2017 17th International Conference on Control, Automation and Systems (ICCAS 2017)
Oct. 18–21, 2017, Ramada Plaza, Jeju, Korea

3D Pose Tracking for GPS-denied Terrain Rovers by Fast State Variable Extension and Enhanced Motion Model

Nilesh Suriyarachchi[1], Peshala G. Jayasekara[1] and Takashi Kubota[2]

[1]Department of Electronic and Telecommunication Engineering, University of Moratuwa, Sri Lanka
[2]Spacecraft Engineering Division, Institute of Space and Astronautical Science (ISAS), JAXA, Japan

Abstract: Pose tracking for outdoor rovers is generally a complex task which is further complicated in conditions where a Global Positioning System (GPS) signal is denied such as in planetary exploration, underground mines and covered areas. In these conditions the rover's pose needs to be calculated purely based on the rover's current environment observations. However, conventional wheel odometry is not reliable on rough terrain where wheels are prone to slip and the wheels do not have a common plane of motion due to suspension systems. This paper proposes a Fast State Variable Extension (Fast-SVE) method in which 2D state variables (x, y, yaw) are extended to the full 3D state $(x, y, z, roll, pitch, yaw)$ to achieve effective real time 3D pose tracking of the rover. A particle filter implementation incorporating the Fast-SVE method is used to track the 3D pose of the rover with roll and pitch values used for weighting. An Enhanced Motion Model (EMM) is also proposed to further improve the accuracy of 2D pose prediction in the particle filter.

Keywords: Pose Tracking, Localization, State Variable Extension, Particle Filter, Odometry, Terrain Rover

1. INTRODUCTION

Pose tracking methods for terrain rovers are commonly based on GPS. However, in GPS denied situations such as in underground mines, under a forest canopy or in planetary exploration a different solution is needed for position tracking. In these type of situations the most frequently used tracking methods are based on the conventional odometry method [1]. The problem with this technique is that on rough terrain the rover's wheels tend to slip creating high accumulated wheel encoder errors. In addition to this issue most outdoor rovers have independent suspension for each wheel, which causes difficulty in predicting along what plane the rover chassis has actually moved.

The use of laser depth sensors allows us to obtain knowledge of the rover's terrain in the form of 3D point clouds. In the case of outdoor rovers a definite relationship exists between the terrain and the rover's position and orientation. This relationship is based on the fact that at a specific 2D pose $(x,\ y,\ yaw)$ of the rover, in order for the rover to be in contact with the terrain, there is only one possible 3D pose $(x,\ y,\ z,\ roll,\ pitch,\ yaw)$. Using this relationship a State Variable Extension (SVE) method [2] has been proposed in which 2D state variables are extended to the full 3D state using terrain knowledge. A particle filter based tracking system can be implemented incorporating the SVE method to obtain the 3D pose of the rover. However, particle filter based high accuracy pose tracking involves the maintainence of an adequately high particle count as well as a reasonably high filter update frequency. As the SVE based particle filter tracking process is quite computationally expensive, effective real time 3D tracking becomes very difficult. This paper proposes a Fast State Variable Extension (Fast-SVE) method which contains a more computationally efficient SVE process which enables high accuracy real time 3D pose tracking.

In order to perform SVE the 2D state of the rover needs to be predicted. This is usually carried out using conventional odometry tracking which uses both IMU measurements and wheel encoder readings. However, both these readings tend to be inaccurate over time due to accumulation of errors by wheel slippage and IMU drift. The paper introduces an Enhanced Motion Model (EMM) in order to improve 2D pose prediction of the rover. The EMM process implements improvements to the standard motion model [3] in order to counter these increasing errors. Initially the clustering of particles is increased by biasing low weight particles in the direction of the best pose of the previous particle filter iteration. A Gaussian kernal is also added to the predicted 2D pose of each particle. Furthermore, the EMM process selectively repurposes a portion of the lowest weight particles in order to track and correct wheel slippage using a Gaussian kernal centered around the best pose of the previous filter iteration. These improvements lead to improved tracking in low friction environments as well as the reduction of the total number of particles required in the particle filter. Together the proposed Fast-SVE method and the EMM provides highly efficient and accurate real time 3D pose tracking of an outdoor rover in GPS denied conditions.

