Relative Optical Navigation for a Lunar Lander Mission

M. J. Verveld

Abstract This work explores the problem of providing relative velocity navigation for an autonomous precision landing approach on the moon without the use of telemetry or known points of support. An error-state Unscented Kalman Filter for the fusion of inertial and optical imaging sensors is presented. These sensors include a star tracker, a monocular surface camera and a laser altimeter. The filter estimates position, velocity and attitude, which, together with an initial position based on crater matching, allows for trajectory following to the surface. A main difficulty is the scale ambiguity in optical flow. The laser altimeter has been included to resolve this ambiguity and allow for velocity and altitude estimation. The scenario of a lunar landing from parking orbit was chosen to test and verify the developed navigation method in simulation using a high resolution surface model of the moon.

1 Introduction

The goals of future space exploration missions include investigation of local surface phenomena on moons, planets and asteroids as well as the building and support of outposts. Autonomous, precise and safe landings near hazardous terrain are key requirements for such missions.[17] These requirements call for a complex navigation system capable of providing accurate state estimation independent of supporting telemetry. Optical imaging sensors may form an important component in such a system as they allow low-latency measurements to be taken independently from Earth, enabling autonomy. Integrating optical measurements into the navigation system provides position and attitude determination with respect to the target body allowing for a precise autonomous landing.
Several past and present projects have investigated the problem of autonomous planetary precision landing. The ESA commissioned Navigation for Planetary Ap-

M. J. Verveld
DLR, Lilienthalplatz 7, 38108 Braunschweig, Germany, Mark.Verveld@dlr.de

proach & Landing (NPAL)[3, 15] project focused on guidance and navigation algorithms based on the tracking of unknown landmarks as well as navigation camera hardware design and the PANGU planetary surface generator. An Extended Kalman Filter (EKF) was used for state estimation. The following VisNav[16] project extended NPAL developed vision based navigation schemes for use in a broader scope of planetary navigation. The NASA commissioned Autonomous Precision Landing and Hazard Avoidance Technology (ALHAT)[1, 21] project develops precision navigation and hazard detection and avoidance for planetary landing. The goals include landing ability without surface illumination, thus requiring active sensors. Development is directed at a flash LIDAR for this purpose. Map matching is also mentioned. Johnson et al.[4, 14] match descent images to a map in which features have a known 3D position using SIFT keypoints. They combine this with persistent and image-to-image feature tracking and fuse the optical data with inertial measurements in an EKF. S. Li et al.[11] combine feature tracking with a LIDAR and apply it to navigation during the landing on asteroids. They match features found in the image of a camera with distances found using the LIDAR to enable 6 degree-of-freedom relative position estimation. In a second paper[12] by the same authors a navigation scheme for planetary landing is discussed on the basis of an EKF which fuses an IMU and the optical flow from a single camera without known support points. It shows a significant reduction in the position error growth rate compared to inertial navigation alone.

The Autonomous Terrain based Optical Navigation for landers (ATON) project, part of which is the work presented in this paper, develops the optical navigation and obstacle avoidance technology to satisfy planetary precision landing requirements. The scenario studied in the project is that of a lunar landing from a 100 km circular parking orbit. The sensors available for this task are a six degree of freedom inertial measurement unit (IMU), a star tracker for inertial attitude determination, a surface camera and a laser altimeter. There are three modes of navigation employed, which may be combined depending upon the available visual input.

Initially, navigation uses the surface camera together with a database of known surface features (craters in the case of the moon), the star tracker and IMU. This allows for direct position and attitude determination in a moon fixed frame of reference. As the lander gets closer to the surface fewer features in the camera field of view may be matched to the database. Therefore a relative velocity navigation mode is included which uses the optical flow (OF) from unknown surface features to estimate the vehicle velocity with respect to the lunar surface and integrate this to yield the position. Finally, as the lander gets the intended landing area in sight, it will navigate relative to a hazard free landing spot. This work presents the relative velocity navigation mode and assumes navigation based solely on this mode.

The navigation filter uses the Unscented Kalman Filter (UKF) algorithm at it's core. We have chosen the UKF for its reported[5, 6, 10, 22] accuracy benefits over the EKF in problems with strong nonlinearities in their observation models. The delays associated with especially image processing required for both the star tracker and surface camera have to be accounted for in the navigation filter. Using an indirect or error state approach simplifies this process.[18,20] We have combined both concepts into the error-state UKF (eUKF).

The paper is composed as follows. Section 2 explains the sensor concept and develops the equations modeling sensor behavior. Section 3 describes the navigation filter and its eUKF equations. A significant part of the ATON project has been the development of a comprehensive lunar landing simulation environment including accurate sensor emulation and detailed lunar surface rendering using ray tracing and specular lighting techniques. The surface digital elevation map (DEM) for the visual simulation uses data from the Kaguya mission.[8] This simulation environment used for testing and validation of the presented method will be further explained in section 4. The results obtained in the simulation will be presented and discussed in section 5. Finally, section 6 draws overall conclusions about the presented navigation method and gives recommendations for further work.

2 Modeling

This section provides an overview of the spacecraft equations of motion and the observation models used for the sensor fusion. The navigation filter as presented in section 3 uses error states in its prediction step so the equations developed in this section will then be cast into their error state formulation.

2.1 Kinematics

The vehicle navigation state $\mathbf{x}$ is expressed in Cartesian moon-fixed, moon-centered coordinates denoted by subscript m. It consists of the velocity $\mathbf{v}_m$, the position $\mathbf{r}_m$, the lander attitude quaternion q_m, the bias of the accelerometers $\mathbf{b}_a$ and the bias of the gyroscopes $\mathbf{b}_g$. The IMU, delivering a specific force vector $\mathbf{a}$ and a rotational rate vector ω, is used in the filter's prediction step to propagate this state vector. The subscript *imu* denotes the measured values, whereas the subscript b denotes the true values in the body-fixed frame of reference. We employ a stochastic model due to [2] using Gaussian white noise processes designated by the $\mathbf{n}$ terms in $\mathbf{y}$ as

$$\mathbf{y} = \begin{bmatrix} \mathbf{a}_{imu} \\ \omega_{imu} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_b + \mathbf{b}_a + \mathbf{n}_{a_1} \\ \omega_b + \mathbf{b}_g + \mathbf{n}_{g_1} \end{bmatrix}, \qquad \begin{bmatrix} \dot{\mathbf{b}}_a \\ \dot{\mathbf{b}}_g \end{bmatrix} = \begin{bmatrix} \mathbf{n}_{a_2} \\ \mathbf{n}_{g_2} \end{bmatrix}. \tag{1}$$

