Investigating the Performance of LCNS with Visual-Inertial Odometry for Lunar Rover Navigation

Johann Diep
European Space Agency
Noordwijk, Netherlands
johann.diep@esa.int

Richard Dennis Swinden
European Space Agency
Noordwijk, Netherlands
richard.swinden@esa.int

Paolo Zoccarato
European Space Agency
RHEA Group
Noordwijk, Netherlands
paolo.zoccarato@esa.int

Martin Azkarate Vecilla
European Space Agency
HE Space Operations
Noordwijk, Netherlands
martin.azkarate@esa.int

Pietro Giordano
European Space Agency
Noordwijk, Netherlands
pietro.giordano@esa.int

Javier Ventura-Traveset
European Space Agency
Toulouse, France
javier.ventura-traveset@esa.int

Paolo Crosta
European Space Agency
Noordwijk, Netherlands
paolo.crosta@esa.int

I. Introduction

Moon exploration has gained more interests in the scientific research and commercial community in recent years. The International Space Exploration Coordination Group (ISECG) identified the Moon as the first step towards a large-scale Solar System exploration [1]. The natural satellite of the Earth can act as an attractive testbed to develop the required technologies and capabilities for future manned deep space exploration.

Communication and navigation are essential building blocks for sustained lunar exploration. The steadily increasing number of missions to the Moon creates the potential need for the deployment of a dedicated lunar communication and navigation infrastructure that would support and enhance the lunar exploration. Thereby, as part of the Moonlight initiative, ESA presents its plan to develop the Lunar Communication and Navigation Services (LCNS) system, which shall support current as well as the next generation of institutional and private lunar exploration missions. The availability of such an infrastructure would enable new cost-effective mission concepts and business models and benefit lunar surface users such as as lander spacecrafts and rovers. Assuming a LCNS constellation of 3-5 satellites orbiting around the moon, a 3-sigma precision of 30m can be reached at times for the lunar landing use case.

The European Space Agency (ESA) is an European intergovernmental organisation of 22 member states dedicated to the exploration and use of space for the benefit of humankind.

The work described in this paper was jointly supported by the Commercial User Segment and Navigation System Validation (TEC-ESG), Radio Navigation Systems and Techniques (TEC-ESN) and Automation & Robotics (TEC-MMA) sections as well as the Strategy and Programme Department (NAV-P) at the European Space Research and Technology Centre (ESTEC) as part of ESA in the Netherlands.

However, for the autonomous rover use case, higher accuracy is required. Currently, as described by Geromichalos et al. [2], rovers on Mars rely heavily on visual-inertial techniques for the navigation task. In the standard visual-inertial odometry implementation [3], a body-fixed camera and an inertial measurement unit (IMU) are used to determine the relative position and orientation of the platform with respect to an initial state by processing the flow of features in sequential image frames and integrating accelerometer and gyroscope measurements respectively. Instances of visual odometry, such as Visual Simultaneous Localization and Mapping (V-SLAM) [4], which generates a globally consistent estimate of the camera trajectory as well as a map of the environment, can achieve high positioning accuracy for a medium-length trajectory. However, in contrast to satellite-based navigation, a drift which increases over the the traversed distance or time, can be observed due to the propagation of estimation errors. Under well illuminated conditions, this drift was reported to be 1-5% per travelled distance. Low illumination at the lunar poles or permanently shadowed craters on the Moon therefore pose challenging conditions for visual odometry. Moreover, distinct pixels within the current image, so-called features, are used for the processing step. Absence of features is less of an issue in dense metropolitan areas on Earth than it is in open textureless lunar landscape, which would further increase the drift. On the other hand, the behavioural pattern of satellite-based navigation[1] depends on the physical impediments as well as its constellation geometry. While it performs robustly in open environments where large number of satellites are visible, its

[1]It is noted that this behaviour applies generally for all types of satellite-based navigation, i.e. Global Navgation Satellite System (GNSS) as well as LCNS.

accuracy is degraded in urban situations due to effects such as signal obstruction, refraction, fading and reflection. Therefore, a fusion between both navigation techniques can benefit from their respective complementary nature as showcased by Afia, Escher and Macabiau [5].

In all earth-based studies, satellite-based positioning forms the basis for the navigation task, whereas visual-inertial pose data serve as addition in order to overcome its drawbacks. In contrast, for the lunar navigation case, additional LCNS measurements shall correct the accumulated drift and provide absolute, i.e. non-relative, accuracy for the deployed visual-inertial pose estimation methods. Therefore, the goal of this work is to establish a fusion between both LCNS and visual-inertial navigation techniques. The user navigation algorithm was implemented and its performance was assessed. They give insights into the achievable combined navigation performance for the lunar rover use case. A planetary rover dataset [6] consisting of raw images recorded from a monocular camera, IMU sensor measurements and GNSS Real Time Kinematic (RTK) solutions was used for the performance evaluation, whereby the RTK data were degraded or temporarily removed in order to simulate real LCNS conditions. An open source visual odometry algorithm called OpenVINS [7] was evaluated for the visual processing task. The resulting visual-inertial pose estimates were combined with simulated LCNS data using an error-state Extended Kalman Filter (EKF) [8] in a loosely-coupled approach in order to evaluate the worst case performance of the adopted sensor fusion approach. Here, the NaveGo navigation software [9] was used as a baseline framework. Multiple experiments with different parameters were conducted which will be discussed in this publication in detail.

The rest of the paper is organized as follows. Section 2 introduces the preliminaries. Section 3 highlights the data workflow and its various modules. Section 4 provides the results and their findings. Lastly, a summary of the performance analysis with a short outlook can be found in section 5.

