2021

Planetary Rover Inertial Navigation Applications: Pseudo Measurements and Wheel Terrain Interactions

Cagri Kilic
West Virginia University, cakilic@mix.wvu.edu

Follow this and additional works at: <https://researchrepository.wvu.edu/etd>

Part of the [Navigation, Guidance, Control and Dynamics Commons](#), [Robotics Commons](#), and the [Space Vehicles Commons](#)

Recommended Citation

Kilic, Cagri, "Planetary Rover Inertial Navigation Applications: Pseudo Measurements and Wheel Terrain Interactions" (2021). *Graduate Theses, Dissertations, and Problem Reports*. 10345.
<https://researchrepository.wvu.edu/etd/10345>

This Dissertation is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Dissertation in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Dissertation has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.

Planetary Rover Inertial Navigation Applications: Pseudo-Measurements and Wheel-Terrain Interactions

CAGRI KILIC

DISSERTATION SUBMITTED TO THE
BENJAMIN M. STATLER COLLEGE OF ENGINEERING AND MINERAL RESOURCES
AT WEST VIRGINIA UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY IN
AEROSPACE ENGINEERING

JASON N. GROSS, PH.D., CHAIR
YU GU, PH.D.

POWSIRI KLINKHACHORN, PH.D.
GUILHERME A. S. PEREIRA, PH.D.

NATALIA A. SCHMID, PH.D.
KYOHEI OTSU, PH.D.

DEPARTMENT OF MECHANICAL AND AEROSPACE ENGINEERING

MORGANTOWN, WEST VIRGINIA

2021

KEYWORDS: LOCALIZATION, PLANETARY ROVER, PROPRIOCEPTIVE SENSORS

COPYRIGHT © 2021 - CAGRI KILIC

CC BY-NC-ND 4.0

ABSTRACT

Planetary Rover Inertial Navigation Applications: Pseudo-Measurements and Wheel-Terrain Interactions

Cagri Kilic

Accurate localization is a critical component of any robotic system. During planetary missions, these systems are often limited by energy sources and slow spacecraft computers. Using proprioceptive localization (e.g., using an inertial measurement unit and wheel encoders) without external aiding is insufficient for accurate localization. This is mainly due to the integrated and unbounded errors of the inertial navigation solutions and the drifted position information from wheel encoders caused by wheel slippage. For this reason, planetary rovers often utilize exteroceptive (e.g., vision-based) sensors. On the one hand, localization with proprioceptive sensors is straightforward, computationally efficient, and continuous. On the other hand, using exteroceptive sensors for localization slows rover driving speed, reduces rover traversal rate, and these sensors are sensitive to the terrain features. Given the advantages and disadvantages of both methods, this thesis focuses on two objectives. First, improving the proprioceptive localization performance without significant changes to the rover operations. Second, enabling adaptive traversability rate based on the wheel-terrain interactions while keeping the localization reliable.

To achieve the first objective, we utilized the zero-velocity, zero-angular rate updates, and non-holonomicity of a rover to improve rover localization performance even with the limited available sensor usage in a computationally efficient way. Pseudo-measurements generated from proprioceptive sensors when the rover is stationary conditions and the non-holonomic constraints while traversing can be utilized to improve the localization performance without any significant changes to the rover operations. Through this work, it is observed that a substantial improvement in localization performance, without the aid of additional exteroceptive sensor information.

To achieve the second objective, the relationship between the estimation of localization uncertainty and wheel-terrain interactions through slip-ratio was investigated. This relationship was exposed with a Gaussian process with time series implementation by using the slippage estimation while the rover is moving. Then, it is predicted when to change from moving to stationary conditions by mapping the predicted slippage into localization uncertainty prediction. Instead of a periodic stopping framework, the method introduced in this work is a slip-aware localization method that enables the rover to stop more frequently in high-slip terrains whereas stops rover less frequently for low-slip terrains while keeping the proprioceptive localization reliable.

Acknowledgments

This work would not have been possible without the support from friends, family, and colleagues. First, I foremost thank my advisor Dr. Jason Gross for his invaluable support, trust, patience in my research. I am also very grateful to Dr. Yu Gu for his ideas and suggestions throughout my studies. I would like to show my gratitude to my committee members Dr. Guilherme Pereira, Dr. Natalia Schmid, Dr. Powsiri Klinkhachorn, and Dr. Kyohei Otsu for their time to review my research. My research has greatly benefited from the great contributions of many current and previous WVU Robotics researchers, great friends, and colleagues, namely Ryan Watson, Jonas Bredu, Ali Baheri, Shounak Das, Jared Beard, Chris Brindle, Benjamin Buzzo, Thomas Swiger, Jennifer Nguyen, Scott Harper, Kyle Lassak, Derek Ross, Conner Castle, Dylan Covell, Chizhao Yang, Jacob Hikes, Andrew Rhodes, Uthman Olawoye, Maria Gonzales, Eduardo Gutierrez, Kieren Samarakoon, and Rogerio Lima. Special thanks go to Nicholas Ohi for his extraordinary support in this study; Bernardo Martinez, Chris Tatsch, Matteo De Petrillo, and Jared Strader for the fruitful discussions, close collaboration, and their candid friendship. I would like to express my gratitude to Ozcan Ozmen for being a big brother to me; Michal Dawson Connor, Deniz Talan, and Deniz Tuncay for their kind and lasting friendship. I would also like to thank my mother Canan Kilic, and father Veli Kilic, my brother Cagatay Kilic, and my family in law Sibel Sen, Sinan Sen, and Oktay Sen for always believing in me. Last, I express my deepest gratitude to my best friend and wife Emel Sen Kilic for helping me regain hope after despair, resume life after obstructions, making my heart smile, and filling happiness into my life.

Financial Support

This research is supported in part by Fulbright Ph.D. Scholarship, Benjamin M. Statler Fellowship, The Honor Society of Phi Kappa Phi Graduate Research Grant, and NASA EPSCoR Research Cooperative Agreement WV-80NSSC17M0053.

Contents

1	INTRODUCTION	1
1.1	Motivation	2
1.2	Research Objectives	4
1.3	Contributions	5
1.4	Thesis Organization	7
2	LITERATURE REVIEW	9
3	EXPERIMENTAL SETUP	16
3.1	Rovers	17
3.1.1	Husky	18
3.1.2	Pathfinder	18
3.2	Robot Sensors	19
3.3	Truth Reference	20
3.4	Initial Pose Condition Estimation	21
4	FUNDAMENTALS	22
4.1	Wheel Odometry	23
4.2	Linear and Non-Linear Kalman Filters	26
4.2.1	Kalman Filter	26
4.2.2	Extended Kalman Filter	27
4.2.3	Error State Extended Kalman Filter	29
4.3	Gaussian Process	30
5	INERTIAL LOCALIZATION WITH PSEUDO-MEASUREMENT UPDATES	34
5.1	Introduction	35
5.2	Inertial Navigation System Mechanization	36
5.3	Error State Extended Kalman Filter	40
5.4	Pseudo-Measurement Updates	44

5.4.1	Zero Updates	46
5.4.2	Non-Holonomic Constraints	49
5.4.3	Wheel Odometry	50
5.5	Field Tests	53
5.6	Results and Discussion	55
5.6.1	Key Takeaways	61
6	PROPRIOCEPTIVE SLIP DETECTION	63
6.1	Introduction	64
6.2	Proprioceptive Slip Detection	66
6.2.1	Evaluation	69
6.3	Key Takeaways	77
7	SLIP-BASED AUTONOMOUS ZERO VELOCITY UPDATE	78
7.1	Introduction	79
7.2	Kernel Function Selection	81
7.3	Slip Estimation with Time Series Modeling	82
7.4	Wheel Odometry Velocity Prediction	87
7.5	Forecasting Localization Error	88
7.6	Summary of the Operational Process	91
7.7	Field Tests	92
7.8	Results and Discussion	97
7.8.1	Key Takeaways	102
8	CONCLUSION	104
8.0.1	Discussion Summary	105
8.0.2	Contributions	105
8.0.3	Future Work	107
	REFERENCES	109
	APPENDICES	119
A.1	INS Mechanization	120
A.2	Wheel Odometry aided Inertial State Estimation	122
A.2.1	Error State Extended Kalman Filter	122
A.2.2	INS System Matrix	123
A.2.3	INS State Transition Matrix	125
A.2.4	INS Noise Covariance Matrix	126
A.3	Supplementary Figures	128

List of Figures

3.1.1	The research presented in this thesis has been evaluated on the robotic platforms Clearpath Husky UGV and different variations of Pathfinder. Credit: Clearpath .	17
3.2.1	Accelerometer and gyroscope axes assignments on ADIS IMU (both 16495 and 16488 OEM models). Credit: Analog Devices.	20
3.4.1	Demonstration of the initial pose generation.	21
4.1.1	Turning motions of a four wheel skid steering robot.	24
4.3.1	Representation of some common kernel functions.	31
5.1.1	This figure presents a graphical depiction of the implemented navigation system when all the algorithm components are utilized. Within the algorithm, the IMU observations are used to propagate the current state estimate. In conjunction with non-holonomic constraints, wheel odometry observations are leveraged to reduce the unbounded error growth of the unaided inertial navigation system. The zero updates (ZUPT, ZARU) can be triggered by an error observer or manual stops.	37
5.4.1	Toy example for using pseudo-measurements in the inertial positioning estimation. Left figure (a): INS based positioning with a static IMU under Earth gravity vector. Black line shows the INS only positioning estimation error, red line shows the error when using zero updates (labeled as ZU for both ZUPT and ZARU) in the INS estimation, the blue line shows the error when using zero updates and nonholonomic (labeled as NH) constraints in the INS estimation, and dashed black lines indicates the time step when the zero update is enabled for a step size based on the IMU data rate (0.02 s). The 3D positioning error is shown with a logarithmic scale. Right figure (b): The figure is the magnified part of the left figure (gray shaded area) between 0 to 50 seconds. The position error is plotted with a linear scale.	46
5.5.1	Husky rover on traversed terrain types to test the developed navigation approach.	54

5.6.1	North (top) and east (bottom) error position estimates for Rough-Terrain case. Black line is the GPS solution, red line is the position estimate, and the gray region is the 3σ covariance hull. RMSE East is 1.28 m and RMSE North is 0.41 m, total distance is 150 m.	58
5.6.2	3D view of the rover traversal for Rough-Terrain in Latitude, Longitude, Height (LLH) coordinates. The GPS solution and filter estimation are plotted on USGS Digital Elevation Model (DEM) data of the traversed region. The 3D RMS position error at the end is 1.62% of the traveled distance.	58
5.6.3	Ground track view of Pathfinder (Slow-Rectangle) scenario. Red line shows the filter estimated solution, black line shows the GPS solution. Overall 2D position error is 0.92 m with 0.84 m standard deviation.	59
5.6.4	North and East error estimates without using wheel odometry for Fast-Rectangle case. Black line is GPS estimated solution, red line is position estimate, and gray region is 3σ covariance hull. The highest peak values are the maximum worst case distance error. Zero updates recover the position solution when the rover stops. .	60
5.6.5	North error estimate without odometry aiding for Concrete-Turn case. Black line is DGPS solution, red line is north position estimate, and gray region is 3σ covariance hull. Filter estimation exceeds the covariance bounds while turning with wheel slips. However, ZUPT reduces the cubic error growth to linear for positioning and correct most of the position drift when the rover stops.	61
6.2.1	Depiction of a comparison for position estimation accuracy of the proposed approach against Intel RealSense T265 VIO, and traditional dead-reckoning in a visually low-feature terrain. 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.	70
6.2.2	Ground track depictions of the proposed method, VIO estimation, WO with IMU heading estimation, and DGPS solution. The DGPS solution is treated as the truth ground track and given as a black line. VIO estimation is given as a green line, WO-based estimation given as cyan line, and the proposed estimation given as red line with blue circles (the blue circles are the detected significant slip, $s > 0.2 $). The Google Map representation of the truth solution is provided to visualize the environment (Point Marion, PA, Ashpiles Mars Analog Environment). .	72
6.2.3	Left figures show the velocity estimation compared to post processed DGPS velocity solution that assumed as truth velocity. Right histogram figures are the velocity error distributions between the estimated and the truth velocities.	74

6.2.4	The unwrapped heading estimation comparisons. The heading estimation accuracy is critically important to have a reliable translational velocity for the slip detection. Proposed method overall accuracy at the end amount, LongFast: 8 deg, ShortSlow: 2 deg, ShortFast = 0.01 deg. Traditional direct integration estimation at the end amount, LongFast: 74 deg, ShortSlow: 14 deg, ShortFast: 17 deg. . . .	76
7.1.1	The architecture for actively predicting the localization error boundaries. CoreNav* is the the real-time portion of this work that is described in Chapters 5 and in [1]. The real time portion measures the slip ratio and estimates the state in an error state EKF, (CoreNav*), consists of INS, ZUPT, ZARU, WO, and non-holonomic updates. When the rover actively initiated the navigation stops, CoreNav applies the ZUPT and ZARU to reduce localization error growth. The future prediction part uses a time series Gaussian process with the the measured slip input and utilizes the current state estimation and an error observer to predict the future localization error. When the predicted localization error exceeds a pre-determined threshold in a future time, the rover computer generates a stopping command to be activated at that specific time.	80
7.2.1	This figure shows a Brownian, RBF, and a composite kernel for five different random seeds. The composite kernel captures the characteristics of both kernels that can be used to model slip behavior of our test platform.	83
7.2.2	Kernel analysis for different composite kernels. In this analysis, different combination of the Matern 5/2 (MA5), Matern 3/2 (MA3), RBF kernel, linear kernel, constant kernel, and Brownian kernel are shown against the slip data.	84
7.2.3	A composite kernel with multiplying Brownian and RBF kernels together is used to capture both slip behaviors during traversing.	84
7.3.1	An example scenario that starts from slip detection and goes through each elements in the framework. The sub-figure (a) shows the filter estimated and WO estimated velocity to be used for slip detection as described in section 6.2.1. The sub-figure (b) shows the slip input and slip prediction in the GP time series as described in section 7.3. The sub-figure (c) is a depiction of unscented transform that used for mapping the WO velocities for error prediction as described in section 7.4. Finally, the sub-figure (d) shows how the predicted error is generated as described in section 7.5. The input slip data is collected within a time window ($[T_0, T_1^-]$) 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.	86

7.6.1	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 slip check interval (10m) [2]. 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 [3], 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 pre-determined 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.	93
7.7.1	Pathfinder during one of the paved road tests. Post processed PPP Kinematic GPS solution from RTKLib software is shown as red line, the filter estimation is shown as white line. Traversed distance 488 m, horizontal error during driving is less than 0.5%, maximum error 2.1 m. The paved road tests are mainly performed to check the software reliability.	94
7.7.2	Pathfinder during field tests in an environment with similar visual and chemical characteristics found in Martian terrains.	95
7.8.1	Comparison of stop time interval for terrain types. Δ 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 ($p<0.0001$).	98
7.8.2	Ash-Pile Test3 results against traditional 2D Dead Reckoning.	102

A.3.1	Various rate gyro outputs (norm value) during traversal on different terrain types. The difference on the IMU outputs are due to roughness of the terrain which induces different vibration acting on wheels.	128
A.3.2	Supplementary Velocity Error Comparison Figures	129

List of Acronyms

EKF Extended Kalman Filter	27
STM state transition matrix	41
IMU inertial measurement unit	2
VO visual odometry	6
VIO visual inertial odometry	69
SLAM Simultaneous Localization and Mapping	14
ZUPT Zero-Velocity Update	14
ZARU Zero Angular Rate Update	14
TRCTL Traction Control	99
STM State Transition Matrix	41
RMSE Root Mean Square Error	69
GNSS Global Navigation Satellite System	20
GPS Global Positioning System	2
DGPS Differential Global Positioning System	20
GP Gaussian Process	13
GIS Geographic Information System	2
TWD think-while-driving	3
MSL Mars Science Laboratory	3
MER Mars Exploration Rover	10
MSR Mars Sample Return	3
INS inertial navigation system	4

DEM digital elevation map	10
HiRISE High Resolution Imaging Science Experiment	10
FPGA field-programmable gate array	11
WO wheel odometry	10
TWD think while driving	3
ESA European Space Agency	66

Contributing Papers

The content of this thesis lead to several peer-reviewed publications, submissions and technical reports. A portion of the results and discussion presented within this document were originally contained within other articles.

- **Kilic, C.**, Gross J.N., 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*, International Conference on Intelligent Robots and Systems (IROS), IEEE, 2019
- **Kilic, C.**, Agha-mohammadi A., *Improved Inertial Navigation Performance and Terramechanics Estimations for Fast Traverse Mars Rovers*, NASA Jet Propulsion Laboratory, JVSRRP Report, 2019.
- **Kilic, C.**, Tatsch C.A., Rocamora Jr B.M., Beard J.J., Ross D.W., and Gross J.N., *Team Mountaineers Space Robotic Challenge Phase-2 Qualification Round Preparation Report*, arXiv preprint, arXiv:2003.09968, 2020.
- **Kilic, C.**, Ohi N., Gu Y., and Gross J.N., *Slip-Based Autonomous ZUPT through Gaussian Process to Improve Planetary Rover Localization*, Robotics and Automation Letters (RA-L), IEEE, 2021
- **Kilic, C.**, Das S., Gutierrez E., Watson R., and Gross J.N., *ZUPT aided GNSS Factor Graph with Inertial Navigation Integration for Wheeled Robots*, ION GNSS+, 2021
- **Kilic, C.**, Martinez B.R.Jr., Tatsch C., Beard J., Strader J., Das S., Ross D., Gu Y., Pereira G.A.S., and Gross J.N., *NASA Space Robotics Challenge 2 Qualification Round: An Approach to Autonomous Lunar Rover Operations*, Aerospace and Electronic Systems Society Magazine (AESS), IEEE, 2021
- **Kilic, C.**, Gu Y., and Gross J.N., *Proprioceptive Slip Detection for Planetary Rovers in Perceptually Degraded Extraterrestrial Environments*, Field Robotics (FR), (Submitted)

Contributing Software

To generate the results presented within this dissertation, several software packages have been developed. These software packages have been made publicly available in the hope that it will be found beneficial in enabling future research.

- **Software Name:** CoreNav
Software Link: <https://github.com/wvu-navLab/corenav>
Summary: The software used in Chapter 5. This repository provides a localization algorithm that compensates for the high likelihood of odometry errors by providing a reliable localization solution that leverages nonholonomic vehicle constraints as well as state-aware pseudo-measurements (e.g., zero velocity and zero angular rate) updates during periodic stops.
- **Software Name:** CoreNav-GP
Software Link: <https://github.com/wvu-navLab/corenav-GP>
Summary: The software used in Chapter 7. This repository provides a autonomous stop planning algorithm through Gaussian process with time series modeling to improve planetary rover localization.
- **Software Name:** CoreNav-Moon
Software Link: <https://github.com/wvu-navLab/SRC2-localization-ins>
Summary: This software uses the elements from Chapter 4 and 5. This repository provides an algorithm that enables to use the CoreNav package in a virtual Moon environment.

Also, the software generated to collect and process the data are used within other studies. The majority of these packages have been made publicly available and can be found in publications in the *Contributing Papers* section. The raw dataset collected during the work is available in

- **Dataset:** <https://ieee-dataport.org/open-access/pathfinder-gps-imu-and-wheel-odometry-data-various-terrains>

1

Introduction

1.1 Motivation

Current autonomous driving technology with sophisticated sensors, high-powered computers, and a vast amount of geospatial information from Global Positioning System (GPS) and Geographic Information System (GIS) allows for accurate mapping, localization, and navigation of the vehicles on easy-to-drive smoothed or paved surfaces. On the other hand, achieving accurate real-time localization performance is challenging for planetary rovers with limited energy sources and slow spacecraft computers due to radiation-hardened hardware requirements on obstacle-rich, harsh terrain conditions [4, 5, 6, 7].

Perception in robotic systems can be achieved by proprioceptive and exteroceptive sensors. Proprioceptive sensors, such as inertial measurement unit (IMU), perceive information from the state of the robot itself. On the other hand, exteroceptive sensors measure the state of the environment (e.g., camera). There is no single sensor that provides sufficient information for localization typically. For this reason, the information gathered by these sensors is often fused. For example, using proprioception without external aiding is insufficient for reliable localization; therefore, planetary rovers are often utilized exteroceptive (e.g., vision-based) sensors to provide feedback from environmental features [8]. Despite its safety and reliability, vision-based systems operate on the assumption that the terrain contains sufficient visual texture for localization. This assumption poses a challenge on planetary terrains that adequate visual features are lacking in the region (e.g., glacier ice, regolith, salt evaporites, permanently shadowed regions) [9]. Hence, the rover operation usually needs to be altered to detect unique terrain features such as moving mast cameras [10] to the point of interest or using wheel tracks left by the vehicle [11, 12] if the illumination conditions are sufficient. In the case of having perceptually degraded performance due to the unavailability of using a vision-based system, a robotic system can leverage the information from some known

conditions with proprioceptive observers to provide a viable localization estimation. For example, planetary rovers frequently stop for a variety of reasons such as safety checks [13], image processing [11], and conducting scientific experiments [14]. Even with the ideal case of think while driving (TWD) approach [15] that is recently developed to minimize how often the rover needs to stop for Perseverance rover, the stopping is inevitable. Since a rover is in stationary conditions in many instances, leveraging this state information is a natural fit for wheeled planetary robots.

Fast traversing is a challenging problem for planetary rovers due to safety reasons, limited energy resources, and computational power. Specifically, Mars Science Laboratory (MSL) rover reaches a maximum velocity of 140 m/h in blind-drive mode, 45 m/h in hazard avoidance mode, and only 20 m/h in fully autonomous mode [2].

Slow-paced driving was not a critical problem for the Mars exploration rovers by now; however, the next proposed mission to Mars will bring some new constraints with its fetch rover. The Mars fetch rover will become the first ESA/NASA planetary rover with a primary objective of sample retrieval instead of exploration [6], depending on the landing location. The solar powered fetch rover may be required to traverse up to 20 km from the lander to the cached samples left by Perseverance and back to its landing area within 150 sols [16]. This required traversal rate is much faster than Curiosity, which has traveled 19.75 km in 6 years [17] and Opportunity that had traversed 45.16 km in 15 years [18] on Mars before the Planet-Encircling Dust Event. The Mars Sample Return (MSR) rover may need to travel greater distances per sol to the cached sample, which is beyond the capability for the current daily operations planning. The increased rover autonomy and physically faster pace traversing requirement, in turn, require more accurate and computationally efficient onboard rover localization capabilities. The slow pace driving can be alleviated by using dead-reckoning methods (i.e., blind-drive [3] mode is the fastest driving mode for Mars rovers), which make use

of wheel odometry readings and inertial measurements to keep track of the rover's motion if the terrain ahead is considered safe and traversable (e.g., no obstacles) by the rover operation team. However, using a dead-reckoning technique causes increasing uncertainty in the rover state due to wheel slippage and inertial navigation system (INS) drift, resulting in a significant problem for rover localization over time. Therefore, mitigating the disadvantages of blind-driving by using a framework that is more aware of the rover state can provide not only a capability of faster driving and longer distance traversal but also more reliable proprioceptive localization performance with increased onboard autonomy that may reduce the mean time between human interventions during the mission.

1.2 Research Objectives

This research aims to find computationally efficient solutions to the given challenges with the current rover operations and mitigate the concerns for the upcoming planetary mission constraints without affecting any major operations. Given that utilizing the observation from the sensors already onboard for multiple purposes are desirable characteristics for planetary missions [19], we investigated the potentialities of using the available standard proprioceptive sensors (e.g., IMU and wheel encoders) to improve localization performance.

