

Precise positioning through a loosely-coupled sensor fusion of GNSS-RTK, INS and LiDAR for autonomous driving

Andreas Schütz

Institute of Space Technology and Space Applications
Bundeswehr University Munich
Munich, Germany
andreas.schuetz@unibw.de

Daniela E. Sánchez-Morales

Institute of Space Technology and Space Applications
Bundeswehr University Munich
Munich, Germany
daniela.sanchez@unibw.de

Prof. Thomas Pany

Institute of Space Technology and Space Applications
Bundeswehr University Munich
Munich, Germany
thomas.pany@unibw.de

Abstract—In this paper we describe the integration done between GNSS-RTK/INS/LiDAR in a loosely coupled Kalman Filter in the context of autonomous driving applications. Specifically, we focus in the assessment of potential LiDAR updates by comparing the velocity profile obtained by the GNSS/INS integration solution and the LiDAR observations. The results from a test drive are shown to provide an insight of the advantages of using the LiDAR updates in GNSS denied environments.

Keywords—*sensor fusion; GNSS/INS/LiDAR integration; loosely coupled filter; precise positioning; autonomous driving*

I. INTRODUCTION

Accurate, reliable and robust positioning and navigation is one of the most challenging and key aspects for autonomous driving today. At the Institute of Space Technology and Space Applications of the Bundeswehr University Munich, research projects taking on all three of these features are conducted. As common research platform among the project groups, we use our Institute's own navigation software, the Multi-Sensor Navigation and Analysis Tool (MuSNAT) [1]. Based upon the ipexSR from the late 2000s, a GNSS software receiver based navigation engine was developed. At the current state multi-frequency, multi-GNSS observations can be processed in various GNSS techniques, and additional sensors such as IMU and LiDAR are supported. The MuSNAT may be tailored to the requirements of the application, meaning that all grades of antennas, IMUs and LiDARs are supported within processing. The MuSNAT also comprises an additional simulation mode, in which GNSS observations are synthesized at IF sample level, which can be used for testing and validation also in combination

with artificial IMU data. This work, however, focuses on the accuracy and robustness aspects of positioning and navigation for automotive applications, especially on the feasibility of LiDAR aided GNSS/INS with respect to pure GNSS/INS in GNSS denied and degraded regions.

It was shown previously [2] in UAV applications that LiDAR measurements can be used to improve the GNSS only positioning significantly, namely from a mean squared error of 2 meters to 1 meter, yet this is very susceptible to the definition of the LiDAR velocity noise model, the pointcloud features used for velocity computation and the GNSS positioning technique. This work however investigates the feasibility of LiDAR velocity updates for a street vehicle, requiring the position accuracy restraint of “driving lane accuracy”, i.e. around 50 cm.

The core of the navigation unit is a loosely-coupled GNSS/INS error state Kalman filter according to [3]. To ensure high quality measurements and best possible estimation results using GNSS updates, only ambiguity-fixed RTK positions are considered valid, while float and SPP solutions are discarded. The open source software RTKLib, accessed through a wrapper in the MuSNAT, serves as RTK engine for this purpose, given also that the above mentioned filter was developed within a cooperation between the Bundeswehr University Munich and the Tokyo University for Marine Science and Technology (TUMSAT), where at the latter it is currently integrated as new feature in the RTKLib and will be available for download in the near future.

However, in an automotive scenario, especially in urban areas, RTK is not available for most of the time, and neither SPP nor float solutions render the level of accuracy and robustness