II. Preliminaries

This section includes all the preliminaries and mathematical foundations to understand the work in this thesis. The information are kept short and there is no intended order or dependencies between the different sections. The reader can also choose to skip this chapter completely and use it only as a reference later on if required.

A. Mathematical Notations

This subsection provides few small notes on the used mathematical notations in this section in order to avoid confusions. In general, whenever a nomenclature symbol is duplicated, their respective meaning should be clear from the context. Additionally, the following notation rules are made:

- throughout this section, variables, which can be substituted with numerical values or expressions,
- as well as the denomination for coordinate frames are italicized,
- vectors are denoted with bold lowercase characters,
- matrices with bold uppercase characters,
- and discrete-time dependency is described with a subscript k.

B. Recursive Bayesian Estimation

A Bayes filter keeps track of the probability distribution of the system state $\boldsymbol{x}_k$ conditioned on all past sensor observations $\boldsymbol{z}_{1:k}$ [10]. It consists of a prior update step based on the process model (equation 1) and a posterior update step based on the measurement model (equation 2):

$$p(\boldsymbol{x}_k \mid \boldsymbol{z}_{1:k-1}) = \int \underbrace{p(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1})}_{\text{process model}} \underbrace{p(\boldsymbol{x}_{k-1} \mid \boldsymbol{z}_{1:k-1})}_{\text{previous posterior}} d\boldsymbol{x}_{k-1} \tag{1}$$

$$p(\boldsymbol{x}_k \mid \boldsymbol{z}_{1:k}) \propto \underbrace{p(\boldsymbol{z}_k \mid \boldsymbol{x}_k)}_{\text{measurement model}} \underbrace{p(\boldsymbol{x}_k \mid \boldsymbol{z}_{1:k-1})}_{\text{prior}} \tag{2}$$

Here, $\boldsymbol{z}_{1:k-1}$ and $\boldsymbol{z}_{1:k}$ describe the sensor measurements obtained up to timestep $k-1$ and k respectively. The process model is a probabilistic model of the system dynamics. The measurement model describes the likelihood of making an observation z_k at the state x_k. Both models are usually characterized using a finite set of parameters.

C. Extended Kalman Filter

The Kalman filter (KF) provides the exact solution to the Bayesian state estimation problem for a linear time-invariant system with Gaussian distributions for the process and measurement noise. In contrast, the extended Kalman filter (EKF) is a suboptimal extension of the KF to nonlinear systems [10]. It is derived by linearizing the nonlinear system equations about the latest state estimate. The following lists the recursive formulation of the discrete-time EKF with the assumption of non-additive process and measurement noise.

By initializing $\hat{\boldsymbol{s}}_0^+ = \boldsymbol{s}_0$, $\boldsymbol{P}_0^+ = \boldsymbol{P}_0$, the prior update is given as:

$$\hat{\boldsymbol{s}}_k = \boldsymbol{q}_{k-1}(\hat{\boldsymbol{s}}_{k-1}^+, \boldsymbol{u}_{k-1}, \boldsymbol{w}_{k-1} = \boldsymbol{0}) \tag{3}$$

$$\boldsymbol{P}_k = \boldsymbol{A}_{k-1}\boldsymbol{P}_{k-1}^+\boldsymbol{A}_{k-1}^\top + \boldsymbol{L}_{k-1}\boldsymbol{Q}_{k-1}\boldsymbol{L}_{k-1}^\top \tag{4}$$

Furthermore, the equations for the posterior update is written as following, where $\hat{\boldsymbol{z}}_{\boldsymbol{k}} = \boldsymbol{h}_k(\hat{\boldsymbol{s}}_k, \boldsymbol{n}_k = \boldsymbol{0})$:

$$\boldsymbol{K}_k = \boldsymbol{P}_k\boldsymbol{H}_k^\top\left[\boldsymbol{H}_k\boldsymbol{P}_k\boldsymbol{H}_k^\top + \boldsymbol{M}_k\boldsymbol{R}_k\boldsymbol{M}_k^\top\right]^{-1} \tag{5}$$

$$\hat{\boldsymbol{s}}_k^+ = \hat{\boldsymbol{s}}_k + \boldsymbol{K}_k\left[\tilde{\boldsymbol{z}}_{\boldsymbol{k}} - \hat{\boldsymbol{z}}_{\boldsymbol{k}}\right] \tag{6}$$

$$\boldsymbol{P}_k^+ = \left[\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k\right]\boldsymbol{P}_k \tag{7}$$

$\hat{\boldsymbol{s}}_k$ and $\hat{\boldsymbol{s}}_k^+$ define the prior and posterior state estimates with $\boldsymbol{P}_k$ and $\boldsymbol{P}_k^+$ as the corresponding covariance matrices. $\boldsymbol{q}_{k-1}$ and $\boldsymbol{h}_k$ define the process and measurement equations

respectively, $\boldsymbol{u}_{k-1}$ the system input and $\tilde{\boldsymbol{z}}_{\boldsymbol{k}}$ the measurement. The process $\boldsymbol{w}_{k-1}$ and measurement noise $\boldsymbol{n}_k$ are assumed to be Gaussian distributed with zero means and covariances $\boldsymbol{Q}_{k-1}$ and $\boldsymbol{R}_k$ respectively. The Jacobian matrices $\boldsymbol{A}_{k-1}$ and $\boldsymbol{H}_k$ of the model equations with respect to the state evaluated at $\hat{\boldsymbol{s}}^+_{k-1}$ and $\hat{\boldsymbol{s}}_k$ respectively linearize the system equations around the current state estimates. The matrices $\boldsymbol{L}_{k-1}$ and $\boldsymbol{M}_k$ represents the Jacobians of the prediction and update equations with respect to the process (evaluated at the prior state estimate) and measurement noises (evaluated at the posterior state estimate) respectively.

