

Undelayed Initialization of Inverse Depth Parameterized Landmarks in UKF-SLAM with Error State Formulation

Nikolaus Ammann[§]

Laura García Mayo[†]

Abstract—In this paper we present an approach to combine error state estimation with total state monocular simultaneous localization and mapping (SLAM) in a single Unscented Kalman Filter (UKF). The map features use the inverse depth parametrization for undelayed initialization and for the ability to use low-parallax features with unknown depth information. Furthermore, a new map feature initialization method is presented using the Unscented Transform (UT). This method allows to capture all correlations between the map features and the error state variables without the necessity to calculate any Jacobian matrices.

I. INTRODUCTION

Estimating the motion of a moving object can be a challenging task, depending on the available sensor data. A widely used scheme is the combination of an inertial navigation system (INS) based on measurements of an inertial measurement unit (IMU) and an error state estimator. There, the solution of the INS is corrected using the estimated errors.

The estimation of the navigation errors is done by fusing data of possibly multiple sensors, like GNSS, altimeter, barometer, etc.. Another promising sensor for this purpose is a monocular camera, which is a projective sensor that measures the bearing of image features.

It is well known that it is possible to compute both a scene structure and a camera motion from a given image sequence of a rigid 3D scene, but with an unknown scale factor. Visual Simultaneous Localization And Mapping (SLAM) approaches that use a probabilistic filtering approach to sequentially update estimates of the positions of features and the current camera location in the total state space have been developed over the past years.

In this paper we present an approach to combine error state estimation with total state monocular SLAM in a single deeply-coupled Unscented Kalman Filter (UKF). This allows the calculation of high frequency inertial navigation solutions, which are corrected using the low frequency error estimates. Furthermore, a new map feature initialization method is presented using the Unscented Transform (UT). This method allows to capture all correlations between the map features and the error state variables without the necessity to calculate any Jacobian matrices.

[§]Nikolaus Ammann is with the German Aerospace Center (DLR), Institute of Flight Systems, 38108 Braunschweig, Germany. nikolaus.ammann@dlr.de

[†]Laura García Mayo is a graduate student in the *Joint European Master in Space Science and Technology* program.

II. INVERSE DEPTH PARAMETRIZATION

In monocular SLAM a sparse map of 3D points is built on the fly as the camera’s motion is simultaneously estimated. Solving this problem in real-time for a 6 DOF motion has been proven feasible since the work of Davison [1].

Over the years different solutions have been evaluated to tackle the SLAM problem. One of the most widely used methods is based on probabilistic Kalman filters, like the Extended Kalman Filter [2], [3] or the Unscented Kalman Filter [4]. These approaches usually use the straightforward Euclidean XYZ parametrization of map feature as described by Davison [1]. This parametrization suffers from the drawback that only features with a significant parallax during the camera motion can be used. Therefore, distant features can not be accounted for in the calculations. Additionally, features have to be reobserved with a changed parallax to initialize the representation of the feature within the map or state. This introduces a delay in the initialization of the feature. To overcome these drawbacks several authors proposed alternative initialization schemes for parametrization and initialization [5], [6], [7], [8], [9]. With the “inverse depth” parametrization of map features Civera et al. proposed a powerful alternative to the Euclidean XYZ parametrization [7], [8].

The “inverse depth” parametrization of the point \mathbf{p}_i expresses that point by a ray from the position from which the feature was first observed by the camera. Therefore, this ray is defined by the position $[x_i, y_i, z_i]^T$, the normalized directional vector \mathbf{m} , and the inverse depth ρ_i . In this parametrization scheme, a 3D point \mathbf{p}_i is represented by the vector:

$$\mathbf{p}'_i = [x_i, y_i, z_i, \theta_i, \phi_i, \rho_i]^T \in \mathbb{R}^6 \quad (1)$$

This six dimensional vector models the three dimensional point \mathbf{p}_i in space by the following equation:

$$\mathbf{p}_i = \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} + \frac{1}{\rho_i} \cdot \mathbf{m}(\theta_i, \phi_i) \quad (2)$$

Where X_i , Y_i and Z_i are the Cartesian coordinates of the three dimensional point \mathbf{p}_i .