A kinematic system of continuous time differential equations describes the state derivative w.r.t. time as a function of the state itself and the IMU outputs, where $w(t)$ collects all additive Gaussian white noise terms:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{\mathbf{v}}_m \\ \dot{\mathbf{r}}_m \\ \dot{q}_m \\ \dot{\mathbf{b}}_a \\ \dot{\mathbf{b}}_g \end{bmatrix} = f(\mathbf{x}(t), \mathbf{y}(t)) + w(t) = \begin{bmatrix} R_{b\to m}\mathbf{a}_b + \mathbf{g}_m \\ \mathbf{v}_m \\ \frac{1}{2}B(q_m)\,\omega_b \\ 0 \\ 0 \end{bmatrix} + w(t) \ . \qquad (2)$$

The vector $\mathbf{g}_m$ is the gravitational acceleration at position $\mathbf{r}_m$ above the lunar surface. This may be expressed using Newton's law of gravitation as

$$\mathbf{g}_m = -\mathbf{r}_m \frac{GM}{\|\mathbf{r}_m\|_2^3} \ ,$$

where GM is the gravitational parameter of the moon in this case. The 4 × 3 matrix $B(q_m)$ is the quaternion derivative matrix[19] and $R_{b\to m}$ is the rotation matrix[19] from body-fixed to moon-fixed frame.
Since the Kalman filter will be computing in discrete time, a state transition function

$$\mathbf{x}(k+1) = \phi(\mathbf{x}(k), \mathbf{y}(k)) + w(k) \ , \quad w(k) \sim \mathcal{N}(\mathbf{0}, Q_k) \qquad (3)$$

may be derived from Eq. (2) by applying Euler discretization. The estimated state is computed using the same ϕ:

$$\hat{\mathbf{x}}(k+1) = \phi(\mathbf{x}(k), \mathbf{y}(k)) \qquad (4)$$

2.2 Observation Models

The optical sensors used for navigation include a surface camera with 40° field of view and 1024 × 1024 pixel resolution, an optical star tracker with an accuracy in the order of 10 arcsec and optionally a laser altimeter with 5 m standard deviation. The configuration is shown in Fig. 1. In order to fuse these optical sensors with the IMU, they have to be modeled by equations describing their outputs as a function of time, the state vector $\mathbf{x}$ and the IMU outputs $\mathbf{y}$.

The star tracker delivers the attitude quaternion of the spacecraft in the International Celestial Reference Frame (ICRF). This may be computed using time, the lunar ephemeris and $\mathbf{x}$.
Images from the surface camera are processed using a feature tracker based on the Lucas-Kanade[13] algorithm. It yields a set of feature positions, represented by unit length direction vectors ξ_c in the camera frame (as denoted by the subscript c), plus their projected displacement rates χ_c, also referred to as optical flow. The directions ξ_c themselves do not contain information on the camera motion and they are taken as a given. The displacement rates however may be expressed as a function of $\mathbf{x}$, $\mathbf{y}$ and ξ_c assuming all motion is due to the spacecraft. The starting point is to express the motion of these features as an angular rate Ω_c. This angular rate consists

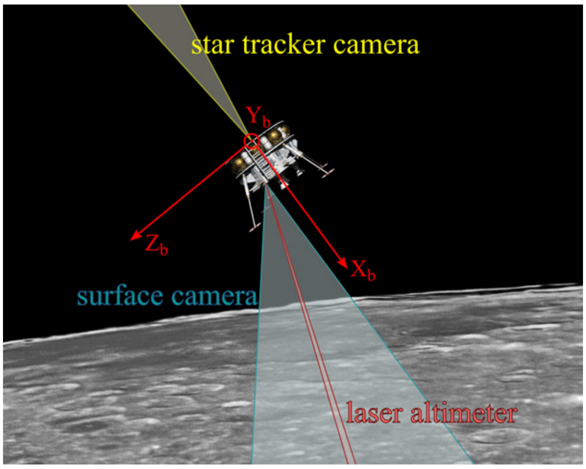


Fig. 1 Schematic optical sensor configuration.

of a component due to translational motion Ω_{c_t} and a component due to rotational motion Ω_{c_r}. The translational velocity vector of the spacecraft in the camera frame $\mathbf{v}_c$ must be considered with respect to the direction ξ_c and the distance D to each feature. Only the component of $\mathbf{v}_c$ which is perpendicular to ξ_c contributes to Ω_{c_t}. This is expressed by the crossproduct. After dividing by D one arrives at the angular rate (in rad/s) due to translation:

$$\Omega_{c_t} = \frac{\mathbf{v}_c \times \xi_c}{D} \tag{5}$$

The rotational component of Ω_c is equal to the component of the rotational rate of the spacecraft which is again perpendicular to ξ_c. Applying the crossproduct again achieves this although the resulting direction is 90° false. By applying the crossproduct with the unit vector ξ_c a final time we arrive at the desired angular rate due to rotation:

$$\Omega_{c_r} = (\omega_c \times \xi_c) \times \xi_c \tag{6}$$

The total angular rate of the feature's motion in the camera frame of reference is the sum of the translational and rotational components:

$$\Omega_c = \Omega_{c_t} + \Omega_{c_r} = \frac{\mathbf{v}_c \times \xi_c}{D} + (\omega_c \times \xi_c) \times \xi_c \,, \tag{7}$$

The camera frame is defined such that the image is projected parallel to the X-Y plane, with the X-axis pointing up and the Y-axis to the right when looking through the lens. The Z-axis coincides with the optical axis to form a right-handed coordinate frame.

The two dimensional optical flow vector χ_c is the projected feature displacement rate as seen in the camera frame. We use rectified images such that a pinhole projection model may be applied. The geometry is drawn in Fig. 2, from which the following expression may be found:

$$\chi_c = \begin{bmatrix} -\Omega_{c,z}\frac{\xi_{c,y}}{\xi_{c,z}} + \frac{1+\xi_{c,x}^2}{\xi_{c,z}^2}\Omega_{c,y} \\ \Omega_{c,z}\frac{\xi_{c,x}}{\xi_{c,z}} - \frac{1+\xi_{c,y}^2}{\xi_{c,z}^2}\Omega_{c,x} \end{bmatrix} . \tag{8}$$

The subscripts x, y and z denote the component along that axis in the camera frame of reference.