Intuitively, the mean state estimate is predicted forward and the deviation between the actual and the predicted measurement is corrected using the undisturbed non-linear formulations. In contrast, the corresponding variances are calculated using the linearized forms.

D. Error-State Kalman Filter

One drawback of the EKF is its dependency on linearization, which is a disadvantage for non-linear systems. However, one interesting aspect is that in comparison to the non-linear states, the corresponding state errors showcase a more linear behaviour. Hence, by using the Error-State Kalman Filter (ESKF) formulation and modeling the Gaussian estimates of the state errors instead, disturbances due to non-linear effects can be avoided. Moreover, the assumption of the linearization is respected since the error-state system always operates close to the origin. The interested reader is referred to the publication by Solà [11] for further explanations and the detailed derivations.

E. Visual Odometry

In camera-based navigation, the objective is to determine the motion of a camera which is rigidly mounted to a mobile vehicle. Here, one distinguishes between navigation in a known and an unknown environment [5]. Within the context of this project, only navigation in the latter case is relevant. The method is based on the principle of dead reckoning, where the camera motion is computed based on the comparison of pixels in successive images.

There are three frames that need to be defined [12]. The image frame defines the 2-dimensional captured image, the camera frame is attached to the camera and the vision frame defines the camera motion in the environment. The mathematical relation between the coordinates of image points and the positions of 3D landmarks are approximated by the pinhole model, where the camera aperture is described as a point. The geometric transformation between image plane coordinates and camera frame landmarks is given by the following equation:

$$\begin{pmatrix} \lambda u \\ \lambda v \\ \lambda \end{pmatrix} = \begin{pmatrix} f^x & 0 & c^x \\ 0 & f^y & c^y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x^c \\ y^c \\ z^c \end{pmatrix} = \boldsymbol{K} \begin{pmatrix} x^c \\ y^c \\ z^c \end{pmatrix} \tag{8}$$

Here, u and v denote the image coordinates whereas x^c, y^c and z^c describe the landmark in the camera frame. $\boldsymbol{K}$ is the calibration matrix consisting of focal lengths f^x, f^y and camera optical center c^x, c^y.

The consecutive homogeneous transformation matrix is then used in order to transform the landmark from the camera over the body-fixed to the vision frame:

$$\begin{pmatrix} x^v \\ y^v \\ z^v \\ 1 \end{pmatrix} = \left(\begin{array}{c|c} \boldsymbol{R}^{vc} & t^{vc} \\ \hline 0_{1x3} & 1 \end{array} \right) \begin{pmatrix} x^c \\ y^c \\ z^c \\ 1 \end{pmatrix} \tag{9}$$

The landmark in the vision frame is denoted by x^v, y^v and z^v.

Using a least square minimization approach, the camera calibration process can be conducted which estimates the intrinsic calibration matrix as well as the extrinsic transformation matrix parameters. Multiple open-source tools has been developed to calculate these parameters offline, such as Kalibr [13] as well as OCamCalib [14].

Having those parameters estimated, visual odometry relies on three main steps in order to estimate the relative trajectory of the camera [5]. First feature matches between two successive images need to be established. This is followed by the removal of the outlier matches. Lastly, the motion that occurs between two images can be estimated.

Feature matching relies on matching similar attributes and is usually performed using the optical flow or feature tracking method. The former case exploits all the pixel information available in the image while the latter only uses distinct image locations such as corners or edges. Since wrong matches can cause an increase in the estimation errors, those outliers need to be removed beforehand. Here, the RANSAC algorithm [15] can be applied, which is based on the epipolar geometry constraint:

$$\bar{\boldsymbol{p}}_k^\top \boldsymbol{E}_k \bar{\boldsymbol{p}}_{k-1} = 0 \tag{10}$$

Here, $\bar{\boldsymbol{p}}$ defines the normalized image coordinates and $\boldsymbol{E}_k$ describes the essential matrix. The epipolar constraint describes a geometrical condition when viewing a 3D landmark from different perspectives. At least 5 features must be correctly matched between images to calculate the essential matrix. Further matches would lead to a least square optimization and hence provide greater robustness against noise.

Having an estimate of the essential matrix, the rotational as well as translational motion can be extracted from it using the following formulation:

$$\boldsymbol{E}_k = \left[\bar{\boldsymbol{p}}_{k-1,k}\right]_\times \boldsymbol{R}_{k-1,k} \tag{11}$$

Where $\bar{\boldsymbol{p}}_{k-1,k}$ and $\boldsymbol{R}_{k-1,k}$ defines the displacement as well as the rotation between time instance $k-1$ and k respectively.

F. Visual Simultaneous Localization and Mapping

V-SLAM extends visual odometry by adding simultaneous environment reconstruction into the context. In particular, 3D feature positions are triangulated from the image points detected in two successive time instances.