2. RELATED WORKS

The work closest to this research is the work by Jayasekara et al. [2]. It presents the original SVE method. However, it is quite computationally expensive and struggles at performing real time 3D pose tracking with a sufficient number of particles for high accuracy tracking. This paper will look into improving the SVE method to be more computationally efficient thus enabling effective real time pose tracking. Accuracy of the 3D pose tracking process will also be improved upon with the use of the EMM for 2D pose prediction.

The paper by Suzuki et al. [4] presents an outdoor localization method for a mobile robot using a laser scanner and a 3D voxel map based on point clouds. This is applicable only if the four wheels of a rover stay on one plane which rarely occurs in terrain rovers as independent wheel suspension systems allow wheels to be moving in different planes. Methods used to detect and compensate for wheel slippage using visual odometry [5] [6] have been documented. However, due to the fact that visual odometry requires specific conditions such as good lighting conditions and well identifiable camera features, our method does not use visual odometry. A description of the difficulties and challenges faced in unknown terrain exploration, have been presented by Lacroix et al. [1] and Lamon [7]. This information along with the work done by Kmmerle et al. [8] on particle clustering and particle filter based localization was quite useful in formulating the EMM used to improve clustering and accuracy of the 3D pose estimation.

3. THE PROPOSED METHOD

3.1 Concept Overview

Considering the full 6 Degrees of Freedom (DOF) $(x,\, y,\, z,\, roll,\, pitch,\, yaw)$ of a terrain rover, a definite relationship between terrain dependent variables $(z,\, roll,\, pitch)$ of the rover and the terrain can be established given a specific independent location and heading $(x,\, y,\, yaw)$. Additionally, if actual values of roll and pitch can be identified using a sensor such as an IMU, a probabilistic filter may be used to track the rover via a predict-update cycle. However, since roll and pitch at multiple locations may be similar, multiple hypotheses of the rover's location needs to be maintained. Therefore, a particle filter may be used in these conditions with the weights for each particle based on a suitable observation model which takes into account the difference in estimated and actual roll and pitch values.

3.2 Fast State Variable Extension (Fast-SVE)

The original SVE model [2] describes the process of extending the 3 DOF $(x,\, y,\, yaw)$ 2D state (X_{2D}) of a rover to the full 6 DOF $(x,\, y,\, z,\, roll,\, pitch,\, yaw)$ extended state (X_{ext}) by adding the extension variables (X_{sve}) estimated using available knowledge of the terrain in the form of point cloud data.

$$X_{2D} = [x, y, yaw]^T \tag{1}$$

$$X_{sve} = [z, roll, pitch]^T \tag{2}$$

$$X_{ext} = [x, y, z, roll, pitch, yaw]^T \tag{3}$$

This SVE system is based on the fact that for given $(x,\, y,\, yaw)$ pose of the rover on a terrain there is a specific $(z,\, roll,\, pitch)$ value that the rover could take. As illustrated by Fig. 1, the rover should lie on the surface of the terrain and it can not lie sunk into the terrain or floating over the terrain.

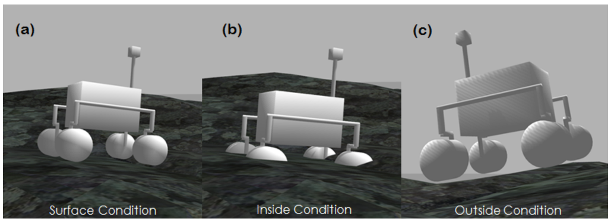

Fig. 1 The rover positioned on the (a) surface of the terrain, (b) sunk into the terrain or (c) floating over the terrain.

While the original SVE system had difficulty tracking a high number of particles in real time, the proposed Fast-SVE system is built based on multiple improvements and enhancements to the original SVE system, enabling it to perform real time 3D pose tracking of high particle counts thereby improving the efficiency and accuracy of the entire system.

3.3 Stages in Fast-SVE

The Fast-SVE process involves multiple stages.