In order to use Eqs. 7 and 8 in the navigation filter, the scale of the optical flow geometry in Figure 3 must be resolved. This scale appears in Eq. (7) as the distance D to a feature on the lunar surface. D must be described as a function of $\mathbf{x}$ and $\mathbf{y}$.

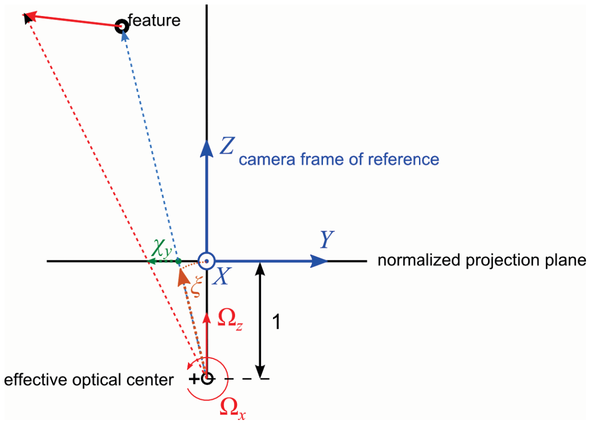


Fig. 2 Diagram showing the y-component of χ.

One approach, used in the NPAL project,[15] relies on the rate of change in the translational optical flow to determine distances to tracked features. From Eq. (7), only optical flow due to translation scales with distance. By comparing the acceleration of features due to translational motion of the camera to the specific force as measured by the accelerometers, one can observe the distances to these features. However, the distance estimates require sufficiently large accelerations in relation to the distances for this method to work. It may be expected to work in the final approach phase where the vertical descent generates a diverging optical flow field. The current application requires the relative velocity navigation earlier in the approach where conditions are unfavorable. We therefore developed another approach.

This approach to derive a relation between the navigation state $\mathbf{x}$ and the distance to a tracked feature on the lunar surface involves using the estimated vehicle position $\mathbf{r}_m$, its attitude q_m and information about the shape of the moon. This method has

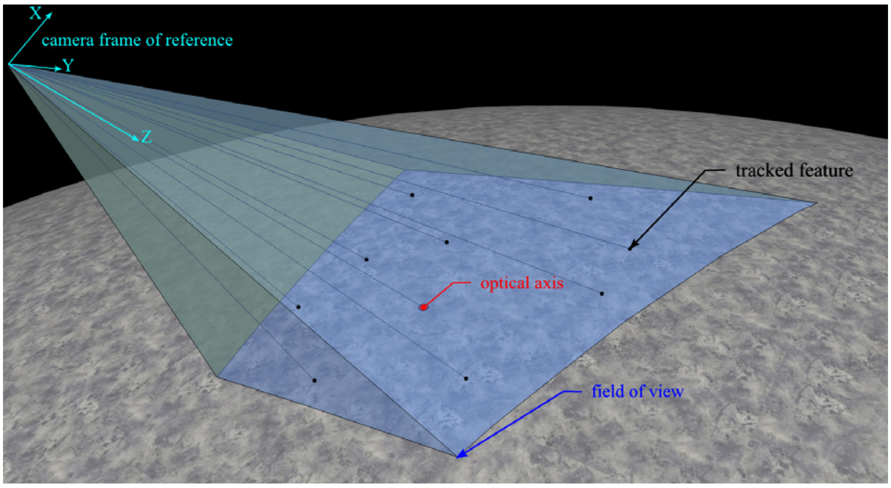


Fig. 3 Schematic geometry for the surface camera.

the advantage that its accuracy does not depend on the details of the vehicle motion. The current implementation is based on an ellipsoid model, although one could in principle substitute a detailed DEM of the surface. The method is well suited to incorporate corrections from the laser altimeter. This presents another advantage over the derivative based method.

The distances D belonging to the feature directions ξ_c are estimated by calculating the intersection with the lunar ellipsoid Λ which, in a moon-centered, Cartesian reference frame, has the following form

$$\Lambda : \mathbf{r}_{le}^T \Theta \mathbf{r}_{le} = 1 \; , \tag{9}$$

with $\mathbf{r}_{le}$ being the set of position vectors on the lunar ellipsoid and Θ being a 3×3 matrix defining the size and shape of the ellipsoid. To calculate D belonging to the intersection of a feature in direction ξ_m with Λ from viewing position $\mathbf{r}_m$, we define a line of sight $\mathbf{s}$ parametrized in terms of D:

$$\mathbf{s}\left(D\right) = D\xi_m + \mathbf{r}_m \; . \tag{10}$$

The problem may now be formulated as follows:

$$\mathbf{s} = \mathbf{r}_{le} \; . \tag{11}$$

Substituting Eq. (11) into Eq. (9) yields

$$\mathbf{s}^T \Theta \mathbf{s} = 1 \; . \tag{12}$$

Further, substituting Eq. (10) makes the distance appear:

$$(D\xi_m + \mathbf{r}_m)^T \Theta (D\xi_m + \mathbf{r}_m) = 1 . \qquad (13)$$

which is a quadratic equation having either 0, 1 or 2 solutions for D. The distance to the surface is the smallest positive solution. If no positive solution is found, the distance may be considered infinite to represent the fact that the camera is looking at the sky. In this case Eq. (7) only depends on the rotational rate ω, which still provides information to correct the gyroscope bias $\mathbf{b}_g$.

In order to correct for the local surface height w.r.t. Λ, a common scaling factor c is applied to Θ yielding a local fit for the ellipsoid:

$$\Theta_{local} = \frac{1}{c^2}\Theta . \qquad (14)$$

Substituting Eq. (14) into Eq. (13) yields

$$c^2 = (D_{laser}\xi_m + \mathbf{r}_m)^T \Theta (D_{laser}\xi_m + \mathbf{r}_m) . \qquad (15)$$

When a laser altimeter measurement arrives, Eq. (15) may be solved for c by selecting ξ_m equal to the laser altimeter direction and substituting the laser distance measurement value D_{laser}.

