Journal of Physics: Conference Series

PAPER • OPEN ACCESS

A benchmark study on the model-based estimation of the go-kart side-slip angle

To cite this article: M D’Inverno *et al* 2021 *J. Phys.: Conf. Ser.* **2090** 012156

A benchmark study on the model-based estimation of the go-kart side-slip angle

M D'Inverno[1], V M Arricale[1], A Zanardi[2], E Frazzoli[2], A Sakhnevych[1], F Timpone[1]

[1] Dept. of Industrial Engineering, University of Naples "Federico II", Naples, Italy
[2] Institute for Dynamic Systems and Control, ETH Zurich, Zurich, Switzerland

E-mail: `vincenzomaria.arricale@unina.it`

Abstract. Nowadays, the active safety systems that control the dynamics of passenger cars usually rely on real-time monitoring of vehicle side-slip angle (VSA). The VSA can't be measured directly on the production vehicles since it requires the employment of high-end and expensive instrumentation. To realiably overcome the VSA estimation problem, different model-based techniques can be adopted. The aim of this work is to compare the performance of different model-based state estimators, evaluating both the estimation accuracy and the computational cost, required by each algorithm. To this purpose Extended Kalman Filters, Unscented Kalman Filters and Particle Filters have been implemented for the vehicle system under analysis. The physical representation of the process is represented by a single-track vehicle model adopting a simplified Pacejka tyre model. The results numerical results are then compared to the experimental data acquired within a specifically designed testing campaign, able to explore the entire vehicle dynamic range. To this aim an electric go-kart has been employed as a vehicle, equipped with steering wheel encoder, wheels angular speed encoder and IMU, while an S-motion has been adopted for the measurement of the experimental VSA quantity.

Keywords: Kalman filters, Particle filters, vehicle state estimation, vehicle dynamics modeling, side-slip angle

1. Introduction

The typical active safety systems that control the dynamics of passenger cars rely on real-time monitoring of vehicle side-slip angle (VSA), but the VSA is not measured directly because it requires the use of high-end instruments, which usually cannot be equipped in the passenger cars due to the significant related costs and bulky instrumentation [1]. However, this is not the only application field, indeed an accurate knowledge of the VSA may improve the ADAS system or may be used to improve the trajectory in autonomous vehicle [2]. In the motorsport field, the application of the VSA is used to enhance the overall vehicle performance during the race. In all these application fields the VSA estimation is increasingly diffused, and it is evaluated employing different measurements available onboard, such as wheel velocities, linear and angular accelerations [3] [4].

The technical literature is plenty of articles about VSA estimation. There are different approaches to estimate the VSA starting from the observer-based methods [5], [6], [7] up to neural network data-based techniques [8], [9]. The observer-based methods are characterized by the type of state estimator adopted, e.g. Extended Kalman filters [10] [11], in which the

Jacobian matrix computation is required and the nonlinear problem is linearized through a Taylor expansion. In [12] and [13], Unscented Kalman filters are adopted. The application of such filters is widely used due to an easier implementation because they do not require the computation (analytical or numerical) of the Jacobian matrix. In this work, another type of state estimator is considered in addition to the ones mentioned above, consisting in the Particle filter. The main applications of this latter involve the tracking problems, as reported in [14].

All the filters are based on the mathematical modeling of the process to estimate, a mathematical representation of the vehicle is required in this application. Different assumption have to be stated in order to model the vehicle dynamics. Two kinds of vehicle models can be found in the literature, which are denoted respectively as kinematic and dynamic [15]. The kinematic model is concerned with the vehicle motion with no reference to forces; thus, it does not need complex parameters such as those regarding tyres, which often are the cause of the non-linearity of the vehicle model. However, the main issue of VSA estimation using a kinematic vehicle model lays in the fact that it does not work when the vehicle yaw rate is relatively small or zero, this leads to the system unobservability, as reported in [16]. The dynamic model, on the other hand, provides a more detailed description of the vehicle dynamics, as it is based on the equilibrium equations. It can have different levels of detail/complexity and hypotheses used, each of them affects the estimation accuracy. Several authors introduce simplifying hypotheses, such as a single-track vehicle model as in [17] and [18]. Additional assumptions may be adopted, as the availability of the vehicle longitudinal speed or the hypothesis of small steering angles. Quite often the equilibrium equations are coupled with a tyre model but it is not strictly necessary as proposed in [19], there are several approaches, the most used are: linear models, Pacejka models, rational tyre model [20] and Dugoff model [21]. The use of a dynamic model can lead to a good VSA estimation, however, the accuracy of the results strongly depends on the tyre model parameters. Unmodeled effects, such as road conditions and tyre wear, can dramatically worsen the reliability of the estimation, meanwhile other secondary effects as the type of suspension adopted [22] and the thermodynamics of the tyre inner chamber [23] can be neglected easier in certain hypothesis. Several authors attempt to deal with this issue employing algorithms which provide an online update of tyre parameters as in [24] and [25]. A single-track vehicle model and a simplified Pacejka tyre model are adopted in this work, the main reason is that the aim of this paper is to compare the performance of different types of state estimators using the same plant model. The benchmark is not only based on the estimate accuracy but also on the run-time capability of each proposed algorithm.

An electric go-kart is employed to acquire the data-set. It is equipped with sensors in order to acquire the necessary input signals to feed the process model of the filters. In addition, an S-motion is used to acquire the true VSA and to validate the estimate one [26].

The paper is organized as follows: an overview of the implemented state estimators is reported in section 2, the mathematical formulation of the vehicle and tyre models is described in section 3. In Section 4, the implementation of the filters and the experimental data acquisition are shown. In section 5, the results obtained are shown comparing them towards the experimental VSA. Finally, the conclusions are reported in section 6.

2. State Estimators

State estimators implemented for VSA estimation are presented in this section, they are based on the filtering technique and, in particular, their discrete-time form is considered. The plant model is assumed the same for each filter. It is based on the physical equations presented in the next section, considered in a discrete time form applying the Euler method. Basically, it is a set of nonlinear equations that can be summarized by the following equation:

$$\begin{aligned} \hat{x}_k^- &= f(\hat{x}_{k-1}^+, u_{k-1}) + w_{k-1} \\ z_k &= h(\hat{x}_k^-, u_k) + v_k \end{aligned} \tag{1}$$

Concerning the first equation, the term on the left side is the *a-priori state estimate*, on the right side there are the functions $f(\cdot)$ that represent the nonlinear system of equations and the process noise term, respectively. $\hat{x}_{k-1}^+$ is the previous time *a-posteriori state estimate.* The second equation is a set of nonlinear and linear equations that gives in output the estimate measurements relying on the *a-priori state estimate.* u_{k-1} and u_k are additional inputs at the previous and current time step, respectively. The terms w_{k-1} and v_k represent the noise terms; in this work they are considered *additive*, *white*, *Gaussian*, *zero-mean* and *uncorrelated.* The noise covariance matrices are respectively Q and R and are computed adopting a "training" method, that runs the filter and tries to minimize the prediction error through a surrogate optimization algorithm [27].

2.1. Kalman filters

Numerous nonlinear Kalman filters can be found in literature, however all them descend from the linear kalman filter (KF) theory [28]. The KF algorithm can be divided into two steps:

- Time update, it projects the last computed state estimate ahead in time. It is also called *prediction step.*
- Measurement update, it adjusts the projected estimate by an actual measurement at that time. It is also called *correction step.*

However the KF is adopted only with linear process, in this application nonlinear Kalman filters are required. This briefly introduction is necessary in order to point out the basis of the next KFs, in fact the steps are common to every KFs but each one is characterized by a specific strategy to deal with the nonlinearity of the process.

2.1.1. Extended Kalman Filters The EKFs attempt to approximate the nonlinear system around the state estimate. Three types of EKFs are implemented in this work and they are briefly presented.

