Vision-Aided Inertial Navigation for Pin-Point Landing using Observations of Mapped Landmarks

Nikolas Trawny
Dept. of Computer Science & Engineering
University of Minnesota
Minnesota, MN 55455
trawny@cs.umn.edu

Anastasios I. Mourikis
Dept. of Computer Science & Engineering
University of Minnesota
Minnesota, MN 55455
mourikis@cs.umn.edu

Stergios I. Roumeliotis
Dept. of Computer Science & Engineering
University of Minnesota
Minnesota, MN 55455
stergios@cs.umn.edu

Andrew E. Johnson
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109
Andrew.E.Johnson@jpl.nasa.gov

James Montgomery
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109
James.F.Montgomery@jpl.nasa.gov

Abstract

In this paper we describe an Extended Kalman Filter (EKF) algorithm for estimating the pose and velocity of a spacecraft during Entry, Descent and Landing (EDL). The proposed estimator combines measurements of rotational velocity and acceleration from an Inertial Measurement Unit (IMU) with observations of *a priori* Mapped Landmarks (MLs), such as craters or other visual features, that exist on the surface of a planet. The tight coupling of inertial sensory information with visual cues results in accurate, robust state estimates available, at a high bandwidth. The dimensions of the landing uncertainty ellipses achieved by the proposed algorithm are *three orders of magnitude* smaller than those possible when relying exclusively on IMU integration. Extensive experimental and simulation results are presented, which demonstrate the applicability of the algorithm on real-world data and analyze the dependence of its accuracy on several system design parameters.

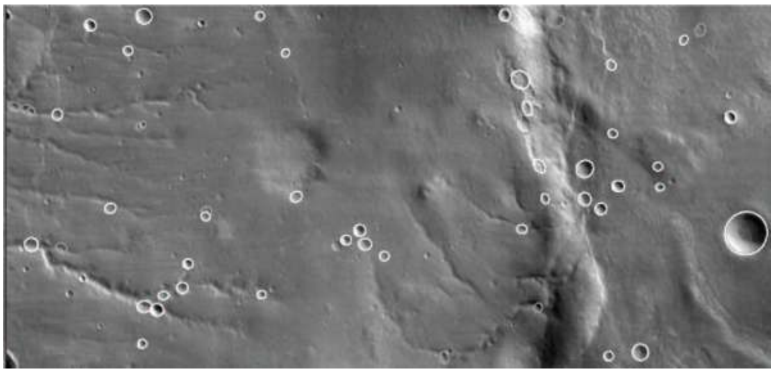


Figure 1: Example output of crater detection algorithm (Cheng et al., 2003). Detected craters are denoted by white ellipses.

1 Introduction

Space missions that involve landing on planets, moons, or asteroids are inherently challenging. Large communication delays rule out earth-based remote control during Entry, Descent and Landing (EDL). High-accuracy position and attitude (pose) determination, a necessity for precise trajectory control during EDL, is difficult due to the absence of navigation aids such as GPS or radio-beacons. Current EDL systems are based on the integration of acceleration and rotational velocity measurements from an Inertial Measurement Unit (IMU). As with all forms of dead reckoning, pose estimates obtained exclusively from an IMU are subject to *unbounded error accumulation* due to the integration of noise and biases. This error accumulation, when combined with uncertainties in surface-relative position and attitude at entry, has resulted in large landing error ellipses (e.g., for the Viking Landers, Mars Pathfinder, Mars Polar Lander, and the Mars Exploration Rovers) with axes of dimensions in the order of 100-300 km. Reducing the landing position uncertainty of future missions to *sub-kilometer accuracy* is of great importance. Specifically, providing spacecraft with such Pin-Point Landing (PPL) capabilities will (i) *increase science return* by precise landing at sites of scientific interest, and (ii) *improve mission safety* by permitting the *a priori* selection of safe landing areas, as well as by estimating and limiting touchdown velocities to within airbag or landing gear specifications.

One possible approach for enabling precise landing is to use *only camera measurements* of *known features* on the surface, such as craters (Cheng and Ansar, 2005) (cf. Fig. 1). Craters appear in sufficient density on many solar-system bodies of interest, and are fairly stable in appearance over time or under different lighting conditions, thus making them a suitable type of feature to observe. Additionally, images from satellites, such as the Mars Global Surveyor and the Mars Reconnaissance Orbiter, with resolution in the order of 1 meter per pixel, will allow to determine the position of numerous craters of various sizes (as small as 10 meters in diameter) *a priori* along the direction of the intended EDL trajectory. Similar maps of craters can also be obtained for other planetary objects during the pre-entry phase of an exploration mission. However, purely visual feature-based pose estimation has certain disadvantages. It requires the concurrent detection of *at least* 3 mapped features at every time instant, and yields only *infrequent* pose estimates limited by the throughput of the image processing algorithm. Additionally, only crude estimates of the vehicle's velocity and acceleration are obtained by differentiation of the computed position and attitude. These quantities, though, need to be precisely estimated at a high rate in order to allow for active control of the spacecraft during EDL.

Cameras and IMUs are *complementary* sensors in terms of the type and rate of information they provide. Over short intervals, IMUs are capable of precisely tracking the dynamic profile of the motion of a vehicle, at high rates and with minimal computational cost. Pose estimates based on IMU integration, however, diverge over longer periods of time. In contrast, image-based detection of known features provides pose estimates of bounded uncertainty, but at the expense of high processing requirements and limited bandwidth. Hence, *fusion* of image measurements of Mapped Landmarks (MLs), such as craters, with IMU measurements result in accurate, high-bandwidth pose estimates that cannot be achieved by either of the two sensors alone.

In this paper, we present an Extended Kalman Filter (EKF) algorithm for processing IMU measurements and visual observations of Mapped Landmarks (MLs)[1]. When operating in conjunction with a feature detection algorithm and a visual map of the landing site, the proposed approach can be directly applied to provide precise, real-time estimates of attitude, position, and velocity of a spacecraft, as well as IMU biases, during EDL. The filter's performance is verified extensively both in simulation and experimentally. The experiments presented in Section 4 utilize data from (i) the 2002 field testing of the Descent Image Motion Estimation System (DIMES) of the Mars Exploration Rovers (Johnson et al., 2005), and (ii) the 2004 NASA Mars Science Lab (MSL) subsonic Parachute Drop. In these tests our algorithm attained final error bounds (3σ) in the order of ± 3 m for position and $\pm 0.5°$ for orientation. Additionally, Monte Carlo simulation experiments were conducted in order to quantify the dependence of the estimation accuracy on factors such as (i) feature detection accuracy, (ii) density of MLs, and (iii) image frame rate. These results allow for determining the required performance of the feature extraction algorithm to meet mission precision requirements.

Following a brief review of related literature in Section 2, we describe our estimation algorithm in Section 3. Extensive experimental and simulation results are presented in Sections 4 and 5, respectively. Section 6, finally, provides concluding remarks and an outlook on future work.

2 Literature Review

Simultaneous Localization and Mapping (SLAM) methods constitute one family of approaches for combining inertial measurements with camera observations (Langelaan, 2006; Strelow, 2004; Kim and Sukkarieh, 2004). These algorithms jointly estimate the camera pose and the 3D feature positions, thus eliminating the need for a pre-loaded map. However, a serious drawback of SLAM in the context of EDL is that when no MLs are detected, no *absolute pose* information is available, and the system state is unobservable (Andrade-Cetto

[1] From here on we will be using the terms Mapped Landmarks (MLs) and known features interchangeably to denote visual features on the ground whose positions are known *a priori* (e.g., from satellite images).

and Sanfeliu, 2004). Therefore, SLAM methods on their own cannot provide the precision necessary for pin-point landing. Furthermore, its high computational complexity (quadratic in the number of features) makes real time, 3D visual SLAM with a large number of features prohibitive for space applications with inherently limited computational resources. The algorithm proposed in this paper fuses observations of *known features* with IMU measurements, which guarantees observability (when at least 3 MLs are visible), and thus results in excellent estimation accuracy.

In what follows, we present an overview of relevant work on the problem of motion estimation for aerospace vehicles using observations of known features with or without inertial measurements. In particular, we describe: (i) two vision-only based algorithms using beacons for spacecraft docking (Crassidis et al., 2000), (Mortari et al., 2004), (ii) a vision-only based approach to EDL using observations of mapped craters (Cheng and Ansar, 2005), and (iii) two methods for fusing vision and IMU data for estimating position and attitude by tracking a single planar target (Wu et al., 2005), or only position from MLs (Bayard and Brugarolas, 2005).

Crassidis *et al.* (Crassidis et al., 2000) use unit vector observations of known features (beacons) to estimate position and orientation. Instead of propagating the state using inertial measurements, they employ a stochastic zero-acceleration model in a recursive, one-step ahead predictive filter. This method is applicable for modeling maneuvers with *slow dynamics* that occur, for example, during rendezvous and docking. Simulation results for trajectories with linear and rotational velocities of 0.03 m/s and 10^{-4} rad/s respectively, show pose estimation accuracies in the order of millimeters and 0.04^{o} using observations of 6 beacons. An observability analysis (Crassidis et al., 2000; Sun and Crassidis, 2002) shows that in order to determine both position and orientation from image measurements alone, 3 non-collinear unit vector observations are necessary for the pose to be observable; 4 or more such measurements allow for determining a unique pose estimate.

In (Mortari et al., 2004), one linear algebra resection algorithm to solve for position and attitude, and two methods for determining position only (without knowledge of attitude) are presented. Two of these methods are especially useful if no initial pose estimate is available and can be used as initialization algorithms for existing iterative methods. The linear resection method is non-iterative and requires 6 simultaneous measurements to known beacons. The position-only estimation eliminates the unknown attitude and uses a modified Sylvester resultant to reduce the problem to solving a polynomial of degree 8. In addition, an iterative Newton-Raphson algorithm is presented. Similar algorithms for estimating the pose of a camera from observations of mapped points are also known from the Computer Vision community (see, for example, (Ansar and Daniilidis, 2003; Quan and Lan, 1999) and references therein).

The above mentioned *vision-only* algorithms are geared specifically to the case where *beacons* consist of controllable LEDs. This eliminates the need for complex feature extraction and identification and enables high frequency measurements. Albeit valid for slow rendezvous and docking maneuvers, such methods are obviously not applicable for the case of EDL.

Cheng and Ansar (Cheng and Ansar, 2005) propose a vision algorithm for *crater detection* *and matching* with application to EDL. Craters are identified based on the size and orientation of their outlining ellipses, which are compared against a database of previously mapped craters. Position is estimated using the centroids of a subset of *at least 3* matched craters in a linear pose estimation algorithm detailed in (Ansar and Daniilidis, 2003). A coarse velocity estimate is formed by combining the *average* velocity between two image-based position estimates and the velocity computed by integrating the accelerometer measurements over the same interval. We should note that this process provides velocity estimates at *low rates*, dictated by the throughput of the image processing algorithm.

In (Wu et al., 2005), the authors present a vision-aided inertial navigation approach based on tracking a *rectangular target*; in particular a dark window against white background. The position, attitude, and area of the target are considered known. A camera is used to measure the centroid of the target and its projected area in the image. This visual information, along with heading measurements from a *magnetometer*, is processed by an EKF to update the state of the vehicle and correct errors incurred by integrating the IMU signals. This method was demonstrated in practice, where it provided position errors smaller than 3.2 m when a helicopter flew in the vicinity of, and tracked one target of area 4 m^2. As explained by the authors, this approach is sensitive to variations in the magnetometer measurements, and its accuracy reduces with the distance from the target. This is due to the fact that there exist multiple poses (position and attitude) of the camera that explain the received image measurements.

Bayard and Brugarolas (Bayard and Brugarolas, 2005) employ IMU and camera measurements to the problem of position and velocity estimation for spacecraft orbiting asteroids or comets. In their work, attitude is considered known, for example from star tracker measurements and is *not* estimated. Furthermore, IMU measurements are only used during thruster firings and not during coast phases, due to the low signal to noise ratio. Since EDL is highly dynamic and barely any stars are visible during atmospheric reentry, our algorithm cannot consider attitude to be known. Instead, we employ gyroscope measurements to estimate attitude in addition to position and velocity. We furthermore account for biases in gyroscopes and accelerometers in our estimator, which is not the case in (Bayard and Brugarolas, 2005).

The algorithm presented in this paper supports *tight integration* of IMU measurements with unit vector observations of MLs (e.g., craters) within a Kalman filter estimator. By integrating IMU measurements, our approach provides motion estimates with *higher bandwidth* than vision-only algorithms, and improves *robustness* to errors resulting from occlusions or variations in lighting conditions. By processing camera observations, our algorithm *bounds the error growth* inherent to purely IMU-based pose estimation.

Contrary to previous approaches, by simultaneously estimating *position, attitude, velocity, and IMU biases*, the proposed filter computes the necessary high-precision motion information required by a controller during powered entry. Additionally, our algorithm can operate with *any* number of detected features since it relies on the IMU data when none or only few features are detected, while it depends more on visual observations when a large number of these becomes available. Finally, we should mention that our algorithm is resource-adaptive. In particular, in a system where the available CPU resources are limited, image measurements can be processed intermittently without significant loss of positioning accuracy (cf.

Section 5).

3 Estimator Description

In this section, we outline the state and covariance propagation and update models used to fuse inertial measurements from a strap-down IMU, with observations of MLs detected in camera images.