1. Obtaining Terrain Point Cloud

The point cloud representing the terrain can be generated using a laser scanner and could be obtained in advance via a scout robot or in real-time using a laser scanner mounted in front of the rover. The density of the point cloud is then reduced by using a voxel grid filter in order to reduce the amount of calculations in the SVE process. The voxel grid size should be determined based on the size of the rover's wheels in order to ensure at least a few points lie within each wheel. Additionally, noise filtering may need to be carried out to remove outliers in the generated terrain point cloud.

In the SVE process, in order to reduce computation, the terrain point cloud is chopped in each particle filter iteration to include only terrain points that are located directly below the rover. This is done by considering an area of $(Rover\,position - R)\, to\, (Rover\,position + R)$ of the point cloud in both x and y directions, where R is the maximum distance from the center of the rover to the radius of the circumscribed circle around the rover as shown in Fig. 2.

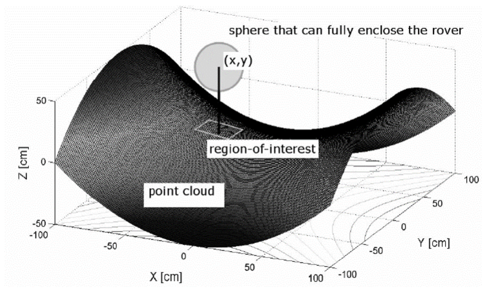

Fig. 2 Chopped point cloud

2. Optimization

Since an arbitrary terrain cannot be analytically specified, a numerical optimization technique needs to be used in order to calculate X_{ext} from the X_{2D} state of the rover by adding the extension state variables X_{sve}.

However, since the value of state variables cannot change suddenly in a high frequency update system the previous values of X_{sve} is a suitable starting point for the optimization process and therefore a local search may be performed around the previous values of X_{sve}. The original SVE method [2] used a continuous space hill climb algorithm [9] while the proposed Fast-SVE system uses a modified hill climb algorithm. While the original hill climb algorithm iterates over a range of possible values for the X_{sve} variables, the modified hill climb algorithm identifies the cost of the rover having a certain X_{sve} pose and optimizes in a direction which would minimize this cost. Therefore, based on the cost of each hill climb iteration the modified algorithm determines the direction in which it should optimize the selected variable. This reduces the total number of iterations per X_{sve} variable in the hill climb optimization phase of SVE and thus improves the real time operation capability of the Fast-SVE system.

The original SVE system [2] also incorporated a state variable known as rocker angle in the extended state X_{ext} of the rover. This variable, which represents the angle of the rocker suspension system of a terrain rover was used in the measurement model of the particle filter. However, upon analysis it was identified that removing the rocker angle from the X_{ext} state greatly improves the time efficiency of the overall system. No loss in tracking capability was identified after rocker angle removal. Thus, as the rocker angle is an externally added variable, which is not included in the 6 DOF state of the rover, the Fast-SVE algorithm does not consider rocker angle in its X_{ext} state.

3. Cost Calculation

The positioning of each wheel of the rover relative to the terrain is considered in selecting the most suitable X_{sve} state variable values. The wheels of the rover are modeled as cylinders and a point inside cylinder test is carried out for all relevant points in the terrain point cloud. Costs are then assigned as follows;

- $Inside\,Cost$ - Minimum distance from center of wheel to a point when the points are inside the cylinder. High cost. Rover has sunk into terrain.
- $Outside\,Cost$ - Minimum distance from center of wheel to a point when the points are outside the cylinder. Medium cost. Rover is floating over the terrain.
- $Surface\,Cost$ - Minimum distance from center of wheel to a point when the points are on the cylinder surface. Low cost. Rover is on the terrain.

Total cost is then defined as,

$$Total\,cost = Inside\,Cost + Outside\,Cost + Surface\,Cost \quad (4)$$

Both inside and outside cost conditions would result in a high total cost. Additionally a region from $(r-\gamma)\,to\,(r+\gamma)$ (where r is the wheel radius and γ represents the point cloud noise) is used to account for the effect of point cloud noise in the surface cost calculation.

This cost calculation is one of the most time consuming processes in the SVE method as it needs to be repeated for every point in the considered point cloud region. The Fast-SVE algorithm optimizes this process by only calculating costs for points lying within the horizontal cross section of each wheel as shown in Fig. 3.

