

Velocity Estimation via Wheel Circumference Identification

Máté Fazekas^{1*}, Péter Gáspár¹, Balázs Németh¹

¹ Systems and Control Laboratory, Institute for Computer Science and Control (SZTAKI), H-1111 Budapest, Kende street 13-17, Hungary

* Corresponding author, e-mail: mate.fazekas@sztaki.hu

Received: 21 May 2021, Accepted: 15 June 2021, Published online: 09 August 2021

Abstract

The article presents a velocity estimation algorithm through the wheel encoder-based odometry and wheel circumference identification. The motivation of the paper is that a proper model can improve the motion estimation in poor sensor performance cases. For example, when the GNSS signals are unavailable, or when the vision-based methods are incorrect due to the insufficient number of features, furthermore, when the IMU-based method fails due to the lack of frequent accelerations. In these situations, the wheel encoders can be an appropriate choice for state estimation. However, this type of estimation suffers from parameter uncertainty. In the paper, a wheel circumference identification is proposed to improve the velocity estimation. The algorithm listens to the incoming sensor measurements and estimates the wheel circumferences recursively with a nonlinear least squares method. The experimental results demonstrate that with the application of the identified parameters in the wheel odometry model, accurate velocity estimation can be obtained with high frequency. Thus, the presented algorithm can improve the motion estimation in the driver assistant functions of autonomous vehicles.

Keywords

velocity estimation, odometry, identification

1 Introduction

In autonomous vehicle functions, vehicle localization and motion estimation has become a key question nowadays. The results of these estimations are utilized in the trajectory planning layer, such as the design of overtaking maneuvers (Németh et al., 2019), and in the vehicle control layer, e.g. suspension control (Basargan et al., 2020; Basargan et al., 2021). Therefore, the as accurate as possible estimation of the motion signals, such as velocity is an important requirement in series-produced vehicles.

The motion estimation of vehicles can be performed by several methods using a wide range of sensors, such as camera, LiDAR, GNSS (Global Navigation Satellite System), IMU (inertial measurement unit), and wheel encoders. The perception-based methods (Bloesch et al., 2015) demand prior teaching, and well recognizable features are required (Fazekas et al., 2020a). The fusion of GNSS and IMU measurements could be precise in higher velocity scenarios, but the method requires the actual knowledge of the covariances of signals (Caron et al., 2006). Moreover, the IMU-based method fails, when there is a lack of strong and frequent accelerations (Funk et al., 2017; Thrun et al, 2006). Furthermore, the GNSS signal can be unavailable in some cases e.g. in parking garages (Schanz et al., 2003),

and moreover, the accuracy can be weak in several urban areas, e.g. between high-rise buildings in urban areas. In these situations, wheel-encoder based odometry may be the appropriate choice for vehicle localization (Thrun et al., 2006), and also it can be integrated into the state estimation layer to improve the performance. Furthermore, this method is the most cost-effective.

In mobile robot applications, encoder-based odometry has been a widely used method for a long time (Moutarlier and Chatila, 1990). In the automotive industry, this type of motion estimation appears with the parking assist functions (Kochem et al., 2002). Several papers deal with the odometry for car-like vehicles but in the case of small robots. In (Lemmer et al., 2010), the basic velocity calculation through wheel motion is examined and calibrated, (Bohlmann et al., 2012) deals with the calibration of a four-wheel-steered small RC car. The effect of kinematic parameter calibration is presented and tested on a small car in (Jung et al., 2016). In the case of real-sized vehicles, only a few studies exist. A comparison of the rear and front axle models for parking is presented in (Kochem et al., 2002). The possible vehicle models are illustrated in (Brunker et al., 2018), with the scope of wheel slippage and parking. Other

works (Brossard and Bonnabel, 2019; Toledo et al., 2018), do not apply vehicle models directly but learn them from data.

In this paper, the velocity estimation with the application of vehicle odometry models, utilizing wheel-encoder measurements is examined. The algorithm does not focus on any special case, such as parking, but the general moving scenario of vehicles is illustrated. Since the accuracy of the velocity estimation highly depends on the proper wheel parameters, a recursive wheel circumference identification is proposed.

The remainder of the paper is organized as follows. The detailed equations of the front- and rear-odometry models can be found in Section 2. The identification of the nonlinear model parameters with a least squares based method is presented in Section 3. The paper deals with signals of a real series-produced vehicle, the applied measurement environment is explained in Section 4. The results of the wheel circumference identification and the velocity estimation are illustrated with experimental tests in Section 5, and finally, the paper is concluded in the last Section 6.

2 Vehicle models for the velocity estimation and wheel circumference identification

Although the main scope in this paper is the velocity estimation, for the parameter estimation the wheel encoder and the pose measurements should be connected. This can be done with the wheel odometry model in which the position $p_{x,k}$ and $p_{y,k}$, and the orientation ψ_k are calculated, such as in Eq. (1)

$$\begin{bmatrix} p_{x,k} \\ p_{y,k} \\ \psi_k \end{bmatrix} = \begin{bmatrix} p_{x,k-1} + v_{k-1} \cdot \cos(\psi_{k-1} + \omega_{k-1} / 2) \\ p_{y,k-1} + v_{k-1} \cdot \sin(\psi_{k-1} + \omega_{k-1} / 2) \\ \psi_{k-1} + \omega_{k-1} \end{bmatrix}, \quad (1)$$

where v_k and ω_k are the longitudinal and angular velocities of the vehicle.

The calculation of the velocities is based on the applied vehicle model. The well-known models are the two-wheel and kinematic bicycle models (Fig. 1), but in this paper, all wheels are integrated. The wheel velocities can be computed from those of the vehicle, such as in Eq. (2)

$$\begin{bmatrix} v_{i,x} \\ v_{i,y} \\ 0 \end{bmatrix} = \begin{bmatrix} v \cdot \cos(\beta) \\ v \cdot \sin(\beta) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \omega \end{bmatrix} \cdot \begin{bmatrix} r_{i,x} \\ r_{i,y} \\ 0 \end{bmatrix}, \quad (2)$$

where the $v_{i,x}$ and $v_{i,y}$ ($i=\{FL, FR, RL, RR\}$, F: front, R: rear, L:R) are the lateral and longitudinal components of the wheel velocities with respect to the vehicle frame, and the r_i values are the distances between the reference point and

