



NASA Public Access

Author manuscript

IEEE Robot Autom Lett. Author manuscript; available in PMC 2022 July 01.

Published in final edited form as:

IEEE Robot Autom Lett. 2021 July ; 6(3): 4782–4789. doi:10.1109/lra.2021.3068893.

Slip-Based Autonomous ZUPT through Gaussian Process to Improve Planetary Rover Localization

Cagri Kilic, Nicholas Ohi, Yu Gu, Jason N. Gross

Department of Mechanical and Aerospace Engineering, West Virginia University, Morgantown, WV 26506 USA

Abstract

The zero-velocity update (ZUPT) algorithm provides valuable state information to maintain the inertial navigation system (INS) reliability when stationary conditions are satisfied. Employing ZUPT along with leveraging non-holonomic constraints can greatly benefit wheeled mobile robot dead-reckoning localization accuracy. However, determining how often they should be employed requires consideration to balance localization accuracy and traversal rate for planetary rovers. To address this, we investigate when to autonomously initiate stops to improve wheel-inertial odometry (WIO) localization performance with ZUPT. To do this, we propose a 3D dead-reckoning approach that predicts wheel slippage while the rover is in motion and forecasts the appropriate time to stop without changing any rover hardware or major rover operations. We validate with field tests that our approach is viable on different terrain types and achieves a 3D localization accuracy of ~97% over 650 m drives on rough terrain.

Index Terms—

Space Robotics and Automation; Localization; Planetary Rovers; Zero Velocity Update

I. Introduction

ACHIEVING accurate real-time localization performance is challenging for planetary rovers with limited-performance computers traversing on harsh and unknown terrains that cause wheel slippage. Rover slip is often estimated using visual odometry (VO) [1], [2]. Despite its safety and reliability, using VO for long periods comes with some concerns: 1) substantial traversal rate reduction since the rover needs to stop to acquire images [3], and needs to drive slow due to limited computational resources [4]; 2) the low number of detected and tracked features on indistinguishable terrains (e.g., bright areas, sand dunes, shadowed areas) can lead to poor accuracy of motion estimates [5] and limit the usage of VO. Specifically, Mars Science Laboratory (MSL) rover reaches a maximum speed of 140 m/h in blind-drive mode (without VO), 45 m/h in hazard avoidance mode (VO update every 10 meters), and only 20 m/h in fully autonomous mode (VO update every half-vehicle length) [6].

Personal use is permitted, but republication/redistribution requires IEEE permission. See <https://www.ieee.org/publications/rights/index.html> for more information.

cakilic@mix.wvu.edu.

For current Mars rovers, the slow pace driving can be alleviated by using the blind-driving mode, which makes use of wheel odometry (WO) and inertial measurement unit (IMU) to keep track of the rover's motion if the terrain ahead is considered to be safely traversable by the rover operation team. However, using only blind-driving causes unbounded pose error growth over time and increasing uncertainty in the rover state due to wheel slippage and INS drift. For this reason, the rover localization is corrected with computationally expensive methods after a short period of blind-driving [1].

Leveraging “free” information without affecting any other operations and using observations for multiple purposes are desirable characteristics for planetary missions [7]. In planetary missions, stopping is inevitable for the rovers due to hardware constraints, and so far, the autonomous planetary rovers are stopping approximately every 1–10 meters of driving for various reasons [3], [6]. As the rover is mostly stationary due to these frequent stops, ZUPT can be leveraged to maintain INS alignment. The main advantages of ZUPT for the localization task is that it can bound the velocity error, calibrate IMU sensor biases, and limit the rate of INS localization drift [8]. Using ZUPT in a planetary rover dead-reckoning system can provide a computationally efficient and accurate real-time rover localization capability, even in feature-poor areas, without any major changes to the rover operations. Furthermore, having a more reliable onboard proprioceptive localization approach may help to reduce the frequency of using computationally expensive visual-based corrections. However, knowing how often ZUPT should be employed requires consideration to avoid unnecessarily reducing traverse rate.

In our previous work [9], we presented an approach to enhance planetary rover dead-reckoning localization performance by making use of ZUPT with periodic stops. In this study, we propose an autonomous stopping framework by monitoring wheel slippage and predicting the time when the rover needs to stop to keep the localization drift rate to an acceptable level using only an IMU and wheel encoders. Our contributions are listed as:

- We develop a novel method for predicting localization error, using a time-series Gaussian process model for prediction of slip uncertainty as a function of time, such that ZUPTs can be actively initiated with respect to the wheel slippage frequency and magnitude.
- We evaluate our approach in a set of field tests and demonstrate that the proposed method is able to improve blind-driving localization on different terrain types (e.g., paved, unpaved, graveled, and rough areas) that yield different stopping times.
- We make our software (designed using Robot Operating System [10]), and datasets publicly available in [11].

The rest of the paper is organized as follows. Section II provides a comprehensive overview of related works. In Section III, we introduce the preliminaries for the problem formulation. In Section V, we describe the details of the proposed framework. Section VI explores the concept further and carries out a qualitative analysis of experimental results. Finally, conclusions are presented in Section VII.

II. Related Work

Wheel slippage can occur when the terrain traversed fails [12] or when there is a kinematic incompatibility between wheels (i.e., different wheel speeds) encountered [13]. Because of slippage and imperfect measurement of the wheel radius, VO based localization is inherently subject to drift.

Knowledge of the terrain geometry is a critical asset for the rovers in unknown environments for safe traversal. For example, MSL uses stereo vision to generate a digital elevation map (DEM) of the surrounding terrain enhanced by leveraging High Resolution Imaging Science Experiment (HiRISE) images [14] similar to Mars Exploration Rovers (MERs) [2]. VO is an accurate and reliable source of information for slip estimation; however, it is computationally expensive for planetary rovers. Even with the field-programmable gate array (FPGA) processors [15], the other limitations of VO arise that it suffers from low-feature terrains and it relies on proper lighting conditions [16]. Similarly, insufficiently detected and tracked features may lead to poor accuracy of motion estimate [5].

Various studies have modeled slip as a function of terrain geometry. Past studies have yielded important insights into the relationship between visual terrain information and the measured slip using training examples by casting the problem into a Mixture of Experts (MoE) framework [17]. However, this terrain geometry knowledge does not guarantee to localize the rover relative to terrain traversed since the rover slip is measured infrequently, and it causes a substantial reduction of the traversal rate due to computational expenses [3].

Moreover, the wheel-terrain interactions (terramechanics) are not dictated by the visible topsoil of the terrain [5]. To address this, a recent line of research has focused on data-driven cubic regression metrics to predict slip with respect to the slope by using proprioceptive and exteroceptive sensors [18]. Although slippage is strongly affected by increasing absolute value of a slope, wheel slippage can also be observed on flat terrains while encountering local obstacles (e.g., small rocks that rover can traverse on) due to kinematic incompatibility [13].

Martian soil is extremely challenging for traversability; even throughout a single drive, Mars rovers traverse various terrains [14]. Employing a terramechanics model to estimate slip requires the knowledge of terrain parameters and variables, which are challenging to measure or estimate accurately online. Due to the complexity of terramechanics modeling, considerable research has been devoted to simplified models. For example, [12] presented a tool for online estimation of terrain parameters based on a simplified terramechanics model for deformable terrains.

Apart from terramechanics modeling, machine learning algorithms have also been utilized as slip estimation tools. Locally adaptive slip-model learning with respect to slope values is demonstrated in [19] using a Gaussian process (GP) regression for visually classified terrain types. Using visual information is one of the common ways to classify a terrain and estimate an equivalent slip value for planetary missions. However, unexpected small variances on the terrain can be deceptive for a vision based slip-learning approach [20].

