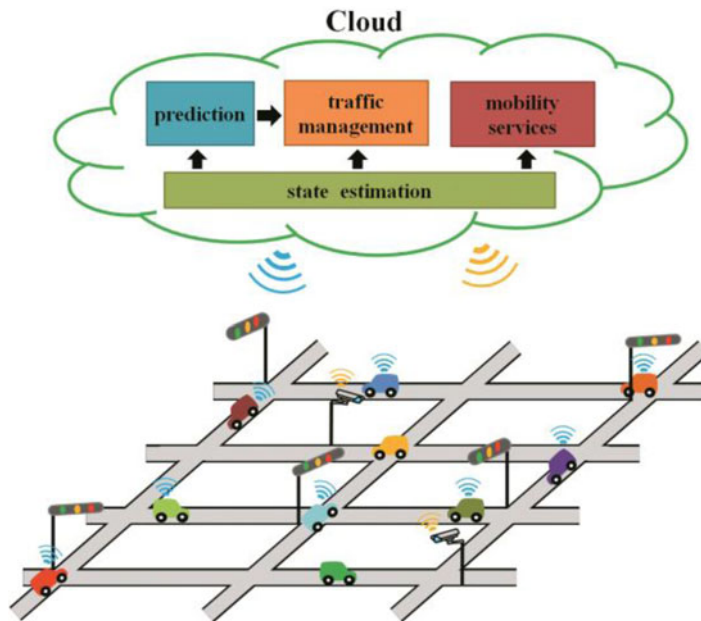
It is remarkable that the EKF-CTRV is about 2.6 times faster than the UKF-CTRA while their accuracies are almost equal (see Tables 5–8). EKF-CTRV may not make a significant improvement in efficiency in most current applications, where just the states of several vehicles need to be estimated. When the amount of vehicles increases substantially, however, it becomes another situation, such as SENSORIS (SENSORIS, 2017) where data is exchanged between the vehicle sensors and a dedicated cloud. As Figure 7 illustrates, the data from vehicles and roadside sensors are gathered into the cloud, where vehicle state estimation is essential for mobility related applications. Such a system has urgent need of real-time processing. In this situation, the EKF-CTRV filter will show its distinguished performance. For example, in the same amount of time, the EKF-CTRV can process information from nearly 2,600 vehicles; however, the UKF-CTRA can only process information from 1,000 vehicles. The efficiency of the system can be greatly improved, with negligible accuracy loss.

In summary, for a specific application, an ideal filter is composed of the most appropriate motion model and KF form, taking both accuracy and efficiency into consideration. In vehicular state estimation, EKF and UKF have not made distinct differences in accuracy; however, in the efficiency aspect, EKF is faster than UKF. The state estimation performance of the CTRV and CTRA model based filters is roughly identical.

### 3.4. Comparison of the models' performance in path prediction

To investigate the models' properties and the affecting factors in path prediction, the predicted path is defined by Equation (29).

$$\text{path} = \{\mathbf{p}_{t_0}, \mathbf{p}_{t_1}, \dots, \mathbf{p}_{t_n}\}, \text{ where } \mathbf{p}_{t_i} = [x, y]^T = \begin{bmatrix} 1 & 0 & \mathbf{0} \\ 0 & 1 & \mathbf{0} \end{bmatrix} F(\mathbf{x}_{t_{i-1}}), t_i \in \{0, 0.1, \dots, 3\} \quad (29)$$



**Figure 7.** Dedicated cloud for vehicle and road side sensor.

where the path consists of sequential position points predicted by the kinematic model  $F$  (CTRV or CTRA). Given an initial state  $\mathbf{x}_{t_0}$ , a path can be generated iteratively via Equation (29).