As such, the ellipsoid has been fit to the point on the lunar surface measured by the laser altimeter. The direction of the laser altimeter is chosen to coincide with the optical axis of the surface camera, see Figure 3. This implicitly assumes a smooth ellipsoidal surface and any height deviation, e.g. due to mountain ridges or craters, must be treated as a measurement error. This has been incorporated in the observation model by making the measurement covariance for each tracked feature a function of the square of its Euclidean distance d to the intersection of the optical axis with the lunar surface.

$$R_\chi(d) = R_{\chi,0} + \delta R_\chi d^2 \qquad (16)$$

3 Filter Algorithm

We have chosen to use the Unscented Kalman Filter[6] (UKF) as mentioned in the introduction as it is equivalent in accuracy to a second order filter without the need for deriving the Jacobian and Hessian matrices of the process and observation models. In the ATON lunar lander, measurements from the IMU are available at a high sample rate and without significant delay while measurements from the optical sensors have a varying delay of up to 0.5 s and will be available at lower sample rates. This asynchronous, delayed nature of correction updates must be taken into account in the navigation filter. An indirect, or error-state formulation allows the full state to be propagated quickly based on inertial data, while the error state is updated using the other sensors. Feedback after such an update corrects the full state. The IMU is able to follow the high frequency motion of the spacecraft very accurately, while the inertial error propagation equations are low frequency.[18] The processing of optical sensor data may therefore be done in a separate thread as data become available. Logged recording time stamps allow the filter to account for delays by taking the error state and covariance at the corresponding filter timestep and applying the resulting innovation to the current filter state.

Let the error state be defined as $\delta\mathbf{x} =: \hat{\mathbf{x}} - \mathbf{x}$. By combining this definition with Eq. (3) and Eq. (4), the error state transition function becomes

$$\begin{aligned}\delta\mathbf{x}(k+1\,|\,k) &= \hat{\mathbf{x}}(k+1\,|\,k) - \mathbf{x}(k+1)\\ &= \phi\left(\hat{\mathbf{x}}(k\,|\,k), \mathbf{y}(k)\right) - \phi\left(\mathbf{x}(k), \mathbf{y}(k)\right) - w(k)\\ &= \phi\left(\hat{\mathbf{x}}(k\,|\,k), \mathbf{y}(k)\right) - \phi\left(\hat{\mathbf{x}}(k\,|\,k) - \delta\mathbf{x}(k\,|\,k), \mathbf{y}(k)\right) - w(k)\end{aligned} \tag{17}$$

where the notation $(k+1\,|\,k)$ means for timestep $k+1$ with information from timestep k. The full state propagation as presented in section 2 uses a quaternion attitude description for computational efficiency and stability. The error of the quaternion is not additive, but rather has the form:

$$\delta q = q \otimes \hat{q}^{-1} \; .$$

However, the error state definition as well as the Unscented transform used in the UKF assume additive error behavior. Therefore, in the error state filter the attitude is described using the axis-angle vector θ_m:

$$\theta_m =: \mathbf{e}\cdot\Phi, \qquad q_m = \begin{bmatrix}\mathbf{e}\sin\left(\frac{\Phi}{2}\right)\\ \cos\left(\frac{\Phi}{2}\right)\end{bmatrix}, \qquad \mathbf{e}^T\mathbf{e} = 1 \; , \tag{18}$$

where $\mathbf{e}$ is the principle axis and Φ the angle of rotation about that axis. θ_m has the required additive error behavior. In the case of the error state, we assume incremental attitude changes. The error quaternion δq therefore corresponds very closely to a small rotation, so the fourth component will be close to unity and the attitude information is contained in the vector component $\delta\mathbf{q}$: $\delta q \simeq \left[\delta\mathbf{q}^T \;\; 1\right]^T$. Further, if the rotation $\delta\Phi$ is sufficiently small, $\delta\mathbf{q}$ may be approximated as $\delta\mathbf{q} \simeq \frac{1}{2}\delta\theta$. Using this assumption and the error state definition, the continuous time differential equation for $\delta\theta_m$ becomes:[18]

$$\frac{d}{dt}\delta\theta_m = [[\omega_{imu}]]\,\delta\theta_m - \delta\mathbf{b}_g - \mathbf{n}_{g_1} \; , \tag{19}$$

where $[[\omega_{imu}]]$ is a 3×3 skew symmetric matrix. Equation (19) may be discretized to yield the state transition function for $\delta\theta_m$.

The error state $\delta\mathbf{x}$ shows the same stochastic behavior as the full state $\mathbf{x}$ since the estimated state $\hat{\mathbf{x}}$ is not a stochastic variable. Therefore, the error covariance matrix P and the process noise covariance matrix for the error state are identical to those of the full state.

The delayed feedback error state UKF equations may be summarized as follows.

1. The error state estimate and covariance are augmented with the mean and covariance of the process noise $w(k)$:

$$\delta\mathbf{x}^a(k|k) = \left[\,\delta\mathbf{x}^T(k|k)\;E\left\{w^T(k+1)\right\}\right]^T \;,$$

$$P^a(k|k) = \begin{bmatrix} P(k|k) & 0 \\ 0 & Q_k \end{bmatrix} \;.$$

2. A set of $2L+1$ prediction sigma points is derived from the augmented state and covariance where L is the dimension of the augmented error state:

$$\varsigma_{p,0}(k|k) = \delta\mathbf{x}^a(k|k) \;,$$

$$\varsigma_{p,i}(k|k) = \delta\mathbf{x}^a(k|k) + \left(\sqrt{(L+\lambda)P^a(k|k)}\right)_i \text{, for } i = 1\ldots L \;,$$

$$\varsigma_{p,i}(k|k) = \delta\mathbf{x}^a(k|k) - \left(\sqrt{(L+\lambda)P^a(k|k)}\right)_{i-L} \text{, for } i = (L+1)\ldots 2L \;,$$

where

$$\left(\sqrt{(L+\lambda)P^a(k|k)}\right)_i$$

is the ith column of the matrix square root of

$$(L+\lambda)P^a(k|k)$$

using the definition: The matrix square root A of B satisfies $B =: AA^T$.