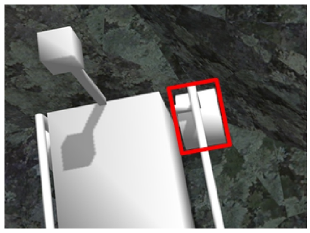

Fig. 3 Horizontal cross section around wheel

This process is carried out by identifying two unit vectors, one along the axis of the wheel and the other perpendicular to the wheel axis and on the horizontal plane. Each point in the chopped point cloud is then tested to identify whether it lies within the wheel's cross section. Two dot products between each unit vector and the vector from the edge of the wheel to the test point are obtained. If these dot products are less than the wheel thickness or wheel diameter respectively, the test point is considered to be within the wheel's cross section. This process drastically improves the computational efficiency of the Fast-SVE method and also ensures that the complexity of the Fast-SVE process becomes independent on rover size and only dependent on the size of the rovers wheels. This is especially important in planetary exploration rovers which tend to have a large overall size but a significantly smaller wheel size.

Most terrain rovers utilize a 'go straight' and then 'rotate in place' type of motion. The Fast-SVE algorithm considers the possibility that the rover may be in the 'rotate in place' motion condition and ensures that rover pose tracking is unaffected during this rotational motion. This is achieved by identifying the steering angle of the rover and computing the exact wheel position before performing the SVE process. Here, it is assumed that the steering angle on all wheels is constant as the rover is rotating in place. The addition of rotation invariance does not improve SVE efficiency but rather ensures better tracking accuracy.

3.4 Implementation of the Particle Filter for 3D Pose Tracking

In order to carry out 3D pose tracking of the terrain rover with incorporated Fast-SVE algorithm, a Sequence Importance Resampling (SIR) particle filter [10] was used.

$$\overline{bel(X_k)} = \int p(X_k|X_{k-1}, u_k)bel(X_{k-1})dX_{k-1} \quad (5)$$

$$bel(X_k) = p(X_k|z_{1:k}, u_{1:k}) = \eta p(z_k|X_k)\overline{bel(X_k)} \quad (6)$$

Here u is the input to the rover, z is the measurements obtained from sensors, η is the coefficient for normalization, X_k is the state of the robot at time k and $\overline{bel(X_k)}$ is the predicted X_{2D} pose from the motion model of the particle filter.

The key stages of the particle filter with N particles are,

1. Enhanced Motion Model (EMM)

   The x, y and yaw value of each particle at time k is calculated based on the values at time $k-1$ and the inputs to the rover obtained during the relevant time step. The conventional rover wheel odometry model [1] is modified to achieve the enhanced motion model. Considering that the rover motion takes the form of 'go straight' or 'rotate in place' commands, the rover is assumed to have a position t_k and rotation matrix R_k at time k. R_k and t_k are calculated based on wheel encoder readings and gyroscope readings based on the conventional odometry model. Therefore, the transform matrix T_k of the rover state at time k is given by,

$$T_k = \left[\begin{array}{c:c} R_k & t_k \\ \hdashline 0 & 1 \end{array}\right] \quad (7)$$

   The relative transform $T_{k-1:k}$ between T_k and T_{k-1} is then given by,

$$T_{k-1:k} = [T_{k-1}]^{-1} * T_k \quad (8)$$

   State transition of individual particles is carried out using the relative transform $T_{k-1:k}$. Additionally, in order to propagate particles further to accommodate for conventional odometry drift a Gaussian kernel is added to the predicted x, y and yaw values at time k.

   Due to the uneven nature of the environment it was observed that a significant number of particles tend to degenerate quite quickly if they stray from the path of the rover. The EMM attempts to improve the probability of survival of these particles with low effectiveness (low weight) which would otherwise simply be deleted in the resampling phase of the next particle filter iteration. This is carried out by selecting particles with very low effectiveness and then providing these particles with a mean shift in the direction of particles with high effectiveness. This also leads to a more concentrated group of particles.