3.1 Propagation

The estimated state vector includes the attitude quaternion, ${}^{B}_{G}\bar{q}$, that describes the orientation of the rotating, planet-centered, planet-fixed *global frame* of reference $\{G\}$ with respect to the *body frame* of reference $\{B\}$. We use the quaternion definition (Breckenridge, 1999) that, contrary to the classical Hamiltonian definition, relates products of quaternions and their corresponding rotational matrices $\mathbf{C}(\bar{q})$ as

$$\mathbf{C}(\bar{q}_1 \otimes \bar{q}_2) = \mathbf{C}(\bar{q}_1) \cdot \mathbf{C}(\bar{q}_2) \tag{1}$$

From here on, we will denote the particular rotation matrix corresponding to ${}^{B}_{G}\bar{q}$ as $\mathbf{C}_q$ for brevity. Further estimated quantities are the body velocity, ${}^{G}\mathbf{v}$, and position, ${}^{G}\mathbf{p}$, as well as the biases of the gyroscopes, $\mathbf{b_g}$, and of the accelerometers, $\mathbf{b_a}$. Concatenation of these quantities defines the state vector as $\mathbf{x}(t) = \begin{bmatrix} {}^{B}_{G}\bar{q}^{\mathrm{T}}(t) & \mathbf{b_g}^{\mathrm{T}}(t) & {}^{G}\mathbf{v}^{\mathrm{T}}(t) & \mathbf{b_a}^{\mathrm{T}}(t) & {}^{G}\mathbf{p}^{\mathrm{T}}(t) \end{bmatrix}^{\mathrm{T}}$.

3.1.1 Kinematics & State Propagation

The propagation of the state estimate is carried out using only acceleration and rotational velocity measurements from the IMU[2]. The kinematics can be formulated as a first-order system of differential equations, with the derivatives taken with respect to a planet-centered, non-rotating (quasi-inertial) frame of reference. For navigation purposes, the estimated quantities are expressed with respect to a rotating, planet-fixed coordinate frame, taking into account effects due to the planet rotation rate $\boldsymbol{\omega}_G$.

The gyroscope (accelerometer) measurements are assumed to be corrupted by zero-mean, white Gaussian noise $\mathbf{n_g}$ ($\mathbf{n_a}$), and a slowly-changing bias, $\mathbf{b_g}$ ($\mathbf{b_a}$). The biases are modeled as first-order random processes driven by zero-mean, white Gaussian noises $\mathbf{n_{rg}}$ and $\mathbf{n_{ra}}$, respectively.

[2] For detailed derivations of the propagation model used, the interested reader is referred to (Chatfield, 1997; Lefferts et al., 1982; Trawny and Roumeliotis, 2005).

The kinematic equations for this model are summarized by

$$\begin{aligned}
{}^{B}_{G}\dot{\bar{q}}(t) &= \frac{1}{2}\boldsymbol{\Omega}\Big(\mathbf{C}_q \cdot (\boldsymbol{\omega} - \boldsymbol{\omega}_G)\Big)\,{}^{B}_{G}\bar{q}(t) \\
\dot{\mathbf{b}}_{\mathbf{g}} &= \mathbf{n}_{\mathbf{rg}} \\
{}^{G}\dot{\mathbf{v}} &= \mathbf{a} - 2\lfloor \boldsymbol{\omega}_G \times \rfloor\, {}^{G}\mathbf{v} - \lfloor \boldsymbol{\omega}_G \times \rfloor^2\, {}^{G}\mathbf{p} \\
\dot{\mathbf{b}}_{\mathbf{a}} &= \mathbf{n}_{\mathbf{ra}} \\
{}^{G}\dot{\mathbf{p}} &= {}^{G}\mathbf{v}
\end{aligned} \tag{2}$$

The vehicle's (inertial) rotational velocity $\boldsymbol{\omega}$ and acceleration $\mathbf{a}$ are measured by the IMU as $\boldsymbol{\omega}_m = \mathbf{C}_q\boldsymbol{\omega} + \mathbf{b}_{\mathbf{g}} + \mathbf{n}_{\mathbf{g}}$ and $\mathbf{a}_m = \mathbf{C}_q(\mathbf{a} - {}^{G}\mathbf{g}) + \mathbf{b}_{\mathbf{a}} + \mathbf{n}_{\mathbf{a}}$. The gravitational acceleration, ${}^{G}\mathbf{g}$, can be obtained, for example, from a spherical harmonics model of the gravity potential (Chatfield, 1997). Further symbols used in Eq. (2) are the matrix operator $\boldsymbol{\Omega}(\boldsymbol{\omega})$, and the cross product skew-symmetric matrix, $\lfloor \boldsymbol{\omega} \times \rfloor$:

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} -\lfloor \boldsymbol{\omega} \times \rfloor & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^{\mathrm{T}} & 0 \end{bmatrix}, \text{and } \lfloor \boldsymbol{\omega} \times \rfloor = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

The equations for state prediction are obtained by computing the expected value of Eq. (2). Letting "ˆ" denote estimated quantities, we can write

$$\begin{aligned}
{}^{B}_{G}\dot{\hat{\bar{q}}}(t) &= \frac{1}{2}\boldsymbol{\Omega}(\hat{\boldsymbol{\omega}})\,{}^{B}_{G}\hat{\bar{q}}(t) \\
\dot{\hat{\mathbf{b}}}_{\mathbf{g}} &= \mathbf{0}_{3\times 1} \\
{}^{G}\dot{\hat{\mathbf{v}}} &= \mathbf{C}_{\hat{q}}^{\mathrm{T}} \cdot \hat{\mathbf{a}} + {}^{G}\mathbf{g} - 2\lfloor \boldsymbol{\omega}_G \times \rfloor^{G}\hat{\mathbf{v}} - \lfloor \boldsymbol{\omega}_G \times \rfloor^{2G}\hat{\mathbf{p}} \\
\dot{\hat{\mathbf{b}}}_{\mathbf{a}} &= \mathbf{0}_{3\times 1} \\
{}^{G}\dot{\hat{\mathbf{p}}} &= {}^{G}\hat{\mathbf{v}}
\end{aligned} \tag{3}$$

with $\hat{\boldsymbol{\omega}} = \boldsymbol{\omega}_m - \hat{\mathbf{b}}_{\mathbf{g}} - \mathbf{C}_{\hat{q}} \cdot \boldsymbol{\omega}_G$ and $\hat{\mathbf{a}} = \mathbf{a}_m - \hat{\mathbf{b}}_{\mathbf{a}}$.

Propagation of the state estimate is carried out using Runge-Kutta numerical integration.

3.1.2 Error-State Model & Covariance Propagation

In order to describe the error between the true and estimated value of a quantity, we use a standard additive error model for all quantities except the quaternion, for which a multiplicative error model is employed (Lefferts et al., 1982). The attitude error is modeled as the infinitesimal rotation that causes true and estimated attitude to match. In quaternion algebra, this is expressed as

$$\bar{q} = \delta\bar{q} \otimes \hat{\bar{q}} \Leftrightarrow \delta\bar{q} = \bar{q} \otimes \hat{\bar{q}}^{-1} \tag{4}$$

Since orientation errors are small, the following small angle approximation can be applied

$$\delta\bar{q} \simeq \begin{bmatrix} \delta\mathbf{q} \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\theta} \\ 1 \end{bmatrix} \tag{5}$$

We can see from Eq. (5) that the error information is contained primarily in the tilt angle vector $\delta\boldsymbol{\theta}_{3\times1}$. We can therefore characterize the attitude uncertainty by the covariance matrix of the 3×1 tilt angle vector instead of the full 4×1 quaternion. The resulting attitude covariance matrix is thus of dimension 3×3, and does not suffer the loss of rank that would arise in a 4×4 covariance matrix due to the unit quaternion constraint.

The continuous-time linearized dynamics for the error state $\tilde{\mathbf{x}} = \begin{bmatrix}\delta\boldsymbol{\theta}^{\mathrm{T}} & \tilde{\mathbf{b}}_{\mathbf{g}}^{\mathrm{T}} & {}^{G}\tilde{\mathbf{v}}^{\mathrm{T}} & \tilde{\mathbf{b}}_{\mathbf{a}}^{\mathrm{T}} & {}^{G}\tilde{\mathbf{p}}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$ can be written as

$$\dot{\tilde{\mathbf{x}}} = \mathbf{F}_c \cdot \tilde{\mathbf{x}} + \mathbf{G}_c \cdot \mathbf{n} \tag{6}$$

where

$$\mathbf{F}_c = \begin{bmatrix} -\lfloor\hat{\boldsymbol{\omega}}\times\rfloor & -\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ -\mathbf{C}_{\hat{q}}^{\mathrm{T}}\lfloor\hat{\mathbf{a}}\times\rfloor & \mathbf{0}_{3\times3} & -2\lfloor\boldsymbol{\omega}_G\times\rfloor & -\mathbf{C}_{\hat{q}}^{\mathrm{T}} & -\lfloor\boldsymbol{\omega}_G\times\rfloor^2 \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix}$$

$$\mathbf{G}_c = \begin{bmatrix} -\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{C}_{\hat{q}}^{\mathrm{T}} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix}$$

are the continuous-time state transition matrix and the input noise matrix, and $\mathbf{n} = \begin{bmatrix}\mathbf{n_g}^{\mathrm{T}} & \mathbf{n_{rg}}^{\mathrm{T}} & \mathbf{n_a}^{\mathrm{T}} & \mathbf{n_{ra}}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$ denotes the system noise vector. In particular, the zero-mean continuous-time noise vector has autocorrelation $E[\mathbf{n}(t)\mathbf{n}(\tau)^{\mathrm{T}}] = \mathbf{Q}_c \cdot \delta(t-\tau)$, where $\delta(t-\tau)$ denotes the Dirac delta function and $\mathbf{Q}_c$ is known from the IMU specifications.

In order to implement the discrete-time form of the covariance propagation equation of the EKF, we need to discretize the error-state equation (Eq. (6)) as

$$\tilde{\mathbf{x}}_{k+1} = \boldsymbol{\Phi}_k \cdot \tilde{\mathbf{x}}_k + \mathbf{n}_d \tag{7}$$

This process requires determining the discrete-time state transition matrix $\boldsymbol{\Phi}_k$, as well as the discrete-time system noise covariance matrix $\mathbf{Q}_d = E[\mathbf{n}_d\mathbf{n}_d^{\mathrm{T}}]$. The state transition matrix can be computed as

$$\boldsymbol{\Phi}_k = \exp\left(\int_{t_k}^{t_{k+1}} \mathbf{F}_c(\tau)\delta\tau\right) \tag{8}$$

while the discrete-time noise covariance matrix $\mathbf{Q}_d$ is calculated according to

$$\mathbf{Q}_d = \int_{t_k}^{t_{k+1}} \boldsymbol{\Phi}(t_{k+1},\tau)\mathbf{G}_c(\tau)\mathbf{Q}_c\mathbf{G}_c^{\mathrm{T}}(\tau)\boldsymbol{\Phi}^{\mathrm{T}}(t_{k+1},\tau)\,\mathrm{d}\tau$$

Following the regular EKF equations (Maybeck, 1979), we can now compute the covariance of the propagated state estimate as

$$\mathbf{P}_{k+1|k} = \boldsymbol{\Phi}_k\mathbf{P}_{k|k}\boldsymbol{\Phi}_k^{\mathrm{T}} + \mathbf{Q}_d \tag{9}$$

where $\mathbf{P}_{\ell|k}$ denotes the discrete-time covariance at time step t_ℓ based on image measurements up to time step t_k.

3.2 Update Model for Camera Observations of MLs

As with any dead-reckoning technique, the estimation errors incurred by exclusive propagation of IMU measurements will grow without bounds. In order to limit this error, camera observations to known landmarks are employed to update the state estimates.

3.2.1 Feature Extraction and Matching

Several techniques have been proposed for feature extraction from images for planetary landing applications. Since this is not the main focus of this paper, we will only briefly discuss the most pertinent approaches and kindly refer the interested reader to the literature.

Cheng *et al.* (Cheng et al., 2003; Cheng and Ansar, 2005) propose using craters as landmarks for navigation, because their shape follows a well-known geometric model, allowing robust detection under a wide range of illumination conditions. Illumination invariance is a particularly desirable property for planetary landing, because there is no guarantee that the illumination conditions during descent imaging will be the same as those when the orbital map imagery was acquired. Matching is carried out using conic invariants, making this approach independent of a prior estimate of position or attitude, although knowledge of either quantity enhances speed and robustness. The main disadvantage of using craters as MLs is that there are places on Mars and other solar-system bodies where craters are not present.

When craters are not present, a more general landmark type is required, such as SIFT (Lowe, 2004) keypoints. All that is required to generate these features is unique image texture, and, contrary to crater detection, a model of terrain shape is not required or used. SIFT keypoints share the scale invariance properties of crater landmarks, but they have been demonstrated to be sensitive to illumination variations between the descent image and map (Ansar, 2004). Similar to craters, SIFT keypoints can be matched without knowledge of the spacecraft altitude or orientation relative to the surface map. Although the extraction of SIFT features is computationally expensive, it lends itself to implementation on specialized hardware, such as FPGAs (Barfoot, 2005) and GPUs (Sinha et al., 2006), resulting in a significant increase in processing speed. This type of features was used in this work for detecting MLs in the PDT images (cf. Fig. 4).

Planetary landers often have an altimeter and gyroscopes that can provide on-board measurements of altitude and surface-relative attitude (e.g., by propagating the initial, pre-entry attitude estimate acquired from celestial observations via gyro integration). Based on these measurements, descent images can be warped to a local level coordinate system so that 2D correlation can be applied to find matches between the descent image and an *a priori* visual map of the landing site. This area-correlation approach has been applied successfully to guide cruise missiles (Carr and Sobek, 1980) and is now being considered for planetary landers (Head et al., 2005). The advantage of the area-correlation method compared to the SIFT and crater approaches is that it is based on systems already proven in the field and it can be easily implemented in DSP or FPGA chips, thereby enabling efficient processing. Since the filtering framework readily provides initial pose estimates, area correlation presents a viable method to obtain measurements of MLs during EDL.