The methodology in [21] demonstrated an offline wheel slippage learning approach, where the model is learned on training runs and evaluated in a test environment using SLAM in a planetary rover navigating an unstructured environment. On the other hand, [22] suggested that the mapping between inputs and resultant behavior depends critically on terrain conditions which vary significantly over time and space (spatio-temporal). Therefore, offline techniques for slip estimation are most likely to suffer from learning changes in wheel-terrain interactions.

Leveraging ZUPT is a natural fit for wheeled planetary robots because rovers are in stationary conditions in many instances [23] such as capturing images for obstacle avoidance, re-planning, processing VO, and conducting scientific experiments. When a rover is in stationary conditions, localization performance can be improved by using the pseudo-measurements generated (i.e., ZUPT) as detailed in our previous work [9]. ZUPT is a well-known concept that was initially popularized to aid inertial pedestrian navigation [24], [25]. Zero-velocity detection and application on paved road for automobile applications are shown in [26]–[28].

III. Preliminaries

This section introduces several essential framework elements for planetary rover proprioceptive localization from our previous study for the sake of completeness. Detailed descriptions can be found in [9].

A. Rover Filter States

An error state extended Kalman filter (ES-EKF), based on the method detailed in [8], is implemented to enhance proprioceptive localization and provide uncertainty bounds. The error state vector is formed in a local navigation frame,

$$\mathbf{x}_{err}^n = \left(\delta\Psi_{nb}^n \ \delta\mathbf{v}_{eb}^n \ \delta\mathbf{p}_b \ \mathbf{b}_a \ \mathbf{b}_g \right)^T \quad (1)$$

where, $\delta\Psi_{nb}^n$ is the attitude error, $\delta\mathbf{v}_{eb}^n$ is the velocity error, $\delta\mathbf{p}_b$ is the position error, \mathbf{b}_a is the IMU acceleration bias, and \mathbf{b}_g is the IMU gyroscope bias.

It is assumed that the error-state vector is defined by (1) and the total state vector is

$$\mathbf{x}^n = \left(\Psi_{nb}^n \ \mathbf{v}_{eb}^n \ \mathbf{p}_b \right)^T \quad (2)$$

where each of the nine total states correspond to the first nine error-states.

B. Non-Holonomic Constraints

A non-holonomic rover is subjected to two motion constraints: 1) zero velocity along the rotation axis of the rover wheels, and 2) zero velocity in the direction perpendicular to the traversed terrain [29]. These constraints can be leveraged as a pseudo-measurement update. Assuming that the rear-wheel frame axes are aligned with the body frame, this measurement update can be given as

$$\delta \mathbf{z}_{RC}^n = - \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} (\mathbf{C}_n^b \mathbf{v}_{eb}^n - \omega_{ib}^b \times \mathbf{L}_{rb}^b) \quad (3)$$

where \mathbf{C}_n^b is the coordinate transformation matrix from the body frame to the locally level frame, \mathbf{L}_{br}^b is body to rear wheel lever arm, and ω_{ib}^b is angular rate measurement. The approximate measurement matrix can then be found as

$$\mathbf{H}_{RC}^n = \begin{pmatrix} \mathbf{0}_{2,3} & \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{C}_n^b & \mathbf{0}_{2,3} & \mathbf{0}_{2,3} & \mathbf{0}_{2,3} \end{pmatrix} \quad (4)$$

Note that the lateral velocity constraint is invalid in excessive sideslip conditions. The sideslip angle estimation (see Subsection V-A) can be used to verify whether the rover is experiencing an excessive sideslip and this verification can be used to decide the lateral velocity measurement should be omitted or not.

C. Zero-Velocity Update - (ZUPT)

During stationary conditions, IMU output is dominated by planetary rotational motion and sensor errors. Therefore, ZUPT can be used to maintain INS accuracy.

ZUPT bounds the velocity error and calibrates IMU sensor biases [30]. Hence, the measurement innovation for ZUPT can be expressed as

$$\delta \mathbf{z}_{Z,k}^n = [-\hat{\mathbf{v}}_{eb,k}^n, -\hat{\omega}_{ib,k}^b]^T \quad (5)$$

where $\delta \mathbf{z}_{Z,k}^n$ is measurement innovation matrix, $\hat{\mathbf{v}}_{eb,k}^n$ is estimated velocity vector, and $\hat{\omega}_{ib,k}^b$ is estimated gyro bias. The measurement matrix is given as

$$\mathbf{H}_{Z,k}^n = \begin{bmatrix} \mathbf{0}_3 & -\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & -\mathbf{I}_3 \end{bmatrix}. \quad (6)$$

IV. Gaussian Process with Time-Series Modeling Overview

In this study, we employ a GP to model the wheel slippage as time-series data. The primary reason for choosing the GP is to leverage its prediction of uncertainty estimates, which are used for predicting the error-covariance of odometry measurements (see Section V-D).

A GP is uniquely defined by its mean function $\mu(x)$ and covariance function $k(x, x')$ [31].

$$f(x) \sim GP(\mu(x), k(x, x')) \quad (7)$$

For any collection of input points, $\mathbf{x} = \{x_1, \dots, x_n\}$, with defining a probability distribution $p(f(x_1), \dots, f(x_n))$, has a joint Gaussian distribution such that

$$p(f(x_1), \dots, f(x_n) \mid x_1, \dots, x_n) = \mathcal{N}(\boldsymbol{\mu}, \mathbf{K}) \quad (8)$$

where the matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$ is the kernel matrix whose entries are given by $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, $i, j = 1, \dots, n$, and $\boldsymbol{\mu}$ is the corresponding mean vector. The covariance (kernel) function encodes the similarity between the outputs in GP [32]. To model the different characteristics of the training dataset, which is collected while the rover is in motion, we combine two kernels as a product to capture the different slip behavior of the rover with respect to the terrain.

Assuming that the slip can be occurred randomly and significantly (e.g., impulsive high slippage) due to unexpected kinematic incompatibility, we adopted the Brownian kernel, $k_B = \min(t, t')$. On the other hand, from the mathematical expression of Radial Basis Function (RBF) kernel, $k_{RBF} = \exp(-\|t - t'\|^2/2\ell^2)$, it can be assumed that if inputs are similar, then the outputs would be similar [33]. In the case that the rover does not encounter significant slippage, we assumed the subsequent measurements should be similar to each other for a short time-interval (the time-interval between two successive slip measurements is 0.1s in our setup) resulting to a repetitive-low slippage. Based on this intuition and a heuristic approach from field test results, we also used RBF kernel in our GP model, resulting in a composite kernel (i.e., multiply kernels together) [32] such as $k(t, t') = k_B(t, t') k_{RBF}(t, t')$. Note that the assumptions mentioned above are for blind-driving mode, and the mode can be activated when the terrain is considered safe to be driven for planetary rovers. The aim of a regression problem is to learn the mapping from inputs to outputs [34], given a training set of input and output pairs $(x, y) = (x_i, y_i)_{i=1}^N$, where N is the number of training examples, predictions can be made at test indices \mathbf{x}_* by computing the conditional distribution and with assuming a zero mean $\epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2)$, results in a Gaussian distribution and given by:

$$p(\mathbf{y}_* \mid \mathbf{x}_*, \mathbf{x}, \mathbf{y}) = \mathcal{N}(\mathbf{y}_* \mid \boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*) \quad (9)$$

where

$$\boldsymbol{\mu}_* = \mathbf{K}_*^T \mathbf{K}_*^{-1} \mathbf{y}, \quad \mathbf{K}_* = \mathbf{K}(\mathbf{x}, \mathbf{x}_*) \quad (10)$$

$$\boldsymbol{\Sigma}_* = \mathbf{K}_{**} - \mathbf{K}_*^T \mathbf{K}_*^{-1} \mathbf{K}_*, \quad \mathbf{K}_{**} = \mathbf{K}(\mathbf{x}_*, \mathbf{x}_*). \quad (11)$$

V. Methodology

The proposed wheeled-robot localization framework consists of a series of actions in current-time and future-time, both of which are computed onboard the rover. The current-time portion consists of our previous work [9], an INS mechanization aided with WO, pseudo-measurements, and kinematic constraints in an ES-EKF as briefly summarized in Section III. The future-time part of the framework uses the estimated slip events and prior estimated error state information to predict the robot's localization error. A depiction of the proposed framework and its elements is demonstrated in Fig. 1.