   Additionally certain areas of the rover's environment with low friction in which wheel slippage occurs such as steep slopes and loose gravel cause wheel encoder based odometry to become highly inaccurate. In these conditions a backup strategy is required to track the rover's 2D pose. The EMM process selectively repurposes a portion of the lowest weight particles to achieve this task. After the prediction cycle of the particle filter, particles with the lowest weights are deleted and replaced by particles generated from a Gaussian kernal centered around the best pose of the previous particle filter iteration. This system identifies slippage by checking whether the rover is still within a window of its previously known position. Therefore when such a slippage occurs the rover's correct pose is detected and the particles which have propogated away from the rover's actual position due to incorrect odometry values are then provided with a mean shift back towards the correct rover position. The EMM thus ensures that accurate pose tracking is maintained even in low friction conditions.

2. Fast-SVE Application

   This is the stage in which Fast-SVE is carried out on the particles having X_{2D} state in order to extend them into the full X_{ext} state.

3. Measurement Model

   The measurement model carries out the process of weighting each particle based on the error between the particles estimated roll and pitch value from the Fast-SVE system and the measured roll and pitch values obtained from IMU sensors. These importance weights $\widehat{w}_k^{(i)}$ are calculated as follows,

$$\widehat{w}_k^{(i)} = w_{k-1}^{(i)} p(z_k|X_k^{(i)}). \quad (9)$$

   Since roll and pitch state variables can be modeled as Gaussian as in [2], $p(z_\alpha|X_\alpha)$ may be modeled as follows where α is either roll or pitch and σ_α is IMU noise.

$$p(z_\alpha|X_\alpha) \propto \frac{1}{\sigma_\alpha\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{X_\alpha - z_\alpha}{\sigma_\alpha}\right)^2\right] \quad (10)$$

   Assuming measurements of roll and pitch are independent of each other, the likelihood $p(z_k|X_k)$ of the probabilistic measurement model can be derived as follows.

$$p(z_k|X_k) \propto p(z_{roll}|X_{roll})p(z_{pitch}|X_{pitch}) \quad (11)$$

$$p(z_k|X_k) \propto \exp\left[-\frac{1}{2}\sum\left(\frac{X_\alpha - z_\alpha}{\sigma_\alpha}\right)^2\right] \quad (12)$$

   Normalizd weights $w_k^{(i)}$ are then calculated from the weights assigned to each particle by the measurement model.

$$w_k^{(i)} = \frac{\widehat{w}_k^{(i)}}{\sum_{i=1}^{N} \widehat{w}_k^{(i)}} \quad (13)$$

4. Resampling Stage

   The process of resampling is carried out if the number of effective particles (N_{eff}) fall below a defined threshold.

$$N_{eff} = \left[\sum_{i=1}^{N}\left(w_k^{(i)}\right)^2\right]^{-1} \quad (14)$$

   In the proposed implementation the systematic resampling technique [11] is used to carry out the resampling process. Even though this process is computationally expensive, the frequency of resampling is reduced due to the EMM modification.

4. SIMULATIONS AND RESULTS

The Gazebo framework [12] was used to develop a simulation of a terrain rover and its 3D environment as shown in Fig. 1. 3D pose tracking using the Fast-SVE with EMM system was implemented and verified based on results obtained from the simulator. The computer used for simulations and testing was an Intel i7 6th generation 4.2GHz PC with 16GB of RAM.

4.1 Fast-SVE Improvements and Validation

Multiple improvements allowed the Fast-SVE method to achieve a 4.5x speedup over the original SVE system [2]. Due to this speedup the Fast-SVE method is capable of handling upto 4.5 times more particles or a 4.5 times increase in filter update frequency leading to more accurate 3D pose tracking.

In the simulation the rover model was driven over a rough terrain and in each time step the X_{sve} states were estimated and then compared with their ground truth values. Results from the comparison of ground truth (actual) values with estimated values (from Fast-SVE) for the three X_{sve} state variables are shown in Fig. 4. As evidenced in Fig. 4 the estimated values of the X_{sve} state variables closely follow their acual ground truth values. This indicates that the Fast-SVE system is capable of obtaining a significant speedup without any noticable loss in tracking capability. The overall Root Mean Square (RMS) error in estimation for each X_{sve} state variable is as follows,