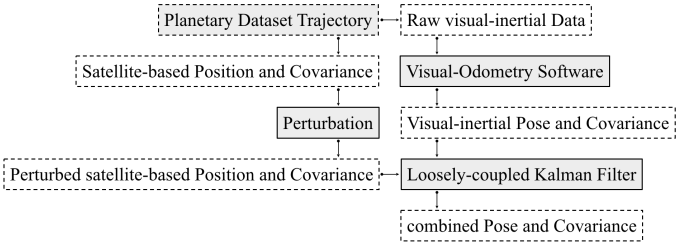

Fig. 1. Flowchart of the various modules and their corresponding interactions in order to estimate the combined poses and covariances.

One distinguish between keyframe-based and filtering-based methods [5]. Filtering-based methods estimate the pose and the landmarks that define the map as well as their respective uncertainties using a KF approach. Keyframe-based methods select some specific frames to reconstruct the 3D map of the environment based on a specific number of correspondences that should exist between two frames. An additional refinement step called bundle adjustment is used to refine the pose estimates given the 3D landmarks by minimizing the respective reprojection errors.

III. Method

The fusion algorithm was developed in Matlab using an existing software framework. This section covers the reasoning and the implementation.

A. Workflow

A graphical illustration of the workflow can be found in Fig. 1 which helps understanding the explained concepts. The source data is provided by the planetary dataset. This consists of real-time kinematic positioning (RTK) measurements of latitude, longitude and height on the WGS84 ellipsoid as well as their corresponding standard deviations and raw visual-inertial data. The former measurements are perturbed in order to simulate LCNS conditions on the Moon. The latter serve as inputs to the visual odometry software, which estimates the visual-inertial poses and covariances. These two estimate streams are then combined in a loosely-coupled fashion in order to generate the combined poses and covariances.

B. Visual-Odometry Software

Three open-source visual-odometry software were evaluated based on reliability, usability as well as the availability of documentation. The initial thought was whether to adapt an earth-based software or reuse an application developed specifically for the operation on Mars for lunar environments. Hence, the review was conducted between three different V-SLAM solutions, namely ORB-SLAM [16], OpenVINS and SPARTAN [17]. It is noted that although the chosen algorithms are classified as V-SLAMs, only the pose estimations were needed in this project.

ORB-SLAM is an open-source keyframe-based SLAM solution which is able to estimate the camera pose as well as a sparse 3D reconstruction of the environment in real-time. It

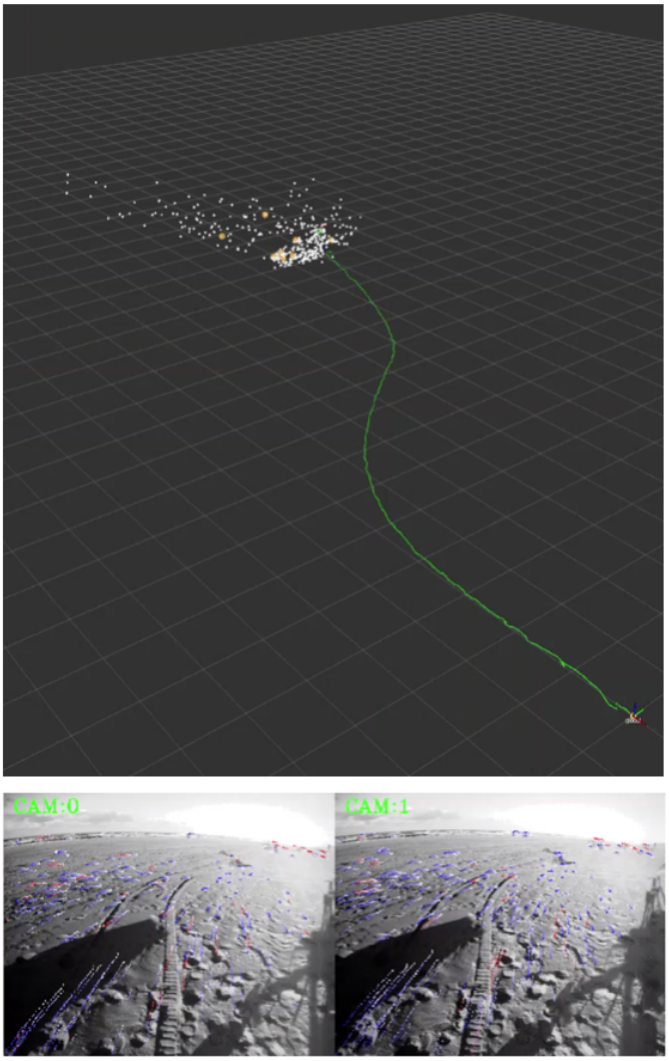

Fig. 2. Feature tracking and trajectory estimation using the Katwijk planetary dataset with OpenVINS.

consists of three interacting modules which are responsible for tracking, mapping and loop closing respectively. The tracking module is responsible for estimating the pose of the camera at each frame and deciding when to choose a new keyframe for map creation. The mapping module reconstructs the 3D map from the selected keyframes. Moreover, a bundle adjustement is performed locally to achieve a greater reconstruction accuracy in the surroundings of the camera. Lastly, the task of the loop closing module is to detect features which have already been detected and reconstructed. This step mitigates the accumulated drift, which further improves the tracking and mapping accuracy.

OpenVINS is an example for a filtering-based SLAM (see Fig. 2), in which an EKF fuses inertial measurements with sparse visual feature tracks. These visual feature tracks are fused by means of the Multi-State Constraint Kalman Filter (MSCKF) sliding window method [18].

Fig. 3. Rover used for sensor measurement collection as part of the Canadian planetary dataset (https://starslab.ca/enav-planetary-dataset/).

