AN AUTONOMOUS TRANSPORTATION VEHICLE FOR ALGORITHM AND SENSOR TESTING

NAVITEC 2018

5-7 December 2018
ESA/ESTEC, Noordwijk, The Netherlands

Benjamin Tennstedt and Steffen Schön

Leibniz University Hannover
Institut für Erdmessung (IfE)
Schneiderberg 50, D-30167 Hannover, Germany
Email: {tennstedt | schoen}@ife.uni-hannover.de

Abstract

This paper presents the implementation of an autonomous transportation vehicle as a kinematic test platform for navigation sensors and algorithms. The repeatability of the vehicle trajectory in an outdoor environment with GNSS-visibility will be shown. It is at the same level as most state of the art outdoor kinematic test sites. In addition, a first experiment in estimating the system's accuracy in areas without GNSS will be demonstrated using a motion model. This model is integrated into a linearized extended Kalman filter framework with autonomous sensors, i.e. IMU and on-board odometry, as control and observation data, respectively.

I. Introduction

To test new sensors and innovative filter algorithms and models for vehicle navigation, it is quite common to use a multisensor platform, equipped with an array of GNSS antennas, one or more inertial measurement units, as well as other sensors like LiDaR or stereo cameras, on the roof of a car for instants, and undergo different test scenarios with it. Mostly this strategy is very time-consuming, difficult to repeat, and several scientists have to be involved in the process [1]. Furthermore, when using rail-bounded systems, the flexibility in designing trajectories is lost, and they often lack the possibility to control the speed of the moving platform [2], [3]. To reduce time and resources needed for the preparation of those real tests, and to improve at the same time the repeatability of the test scenarios, a reliable system on a smaller scale is developed at the Institut für Erdmessung (IfE).

In modern logistics, such a system can be found in shape of small autonomous transport units. A sub-group of those vehicles, equipped with an optical line-following system and a propulsion unit able to turn around its height-axis, allows the use of motion models comparable to those of real cars with front wheel drives. The system enables the free definition of different reference trajectories and can be used as a carrier platform for a large number of sensor systems. The vehicle in use (KATE, or "Kinetic automate for transport enhancement") is equipped with an open CAN interface which allows accessing the process data of the systems propulsion unit, like the current turn rate and the wheel odometry. Its main purpose is to follow any given trajectories and velocity input, in any given environment, indoor and outdoor. In outdoor areas, GNSS visibility is provided and therefore a ground truth can be calculated by IMU and GNSS filtering, whereas it is more difficult to achieve a reference in indoor environment without satellite visibility. For this purpose, a proper motion model will be developed to solve the system equations and then later be merged with the odometry data in an extended Kalman filter framework.

A huge amount of different vehicle models at different levels of detail is available in literature, like e.g. [4], [5], but since some of the dynamics of the KATE vehicle can be neglected, a simpler version is sufficient. Current motion models often start from a time-discrete state space form without any information about the derivation of the equations [6], [7]. It seems that a relevant part of information could be lost in the progress that is needed for the calculation of the process noise for example. Therefore, this paper will start with a system of time-continuous differential equations describing the motion of the vehicle and all the following steps in developing the filter, to generate a solid base for any further work on this subject.

The remainder of the paper is structured as follows: Section II describes the solution derived from the differential equations of a motion model of the vehicle (II-A), as well as the set up of the filter to include velocity observations (II-B). Section III describes the experimental setup, with the results presented in section IV, namely the repeatability of the vehicle trajectory (IV-A), as well as a first experiment testing the vehicle model with simulated odometry, since the CAN-information is not yet accessible (IV-B). The results will then be summarized in Section V.

II. State estimation of the vehicle

A. Motion Model

Before starting to derive a motion model for the vehicle, a few assumptions have to be made to simplify the further steps. The motion model will be part of the prediction step of an extended Kalman filter framework, involving the use of acceleration

