TERRAIN AIDED NAVIGATION FOR PLANETARY EXPLORATION MISSIONS

Jakob Schwendner and Javier Hidalgo

Robotics Innovation Center, DFKI, Bremen, Germany

ABSTRACT

A central part of current space activities is to learn more about our solar system, its origins, its resources and its conditions for harbouring life. A key technology for performing surface exploration of celestial bodies are mobile robotic systems, as they are able to withstand the harsh conditions with reasonable effort. One important requirement for performing navigation of a mobile robot is the ability to localise the system within a known reference. Visual methods, which have proven useful in this context, can add constraints on processing power and environmental conditions. In this paper an alternative approach is presented which only uses inertial sensors and encoders in order localise within a known map. The method is evaluated in a simulated lunar environment based on LRO digital elevation maps. The results show an average localisation error of 11 m for a travelled distance of 2.3 km, assuming low-precision MEMS gyros. The proposed method can be applied to perform resource efficient localisation in situations where visual methods fail or are too costly to perform.

Key words: Terrain Aided Navigation; Exploration; Localisation; LRO DEM.

1. INTRODUCTION

Mobile robotic systems will without a doubt become even more relevant for space exploration missions than they already are [11]. High cost for manned programs and recent success of robotic mission (e.g. MER [Mai06]) are likely to form a shift towards robotic missions. One of the key aspects of exploration systems is mobility. Apart from the physical capabilities to negotiate complex and difficult terrain, the aspect of navigation is of great importance in this context. Navigation requires the three key components of localisation, mapping and control. The subject of this paper is the problem of robot localisation. A mobile robot requires the knowledge of its position with respect to its goal, as well as a model of its surrounding to generate steering commands which will lead it to the goal. This is true regardless of whether the system is tele-operated or performs its navigation autonomously. Earthbound activities can make use of the Global Positioning System (GPS), which provides a means of localisation with respect to the earth fixed frame. Although missions for mobile surface exploration systems often have an orbiter component, they can not provide sufficient accuracy in localising the robot. Most potential mission targets however have a map generated from orbiters. Most probably these maps are not provided with an accuracy detailed enough for control of the system. They do however provide a global reference map, which can put mission targets in a spatial context and aid global path planning [Mur10].

Most previous successful approaches for mobile robot navigation in space are vision based [MMJ+07]. A great number of positive sides like environmental awareness and exploitation of the images for science are on the upside of this. There are however some design constraints in order to make the use of vision based navigation feasible. Image processing does require a certain amount of processing resources and a favorable positioning of the cameras [HK97].

One possible alternative could be the use of Terrain Aided Navigation [GGB+02, Lon82, Hos78] methods, which use a-priori map knowledge to compensate drift errors from dead-reckoning navigation. These approaches have been applied to the problem of aerial localisation with the aid of radar sensors and terrain height maps. Matching of 3D optical sensor information to known terrain maps [Car10] has been shown to provide absolute referencing. A proof of concept for non-visual methods was given by [CVGL07] on a four legged robot. In [SJ11] it was shown that it is possible to localise using only inertial sensor and encoder data on a 50 m by 30 m sand field to within an accuracy of 1.5 m. The map resolution was 5 $^{cm}/_{pixel}$, which is not realistic for orbiter based maps.

In this paper a method based on [SJ11] is presented, which is able to perform non-visual global referencing of a ground based rover using maps from an orbiter. Inertial sensor readings and encoder values are the only sensor data used to perform the localisation in a Digital Elevation Map (DEM) from the LRO [CBF+07] mission. The feasibility of the approach is demonstrated in a simulated environment which takes the lunar DEM and a simulated model of the Asguard v4 [JSK+11] system. The simulation takes gyro noise into account to generate realistic heading errors for the odometry model. The method is applied to the simulated data, and compared to the con-

tact point based odometry model also presented in this paper.

The contributions of this work are (i) a method to extract terrain slope from an odometry model, (ii) a measurement model for evaluating terrain slope against a DEM model and (iii) evaluation of these methods in a Bayesian filter framework for estimating the position in a km scale simulated scenario.

While the proposed method should not be seen as a complete replacement of visual navigation, it can augment such systems. It can also provide a localization solution with a bounded error for missions where visual processing is not feasible due to resource or engineering constraints.