A. Slip Detection

The slippage is monitored with the slip ratio calculation for front and rear wheels velocity with respect to the INS velocity. Example estimates of WO based velocity, INS (filter) estimated velocity, and truth (DGPS) velocity are shown in Fig. 2 (a).

The longitudinal slip ratio, $s \in [-1, 1]$, is defined as:

$$s = \begin{cases} 1 - \frac{v_x}{r\omega} & (\text{if } \omega \neq 0, v_x < r\omega, \text{ driving}) \\ \frac{r\omega}{v_x} - 1 & (\text{if } \omega \neq 0, v_x > r\omega, \text{ braking}) \end{cases} \quad (12)$$

where v_x is the translational velocity estimated from INS, r is the wheel radius, and ω is the wheel angular velocity estimated from the WO measurements. The motion estimates from the filter are compared to the computed velocity based on the vehicle kinematics to determine if any slippage has occurred. Detected slippage input is demonstrated in Fig. 2 (b).

Also, sideslip can be expressed using the slip angle, β , and can be given as the angle between lateral velocity, v_y , and translational velocity

$$\beta = \tan^{-1}\left(\frac{v_y}{v_x}\right) \quad (13)$$

Although there are several methods to detect slippage as discussed in Section II, we adopt this proprioceptive slip detection since it is computationally efficient and not required any visual-sensor information to observe the wheel slippage for the proposed method.

B. Wheel Slippage with GP Time-Series Modeling

In our case, there is one input ($\mathbf{x} = \mathbf{T}$) and one output ($\mathbf{y} = \mathbf{s}$) in the GP. The input $\mathbf{T} = \{t_1, t_2, \dots, t_N\}$ is the time tags of each corresponding slip ratio value, and the output $\{s_1, s_2, \dots, s_N\} = \mathbf{s} \in [-1, 1]$ is the estimated slip ratio value, assuming training input and output pairs such that $\mathcal{D} = (\mathbf{T}, \mathbf{s})$.

The collected training data for wheel slip ratio values, $\mathbf{s} = \{s_1, \dots, s_N\}$, and corresponding time tags $\mathbf{T} = \{t_1, \dots, t_N\}$ for a time window are used to learn the model

$$s = f(t) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2). \quad (14)$$

The time window for learning is kept short to capture the most current (the last 12 m of drive) terrain-wheel information based on the MSL Hazard Avoidance slip check interval (~10 m) [6]. In that time window, the rover is in free driving (i.e., rover does not perform any stops). The learned model is then processed in the GP forecast model to make predictions at future test indices $\mathbf{t}^* = \{t_i^*\}_{i=1}^{N^+}$ for future unknown wheel slip ratio

observations $\mathbf{s}^* = \{s_i^*\}_{i=1}^{N^+}$ where N^+ is the number of test indices which in our case it corresponds to a future time tag. For a detailed demonstration of slip input and slip prediction by using the slip ratio definition, see Fig. 1(b). A python GP library [35] is used in our rover's ROS framework to optimize the hyperparameters (e.g., the length parameter l in the RBF kernel), and to predict the slip values while the rover is in motion.

C. Wheel Odometry Velocity Prediction

To predict the simulated odometry velocity error boundaries, a statistical sigma point transformation inspired by unscented transformation [36] where the slip ratio definition in (12) is used to generate this transformation function:

$$\sigma_{\chi_{est}(t)} = \frac{1}{N} \sum_{i=1}^N (\chi_{i(t)} - \mu_{\chi_{est}(t)})^2, \quad t \in [T_1^+, T_3] \quad (15)$$

where T_1^+ is the time when the prediction is being generated, T_3 is the time when the generated prediction ends (i.e., $T_3 = T_1 + 60$ s, see Fig. 1(c)), N is the number of the sigma points, χ_i is velocity term mapped from slip measurement, defined as $\chi_1 = \mu_{vel}(1 - \mu_s)$, $\chi_2 = \mu_{vel}(1 - \mu_s - \sigma_s)$, and $\chi_3 = \mu_{vel}(1 - \mu_s + \sigma_s)$ where μ_s and σ_s are mean and variance of s , respectively, and χ_{est} is the mean of χ_i values for $N=3$.

In constituting the observation noise covariance matrix in the localization forecasting phase, \mathbf{R}^{GP} , we assumed that the constant WO velocity related \mathbf{R} values on the filter could be interchangeable with varying $\sigma_{\chi_{est}}$ values between T_1^+ and T_3 come from predicted observation covariance.

$$\mathbf{R}^{GP} = \begin{bmatrix} (\sigma_{\chi_{est}}^2 \mathbf{I}_{3 \times 3}) & 0 \\ 0 & \mathbf{1}_{4 \times 4} \end{bmatrix} \quad (16)$$

These mapped velocity values and their prediction with this statistical sigma point transformation method are depicted in Fig. 1(c).

D. Forecasting Localization Error

When the forecasted GP data arrives, the algorithm uses the latest filter error covariance estimate, $\mathbf{P}_{T_1^-}$, to initialize the error covariance prediction.

$$\hat{\mathbf{P}}_0^{GP} = \mathbf{P}_{T_1^-} \quad (17)$$

The most recent state transition matrix, $\mathbf{F}_{T_1^-}$, process noise covariance, $\mathbf{Q}_{T_1^-}$, and WO observation matrix, $\mathbf{H}_{T_1^-}$ are being kept fixed during the forecasting error covariance process (see the left side of the Fig. 1).

$$\tilde{\mathbf{P}}_k^{GP} = \mathbf{F}_{T_1} \hat{\mathbf{P}}_{k-1}^{GP} \mathbf{F}_{T_1}^T + \mathbf{Q}_{T_1} \quad (18)$$

Then, the algorithm simulates an INS error covariance propagation. In our setup, simulated odometry update is assumed to take place in every 5th IMU time step (IMU data rate is 50 Hz, WO data rate is 10 Hz). When this simulated odometry update is available, transformation function predicts the simulated odometry velocity error boundaries. Finally, the simulated Kalman gain is calculated and simulated estimate covariance is updated.

$$\mathbf{K}_k^{GP} = \tilde{\mathbf{P}}_k^{GP} \mathbf{H}_{T_1}^T (\mathbf{H}_{T_1} \tilde{\mathbf{P}}_k^{GP} \mathbf{H}_{T_1}^T + \mathbf{R}^{GP})^{-1} \quad (19)$$

$$\hat{\mathbf{P}}_k^{GP} = (1 - \mathbf{K}_k^{GP} \mathbf{H}_{T_1}) \tilde{\mathbf{P}}_k^{GP} \quad (20)$$

For each updated covariance prediction, the algorithm calculates the position error covariances as a function of time. An example calculation is illustrated in Fig. 1(d). When the horizontal error gets more prominent than a predetermined threshold, the algorithm takes the corresponding time for that event, calculates the remaining time to stop with respect to the current time, and alerts the rover to stop. If there is no need for stopping (e.g., the positioning error prediction is below the threshold within the prediction time limit), the rover keeps driving. Otherwise, the rover stops traversing, applies ZUPT, then keeps driving. A detailed example scenario is given Fig. 2. The details to model state transition matrix \mathbf{F} , process noise covariance \mathbf{Q} , observation matrix \mathbf{H} and the observation noise covariance \mathbf{R} can be found in [9] and [8].