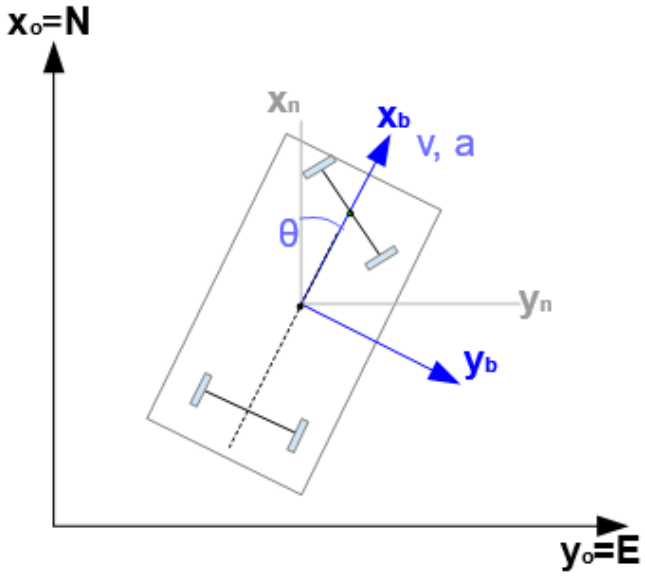

Fig. 1. Coordinate frames and system variables that are used. The IMU is positioned at the center of the b-frame (blue) and aligned with the orientation.

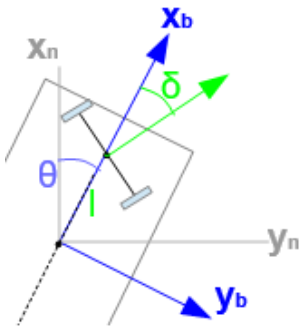

Fig. 2. Direction of the propulsion unit's velocity, if it was considered properly, including the lever arm **l** and the turn angle δ (both in green). In this paper, the simulated odometry is assumed to be already transformed into the bodyframe.

and angular rates of an IMU. This allows to predict the system states based on the data rate of the IMU.

The initial orientation between the body-frame of the vehicle (b-frame) and the navigation frame (n-frame) is estimated by a self-alignment. In addition, a strapdown algorithm will provide the corrected data in the body-frame of the vehicle, reducing Coriolis effects, Earth rate and transportation rate between the n- and b-frame. Furthermore, the orientation C_b^n is already calculated and will be used as additional input for the state estimation. Given the relatively low speed of the vehicle and the high update rate of the IMU, a linearly accelerated motion (constant acceleration, or CA) can be expected between two time steps. The differential equations describing the movement of the propulsion unit in xy-plane are given by:

$$\dot{x}(t) = v(t) \cdot \cos\theta(t) \tag{1}$$
$$\dot{y}(t) = v(t) \cdot \sin\theta(t) \tag{2}$$
$$\dot{\theta}(t) = 0 \tag{3}$$
$$\dot{v}(t) = a \tag{4}$$
$$\dot{a}(t) = 0 \tag{5}$$

Where x and y denote the position in North and East direction, respectively, while v equals the velocity in driving direction. The three variables are expressed as the state vector:

$$\mathbf{x} = \begin{pmatrix} x \\ y \\ v \end{pmatrix} \tag{6}$$

Additionally, the scalar acceleration a and the current yaw angle θ are introduced as control input:

$$\mathbf{u} = \begin{pmatrix} \theta \\ a \end{pmatrix} \tag{7}$$

The change of the state vector can be described as a function of the previous state and the current control input at a time t:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \tag{8}$$

To convert the equations into their state-space representation (9) a linearization would normally be needed. Thus, the partial derivatives of $\mathbf{f}$ with respect to the state vectors have to be calculated, each at a respective linearization point $\tilde{\mathbf{x}}, \tilde{\mathbf{u}}$. Since θ will be used as a constant (regarding the high data rate) control input at each epoch later on, the differential equations can already be considered linear and it is possible to work with the total state of $\mathbf{x}$ rather than an error state $\delta\mathbf{x} = \mathbf{x} - \tilde{\mathbf{x}}$ which would normally result from the linearization.

$$\dot{\mathbf{x}} = \mathbf{A} \cdot \mathbf{x} + \mathbf{B} \cdot \mathbf{u} \tag{9}$$

The matrices $\mathbf{A}$ and $\mathbf{B}$ read:

$$\mathbf{A} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\Big|_{\tilde{\mathbf{x}},\tilde{\mathbf{u}}} = \begin{pmatrix} 0 & 0 & cos(\tilde{\theta}) \\ 0 & 0 & sin(\tilde{\theta}) \\ 0 & 0 & 0 \end{pmatrix} \tag{10}$$

$$\mathbf{B} = \frac{\partial \mathbf{f}}{\partial \mathbf{u}}\Big|_{\tilde{\mathbf{x}},\tilde{\mathbf{u}}} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \tag{11}$$