Using proprioceptive sensors such as IMU and wheel encoders are widely used to localize wheeled and tracked vehicles because of their simplicity, computational efficiency, and continuity. However, the localization performance is limited and degraded after a short period due to integrated and unbounded errors of the inertial navigation solutions and the drifted position information from wheel encoders caused by wheel slippage. For this reason, it usually requires computation-

ally expensive methods or human-in-the-loop process to refine rover position estimates [20] [21]. Nevertheless, a planetary rover can still benefit from available sensor measurements to improve its localization performance even with the limited available sensor usage in a computationally efficient way. For example, pseudo-measurements generated from proprioceptive sensors when the rover is in stationary conditions and the motion constraints while traversing can be utilized without any significant changes to the rover operations. Moreover, a rover can be aware of the terrain conditions by leveraging wheel-terrain interactions (e.g., slip-ratio) and adapt its traversal rate by predicting when to utilize measurement updates.

Therefore, the two main research objectives of this study can be summarized with the following items.

- **Research Objective 1:** Providing computationally efficient and accurate real-time rover localization capability without any major changes to the rover operations using only proprioceptive sensors already onboard.
- **Research Objective 2:** Enabling autonomous adaptive traversability rate based on the wheel-terrain interactions while keeping the proprioceptive localization reliable.

1.3 Contributions

This dissertation provides several methods to improve the blind-driving based localization by leveraging the motion constraints while traversing and pseudo-measurements generated from proprioceptive sensors when the rover is stationary. These methods provide reliable localization using only proprioceptive sensors (i.e., IMU and wheel encoders) without any significant changes to the rover operations. Besides, these localization methods could play an essential role in providing

computationally efficient real-time rover localization since the used sensors are available almost all of the time during the mission. While the main area of interest is planetary rovers, the contributions of this thesis can be benefited by any wheeled robotic system where the objective is improving localization accuracy.

The contributions are outlined as follows:

- **Contribution 1 - Method:** It is demonstrated that leveraging pseudo-measurements and motion constraints can significantly reduce the rate of unbounded inertial navigation error growth due to continually integrating acceleration and gyro rate with respect to time. In this respect, providing a reliable proprioceptive localization approach can reduce the usage of computationally expensive positioning corrections.
- **Contribution 2 - Method:** It is shown that wheel slip can be detected as velocity discrepancies between wheel odometry measurements and developed INS solution. Even the wheel slippage can be detected by visual odometry (VO) estimations, in the places where visual perception is unavailable, the proprioceptive slip detection can play a significant role for rover safety and provide a continuous localization estimation.
- **Contribution 3 - Method:** A method is developed for predicting localization error, using a time-series Gaussian process model for slip uncertainty prediction as a function of time, such that stationary conditions can be actively initiated with respect to the wheel slippage frequency and magnitude.
- **Contribution 4 - Test Platform:** A lightweight testbed mobile platform, called Pathfinder, is further improved with numerous software and hardware design iterations to excel in reliably collecting and processing IMU, camera, GNSS, and wheel encoder outputs in various

environments. Thanks to its compact form factor, this testbed rover is easy to carry and can get ready in minutes to conduct field tests. In this respect, developed methods are tested on actual hardware in a planetary-analog environment. Detailed descriptions of our implementation and hardware specifications are provided, so the reader can more easily replicate the work.

- **Contribution 5 - Software:** The developed software packages, which are designed for real-time usage under ROS[22], and the post-processing scripts are made publicly available.
- **Contribution 6 - Dataset:** During verification and validation of the methods, IMU, wheel odometry, GNSS, and camera outputs are collected in more than 120 field experiments, including test cases in a planetary-analog environment. The majority of the collected dataset has been made publicly available to enable future research.

1.4 Thesis Organization

The rest of this proposal is organized in the following manner. In Chapter 2, a literature review for the subject discussed within this dissertation is provided. This review begins with the planetary rover operations for state estimation. Then, it focuses on slip estimation and pseudo measurements. In Chapter 3, the experimental setup with rovers and sensors for the data collection, validation, and verification is detailed. Chapter 4 provides the formulation of the fundamental methods used in this dissertation. Within Chapter 5, a method to improve rover localization with pseudo-measurements from the IMU sensor is provided. This chapter also provides quantitative and qualitative comparisons as well as the performance analysis of the provided method. In Chapter 7, a method of slip-aware localization to increase the traverse rate with keeping the proprioceptive lo-

calization reliable is discussed with experimental field evaluations. Finally, Chapter 8 concludes the dissertation with an overview of the topics covered and provides possible future research directions.

2

Literature Review

The major requirement of any robotic planetary mission is getting the maximum scientific value during the rover working hours. A rover should traverse the harsh terrain autonomously to reach the desired scientific target and move at a maximum safe speed to achieve that goal successfully. These challenges require the improvement of safety features and efficiency for future Mars rovers.

Previous Mars rovers have exhibited significant onboard localization errors and have had to rely primarily on human-in-the-loop operations [4]. Teams of scientists and engineers meticulously programmed task lists at the beginning of each Mars day (sol). The Mars Exploration Rovers (MERs) and MSL rover use a similar approach for localization. When moving, the rover's attitude angles are propagated through the integration of angular-rate measurements [23]. When the rover is stopped, the rover's pitch and roll angles are estimated based on the gravity vector, which is measured using tri-axial accelerometers [23]. Also, the rover improves its attitude by tracking the Sun. When the terrain is safe to traverse, the rover switches to blind driving and goes from A to B. In that phase, wheel odometry (WO), and IMU integration [12] are used to estimate the rover state. The rover does not utilize any advanced autonomy in the blind-drive mode, but it ensures that the mobility system is operating within predetermined safety limits [3]. Since blind-driving is a dead-reckoning technique, it causes the uncertainty of the rover's state to increase with distance. As a consequence, this technique can only be used over short distances in most situations.

Another key capability for safe traversal of rovers in extraterrestrial environment is the knowledge of the terrain geometry. For instance, Curiosity generates a digital elevation map (DEM) of the surrounding terrain using stereo vision which is enhanced by High Resolution Imaging Science Experiment (HiRISE) images [24] similar to MERs [13]. The rover planners closely inspect the generated DEM to command the rover driving steps. The length of blind-driving is limited to the distance chosen by the rover planners (e.g., 25 m - 50 m), which is based on the rover camera visi-

bility range prior to employing this mode.[3]. This is an important action for the rover to properly select obstacle-free paths; however, this terrain geometry knowledge does not guarantee to localize the rover relative to terrain traversed since it causes a substantial cost of traversal rate due to computational expense, and the rover slip is measured infrequently [25].

Wheel slippage is one of the most critical issues to be dealt with for planetary rovers. In general, wheel slippage is observed when the terrain traversed fails [26] or when the rover wheel speeds are different from each other (i.e., kinematic incompatibility [27]. Unexpected variances of terrains arise as non-systematic errors [28], and they can cause significant position errors. Also, driving across loose soil and sloped regions [5] poses a substantial risk for wheel slippage. For example, if the rover traverses on a downward slope, the rover weight fraction in the movement's direction becomes greater, leading to an increased slippage. Vision-based approaches (e.g., VO) are mostly used to estimate the rover slip [5], [29] [30]. Using VO for the slip estimation is considered an accurate and reliable way; however, it is computationally expensive for planetary rovers, especially when the rover is in motion. Using VO considerably slows rover driving speed due to limited computational resources [31]. Also, using VO in a long period decreases the rover traversal rate since the rover needs to stop periodically to acquire images [25]. Even with the addition of the field-programmable gate array (FPGA) processors [32], usage of VO is limited in the visually low-featured terrains and needs proper lighting conditions [10]. Similarly, poor accuracy of state estimate is often observed in the areas when there are inadequate features to detect and track [9]. This limitation can be a decisive factor when the terrain is covered in sand, making the terrain visually indistinguishable.

Martian soil is exceptionally challenging for traversability; even throughout a single drive, Mars rovers traverse on various terrains with different slopes [24]. Although the Martian terrain is rough

and uneven, which contains diverse types of local obstacles [33], the MSL rover drives under the assumption of flat terrain if it does not run the Traction Control (TRCTL) algorithm, which is designed to reduce the rover wheel damage rate [25]. Various studies have modeled slip as a function of terrain geometry. For example, in [34], the correlation between visual terrain information and the estimated slip is investigated using a Mixture of Experts (MoE) framework. However, the visible topsoil of the terrain may provide misleading information since the wheel-terrain interactions are not governed by this topmost layer of the terrain [9]. In particular, highly deformable sulfate-rich sands were buried under the thin cover of basaltic sands, which are not visually perceptible [24] on the Martian surface. During the traversal over these terrains, the rover wheels experience sinkage-related slippage due to increased compaction resistance. A recent study examined the data-driven cubic regression metrics to assess the relationship between the slip estimation and terrain slope information by leveraging proprioceptive sensors in addition to the exteroceptive sensors [35]. It is well established from a variety of studies that slippage is affected by slope; however, wheel slippage can also be seen when a rover traverses on small rocks due to kinematic incompatibility [27]. For example, if one of the rover wheels traverses over a rock, that wheel relatively travels a longer path than the other wheels, and in turn, the other wheels push forward, the wheel traverses over a rock [25].

Rover safety is more important than accuracy for planetary missions [7] since a collision or an unforeseeable slippage-related wheel sinkage can terminate a mission. For this reason, the rovers usually avoid slopes more than 25 degrees on undeformable terrains (bedrock), sandy slopes more than 12.5 degrees, and relatively bright terrains (most likely dominated by soft/fine soils), which may lead to significant wheel sinkage, increased compaction resistance, high slippage, and possible embedding [24]. For example, MERs both became embedded into the soft surface of Mars [36, 37]

due to the significant amount of slip. In May 2009, Spirit became permanently entrapped in soft soil [38]. Moreover, Curiosity faced a significant challenge to avoid sinking because of excessive wheel slippage on a sandy surface in Hidden Valley [39, 40]. To avoid these problems, commanding the rover over non-deformable terrain may not be the solution. Since Mars rovers have lacked the onboard capability to identify non-geometric hazards, Curiosity has received holes, tears, and rips on its rigid aluminum wheels [41].

Employing a terramechanics (wheel-terrain interaction) model to estimate slip requires the knowledge of terrain parameters and variables, which are challenging to measure or estimate accurately online. Substantial studies have been devoted to simplified models due to terramechanics modeling complexity. For example, [26] presented an online estimation tool for the assessment of mechanical soil parameters (e.g., wheel sinkage, cohesion, angle of friction) based on a simplified terramechanics model for deformable terrains (i.e., not applicable to a hard surface such as rocks), and [42] uses a simplified terrain parameter estimation method to establish a link between the driving energy consumption and terramechanics.

Another considerable research attention has been devoted to investigating the use of machine learning algorithms as a slip estimation tool. Locally adaptive slip-model learning with respect to slope values is demonstrated in [43] using a Gaussian Process (GP) regression for visually classified terrain types. Machine learning algorithm comparisons for discrete slip event detection are detailed in [44] and immobilization detection is addressed in [27]. The key part of these researches focuses on the detection of embedding when slippage occurred and informing the rover with in-situ slippage checks. 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 [45]. The methodology in [46]

demonstrated an offline wheel slippage learning approach, where the model is learned on training runs and evaluated in a test environment using Simultaneous Localization and Mapping (SLAM) in a planetary rover navigating an unstructured environment.

On the other hand, [47] 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. Similarly, in [48], the researchers described an online learning method to calibrate the vehicle’s mobility model using the sensor inputs from the IMU, wheel encoders, and GPS.

Continuous perception is the critical capability for resilient localization. Since no single observation can provide this, as previously mentioned, the vehicle can encounter critical navigation failures in perceptually degraded situations. However, proprioceptive sensing can be leveraged in addition to geometric and semantic information. A robotic system can utilize the information from some known conditions with proprioceptive sensing capabilities to provide a viable localization estimation in permanently shadowed regions, extremely bright areas, uniform, and visually indistinguishable terrains. Besides, having a more reliable onboard proprioceptive localization approach may help to reduce the frequency of using computationally expensive visual-based corrections. For instance, rover localization performance can be improved by using the pseudo-measurements during stationary conditions (e.g., Zero-Velocity Update (ZUPT), Zero Angular Rate Update (ZARU)) and leveraging rover kinematics constraints (e.g., non-holonomic constraints). In general, rovers are in stationary conditions in many instances (e.g., capturing images, obstacle avoidance, re-planning, the processing time for visual odometry, and conducting scientific experiments). For instance, the Curiosity rover needs to stop to acquire an image, process

the image, and then the rover runs the VO to gather slip measurements [25]. ZUPT is a well-known concept, which was initially popularized as a method to aid inertial pedestrian navigation [49, 50]. Zero velocity detection and application in standard road conditions for automobile applications is shown in [51]. Moreover, the study in [52] demonstrates the capability of using zero-updates during stopping at traffic lights. Further, a recurrent neural network method to detect the zero velocity conditions on a publicly available car dataset is proposed in [53]. Apart from using stationary conditions to improve localization for autonomous cars, using ZUPT can be considered as a well-suited application to deal with drifts, given that planetary rovers stop more regularly than cars. Leveraging pseudo-measurements can improve the rover localization performance by a significant margin without any hardware changes or significant changes in rover operations [1, 54].

3

Experimental Setup

Figure 3.1.1: The research presented in this thesis has been evaluated on the robotic platforms Clearpath Husky UGV and different variations of Pathfinder. Credit: Clearpath

3.1 Rovers

This section introduces two rovers used as the testbed platforms for the implementation, data collection, verification, and evaluation of the findings for this research. In the earlier research stage, an out-of-the-box robotic platform, Clearpath Husky UGV, is used. Using Husky minimizes the hardware-related difficulties, such as wheel encoder calibration, motor controller tuning, and interfacing with the hardware. Also, the platform vendor provides an open-source repository with common software implementations, which allows for rapid software prototyping. After having reliable and repetitive experimental results with Husky, the testing platform is changed to WVU Pathfinder. This testbed platform has several variations and significantly evolved along with the study. Having two different rovers also allows us to demonstrate the general applicability of the developed algorithms. Those two testbed platforms are detailed in the following sections.

3.1.1 Husky

Husky has four wheels, both left and right wheels are connected by a single drive-train, and uses a differential steering system similar to tracked vehicles which they take two inputs, left and right wheel velocity commands. Therefore, the rover itself can be classified as a skid-steer vehicle since the rotation relies on differing the left and right wheel velocities. The rover can turn-in-place by rotating the left and right wheels in opposite directions at the same speed or generate arc trajectories with a combination of the rotational and translational motion by varying the speeds and directions. The motor drivers use these rotational and translational commanded motions from the user, and a built-in differential drive controller calculates the right and left wheel velocities to drive the rover. Husky has 33 *cm* diameter lug tread all-terrain wheels. Husky track-width (distance between the left and right wheels) is 0.555 *m*, and wheelbase (distance between front and back wheels) is 0.512 *m*. The rover is 50 *kg*, and it has a maximum achievable speed of 1.0 *m/s*.

3.1.2 Pathfinder

Pathfinder is a lightweight, four-wheeled, custom-made, quick testing and software prototyping platform. The rover drive-train is a differential-bogie (as opposed to a rocker-bogie) system, where the two wheels on one side are attached to the same bogie, and the robot body and other bogie are constrained by a rotational differential bar. Left and right wheel pairs are controlled by motor controllers and mounted on each side of the rover. Pathfinder is a skid-steer rover, and it has 24 *cm* diameter polyurethane deformable wheels. Pathfinder track-width is 0.685 *m*, and wheelbase is 0.544 *m*. The rover is 15 *kg*, and its maximum speed is 0.8 *m/s*.

3.2 Robot Sensors

In order to replicate the developed methods in this thesis work more easily, a detailed description of the sensor specifications used is provided in this section. Different proprioceptive and exteroceptive sensor modalities such as IMU, wheel encoders, and tracking cameras, as well as GNSS receivers and antennas, are leveraged to evaluate the results with reference solutions and for comparisons. To measure the acceleration and rate gyro signals, Analog Digital 16488 (ADIS 16488-OEM) and 16495 (ADIS 16495-2BMLZ) 6-degree of freedom (DOF) IMU models, both include a tri-axis gyroscope and a tri-axis accelerometers, are utilized. ADIS 16488-OEM IMU reports having an in-run bias. Angular random walk values are $6.25^\circ/hr$, $0.3^\circ/\sqrt{hr}$ for the gyroscopes, and a bias-stability and velocity random walk of $0.1\ mg$ and $0.029\ m/sec/\sqrt{hr}$ for the accelerometers, respectively. ADIS 16488-OEM IMU is used in the early stages of the research and mainly in the field tests with the Husky rover. ADIS 16495-2BMLZ IMU in-run bias and angular random walk values are $1.6^\circ/hr$, $0.1^\circ/\sqrt{hr}$ for the gyroscope, and $3.2\ \mu g$, $0.008\ m/sec/\sqrt{hr}$ for the accelerometer, respectively. The specifications of these sensors are given in Table 3.2.1 and the axes assignments for both IMU are given in Fig. 3.2.1. Wheel odometry estimations are used as a standalone dead-reckoning solution for comparisons and as an aiding measurement update to the inertial localization system. The WO inputs are generated by quadrature Hall effect encoder feedback with $78,000\ pulses/m$ for Husky and $47,000\ pulses/m$ for Pathfinder. Global positioning information is collected with dual-frequency Novatel OEM-615 GNSS receiver and L1/L2 Pinwheel antenna that mounted to the rovers. The GNSS solution is used for the initialization of the inertial localization framework. The post-processed high precision solutions (e.g., DGPS, PPP) are leveraged for comparisons and truth generation for this work. Additionally, a commercially-of-the-shelf system, RealSense T265 tracking camera is used for localization performance comparisons. This

system includes two fisheye lens sensors, an IMU, and an Intel visual processing unit (VPU).

Table 3.2.1: Inertial Measurement Unit Specifications

IMU	Gyroscope			Accelerometer			Physical Specs	
	Range (deg/sec)	In-run bias (deg/hr)	ARW* (deg/ $\sqrt{hr}$)	Range (g)	In-run bias (mg)	VRW* (m/s/ $\sqrt{hr}$)	Dimensions (mm)	Weight (g)
ADIS-16488OEM	± 450	6	0.30	± 18	0.1	0.029	47 x 44 x 14	48
ADIS-16495-2	± 480	1.6	0.1	± 8	0.0032	0.008	47 x 44 x 14	42

* ARW: Angular Random Walk, VRW: Velocity Random Walk

Figure 3.2.1: Accelerometer and gyroscope axes assignments on ADIS IMU (both 16495 and 16488 OEM models). Credit: Analog Devices.

3.3 Truth Reference

Integer-ambiguity-fixed carrier-phase Differential Global Positioning System (DGPS) is used to determine a truth reference solution. The setup for the DGPS solution consisted of two dual-frequency Global Navigation Satellite System (GNSS) receivers and dual-frequency antennas, with one set mounted on a static base station and another affixed on top of the test rover platforms. During the experiments, 10 Hz carrier-phase and GPS pseudorange measurements were logged on both receivers. GNSS measurements are collected only externally for generating the truth through

post-processing. The open-source software library, RTKLIB 2.4.2 [55], is used to post-process the DGPS solutions with a cm-to-dm expected level accuracy [56].

3.4 Initial Pose Condition Estimation

The accuracy of a dead-reckoning based localization is highly dependent on the initial pose accuracy. Rover position and heading are initialized with a loosely coupled GNSS-INS sensor fusion algorithm based on the formulation in [57]. The initial pose condition estimation is performed by driving straight with a short distance ($\sim 10\text{ m}$) for estimating the absolute initial heading and being stationary for a period of time ($\sim 30\text{ s}$) to initialize position before testing. The rovers are assumed as they start their driving on a flat region (i.e., initial roll and pitch values are equal to zero degrees). A depiction of this process is given in Fig. 3.4.1.

Figure 3.4.1: Demonstration of the initial pose generation.

4

Fundamentals

This chapter incorporates material from the following publication:

Kilic, C., Gross, J.N., Ohi, N., Watson, R., Strader, J., Swiger, T., Harper, S. and Gu, Y., 2019, November. “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)* (pp. 552-559). IEEE.

4.1 Wheel Odometry

Odometry uses motion sensor data to estimate the position change over time. Wheel odometry-based state estimation can be performed by using the wheel encoders as motion sensors, which is broadly utilized for localizing wheeled and tracked vehicles because of its simplicity and continuity. The introduced errors from the wheel odometry are integrated and unbounded. These odometry errors can be classified as systematic and non-systematic errors [58]. Systematic errors are related to wheel geometry uncertainties and encoder quality. Non-systematic errors are related to wheel-terrain interactions, representing the force acting on the robot and the deformation of the terrain surface. Wheel odometry estimation is often fused with the estimations from other sensor modalities [59] in order to correct the drifted estimation from the robot’s true motion for maintaining the robot pose reliable.

For errors due to non-systematic reasons, the placement of odometry sensors is one of the issues in any car-type locomotion system (i.e., Ackermann configuration) where more than one wheel is providing thrust. As in a car-like driving configuration, the rear wheels will slip when turning because the wheel on the outside travels farther than the wheel on the inside (if the rear wheels are driven by a solid axle and a differential is not used). This slippage is more prominent in skid steer driving configuration, where the wheels at the turning direction travel a lesser distance than the wheels on the outside. Turning motion for arc trajectory generation and turn-in-place maneuver

examples of a four-wheel skid steering robot are demonstrated in Fig. 4.1.1.

Figure 4.1.1: Turning motions of a four wheel skid steering robot.

The state of a skid steer vehicle is defined with the planar position and heading angle. The motion of the rover based on wheel odometry can be described with the following state equations:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{R}{2} (V_R + V_L) \cos \psi \\ \frac{R}{2} (V_R + V_L) \sin \psi \\ \frac{R}{L} (V_R - V_L) \end{bmatrix} \quad (4.1)$$

where L is wheelbase, how far the left and right wheels are separated to each other, R is the radius of

the wheel, V_R is right wheel velocity, V_L is left wheel velocity, and ψ is heading. Skid steer vehicles rely on differing left and right wheel velocity directions to turn the vehicle. The state information can be obtained by the fact that wheel encoders give the distance moved by each wheel. Assume the wheels are following an arc;

$$\begin{bmatrix} x_k \\ y_k \\ \psi_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \psi_{k-1} \end{bmatrix} + \begin{bmatrix} D_c \cos(\psi) \\ D_c \sin(\psi) \\ \frac{D_r - D_\ell}{R} \end{bmatrix} \quad (4.2)$$

where

$$D_c = \frac{D_\ell + D_r}{2} \quad (4.3)$$

The distance moved by individual wheels, D , can be obtained by assuming each wheel has N ticks per revolution. Most wheel encoders give the total encoder tick count since the beginning. Therefore, differentiating the encoder tick counts for a time interval will give the delta encoder tick count. This tick count can be used with the known (or estimated) geometric properties of the corresponding wheel. For a rigid wheel, assuming the circular shape of the wheel does not change (i.e., constant wheel radius) during traversal on a rigid terrain, this can be given as:

$$\begin{aligned} \Delta \text{tick} &= \text{tick}' - \text{tick} \\ D &= 2\pi R \frac{\Delta \text{tick}}{N} \end{aligned} \quad (4.4)$$

4.2 Linear and Non-Linear Kalman Filters

4.2.1 Kalman Filter

The Kalman filter is a recursive filter that takes a probabilistic state estimate and updates it in real time using prediction and corrections steps [60]. The process starts from an initial estimate of the state and a motion model is used to predict the new state. Then, an observation model is used to correct the prediction of the state. For a linear Kalman filtering application, a motion model at time k can be given as

$$\mathbf{x}_k = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1}, \quad \mathbf{w}_k \sim N(\mathbf{0}, \mathbf{Q}_k) \quad (4.5)$$

where $\mathbf{x}$ is the state, $\mathbf{F}$ is the state transition model, $\mathbf{G}$ is the control input model, $\mathbf{u}$ is input, $\mathbf{w}$ is the process noise. The process noise is assumed to be a zero mean, white Gaussian noise with covariance $\mathbf{Q}$. The measurement model at time k is

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k, \quad \mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R}_k) \quad (4.6)$$

where $\mathbf{H}$ is measurement model, and $\mathbf{v}_k$ is measurement noise. The measurement noise is assumed to be a zero mean, white Gaussian noise with covariance $\mathbf{R}$.