2. METHOD

2.1. Overview

The method which is proposed in this paper uses a particle distribution in order to represent the pose uncertainty of a planetary exploration rover. The rover provides information from the wheel encoders and wheel position information in case of a non-rigid chassis. Further, the rover uses inertial sensors and sensor fusion in order to estimate the direction of the gravity vector. The fused inertial sensors also provide an estimate of the heading, which is prone to drift unbounded because of accumulated gyro errors. This information is used in a probabilistic odometry model of the system, in order to forward project the state distribution. A terrain slope measurement is generated based on accumulated environment contact information over time. This terrain slope information is compared against a digital elevation model of the environment in order to correct the pose distribution.

2.2. Odometry Model

The odometry model provides a way to make a prediction of the pose delta of the robot, so the change in position and orientation, between two time steps. Odometry models have seen extensive research (e.g. [GZZD05]) and usually depend on the class and configuration of the system. The work presented in this paper uses an environment contact based approach. The underlying assumption is that a set of contact points C, which have a defined, time variate position expressed in the body frame of the robot B stay fixed in the environment. What is required now is the full transform

$$C_{B_{k+1}}^{B_k} = \begin{bmatrix} R_{B_{k+1}}^{B_k} & T_{B_{k+1}}^{B_k} \\ 0 & 1 \end{bmatrix} \tag{1}$$

from the body frame B at time step k, to the body frame B at time step $k+1$. T and R are the rotational and translational parts of the transform.

The rotational part of the transform is already known through the estimate of the inertial measurement unit. The IMU provides a rotation R_{IMU}^{W} from the IMU fixed frame to a local geo fixed frame W, and the transformation C_{IMU}^{B} from the IMU frame to the body frame B is known. Then the rotational part of the delta can be given as

$$R_{B_{k+1}}^{B_k} = (R_{IMU_k}^{W} R_{B}^{IMU})^{-1} R_{IMU_{k+1}}^{W} R_{B}^{IMU} \tag{2}$$

Finding the translation is based on the assumption that the contact points are fixed in the environment. By rotating the contact points from two time steps into the same reference frame, the frame translation is given by the position difference of the same contact point at two different time steps. For a single point, the assumptions that the contact point stays fixed in the environment always holds. However, for multiple points the problem is over-specified, and the translation is required such that the squared sum of the individual displacements is minimal. It can be easily verified that this is just the average of all the displacement, so that finally the translational delta can be given as

$$T_{B_{k+1}}^{B_k} = \frac{1}{|C|} \sum_{c \in C} c_k - R_{B_{k+1}}^{B_k} c_{k+1} \tag{3}$$

This delta transformation is assumed to be the mean μ_O of the odometry transform. To model for different types of errors, a simple Gaussian distribution with covariance Σ_O is parametrized based on certain dynamic and static aspects.

$$\Sigma_O = diag\left(A \begin{bmatrix} \cos^{-1}(e_z \cdot R_{B_k}^{W} e_z) \\ |T_{B_{k+1}}^{B_k}| \\ R_z(R_{B_{k+1}}^{B_k}) \\ 1 \end{bmatrix} \right) \tag{4}$$

In (4) e_z is the unity vector in the direction of the z-axis. A is a configuration matrix, which relates the tilt of the robot, the delta translation and the change in heading to the pose errors. A robot which is situated on an incline is more likely to slip than a robot on a flat surface. Equally, higher translational velocities, which are expressed in a larger translation over a fixed time period, are more likely to generate errors due to system dynamics. The actual values for the coefficients of A depends on the system and the environment.

Based on the mean and covariance values, the final odometry distribution can be given as

$$p(x_k | u_k, x_{k-1}) = \mathcal{N}(\mu_O, \Sigma_O) \tag{5}$$

where u_k is the odometry input consisting of the contact point values and the orientation reading from the IMU.

2.3. Measurement Model

In Section 2.2 a method was developed to estimate the transformation between the body frame in two consecutive time steps. This is an estimate and prone to errors from different sources. These errors accumulate over time and lead to drift in the pose estimate of the robot. Depending on the quality of the sensor data, the values which are generated are locally stable. The principle idea behind the measurement model used in this approach is to extract information on the contact points with the environment and place them in a locally consistent correlation. Further, it is assumed that the measurements from the IMU can provide information on the pitch and roll of the robot, since they are able to compensate for drift errors by measuring the gravity vector. This is not possible for the heading error. The contact points can be related to the body frame and the gravity vector. By fitting a plane through the contact points, an estimate of the slope of the terrain the robot has traversed can be performed. Given a map with slope information, and a position and heading, the measurement from the contact points can be related to that map, by comparing the normals of the respective surfaces. This information can then be used in the particle filter to estimate the likelihood of a particular hypothesis.