The First-Order Extendend Kalman Filter (FO-EKF) is based on the linearization of the nonlinear system around the state estimate using the first-order Taylor expansion [28] [29]. It approximates *measurement equations* by expanding it in a Taylor series around $\hat{x}_k^-$, the reason is because it is the best estimate of x_k before the measurement at time k is taken into account. When the *a-posteriori state estimate* is obtained, the best estimate of x_k becomes $\hat{x}_k^+$.

The basic idea of the Iterated Extended Kalman Filter (I-EKF) is to reduce the linearization error by reformulating the Taylor series expansion around $\hat{x}_k^+$ [28]. This process can be repeated as many times as desired: although, for most problems the majority of the possible improvements is obtained by re-linearizing only one time. The main difference between SO-EKF and I-EKF is the iteration cycle that refines the measurement update equations at time k, so the more the measurement equations are nonlinear the more effective the refinement is.

The Second Order Extended Kalman Filter (SO-EKF) perform a second order Taylor expansion of the process equations, $f(\cdot)$ and $h(\cdot)$ [28].

Summing up, the EKFs rely on the computation of the Jacobian matrices (and also Hessian matrices in SO-EKF), their computation may be quite difficult and and computationally expensive especially for high nonlinear systems. These matrices can be evaluated analytically or numerically, in the first case the computational burden required is smaller than in the second case.In this application the first method is adopted.

The main EKFs flaw is the linearization of a nonlinear process, in fact the linearized transformation is a good estimation method only when error propagation can be well approximated by a linear model, this could force the use of very small sampling times.

2.1.2. Unscented Kalman Filters Unscented Kalman Filters (UKFs) aims to overcome the main EKFs' flaw, it provides a simpler and more immediate way to propagate mean and covariance of random variables through a non-linear transformation [30] [31]. The *unscented transformation* is used to calculate the statistics of a random variable which undergoes a nonlinear transformation. The UKFs propagate the mean and covariance of the *sigma points* using system nonlinear equations and the *a-priori state estimate* is the weighted mean of them. As well, predicted measurements for each propagated *sigma point* can be computed the using the measurement equations and the predicted measurements vector is the weighted mean of them. The implemented UKFs differ in number of sigma points and/or weights formulation. An example of *sigma points* are reported in (2), they refer to the General Unscented Kalman Filter. The parameters α and k control the spread of the sigma points around the mean state value.

$$\begin{aligned} x^{(0)} &= \bar{x} \\ x^{(i)} &= \bar{x} + \tilde{x}^{(i)} \qquad i = 1, \dots, 2n \\ \tilde{x}^{(i)} &= (\sqrt{\alpha^2(n+k)P_{xx}})_i^T \qquad i = 1, \dots, n \\ \tilde{x}^{(n+i)} &= -(\sqrt{\alpha^2(n+k)P_{xx}})_i^T \qquad i = 1, \dots, n \end{aligned} \tag{2}$$

The simpliest UKF is here called Simply Unscented Kalman Filter (S-UKF), it uses $2n$ *sigma points* and also they have the same weight. Referring to the (2), in this case α is set equal to 1 and k is null.

Based on [32] and [33], it can be shown the same order of mean and covariance estimation accuracy can be obtained by choosing $(2n + 1)$ *sigma points* instead of $(2n)$ as before. This type of UKF is called General Unscented Kalman Filter (G-UKF). This means that the $2n$ *sigma points* are symmetrically distributed around the mean value. The G-UKF's algorithm and the S-UKF's algorithm are quite the same, the main difference is the number of *sigma points* computed at each time step and their weight factors as shown in [28].

A new set of *sigma points* and weight factors can be introduced. It can be shown that if the state vector, x, has n elements then the minimum number of sigma points, that gives the order of estimation accuracy of the previous section, is equal to $(n+1)$ [33]. These *sigma points* are called simplex sigma points and for this reason the algorthm is called Simplex Unscented Kalman Filter (SIMP-UKF). However, the number of *sigma points* can be reduced to $(n + 1)$ by choosing one of the weights to be zero. This type of filter aims to reduce the computational effort reducing the number of *sigma points* without losing in estimation accuracy.

The Spherical Unscented Transformation aims to rearrange the *sigma points* of the simplex algorithm in order to obtain better numerical stability, as reported in [31] and [33]. The filter based on this transformation is called Spherical Unscented Kalman Filter (SPHE-UKF).

The UKFs don't require the computation of Jacobian o Hessian, unlike the EKFs, this is a great advantage that make them easy to implement for all the nonlinear system. The main obstacle lays in the fact that the state covariance matrices must be semi-definite positive in order to have real matrices after applying the Cholesky decomposition [28]. This goal can be achieved tuning the noise covariance matrices.

2.2. Particle Filters

Particle Filter (PF) aims to estimate the state of a nonlinear process investigating the properties of sets of particles rather than the properties of individual particles as in UKFs. It is a completely nonlinear state estimator, unlike the UKFs and the EKFs presented before which are based on the approximation of the nonlinear system. It is a technique for implementing a recursive Bayesian filter by Monte Carlo (MC) simulations [34]. The key idea is to represent the required posterior density function by a set of random samples with associated weights and to compute estimates based on these samples and weights. As the number of samples becomes very large, the MC characterization becomes an equivalent representation to the usual functional description of the posterior pdf [14]. Therefore, the basic idea is to randomly generate a given number N state vectors based on the initial pdf $p(x_0)$, that is known. These state vectors are called *particles* and are denoted as $x^+_{0,i}$, with $i = 1, \ldots, N$. At each time step, each *particle* is propagated to the next time step using the system of equations $f(\cdot)$ in (1). Each propagated particle is characterized by a weight that reflects the likelihood that the particle represents the true state of the system. This weight factor is based on the of the measurement equation and of the pdf of the measurement noise, the relative likelihood q_i that the estimate measurement z_k is equal to a specific measurement y_k, given the premise that x_k is equal to the particle $\hat{x}^-_{k,i}$, can be computed as follows:

$$\begin{aligned} q_i &= P[(z_k = y_K)|(x_k = \hat{x}^-_{k,i})] = P[v_k = y_K - h(\hat{x}^-_{k,i})] \\ &\sim \frac{1}{(2\pi)^{m/2}|R|^{1/2}} exp\left(\frac{-[y_k - h(\hat{x}^-_{k,i})]^T R^{-1}}{2} [y_k - h(\hat{x}^-_{k,i})] \right) \end{aligned} \tag{3}$$

A *resampling strategy* is applied, in other words a brand new set of particles is computed: the basic idea is to eliminate particles that have small relative likelihood and to concentrate on particles with large one. A lot of *resampling strategy* can be found in literature, an exhaustive collection is done in [35], only three strategies are implemented in this application: Multinomial Resampling, Stratified Resampling, Systematic Resampling. Finally, for each time step the a-posteriori state estimate is computed as a weighted sum of the resampled particles.

The main implementation issue consists of the sample impoverishment, this can be overcome simply increasing the number of particles Ns, but this can quickly lead to an unreasonable demand of computational cost , and often simply delays the inevitable sample impoverishment.

3. Mathematical Model

Several vehicle and tyre models are present in literature [15]. In this case, considering the main goal of this work, a single-track vehicle model and a simplified Pacejka tyre model has been chosen [36].

3.1. Vehicle model

The vehicle model adopted in this work is a single-track vehicle model, which does not require the knowledge of such not easily measurable vehicle parameters. Taking into account the assumption behind this model as reported in [15], in this application the hypothesis of small steering angle is not considered, because the steering angle values reached during the on-track tests are not so small to be neglected. The vehicle involved is an electric go-kart, in which the braking and tractive forces affect only the rear axle.

The kinematic equations of the vehicle model are:

$$\begin{aligned} a_x &= \dot{u} - v\ r \\ a_y &= \dot{v} + u\ r \end{aligned} \tag{4}$$

With a the linear acceleration, u the longitudinal velocity of the CoG along the x-axis, v the lateral velocity along the y-axis, and r the angular velocity around the z-axis.