To solve the system of linear time-invariant differential equations (Eq. 9) for the state vector $\mathbf{x}$, the matrix exponential $e^{\mathbf{A}t}$ has to be solved. The analytical method uses Laplace transformation. The transition matrix $\mathbf{\Phi}_{t,t_0} = e^{\mathbf{A}(t-t_0)}$ will then describe the transformation of the initial state vector to the next time step. Utilizing the $\mathcal{L}\{...\}$ operator, equation (9) results in:

$$\mathcal{L}\{\dot{\mathbf{x}}\} = \mathcal{L}\{\mathbf{A}\mathbf{x}\} + \mathcal{L}\{\mathbf{B}\mathbf{u}\} \tag{12}$$

$$s\mathbf{X}(s) - \mathbf{x_0} = \mathbf{A}\mathbf{X}(s) + \mathbf{B}\mathbf{U}(s) \tag{13}$$

Solving the equation for $\mathbf{X}(s)$ delivers:

$$\mathbf{X}(s) = [s\mathbf{I} - \mathbf{A}]^{-1}\mathbf{x_0} + [s\mathbf{I} - \mathbf{A}]^{-1}\mathbf{B}\mathbf{U}(s) \tag{14}$$

By defining $[s\mathbf{I} - \mathbf{A}]^{-1}$ as $\mathbf{\Phi}(s)$ and converting it back into time domain, the transition matrix $\mathbf{\Phi}_{(t,t_0)}$ can be obtained. The solution of the state space equation in time domain describing the transition between time t_0 and the following instance t is then:

$$\mathbf{x}(t) = \mathbf{\Phi}_{(t,t_0)}\mathbf{x}(t_0) + \mathbf{B}(t)\mathbf{u}(t) \tag{15}$$

The time-discrete version of (15) reads:

$$\mathbf{x}_k = \mathbf{\Phi}_k\mathbf{x}_{k-1} + \mathbf{B}_k\mathbf{u}_k \tag{16}$$

Recapturing (14), $\mathbf{\Phi}(s)$ results in:

$$\mathbf{\Phi}(s) = \begin{pmatrix} \frac{1}{s} & 0 & \frac{1}{s^2} \cdot \cos\tilde{\theta} \\ 0 & \frac{1}{s} & \frac{1}{s^2} \cdot \sin\tilde{\theta} \\ 0 & 0 & \frac{1}{s} \end{pmatrix} \tag{17}$$

and therefore:

$$\mathbf{\Phi}_k = \begin{pmatrix} 1 & 0 & \Delta t \cdot \cos\tilde{\theta} \\ 0 & 1 & \Delta t \cdot \sin\tilde{\theta} \\ 0 & 0 & 1 \end{pmatrix} \tag{18}$$

The control input matrix $\mathbf{B}_k$ can then be calculated by [8], p. 16:

$$\mathbf{B}_k \approx \frac{1}{2}(\mathbf{I} + \mathbf{\Phi}_k)\mathbf{B}\Delta t \tag{19}$$

This approximation to the linear term is exact, since $\mathbf{A}^n = \mathbf{0}$ for $n \geq 2$.
Finally, $\mathbf{B}_k$ results in:

$$\mathbf{B}_k = \begin{pmatrix} 0 & \frac{\Delta t^2}{2}\cos\tilde{\theta} \\ 0 & \frac{\Delta t^2}{2}\sin\tilde{\theta} \\ 0 & \Delta t \end{pmatrix} \tag{20}$$

B. Filter Model

Consider now the framework of a linearized EKF for the estimation of the state vector $\mathbf{x}$. First, the state will be predicted using the prediction model from the previous section:

$$\mathbf{x}_k = \mathbf{\Phi}_k \cdot \mathbf{X}_{k-1} + \mathbf{B}_k \cdot \mathbf{u}_k + \mathbf{G}_k \cdot \mathbf{w}_k \tag{21}$$

Note that a capital $\mathbf{X}$ is used for the filtered state vector to distinguish it from the predicted one. Additionally, a random noise vector $\mathbf{w}_k$ is introduced to model effects of the integrated acceleration noise.
As soon as the drive units scalar speed information v_{odo} is available, the state is then corrected:

$$\mathbf{X}_k = \mathbf{x}_k + \mathbf{K} \cdot \mathbf{i} \tag{22}$$

With the observation matrix $\mathbf{H} = [0, 0, 1]$ the predicted state can be directly projected into observation space to calculate the innovation $\mathbf{i}$:

$$\mathbf{i} = (v_{odo} - \mathbf{H} \cdot \mathbf{x}_k) \tag{23}$$

To weight the observation innovation $\mathbf{i}$, the Kalman Gain $\mathbf{K}$ has to be calculated. The optimal solution is defined by [9]:

$$\mathbf{K} = \mathbf{Q}_{xx,k} \cdot \mathbf{H} \cdot (\mathbf{Q}_{ll} + \mathbf{H} \cdot \mathbf{Q}_{xx,k} \cdot \mathbf{H}^T)^{-1} \tag{24}$$

