# Odometry 2.0: A Slip-Adaptive EIF-Based Four-Wheel-Odometry Model for Parking

Alexander Brunker, Thomas Wohlgemuth, Michael Frey, and Frank Gauterin

***Abstract*—This paper investigates the possibility of estimating the actual pose of a passenger car while parking by adaptive multi-sensor data fusion. The newly developed Odometry 2.0 estimator includes multiple dead-reckoning models implemented by the adaptive measurement model of an Extended Information filter. The measurements of orientation and velocity of each of the four wheels, corrected kinematic side-slip angles and the yaw rate serve as input for the estimator. The chosen architecture allows the measurements to be combined individually. This makes it possible to deal with systematic errors in modeling, depending on the quality of the measurements, by changing the composition of the measurement model. In addition, a Slip Detection Algorithm (SDA) is implemented, which sets the information measurement covariance of the slipping wheel measurement to 0, so that the inaccurate measurement cannot influence the estimated pose. Separate modeling of translation and rotation allows to estimate the vehicle's position from the information of a single wheel, making it a true four-wheel-odometry model. Experimental results confirm the excellent performance of the Odometry 2.0 estimator in contrast to the well-known state-of-the-art dead-rocking models for parking, analyzed by position data obtained from a Dual-DGNSS-IMU reference system. The SDA was tested under real conditions on partially snow-covered areas.**

***Index Terms*—Dead-reckoning, odometry, unscented information filter, adaption, fusion.**

## I. Introduction

AUTOMATIC parking systems are currently one of the most popular driving assistance systems. The customer expects a reproducible, i.e. consistently good, parking result even if the customer does not know the technical limits of the sensors. Guaranteeing a good and consistent parking result requires a well-calibrated and robust localization of the vehicle during the entire parking process. Estimation of the pose of the vehicle with the help of odometry for application to parking systems is an efficient and well-suited method. The calibration of the kinematic parameters for the odometry-based localization in difficult environments is a well-explored area [1], [2]. There are also a number of papers on the topic of multi-sensor data fusion for localization [3], [4], [5], [6], but this paper focuses on the method of odometry-based dead-reckoning localization itself and presents a method of intelligent adaption for application to a vehicle while parking.

Manuscript received November 23, 2017; revised April 28, 2018; accepted August 19, 2018. Date of publication December 13, 2018; date of current version February 22, 2019. *(Corresponding author: Alexander Brunker.)*

A. Brunker is with the Department of Parking Systems, Daimler AG, Sindelfingen 71063, Germany, and also with the Institute of Vehicle System Technology, Karlsruhe Institute of Technology, Karlsruhe 76131, Germany (e-mail: alexander.brunker@gmail.com).

T. Wohlgemuth is with the Department of Parking Systems Daimler AG, Sindelfingen 71063, Germany (e-mail: thomas.wohlgemuth@daimler.com).

M. Frey and F. Gauterin are with the Institute of Vehicle System Technology, Karlsruhe Institute of Technology, Karlsruhe 76131, Germany (e-mail: michael.frey@kit.edu; frank.gauterin@kit.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIV.2018.2886675

### A. Literature Review

There are basically three state-of-the-art dead-reckoning models that can be used for localization during parking.

- For the parking systems, the *Kinematic-Double-Track Model* is conventionally used where the movement is constructed from only the input of the two velocities of the non-steered rear wheels. This model comes from robotics and if the wheels do not slip, this methodology provides a surprisingly accurate description of the movement despite its simplicity [7]. The slip of individual wheels, however, causes the model to deviate from the true trajectory. The *Extended Kinematic-Double-Track Model* is an extension of the model and uses all four independent wheels with their respective wheel angle.
- The most common vehicle model is the *Single-Track Model*, from the field of Electronic Stability Control (ESC) [8], which describes the movement of the vehicle around the center of gravity with kinematic and dynamic inputs from which a kinematic model, the *Kinematic-Single-Track Model*, can be derived. It separates the input of the translation motion by the average of the absolute value of the velocities of the rear wheels and the rotation by the virtual wheel angles. This model is extremely sensitive to changes in the orientation angle, which is influenced by several errors such as tolerances, wheel alignment errors caused by the Ackermann error [9] and elastokinematic multi-angles. These multi-angles are based on elastokinematic effects, as the axes are influenced by different force directions when driving forwards and backwards. The *Corrected Kinematic Single-Track Model* uses direction-dependent corrected kinematic side-slip angles.
- In addition, a kinematic model, the *Inertial-Sensor Model*, can be derived in which the translation motion is determined from the double integrated acceleration and the rotation from the integrated yaw rate. A *Yaw-Rate-Odometry Model* was constructed [10] and it uses the yaw rate for the rotation and the absolute value of the velocities of the

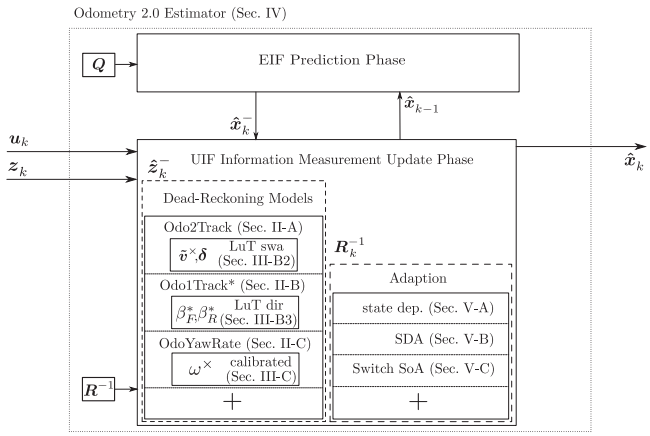


Fig. 1. Odometry 2.0 estimator block diagram.

rear wheels for the translation motion as accelerations on the basis of the specification are too imprecise.

### B. Outline and Main Contribution

An Extended Information filter (EIF) is constructed to deal with the technical challenges of the fusion of different dead-reckoning models, depending on the availability and quality of sensor data. Here the EIF is essentially a Bayes filter which uses the Kalman filter equations expressed in terms of the inverse of the covariance matrix. Its structure is computationally simpler than the Kalman filter [11] especially if the measurement vectors are larger than the state vectors [12]. This means that in the innovation phase no inversion of the measurement covariance matrix is necessary, since we directly use an inverse form. This allows adapting entries of the matrix up to 0 to switch off inputs for the estimator. The non-linear system model operates without input and solely with the information of the previous state in the prediction phase (Systems Theory: Autonomous System). The input to the estimator comes only during the innovation resp. information measurement update phase and from the corresponding non-linear measurement model, so that the number of measurements is easily adapted without affecting the system model operation.

The Odometry 2.0 estimator architecture (Fig. 1) allows adapting the measurement model while changing the composition of the dead-reckoning models all the way to switching off a complete model. The composition of the dead-reckoning models can be changed in every time step through an adaptive information covariance matrix. No stability is lost because care is taken to switch off only redundant measurements. The architecture allows easily switching off entire measurements when non-systematic errors such as wheel slip are detected. A Slip Detection Algorithm (SDA) makes the localization robust towards the longitudinal slippage of individual wheels during acceleration or braking maneuvers. Furthermore, fusing the information of the individual dead-reckoning models means the errors in the measurements do not affect the operation accuracy as much as in the state-of-the-art dead-reckoning models. Besides, it is easily possible to extend the number of measurements when the sensor set expands, e.g., through optical-flow-based Visual Odometry [13] or Radar-based ego-motion [14], if the sensor data can be constructed as in the published patent application [15] describes.

To check the performance of the Odometry 2.0 estimator, results were generated by verifying the models with test drives using a very accurate Dual-DGNSS-IMU-reference measurement system (Fig. 10). To find the best implementation for the Odometry 2.0 estimator, several possible Bayesian filter types (EKF, UKF, EIF, UIF) were implemented. The evaluation has shown that the EIF is the best filter implementation because it requires the shortest simulation time and the results of the different implementations do not differ significantly. Subsequently, the sensitivity analysis of the Odometry 2.0 estimators showed that it is less sensitive to errors in wheel circumference, wheelbase, yaw rate and wheel angles compared to the three state-of-the-art models. This is the result of the fusion of the equivalent information of the individual measurements in separated dead-reckoning models. Compared to the *Kinematic-Double-Track Model*, the Odometry 2.0 estimator shows better performance because of independent measurements for translation motion and rotation. Also, compared to the *Corrected Kinematic-Single-Track Model*, the algorithm offers the advantage of being able to estimate the rotation from additional sensors information together with the corrected kinematic side-slip angles. Individual measurement uncertainties do not have any effect on the estimated pose of the Odometry 2.0 estimator, as would be the case with the other two models. The *Yaw-Rate-Odometry Model*, on the other hand, is affected only in the translation motion by errors in the wheel circumference or in the rotation by errors in the directly measured yaw rate.

The division of this work is organized as follows: In Section II, the physical derivations of the dead-reckoning models for the localization during parking are shown. A detailed description of the systematic and non-systematic errors follows in Section III. Section IV and V outline the mathematical description of the EIF and its adaptations. Finally, experimental results are shown in Section VI, where the performance is demonstrated in real test scenarios with simulated errors in a sensitivity analysis and with real test scenarios on surfaces partially covered by snow to illustrate the performance of the SDA.

## II. Dead-Reckoning Models

In this section, the state-of-the-art dead-reckoning models for parking are presented and formulated as time-discrete state-space models and the trajectory of the point centered on the rear axle of a vehicle is described (Fig. 2). For parking scenarios it is assumed that the motion of the vehicle follows a circular arc with the length $s_k$ by a constant velocity $v_k$ and rotation rate $\omega_k$ of the vehicle in a time step $\Delta t$ with a curvature factor $\kappa \approx 1$ since small angles are assumed:

$$s_k = v_k \cdot \Delta t \cdot \underbrace{\frac{2 \cdot \sin\left(\frac{\omega_k \cdot \Delta t}{2}\right)}{\omega_k \cdot \Delta t}}_{=\kappa \approx 1} \approx v_k \cdot \Delta t. \tag{1}$$

The motion can then be described with a global coordinate frame $[x, y, \theta]$ at the point centered on the rear axle with a

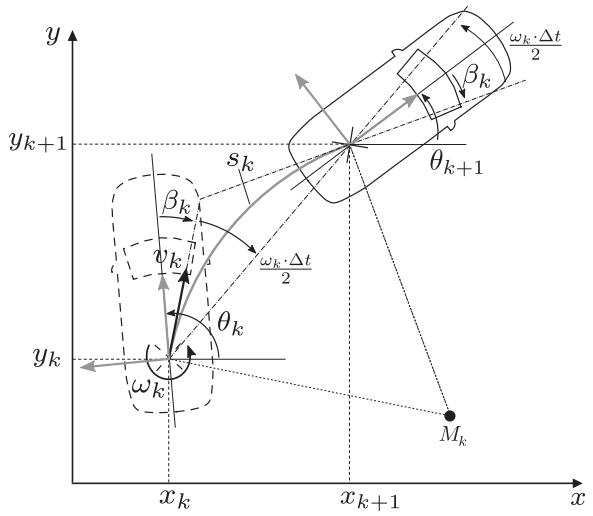


Fig. 2. Motion of the vehicle in the global coordinate system.