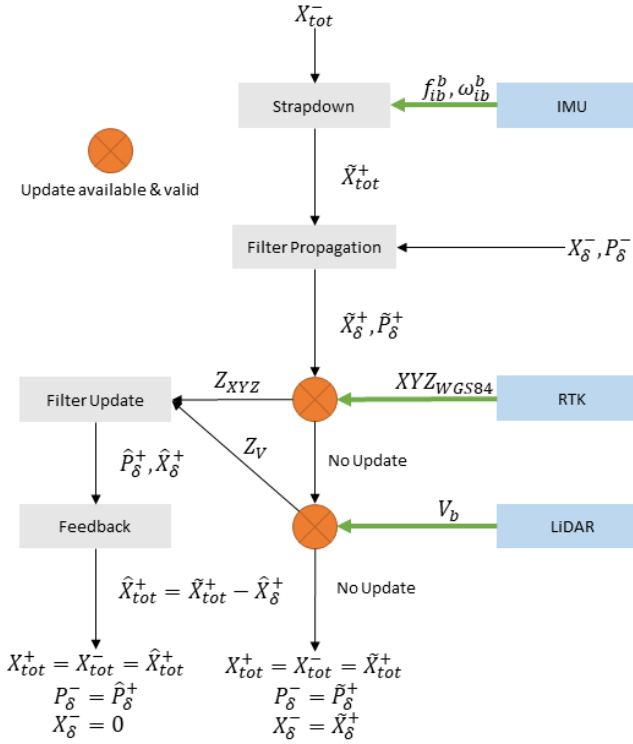


Figure 1: Navigation engine processing flowchart

necessary for reliable GNSS filter updates at driving lane accuracy. By discarding all non RTK updates, we treat all epochs with float or SPP solutions as GNSS outage. Presuming the availability of RTK at filter initialization phase to provide accurate enough bias estimates, we attempt to stabilize the strapdown algorithm through LiDAR velocity updates when entering regions of degraded GNSS measurements in contrast to an IMU only solution.

Within the scope of this paper, we first introduce our GNSS/INS/LiDAR navigation engine currently being developed and embedded in the MuSNAT and its underlying algorithms. For its latter application, that is the navigation of cars, we want to investigate the use of LiDAR measurements in GNSS denied regions to mitigate the INS solutions' drift. Therefore we construct an experiment of an extreme case of GNSS outage during the vehicle trajectory. Then we investigate the provided LiDAR measurements regarding their feasibility to serve as measurement updates instead of GNSS for the navigation filter, and, finally, we give an outlook on the future development focus based on the knowledge we gathered during this campaign.

II. GNSS-RTK/INS/LiDAR INTEGRATION

A. Loosely coupled Kalman Filter architecture

The current navigation engine comprises of a loosely coupled GNSS/INS error-state Kalman Filter according to [3] and [4] estimating 15 states, namely position error in m (WGS84), velocity error in $\frac{m}{s}$ (NED), attitude error in rad (Roll,

Pitch, Yaw), static accelerometer bias in $\frac{m}{s^2}$ (IMU body axes) and static gyroscope bias in $\frac{rad}{s}$ (IMU body axes). Supported update types are position only, velocity only and position/velocity combined. Whereas position updates are a pure GNSS domain at the moment, velocity updates may also be provided by other sensors, specifically LiDAR in this case of investigation. Figure 1 depicts the implementation algorithm of the sensor fusion algorithm in the MuSNAT, where X denotes the states with subscripts tot for the total states such as position, velocity, attitude and static biases on accelerometers and gyroscopes, and δ for the corresponding error states, $\hat{\cdot}$ the estimation and \sim the propagation. Superscripts $-$ and $+$ indicate the previous respective current epoch. Z denotes the Kalman Filter observation vector, XYZ a position and V a velocity. Subscript/superscript b indicates the vehicle body frame, $WGS84$ the GNSS frame, i the inertial frame and finally f and ω denote the accelerations and turn rates sensed by the IMU.

The strapdown computation and filter propagation are done at IMU measurement rate, where the strapdown algorithm comprises precision attitude updates [3] for both its selectable attitude computation modes (Euler or Quaternions) and all rotating-frame related corrections such as the Coriolis effect are removed. Time synchronization between IMU, RTK position and LiDAR velocity is realized by an independent GPS timestamp on all three sensor measurements.

Currently solely RTK fixed positions are accepted as GNSS updates, with a measurement accuracy fixed to conservative $0.03 m$ horizontally and $0.05 m$ vertically. In order to be accepted, the observation Z_{XYZ} must be within twice the predicted standard deviation of the total position states σ_{XYZ} , i.e.

$$Z_{XYZ} < 2 * \sigma_{XYZ}, \quad (1)$$

where σ_{XYZ} consists of the first three elements in $\sqrt{\text{diag}(\hat{P}_{\delta}^+)}$. Furthermore the potential update must be preceded by an already ambiguity-fixed position, in order to avoid single false fixes within regions of otherwise only float or SPP solutions.