The directional vector \mathbf{m} is encoded by the azimuth and elevation angles (θ_i & ϕ_i), which are given in the world coordinate frame:

$$\mathbf{m}(\theta_i, \phi_i) = \begin{bmatrix} \cos \phi_i \sin \theta_i \\ -\sin \phi_i \\ \cos \phi_i \cos \theta_i \end{bmatrix} \quad (3)$$

This parametrization improves the Gaussianity of the measurement equation significantly for features at all depths; even for features with potentially “infinite” depths. Additionally, the “inverse depth” parametrization allows direct, non-delayed initialization of map features and therefore, undelayed inclusion of orientation information from low-parallax features with unknown depths.

The only drawback of the inverse depth approach is the number of parameters used to express the map features. Because every map feature is encoded using six parameters instead of the three parameters used in the more usual Euclidean XYZ approach. This leads to a significant increase in the computational costs, since computational complexity for each update of an EKF or UKF scales at least with the square of the total size of the state [10].

Nevertheless, the “inverse depth” parametrization was chosen because of the non-delayed initialization of map features and the possibility to include features with potentially “infinite” depths. These benefits play an important role in possible high altitude scenarios like spacecraft navigation for lunar landing [11], [12] or High Altitude Long Endurance (HALE) aircraft [13], where the large distance to the surface results in low-parallax.

III. STATE ESTIMATION

The purpose of the developed algorithm is to provide a valid navigation solution. Since this algorithm is divided into a high rate inertial navigation part and a low rate error state estimation, there have to be defined two different state representations.

The first state is the total state \mathbf{x} , representing the navigation solution. It consists of the position $\mathbf{r}_{\mathbb{W}}$, the velocity $\mathbf{v}_{\mathbb{W}}$, and the attitude $\mathbf{q}_{\mathbb{B}}^{\mathbb{W}}$; see (4). This total state is updated by the INS with every new inertial measurement. The implementation is described in [11].

$$\mathbf{x} = \begin{bmatrix} \mathbf{r}_{\mathbb{W}} \\ \mathbf{v}_{\mathbb{W}} \\ \mathbf{q}_{\mathbb{B}}^{\mathbb{W}} \end{bmatrix} \in \mathbb{R}^{10} \quad (4)$$

Since the resulting total state \mathbf{x} is prone to errors due to defective inertial measurements and other model uncertainties, it has to be corrected to remain reliable. Therefore, the error state $\delta\mathbf{x}$, which is used to compensate the errors of the INS is estimated. The error state $\delta\mathbf{x}$ is defined as shown in (5). It consists of the position error $\delta\mathbf{r}_{\mathbb{W}}$, the velocity error $\delta\mathbf{v}_{\mathbb{W}}$, and the attitude error¹ $\delta\boldsymbol{\theta}_{\mathbb{W}}^{\mathbb{W}}$ to correct the estimated total state $\tilde{\mathbf{x}}$.

$$\delta\mathbf{x} = \begin{bmatrix} \delta\mathbf{r}_{\mathbb{W}} \\ \delta\mathbf{v}_{\mathbb{W}} \\ \delta\boldsymbol{\theta}_{\mathbb{W}}^{\mathbb{W}} \end{bmatrix} \in \mathbb{R}^9 \quad (5)$$

Additionally sensor and model errors might be included in the error state as well; as can be seen in [11]. For the sake of simplicity, these additional errors are left out of this paper.

¹The attitude error $\delta\boldsymbol{\theta}_{\mathbb{W}}^{\mathbb{W}}$ is given as a slight tilt of the computation frame \mathbb{W} . This rotation is parameterized using the axis-angle representation and can be easily converted to a rotation quaternion $\delta\mathbf{q}_{\mathbb{W}}^{\mathbb{W}}$ or a rotation matrix $\delta\mathbf{R}_{\mathbb{W}}^{\mathbb{W}}$. See [11] for more details.

For the estimation of the error state $\delta\mathbf{x}$ a UKF is used to fuse the measurements from the IMU with additional aiding sensor measurements, in such a way that the resulting variance of the error state is minimal. This kind of filter was described by Julier and Uhlmann [14], [15], [16].

The UKF has some advantages over the widely used EKF. The EKF uses the non-linear state transition and observation models for the state estimation but uses linearization to be able to apply the standard Kalman Filter equations for the covariance calculation. The UKF, on the other hand, uses a deterministic sampling technique to calculate the state and covariance using the non-linear models directly. Therefore, the UKF is able to estimate the state of non-linear systems more accurately than the EKF [14].

The process model f and the measurement model h of the error state UKF have been described in [11]. This description includes the estimated sensor and model errors as well as the measurement models for position, attitude and altitude measurements.