Given that the estimate at time step k is a linear combination of the estimate at time step $k-1$, prediction and correction steps can be summarized as

Prediction:

$$\check{\mathbf{x}}_k = \mathbf{F}_{k-1} \mathbf{x}_{k-1} + \mathbf{G}_{k-1} \mathbf{u}_{k-1} \quad (4.7)$$

$$\check{\mathbf{P}}_k = \mathbf{F}_{k-1} \hat{\mathbf{P}}_{k-1} \mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1} \quad (4.8)$$

where $\check{\mathbf{x}}_k$ is a *a priori* state estimate and $\check{\mathbf{P}}_k$ is a *a priori* estimate covariance, given motion model at time k.

Optimal Gain:

$$\mathbf{K}_k = \check{\mathbf{P}}_k \mathbf{H}_k^T (\mathbf{H}_k \check{\mathbf{P}}_k \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (4.9)$$

where $\mathbf{K}_k$ is the Kalman gain **Correction:**

$$\hat{\mathbf{x}}_k = \check{\mathbf{x}}_k + \mathbf{K}_k (\mathbf{y}_k - \mathbf{H}_k \check{\mathbf{x}}_k) \quad (4.10)$$

$$\hat{\mathbf{P}}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \check{\mathbf{P}}_k \quad (4.11)$$

where $\hat{\mathbf{x}}_k$ is a *posteriori* state estimate and $\hat{\mathbf{P}}_k$ is a *posteriori* estimate covariance – given measurement at time k. The hat indicates a corrected prediction at a particular time step. Whereas a check indicates a prediction before the measurement is incorporated.

4.2.2 Extended Kalman Filter

Kalman filter is a linear estimator and cannot be used directly to estimate states that are non-linear functions of either the measurements or the control inputs. The Extended Kalman Filter (EKF) is the idea of linearization a non-linear system [61]. Differentiable motion and measurement models

for EKF can be given as

$$\mathbf{x}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}), \quad \mathbf{w}_k \sim N(\mathbf{0}, \mathbf{Q}_k) \quad (4.12)$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{v}_k), \quad \mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R}_k). \quad (4.13)$$

where $\mathbf{x}_k$ is current state, $\mathbf{x}_{k-1}$ is previous state, $\mathbf{u}$ is input, and $\mathbf{y}_k$ is measurement similar to the linear Kalman filter equations. The process noise $\mathbf{w}_{k-1}$ and the measurement noise $\mathbf{v}_k$ are assumed to be zero mean, Gaussian noise with covariance $\mathbf{Q}$ and $\mathbf{R}$, respectively. The process noise covariance matrix, $\mathbf{Q}$, represents the confidence placed in the state estimates where the diagonal terms contain the variance of the states, and off-diagonal terms comprise the covariances between the different state variables.

The motion model can be linearized about the posterior estimate of the previous state, and the measurement model can be linearized about the prediction of the current state based on the motion model. This linearization can be done by using the Taylor series expansion by a tangent line at an operating point. The operating point can be selected as the most recent state estimate.

$$\mathbf{f}(\mathbf{x}) \approx \mathbf{f}(\mathbf{a}) + \left. \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{a}} (\mathbf{x} - \mathbf{a}) \quad (4.14)$$

Correspondingly, motion model and measurement model Jacobians can be given as

$$\mathbf{F}_{k-1} = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}_{k-1}} \right|_{\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}, 0} \quad (4.15)$$

$$\mathbf{H}_k = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}_k} \right|_{\check{\mathbf{x}}_k, 0}. \quad (4.16)$$

4.2.3 Error State Extended Kalman Filter

The idea of the error state extended Kalman filter can be explained as the total state, $\mathbf{x}$, is composed of two parts; nominal state, $\hat{\mathbf{x}}$, and error state, $\delta\mathbf{x}$ [62].

$$\mathbf{x} = \hat{\mathbf{x}} + \delta\mathbf{x} \quad (4.17)$$

In the error state EKF, the error state is estimated instead of the full state. Then the estimate of this error state is used as a correction to the nominal state. Starting with predicting the nominal state using Equation 4.12 and the current estimate of the state as well as keep tracking the state covariance, the prediction part can be given as;

$$\check{\mathbf{x}}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{0}) \quad (4.18)$$

$$\check{\mathbf{P}}_k = \mathbf{F}_{k-1} \mathbf{P}_{k-1} \mathbf{F}_{k-1}^T + \mathbf{L}_{k-1} \mathbf{Q}_{k-1} \mathbf{L}_{k-1}^T \quad (4.19)$$

where $\mathbf{L}_k$ is the process noise related Jacobian, and it is identity in the case where noise is assumed to be additive. Kalman gain is used as same as Equation 5.11, and the estimate of the error state can be given as;

$$\delta\hat{\mathbf{x}}_k = \mathbf{K}_k (\mathbf{y}_k - \mathbf{h}_k(\check{\mathbf{x}}_k, \mathbf{0})) \quad (4.20)$$

After having the error state estimation, it can be used to correct the full state:

$$\hat{\mathbf{x}}_k = \check{\mathbf{x}}_k + \delta\hat{\mathbf{x}}_k \quad (4.21)$$

$$\hat{\mathbf{P}}_k = (1 - \mathbf{K}_k \mathbf{H}_k) \check{\mathbf{P}}_k. \quad (4.22)$$

4.3 Gaussian Process

GP deals with the prediction problem of given some noisy observations of a dependent variable at certain values of the independent variable, determining the best estimate of the dependent variable at a new value [63]. A GP is a Bayesian nonparametric model [64] which uniquely defined by its mean function $\mu(x)$ and covariance function $k(x, x')$ [65]

$$f(x) \sim GP(\mu(x), k(x, x')). \quad (4.23)$$

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

$$p(f(x_1), \dots, f(x_n) | x_1, \dots, x_n) = \mathcal{N}(\boldsymbol{\mu}, \mathbf{K}) \quad (4.24)$$

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 aim of a regression problem is to learn the mapping from inputs to outputs [66], given a training set of input and output pairs $(\mathbf{x}, \mathbf{y}) = (x_i, y_i)_{i=1}^N$, 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:

$$p(\mathbf{y}_* | \mathbf{x}_*, \mathbf{x}, \mathbf{y}) = \mathcal{N}(\mathbf{y}_* | \boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*) \quad (4.25)$$

where

$$\boldsymbol{\mu}_* = \mathbf{K}_*^T \mathbf{K}_*^{-1} \mathbf{y}, \quad \mathbf{K}_* = \mathbf{K}(\mathbf{x}, \mathbf{x}_*) \quad (4.26)$$

$$\boldsymbol{\Sigma}_* = \mathbf{K}_{**} - \mathbf{K}_*^T \mathbf{K}_*^{-1} \mathbf{K}_*, \quad \mathbf{K}_{**} = \mathbf{K}(\mathbf{x}_*, \mathbf{x}_*). \quad (4.27)$$

A kernel describes the covariance of the GP variables; it is also called a covariance function. Kernel encodes the similarity between the outputs in GP [67]. To be a valid kernel, it must be positive semidefinite [65]. However, it is often difficult to define a new valid kernel [68]. For this reason, a common approach is combining predefined kernels to model a new kernel. Some of the common kernels are produced with GPy software [69] and given in Fig. 4.3.1.

Figure 4.3.1: Representation of some common kernel functions.

The mathematical definitions of the kernels in Fig 4.3.1 can be found through Equations 4.28-4.33 for scalar value inputs.

Radial Basis Function: (Also known as Squared Exponential)

$$k_{RBF}(x, x') = \sigma^2 \exp \left(-\frac{\|x - x'\|^2}{2\ell^2} \right) \quad (4.28)$$

Matern 5/2:

$$k_{M5/2}(x, x') = \sigma^2 \left(1 + \frac{\sqrt{5}|x - x'|}{\ell} + \frac{5\|x - x'\|^2}{3\ell^2} \right) \exp \left(-\frac{\sqrt{5}|x - x'|}{\ell} \right) \quad (4.29)$$

Matern 3/2:

$$k_{M3/2}(x, x') = \sigma^2 \left(1 + \frac{\sqrt{3}|x - x'|}{\ell} \right) \exp \left(-\frac{\sqrt{3}|x - x'|}{\ell} \right) \quad (4.30)$$

Exponential:

$$k_{exp}(x, x') = \sigma^2 \exp \left(-\frac{|x - x'|}{\ell} \right) \quad (4.31)$$

Linear:

$$k_{lin}(x, x') = \sigma^2 x x' \quad (4.32)$$

Brownian:

$$k_B(x, x') = \sigma^2 \min(x, x') \quad (4.33)$$

where σ^2 is variance and ℓ is called length-scale.

Using a kernel combination from the existing kernels is a way of making a new kernel. This is specifically beneficial if the data has more than one type of feature. The most common combination

approaches are multiplication and addition of the kernels such that

The product of two kernels:

$$k_m(x, x') = k_1(x, x')k_2(x, x') \quad (4.34)$$

The sum of two kernels:

$$k_a(x, x') = k_1(x, x') + k_2(x, x') \quad (4.35)$$

The decision of choosing which kernel to use in the GP is a design choice that needs to be incorporated with the modeling data. This decision is highly dependent on prior knowledge about the data such that the kernel is expected to reflect the particular characteristics of the data [68]. The process of kernel expressions selection is often needed a systematic or heuristic search; however, there are methods for automating the choice of kernel. For example, the researchers in [67] proposed a kernel search procedure by using addition and multiplication combinations among several base kernels to automatize the process.

5

Inertial Localization with Pseudo-Measurement Updates

This chapter incorporates material from the following publications:

Kilic, C., Gross, J.N., Ohi, N., Watson, R., Strader, J., Swiger, T., Harper, S. and Gu, Y., 2019, November. “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)* (pp. 552-559). IEEE.

Kilic, C., Das S., Gutierrez E., Watson R., and Gross J.N., *ZUPT aided GNSS Factor Graph with Inertial Navigation Integration for Wheeled Robots*, ION GNSS+, 2021

5.1 Introduction

Without external aiding, inertial sensor-based navigation solutions inherently exhibit accumulated error. A constant accelerometer bias causes a position error that grows quadratically in time, and a constant gyro rate bias results in a cubic error growth in position [59]. Furthermore, for planetary rovers that are traversing diverse types of terrain, wheel odometry is often unreliable for use in long-term localization, mostly due to wheel slippage. Due to these problems, several error correction approaches are used to maintain the rover pose; however, these strategies often require additional computational power, energy resources, adequate features in the environment, and slow down the rover traverse speed as discussed previously in Chapters 1 and 2.

This chapter provides a proprioceptive approach that compensates for the high likelihood of inertial sensor-based navigational errors by leveraging non-holonomic vehicle constraints during motion and zero updates (i.e., ZUPT, ZARU) during periodic stops. By using this, the unbounded error growth due to continually integrating acceleration and gyro rate with respect to time could be reduced, and computationally expensive positioning corrections could be performed less often. In this localization framework, an error-state EKF is implemented based on the method detailed in [57]. In this formulation, a planet-centered, planet-fixed frame is used as the reference frame, while

a locally-level navigation frame (NED) comprises the resolving axes. In the filter, wheel odometry-based wheel velocities and heading rate calculations are treated as an aiding sensor for the INS. In addition, zero updates and non-holonomic motion constraints are utilized as additional pseudo-measurement updates whenever they are available. Slip is estimated using a threshold on the difference between the estimated INS solution and wheel odometry velocity vectors. The following sections detail the formulation and methods used in this localization framework. The benefits of this approach can be leveraged for planetary rovers with limited computational resources, without a need for a dedicated additional computer, and in the perceptually degraded regions especially when the visual-based sensors are unavailable. The approach can also be used as a complimentary localization technique to provide continuous state information.

The following sections describe the INS formulation, WO update, using pseudo-measurement updates in the filter, and proprioceptive slip detection. The field tests corresponding to this implementation are given at the end of this chapter. The overall system architecture of the method is given in Fig 5.1.1.

5.2 Inertial Navigation System Mechanization

INS is a self-contained dead-reckoning system that integrates IMU acceleration and angular rate measurements to provide a navigation solution. This solution often suffers from error accumulation, integration of misaligned sensors, and modeling errors. Estimating attitude, velocity, and position can be handled by an iterative navigation filter implementation in the system's processor unit. For attitude estimation, attitude sensors must account for the body-frame rotation over time. For velocity estimation, the accelerometers must be rotated from the body frame to the re-

Figure 5.1.1: This figure presents a graphical depiction of the implemented navigation system when all the algorithm components are utilized. Within the algorithm, the IMU observations are used to propagate the current state estimate. In conjunction with non-holonomic constraints, wheel odometry observations are leveraged to reduce the unbounded error growth of the unaided inertial navigation system. The zero updates (ZUPT, ZARU) can be triggered by an error observer or manual stops.

solving frame. Also, the gravity and planet's rotation must be modeled accurately to have a reliable estimation. For position estimation, integrating the velocity causes a double accumulation of the error. Furthermore, the quality of an INS solution depends on the state's initialization since it is a recursive process. The initialization process for this implementation is described in Chapter 3.4.

In this dissertation, the navigation equations are implemented in a locally level navigation frame following the strapdown INS mechanization process as detailed in [70, 71, 72] with closely following the notation in [57], and used in [1, 54]. A more detailed derivation of the state updates and INS error-state model are provided in Appendix A.1. Each component of the INS estimation framework is discussed below.

Attitude Update: The attitude estimation in a locally level navigation frame implementation can be expressed as the body to navigation frame coordinate transformation matrix. The attitude update is given as

$$\mathbf{C}_b^{n(+)} \approx \mathbf{C}_b^{n(-)} (\mathbf{I}_3 + \boldsymbol{\Omega}_{ib}^b \tau_s) - (\boldsymbol{\Omega}_{ie}^n + \boldsymbol{\Omega}_{en}^n) \mathbf{C}_b^n \tau_s \quad (5.1)$$

where $\mathbf{C}_b^n$ is the coordinate transformation matrix from the body frame to the locally level frame, $\mathbf{I}_3$ is a 3-by-3 identity matrix, $\boldsymbol{\Omega}_{ib}^b$ is the skew symmetric matrix of the IMU angular rate measurement, $\boldsymbol{\Omega}_{ie}^n$ is the skew symmetric matrix of the planet's rotation vector represented in the locally level frame, $\boldsymbol{\Omega}_{en}^n$ is the transport term, and τ_s is the IMU sampling interval.

Velocity Update: Assuming that the variations of the acceleration due to gravity, Coriolis, and transport rate terms are all negligible over the integration interval, the velocity update is given as,

$$\mathbf{v}_{eb}^{n(+)} \approx \mathbf{v}_{eb}^{n(-)} + (\mathbf{f}_{ib}^n + \mathbf{g}_b^n(L_b^{(-)}, h_b^{(-)}) - (\boldsymbol{\Omega}_{en}^{n(-)} + 2\boldsymbol{\Omega}_{ie}^{n(-)})\mathbf{v}_{eb}^{n(-)})\tau_s \quad (5.2)$$

where $\mathbf{v}_{eb}^n$ is the velocity update, $\mathbf{f}_{ib}^n$ is the specific force measurements from the IMU acceleration sensors, $\mathbf{g}_b^n$ is the gravity vector. The velocity estimation is given as planet referenced in locally level navigation frame.

Accelerometers measure the specific force which is the acceleration relative to free fall or the total non-gravitational force per unit mass. After the attitude has been updated, the accelerometer measured specific force transformation is applied. The transformation is necessary since the accelerometers measure specific force along the body-axis [57].

$$\mathbf{f}_{ib}^n \approx \frac{1}{2} (\mathbf{C}_b^{n(-)} + \mathbf{C}_b^{n(+)}) \mathbf{f}_{ib}^b \quad (5.3)$$

Position Update: Assuming the velocity variation is linear over the integration interval, the position update in the curvilinear form (latitude, longitude, height) is given as

$$h_b^{(+)} = h_b^{(-)} - \frac{\tau_s}{2} \left(\mathbf{v}_{eb,D}^{n(-)} + \mathbf{v}_{eb,D}^{n(+)} \right) \quad (5.4)$$

$$L_b^{(+)} = L_b^{(-)} + \frac{\tau_s}{2} \frac{\mathbf{v}_{eb,N}^{n(-)}}{R_N(L_b^{(-)}) + h_b^{(-)}} + \frac{\tau_s}{2} \frac{\mathbf{v}_{eb,N}^{n(+)}}{R_N(L_b^{(-)}) + h_b^{(+)}} \quad (5.5)$$

$$\lambda_b^{(+)} = \lambda_b^{(-)} + \frac{\tau_s}{2} \frac{\mathbf{v}_{eb,E}^n{}^{(-)}}{(R_E(L_b^{(-)}) + h_b^{(-)}) \cos L_b^{(-)}} + \frac{\tau_s}{2} \frac{\mathbf{v}_{eb,E}^n{}^{(+)}}{(R_P(L_b^{(-)}) + h_b^{(+)}) \cos L_b^{(+)}} \quad (5.6)$$

where h_b , L_b and λ_b are updated position estimates (expressed in terms of height, latitude, and longitude, respectively), R_N is the variation of the meridian, and R_P is transverse radii of curvature.

5.3 Error State Extended Kalman Filter

In the error state EKF, the error state is estimated instead of the total state. Then the estimate of this error is used to correct the total state [62, 57]. The total state vector is given as

$$\mathbf{x}^n = \left(\Psi_{nb}^n \mathbf{v}_{eb}^n \mathbf{p}_b \right)^T \quad (5.7)$$

Error State Model: The error state vector is constructed in a local navigation frame,

$$\delta \mathbf{x}^n = \left(\delta \Psi_{nb}^n \delta \mathbf{v}_{eb}^n \delta \mathbf{p}_b \mathbf{b}_a \mathbf{b}_g \right)^T, \quad \delta \mathbf{p}_b = \left(\delta L_b \delta \lambda_b \delta h_b \right)^T \quad (5.8)$$

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. The position error is expressed in terms of the latitude, longitude, and height, respectively. The steps of the filter are given as propagation and correction in the following pages. Then, the elements in these steps are described.

Propagation Step: In the propagation step, the error state is propagated by using the the State Transition Matrix (STM) in discrete time such that

$$\delta \tilde{\mathbf{x}}_k = \Phi_{k-1} \delta \hat{\mathbf{x}}_{k-1} \quad (5.9)$$

where $\Phi \in \mathbb{R}^{15 \times 15}$ is the STM. Similarly, the error covariance matrix $\mathbf{P} \in \mathbb{R}^{15 \times 15}$ is propagated through if pseudo-measurements are unavailable to use

$$\check{\mathbf{P}}_k = \Phi_{k-1} \mathbf{P}_{k-1} \Phi_{k-1}^T + \mathbf{L}_{k-1} \mathbf{Q}_{k-1} \mathbf{L}_{k-1}^T \quad (5.10)$$

where $\mathbf{Q}_k$ is the process noise covariance, $\mathbf{L}_k$ is the process noise related Jacobian, and it is identity in our case since the noise is assumed to be additive. The optimal gain can be calculated as;

$$\mathbf{K}_k = \check{\mathbf{P}}_k \mathbf{H}_k^T (\mathbf{H}_k \check{\mathbf{P}}_k \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (5.11)$$

where $\mathbf{H}_k$ is the measurement matrix and $\mathbf{R}_k$ is the measurement noise covariance. The estimate of the error state can be given as;

$$\delta \hat{\mathbf{x}}_k = \mathbf{K}_k (\delta \mathbf{z}_k - \mathbf{H}_k \delta \tilde{\mathbf{x}}_k) \quad (5.12)$$

where $\mathbf{z}_k$ is the corresponding available pseudo-measurement innovation that is given for each updates in 5.4.

Correction Step: After having the error state estimation, it can be used to correct the state. Starting from coordinate transformation matrix correction,

$$\hat{\mathbf{C}}_n^b = (\mathbf{I}_3 - [\delta\mathbf{\Psi}_k] \times) \check{\mathbf{C}}_n^{bT}. \quad (5.13)$$

Then, the attitude correction can be given as;

$$\hat{\mathbf{\Psi}}_{nb}^n = \begin{bmatrix} \text{atan2}(\hat{\mathbf{C}}_{n(3,2)}^b, \hat{\mathbf{C}}_{n(3,3)}^b) \\ \text{asin}(-\hat{\mathbf{C}}_{n(3,1)}^b) \\ \text{atan2}(\hat{\mathbf{C}}_{n(2,1)}^b, \hat{\mathbf{C}}_{n(1,1)}^b) \end{bmatrix}. \quad (5.14)$$

The velocity and position corrections are given as;

$$\hat{\mathbf{v}}_{eb}^n = \check{\mathbf{v}}_{eb}^n + \delta\mathbf{v}_{eb}^n \quad (5.15)$$

$$\hat{\mathbf{p}}_b = \check{\mathbf{p}}_b + \delta\mathbf{p}_b. \quad (5.16)$$

Finally, the error covariance matrix can be corrected as;

$$\hat{\mathbf{P}}_k = (1 - \mathbf{K}_k \mathbf{H}_k) \check{\mathbf{P}}_k. \quad (5.17)$$

INS System Matrix: To calculate the INS system matrix, the Jacobian of the error-state equations is determined. After defining the time derivatives of the error-state equations, the system matrix and the STM are defined. Then, the errors are transformed into the local navigation frame, and the time derivative of the velocity error is constructed by adding the transport rate term. The

system matrix can be given as;

$$\mathbf{F}_{INS}^n = \begin{pmatrix} -[\hat{\boldsymbol{\omega}}_{in}^n \wedge] & \mathbf{F}_{12}^n & \mathbf{F}_{13}^n & \mathbf{0}_3 & \hat{\mathbf{C}}_b^n \\ -[\left(\hat{\mathbf{C}}_b^n \hat{\mathbf{f}}_{ib}^b\right) \wedge] & \mathbf{F}_{22}^n & \mathbf{F}_{23}^n & \hat{\mathbf{C}}_b^n & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{F}_{32}^n & \mathbf{F}_{33}^n & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{pmatrix}. \quad (5.18)$$

where $\hat{\boldsymbol{\omega}}_{in}^n$ is the IMU gyro rate outputs resolved in navigation frame, $\hat{\mathbf{f}}_{ib}^b$ is the IMU acceleration outputs in the body frame, and notation $\wedge$ stands for the skew-symmetric matrix of the vector. The remaining system matrix elements are given in Appendix A.2.