The observation uncertainty is given in the matrix $\mathbf{Q}_{ll}$, which is apparently a scalar in this case. The state covariance matrix $\mathbf{Q}_{xx,k}$ is propagated and updated each iteration just like the state vector $\mathbf{x}$:

$$\mathbf{Q}_{xx,k} = \mathbf{\Phi}_k \cdot \mathbf{Q}_{XX,k-1} \cdot \mathbf{\Phi}_k^T + \mathbf{Q}_{ww,k} \tag{25}$$

$$\mathbf{Q}_{XX,k} = (\mathbf{I} - \mathbf{KH})\mathbf{Q}_{xx,k}(\mathbf{I} - \mathbf{KH})^T + \mathbf{K}\mathbf{Q}_{ll}\mathbf{K}^T \tag{26}$$

expressed in Joseph-Form for numerically reasons [10], p. 122, with $\mathbf{I}$ being the identity matrix.
Now the noise processes need to be defined. The state noise vector $\mathbf{w}$ is defined as a random ramp process based on an acceleration white noise density w_a. The time discrete process noise matrix $\mathbf{Q}_k$ can be estimated by [10], p. 153:

$$\mathbf{Q}_k = \int_0^\tau \mathbf{\Phi}_k(\tau)\mathbf{Q}\mathbf{\Phi}_k^T(\tau)d\tau \tag{27}$$

with $\mathbf{Q} = \mathbf{G}w_a\mathbf{G}^T$ as the acceleration white noise density projected on the state space via $\mathbf{G} = [0, 0, 1]^T$. Finally, with the interval $[0, \tau]$ being equal to Δt, the resulting discrete process noise matrix is:

$$\mathbf{Q}_k = w_a \cdot \begin{pmatrix} \frac{1}{3}\Delta t^3 \cos^2\theta & \frac{1}{3}\Delta t^3 \cos\theta \sin\theta & \frac{1}{2}\Delta t^2 \cos\theta \\ \frac{1}{3}\Delta t^3 \cos\theta \sin\theta & \frac{1}{3}\Delta t^3 \sin^2\theta & \frac{1}{2}\Delta t^2 \sin\theta \\ \frac{1}{2}\Delta t^2 \cos\theta & \frac{1}{2}\Delta t^2 \sin\theta & \Delta t \end{pmatrix} \tag{28}$$

Since the analytical solution becomes difficult for more complex systems, another common way to solve the integral in (27) is the numerical evaluation by Van Loan [11], which essentially leads to the same results with neglectible errors. Initially the Van Loan method was invented to solve the above derived matrix exponential $\mathbf{\Phi}_{(t,t_0)}$, so there may be another advantage if a quick and easy solution without too much insight into the equations is intended.

III. Experimental setup

For the following experiment the vehicle (cf. figure 3) is equipped with a fiber optical gyro IMU by iMAR [12] with a data rate of 200 Hz. Additionally, two GNSS antennas are added to observe the position and the heading of the vehicle. All sensors are connected to a Novatel SPAN Receiver [13]. Note that there is also a 360° prism in use to enable tracking the vehicle by a total station with autotarget function, but the total station data will not be discussed in this paper, due to the low data rate of 10 Hertz which is not high enough for the purpose of comparing the trajectory especially in the corner pieces. Instead, the reference trajectory is calculated by an INS/GNSS tightly coupled filtered solution, processed with the TerraPos software.
At the time of the recording of the data for this work, the programming of the CAN interface was not finished yet, which would include reading the current odometry data, recognizing RFID transponder codes and controlling the turn angle and the velocity of the car. The manufacturer's software "Transport Control" had to be used instead, which enabled us to command the velocity of the vehicle via wifi, while the trajectory, and therefore the direction of the moving platform, was determined by an optical contrast line.
The test trajectory is a round course with a dimension of about 4.1 by 1.6 meters, as depicted in figure 4. The mats are used to dampen the effects of the gaps between the tiles on the underground on the moving vehicle, including the acceleration measurements. The vehicle will drive more than twenty turns in clockwise direction.

