Master's Thesis

Improving Odometry in Rimless Wheeled Rover using Sensor Fusion

Ravisankar Selvaraju

Submitted to Hochschule Bonn-Rhein-Sieg,
Department of Computer Science
in partial fullfilment of the requirements for the degree
of Master of Science in Autonomous Systems

Supervised by

Prof. Dr. Sebastian Houben
Prof. Dr.-Ing. Udo Frese[1]
M. Sc. Raúl Domínguez[2]
Dr.-Ing. Christoph Hertzberg[2]

February 2026

[1] University of Bremen, Bremen
[2] Robotics Innovation Center, German Research Center for Artificial Intelligence (DFKI), Bremen

© 2020 Ravisankar Selvaraju. All Rights Reserved

[If AI assistants have not been used, use this sentence] I, the undersigned below, declare that this work has not previously been submitted to this or any other university and that it is, unless otherwise stated, entirely my own work.

[If an AI assistant has been used, use this sentence] I, the undersigned below, declare that this work has not previously been submitted to this or any other university and that it is, unless otherwise stated, entirely my own work. The report was, in part, written with the help of the AI assistant [AI assistant name] as described in the appendix. I am aware that content generated by AI systems is no substitute for careful scientific work, which is why all AI-generated content has been critically reviewed by me, and I take full responsibility for it.

Date

Ravisankar Selvaraju

Abstract

Reliable state estimation is vital for Rimless Wheeled Rovers (RWR) operating in unstructured and planetary environments where external positioning systems are unavailable. Traditional dead-reckoning odometry approaches are prone to significant errors due to sensor drift and unaccounted wheel slippage on loose terrains such as sand and gravel. This thesis proposes an improved odometry solution using sensor fusion within an Invariant Extended Kalman Filter (InEKF) framework. The approach utilizes a Potential Contact Velocity (PCV) based update model combined with a slip filtering mechanism to detect and discard erroneous wheel measurements when slippage occurs. By monitoring the consistency between Inertial Measurement Unit (IMU) data and wheel encoder data, the system aims to exclude wheel encoder data with slip from the state estimation process. The performance of this slip-aware framework will be evaluated against the current baseline odometry through simulations on inclined terrains and real-world experiments with the Coyote 3 rover, utilizing metrics such as Mean Squared Error (MSE) and error per unit distance.

Source: proposal

Acknowledgements

Thanks to ....

Contents

1

Introduction

Reliable state estimation is vital for efficient operation of Rimless Wheeled Rover (RWR) in unstructured and planetary environments. Traditional dead-reckoning-based odometry approaches are prone to drift and slippage-related errors. This interoceptive sensor-based state estimation is critical in scenarios where the use of visual data for odometry is limited due to computational power constraints and environmental conditions [1]. The availability of reliable proprioceptive odometry enables the possibility of sensor fusion with visual and other forms of odometry. Relying on a single source of odometry is susceptible to errors, so using sensor fusion techniques can improve the quality of odometry. The introduction section is organised by dividing the project title into multiple parts. Each part explains a specific aspect of the topic, making it easier for the reader to understand.

Source: proposal

1.1 Motivation

Mobile robots in general need to estimate their location relative to either their previous location or some landmarks in the environment for an effective autonomous navigation task. Proprioceptive sensor-based odometry estimation is effective in scenarios where external positioning systems are unavailable. However, this form of odometry suffers from sensor drift over time and wheel slippage. In order to combat these issues, sensor fusion techniques are commonly employed.

Source: proposal

1.1.1 ...

Rimless Wheeled Rovers (RWR) are used in planetary exploration and search operations in tunnels Can climb obstacles and operate in variety of terrain Efficiency and traversability [1] Reliable odometry is crucial for autonomous systems in unstructured environment in the absence of GPS [2] Odometry is the foundation for tasks such as SLAM and Trajectory following

Source: Slides

1.1.2 ...