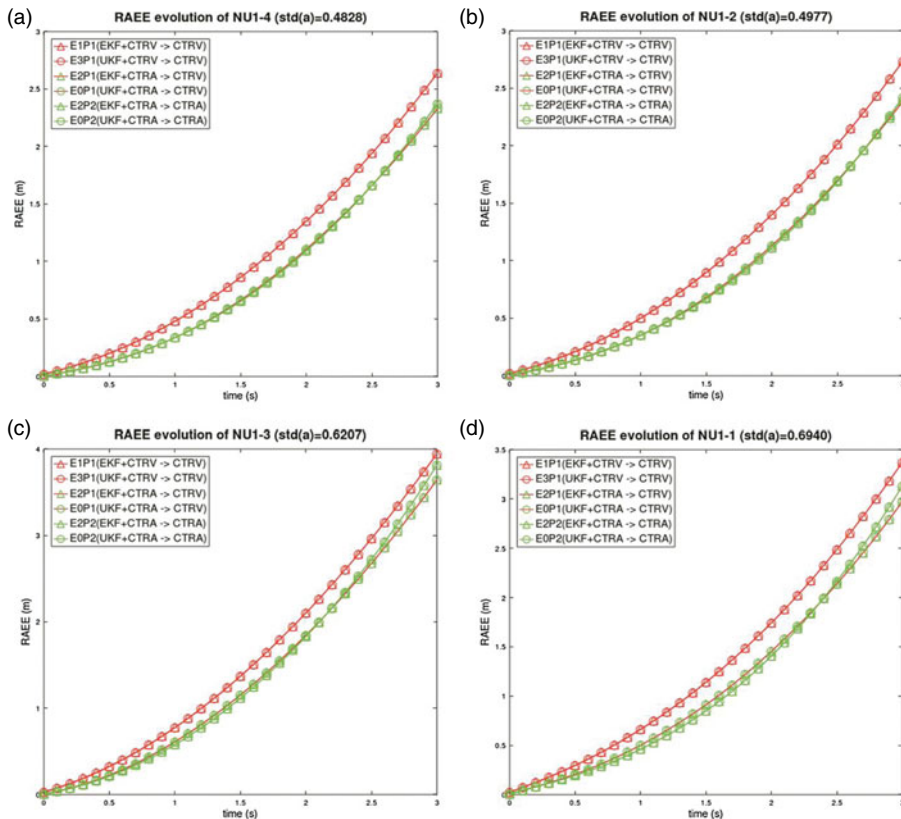
The relative average Euclidean error (RAEE) at instant  $t_i$  of a certain *PATH* defined in Equation (30) is used to measure the path's reliability evolution, where  $path_j$  denotes the  $j$ -th predicted path;  $N$  is the number of predicted paths and  $PATH \in \{E1P1, E3P1, E2P1, E0P1, E2P2, E0P2\}$ .

$$RAEE(t_i, PATH) = \frac{1}{N} \sum_{j=1}^N \|\mathbf{p}_{t_i}^{path_j} - \mathbf{p}_{t_i}^{baseline}\|_2, \mathbf{p}_{t_i}^{path_j} \in path_j, path_j \in PATH \quad (30)$$

Since the baseline estimated based on GPS measurements still retained residuals of several metres, for reasons of caution, the path prediction experiment was not conducted on the highway drive. In contrast, LiDAR provided sub-metre positioning results, at least (Liu et al., 2020). In this section, just the experiments at Nagoya University are discussed.

The evolution of RAEE over the time of each drive on routes NU1 and NU2 is presented in Figures 8 and 9, respectively. In the figures, the RAEE evolution of a certain *PATH* is represented by a marked line with different colour. The red and green lines indicate CTRV and CTRA based prediction, respectively. The triangular and circular marks, respectively, indicate that the EKF or UKF estimate is used in the prediction. The colour of the mark indicates the process model used in state estimation; red is CTRV and green is CTRA.

In Figures 8 and 9, there are three outstanding characteristics: (1) the red lines with red marks have the largest RAEE, which means the paths predicted by CTRV using the state estimates of CTRV-based filters have the largest relative difference; (2) the triangular and circular marks are superposed, which means the KF form (EKF or UKF) barely affects the path prediction; (3) all the RAEEs increase exponentially over time. It can be inferred that (1) is caused by the different initial velocity estimates because the red lines with green marks have the lowest RAEE and the only obvious differences between them are the initial velocities, see Tables 7 and 8. The reason for (2) is the negligible estimate difference between identical-model based EKF and UKF, see Tables 7 and 8. The cause of (3) is that the prediction, in fact, is an open-loop KF without any updates, thus the prediction becomes divergent over time.