INS State Transition Matrix: Using the time derivatives of the error-state equations, the state transition model can be assumed as;

$$\Phi_k \approx e^{\mathbf{F}_k \tau_s} = \sum_{\alpha=0}^{\infty} \frac{\mathbf{F}_k^\alpha \tau_s^\alpha}{\alpha!}. \quad (5.19)$$

Neglecting the higher order terms after the first order, the elements of the STM for discrete time can be approximated to

$$\Phi_k^n \approx \begin{bmatrix} \mathbf{I}_3 + \mathbf{F}_{11}^n \tau_s & \mathbf{F}_{12}^n \tau_s & \mathbf{F}_{13}^n \tau_s & \mathbf{0}_3 & \hat{\mathbf{C}}_b^n \tau_s \\ \mathbf{F}_{21}^n \tau_s & \mathbf{I}_3 + \mathbf{F}_{22}^n \tau_s & \mathbf{F}_{23}^n \tau_s & \hat{\mathbf{C}}_b^n \tau_s & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{F}_{32}^n \tau_s & \mathbf{I}_3 + \mathbf{F}_{33}^n \tau_s & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \quad (5.20)$$

INS Noise Covariance Matrix: Integrating the power spectral densities of the IMU noise through the state propagation, the INS system noise covariance matrix can be constructed as;

$$\mathbf{Q}_{INS}^n = \begin{pmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{21}^{nT} & \mathbf{Q}_{31}^{nT} & \mathbf{0}_3 & \frac{1}{2}S_{bgd}\tau_s^2\hat{\mathbf{C}}_b^n \\ \mathbf{Q}_{21}^n & \mathbf{Q}_{22}^n & \mathbf{Q}_{32}^{nT} & \frac{1}{2}S_{bad}\tau_s^2\hat{\mathbf{C}}_b^n & \frac{1}{3}S_{bgd}\tau_s^3\mathbf{F}_{21}^n\hat{\mathbf{C}}_b^n \\ \mathbf{Q}_{31}^n & \mathbf{Q}_{32}^n & \mathbf{Q}_{33}^n & \mathbf{Q}_{34}^n & \mathbf{Q}_{35}^n \\ \mathbf{0}_3 & \frac{1}{2}S_{bad}\tau_s^2\hat{\mathbf{C}}_n^b & \mathbf{Q}_{34}^{nT} & S_{bad}\tau_s\mathbf{I}_3 & \mathbf{0}_3 \\ \frac{1}{2}S_{bgd}\tau_s^2\hat{\mathbf{C}}_n^b & \frac{1}{3}S_{bgd}\tau_s^3\mathbf{F}_{21}^n\hat{\mathbf{C}}_n^b & \mathbf{Q}_{35}^{nT} & \mathbf{0}_3 & S_{bgd}\tau_s\mathbf{I}_3 \end{pmatrix} \quad (5.21)$$

where S_{rg} , S_{ra} , S_{bad} , and S_{bgd} are the power spectral densities of the gyro random noise, accelerometer random noise, accelerometer bias variation, and gyro bias variation, respectively [57]. INS system noise covariance matrix is defined by the random walk of the velocity error due to noise on the accelerometer specific-force measurements and random walk of the attitude error due to noise on the gyro angular rate measurements by closely following the notation in [57]. The elements of the INS system noise covariance matrix in 5.21 are given in Appendix A.2.

5.4 Pseudo-Measurement Updates

Providing a reliable inertial localization system requires the calibration of the IMU outputs. One way to calibrate the bias can be achieved by using the additional sensor outputs (e.g., magnetometers, sun sensors). However, magnetometers are usually not useful for obtaining global orientation when the planet of interest does not possess a global magnetic field. Similarly, sun sensors are only useful when the sun is within the sensor's field of view. Another way to calibrate the IMU

sensor biases can be achieved by utilizing pseudo-measurements in certain conditions from the sensors already onboard. For example, planetary rovers frequently stop for a variety of reasons, such as safety checks [13], mast pointing [10], image processing [73], and conducting scientific experiments [14]. Stopping is unavoidable even with the ideal case of the thinking-while-driving (TWD) approach, which has recently been developed to minimize how often the rover needs to stop for Perseverance [25]. Since a rover is in stationary conditions in many instances, leveraging this state information is a natural fit for planetary robots.

Pseudo-measurements are the constraints that observable in certain kinematic and physical conditions. These pseudo-measurements can be applied as a measurement update to enforce constraints on the states of a system and can be used in the system state estimation process in a cost-effective way because the information is mostly gathered from the sensors already on-board. There are several strategies to take advantage of pseudo-measurements including, but not limited to, zero velocity/angular rate [74, 54], zero displacement [75], constant height/slope [76], and non holonomicity [77, 1]. A toy example to demonstrate the pseudo-measurement capability of reducing error growth in the INS-based localization is given in Fig. 5.4.1. In this figure, a static IMU output with 50 Hz data rate under the effect of Earth's gravity field is processed with and without pseudo-measurement updates. Since the used IMU is stationary for this toy example, we were able to control when to enable pseudo-measurement updates in the state estimation. For example, ZUPT and ZARU (zero updates) are enabled for one step size of the estimation (0.02 s), every 40 seconds. Non-holonomic constraint updates are used in each time step of the INS estimation. After 200th second, zero updates are kept active. While it may be intuitive that these constraints provide information useful for calibrating IMU sensor biases, this figure also illustrates that the position errors are further reduced to some extent after an update. This is due to the fact that correlation of IMU

biases to position errors are modeled and integrated over time in the INS process noise covariance matrix. In the following sections, implementation details of the pseudo-measurement updates in an error state EKF for INS based state estimation are given.

Figure 5.4.1: Toy example for using pseudo-measurements in the inertial positioning estimation. Left figure (a): INS based positioning with a static IMU under Earth gravity vector. Black line shows the INS only positioning estimation error, red line shows the error when using zero updates (labeled as ZU for both ZUPT and ZARU) in the INS estimation, the blue line shows the error when using zero updates and nonholonomic (labeled as NH) constraints in the INS estimation, and dashed black lines indicates the time step when the zero update is enabled for a step size based on the IMU data rate (0.02 s). The 3D positioning error is shown with a logarithmic scale. Right figure (b): The figure is the magnified part of the left figure (gray shaded area) between 0 to 50 seconds. The position error is plotted with a linear scale.

5.4.1 Zero Updates

Zero Velocity Update

ZUPT is a commonly used concept that was initially popularized to aid inertial pedestrian navigation [49, 50, 78, 79]. It can bound the velocity error and help to calibrate IMU sensor noises [80].

This process helps to reduce the INS positioning error growth since the error state model justifies the correlation between the position and velocity errors of the error covariance matrix. Using ZUPT in state estimation does not need any dedicated sensor to provide zero velocity information, and this information can be obtained by the sensors already on-board (e.g., IMU, wheel encoders). ZUPT requires stationary conditions, and it can be used as an opportunistic navigational update if a wheeled robot stops for other reasons (e.g., obstacle avoidance, re-planning, waiting for pedestrians, stopping at traffic lights). Also, ZUPT can be used to improve WO/INS proprioceptive localization with periodic stops in GNSS-denied/degraded [75], poor lighting/feature areas [1] and with autonomous stops by deciding when to stop [54]. Since planetary rovers are in stationary conditions in many instances, leveraging this state information is a natural fit for wheeled planetary robots. The ZUPT algorithm can provide this valuable information as a pseudo-measurement update. Using ZUPT in a planetary localization method can provide computationally efficient and accurate real-time rover localization capability without changing any rover hardware or major operations except the acquisition of the IMU and wheel encoder observations. This is a particularly desirable consequence of using ZUPT in space robotics, as computational resources in planetary rovers are limited by radiation-hardened hardware.

IMU sensor outputs are governed by sensor errors during the zero velocity. The measurement noise covariance describes the variance and covariance of the nominally-zero velocity due to vibration and disturbances. This fact is used when performing zero velocity update and the measurements fed into EKF to reduce the position error growth of the system. The measurement innovation for ZUPT is

$$\delta \mathbf{z}_{ZV,k}^{\gamma-} = -\hat{\mathbf{v}}_{eb,k}^{\gamma} \quad \gamma \in e, n \quad (5.22)$$

and the measurement matrix is

$$\mathbf{H}_{ZV,k}^\gamma = \begin{pmatrix} \mathbf{0}_3 & -\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{pmatrix} \quad \gamma \in e, n. \quad (5.23)$$

Although ZUPT does not provide absolute position information, the Kalman filter system model builds up information on the correlation between the velocity and position errors in the off-diagonal elements of the error covariance matrix and the cubic error growth for positioning is reduced to linear [80]. This enables a ZUPT to correct most of the position drift since the last measurement update.

Zero Angular Rate Update

During stationary conditions, ZUPT can bound the roll and pitch errors; however, heading error may grow rapidly due to poor observation of heading [74, 81]. In this case, ZARU can be performed to reduce the heading drift. Therefore, a ZARU can be performed whenever a ZUPT is performed. Additionally, this can be detected by comparing the standard deviation of the recent yaw-rate gyro measurements, the standard deviation of the steering-angle commands (considering steerable wheels) and the yaw rate obtained from differential odometry.

The measurement innovation for a ZARU can be given independent of the coordinate frames used for position, velocity and attitude states as

$$\delta \mathbf{z}_{ZA,k}^- = -\hat{\omega}_{ib,k}^b \quad (5.24)$$

and the measurement matrix is

$$\mathbf{H}_{ZA,k} = \begin{pmatrix} \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & -\mathbf{I}_3 \end{pmatrix} \quad (5.25)$$

5.4.2 Non-Holonomic Constraints

A rover is a non-holonomic system if its number of controllable degrees of freedom is less than its total degrees of freedom. A skid-steer rover is subject to two motion constraints if the rover is not experiencing side slip and motion normal to the surface: 1) the velocity of the vehicle is zero along the rotation axis of any of its wheels, and 2) velocity is also zero in the direction perpendicular to the traversal surface [82]. This can be interpreted as that these constraints become invalid when the rover experiences a slip on its sideways or a wheel loses its contact to the surface for an extended period. Due to frame rotations, zero vertical and lateral velocity does not mean that acceleration on these directions are zero.

Following the similar process as in [57], it is assumed that the error-state vector is defined by (5.8) and the total state vector is

$$\mathbf{x}^n = \begin{pmatrix} \boldsymbol{\Psi}_{nb}^n & \mathbf{v}_{eb}^n & \mathbf{p}_b \end{pmatrix}^T. \quad (5.26)$$

The rover velocity constraints can be applied as a pseudo-measurement update, assuming that the axes of the rear-wheel frame are aligned with the body frame. This measurement update can be expressed as

$$\delta \mathbf{z}_{RC}^n = - \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} (\mathbf{C}_n^b \mathbf{v}_{eb}^n - [\omega_{ib}^b \wedge] \mathbf{L}_{rb}^b) \quad (5.27)$$

where $\mathbf{L}_{br}^b$ is body to rear wheel lever arm, and ω_{ib}^b is angular rate measurement. Then, correspond-

ing measurement matrix may be approximated as

$$\mathbf{H}_{RC}^n = \begin{pmatrix} \mathbf{0}_{2,3} & \begin{bmatrix} -\mathbf{H}_l \\ -\mathbf{H}_v \end{bmatrix} & \mathbf{0}_{2,3} & \mathbf{0}_{2,3} & \mathbf{0}_{2,3} \end{pmatrix} \quad (5.28)$$

where $\mathbf{H}_l$ is lateral constraint part. and $\mathbf{H}_v$ is the vertical part of the measurement matrix.

$$\begin{bmatrix} -\mathbf{H}_l \\ -\mathbf{H}_v \end{bmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{C}_n^b \quad (5.29)$$

Although the non-holonomic constraint measurement update is performed whenever the navigation solution is updated, the excessive sideslip invalidates the lateral velocity constraint, and this adds extra biases to velocity solution. While turning, the heading rate of the rover can be observed, and if it exceeds a predetermined threshold, the lateral velocity constraint measurement can be omitted.

5.4.3 Wheel Odometry

In a traditional wheel odometry based dead-reckoning system, horizontal position and heading are estimated by measuring the wheel rotations with encoders. The wheel odometry outputs consist of wheel forward speed and heading rate which are functions of wheel diameter, trackwidth and wheelbase values. Heading rate is calculated by differencing the average linear velocity values of left and right wheels. The wheel odometry can also be assisted with gyroscope estimations from an IMU to give a better pose estimation. Instead, in this implementation, the wheel odometry is used to assist the INS based state estimation as a measurement update in the filter. The measurement

innovation is derived as a similar procedure as described in [82] and [57] with using the fact that the outputs of the wheel odometry do not consist of vertical and lateral velocities, which can be interpreted as vertical and lateral velocity constraint (pseudo-measurements, see Section 5.4) as

$$\delta \mathbf{z}_O = \begin{pmatrix} \tilde{v}_{lon,O} - \tilde{v}_{lon,i} \\ -\tilde{v}_{lat,i} \\ -\tilde{v}_{ver,i} \\ \tilde{\psi}_{nb,o} - \tilde{\psi}_{nb,i} \cos \hat{\theta}_{nb} \end{pmatrix} \quad (5.30)$$

where $\tilde{v}_{lon}$, $\tilde{v}_{lat}$, and $\tilde{v}_{ver}$ are estimated longitudinal, lateral and vertical rear wheel speed, respectively. The subscript i denotes the estimated INS solution, and O denotes the wheel odometry measurements. Heading rate is given as $\tilde{\psi}_{nb}$ and $\hat{\theta}_{nb}$ is the estimated pitch angle of the rover body frame with respect to the navigation frame. The frame alignment of the estimated inertial velocity can be obtained as

$$\begin{bmatrix} \tilde{v}_{lon,i} \\ \tilde{v}_{lat,i} \\ \tilde{v}_{ver,i} \end{bmatrix} = \frac{1}{\tau_0} \int_{t-\tau_0}^t \mathbf{I}_3 (\mathbf{C}_n^b(\tau) \mathbf{v}_{eb}^n(\tau) + [\omega_{eb}^b(\tau) \wedge] \mathbf{L}_{br}^b) d\tau \quad (5.31)$$

where $\mathbf{L}_{br}^b$ is body to rear wheel lever arm with the assumption that the axes of the rear and front wheel frames are aligned with the body frame and ω_{eb}^b is angular rate measurements. The aligned velocities with respect to the lever arm are pre-integrated between each odometry update.

The measurement matrix is,

$$\mathbf{H}_{O,k}^n = \begin{bmatrix} \mathbf{H}_{O1,1}^n & \mathbf{H}_{O1,2}^n & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{H}_{O2,1}^n & \mathbf{H}_{O2,2}^n & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{H}_{O3,1}^n & \mathbf{H}_{O3,2}^n & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{H}_{O4,1}^n & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{H}_{O4,4}^n & \mathbf{0}_3 \end{bmatrix}, \quad \mathbf{H}_{O,k}^n \in \mathbb{R}^{4 \times 15} \quad (5.32)$$

where

$$\begin{bmatrix} \mathbf{H}_{O1,1}^n \\ \mathbf{H}_{O2,1}^n \\ \mathbf{H}_{O3,1}^n \end{bmatrix} = -\frac{1}{\tau_0} \int_{t-\tau_0}^t \mathbf{I}_3 \mathbf{C}_n^b(\tau) [\mathbf{v}_{eb}^n(\tau) \wedge] d\tau, \quad (5.33)$$

$$\begin{bmatrix} \mathbf{H}_{O1,2}^n \\ \mathbf{H}_{O2,2}^n \\ \mathbf{H}_{O3,2}^n \end{bmatrix} = -\frac{1}{\tau_0} \int_{t-\tau_0}^t \mathbf{I}_3 \mathbf{C}_n^b(\tau) d\tau, \quad (5.34)$$

$$\mathbf{H}_{O4,1}^n = \frac{1}{\tau_0^2} [\hat{\psi}_{nb}(t) - \hat{\psi}_{nb}(t - \tau_0)] \int_{t-\tau_0}^t \sin \hat{\theta}_{nb} \begin{bmatrix} 0 \\ \cos \hat{\phi}_{nb} \\ \sin \hat{\phi}_{nb} \end{bmatrix}^T \mathbf{C}_n^b(\tau) d\tau, \quad (5.35)$$

$$\mathbf{H}_{O4,4}^n = -\frac{\cos \hat{\theta}_{nb}}{\tau_0} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \mathbf{C}_b^n. \quad (5.36)$$

Note that the coordinate transformation matrix in $\mathbf{H}_{O44}^n$ is from the body to the inertial navigation frame.

5.5 Field Tests

Preliminary tests were performed to verify the software reliability, observe the robot behaviour, and determine the practical limits of the study. In these tests, Husky and Pathfinder rovers are driven in both indoor and outdoor environments. Apart from preliminary tests, a larger set of experiments on the West Virginia University campus were performed on flat gravel/pebble terrain and non-flat rough terrain. It is observed that the gravel road consists of small broken rock materials similar to the sharp-edged Martian rocks (ventifacts) and smooth-edged river rocks. Due to the shape of these materials, there is less traction on the wheels on the gravel road. This loose surface provides the opportunity to test our framework to estimate slip, primarily due to wheel kinematic incompatibilities. Non-flat rough terrain consists of several unknown local variances (e.g., bumps and holes). These local variances most likely causes slippage due to kinematic incompatibility while traversing. Overall, field test environments for the Husky and Pathfinder rovers include hard tile, linoleum, concrete, lawn, wet sand, gravel, asphalt, and river rocks (see Fig. 5.5.1).

The experiments are conducted at relatively high speed compared to what is expected from a planetary mission. This is mainly due to limited generated torque resolution from the wheel motors for the Pathfinder rover. The high speed results are expected to have more noisy IMU outputs than slow speed results due to the generated vibration during traversal. On the other hand, increasing the traversal time without using pseudo-measurements will deteriorate the INS estimation. Due to the limitation of achieving extreme low speeds as in planetary missions, several possible velocity profiles are tested to observe and compare differences.

To demonstrate the proposed approach, specifically four different scenarios are detailed: 1) Concrete-Turn, flat, L-Shaped concrete path. 2) Rough-Terrain, long path on a non-flat, muddy, and grassy

Figure 5.5.1: Husky rover on traversed terrain types to test the developed navigation approach.

Table 5.5.1: Details of Performed Scenarios

Scenario Type	Σ Distance (m)	ADTS(s)	Σ Stops	Σ Driving Time (s)	Rover	Velocity (m/s)
Concrete-Turn	34	12	7	85	Husky	0.4
Rough-Terrain	151	9	42	406	Husky	0.4
Fast-Rectangle	87	15	8	120	Pathfinder	0.8
Slow-Rectangle	85	15	21	330	Pathfinder	0.2

terrain with embedded rocks. 3) Fast-Rectangle, rectangle path on a non-flat, grassy terrain with high speed. 4) Slow-Rectangle, rectangle path on a non-flat, grassy terrain with slow speed. For each scenario, navigation stops were periodically commanded. The total distance traversed, Average Driving Time between Stops (ADTS), the number of performed stops, and the total driving time are listed in Table 5.5.1. The Husky rover has 0.4 m/s commanded velocity while driving in Concrete-Turn and Rough-Terrain scenarios. The Pathfinder rover has 0.8 m/s, and 0.2 m/s average commanded velocity in Fast-Rectangle and Slow-Rectangle scenarios, respectively.

5.6 Results and Discussion

To assess the relative benefit of each aspect of the approach, many of the update combinations are performed for each of the test runs. The positioning error values with respect to each of the update combinations are given for each scenario in Tables 5.6.1 - 5.6.4, where RMS denotes root mean square, STD denotes standard deviation of the horizontal error, and Max. is the worst-case horizontal distance error that rover encounters during each test.

The findings from these results suggest that without using zero updates, the rover localization performance is inadequate due to INS integration accumulated error. Wheel odometry helps to reduce drift along with non-holonomic constraints within the INS solution; however, if the zero updates are applied, most of the accumulated error is mitigated, as explained in Section 5.4.

While it is intuitive to see the effectiveness of the zero updates from the position error results, the

Table 5.6.1: Position Error for Concrete-Turn Scenario

Estimation I O Z N B *	Horizontal Error (m)			RMS Error (m)		
	Median	STD	Max.	East	North	Up
I	518.44	494.77	1606.2	783.75	64.75	153.07
I+O	23.73	12.59	34.26	21.80	10.08	7.07
I+O+N	23.69	11.89	32.34	20.57	10.57	7.84
I+O+N+B	23.75	12.25	33.30	21.18	10.45	7.57
I+Z	2.86	6.42	31.09	7.38	4.28	4.46
I+Z+B	0.60	1.58	31.62	1.23	1.46	1.68
I+Z+N	0.58	1.19	8.54	1.28	0.81	0.81
I+Z+N+B	0.63	1.13	7.69	1.28	0.74	0.95
I+Z+O	0.68	0.31	1.33	0.67	0.31	0.67
I+Z+B+O	0.66	0.31	1.33	0.66	0.31	0.69
I+Z+N+O	0.49	0.25	1.20	0.43	0.36	0.71
I+Z+N+B+O	0.48	0.25	1.21	0.43	0.36	0.73

* I:INS, O:Odometry, Z:Zero-Type, N:Non-Holonomic, B:Backward Smoothing.

Table 5.6.2: Position Error for Rough-Terrain Scenario

Estimation I O Z N B *	Horizontal Error (m)			RMS Error (m)		
	Median	STD	Max.	East	North	Up
I	14253.	20742.	68676.	29200.	6045.	11334.
I+O	21.31	18.69	63.90	23.37	17.45	18.07
I+O+N	30.68	16.30	56.53	29.23	16.94	11.65
I+O+N+B	30.84	16.35	55.71	28.92	17.06	11.97
I+Z	1.77	3.13	31.40	4.02	1.09	2.19
I+Z+B	1.55	1.23	29.75	2.06	0.96	0.86
I+Z+N	0.93	0.99	4.08	1.46	0.73	0.92
I+Z+N+B	1.59	1.07	4.01	1.79	1.09	0.58
I+Z+O	1.36	1.29	4.28	2.16	0.29	1.18
I+Z+B+O	1.21	1.28	4.29	2.13	0.32	0.90
I+Z+N+O	0.47	0.90	2.86	1.25	0.27	1.04
I+Z+N+B+O	0.54	0.93	2.89	1.28	0.41	0.80

* Same as Table 5.6.1

Table 5.6.3: Position Error for Fast-Rectangle Scenario

Estimation I O Z N B *	Horizontal Error (m)			RMS Error (m)		
	Median	STD	Max.	East	North	Up
I	7801.5	8945.6	29937.	13482.	1450.	3783.
I+O	15.85	8.83	30.43	13.97	8.22	8.52
I+O+N	7.04	9.26	29.08	12.15	6.08	5.06
I+O+N+B	4.90	7.00	22.19	9.38	4.64	6.71
I+Z	5.42	10.27	73.16	12.18	3.01	5.04
I+Z+B	4.20	4.85	118.28	4.73	4.52	5.24
I+Z+N	2.26	1.38	4.69	1.56	1.82	1.82
I+Z+N+B	0.96	1.45	8.29	1.54	1.35	1.81
I+Z+O	1.29	1.05	3.44	1.06	1.33	1.31
I+Z+B+O	1.78	1.06	3.64	1.47	1.19	0.62
I+Z+N+O	1.34	1.03	3.50	1.09	1.33	1.98
I+Z+N+B+O	0.92	0.84	2.77	0.80	1.06	1.19

* Same as Table 5.6.1

solution is further enhanced with the non-holonomic constraints. Taken together, the results reported here supports the idea that usage pseudo-measurement significantly improves the proprioceptive localization reliability during the blind driving. Although non-holonomic constraints handle most of the lateral motion drift, when the rover heading rate is significant (e.g., if it exceeds 0.1

Table 5.6.4: Position Error for Slow-Rectangle Scenario

Estimation I O Z N B *	Horizontal Error (m)			RMS Error (m)		
	Median	STD	Max.	East	North	Up
I	25771.	31574.	98873.	46708.	6222.	14016.
I+O	8.71	8.51	28.19	14.22	4.77	4.09
I+O+N	12.81	10.81	34.76	16.71	9.28	8.84
I+O+N+B	10.95	8.19	29.09	14.04	3.48	9.44
I+Z	4.34	5.50	43.70	6.84	3.81	3.14
I+Z+B	4.68	2.66	97.09	3.80	3.73	2.93
I+Z+N	3.17	1.23	6.89	1.73	2.55	1.08
I+Z+N+B	1.77	1.24	13.44	1.95	1.19	1.04
I+Z+O	3.74	1.13	4.61	2.09	2.75	1.88
I+Z+B+O	3.99	1.19	5.05	3.04	2.40	1.18
I+Z+N+O	1.67	0.68	2.87	0.93	1.51	1.69
I+Z+N+B+O	1.34	0.69	2.90	0.95	1.34	1.77

* Same as Table 5.6.1

rad/s), lateral velocity constraint measurement updates become less reliable.