Let $C^O_{B_k}$ be the transform from the body frame at time k to an arbitrary odometry frame O, such that $C^O_{B_{k+1}} = C^O_{B_k} C^{B_k}_{B_{k+1}}$. The set S of contact points in the odometry frame can be constructed by transforming the contact points for the last n time steps.

$$S = \bigcup_{i=1..n} \bigcup_{c \in C_{k-i}} C^O_{B_{k-i}} c \tag{6}$$

In this approach n is chosen so that the total translation $|T^{B_{k-n}}_{B_k}| < t_{\text{meas}}$ is smaller than a threshold value.

The next step is to fit a plane through the set S of contact points. There are multiple methods on how this can be done. For simplicity the following system of linear equations is chosen

$$\sum_{p \in S} \begin{bmatrix} p_x^2 & p_x p_y & p_x \\ p_y p_x & p_y^2 & p_y \\ p_x & p_y & 1/|S| \end{bmatrix} x = \sum_{p \in S} \begin{bmatrix} p_x p_z \\ p_y p_z \\ p_z \end{bmatrix} \tag{7}$$

which when solved for x results in a scaled normal of the surface $\tilde{n} = (-x_x, -x_y, 1)$ which can be normalized to $n = \tilde{n}/|\tilde{n}|$. The surface normal is expressed in the arbitrary odometry frame O. For the slope to be comparable to the slop information from the map, the normal is rotated into the body frame. Rather than using the full R^O_B rotation, only the component around the z-axis is applied to preserve the direction of the gravity vector. This results in the measured normal in body coordinates as

$$z_k = R_z(-R_z^{-1}(R^O_B))n. \tag{8}$$

The measurement model is used to estimate the likelihood of a particular state hypothesis x_k given the measurement and the map $p(x_k|z_k, m)$. The measurement in this case is the surface normal in body coordinates, and the map is a digital elevation map of the environment. In order to compare the slope measured by the odometry model to the map, the map needs to provide slope information as well. If this information is not available directly, it can be estimated based on the neighbouring map cells. Let m_{pq} be the elevation value of the map cell at index position pq which correspond to the x and y coordinates of the state variable x_k. The slope values in x and y direction can then be directly derived given the cell size c, such that the surface normal at the position of x_k is

$$\tilde{n}_m = \left(-\frac{(m_{p+1q} - m_{p-1q})}{2c}, -\frac{(m_{pq+1} - m_{pq-1})}{2c}, 1 \right). \tag{9}$$

Which again is normalized such that $n_m = \tilde{n}_m/|\tilde{n}_m|$. Similar to the measured normal from the odometry, the map normal needs to be rotated into body coordinates, in order to be comparable. The comparison is performed by calculating the angle between the two normals. The error due to measurement uncertainties and other effects is assumed to be normally distributed so that for the likelihood of the measurement we can finally give

$$p(x_k|z_k, m) = \phi(\frac{1}{\zeta} \cos^{-1}(z_k \cdot R_z(-R_z^{-1}(x_k))n_m)), \tag{10}$$

where ζ is the measurement noise factor and ϕ the standard normal distribution.

2.4. Particle Filter

The pose of the robot is modelled as an uncertainty distribution over x, y and θ, which represent the position on the plane and the heading of the robot. The z-position is not modelled explicitly, since it can be derived from the elevation model, given the assumption that the robot is in contact with the ground all the time. Due to the nature of the problem, the probability distribution can not be modelled using Gaussians, and a sampling representation is used instead. A particle filter is then used to update the representation according to the odometry and the measurement model. The odometry performs a forward propagation of the pose samples in the distribution. The odometry model provides a mean and an error distribution over the pose deltas, so the change of pose between two time steps. In the projection step of the filter, each of the pose particles gets updated with a sample from the odometry model distribution. In that way the prior error distribution is increased by the error that is added from the odometry model. The update step is called repeatedly in a fixed time interval. Whenever the robot has travelled a distance larger than t_{meas}, the measurement model is used. According to (8), the measurement of the slope of the environment in body coordinates is calculated as z_k, and updated according the update model $p(x_k|z_k, m)$ defined in (10). This measurement model provides a likelihood value for the state x_k given the measurement and the map. Each particle has a relative weight value, which gets

updated by the measurement function. In this way particles that represent a pose with a high likelihood receive a larger weight than those with a lesser likelihood. After a while, certain particles will have a very low weight, and do not contribute significantly to representing the probability distribution. This can be measured by the effective number of particles N_{eff}. Whenever this value falls below a certain threshold, the particles get resampled according to their relative particle weight.