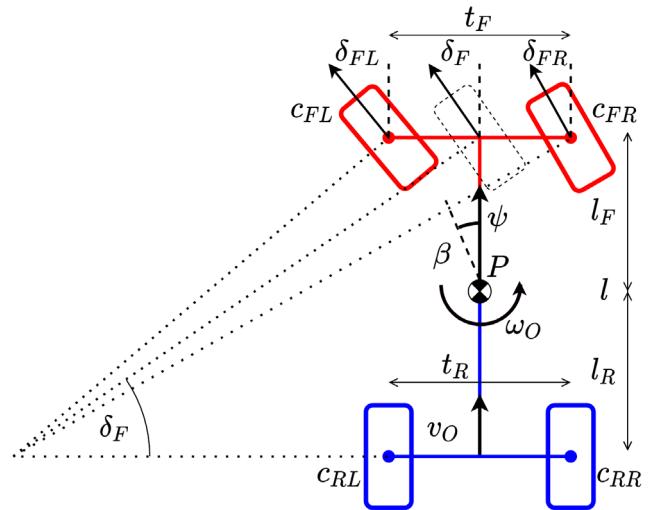


Fig. 1 Front- and rear odometry models

the center of the wheel. For us, the inverse calculation is required, but with the wheel encoder only the velocity component in the direction of rotation is measured (this is the longitudinal component in the wheel frame). However, this can be expressed with the coordinate components of the vehicle frame, such as in Eq. (3)

$$\tilde{v}_i = v_{i,x} \cdot \cos(\delta_i) + v_{i,y} \cdot \sin(\delta_i), \quad (3)$$

where \tilde{v}_i denotes the measured wheel velocities, and δ_i is the wheel angle. Based on these, the vehicle velocities can be expressed utilizing two wheel velocity measurements in the following way:

$$\begin{aligned} v_{num} &= \tilde{v}_{i2} \cdot r_{i1,x} \cdot \sin(\delta_{i1}) - \tilde{v}_{i1} \cdot r_{i2,x} \cdot \sin(\delta_{i2}) \\ &\quad - \tilde{v}_{i2} \cdot r_{i1,y} \cdot \cos(\delta_{i1}) + \tilde{v}_{i1} \cdot r_{i2,y} \cdot \cos(\delta_{i2}), \end{aligned} \quad (4)$$

$$v_{num} = \tilde{v}_{i1} \cdot \cos(\delta_{i2} - \beta) - \tilde{v}_{i2} \cdot \cos(\delta_{i1} - \beta), \quad (5)$$

$$\begin{aligned} v_{den} &= \omega_{den} \\ &= \cos(\delta_{i2} - \beta) \cdot r_{i1,x} \cdot \sin(\delta_{i1}) - \cos(\delta_{i1} - \beta) \cdot r_{i2,x} \cdot \sin(\delta_{i2}) \\ &\quad - \cos(\delta_{i2} - \beta) \cdot r_{i1,y} \cdot \cos(\delta_{i1}) + \cos(\delta_{i1} - \beta) \cdot r_{i2,y} \cdot \cos(\delta_{i2}), \end{aligned} \quad (6)$$

$$v = \frac{v_{num}}{v_{den}}, \quad \omega = \frac{\omega_{num}}{\omega_{den}}. \quad (7)$$

The detailed explanation can be found in the appendices of (Fazekas, 2019). The distance values are the following:

$$r_x = [r_{FL,x} \quad r_{FR,x} \quad r_{RL,x} \quad r_{RR,x}] = [l \quad l \quad 0 \quad 0],$$

$$\begin{aligned} r_y &= [r_{FL,y} \quad r_{FR,y} \quad r_{RL,y} \quad r_{RR,y}] \\ &= [t_F / 2 \quad -t_F / 2 \quad t_R / 2 \quad -t_R / 2]. \end{aligned}$$

For example in the case of the well-known two-wheel model, where only the rear-wheel speeds are applied ($i1=RR$, $i2=RL$):

$$v = \frac{\tilde{v}_{RL} + \tilde{v}_{RR}}{2 \cdot \cos(\beta)} \omega = \frac{-\tilde{v}_{RL} + \tilde{v}_{RR}}{t_R}, \quad (8)$$

due to $\delta_{RL} = \delta_{RR} = 0$. Finally, the n_i wheel encoder measurements are inserted, such as:

$$v = \frac{n_{RL} \cdot c_{RL} + n_{RR} \cdot c_{RR}}{2 \cdot \cos(\beta)}, \quad \omega = \frac{-n_{RL} \cdot c_{RL} + n_{RR} \cdot c_{RR}}{t_R}, \quad (9)$$

where c_i is the wheel circumference. These velocities are applied in Eq. (1) to calculate the vehicle pose values, and the following nonlinear system is resulted:

$$x_k = f(x_{k-1}, u_{k-1}, \theta), \quad y_k = x_k, \quad (10)$$

$$x_k = [p_{x,k} \quad p_{y,k} \quad \psi_k]^T. \quad (11)$$

where the inputs and the parameters are:

$$u_{k-1} = [n_{RL,k-1} \quad n_{RR,k-1}]^T, \quad \theta = [c_{RL} \quad c_{RR}]. \quad (12)$$

Utilizing the measurements of the front wheels, the values can be determined in the same way, but the δ_{FL} and δ_{FR} wheel angles have to be included, since the front axle of the vehicles is steered. The values can be calculated based on the Ackermann-steering geometry (Scott et al., 2006), such as in Eq. (13):

$$\begin{aligned} \hat{\delta}_{FL,k} &= \text{atan}\left(\frac{l}{\frac{l}{\tan(\hat{\delta}_F)} - \frac{t_F}{2}}\right), \\ \hat{\delta}_{FR,k} &= \text{atan}\left(\frac{l}{\frac{l}{\tan(\hat{\delta}_F)} + \frac{t_F}{2}}\right), \end{aligned} \quad (13)$$

where $\hat{\delta}_F$ are the virtual front wheel angles which can be computed with the $\tilde{\gamma}$ measured steering wheel angle with the following polynomial function:

$$\hat{\delta}_F = p_3 \cdot \tilde{\gamma}^3 + p_2 \cdot \tilde{\gamma}^2 + p_1 \cdot \tilde{\gamma}^1 + p_0. \quad (14)$$

A possible calibration of the p_i parameters is presented in (Fazekas et al., 2021a). Thus, there are two separate systems, f_R and f_L , and θ_R and θ_L , respectively. However, the whole parameter calibration method has the same structure, thus for simplicity, a general f will be used in the rest of the paper, and the algorithm is illustrated with the equations of the rear odometry, Eqs. (4) to (7).

The velocity can be calculated with Eq. (8), but it contains the wheel circumference values as parameters. Since these values vary with the tire wear and the load of the vehicle, the parameters should be estimated. The presented odometry model can be improved with a dynamic

wheel model (Fazekas et al., 2020b), however in the case of velocity estimation it has no significant effect, thus in this paper the wheel circumference is assumed to be constant. The presented odometry model connects these parameters and the pose measurements, which are generally measured with onboard GNSS and IMU sensors in modern series-produced vehicles.