IV. UKF-SLAM

To include optical measurements from a monocular camera into the state estimation, the estimated state vector, the process model f and the measurement model h have to be extended to be able to solve the SLAM problem.

At first, the error state vector $\delta\mathbf{x}$ from (5) is augmented by the tracked map features \mathbf{p}'_i , which use the “inverse depth” parametrization in the total state space. This leads to an error state $\delta\mathbf{x}$, which is defined as follows:

$$\delta\mathbf{x} = \begin{bmatrix} \delta\mathbf{r}_{\mathbb{W}} \\ \delta\mathbf{v}_{\mathbb{W}} \\ \delta\boldsymbol{\theta}_{\mathbb{W}}^{\mathbb{W}} \\ \mathbf{p}'_1 \\ \vdots \\ \mathbf{p}'_n \end{bmatrix} \in \mathbb{R}^{9+6 \cdot n} \quad (6)$$

Where n equals the number of tracked map features.

For the UKF-SLAM algorithm the process model f and the measurement model h have to be extended to support updating of the tracked map features in the state and using the measured feature positions in the camera image in the correction step of the filter:

To correct the predicted error state in the UKF update step the transformed sigma points $\boldsymbol{\chi}$ are projected through the observation function $h : \mathbb{R}^{\dim(\delta\mathbf{x})+\dim(\mathbf{z}_{k+1})} \rightarrow \mathbb{R}^{\dim(\mathbf{z}_{k+1})}$. This will compute the gamma points $\boldsymbol{\gamma}_{k+1}$.

$$\boldsymbol{\gamma}_{k+1} = h(\boldsymbol{\chi}) \quad (7)$$

$$= h\left(\begin{bmatrix} \delta\mathbf{x} \\ \mathbf{v}_{k+1} \end{bmatrix}\right) \quad (8)$$

$$= h'(\delta\mathbf{x}) + \mathbf{v}_{k+1} \quad (9)$$

Similar to the sigma points, the gamma points have been constructed in a way that each point can be divided into different parts. The first part corresponds to the actual error state vector $\delta\mathbf{x}$ and the second part corresponds to the measurement noise \mathbf{v}_{k+1} . Equations (7) to (9) show how each

part is handled. The state vector part is projected through the actual observation function h' , whereas the measurement noise part is added to the predicted observation.

As mentioned the measurement models for position $h'_{r_{\mathfrak{W}}}$, attitude $h'_{\mathfrak{A}}$ and altitude h'_{alt} have already been described in [11]. Therefore only the measurement model for the feature observations in the camera images h'_c is presented here.

For every visible feature in the camera image with index i , which is also tracked in the state vector, the following calculations are performed to get the predicted image coordinate of that feature. At first, the tracked map feature \mathbf{p}_i is transformed from the world coordinate frame \mathfrak{W} to the camera frame \mathfrak{C} . Where $\mathbf{p}_i \in \mathbb{R}^3$ depends on the ‘‘inverse depth’’ parametrization of that map feature $\mathbf{p}'_i \in \mathbb{R}^6$ in the state vector.

$$h''_c(\boldsymbol{\chi}) = \mathbf{R}_{\mathfrak{W}}^{\mathfrak{C}} \cdot (\mathbf{p}_i - (\tilde{\mathbf{r}}_{\mathfrak{W}} + \delta \mathbf{r}_{\mathfrak{W}})) \quad (10)$$

Where $\tilde{\mathbf{r}}_{\mathfrak{W}}$ is the position from the total state \mathbf{x} calculated by the strapdown algorithm, $\delta \mathbf{r}_{\mathfrak{W}}$ is the position error included in the error state $\delta \mathbf{x}$ or in the sigma points $\boldsymbol{\chi}$ resp., and $\mathbf{R}_{\mathfrak{W}}^{\mathfrak{C}}$ is the rotation matrix from the world coordinate frame \mathfrak{W} to the camera coordinate frame \mathfrak{C} . It is the product of the constant rotation matrix $\mathbf{R}_{\mathfrak{B}}^{\mathfrak{C}}$ from the body frame \mathfrak{B} to the camera frame \mathfrak{C} and the inverse of the attitude $\mathbf{R}_{\mathfrak{B}}^{\mathfrak{W}} = \delta \mathbf{R}_{\mathfrak{B}}^{\mathfrak{W}} \cdot \tilde{\mathbf{R}}_{\mathfrak{B}}^{\mathfrak{W}}$.