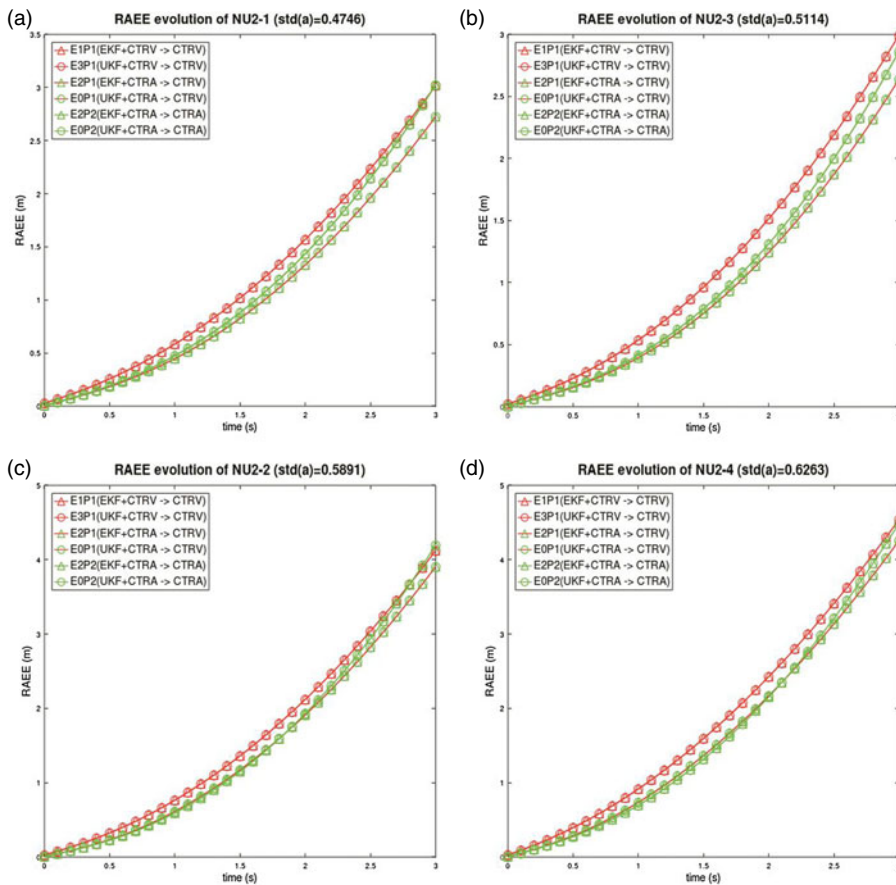


**Figure 8.** RAEE evolution over time of each drive on route NU1.

It can be observed in NU1-4 and NU1-2 drives [Figures 8(a) and 8(b)] that the CTRV and CTRA models have almost identical performance. This is because the acceleration was steady in the drives [low standard deviation of acceleration ( $\text{std-a}$ )] which results in low standard deviation of velocity ( $\text{std-v}$ ), see Table 3. In such condition, the vehicle was likely to move uniformly; therefore, the two models produced almost identical predictions. However, with low  $\text{std-a}$ , in NU2-1 and NU2-3 drives [Figures 9(a) and 9(b)], CTRV predicted better than CTRA. The reasons are associated with the road conditions. The NU2 route is the most curved and sloped, so it is harder for the driver to keep a constant acceleration than a constant velocity on NU2. Therefore, the constant velocity hypothesis of CTRV is more realistic than the constant acceleration hypothesis of CTRA. It can be summarised that in low  $\text{std-a}$  situations (low dynamics) CTRV model predicts more reliably. With  $\text{std-a}$  increasing, RAEE of the CTRA prediction decreased with respect to that of CTRV at the beginning (Figures 8 and 9, (c) and (d)); however, RAEE of the CTRA prediction became larger than that of CTRV about 2 s later. This reveals the fact that continuous long-term accelerating is not permitted on the Nagoya University campus. It can be inferred that the CTRA model predicts more reliably when the vehicle has high dynamics [Figures 8(d) and 9(d)].