VI. Experimental Results

A. Setup

Pathfinder, a custom-built testbed rover, is employed for the experimental evaluation of the proposed method (see Fig. 3). The platform is a lightweight, 4-wheeled, skid-steered robot. Rover uses a rocker system with a differential bar connected to the front wheels. In general, planetary rovers use wheels with grousers, which increase traction and traversability performance (e.g., MSL, MERs, ExoMars). However, Pathfinder is utilized with slick wheels to test our localization algorithm against significant slippage. Slick wheels lead to encounter more slippage with larger frequency and occurrence which aid to detect slippage but degrade the localization performance significantly.

The IMU used on the rover is an ADIS-16495 with 50 Hz data rate [37] and the quadrature encoders are used for WO readings with 10 Hz data rate. Integer-ambiguity-fixed carrier-phase differential GPS (DGPS) is used to determine a truth reference solution. Just as in [9], dual-frequency Novatel GPS receivers and L1/L2 Pinwheel antennas [38] are mounted to the rover and a stationary base station. During the experiments, 10 Hz carrier-phase and GPS pseudorange measurements were logged on both receivers. Rover state is initialized with a loosely-coupled GPS-IMU sensor fusion algorithm, such as driving straight with a short

distance (~10 m) for estimating initial heading and being stationary for a period of time (~30 s) to initialize position before testing. After initialization, GPS measurements are collected only externally for generating the truth through post-processing. The open-source software library, RTKLIB 2.4.2 [39], is used to post-process the DGPS solutions with a cm-to-dm expected level accuracy [40]. Rover is teleoperated and commanded for 0.8 m/s forward speed in field tests.

B. Evaluation

A series of tests were performed on several terrains, including paved, unpaved, gravel, and rough terrains. Paved terrains are relatively flat roads with minimal slippage observation. Unpaved terrains are also rigid roads with small scattered rocks that rover can easily traverse. Gravel terrain consists of small broken rock materials. Due to the shape of these materials, there is less traction on the wheels on the gravel road. This loose surface creates slippage primarily due to wheel kinematic incompatibilities. This letter particularly focuses on the rough terrain results because of its similarities with the Martian terrain (see Fig. 3). This terrain is a burnt coal ash pile located at Point Marion, PA, with complex geometric (e.g., sloped, pitted, fractured, and sandy areas) and chemical terrain properties [41] similar to the abundant chemical compounds found in Martian regolith [42].

A stopping time comparison analysis against four terrain types is shown in Fig. 4. In this analysis, the rover is driven on different terrains and the corresponding stop time intervals are stored. Paved and unpaved roads are rigid, and the terrain underneath the wheels is not moving, and the robot wheels do not encounter significant slippage, resulting in better WO. However, the rover encounters significant slippage on gravel (kinematic incompatibility) and rough terrain (sinkage, slope, and kinematic incompatibility). The important result of this analysis is that the average stopping time intervals are shorter on gravel and rough terrain than on benign roads. Correspondingly, the algorithm enforces the rover stops more often on more slippery terrains (minimum stop frequency is 15 s).

To further evaluate the method, the localization accuracy of the proposed estimation is compared against the DGPS solution. As detailed in Table I, we achieved approximately 1% of 3D localization error (ENU) in short (152 m) and medium (339 m) range distances on rough terrain with keeping the stopping error threshold as 2 m. Also, in long (650 m) range distances, the threshold is varied as 2 m, 3 m, and 5 m to observe the localization accuracy performance against stopping time prediction. In these field test results, the algorithm reasonably predicts the stopping time to keep the localization drift approximately 3% for the 5 m threshold and less than 2% for the 3 m threshold. We also monitored that the rover often does not need to stop for the 5 m threshold due to not exceeding the threshold in the prediction time limit.

Ground-track depiction of an example scenario from ash-pile field testing is given in Fig. 5. The results show that traditional 2D dead-reckoning (WIO) is reliable only for short distances due to slippage, whereas the proposed estimation (3D WIO+ZUPT) can be used for longer distances if the terrain is safe to drive blindly. The localization design goal for MER was to maintain a position estimate that drifted less than 10% during a 100 m drive

[1]. Without using ZUPT and kinematic constraints in blind-driving, the drift can quickly elevate and exceed that design limit, as shown in Fig. 5.

Moreover, a comparison analysis between autonomous (proposed) and periodic [9] stopping methods is provided in Table II. Using autonomous stopping leads to an average stop rate (S_R) decrease over 65% compared to periodic stopping while keeping the localization accuracy more than 98%. Consequently, when using ZUPT, autonomous stopping increases the traversal rate by stopping less, and keeps the localization accuracy to an acceptable level.

A comparison between our localization approach against a commercially off-the-shelf RealSense T265 tracking system [43] visual-inertial odometry (VIO) solution is provided in Fig. 6. In this field test, rover traversed for 150 m on a low-feature terrain. The tracking system is able to provide reliable solution in feature rich areas whereas it suffers in the areas with a lack of detectable and trackable features. This is a common issue of visual-based localization approaches because these approaches require reasonable distinct visual features in view to operate accurately [2], [16], [44].

VII. Conclusion and Future Work

We presented a slip-based localization error prediction framework, which effectively balances the traversal-rate and localization accuracy for wheeled planetary rovers. Instead of periodic stopping, ZUPTs can be autonomously initiated with respect to the wheel slippage frequency and magnitude using a time-series GP model for prediction of slip uncertainty as a function of time. Planetary robot slip related localization drift can be alleviated with ZUPTs and can provide reliable localization performance for longer distances. The main value of the proposed approach is that it can be easily integrated into planetary rover operations (and many other wheeled robots) to improve onboard localization performance with no hardware changes and minimal operational changes. Since planetary rovers are already stopping frequently for using VO or other operational reasons, using ZUPT along with the blind-drive is a natural fit.

Future work will focus on 1) using deformable planetary spring tires with a traction control mechanism to help alleviating the limitation of the method when stopping on a steep slope and sliding down, 2) improving the method with adaptive and robust filtering techniques.

Collected dataset for experimental validation is available in [11] for the community to use. Developed software and supplementary analyses for this paper are available at: <https://github.com/wvu-navLab/CN-GP>.

Acknowledgments

This paper was recommended for publication by Editor Pauline Pounds upon evaluation of the Associate Editor and Reviewers' comments. This work was supported in part by NASA EPSCoR Research Cooperative Agreement WV-80NSSC17M0053, in part by NASA WVSGC Agreement 80NSSC20M0055, and in part by the Benjamin M. Statler Fellowship.