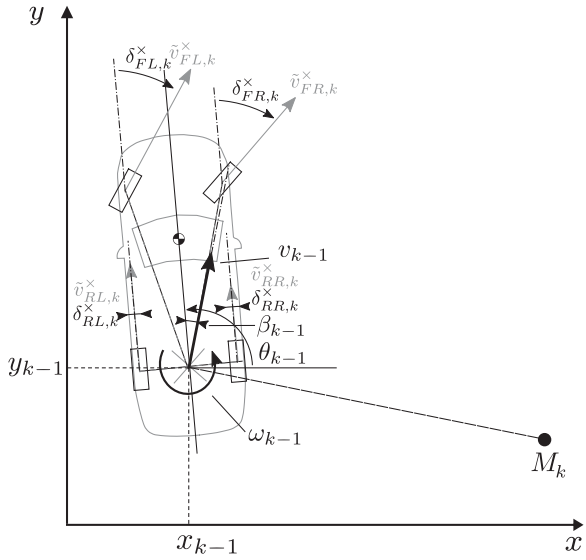


Fig. 3. Relevant motion input for the *Kinematic-Double-Track Model*.

kinematic side-slip angle $\beta_k$ by:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} + \begin{bmatrix} v_k \cdot \Delta t \cdot \cos\left(\beta_k + \theta_k + \frac{\omega_k \cdot \Delta t}{2}\right) \\ v_k \cdot \Delta t \cdot \sin\left(\beta_k + \theta_k + \frac{\omega_k \cdot \Delta t}{2}\right) \\ \omega_k \cdot \Delta t \end{bmatrix}. \tag{2}$$

For dead-reckoning modelling, the following input $\boldsymbol{u}$ of the current sensor set of a production vehicle are given:

1) Vehicle data sheet: wheel contact points in $x$-direction $\boldsymbol{r}_x = \begin{bmatrix} r_{FL,x} & r_{FR,x} & r_{RL,x} & r_{RR,x} \end{bmatrix}^T$,
2) Vehicle data sheet: wheel contact points in $y$-direction $\boldsymbol{r}_y = \begin{bmatrix} r_{FL,y} & r_{FR,y} & r_{RL,y} & r_{RR,y} \end{bmatrix}^T$,
3) Vehicle bus: single wheel velocities $\tilde{\boldsymbol{v}}^{\times} = \begin{bmatrix} \tilde{v}^{\times}_{FL} & \tilde{v}^{\times}_{FR} & \tilde{v}^{\times}_{RL} & \tilde{v}^{\times}_{RR} \end{bmatrix}^T$,
4) Vehicle bus: yaw rate $\omega^{\times}$,
5) Vehicle bus: virtual front and rear wheel angle $\delta^{\times}_F$, $\delta^{\times}_R$,
6) Steering Measurements: single wheel angles $\boldsymbol{\delta} = \begin{bmatrix} \delta_{FL}(\delta^{\times}_F) & \delta_{FR}(\delta^{\times}_F) & \delta_{RL}(\delta^{\times}_R) & \delta_{RR}(\delta^{\times}_R) \end{bmatrix}^T$ (Section III-B2),
7) Steering Measurements: driving direction corrected kinematic side-slip angles $\beta^*_F(\delta^{\times}_F, \delta^{\times}_R, d)$, $\beta^*_R(\delta^{\times}_F, \delta^{\times}_R, d)$ (Section III-B3).

In the following, the subscript $i$ is referenced to the individual wheels $i \in \{FL, FR, RL, RR\}$. Sensor measurements or quantities derived from the sensor model are illustrated with $^{\times}$ and empirically corrected quantities are illustrated with $^*$.

### A. Kinematic-Double-Track Model

The *Kinematic-Double-Track Model* determines the pose solely on the basis of two single wheel velocities. The translation motion is calculated by the average absolute value of the velocities of the two wheels. The rotation is determined on the basis of the velocity-difference with respect to the single wheel angles $\boldsymbol{\delta} \in \mathbb{R}^4$ (Section III-B2) and the wheel contact points $\boldsymbol{r}_x, \boldsymbol{r}_y \in \mathbb{R}^4$. Equation (3) describes how a connection between the velocity on the wheel contact points $\bar{\boldsymbol{v}}_i \in \mathbb{R}^3$, the vehicle velocity vector $\boldsymbol{v} \in \mathbb{R}^3$, the rotation rate vector $\boldsymbol{\omega} \in \mathbb{R}^3$ and the kinematic side-slip angle $\beta$ can be established:

$$\underbrace{\begin{bmatrix} \bar{v}_{i,x} \\ \bar{v}_{i,y} \\ 0 \end{bmatrix}}_{=:\bar{\boldsymbol{v}}_i} = \underbrace{\begin{bmatrix} v \cdot \cos(\beta) \\ v \cdot \sin(\beta) \\ 0 \end{bmatrix}}_{=:\boldsymbol{v}} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ \omega \end{bmatrix}}_{=:\boldsymbol{\omega}} \times \underbrace{\begin{bmatrix} r_{i,x} \\ r_{i,y} \\ 0 \end{bmatrix}}_{=:\boldsymbol{r}_i}. \tag{3}$$

From the velocity on the wheel contact points $\bar{\boldsymbol{v}}_i$ only the component of the translation motion direction, defined by the single wheel angle $\delta_i$, is applicable (Fig. 3). The translation motion direction is the true direction in which the wheel is rolling:

$$\begin{aligned} \tilde{v}_i &= \bar{v}_{i,x} \cdot \cos(\delta_i) + \bar{v}_{i,y} \cdot \sin(\delta_i) \\ &= v \cdot \cos(\delta_i) \cdot \cos(\beta) + v \cdot \sin(\delta_i) \cdot \sin(\beta) \\ &\quad + \omega \cdot (r_{i,x} \cdot \sin(\delta_i) - r_{i,y} \cdot \cos(\delta_i)) \\ &= v \cdot \cos(\delta_i - \beta) + \omega \cdot (r_{i,x} \cdot \sin(\delta_i) - r_{i,y} \cdot \cos(\delta_i)). \end{aligned} \tag{4}$$

With the help of Equation 4 the velocity for two individual wheels $v_1, v_2$ can be set up and solved to determine the velocity of the vehicle $v$,

$$v_{1,2} = \frac{\begin{matrix} r_{1,x} \cdot v_2 \cdot \sin(\delta_1) - r_{1,y} \cdot v_2 \cdot \cos(\delta_1) \\ -r_{2,x} \cdot v_1 \cdot \sin(\delta_2) + r_{2,y} \cdot v_1 \cdot \cos(\delta_2) \end{matrix}}{\begin{matrix} r_{1,x} \cdot \sin(\delta_1)\cos(\delta_2 - \beta) - r_{1,y} \cdot \cos(\delta_1)\cos(\delta_2 - \beta) \\ -r_{2,x} \cdot \sin(\delta_2)\cos(\delta_1 - \beta) + r_{2,y} \cdot \cos(\delta_2)\cos(\delta_1 - \beta) \end{matrix}}, \tag{5}$$

and solved to find the rotation rate of the vehicle $\omega$,

$$\omega_{1,2} = \frac{v_1 \cdot \cos(\delta_2 - \beta) - v_2 \cdot \cos(\delta_1 - \beta)}{\begin{array}{l} r_{1,x} \cdot \sin(\delta_1)\cos(\delta_2 - \beta) - r_{1,y} \cdot \cos(\delta_1)\cos(\delta_2 - \beta) \\ -r_{2,x} \cdot \sin(\delta_2)\cos(\delta_1 - \beta) + r_{2,y} \cdot \cos(\delta_2)\cos(\delta_1 - \beta) \end{array}}. \tag{6}$$

For a vehicle without rear axle steering $\delta_R^\times = 0 \Rightarrow \delta_{RL}$, $\delta_{RR} = 0$ the simplest form can be build: the *Kinematic-Double-Track Model* (Odo2Track) sometimes also called *Differential-Velocity Model*:

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \\ \beta_k \\ v_k \\ \omega_k \end{bmatrix} = \begin{bmatrix} x_{k-1} + v_{k-1} \cdot \Delta t \cdot \cos\left(\beta_{k-1} + \theta_{k-1} + \frac{\omega_{k-1} \cdot \Delta t}{2}\right) \\ y_{k-1} + v_{k-1} \cdot \Delta t \cdot \sin\left(\beta_{k-1} + \theta_{k-1} + \frac{\omega_{k-1} \cdot \Delta t}{2}\right) \\ \theta_{k-1} + \omega_{k-1} \cdot \Delta t \\ 0 \\ \frac{\tilde{v}_{RL,k}^\times + \tilde{v}_{RR,k}^\times}{2} \\ \frac{\tilde{v}_{RR,k}^\times - \tilde{v}_{RL,k}^\times}{(r_{RR,y} - r_{RL,y})} \end{bmatrix}. \tag{7}$$

### B. Kinematic-Single-Track Model

For the *Kinematic-Single-Track Model* (Odo1Track), the translation motion is determined by the average velocity of the rear wheels. The rotation is determined by the geometric orientation of the kinematic side-slip angles $\beta_F$, $\beta_R \in (-\frac{\pi}{2}, \frac{\pi}{2})$. To calculate the rotation rate $\omega$, the fact is used that the velocity vector $\boldsymbol{v} \in \mathbb{R}^3$, the length vector $\boldsymbol{r}_{RM} \in \mathbb{R}^3$ and the rotation rate vector $\boldsymbol{\omega} \in \mathbb{R}^3$ are perpendicular to each other:

$$\boldsymbol{v} = \boldsymbol{\omega} \times \boldsymbol{r}_{RM} \Rightarrow \|\boldsymbol{v}\| = \|\boldsymbol{\omega} \times \boldsymbol{r}_{RM}\| \Leftrightarrow \omega = \frac{v}{r_{RM}}. \tag{8}$$

Fig. 4 shows how to calculate the length $r_{RM}$ to the instantaneous center of rotation $M$, by set the projection of the triangle ($F$, $R$, $M$) with the help of the wheel base $r_b$ and the cosine theorem:

$$r_{RM} = r_b \cdot \cos\left(\frac{\pi}{2} + \beta_R\right) + r_{FM} \cdot \cos\left(\beta_F - \beta_R\right). \tag{9}$$

Subsequently, $r_{FM}$ is substitute by $r_b \cdot \frac{\sin\left(\frac{\pi}{2} + \beta_R\right)}{\sin(\beta_F - \beta_R)}$ using the sine theorem:

$$r_{RM} = r_b \cdot \left(\cos\left(\frac{\pi}{2} + \beta_R\right) + \frac{\sin\left(\frac{\pi}{2} + \beta_R\right)}{\tan\left(\beta_F - \beta_R\right)}\right). \tag{10}$$

The symmetries of sine and cosine continue to:

$$r_{RM} = r_b \cdot \left(\frac{1}{\cos(\beta_R) \cdot (\tan(\beta_F) - \tan(\beta_R))}\right) = \frac{v}{\omega}, \tag{11}$$

solving Equation (11) for the rotation rate $\omega$ results in:

$$\omega = \frac{v}{r_b}\left(\cos(\beta_R) \cdot (\tan(\beta_F) - \tan(\beta_R))\right), \tag{12}$$

and for a vehicle without rear axle steering $\beta_R = 0$ in:

$$\omega = \frac{v}{r_b}\tan(\beta_F). \tag{13}$$

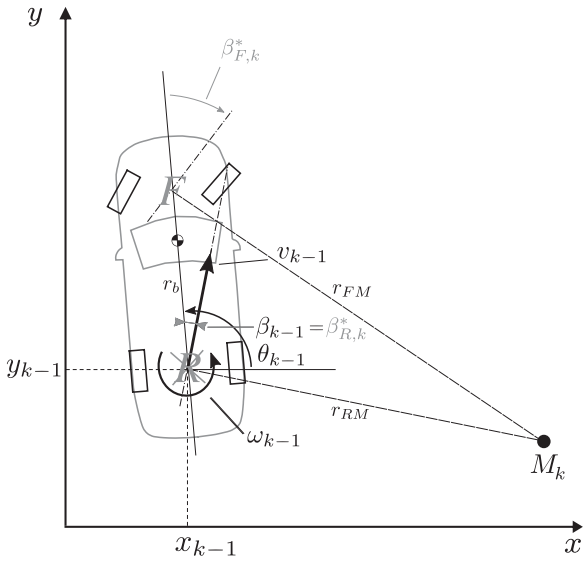


Fig. 4. Relevant motion input for the *Kinematic-Single-Track Model*.