From the experiments, some conclusions can be drawn. Firstly, the reliability of predicted path decreases over time; a standalone model should not be used for long-term prediction. Secondly, velocity estimate affects path prediction. Thirdly, the model that matches with the vehicle's driving status predicts better; the prediction model should be chosen dynamically considering the vehicle's behaviour, as done by Lytrivis et al. (2011).





**Figure 9.** RAEE evolution over time of each drive on route NU2.

#### 4. Conclusions

This paper has been devoted to comparing the accuracy and efficiency performance of UKF and EKF with different motion models, in vehicular state estimation. The models' properties and the factors affecting path prediction have also been investigated. UKF and EKF show roughly the same accuracy; the only obvious difference occurs in velocity estimation which is caused by different velocity hypotheses of the CTRA and CTRV models. However, they differed significantly in efficiency. With an identical process model, EKF works faster than UKF; with identical KF form, the filter using CTRV is faster than that using CTRA. The fastest filter, EKF-CTRV, is about 2-6 times faster than the slowest, UKF-CTRA. For the application that needs to process mass data with strict real-time demands, EKF-CTRV might be an ideal choice. The velocity estimate and the motion model used affect the reliability in path prediction. A realistic model that reflects the real driving status generates a reliable path. A standalone motion model is not recommended for performing long-term prediction; some additional information should be taken into account, such as the road geometry and topology.

The experiment in this paper was conducted according to urban and highway scenarios. We can also consider more complex scenarios with sensor problems: multipath effect and signal blockage in GPS, drifting in IMU, and LiDAR's sensitivity to the weather. Experiments with UKF and EKF with these factors will be included in the authors' future work.

**Acknowledgements.** This work was supported by JST-OPERA Programme, Grant Number JPMJOP1612. The support provided by the China Scholarship Council (CSC) during the studies of Lu Tao at Nagoya University are also acknowledged.