The vehicle side-slip angle is given by

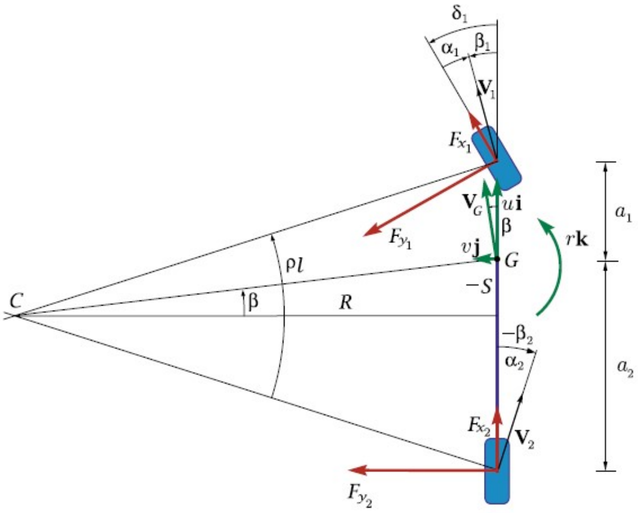

Figure 1. Single-track vehicle model basic scheme

$$\beta = \tan^{-1}\left(\frac{v}{u}\right) \tag{5}$$

Considering the equations (4), the equilibrium equations are:

$$\begin{aligned} m\,(\dot{u} - v\,r) &= F_{x_1}\ \cos(\delta_1) - F_{y_1}\ \sin(\delta_1) + F_{x2} \\ m\,(\dot{v} + u\,r) &= F_{y_1}\ \cos(\delta_1) + F_{x_1}\ \sin(\delta_1) + F_{y2} \\ J_z\ \dot{r} &= F_{y_1}\ a_1\cos(\delta_1) + F_{x_1}\ a_1\sin(\delta_1) - F_{y_2}\ a_2 \end{aligned} \tag{6}$$

The equations above are defined in the ISO body-fixed reference system, the forces F_{x_i} and F_{y_i}[1] are given by the tyre model adopted and are defined in the ISO tyre reference framework.

3.2. Tyre model

As mentioned, a simplified version of the Pacejka tyre model, based on the macro-parameters instead of micro-parameters, has been adopted [36]:

$$F_0 = D\ \sin\left\{C\ \arctan\left[B\ x_s - E(B\ x_s - \arctan\left(B\ x_s\right))\right]\right\} \tag{7}$$

The formulation is the same for both planar forces, the independent variable x_s identify the slip ratio or the tyre slip angle. The tunable parameters are not all considered as fixed parameters, indeed a vertical load dependence is adopted for the peak value D:

$$D = D(F_z) = \mu F_z = (a_1\ F_z + a_2)F_z \qquad \textit{with} \quad a_1 < 0 \tag{8}$$

in this case the terms a_1 and a_2 are fixed value parameters. The parameters B, C and D define the slope in the linear region [36].

The effect of the combined slip case has been taken into account by the G-function (or Hill function), again considering the shift null:

$$G = \cos\left\{C_c\ \arctan[B_c\ x_c - E_c(B_c\ x_c - \arctan(B_c\ x_c))]\right\} \tag{9}$$

[1] where the index i refers to the axle, 1 for the front axle and 2 for the rear one.

Note the difference between the independent variable in (9) and in (7), if x_s is the tyre slip angle so x_c is the slip ratio and vice versa. All the macro-parameters are computed starting from the output obtained using a T.R.I.C.K. methodology [37], which provides a virtual telemetry requiring the acquired data set as input. The obtained forces and slips are used as true values in order to calibrate the MF model presented above.

Finally the formulation of the MF force in combined slip case is given by:

$$F_c = F_0 \; G \tag{10}$$

These tangential forces are assumed to be dependent only on: tyre vertical load F_z, tyre slip angle α and slip ratio k. The dependence on the camber angle γ and on the spin slip ϕ as well is neglected in this work.
The vertical load may be evaluated with different levels of accuracy. In this work the aerodynamics effect is neglected, and because of the chosen vehicle dynamics model (Single track) only longitudinal load transfer and the static load distribution affect the vertical forces acting on the single wheel at the front and rear axle.

The mathematical formulation of the slip ratio and the tyre slip angle, for each axle, is given by:

$$\begin{aligned}
\alpha_1 &= \arctan\left(\frac{(v + r\ a_1)\cos(\delta_1) - u\sin(\delta_1)}{|u\cos(\delta_1) + (v + r\ a_1)\sin(\delta_1)|}\right) \\
k_1 &= \frac{\omega_1 R_{r_1} - u\cos(\delta_1) + (v + r\ a_1)\sin(\delta_1) -}{u\cos(\delta_1) + (v + r\ a_1)\sin(\delta_1)} \\
\alpha_2 &= \arctan\left(\frac{(v - r\ a_2)}{|u|}\right) \\
k_2 &= -\frac{\omega_2 R_{r_2} - u}{u}
\end{aligned} \tag{11}$$

4. Implementation of the filters and experimental data acquisition

The mathematical formulation of the physical model described in the previous section is implemented as process of each filters. In this application, the state vector of each filter is constituted by:

$$x_k = [u_k, v_k, r_k]^T \tag{12}$$

longitudinal and lateral vehicle velocity $[m/s]$ and yaw rate $[rad/s]$ respectively. The filter requires the knowledge of some measurements acquired at each time step, in order to correct the *a-priori state estimate* provided by the process. In this application, the elements of the measurement vector are:

$$y_k = \left[\omega_{1,1_k}^{ENCODER}, \omega_{1,2_k}^{ENCODER}, r_k^{IMU}, a_{y_k}^{IMU}, a_{x_k}^{IMU}\right]^T \tag{13}$$

respectively rotational velocity of the front-left and front-right wheels $[rad/s]$, yaw rate $[rad/s]$ and longitudinal and lateral vehicle linear acceleration $[m/s^2]$. The elements of this vector are compared with the corresponding estimate ones, the difference is proportional to the amount of correction of the *a-priori state estimate.*

The system equations adopted in each filter descend from the equilibrium equations (6), they are computed starting from the (14) and applying the Euler method.

$$\begin{aligned}
\dot{u} &= v\ r + \frac{1}{m}\big(F_{x_1}\ \cos(\delta_1) - F_{y_1}\ \sin(\delta_1) + F_{x_2}\big) \\
\dot{v} &= -u\ r + \frac{1}{m}\big(F_{y_1}\ \cos(\delta_1) + F_{x_1}\ \sin(\delta_1) + F_{y_2}\big) \\
\dot{r} &= \frac{1}{J_z}\big(F_{y_1}\ a_1 \cos(\delta_1) + F_{x_1}\ a_1 \sin(\delta_1) - F_{y_2}\ a_2\big)
\end{aligned} \tag{14}$$

the expressions on the right side are strongly non-linear, especially because of the tyre model chosen. As mentioned above a linearization is needed to implement the FO-EKF and the I-EKF or in other words the Jacobian matrix has to be computed, it is given by the partial derivatives of $f(\cdot)$ with respect to the state vector x:

$$\left.\frac{\partial f}{\partial x}\right|_{\hat{x}^+} = \frac{\partial f(\hat{x}^+, u, 0)}{\partial x} = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix} \tag{15}$$

where the expressions of each term are reported in the Appendix B. The Hessian computation is required for the implementation of the SO-EKF, it is not reported here for sake of simplicity. The same can be done with the measurement equations $h(\cdot)$ given by the expressions below:

$$\begin{aligned}
\omega_{11} &= \frac{\cos(\delta_{11})}{R_{r_1}}\hat{u}^- + \frac{\sin(\delta_{11})}{R_{r_1}}\hat{v}^- + \frac{a_1 \sin(\delta_{11}) - t_1/2\cos(\delta_{11})}{R_{r_1}}\hat{r}^- \\
\omega_{12} &= \frac{\cos(\delta_{12})}{R_{r_2}}\hat{u}^- + \frac{\sin(\delta_{12})}{R_{r_2}}\hat{v}^- + \frac{a_1 \sin(\delta_{12}) - t_1/2\cos(\delta_{12})}{R_{r_2}}\hat{r}^- \\
r &= \hat{r}^- \\
a_y &= \frac{1}{m}\Big(F_{y_1}\ \cos(\delta_1) + F_{x_1}\ \sin(\delta_1) + F_{y_2}\Big) \\
a_x &= \frac{1}{m}\Big(F_{x_1}\ \cos(\delta_1) - F_{y_1}\ \sin(\delta_1) + F_{x_2}\Big)
\end{aligned} \tag{16}$$

On the right side of the equations above there are the estimate measurements, which are compared to the acquired ones computed by (13).