The LiDAR updates are constructed slightly different. Since the LiDAR yields velocities in the vehicle body frame and the INS solution is represented in North-East-Down (NED), the LiDAR velocities need to be transformed to NED in order to compute Z_V . The transformation is given by the current Roll, Pitch and Yaw angle of \hat{X}_{tot}^+ . This however introduces the accumulated attitude error to the independent LiDAR measurements. At this point an outlier rejection would no longer be robust and the filter observations depend on the propagated filter states, rendering them statistically dependent. For these reasons, the propagated NED velocity must be transformed to the vehicle body frame, containing all accumulated errors so the LiDAR velocity can be introduced as a truly independent observation.

Whenever concurrent updates occur, i.e. a valid GNSS update and a valid LiDAR update correspond to the same IMU measurement, the GNSS update is always preferred and the LiDAR update is rejected. It has to be noted though that the GNSS update rejection as in equation 1 needs to be improved to

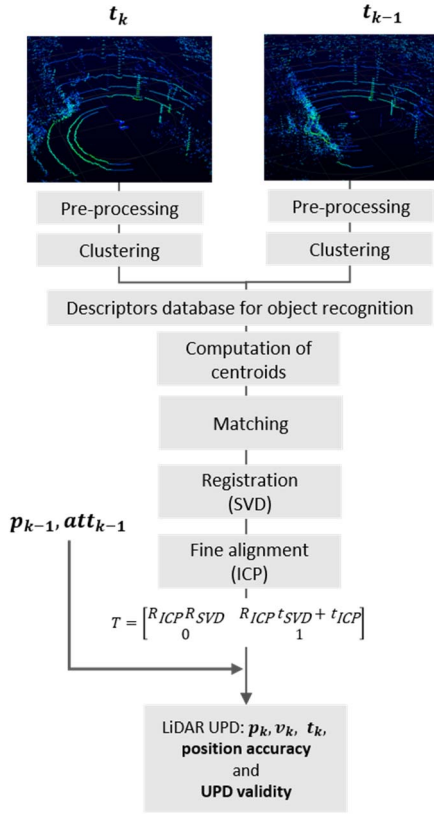


Figure 2: Pointcloud processing workflow

a more sophisticated model regarding the detection of false fixes, and an outlier rejection for LiDAR updates in the same manner as the current one for GNSS updates is not sufficient.

B. LiDAR velocity computation

As seen in the previous section, the integration between GNSS and INS sensors is implemented using a loosely coupled architecture. INS sensors are capable to provide real-time measurements, computing position, velocity and attitude by the integration of the observed accelerations at a low computation complexity and at a high frequency. Nonetheless, the

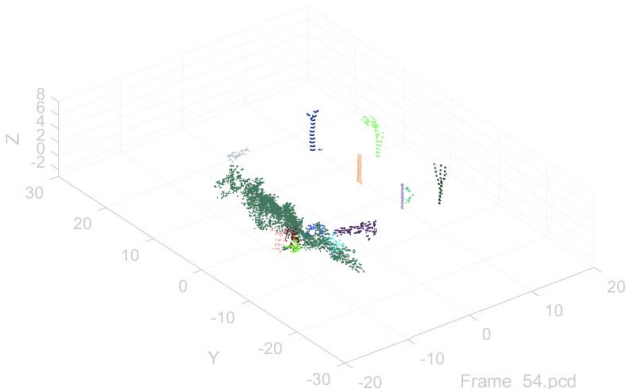


Figure 3: Example of an initial segmentation to identify clusters for a specific frame

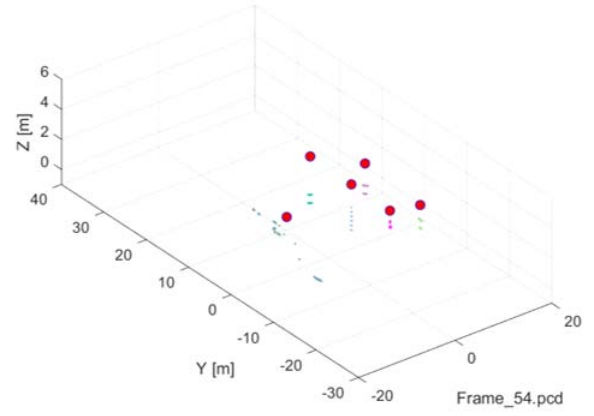


Figure 4: Example of clusters identified as cylinders for a specific frame (red dots represent the centroid of the object)

accumulated error over the time during the integration has a result a drift in the navigation solution. To overcome this disadvantage, especially during long-term GNSS outages, we target to use LiDAR observations as updates.