Since the terrain of the Concrete-Turn scenario is relatively benign and short to traverse, another test is performed on a more difficult terrain for a longer distance. This test case is called as Rough-Terrain. The position estimates with 3σ error covariances are given in Fig. 5.6.1 for the Rough-Terrain case. Also, a 3D view of the position estimation when using all of the proposed updates is given in Fig. 5.6.2. The proposed algorithm is also tested on Pathfinder by keeping the algorithm same, but changing the rover specific values (e.g., wheel radius, IMU mount distance, wheelbase). Moreover, Pathfinder is used for a higher velocity (0.8 m/s) scenario, called as Fast-Rectangle. In this scenario, the rover is remotely controlled to follow a rectangle shaped path. In order to observe the behaviour of the rover at a lower velocity profile, another test is performed following the same path as the Fast-Rectangle; but with a lower velocity (0.2 m/s). This scenario is called as Slow-Rectangle. A ground track view of the Slow-Rectangle scenario when using all of the pro-

Figure 5.6.1: North (top) and east (bottom) error position estimates for Rough-Terrain case. Black line is the GPS solution, red line is the position estimate, and the gray region is the 3σ covariance hull. RMSE East is 1.28 m and RMSE North is 0.41 m, total distance is 150 m.

Figure 5.6.2: 3D view of the rover traversal for Rough-Terrain in Latitude, Longitude, Height (LLH) coordinates. The GPS solution and filter estimation are plotted on USGS Digital Elevation Model (DEM) data of the traversed region. The 3D RMS position error at the end is 1.62% of the traveled distance.

Figure 5.6.3: Ground track view of Pathfinder (Slow-Rectangle) scenario. Red line shows the filter estimated solution, black line shows the GPS solution. Overall 2D position error is 0.92 m with 0.84 m standard deviation.

posed updates is given in Fig. 5.6.3. In this test, horizontal position error against DGPS solution is calculated as 0.92 m at the end of 85 m traversal distance. Based on these results, the achieved localization accuracy can be approximated as greater than 97% for these terrains.

To further evaluate the method and better visualize the effects of the zero updates, the collected data for Fast-Rectangle and Concrete-Turn cases are post-processed only using INS and zero-updates without leveraging the wheel odometry information for Pathfinder in Fig. 5.6.4 and for Husky in Fig. 5.6.5, respectively. Similar to the toy example that is given in Fig. 5.4.1, these figures support the idea that using zero updates quickly reduces the growth of the position error and also corrects

Figure 5.6.4: North and East error estimates without using wheel odometry for Fast-Rectangle case. Black line is GPS estimated solution, red line is position estimate, and gray region is 3σ covariance hull. The highest peak values are the maximum worst case distance error. Zero updates recover the position solution when the rover stops.

most of the position drift.

In addition to the pseudo-measurement update cases, the backward smoothing extension, which is used for smoothing out the estimated state along the traversed path between zero updates, is added in the position error tables for observing its effectiveness along with other updates. This smoothing method is the Rauch-Tung-Striebel (RTS) backward recursive method [83], which utilizes all the available state information and goes backward in time. In this formulation, the most interesting observation was backward smoothing is shown to be most effective only if non-holonomic constraints are not applied mainly for the Husky rover. Since the test platforms (both Husky and Pathfinder) are non-holonomic rovers, this backward smoothing implementation could be useful

Figure 5.6.5: North error estimate without odometry aiding for Concrete-Turn case. Black line is DGPS solution, red line is north position estimate, and gray region is 3σ covariance hull. Filter estimation exceeds the covariance bounds while turning with wheel slips. However, ZUPT reduces the cubic error growth to linear for positioning and correct most of the position drift when the rover stops.

for a holonomic rover. Additional details of this implementation can be found in [1].

5.6.1 Key Takeaways

Existing dead reckoning or wheel-odometry solutions are not reliable for the long term due to inherent navigation drift accumulation. This is a well-known curse of inertial navigation. A layered approach can be implemented by taking advantage of common sensor information sources to achieve the required localization performance with a minimum burden on computational resources. If the navigation error is less than a mission-specific, predetermined threshold, there would be no need to run the more computationally intensive localization functions. Therefore, this chapter's aim was to provide quality rover localization performance between periods in which visual perception is unavailable or insufficiently performed. To provide a reliable localization framework, an error-state EKF was implemented. Due to the complex non-linear behaviors of the total state, dealing with the integration of the error state is less complicated than the total state. Also, since we

are using the pseudo-measurement updates to enhance the localization solution, estimating the error states directly and removing the accumulated modeling errors and process noise from the total state in the state correction step is more preferable. For this reason, the error state EKF is chosen instead of EKF. To estimate the error state, 6-DOF angular rate and specific force measurements from an IMU, measurement updates from rover wheel odometry (i.e., forward speed and yaw rate), as well as pseudo-measurement updates from zero velocity, zero angular rates, and non-holonomic constraints were used. The proposed approach was demonstrated to reduce the rover positioning error rate significantly when using all of the presented update strategies together. Besides, it was shown that slippage could be detected by using the slip ratio estimate. The primary value of this approach is that it can be used within current and future planetary rovers and many other wheeled robots to improve onboard localization performance without any hardware changes or major alterations in operations.

6

Proprioceptive Slip Detection

This chapter incorporates material from the following submission:

Kilic, C., Gu Y., and Gross J.N., “Proprioceptive Slip Detection for Planetary Rovers in Perceptually Degraded Extraterrestrial Environments”

6.1 Introduction

Slip detection is of fundamental importance for the safety and efficiency of rovers driving on the surface of extraterrestrial bodies. Current planetary rover slip detection systems rely on visual perception on the assumption that sufficient visual features can be acquired in the environment. However, visual-based methods are prone to suffer in perceptually degraded planetary environments with dominant low terrain features such as regolith, glacial terrain, salt-evaporites, as well as, poor lighting conditions such as dark caves and permanently shadowed regions. Relying only on visual sensors for slip detection also requires additional computational power and reduces the rover traversal rate. This chapter provides a wheel slip detection method without depending on visual perception. The method obtains its information from the proprioceptive localization framework described in Chapter 5 that is capable of providing reliable, continuous, and computationally efficient state estimation over hundreds of meters.

Acquiring accurate slip detection is one of the critical capabilities required for planetary rovers [84] to maintain safer driving conditions. Wheel slippage is often unavoidable for a planetary rover, it affects the traction and energy consumption and causes the significant drift from rover’s planned path and poor results in the rover state estimates [5]. Due to radiation-hardened hardware requirements, slip detection capability is challenging to achieve for the rovers with limited energy sources and computational power.

Mars rovers have been substantially benefited from stereo vision-based odometry to detect slip and

compute position updates whereas IMU provides the attitude solution[13]. Despite their reliability, vision-based systems operate with assuming that the terrain contains sufficient visual texture for localization. This assumption poses a challenge on extraterrestrial bodies that adequate visual features are lacking in the region (e.g., glacier ice, regolith, salt evaporites) [9] or when the lighting conditions are insufficient [85].

High-slip events and entrapment of the rovers are often experienced in terrains with unconsolidated sandy terrains [33, 25, 86, 24]. Also, the majority of the VO failures on Curiosity happened when the rover is at a sandy terrain with a few obvious unique features (66/94 by sol 2488) [13]. Hence, the rover operation usually needs to be altered to detect unique terrain features, such as moving mast cameras [10] to the point of interest or using wheel tracks left by the vehicle [73]. Degraded performance and unavailability of visual based systems are prone to increased localization drift on featureless terrains.

Apart from Martian exploration missions, the interest of the exploration of Europa has been increased [87]. Although the knowledge of Europa's surface is extremely limited, experiments for rover mobility purposes have been recently performed on salt-evaporites and icy terrains analogous to the Europa surface [88]. Similar to the visually imperceptible unconsolidated sands buried under the thin cover of basaltic sands on Mars [24], low-feature terrains on Europa are extremely challenging for vision-based systems and sparse features can degrade the perception performance leading to increased localization drift. Therefore, given that slip detection is only performed by VO-based methods in current rover operations[25, 13], the concern for providing continuous slip detection significantly increases when visual-based systems are unavailable.

Even this study is very much centered on the planetary rover literature, there has been plenty of work on VO that mainly focuses on urban navigation, including some very fast and accurate meth-

ods [89, 90]. These algorithms are designed for faster and more powerful computers than current RAD 750 processor and cannot directly be used in the planetary rovers due to intractable computational cost [7]; however, there are a few comparison analyses to evaluate their performance in challenging planetary analog datasets [91]. Moreover, a recent research from European Space Agency (ESA) in [92] reports that accumulated localization error at the end of trajectory for the state of the art VO algorithms are typically in a range of 1 to 2%, which often requires global pose correction techniques (e.g., map matching) to correct the drift. Also, it is observed that these similar concerns are quite visible in one of the state-of-the arts algorithm in [93] for ORB-SLAM3 performance at outdoor tests due to the poor-features in the environment. Note that the provided outdoor test environments in [93] have richer features than our Mars analog terrain.

6.2 Proprioceptive Slip Detection

Wheel slippage affects the traction and energy consumption, and causes the significant drift from rover’s planned path and poor results of localization [5]. Wheel slippage can occur when the terrain traversed fails [26] or when there is a kinematic incompatibility between wheels (i.e., different wheel speeds) encountered [27]. Having a continuous slip detection is a critical asset for planetary rovers, given that there are certain cases when visual-based systems suffer from low-feature environments. Apart from the exteroceptive slip detection in the current planetary rovers (i.e., Curiosity, Perseverance) that require camera outputs, this method utilizes the INS estimated velocity.

Skid steer vehicles rely on differing left and right wheel velocity directions to turn the vehicle. Due to redundant points of contact (i.e., two wheels are driven by the same drive-train on each side), slippage is often expected when turning motion is performed[94]. When a wheel slips, the asso-

ciated encoder will register a revolution, though this revolution does not correspond to any linear displacement of the wheel. However, if the wheel skids, no revolutions are registered by the encoder even though a skid results in a linear wheel displacement. Since the IMU measurements are independent of the wheel odometry, the motion estimates from the EKF can be compared to the computed velocity based on the vehicle kinematics to determine if any statistically significant slip (or skid) has occurred.

The slip ratio is a simple indicator to observe the slip, which can be formulated in different ways. This formulation depends on the sensors used for estimating wheel velocity and body velocity [27]. Observing the commanded distance and traversed distance, or commanded wheel velocity and actual velocity, can be given as examples. Wheel slip can also be monitored with the slip ratio calculation for wheel odometry velocity with respect to the INS estimated velocity. Slip ratio is usually distinguished as low, medium, and high slips. For example, defining the slip ratio as in Equation 6.1, upper and lower thresholds for these slip ratio classifications as low slip ($0 < |s| \leq 0.3$), medium slip ($0.3 < |s| \leq 0.6$), and high slip ($0.6 \leq |s|$) in [27, 44, 95]. It is reported that Curiosity rover traversal are forced to stop if a single significant slip measurement exceeds a threshold between 0.7 and 0.9 or if there is consecutive slip measurement between 0.4 and 0.7 based on the past experience with the terrain and testing [33]. Moreover, based on the terramechanic observations, 0.2 slip ratio is widely accepted as a significant slip threshold due to its effect on the drawbar pull. This effect can be observable as the rate of drawbar pull increasing with slip is higher between 0-0.2 slip range than 0.2 to 0.8 slip range[35]. This results in observing more sinkage in the 0.2 - 0.8 slip range.

Proprioceptive slippage can be monitored with the slip ratio calculation for front and rear wheels

velocity with respect to the INS estimated velocity. The slip ratio [96], $s \in [-1, 1]$, is defined as

$$s = \begin{cases} 1 - \frac{v_x}{r\omega} & (\text{if } \omega \neq 0, v_x < r\omega, s > 0) \\ \frac{r\omega}{v_x} - 1 & (\text{if } v_x \neq 0, v_x > r\omega, s < 0) \\ 0 & (\text{if } v_x = r\omega \vee v_x = \omega = 0) \end{cases} \quad (6.1)$$

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 in our method. If slip ratio is equal to zero, it means the rover does not encounter any wheel slippage. The motion estimates from the filter are compared to the computed velocity based on the vehicle kinematics to determine if any slippage has occurred.

The slippage is almost inevitable on rough, sloped, and fine soil terrains. Experiencing a long term significant slip may cause catastrophic results as in Spirit's mission; however, rovers can halt their driving before reaching a significant slippage point and correct its route to reliably arrive the end goal. For this reason, the rover should be aware when experiencing a slip, even without using a visual-based system, given that most of the excessive slippage events happened in highly deformable and not visually perceptible unconsolidated sands buried under the thin cover of basaltic sands[9, 24, 36, 37, 38, 39, 40, 13]. Considering the Eq.6.1, the proprioceptive slip detection accuracy depends on the quality of the velocity estimation from the filter. In an extreme case with a perfect INS state estimation (i.e., no INS drift), rover localization would not be affected by this slip. In fact, wheel slippage only affects the wheel odometry estimation considering the problem in the localization perspective. However, a dead-reckoning system is prone to drift in the real world scenarios. Using pseudo-measurement updates can improve the INS estimated velocity accuracy. Consequently, improved INS velocity estimation can be used as a complementary method to de-

test slippage by comparing it with wheel encoder based velocities when the camera information is not available.

6.2.1 Evaluation

To evaluate the proposed proprioceptive slip detection method, several comparison analyses are performed. First, the visual inertial odometry (VIO) estimates from the tracking system are statistically compared with the proposed method using Root Mean Square Error (RMSE) values of the 3D position estimates. Then, the slip detection accuracy and velocity estimations are examined with respect to the DGPS based velocity and slip detection. Finally, the heading estimation for WO, DGPS, and proposed method are compared.

In position estimation comparisons on a perceptually degraded terrain, rover is remotely controlled for approximately 150 m and the comparison results are given for five cases in Table 6.2.1 using Root Mean Square Error (RMSE) values of the 3D position estimates. The naming convention for the test cases are selected based on relative to the distance traversed and the rover velocity. For example, ShortFast stands for short distance (~ 150 m) with fast speed (~ 0.8 m/s), LongFast is long distance (> 500 m) with fast speed, and ShortSlow is short distance with slow speed (~ 0.3 m/s). ShortSlow case is used for two specific observations: 1) to observe the effect of having a slower velocity in the state estimation, 2) to observe the INS drift for a longer traversal time.

In areas with many visual features, the tracking system can produce dependable solutions, but it often fails in places with few detectable and trackable elements. This is a typical problem for visual-based localization methods since they require a decent number of distinct visual components to function properly. [93, 10]. For example, it is shown that the VO method working on Curiosity

Table 6.2.1: Statistical Position Error Comparison against VIO

Case	VIO RMSE (m)			Proposed RMSE (m)		
	East	North	Up	East	North	Up
ShortFast1	44.04	27.24	0.42	1.23	2.51	1.97
ShortFast2	2.70	10.41	2.12	1.03	0.49	0.65
ShortFast3	13.27	56.62	2.14	0.66	0.40	2.89
ShortFast4	1.41	12.22	1.59	2.26	1.76	0.38
ShortSlow	13.25	44.69	2.83	1.01	0.28	1.47

rover has a remarkable convergence rate in one of the most recent works [13]; however, in the same work, it can be seen that the majority of the VO failures on MSL are due to scarcity of sufficient features at sandy terrains. This supports the idea that using a complementary proprioceptive slip detection in addition to exteroceptive slip detection system can be useful in perceptually degraded environments.

Figure 6.2.1: Depiction of a comparison for position estimation accuracy of the proposed approach against Intel RealSense T265 VIO, and traditional dead-reckoning in a visually low-feature terrain. 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 3D position estimation between our approach against the tracking system visual-inertial odometry (VIO) solution is provided in Fig. 6.2.1. The purpose of this analysis is demonstrating some cases in perceptually degraded regions where the visual perception is not reliable. Since the proposed proprioceptive slip detection method accuracy relies on the state estimation quality, illustrating the position estimations are one of the good intuitive ways when VO suffers to provide accurate state information.

In our analysis in a low-feature environment, we observed the similar issues for using VO. For example, in ShortFast4 case, the VIO failed after 124 m of traversal which can be seen in Fig. 6.2.2(a). Moreover, even it can keep the East positioning in a reasonable accuracy for ShortFast2 and ShortFast4 case, the positioning estimations of the North axis are more than 10 m RMSE for all tests. On the other hand, the proposed method in this work often outperforms the VIO solution in this environment. Along with the previously discussed VO related issues, these results suggest that leveraging pseudo-measurements and kinematic constraints during the mission can keep the localization accuracy reliable where the visual based approaches suffer from lack of features.

Therefore, using a proprioceptive localization method can be used as a complementary localization technique to assist the main VO-based localization method in planetary missions. Then, the velocity estimation from this proprioceptive localization method can be leveraged as a secondary slip detection method when the visual sensor information is not available. Having a complementary method along with the main method can also assist to build a resilient localization system with a continuous slip detection method.

(a) ShortFast4

(b) ShortSlow

(c) ShortFast3

(d) ShortFast1

Figure 6.2.2: Ground track depictions of the proposed method, VIO estimation, WO with IMU heading estimation, and DGPS solution. The DGPS solution is treated as the truth ground track and given as a black line. VIO estimation is given as a green line, WO-based estimation given as cyan line, and the proposed estimation given as red line with blue circles (the blue circles are the detected significant slip, $s > |0.2|$). The Google Map representation of the truth solution is provided to visualize the environment (Point Marion, PA, Ashpiles Mars Analog Environment).

Table 6.2.2: Slip Detection Accuracy

Case	Distance (m)	Velocity (m/s)	Detection Failure	Number of slip (Truth)	Accuracy %
LongFast	659	0.8	1682	10283	83.64
ShortFast1	141	0.8	221	3048	92.75
ShortFast2	145	0.8	420	6483	93.52
ShortFast3	148	0.8	51	2278	97.76
ShortSlow	146	0.3	228	3219	92.92

Based on both previous literature and Curiosity rover thresholds for slip [33, 35], as well as our field test observations, the measured slip ratio greater than $|0.2|$ is defined as significant slip. Using this definition, a statistical validation of the significant slip detection performance is given in Table 6.2.2. The slip detection is performed each time when the wheel encoder data is available (10 Hz for both truth and estimated). In this table, the estimated slip detection accuracy is calculated using DGPS-based slip detection such that

$$s_{TR} = \begin{cases} 1 - \frac{v_{xTR}}{r\omega} & (\text{if } \omega \neq 0, v_{xTR} < r\omega, s > 0) \\ \frac{r\omega}{v_{xTR}} - 1 & (\text{if } v_{xTR} \neq 0, v_{xTR} > r\omega, s < 0) \\ 0 & (\text{if } v_{xTR} = r\omega \vee v_{xTR} = \omega = 0) \end{cases}, \quad s_{TR} \in [-1, 1] \quad (6.2)$$

where s_{TR} is DPGS-based slip ratio. The detection failure in Table 6.2.2 shows the total number of slip detection failure. This happens when a significant slip cannot be detected by the proposed method. Also, the number of slip values in the table are the significant slip values detected by DGPS (truth). In these test cases, the LongFast has the longest traversal distance (~ 650 m), which is an additional test case to show the slip detection accuracy limit for longer distances, whereas other test cases are between 140 m to 150 m. The estimated velocity is compared with the DGPS-based velocity solution to evaluate the performance of the proposed velocity estimation model and the overall distribution of the translational velocity errors in the tests are given in Fig. 6.2.3.

(a) ShortFast3

(b) ShortSlow

(c) LongFast

Figure 6.2.3: Left figures show the velocity estimation compared to post processed DGPS velocity solution that assumed as truth velocity. Right histogram figures are the velocity error distributions between the estimated and the truth velocities.

The observed accuracy increase for short distances confirms the reliability of the method up to 141 m distance for the used sensor setup in this field. However, for longer distances (e.g., more than 500 m), the accuracy decrease could be attributed to the limitation of using a dead-reckoning method without any external updates and also the quality of the IMU. The IMU used in this study is a relatively low-cost sensor compared to Northrop Grumman LN-200S IMU with fiber optic gyroscopes and solid state silicon MEMS accelometers used in MERs, Curiosity, and Perseverance. The error histograms in Fig. 6.2.3 can also be interpreted as the accuracy of the slip detection .

Overall, these cases support the reliability of the method for distances around 150 m in the test field for the used rover platform with the given sensor setup, and also reveal the limitation of the method for the longer distances without external update.

Given that the attitude of the robot is used to integrate the velocity vector in the INS mechanism, the attitude estimation accuracy is critically important to have a reliable translational velocity for the slip detection. In this respect, to further evaluate the method, the estimated heading with the proposed architecture, directly integrated heading estimation, and wheel encoder based heading are compared with the DGPS-based heading estimation in Fig. 7.2.1.

In this figure, the unwrapped heading comparisons show the error growth of unbounded estimations for three different driving time interval including 1000 s (long), 600 s (medium), and 300 s (short). After a relatively short drive, WO-based heading estimation accumulates significant heading error which is a well-known problem due to wheel slippage. Also considering the used rover is a skid-steer rover, any small maneuvers during traversal adds more error to the WO-based heading estimation. Using a direct integration from the IMU for heading estimation could be useful for short drives. This heading estimation is usually used to support the wheel odometry such that using

(a) LongFast

(b) ShortSlow

(c) ShortFast2

Figure 6.2.4: The unwrapped heading estimation comparisons. The heading estimation accuracy is critically important to have a reliable translational velocity for the slip detection. Proposed method overall accuracy at the end amount, LongFast: 8 deg, ShortSlow: 2 deg, ShortFast = 0.01 deg. Traditional direct integration estimation at the end amount, LongFast: 74 deg, ShortSlow: 14 deg, ShortFast: 17 deg.

the velocities from the wheel encoders and using the heading information from the IMU. However, when the rover drives longer time, the drift becomes more significant. This is also related to

the grade of the used IMU, such that given a higher grade IMU may provide better estimations for longer drives without any external update (e.g., sun sensors). On the other hand, the proposed method, with using pseudo-measurements, closely follows the truth (DGPS) heading. The overall accuracy at the end amount for LongFast test is 8 deg (Fig. 7.2.1(a)), for ShortSlow is 2 deg (Fig. 7.2.1(b)), and for ShortFast is 0.01 deg (Fig. 7.2.1(c)) for these test cases. The comparison shows that the proposed method outperforms the traditional heading estimation techniques, which can be leveraged to estimate heading for short and medium driving duration reliably; however, it requires external updates after a long driving time.