Finally, the image coordinates are calculated from the transformed map feature using the following equation:

$$h'_c(\boldsymbol{\chi}) = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} u_0 - \frac{f}{(h''_c(\boldsymbol{\chi}))_z} \cdot (h''_c(\boldsymbol{\chi}))_x \\ v_0 - \frac{f}{(h''_c(\boldsymbol{\chi}))_z} \cdot (h''_c(\boldsymbol{\chi}))_y \end{bmatrix} \quad (11)$$

Where (u_0, v_0) is the principal point of the camera, and f is the focal length.

Depending on which kind of measurements are available at time k , the dimensions of the measurement vector \mathbf{z}_k , the measurement noise \mathbf{v}_k as well as the observation function h itself are adjusted accordingly.

V. LANDMARK INITIALIZATION

As described in the last section, the measurement for each visible feature in the camera image, which is also tracked in the state vector, is predicted using the measurement model h'_c . But before a feature can be tracked in the state vector, each feature has to be initialized. This includes the augmentation of the error state $\delta \mathbf{x}$ by the six parameters per feature like described in (1), as well as the augmentation of the corresponding covariance matrix \mathbf{P} .

Several papers deal with the problem of undelayed map feature initialization for the monocular SLAM problem [5], [7], [8], [17], [18]. All these publications share one common aspect: All estimated state variables are defined in the total state space, which is the same for the map features. In this paper however, the estimated state includes variables defined in the error state space along with the map features, which are defined in the total state space.

Therefore, we propose a new method for initializing new features. The proposed initialization method is based on the Unscented Transform (UT) and uses the full state vector and covariance matrix to capture all possible correlations between the error state variables and the total state map features.

The calculation is divided into four steps; similar to the augmented Unscented Transform described in [19]:

- 1) The estimated error state $\delta \mathbf{x}$ and covariance \mathbf{P} are augmented with the image coordinates $[u_i, v_i]^T \in \mathbb{R}^2$ of every new feature with index i which should be included into the state vector. Additionally, an initial guess for the inverse depth $\rho \in \mathbb{R}$ for all new features observed in the current image is added to the augmented state as well. See Civera et al. [7] for details on how to choose ρ .

$$\delta \mathbf{x}^a = \left[\delta \mathbf{x}^T, [u_1, v_1]^T, \dots, [u_n, v_n]^T, \rho \right]^T$$

$$\mathbf{P}^a = \begin{bmatrix} \mathbf{P} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{uv} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R}_\rho \end{bmatrix}$$

- 2) A set of $2L + 1$ sigma points is derived from the augmented state $\delta \mathbf{x}^a$ and covariance \mathbf{P}^a , where $L = \dim(\delta \mathbf{x}) + 2 \cdot n + 1$ is the dimension of the augmented state. With $\left[\sqrt{(L + \lambda) \mathbf{P}^a} \right]_j$ is the j th column of the matrix square root² of $(L + \lambda) \mathbf{P}^a$ the sigma points $\boldsymbol{\chi}^j$, with $j \in [0, 2L]$ are defined as follows:

$$\boldsymbol{\chi}^0 = \delta \mathbf{x}^a$$

$$\boldsymbol{\chi}^j = \delta \mathbf{x}^a + \left[\sqrt{(L + \lambda) \mathbf{P}^a} \right]_j, \quad j \in [1, L]$$

$$\boldsymbol{\chi}^{j+L} = \delta \mathbf{x}^a - \left[\sqrt{(L + \lambda) \mathbf{P}^a} \right]_j, \quad j \in [1, L]$$

- 3) The sigma points $\boldsymbol{\chi}^j$ are propagated through the initialization function $m : \mathbb{R}^L \rightarrow \mathbb{R}^{\dim(\delta \mathbf{x}) + 6 \cdot n}$.

$$\boldsymbol{\gamma}^j = m(\boldsymbol{\chi}^j), \quad j \in [0, 2L]$$

This function is defined as follows:

$$m(\delta \mathbf{x}^a) := \left[\delta \mathbf{x}^T, \mathbf{p}'_1^T, \dots, \mathbf{p}'_n^T \right]^T \quad (12)$$

Where \mathbf{p}'_i is calculated from $\delta \mathbf{x}^a$ like:

$$\mathbf{p}'_i = \begin{bmatrix} \tilde{\mathbf{r}}_{\mathfrak{W}} + \delta \mathbf{r}_{\mathfrak{W}} \\ \text{atan2}((z_{\mathfrak{W},i})_x, (z_{\mathfrak{W},i})_z) \\ \text{atan2} \left(-(z_{\mathfrak{W},i})_y, \sqrt{(z_{\mathfrak{W},i})_x^2 + (z_{\mathfrak{W},i})_z^2} \right) \\ \rho \end{bmatrix} \quad (13)$$