The measurements mentioned in (13) are acquired by sensors, in addition some of them feed the system equations as seen in (1) and are called *additional inputs* u_{k-1} or u_k. These measurements require sensors which are commonly used for the basic data acquisition, sometimes they equip common vehicles and are used to prevent critical events. In this application an electric go-kart is used in order to acquire the experimental data, several indoor on-track tests have been conducted with the aim to explore the vehicle behaviour in different operating conditions. The vehicle is rear drive and also the brakes affect only the rear axle. The sensors adopted consist of: an encoder per each front wheel to acquire the rotational velocity whereas the rotational velocities of the rear ones is measured through the electric motors, an IMU which acquires the in-plane accelerations and the yaw rate, an S-motion provides the true values of the longitudinal and lateral velocities (thus the side slip angle) and finally an encoder measures the steering wheel angle. Starting from this latter, the steering angle of the front wheels is computed using a non-linear function identified using experimental data. All acquired data are subjected to post-processing, consisting in the cleaning (deleting the incorrect and/or duplicated samples), synchronisation, filtering and resampling (20 Hz in this application). The signals acquired by the IMU and S-motion are transported to the CG using rotational matrices that are based on the geometrical distances of the sensors from the CG.

5. Results and validation

In this section the obtained results are presented, comparing each estimate VSA to the experimental one acquired by the S-Motion. All the filters are "trained" using the same algorithm that provides the process and measurements covariance matrices.

As can be seen in the figures below, the VSA estimation depends on the type of filter implemented. They refer to one of the led tests, the other ones are taken into account in term of mean RMSE of the estimate VSA in the table (1).

Table 1. RMSE mean values for each test.

[deg]	TEST 1	TEST 2	TEST 3	TEST 4	MEAN
FO-EKF	2.52	1.18	2.23	2.02	2.03
I-EKF	2.46	1.26	2.09	2.26	2.02
SO-EKF	2.48	1.20	2.36	2.22	2.06
S-UKF	2.10	1.32	2.38	1.59	1.85
G-UKF	2.40	1.06	2.72	2.87	2.26
SIMP-UKF	2.60	1.39	2.95	1.86	2.20
SPHE-UKF	2.17	1.43	2.82	2.70	2.28
MULTIres-PF	4.62	1.32	3.53	3.80	3.32
SYSTres-PF	3.26	1.38	3.24	3.65	2.88
STRAres-PF	4.28	1.44	3.88	3.99	3.40

Another filter's index of performance taken into account in this work is the run-time. The mean run-time required by each algorithm is reported in the table (2). This may be useful in the real-time application, for this reason the time taken by each implemented filter to estimate 1 s of real time acquired data is considered.

Considering each test, the implemented EKFs provide quite similar VSA estimations. The figure (2) refers to the test #3, in which the I-EKF outperforms the others at the beginning, this is due to the refining provided by the iteration cycles. The time taken by the EKFs is different, as can be seen in table 2, the FO-EKF needs less than the half of the time taken by the others.

The implemented UKFs provide slightly different VSA estimations. The best performance in term of RMSE are achieved by the S-UKF and the G-UKF, that uses more sigma points than the others. Note that the S-UKF outperforms the G-UKF on three out of the four tests. The time taken by the EKFs is different and the same trend is repeated for each test. The G-UKF require more time than the others, as expected, because it adopted 7 sigma points and also their weight factors are different, while in the S-UKF they are the same and the number of sigma points involved is 6. The SIMP-UKF and the SPHE-UKF require a very similar execution time, they differ only in the weight factor definition and both use 5 sigma points.

The implemented PFs always provide noisy VSA estimations and also higher value of RMSE. In general, the SYSres-PF outperforms the others, but they are not so far one each other as shown in figure (4). The execution time is very high if compared to the other filters ones. In addition, note that if the random number generator is not fixed, the results depends on the random numbers generated as well, that could be very different at each run of the PF.

Table 2. Time taken by each implemented filter to estimate 1 *s* of real time acquired data.

[ms/s]	TEST 1	TEST 2	TEST 3	TEST 4	MEAN
FO-EKF	8.20	8.86	8.64	7.49	8.30
I-EKF	22.91	24.35	25.19	22.37	23.71
SO-EKF	18.26	20.63	19.31	17.84	19.01
S-UKF	27.78	27.36	27.79	25.952	27.22
G-UKF	39.08	37.20	35.98	34.08	36.59
SIMP-UKF	25.73	25.13	23.28	22.24	24.10
SPHE-UKF	25.08	25.42	23.52	22.45	24.12
MULTIres-PF	311.0	310.9	306.0	305.8	308.5
SYSTres-PF	309.4	303.2	306.1	304.0	305.7
STRAres-PF	308.2	302.9	308.1	306.8	306.5