- $Z_{RMSE} = 3.469$ mm
- $Roll_{RMSE} = 0.668$ deg
- $Pitch_{RMSE} = 0.759$ deg

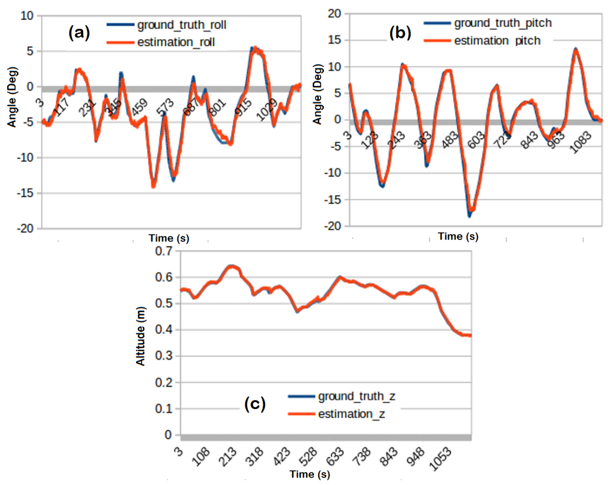

Fig. 4 Comparison of (a) roll, (b) pitch and (c) z state variables.

4.2 3D Pose Tracking Validation using Fast-SVE

The particle filter based 3D pose tracking system with incorporated Fast-SVE and EMM was tested by driving the rover model over a simulated rough terrain using both 'go straight' and 'rotate in place' forms of movement commands. Multiple tests were conducted with varying numbers of particles and varying levels of terrain friction. Additionally, for comparison the same test was carried out under identical conditions for the basic conventional odometry (Conv-Odom) based tracking method as well as for the original SVE (Orig-SVE) [2] based tracking method.

Figure 5 shows the generated plots containing the error in tracking for all three tracking methods mentioned above. Here, error in tracking is defined as the difference between the absolute 3D pose and the estimated 3D pose of the rover. These tests were carried out using a friction value of 0.8 to allow the wheels to slip on the terrain. The particle count in the particle filter was set to 25 particles for the first test and to 50 particles for the second test. These levels of low particle counts are acceptable in SVE based particle filter systems since only the X_{2D} state variables are used in the prediction stage of the particle filter (since SVE reduces the number of state variables needed to be predicted).

The results in Fig. 5 show that overall the Fast-SVE algorithm performs better than the other two methods and has a significantly lower mean error. The Conv-Odom method performs the worst especially since the error grows rapidly due to slipping caused by low friction surfaces. The Fast-SVE method also shows improved performance when the particle count is increased from 25 to 50 particles while the original SVE method has a reduction in performance.

This significant relative performance gain of the Fast-SVE algorithm compared to the original SVE algorithm, with increasing number of particles, is due to the fact that even though mean error in pose tracking reduces with increasing particle count, the original SVE algorithm struggles to perform computations in real time at high particle counts. The mean errors of three tracking methods for both particle counts are given in Table 1.

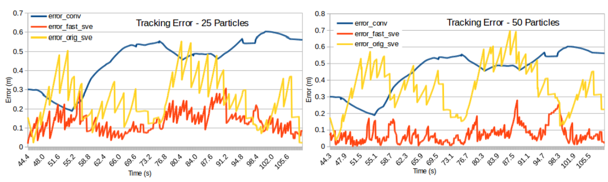

Fig. 5 Comparison of Fast-SVE, Orig-SVE and Conv-Odom based tracking error using 25 particles and 50 particles

Table 1 Mean error (m) for different particle counts

Particle Count	Conv-Odom	Original-SVE	Fast-SVE
25 Particles	0.450	0.245	0.127
50 Particles	0.450	0.359	0.073

The effect of friction on the 3D pose tracking process was also tested. Friction values used were 0.5 for low friction and 0.9 for high friction. Results of these friction modification tests are shown in Fig. 6. In these tests the particle count was kept at 25 particles to ensure consistency in comparison. The mean errors in the 3D pose tracking process for all three methods under varying friction values is shown in Table 2. The results show that, as expected, tracking accuracy increases with increasing levels of friction. It is also evident from the values in Table 2 that the Fast-SVE method outperforms the other methods over both friction conditions. The low friction plot clearly indicates the action of the EMM as when a large slip occurs at time 120s the Fast-SVE with EMM method is capable of maintaining accurate tracking while both other systems fail. The observed trend is that the relative benefits of these SVE based systems increase as the value of friction reduces.