LiDAR is an active sensor with a laser source that can be used independently of the presence of natural or artificial light sources, in contrast with visual sensors (e.g. cameras). The approach chosen in our work is a relative navigation algorithm based on feature detection. Figure 2 shows the workflow from the procurement of the point clouds until the estimation of the values to be given to the filter.

1) Pre-processing

The cloud is filtered based on a maximum distance. We limit the area of interest in which we search for the features. Objects that are far from the sensor are likely to be missed by the algorithm as they have few scan lines crossing them. Therefore, we focus in the section where objects are denser in points.

2) Clustering

The cloud is segmented into smaller sets based on the Euclidean distance. Each of these sets are assumed to represent single objects and are called clusters. A maximum and minimum number of points belonging to the cluster is configured, as well as the threshold to define whether a point belongs to the cluster or not.

3) Descriptors database for object recognition

In this module a series of descriptors are gather in order to compare their performance and use cases. In the current work, two combinations were tested: 1) the ESF descriptor [5], which is meant to work better for partial point clouds and computes a histogram describing the characteristic properties of the analyzed cluster. This method do not require the computation of the surface's normal as it is often seen in other descriptors and therefore, improves the processing speed; 2) Cylinder recognition employing RANSAC [6], under the assumption that fixed objects like trees and light poles cab be described by this model. The algorithm switches back to the ESF when not enough cylinders are found in the current pair of frames.

4) Computation of centroids and matching

In the case of using the ESF descriptor, the computed signatures of the clusters are stored in order to compare them with the signatures of the next frame. Those which match and also fulfill a maximum distance threshold are considered to belong to the same cluster.

On the other hand, with the cylinders a matching procedure is also performed with the nearest neighbor of the next epoch, taking care of multiple assignments, fulfilling a maximum distance threshold and having reciprocal correspondences in both epochs. In Figure 3 is shown an example of a clustered point cloud and in Figure 4 the respective clusters that fulfilled the cylindrical model. Finally, the centroids of the resulting clusters are computed.

5) Registration

A coarse alignment is performed using the centroids obtained in the previous section through the Singular Value Decomposition method [7, 8]. As a result, we obtained a rotation matrix and a translation vector, which is applied to the original cloud.

When any of the centroids sets do not have enough correspondences, the validity of the registration is set as an unsuccessful operation.

6) Fine alignment

Once we get a coarse rotation and translation, we applied it to the original point cloud and use the ICP algorithm to bring it closer to the cloud belonging to the next epoch. The initial rough estimate helps the ICP to converge faster to a solution, giving the final transformation matrix.

7) LiDAR Update

Finally, the position computed in the previous epoch is multiplied with the final transformation matrix, giving the position of the vehicle in the current epoch. As we have the initial and final position and the time tags of both frames, the velocity can be computed.

The update is set to valid if the registration is successful and the ICP converged with a certain score. And finally, the accuracy of the update is set to the ICP score, which is the sum of squared Euclidean distances resulted from the ICP.

Disadvantages of this approach occur when the environment is featureless or when we get a poor outcome from the feature extraction module. In can be seen the direct effect of a drop in the number of features and the velocity outliers. In principle, even with few features, as long as they represent correctly and accurately real objects in the environment, a good estimate of the displacement can be computed, although this is not often the case.