References

- [1]. Maimone M, Cheng Y, and Matthies L, “Two years of visual odometry on the mars exploration rovers,” *Journal of Field Robotics*, vol. 24, no. 3, pp. 169–186, 2007.
- [2]. Rankin A, Maimone M, Biesiadecki J, Patel N, Levine D, and Toupet O, “Driving curiosity: Mars rover mobility trends during the first seven years,” *2020 IEEE Aerospace Conference*, pp. 1–19, 2020.
- [3]. Toupet O, Biesiadecki J, Rankin A, Steffy A, Meirion-Griffith G, Levine D, Schadeegg M, and Maimone M, “Terrain-adaptive wheel speed control on the curiosity mars rover: Algorithm and flight results,” *Journal of Field Robotics*, 2019.
- [4]. Li R, Wu B, Di K, Angelova A, Arvidson RE, Lee I-C, Maimone M, Matthies LH, Richer L, Sullivan R et al., “Characterization of traverse slippage experienced by spirit rover on husband hill at gusev crater,” *Journal of Geophysical Research: Planets*, vol. 113, no. E12, 2008.
- [5]. Gonzalez R and Iagnemma K, “Slippage estimation and compensation for planetary exploration rovers. state of the art and future challenges,” *Journal of Field Robotics*, vol. 35, no. 4, pp. 564–577, 2018.
- [6]. Grotzinger JP, Crisp J, Vasavada AR, Anderson RC, Baker CJ, Barry R, Blake DF, Conrad P, Edgett KS, Ferdowski B et al., “Mars science laboratory mission and science investigation,” *Space science reviews*, vol. 170, no. 1–4, pp. 5–56, 2012.
- [7]. Arvidson R, Anderson R, Haldemann A, Landis G, Li R, Linde-mann R, Matijevic J, Morris R, Richter L, Squyres S et al., “Physical properties and localization investigations associated with the 2003 mars exploration rovers,” *Journal of Geophysical Research: Planets*, vol. 108, no. E12, 2003.
- [8]. Groves PD, *Principles of GNSS, inertial, and multisensor integrated navigation systems*. Artech house, 2013.
- [9]. Kilic C, Gross JN, Ohi N, Watson R, Strader J, Swiger T, Harper S, and Gu Y, “Improved planetary rover inertial navigation and wheel odometry performance through periodic use of zero-type constraints,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 11 2019, pp. 552–559.
- [10]. Quigley M, Conley K, Gerkey B, Faust J, Foote T, Leibs J, Wheeler R, and Ng AY, “Ros: an open-source robot operating system,” in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.
- [11]. Kilic C and Gross JN, “Pathfinder gps, imu, and wheel odometry data on various terrains,” 2020. [Online]. Available: 10.21227/vz7z-jc84
- [12]. Iagnemma K and Dubowsky S, *Mobile robots in rough terrain: Estimation, motion planning, and control with application to planetary rovers*. Springer Science & Business Media, 2004, vol. 12.
- [13]. Gonzalez R, Apostolopoulos D, and Iagnemma K, “Slippage and immobilization detection for planetary exploration rovers via machine learning and proprioceptive sensing,” *Journal of Field Robotics*, vol. 35, no. 2, pp. 231–247, 2018.
- [14]. Arvidson R, Bellutta P, Calef F, Fraeman A, Garvin JB, Gasnault O, Grant JA, Grotzinger J, Hamilton V, Heverly M et al., “Terrain physical properties derived from orbital data and the first 360 sols of mars science laboratory curiosity rover observations in gale crater,” *Journal of Geophysical Research: Planets*, vol. 119, no. 6, pp. 1322–1344, 2014.
- [15]. Lentaris G, Stamoulias I, Soudris D, and Lourakis M, “Hw/sw codesign and fpga acceleration of visual odometry algorithms for rover navigation on mars,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 8, pp. 1563–1577, 2015.
- [16]. Strader J, Otsu K, and Agha-mohammadi A.-a., “Perception-aware autonomous mast motion planning for planetary exploration rovers,” *Journal of Field Robotics*, vol. 37, no. 5, pp. 812–829, 2020.
- [17]. Angelova A, Matthies L, Helmick D, Sibley G, and Perona P, “Learning to predict slip for ground robots,” in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006 IEEE, 2006*, pp. 3324–3331.
- [18]. Skonieczny K, Shukla DK, Faragalli M, Cole M, and Iagnemma KD, “Data-driven mobility risk prediction for planetary rovers,” *Journal of Field Robotics*, vol. 36, no. 2, pp. 475–491, 2019.

- [19]. Cunningham C, Ono M, Nesnas I, Yen J, and Whittaker WL, "Locally-adaptive slip prediction for planetary rovers using gaussian processes," in 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017, pp. 5487–5494.
- [20]. Brooks CA and Iagnemma K, "Vibration-based terrain classification for planetary exploration rovers," IEEE Transactions on Robotics, vol. 21, no. 6, pp. 1185–1191, 2005.
- [21]. Hidalgo-Carrio J, Hennes D, Schwendner J, and Kirchner F, "Gaussian process estimation of odometry errors for localization and mapping," in 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017, pp. 5696–5701.
- [22]. Rogers-Marcovitz F, Seegmiller N, and Kelly A, "Continuous vehicle slip model identification on changing terrains," in RSS 2012 Workshop on Long-term Operation of Autonomous Robotic Systems in Changing Environments. Citeseer, 2012.
- [23]. Biesiadecki JJ, Baumgartner ET, Bonitz RG, Cooper B, Hartman FR, Leger PC, Maimone MW, Maxwell SA, Trebi-Ollennu A, Tunstel EW et al., "Mars exploration rover surface operations: Driving opportunity at meridiani planum," IEEE robotics & automation magazine, vol. 13, no. 2, pp. 63–71, 2006.
- [24]. Foxlin E, "Pedestrian tracking with shoe-mounted inertial sensors," IEEE Computer graphics and applications, no. 6, pp. 38–46, 2005. [PubMed: 16315476]
- [25]. Norrdine A, Kasmi Z, and Blankenbach J, "Step detection for zup-tailed inertial pedestrian navigation system using foot-mounted permanent magnet," IEEE Sensors Journal, vol. 16, no. 17, pp. 6766–6773, 2016.
- [26]. Xiaofang L, Yuliang M, Ling X, Jiabin C, and Chunlei S, "Applications of zero-velocity detector and kalman filter in zero velocity update for inertial navigation system," in Proceedings of 2014 IEEE Chinese Guidance, Navigation and Control Conference. IEEE, 2014, pp. 1760–1763.
- [27]. Ramanandan A, Chen A, and Farrell JA, "Inertial navigation aiding by stationary updates," IEEE Transactions on Intelligent Transportation Systems, vol. 13, no. 1, pp. 235–248, 2012.
- [28]. Brossard M, Barrau A, and Bonnabel S, "Rins-w: Robust inertial navigation system on wheels," arXiv preprint arXiv:1903.02210, 2019.
- [29]. Dissanayake G, Sukkarieh S, Nebot E, and Durrant-Whyte H, "The aiding of a low-cost strapdown inertial measurement unit using vehicle model constraints for land vehicle applications," IEEE transactions on robotics and automation, vol. 17, no. 5, pp. 731–747, 2001.
- [30]. Skog I, Handel P, Nilsson J-O, and Rantakokko J, "Zero-velocity detection—an algorithm evaluation," IEEE transactions on biomedical engineering, vol. 57, no. 11, pp. 2657–2666, 2010.
- [31]. Williams CK and Rasmussen CE, Gaussian processes for machine learning. MIT press Cambridge, MA, 2006, vol. 2, no. 3.
- [32]. Duvenaud D, "Automatic model construction with gaussian processes," Ph.D. dissertation, University of Cambridge, 2014.
- [33]. Vert J-P, Tsuda K, and Scholkopf B, "A primer on kernel methods," Kernel methods in computational biology, vol. 47, pp. 35–70, 2004.
- [34]. Richardson RR, Osborne MA, and Howey DA, "Gaussian process regression for forecasting battery state of health," Journal of Power Sources, vol. 357, pp. 209–219, 2017.
- [35]. GPy, "GPy: A gaussian process framework in python," <http://github.com/SheffieldML/GPy>, 2012.
- [36]. Julier S, Uhlmann J, and Durrant-Whyte HF, "A new method for the nonlinear transformation of means and covariances in filters and estimators," IEEE Transactions on automatic control, vol. 45, no. 3, pp. 477–482, 2000.
- [37]. ADIS16495 Data Sheet, Analog Devices, 2017, rev. A.
- [38]. OEM615 Receivers Data Sheet, Novatel, 2015, ver.8.
- [39]. Takasu T, "Rtklib: Open source program package for rtk-gps," Proceedings of the FOSS4G, 2009.
- [40]. Misra P and Enge P, "Global positioning system: signals, measurements and performance second edition," Massachusetts: Ganga-Jamuna Press, 2006.
- [41]. Ramme BW and Tharaniyil MP, We energies coal combustion products utilization handbook. We Energies, 2004.