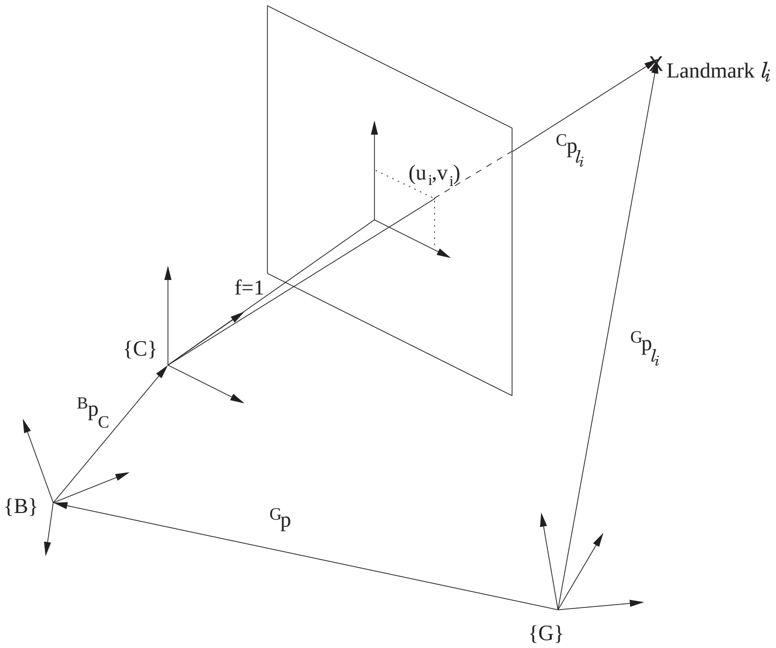


Figure 2: Spatial relationship between the global frame $\{G\}$, the spacecraft body frame $\{B\}$, the camera frame $\{C\}$, and the landmark ℓ_i.

3.2.2 Measurement Model

We hereafter address the case where a set of ground features ℓ_i are extracted from a descent image and matched to a visual map of the area, using any of the methods discussed in the previous section. Hence, the position, ${}^{G}\mathbf{p}_{\ell_i}$, of each of these MLs is considered known in global coordinates. For clarity of presentation, we employ a perspective projection model, assuming a fully calibrated camera (cf. Fig. 2). This can be readily adapted to other projective models.

The vector from the camera to the landmark, ${}^{C}\mathbf{p}_{\ell_i}$, can be expressed in terms of the current spacecraft pose,$\left({}^{G}\mathbf{p}, \mathbf{C}_q\right)$, the known landmark position, ${}^{G}\mathbf{p}_{\ell_i}$, and the known transformation between body frame and camera frame, $\left({}^{B}\mathbf{p}_C, {}^{B}_{C}\mathbf{C}(\bar{q}_C)\right)$, as[3]

$$ {}^{C}\mathbf{p}_{\ell_i} = \mathbf{C}_{q_C}^{\mathrm{T}}\Big(\mathbf{C}_q({}^{G}\mathbf{p}_{\ell_i} - {}^{G}\mathbf{p}) - {}^{B}\mathbf{p}_C\Big) \tag{10} $$

The measurement $\mathbf{z} = [u_i \ v_i]^{\mathrm{T}}$ consists of the normalized image coordinates of the landmark ℓ_i, whose measurement model can be written as

$$ \mathbf{z} = \mathbf{\Pi}\frac{{}^{C}\mathbf{p}_{\ell_i}}{\begin{bmatrix}0 & 0 & 1\end{bmatrix}\cdot {}^{C}\mathbf{p}_{\ell_i}} + \boldsymbol{\eta}' \tag{11} $$

where $\mathbf{\Pi}$ is the projection matrix

$$ \mathbf{\Pi} = \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix}, \tag{12} $$

[3] The matrix ${}^{B}_{C}\mathbf{C}(\bar{q}_C)$ will from here on be written as $\mathbf{C}_{q_C}$ for brevity.

$\boldsymbol{\eta}'$ is the noise term that represents the errors in feature detection. Additionally, it encapsulates errors due to the limited accuracy of the map and of the alignment (coordinate transformation) between the camera and the IMU. These last two types of errors can be either estimated in real time or compensated for by covariance inflation (Maybeck, 1979), as shown in what follows.

The expected measurement, $\hat{\mathbf{z}}$, is formed using the current state estimate as

$$\hat{\mathbf{z}} = \boldsymbol{\Pi}\frac{{}^C\hat{\mathbf{p}}_{\ell_i}}{\begin{bmatrix}0 & 0 & 1\end{bmatrix}\cdot {}^C\hat{\mathbf{p}}_{\ell_i}} \tag{13}$$

with the estimate of the vector from camera to landmark given by

$${}^C\hat{\mathbf{p}}_{\ell_i} = \mathbf{C}^{\mathrm{T}}_{\hat{\bar{q}}_C}\Big(\mathbf{C}_{\hat{\bar{q}}}({}^G\hat{\mathbf{p}}_{\ell_i} - {}^G\hat{\mathbf{p}}) - {}^B\hat{\mathbf{p}}_C\Big) \tag{14}$$

Let us assume that the relative transformation between camera and IMU is uncertain

$${}^B\mathbf{p}_C = {}^B\hat{\mathbf{p}}_C + {}^B\tilde{\mathbf{p}}_C \tag{15}$$

$$\bar{q}_C = \delta\bar{q}_C \otimes \hat{\bar{q}}_C \tag{16}$$

and define the corresponding alignment error vector as $\tilde{\mathbf{x}}_{BC} = \begin{bmatrix}\delta\boldsymbol{\theta}_C^{\mathrm{T}} & {}^B\tilde{\mathbf{p}}_C^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$. We also define the map error[4]

$${}^G\tilde{\mathbf{p}}_{\ell_i} = {}^G\mathbf{p}_{\ell_i} - {}^G\hat{\mathbf{p}}_{\ell_i} \tag{17}$$

and form a linearized approximation of the measurement residual, $\mathbf{r} = \mathbf{z} - \hat{\mathbf{z}}$ as

$$\mathbf{r} = \mathbf{z} - \hat{\mathbf{z}} \simeq \mathbf{H}\tilde{\mathbf{x}} + \mathbf{H}_{BC}\tilde{\mathbf{x}}_{BC} + \mathbf{H}_{\ell_i}{}^G\tilde{\mathbf{p}}_{\ell_i} + \boldsymbol{\eta}' \tag{18}$$

$$= \mathbf{H}\tilde{\mathbf{x}} + \boldsymbol{\eta} \tag{19}$$

with

$$\mathbf{H} = \mathbf{H}_C \cdot \mathbf{C}^{\mathrm{T}}_{\hat{\bar{q}}_C}\begin{bmatrix}\lfloor \mathbf{C}_{\hat{\bar{q}}}({}^G\hat{\mathbf{p}}_{\ell_i} - {}^G\hat{\mathbf{p}})\times\rfloor & \mathbf{0}_{3\times 9} & -\mathbf{C}_{\hat{\bar{q}}}\end{bmatrix} \tag{20}$$

$$\mathbf{H}_{BC} = \mathbf{H}_C \cdot \mathbf{C}^{\mathrm{T}}_{\hat{\bar{q}}_C}\begin{bmatrix}-\lfloor \mathbf{C}_{\hat{\bar{q}}}({}^G\hat{\mathbf{p}}_{\ell_i} - {}^G\hat{\mathbf{p}}) - {}^B\hat{\mathbf{p}}_C\times\rfloor & -\mathbf{I}_{3\times 3}\end{bmatrix} \tag{21}$$

$$\mathbf{H}_{\ell_i} = \mathbf{H}_C \cdot \mathbf{C}^{\mathrm{T}}_{\hat{\bar{q}}_C}\mathbf{C}_{\hat{\bar{q}}} \tag{22}$$

where $\mathbf{H}$ is the measurement matrix, $\mathbf{H}_{BC}$ is the Jacobian with respect to the error in the transformation between the camera and the IMU frames, $\mathbf{H}_{\ell_i}$ is the Jacobian with respect to the map error, and finally $\mathbf{H}_C$ denotes the Jacobian with respect to the feature vector in camera coordinates

$$\mathbf{H}_C = \frac{1}{\begin{bmatrix}0 & 0 & 1\end{bmatrix}\cdot {}^C\hat{\mathbf{p}}_{\ell_i}}\begin{bmatrix}\mathbf{I}_{2\times 2} & -\hat{\mathbf{z}}\end{bmatrix} \tag{23}$$

The noise vector $\boldsymbol{\eta}$ comprises the normalized pixel noise $\boldsymbol{\eta}'$, assumed to be zero-mean, white Gaussian noise with covariance $\mathbf{R}'$, the effects of errors $\tilde{\mathbf{x}}_{BC}$ due to misalignment, as well as those due to map errors ${}^G\tilde{\mathbf{p}}_{\ell_i}$. The noise term $\boldsymbol{\eta}$ can then be treated as noise with covariance

$$\begin{aligned}\mathbf{R} =& E[\boldsymbol{\eta}\boldsymbol{\eta}^{\mathrm{T}}] \qquad (24)\\ =& \mathbf{H}_{\ell_i}E[{}^G\tilde{\mathbf{p}}_{\ell_i}{}^G\tilde{\mathbf{p}}_{\ell_i}^{\mathrm{T}}]\mathbf{H}_{\ell_i}^{\mathrm{T}} + \mathbf{H}_{BC}E[\tilde{\mathbf{x}}_{BC}\tilde{\mathbf{x}}_{BC}^{\mathrm{T}}]\mathbf{H}_{BC}^{\mathrm{T}} + \mathbf{R}'\end{aligned}$$

where the first two terms correspond to the covariance inflation components mentioned previously.

[4] In order to account for uncertainties in the map, there are generally two possible solutions. The first is to estimate the ML positions online and include them in the state vector (e.g., (Kim and Sukkarieh, 2004)), a process also known as Simultaneous Localization and Mapping (SLAM). Due to its high computational complexity, we have opted to avoid SLAM in spacecraft EDL. Alternatively, and similar to the treatment of the alignment errors, in this work we compensate for map uncertainty by covariance inflation.

3.2.3 Iterated Extended Kalman Filter Update

In order to better address linearization errors, we employ the Iterated Extended Kalman Filter (IEKF) (Maybeck, 1979) to update the state.

The iterative scheme proceeds as follows:

1. Compute $\hat{\mathbf{z}}$ as a function of the current j-th iterate $\hat{\mathbf{x}}_{k+1|k+1}^{(j)}$ using Eq. (13).
2. Linearize the measurement model around the current iterate to obtain $\mathbf{H}^{(j)}$ (Eq. (20)), $\mathbf{H}_{BC}^{(j)}$ (Eq. (21)), and $\mathbf{H}_{\ell_i}^{(j)}$ (Eq. (22)).
3. Form the residual $\mathbf{r}^{(j)} = \mathbf{z} - \hat{\mathbf{z}}^{(j)}$, and compute its covariance $\mathbf{S}^{(j)}$, (using Eq. (24)), according to
$$\mathbf{S}^{(j)} = \mathbf{H}^{(j)}\mathbf{P}_{k+1|k}\mathbf{H}^{(j)\,\mathrm{T}} + \mathbf{R}^{(j)} \tag{25}$$
4. Using the Kalman gain $\mathbf{K}^{(j)}$
$$\mathbf{K}^{(j)} = \mathbf{P}_{k+1|k}\mathbf{H}^{(j)\,\mathrm{T}}\,(\mathbf{S}^{(j)})^{-1} \tag{26}$$
compute the correction
$$\Delta\mathbf{x}^{(j)} = \begin{bmatrix} \delta\hat{\boldsymbol{\theta}}_{k+1} \\ \Delta\hat{\mathbf{b}}_{\mathbf{g}\,k+1} \\ \Delta^{G}\hat{\mathbf{v}}_{k+1} \\ \Delta\hat{\mathbf{b}}_{\mathbf{a}\,k+1} \\ \Delta^{G}\hat{\mathbf{p}}_{k+1} \end{bmatrix}^{(j)} = \mathbf{K}^{(j)}(\mathbf{r}^{(j)} + \mathbf{H}^{(j)} \cdot \Delta\mathbf{x}^{(j-1)}) \tag{27}$$
5. We can now obtain the next iterate of the updated state estimate, $\hat{\mathbf{x}}_{k+1|k+1}^{(j+1)}$, using the following update procedure:
   (a) The update for the quaternion is given by
$$ {}^{B}_{G}\hat{\bar{q}}_{k+1|k+1}^{(j+1)} \;=\; \delta\hat{\bar{q}}_{k+1}^{(j)} \otimes {}^{B}_{G}\hat{\bar{q}}_{k+1|k} \tag{28}$$
   where
$$\delta\hat{\bar{q}}_{k+1}^{(j)} = \begin{bmatrix} \frac{1}{2}\delta\hat{\boldsymbol{\theta}}_{k+1}^{(j)} \\ \sqrt{1 - \frac{1}{4}\delta\hat{\boldsymbol{\theta}}_{k+1}^{j\,\mathrm{T}}\delta\hat{\boldsymbol{\theta}}_{k+1}^{(j)}} \end{bmatrix}$$
   (b) The update for the remaining elements of the state estimate is simply
$$\begin{bmatrix} \hat{\mathbf{b}}_{\mathbf{g}\,k+1|k+1} \\ {}^{G}\hat{\mathbf{v}}_{k+1|k+1} \\ \hat{\mathbf{b}}_{\mathbf{a}\,k+1|k+1} \\ {}^{G}\hat{\mathbf{p}}_{k+1|k+1} \end{bmatrix}^{(j+1)} = \begin{bmatrix} \hat{\mathbf{b}}_{\mathbf{g}\,k+1|k} \\ {}^{G}\hat{\mathbf{v}}_{k+1|k} \\ \hat{\mathbf{b}}_{\mathbf{a}\,k+1|k} \\ {}^{G}\hat{\mathbf{p}}_{k+1|k} \end{bmatrix} + \begin{bmatrix} \Delta\hat{\mathbf{b}}_{\mathbf{g}\,k+1} \\ \Delta^{G}\hat{\mathbf{v}}_{k+1} \\ \Delta\hat{\mathbf{b}}_{\mathbf{a}\,k+1} \\ \Delta^{G}\hat{\mathbf{p}}_{k+1} \end{bmatrix}^{(j)} \tag{29}$$

This process is iterated to convergence. The iteration is started using the propagated state estimate $\hat{\mathbf{x}}_{k+1|k+1}^{(0)} = \hat{\mathbf{x}}_{k+1|k}$ as zeroth iterate, making the first iteration equivalent to a regular EKF update.