6.3 Key Takeaways

This chapter offers the contribution of detailing a proprioceptive slip detection technique that leverages the accurate velocity estimation from the localization framework detailed in Chapter 5. Current planetary rovers depend upon sufficiently detected and tracked features for exteroceptive slip detection. The proposed method does not depend on the visual characteristics of the environment and also only uses the proprioceptive rover sensors already onboard. Therefore, it can be used as a complementary slip estimation technique when the visual sensor information is unavailable for the current and the upcoming planetary rover missions. The effectiveness of the proposed method is demonstrated with field tests in a perceptually degraded planetary-analog environment by qualitatively comparing with commercially off the shelf VIO solution, wheel encoder based velocity estimation, and DGPS velocity solutions.

7

Slip-Based Autonomous Zero Velocity

Update

This chapter incorporates material from the following publications:

Kilic, C., Ohi, N., Gu, Y. and Gross, J.N., 2021. “Slip-Based Autonomous ZUPT through Gaussian Process to Improve Planetary Rover Localization”. *IEEE Robotics and Automation Letters (RA-L)*, 6(3), pp.4782-4789.

Kilic, C., Gross, J.N., Ohi, N., Watson, R., Strader, J., Swiger, T., Harper, S. and Gu, Y., 2019, November. “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)* pp. 552-559.

7.1 Introduction

When stationary conditions are sufficed, the ZUPT and ZARU algorithms offer valuable information to maintain the INS-based state estimation reliability. Employing these algorithms along with leveraging non-holonomic constraints can greatly benefit wheeled mobile robot dead-reckoning localization accuracy as previously discussed in Chapter 5. However, determining how often they should be employed requires consideration to balance localization accuracy and traversal rate for planetary rovers. To address this, this section investigates when to autonomously initiate stops to improve wheel-inertial odometry (WIO) localization performance. This is achieved by a 3D dead-reckoning approach that measures the wheel slippage while the rover is in motion and forecasts the appropriate time to stop without changing rover hardware. A depiction of the algorithm workflow is shown in Fig. 7.1.1.

This wheeled robot localization framework consists of a series of actions that take place in real-time and in a future prediction. The real-time portion consists of the work in Chapter 5, which is an INS mechanization method aided by wheel odometry, pseudo-measurements, and kinematic constraints in an error state EKF. In this section, the future prediction part of the framework is the

Figure 7.1.1: The architecture for actively predicting the localization error boundaries. CoreNav* is the the real-time portion of this work that is described in Chapter5 and in [1]. The real time portion measures the slip ratio and estimates the state in an error state EKF, (CoreNav*), consists of INS, ZUPT, ZARU, WO, and non-holonomic updates. When the rover actively initiated the navigation stops, CoreNav applies the ZUPT and ZARU to reduce localization error growth. The future prediction part uses a time series Gaussian process with the the measured slip input and utilizes the current state estimation and an error observer to predict the future localization error. When the predicted localization error exceeds a predetermined threshold in a future time, the rover computer generates a stopping command to be activated at that specific time.

main focus. This prediction part uses the measured slip events and prior estimated error state information to predict the localization error of the rover. In this study, the Gaussian process (GP) is utilized to predict a time series model of the detected wheel slippage. The primary reason for choosing the GP is to leverage its prediction of uncertainty estimates, which are used later for predicting the error-covariance of odometry measurements. The kernel function selection, slip estimation with time series modeling, WO velocity prediction, and forecasting the localization error are detailed in the following sections. After explaining the method, initial tests to verify the algorithm, validation with field tests, and comparison analyses are provided.

7.2 Kernel Function Selection

A kernel describes the covariance of the GP variables, it is also called as a covariance function. Kernel encodes the similarity between the outputs in GP [67]. As previously discussed in Section 4.3, a GP is uniquely defined by its mean function $\mu(x)$ and a kernel $k(x, x')$ [65].

$$f(x) \sim GP(\mu(x), k(x, x')) \quad (7.1)$$

In order to properly identify a suitable kernel function to model the slip characteristics, it is assumed that slip could happen mainly in two ways based on the field test observations: 1) Randomly due to traversing over a rock, entering a pothole, or small local terrain imperfections, 2) continuously due to the mechanical properties of deformable terrains. To capture the different characteristics of the training dataset, we opt for combining at least two kernels and assessed different combination of the kernels such as, Matern kernels (Matern $5/2$ and Matern $3/2$), RBF kernel, linear kernel, constant kernel, and Brownian kernel. Some of the kernel selection compar-

ison analyses results are given in Fig. 7.2.2. These analyses are performed by using GPy[69] and scikit-learn[97] GP libraries. After a careful consideration through mathematical expressions of the kernels, field tests, and simulation analyses for the evaluation of these kernels, we adopted the Brownian kernel, $k_B = \sigma^2 \min(x, x')$, to characterize the unexpected kinematic incompatibility induced slip, and RBF kernel, $k_{RBF} = \sigma^2 \exp(-\|x - x'\|^2 / 2\ell^2)$, to model similar continuous slip values in the GP. Consequently, a composite kernel with multiplying these two kernels together, $k(x, x') = k_B(x, x')k_{RBF}(x, x')$, is used to capture both slip behaviors during traversing. An example demonstration of using this kernel against a collected slip data is given in Fig. 7.2.3.

7.3 Slip Estimation with Time Series Modeling

A time series is indexed data points ordered in a sequence. In a time series analysis, the aim is extracting characteristics of the data that can be used for forecasting based on previously observed values. In our case, forecasting refers to estimating the future values of slip. To model the time series of the wheel slippage, GP is utilized. The primary reason for choosing the GP is to leverage its prediction of uncertainty estimates, which are later used for predicting the error-covariance of wheel odometry velocity measurements detailed in Section 7.4.

Given a training set of input and output pairs $(\mathbf{x}, \mathbf{y}) = (x_i, y_i)_{i=1}^N$, 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 given by:

$$p(\mathbf{y}_* | \mathbf{x}_*, \mathbf{x}, \mathbf{y}) = \mathcal{N}(\mathbf{y}_* | \boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*) \quad (7.2)$$

(a) Brownian kernel

(b) Radial Basis Function kernel

(c) Composite Brownian and RBF Kernel

Figure 7.2.1: This figure shows a Brownian, RBF, and a composite kernel for five different random seeds. The composite kernel captures the characteristics of both kernels that can be used to model slip behavior of our test platform.

Figure 7.2.2: Kernel analysis for different composite kernels. In this analysis, different combination of the Matern 5/2 (MA5), Matern 3/2 (MA3), RBF kernel, linear kernel, constant kernel, and Brownian kernel are shown against the slip data.

Figure 7.2.3: A composite kernel with multiplying Brownian and RBF kernels together is used to capture both slip behaviors during traversing.

where

$$\boldsymbol{\mu}_* = \mathbf{K}_*^T \mathbf{K}_*^{-1} \mathbf{y}, \quad \mathbf{K}_* = \mathbf{K}(\mathbf{x}, \mathbf{x}_*) \quad (7.3)$$

$$\boldsymbol{\Sigma}_* = \mathbf{K}_{**} - \mathbf{K}_*^T \mathbf{K}_*^{-1} \mathbf{K}_*, \quad \mathbf{K}_{**} = \mathbf{K}(\mathbf{x}_*, \mathbf{x}_*). \quad (7.4)$$

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 N 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

$$s = f(t) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2). \quad (7.5)$$

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) [2]. During this 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. 7.3.1(b). A python GP library [69] is used in our rover's ROS framework to optimize the hyperparameters (e.g., the length parameter ℓ), and to predict the slip values while the rover is in motion.

Figure 7.3.1: An example scenario that starts from slip detection and goes through each elements in the framework. The sub-figure (a) shows the filter estimated and WO estimated velocity to be used for slip detection as described in section 6.2.1. The sub-figure (b) shows the slip input and slip prediction in the GP time series as described in section 7.3. The sub-figure (c) is a depiction of unscented transform that used for mapping the WO velocities for error prediction as described in section 7.4. Finally, the sub-figure (d) shows how the predicted error is generated as described in section 7.5. The input slip data is collected within a time window $([T_0, T_1^-])$ 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.

7.4 Wheel Odometry Velocity Prediction

To predict the simulated odometry velocity error boundaries, a statistical sigma point transformation inspired by unscented transformation [98] where the slip ratio definition in (6.2) is used to generate this transformation function:

$$\sigma_{\chi_{est}(t)} = \frac{1}{N} \sum_{i=1}^N \left(\chi_{i(t)} - \mu_{\chi_{est}(t)} \right)^2, \quad t \in [T_1^+, T_3] \quad (7.6)$$

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. 7.3.1(c)), N is the number of the sigma points, χ_i is velocity term mapped from slip measurement, defined as

$$\chi_1 = \mu_{vel} / (1 - \mu_s) \quad (7.7)$$

$$\chi_2 = \mu_{vel} / (1 - \mu_s - \sigma_s) \quad (7.8)$$

$$\chi_3 = \mu_{vel} / (1 - \mu_s + \sigma_s) \quad (7.9)$$

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.

$$\mathbf{R}^{GP} = \begin{bmatrix} (\sigma_{\chi_{est}}^2 \mathbf{I}_{3 \times 3}) & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}_{4 \times 4}. \quad (7.10)$$

These mapped velocity values and their prediction with this statistical sigma point transformation method are depicted in Fig. 7.3.1(c).

7.5 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.

$$\hat{\mathbf{P}}_0^{GP} = \mathbf{P}_{T_1^-} \quad (7.11)$$

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. 7.3.1).

$$\check{\mathbf{P}}_k^{GP} = \mathbf{F}_{T_1^-} \hat{\mathbf{P}}_{k-1}^{GP} \mathbf{F}_{T_1^-}^T + \mathbf{Q}_{T_1^-} \quad (7.12)$$

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). The EKF implementation is designed to be able to process asynchronous measurement updates in a sequential manner as they arrive. That is, each prediction or update outputs the estimated error state and error covariance. Therefore, the prediction from IMU is processed five times more frequently than the wheel odometry update. 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.

$$\mathbf{K}_k^{GP} = \check{\mathbf{P}}_k^{GP} \mathbf{H}_{\mathbf{T}_1^-}^T \left(\mathbf{H}_{\mathbf{T}_1^-} \check{\mathbf{P}}_k^{GP} \mathbf{H}_{\mathbf{T}_1^-}^T + \mathbf{R}^{GP} \right)^{-1} \quad (7.13)$$

$$\hat{\mathbf{P}}_k^{GP} = \left(1 - \mathbf{K}_k^{GP} \mathbf{H}_{\mathbf{T}_1^-} \right) \check{\mathbf{P}}_k^{GP} \quad (7.14)$$

For each updated covariance prediction, the algorithm calculates the position error covariances as a function of time. An example calculation is illustrated in Fig. 7.3.1(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. 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 Chapter 5 and Appendix A.2.

In order to enable zero updates, the rover simply require a stationary condition. Although the rover is often stationary, the navigation solution may require extra stops while driving. This can be achieved by sequential stops such as stopping after driving a predetermined interval, or actively initiating the navigation stops by monitoring the estimated localization uncertainty. Sequential stops may correct most of the drifts while driving as shown in Chapter 5; however, observing the position or velocity uncertainty will give a more intuitive answer where a high level position correction is required. While driving, the variance of the position, x , after a period of t seconds can be given

by

$$\text{Var}[\hat{x}^{(1)}] = \text{Var} \left[x + \int_0^t \int_0^t \epsilon(\tau) d\tau d\tau \right] \quad (7.15)$$

$$= T_s^4 \sum_{i=1}^n (n - i + 1)^2 \text{Var} [X_i] \quad (7.16)$$

where $\epsilon(\tau)$ is the white noise signal. If the sampling frequency, $1/T_s$, is assumed to be large and $\sum_{i=1}^n (n - i + 1)^2$ is equivalent to the sum of the first n positive squared integers, the variance of the position can be approximated as

$$\text{Var}[\hat{x}^{(1)}] \approx \sigma^2 \frac{T_s t^3}{3} \quad (7.17)$$

Therefore, the noise introduced to the signal is second order that grows with time with zero mean and standard deviation given by

$$\sigma_x(t) = \sigma t^{3/2} \sqrt{\frac{T_s}{3}} \quad (7.18)$$

where $\sigma_x^{(1)}(t)$ is the standard deviation at time t of the position obtained by integrating a single accelerometer. Denoting P_{lat} , P_{lon} , P_h (i.e., P_{77} , P_{88} , P_{99}) as the diagonal position elements of covariance of state vector estimate matrix, $\mathbf{P}$, the standard deviation of the measured uncertainty of the error estimates can be calculated. Since it is usually more meaningful to observe the errors in East North Up (ENU) coordinate frame instead of locally-level navigation frame (latitude, longitude, height), from the transformed covariance matrix, the errors can be defined as

$$\sigma_E = \sqrt{P_{east}} \quad \sigma_N = \sqrt{P_{north}} \quad \sigma_U = \sqrt{P_{up}} \quad (7.19)$$

and the horizontal error is

$$\sigma_{horizontal} = \sqrt{\begin{bmatrix} \sigma_E \\ \sigma_N \end{bmatrix}^T \begin{bmatrix} \sigma_E \\ \sigma_N \end{bmatrix}} \quad (7.20)$$

Since the expected value of the position after a period of t seconds is x and the corresponding estimated error grows with time, a certain threshold to initiate stops may be used. The threshold can be selected with respect to the predetermined blind-driving distance error limit. For example, MERs mission design criteria for position error was set as it must be less than 10%, (i.e., 10 m error for 100 m driving) [73, 99]. Similar to this, in order to keep the localization solution reliable, we used error thresholds as the standard deviation of the measured uncertainty of the horizontal error in a range of $2 \leq \sigma_{horizontal} \leq 5$ m to decide when to apply zero-update. Different thresholds are set prior to each run in the Martian analog field tests since our testbed rover was teleoperated with a remote controller during traversal and did not have fixed traversal distance to drive from point A to B. The other reason of using different thresholds was testing the operational limits of the algorithm against longer distances. Thanks to flexibility of the algorithm design, horizontal error threshold usage can be extended to percentage-based threshold assessment with respect to the given fixed distances, e.g., given a 200 m distance to blind-drive, use 1% error threshold.

7.6 Summary of the Operational Process

This section gives a summary of the proposed algorithm's steps and high-level information of how these steps connected to each other. The process can be divided into two sections: 1) the real-time filter estimations, and 2) future GP predictions. Real-time part is the method described in Chapter 5, where the filter estimated error is error state EKF provided estimations. The algorithm

also collects the detected slippages by considering the WO and filter estimated velocity for 15 s intervals. Error prediction time limit is set to 60 s with a predetermined error threshold value. In the case where the predicted localization error does not exceed a threshold, the rover continues driving. Otherwise, the algorithm sets an internal countdown for stopping at the point when the error prediction exceeds the threshold. A demonstration of these actions with a detailed example scenario is given in Fig. 7.6.1. The time limits and intervals are decided based on the blind driving operating limits for MSL to keep the current planetary rover operations as close as possible to our tests. These operation limits are given in Section 7.8.

7.7 Field Tests

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. The preliminary tests to verify the algorithm are performed in paved (asphalt road), unpaved, and gravel terrains. Overall, more than 70 short distance (less than 150 m) tests used to verify the algorithm, and after that approximately 30 tests used for the validation. The details of these preliminary tests are given in Appendix A.3 and an example test on a paved environment is given in Fig. 7.7.1.

The validation part of the field tests particularly performed in a planetary analog terrain results because of its similarities with the Martian terrain (see Fig. 7.7.2). The selected terrain for the field

Figure 7.6.1: 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 slip check interval (10m) [2]. 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 [3], 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.

Figure 7.7.1: Pathfinder during one of the paved road tests. Post processed PPP Kinematic GPS solution from RTKLib software is shown as red line, the filter estimation is shown as white line. Traversed distance 488 m, horizontal error during driving is less than 0.5%, maximum error 2.1 m. The paved road tests are mainly performed to check the software reliability.

tests 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 similar to the abundant chemical compounds found in Martian regolith [100]. The common composition of coal ash includes substantial amounts of aluminum oxide (Al_2O_3), iron oxide (Fe_2O_3), silicon dioxide (SiO_2), and calcium oxide (CaO) [101], similar to the abundant chemical compounds found in Martian regolith¹ [100, 102]. For this reason, visual (see Fig. 7.7.2) and chemical characteristics of this test field are considered sufficient for a planetary analog environment.

¹Although the composition percent by weight of the soil is the key part to creating a Mars simulant, the measurement of the chemical properties composition by weight is beyond of the scope of our study.

Figure 7.7.2: Pathfinder during field tests in an environment with similar visual and chemical characteristics found in Martian terrains.

In order to evaluate the proposed algorithm in field tests, we adopted the following blind driving operating limits:

- The blind drive is usually commanded on the basis that the path in the immediate front of the rover can be seen very clearly in images and is considered safe. For this reason, Pathfinder is driven on obstacle free regions or by teleoperating to avoid obstacles.
- Since the blind driving is a dead-reckoning technique (which causes the uncertainty of the state of the rover to increase with distance), this technique can only be used over short distances in most situations. The length is limited to the distance chosen by the rover planners, which is based on the rover camera visibility range, prior to employing this mode. We tested

our localization algorithm with a maximum 670 m length of blind driving to see its operational limits. Rover safety is more important than accuracy for planetary missions, therefore, these lengths of drivings without human-in-the-loop process may not be suitable for planetary rovers in a manner of rover safety.

- Since an unforeseeable slippage related wheel sinkage can terminate a mission (e.g., Spirit mission), the Mars rovers usually avoid slopes more than 25 degrees on undeformable terrains (bedrock), sandy slopes more than 12.5 degrees, and relatively bright terrains (most likely dominated by soft/fine soils). Our tests have maximum 29.8% degrees slope.

The time limits and interval parameters for the algorithm are chosen based on the current planetary rover operations as best as we can and engineering judgments during the experiments. To elaborate more, Mars Rovers are using VO update in every 10 meters in Hazard Avoidance Mode [2]. This can be inferred as the rover checks slippage after every 10 meters of traversal. In other words, traversed terrain can be assumed as not changing for 10 meters. Given this intuition and our rover's average speed of 0.8 m/s, we decided to train the GP with the last 12 m of slip data. The extra 2.5 sec (which corresponds to 2 m extra traversal) is included to compensate for the possible slippage. The Mars rovers blind-driving mode is often limited to a maximum of 50-60 m drive if the terrain is safe to traverse. Based on this and the fact that GP time series prediction becomes less reliable over time, we decided to predict a maximum of 60 s slippage. These values can be changed with respect to the wheel velocity, and the reliability of the GP time series prediction algorithm. However, we used these specific times based on the current rover operations in planetary missions assuming that the real Mars driving operations are conservative enough due to safety reasons. Obviously, shortening the GP training time will affect the prediction reliability, therefore limiting the prediction duration. A systematic way to determine these parameters would be a subject of future research (e.g., adding

another layer in the framework to select these GP training and prediction time parameters based on the rover wheel model, slip measurement resolution, wheel encoder resolution).

Given that planetary rovers have limited onboard computation, one can argue that using a GP is usually computationally expensive. Using GP for blind driving will increase the computation expense; however, usage of GP in the planetary missions has been widely proposed in many papers, as it is referred to in Chapter 2. Additionally, the GP that is used in this work is a lightweight architecture because it uses one input and one output, which makes it faster than multi-input-output structures. After 15 s of the drive, each process for prediction takes less than a second with Intel Core i7-8650U CPU (Intel NUC Board NUC7i7DN) without extensive code optimization. Nevertheless, we acknowledge that the capability of RAD750 CPU on the current Mars rovers will be less capable than this.

7.8 Results and Discussion

To demonstrate relation between the stopping time and the wheel slip, a stopping time comparison analysis against four terrain types is shown in Fig. 7.8.1. In this comparison figure, Delta Time refers to the remaining time to stop after 15 s of data collection for each terrain type. For example, Delta Time median value of benign road is 35 s, and rough road is 10 s. This means the rover needs to stop after 50 s driving for the benign road, and after 25 s driving for the rough road. The data collection is fixed to 15 s regardless of the terrain type with the assumption of the type of the terrain traversed does not change for each 12 meters of driving based on the MSL Hazard Avoidance slip check interval (~ 10 m) [2]. The measured slip values are the decisive factors to identify the driving duration between each stops.

Figure 7.8.1: Comparison of stop time interval for terrain types. Δ 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 ($p<0.0001$).

In the analysis given in Fig. 7.8.1, 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. Although stopping too often on slippery terrains can be a disadvantage for rover mobility, this is a common mobility limitation for rovers without using an actual traction con-

trol mechanism. Unfortunately, neither Curiosity nor Perseverance rovers (and also our testbed rovers) include force sensors on the mobility subsystem for this actual traction control. However, the Traction Control (TRCTL) software patch has been used on Curiosity since sol 1678 and used as a traction approximation [25]. Therefore, this limitation can be alleviated by leveraging TRCTL with this method. The planetary rovers are extremely slow (~ 4 cm/s max speed), and the rover operations have already forced the rover stop on slippery terrains [13, 25]. This slippage is primarily due to sinking of the wheels. When a rover starts slipping on these loose terrains, it also starts sinking [27]. Due to this, for the sake of safety of the rover, the stopping and changing the rover's navigational plan would be the best option to avoid sinking based on the observations from previous rovers (e.g., MERs and MSL rovers). In these high slippage cases, the localization accuracy and driving long distances would become secondary issues, and the rover operators will focus on rover safety. The slippery terrains in our field tests are mostly the terrains that would not lead the rover to be immobile, which blind-driving can be used safely. Also, in order to be consistent with the current rover operations and not stop too often on these slippery terrains in our field tests, the minimum frequency of stop is set to 12 m (15 s) given that our rover's average forward speed is 0.8 m/s. This is the free driving part of the algorithm where we collect the data.

To further evaluate the method, the localization accuracy of the proposed estimation is compared against the DGPS solution which is assumed as the truth approximation. As detailed in Table 7.8.1, 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 the long tests, Test1 has a 5 m threshold whereas Test2 and Test3 have a 3 m threshold. Therefore, if all the long tests are assumed

as approximately the same lengths, Test1 should have much more error than Test2 and Test3 since it has a bigger error threshold before triggering a ZUPT. This also can be seen that the Test1 stop count is less than the others, and consequently, the error progression is expected to be more significant than the other tests. It also can be seen that Test2 and Test3 almost have the same stop counts and similar errors, which are less than Test1. Moreover, since the error will continue to grow over time, providing only short drive tests with short distances will consequently produce less localization error. Instead, we drive the rover over long time and show the worst and most interesting scenarios. For example, long-distance tests (650 m) can be given as divided short-distance tests instead of one long test. In these field test results, the algorithm still 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 Test1 due to not exceeding the threshold in the prediction time limit. Since this is a dead-reckoning technique, the localization accuracy will degrade over time and the error progression of this error is typically non-linear. Obviously, using an autonomous blind-driving without any external positioning update for these long distances would not be recommended for planetary missions for safety reasons. These long distances are selected to observe the possible limitations of the algorithm and relatively ordered with each other in this work. In fact, most of the short-range tests (~ 150 m) in our tests are longer than the Curiosity rover's longest drive (142.5 m) by sol 2488 [13].

Ground-track depiction of an example scenario from ash-pile field testing is given in Fig. 7.8.2. 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. The localization design goal for MER was to maintain a position estimate that drifted less than 10%

Table 7.8.1: 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	STD	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.

during a 100 m drive [30]. Without using ZUPT and kinematic constraints in blind-driving, the drift can quickly elevate and exceed that design limit, as shown in Fig. 7.8.2.