III. EXPERIMENTAL SETUP

A. Equipment

The equipment in the experiment is listed in Table 1. We can see 2 LiDARs mounted in each side of the vehicle, as well as the X-Sens IMU and the Zephyr GNSS antenna. Considering

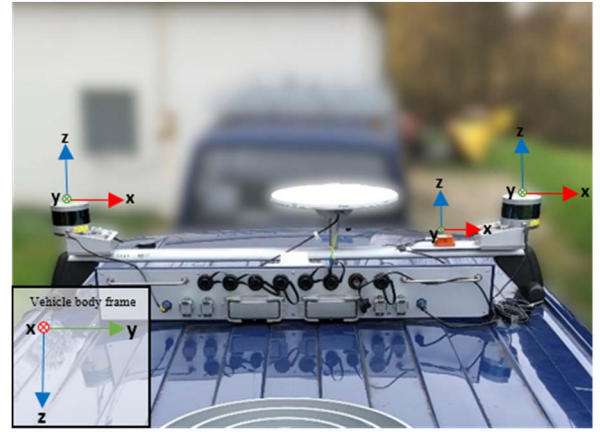


Figure 5: Sensor setup and coordinate system definitions

an automotive scenario, especially the grade of GNSS antenna and receiver are too high, yet the IMU, although being again of slightly higher grade than the average automotive sensors, is representative for the MEMS category. The commercial receiver used at the moment will be replaced by ISTA's software receiver. The sensors and GNSS antenna are mounted on top of our measurement vehicle (compare Figure 5 with the respective sensor coordinate orientations). The vehicle frame, also previously mentioned as body frame, has its origin in the center point of the IMU body coordinate frame, with its X axis pointing towards the vehicle nose, the Y axis pointing towards the passengers' side, and the Z axis pointing down. All further velocity plots of both the IMU and the LiDAR refer to this frame. All of the sensors were mounted in the same rail, therefore the lever arm consists only in offsets along the Y- and Z-axis, which were measured with a sub-centimeter level accuracy. Respective axis rotations and lever arms are accounted for and not mentioned in the processing flow for reasons of display. As mentioned above, only RTK ambiguity-fixed positions fulfilling the criteria from section II are accepted as valid GNSS updates for the navigation filter.



Figure 6: GNSS/INS trajectory with solution types

TABLE 1: LIST OF SENSORS AND CONFIGURATION

Sensors/ Subsystem s	System description	
	Measurement rate	Performance specification
GNSS Trimble NETR9 & Zephyr 2 antenna	1Hz	<ul style="list-style-type: none"> GNSS: GPS, Galileo
IMU MTi-G-710	400 Hz	<ul style="list-style-type: none"> Gyroscopes Initial bias error: 0.2 [deg/s] In-run bias stability: 10 [deg/h] Noise density: 0.01 [deg/s/$\sqrt{\text{Hz}}$] Accelerometers Initial bias error: 0.05 [m/s²] In-run bias stability: 10 [deg/h] Noise density: 0.01 [deg/s/$\sqrt{\text{Hz}}$]
LiDAR VLP-16	10 Hz	<ul style="list-style-type: none"> Accuracy: +/- 2 cm Hz. angular resolution: 0.1-0.4 [deg] Vert. angular resolution: 2 [deg] 903 nm wavelength Phase lock activated
SAPOS Base station	1 Hz	<ul style="list-style-type: none"> Baseline: approx. 8 [Km]

B. Trajectory

The biggest problem when using an IMU in a positioning system, especially a MEMS IMU, is the solution drift caused by the accumulated biases of the sensor. This issue becomes particularly present whenever the vehicle is performing low-dynamic maneuvers, and the movement sensed by the IMU is somewhat close to the noise level of the sensor. If this is the case within an area of no or very few high quality GNSS updates, the computed position and velocity will drift away from the true value due to the increasing error in the measured turn rates and accelerations. Especially in a scenario like this, we attempt to use the LiDAR to mitigate this drift. Figure 6 shows exactly such a scenario, with 65 seconds of GNSS outage and a long straight drive with only two significant maneuvers but a straight line and very moderate velocity (max. 10 m/s), with no stationary phase. It is worth to be mentioned, that this is a critical scenario and does not represent the typical environment. The trajectory