Algorithm 1 Particle Filter

$\bar{\chi}_k = \chi_k = \emptyset$
$D = I$
for all $m = 1 \dots M$ **do**
  $\Delta T \sim p(x_k | u_k, x_{k-1})$
  $x_k^{[m]} = x_k^{[m]} + \Delta T$
  $D = \Delta T D$
  if $D_{trans} > t_{\text{meas}}$ **then**
    $w_k^{[m]} = w_{k-1}^{[m]} p(z_k | x_k^{[m]})$
    $D = I$
  end if
  $\bar{\chi}_k = \bar{\chi}_k + < x_k^{[m]}, w_k^{[m]} >$
end for
$\hat{N}_{\text{eff}} = \frac{1}{\sum_{i=1}^{M} \left(w_k^{[i]}\right)^2}$
if $\hat{N}_{\text{eff}} < N_{thr}$ **then**
  draw i with probability $\propto w_k^{[i]}$
  add $x_k^{[i]}$ to χ_k
else
  $\chi_k = \bar{\chi}_k$
end if

The Particle Filter used is a Sequential Importance Resampling (SIR) [GSS93] filter, where an initial set of particles is generated using a random distribution over $X \sim \mathcal{N}(\mu_0, \Sigma_0)$. A more in depth discussion of the filter in the context of this application can be found in [GGB+02].

3. EXPERIMENTS

In order to evaluate the feasibility of the approach for localisation in an lunar exploration scenario a representative set up was generated in a simulation environment. The Lunar Reconnaissance Orbiter (LRO) [CBF+07] carries instruments for geological inspection of the lunar surface. The LOLA instrument generates globally consistent elevation data. Further, the narrow angle camera (NAC) produces high-resolution images in the order of 0.5[m/pixel]. This image data can be used to extract high resolution digital elevation maps (DEM) from the lunar surface [MBB+10]. One of these maps from the Aitken Basin near the lunar south pole was selected (see Fig. 1). The map was generated by the United States Geological Survey (USGS) and made publicly available through the LMMP website [1]. The identifier for the map is [LRO_NAC_DEM_60S200E_150cmp.tif]. From this map, a 1500 m by 1500 m segment was extracted for further processing. The coordinates of the map corners are:

$$(160°14'11.28"W, 59°39'50.92"S)$$
$$(160°14'11.28"W, 59°42'49.00"S)$$
$$(160°8'15.12"W, 59°39'50.92"S)$$
$$(160°8'15.12"W, 59°42'49.00"S)$$

The rendered image shown in Fig. 2 provides an impression of the local shape of the environment.

During the experiments a gyroscope model is used in order to be precise. The model is balanced between accuracy and complexity covering the commonly encountered errors in inertial sensors but does not include g-sensitive drift, scale factor asymmetry or other errors which depend on the design of a particular sensor and technology. The angle random walk and the rate angle random walk are considered the dominant stochastic errors for gyros. The first error is affected by white noise while the second one affects the bias instability directly altering the gyros drift. It is a variation of Farrenkopf's gyro model [Far78] which has been widely accepted and used in the aerospace sector for many years. The model is given as:

$$\tilde{\boldsymbol{\omega}}(t) = (I_{3\times3} + M(SF, MA))\boldsymbol{\omega}(t) + \boldsymbol{\beta}_\omega(t) + \boldsymbol{n}_{rw}(t) \tag{11a}$$

$$\dot{\boldsymbol{\beta}}_\omega(t) = \boldsymbol{n}_{rrw}(t) \tag{11b}$$

where $\tilde{\boldsymbol{\omega}}(t)$ is the sensor readout, $\boldsymbol{\omega}(t)$ is the ideal value, $\boldsymbol{\beta}_\omega(t)$ is the bias and $\boldsymbol{n}_{r\omega}(t)$ and $\boldsymbol{n}_{rr\omega}(t)$ are independent zero-mean Gaussian white noise of the characterized stochastic processes defined with