3 Parameter identification method

3.1 Estimation with nonlinear least squares

The parameter estimation is handled as a fitting in this paper, which can be solved with the least squares (LS) method. The minimization problem is formulated, such as Eq. (15):

$$\hat{\theta}_{opt} = \arg \min_{\theta} S_k = \arg \min_{\theta} \frac{1}{K} \sum_{k=k_0}^{k_0+K-1} (\tilde{y}_k - y_k(\theta))^2, \quad (15)$$

where the predictor $y_k(\theta)$ is the value of the vehicle model at time k , and \tilde{y}_k is the measurement used for the estimation. If the system is linear, $y_k(\theta)$ has the form of $\phi_k \cdot \theta$ and the optimization is a convex problem, which has a unique solution and can be computed in one step (Ljung, 1998). However, the consequence of the nonlinear odometry model is that nonlinear LS method should be applied. These methods are numerical search techniques and approximate the minimization task of Eq. (15). The applied method handles the nonlinearity with the first order Taylor-approximation of the predictor (Tangirala, 2014), such as in Eq. (16):

$$y_k(\theta) \approx + \underbrace{\frac{\partial y_k(\theta)}{\partial \theta} \Big|_{\theta_{i-1}}}_{j_k} \underbrace{(\theta - \theta_{i-1})}_{\Delta \theta}. \quad (16)$$

The approximation results in a locally linear LS problem:

$$\widehat{\Delta \theta}_{opt} = \arg \min_{\theta} \frac{1}{K} \sum_{k=k_0}^{k_0+K-1} ((\tilde{y}_k - y_k(\theta)) - j_k \cdot \Delta \theta)^2, \quad (17)$$

which can be solved with the basic LS solution, with the result as:

$$\widehat{\Delta \theta}_{opt} = (J(\hat{\theta}_{i-1})^T J(\hat{\theta}_{i-1}))^{-1} J(\hat{\theta}_{i-1})^T (\tilde{Y} - Y(\hat{\theta}_{i-1})). \quad (18)$$

Thus, the new value of the parameter vector is:

$$\hat{\theta}_i = \hat{\theta}_{i-1} + (J(\hat{\theta}_{i-1})^T W J(\hat{\theta}_{i-1}))^{-1} J(\hat{\theta}_{i-1})^T W (\tilde{Y} - Y(\hat{\theta}_{i-1})), \quad (19)$$

where W is a weight matrix that takes into account the different range of the position measurements in *meter* and the orientation measurements in *radian*. The $J(\hat{\theta}_{i-1})$ matrix is formed from the $j_k(\hat{\theta}_{i-1})$ values, thus this is the Jacobian of the predictor $y_k(\theta)$, such as:

$$j_k(\hat{\theta}_{i-1}) = \begin{bmatrix} \frac{\partial p_{x,k}}{\partial c_{RL}} & \frac{\partial p_{x,k}}{\partial c_{RR}} \\ \frac{\partial p_{y,k}}{\partial c_{RL}} & \frac{\partial p_{y,k}}{\partial c_{RR}} \\ \frac{\partial \psi_k}{\partial c_{RL}} & \frac{\partial \psi_k}{\partial c_{RR}} \end{bmatrix}, J(\hat{\theta}_{i-1}) = \begin{bmatrix} j_{k0}(\hat{\theta}_{i-1}) \\ \vdots \\ j_K(\hat{\theta}_{i-1}) \end{bmatrix}. \quad (20)$$

In the last term, $Y(\hat{\theta}_{i-1})$, and \tilde{Y} matrices are formed from the $y_k(\hat{\theta}_{i-1})$, and \tilde{y}_k values, respectively. The term contains the integrated vehicle model with the previous parameters and the residuals are calculated utilizing the reference measurements:

$$\tilde{y}_k - y_k(\hat{\theta}_{i-1}) = \tilde{y}_k - f(x_{k-1}, u_{k-1}, \hat{\theta}_{i-1}) = \begin{bmatrix} \tilde{p}_{x,k} - p_{x,k} \\ \tilde{p}_{y,k} - p_{y,k} \\ \tilde{\psi}_k - \psi_k \end{bmatrix}, \quad (21)$$

$$\tilde{Y} - Y(\hat{\theta}_{i-1}) = \begin{bmatrix} \tilde{y}_{k0} - y_{k0}(\hat{\theta}_{i-1}) \\ \vdots \\ \tilde{y}_K - y_K(\hat{\theta}_{i-1}) \end{bmatrix}. \quad (22)$$

Since the predictor is a dynamic system, the minimization task has to be initialized at time $k0$ with state values of time k_0 , such as:

$$\tilde{y}_{k0} - y_{k0}(\hat{\theta}_{i-1}) = \tilde{y}_{k0} - f(x_{k0-1}, u_{k0-1}, \hat{\theta}_{i-1}). \quad (23)$$

The possible choice is to apply the reference measurements \tilde{y}_{k0-1} , but its uncertainty can result in unfeasible parameter estimation (Fazekas et al., 2021b). The numerical search with this Taylor-approximation is an iterative technique, thus the parameter vector has to be initialized as well. The nominal values of the circumferences are used, which can be found in the vehicle's datasheet.

The iterative estimation runs until the residual decreases or the maximum iteration is reached, the final estimated values are denoted with $\hat{\theta}_{opt}$.

3.2 The estimation architecture

With the presented estimation method the model is calibrated from K input and output values. The value of K is determined based on two facts. First, it is clear in a LS fitting that by increasing the number of measurement points the effect of noise is decreasing. However, the uncertainty in the state initialization can result in divergence easily in the vehicle path and in parallel in the parameter estimation. Thus, the measurement length is chosen for 30 s, which corresponds to $K=1200$ with a 40 Hz measurement frequency. Furthermore, the 30 s results in around 300 m path length, assuming 10 m/s average speed, which is realistic in city driving.

The other advantage of the shorter subtraces used for the calibration is that further filtering techniques can be applied to stabilize the circumference estimation. A basic recursive averaging is proposed, in which the Θ_s denotes the actual value of the circumferences, and updated every time in the following way when an estimated parameter $\hat{\theta}_{opt}$ of a separate subtrace is available:

$$\hat{\Theta}_s = \frac{s-1}{s} \hat{\Theta}_{s-1} + \frac{1}{s} \hat{\theta}_{opt}, \quad (24)$$

where s is the number of subtraces applied. Thus, the proposed calibration method is a moving window estimator. The shift between the separate subtrace estimations is 2.5 s. The whole calibration architecture can be found in Fig. 2. The algorithm listens to the incoming sensor measurements, and estimates the wheel circumferences with the nonlinear LS technique in every $\Delta t = 2.5$ s based on the last 30 s of driving, and the individual subtrace estimations are filtered recursively. The circumference values $\Theta_s = [c_{RL} \ c_{RR}]$ (and $[c_{FL} \ c_{FR}]$ in the case of front odometry of course) can be used anytime to calculate the velocities of the vehicle. Furthermore, the architecture could handle sensor errors easily, e.g. when the GNSS signals are unavailable or highly uncertain, the recursive filtering is not executed and for the velocity or pose calculation the last Θ_s values are utilized.