IV. Results

A. Absolute Repeatability

At first, the repeatability of the system's trajectory estimated by the tightly coupled filtered solution is tested by calculating the scatter of the individual track with respect to the mean out of a set of turns.
To get the right time stamps of the beginning of each turn, the autocorrelation function of the whole signal in North direction is calculated. The maxima of this autocorrelation function show the time differences of the respective most similar position each turn, which correspond to the duration of each turn. Thus, a set of 17 consecutive single turns can be produced from the initial dataset, and the mean and standard deviation of the North and East position at each time step can be calculated. The mean deviation of each turn indicate the repeatability of the temporal and spatial position of the vehicle. In future implementations, the transponders can be used to determine the starting time of each track.
Figure 5 shows the trajectory as seen by the GNSS/INS solution in blue. The calculated mean is shown in red. In figures 7 and 8 the deviation of each single turn from the mean course is shown in East and North direction, respectively. There are two timespots where the deviations reach their minimum. They correspond to the parts of the course where the trajectories' orientation is perpendicular to the North or East direction. This indicates a higher across-track repeatability due to the line

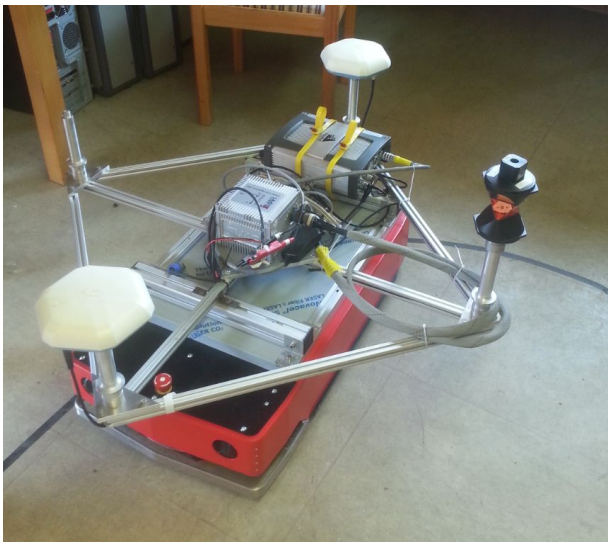

Fig. 3. Sensors used in the test scenario.

Fig. 4. Trajectory used in the experiment: dark rubber mats with white contrast tape for the optical trace sensor.

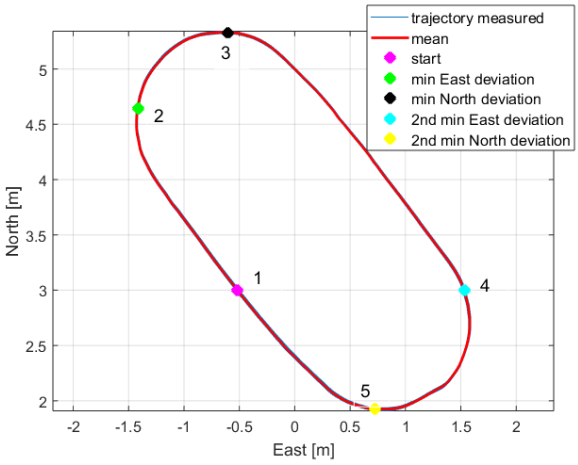

Fig. 5. Generating a mean course from several turns of the tightly coupled solution. The start of the course is indicated by position 1. The driving direction is clockwise.

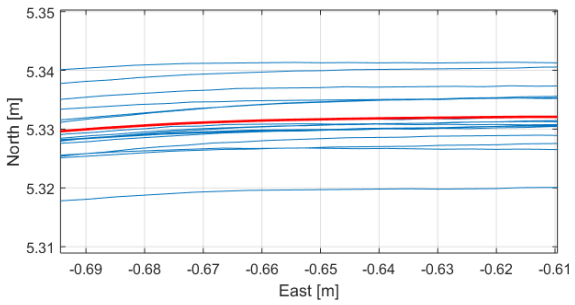

Fig. 6. Detail view of the measured tracks and the calculated mean track at position 3 in figure 5. In this section, the deviation in North-direction has the smallest value because of the perpendicular track and the lateral accuracy of the line following controller. The same circumstance is also valid for position 5 in the South of the track, as well as positions 2 and 4 regarding the deviation in East-direction.

follower sensor.

The timeseries of the standard deviations is shown in figure 9. Most of the values are between 0.5 and 1.5 centimeters. The cumulative distribution in figure 10 illustrates that area more clearly. The mean value of the standard deviations in North direction is 11.1 millimeters, the mean in East direction is slightly smaller with about 9.7 millimeters, and so is the repeatability of the trajectory.