The context from Equation (11) allows also solving for the kinematic side-slip angles on the front axle $\beta_F$:

$$\beta_F = \arctan\left(\frac{\omega \cdot r_b}{v \cdot \cos(\beta_R)} + \tan(\beta_R)\right). \tag{14}$$

The state-space equation for the *Corrected Kinematic-Single-Track Model* (Odo1Track*), which uses the driving direction dependent corrected kinematic side-slip angles $\beta_F^*(\delta_F^\times, \delta_R^\times, d)$, $\beta_R^*(\delta_F^\times, \delta_R^\times, d)$ (Section III-B3) is then assembled to:

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \\ \beta_k \\ v_k \\ \omega_k \end{bmatrix} = \begin{bmatrix} x_{k-1} + v_{k-1} \cdot \Delta t \cdot \cos\left(\beta_{k-1} + \theta_{k-1} + \frac{\omega_{k-1} \cdot \Delta t}{2}\right) \\ y_{k-1} + v_{k-1} \cdot \Delta t \cdot \sin\left(\beta_{k-1} + \theta_{k-1} + \frac{\omega_{k-1} \cdot \Delta t}{2}\right) \\ \theta_{k-1} + \omega_{k-1} \cdot \Delta t \\ \beta^*_{R,k} \\ \frac{\tilde{v}_{RL,k}^\times + \tilde{v}_{RR,k}^\times}{2} \\ \frac{v_{k-1}}{r_b}\left(\cos(\beta^*_{R,k}) \cdot (\tan(\beta^*_{F,k}) - \tan(\beta^*_{R,k}))\right) \end{bmatrix}, \tag{15}$$

with the wheelbase $r_b$:

$$r_b = \frac{r_{FL,x} + r_{FR,x}}{2} - \frac{r_{RL,x} + r_{RR,x}}{2}. \tag{16}$$

### C. Yaw-Rate-Odometry Model

The *Yaw-Rate-Odometry Model* (OdoYawRate) uses the yaw rate $\omega^\times$ directly measured by the gyroscope sensor as the information for the rotation (Fig. 5).

The model also uses the average rear wheel velocities as the information on the translation motion. For a vehicle without rear

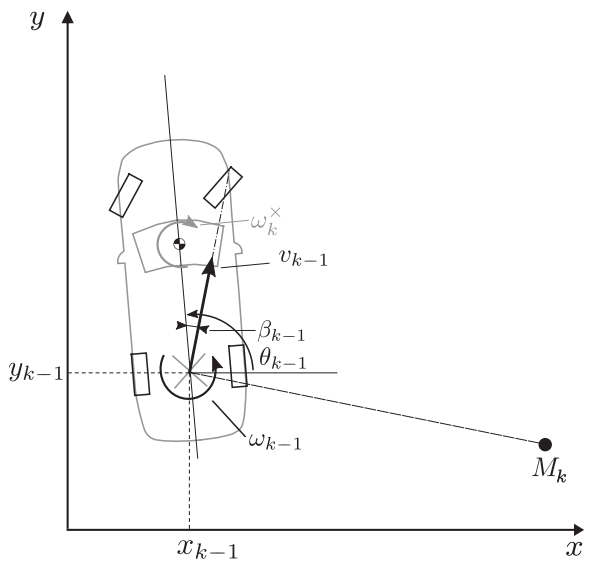


Fig. 5. Relevant motion input for the *Yaw-Rate-Odometry Model*.

axle steering $\delta_R^\times = 0$, one obtains:

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \\ \beta_k \\ v_k \\ \omega_k \end{bmatrix} = \begin{bmatrix} x_{k-1} + v_{k-1} \cdot \Delta t \cdot \cos\left(\beta_{k-1} + \theta_{k-1} + \dfrac{\omega_{k-1} \cdot \Delta t}{2}\right) \\ y_{k-1} + v_{k-1} \cdot \Delta t \cdot \sin\left(\beta_{k-1} + \theta_{k-1} + \dfrac{\omega_{k-1} \cdot \Delta t}{2}\right) \\ \theta_{k-1} + \omega_{k-1} \cdot \Delta t \\ 0 \\ \dfrac{\tilde{v}_{RL,k}^\times + \tilde{v}_{RR,k}^\times}{2} \\ \omega_k^\times \end{bmatrix}. \tag{17}$$

## III. Systematic and Non-Systematic Errors

There are two different classes of specified errors: systematic and non-systematic errors. Systematic errors in the measurements are vehicle specific and change slowly while driving. Systematic errors contribute to inaccuracies in odometry calculation. However, by knowing the effects of the dominant sources of error, this effect can be eliminated by calibration, which reduces the influence on the calculations. Errors in odometry calculations arise either from uncertainties in parameters used in the model or the model itself, where non-modelled dynamics may cause the false caption of the operation data. The limits of the errors are expressed with the values $A$ and $B$ and the errors are summaries in the error factor $\rho \in \{\Delta c_i, \Delta c_{RR}, \Delta t_R, \Delta\delta_F, \Delta\omega\}$.

### A. Systematic Errors by Wheel Setup

*1) Wheel Circumference Error:* The radii are greatly dependent on the dimensions of the mounted wheels (specific wheel type: rims and tires). On a specific vehicle model, the delta value of the wheel circumference $c_{default} = 2.080$ m is assumed to lie between $\rho_A(\Delta c_i) := -0.040$ m $\leq \Delta c_i \leq \rho_B(\Delta c_i) := +0.030$ m, depending on the wheel type (experimentally measured). When the value of the radius is poorly calibrated, this corresponds to a systematic error in the calculated traveled distance of the vehicle. An error in the wheel circumference calibration results in an error in the single wheel velocity:

$$\tilde{v}_{i,k}^\times := \frac{w_{i,k}^\times \cdot d \cdot (c_{default} + \Delta c_i)}{w_{max} \cdot \Delta t},$$
$$w_{i,k}^\times := \mathrm{mod}\left[\left(En_{i,k}^\times - En_{i,k-1}^\times\right), En_{max}\right]. \tag{18}$$

The encoder value $En_{i,k}^\times$ represents the pulse count of the wheel encoder sensor at the time $k$ and for the Modulo-operation: $En_{max}$ is the maximum value of the counter and $w_{max}$ is the number of counts for one revolution. Further, the average velocity $\tilde{v}_k^\times$ is defined as the respective movement and can be described in terms of delta-wheels-pulse count $w_i^\times$. The direction signal $d \in \{-1, 1\}$ is detected by the change in the magnetic field of the encoder sensor. Because there could be a delayed detection of the direction signal, other methods have been developed to obtain a faster direction estimation [16], [17] and additional methods have been developed to use these signals in a Bayes-filter environment [18]. The standard deviation of the velocity measurement would be assumed with 0.01 m/s.

*2) Track Width Error:* The vehicle track width is defined as the distance between the two wheel contact points for the same axle. The wheel contact points depend on the current wheel type and is an example of a systematic error. For example, a wider wheel pushes the wheel contact points towards the vehicle center-line, because the outer edge is limited by the vehicle contour. Also, the degradation of the tires and the alignment of the wheels have an impact on the contact points. On a specific vehicle model, the delta value of the track width of the rear axle $t_R = 1.604$ m is assumed to lie in a span of $\rho_A(\Delta t_R)$ : $-0.020$ m $\leq \Delta t_R \leq \rho_B(\Delta t_R)$ : $+0.016$ m depending on the wheel type (specifications) and is defined as:

$$r_{RL,y} = -r_{RR,y} := \frac{t_R + \Delta t_R}{2}. \tag{19}$$

### B. Systematic Errors in the Steering

*1) Tolerance From the Rack Position Sensor:* The virtual wheel angle is determined indirectly by the measurement of the rack position $x_{\delta_F}^\times$ and a steering kinematic look-up-table $\mathrm{LuT}_{\mathrm{steer}}(\cdot)$. The toothed rack can be positioned by tolerances or by adjusting the tie rods. This ensures an offset in the value and represents a systematic error. The virtual wheel angle tolerance is assumed to lie between $\rho_A(\Delta\delta_F) := -1^\circ \leq \Delta\delta_F \leq \rho_B(\Delta\delta_F) := +1^\circ$ (specifications) and is defined as:

$$\delta_F^\times := \mathrm{LuT}_{\mathrm{steer}}(x_{\delta_F}^\times) + \Delta\delta_F. \tag{20}$$

*2) Virtual Wheel Angles Depended Single Wheel Angles:* The virtual wheel angles $\delta_F^\times$, $\delta_R^\times$ from the vehicle bus are obtained from the average angle of the two wheels of one axle. This was introduced, because of the Electronic Stability Control (ESC), to obtain an input for the *Single-Track Model* which is specified on small wheel angles. For large wheel angles, the *Kinematic-Double-Track Model* needs the true single wheel angles $\boldsymbol{\delta}$ to take the velocities measurements into account. The

offline steering test stand measurements provide the original single wheel angles $\boldsymbol{\delta}$ depending on the virtual wheel angle $\delta^{\times}$ in the single wheel look-up-tables $\mathrm{LuT}_{\mathrm{steer,swa,F}}(\cdot)$ and $\mathrm{LuT}_{\mathrm{steer,swa,R}}(\cdot)$:

$$\boldsymbol{\delta} = \begin{bmatrix} \delta_{FL} \\ \delta_{FR} \\ \delta_{RL} \\ \delta_{RR} \end{bmatrix} := \begin{bmatrix} \mathrm{LuT}_{\mathrm{steer,swa,F}}(\delta_F^{\times}) \\ \mathrm{LuT}_{\mathrm{steer,swa,R}}(\delta_R^{\times}) \end{bmatrix}. \tag{21}$$

*3) Driving Direction Depended Kinematic Side-Slip Angles:* Large wheel angles are achieved while parking and the wheels do not roll in rolling direction. This happens because the axis kinematics builds wheel angle errors known as Ackermann's error [9]. In addition, the wheel angles $\delta_F^{\times}, \delta_R^{\times}$ have different offsets depending on the driving direction of the vehicle. These are based on elastokinematic effects, as the axes are influenced by different force directions when driving forwards and backwards. These effects result into skew angles $\alpha_i$ and elastokinematic multi-angles $\Delta\delta_i$. It applies: $\beta = \delta^{\times} + \Delta\delta + \alpha$. These effects are corrected by test drives with the Dual-DGNSS-IMU reference measuring system of Section VI. During these test drives, turning circles, forward and backward, were performed to create the driving direction corrected kinematic side-slip angle look-up-tables $\mathrm{LuT}_{\mathrm{steer,d,F}}(\cdot)$ and $\mathrm{LuT}_{\mathrm{steer,d,R}}(\cdot)$:

$$\begin{aligned} \beta_F^* &:= \mathrm{LuT}_{\mathrm{steer,d,F}}(\delta_F^{\times}, \delta_R^{\times}, d), \\ \beta_R^* &:= \mathrm{LuT}_{\mathrm{steer,d,R}}(\delta_F^{\times}, \delta_R^{\times}, d). \end{aligned} \tag{22}$$

This method provides a kinematic side-slip angle of the rear axle $\beta_R^* \neq 0$ without rear axle steering $\delta_R^{\times} = 0$ simply on account of the skew angles $\alpha_i$. The published patent applications [19] and [20] show the entire method in detail. The method determines the values with a standard deviation of 0.4°.