3.3 Tuning of the method

The tuning of the method aims to determine the value of the weight matrix W , and the stopping conditions, such

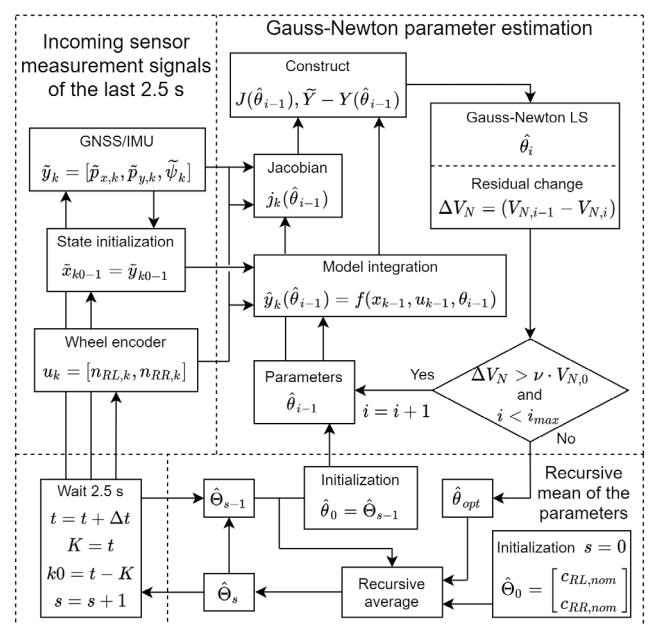


Fig. 2 Architecture of the parameter identification

as the maximum iteration number i_{max} and the residual decreasing rate parameter v .

The last is limited by the requirement of the computation capacity. In our case, $v = 0$ is chosen, which means that the iterative estimation runs until the residual decreases or the maximum iteration is reached. This can be applied, because the computation time is not critical in this application, since 2.5 seconds are available between two calculations. The maximum iteration should be chosen to guarantee the convergence of the iterative LS estimation. Several tests demonstrate that the relative change of the residual after 3 iterations is below 0.1 %, which verifies that the method converges to an optimum.

The determination of the optimal value of the weight matrix is more difficult. The matrix is generally included in the minimization task Eq. (17), to take the different magnitude of the equation noises into account. In the presented case the positions are measured in meter while the orientation in radian, thus the error of the orientation equations are significantly lower. Consequently, without weighting, the numerical optimization would only try to reduce position errors, which means that the circumferences that resulted in the minimal position error would be the optimal values. However, the estimation is based only on subtraces and due to the measurement noises, several local minimums can exist in a nonlinear dynamic model calibration problem (Schoukens and Ljung, 2019).

The aim is to estimate that parameters of the model from measurements of a subtrace, which result in minimum errors on different subtraces, as well. To guarantee this generalization capability the whole model Eq. (1), which defines the connection between the parameters and the measurements applied for its calibration, should be taken into account. Thus, the form of the weight matrix is:

$$W = \text{diag}(w, \dots, w)_{3K \times 3K}, \quad (25)$$

$$w = \begin{bmatrix} w_{p,x}^2 & 0 & 0 \\ 0 & w_{p,y}^2 & 0 \\ 0 & 0 & w_\psi^2 \end{bmatrix}, \quad w_{p,x} = w_{p,y} = 1. \quad (26)$$

Because only the ratio of the weights matters, the only tuning parameter is the w_ψ . The optimal value is difficult to obtain since the weights can not be expressed as the reciprocals of the sensor measurement noises, because of the dynamic model.

Therefore, an experimental tuning is applied. The appropriate order of magnitude of weight is calculated by test with the nominal setting. The odometry model with

$\theta_{nom} = [c_{RL,nom} \ c_{RR,nom}]$ parameters was integrated on every subtrace, and the ratio of the position and orientation errors are calculated such as:

$$n = \frac{\sum_{k=1}^K \sqrt{(\tilde{p}_{x,k} - p_{x,k})^2 + (\tilde{p}_{y,k} - p_{y,k})^2}}{\sum_{k=1}^K |\tilde{\psi}_k - \psi_k|}. \quad (27)$$

The values of the subtraces can be found in Fig. 3. The mean is 99, but the value is varying significantly, and it is the result with the nominal setting, which is applied only in the first iteration of the estimations. However, it can be seen that the ratios are in a relatively narrow range. Therefore, the whole estimation is tested with w_ψ values between 50 and 250, and the optimal setting of the circumferences resulted by $w_\psi = 173$ setting. It is significantly higher than the mean of 99, but if we examine the vehicle model Eq. (1), it is evident that an accurate orientation value is also essential for low position error.

4 Test vehicle and measurements

The test vehicle was a Nissan Leaf electric compact car that is equipped with automotive-grade GNSS and compass. From the vehicle CAN bus the wheel encoder signals were also saved. The sampling frequency was 40 Hz. The vehicle is also equipped with a dual antenna GNSS/IMU sensor, which contains internal filter algorithms and provides absolute velocity signals. This is applied as a reference to validate our results, and the raw GNSS signal is utilized for the calibration.

The test track was a 20 km long route in suburb and city driving with full traffic (Fig. 4). The track contained several sharp and large curved bends, two roundabouts, and lots of crossroads. Because of this various track and traffic characteristics, the velocity was varied, the maximum was 18.72 m/s, the mean was 9 m/s while the standard deviation was 4 m/s.

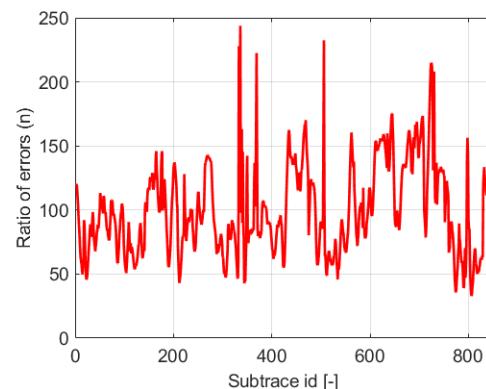


Fig. 3 Ration of errors in the subtraces

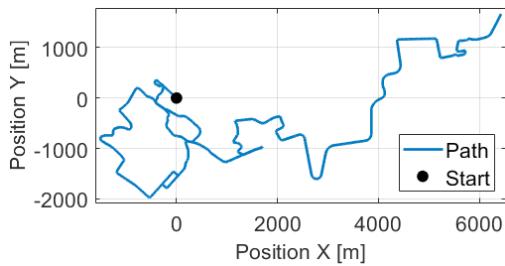


Fig. 4 Path of the vehicle in the experimental test

The odometry model applies kinematic vehicle parameters to calculate the pose change from the wheel rotations. The value of these parameters can be obtained from the vehicle datasheet. Table 1 shows the numerical values of our test vehicle. The nominal value of the circumferences is computed by the geometry size of the tire, thus not containing the deformation resulted from the load and the wear.