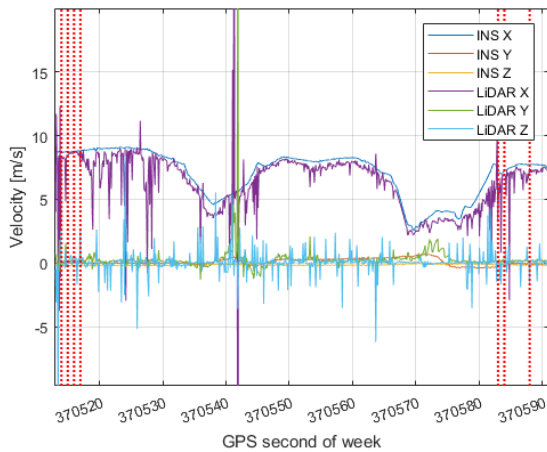


Figure 7: Velocities in vehicle body frame obtained from LiDAR and INS

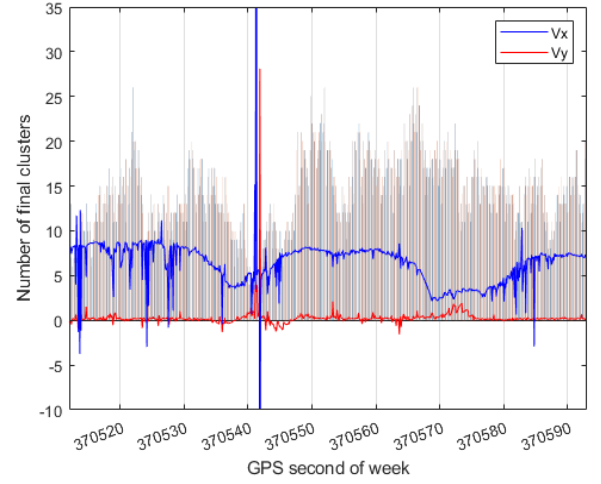


Figure 8: Number of clusters used for registration with corresponding velocity estimate

begins on the left hand side of Figure 6. At the end of the GNSS outage (right hand side), a large jump in the position is visible after the GNSS update.

IV. EVALUATION OF POTENTIAL LiDAR UPDATES

A. Velocities

The body frame velocities of both the GNSS/INS and LiDAR for the whole trajectory are displayed in Figure 7, where the red dashed lines indicate the accepted GNSS updates. Upon first glance the LiDAR velocities follow the INS quite nicely, especially in the X component, which corresponds to the driving direction of the vehicle. However, it also becomes obvious that the LiDAR velocities are quite noisy, not taking into account the large outliers at GPS second of week (SoW) 370542. These large velocities can easily be linked to the number of clusters found at the corresponding epochs (see Figure 8), where the segmentation is not able to find more than 5 clusters. This arises the assumption of a nearly featureless environment, which means that the detected features may likely be artifacts, or at least unreliable and not well determined, too.

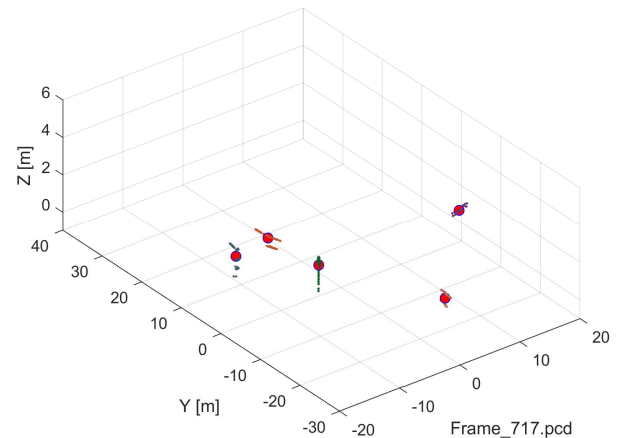


Figure 9: Example of frame with a poor feature detection

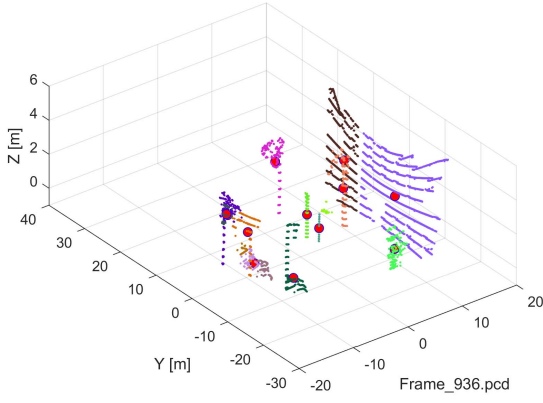


Figure 10: Example of frame rich of features and successful detection

Considering the previous point with the low redundancy such a small number of final clusters provide, such epochs can be immediately discarded for reasons of robustness. To illustrate the issues for the LiDAR when clustering, Figure 9 depicts an exemplary frame with scarce features that produced one of the large outliers in Figure 8. Figure 10 on the other hand displays an example of a frame rich on features as a representative base for a well-conditioned velocity estimation.