1.2 Challenges and Difficulties

Baseline Odometry in Coyote rover Decoupled odometry: Wheel Odometry for translation, IMU for rotation Wheel odometry ignores RW geometry and contacts IMU based orientation suffers from drift overtime Unaccounted slippage leads to translational errors Higher risk of slippage in sand, loose rock and inclined terrain

Source: Slides

1.2.1 ...

1.2.2 ...

1.2.3 ...

1.3 Problem Statement

Based on the literature survey and the limitations discussed, a new approach of using Potential Contact Velocity (PCV) based odometry using Invariant Extended Kalman Filter (InEKF) framework is proposed. This proposed approach will be compared with the baseline odometry to evaluate the performance of sensor fusion in RWR. The InEKF implementation is developed by Ross Hartley [2] and is available as open-source[1].

On top of the InEKF implementation, a slip filtering mechanism will be integrated into the state estimation process to discard the erroneous wheel speed measurement data when slippage events occur. The literature proposes various methods for slip modelling, such as using wheel motor current measurements [3] or offline analysis of recorded data to estimate terrain slip parameters [4]. However, these slip modelling and estimation techniques have not been tested on rimless wheeled rovers. Consequently, the baseline odometry model implemented in our rimless wheeled rover currently does not include a slippage model or account for slip in any way.

Finally, the prospect of using the InEKF framework to estimate the Potential Contact (PC) will be explored. The PC can be inferred by the PCV estimated using IMU data and the wheel velocity data. When the PCV vector direction from the IMU estimate and the wheel velocity measurement match, then that spoke can be considered as a PC.

A simple block diagram showcasing the proposed approach is shown in Figure The proposed approach will be tested in simulation on inclined terrains and in real-world scenarios with sand and loose rocks inside DFKI's multifunction hall. The evaluation will compare the estimation error of standard odometry against the proposed InEKF framework using metrics like MSE and max error per distance in percentage.

[1] InEKF implementation: `https://github.com/RossHartley/invariant-ekf`

A possible Key Performance Indicator (KPI) could be errors (m) per unit distance(m). This quantitative measure can assess the performance of the proposed approach, and the objective is to keep the KPI as low as possible for the SLAM system to work. This demonstrates that an improved odometry can lead to better performance of SLAM.

Source: proposal

1.3.1 ...

1.3.2 ...

1.3.3 ...

2
Background

Change the chapter title if necessary

2.0.1 Odometry

The process of estimating a robot's current pose based on its sensor data, such as wheel encoder and Inertial Measurement Unit (IMU), is called odometry. This can be calculated using the rover's proprioceptive sensors, such as wheel encoders and IMU, as well as exteroceptive sensors like cameras and LiDARs. In situations like an underground cave, tunnels, and space exploration, Odometry plays a vital role in tracking the movements of mobile rovers when external positioning systems like GPS (Global Positioning System) and other motion capture systems might not be available. The use of an exteroceptive sensor can improve the state estimation; nevertheless, a reliable proprioceptive sensor-based state estimation provides a better starting point for odometry, which can then later be fused with other sources of odometry.

The current implementation[1] of baseline odometry in DFKI's rimless wheeled rovers is a combination of both wheel and inertial odometry. The linear motion is calculated from the wheel odometry based on a simplified wheel model, which doesn't consider the spoked geometry and contact position of the rimless wheel's spokes. The attitude/orientation changes are calculated from the inertial odometry.

The baseline odometry suffers from the disadvantages of orientation drift over time from the inertial odometry and error due to unaccounted slippage from the wheel odometry. These limitations motivate us to consider sensor fusion techniques to improve the quality of odometry and reduce the error from individual odometry sources.