5 Results

In this section, the results of the estimation are presented. First, the estimation method is illustrated in detail on a given subtrace, next the estimated circumferences are shown on all subtraces, and finally the accuracy of the method is validated with the demonstration of the velocity estimation with the calibrated model. Due to the linearization in the LS method, initial parameter guess is required. For these, the nominal wheel circumference values are applied.

5.1 The estimation method

The proposed parameter calibration method is illustrated with the estimation of the rear wheel circumferences on 5 separate subtraces. In Figs. 5 and 6, the estimated values of the iterations can be found in 5 different subtraces. The estimated values are stabilized during 3 iterations, and the left and right values are close to each other, as it is expected. The change in the first iteration is huge, which can be explained if we examine the relative residual decrease in Fig. 7. In the first iteration, the cumulated weighted error decreased by 95–99%, while in the next iterations the decrease was only a few percent. This

Table 1 Vehicle parameters of the test car from datasheet

Parameter		Value
Nominal wheel circumference	$c_{i,nom}$	2 m
Front track width	t_F	1.539 m
Rear track width	t_R	1.534 m
Wheelbase	l	2.7 m

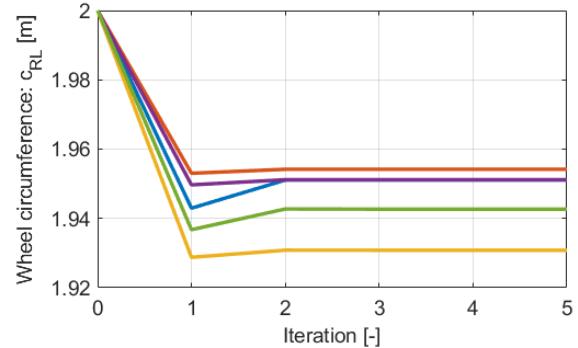


Fig. 5 Estimated wheel circumference in the iterations 1

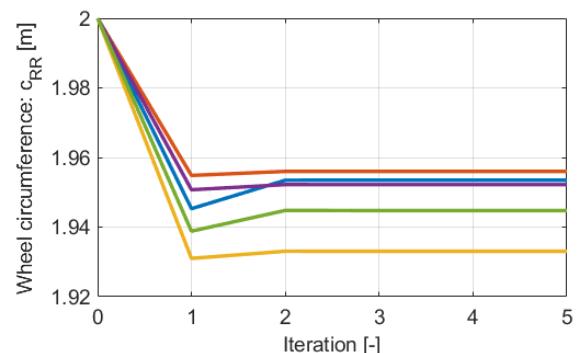


Fig. 6 Estimated wheel circumference in the iterations 2

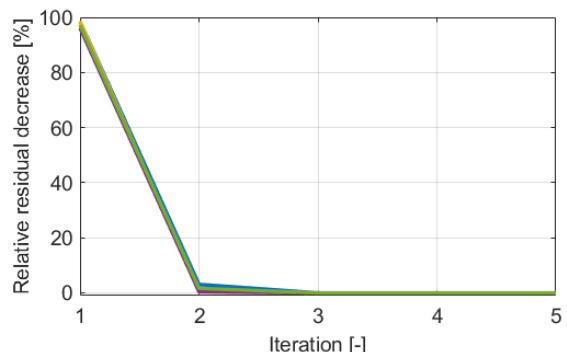


Fig. 7 Relative residual decrease in the iterations

illustrates two facts, firstly the uncalibrated model is not applicable for the estimation, and secondly the nonlinear LS estimation can converge fast. Furthermore, in the last two iterations the residual change is almost zero, thus a minimum is achieved.

5.2 The estimated circumferences

The final values of the estimated circumferences of all subtraces can be found in Figs. 8 and 9. Both the rear and front values are varying significantly, which is the consequence of the noisy reference pose measurements used for calibration. In our previous paper (Fazekas et al., 2021b), the issue is examined in detail and an internal filtering is proposed to mitigate the effect of measurement noise. In

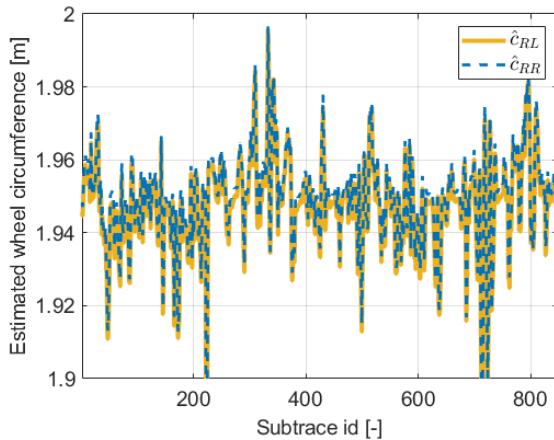


Fig. 8 Estimated wheel circumferences of all segments 1

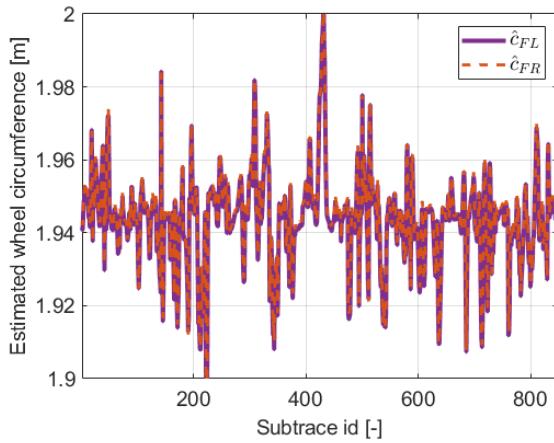


Fig. 9 Estimated wheel circumferences of all segments 2

In most cases, the estimated rear and front values are close to each other, only in some cases, for example around subtrace 410, differ the values from each other significantly.

Because of the high variety of the estimations, a recursive filtering technique Eq. (24) has been proposed. The filtered circumferences can be found in Fig. 10. As we can see after the first 150 estimations the values of the

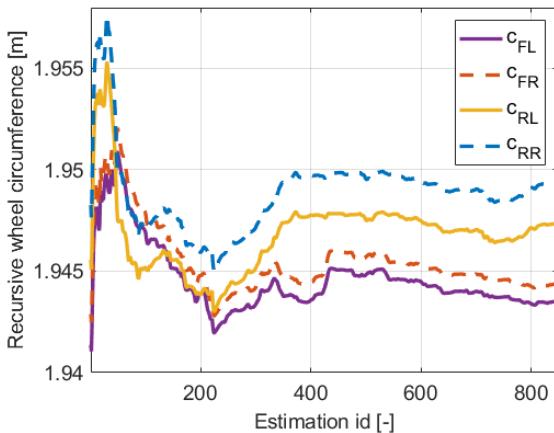


Fig. 10 Estimated circumferences with the recursive filtering

front circumferences are stabilized in a ± 2 mm range, and the rear ones also after 350 estimations. The convergence to the optimum values can be verified also with the examination of the standard deviation of the estimations, which can be found in Fig. 11. The deviations are maximized around 12.5 mm, and it is reached after 250 iterations. Thus, it is clear that after 350 estimations, which is equal to 8 km, the calibrated model can be applied for the motion estimation task.