Finally, the covariance matrix for the current state is updated, using the values for $\mathbf{K}$ and $\mathbf{S}$ from the final iteration:

$$\mathbf{P}_{k+1|k+1} = \mathbf{P}_{k+1|k} - \mathbf{KSK}^T \tag{30}$$

3.2.4 Outlier Rejection

In order to detect and reject mismatched MLs, a Mahalanobis distance test is employed. Every time a new measurement $\mathbf{z}$ of a known feature ℓ_i becomes available, the following quantity

$$\mathbf{r}^{(0)\,\mathrm{T}} \left(\mathbf{S}^{(0)}\right)^{-1} \mathbf{r}^{(0)} \tag{31}$$

is computed and compared to a preset threshold γ selected based on the chi-square distribution. A measurement to a landmark ℓ_i is processed by the filter only when it passes this test.

Notice that in purely vision-based localization approaches, a number of more sophisticated techniques are usually needed for outlier rejection. In our case, however, the filtering algorithm makes it possible to have an initial camera-pose estimate based on IMU propagation. Using this estimate, guided feature extraction and matching can be performed, which significantly increases performance and reduces the number of outliers. Moreover, the image processing algorithm usually provides a figure of merit for the probability of each match (e.g., the Euclidean distance between SIFT keys). Based on this criterion, only highly probable matches could be used for pose updates. Finally, if increased outlier protection is required, and the additional computing power is available, a RANSAC-based compatibility test can be applied. In particular, once a set of landmark matches has been determined, an image-based pose estimation algorithm (e.g., (Haralick et al., 1989)), can be employed in conjunction with RANSAC, to identify a subset of measurements that is self-consistent.

All the aforementioned steps can be applied as pre-processing steps, before applying the Mahalanobis-distance gating test. From our experience with the datasets used in this paper however, the Mahalanobis distance test was a very effective method for outlier rejection, and was sufficient on its own to guarantee robust filter performance.

3.2.5 Treating Image Processing Delays

The image processing algorithm for detecting MLs will give rise to a delay between image acquisition and actual use of extracted observations for pose updates. As detailed in (Montgomery et al., 2006; Mourikis et al., 2007), it is possible to appropriately account for latency by state augmentation. In particular, at the point of image acquisition a copy of the current pose is stored in the state vector, which is at the beginning fully correlated with the current state; a process termed “state cloning” (Roumeliotis and Burdick, 2002). As the current state evolves, the clone remains constant, but the cross-correlations between the state at image acquisition and the state at the point of the actual update are properly evaluated. The update is carried out using the cloned state as linearization point, affecting the current state via the cross-correlation terms. After the update, the state clone can be discarded by marginalizing it from the probability density function.

4 Experimental Results

The motion estimation algorithm described in the preceding section was tested using data collected during the MSL subsonic Parachute Drop Test (PDT) (Johnson et al., 2004), carried out in Fall 2004 at the NASA Scientific Balloon Facility in Ft. Sumner, NM. In this experiment, a gondola equipped with an IMU, GPS and camera was lifted to an altitude of approximately 36 km by a balloon, where it was released and descended to Earth on a parachute. Further testing of the algorithm was conducted using data from the 2002 Descent Image Motion Estimation System (DIMES) field tests (Johnson et al., 2005). The primary purpose of these tests was to demonstrate that the velocity estimates produced by the DIMES algorithm (Cheng et al., 2004) would be reasonable when imaging Mars-like terrain at representative altitudes. The nature, however, of the recorded camera and IMU data made them ideal for testing our ML-based pose estimation algorithm. A description of the experimental results follows.

4.1 The MSL Subsonic Parachute Drop Test

4.1.1 Experiment Description

During the MSL subsonic Parachute Drop Test (PDT), an instrumentation package was included on the gondola's payload, in order to collect data suitable for testing pin-point landing algorithms (Fig. 3 shows the experimental setup). The instruments used for our experiment consist of an LN-200 IMU, a Sony XC55 camera, and a Horita GPS-based master video sync generator, allowing high-accuracy time-stamping of the aerial images. During the experiment, the balloon carried the gondola to an altitude of 36 km within a period of about 2 hours. The nadir-pointing camera started operating 2 minutes before separation, recording a time-stamped video stream at 30 frames per second. During the drop, which lasted approximately 14 minutes, the maximum vertical velocity was approximately 120 m/s, while the horizontal velocities were smaller than 20 m/s. The gondola traversed a ground distance of about 5.5 km during its descent. Unfortunately, upon deployment of the parachute, the onboard GPS lost lock of several satellites, and was unable to re-acquire the signals. As a result, no reliable ground-truth for the position estimate is available for the experiment.

The imagery from the onboard camera was processed off-line, using the SIFT algorithm (Lowe, 2004) to extract features from the aerial images, processing only one out of every 30 frames (1 sec interframe time). The resulting SIFT keys were matched against a database of SIFT features with known 3D positions. These MLs were extracted in advance from a map given in the form of an 11×12 km patch of 1 m/pixel resolution, grayscale orthoimagery of the landing area, combined with altitude information from the 1/3" National Elevation Dataset (NED) (U.S. Geological Survey, 2006). A preprocessing step, adjusting the histogram of the images, was applied in order to increase robustness of the feature matching algorithm to illumination changes that occurred during the drop. Further, to adjust for the differences in scale, between 8 km and 1.5 km altitude, the original map was downsampled to 8 m/pixel, whereas for the final descent the map was used at its full resolution, but cropped to 1×1 km. The difference in scale and resolution results in a changing number of

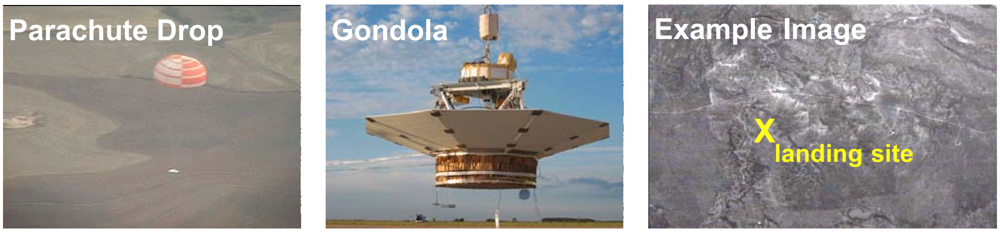


Figure 3: Pin-Point Landing Experiment on the NASA MSL Subsonic Parachute Drop Test (PDT). The gondola (center) was attached to a balloon that carried it to approximately 36 km altitude and then released it for descent on a parachute (left). During the descent, a nadir-pointing camera recorded images of the landing site (right), and an IMU provided measurements of acceleration and rotational velocity.

ML matches, as seen in Fig. 6(d). Figure 4 shows the resulting matches between an image taken at an altitude above ground of approximately 3.5 km and the map of the landing area.

4.1.2 Description of Results

The operation of the filter began 6 minutes into the drop, at an altitude of approximately 8 km above the ground, when a sufficient number of ML matches were found. Since, due to the GPS failure, no prior state estimates were available, an initial pose estimate was obtained using a nonlinear least-squares minimization algorithm (Haralick et al., 1989). Similarly, by computing the difference of the image-only position estimates from two successive images, an initial estimate of the velocity is obtained. In Fig. 5, the estimated trajectory during the drop is shown. In this plot, the z axis represents altitude above the reference ellipsoid, while the x and y axes represent latitude and longitude, respectively. Fig. 6 shows the standard deviation of the estimated attitude, velocity, and position, as computed by the filter. The horizontal axis corresponds to altitude above ground level (AGL) in these plots. We note that in the initial moments after the EKF begins processing images, the covariance of the estimates (especially for the position and velocity) undergoes a large reduction. For this reason, in the plots of Figs. 6(b) and 6(c), the scale of the axes has been adjusted, so that the accuracy during the largest portion of the trajectory (i.e., the last approximately 7 km) is clearly visible. It should be pointed out that at touchdown, the computed standard deviation of the errors is less than 1 m for position, and less than 0.05° for orientation, in all axes. This impressive accuracy is due to the fact that SIFT keys can be reliably matched until approximately 75 m AGL (cf. Fig. 6(d)).

In order to compare the estimation accuracy obtained by the proposed EKF estimator to that attainable by image-based pose estimation, in Fig. 7 we plot the standard deviation of the attitude and position errors for the two approaches. The dashed lines represent the standard deviation of the estimates computed using the camera observations only (cf. (Haralick et al., 1989)), while the solid lines represent the standard deviation of the EKF estimate, at the time instant of the corresponding image. It becomes apparent that combining visual measurements with inertial information results in substantial improvement of the accuracy of the pose estimates. Moreover, it should be pointed out that using a vision-only approach, we are able to obtain estimates *only* for the time instants when the images are recorded, and

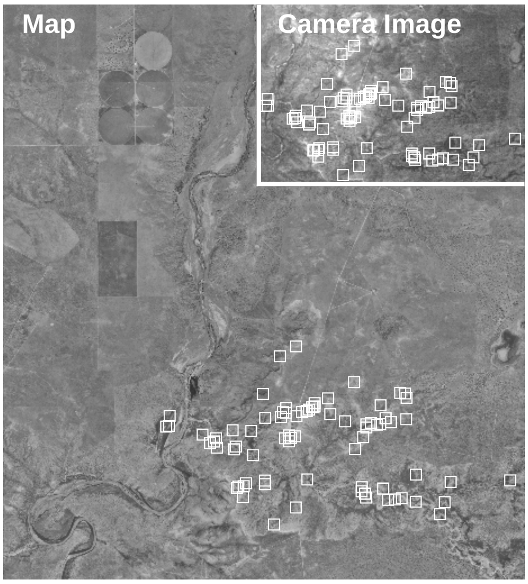


Figure 4: Matched SIFT features between an aerial image from the Parachute Drop Test and a map of the landing area, given in the form of an 11×12 km patch of grayscale orthoimagery. The image was taken at an altitude of approximately 3.5 km above ground.

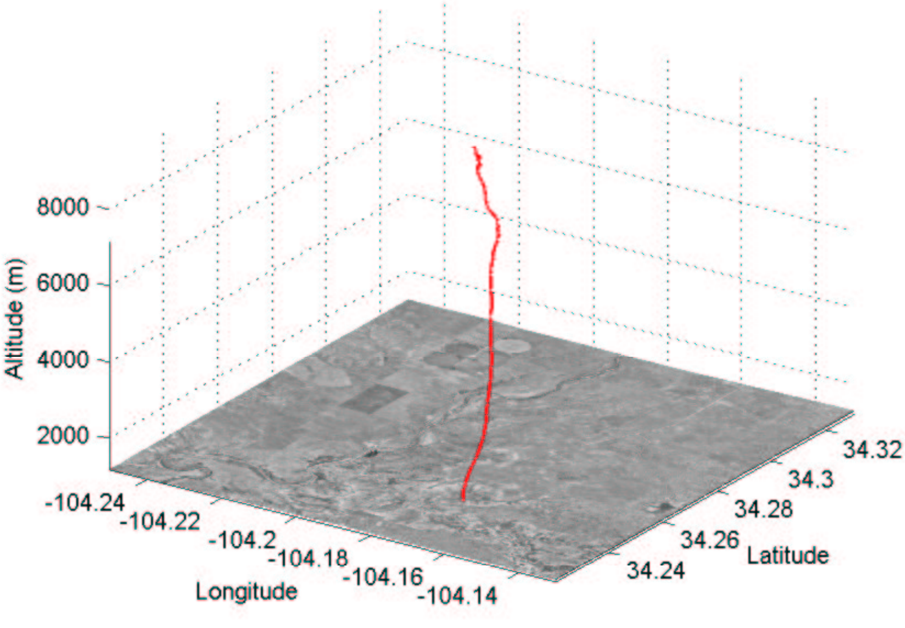
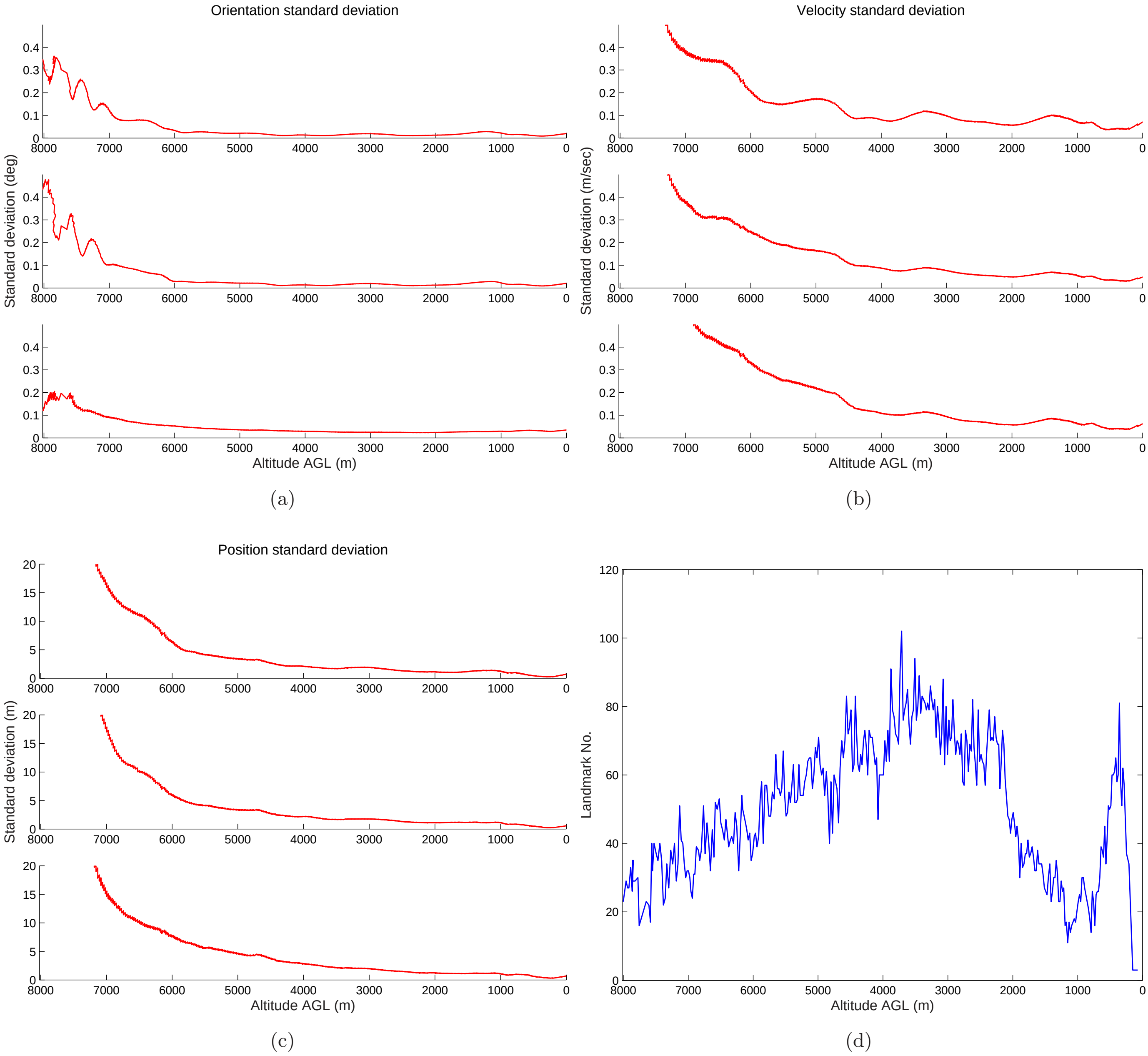