Physically, the along-track accuracy of the system is given by 2 millimeters at a speed of 20 centimeters per second [14], while the lateral accuracy is given by 3 millimeters. Unfortunately, no information is given about the accuracy at higher speeds, as the used $1\frac{m}{s}$ (which is most probably worse).

B. Solution without GNSS Visibility

To find out how well the vehicle's state can be estimated when only relying on the on-board sensors and an IMU in difficult areas without GNSS-visibility and absent from the total station's line of sight, the linearized Kalman filter framework developed before will be used and compared to the tightly coupled INS/GNSS relative solution processed in the previous section. Since

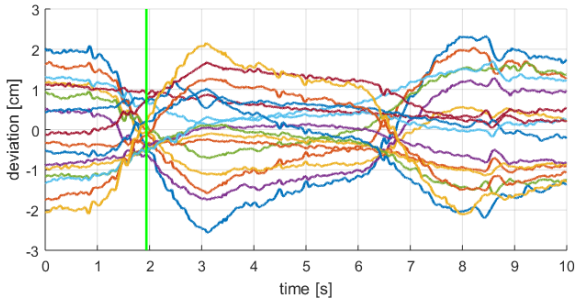

Fig. 7. East direction. Showing the deviations from the calculated mean of each single turn. The green vertical line represents the minimum of the deviations, also seen in figures 5 and 9.

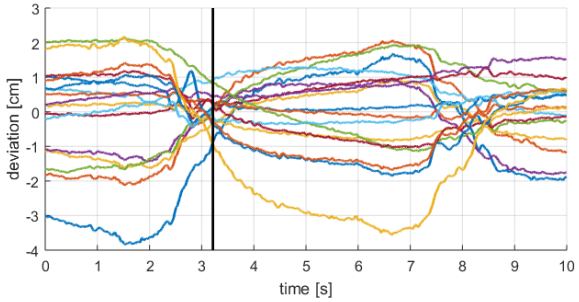

Fig. 8. North direction. Showing the deviations from the calculated mean of each single turn. The black vertical line represents the minimum of the deviations, also seen in figures 5 and 9.

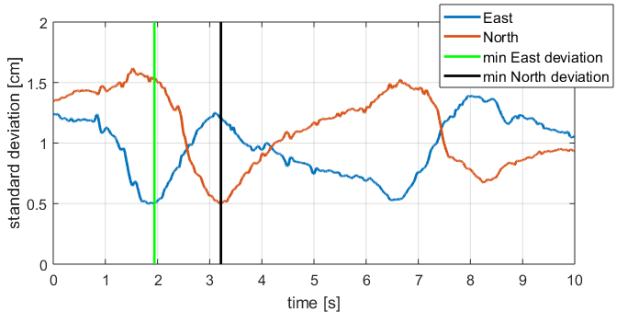

Fig. 9. The timeline of the mean of all deviations in East and North direction.

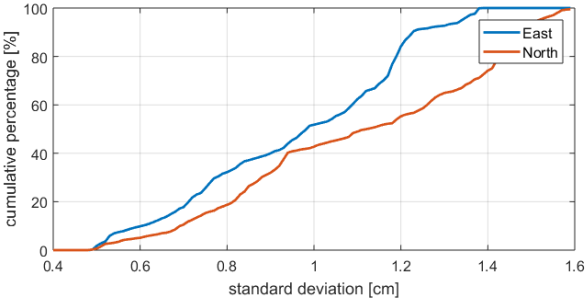

Fig. 10. Cumulative deviation in East and North direction.

the odometry data is not available yet, a simulated vehicle velocity based on the tightly coupled solution will be used instead. The results delivered this way enable first insights into the behaviour of the filter. To keep the results clear, only the first round trip of the filtered solution is shown, this time including the acceleration part, cf. figure 11. Figure 12 indicates the deviations in North and East direction between the EKF- and the INS/GNSS solution.

The velocity in figure 13 shows a few interesting things. Right after the start the velocity takes about two seconds to reach its final value - right before the corner starts. Therefore, the acceleration should have a value of about 0.5 meters per squared seconds, which aligns with the parameter of the maximum acceleration set in the microcontroller of the propulsion unit by the manufacturer. This value can be changed at will, but since wheel slip and limits of the electric current supply necessary to build up force will occur at some point, the optimal value has to be carefully estimated in another work.