With $z_{\mathfrak{W},i} = \mathbf{R}_{\mathfrak{C}}^{\mathfrak{W}} \cdot [u_i, v_i, 1]^T$ and $\mathbf{R}_{\mathfrak{C}}^{\mathfrak{W}}$ being the rotation from the camera frame \mathfrak{C} to the world frame \mathfrak{W} .

²The matrix square root \mathbf{A} of matrix \mathbf{B} satisfies $\mathbf{B} \triangleq \mathbf{A} \mathbf{A}^T$. This could be calculated using the numerically efficient and stable Cholesky decomposition.

- 4) The gamma points γ^j are weighted and recombined to calculate the new state $\delta\mathbf{x}$ and covariance \mathbf{P} including the new initialized map features.

$$\delta\mathbf{x} = \sum_{i=0}^{2L} W_s^i \gamma^i$$

$$\mathbf{P} = \sum_{i=0}^{2L} W_c^i [\gamma^i - \delta\mathbf{x}] [\gamma^i - \delta\mathbf{x}]^T$$

Where the weights are given by:

$$W_s^0 = \frac{\lambda}{L + \lambda}$$

$$W_c^0 = \frac{\lambda}{L + \lambda} + (1 - \alpha^2 + \beta)$$

$$W_s^j = W_c^j = \frac{1}{2(L + \lambda)}$$

with $\lambda = \alpha^2(L + \kappa) - L$.

The parameters α , β , and κ control the spread of the sigma points [10].

The new state $\delta\mathbf{x}$ now includes the newly initialized map features and the covariance matrix \mathbf{P} has the correct correlation between the error state space variables and the total state space map features.

This method can also be used for the normal cases described in [5], [7], where both, the state variables and the map features are defined in the total state space.

VI. RESULTS

To show the functionality of the proposed initialization method and the complete UKF-SLAM algorithm an open-loop Matlab/Simulink-based simulation environment has been used. It simulates a landing trajectory of a spacecraft from the powered descent initiation maneuver to landing. The trajectory begins in approximately 11 km above ground with a flight speed of $1600 \frac{m}{s}$. After a flight time of around 52 min the spacecraft touches down at the final landing site.

From this trajectory, sensor data for an IMU, a star tracker, and an absolute positioning sensor are generated. In contrast to [20], the optical features used for the SLAM algorithm are projections of randomly distributed points on the surface of an ellipsoid representing the Moon. This allows for an direct evaluation of the mapping. For the projection the pinhole camera model is used, with the camera pointing downwards.

While measurements from the IMU and the star tracker are available during the whole simulation run, absolute position measurements are only available at some stages of the first half of the trajectory and shortly prior to the landing. This is due to the availability of a valid crater catalog for the entire surface as well as for all altitudes. Therefore the SLAM algorithm is used to compensate the drift of the navigation solution during the phases without absolute position measurements.

A simulation run generates ground truth data for the position, velocity, and attitude of the spacecraft as well as the true position of the randomly chosen points used

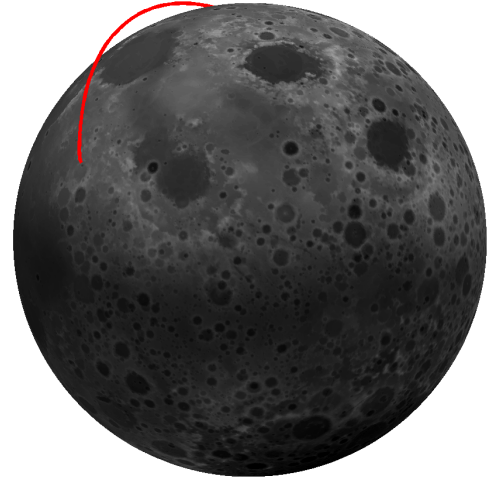


Fig. 1. Partial plot of the simulated landing trajectory. The trajectory is visible from the powered descent initiation maneuver to the touchdown at the landing site.

as optical features. Furthermore, the estimated state vector including the position, velocity, attitude, and the mapped optical features is logged along with its covariance.