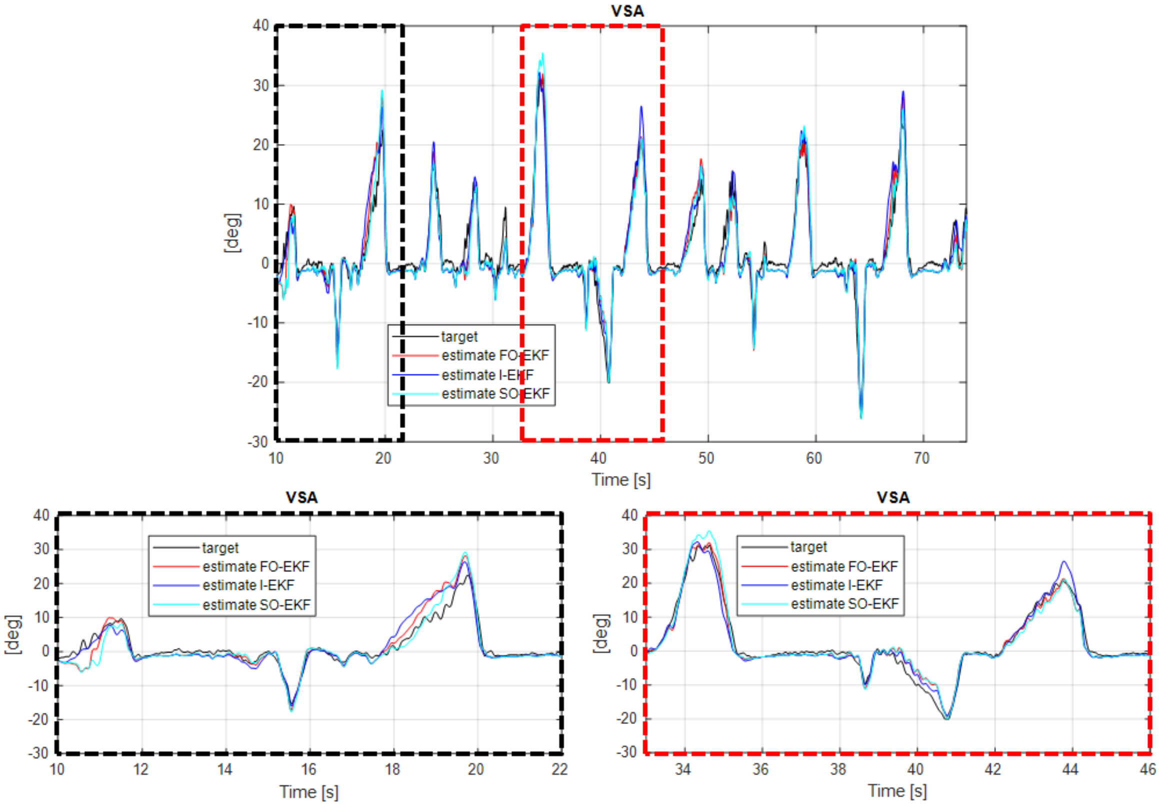

Figure 2. VSA estimated by EKFs and acquired by S-motion

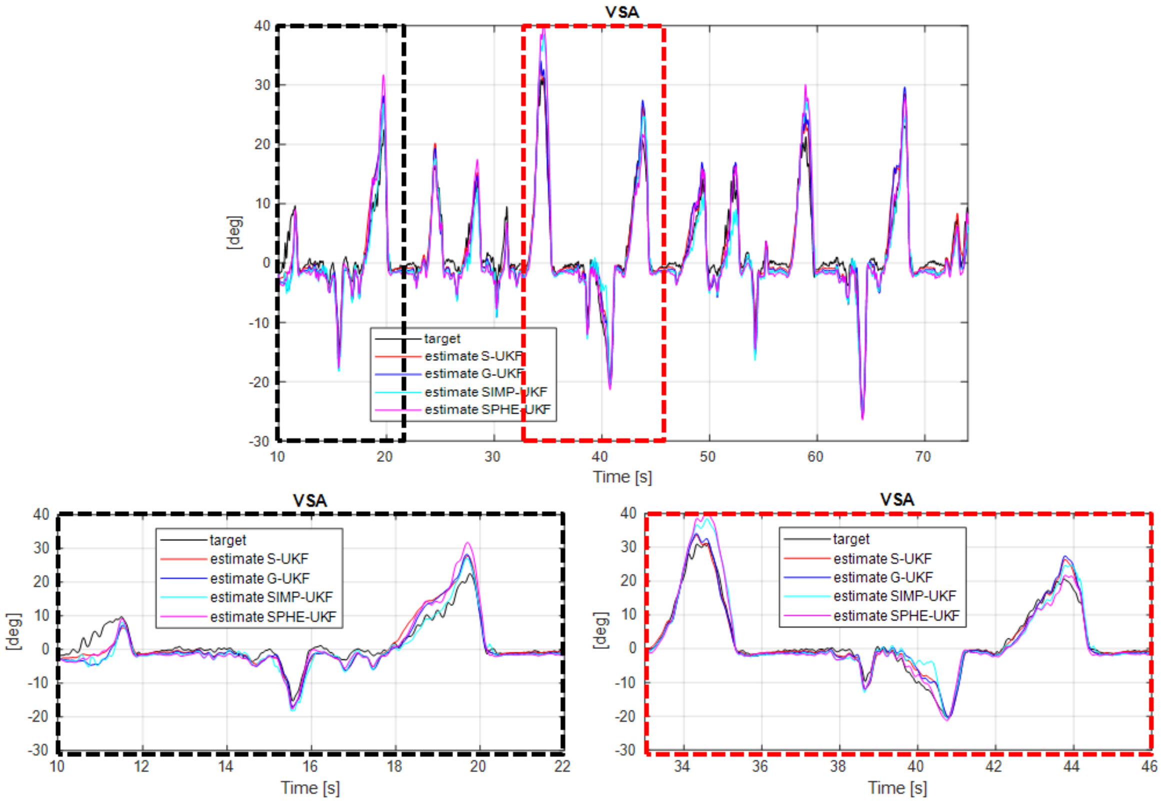

Figure 3. VSA estimated by UKFs and acquired by S-motion

The estimated VSA, but also all the other estimated quantities as well, are characterized by a noisy trend. This is due to the fact that the samples are very close one each other but the same sample have also different weight factors at each time step: this means that at consecutive time step the estimated value is not so close to the previous as in the previous filters.

At that point it becomes important to highlight that all the results presented are strongly dependent on the Q and R matrices. As said, they are computed using a training method based on an optimization method, the surrogate optimization for global minimization of time-consuming objective functions has been adopted in thesis work, but others can be found in literature [27]. However, the stopping condition adopted is the maximum number of function evaluations, that has been set on 50, a relatively low value. In other words, the optimizator runs 50 times the filter, changing the Q and R values, storing only the ones that minimize the residual prediction error computed till now. In addition, this is repeated 2 times: the initial values of Q and R obtained in the second optimization cycle are the best values provided by the previous optimization cycle. Best VSA estimation may be achieved if higher limit is set for the maximum number of function evaluations, but of course this requires much more time, especially considering the PFs. The reason behind the low value chosen is that the the aim of this work is to compare the different type of filters, adopting the same boundary conditions, not looking for the best state estimate that each filter may provide.

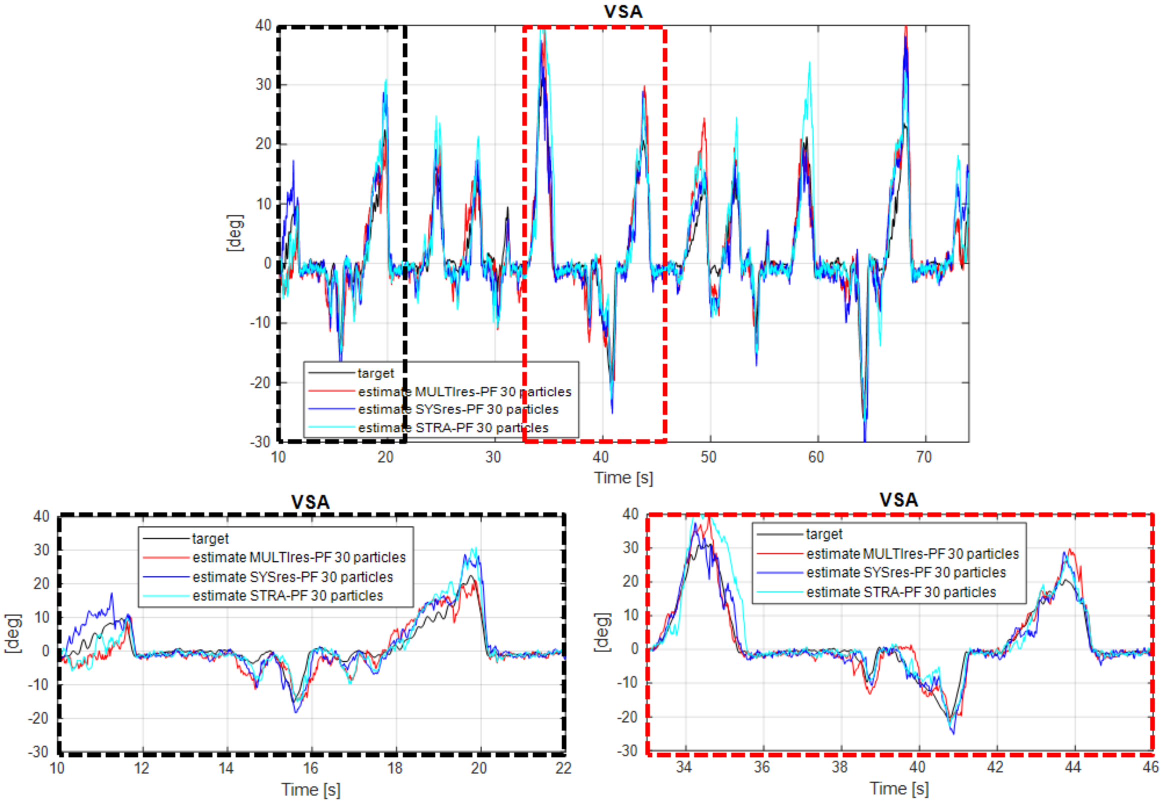

Figure 4. VSA estimated by PFs and acquired by S-motion

6. Conclusions

A vehicle sideslip angle estimation based on dynamic model is proposed in this work. The purpose of this paper is to compare the performance of different model-based state estimators (Extended Kalman Filters, Unscented Kalman Filters and Particle Filters) in terms of estimation accuracy and the computational cost for a chosen vehicle. The mathematical model of the vehicle and the features of each implemented filter have been mathematically presented.