3. The prediction sigma points are propagated through the state transition function for $\delta\mathbf{x}$:

$$\varsigma_{p,i}(k+1|k) = \phi\left(\hat{\mathbf{x}}(k|k),\mathbf{y}(k)\right) - \phi\left(\hat{\mathbf{x}}(k|k) - \varsigma_{p,i}(k|k),\mathbf{y}(k)\right) \text{, for } i = 0\ldots 2L \;.$$

4. The propagated sigma points are recombined to produce the predicted state and covariance:

$$\delta\mathbf{x}(k+1|k) = \sum_{i=0}^{2L} W_s(i)\,\varsigma_{p,i}(k+1|k) \;,$$

$$P(k+1|k) = \sum_{i=0}^{2L} W_c(i)\left[\varsigma_{p,i}(k+1|k) - \delta\mathbf{x}(k+1|k)\right]\left[\varsigma_{p,i}(k+1|k) - \delta\mathbf{x}(k+1|k)\right]^T \;,$$

where the weights for the state and covariance are given by:

$$W_s(0) = \frac{\lambda}{L+\lambda} \;,$$

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

$$W_s(i) = W_c(i) = \frac{1}{2(L+\lambda)} \;,$$

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

Values for α, β and κ have to be chosen to tune the prediction step. Some guidelines to choose these constants for a particular problem are given in Ref. 7.

5. For the update step the availability of new data from the surface camera and/or star tracker is checked. If these are not available, the current prediction is used for the next timestep: So $\delta\mathbf{x}(k+1\,|\,k+1) = \delta\mathbf{x}(k+1\,|\,k)$ and $P(k+1\,|\,k+1) = P(k+1\,|\,k)$ and no feedback to the full state takes place. Otherwise, the predicted state and covariance corresponding to timestep $l \leq k$ of the incoming data are augmented with the mean and covariance of the measurement noise of the available sensors R^*:

$$\delta\mathbf{x}_u^a(l+1\,|\,l) = \left[\, \delta\mathbf{x}^T(l+1\,|\,l)\; E\left\{v^T(l)\right\} \right]^T \;,$$

$$P_u^a(l+1\,|\,l) = \begin{bmatrix} P(l+1\,|\,l) & 0 \\ 0 & R^* \end{bmatrix} \,.$$

6. A set of $2L+1$ update sigma points is derived from $\delta\mathbf{x}_u^a(l+1\,|\,l)$ and $P_u^a(l+1\,|\,l)$ where L is the dimension of the augmented state:

$$\varsigma_{u,0}(l+1\,|\,l) = \delta\mathbf{x}_u^a(l+1\,|\,l) \;,$$

$$\varsigma_{u,i}(l+1\,|\,l) = \delta\mathbf{x}_u^a(l+1\,|\,l) + \left(\sqrt{(L+\lambda)P_u^a(l+1\,|\,l)}\right)_i \;, \text{for } i = 1\ldots L \,,$$

$$\varsigma_{u,i}(l+1\,|\,l) = \delta\mathbf{x}_u^a(l+1\,|\,l) - \left(\sqrt{(L+\lambda)P_u^a(l+1\,|\,l)}\right)_{i-L} \;, \text{for } i = (L+1)\ldots 2L \,.$$

7. An appropriate observation function $h^*(\mathbf{x})$ is composed from the complete observation function by selecting the sensors having new data since the last IMU timestep.

8. The update sigma points are fed to h^*, using $\mathbf{x} = \hat{\mathbf{x}} - \delta\mathbf{x}$ where the sigma points $\varsigma_{u,i}$ are substituted for $\delta\mathbf{x}$:

$$\gamma_i(l+1\,|\,l) = h^*\left(\hat{\mathbf{x}}(l+1\,|\,l) - \varsigma_{u,i}(l+1\,|\,l)\right) \;, \text{for } i = 0\ldots 2L \,.$$

9. The result is recombined to yield the predicted measurement and predicted measurement covariance:

$$\mathbf{z}(l+1\,|\,l) = \sum_{i=0}^{2L} W_s(i)\,\gamma_i(l+1\,|\,l) \;,$$

$$P_{zz} = \sum_{i=0}^{2L} W_c(i) \left[\gamma_i(l+1\,|\,l) - \mathbf{z}(l+1\,|\,l)\right] \left[\gamma_i(l+1\,|\,l) - \mathbf{z}(l+1\,|\,l)\right]^T .$$

10. The UKF Kalman gain is computed as:

$$K_{l+1} = P_{xz} P_{zz}^{-1} ,$$

where the state-measurement cross-covariance matrix is expressed as:

$$P_{xz} = \sum_{i=0}^{2L} W_c(i) \left[\varsigma_{u,i}(l+1\,|\,l) - \delta\mathbf{x}(l+1\,|\,l)\right] \left[\gamma_i(l+1\,|\,l) - \mathbf{z}(l+1\,|\,l)\right]^T .$$

11. The error state update equation using $\mathbf{z}(l+1)$ is:

$$\delta\mathbf{x}(k+1\,|\,k+1) = \delta\mathbf{x}(k+1\,|\,k) + K_{l+1}\left(\mathbf{z}(l+1) - \mathbf{z}(l+1\,|\,l)\right) .$$

12. The updated covariance is:

$$P(k+1\,|\,k+1) = P(k+1\,|\,k) - K_{l+1} P_{zz} K_{l+1}^T .$$

13. At this point the updated error state $\delta\mathbf{x}(k+1\,|\,k+1)$ is fed back to the estimated full state $\hat{\mathbf{x}}$ which has been propagated using IMU data to timestep $k+1$ since the last update at timestep $k-n$:

$$\hat{\mathbf{x}}(k+1\,|\,k+1) = \hat{\mathbf{x}}(k+1\,|\,k-n) - \delta\mathbf{x}(k+1\,|\,k+1) .$$

Subsequently, the error state must be reset to zero: $\delta\mathbf{x}(k+1\,|\,k+1) \Rightarrow 0$. The covariance $P(k+1\,|\,k+1)$ retains its value however, since it represents the uncertainty in $\delta\mathbf{x}$ which is the same as the uncertainty in $\hat{\mathbf{x}}$.

4 Simulation Environment

The simulation environment has been designed in a modular fashion to allow different configurations to work with common parts. For testing during development there is a Software-in-the-Loop (SiL) model which uses logged state data and pre-rendered images from a selection of scenarios to generate further sensor outputs including realistic signal delays. Figure 4 shows an image from the lunar surface rendering. The surface model is based on data from the Kaguya[8] mission.