Improving the localization accuracy with periodic stopping was detailed in Chapter 5. In this chapter's work, the main aim is keeping the localization accuracy as similar as possible to the prior approach and autonomously decide to the traversal rate based on detected slippage frequency and magnitude. In this respect, to quantitatively compare these two methods such as the time gained by stopping less with this method, as well as the difference of accuracy, a comparison analysis between autonomous and periodic stopping methods is provided in Table 7.8.2. Comparison of the findings with autonomous and periodic stopping algorithm confirms that 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, autonomous stopping increases the traversal rate by stopping less, and keeps the localization accuracy to an acceptable level.

Figure 7.8.2: Ash-Pile Test3 results against traditional 2D Dead Reckoning.

7.8.1 Key Takeaways

In the extreme case, while only using inertial navigation without wheel odometry, zero-updates with periodic stops can be employed to reduce the error-growth on the position states. On the other extreme, when there are near-perfect wheel odometry measurements available (i.e., no wheel slipping) the velocity information can be combined with IMU to have a reliable navigation source that works while the rover is moving. Since assuming no wheel slippage is impractical for planetary rover applications and stopping too often is not ideal for missions that require fast traversal, in this chapter, the main focus is balancing the rate of traversal (i.e., not requiring periodically scheduled navigation stops) and keeping the localization error-growth rate bounded to an acceptable level. In this chapter, a slip-based localization error prediction framework is presented, which effectively balances the traversal-rate and localization accuracy for wheeled planetary rovers. Instead of peri-

Table 7.8.2: 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 7.8.1, S_R : Stop rate, ΣStop : Stop count

odic 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. The results show that planetary robot slip related localization drift can be alleviated with ZUPTs and can provide reliable proprioceptive localization performance for longer distances. It is obvious that using an autonomous blind-driving without any positioning update for long distances would not be recommended for planetary missions for safety reasons. The long distances in the field tests are selected to observe the possible limitations of the algorithm. In fact, most of the short-range tests (150 m) in our tests are longer than the Curiosity rover's longest drive (142.5 m) and provide approximately 1% of 3D localization error. In this respect, this work for blind-driving with pseudo-measurement updates does not underestimate the effectiveness of high-level localization solutions and current advancements for planetary autonomous navigation approaches. Instead, it shows that a complimentary proprioceptive method can be used along with the current methods. Also, we acknowledge that there is a fundamental limit that zero-updates can only slow the rate of localization error growth. After that threshold, the higher-level positioning updates (whenever it is possible to process) should take over and correct the rover's course.

8

Conclusion

8.0.1 Discussion Summary

Without the capability of providing a reliable proprioceptive dead reckoning, the robot can be lost and required extensive recovery actions in perceptually degraded situations, e.g., completely dark, bright, uniform, or foggy scenes. The proprioceptive sensors are often used as an aiding sensor for localization. However, the challenge of providing a reliable and continuous localization estimation in the situations when the unavailability of exteroceptive sensors still remains. Due to the drifting curse of using inertial localization techniques, the observations from the proprioceptive sensors are often overlooked and not utilized to their full potential. In this dissertation, we investigated the performance limit of onboard proprioceptive sensors to deal with some of these localization problems.

This dissertation provides several methods to improve the blind-driving based localization by leveraging the motion constraints while traversing and pseudo-measurements generated from proprioceptive sensors when the rover is stationary. These methods provide reliable and continuous localization using only proprioceptive sensors without any significant changes to the rover operations. Besides, these localization methods could play an essential role in providing computationally efficient real-time rover localization since the used sensors are available most of the time during the mission. While the main area of interest is planetary rovers, the contributions of this thesis can be benefited by any wheeled robotic system where the objective is improving localization accuracy.

8.0.2 Contributions

The work in this study leads several contributions: 1) It is demonstrated that leveraging pseudo-measurements and motion constraints can significantly reduce the rate of unbounded inertial navigation error growth due to continually integrating acceleration and gyro rate with respect to time.

In this respect, providing a proprioceptive approach for wheeled rovers that compensates for the high likelihood of inertial sensor-based navigational errors can reduce the usage of computationally expensive positioning corrections. 2) It is shown that wheel slip can be detected as velocity discrepancies between wheel odometry measurements and developed INS solution. Even the wheel slippage can be detected by VO estimations, in the places where visual perception is unavailable, the proprioceptive slip detection can play a significant role for rover safety and provide a continuous localization estimation. 3) A method is developed for predicting localization error, using a time-series Gaussian process model for slip uncertainty prediction as a function of time, such that stationary conditions can be actively initiated with respect to the wheel slippage frequency and magnitude. 4) A lightweight testbed mobile platform, called Pathfinder, is further improved with numerous software and hardware design iterations to excel in reliably collecting and processing IMU, camera, GNSS, and wheel encoder outputs in various environments. Thanks to its compact form factor, this testbed rover is easy to carry and can get ready in minutes to conduct field tests. Also, developed methods are tested on actual hardware in a planetary-analog environment. Detailed descriptions of our implementation and hardware specifications are provided, so the reader can more easily replicate the work. 5) The developed software packages, which are designed for real-time usage under ROS[22], and the post-processing scripts are made publicly available. 6) During verification and validation of the methods, IMU, wheel odometry, GNSS, and camera outputs are collected in more than 120 field experiments, including test cases in a planetary-analog environment. The majority of the collected dataset has been made publicly available to enable future research.

8.0.3 Future Work

The proposed methods are not without limitations, and in this regard, we discuss potential improvements and future work.

Pathfinder is utilized with deformable 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. Also using slick wheels limit to test the algorithm on unconsolidated soils with high slope values (more than 30 degree). These wheels are non-representative wheels for planetary missions because maximizing wheel traction is one of the most important design criteria for the rovers. In this respect, using the algorithms with better representative wheels such as mesh-woven spring wheels or aluminum made wheel design choices would expose further limitations and would be a good additional test.

The drift in the heading estimation is one of the main concerns for navigational purposes. Our method provides quality heading estimations and it is extendable to use the information from sensing the Sun. Although this may not be useful in the permanently shadowed regions or dark environments for future lunar mission, the current Mars rovers utilize the sun sensor information to correct their attitude. Therefore, leveraging this information along with pseudo-measurement updates can further improve the attitude of the rover.

In order to identify a suitable compound kernel function, we assessed several different compound kernels and validate the selection of the kernel through mathematical expressions of the kernels, field tests, and simulation analyses. After identifying the kernel function, the GP uses the same kernel for all the terrains. The kernel function decision for our tests is dictated by the measured slip on the terrains, the rover physical characteristics, and the terrain-wheel interactions. This usually

enforces the rover to become more strict when it faces a high slippage even in relatively benign roads. A more intuitive approach can be used with an adaptive kernel framework by learning the kernel from the slip measurements during traversal instead of using one compound kernel. This can be achieved by similar to our approach with adding one more layer to the framework such that rover may process the slip data during traversal to learn a kernel and use this new kernel to forecast the localization error.

During initial data collection process, we observed that the gyro rate measurements from the IMU sensor behave differently on different terrain types. These measurements are also compared with generated motor torque values and an observable relation is assessed. However, due to low resolution of the wheel motor torques, and the absence of force sensors, this relation were not be able to used for assessing a viable information in the framework. To enable a potential research direction, the performed analyses are given in Appendix A.3 as a starting point. Torque values can be used as a bridge between wheel and terrain interactions as well as identifying the slip ratio which can be used as an additional information in the proposed framework.

Using camera information may not be useful when there is not sufficient or unique features in the environment. However, using cameras to observe wheel sinkage can still be useful to identify the sinkage related wheel slip values such as mounting cameras to see the wheel and terrain interactions directly. One important observation is that the rigid wheel radius is assumed as constant; however, the deformable wheel has a changing diameter due to its interaction with the terrain. For instance, Curiosity consistently drives 1.65 meters with one full rigid wheel rotation on terrain with no slippage. This distance would decrease if the wheels were deformable. If the terrain is deformable, the sinkage related wheel slip can be estimated by determining the entry angle using the camera output that can see the interaction between wheel and terrain.

References

- [1] Cagri Kilic, Jason N Gross, Nicholas Ohi, Ryan Watson, Jared Strader, Thomas Swiger, Scott Harper, and Yu Gu. 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)*, pages 552–559, Nov 2019.
- [2] John P Grotzinger, Joy Crisp, Ashwin R Vasavada, Robert C Anderson, Charles J Baker, Robert Barry, David F Blake, Pamela Conrad, Kenneth S Edgett, Bobak Ferdowski, et al. Mars science laboratory mission and science investigation. *Space science reviews*, 170(1-4):5–56, 2012.
- [3] Emily Lakdawalla. *The Design and Engineering of Curiosity: How the Mars Rover Performs Its Job*. Springer, 2018.
- [4] Rongxing Li, Brent A Archinal, Raymond E Arvidson, Jim Bell, Philip Christensen, Larry Crumpler, David J Des Marais, Kaichang Di, Tom Duxbury, Matt Golombek, et al. Spirit rover localization and topographic mapping at the landing site of gusev crater, mars. *Journal of Geophysical Research: Planets*, 111(E2), 2006.
- [5] Giulio Reina, Genya Ishigami, Keiji Nagatani, and Kazuya Yoshida. Odometry correction using visual slip angle estimation for planetary exploration rovers. *Advanced Robotics*, 24(3):359–385, 2010.
- [6] Eric Klein, Erik Nilsen, Austin Nicholas, Charles Whetsel, Joseph Parrish, Richard Mattingly, and Lisa May. The mobile mav concept for mars sample return. In *Aerospace Conference, 2014 IEEE*, pages 1–9. IEEE, 2014.
- [7] Kyohei Otsu, Guillaume Matheron, Sourish Ghosh, Olivier Toupet, and Masahiro Ono. Fast approximate clearance evaluation for rovers with articulated suspension systems. *Journal of Field Robotics*, 2019.

- [8] Tom Duckett and Ulrich Nehmzow. Mobile robot self-localisation and measurement of performance in middle-scale environments. *Robotics and Autonomous Systems*, 24(1-2):57–69, 1998.
- [9] Ramon Gonzalez and Karl Iagnemma. Slippage estimation and compensation for planetary exploration rovers. state of the art and future challenges. *Journal of Field Robotics*, 35(4):564–577, 2018.
- [10] Jared Strader, Kyohei Otsu, and Ali-akbar Agha-mohammadi. Perception-aware autonomous mast motion planning for planetary exploration rovers. *Journal of Field Robotics*, 37(5):812–829, 2020.
- [11] 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.
- [12] Yang Cheng, Mark W Maimone, and Larry Matthies. Visual odometry on the mars exploration rovers. *IEEE Robotics and Automation magazine*, 13(2):54, 2006.
- [13] Arturo Rankin, Mark Maimone, Jeffrey Biesiadecki, Nikunj Patel, Dan Levine, and Olivier Toupet. Driving curiosity: Mars rover mobility trends during the first seven years. In *2020 IEEE Aerospace Conference*, pages 1–19, 2020.
- [14] Jeffrey J Biesiadecki, Eric T Baumgartner, Robert G Bonitz, B Cooper, Frank R Hartman, P Chris Leger, Mark W Maimone, Scott A Maxwell, Ashitey Trebi-Ollennu, Edward W Tunstel, et al. Mars exploration rover surface operations: Driving opportunity at meridiani planum. *IEEE robotics & automation magazine*, 13(2):63–71, 2006.
- [15] Olivier Toupet, Tyler Del Sesto, Masahiro Ono, Steven Myint, Joshua vander Hook, and Michael McHenry. A ros-based simulator for testing the enhanced autonomous navigation of the mars 2020 rover. In *2020 IEEE Aerospace Conference*, pages 1–11, 2020.
- [16] Brian K Muirhead, Austin Nicholas, and Jeff Umland. Mars sample return mission concept status. In *2020 IEEE Aerospace Conference*, pages 1–8. IEEE, 2020.
- [17] National Aeronautics and Space Administration, Curiosity’s Traverse Map Through Sol 2166. (Date last accessed 15-September-2018).
- [18] National Aeronautics and Space Administration, Opportunity Updates. (Date last accessed 16-January-2019).

- [19] RE Arvidson, RC Anderson, AFC Haldemann, GA Landis, R Li, RA Lindemann, JR Matijevic, RV Morris, L Richter, SW Squyres, et al. Physical properties and localization investigations associated with the 2003 mars exploration rovers. *Journal of Geophysical Research: Planets*, 108(E12), 2003.
- [20] Alex Ellery. *Planetary rovers: robotic exploration of the solar system*. Springer, 2015.
- [21] K Di, J Wang, S He, B Wu, W Chen, R Li, LH Matthies, and AB Howard. Towards autonomous mars rover localization: Operations in 2003 mer mission and new developments for future missions. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37(B1):957–962, 2008.
- [22] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.
- [23] Khaled S Ali, C Anthony Vanelli, Jeffrey J Biesiadecki, Mark W Maimone, Yang Cheng, A Miguel San Martin, and James W Alexander. Attitude and position estimation on the mars exploration rovers. In *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, volume 1, pages 20–27. IEEE, 2005.
- [24] RE Arvidson, P Bellutta, F Calef, AA Fraeman, James B Garvin, O Gasnault, John A Grant, JP Grotzinger, VE Hamilton, M Heverly, 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*, 119(6):1322–1344, 2014.
- [25] Olivier Toupet, Jeffrey Biesiadecki, Arturo Rankin, Amanda Steffy, Gareth Meirion-Griffith, Dan Levine, Maximilian Schadeegg, and Mark Maimone. Terrain-adaptive wheel speed control on the curiosity mars rover: Algorithm and flight results. *Journal of Field Robotics*, 2019.
- [26] Karl Iagnemma and Steven Dubowsky. *Mobile robots in rough terrain: Estimation, motion planning, and control with application to planetary rovers*, volume 12. Springer Science & Business Media, 2004.
- [27] Ramon Gonzalez, Dimi Apostolopoulos, and Karl Iagnemma. Slippage and immobilization detection for planetary exploration rovers via machine learning and proprioceptive sensing. *Journal of Field Robotics*, 35(2):231–247, 2018.
- [28] Johann Borenstein and Liqiang Feng. Measurement and correction of systematic odometry errors in mobile robots. *IEEE Transactions on robotics and automation*, 12(6):869–880, 1996.

- [29] Andrew E Johnson, Steven B Goldberg, Yang Cheng, and Larry H Matthies. Robust and efficient stereo feature tracking for visual odometry. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 39–46. IEEE, 2008.
- [30] Mark Maimone, Yang Cheng, and Larry Matthies. Two years of visual odometry on the mars exploration rovers. *Journal of Field Robotics*, 24(3):169–186, 2007.
- [31] Rongxing Li, Bo Wu, Kaichang Di, Anelia Angelova, Raymond E Arvidson, I-Chieh Lee, Mark Maimone, Larry H Matthies, Lutz Richer, Robert Sullivan, et al. Characterization of traverse slippage experienced by spirit rover on husband hill at gusev crater. *Journal of Geophysical Research: Planets*, 113(E12), 2008.
- [32] George Lentaris, Ioannis Stamoulias, Dimitrios Soudris, and Manolis Lourakis. Hw/sw codesign and fpga acceleration of visual odometry algorithms for rover navigation on mars. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(8):1563–1577, 2015.
- [33] Raymond E Arvidson, Karl D Iagnemma, Mark Maimone, Abigail A Fraeman, Feng Zhou, Matthew C Heverly, Paolo Bellutta, David Rubin, Nathan T Stein, John P Grotzinger, et al. Mars science laboratory curiosity rover megaripple crossings up to sol 710 in gale crater. *Journal of Field Robotics*, 34(3):495–518, 2017.
- [34] Anelia Angelova, Larry Matthies, Daniel Helmick, Gabe Sibley, and Pietro Perona. Learning to predict slip for ground robots. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 3324–3331. IEEE, 2006.
- [35] Krzysztof Skonieczny, Dhara K Shukla, Michele Faragalli, Matthew Cole, and Karl D Iagnemma. Data-driven mobility risk prediction for planetary rovers. *Journal of Field Robotics*, 36(2):475–491, 2019.
- [36] Mark Maimone, Jeffrey Biesiadecki, Edward Tunstel, Yang Cheng, and Chris Leger. Surface navigation and mobility intelligence on the mars exploration rovers. *Intelligence for Space Robotics*, pages 45–69, 2006.
- [37] Raymond E Arvidson, James F Bell III, P Bellutta, Nathalie A Cabrol, JG Catalano, J Cohen, Larry S Crumpler, DJ Des Marais, TA Estlin, WH Farrand, et al. Spirit mars rover mission: Overview and selected results from the northern home plate winter haven to the side of scamander crater. *Journal of Geophysical Research: Planets*, 115(E7), 2010.
- [38] Caleb Thomas Leslie. *Electrical and computer architecture of an autonomous mars sample return rover prototype*. PhD thesis, University of Alabama, 2016.

- [39] Long Xue, Zhaolong Dang, Baichao Chen, Jianqiao Li, and Meng Zou. Pressure-bearing parameter identification for martian soil based on a terramechanics model and genetic algorithm. *Journal of Aerospace Engineering*, 31(2):04017104, 2017.
- [40] Christopher Cunningham. *Improving Prediction of Traversability for Planetary Rovers Using Thermal Imaging*. PhD thesis, Carnegie Mellon University, 2017.
- [41] RE Arvidson, P DeGrosse Jr, JP Grotzinger, MC Heverly, J Shechet, SJ Moreland, MA Newby, N Stein, AC Steffy, F Zhou, et al. Relating geologic units and mobility system kinematics contributing to curiosity wheel damage at gale crater, mars. *Journal of Terramechanics*, 73:73–93, 2017.
- [42] Shoya Higa, Yumi Iwashita, Kyohei Otsu, Masahiro Ono, Olivier Lamarre, Annie Didier, and Mark Hoffmann. Vision-based estimation of driving energy for planetary rovers using deep learning and terramechanics. *IEEE Robotics and Automation Letters*, 4(4):3876–3883, 2019.
- [43] Chris Cunningham, Masahiro Ono, Issa Nesnas, Jeng Yen, and William L Whittaker. Locally-adaptive slip prediction for planetary rovers using gaussian processes. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5487–5494. IEEE, 2017.
- [44] Ramon Gonzalez, Samuel Chandler, and Dimi Apostolopoulos. Characterization of machine learning algorithms for slippage estimation in planetary exploration rovers. *Journal of Terramechanics*, 82:23–34, 2019.
- [45] Christopher Brooks, Karl Iagnemma, and Steven Dubowsky. Vibration-based terrain analysis for mobile robots. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 3415–3420. IEEE, 2005.
- [46] Javier Hidalgo-Carrió, Daniel Hennes, Jakob Schwendner, and Frank Kirchner. Gaussian process estimation of odometry errors for localization and mapping. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5696–5701. IEEE, 2017.
- [47] Forrest Rogers-Marcovitz, Neal Seegmiller, and Alonzo Kelly. 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.
- [48] Forrest Rogers-Marcovitz, Michael George, Neal Seegmiller, and Alonzo Kelly. Aiding off-road inertial navigation with high performance models of wheel slip. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 215–222. IEEE, 2012.

- [49] Eric Foxlin. Pedestrian tracking with shoe-mounted inertial sensors. *IEEE Computer graphics and applications*, (6):38–46, 2005.
- [50] Abdelmoumen Norrdine, Zakaria Kasmi, and Jörg Blankenbach. Step detection for zupt-aided inertial pedestrian navigation system using foot-mounted permanent magnet. *IEEE Sensors Journal*, 16(17):6766–6773, 2016.
- [51] Li Xiaofang, Mao Yuliang, Xie Ling, Chen Jiabin, and Song Chunlei. 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*, pages 1760–1763. IEEE, 2014.
- [52] Arvind Ramanandan, Anning Chen, and Jay A Farrell. Inertial navigation aiding by stationary updates. *IEEE Transactions on Intelligent Transportation Systems*, 13(1):235–248, 2012.
- [53] Martin Brossard, Axel Barrau, and Silvere Bonnabel. Rins-w: Robust inertial navigation system on wheels. *arXiv preprint arXiv:1903.02210*, 2019.
- [54] Cagri Kilic, Nicholas Ohi, Yu Gu, and Jason N. Gross. Slip-based autonomous zupt through gaussian process to improve planetary rover localization. *IEEE Robotics and Automation Letters*, 6(3):4782–4789, 2021.
- [55] Tomoji Takasu. Rtklib: Open source program package for rtk-gps. *Proceedings of the FOSS4G*, 2009.
- [56] Pratap Misra and Per Enge. Global positioning system: signals, measurements and performance second edition. *Massachusetts: Ganga-Jamuna Press*, 2006.
- [57] Paul D Groves. *Principles of GNSS, inertial, and multisensor integrated navigation systems*. Artech house, 2013.
- [58] Johann Borenstein and Liqiang Feng. Umbmark: A method for measuring, comparing, and correcting dead-reckoning errors in mobile robots. Technical report, 1994.
- [59] Bruno Siciliano and Oussama Khatib. *Springer handbook of robotics*. Springer, 2016.
- [60] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.
- [61] Andrew H Jazwinski. *Stochastic processes and filtering theory*. Courier Corporation, 2007.
- [62] Stergios I Roumeliotis, Gaurav S Sukhatme, and George A Bekey. Circumventing dynamic modeling: Evaluation of the error-state kalman filter applied to mobile robot localization. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, volume 2, pages 1656–1663. IEEE, 1999.

- [63] Mark Ebden. Gaussian processes: A quick introduction. *arXiv preprint arXiv:1505.02965*, 2015.
- [64] Peter Orbanz and Yee Whye Teh. Bayesian nonparametric models. *Encyclopedia of machine learning*, 1, 2010.
- [65] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.
- [66] Robert R Richardson, Michael A Osborne, and David A Howey. Gaussian process regression for forecasting battery state of health. *Journal of Power Sources*, 357:209–219, 2017.
- [67] David Duvenaud. *Automatic model construction with Gaussian processes*. PhD thesis, University of Cambridge, 2014.
- [68] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003.
- [69] GPy. GPy: A gaussian process framework in python. <http://github.com/SheffieldML/GPy>, since 2012.
- [70] Paul G Savage. Strapdown inertial navigation integration algorithm design part 2: Velocity and position algorithms. *Journal of Guidance, Control, and Dynamics*, 21(2):208–221, 1998.
- [71] Paul G Savage. Strapdown inertial navigation integration algorithm design part 1: Attitude algorithms. *Journal of Guidance, Control, and Dynamics*, 21(1):19–28, 1998.
- [72] David Titterton, John L Weston, and John Weston. *Strapdown inertial navigation technology*, volume 17. IET, 2004.
- [73] M. Maimone, Y. Cheng, and L. Matthies. Two years of visual odometry on the mars exploration rovers. *Journal of Field Robotics*, 24(2):169–186, 2007.
- [74] Eun-Hwan Shin. *Estimation techniques for low-cost inertial navigation*. PhD thesis, University of Calgary, 2005.
- [75] Cagri Kilic, Shounak Das, Eduardo Gutierrez, Ryan Watson, and Jason Gross. Zupt aided gnss factor graph with inertial navigation integration for wheeled robots. In *Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021)*, pages 3285–3293, 2021.
- [76] Itzik Klein, Sagi Filin, and Tomer Toledo. Pseudo-measurements as aiding to ins during gps outages. *NAVIGATION, Journal of the Institute of Navigation*, 57(1):25–34, 2010.