Concerning the obtained results, the EKFs and the UKFs show a better state estimation using the vehicle model presented and the Pacejka macroparameters computed. Considering the mean value of the RMSE computed per each tests, the S-UKF exhibits the lowest value whereas the other UKFs exhibit a value which is about the 20% higher; the EKFs show the same mean value if compared one each other but this is about the 10% higher than the one reached by the S-UKF; Finally, the SYSTres-PF shows the lowest RMSE mean value if compared with the other implemented PFs, but if compared to the others the PFs exhibit the higher values, in particular the STRAres-PF shows the highest one. Concerning the computational burden required by each state estimator which can be considered as proportional to the time taken by the filter to estimate 1 second of real time, the FO-EKF is characterized by the lowest amount of time required, thanks to its simple algorithm. The other EKFs present a value which is about the 150% higher. Considering the UKFs, the SIMP-UKF and the SPHE-UKF require lower run-time than the S-UKF and the G-UKF, this latter is characterized by the highest one if considering only the Kalman-based filters, it is about the 340% higher than the lowest one. Considering the PFs, the time required is one order of magnitude higher than the Kalman-based one, they take about one-third of second to estimate 1 second of real time. However, the overall state estimate is not always accurate, in same cases the estimate VSA is quite different than the actual one, this may be mitigated adopting a complete Magic Formula, with micro-parameters instead of macro-parameters. Moreover, it may be interesting to consider a tricycle vehicle model instead of a bicycle one in order to take into account the lateral load transfer effect.

In order to improve the VSA estimation the tyre model may be a key factor, in this work the macro-parameters are fixed, but considering their update during filter execution, due to thermodynamics and wear effect, could lead to better results.

Appendix A

The algorithms of the implemented filters are here reported.

Algorithm 1 First-Order Extended Kalman Filter algorithm

```
$\hat{x}_0^+ = E[x_0]$                                         ▷ Initial state
$P_0^+ = E[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T]$              ▷ Initial state Covariance
procedure FO-EKF($T$, $\{u\}_{k=1}^T$)
    for $k = 1 \to T$ do
        $\hat{x}_k^- = f(\hat{x}_{k-1}^+, u_{k-1}, 0)$                 ▷ a-priori state estimate
        $P_k^- = F_{k-1} P_{k-1}^+ F_{k-1}^T + Q$                ▷ a-priori state estimate covariance
        $z_k = h(\hat{x}_k^-, u_k, 0)$                            ▷ a-priori measurement estimate
        $K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R)^{-1}$           ▷ Kalman gain
        $\hat{x}_k^+ = \hat{x}_k^- + K_k(y_k - z_k)$                   ▷ a-posteriori state estimate
        $P_k^+ = (I - K_k H_k) P_k^-$                          ▷ a-posteriori state estimate covariance
    end for
end procedure
```

Algorithm 2 Iterated Extended Kalman Filter algorithm

```
$\hat{x}_0^+ = E[x_0]$
$P_0^+ = E[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T]$
procedure I-EKF($T$, $\{u\}_{k=1}^T$, N)
    for $k = 1 \to T$ do
        $\hat{x}_k^- = f(\hat{x}_{k-1}^+, u_{k-1}, 0)$
        $P_k^- = F_{k-1} P_{k-1}^+ F_{k-1}^T + Q$
        $\hat{x}_{k,1}^+ = \hat{x}_k^-$
        for $i = 1 \to N$ do
            $z_{k,i} = h(\hat{x}_{k,i}^+, u_k, 0) - H_{k,i}(\hat{x}_k^- - \hat{x}_{k,i}^+)$
            $K_{k,i} = P_k^- H_{k,i}^T (H_{k,i} P_k^- H_{k,i}^T + R)^{-1}$
            $\hat{x}_{k,i+1}^+ = \hat{x}_k^- + K_{k,i}(y_k - z_{k,i})$
            $P_{k,i+1}^+ = (I - K_k H_k) P_k^-$
        end for
    end for
end procedure
```

Algorithm 3 Second Order Extended Kalman Filter algorithm

```
$\hat{x}_0^+ = E[x_0]$
$P_0^+ = E[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T]$
**procedure** SO-EKF($T$, $\{u\}_{k=1}^T$)
    **for** $k = 1 \to T$ **do**
        $\hat{x}_k^- = f(\hat{x}_{k-1}^+, u_{k-1}, 0) + \frac{1}{2}\sum_{i=1}^{n} \phi_i Tr \left[ \left.\frac{\partial^2 f_i}{\partial x^2}\right|_{\hat{x}_{k-1}^+} P_{k-1}^+ \right]$
        $P_k^- = F_{k-1} P_{k-1}^+ F_{k-1}^T + Q$
        $z_k = h(\hat{x}_k^-, u_k, 0) + \frac{1}{2}\sum_{i=1}^{m} \phi_i \,\mathrm{Tr} \left[ \left.\frac{\partial^2 h_i}{\partial x^2}\right|_{\hat{x}_k^-} P_k^- \right]$
        $K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R)^{-1}$
        $\hat{x}_k^+ = \hat{x}_k^- + K_k(y_k - z_k)$
        $P_k^+ = (I - K_k H_k) P_k^-$
    **end for**
**end procedure**
```

Algorithm 4 Simply Unscented Kalman Filter algorithm

```
$\hat{x}_0^+ = E[x_0]$
$P_0^+ = E[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T]$
**procedure** S-UKF($T$, $\{u\}_{k=1}^T$)
    **for** $k = 1 \to T$ **do**
        $\hat{x}_{k-1}^{(i)} = \hat{x}_{k-1}^+ + \left(\sqrt{nP_{k-1}^+}\right)_i^T \qquad i = 1, \ldots, n$
        $\hat{x}_{k-1}^{(n+i)} = \hat{x}_{k-1}^+ - \left(\sqrt{nP_{k-1}^+}\right)_i^T \qquad i = 1, \ldots, n$
        $\hat{x}_k^{(i)} = f(\hat{x}_{k-1}^{(i)}, u_{k-1}, 0)$
        $\hat{x}_k^- = \frac{1}{2n}\sum_{i=1}^{2n} \hat{x}_k^{(i)}$
        $P_k^- = \frac{1}{2n}\sum_{i=1}^{2n} (\hat{x}_k^{(i)} - \hat{x}_k^{(-)})(\hat{x}_k^{(i)} - \hat{x}_k^{(-)})^T + Q$
        $z_k^{(i)} = h(\hat{x}_k^i, u_k, 0)$
        $z_k = \frac{1}{2n}\sum_{i=1}^{2n} z_k^{(i)}$
        $P_k^y = \frac{1}{2n}\sum_{i=1}^{2n} (z_k^{(i)} - z_k)(z_k^{(i)} - z_k)^T + R$
        $P_k^{xy} = \frac{1}{2n}\sum_{i=1}^{2n} (\hat{x}_k^{(i)} - \hat{x}_k^{(-)})(z_k^{(i)} - z_k)^T$
        $K_k = P_k^{xy} (P_k^y)^{-1}$
        $\hat{x}_k^+ = \hat{x}_k^- + K_k(y_k - z_k)$
        $P_k^+ = P_k^- - K_k P_k^- K_k^T$
    **end for**
**end procedure**
```