Also quite prominent is the oscillation of the velocity between about $0.9\frac{m}{s}$ and $1.0\frac{m}{s}$. This is mainly due to the fact that while the propulsion unit follows the optical contrast trace with the speed held by its controller, the rest of the vehicle is dragged behind, with the IMU, and the origin of the bodyframe alike, positioned at the connecting line between the height axis of the drive and the back of the vehicle (as depicted in figure 1). When using real odometry from the unit, the observed velocity data has to be transformed into the bodyframe first. Another possible reason for the deviation of the velocity are effects due to the wheel slip, especially in the corner parts of the track, even though they should be kept small because of the software-implemented differential drive.

Lastly, the detail view in figure 14 shows that in the short straight piece of the trajectory in-between seconds 6 and 8, the control velocity is reached very well, with some small deviations.

Based on these first experimental results, the following topics need to be considered in more detail:

- It is quite obvious that the error in orientation grows boundlessly with time, so there is a necessity to add another sensor for an observation at least for the yaw angle. Furthermore, in enhancing the strapdown algorithm used for calculating the attitude by taking coning effects into account, the drift of the angle can be reduced.
- Even though the turn angle of the vehicle does not provide absolute attitude information, it can be used in the transformation from the propulsion unit frame to the body frame of the vehicle, which will be necessary for the CAN odometry data (cf. figure 2).
- Using a constant turn rate and acceleration model (CTRA) instead of the CA model has shown no improvement on the results in this work, since the IMU turn rates are already integrated in the strapdown algorithm the same way they would be integrated in the motion model. By taking the vehicle odometry and turn rates into account instead, there will be a

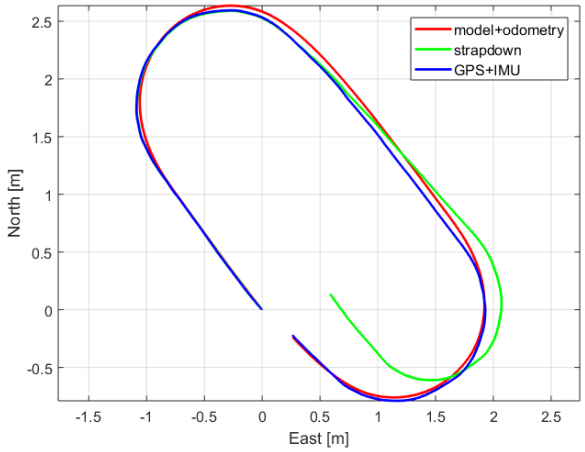

Fig. 11. Linearized EKF-solution with vehicle model and simulated odometry (red) vs. tightly coupled INS/GNSS solution (blue). Origin of the frame is the start point of the vehicle. For a comparison, the IMU strapdown solution (green) is included.

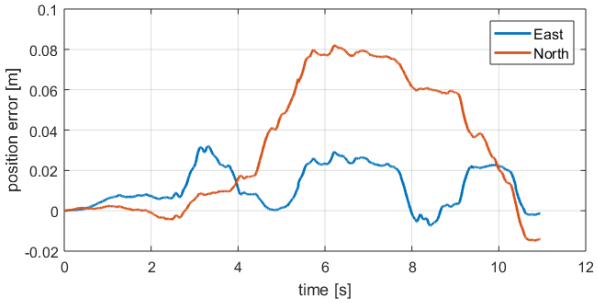

Fig. 12. Position error of the EKF solution with regards to the INS/GNSS solution.

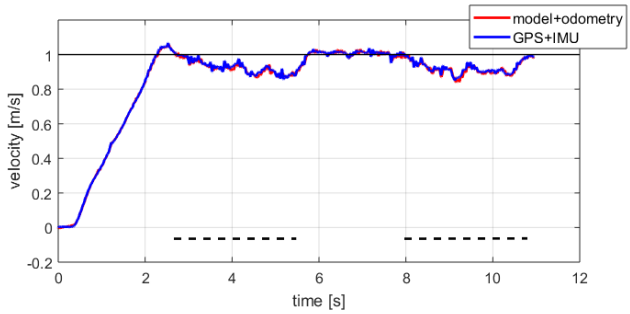

Fig. 13. Velocity state. The commanded velocity transmitted by "Transport Control" via wifi is 1 meter per second (black horizontal line). The difference in the corners (about seconds 3 to 6, and 8 to 11, indicated by black dashed lines) can be explained by the lever arm of the driving unit to the bodyframe centered in the IMU position, as well as the random effects of slipping wheels leading to a wrong odometry.