## References

- Gelb, A. (1974). *Applied Optimal Estimation*. Cambridge, MA: MIT Press.
- Houenou, A., Bonnifait, P., Cherfaoui, V. and Yao, W. (2013). Vehicle Trajectory Prediction Based on Motion Model and Maneuver Recognition. *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, pp. 4363–4369.
- Julier, S. J. (2002). The scaled unscented transformation. *IEEE*, **6**, 4555–4559. doi:10.1109/ACC.2002.1025369
- Julier, S. J. and Durrant-Whyte, H. F. (2003). On the role of process models in autonomous land vehicle navigation systems. *IEEE Transactions on Robotics and Automation*, **19**, 1–14. doi:10.1109/TRA.2002.805661
- Julier, S. J. and Uhlmann, J. K. (1997). A New Extension of the Kalman Filter to Nonlinear Systems. *Proc. SPIE 3068. Signal Processing, Sensor Fusion, and Target Recognition, VI*. International Society for Optics and Photonics, pp. 182–194.
- Julier, S. J., Uhlmann, J. K. and Durrant-Whyte, H. F. (1995). A New Approach for Filtering Nonlinear Systems. *Procs. 1995 American Control Conference - ACC '95*, **3**, pp. 1628–1632. doi:10.1109/ACC.1995.529783
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, **82**, 35–45. doi:10.1115/1.3662552
- Karagiannis, G., Altintas, O., Ekici, E., Heijenk, G., Jarupan, B., Lin, K. and Weil, T. (2011). Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions. *IEEE Communications Surveys & Tutorials*, **13**, 584–616. doi:10.1109/SURV.2011.061411.00019
- Kato, S., Takeuchi, E., Ishiguro, Y., Ninomiya, Y., Takeda, K. and Hamada, T. (2015). An open approach to autonomous vehicles. *IEEE Micro*, **35**, 60–68. doi:10.1109/MM.2015.133
- Lefebvre, T., Bruyninckx, H., De Schutter, J. (2004). Kalman filters for nonlinear systems: A comparison of performance. *International Journal of Control*, **77**, 639–653.
- Lefevre, S., Vasquez, D. and Laugier, C. (2014). A survey on motion prediction and risk assessment for intelligent vehicles. *Robomech Journal*, **1**, 1.
- Liu, R., Wang, J. and Zhang, B. (2020). High definition map for automated driving: Overview and analysis. *The Journal of Navigation*, **73**, 324–341. doi:10.1017/S0373463319000638
- Lytrivis, P., Thomaidis, G., Tsogas, M. and Amditis, A. (2011). An advanced cooperative path prediction algorithm for safety applications in vehicular networks. *IEEE Transactions on Intelligent Transportation Systems*, **12**, 669–679. doi:10.1109/TITS.2011.2123096
- Madhavan, R., Kootbally, Z. and Schlenoff, C. (2006). Prediction in Dynamic Environments for Autonomous On-Road Driving. *Control, Automation, Robotics and Vision, 2006. ICARCV '06. 9th International Conference On*, IEEE, pp. 1–6.
- Ndjeng Ndjeng, A., Lambert, A., Gruyer, D. and Glaser, S. (2009). Experimental Comparison of Kalman Filters for Vehicle Localization. *2009 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, Xi'an, China, pp. 441–446. doi:10.1109/IVS.2009.5164318
- Polychronopoulos, A., Tsogas, M., Amditis, A. J. and Andreone, L. (2007). Sensor fusion for predicting vehicles' path for collision avoidance systems. *IEEE Transactions on Intelligent Transportation Systems*, **8**, 549–562. doi:10.1109/TITS.2007.903439
- Schneider, R. and Georgakis, C. (2013). How to not make the extended Kalman filter fail. *Industrial & Engineering Chemistry Research*, **52**, 3354–3362. doi:10.1021/ie300415d
- Schubert, R., Richter, E. and Wanielik, G. (2008). Comparison and Evaluation of Advanced Motion Models for Vehicle Tracking. *2008 11th International Conference on Information Fusion*, IEEE, pp. 1–6.
- Schubert, R., Adam, C., Obst, M., Mattern, N., Leonhardt, V. and Wanielik, G. (2011). Empirical Evaluation of Vehicular Models for Ego Motion Estimation. *2011 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, Baden-Baden, Germany, pp. 534–539. doi:10.1109/IVS.2011.5940526
- SENSORIS (2019). <https://sensor-is.org/> (accessed 29 May 2019).
- St-Pierre, M. and Gingras, D. (2004). Comparison Between the Unscented Kalman Filter and the Extended Kalman Filter for the Position Estimation Module of an Integrated Navigation Information System. *IEEE Intelligent Vehicles Symposium*, Citeseer, pp. 831–835.
- Tan, H.-S. and Huang, J. (2006). DGPS-based vehicle-to-vehicle cooperative collision warning: Engineering feasibility viewpoints. *IEEE Transactions on Intelligent Transportation Systems*, **7**, 415–428. doi:10.1109/TITS.2006.883938
- Tsogas, M., Polychronopoulos, A. and Amditis, A. (2005). Unscented Kalman Filter Design for Curvilinear Motion Models Suitable for Automotive Safety Applications. *2005 7th International Conference on Information Fusion. Presented at the 2005 7th International Conference on Information Fusion*, IEEE, Philadelphia, PA, USA, p. 8. doi:10.1109/ICIF.2005.1592006
- Wan, E. A. and Van Der Merwe, R. (2000). *The Unscented Kalman Filter for Nonlinear Estimation*. IEEE, 153–158. doi:10.1109/ASSPCC.2000.882463
- Watanabe, Y., Sato, K. and Takada, H. (2018). Dynamicmap 2.0: A traffic data management platform leveraging clouds, edges and embedded systems. *International Journal of Intelligent Transportation Systems Research*, doi:10.1007/s13177-018-0173-7
- Weiß, C. (2011). V2x communication in Europe – from research projects towards standardization and field testing of vehicle communication technology. *Computer Networks*, Deploying vehicle-2-x communication, **55**, 3103–3119. doi:10.1016/j.comnet.2011.03.016
- Yang, C., Shi, W. and Chen, W. (2017). Comparison of unscented and extended kalman filters with application in vehicle navigation. *The Journal of Navigation*, **70**, 411–431. doi:10.1017/S0373463316000655