The G-UKF's algorithm is similat to the S-UKF's one, but it involves $(2n+1)$ *sigma points* as shown in (2) instead of $2n$. The formulation of *sigma points* and weight factors are here reported:

$$
\begin{aligned}
x^{(0)} &= \bar{x} \\
x^{(i)} &= \bar{x} + \tilde{x}^{(i)} \qquad i = 1, \ldots, 2n \\
\tilde{x}^{(i)} &= (\sqrt{\alpha^2(n+k)P_{xx}})_i^T \qquad i = 1, \ldots, n \\
\tilde{x}^{(n+i)} &= -(\sqrt{\alpha^2(n+k)P_{xx}})_i^T \qquad i = 1, \ldots, n \\
W_m^{(0)} &= \frac{\alpha^2(n+k)-n}{\alpha^2(n+k)} \qquad W_m^{(i)} = \frac{1}{2\alpha^2(n+k)} \qquad i = 1, \ldots, 2n \\
W_c^{(0)} &= (2-\alpha^2+\beta) - \frac{n}{\alpha^2(n+k)} \qquad W_c^{(i)} = \frac{1}{2\alpha^2(n+k)} \qquad i = 1, \ldots, 2n
\end{aligned}
$$

The SPHE-UKF and the SIMP-UKF adopt a different formulation of the *sigma points* as reported below:

Algorithm 5 Spherical Unscented Transformation

```
$W^{(0)} \in [0,1)$                                                         ▷ Initial choice
$W^{(i)} = \frac{1-W^{(0)}}{n+1}$    $i = 1, \ldots, n+1$                          ▷ Weights initialization
$\sigma_0^{(1)} = 0$    $\sigma_1^{(1)} = \frac{-1}{\sqrt{2W^{(1)}}}$    $\sigma_2^{(1)} = \frac{1}{\sqrt{2W^{(1)}}}$      ▷ Sigma vector initialization
for $j = 2 \to n$ do
    $\sigma_i^{(j)} = \begin{cases} \begin{bmatrix} \sigma_0^{(j-1)} \\ 0 \end{bmatrix} & i = 0 \\ \begin{bmatrix} \sigma_i^{(j-1)} \\ \frac{-1}{\sqrt{j(j+1)W^{(1)}}} \end{bmatrix} & i = 1, \ldots, j \\ \begin{bmatrix} 0_{(j-1)} \\ \frac{j}{\sqrt{j(j+1)W^{(1)}}} \end{bmatrix} & i = j+1 \end{cases}$     ▷ Sigma vector building
end for
$x^{(i)} = \bar{x} + \sigma_i^{(n)}\sqrt{P_{xx}}$    $i = 0, \ldots, n+1$                      ▷ Sigma points
```

Algorithm 6 Simplex Unscented Transformation

```
$W^{(0)} \in [0,1)$    ▷ Initial choice
$W^{(i)} = \begin{cases} 2^{-n}(1-W^{(0)}) & i=1,2 \\ 2^{i-2}W^{(1)} & i=3,\dots,n+1 \end{cases}$    ▷ Weights initialization
$\sigma_0^{(1)} = 0 \qquad \sigma_1^{(1)} = \frac{-1}{\sqrt{2W^{(1)}}} \qquad \sigma_2^{(1)} = \frac{1}{\sqrt{2W^{(1)}}}$    ▷ Sigma vector initialization
for $j = 2 \to n$ do
    $\sigma_i^{(j)} = \begin{cases} \begin{bmatrix} \sigma_0^{(j-1)} \\ 0 \end{bmatrix} & i=0 \\ \begin{bmatrix} \sigma_i^{(j-1)} \\ \frac{-1}{\sqrt{2W^{(j+1)}}} \end{bmatrix} & i=1,\dots,j \\ \begin{bmatrix} 0_{(j-1)} \\ \frac{j}{\sqrt{2W^{(j+1)}}} \end{bmatrix} & i=j+1 \end{cases}$    ▷ Sigma vector building
end for
$x^{(i)} = \bar{x} + \sigma_i^{(n)}\sqrt{P_{xx}} \qquad i=0,\dots,n+1$    ▷ Sigma points
```

Appendix B

The terms in equations (15) are:

$$a_1 = 1 + \frac{\Delta t}{m}\left(\frac{\partial F_{x_1}}{\partial u} + \frac{\partial F_{x_2}}{\partial u} - \frac{\partial F_{y_1}}{\partial u}\delta_1\right)$$
$$a_2 = \Delta t\left(\hat{r}^{+} + \frac{1}{m}\left(\frac{\partial F_{x_1}}{\partial v} + \frac{\partial F_{x_2}}{\partial v} - \frac{\partial F_{y_1}}{\partial v}\delta_1\right)\right)$$
$$a_3 = \Delta t\left(\hat{v}^{+} + \frac{1}{m}\left(\frac{\partial F_{x_1}}{\partial r} + \frac{\partial F_{x_2}}{\partial r} - \frac{\partial F_{y_1}}{\partial r}\delta_1\right)\right)$$
$$a_4 = \Delta t\left(-\hat{r}^{+} + \frac{1}{m}\left(\frac{\partial F_{y_1}}{\partial u} + \frac{\partial F_{y_2}}{\partial u} + \frac{\partial F_{x_1}}{\partial u}\delta_1\right)\right)$$
$$a_5 = 1 + \frac{\Delta t}{m}\left(\frac{\partial F_{y_1}}{\partial v} + \frac{\partial F_{y_2}}{\partial v} + \frac{\partial F_{x_1}}{\partial v}\delta_1\right)$$
$$a_6 = \Delta t\left(-\hat{u}^{+} + \frac{1}{m}\left(\frac{\partial F_{y_1}}{\partial r} + \frac{\partial F_{y_2}}{\partial r} + \frac{\partial F_{x_1}}{\partial r}\delta_1\right)\right)$$
$$a_7 = \frac{\Delta t}{J_z}\left(\frac{\partial F_{y_1}a_1}{\partial u} - \frac{\partial F_{y_2}a_2}{\partial u} + \frac{\partial F_{x_1}a_1}{\partial u}\delta_1\right)$$
$$a_8 = \frac{\Delta t}{J_z}\left(\frac{\partial F_{y_1}a_1}{\partial v} - \frac{\partial F_{y_2}a_2}{\partial v} + \frac{\partial F_{x_1}a_1}{\partial v}\delta_1\right)$$
$$a_9 = 1 + \frac{\Delta t}{J_z}\left(\frac{\partial F_{y_1}a_1}{\partial r} - \frac{\partial F_{y_2}a_2}{\partial r} + \frac{\partial F_{x_1}a_1}{\partial r}\delta_1\right)$$

References

[1] Julien Caroux et al. “Sideslip angle measurement, experimental characterization and evaluation of three different principles”. In: *IFAC Proceedings Volumes* 40.15 (2007). 6th IFAC Symposium on Intelligent Autonomous Vehicles, pp. 505–510. ISSN: 1474-6670. DOI: https://doi.org/10.3182/20070903-3-FR-2921.00086.

[2] Juraj Kabzan et al. *AMZ Driverless: The Full Autonomous Racing System.* May 2019.

[3] Saeid Niazi and Alireza Toloei. “State Estimation for Target Tracking Problems with Nonlinear Kalman Filter Algorithms”. In: *International Journal of Computer Applications* 98 (July 2014). DOI: 10.5120/17277-7708.

[4] Alexander Wischnewski et al. “Vehicle Dynamics State Estimation and Localization for High Performance Race Cars”. In: vol. 52. July 2019, pp. 154–161. DOI: 10.1016/j.ifacol.2019.08.064.