### C. Systematic Errors in the Yaw Rate Sensor

The yaw rate delivered by the gyroscope of the ESC $\omega_{ESC}$ can be affected by two typical kinds of systematic errors: an offset (also called bias) $Drift_{\omega}$ and a scaling factor error $Gain_{\omega}$ [21]. For simplicity, the maximum specified error is defined as $\Delta\omega$ and is assumed to lie between $\rho_A(\Delta\omega) := -0.7\ ^{\circ}/\mathrm{s} \leq \Delta\omega \leq \rho_B(\Delta\omega) := +0.7\ ^{\circ}/\mathrm{s}$ (specifications) and is defined as:

$$\omega^{\times} = (\omega_{ESC} - Drift_{\omega})\frac{1}{Gain_{\omega}} := \omega_{ESC} + \Delta\omega. \tag{23}$$

The yaw rate signal noise is specified as 0.1 °/s.

### D. Non-Systematic Errors by Slip of the Wheels

Non-systematic errors are caused by the vehicle's interaction with non-predictable changes in the environment. Cobblestones, cracks or bumps in the road surface will cause the wheels to move irregularly on the vertical axis. These non-systematic errors arise, e.g., when one or more wheels of the vehicle are spinning due to relatively low friction between the tire and the road (wheel slip). The Odometry 2.0 estimator systematically minimizes the effect of a non-systematic error caused by equivalent measurements. Additionally, it provides the possibility of switching off faulty measurements. The Slip Detection Algorithm (SDA) was developed for this reason. By merging equivalent information for rotation, odometry can also work when only one wheel is rolling, so a logic is included to detect up to three slipping wheels. The Boolean variable for the slipping wheels $\boldsymbol{S}_k$ is calculated in Algorithm 1 and is defined as:

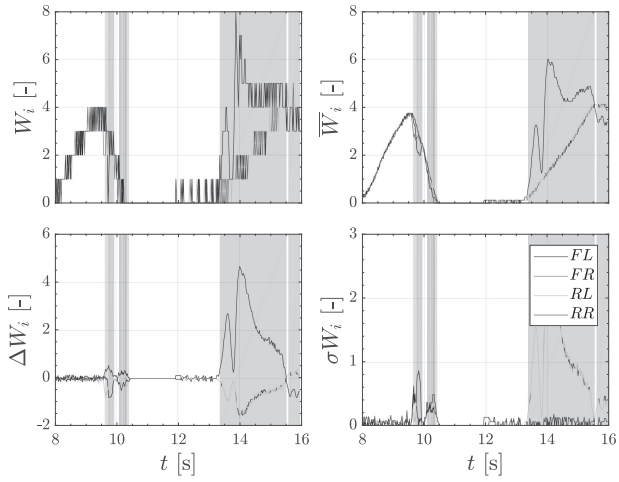


Fig. 6. $W(t)$, $\overline{W}(t)$, $\Delta\overline{\overline{W}}(t)$ and $\sigma\overline{\overline{W}}(t)$ for slipping wheels calculated by the SDA.

$$\boldsymbol{S}_k = \mathrm{SDA}(\boldsymbol{W}_k) \in \mathbb{R}^4. \tag{24}$$

A normalized wheel-pulse count vector $\boldsymbol{W}_k$ which serves as input for Algorithm 1 is first defined. The individual normalized wheel-pulse count $W_{i,k}$ is determined as follows:

$$W_{i,k} = w_{i,k}^{\times} \cdot \frac{c_{default}}{c_{default} + \Delta c_i} \cdot \frac{||\boldsymbol{r}_{RM,k}||}{||\boldsymbol{r}_{i,k}(\delta_F^{\times}, \delta_R^{\times}) - \boldsymbol{r}_{RM,k}||}. \tag{25}$$

Fig. 6 shows how the SDA works when the vehicle's right side is on a low-friction surface. Top left and right, the influence of the step size of the floating average of the normalized wheel-pulse count is shown (cf. Algorithm 1 line 7). The floating average leads to better performance, since the quantization error is taken less into account. It can be seen that the SDA detects the slip on the right wheels at approx. 10 s, the colored background indicates that slippage has been detected on the corresponding wheel. In Fig. 6, bottom left, the differences of the remainder of the normalized wheel-pulse count to the other wheels and ,bottom right, the standard deviation are shown, calculated in Algorithm 1 Line 15+16. One can clearly see why a slip is detected by second 13.4, $\Delta\overline{\overline{W}}_{RR}$ is above the threshold $\Delta\overline{\overline{W}}_{min}$ and the standard deviation $\sigma\overline{\overline{W}}_{RR}$ is within the allowed range $\sigma\overline{\overline{W}}_{max}$. If more than two wheels slip, a distinction is made between launching and braking. During deceleration, it is assumed that regardless of the type of drive, the wheel always rolls with the highest speed through the Drive Slip Control (DSC). When launching, it is assumed that for a 2WD (2-Wheel Drive), the wheels of the non-driven axle and in the case of a 4WD vehicle the wheel at the lowest velocity rolls.

**Algorithm 1:** SDA: Slip Detection Algorithm.

**function** SDA(W(4,1))
  $S = false(4,1)$ ▷ *default no slip*
  $\Delta\overline{\overline{W}}_{min} = 0.15$ ▷ *tuning parameter*
  $\sigma\overline{\overline{W}}_{max} = 0.20$ ▷ *tuning parameter*
  $n = 5$ ▷ *tuning parameter*
  $J = [2,3,4;1,3,4;1,2,4;1,2,3]$
  $\overline{W} = \text{floatingmean}(W, n)$
  $\overline{acc} = \text{diff}(\text{mean}(\overline{W}))$
  **for** $j = 1:4$ **do**
    **for** $k = 1:3$ **do**
      $\Delta\overline{W}(j,k) = \overline{W}(j) - \overline{W}(J(j,k))$
    **end for**
  **end for**
  **for** $j = 1:4$ **do**
    $\Delta\overline{\overline{W}}(j) = \text{mean}(\Delta\overline{W}(j,:))$
    $\sigma\overline{\overline{W}}(j) = \text{std}(\overline{W}(J(j,:)))$
    **if** $acc > 0$ **then**
      $S(j) = \Delta\overline{\overline{W}}(j) > \Delta\overline{\overline{W}}_{min} \wedge \sigma\overline{\overline{W}}(j) < \sigma\overline{\overline{W}}_{max}$
    **else**
      $S(j) = \Delta\overline{\overline{W}}(j) < -\Delta\overline{\overline{W}}_{min} \wedge \sigma\overline{\overline{W}}(j) < \sigma\overline{\overline{W}}_{max}$
    **end if**
    **if** $\text{mean}(S) == true$ **then** ▷ *if all wheels slip*
      **if** $\overline{acc} > 0$ **then** ▷ *full acceleration*
        $[\sim, j] = \min(\text{abs}(\overline{W}))$
        $S(j) = false$
        **if** $2WD$ **then** ▷ *2WD: slip ≤ 2 Wheels*
          $\overline{W}(j) = \text{mean}(\overline{W})$
          $[\sim, j] = \min(\text{abs}(\overline{W}))$
          $S(j) = false$
        **else** ▷ *4WD: slip ≤ 3 Wheel*
        **end if**
      **else** ▷ *full breaking*
        $[\sim, j] = \max(\text{abs}(\overline{W}))$
        $S(j) = false$
      **end if**
    **end if**
  **end for**
  **return** $S(4,1)$
**end function**

## IV. EIF-Based Four-Wheel-Odometry Model

In order to create a robust and accurate estimation of the pose of the vehicle, the Extended Information filter (EIF) for multi-sensor data-fusion has been developed. The use of the EIF allows merging information from different sensor measurements to obtain an accurate estimation performance. Here, the non-linear system model is used for state prediction and linearized using the partial derivatives. The innovation resp. information measurement update phase of the EIF corrects the predicted states with the help of multiple different sensors. New information from the sensors enters the system model only during through the adaptable information measurement update phase.

### A. EIF State Definition

The used system model is derived as a dead-reckoning model for parking (cf. Equation (2)). However, in this case the angle of the velocity $\beta_k$, the velocity $v_k$ and the rotation rate $\omega_k$ of the vehicle are included in the state vector $\boldsymbol{x}_k$ instead of being observed as measurements. This method has the benefit in the estimation of these states of providing information on equivalent measurements. Multiple sensors act together to provide measurement correction for the same state and therefore, if properly tuned and used, increase the accuracy compared to using a single sensor. The state vector is then given by:

$$\boldsymbol{x}_k = \begin{bmatrix} x_k & y_k & \theta_k & \beta_k & v_k & \omega_k \end{bmatrix}^T \in \mathbb{R}^6. \tag{26}$$

### B. EIF Linearization

The EIF deals with a problem of estimation of process condition of a nonlinear time-discrete system. It uses the first-order Taylor linearization around the current operating point to linearize the nonlinear system model and uses the partial derivations of the process and measurement functions to calculate predictions of the future process covariance, which are defined as:

$$\boldsymbol{F}_k = \frac{\partial \boldsymbol{f}}{\partial \hat{\boldsymbol{x}}_k}, \quad \boldsymbol{H}_k = \frac{\partial \boldsymbol{h}}{\partial \hat{\boldsymbol{x}}_k}. \tag{27}$$

### C. EIF System Model Prediction Phase

In the system model prediction phase of the EIF, the state $\hat{\boldsymbol{x}}_k$ is estimated without the deterministic noise, which is later depicted by the process covariance $\boldsymbol{Q}_k$ and where $\boldsymbol{P}_k$ is the corresponding covariance matrix of the state vector $\boldsymbol{x}$:

$$\begin{aligned} \hat{\boldsymbol{x}}_k^- &= \boldsymbol{f}\left(\hat{\boldsymbol{x}}_{k-1}, \boldsymbol{u}_k\right), \\ \boldsymbol{P}_k^- &= \boldsymbol{F}_k \cdot \boldsymbol{P}_{k-1} \cdot \boldsymbol{F}_k^T + \boldsymbol{Q}_k. \end{aligned} \tag{28}$$

The system model $\boldsymbol{f}(\cdot)$ predicts the movement of the vehicle:

$$\boldsymbol{f}(\cdot) = \begin{bmatrix} x + v \cdot \Delta t \cdot \cos\left(\beta + \theta + \dfrac{\omega \cdot \Delta t}{2}\right) \\ y + v \cdot \Delta t \cdot \sin\left(\beta + \theta + \dfrac{\omega \cdot \Delta t}{2}\right) \\ \theta + \omega \cdot \Delta t \\ \beta \\ v \\ \omega \end{bmatrix}. \tag{29}$$

The states $\beta_k$, $v_k$, $\omega_k$ are not updated by the system model in the prediction phase and remain constant, the update is performed in the information measurement update phase.