B. Region of interest

For a more detailed examination, we now focus on the part close to the GNSS updates at the end of the trajectory. Figure 11 depicts the velocities of the region of interest, where again the red dashed lines indicate the accepted GNSS updates. Again the LiDAR velocities show a very noisy character, yet the overall characteristics match. Moreover, after the first GNSS update after the long outage, the INS velocity in X and Y get pulled towards the LiDAR. There are two characteristic points where the INS and LiDAR increase diverging, specifically at GPS SoW 370572 and 370578. Both of these incidents relate to the completion of a very low dynamic turning maneuver, with low

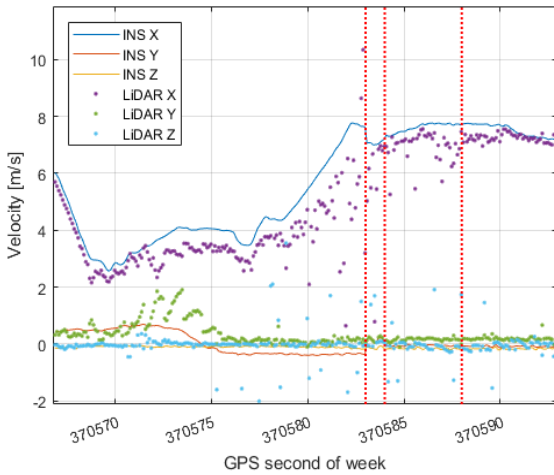


Figure 11: Velocities in vehicle body frame focused in the region of interest

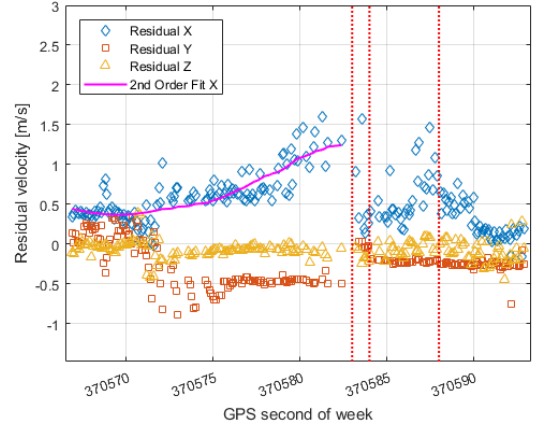


Figure 12: Residual velocities in vehicle body frame (INS-LiDAR), with 2nd order polynomial fit in X

velocity in X and very low velocity in Y. Where after the first maneuver the LiDAR Y velocity is centered around zero, the INS overshoots and assumes a constant negative velocity in Y. Especially in this case, LiDAR velocity updates would be of great benefit for the INS solution.

The differences and hard setback before and after the GNSS update become nicely visible in the residual velocities. Note for a more meaningful comparison, residuals larger than one standard deviation of the respective component were removed. The residuals are shown in Figure 12. To indicate the tendency, a second order polynomial was fitted to the residuals in X until the first GNSS update occurs. To underline the behavior of the residuals in all components, Table 2 states the root mean square error (RMSE) before and after the first GNSS update after the outage. While the RMSE in X and especially in Y becomes significantly smaller, it increases in Z. Where Y and Z seem to converge to a constant value close to zero after the update, the X component unfortunately shows a noisy peak centered at the third GNSS update after the outage before the residuals approximate zero.