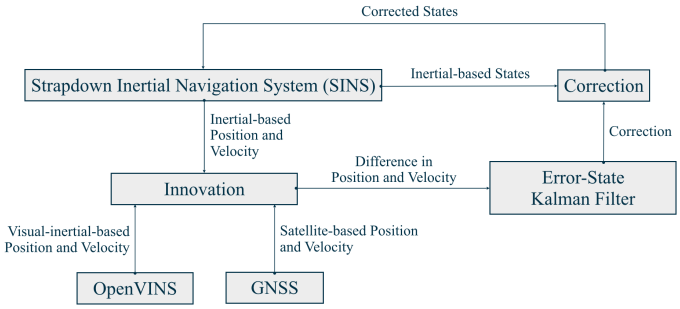

Fig. 4. Flowchart of the extended NaveGo architecture.

According to the SPARTAN documentation, its visual odometry component estimates the relative motion of the camera with respect to the starting position by examining and correlating consecutive stereo images. The algorithm selects image features and matches them in stereo image pairs in order to reconstruct their 3D coordinates and obtain the resulting point clouds. The motion between successive stereo pairs can then be estimated by fitting two 3D point clouds.

A review of the three visual odometry has led to the following conclusion. The usage of SPARTAN requires the installation of the additional Rock framework which seems inconvenient to use. Furthermore, the distinction between Earth- or Mars-based SLAM mostly impact computational cost, since the hardware of planetary rovers have some limitations compared to Earth-based use cases. In contrary, this project focuses on the achievable pose accuracy without the consideration of the required computational effort. Furthermore, ORB-SLAM only reports the pose estimations without providing a measure for their accuracy. Hence, the filtering-based OpenVINS SLAM was chosen for the visual odometry part of this project.

C. Planetary Dataset

As OpenVINS requires inertial as well as visual measurements as an input for the estimation of the state, open-source datasets were evaluated which provide the mentioned criteria. The Kalibr tool[2] was used in order to parse the datasets into the proper rosbag-format. Moreover, since this project deals with the navigation performance on lunar surfaces, the images shall represent planetary environments. Two datasets were examined in this regards.

In both datasets, the camera intrinsic parameters consisting of focal length, camera center and distortion coefficients as well as the extrinsic parameters consisting of the transformation between the camera and IMU frame are given.

The first dataset [19] corresponds to field tests conducted in the beach area of Katwijk (see Fig. 2). The tests were performed on a roving platform equipped with payload instruments which include an inertial measurement unit (IMU) as well as a stereo camera. The ground-truth is provided by RTK using a rover-mounted GNSS receiver and a base station. The roving platform traversed during a round trip along the beach through a field made of artificial boulders. These were evenly distributed in order to resemble the environments seen in Mars Reconnaissance Orbiter (MRO) images. It is noted that only during the return traverse, the sun was facing the rover which lowers the quality of the stereo images. Nonetheless, OpenVINS showcased robust performance even under the harsher conditions in those instances.

The second dataset [20] corresponds to field tests conducted at the Mars emulation terrain by the Canadian Space Agency (see Fig. 3). The rover used for data collection was a four-wheeled, skid-steered mobile robot. It is equipped with an IMU, monocular camera and a GPS receiver. Multiple long-distance runs were conducted at the terrain. This dataset was showcased to be very reliable and therefore was used for further analysis.

D. NaveGo

NaveGo provides an open-source loosely-coupled integration framework of IMU and GNSS data. In its baseline model, the strapdown inertial navigation system (SINS) module predicts the state using inertial measurements. Once a satellite measurement arrives, the correction of the initial prediction is determined through an ESKF. In this project, the baseline framework was extended to take an additional source of visual-inertial position and velocity measurements coming from OpenVINS (see Fig. 4).

E. Inertial Parameters Optimization

The SINS module of NaveGo requires the knowledge of the IMU noise parameters, which among others include angle and velocity random walk, static and dynamic biases. In theory, those are estimated using an Allan-variance analysis [21]. However, this requires a collection of static IMU measurements over a long period of time. Both datasets do not contain such a collection. Therefore, these parameters were determined using a brute-force optimization procedure, where the set of parameters which minimizes the least-square position estimation error was picked.