### D. EIF Information Transformation Phase

In the information transformation phase, the prediction equations are derived by implementing the transformation of the information state vector $\hat{\boldsymbol{y}}_k^-$ with the corresponding information

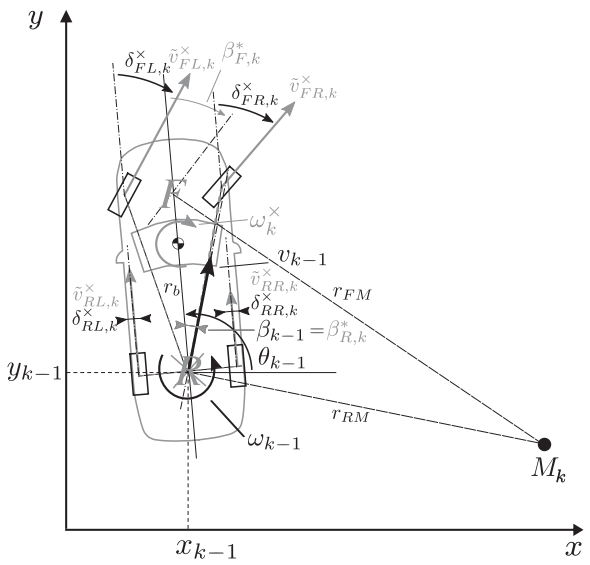


Fig. 7. Relevant motion input for the EIF.

matrix $\hat{\boldsymbol{Y}}_k^-$:

$$\hat{\boldsymbol{y}}_k^- = (\boldsymbol{P}_k^-)^{-1} \cdot \hat{\boldsymbol{x}}_k^-,$$
$$\hat{\boldsymbol{Y}}_k^- = (\boldsymbol{P}_k^-)^{-1}. \tag{30}$$

### *E. EIF Measurement Model Prediction Phase*

The measurement model includes the possibility to introduce as much information about the movement into the system model as possible. In this case, the measured velocities of the wheels $\tilde{\boldsymbol{v}}_k^{\times} \in \mathbb{R}^4$, a summed velocity in $x$-direction $\bar{v}_k^*$, the direct measured yaw rate $\omega_k^{\times}$, the corrected kinematic side-slip angles $\beta_{F,k}^*$, $\beta_{R,k}^*$ (Section III-B3) are identified as measurements and are summarized in a measurement vector $\boldsymbol{z}_k$ (see 7):

$$\boldsymbol{z}_k = \begin{bmatrix} \tilde{\boldsymbol{v}}_k^{\times} & \bar{v}_k^* & \omega_k^{\times} & \beta_{F,k}^* & \beta_{R,k}^* \end{bmatrix}^T \in \mathbb{R}^8,$$
$$\bar{v}_k^* = \frac{\tilde{v}_{RL,k}^{\times} + \tilde{v}_{RR,k}^{\times}}{2}. \tag{31}$$

Here the summed velocity in $x$-direction $\bar{v}_k^*$ was added to easy present the individual models later. The inputs of the measurement model $\boldsymbol{u}_k$ are the single wheel angles $\boldsymbol{\delta}_k \in \mathbb{R}^4$ (Section III-B2), the wheel contact points in both the $x$-direction $\boldsymbol{r}_x \in \mathbb{R}^4$ and $y$-direction $\boldsymbol{r}_y \in \mathbb{R}^4$:

$$\boldsymbol{u}_k = \begin{bmatrix} \boldsymbol{\delta}_k & \boldsymbol{r}_x & \boldsymbol{r}_y \end{bmatrix}^T \in \mathbb{R}^{12}. \tag{32}$$

The predicted state $\hat{\boldsymbol{x}}_k^-$ and the input $\boldsymbol{u}_k$ can be used to calculate the predicted measurement vector $\hat{\boldsymbol{z}}_k$ using the measurement model $\boldsymbol{h}(\cdot)$:

$$\hat{\boldsymbol{z}}_k^- = \boldsymbol{h}\left(\hat{\boldsymbol{x}}_k^-, \boldsymbol{u}_k\right). \tag{33}$$

where the equations are derived from the geometrical correlation of the dead-reckoning models Equation (4) and (14) in Section II:

$$\boldsymbol{h}(\cdot) = \begin{bmatrix} v \cdot \cos(\boldsymbol{\delta} - \beta) + \omega \cdot (\boldsymbol{r}_x \cdot \sin(\boldsymbol{\delta}) - \boldsymbol{r}_y \cdot \cos(\boldsymbol{\delta})) \\ v \\ \omega \\ \hat{\beta}_F \\ \beta \end{bmatrix},$$
$$\hat{\beta}_F = \begin{cases} \arctan\left(\dfrac{\omega \cdot r_b}{v \cdot \cos(\beta)} + \tan(\beta)\right) & \text{if } v \neq 0 \\ \beta & \text{else} \end{cases}. \tag{34}$$

For $v = 0$ the calculated front kinematic side-slip angle $\beta_F$ is not well defined.

### *F. EIF Information Measurement Update Phase*

In the information measurement update phase, the measurement vector $\boldsymbol{z}_k$ and the predicted measurement vector $\hat{\boldsymbol{z}}_k^-$ are merged in order to accurately estimate the innovation of the information state $\boldsymbol{y}_k$ and the associated information matrix $\boldsymbol{Y}_k$ with the help of the measurement covariance matrix $\boldsymbol{R}_k$, the measurement matrix $\boldsymbol{H}_k$ and the cross-correlation covariance matrix $\boldsymbol{P}_{xz,k}$:

$$\hat{\boldsymbol{y}}_k = \hat{\boldsymbol{y}}_k^- + \boldsymbol{H}_k^T \cdot \boldsymbol{R}_k^{-1} \cdot (\boldsymbol{z}_k - \hat{\boldsymbol{z}}_k^- + \boldsymbol{H}_k \cdot \hat{\boldsymbol{x}}_k^-),$$
$$\hat{\boldsymbol{Y}}_k = \hat{\boldsymbol{Y}}_k^- + \boldsymbol{H}_k^T \cdot \boldsymbol{R}_k^{-1} \cdot \boldsymbol{H}_k. \tag{35}$$

### *G. EIF State Transformation Phase*

In the state transformation phase, the measurement updated information vector $\hat{\boldsymbol{y}}_k$ and the information matrix $\hat{\boldsymbol{Y}}_k$ will retransformed into the state vector $\hat{\boldsymbol{x}}_k$ and its corresponding covariance matrix $\boldsymbol{P}_k$:

$$\hat{\boldsymbol{x}}_k = (\hat{\boldsymbol{Y}}_k)^{-1} \cdot \hat{\boldsymbol{y}}_k,$$
$$\boldsymbol{P}_k = (\hat{\boldsymbol{Y}}_k)^{-1}. \tag{36}$$

### *H. EIF Tuning Phase*

Tuning of the above stated EIF is performed with the help of the time-invariant matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$, corresponding to the process and the measurement noises. The diagonal matrices diag([]) for the start covariance matrix $\boldsymbol{P}_0$, the process covariance matrix $\boldsymbol{Q}$ and the measurement covariance matrix $\boldsymbol{R}$ are set to:

$$\boldsymbol{P}_0 = \boldsymbol{Q} = \mathrm{diag}([\sigma_{x,y}^2\ \sigma_{x,y}^2\ \sigma_{\theta}^2\ \sigma_{\beta}^2\ \sigma_{v,f}^2\ \sigma_{\omega,f}^2]),$$
$$\boldsymbol{R} = \mathrm{diag}([\boldsymbol{\Sigma}^2\ \sigma_{\bar{v}^*,h}^2\ \sigma_{\omega,h}^2\ \sigma_{\beta_F}^2\ \sigma_{\beta_R}^2]),$$
$$\boldsymbol{\Sigma}^2 = \mathrm{diag}([\sigma_{\tilde{v},h}^2\ \sigma_{\tilde{v},h}^2\ \sigma_{\tilde{v},h}^2\ \sigma_{\tilde{v},h}^2]). \tag{37}$$

The process and measurement covariance are adapted empirically. As a starting point for the measurement standard deviations the specified sensor noises from Section III were used. The values are summarized in Table I.

## V. EIF Adaption Architecture

In this section, the adaptive adjustments of the EIF are presented. There are basically two different ways of adapting a Bayes filter. The proportions of the process covariance matrix

TABLE I
STANDARD DEVIATIONS FOR THE COVARIANCE MATRICES OF THE EIF

| Variable | Value | Unit | Description |
|---|---|---|---|
| $\sigma_{x,y}$ | $\pm 1 \cdot 10^{-5}$ | m | Position error |
| $\sigma_{\theta}$ | $\pm 1 \cdot 10^{-5}$ | ° | Heading error |
| $\sigma_{\beta}$ | $\pm 1 \cdot 10^{-4}$ | ° | Side-slip error |
| $\sigma_{v,f}$ | $\pm 2 \cdot 10^{-3}$ | m/s | Velocity error |
| $\sigma_{\omega,f}$ | $\pm 3 \cdot 10^{-3}$ | °/s | Yaw rate error |
| $\sigma_{\tilde{v},h}$ | $\pm 1 \cdot 10^{-2}$ | m/s | Velocity noise |
| $\sigma_{\bar{v}^*,h}$ | $\pm 1 \cdot 10^{-2}$ | m/s | Mean velocity noise |
| $\sigma_{\omega,h}$ | $\pm 1 \cdot 10^{-1}$ | °/s | Yaw rate noise |
| $\sigma_{\beta_F}$ | $\pm 4 \cdot 10^{-1}$ | ° | Front side-slip angle noise |
| $\sigma_{\beta_R}$ | $\pm 4 \cdot 10^{-1}$ | ° | Rear side-slip angle noise |

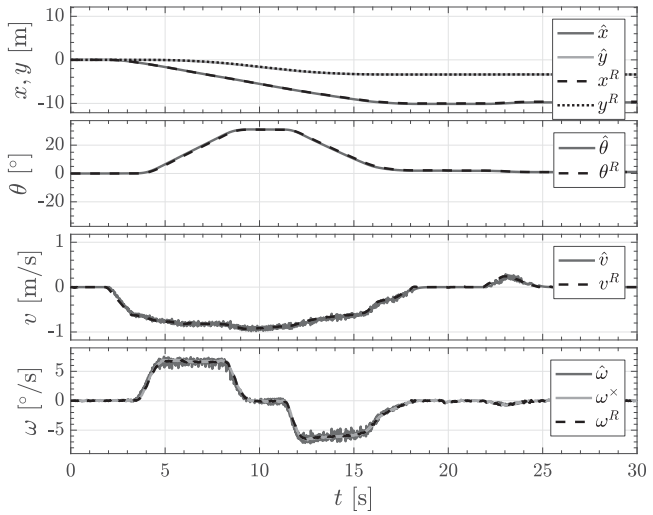


Fig. 8. Values of the measurements of the EIF states and reference value during a longitudinal parking operation.

$\boldsymbol{Q}$ and the measurement covariance matrix $\boldsymbol{R}$ can be changed within each matrix and vice versa. Since the process of the prediction phase uses no inputs, the relationship of the measurement covariance matrix $\boldsymbol{R}$ can be determined. The use of an Information filter with an information covariance matrix $\boldsymbol{R}^{-1}$ concentrates efforts on adaptions of the relation of the time-variant adaptive information covariance matrix $\boldsymbol{R}_k^{-1}$:

$$\boldsymbol{R}_k^{-1} = \boldsymbol{R}^{-1} \cdot \boldsymbol{\Lambda}_k \cdot \boldsymbol{\Pi}_k \cdot \boldsymbol{Z}_k. \tag{38}$$

### A. State-Depending Coefficient Matrix $\boldsymbol{\Lambda}_k$

The information covariance matrix $\boldsymbol{R}_k^{-1}$ can be adapted in a state-dependent manner through $\boldsymbol{\Lambda}_k$. Fig. 8 shows the estimated states of the EIF and corresponding measurements for a longitudinal parking operation. For comparison, the values of the Dual-DGNSS-IMU reference system are shown. The state-dependent tuning can be recognized by the colored background. The estimated position and orientation are close to the reference. The measured yaw rate of the vehicle and the reference are very close to each other, but the estimated velocity and yaw rate oscillates most at higher amplitudes. When launching, the quantization error of the delta-wheels-pulse-counts $w_i^{\times}$ from Equation (18) is noticeable. This is also one of the reasons why the pre-set side-slip angle deviates as much from the measurement as it does in Fig. 9 (middle), but the states are not affected

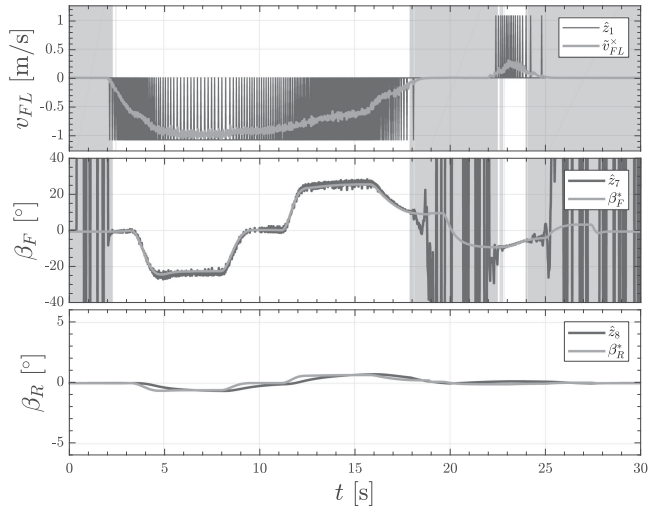


Fig. 9. Values of the model-based measurements of the EIF during a longitudinal parking operation.

because of the coefficients matrix. Within 30 s, the parking is completed.