The results in this work are generated using the SiL configuration with the following characteristics:

- the IMU sensor model has a bias drift of 1°/hr, 300 ppm scale factor stability and $0.03°/\sqrt{\text{hr}}$ white noise running at 100 Hz update rate,
- the star tracker model has 5 arcsec angular accuracy, 5 Hz update rate and variable delay (jitter) of up to 0.2 s,

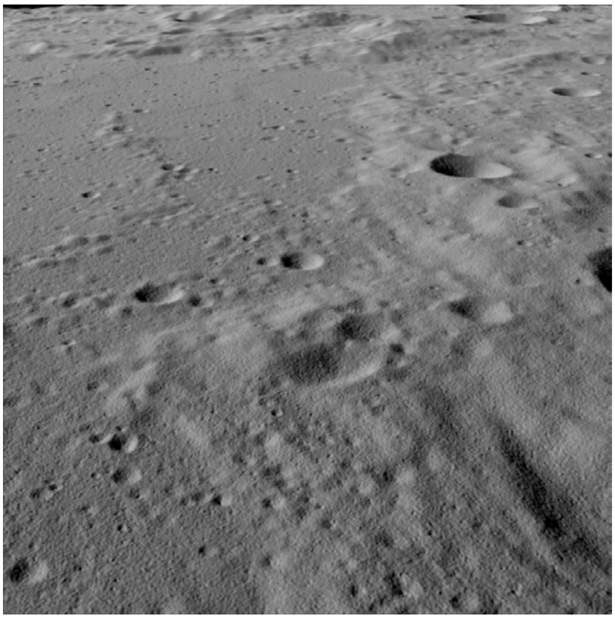


Fig. 4 An example image of the surface camera simulation.

- the surface camera has a resolution of 1024 × 1024 pixels, 40° × 40° field of view and produces 30 frames per second. Including the Lucas-Kanade tracker, the delay for optical flow may be up to 0.2 s depending on the number of features to be tracked which was capped at 40,
- the laser altimeter is accurate to 5 meter with 0.5 meter bias and an update rate of 0.5 Hz.

5 Results

The scenario presented here starts at 700 km downrange from the landing zone at an altitude of about 10 km. The spacecraft travels at 1700 m/s at that point. The trajectory is shown in Fig. 5. It is assumed that at this point the absolute crater navigation does not find enough matching craters in its database anymore and hands over its estimated state to the relative navigation. To assess the expected error accumulation due to the relative velocity navigation mode, the true state has been taken as initial state. Due to the fact that the sensor signals don't give direct positioning information, an initial position error would not contribute to the error growth during this phase of the landing.

The same scenario has been flown with and without the surface camera to show the precision gained from using optical flow. With the surface camera and laser al-

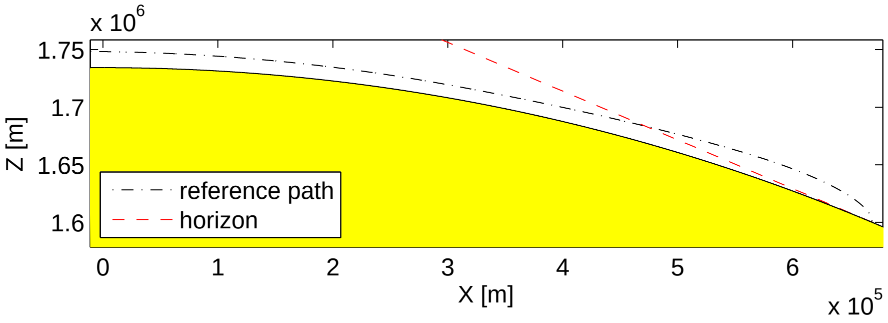
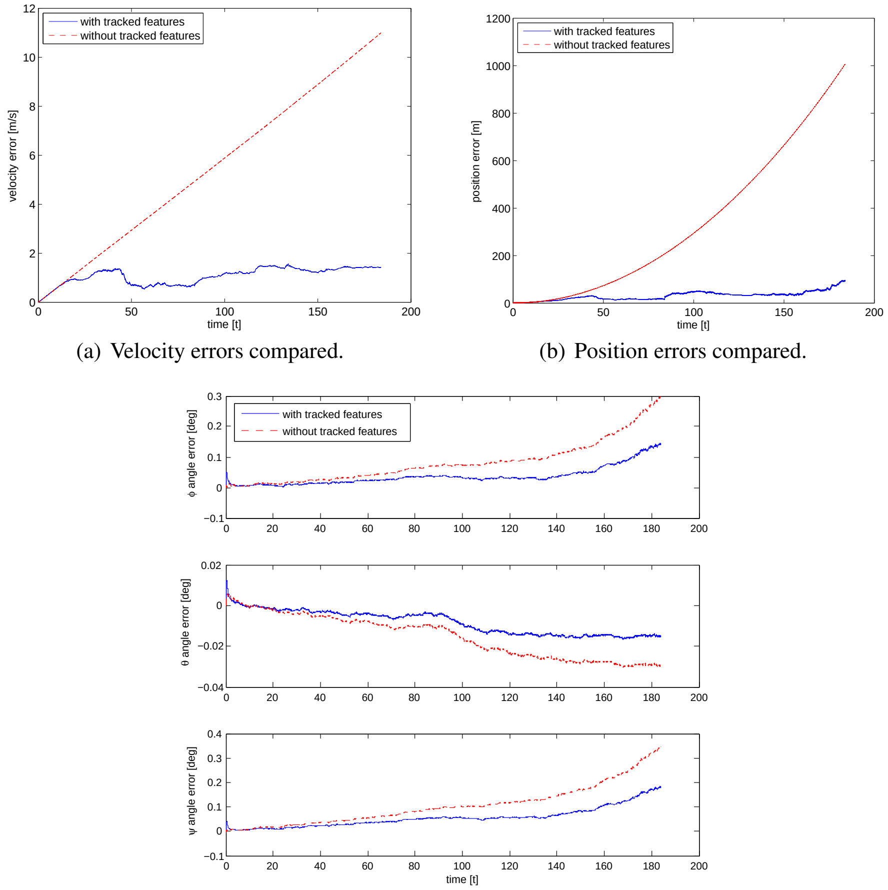


Fig. 5 Spacecraft trajectory from scenario start to landing.