Fig. 1 shows a partial plot of the simulated landing trajectory. The trajectory is visible from the powered descent initiation maneuver to the touchdown at the landing site. Due to the order of magnitude of the position the difference between the ground truth and estimate is not visible at this scale. Therefore, additional error plots give a more detailed picture of the performance.

The resulting position error of one simulation run is given in Fig. 2. The plot shows the position error during the last 32 min of the landing trajectory. The error is plotted for the x , y , and z axes in the colors red, green, and blue respectively. The estimated sigma bound of the error is plotted with dashed lines in the corresponding colors.

One can clearly see the corrections of the sensor fusion as a result of the available absolute position measurements in the time from 1200 s to 2200 s. Each correction results in a decrease of the sigma bound and a sudden change of the position error. Also shortly prior to the landing in the time from 3050 s to 3150 s absolute position updates can be observed. Between these absolute position updates a slight drift of the position error can be seen. The resulting position error is in a range of ± 600 m at the beginning of the plot and converges to less than 50 m at the end. Also the sigma bound increases during these phases.

Compared to previous SLAM implementations (see [20]) the observed drift has been decreased significantly with the presented UKF-SLAM implementation. Furthermore, the uncertainties have been integrated properly in this approach

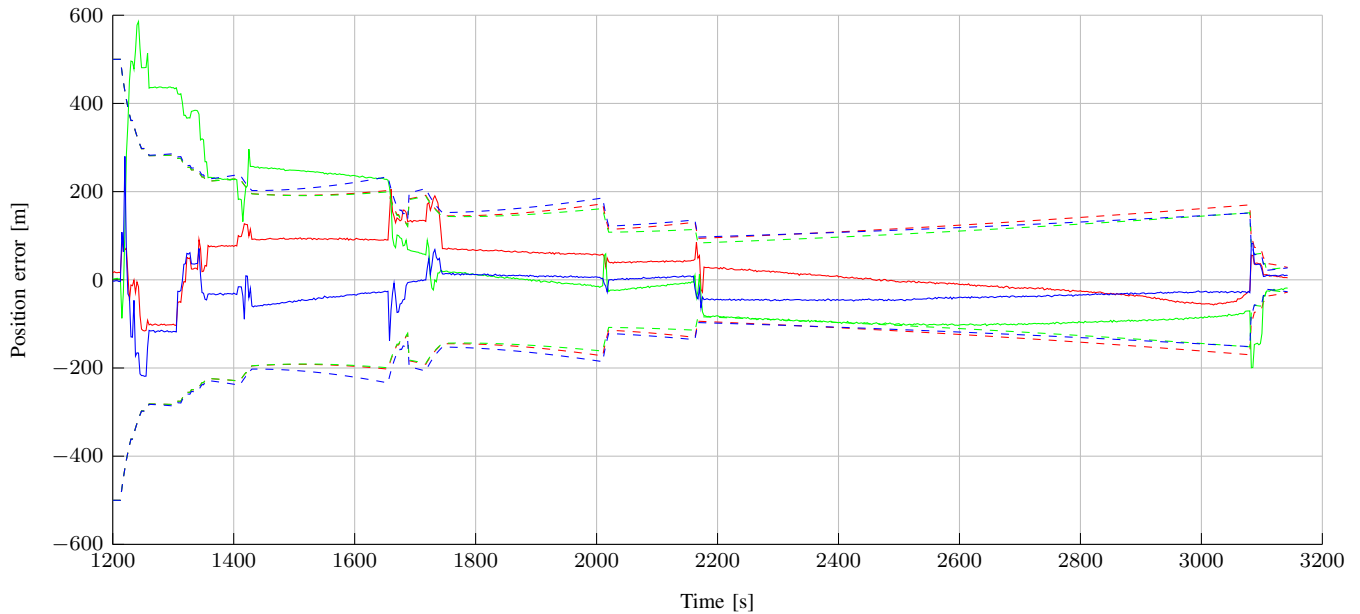


Fig. 2. Plot of the position error during the last 32 min. The error is plotted for the x , y , and z axes in the colors red, green, and blue respectively. The estimated sigma bound of the error is plotted with dashed lines in the corresponding colors.

resulting in a more precise estimation of the variance.

To evaluate the mapping of this UKF-SLAM implementation a snapshot of the estimated map is presented in Fig. 3. The plot shows the current position of the spacecraft marked in black, the true position of the features marked in pink, and the estimated feature positions marked in green with the spread of the standard deviation indicated by the blue dots. The three dimensional position of the features is calculated using (2).