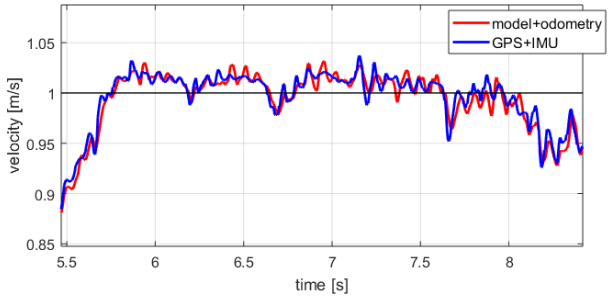

Fig. 14. Detail view on the velocity in the straight piece of the track.

few more options to combine the different sensors.

- To create a solution really independent from external observations, the use of transponders for absolute position information will be investigated.

V. Conclusion

The repeatability of a simple experiment in a round track was shown by analyzing the results of a GNSS/INS tightly coupled relative filtered solution. Out of the test sites of other institutions, the University of Nottingham did a comparable kinematic experiment [2]. Using a Network RTK, they reached a mean offset of less than 13 millimeters on the horizontal plane with their rail-based locomotive system for some chosen reference points. With KATE and the tightly coupled INS/GNSS solution a mean horizontal offset of 14.7 millimeters has been reached for the whole test trajectory.
A first experiment for calculating an indoor reference solution without using external sensors has been realised. Some additional topics to improve this solution have been stated in section IV-B.
The vehicle offers several advantages. The ability to control its velocity and the option to freely generate reference trajectories makes it a powerful and versatile tool for testing navigation sensors and algorithms.

Acknowledgment

This work was partially supported by i.c.sens Research Training Group (GRK2159).

References

[1] S. Schön et al., "Integrity and Collaboration in Dynamic Sensor Networks," *Sensors*, 2018, 18, 2400, June 2018.
[2] S. Stephenson, X. Meng, T. Moore, and T. Edwards, "Precision of Network Real Time Kinematic Positioning for Intelligent Transport Systems," Nottingham Geospatial Institute, University of Nottingham, Nottingham, 2011.
[3] E. Heinz, L. Klingbeil, and H. Kuhlmann, "Development of a Test Field for the Calibration and Evaluation of Kinematic Multi Sensor Systems," *Geodätische Woche 2017*, Berlin, 2017.

[4] X. R. Li and V. P. Jilkov, “Survey of Maneuvering Target Tracking. Part I: Dynamic Models,” *IEEE Transactions On Aerospace And Electronic Systems*, vol. 39, no. 4, October 2003.
[5] J. Ackermann, J. Guldner, W. Sienel, R. Steinhauser, and V. I. Utkin, “Linear and Nonlinear Controller Design for Robust Automatic Steering,” *IEEE Transactions On Control Systems Technology*, vol.3, no. 1, March 1995.
[6] R. Schubert et al., “Empirical Evaluation Of Vehicular Models For Ego Motion Estimation,” *IEEE Intelligent Vehicles Symposium (IV)*, Baden-Baden, Germany, June 2011.
[7] G. Zhai, H. Meng, and X. Wang, “A Constant Speed Changing Rate and Constant Turn Rate Model for Maneuvering Target Tracking,” *Sensors*, 2014, 14, pp. 5239-5253, March 2014.
[8] J. Wendel, “Integrierte Navigationssysteme. Sensordatenfusion, GPS und Inertiale Navigation,” Oldenbourg, 2007.
[9] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” *ASME Transactions*, vol. 82, part D (Journal of Basic Engineering), pp. 35-45, 1960.
[10] M. S. Grewal and A. P. Andrews, “Kalman Filtering: Theory and Practice Using MATLAB,” John Wiley & Sons Inc., 2001.
[11] C. F. Van Loan, “Computing Integrals Involving the Matrix Exponential,” *IEEE Transactions On Automatic Control*, vol. ac-23, no. 3, June 1978.
[12] iMAR Gesellschaft für inertiale Mess-, Automatisierungs- und Regelsysteme mbH, “iIMU-FSAS-E-EI-SN,”, http://www.imar-navigation.de, last access on 31.10.2018.
[13] Novatel, “SPAN-SE,” http://www.novatel.com/assets/Documents/Papers/SPAN-SE.pdf, last access on 31.10.2018.
[14] Götting KG, “Fahrlenkmodul HG G-08230YA FLM01 für Kleinfahrzeuge,” http://www.goetting.de, last access on 31.10.2018.