$$E\{\boldsymbol{n}_{r\omega}(t)\boldsymbol{n}_{r\omega}^T(\tau)\} = I_{3\times3}N^2\delta(t-\tau) \tag{12a}$$
$$E\{\boldsymbol{n}_{rr\omega}(t)\boldsymbol{n}_{rr\omega}^T(\tau)\} = I_{3\times3}K^2\delta(t-\tau) \tag{12b}$$

where E denotes expectation, $\delta(t-\tau)$ is the Dirac delta function, N and K are the angle random walk and rate random walk coefficients in $rad/\sqrt{s}$ and $rad/s/\sqrt{s}$ respectively. M is the deterministic errors matrix corresponding to the misalignment (MA) and scale factor (SF). Assuming to be temperature independent these errors can be calibrated and therefore the matrix M has been set to zero in this case.

The resulting DEM was used in the MARS [RKK09] simulation environment. The rover which was used in the simulation is modelled after the Asguard v3 system [JSK+11]. The simulation provides true position and orientation values as well as information about contact points with the environment. The position values from the simulation are only used as reference. The orientation values have been perturbed by additional noise from a simulated gyro. The parameters for the gyro model have been chosen to be K $= 1.5 \cdot 10^{-4} rad/s/\sqrt{s}$ and

[1] http://lmmp.nasa.gov

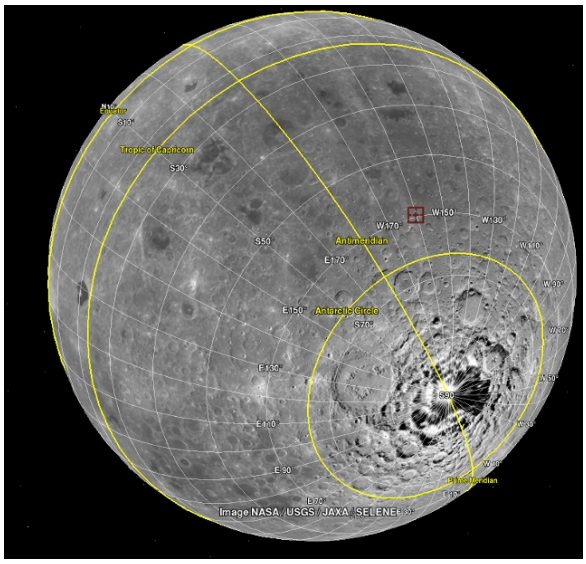

Figure 1. The experiments were performed on digital elevation maps (DEM) from the lunar south pole. A 1500 m by 1500 m patch with a resolution of 1.5 m/pixel was selected at [160°14'11.28" W, 59°39'50.92" S]. The patch was extracted from a DEM (LRO_NAC_DEM_60S200E_150cmp.tif), which was made publicly available by the United States Geological Survey (USGS) through the LMMP website.

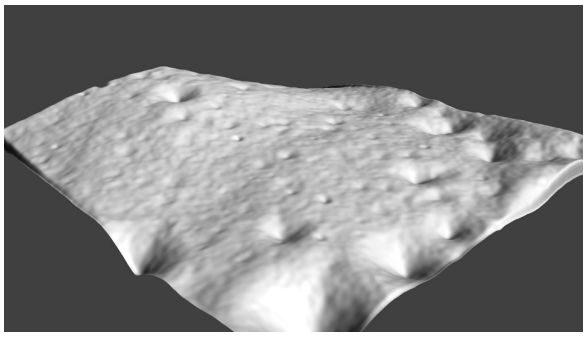

Figure 2. A rendered image of the DEM which was used for the experiments. The height values are not to scale. The difference between the lowest and the highest point for this 1500 m by 1500 m patch is 118 m

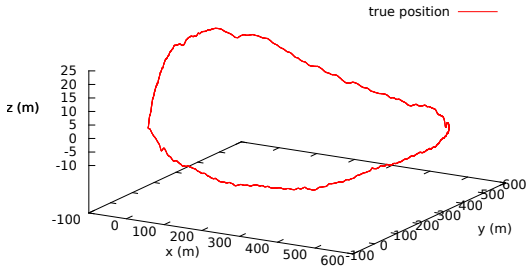

Figure 3. A rigid body simulator [RKK09] was used to run a model of the Asguard system [JSK+11] on the lunar DEM. The graph shows the trajectory of the robot in the simulated environment, starting and ending at the origin. The trajectory has a total length of 2300 m. The difference between the lowest and the highest point on the trajectory is 32 m.