TABLE 2: RMSE VELOCITIES DURING GNSS OUTAGE AND AFTER ACCEPTED GNSS UPDATES

RMSE	Velocity RMSE (INS – LiDAR)	
	Before GNSS Update	After GNSS Update
X (m/s)	0.677	0.522
Y (m/s)	0.418	0.247
Z (m/s)	0.122	0.153

V. CONCLUSION

Several conclusions for the LiDAR processing algorithm and velocity estimation quality may be drawn here, as well as for the Kalman Filter itself. First of all, the estimated LiDAR velocities nicely coincide in shape with the trajectory provided by the INS. Moreover the drift of the IMU due to the accumulated sensor errors can be detected using the LiDAR velocities, shown by the

growing difference in residuals in a long period of no GNSS updates and sudden congruence of velocities after the GNSS update. Considering the rather consistent offset between the velocities after the GNSS update, at least in Y and Z, indicates the potential need of introducing an additional bias mitigation or estimation of the LiDAR velocity with respect to the INS.

At this point, the LiDAR measurements' noise behavior is not sufficiently stable enough for them to serve as filter updates without a very sophisticated distinction between valid and invalid velocity estimates. Considering this, a simple variance based velocity update rejection in the filter similar to eq. (1) is not feasible, even when computed using vehicle body frame velocities.

The number of clusters found in a frame relates directly to the quality of the velocity estimate, and yields potential for a first outlier rejection at a very early stage. A further goal is to improve the clustering and object detection by extending the feature database. Another possibility is to filter out unwanted objects reducing in consequence the noise in the registration module as suggested in [9]. This enables the algorithm to retrieve more features per frame, reduce the probability of featureless environments and add further redundancy to the velocity estimation.

At this point the targeted "driving lane accuracy" of the INS solution using LiDAR updates for stabilization in the current implementation is not achieved. This opens the door for future investigations, such as improving the LiDAR velocity estimation at LiDAR side, and the outlier rejection, both at LiDAR and at filter side. The overall level of accuracy the LiDAR measurements provide show promise to suffice for stable filter velocity updates, yet the noise renders them unusable. Decreasing the noise in the velocity estimation will be key to selecting proper filter updates, because only then high precision sensor characteristics may be introduced to the filter and outlier statistics allow for a more narrow selection of valid updates, as well as the derivation of integrity information.

ACKNOWLEDGMENT

The results presented in this work were developed within the project "OPTimal Assistierte, hoch Automatisierte, Autonome und cooperative Fahrzeugnavigation und Lokalisation (short:

OPA3L)", funded by the German Federal Ministry for Economic Affairs and Energy (BMWi) and administered by the Project Management Agency for Aeronautics Research of the German Space Agency (DLR) in Bonn, Germany (grant no. 50NA1910). Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author and do not necessarily reflect the views of UniBwM nor DLR.

REFERENCES

- [1] T. Pany, D. Dötterböck, H. Gomez-Martinez, M. Subhan Hamed, F. Hörkner, T. Kraus, D. Maier, D. Sanchez-Morales, A. Schütz, Universität der Bundeswehr München, Neubiberg, Germany, P. Klima, D. Ebert, Axtesys GmbH, Graz, Austria, "The Multi-Sensor Navigation Analysis Tool (MuSNAT) – Architecture, LiDAR, GPU/CPU-GNSS-Signalprocessing," in *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019)*, Miami, FL, 2019.
- [2] A. Shetty and G. Xingxin Gao, "Adaptive covariance estimation of LiDAR-based positioning errors for UAVs," *NAVIGATION*, vol. 66, no. 2, pp. 463–476, 2019.
- [3] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. London: Artech House, 2013.
- [4] J. Wendel, *Integrierte Navigationssysteme: Sensordatenfusion, GPS und Inertiale Navigation*: Walter de Gruyter, 2011.
- [5] W. Wohlkinger and M. Vincze, Eds., *Ensemble of shape functions for 3d object classification*: IEEE, 2011.
- [6] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [7] D. Holz, A. E. Ichim, F. Tombari, R. B. Rusu, and S. Behnke, "Registration with the point cloud library: A modular framework for aligning in 3-D," *IEEE Robotics & Automation Magazine*, vol. 22, no. 4, pp. 110–124, 2015.
- [8] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," *Numerische mathematik*, vol. 14, no. 5, pp. 403–420, 1970.
- [9] M. Joerger, G. Duenas Arana, M. Spenko, and B. Pervan, "A New Approach to Unwanted-Object Detection in GNSS/LiDAR-Based Navigation," *Sensors*, vol. 18, no. 8, p. 2740, 2018.