- [42]. Peters GH, Abbey W, Bearman GH, Mungas GS, Smith JA, Anderson RC, Douglas S, and Beegle LW, “Mojave mars simulant—characterization of a new geologic mars analog,” *Icarus*, vol. 197, no. 2, pp. 470–479, 2008.
- [43]. “Intel realsense tracking camera,” Product Documentation, 2020.
- [44]. Campos C, Elvira R, Rodríguez JJG, Montiel JM, and Tardós JD, “Orb-slam3: An accurate open-source library for visual, visual-inertial and multi-map slam,” arXiv preprint arXiv:2007.11898, 2020.

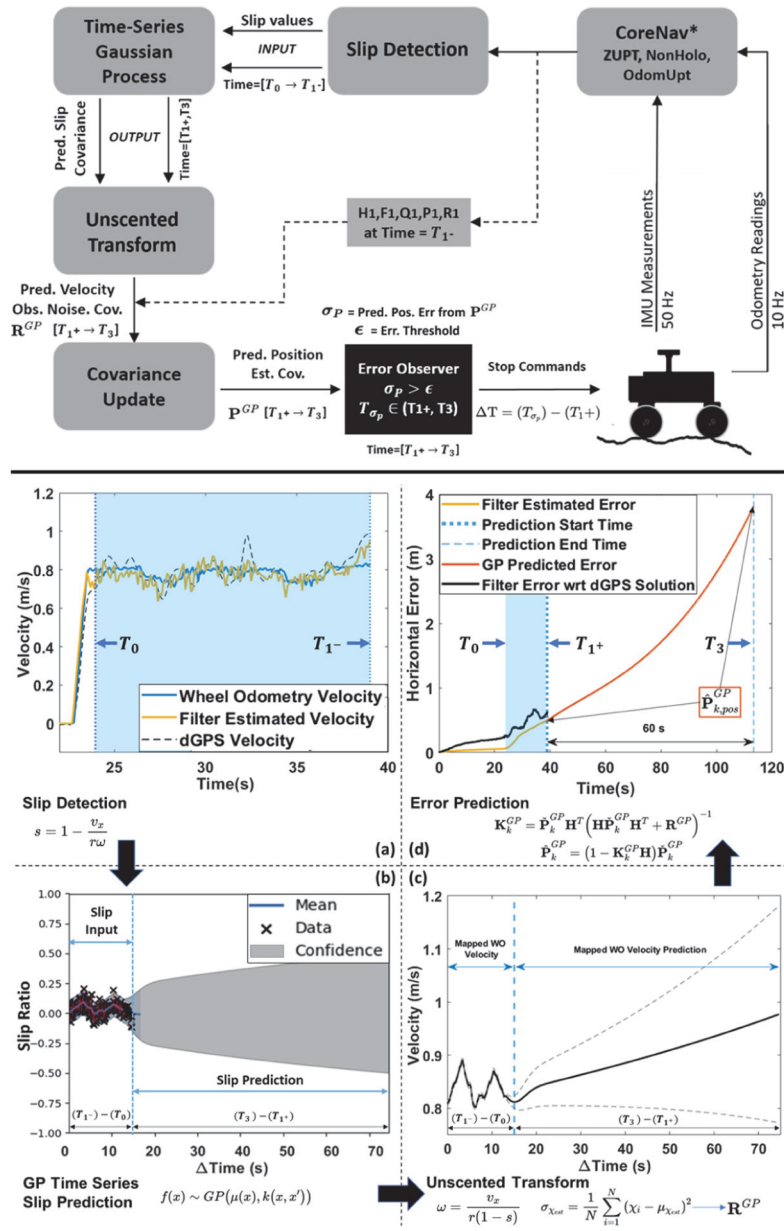


Fig. 1. The proposed framework is demonstrated on the top figure. Each elements of the framework are shown on the bottom sub-figures. The sub-figure (a) shows the filter estimated and WO estimated velocity to be used for slip detection. The sub-figure (b) shows the slip input and slip prediction. The sub-figure (c) is a depiction of unscented transform that used for mapping the WO velocities for error prediction. Finally, the sub-figure (d) shows how the predicted error is generated. The input slip data is collected within a time window (T_1^-) which represented in the blue area in (a). The dotted blue line (T_1^+) in sub-figure (d) represents the time when the future error prediction is generated for 60s. The post processed DGPS outputs are assumed as truth and given for comparison purposes.

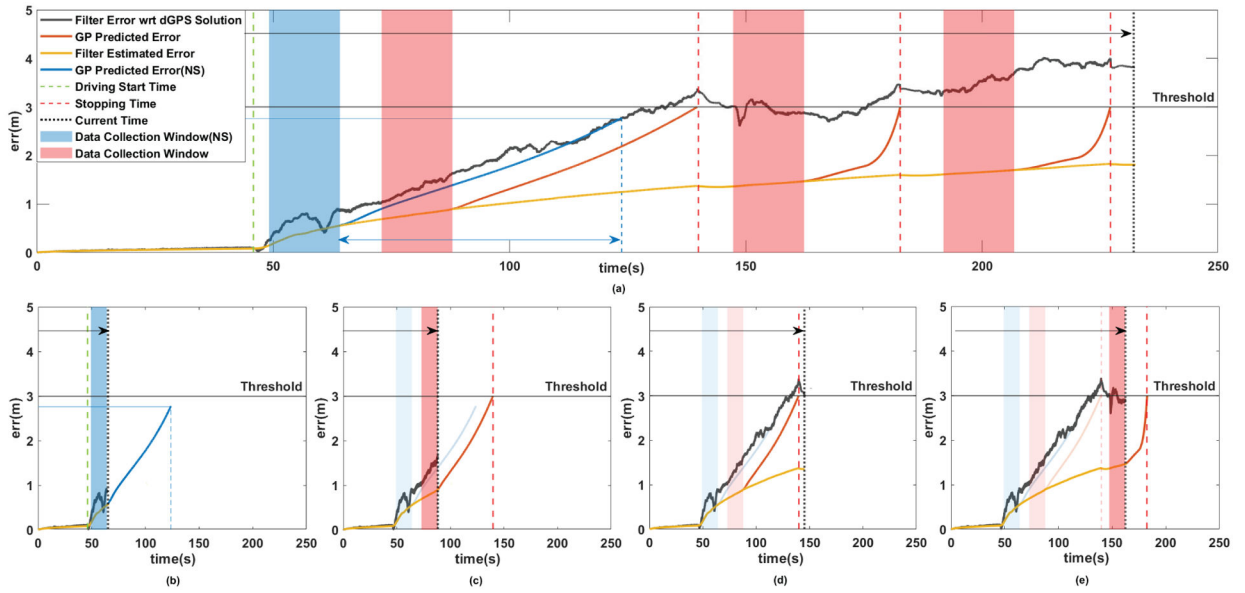


Fig. 2.