N $= 5.7 \cdot 10^{-4} rad/\sqrt{s}$. These parameters have been extracted experimentally using the Allan variance [EHN08] technique [2] from the commercially available XSens-MTI. The gyro rate error is evaluated for a single gyro only, integrated to produce the heading error and finally applied to the orientation values from the simulation. The noise added orientation data as well as the contact information is sufficient information for the odometry model described in Section 2.2. The odometry result is the base reference against which the filter performance is compared. The filter described in Section 2.4, receives the same inputs as the odometry model, but also holds a reference to the DEM model. The initial condition of the particle filter is a sampling around the start position of the trajectory, which is considered known. The orientation with respect to the map is considered unknown and needs to be recovered by the filter. The t_{meas} variable was set to the cell size of the DEM.

There are a number of ways to represent the particle pose distribution as a single position value. For simplicity reason, the particle centroid has been chosen. This is the weighted average of the particle position and orientation values.

[2] http://cran.r-project.org/web/packages/allanvar/

4. RESULTS

The MARS simulator was able to generate position and orientation values as well as environment contact points for the Asguard v3 model in real-time. The total simulation time for the provided trajectory is 70 min. The total distance the rover has travelled is 2300 m, which results in an average velocity of 0.54 m/s. The loop the rover follows, starts and ends at the same position. The overall height difference between the highest and the lowest point of the trajectory is 32 m. A plot of the trajectory can be seen in Fig. 3.

The orientation values are perturbed by gyro noise. The results of the gyro noise modelling can be seen in Fig. 4. The integration of the rate noise results in a final heading error of approximately 6 rad. The gyro noise parameters were chosen to reflect characteristics of low-precision MEMS gyroscopes, and thus represent a very basic sensor configuration. The trajectory the odometry follows can be seen in Fig. 6. The error in heading has a significant impact on the trajectory of the odometry. The global

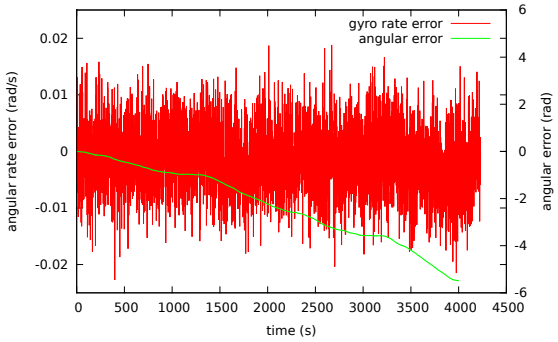

Figure 4. In order to simulate realistic sensor condition, the true orientation provided by the simulation was perturbed by simulated gyro noise. The graph shows the angular rate noise over the course of 70 min of runtime. The plot also shows the angular error, which is the integration of the rate error.

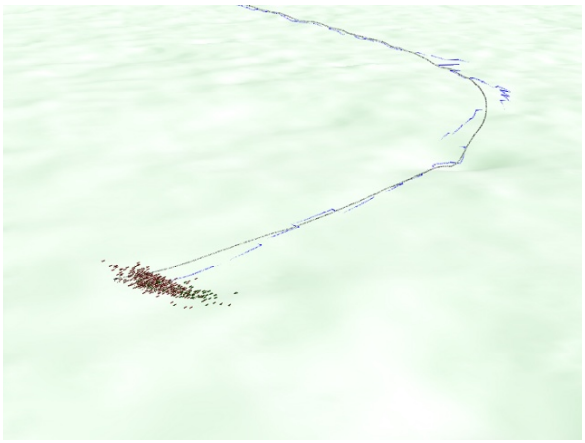

Figure 5. The simulated sensor data was processed by the localisation filter proposed in the methods section. The illustration shows the true pose (black) and the centroid of the particle distribution (blue). The particles represent the uncertainty distribution of the current pose of the robot.