For the state-dependent tuning the state-dependent coefficient matrix $\boldsymbol{\Lambda}_k$ is defined as:

$$\boldsymbol{\Lambda}_k = \operatorname{diag}\left(\begin{bmatrix} (1:4) & = \begin{cases} 1 \cdot 10^{-2} & \text{if} \quad \|v_k\| < 0.1 \text{ m/s} \\ 1 & \text{else} \end{cases} \\ (5) & = 1 \\ (6) & = 1 \\ (7) & = \begin{cases} 0 & \text{if} \quad \|v_k\| < 0.1 \text{ m/s} \\ 1 & \text{else} \end{cases} \\ (8) & = 1 \end{bmatrix}\right) \in \mathbb{R}^{8\times 8}. \tag{39}$$

### B. SDA-Dependent Coefficient Matrix $\boldsymbol{\Pi}_k$

When incorrect values of a measurement are recognized, the EIF must be adapted and the measurement switched off. Section III-D depicts how a wheel slip detection operates. If the SDA detects a slipping wheel, the corresponding measurement in the measurement model Equation (34) could be switched off. Therefore matrix $\boldsymbol{\Pi}_k$ is included to switch the information covariance value to 0:

$$\boldsymbol{\Pi}_k = \operatorname{diag}([\operatorname{not}(\boldsymbol{S}_k)\ 1\ 1\ 1\ 1]) \in \mathbb{R}^{8\times 8}. \tag{40}$$

### C. Dead-Reckoning Model Depending Coefficient Matrix $\boldsymbol{Z}_k$

In addition, a method is presented to adapt the EIF to the corresponding dead-reckoning models. The different dead-reckoning models can be represented by the following coefficient vector $\boldsymbol{\zeta}_k$ from Table II time-dependent in the coefficient matrix $\boldsymbol{Z}_k$:

$$\boldsymbol{Z}_k = \operatorname{diag}([\boldsymbol{\zeta}_k]) \in \mathbb{R}^{8\times 8}. \tag{41}$$

In addition, the different tuning have different information contents and redundancy for the translation motion $n_v$ and rotation $n_\omega$. The OdoExt2Track can be calculated in six different

TABLE II
COEFFICIENT VECTOR $\boldsymbol{\zeta}_k$ FOR THE DEAD-RECKONING MODELS

| Dead-Reckoning Model | Vector $\boldsymbol{\zeta}_k$ | $n_v$ | $n_\omega$ |
|---|---|---|---|
| Odometry 2.0 | 11111111 | 6 | 8 |
| Odo2Track | 00110000 | 1 | 1 |
| OdoExt2Track | 11110000 | 6 | 6 |
| Odo1Track* | 00001011 | 1(2) | 1 |
| OdoYawRate | 00001100 | 1(2) | 1 |

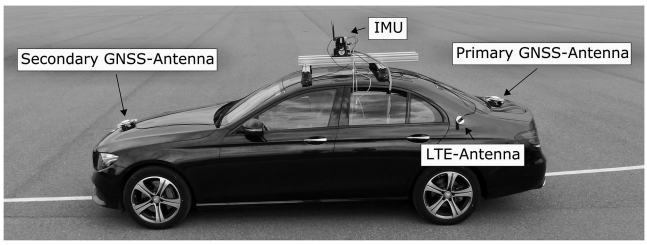


Fig. 10. Test vehicle with Dual-DGNSS-IMU reference measurement system.

ways, depending on which wheel combinations are chosen. The Odo1Track* and OdoYawRate are not more redundant than the Odo2Track because there is no additional velocity measurement, but they are even more robust when a measurement in the velocity is disturbed.

## VI. EXPERIMENTAL VALIDATIONS

In order to test the performance of the proposed Odometry 2.0, test drives with manipulated measurements and real test scenarios on surfaces partially covered by snow have been carried out. A test vehicle was equipped with a Dual-DGNSS-IMU reference system consisting of 2 Differential-Global Navigation Satellite Systems (DGNSS) with RTK-differential data from a ground-based reference stations (communication via LTE). The DGNSS system allows measuring the absolute vehicle position $\boldsymbol{p}^{(R)}$ with $\boldsymbol{p} = [x\ y]^T$ up to an accuracy of $\pm 0.02$ m. Using 2 DGNSS units, the vehicle heading angle $\theta$ can be measured directly without driving to get the entire pose of the vehicle. Fig. 10 illustrates how this was mounted on the vehicle. The absolute pose is supported by the Inertial Measurement Unit (IMU) during a bad GNSS connection and serves as a reference system for the yaw rate. During the different test drives, reference data was saved together with the vehicle signals on a measurement computer.

The measurement drives were carried out under best conditions and an evaluation could be carried out live in the vehicle, since an interface between Matlab, the vehicle bus and the reference system was used.

### A. Comparison of Simulation Time

First, the simulation time of the different Bayesian filter implementations and the state-of-the-art dead-reckoning models are examined. Matlab's so-called tic-toc method is used to compare the simulation time. A stopwatch timer is started by the tic command and stopped by the toc command. This function records the internal time when the tic-toc command is executed. Fig. 11 shows the simulation time $toc$.

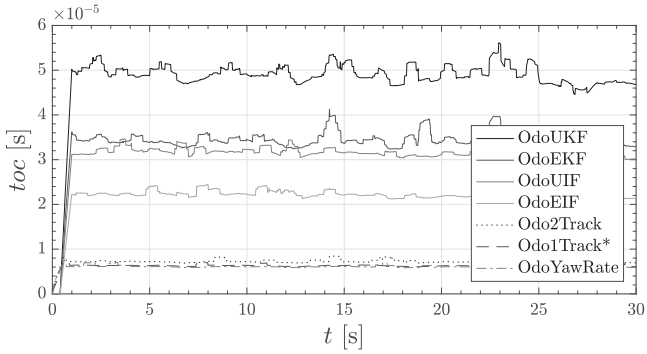


Fig. 11. Comparison of the simulation time of different implementations of the localization filter and the state-of-the-art dead-reckoning models.

The different implementations of Bayesian filter with an average of $3.3 \cdot 10^{-5}$ s need about $6\times$ more computing time, which is easy to understand, considering that they actually include all three models and also include information from the past through the covariance matrix calculations. Both implementations of the Information filter are faster than those of the Kalman filter, this corresponds to the result of the publication [12], which describes that the computing effort for a time-variant system decreases already at a ratio of $3n_z > 4n_x$. The OdoEIF is with an average of $2.2 \cdot 10^{-5}$ s the fastest implementation of localization filters. The OdoYawRate is the fastest model, which is also easy to explain, as it does not require calculation for rotation movement as the measured yaw rate is directly used.

### B. Comparison of Bayesian Filter Types

For comparison, the different implementations of Bayesian filters (EKF, UKF, EIF, UIF) were developed. The examination of the states and thus also the position of the different implementations showed no or only insignificant difference. This result can be explained by the type of linearization, the Extended filters uses the first Tayler series and the Unscented filters uses a linearization up to the third Tayler series, since it is only a small non-linear problem the approximation by the derivation, i.e. the first Taylor series linearization is sufficient. The robustness against non-linear effects is also supported by the tuning of the coefficient matrices from Section V. The Kalman filter and the Information filter differ only in the places where the innovation phase is calculated, which is only expressed by numerical rounding errors that are not recognizable in this case. Since the results of the performance evaluation for the different implementations do not differ, only the simulation time shows how the different filters behave. Therefore only the implementation of the OdoEIF is used for the Odometry 2.0 estimator in the following.

### C. Performance Evaluation

*1) Performance Evaluation Criteria:* In order to evaluate the performance of the Odometry 2.0 estimator compared to the state-of-the-art dead-reckoning models, the following errors $\xi \in \{e_{pos,x}, e_{pos,y}, e_{alig}, e_{loc}\}$ are considered, at the end of each measurement:

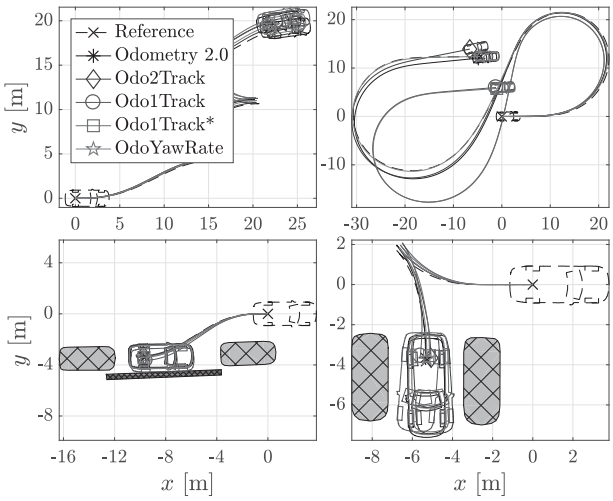


Fig. 12. Trajectories of different odometry maneuvers.

- Position error in global $x$/$y$-coordinate relative to the reference position $e_{pos,x}$, $e_{pos,y}$:

$$\begin{bmatrix} e_{pos,x} \\ e_{pos,y} \end{bmatrix} := e_{pos} := \boldsymbol{T}(\theta^{(R)}) \cdot \left( \boldsymbol{p}^{(R)} - \boldsymbol{p} \right),$$

$$\boldsymbol{T}(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}, \tag{42}$$

- Heading error towards the reference alignment $e_{alig}$:

$$e_{alig} := \theta^{(R)} - \theta, \tag{43}$$

- Cumulative sum of the spread error with respect to the traveled distance $e_{loc}$:

$$e_{loc} := \frac{\sum_{i=0}^{k} \left\| \boldsymbol{p}_i^{(R)} - \boldsymbol{p}_i \right\|}{\sum_{i=1}^{k} \left\| \boldsymbol{p}_i^{(R)} - \boldsymbol{p}_{i-1}^{(R)} \right\|}. \tag{44}$$

In order to obtain a comparable quality measure of a localization error, the distance-dependent cumulative sum of the deviations is defined $e_{loc}$. This value contains the historical data, even if a model has determined a good absolute end position. This can also happen if the deviations in the trajectory at the end of the trajectory have compensated each other.