Figure 5: The estimated trajectory during the PDT experiment. The trajectory is overlaid on a map of the area.



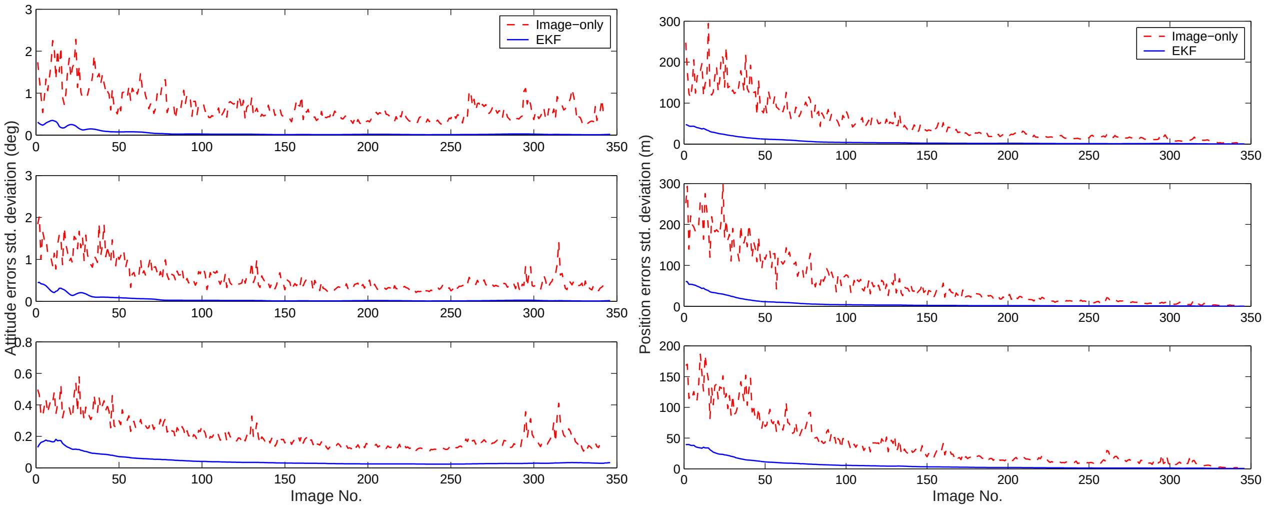
(a)

(b)

(c)

(d)

Figure 6: Filter uncertainty as a function of altitude above ground level (AGL) during the PDT descent. (a) Attitude standard deviation. (b) Velocity standard deviation. (c) Position standard deviation. In all figures, the 3 sub-plots correspond to the x-, y-, and z-axes, respectively. (d) The number of correctly matched SIFT keypoints.



(a)

(b)

Figure 7: Comparison of the uncertainty in the image-only estimates (dashed lines) vs. that in the EKF estimates (solid lines). (a) Attitude standard deviation. (b) Position standard deviation. In these figures, the 3 sub-plots correspond to the x-, y-, and z-axes, respectively.

moreover, no velocity estimates can be directly computed. In contrast to these limitations, with the proposed EKF-based approach attitude, position, and velocity estimates are available at the rate of the IMU measurements (100 Hz in this experiment), and thus they can be employed for motion control during a lander's descent.

4.2 DIMES field test

4.2.1 Experiment Description

Due to the lack of reliable ground truth in the MSL experiment, we further tested the algorithm on datasets acquired onboard a helicopter equipped with a gimballed camera/IMU platform during validation of the Descent Image Motion Estimation System. The flight prototype sensors used in the DIMES field test system consisted of a camera that was form, fit, and function equivalent to the cameras employed on the Mars Exploration Rover mission spacecraft, and an IMU that was from the same product line as the one used on the spacecraft. The IMU data was available at 400 Hz, while the camera recorded one image every 1.7 sec. The camera and IMU were rigidly mounted on a 2-axis gimbal platform, which was then mounted to a 3-axis stabilized platform attached to the front of a helicopter. Ground support equipment was developed, including a data acquisition and control system to command the 2-axis gimbal, and to log field-test data. The rigid transformation between the camera and IMU was determined during pre-flight camera calibration.

The helicopter position and timing data were recorded using an Ashtech GPS Z-Reference Station. This stand-alone unit logs data internally and was not integrated into the support electronics, except for power. The GPS data are employed for providing

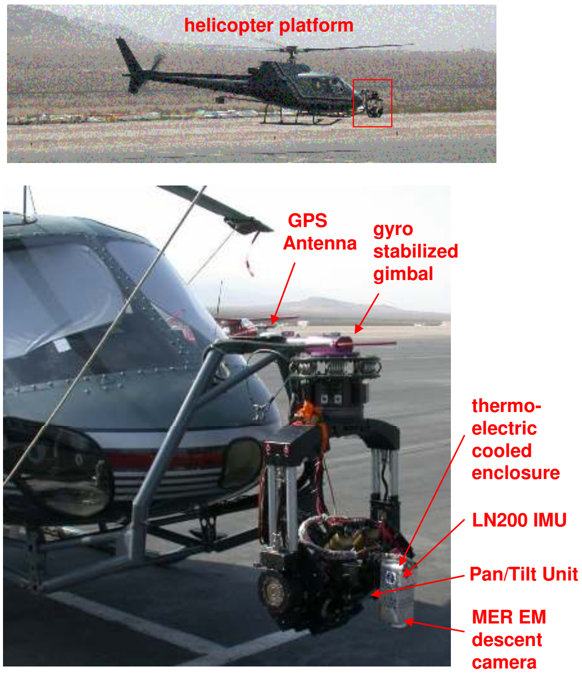


Figure 8: The sensor payload used for the DIMES field testing.

ground truth position estimates; for this purpose, the data were post-processed at JPL using the GIPSY-OASIS II (GOA II) software package, an ultra high-precision GPS satellite orbit determination process providing sub-centimeter position accuracy on the ground (http://gipsy.jpl.nasa.gov/orms/goa/). For maximum reception, an omni-directional GPS antenna was mounted above the sensor payload. A picture of the equipment is given in Fig. 8.

During a typical flight, the helicopter takes off and flies to the test site while climbing to an altitude of 1000 m. While constantly gaining in altitude, it flies back and forth over the terrain in a zigzag pattern along a fixed horizontal line; the typical velocities are 30 m/s horizontal and 1 to 2 m/s vertical. During this time, the camera operator points the gimbal to avoid imaging undesirable terrain. The run ends when the helicopter reaches an altitude of 2000 m. During each one of these runs the pan/tilt can be activated to obtain attitude rates up to 60 °/sec and off-nadir angles up to 45°. Except for vertical velocity, the field test covered the dynamic range expected during spacecraft landing on Mars.

To form an attitude reference for the DIMES algorithm, eleven square targets (1 m edges) were placed on the ground, and a detailed GPS survey determined the latitude, longitude, and altitude of each of the targets. The area containing all of the targets was imaged during a target flyover at the beginning and end of each flight; typically less than 2 minutes of imaging data were acquired for each pass over the targets. An example image is shown in Fig. 9. The centroids of the targets were manually extracted from the images and then matched to the GPS positions from the target survey. For our experiments, these targets were used as MLs.

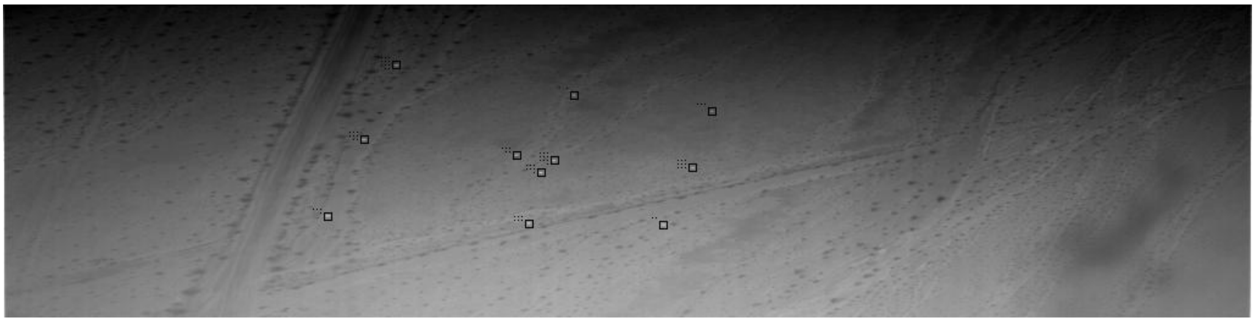


Figure 9: A sample image recorded by the camera, with the detected targets denoted by black squares.

4.2.2 Description of Results

We hereafter report the results of our Extended Kalman Filter (EKF)-based motion estimation algorithm on the data collected during the DIMES field test. Four datasets are available; the first two were recorded at the Ivanpah Dry Lake Bed site, while the remaining two originate from the Pisgah Lava Flow site[5]. During each run, we initialized the filter by processing the ML observations in the first image. The initial 3D pose is estimated using a nonlinear least-squares minimization algorithm (Haralick et al., 1989). This approach was followed in this case, since no other prior estimate was available (using the GPS would not be realistic). We should note, however, that in an actual EDL scenario, the initial estimates for the filter state will be provided from the pre-entry estimation data. After initialization, the filter propagates the state and covariance estimates based on the IMU signals, while it updates these quantities intermittently every time image measurements become available.

In Table 1, the RMS, maximum, and final position errors for all four datasets are shown. The errors reported in this table correspond to the difference between the GPS position data and the EKF estimates. It should be noted that the GPS antenna, which is the reference point for the ground-truth data, is located approximately 1.5 m away from the IMU, whose position is estimated by the EKF. The transformation between the GPS and IMU frames is time-varying, depending on the gimbal angles. Unfortunately, the gimbal angle data were not recorded during the helicopter flights, making it impossible to compensate for the displacement between the GPS antenna and the IMU enclosure. Therefore, the errors reported in the table include the unknown displacement between the GPS and IMU, which implies that the actual filter errors are expected to be *smaller* than the values shown in Table 1.

In Fig. 10, the trajectory of the helicopter during the first flyover, which comprises 36 images recorded over a period of approximately 1 minute, is plotted. The dashed red line represents the GPS ground truth estimates, the solid black line denotes the EKF estimates, the dotted blue line shows the trajectory estimate computed by dead reckoning on the IMU signals, and finally the asterisks depict the position estimates derived using only the camera measurements, based on the method described in (Haralick et al., 1989). During

[5] Two additional datasets were recorded at the Kelso Sand Dunes site, but the IMU measurements suffered from significant bias instability, and were unusable.