A demonstration of the on-board actions and error prediction process of the proposed algorithm. "Filter Estimated Error" is ES-EKF provided estimation and "Filter Error" is the difference between position truth (post-processed differential GPS solution) and the filter position estimation. The testbed rover's average forward speed is 0.8 m/s. The algorithm only considers the slippage collection from WO and filter estimated velocity for 15s intervals. This interval for learning is set based on engineering judgement to capture the most recent (the last 12m of drive) terrain-wheel information based on the MSL Hazard Avoidance slippage check interval (10m) [6]. Threshold is set to 3m and error prediction time limit is set to 60s. The prediction limit is set based on the limitation of blind-driving driving (50m) on MSL operations [6], and reliability of GP prediction over longer times. (a) Overall error prediction and stopping decision for 230s of the operation. (b) Current time = 64s. After collecting 15s of slippage data, the algorithm predicts the horizontal error for 60s. Since the predicted error does not exceed the predetermined threshold, rover continues driving. (c) Current time = 88s. Same process as (b) but this time the error prediction exceeds the threshold before the 60s prediction time limit. Algorithm sets an internal countdown for stopping at the point when the error prediction exceeds the threshold. (d) Current time = 145s. Rover stops, applies ZUPT, and starts driving again. (e) Current time = 162s. Algorithm collects data for 15s, and predicts the stopping time, repeats the process as (c). The GP prediction process took less than a second with IntelCore i7-8650U CPU (Intel NUC Board NUC7i7DN) and is negligible to show in the figure.



Fig. 3.
Pathfinder test platform during field tests in Point Marion, PA.

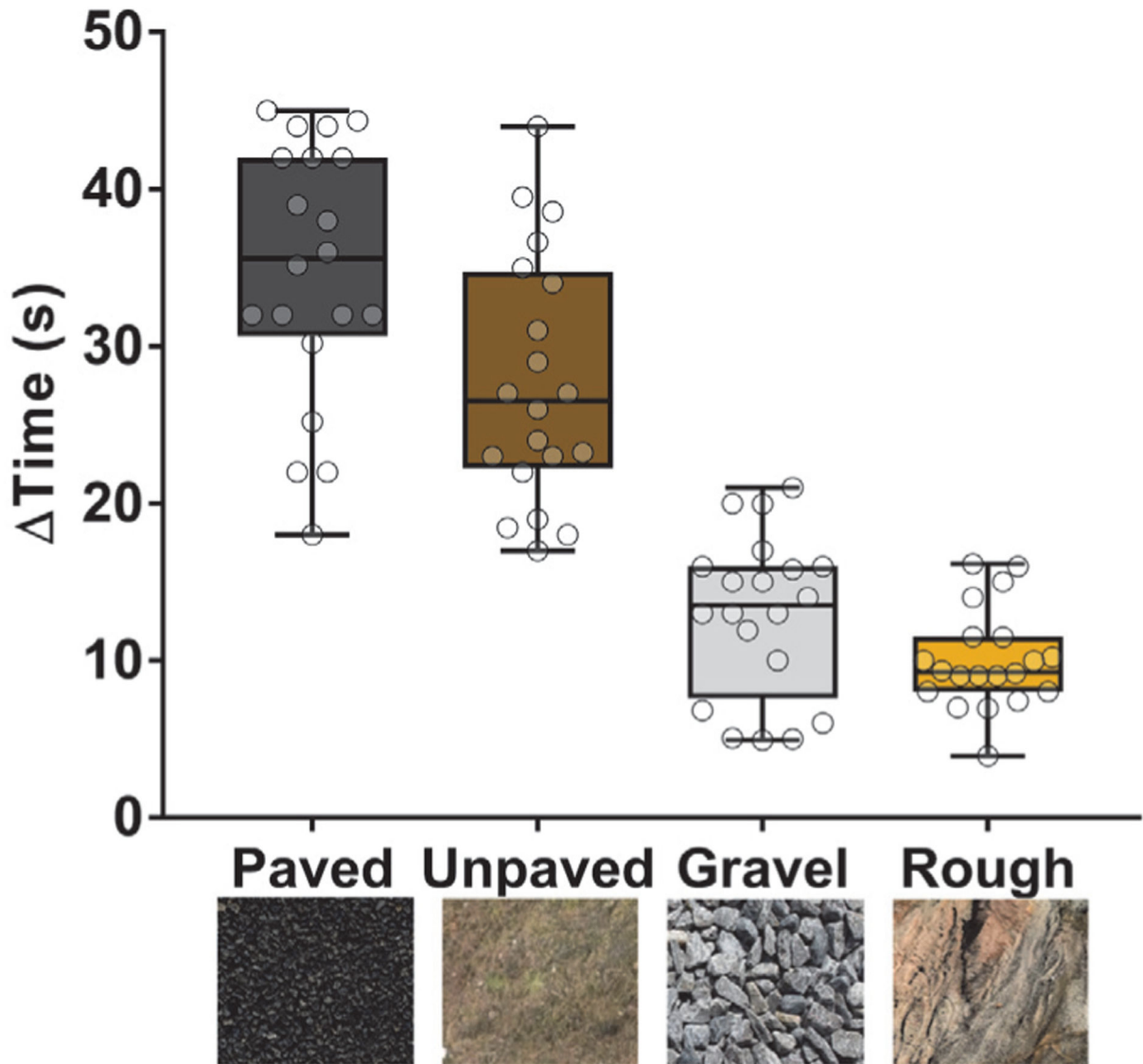


Fig. 4.

Comparison of stop time interval for terrain types. Time axis in box plot shows the remaining duration to stop after 15s of data collection. The middle line in the boxes show the median value of 20 tests for each terrain type. GraphPad Prism software v.7 is used for one-way analysis of variance Tukey's multiple comparison statistical analysis test. Paved: P, Unpaved: U, Gravel: G Rough: R. Non-significant difference: G/R ($p=0.5028$). Significant difference: P/U ($p=0.0049$), R/P, R/U, G/P, and G/U ($p<0.0001$).