The final values of the estimated parameters and their uncertainty can be found in Table 2.

5.3 The estimated velocity

The previous figures demonstrate that the estimation method converges to a stable optimum. However, in a nonlinear dynamic model, several local optima exist, thus the equality of the parameter values resulted from the estimated optimum and the true ones is always an open question in such cases. The validation of the results is performed with the comparison to reference velocity signals. Four cases are examined, the velocity calculated with the derivation of the GNSS position measurements (GNSS position in the following figures), the signals estimated by the GNSS sensor based on the Doppler-effect (GNSS velocity) (Serrano et al., 2014), and the estimated velocity with

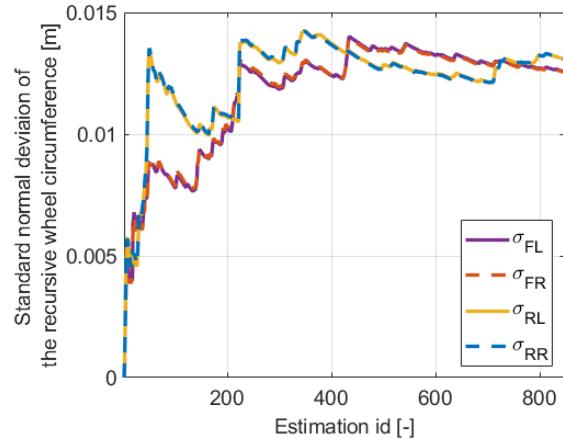


Fig. 11 Standard deviation of the estimated circumferences

Table 2 Estimated parameter values and their standard deviation

Parameter	Value	Standard dev.
Estimated wheel circumference FL	$c_{FL} = 1.943530$ m	$\sigma_{FL} = 12.6$ mm
Estimated wheel circumference FR	$c_{FR} = 1.944398$ m	$\sigma_{FR} = 12.5$ mm
Estimated wheel circumference RL	$c_{RL} = 1.947349$ m	$\sigma_{RL} = 13.1$ mm
Estimated wheel circumference RR	$c_{RR} = 1.949352$ m	$\sigma_{RR} = 13.1$ mm

the presented odometry model Eq. (7), with the calibrated parameters (odometry front and rear).

Parts of the estimated velocity signals and the reference one can be found in Figs. 12 to 15. The Fig. 12 illustrates

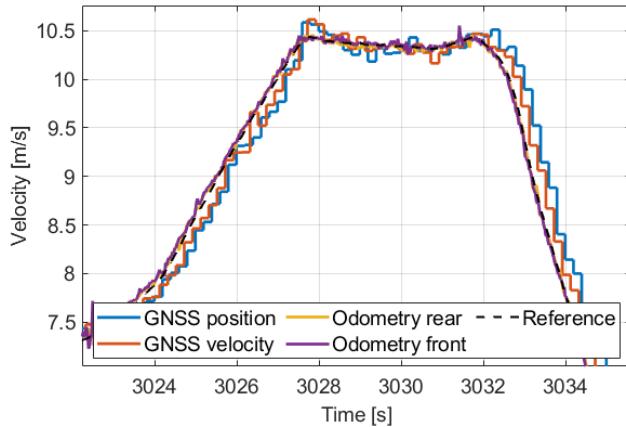


Fig. 12 The estimated velocity signal 1

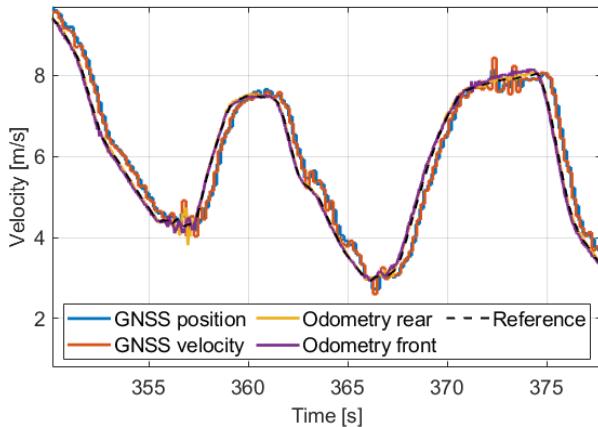


Fig. 13 The estimated velocity signal 2

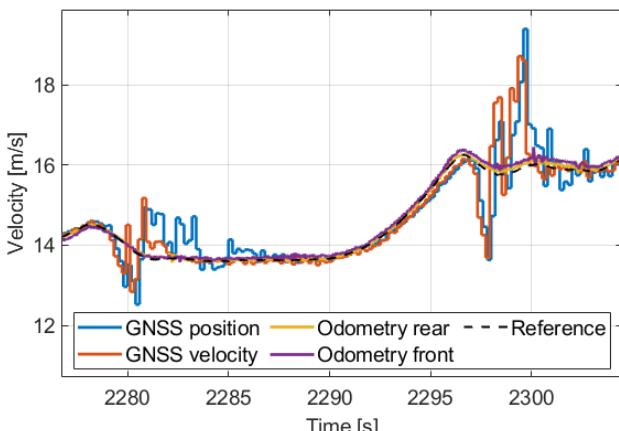


Fig. 14 The estimated velocity signal 3

the main advantage of the wheel-encoder based methods. The signals computed with the usage of the GNSS sensor have significant delay, due to the derivation and

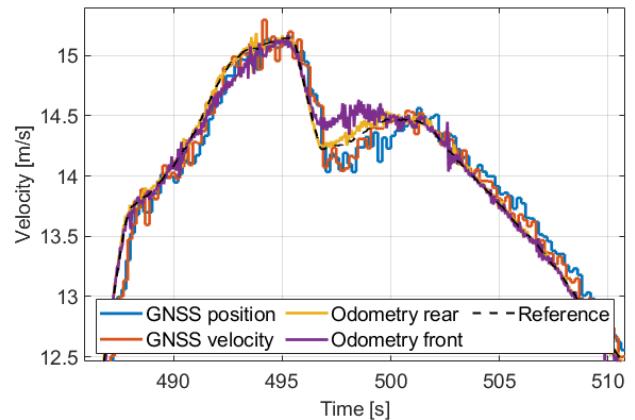


Fig. 15 The estimated velocity signal 4

the low 5 Hz sampling frequency of the sensor. Take into account that the delay resulted not only by the 0.2 s time-step, because the GNSS sensor contains internal optimization algorithms to determine the outputs. In contrast the velocity signals of the odometry models track the reference one clearly, see Fig. 13.