[5] S. Melzi and E. Sabbioni. “On the vehicle sideslip angle estimation through neural networks: Numerical and experimental results”. In: *Mechanical Systems and Signal Processing* 25.6 (2011). Interdisciplinary Aspects of Vehicle Dynamics, pp. 2005–2019. ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2010.10.015.

[6] Xiaoping Du et al. “A prediction model for vehicle sideslip angle based on neural network”. In: *2010 2nd IEEE International Conference on Information and Financial Engineering.* 2010, pp. 451–455. DOI: 10.1109/ICIFE.2010.5609398.

[7] Daniel Chindamo and Marco Gadola. “Estimation of Vehicle Side-Slip Angle Using an Artificial Neural Network”. In: *MATEC Web of Conferences* 166 (Jan. 2018), p. 02001. DOI: 10.1051/matecconf/201816602001.

[8] Francesco Carputo et al. “A Neural-Network-Based Methodology for the Evaluation of the Center of Gravity of a Motorcycle Rider”. In: *Vehicles* 3 (July 2021), pp. 377–389. DOI: 10.3390/vehicles3030023.

[9] Danilo D’Andrea et al. “Development of Machine Learning Algorithms for the Determination of the Centre of Mass”. In: *Symmetry* 13 (Feb. 2021), p. 401. DOI: 10.3390/sym13030401.

[10] M.C. Best, T.J. Gordon, and P.J. Dixon. “An Extended Adaptive Kalman Filter for Real-time State Estimation of Vehicle Handling Dynamics”. In: *Vehicle System Dynamics* 34.1 (2000), pp. 57–75. DOI: 10.1076/0042-3114(200008)34:1;1-K;FT057.

[11] Cristiano Pieralice et al. “Vehicle Sideslip Angle Estimation Using Kalman Filters: Modelling and Validation: Proceedings of the Second International Conference of IFToMM Italy”. In: Jan. 2019, pp. 114–122. ISBN: 978-3-030-03319-4. DOI: 10.1007/978-3-030-03320-0_12.

[12] S. Antonov, A. Fehn, and A. Kugi. “Unscented Kalman filter for vehicle state estimation”. In: *Vehicle System Dynamics* 49.9 (2011), pp. 1497–1520. DOI: 10.1080/00423114.2010.527994.

[13] Hongbin Ren et al. “Vehicle State Information Estimation with the Unscented Kalman Filter”. In: *Advances in Mechanical Engineering* 6 (2014), p. 589397. DOI: 10.1155/2014/589397.

[14] M. Arulampalam et al. “A Tutorial on Particle Filters for Online Nonlinear/ Non-Gaussian Bayesian Tracking”. In: *Signal Processing, IEEE Transactions on* 50 (Mar. 2002), pp. 174–188. DOI: 10.1109/78.978374.

[15] Massimo Guiggiani. *The Science of Vehicle Dynamics.* Springer International Publishing AG, 2018.

[16] "Robust Vehicle Sideslip Estimation Based on Kinematic Considerations". In: *IFAC-PapersOnLine* 50.1 (2017). 20th IFAC World Congress, pp. 14855–14860. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2017.08.2513.

[17] Federico Cheli et al. "A methodology for vehicle sideslip angle identification: Comparison with experimental data". In: *Vehicle System Dynamics* 45 (June 2007), pp. 549–563. DOI: 10.1080/00423110601059112.

[18] Marco Gadola et al. "Development and validation of a Kalman filter-based model for vehicle slip angle estimation". In: *Vehicle System Dynamics* 52 (Jan. 2014). DOI: 10.1080/00423114.2013.859281.

[19] Manuel Acosta Reche, Stratis Kanarachos, and Mike Blundell. "Virtual Tyre Force Sensors: An Overview of Tyre Model-based and Tyre Model-less State Estimation Techniques". In: *Proceedings of the Institution of Mechanical Engineers Part D Journal of Automobile Engineering* In press. (Sept. 2017). DOI: 10.1177/0954407017728198.

[20] Feliciano Di Biase, Basilio Lenzo, and Francesco Timpone. "Vehicle Sideslip Angle Estimation for a Heavy-Duty Vehicle via Extended Kalman Filter Using a Rational Tyre Model". In: *IEEE Access* 8 (2020), pp. 142120–142130. DOI: 10.1109/ACCESS.2020.3012770.

[21] Mingyuan Bian et al. "A Dynamic Model for Tire/Road Friction Estimation under Combined Longitudinal/Lateral Slip Situation". In: vol. 1. Apr. 2014. DOI: 10.4271/2014-01-0123.

[22] Andrea Genovese et al. "Multiphysics design and optimization of a vibration-based energy harvester from pantograph-catenary interaction". In: *IOP Conference Series: Materials Science and Engineering* 922 (Oct. 2020), p. 012012. DOI: 10.1088/1757-899X/922/1/012012.

[23] Luigi Teodosio et al. "A numerical methodology for thermo-fluid dynamic modelling of tyre inner chamber: towards real time applications". In: *IOP Conference Series: Materials Science and Engineering* 922 (2021). DOI: 10.1007/s11012-021-01310-w.

[24] Frank Naets et al. "Design and Experimental Validation of a Stable Two-Stage Estimator for Automotive Sideslip Angle and Tire Parameters". In: *IEEE Transactions on Vehicular Technology* 66.11 (2017), pp. 9727–9742. DOI: 10.1109/TVT.2017.2742665.

[25] Arash Hosseinian and Selahattin Baslamisli. "ADAP-TYRE: DEKF Filtering For Vehicle State Estimation Based On Tyre Parameter Adaptation". In: *International Journal of Vehicle Design* (Dec. 2015). DOI: 10.1504/IJVD.2016.078769.

[26] Kistler. *6-DOF CAN Inertial Measurement Unit datasheet.* URL: https://www.kistler.com.

[27] Pieter Abbeel et al. "Discriminative Training of Kalman Filters". In: June 2005, pp. 289–296. DOI: 10.15607/RSS.2005.I.038.

[28] Dan Simon. *Optimal State Estimation.* John Wiley & Sons, Inc., 2006.

[29] Moustapha Doumiati et al. "Vehicle Dynamics Estimation using Kalman Filtering: Experimental Validation". In: *Vehicle Dynamics Estimation using Kalman Filtering: Experimental Validation* (Dec. 2012). DOI: 10.1002/9781118578988.

[30] S. J. Julier and J. K. Uhlmann. "New extension of the Kalman filter to nonlinear systems". In: *Signal Processing, Sensor Fusion, and Target Recognition* 3068 (1997), pp. 182–193.

[31] *The spherical simplex unscented transformation.* Vol. 3. 2003, 2430–2434 vol.3. DOI: 10.1109/ACC.2003.1243439.

[32] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte. "A new method for the nonlinear transformation of means and covariances in filters and estimators". In: *IEEE Transactions on Automatic Control* 45.3 (2000), pp. 477–482.

[33] *Unscented Filtering and Nonlinear Estimation.* Vol. 92. 3. 2004, pp. 401–422.

[34] Zhe Chen. "Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond". In: *Statistics* 182 (Jan. 2003). DOI: 10.1080/02331880309257.

[35] Tiancheng Li, Miodrag Bolic, and Petar M. Djuric. "Resampling Methods for Particle Filtering: Classification, implementation, and strategies". In: *IEEE Signal Processing Magazine* 32.3 (2015), pp. 70–86. DOI: 10.1109/MSP.2014.2330626.

[36] H. B. Pacejka. *Tire and vehicle dynamics / Hans B. Pacejka.* eng. Warrendale, PA: Published on behalf of Society of Automotive Engineers, 2002. ISBN: 0768011264.

[37] Flavio Farroni. "T.R.I.C.K.-Tire-Road Interaction Characterization & Knowledge - A tool for the evaluation of tire and vehicle performances in outdoor test sessions". In: *Mechanical Systems and Signal Processing* 72-73 (Dec. 2015), pp. 808–831. DOI: 10.1016/j.ymssp.2015.11.019.