Table 1: Position errors for the DIMES datasets

Dataset	Duration (sec)	RMS (m)	max (m)	final (m)
Ivanpah 1	60.9	1.54	2.69	2.64
Ivanpah 2	38.7	1.88	2.36	1.12
Pisgah 1	50.6	2.00	2.49	1.89
Pisgah 2	21.5	2.66	3.08	3.08

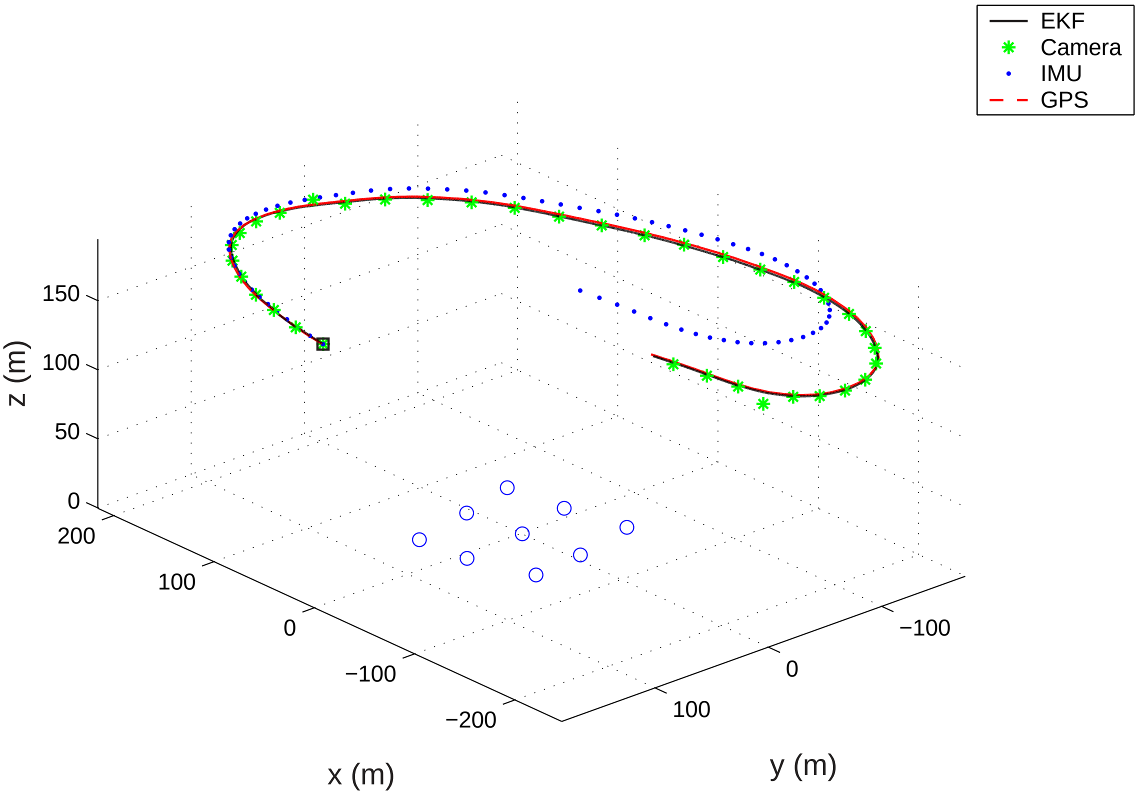
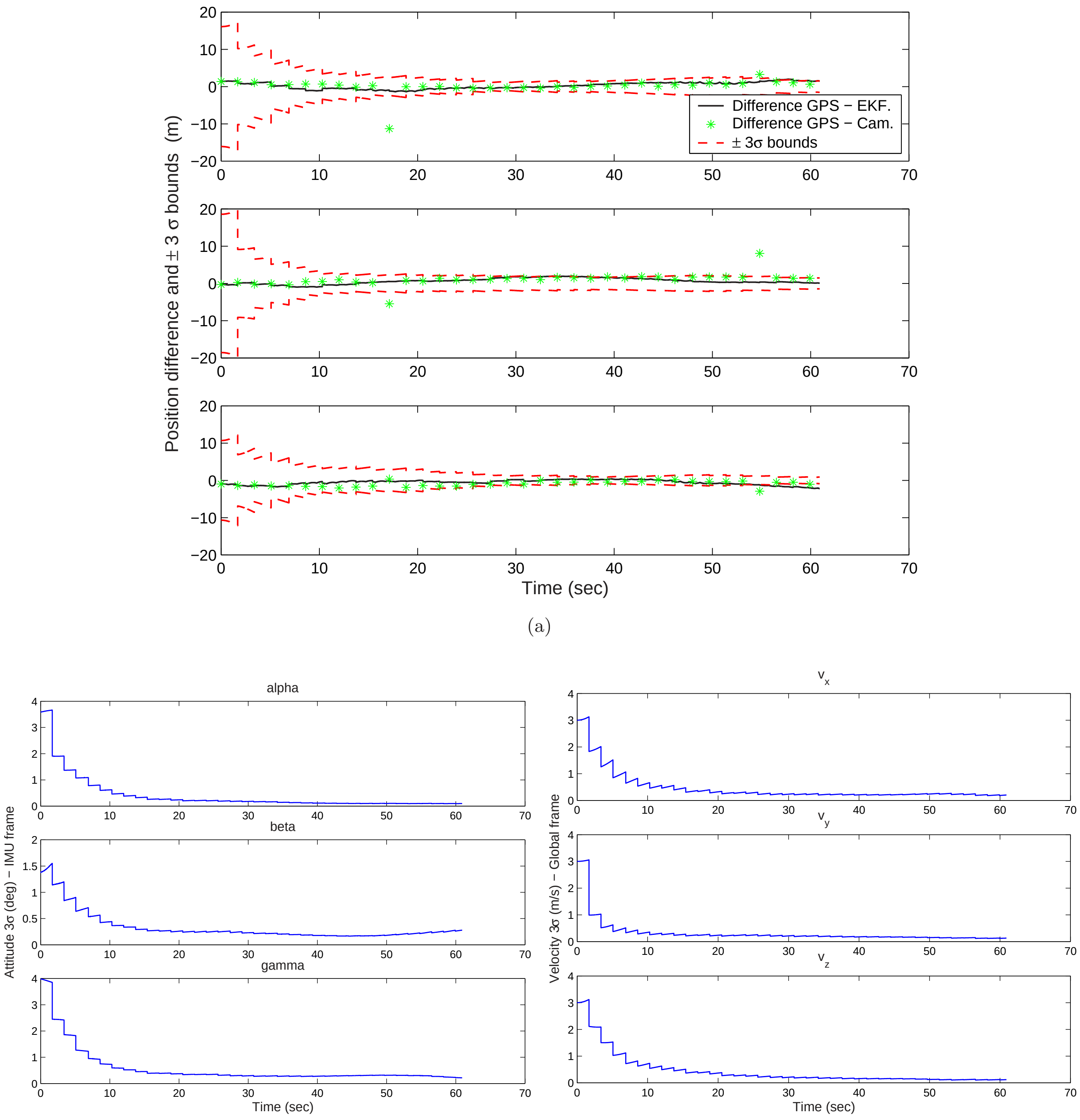


Figure 10: The trajectory of the camera during the first flyover at the Ivanpah site. The start of the trajectory is denoted by a square.

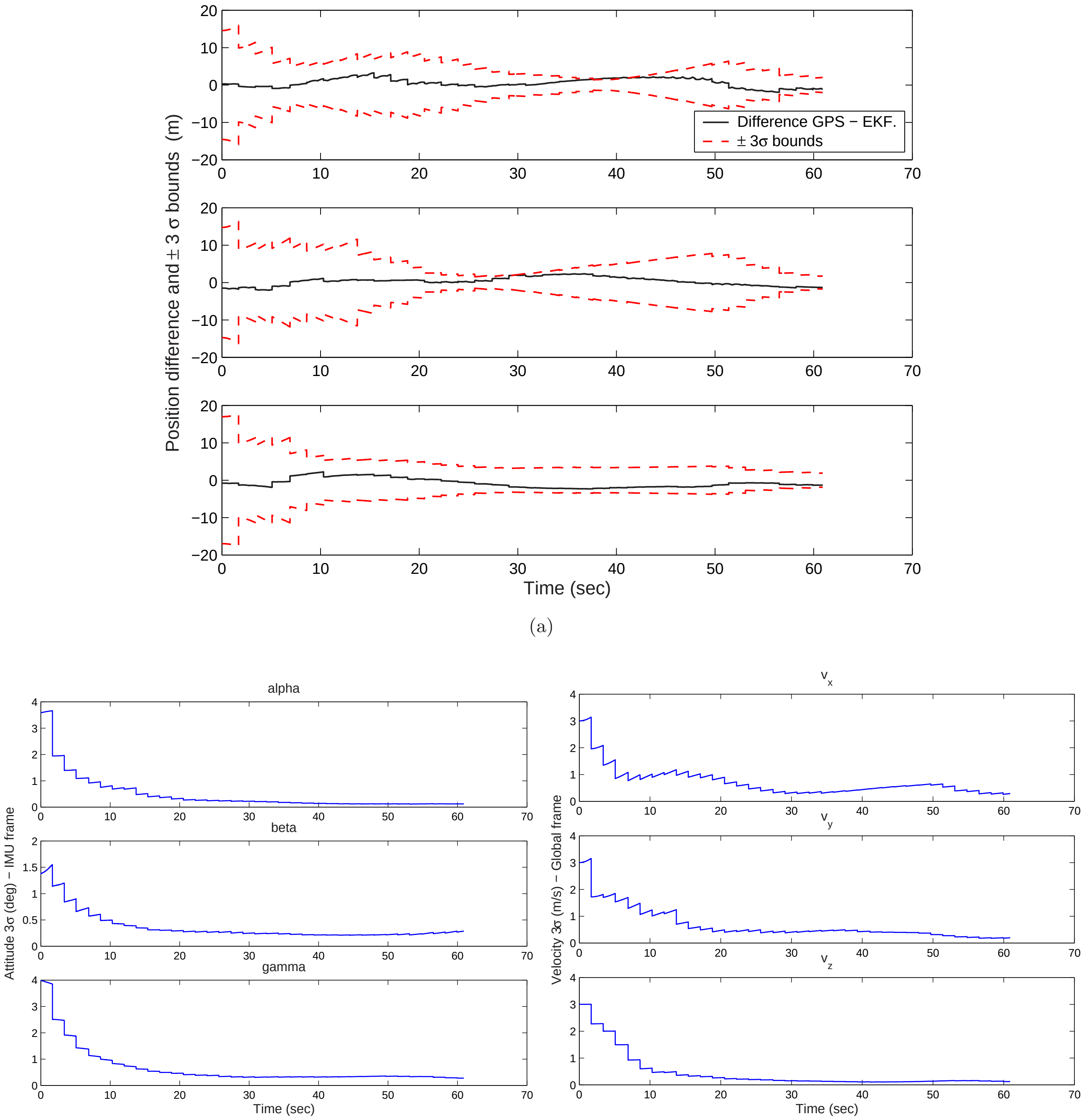


(a)

(b)

(c)

Figure 11: EKF experimental results: (a) The position errors of the filter (solid black line) and the associated ±3σ bounds computed by the EKF. The asterisks denote the camera-only position estimates. Note that the reported errors include the unknown displacement between the GPS antenna and the IMU, whose magnitude is in the same order as the ±3σ values. (b) The 3σ values for the attitude errors computed by the EKF. (c) The 3σ values for the velocity errors computed by the EKF. In all figures, the 3 sub-plots correspond to the x-, y-, and z-axes, respectively.



(a)

(b)

(c)

Figure 12: UT experimental results: (a) The position errors of the filter (solid black line) and the associated $\pm3\sigma$ bounds computed using the UT for updates. (b) The 3σ values for the attitude errors computed using the UT for updates. (c) The 3σ values for the velocity errors computed using the UT for updates. In all figures, the 3 sub-plots correspond to the x-, y-, and z-axes, respectively.

this flyover, the helicopter moved in an approximately circular trajectory above the targets (denoted by blue circles), at almost constant height. Note that the position estimates, for both the targets and body are expressed in Earth-Centered Earth Fixed (ECEF) coordinates in the EKF, but are transformed, for visualization purposes, to a frame centered between the targets (cf. Fig. 10).

In Fig. 11(a), the time evolution of the EKF position errors is shown. The solid line in this plot represents the difference between the EKF and GPS position estimates, while the dashed lines represent the $\pm 3\sigma$ values corresponding to the filter covariance estimates. For comparison, we also plot the errors that would result from camera-only pose estimation using nonlinear least-squares minimization (marked by asterisks). In Figs. 11(b) and 11(c) the 3σ values for the errors in the body attitude and velocity computed by the EKF are shown (no ground truth is available for these quantities).

In the plots of Fig. 11, we observe that the covariance of the filter estimates is initially large, but practically reaches its steady-state value after 15 seconds (i.e., once having processed 10 images). After the initial transient phase, the 3σ values correspond to errors of approximately ± 2 m for position, $\pm 0.5^{o}$ for attitude, and ± 0.2 m/sec for velocity. These errors are *an order of magnitude smaller* than those resulting from dead-reckoning based on the IMU measurements (the RMS error of the IMU-only based estimates for this dataset is 47.8 m, while the maximum error is 135.7 m). The EKF estimates are also more accurate than the camera-only estimates (RMS error 2.2 m, and maximum error 12.6 m), but most importantly, they are available at 400 Hz, i.e., at the IMU sampling rate. We thus see that by combining the positioning information of the IMU and camera, *precise, robust, and high-bandwidth* pose and velocity estimates are obtained, which can be utilized for motion control during EDL.

4.2.3 Alternative Nonlinear Estimators

For the nonlinear estimation problem under consideration, no provably optimal recursive estimator exists. In the absence of theoretical tools, the performance of candidate estimators for a specific nonlinear estimation problem typically has to be assessed experimentally. In our work, we have evaluated the performance of the Particle Filter (PF) (Arulampalam et al., 2002), and the Unscented Transform (UT) (Julier et al., 2000) for performing visual updates in the filter. Due to the large dimension of the state space, unless very large numbers of particles were used, the PF suffered from *particle depletion*. Therefore, the PF was not further considered. Although the UT attained better accuracy than the PF, both methods were outperformed by the EKF estimator. In Fig. 12, we plot the 3σ values for the position, attitude, and velocity that were computed using the UT for updates, for the same dataset as in Fig. 11. Comparing these two sets of plots, we conclude that both the covariance estimates and the errors obtained when using the UT are larger than those produced by the EKF, a result that was typical for all our tests.