The difference between freshly initialized features or features with low-parallax and features with a converged inverse depth can be seen. Also it can be seen that the estimated features converge to the true feature positions as the inverse depth converges.

VII. CONCLUSION

In this paper we presented a way how to augment an error state vector with total state space map features. The adjustments to the measurement model of the used UKF have been documented as well as a new initialization method used for the integration of new map features to the state vector. This method is based on the UT and has some advantages over other approaches:

- It is easy to implement, since no multi-hypotheses and not even Jacobian matrices have to be calculated.
- The complete state including position, attitude, and other possible sensor errors are included in the calculation of the new map features and their covariance.
- The correlation between the total state map features and the error state variables can be easily determined.

The results have shown proper functionality of the initialization and the complete UKF-SLAM algorithm. Also the inverse depth parametrization has been proven to be a good

fit for such high altitude scenarios as the landing trajectory of a spacecraft.

Further evaluation of this UKF-SLAM algorithm should be performed based on realistic optical data. Such evaluations have already been performed with a decoupled SLAM algorithm earlier in the project. These tests have been conducted using a simulation of a lunar landing scenario with realistic image rendering [20] as well as offline and online realtime processing of flight test data from an unmanned aerial vehicle (UAV) in a simulated lunar environment [11] and [12]. These steps should be repeated using this improved and coupled UKF-SLAM algorithm. Also other scenarios like autonomous driving or indoor navigation would be interesting testing scenarios.

REFERENCES

- [1] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Proceedings 9th IEEE International Conference on Computer Vision*, Oct 2003, pp. 1403–1410 vol.2.
- [2] S. Weiss and R. Siegwart, "Real-time metric state estimation for modular vision-inertial systems," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 4531–4537.
- [3] A. Barrau and S. Bonnabel, "Invariant filtering for pose ekf-slam aided by an imu," in *2015 54th IEEE Conference on Decision and Control (CDC)*, Dec 2015, pp. 2133–2138.
- [4] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "On the complexity and consistency of ukf-based slam," in *2009 IEEE International Conference on Robotics and Automation*, May 2009, pp. 4401–4408.
- [5] J. Sola, A. Monin, M. Devy, and T. Lemaire, "Undelayed initialization in bearing only slam," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Aug 2005, pp. 2499–2504.
- [6] E. Eade and T. Drummond, "Scalable monocular slam," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 1, June 2006, pp. 469–476.
- [7] J. Civera, A. J. Davison, and J. M. M. Montiel, "Unified inverse depth parametrization for monocular slam," in *In Proceedings of Robotics: Science and Systems*, 2006.
- [8] —, "Inverse depth parametrization for monocular slam," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 932–945, Oct 2008.