*2) Performance Evaluation Results:* The results of the Odometry 2.0 estimator and the state-of-the-art dead-reckoning models are compared to the accurate absolute measured reference trajectory in order to investigate the filter performance. The performance is tested with typical odometry test scenarios (Z-, 8-maneuvers) and with parking maneuvers (PS+3m, PPS+2m-maneuvers). Fig. 12 shows the trajectories and in Table III, an overview over the errors is given. Here the Odometry 2.0 estimator performs similarly and in no case worse than the state-of-the-art dead-reckoning models. Additionally, the performance increase of the Odo1Track* compared to Odo1Track is shown. The Odometry 2.0 estimator consistently shows good performance and it becomes clear that there is a slightly better improvement in the localization error when parking. However, the strong performance increase of the Odometry 2.0 estimator only becomes clear in the next section, where the robustness increase is investigated by the sensitivity evaluation.

TABLE III
ERRORS $\xi \in \{e_{pos,x}, e_{pos,y}, e_{alig}, e_{loc}\}$ FOR DIFFERENT ODOMETRY MANEUVERS. Z: Z-MANEUVER; 8: 8-MANEUVER; PS+3M: PARALLEL PARKING SLOT, SIZE: VEHICLE LENGTH +3M; PPS+2M: PERPENDICULAR PARKING S., SIZE: VEHICLE WIDTH +2M

| Model | $\xi$ | Z | | 8 | | PS+3m | | PPS+2m | |
|---|---|---|---|---|---|---|---|---|---|
| Odometry 2.0 | $e_{pos,x}$ | 0,54 | m | 1,15 | m | 0,05 | m | 0,14 | m |
| | $e_{pos,y}$ | -0,47 | m | -1,06 | m | 0,12 | m | -0,18 | m |
| | $e_{alig}$ | -0,57 | ° | -5,74 | ° | -0,14 | ° | 1,23 | ° |
| | $e_{loc}$ | 10,21 | - | 8,35 | - | 11,30 | - | 27,07 | - |
| Odo2Track | $e_{pos,x}$ | 0,69 | m | 2,26 | m | 0,14 | m | -0,25 | m |
| | $e_{pos,y}$ | -0,15 | m | -1,52 | m | -0,07 | m | -0,16 | m |
| | $e_{alig}$ | 2,01 | ° | 10,93 | ° | -0,19 | ° | 2,88 | ° |
| | $e_{loc}$ | 8,18 | - | 15,90 | - | 8,39 | - | 26,63 | - |
| Odo1Track | $e_{pos,x}$ | 0,07 | m | -3,17 | m | -0,14 | m | 0,28 | m |
| | $e_{pos,y}$ | -1,02 | m | 6,42 | m | 0,29 | m | -0,27 | m |
| | $e_{alig}$ | -1,06 | ° | 4,05 | ° | 0,43 | ° | -5,03 | ° |
| | $e_{loc}$ | 15,96 | - | 58,83 | - | 26,23 | - | 42,29 | - |
| Odo1Track* | $e_{pos,x}$ | -0,07 | m | 0,56 | m | 0,01 | m | -0,01 | m |
| | $e_{pos,y}$ | -0,42 | m | -0,22 | m | 0,22 | m | -0,30 | m |
| | $e_{alig}$ | -1,51 | ° | 4,81 | ° | 0,54 | ° | -1,17 | ° |
| | $e_{loc}$ | 7,32 | - | 9,97 | - | 21,45 | - | 38,40 | - |
| OdoYawRate | $e_{pos,x}$ | 0,51 | m | 0,11 | m | 0,09 | m | -0,4 | m |
| | $e_{pos,y}$ | -0,48 | m | 0,01 | m | 0,04 | m | -0,13 | m |
| | $e_{alig}$ | 0,99 | ° | 0,60 | ° | 0,53 | ° | -0,03 | ° |
| | $e_{loc}$ | 7,95 | - | 2,82 | - | 6,90 | - | 19,16 | - |

### D. Sensitivity Evaluation

*1) Sensitivity Criterion (SC):* The sensitivity of the Odometry 2.0 estimator and the state-of-the-art dead-reckoning models are examined for different grades. The sensitivities are considered for the different specified maximum errors from Section III in a Single-Worst-Case-Sensitivity Evaluation. In order to determine the sensitivity of an error factor $\rho$, the test scenarios from Table III are simulated once with the nominal value of $\rho$, once with an error $\rho_A$ and once with an error $\rho_B$. The deviation for the nominal simulation is defined with $\xi|_{\rho}$ and for the simulations with the modified inputs for the worst case scenario, $\xi|_{\rho_A}$ or $\xi|_{\rho_B}$ is defined. The sensitivity in the errors $\xi \in \{e_{pos,x}, e_{pos,y}, e_{alig}, e_{loc}\}$ to the error factor $\rho \in \{\Delta c_i, \Delta c_{RR}, \Delta t_R, \Delta \delta_F, \Delta\omega\}$ could be calculated with the following equation:

$$SC(\xi, \rho) := \frac{1}{2} \cdot \left( \left\| \frac{\xi|_{\rho} - \xi|_{\rho_A}}{\rho_A} \right\| + \left\| \frac{\xi|_{\rho} - \xi|_{\rho_B}}{\rho_B} \right\| \right). \tag{45}$$

$SC(\xi, \rho)$ quantifies how rationally the deviation $\xi$ reacts to a change in the magnitude $\rho$. The smaller the value, the more robust the model is against this error. If the value is -, the deviation is not affected by a change in $\rho$. The Odo1Track* does not use the yaw rate $\omega^{\times}$. Therefore, it does not have a sensitivity to the error in the yaw rate $SC(:, \Delta\omega) = 0$. The OdoYawRate has no sensitivity to the virtual wheel angle $SC(:, \Delta\delta_F) = 0$ and the Odo2Track has no sensitivity to either.

*2) Sensitivity Evaluation Results:* Table IV shows the results of the Sensitivity Evaluation. The sensitivity of all scenarios have been averaged for the averaged Sensitivity Criterion $\overline{SC}(\xi, \rho)$. The deviations $e_{pos,x}$, $e_{pos,y}$ and $e_{loc}$ of the grades estimated in the Odometry 2.0 estimator are significantly less sensitive to a change in the wheel circumference $\Delta c_i$, $\Delta c_{RR}$, the rear track width $\Delta t_R$, the virtual front wheel angle $\Delta\delta_F$ and the yaw rate $\Delta\omega$ than all the comparative state-of-the-art dead-reckoning models presented.

TABLE IV
AVERAGED SENSITIVITY CRITERION $\overline{SC}(\xi,\rho)$ IN THE ERRORS $\xi \in \{e_{pos,x}, e_{pos,y}, e_{alig}, e_{loc}\}$ TO THE ERROR FACTOR $\rho \in \{\Delta c_i, \Delta c_{RR}, \Delta t_R, \Delta\delta_F, \Delta\omega\}$ AVERAGED OVER ALL SCENARIOS

| Model | $\rho$ | $\overline{SC}(e_{pos,x},\rho)$ | | $\overline{SC}(e_{pos,y},\rho)$ | | $\overline{SC}(e_{alig},\rho)$ | | $\overline{SC}(e_{loc},\rho)$ | |
|---|---|---|---|---|---|---|---|---|---|
| Odometry 2.0 | $\Delta c_i$ | 0,64 | m/m | 1,40 | m/m | 1,44 | °/m | 15,70 | 1/m |
| | $\Delta c_{RR}$ | 1,40 | m/m | 0,39 | m/m | 1,00 | °/m | 13,09 | 1/m |
| | $\Delta t_R$ | 0,14 | m/m | 0,29 | m/m | 0,12 | °/m | 1,06 | 1/m |
| | $\Delta\delta_F$ | 0,07 | m/° | 0,04 | m/° | 0,52 | °/° | 1,58 | 1/° |
| | $\Delta\omega$ | 0,09 | m/(°/s) | 0,02 | m/(°/s) | 0,98 | °/(°/s) | 1,71 | 1/(°/s) |
| Odo2Track | $\Delta c_i$ | 0,91 | m/m | 1,95 | m/m | 1,35 | °/m | 16,13 | 1/m |
| | $\Delta c_{RR}$ | 6,67 | m/m | 6,74 | m/m | 7,11 | °/m | 170,45 | 1/m |
| | $\Delta t_R$ | 1,10 | m/m | 2,41 | m/m | 1,88 | °/m | 18,03 | 1/m |
| | $\Delta\delta_F$ | - | m/° | - | m/° | - | °/° | - | 1/° |
| | $\Delta\omega$ | - | m/(°/s) | - | m/(°/s) | - | °/(°/s) | - | 1/(°/s) |
| Odo1Track* | $\Delta c_i$ | 0,91 | m/m | 1,99 | m/m | 1,49 | °/m | 17,43 | 1/m |
| | $\Delta c_{RR}$ | 0,45 | m/m | 1,01 | m/m | 1,03 | °/m | 9,15 | 1/m |
| | $\Delta t_R$ | - | m/m | - | m/m | - | °/m | - | 1/m |
| | $\Delta\delta_F$ | 0,16 | m/° | 0,09 | m/° | 1,34 | °/° | 3,42 | 1/° |
| | $\Delta\omega$ | - | m/(°/s) | - | m/(°/s) | - | °/(°/s) | - | 1/(°/s) |
| OdoYawRate | $\Delta c_i$ | 0,36 | m/m | 0,39 | m/m | 0 | °/m | 11,10 | 1/m |
| | $\Delta c_{RR}$ | 0,18 | m/m | 0,20 | m/m | 0 | °/m | 3,33 | 1/m |
| | $\Delta t_R$ | - | m/m | - | m/m | - | °/m | - | 1/m |
| | $\Delta\delta_F$ | - | m/° | - | m/° | - | °/° | - | 1/° |
| | $\Delta\omega$ | 0,25 | m/(°/s) | 0,09 | m/(°/s) | 2,77 | °/(°/s) | 8,22 | 1/(°/s) |

The model errors $e_{pos,x}$, $e_{pos,y}$ and $e_{loc}$ of the localization filter are much more insensitive to sensor errors of the error factor $\rho \in \{\Delta c_i, \Delta c_{RR}, \Delta t_R, \Delta\delta_F, \Delta\omega\}$.

If the wheel circumferences $c_i$ have an incorrect value or are incorrectly calibrated, this has the strongest effect on the localization error $e_{loc}$ of the Odo2Track and OdoYawRate. The Odo1Track* is least affected, the Odometry 2.0 is in between. As soon as an error occurs on the wheel circumference rear right $c_{RR}$, the accuracy of the Odo2Track increases significantly and the model errors are 6.5-54 $\times$ more sensitive than those of the Odometry 2.0. If the error in the wheel circumference is the maximum value $\Delta c_{RR} = 4$ cm, the deviation of the Odo2Track in the localization error is $\Delta c_{RR} \cdot SC(e_{loc}, \Delta c_{RR}) = 1.44$ m, compared to 0.06 m for the Odometry 2.0. The Odo2Track has also the biggest influence on an orientation with an orientation error of $e_{alig} = 5,95$ °/m in comparison with $e_{alig} = 0,11$ °/m of the Odometry 2.0. This can be explained by the fact that the Odometry 2.0 determines the rotation rate by the 6 wheel pairs, the kinematic side-slip angle and the yaw rate.

The sensitivity to the rear track width $SC(:, \Delta t_R)$ only affects the Odo2Track and the Odometry 2.0. However, it can be noted that the influence on a track width error is small. The localization error alone is decisive at 4.40 1/m.

The sensitivity to a wheel angle sensor error $SC(:, \Delta\delta_F)$ only affects the Odo1Track* and the Odometry 2.0, since $\delta_F^{\times}$ is used only by this model. The deviation is very small. An error of $\Delta\delta_F = 1^\circ$ only affects the alignment by 0,0054° of the Odo1Track*. This can be explained by the fact that it is a parallel parking process and the vehicle makes a right and a left turn. The alignment errors are balanced again.