## APPENDIX A: JACOBIAN MATRICE OF PROCESS AND OBSERVATION MODELS

For CTRV model, suppose the state transition function is

$$\mathbf{x}_{k+1} = F(\mathbf{x}_k) = [f_x(\mathbf{x}_k) \ f_y(\mathbf{x}_k) \ f_\theta(\mathbf{x}_k) \ f_v(\mathbf{x}_k) \ f_\omega(\mathbf{x}_k)]^T \quad (\text{A1})$$

Then, the Jacobian matrix of CTRV model is as follows.

$$\mathbf{J}_F = \begin{bmatrix} 1 & 0 & \frac{\partial f_x}{\partial \theta} & \frac{\partial f_x}{\partial v} & \frac{\partial f_x}{\partial \omega} \\ 0 & 1 & \frac{\partial f_y}{\partial \theta} & \frac{\partial f_y}{\partial v} & \frac{\partial f_y}{\partial \omega} \\ 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A2})$$

where

$$\frac{\partial f_x}{\partial \theta} = \frac{v_v}{\omega_k} [\cos(\theta_k + \omega_k T) - \cos(\theta_k)] \quad (\text{A3})$$

$$\frac{\partial f_x}{\partial v} = \frac{1}{\omega_k} [\sin(\theta_k + \omega_k T) - \sin(\theta_k)] \quad (\text{A4})$$

$$\frac{\partial f_x}{\partial \omega} = \frac{-v_k}{\omega_k^2} [\sin(\theta_k + \omega_k T) - \sin(\theta_k)] + \frac{v_v}{\omega_k} \cos(\theta_k + \omega_k T)T \quad (\text{A5})$$

$$\frac{\partial f_y}{\partial \theta} = \frac{v_v}{\omega_k} [\sin(\theta_k + \omega_k T) - \sin(\theta_k)] \quad (\text{A6})$$

$$\frac{\partial f_y}{\partial v} = \frac{1}{\omega_k} [-\cos(\theta_k + \omega_k T) + \cos(\theta_k)] \quad (\text{A7})$$

$$\frac{\partial f_y}{\partial \omega} = \frac{v_k}{\omega_k^2} [\cos(\theta_k + \omega_k T) - \cos(\theta_k)] + \frac{v_v}{\omega_k} \sin(\theta_k + \omega_k T)T \quad (\text{A8})$$

For CTRA model, suppose the state transition function is

$$\mathbf{x}_{k+1} = F(\mathbf{x}_k) = [f_x(\mathbf{x}_k) \ f_y(\mathbf{x}_k) \ f_\theta(\mathbf{x}_k) \ f_v(\mathbf{x}_k) \ f_a(\mathbf{x}_k) \ f_\omega(\mathbf{x}_k)]^T \quad (\text{A9})$$

Then, the Jacobian matrix of CTRA model is as follows.

$$\mathbf{J}_F = \begin{bmatrix} 1 & 0 & \frac{\partial f_x}{\partial \theta} & \frac{\partial f_x}{\partial v} & \frac{\partial f_x}{\partial a} & \frac{\partial f_x}{\partial \omega} \\ 0 & 1 & \frac{\partial f_y}{\partial \theta} & \frac{\partial f_y}{\partial v} & \frac{\partial f_y}{\partial a} & \frac{\partial f_y}{\partial \omega} \\ 0 & 0 & 1 & 0 & 0 & T \\ 0 & 0 & 0 & 1 & T & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A10})$$