- [77] Xiaoji Niu, Hongping Zhang, Kai-Wei Chiang, and Naser El-Sheimy. Using land-vehicle steering constraint to improve the heading estimation of mems gps/ins georeferencing systems. In *Proc. Int. Conf. Canadian Geomatics*, pages 1–5, 2010.
- [78] SP Kwakkel, G Lachapelle, and ME Cannon. Gnss aided in situ human lower limb kinematics during running. In *Proceedings of the 21st International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS 2008)*, pages 1388–1397, 2008.
- [79] Rui Zhang, Hai Yang, Fabian Höflinger, and Leonhard M Reindl. Adaptive zero velocity update based on velocity classification for pedestrian tracking. *IEEE Sensors journal*, 17(7):2137–2145, 2017.
- [80] Isaac Skog, Peter Handel, John-Olof Nilsson, and Jouni Rantakokko. Zero-velocity detection—an algorithm evaluation. *IEEE transactions on biomedical engineering*, 57(11):2657–2666, 2010.
- [81] Johan Wahlström and Isaac Skog. Fifteen years of progress at zero velocity: A review. *IEEE Sensors Journal*, 21(2):1139–1151, 2020.
- [82] Gamini Dissanayake, Salah Sukkarieh, Eduardo Nebot, and Hugh Durrant-Whyte. The aiding of a low-cost strapdown inertial measurement unit using vehicle model constraints for land vehicle applications. *IEEE transactions on robotics and automation*, 17(5):731–747, 2001.
- [83] Arthur Gelb. *Applied optimal estimation*. MIT press, 1974.
- [84] Matt Heverly, Jaret Matthews, Justin Lin, Dan Fuller, Mark Maimone, Jeffrey Biesiadecki, and John Leichty. Traverse performance characterization for the mars science laboratory rover. *Journal of Field Robotics*, 30(6):835–846, 2013.
- [85] Ji Zhang and Sanjiv Singh. Visual-lidar odometry and mapping: Low-drift, robust, and fast. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2174–2181. IEEE, 2015.
- [86] Brian Wilcox and Tam Nguyen. Sojourner on mars and lessons learned for future planetary rovers. Technical report, SAE Technical Paper, 1998.
- [87] Kevin Hand. The europa lander mission concept and science goals of the 2016 europa lander science definition team report. *42nd COSPAR Scientific Assembly*, 42:B5–3, 2018.

- [88] William Reid, Michael Paton, Sisir Karumanchi, Brendan Chamberlain-Simon, Blair Emanuel, and Gareth Meirion-Griffith. Autonomous navigation over europa analogue terrain for an actively articulated wheel-on-limb rover. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1939–1946. IEEE, 2020.
- [89] Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018.
- [90] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5):1255–1262, 2017.
- [91] Lukas Meyer, Michal Smíšek, Alejandro Fontan Villacampa, Laura Oliva Maza, Daniel Medina, Martin J Schuster, Florian Steidle, Mallikarjuna Vayugundla, Marcus G Müller, Bernhard Rebele, et al. The madmax data set for visual-inertial rover navigation on mars. *Journal of Field Robotics*, 2021.
- [92] Martin Azkarate, Levin Gerdes, Luc Joudrier, and Carlos J Pérez-del Pulgar. A gnc architecture for planetary rovers with autonomous navigation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3003–3009. IEEE, 2020.
- [93] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual-inertial and multi-map slam. *arXiv preprint arXiv:2007.11898*, 2020.
- [94] Benjamin Jarrett Ebersole. *Skid-steer kinematics for dual-arm mobile manipulator system with dynamic center of gravity*. PhD thesis, University of Texas at Austin, 2016.
- [95] Masafumi Endo, Shogo Endo, Kenji Nagaoka, and Kazuya Yoshida. Terrain-dependent slip risk prediction for planetary exploration rovers. *Robotica*, pages 1–14, 2021.
- [96] Jo Yung Wong. *Theory of ground vehicles*. John Wiley & Sons, 2008.
- [97] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [98] Simon Julier, Jeffrey Uhlmann, and Hugh F Durrant-Whyte. A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on automatic control*, 45(3):477–482, 2000.

- [99] Khaled S Ali, C Anthony Vanelli, Jeffrey J Biesiadecki, Mark W Maimone, Yang Cheng, A Miguel San Martin, and James W Alexander. Attitude and position estimation on the mars exploration rovers. In *2005 IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 20–27. IEEE, 2005.
- [100] Gregory H Peters, William Abbey, Gregory H Bearman, Gregory S Mungas, J Anthony Smith, Robert C Anderson, Susanne Douglas, and Luther W Beegle. Mojave mars simulant—characterization of a new geologic mars analog. *Icarus*, 197(2):470–479, 2008.
- [101] Bruce W Ramme and Matthew P Tharaniyl. *We energies coal combustion products utilization handbook*. We Energies, 2004.
- [102] DW Ming and RV Morris. Chemical, mineralogical, and physical properties of martian dust and soil. 2017.

Appendices

A.1 INS Mechanization

The INS mechanization is composed of four steps: attitude update, specific force transformation, velocity update, and position update. In this formulation, a planet-centered, planet-fixed frame is used as the reference frame, while a locally-level-navigation frame (NED) comprises the resolving axes. The navigation equations are implemented in a locally level navigation frame following the strapdown INS mechanization process as detailed in [70, 71, 72] with closely following the notation in [57]. The attitude update is given as

$$\mathbf{C}_b^{n(+)} \approx \mathbf{C}_b^{n(-)} (\mathbf{I}_3 + \boldsymbol{\Omega}_{ib}^b \Delta t_i) - (\boldsymbol{\Omega}_{ie}^n + \boldsymbol{\Omega}_{en}^n) \mathbf{C}_b^n \Delta t_i \quad (8.1)$$

where $\mathbf{C}_b^n$ is the coordinate transformation matrix from the body frame to the locally level frame, $\mathbf{I}_3$ is a 3-by-3 identity matrix, $\boldsymbol{\Omega}_{ib}^b$ is the skew symmetric matrix of the IMU angular rate measurement, $\boldsymbol{\Omega}_{ie}^n$ is the skew symmetric matrix of the planet's rotation vector represented in the locally level frame, $\boldsymbol{\Omega}_{en}^n$ is the transport term, and Δt_i is the IMU sampling interval.

Accelerometers measure the specific force which is the acceleration relative to free fall or the total non-gravitational force per unit mass. After the attitude has been updated, the accelerometer measured specific force transformation is applied. The transformation is necessary since the accelerometers measure specific force along the body-axis [57].

$$\mathbf{f}_{ib}^n \approx \frac{1}{2} (\mathbf{C}_b^{n(-)} + \mathbf{C}_b^{n(+)}) \mathbf{f}_{ib}^b \quad (8.2)$$

Assume that the acceleration variations due to gravity, Coriolis, and transport rate terms are neg-

ligible for the integration interval [57]. Noting that the specific force must be resolved in the local level navigation frame, the velocity update is given as;

$$\mathbf{v}_{eb}^{n(+)} \approx \mathbf{v}_{eb}^{n(-)} + (\mathbf{f}_{ib}^n + \mathbf{g}_b^n(L_b^{(-)}, h_b^{(-)}) - (\boldsymbol{\Omega}_{en}^{n(-)} + 2\boldsymbol{\Omega}_{ie}^{n(-)})\mathbf{v}_{eb}^{n(-)})\Delta t_i \quad (8.3)$$

where $\mathbf{v}_{eb}^n$ is the velocity update, $\mathbf{f}_{ib}^n$ is the transformed specific force measurements from the IMU acceleration sensors, $\mathbf{g}_b^n$ is the gravity vector.

Assume that the velocity variation is linear over the integration interval [57]. The position update with this assumption can be given as

$$h_b^{(+)} = h_b^{(-)} - \frac{\Delta t_i}{2} \left(\mathbf{v}_{eb,D}^{n(-)} + \mathbf{v}_{eb,D}^{n(+)} \right) \quad (8.4)$$

$$L_b^{(+)} = L_b^{(-)} + \frac{\Delta t_i}{2} \frac{\mathbf{v}_{eb,N}^{n(-)}}{R_N(L_b^{(-)}) + h_b^{(-)}} + \frac{\Delta t_i}{2} \frac{\mathbf{v}_{eb,N}^{n(+)}}{R_N(L_b^{(-)}) + h_b^{(+)}} \quad (8.5)$$

$$\lambda_b^{(+)} = \lambda_b^{(-)} + \frac{\Delta t_i}{2} \frac{\mathbf{v}_{eb,E}^{n(-)}}{(R_E(L_b^{(-)}) + h_b^{(-)})\cos L_b^{(-)}} + \frac{\Delta t_i}{2} \frac{\mathbf{v}_{eb,E}^{n(+)}}{(R_P(L_b^{(-)}) + h_b^{(+)})\cos L_b^{(+)}} \quad (8.6)$$

where h_b , L_b and λ_b are updated position estimates (expressed in terms of height, latitude, and longitude, respectively), R_N is the variation of the meridian, and R_P is transverse radii of curvature.

A.2 Wheel Odometry aided Inertial State Estimation

The inertial state estimation is obtained by an error state extended Kalman filter. In this appendix, a summary of the Kalman filter and extended Kalman filter are given first, then wheel odometry aiding inertial state estimation using error state extended Kalman filter is detailed.

A.2.1 Error State Extended Kalman Filter

Consider the error state, $\delta \mathbf{x}^n \in \mathbb{R}^{15}$, is constructed in a local navigation frame,

$$\delta \mathbf{x}^n = \left(\delta \Psi_{nb}^n \ \delta \mathbf{v}_{eb}^n \ \delta \mathbf{p}_b \ \mathbf{b}_a \ \mathbf{b}_g \right)^T, \quad \delta \mathbf{p}_b = \left(\delta L_b \ \delta \lambda_b \ \delta h_b \right)^T \quad (8.7)$$

where, $\delta \Psi_{nb}^n \in \mathbb{R}^3$ is the attitude error, $\delta \mathbf{v}_{eb}^n \in \mathbb{R}^3$ is the velocity error, $\delta \mathbf{p}_b \in \mathbb{R}^3$ is the position error, $\mathbf{b}_a \in \mathbb{R}^3$ is the IMU acceleration bias, and $\mathbf{b}_g \in \mathbb{R}^3$ is the IMU gyroscope bias. The position error is expressed in terms of the latitude, longitude, and height, respectively. Consider the error state EKF equations in Chapter 4.2.3

$$\check{\mathbf{x}}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{0}) \quad (8.8)$$

$$\check{\mathbf{P}}_k = \mathbf{F}_{k-1} \mathbf{P}_{k-1} \mathbf{F}_{k-1}^T + \mathbf{L}_{k-1} \mathbf{Q}_{k-1} \mathbf{L}_{k-1}^T \quad (8.9)$$

$$\mathbf{K}_k = \check{\mathbf{P}}_k \mathbf{H}_k^T (\mathbf{H}_k \check{\mathbf{P}}_k \mathbf{H}_k^T + \mathbf{R})^{-1} \quad (8.10)$$

$$\delta \hat{\mathbf{x}}_k = \mathbf{K}_k (\mathbf{y}_k - \mathbf{h}_k(\check{\mathbf{x}}_k, \mathbf{0})) \quad (8.11)$$

$$\hat{\mathbf{x}}_k = \check{\mathbf{x}}_k + \delta \hat{\mathbf{x}}_k \quad (8.12)$$

$$\hat{\mathbf{P}}_k = (1 - \mathbf{K}_k \mathbf{H}_k) \check{\mathbf{P}}_k. \quad (8.13)$$

A.2.2 INS System Matrix

The system matrix can be given as;

$$\mathbf{F}_{INS}^n = \begin{pmatrix} \mathbf{F}_{11}^n & \mathbf{F}_{12}^n & \mathbf{F}_{13}^n & 0_3 & \hat{\mathbf{C}}_b^n \\ \mathbf{F}_{21}^n & \mathbf{F}_{22}^n & \mathbf{F}_{23}^n & \hat{\mathbf{C}}_b^n & 0_3 \\ 0_3 & \mathbf{F}_{32}^n & \mathbf{F}_{33}^n & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \end{pmatrix} \quad (8.14)$$

with denoting $\hat{R}_E = \left(R_E \left(\hat{L}_b \right) + \hat{h}_b \right)$ and $\hat{R}_N = \left(R_N \left(\hat{L}_b \right) + \hat{h}_b \right)$ the system matrix elements can be written with closely following the notation in [57]

$$\mathbf{F}_{11}^n = -[\hat{\boldsymbol{\omega}}_{in}^n \wedge] \quad (8.15)$$

$$\mathbf{F}_{12}^n = \begin{bmatrix} 0 & \frac{-1}{\hat{R}_E} & 0 \\ \frac{1}{\hat{R}_N} & 0 & 0 \\ 0 & \frac{\tan \hat{L}_b}{\hat{R}_E} & 0 \end{bmatrix} \quad (8.16)$$

$$\mathbf{F}_{13}^n = \begin{bmatrix} \omega_{ie} \sin \hat{L}_b & 0 & \frac{\hat{v}_{eb,E}^n}{(\hat{R}_E)^2} \\ 0 & 0 & \frac{-\hat{v}_{eb,N}^n}{(\hat{R}_N)^2} \\ \omega_{ie} \cos \hat{L}_b + \frac{\hat{v}_{eb,E}^n}{(\hat{R}_E) \cos^2 \hat{L}_b} & 0 & \frac{-\hat{v}_{eb,E}^n \tan \hat{L}_b}{(\hat{R}_E)^2} \end{bmatrix} \quad (8.17)$$

$$\mathbf{F}_{21}^n = -\left[\left(\hat{\mathbf{C}}_b^n \hat{\mathbf{f}}_{ib}^b \right) \wedge \right] \quad (8.18)$$

$$\mathbf{F}_{22}^n = \begin{bmatrix} \frac{\hat{v}_{eb,D}^n}{\hat{R}_N} & -\frac{2\hat{v}_{eb,E}^n \tan \hat{L}_b}{\hat{R}_E} - 2\omega_{ie} \sin \hat{L}_b & \frac{\hat{v}_{eb,N}^n}{\hat{R}_N} \\ \frac{\hat{v}_{eb,E}^n \tan \hat{L}_b}{\hat{R}_E} + 2\omega_{ie} \sin \hat{L}_b & \frac{\hat{v}_{eb,N}^n \tan \hat{L}_b + \hat{v}_{eb,D}^n}{\hat{R}_E} & \frac{\hat{v}_{eb,E}^n}{\hat{R}_E} + 2\omega_{ie} \cos \hat{L}_b \\ -\frac{2\hat{v}_{eb,N}^n}{\hat{R}_N} & -\frac{2\hat{v}_{eb,E}^n}{\hat{R}_E} - 2\omega_{ie} \cos \hat{L}_b & 0 \end{bmatrix} \quad (8.19)$$

$$\mathbf{F}_{23}^{n_n} = \begin{bmatrix} -\frac{(\hat{v}_{eb,E}^n)^2 \sec^2 \hat{L}_b}{\hat{R}_E} - 2\hat{v}_{eb,E}^n \omega_{ie} \cos \hat{L}_b & 0 & \frac{(\hat{v}_{eb,E}^n)^2 \tan \hat{L}_b}{(\hat{R}_E)^2} - \frac{\hat{v}_{eb,N}^n \hat{v}_{eb,D}^n}{(\hat{R}_N)^2} \\ \frac{\hat{v}_{eb,N}^n \hat{v}_{eb,E}^n \sec^2 \hat{L}_b}{\hat{R}_E} + 2\hat{v}_{eb,N}^n \omega_{ie} \cos \hat{L}_b - 2\hat{v}_{eb,D}^n \omega_{ie} \sin \hat{L}_b & 0 & -\frac{\hat{v}_{eb,N}^n \hat{v}_{eb,E}^n \tan \hat{L}_b + \hat{v}_{eb,E}^n \hat{v}_{eb,D}^n}{(\hat{R}_E)^2} \\ 2\hat{v}_{eb,E}^n \omega_{ie} \sin \hat{L}_b & 0 & \frac{(\hat{v}_{eb,E}^n)^2}{(\hat{R}_E)^2} + \frac{(\hat{v}_{eb,N}^n)^2}{(\hat{R}_N)^2} - \frac{2g_0(\hat{L}_b)}{r_{es}^e(\hat{L}_b)} \end{bmatrix} \quad (8.20)$$

$$\mathbf{F}_{32}^n = \begin{bmatrix} \frac{1}{\hat{R}_N} & 0 & 0 \\ 0 & \frac{1}{(\hat{R}_E) \cos \hat{L}_b} & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (8.21)$$

$$\mathbf{F}_{33}^n = \begin{bmatrix} 0 & 0 & -\frac{\hat{v}_{eb,N}^n}{(\hat{R}_N)^2} \\ \frac{\hat{v}_{eb,E}^n \sin \hat{L}_b}{(\hat{R}_E) \cos^2 \hat{L}_b} & 0 & -\frac{\hat{v}_{eb,E}^n}{(\hat{R}_E)^2 \cos \hat{L}_b} \\ 0 & 0 & 0 \end{bmatrix} \quad (8.22)$$

A.2.3 INS State Transition Matrix

$$\Phi_{INS}^n = \begin{bmatrix} \Phi_{11} & \Phi_{12} & \Phi_{13} & 0_3 & \Phi_{15} \\ \Phi_{21} & \Phi_{22} & \Phi_{23} & \Phi_{24} & 0_3 \\ 0_3 & \Phi_{32} & \Phi_{33} & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & \mathbf{I}_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & \mathbf{I}_3 \end{bmatrix} \quad (8.23)$$

$$\Phi_{11} = \mathbf{I}_3 + \mathbf{F}_{11}^n \tau_s \quad (8.24)$$

$$\Phi_{12} = \mathbf{F}_{12}^n \tau_s \quad (8.25)$$

$$\Phi_{13} = \mathbf{F}_{13}^n \tau_s \quad (8.26)$$

$$\Phi_{15} = \hat{\mathbf{C}}_b^n \tau_s \quad (8.27)$$

$$\Phi_{21} = \mathbf{F}_{21}^n \tau_s \quad (8.28)$$

$$\Phi_{22} = \mathbf{I}_3 + \mathbf{F}_{22}^n \tau_s \quad (8.29)$$

$$\Phi_{23} = \mathbf{F}_{23}^n \tau_s \quad (8.30)$$

$$\Phi_{24} = \hat{\mathbf{C}}_b^n \tau_s \quad (8.31)$$

$$\Phi_{32} = \mathbf{F}_{32}^n \tau_s \quad (8.32)$$

$$\Phi_{33} = \mathbf{I}_3 + \mathbf{F}_{33}^n \tau_s \quad (8.33)$$

A.2.4 INS Noise Covariance Matrix

INS system noise covariance matrix can be defined by the random walk of the velocity error due to noise on the accelerometer specific-force measurements and random walk of the attitude error due to noise on the gyro angular rate measurements [57]. Integrating the power spectral densities of the noise sources over the state propagation interval, the INS system noise covariance matrix can be given with closely following the notation in [57];

$$\mathbf{Q}_{INS}^n = \begin{pmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{21}^{nT} & \mathbf{Q}_{31}^{nT} & 0_3 & \frac{1}{2}S_{bgd}\tau_s^2\hat{\mathbf{C}}_b^n \\ \mathbf{Q}_{21}^n & \mathbf{Q}_{22}^n & \mathbf{Q}_{32}^{nT} & \frac{1}{2}S_{bad}\tau_s^2\hat{\mathbf{C}}_b^n & \frac{1}{3}S_{bgd}\tau_s^3\mathbf{F}_{21}^n\hat{\mathbf{C}}_b^n \\ \mathbf{Q}_{31}^n & \mathbf{Q}_{32}^n & \mathbf{Q}_{33}^n & \mathbf{Q}_{34}^n & \mathbf{Q}_{35}^n \\ 0_3 & \frac{1}{2}S_{bad}\tau_s^2\hat{\mathbf{C}}_n^b & \mathbf{Q}_{34}^{nT} & S_{bad}\tau_s\mathbf{I}_3 & 0_3 \\ \frac{1}{2}S_{bgd}\tau_s^2\hat{\mathbf{C}}_n^b & \frac{1}{3}S_{bgd}\tau_s^3\mathbf{F}_{21}^n\hat{\mathbf{C}}_n^b & \mathbf{Q}_{35}^{nT} & 0_3 & S_{bgd}\tau_s\mathbf{I}_3 \end{pmatrix} \quad (8.34)$$

where S_{rg} , S_{ra} , S_{bad} , and S_{bgd} are the power spectral densities of the gyro random noise, accelerometer random noise, accelerometer bias variation, and gyro bias variation, respectively [57]. The elements of the INS system noise covariance matrix in 8.34 are given with closely following the notation in [57] as:

$$\mathbf{Q}_{11}^n = \left(S_{rg}\tau_s + \frac{1}{3}S_{bgd}\tau_s^3 \right) \mathbf{I}_3 \quad (8.35)$$

$$\mathbf{Q}_{21}^n = \left(\frac{1}{2}S_{rg}\tau_s^2 + \frac{1}{4}S_{bgd}\tau_s^4 \right) \mathbf{F}_{21}^n \quad (8.36)$$

$$\mathbf{Q}_{22}^n = \left(S_{ra}\tau_s + \frac{1}{3}S_{bad}\tau_s^3 \right) \mathbf{I}_3 + \left(\frac{1}{3}S_{rg}\tau_s^3 + \frac{1}{5}S_{bgd}\tau_s^5 \right) \mathbf{F}_{21}^n \mathbf{F}_{21}^{n\mathbf{T}} \quad (8.37)$$

$$\mathbf{Q}_{31}^n = \left(\frac{1}{3}S_{rg}\tau_s^3 + \frac{1}{5}S_{bgd}\tau_s^5 \right) \mathbf{T}_{r(n)}^p \mathbf{F}_{21}^n \quad (8.38)$$

$$\mathbf{Q}_{32}^n = \left(\frac{1}{2}S_{ra}\tau_s^2 + \frac{1}{4}S_{bad}\tau_s^4 \right) \mathbf{T}_{r(n)}^p + \left(\frac{1}{4}S_{rg}\tau_s^4 + \frac{1}{6}S_{bgd}\tau_s^6 \right) \mathbf{T}_{r(n)}^p \mathbf{F}_{21}^n \mathbf{E}_{21}^n \mathbf{T} \quad (8.39)$$

$$\mathbf{Q}_{33}^n = \left(\frac{1}{3}S_{ra}\tau_s^3 + \frac{1}{5}S_{bad}\tau_s^5 \right) \mathbf{T}_{r(n)}^p \mathbf{T}_{r(n)}^p + \left(\frac{1}{5}S_{rg}\tau_s^5 + \frac{1}{7}S_{bgd}\tau_s^7 \right) \mathbf{T}_{r(n)}^p \mathbf{F}_{21}^n \mathbf{F}_{21}^n \mathbf{T}_{r(n)}^p \quad (8.40)$$

$$\mathbf{Q}_{34}^n = \frac{1}{3}S_{bad}\tau_s^3 \mathbf{T}_{r(n)}^p \hat{\mathbf{C}}_b^n \quad (8.41)$$

$$\mathbf{Q}_{35}^n = \frac{1}{4}S_{bgd}\tau_s^4 \mathbf{T}_{r(n)}^p \mathbf{F}_{21}^n \hat{\mathbf{C}}_b^n \quad (8.42)$$

A.3 Supplementary Figures

Figure A.3.1: Various rate gyro outputs (norm value) during traversal on different terrain types. The difference on the IMU outputs are due to roughness of the terrain which induces different vibration acting on wheels.

(a) Test3

(b) Test3

(c) TestS1E2

(d) TestS1E2

(e) Test11

(f) Test11

Figure A.3.2: Supplementary Velocity Error Comparison Figures