Furthermore, mainly in the higher speed region, high frequency and huge errors appear in the GNSS-based signals, which can be found in Fig. 14. These may be resulted because the radio waves were blocked by buildings and were not spread directly in the line-of-sight or a satellite became suddenly unavailable.

In some cases, the wheel encoder based signals have higher error, but these mostly appear in the front odometry case, see Fig. 15.

For the illustration of the accuracy of the different methods, the histograms of the error from the reference signals are generated. Figs. 16 and 17 show the histograms of the GNSS velocity and the rear odometry cases, respectively. The different resolutions of the distributions illustrate the advantage of the model-based estimation and the accuracy

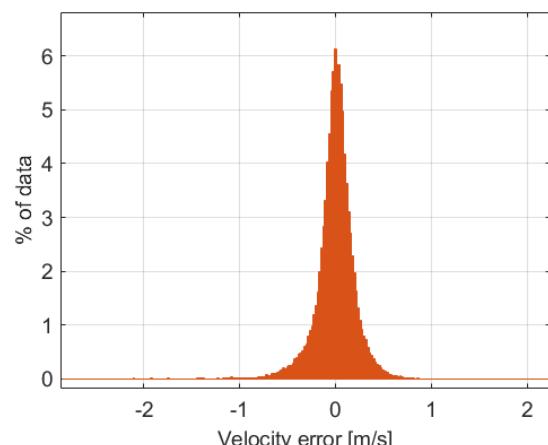


Fig. 16 Histogram of the velocity error of GNSS

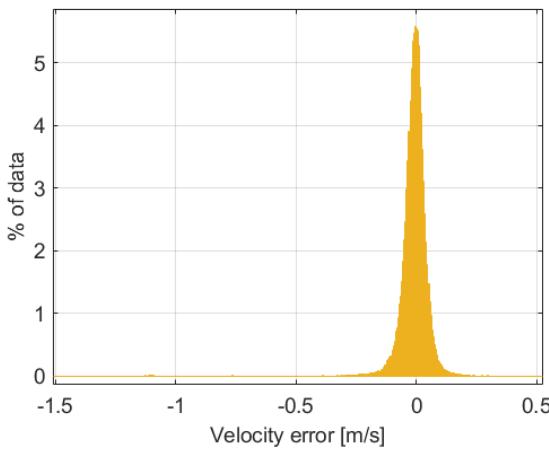


Fig. 17 Histogram of the velocity error of rear odometry

of the calibration. For comparison, the histograms of the 4 cases were generated with the same resolution and can be found in Fig. 18. The diagram validates the accuracy of the calibration, thus the reached optimum should be close to the true parameter values. The higher accuracy of the rear odometry can be explained by the lower sensitivity to the neglected sideslip values.

The disadvantage is the high-frequency changes in the signals which could result in unstable behavior if the signal is applied in a control loop. This problem can be measured with the formulation of the histogram of the error variation which is presented in Fig. 19 for the 4 cases. The histogram illustrates that the uncertainty of the odometry-based signals are 3–4 times lower than the GNSS-based ones. Furthermore, the sampling frequency is one order of magnitude higher, thus in precise control applications, the calibrated odometry is a feasible and cost-effective method for velocity estimation. A summary of the 4 methods can be found in Table 3, which illustrates that the accuracy of the rear odometry is

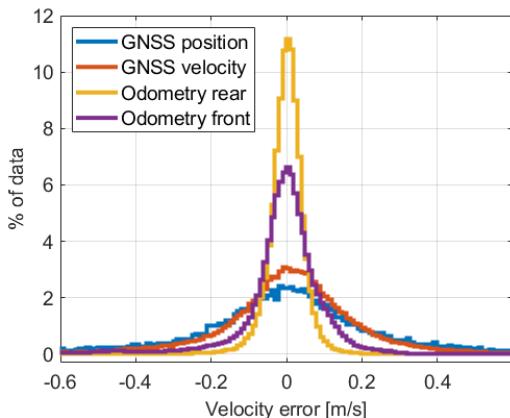


Fig. 18 Histogram of the velocity errors

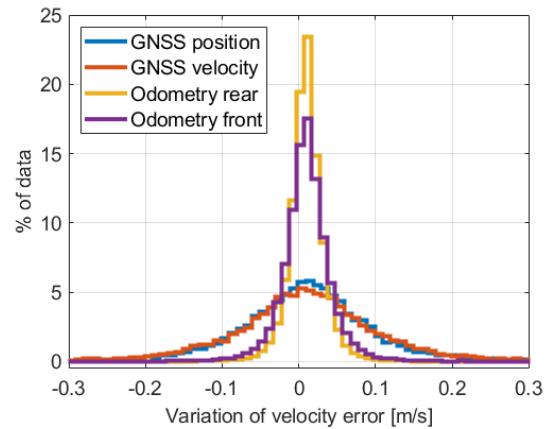


Fig. 19 Histogram of the error variations

Table 3 Error parameters of the 4 cases

Error	GNSS position	GNSS velocity	Odometry rear	Odometry front
Min [m]	-2.2349	-2.2169	-0.5177	-0.6679
Max [m]	4.5084	2.8475	1.5082	1.5122
Mean (abs.) [m]	0.1888	0.1464	0.0368	0.0766
Std [m]	0.1989	0.1588	0.0612	0.1128
Error variation	GNSS position	GNSS velocity	Odometry rear	Odometry front
Min [m]	-4.3858	-2.3610	-0.6867	-1.0058
Max [m]	4.4540	1.7079	0.4522	0.7325
Mean (abs.) [m]	0.0726	0.0779	0.0173	0.0241
Std [m]	0.1223	0.1148	0.0262	0.0372

3–4 times higher than the GNSS-based, and the front odometry performs better, as well.

6 Conclusion

In this paper, a velocity estimation algorithm was presented, which is based on the wheel encoder measurements. Since in the calculation the wheel circumferences are applied as parameters, the values are identified with a nonlinear least squares method. The algorithm listens to the incoming sensor measurements and estimates the wheel circumferences recursively from GNSS and IMU measurements. The experimental tests demonstrate the efficiency of the parameter estimation that results in accurate, and in parallel robust velocity estimation in a cost-effective way. The proposed method can be applied for the motion estimation and localization task in the driver assistance and self-driving functions.

In the future, an integrated state- and parameter estimation algorithm will be developed, with which the accuracy of both tasks can be improved.

Acknowledgement

The research was supported by the Ministry of Innovation and Technology NFDI Office within the framework of the Autonomous Systems National Laboratory Program.