[2]https://github.com/ethz-asl/kalibr/wiki/bag-format

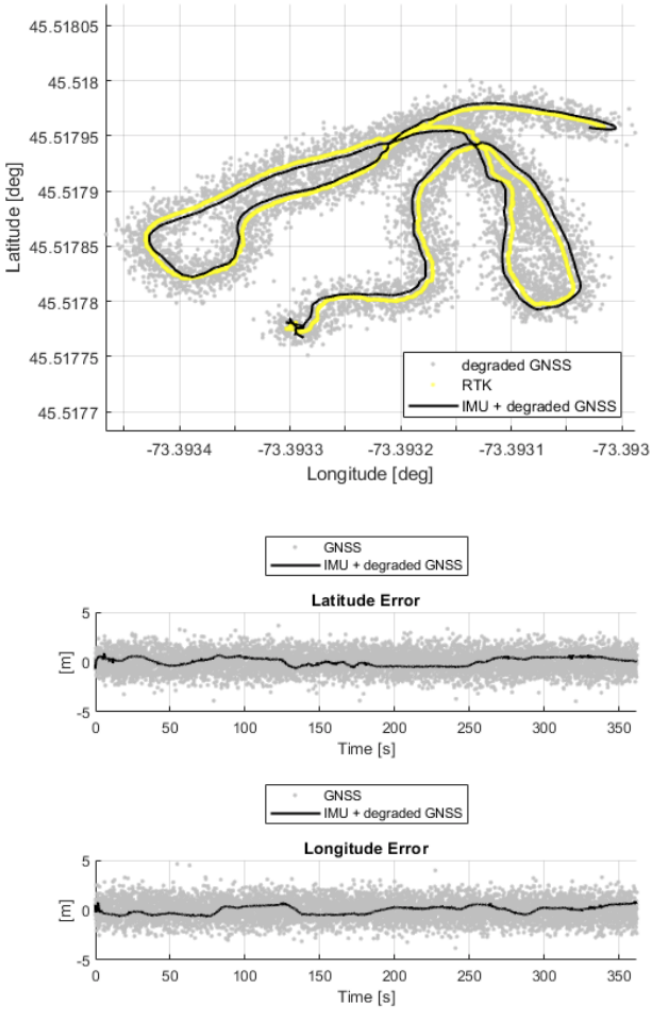

Fig. 5. Estimation using degraded GNSS and IMU. Thereby the GNSS was degraded with a standard deviation of 1m.

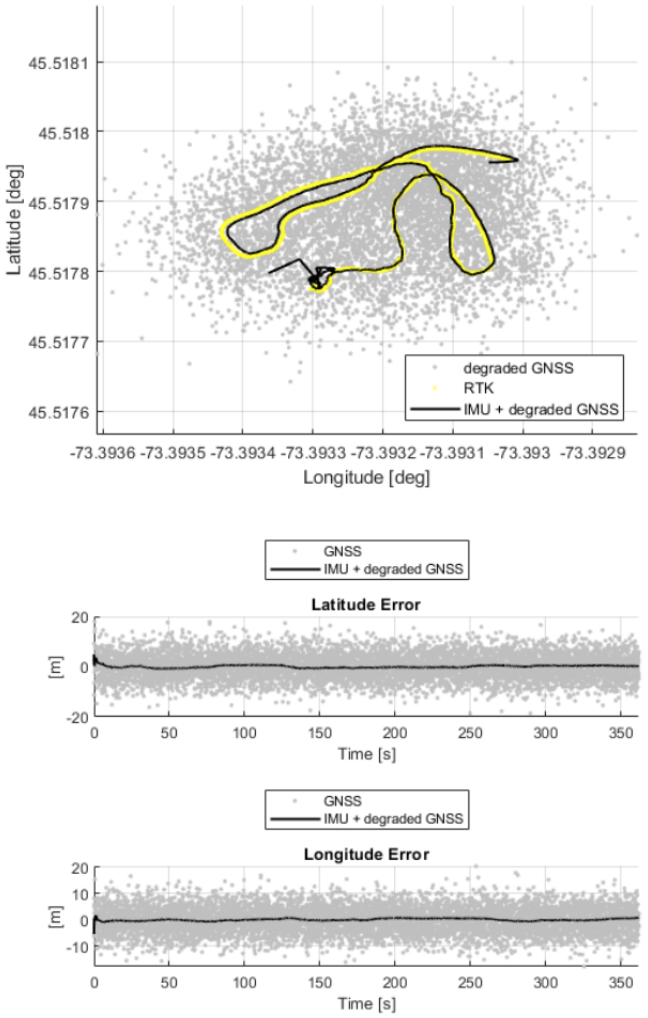

Fig. 6. Estimation using degraded GNSS and IMU. Thereby the GNSS was degraded with a standard deviation of 20m.

IV. Results

Run 3 of the Canadian planetary dataset was used for the experiments, where the rover traversed a total distance of 146 meters during a timespan of 367 seconds.

A. Baseline Performance

In a first step, the standalone fusion of IMU and degraded GNSS was evaluated. Since the measurement model is non-linear, a wrong initialization for the state and its covariance can lead to instability of the filter owing to its linearization. For this reason, the undegraded initial RTK solution with its respective standard deviation was chosen for the initial position and velocity estimate. As for the initial angular estimation, an educated guess based on direction of the rover within a short period at the beginning was made. The result for a GNSS degradation with a standard deviation of 1m can be seen in Fig. 5. Although the GNSS datapoints are dispersed, the combined version is able to estimate the ground-truth trajectory reasonably well. This holds true even with a higher GNSS degradation of 5m standard deviation as can be seen in Fig. 6.

In a further step, the combination of OpenVINS and IMU was evaluated as can be seen in Fig. 7. Since OpenVINS already uses IMU for its estimation step, it is expected that the trajectory of standalone OpenVINS and OpenVINS combined with IMU are aligned. Furthermore, the expected accumulated drift from the ground-truth which increases over the travelled distance can be observed.

B. Combined Performance

For the combined fusion of degraded GNSS with visual-inertial estimation, the same initialization as in the baseline evaluation was made for comparison purpose. The degradation of the GNSS was set to 1m standard deviation. Furthermore, the specific measurements of OpenVINS and GNSS were weighted by multiplying the respective covariance matrices

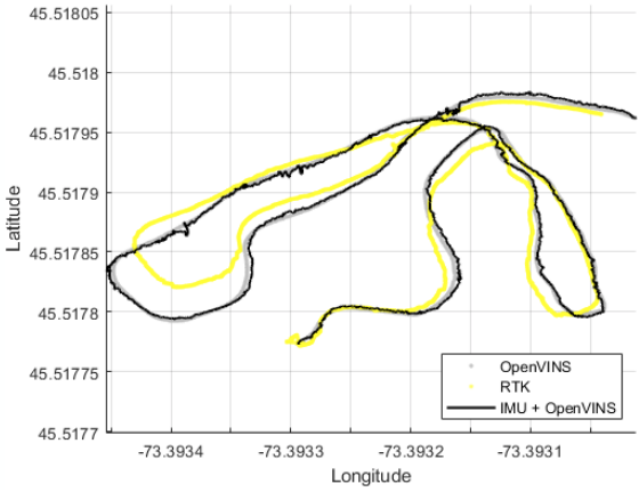

Fig. 7. Estimation using OpenVINS and IMU.

with a scale factor. Specifically, the uncertainties of the OpenVINS measurements were manually increased by a factor of 10^5 in order to lower the drift effect. The result can be seen in figure 8, which showcases the mitigation of the drift compared to the baseline performance in Fig. 7. However, in comparison with figure 5, the estimation is less smooth.