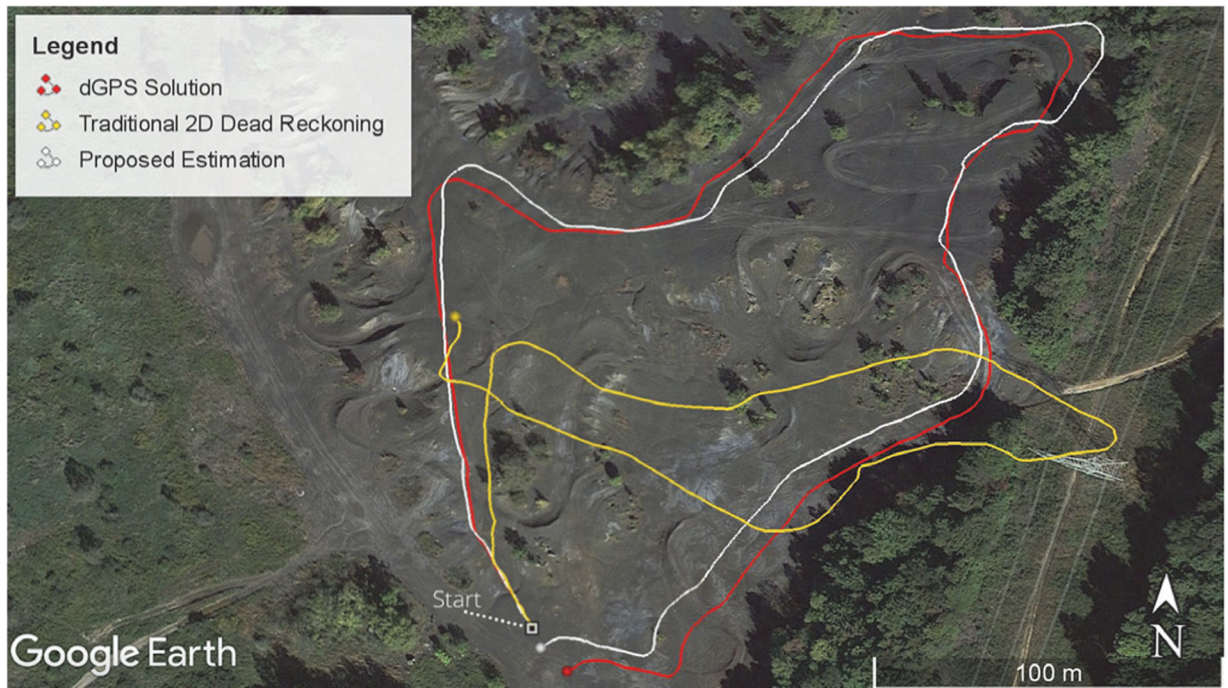


Fig. 5.
Ash-Pile test result for 652m driving with 3m error threshold.

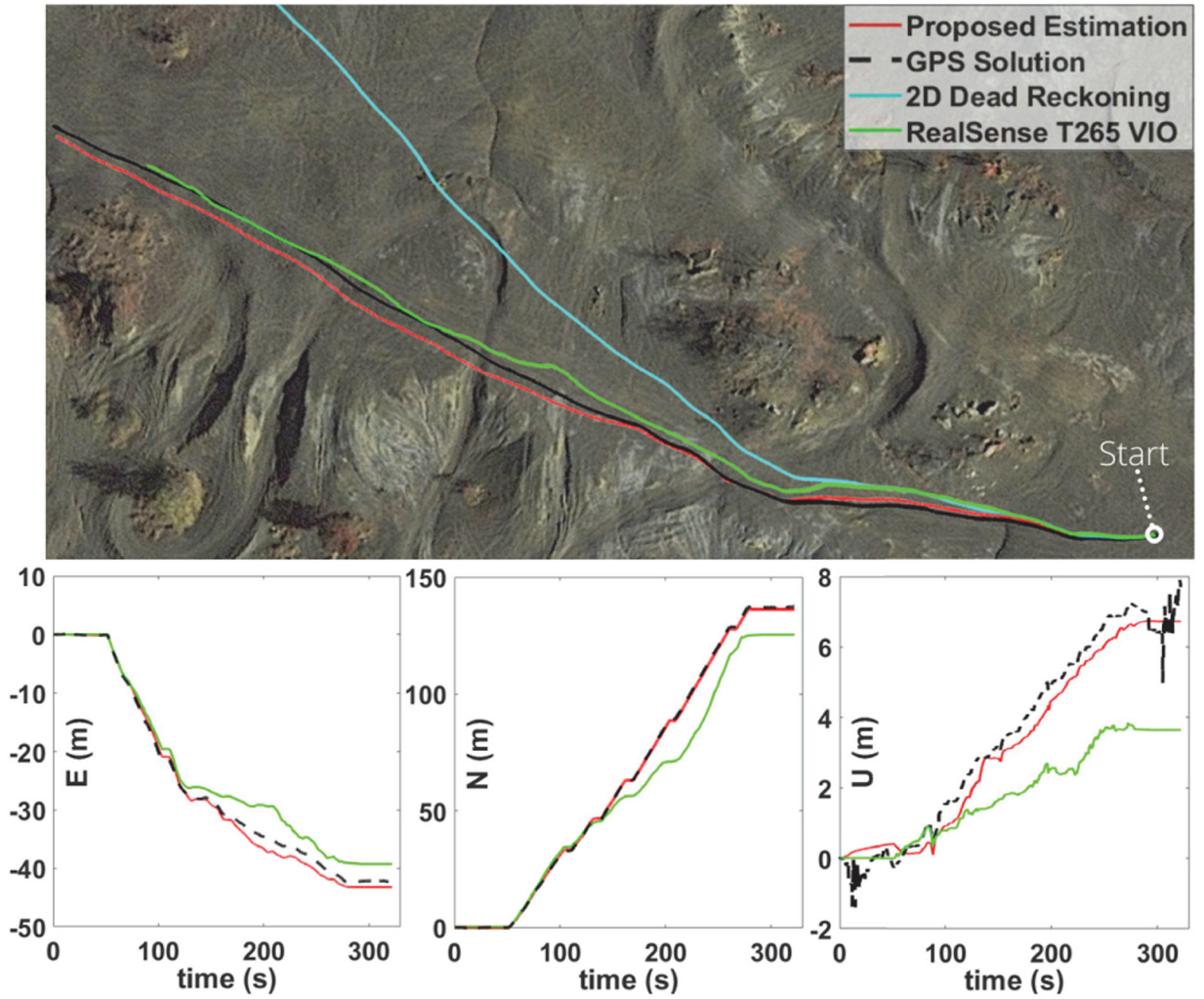


Fig. 6.

Depiction of a comparison for localization accuracy of the proposed approach (3D WIO +ZUPT) in a low-feature rough terrain against RealSense T265 [43] VIO, and 2D dead-reckoning (WIO). Traversed distance 150m. RMSE VIO = East: 2.70m, North: 10.41m, Up: 2.12m. RMSE Proposed = East: 1.03m, North: 0.49m, Up: 0.65m. A detailed analysis with several other examples is available in the VIO Analysis folder at <https://github.com/wvu-navLab/CN-GP>.

TABLE I

Accuracy of the Proposed Approach on Rough Terrain

Ash Pile	Test Specifics*			Error (%)		
	Σ_D (m)	ϵ (m)	Stop Count	Σ_T (s)	ENU	Median
Test1	671	5	8	879	3.07	1.73
Test2	663	3	19	924	1.78	1.04
Test3	652	3	20	915	1.14	1.05
Test4	339	2	9	469	0.91	0.58
Test5	152	2	5	215	0.94	0.82

	Horizontal Error (m)			RMS Error (m)		
	Median	STDD	Max.	East	North	Up
Test1	11.60	12.01	34.48	17.84	7.26	7.25
Test2	6.89	5.68	18.78	9.49	3.08	6.24
Test3	6.84	2.88	11.10	5.67	4.44	1.75
Test4	1.96	1.27	5.86	1.74	1.83	1.78
Test5	1.24	0.80	2.72	1.34	0.47	0.13

* Σ_D : Traversed Distance, ϵ : Error Threshold, Σ_T : Traversal Time.

TABLE II

Periodic versus Autonomous ZUPT Comparison

Periodic	Σ_D (m)	Σ_T (s)	Error(%)	S_R (%)	ΣStop
Rough_A	151	504	0.85	25.02	42
Unpaved_A	87	133	1.53	18.08	8
Unpaved_B	128	181	1.02	11.60	7
Autonomous	Σ_D (m)	Σ_T (s)	Error(%)	S_R (%)	ΣStop
Rough_B	152	215	0.94	7.32	5
Unpaved_C	183	244	1.17	6.15	5
Unpaved_D	161	210	1.56	4.28	3

* Σ_D, Σ_T : Same as Table I, S_R : Stop rate, ΣStop : Stop count