The sensitivity to errors in the yaw rate $SC(:, \Delta\omega)$ can only be compared with the OdoYawRate and the Odometry 2.0. If the error is $\Delta\omega = 0.7$°/s, the Odometry 2.0 alignment error changes by 0.7°, while the OdoYawRate alignment error changes by 2°. The localization error of the OdoYawRate is also very high at 8.22°/(°/s).

The sensitivities basically depend on the measurement covariance matrix. The Odometry 2.0 reacts more strongly to the uncertainties of the yaw rate, which leads to a lower sensitivity to other sensor errors.

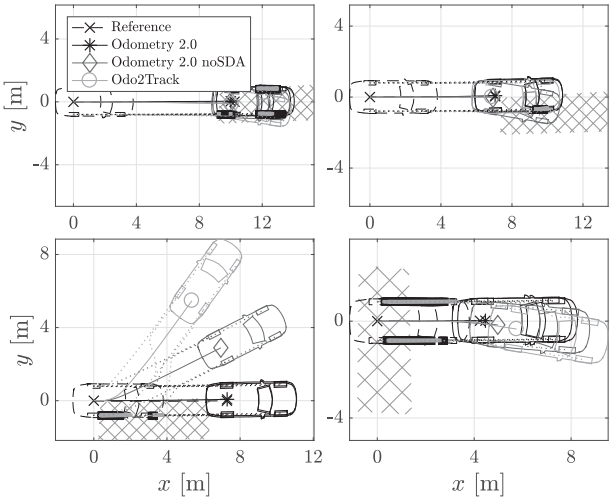


Fig. 13. SDA test maneuver. Top left: slow braking, all wheels in the snow. Top right: slow braking, right wheels in the snow. Bottom left: slow launching, right wheels in the snow. Bottom right: fast launching, rear wheels in the snow.

### E. Slip Detection Algorithm Evaluation

To test the SDA (Section III-D) performing in the Odometry 2.0 estimator, tests with and without SDA are compared, where the setting of the OdoExt2Track (Section V-C) was selected to see the influence of the slip more clearly. Additionally, the Odo2Track is compared.

*1) SDA Evaluation Criteria:* The detected slip (grey) and the reference slip (black) is drawn to the position of the wheel contact point, when the slip determined by the reference $\boldsymbol{S}^{(R)}$ is $true$:

$$\boldsymbol{S}^{(R)} = \begin{cases} \text{true} & \text{if } \left\| \frac{\tilde{v}_i^{\times} - \tilde{v}_i^{(R)}}{\tilde{v}_i^{(R)}} \right\| > 12\% \\ \text{false} & \text{else} \end{cases},$$

$$\tilde{v}_i^{(R)} = v^{(R)} \cdot \cos(\delta_i) + \omega^{(R)} \cdot (r_{i,x} \cdot \sin(\delta_i) - r_{i,y} \cdot \cos(\delta_i)) . \quad (46)$$

It determines the difference between the wheel velocity of the sensor and the velocity of the reference measurement system projected onto the wheel contact point. The threshold for the sensitivity of the reference slip detection is determined experimentally and is set to 12 % and therefore a bit more sensitive than the SDA threshold (Algorithm 1 line 3). In the figures, snow is depicted as light grey hatched. It was measured with an extra DGNSS reference antenna.

*2) Braking in Snow Results:* First the Odometry 2.0 estimator is tested on partially covered snow surfaces while braking. Fig. 13 (top left and right) shows that the SDA performs very well in both situations. It is also interesting that the Odo2Track performs better on the manoeuvre, in which only the front wheels slip, because it does not use these wheel velocities.

*3) Launch in the Snow Results:* The Odometry 2.0 estimator with SDA is then tested on partially covered snow surfaces when launching with a rear-wheel drive vehicle. Fig. 13 (bottom left and right) shows the SDA performance results for test drives

where the vehicle is launched on partially snow-covered surfaces on the rear axle and on the right side of the vehicle. It becomes clear that especially the alignment accuracy can be increased.

## VII. Conclusion

This paper proposes a slip-adaptive EIF-based Four-Wheel-Odometry model for parking, also called Odometry 2.0 estimator. To intelligently take the information on all four wheels, the corrected kinematic side-slip angles and the yaw rate into account, an adaptive fusion architecture was designed. The input of the estimator can be adapted upon detection of quantization errors or even slip on different wheels. The Odometry 2.0 approach leads to similar or slightly better final position performance as the state-of-the-art dead-reckoning models for parking. The type of implementation (EKF, UKF, EIF, UIF) makes no difference due to the low non-linearity. However, the EIF did well in examining the simulation time. Through sensitivity evaluation, it is shown that the Odometry 2.0 approach is much more robust against systematic errors in the measurement than the comparative models. When wheel slip occurs, the presented SDA ensures that the Odometry 2.0 estimator switches off the faulty measurement, resulting in improved pose performance in the event of wheel slip. As part of future work, additional driving situation-dependent tuning for the coefficient matrix could be designed, making the Odometry 2.0 estimator more effective when model specific errors arise in detected driving conditions.

## References


[1] D.-J. Lee, "Nonlinear estimation and multiple sensor fusion using unscented information filtering," *IEEE Signal Process. Lett.*, vol. 15, pp. 861–864, 2008.

[2] A. Brunker, M. Frey, F. Gauterin, and T. Wohlgemuth, "GNSS-shortages-resistant and self-adaptive rear axle kinematic parameter estimator (SARAKPE)," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2017, pp. 456–461.

[3] T. D. Larsen, K. L. Hansen, N. A. Andersen, and O. Ravn, "Design of Kalman filters for mobile robots; evaluation of the kinematic and odometric approach," in *Proc. IEEE Int. Conf. Control Appl.*, vol. 2, Aug. 1999, pp. 1021–1026.

[4] Y. K. Tham, H. Wang, and E. K. Teoh, "Adaptive state estimation For 4-wheel steerable industrial vehicles," in *Proc. IEEE Conf. Decis. Control*, Dec. 1998, pp. 4509–4514.

[5] L. Armesto, J. Tornero, and M. Vincze, "Fast ego-motion estimation with multi-rate fusion of inertial and vision," *Int. J. Robot. Res.*, vol. 26, pp. 577–589, 2007.

[6] M. Ibraheem, "Gyroscope-enhanced dead reckoning localization system for an intelligent walker," in *Proc. Int. Conf. Inf. Netw. Automat.*, Oct. 2010, pp. V1-67–V1-72.

[7] D. Caltabiano, G. Muscato, and F. Russo, "Localization and self-calibration of a robot for volcano exploration," in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 1, Jun. 2004, pp. 586–591.

[8] B. Gao, W. Tao, H. Chu, M. Tian, and H. Chen, "A reference vehicle model applied to electronic stability control (ESC) system," in *Proc. Chin. Control Conf.*, Jul. 2017, pp. 9436–9441.

[9] P. Simionescu and D. Beale, "Optimum synthesis of the four-bar function generator in its symmetric embodiment," *Mech. Mach. Theory*, vol. 37, pp. 1487–1504, 2002.

[10] H. Chung, L. Ojeda, and J. Borenstein, "Accurate mobile robot dead-reckoning with a precision-calibrated fiber-optic gyroscope," *IEEE Trans. Robot. Automat.*, vol. 17, no. 1, pp. 80–84, Feb. 2001.

[11] K. Lee and W. Chung, "Calibration of kinematic parameters of a car-like mobile robot to improve odometry accuracy," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2008, pp. 2546–2551.

[12] N. Assimakis, M. Adam, and A. Douladiris, "Information filter and Kalman filter comparison: Selection of the faster filter," in *Proc. Int. J. Inf. Eng.*, 2012, pp. 1–5.

[13] T. Müller, J. Rannacher, C. Rabe, and U. Franke, "Feature- and depth-supported modified total variation optical flow for 3D motion field estimation in real scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 1193–1200.

[14] H. Lategahn and C. Stiller, "Vision-only localization," in *Proc. IEEE Trans. Intell. Trans. Syst.*, vol. 15, 2014, pp. 1246–1257.

[15] A. Brunker, "Method for determining the proper motion of a vehicle, especially a motor vehicle," DMPA, Patent DE102 017 002 637A1, Oct. 19, 2017.

[16] A. Brunker and F. Stein, "A process for the detection of driving direction of a vehicle," Daimler [DE], Patent DE102 017 003 442A1, Jan. 04, 2018.

[17] A. Brunker, F. Stein, M. Greiner, V. Rodrigo-Marco, and J. Kalkkuhl, "A process for the detection of driving direction of a vehicle," Daimler [DE], Patent DE102 017 007 122A1, Mar. 01, 2018.

[18] A. Brunker and F. Stein, "A method for odometry-based position determination for a vehicle," Daimler [DE], Patent DE102 017 007 773A1, Apr. 19, 2018.

[19] A. Brunker, "A method for correcting a wheel angle of steered wheels of a vehicle," Daimler [DE], Patent DE102 017 006 585A1, Mar. 01, 2018.

[20] A. Brunker and F. Stein, "A method for correcting a wheel angle of steered wheels of a vehicle," Daimler [DE], Patent DE102 017 006 595A1, Mar. 01, 2018.

[21] A. G. Quinchia, G. Falco, E. Falletti, F. Dovis, and C. Ferrer, "A comparison between different error modeling of MEMS applied to GPS/INS integrated systems," *Sensors*, vol. 13, pp. 9549–9588, 2013.


**Alexander Brunker** received the M.Sc. degree in automotive engineering from the Rheinisch-Westfälische Technische Hochschule, Aachen, Germany, in July 2015. He is a Research Associate with the Institute of Vehicle System Technology, Karlsruhe Institute of Technology, Karlsruhe, Germany. Since 2015, he has been with the Research and Development, Daimler AG, Sindelfingen, Germany. His research interests include odometry, localization, state estimation, and vehicle modelling.

**Thomas Wohlgemuth** received the Diploma in electronics/micro-electronics from the University of Applied Sciences, Esslingen, Germany, in 1994. In 1994, he worked for the Steinbeis Foundation as an electronics engineer. From 1995 to 2004, he worked for Daimler RD as an electronics, engineer for transmission controllers and night vision systems. From 2004 to 2013, he was a manager for electronic control units for driving dynamic systems and sensors. Since 2013, he has been a manager for automatic park systems.

**Michael Frey** received the Diploma in mechanical engineering from the University of Karlsruhe, Karlsruhe, Germany, and the Ph.D. degree in mechanical engineering from the University of Karlsruhe, in 1993 and 2004, respectively. He is the manager of the research group Operational Strategies and of the research group Suspension Systems at the Institute of Vehicle Systems Technology. His research interests include driver assistance systems, operational strategies, suspension systems, vehicle dynamics, as well as vehicle modeling and optimization.

**Frank Gauterin** received the Diploma from the University of Münster, Münster, Germany, and the Dr. rer. nat. degree (Ph.D.) from the University of Oldenburg, Oldenburg, Germany, in 1989 and 1994, respectively, all in physics. From 1989 to 2006, he was in various RD positions at Continental AG, Germany, leaving as director of NVH Engineering (noise, vibration, and harshness). Since 2006, he has been a full professor at the Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany. He is head of the KIT Institute of Vehicle System Technology and scientific spokesperson of the KIT Center Mobility Systems. His research interests include vehicle control, vehicle dynamics, vehicle NVH, vehicle suspension, tire dynamics and tire-road-interaction as well as vehicle concepts, vehicle modeling, and identification methods.