In a second evaluation, GNSS data were partially removed along a track of the trajectory, which simulates the scenario where LCNS satellites are visible due to certain satellite constellation geometry. Using the same parameters as before, OpenVINS provides a robust failsafe solution during the timespan where satellite navigation data are unavailable, which can be seen in Fig. 9. Moreover, one can also observe that the overall error reduces quickly once satellite data become available again.

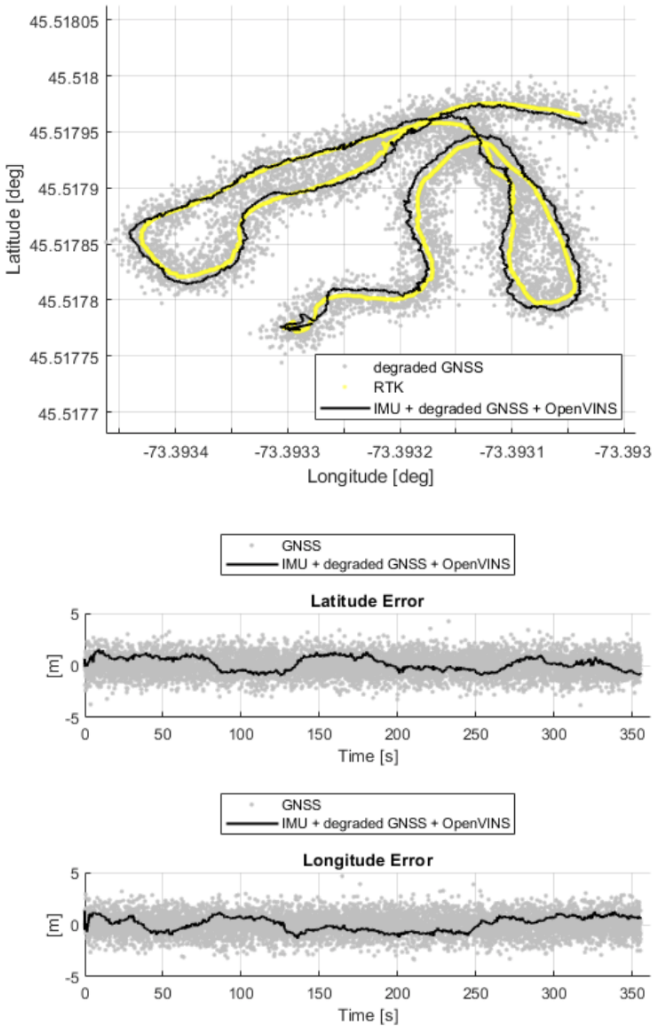

Fig. 8. Estimation using OpenVINS, IMU and GNSS. Thereby the GNSS was degraded with a standard deviation of 1m.

V. Conclusion

This publication deals with the performance investigation of LCNS with visual-inertial odometry for lunar rover navigation. This section summarises its findings and gives an outlook on future work.

A. Findings

Multiple combinations were evaluated using the provided planetary dataset, where a rover traversed a total distance of 146 meters. In an initial baseline evaluation, the combination of IMU and degraded GNSS were evaluated. Not only does this evaluation validates the correctness of the framework, it also showcases that a fusion removes the dispersion effect of the degraded GNSS data. In a further evaluation, OpenVINS was combined with IMU data which highlights the alignment of both estimations. Furthermore, OpenVINS demonstrates its drawback of an accumulated drift which increases over the traversed distance. The combined version of OpenVINS, degraded GNSS and IMU demonstrates its ability of mitigating the drift with the drawback of a less smooth trajectory estimation. The benefit of combining satellite navigation with visual-inertial odometry emerges in cases, when GNSS data are unavailable. In those scenarios, visual-inertial data provides a robust failsafe option.

B. Outlook

In this work, LCNS data were simulated by artificially degrading the provided RTK measurements. It is to be tested how the framework will perform under more realistic conditions, i.e. using a moon-projected rover trajectory in combination with simulated lunar satellite orbits. Here, the respective software tools are part of previous developments at ESA. Moreover, OpenVINS shall be tested in longer traverses as well as in more challenging environments with limited light exposure and textureless surroundings. Further work also needs to be conducted on proper tuning of the inertial parameters, which would reduce the jitter effect of the estimated trajectory. Moreover, it is to be investigated if a tight-coupling between