We attribute the superior performance of the EKF in this particular problem to the relatively small estimation errors, which result in negligible linearization inaccuracies. In particular, when at least 3 landmarks are visible (which is the case in the DIMES datasets), the pose is fully observable, and thus the state errors do not continuously accumulate. The suitability of the EKF for nonlinear estimation problems where the updates are very accurate has also been previously observed (Lefebvre et al., 2001).

5 Simulations

In order to evaluate the effects of various design parameters on the attainable estimation accuracy, we have conducted extensive simulation tests. We here present results analyzing the accuracy with which the landing coordinates of a spacecraft can be estimated using the presented algorithm. For our simulations, we consider the lander trajectory depicted in Fig. 13, which is similar to the trajectory recorded at a recent EDL test carried out by NASA. The camera begins operating at an altitude of approximately 4 km (i.e., after the deployment of the parachute), and is recording images up to an altitude of 150 m. The camera has a Field Of View (FOV) of 35^o, and focal length equal to 1000 pixels. During the descent, the vertical velocity is approximately 11 m/sec, while the maximum horizontal velocity is approximately 30 m/sec. Our goal is to study the impact of the following factors: (i) the frame rate at which images are recorded, (ii) the accuracy with which MLs are detected, and (iii) the density of MLs on the surface at the landing site.

In Fig. 14, the attitude, velocity, and position errors during a typical descent are shown (solid lines), along with the corresponding $\pm 3\sigma$ values computed by the filter (dashed lines). The horizontal axis in this plot is the lander's altitude above ground level. For this particular trial, one image is recorded every 1 sec, landmarks can be localized in images with a standard deviation of 1 pixel, and the average density of landmarks on the surface is 2 per km^2. In these plots, we observe that the uncertainty of the state estimates is initially large, and becomes smaller as more image measurements are processed. Interestingly, however, as the lander approaches the ground, the covariance of the state estimates increases once again - an effect also visible in the PDT experimental results. This occurs because as the camera moves closer to the ground, fewer known landmarks fall within its field of view, and therefore the positioning information contributed by the images is reduced. For example, for the given camera FOV and landmark density, the expected number of visible landmarks at an altitude of 1 km is 0.75. For this particular trial (cf. Fig. 14), only 2 or fewer MLs are visible in the last 900 m of the descent, 1 or fewer MLs for the last 720 m, and no MLs in the last 610 m. During the time interval when fewer than 3 MLs are visible, the pose of the lander is not observable. The errors, however, in attitude, position and velocity are still small since our estimator is capable of processing even sparse ML observations and reduce the positioning uncertainty when compared to IMU-only propagation. This shows that the ability of our algorithm to process *any number* of landmark measurements (even fewer than 3), is important.

The most computationally intensive part of the algorithm described in this paper is the processing of the images to extract MLs. During EDL, hard real-time constraints exist, and, moreover, computing resources in spacecraft are typically not abundant. For example, current interplanetary spacecraft are equipped with CPUs operating at 30 (RAD6000) up to 300 MIPS (RAD750) (Burcin, 2002). In a system where the available CPU power is limited, the images will be utilized at a low rate, thus affecting the accuracy of state estimation. To study the trade-off between computation, i.e., *inter-frame time*, and *positioning accuracy*, in

Fig. 15 we plot the 3σ values for the position errors along the x- and y-axes, at the time of touchdown, as a function of the time needed for processing each image. For these plots, the standard deviation of the feature detection errors is equal to 1 pixel, the density of landmarks is 2 per km^2, and the reported values are averages over 100 runs. As expected, when images are processed less frequently, the position uncertainty increases, but it is important to note that the degradation in accuracy is relatively small. The penalty paid by increasing the inter-frame time from 0.5 sec to 20 sec (a 40-fold reduction in computation) is a mere 50% increase in the dimensions of the landing ellipse. Most importantly, even when images are processed at a low rate of 1 every 20 sec, the landing errors are still *three orders of magnitude smaller* than the errors attainable by the existing technology that relies on IMU dead reckoning.

The amount of computational resources required for image feature extraction is a function of the accuracy specifications of the vision algorithm. For example, sub-sampling images prior to processing can yield substantial gains in terms of computation, at the cost of less accurate feature localization. In Fig. 16, the 3σ values for the errors along the x- and y-axes at touchdown are plotted as a function of the standard deviation of the *image feature detection errors*. For these plots the frame rate of the camera was set to 1 Hz, and the density of landmarks at the landing site is 2 per km^2. From Fig. 16, it becomes evident that the performance loss inflicted by reducing the accuracy of the image measurements is not dramatic: increasing the image error standard deviation from 0.1 to 3 pixels (30-fold), results in a mere 40% increase of the position errors. This observation implies that it is possible to employ less accurate (and thus, less resource-consuming) image processing algorithms, that can attain real-time performance in spaceflight hardware, without substantially compromising estimation accuracy and mission safety.

An additional parameter of interest is the *density of MLs* at the landing site (and at the areas imaged during the entire EDL trajectory). The density of MLs determines the amount of positioning information that can be extracted from each image, and thus affects estimator performance. In Fig. 17, the 3σ values for the errors along the x- and y-axes at landing are plotted, as a function of the landmark density on the planet surface. The values used for this plot (0.1 to 5 landmarks per km^2) are within the range of values of crater density on the surface of Mars, reported in (Hartmann, 1999). The image errors' standard deviation is set to 1 pixel and the camera frame rate to 0.25 Hz for Fig. 17. It becomes evident that the density of landmarks has a much more significant effect on accuracy, than the factors previously considered. For example, when the landmark density is 0.1 per km^2, the position errors' 3σ values are approximately equal to 16 km (these data points are off the plots' scale), but drop to about 1.13 km for ML density of 0.5 per km^2. Thus, careful selection of the EDL trajectory can have a profound effect on the uncertainty of the spacecraft's landing position. An interesting fact is that the density of craters in each part of a planet is a function of the geological age of the surface (Hartmann, 1999). Thus, the geological processes occurring in an area (which determine the scientific interest of a landing site) indirectly affect the difficulty of landing a spacecraft accurately in this area.

As a final remark, we should note that the landing error statistics reported in this section pertain to the case where a camera and an IMU are the *only* sensors used for EDL navigation. In a realistic scenario, additional sensors (e.g., radar altimeter) may be available, in which case the resulting errors will be even *smaller*. Furthermore, it should be pointed out that

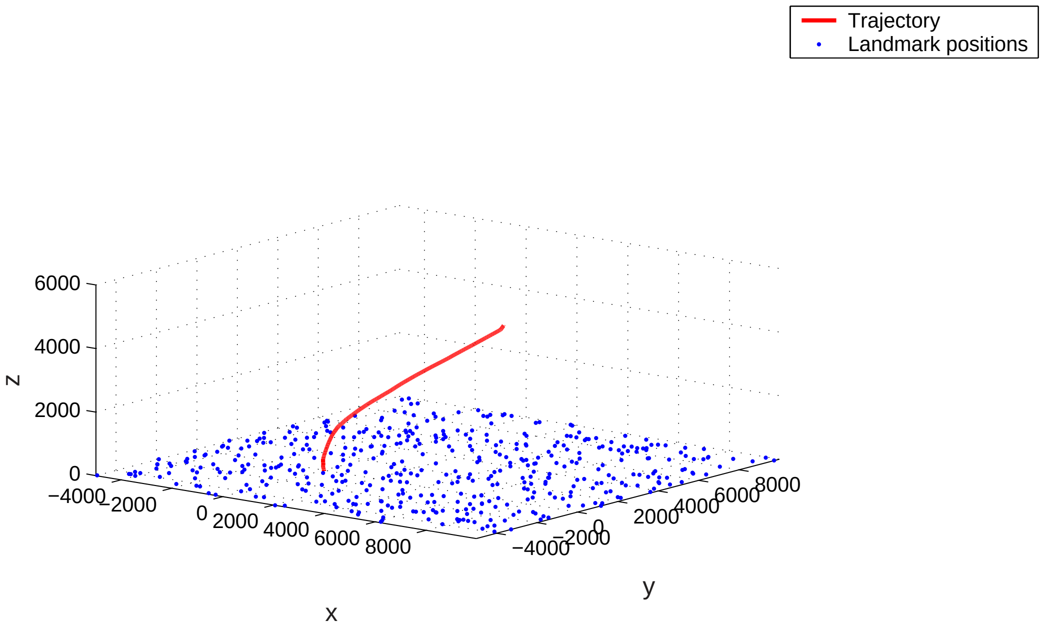


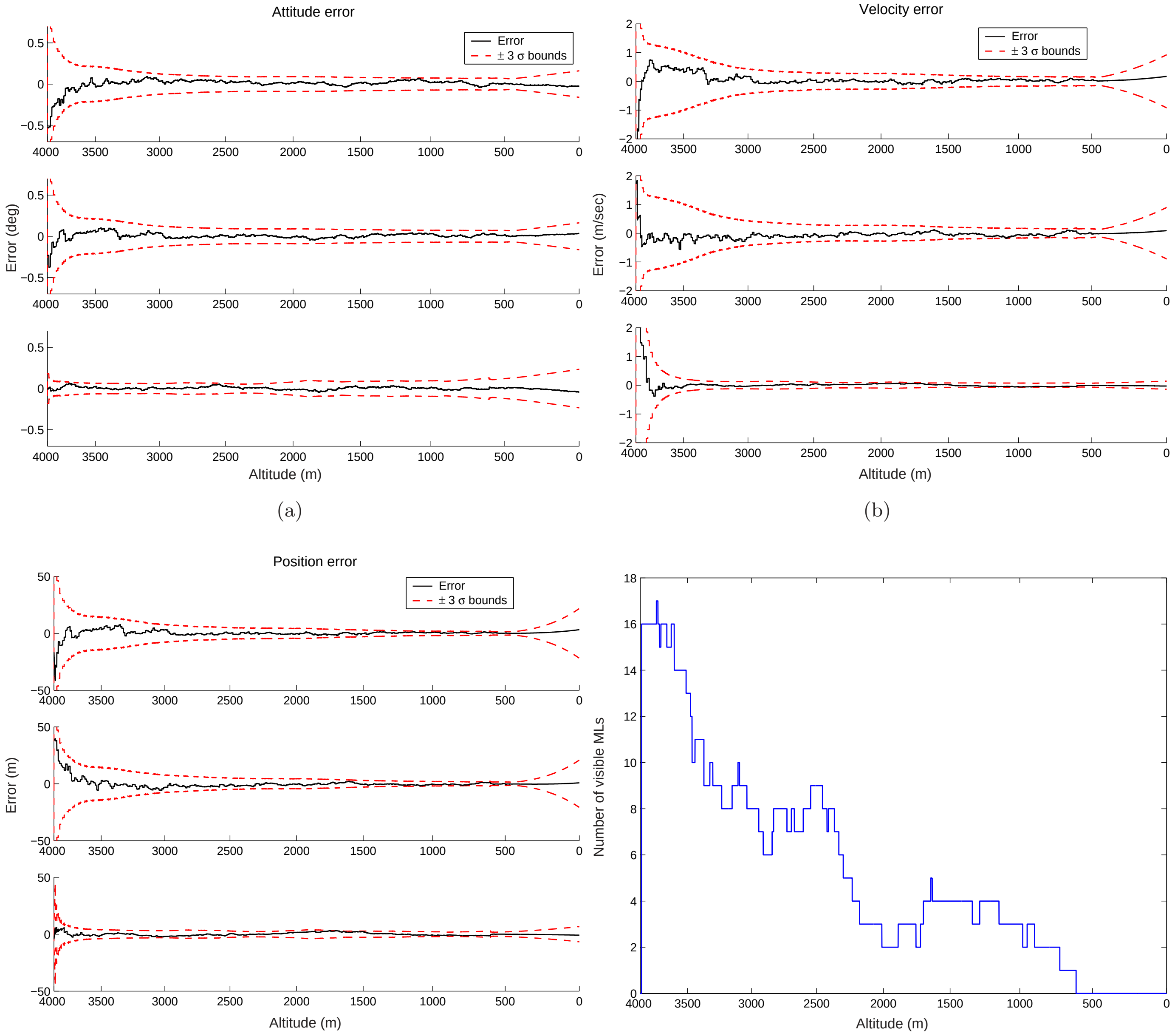
Figure 13: The trajectory of the lander used for the simulations.

there is a multitude of other factors that determine the performance of the proposed EKF-based estimator (e.g., accuracy of the IMU measurements, shape of the EDL trajectory, camera FOV, availability of additional sensors, and intensity as well as modeling accuracy of the gravity field, to name a few). A detailed study of the effects of such parameters is the subject of ongoing work.

6 Conclusion and Future Work

In this paper we have presented an algorithm for estimating spacecraft attitude, position, and velocity during EDL. In particular, we have described a method for fusing IMU measurements and visual observations of MLs, in an Extended Kalman Filter framework. We have shown that the coupling of inertial and vision sensing offers significant advantages for EDL. The use of ML observations (such as craters, or areas of distinctive texture on a planet's surface) makes it possible to obtain extremely accurate estimates of a lander's pose, during its descent. On the other hand, the availability of IMU signals allows the propagation of the state estimates between consecutive image-based updates, at a high rate. As a result, very precise pose and velocity estimates are accessible at high bandwidth. These characteristics of the proposed estimator (i.e., increased precision and high bandwidth), make it suitable for use in the feedback loop of an autonomous spacecraft control algorithm, allowing guided EDL and pin-point landing capabilities required in upcoming NASA missions.

The proposed algorithm has been validated experimentally, using the datasets from the DIMES field test and the MSL Subsonic Parachute Drop test. In both cases, the impressive accuracy attained by the fusion of inertial and visual measurements (sub-degree attitude and a few meters position errors) has been demonstrated. Moreover, the properties of the algo-



(a)

(b)

(c)

(d)

Figure 14: Filter accuracy as a function of altitude during descent. (a) The attitude errors of the filter (solid black line) and the associated $\pm 3\sigma$ values (red dashed line). (b) The velocity errors and the associated $\pm 3\sigma$ values. (c) The position errors and associated $\pm 3\sigma$ values. In all figures, the 3 sub-plots correspond to the x-, y-, and z-axes, respectively. (d) The number of visible landmarks for this descent trajectory.