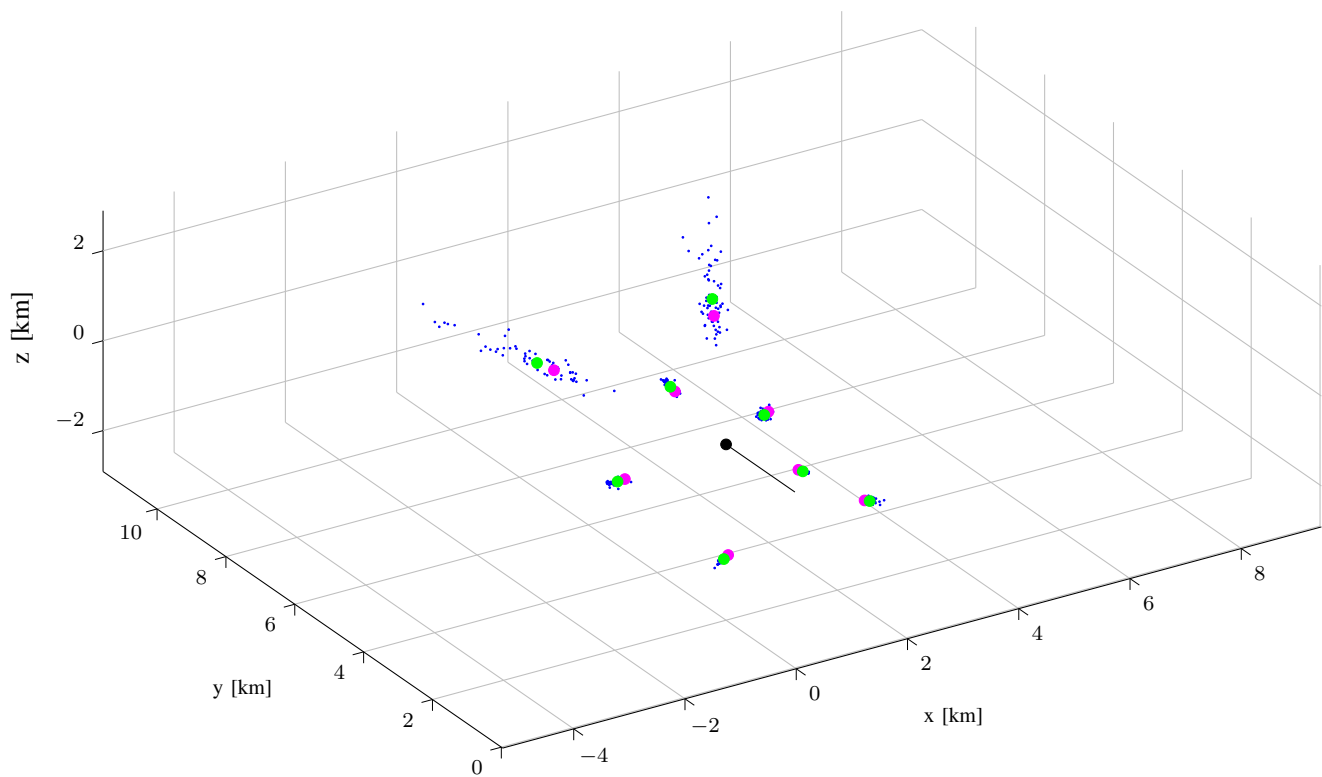


Fig. 3. Plot showing the current position of the spacecraft marked in black, the true position of the features marked in pink, and the estimated feature positions marked in green with the spread of the standard deviation indicated by the blue dots.

- [9] N. Trawny and S. I. Roumeliotis, "A unified framework for nearby and distant landmarks in bearing-only slam," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, May 2006, pp. 1923–1929.
- [10] E. A. Wan and R. V. D. Merwe, "The unscented kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, 2000, pp. 153–158.
- [11] N. Ammann and F. Andert, "Visual navigation for autonomous, precise and safe landing on celestial bodies using unscented kalman filtering," in *2017 IEEE Aerospace Conference*, March 2017, pp. 1–12.
- [12] S. Theil, N. Ammann, F. Andert, T. Franz, H. Krüger, H. Lehner, M. Lingenauber, D. Lüdtkke, B. Maass, C. Paproth, and J. Wohlfeil, "ATON (Autonomous Terrain-based Optical Navigation) for exploration missions: recent flight test results," *CEAS Space Journal*, Mar 2018.
- [13] G. Romeo, G. Frulla, and E. Cestino, "Design of a high-altitude long-endurance solar-powered unmanned air vehicle for multi-payload and operations," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 221, no. 2, pp. 199–216, 2007.
- [14] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *American Control Conference*, vol. 3, Jun. 1995, pp. 1628–1632.
- [15] S. J. Julier and J. K. Uhlmann, "New extension of the kalman filter to nonlinear systems," in *Proceedings of the 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, 1997, pp. 182–193.
- [16] S. Julier, J. Uhlmann, and D. H.F., "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Transactions on Automatic Control*, vol. 45, no. 3, pp. 477–482, 2000.
- [17] C. H. Hsiao and C. C. Wang, "Achieving undelayed initialization in monocular slam with generalized objects using velocity estimate-based classification," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 4060–4066.
- [18] K. Choi, J. Park, Y.-H. Kim, and H.-K. Lee, "Monocular slam with undelayed initialization for an indoor robot," *Robotics and Autonomous Systems*, vol. 60, no. 6, pp. 841 – 851, 2012.
- [19] Y. Wu, D. Hu, M. Wu, and X. Hu, "Unscented kalman filtering for additive noise case: augmented versus nonaugmented," *IEEE Signal Processing Letters*, vol. 12, no. 5, pp. 357–360, May 2005.
- [20] F. Andert, N. Ammann, and B. Maass, "Lidar-aided camera feature tracking and visual slam for spacecraft low-orbit navigation and planetary landing," in *Advances in Aerospace Guidance, Navigation and Control*, J. Bordeneuve-Guibé, A. Drouin, and C. Roos, Eds. Cham: Springer International Publishing, 2015, pp. 605–623.