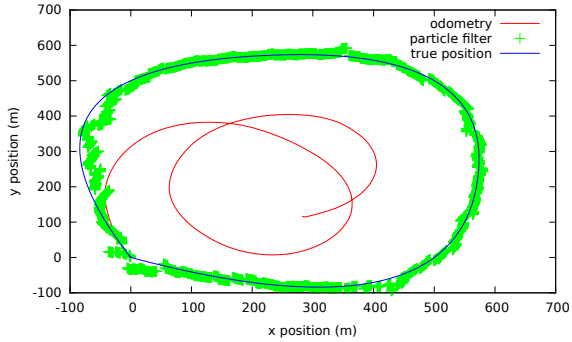

Figure 6. The result of the localisation experiment are shown in this graph. The true position in x and y-direction is plotted in blue. The odometry (red) exhibits significant deviation from the true position, mainly caused by the gyro noise which results in heading errors. The estimate of the particle filter (green) is able to track the position and exhibits a bounded error behavior.

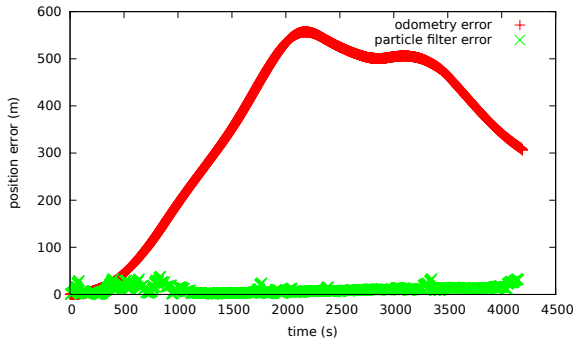

Figure 7. Comparison of odometry error to the filter error. While the odometry error grows up to 500 m, the filter error stays within a 40 m boundary.

position estimate has a significant deviation compared to the reference trajectory, and is only meaningful locally.

The particle filter based approach receives the DEM of the environment as additional information to perform global referencing. The data provided by the simulation was processed according to the description given in the methods section. A total number of 500 particles was selected and the odometry parameters adapted to the simulation and gyro noise environment. Fig. 5 shows a visualisation of the particle filters internal state. The current pose estimate is represented as a set of particles. The distribution of the particles represents the uncertainty over the pose. Because the gyro error is modelled in the error model of the odometry, the particles spread in the typical half-moon shape for orientation uncertainty. Particles which correlate the most with the reference map are given higher weights and are reinforced against low weight particles. In this way, the particle distribution is able to track the true position of the rover much closer, which can be seen in Fig. 6.

Fig. 7 shows a comparison of the odometry error with the error from the filter. The error profile for the odometry is much smoother, but grows up to an error of 500 m, while the filter error stays within a 40 m boundary. The error is calculated as the length of the distance vector between the position on the true pose vs the position on the estimate at any particular time step. The min and max values for the filter error are 0.9 m and 38.8 m, with an average of 11.7 m. The experiments were performed on an Core-i7 processor with 2.8 GHz clock frequency. The runtime for the filter was around 10 min for the 70 min simulation time log. The code was not optimized for this particular task and could be made significantly faster.

5. CONCLUSION

Mobile robotic systems will remain to have a significant influence on planetary and lunar exploration activities. The ability to navigate reliably is a crucial factor in this. While visual methods are a valid option in many cases, they may fail to perform under certain conditions or prove to be too resource intensive. In this paper, a method was proposed, that allows to perform global localisation using digital elevation models of the surface to be explored. These models are often available through orbiters or other sources like DEMs from a lander descent. The experiments performed show that it is feasible to track the global position for large travel distances of up to several kilometres. For a distance of 2.3 km, the filter approach was able to track the position with a maximum error of 40 m, and an average error of 11 m. This accuracy is sufficient for example to return to a lander module autonomously after several kilometers of travel. Also, this method does not make use of visual data other than the DEM models. No visual data needs to be processed on the rover, making it very resource efficient. The method will also work in complete darkness, as well as other conditions like atmospheric clouds or dust storms.

The results provided give a proof of concept in simulation. Although the principle approach has been verified in different scenarios, more experimental work is required to detail the effectiveness under realistic conditions, as well as the resource requirements. The current method of using the centroid of the particles for estimating best pose can be improved by taking particle history and clustering into account.

ACKNOWLEDGMENTS

The work presented here is part of the Project "Intelligent Mobility", which is funded by the Federal Ministry for Economics and Technology (BMWI) through the German Space Agency (DLR) grant number 50 RA 0907.