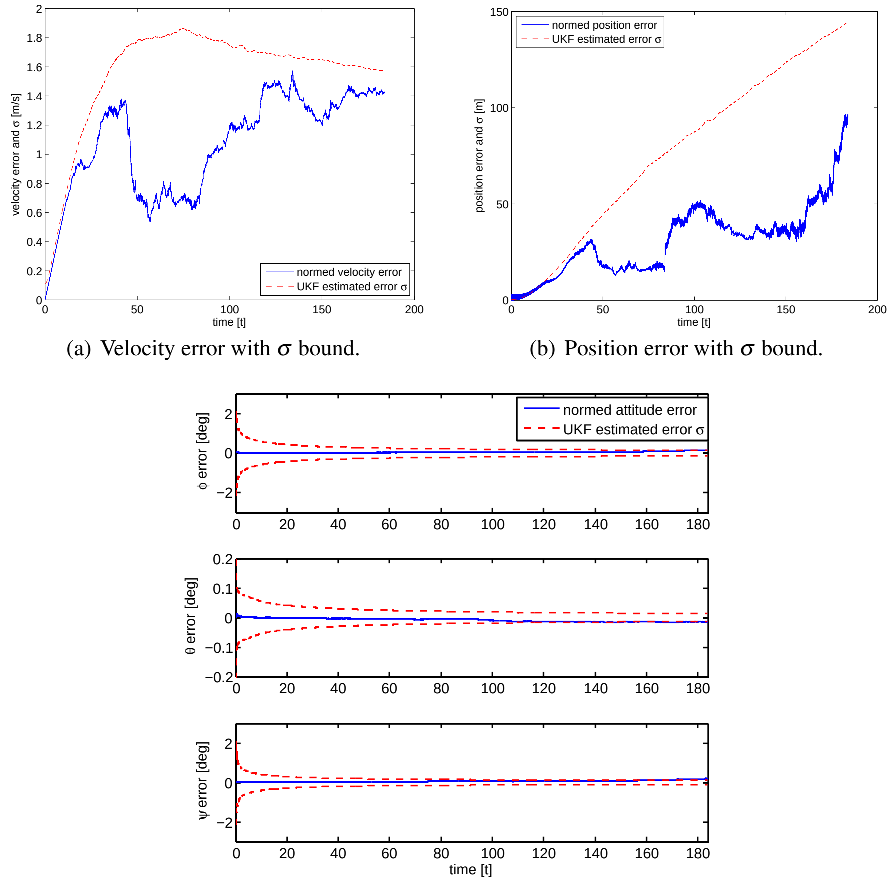


(a) Velocity errors compared.

(b) Position errors compared.

(c) Attitude errors compared.

Fig. 6 Resulting errors for a powered descent trajectory with and without surface camera.



(a) Velocity error with σ bound.

(b) Position error with σ bound.

(c) Attitude error with σ bound.

Fig. 7 Standard deviation error bounds estimated by the UKF for the case with surface camera.

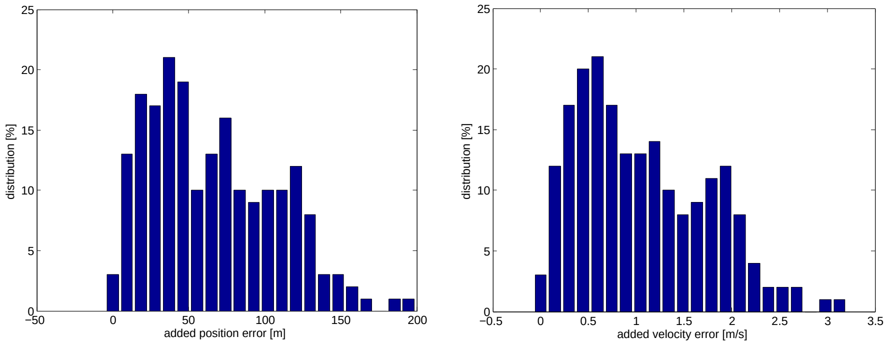
timeter turned off, the remaining sensors are the IMU and star tracker. The results in Fig. 6 show the position, velocity and attitude errors comparing the two cases. The results without the surface camera show a typical error growth expected for an inertial navigation system. The velocity error increases linearly in time and the position quadratically. This can be attributed to the accelerometer biases which cannot be estimated in this case. Attitude is mainly driven by the star tracker, but benefits from optical flow as well. (Note that Euler angles have been used here merely for interpretation purposes: the filter works with quaternions and the axis-angle vector internally as described in subsection 2.1 and section 3.) The relative velocity optical navigation limits the velocity error to about 1.4 m/s. Although the position error is still unbounded, its growth is also significantly reduced. Figure 7 shows the case with surface camera including the 1σ-bounds derived from the auto covariances in P estimated by the UKF. The errors remain below this bound showing that the filter produces a consistent estimate. From the development of σ in time one can further conclude that the velocity and attitude are observable as σ converges. The position is not observable as may be expected when only surface features with unknown position are available.

To investigate the effects of initial position and velocity errors a Monte Carlo simulation with 100 runs was performed. The initial errors were normally distributed. Together with the resulting added errors their statistical properties are listed in Table 1.

	initial position error [m]	initial velocity error [m/s]	added position error [m]	added velocity error [m/s]
mean	95	0.4	65	1.07
standard deviation	32	0.5	42	0.68

Table 1 Monte Carlo Analysis: statistical properties of the errors.

The added errors are the normed additional position and velocity errors at the end of the relative velocity navigation phase. Their distributions are shown in Fig. 8.



(a) Position error distribution.

(b) Velocity error distribution.

Fig. 8 Monte Carlo Analysis: resulting error distributions.

6 Conclusion

The results so far show a significant position error growth reduction when optical flow and star tracker sensors are used to update the IMU state predictions. An important element of the navigation filter is the way it takes the processing delay for the optical data into account. Although currently the filter implementation is single threaded, the algorithm lends itself to a dual threaded structure, allowing fast processing of IMU data and enough processing time for the optical updates including delay compensation.

It should be noted that the crater navigation accuracy will degrade gradually and relative navigation will work in parallel for part of the descent. This combined navigation mode is planned as part of the ATON project, but the current paper concentrates on the evaluation of relative velocity navigation only. Also, the actual landing site will be autonomously selected based on LIDAR and camera image processing. This process starts as soon as the landing site crosses the horizon. From then on, navigation will continue relative to the selected landing site. This marks the endpoint for the relative velocity navigation mode. The main goal is to provide slow position error growth in the gap between crater navigation and landing site relative navigation. The results in Table 1 show that the position error can be expected to be 65 m with a standard deviation of 42 m under the chosen sensor characteristics and navigation errors at the start of the relative phase.