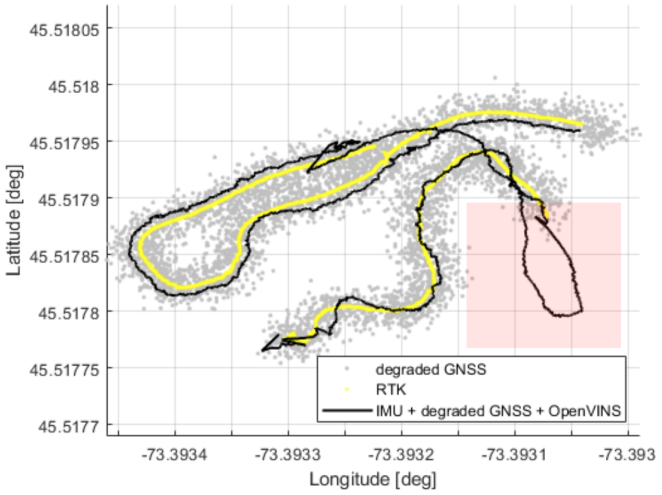

Fig. 9. Estimation using OpenVINS, IMU and GNSS. Thereby the GNSS was degraded with a standard deviation of 1m. Along the red marked track, GNSS data were made unavailable.

visual-inertial and GNSS data would potentially lead to better results.

VI. Disclaimer

This research project was conducted as part of the Young Graduate Trainee (YGT) program by the first author. In case of further inquiry, he can also be reached under his private e-mail address (johanndiep@gmail.com) as well. A special thanks goes to Antoine Grenier for helping us with the tools and analysis. The content of the present article reflects solely the authors' view and by no means represents the official ESA view.

References

[1] International Space Exploration Coordination Group, "Annual report 2020," https://www.globalspaceexploration.org/wordpress/wp-content/uploads/2021/05/ISECG-Annual-Report-2020.pdf, 2021, accessed: 2021-11-02.

[2] D. Geromichalos, M. Azkarate, E. Tsardoulias, L. Gerdes, L. Petrou, and C. Perez-del Pulgar, "Slam for autonomous planetary rovers with global localization," *Journal of Field Robotics*, vol. 37, 02 2020.

[3] R. Y. Siegwart and I. R. Nourbakhsh, *Introduction to autonomous mobile robots*. MIT, 2004.

[4] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[5] A. B. Afia, A.-C. Escher, and C. Macabiau, "A low-cost gnss/imu/visual monoslam/wss integration based on federated kalman filtering for navigation in urban environments," in *Proceedings of the 28th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2015)*, Tampa, Florida, September 2015, pp. 618–628.

[6] O. Lamarre, O. Limoyo, F. Marić, and J. Kelly, "The Canadian Planetary Emulation Terrain Energy-Aware Rover Navigation Dataset," *The International Journal of Robotics Research*, 2020. [Online]. Available: https://doi.org/10.1177/0278364920908922

[7] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "OpenVINS: A research platform for visual-inertial estimation," in *Proc. of the IEEE International Conference on Robotics and Automation*, Paris, France, 2020. [Online]. Available: https://github.com/rpng/open_vins

[8] J. Solà, "Quaternion kinematics for the error-state kalman filter," *CoRR*, vol. abs/1711.02508, 2017. [Online]. Available: http://arxiv.org/abs/1711.02508

[9] R. Gonzalez., C. A. Catania., P. Dabove., J. C. Taffernaberry., and M. Piras., "Model validation of an open-source framework for post-processing ins/gnss systems," in *Proceedings of the 3rd International Conference on Geographical Information Systems Theory, Applications and Management - GISTAM,*, INSTICC. SciTePress, 2017, pp. 201–208.

[10] R. D'Andrea, "Lecture Notes in Recursive Estimation," https://idsc.ethz.ch/education/lectures/recursive-estimation.html, 2022, accessed: 2022-02-15.

[11] J. Solà, "Quaternion kinematics for the error-state kalman filter," 2017.

[12] J. Diep, "Optimization-based Motion Blur aware Camera Pose Estimation," https://gitlab.com/jdiep/semester-thesis/-/blob/master/documentation/Report.pdf, 2018, accessed: 2022-02-15.

[13] L. Oth, P. Furgale, L. Kneip, and R. Siegwart, "Rolling shutter camera calibration," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1360–1367.

[14] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A toolbox for easily calibrating omnidirectional cameras," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 5695–5701.

[15] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, p. 381–395, jun 1981. [Online]. Available: https://doi.org/10.1145/358669.358692

[16] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: A versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, p. 1147–1163, Oct 2015. [Online]. Available: http://dx.doi.org/10.1109/TRO.2015.2463671

[17] I. Kostavelis, L. Nalpantidis, E. Boukas, M. A. Rodrigalvarez, I. Stamoulias, G. Lentaris, D. Diamantopoulos, K. Siozios, D. Soudris, and A. Gasteratos, "Spartan: Developing a vision system for future autonomous space exploration robots," *Journal of Field Robotics*, vol. 31, no. 1, pp. 107–140, 2014. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21484

[18] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 3565–3572.

[19] R. A. Hewitt, E. Boukas, M. Azkarate, M. Pagnamenta, J. A. Marshall, A. Gasteratos, and G. Visentin, "The katwijk beach planetary rover dataset," *The International Journal of Robotics Research*, vol. 37, no. 1, pp. 3–12, 2018. [Online]. Available: https://doi.org/10.1177/0278364917737153

[20] O. Lamarre, O. Limoyo, F. Marić, and J. Kelly, "The canadian planetary emulation terrain energy-aware rover navigation dataset," *The International Journal of Robotics Research*, vol. 39, no. 6, pp. 641–650, 2020. [Online]. Available: https://doi.org/10.1177/0278364920908922

[21] N. El-Sheimy, H. Hou, and X. Niu, "Analysis and modeling of inertial sensors using allan variance," *Instrumentation and Measurement, IEEE Transactions on*, vol. 57, pp. 140 – 149, 02 2008.