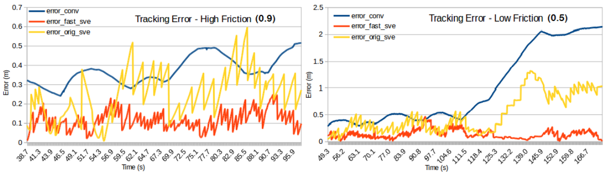

Fig. 6 Comparison of Fast-SVE, Orig-SVE and Conv-Odom based tracking for high friction and low friction values

Table 2 Mean error (m) for different friction levels

Friction level	Conv-Odom	Original-SVE	Fast-SVE
High Friction	0.372	0.249	0.117
Low Friction	1.000	0.516	0.143

5. CONCLUSION

In this paper we have proposed a Fast State Variable Extension (Fast-SVE) method in which 2D state variables (x, y, yaw) are extended to the full 3D state $(x, y, z, roll, pitch, yaw)$ using terrain knowledge, to achieve real time 3D position tracking of a terrain rover with the use of a particle filter. An Enhanced Motion Model (EMM) was also proposed to improve the accuracy of 2D state variable prediction which enabled high accuracy tracking even in low friction environments and reduced the need for very high particle counts. The Fast-SVE system with EMM is therefore able to ensure accurate real time tracking via increases in particle counts and faster particle filter update frequencies.

This system is limited on its dependancy on accurate terrain knowledge. Terrain knowledge can be improved by using high quality depth sensors leading to low point cloud noise. The system also performs poorly on extensively flat terrain since particles weighted based on roll and pitch will all be assigned equal weights. To address this issue other measurements like x, y or yaw obtained from external verification sources such as slip detection and laser based pointcloud odometry should be used in the weighting process of the particle filter when the rover is on flat terrain.

REFERENCES

[1] S. Lacroix, A. Mallet, D. Bonnafous, G. Bauzil, S. Fleury, M. Herrb, and R. Chatila, "Autonomous rover navigation on unknown terrains: Functions and integration", in *The International Journal of Robotics Research*, vol. 21, no. 10-11, pp. 917-942, 2002.

[2] Peshala G. Jayasekara, Genya Ishigami and Takashi Kubota, "Particle Filter based 3D Position Tracking for Terrain Rovers using Laser Point Clouds", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2369-2374, 2014.

[3] Sebastian Thrun, Wolfram Burgard, and Dieter Fox, "Probabilistic Robotics", The MIT Press, 2005.

[4] T. Suzuki, M. Kitamura, Y. Amano, and T. Hashizume, "6-dof localization for a mobile robot using outdoor 3d voxel maps", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5737-5743, 2010.

[5] M. Maimone, Y. Cheng, and L. Matthies, "Two years of visual odometry on the mars exploration rovers", in *Journal of Field Robotics*, vol. 24, no. 3, pp. 169-186, 2007.

[6] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry in Computer Vision and Pattern Recognition", in *Proceedings of the 2004 IEEE Computer Society Conference*, vol. 1. IEEE, pp. I-652, 2004.

[7] P. Lamon, "3d position tracking for all-terrain robots", in *Ph.D. dissertation, E'COLE POLYTECHNIQUE FE'DE'RALE DELAUSANNE*, 2005.

[8] Rainer Kmmerle, Rudolph Triebel, Patrick Pfaff, and Wolfram Burgard, "Monte Carlo localization in outdoor terrains using multilevel surface maps", in *Journal of Field Robotics*, vol. 25, no. 6-7, pp. 346-359, 2008.

[9] D. S. Johnson, C. H. Papadimitriou, and M. Yannakakis, "How easy is local search?", in *Journal of Computer and System Sciences*, vol. 37, no. 1, pp. 79-100, 1988.

[10] N. J. Gordan, D. J. Salmond, and A. F. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation", in *IEE Proceedings F (Radar and Signal Processing)*, vol. 140, no. 2, pp. 107-113, IET, 1993.

[11] M. S. Arulampalam, S. Maskell, N. Gordan, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking", in *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174-188, 2002.

[12] P. G. Jayasekara, G. Ishigami, and T. Kubota, "Testing and validation of autonomous navigation for a planetary exploration rover using opensource simulation tools", in *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2012.