Future work will include demonstration and validation of Technology Readiness Level 4 (TRL4) using a Hardware-in-the-Loop (HiL) configuration which will use the TRON Facility,[9] see Fig. 9, as well as flight tests using superARTIS, see Fig. 10.

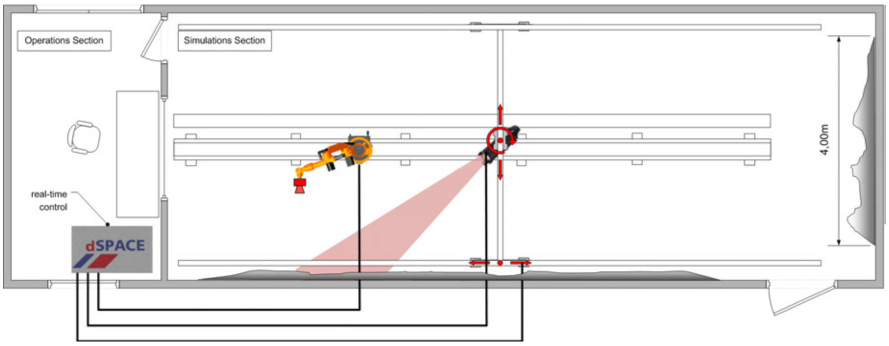


Fig. 9 The TRON facility can generate planetary approach images using an industrial robotic arm, scaled relief models and special lighting.

Acknowledgements The author would like to thank his colleagues in the ATON project team for their work and fruitful cooperation in building the simulation environment and especially Franz Andert for his help building the feature tracker. Furthermore, he is indebted to the referees for their careful reviews and helpful suggestions.

Fig. 10 DLR's pilotless helicopter superARTIS shown at the ILA 2012 in Berlin.

References

1. Tye Brady, Erik Bailey, Timothy Crain, and Stephen Paschall. ALHAT System Validation. In *8th International ESA Conference on Guidance, Navigation & Control Systems*, Karlovy Vary, Czech Republic, June 2011. ESA.
2. R.L. Farrenkopf. Analytic Steady-State Accuracy Solutions for Two Common Spacecraft Attitude Estimators. *Journal of Guidance and Control*, 1(4):282–284, July–August 1978.
3. B. Frapard and S. Mancuso. Vision Navigation for European Landers and the NPAL Project. In *6th International ESA Conference on Guidance, Navigation and Control Systems*, Loutraki, Greece, October 2005. ESA.
4. A. E. Johnson, A. Ansar, L. H. Matthies, N. Trawny, A. I. Mourikis, and S. I. Roumeliotis. A General Approach to Terrain Relative Navigation for Planetary Landing. In *Infotech*, Rohnert Park, California, United States, May 2007. AIAA.
5. Simon Julier and Jeffrey K. Uhlmann. *Handbook of Multisensor Data Fusion*, chapter 13: Data Fusion in Nonlinear Systems. CRC Press LLC, 2001.
6. Simon J. Julier and Jeffrey K. Uhlmann. A New Extension of the Kalman Filter to Nonlinear Systems. *Proceedings of SPIE*, 3068:182 – 193, 1997.
7. Simon J. Julier and Jeffrey K. Uhlmann. Unscented Filtering and Nonlinear Estimation. *Proceedings of the IEEE*, 92(3):401 – 422, March 2004.
8. Manabu Kato, Susumu Sasaki, and Yoshisada Takizawa. The kaguya mission overview. *Space Science Reviews*, 154:3–19, 2010. 10.1007/s11214-010-9678-3.
9. H. Krüger and S. Theil. TRON - Hardware-in-the-loop test facility for lunar descent and landing optical navigation. In *18th IFAC Symposium on Automatic Control in Aerospace*. IFAC, 2010.
10. Daero Lee and Henry Pernicka. Vision-Based Relative State Estimation Using the Unscented Kalman Filter. *International Journal of Aeronautical and Space Sciences*, 12(1):24–36, 2011.
11. Shuang Li, Pingyuan Cui, and Hutao Cui. Autonomous Navigation and Guidance for Landing on Asteroids. *Aerospace Science and Technology*, 10(3):239–247, 2006.
12. Shuang Li, Pingyuan Cui, and Hutao Cui. Vision-aided Inertial Navigation for Pinpoint Planetary Landing. *Aerospace Science and Technology*, 11(6):499–506, 2007.
13. Bruce D. Lucas and Taireo Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. *International Joint Conference on Artificial Intelligence*, 3:674–679, August 1981.
14. Anastasios I. Mourikis, Nikolas Trawny, Stergios I. Roumeliotis, Andrew E. Johnson, Adnan Ansar, and Larry Matthies. Vision-aided Inertial Navigation for Spacecraft Entry, Descent, and Landing. *IEEE Transactions on Robotics*, 25(2):264–280, April 2009.

15. multiple. Navigation for Planetary Approach & Landing. Final report, EADS Astrium, May 2006.
16. multiple. VisNav Executive Summary. Technical Report CMU-CS-91-132, EADS Astrium, September 2009.
17. 2006 Solar System Exploration Roadmap for NASA's Science Mission Directorate. http://solarsystem.nasa.gov/multimedia/downloads/SSE_RoadMap_2006_Report_FC-A_opt.pdf, 2006.
18. Stergios I. Roumeliotis, Gaurav S. Sukhatme, and George A. Bekey. Smoother based 3D Attitude Estimation for Mobile Robot Localization. In *IEEE International Conference on Robotics and Automation*, pages 1979–1986, Detroit, MI , USA, May 1999.
19. Hanspeter Schaub and John L. Junkins. *Analytical Mechanics of Space Systems*. AIAA Education Series, 2003. ISBN 1-56347-563-4.
20. Stephen Steffes. Extended Kalman Filter for HNS Navigator. Technical report, DLR, May 2011.
21. Scott A. Striepe, Chirold D. Epp, and Edward A. Robertson. Autonomous Precision Landing and Hazard Avoidance Technology (ALHAT) Project Status as of May 2010. In *International Planetary Probe Workshop 2010*, Barcelona, Spain, June 2010.
22. M. J. Verveld, Q. P. Chu, C. de Wagter, and J. A. Mulder. Optic Flow Based State Estimation for an Indoor Micro Air Vehicle. In *Guidance, Navigation, and Control Conference and Exhibit*, Toronto, Ontario Canada, August 2010. AIAA.