where

$$\frac{\partial f_x}{\partial \theta} = \frac{1}{\omega_k} [(v_k + a_k T) \cos(\theta_k + \omega_k T) - v_k \cos(\theta_k)] - \frac{a_k}{\omega_k^2} [\sin(\theta_k + \omega_k T) - \sin(\theta_k)] \quad (\text{A11})$$

$$\frac{\partial f_x}{\partial v} = \frac{1}{\omega_k} [\sin(\theta_k + \omega_k T) - \sin(\theta_k)] \quad (\text{A12})$$

$$\frac{\partial f_x}{\partial a} = \frac{1}{\omega_k} [T \sin(\theta_k + \omega_k T)] + \frac{1}{\omega_k^2} [\cos(\theta_k + \omega_k T) - \cos(\theta_k)] \quad (\text{A13})$$

$$\begin{aligned} \frac{\partial f_x}{\partial \omega} = & \frac{1}{\omega_k} [(v_k + a_k T) \cos(\theta_k + \omega_k T)T] - \frac{1}{\omega_k^2} [(v_k + 2a_k T) \sin(\theta_k + \omega_k T) \\ & - v_k \sin(\theta_k)] - \frac{2a_k}{\omega_k^3} [\cos(\theta_k + \omega_k T) - \cos(\theta_k)] \end{aligned} \quad (\text{A14})$$

$$\frac{\partial f_y}{\partial \theta} = \frac{1}{\omega_k} [(v_k + a_k T) \sin(\theta_k + \omega_k T) - v_k \sin(\theta_k)] + \frac{a_k}{\omega_k^2} [\cos(\theta_k + \omega_k T) - \cos(\theta_k)] \quad (\text{A15})$$

$$\frac{\partial f_y}{\partial v} = \frac{1}{\omega_k} [-\cos(\theta_k + \omega_k T) + \cos(\theta_k)] \quad (\text{A16})$$

$$\frac{\partial f_y}{\partial a} = \frac{-1}{\omega_k} [T \cos(\theta_k + \omega_k T)] + \frac{1}{\omega_k^2} [\sin(\theta_k + \omega_k T) - \sin(\theta_k)] \quad (\text{A17})$$

$$\begin{aligned} \frac{\partial f_y}{\partial \omega} = & \frac{1}{\omega_k} [(v_k + a_k T) \sin(\theta_k + \omega_k T)T] + \frac{1}{\omega_k^2} [(v_k + 2a_k T) \cos(\theta_k + \omega_k T) \\ & - v_k \cos(\theta_k)] - \frac{2a_k}{\omega_k^3} [\sin(\theta_k + \omega_k T) - \sin(\theta_k)] \end{aligned} \quad (\text{A18})$$

For the directly observed measurements: position, heading and turn rate, their measurement models' Jacobian matrices are

$$\mathbf{J}_H = \frac{\partial H}{\partial \mathbf{x}} = \mathbf{I} \quad (\text{A19})$$

The velocity measurement models' Jacobian matrices are

$$CTRV : \mathbf{J}_{H_v} = \frac{\partial H_v}{\partial \mathbf{x}} = \begin{bmatrix} 0 & 0 & -v \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & v \cos(\theta) & \sin(\theta) & 0 \end{bmatrix} \quad (\text{A20})$$

$$CTRA : \mathbf{J}_{H_v} = \frac{\partial H_v}{\partial \mathbf{x}} = \begin{bmatrix} 0 & 0 & -v \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & v \cos(\theta) & \sin(\theta) & 0 & 0 \end{bmatrix} \quad (\text{A21})$$

And acceleration measurement model' Jacobian matrix is

$$\mathbf{J}_{H_a} = \frac{\partial H_a}{\partial \mathbf{x}} = \begin{bmatrix} 0 & 0 & -a \sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & a \cos(\theta) & 0 & \sin(\theta) & 0 \end{bmatrix} \quad (\text{A22})$$

## APPENDIX B: SAMPLE DATA

The sample data from LiDAR, GPS and IMU generated by *Autoware* in this paper are available at: <https://github.com/LuTaonuwhu/LiDAR-GPS-IMU-Data.git>.