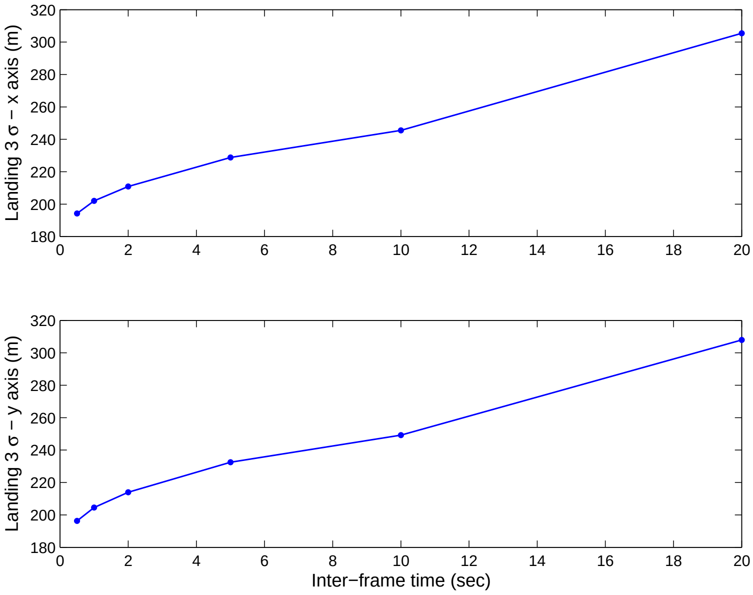


Figure 15: The 3σ values for the errors along the x- and y-axes at touchdown, as a function of the camera inter-frame time.

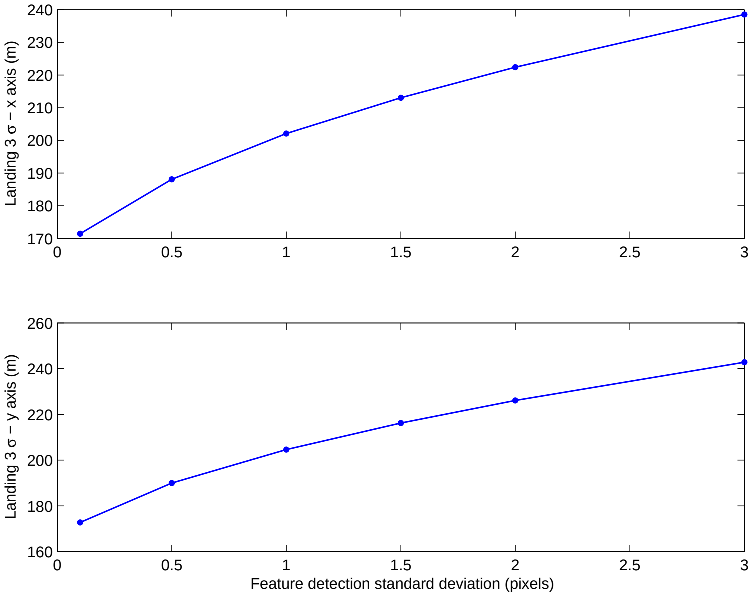


Figure 16: The 3σ values for the errors along the x- and y-axes at touchdown, as a function of the standard deviation of feature detection errors.

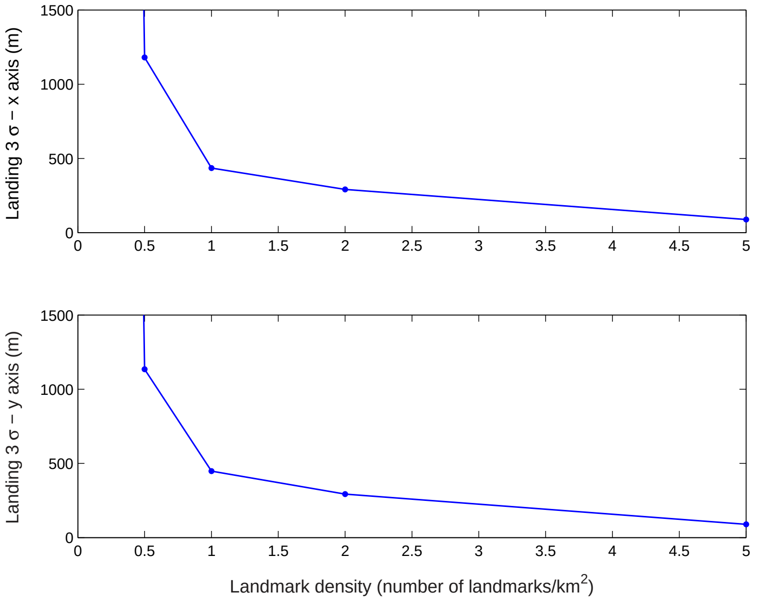


Figure 17: The 3σ values for the errors along the x- and y-axes at touchdown, as a function of the density of landmarks at the landing site.

rithm, and its dependence on several design parameters have been investigated in extensive simulation tests. These tests have indicated that the most significant factor for determining the size of the landing error ellipse is the density of MLs at the landing site. Interestingly, both the accuracy of the unit vector observations, and the frame rate at which such measurements become available, have a much smaller impact on the final landing errors. This observation is significant from a system design perspective, because it indicates that the requirements of the algorithm in terms of hardware capabilities (e.g., CPU speed) are quite flexible. Even under strictly constrained computational resources, the proposed estimator is *orders of magnitude* more accurate than existing flight-proven navigation solutions, which are based on inertial measurements alone.

In this paper, the presentation has focused primarily on the description of the EKF estimator and the analysis of its properties. It should be pointed out that the proposed estimator is general, and can be employed in conjunction with any feature detection algorithm that is capable of identifying features of interest in images and matching them to known landmarks in a map. Moreover, the modular design of the EKF-based estimator allows seamless incorporation of additional measurements from other sensors that may exist on a lander (e.g., radar altimeter). If further improvement of the attainable pose estimation accuracy is desired, the described algorithm can be extended in order to process the measurements of visual features whose map coordinates are not known *a priori*. This would enhance estimation performance in situations where the number of MLs in the landing area is small, or when, for instance, image-to-map matching fails due to illumination differences. Measurements to previously unknown features can be processed either in the SLAM framework (Langelaan, 2006; Strelow, 2004) or by imposing pairwise constraints between camera poses, for increased efficiency (Roumeliotis et al., 2002). The study of these extensions is the subject of future and ongoing work.

Acknowledgments

This work was supported by the University of Minnesota (DTC), the NASA Mars Technology Program (MTP-1263201), and the National Science Foundation (EIA-0324864, IIS-0643680). Portions of the research described in this publication was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

Andrade-Cetto, J. and Sanfeliu, A. (2004). The effects of partial observability in SLAM. In *Proc. of the 2004 IEEE International Conference on Robotics and Automation*, pages 397–402, New Orleans, LA.

Ansar, A. (2004). 2004 small body GN&C research report: Feature recognition algorithms. In *Small Body Guidance Navigation and Control FY 2004 RTD Annual Report (Internal Document)*, number D-30282 / D-30714, pages 151–171. Jet Propulsion Laboratory, Pasadena, CA.

Ansar, A. and Daniilidis, K. (2003). Linear pose estimation from points or lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):578–589.

Arulampalam, S., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188.

Barfoot, T. (2005). Online visual motion estimation using FastSLAM with SIFT features. In *Proc. of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 579–585, Edmonton, Alberta.

Bayard, D. S. and Brugarolas, P. B. (2005). An estimation algorithm for vision-based exploration of small bodies in space. In *Proceedings of the 2005 American Control Conference*, volume 7, pages 4589–4595, Portland, Oregon.

Breckenridge, W. G. (1999). Quaternions proposed standard conventions. Interoffice Memorandum IOM 343-79-1199, Jet Propulsion Laboratory, Pasadena, CA.

Burcin, L. (2002). Rad750 experience: The challenge of SEE hardening a high performance commercial processor. In *Proc. of Microelectronics Reliability & Qualification Workshop (MRQW)*. `http://www.aero.org/conferences/mrqw/2002-papers/A_Burcin.pdf`.

Carr, J. and Sobek, J. S. (1980). Digital scene matching area correlator (DSMAC). In *Image Processing For Missile Guidance, Proceedings of the Society of Photo-Optical Instrumentation Engineers*, number 238, pages 36–41.

Chatfield, A. B. (1997). *Fundamentals of High Accuracy Inertial Navigation.* American Institute of Aeronautics and Astronautics, Inc., Reston, VA.

Cheng, Y. and Ansar, A. (2005). Landmark based position estimation for pinpoint landing on mars. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4470–4475, Barcelona, Spain.

Cheng, Y., Goguen, J., Johnson, A., Leger, C., Matthies, L., Martin, M. S., and Willson, R. (2004). The mars exploration rovers descent image motion estimation system. *IEEE Intelligent Systems*, 19(3):13–21.

Cheng, Y., Johnson, A. E., Olson, C. F., and Matthies, L. H. (2003). Optical landmark detection for spacecraft navigation. In *Proceedings of the 13th Annual AAS/AIAA Space Flight Mechanics Meeting*, Ponce, Puerto Rico.

Crassidis, J. L., Alonso, R., and Junkins, J. L. (2000). Optimal attitude and position determination from line-of-sight measurements. *The Journal of the Astronautical Sciences*, 48(2–3):391–408.

Haralick, R. M., Joo, H., Lee, C., Zhuang, X., Vaidya, V. G., and Kim, M. B. (1989). Pose estimation from corresponding point data. *IEEE Transactions on Systems, Man and Cybernetics*, 19(6):1426–1446.

Hartmann, W. K. (1999). Martian cratering VI. Crater count isochrons and evidence for recent volcanism from Mars Global Surveyor. *Meteoritics and Planetary Science*, 34(2):167–177.

Head, J. N., Hoppa, G. V., Gardner, T. G., Seybold, K. S., and Svitek, T. (2005). Autonomous low cost precision lander for lunar exploration. In *Proc. of the 36th Annual Lunar and Planetary Science Conference*, number 1471, League City, TX.

Johnson, A., Bergh, C., and Bellutta, P. (2004). Pin point landing validation 2004 year end report. Technical report, Jet Propulsion Laboratory, Pasadena, CA.

Johnson, A., Willson, R., Goguen, J., Alexander, J., and Meller, D. (2005). Field testing of the mars exploration rovers descent image motion estimation system. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4463–4469, Barcelona, Spain.

Julier, S., Uhlmann, J., and Durrant-Whyte, H. (2000). A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control*, 45(3):477–482.

Kim, J. and Sukkarieh, S. (2004). Autonomous airborne navigation in unknown terrain environments. *IEEE Transactions on Aerospace and Electronic Systems*, 40(3):1031–1045.

Langelaan, J. W. (2006). *State Estimation for Autonomous Flight in Cluttered Environments*. PhD thesis, Stanford University, Department of Aeronautics and Astronautics.

Lefebvre, T., Bruyninckx, H., and Schutter, J. D. (2001). Kalman filters for nonlinear systems: A comparison of performance. Technical report, Department of Mechanical Engineering Katholieke Universiteit Leuven Belgium.

Lefferts, E. J., Markley, F. L., and Shuster, M. D. (1982). Kalman filtering for spacecraft attitude estimation. *Journal of Guidance, Control, and Dynamics*, 5(5):417–429.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2(60):91–110.

Maybeck, P. S. (1979). *Stochastic Models, Estimation and Control*, volume 1+2. Academic Press, New York.

Montgomery, J., Johnson, A., Roumeliotis, S., and Matthies, L. (2006). The jet propulsion laboratory autonomous helicopter testbed: A platform for planetary exploration technology research and development. *Journal of Field Robotics*, 23(3/4):245–267.

Mortari, D., Rojas, J., and Junkins, J. (2004). Attitude and position estimation from vector observations. In *Proceedings of the 2004 Space Flight Mechanics Meeting Conference*, number AAS 04-140, Maui, Hawaii.

Mourikis, A., Trawny, N., Roumeliotis, S., Helmick, D., and Matthies, L. (2007). Autonomous stair climbing for tracked vehicles. *International Journal of Computer Vision & International Journal of Robotics Research - Joint Special Issue on Vision and Robotics.* To appear.

Quan, L. and Lan, Z.-D. (1999). Linear n-point camera pose determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):774–780.

Roumeliotis, S. and Burdick, J. (2002). Stochastic cloning: A generalized framework for processing relative state measurements. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1788–95, Washington D.C.

Roumeliotis, S., Johnson, A., and Montgomery, J. (2002). Augmenting inertial navigation with image-based motion estimation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4326–33, Washington D.C.

Sinha, S., Frahm, J.-M., Pollefeys, M., and Genc, Y. (2006). GPU-based video feature tracking and matching. In *Workshop on Edge Computing Using New Commodity Architectures*, Chapel Hill, NC.

Strelow, D. (2004). *Motion estimation from image and inertial measurements.* PhD thesis, Carnegie Mellon University.

Sun, D. and Crassidis, J. L. (2002). Observability analysis of six-degree-of-freedom configuration determination using vector observations. *Journal of Guidance, Control, and Dynamics*, 25(6):1149–1157.

Trawny, N. and Roumeliotis, S. I. (2005). Indirect Kalman filter for 6D pose estimation. Technical Report 2005-002, University of Minnesota, Dept. of Comp. Sci. & Eng.

U.S. Geological Survey (2006). Seamless data distribution system. `http://seamless.usgs.gov/index.asp`.

Wu, A., Johnson, E., and Proctor, A. (2005). Vision-aided inertial navigation for flight control. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, number AIAA 2005-5998, San Francisco, CA.