The research was also supported by the National Research, Development and Innovation Office through

References

- Basargan, H., Mihály, A., Gáspár, P., Sename, O. (2020) "Integrated multi-criteria velocity and semi-active suspension control based on look-ahead road information", In: 2020 28th Mediterranean Conference on Control and Automation (MED), Saint-Raphaël, France, pp. 25–30.
<https://doi.org/10.1109/MED48518.2020.9182953>
- Basargan, H., Mihály, A., Gáspár, P., Sename, O. (2021) "Adaptive Semi-Active Suspension and Cruise Control through LPV Technique", Applied Sciences, 11(1), Article number: 290.
<https://doi.org/10.3390/app11010290>
- Bloesch, M., Omari, S., Hutter, M., Siegwart, R. (2015) "Robust visual inertial odometry using a direct EKF-based approach", In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, pp. 298–304.
<https://doi.org/10.1109/IROS.2015.7353389>
- Bohlmann, K., Marks, H., Zell, A. (2012) "Automated odometry self-calibration for car-like robots with four-wheel-steering", In: 2012 IEEE International Symposium on Robotic and Sensors Environments Proceedings, Magdeburg, Germany, pp. 168–173.
<https://doi.org/10.1109/ROSE.2012.6402609>
- Brossard, M., Bonnabel, S. (2019) "Learning Wheel Odometry and IMU Errors for Localization", In: 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, pp. 291–297.
<https://doi.org/10.1109/ICRA.2019.8794237>
- Brunker, A., Wohlgemuth, T., Frey, M., Gauterin, F. (2018) "Odometry 2.0: A Slip-Adaptive EIF-Based Four-Wheel-Odometry Model for Parking", IEEE Transactions on Intelligent Vehicles, 4(1), pp. 114–126.
<https://doi.org/10.1109/TIV.2018.2886675>
- Caron, F., Duflos, E., Pomorski, D., Vanheeghe, P. (2006) "GPS/IMU data fusion using multisensor Kalman filtering: introduction of contextual aspects", Information Fusion, 7(2), pp. 221–230.
<https://doi.org/10.1016/j.inffus.2004.07.002>
- Fazekas, M. (2019) "State and parameter estimation of Car-like robots", Master's thesis, Budapest University of Technology and Economics.
- Fazekas, M., Gáspár, P., Németh, B. (2020a) "Vision-based motion estimation for vehicles on test track via cone markers", In: 2020 11th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), Mariehamn, Finland, pp. 000387–000392.
<https://doi.org/10.1109/coginfocom50765.2020.9237845>
- Fazekas, M., Németh, B., Gáspár, P., Sename, O. (2020b) "Vehicle odometry model identification considering dynamic load transfers", In: 2020 28th Mediterranean Conference on Control and Automation (MED), Saint-Raphaël, France, pp. 19–24.
<https://doi.org/10.1109/mcd48518.2020.9182873>
- Fazekas, M., Gáspár, P., Németh, B. (2021a) "Challenges of the application of front-wheel odometry for vehicle localization", presented at 29th Mediterranean Conference on Control and Automation, Bari, Italy, June 22–25 2021.
- Fazekas, M., Gáspár, P., Németh, B. (2021b) "Calibration and Improvement of an Odometry Model with Dynamic Wheel and Lateral Dynamics Integration", Sensors, 21(2), Article number: 337.
<https://doi.org/10.3390/s21020337>
- Funk, N., Alatur, N., Deuber, R., Gono, F., Messikommer, N., Nubert, J., Patriarca, M., Scheafar, S., Scotoni, D., Bünger, N., Dube, R., Khanna, R., Pfeiffer, M., Wilhelm, E., Siegwart, R. (2017) "Autonomous Electric Race Car Design", presented at Electric Vehicle Symposium 30, Stuttgart, Germany, October 9–11 2017.
- Jung, D., Seong, J., Moon, C. B., Jin, J., Chung, W. (2016) "Accurate calibration of systematic errors for car-like mobile robots using experimental orientation errors", International Journal of Precision Engineering and Manufacturing, 17(9), pp. 1113–1119.
<https://doi.org/10.1007/s12541-016-0135-4>
- Kochem, M., Neddenriep, R., Isermann, R., Wagner, N. (2002) "Accurate local vehicle dead-reckoning for a parking assistance system", In: Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301), Anchorage, AK, USA, pp. 4297–4302.
<https://doi.org/10.1109/acc.2002.1024608>
- Lemmer, L., Heß, R., Krauß, M., Schilling, K. (2010) "Calibration of a Car-Like Mobile Robot with a High-Precision Positioning System", IFAC Proceedings Volumes, 43(23), pp. 174–179.
<https://doi.org/10.3182/20101005-4-RO-2018.00052>
- Moutarlier, P., Chatila, R. (1990) "An experimental system for incremental environment modelling by an autonomous mobile robot", In: Hayward, V., Khatib, O. (eds.) Experimental Robotics I. Lecture Notes in Control and Information Sciences, Springer, Berlin, Heidelberg, Germany, pp. 327–346.
<https://doi.org/10.1007/bfb0042528>
- Németh, B., Hegedűs, T., Gáspár, P. (2019) "Model Predictive Control Design for Overtaking Maneuvers for Multi-Vehicle Scenarios", In: 2019 18th European Control Conference (ECC), Naples, Italy, pp. 744–749.
<https://doi.org/10.23919/ecc.2019.8796013>
- Schanz, A., Spieker, A., Kuhner, K. D. (2003) "Autonomous parking in subterranean garages-a look at the position estimation", In: IEEE IV2003 Intelligent Vehicles Symposium, Proceedings (Cat. No.03TH8683), Columbus, OH, USA, pp. 253–258.
<https://doi.org/10.1109/ivs.2003.1212918>
- Schoukens, J., Ljung, L. (2019) "Nonlinear System Identification: A User-Oriented Road Map", IEEE Control Systems Magazine, 39(6), pp. 28–99.
<https://doi.org/10.1109/MCS.2019.2938121>

- Scott, I., Staniforth, A., Mitchell, W. C. (2006) "Analysis of Ackermann Steering Geometry", SAE Technical Paper, No. 2006-01-3638.
<https://doi.org/10.4271/2006-01-3638>
- Serrano, L., Kim, D., Langley, R. B., Itani, K., Ueno, M. (2014) "A GPS velocity sensor: How accurate can it be? - A first look", In: Proceedings of the 2004 National Technical Meeting of The Institute of Navigation, San Diego, CA, US, pp. 875-885.
- Tangirala, A. K. (2014) "Principles of System Identification: Theory and Practice", CRC Press, Boca Raton, FL, USA.
- Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., ... Mahoney, P. (2006) "Stanley: The robot that won the DARPA grand challenge", Journal of Field Robotics, 23(9), pp. 661–692.
<https://doi.org/10.1002/rob.20147>
- Toledo, J., Piñeiro, J. D., Arnay, R., Acosta, D., Acosta, L. (2018) "Improving Odometric Accuracy for an Autonomous Electric Cart", Sensors, 18(1), Article number: 200.
<https://doi.org/10.3390/s18010200>