REFERENCES

[Car10] PJF Carle. Global rover localization by matching lidar and orbital 3D maps. In *Robotics and Automation (ICRA),*, pages 881–886, 2010.

[CBF+07] Gordon Chin, Scott Brylow, Marc Foote, James Garvin, Justin Kasper, John Keller, Maxim Litvak, Igor Mitrofanov, David Paige, Keith Raney, Mark Robinson, Anton Sanin, David Smith, Harlan Spence, Paul Spudis, S. Stern, and Maria Zuber. Lunar reconnaissance orbiter overview: Theinstrument suite and mission. *Space Science Reviews*, 129:391–419, 2007.

[CVGL07] S. Chitta, P. Vernaza, R. Geykhman, and D.D. Lee. Proprioceptive localization for a quadrupedal robot on known terrain. In *Proc. IEEE ICRA*, 2007.

[EHN08] N. El-Sheimy, H. Hou, and X. Niu. Analysis and modeling of inertial sensors using allan variance. *IEEE Transactions on Instrumentation and Measurement*, 57(1):140–149, 2008.

[Far78] R. L. Farrenkopf. Analytic Steady-State Accuracy Solutions for Two Common Spacecraft Attitude Estimators. *Journal of Guidance, Control, and Dynamics*, 1(4):282–284, 1978.

[GGB+02] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund. Particle filters for positioning, navigation, and tracking. In *IEEE Transactions on Signal Processing*, volume 50, 2002.

[GSS93] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *Radar and Signal Processing*, volume 140, 1993.

[GZZD05] B. Gassmann, F. Zacharias, J.M. Zollner, and R. Dillmann. Localization of Walking Robots. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, number April, pages 1471–1476. Ieee, 2005.

[HK97] WH Huang and EP Krotkov. Optimal stereo mast configuration for mobile robots. In *IEEE Robotics and Automation*, number April, 1997.

[Hos78] L.D. Hostetler. Optimal terrain-aided navigation systems. Technical report, Sandia Labs., Albuquerque, NM (USA), 1978.

[11] International Space Exploration Coordination Group (ISECG). The global exploration roadmap, 2011.

[JSK+11] Sylvain Joyeux, Jakob Schwendner, Frank Kirchner, Ajish Babu, Felix Grimminger, Janosch Machowinski, Patrick Paranhos, and Christopher Gaudig. Intelligent Mobility - Autonomous Outdoor Robotics at the DFKI. *KI - Künstliche Intelligenz*, 2011.

[Lon82] W. E. Longenbaker. Terrain-aided navigation of an unpowered tactical missile using autopilot-grade sensors. In *Guidance and Control Conference*, 1982.

[Mai06] C. Maimone, M.; Biesiadecki, J.; Tunstel, E.; Cheng, Y.; Leger. Surface navigation and mobility intelligence on the Mars Exploration Rovers. In *Intelligence for Space Robotics*, chapter 3, pages 45—-69. TSI Press, 2006.

[MBB+10] Z. M. Moratto, M. J. Broxton, R. A. Beyer, M. Lundy, and K. Husmann. Ames Stereo

Pipeline, NASA's Open Source Automated Stereogrammetry Software. In *Lunar and Planetary Institute Science Conference Abstracts*, volume 41 of *Lunar and Planetary Inst. Technical Report*, page 2364, March 2010.

[MMJ+07] Larry Matthies, Mark Maimone, Andrew Johnson, Yang Cheng, Reg Willson, Carlos Villalpando, Steve Goldberg, Andres Huertas, Andrew Stein, and Anelia Angelova. Computer Vision on Mars. *International Journal of Computer Vision*, 75(1):67–92, 2007.

[Mur10] Liz Murphy. Planning most-likely paths from overhead imagery. In *Robotics and Automation (ICRA),*, pages 3059–3064, 2010.

[RKK09] M. Römmerman, D. Kuhn, and F. Kirchner. Robot design for space missions using evolutionary computation. In *Evolutionary Computation, 2009. CEC '09. IEEE Congress on*, pages 2098 –2105, may 2009.

[SJ11] Jakob Schwendner and Sylvain Joyeux. Self Localisation using Embodied Data for a Hybrid Leg-Wheel Robot. In *IEEE International Conference on Robotics and Biomimetics*, 2011.