Source: proposal

2.0.2 Rimless wheeled rover

Mobile robots use various types of locomotion. Some common mechanisms include legged, wheeled, track-wheel hybrid and whegs (leg-wheel hybrid). Among these, rimless wheeled systems offer a unique

[1] Baseline odometry on Coyote: `https://github.com/rock-slam/slam-odometry/blob/master/src/Odometry.hpp`

Figure 2.1: Coyote 3 rimless wheeled rover with SIMA arm in a rocky terrain

alternative that combines the simplicity of wheels with the terrain adaptability of legs. A Rimless Wheel is a special type of wheel-like structure formed by evenly spaced spokes connected to the centre of the wheel. The discontinuous nature of the surface of the wheel mimics a legged motion. These types of rimless wheels have been used in the RIC (Robotic Innovation Center) of DFKI (German Research Center for Artificial Intelligence) to construct four-wheeled rimless wheeled rovers such as Asguard and Coyote in Figure 2.1. These rovers, due to their spoked wheels, can traverse diverse terrains like soft sand, coarse sand, rocky terrain and inclined terrain. Unlike traditional wheeled rovers, rimless wheeled rovers can overcome challenging obstacles effectively. These rovers, when travelling autonomously, need to keep track of their pose to navigate effectively.

Given that the rimless wheeled rover can travel in diverse terrain, the odometry calculation to track the robot's pose is more challenging than in other terrains. One of the challenging points is to achieve an odometry that works well in different types of terrain. Since the contact behaviours differ based on the shape and texture of materials on the terrain, it is vital to improve the odometry of rimless wheeled rovers for effective navigation on diverse terrains.

Source: proposal

2.0.3 Sensor fusion

One of the well-known solutions to deal with the unreliability of single sensor-based pose estimation is to use sensor fusion techniques. These techniques fuse sensor data from different sensors to get a reliable estimate of the rover's pose. The use of sensor fusion techniques is studied in traditional wheeled [5–8] and legged [9] based locomotion in the literature.

As mentioned in the subsection 2.0.1, the baseline odometry does not perform true sensor fusion of wheel and inertial odometry. Instead, it simply combines selected components from each odometry. The proposed approach uses an Extended Kalman filter (EKF) to fuse the inertial odometry data and wheel odometry data. The key improvement this thesis aims at is to include slip detection and elimination during the sensor fusion process so that error-prone wheel odometry data will not be used in the final pose estimation.

Since the primary advantage of rimless wheels is to traverse challenging terrain, it is vital to study the effects of sensor fusion techniques in improving the rimless wheeled rover's odometry.

Source: proposal

Include de estimator/ equations, and workin server

2.0.4 Slippage

A traditional or rimless wheeled mobile rover in motion experiences two types of slippage at the wheels: Longitudinal slippage (slip) and lateral slip (skid). Slip happens in the direction of motion of wheels, and skid happens perpendicular to the motion of the wheel [10]. Unlike traditional wheeled systems, slip in RWR is more similar to legged robots. Wheeled systems have continuous contact with the surface, making the slip continuous, but in the Rimless Wheel (RW) system, slip happens only during contact phases when the spokes touch the terrain.

Slip can happen due to various reasons. Firstly, when the friction coefficient of the surface is low, the contact points/contact surface of the spokes lose grip. Secondly, when the surface is deformable under stress, like sand terrain, and the wheel contact points exert stress exceeding the terrain's maximum shear strength, deformation occurs, resulting in slip [3]. Finally, when the angular velocity of the wheels is high, causing a higher tangential force at the contact points, it exceeds static friction, leading to slip.

Slip can be computed as the ratio of the difference between the wheel's linear velocity v_{wheel} (calculated as the product of angular velocity ω and wheel radius r) and the body's linear velocity v_{body}, to the wheel's linear velocity [8].

$$\text{Slip} = \frac{v_{\text{wheel}} - v_{\text{body}}}{v_{\text{wheel}}}, \text{ where } v_{\text{wheel}} = \omega \cdot r$$

One of the systems whose performance is directly affected by slippage is the wheel odometry. Since the wheel odometry does not have the knowledge that slip has happened, it calculates the slippage as the rover's motion. This makes the wheel odometry unreliable in slippery conditions [5]. The survey paper [11] by Gonzalez and Lagnemma provide a comprehensive analysis of significant slip-related incidents in planetary exploration. This paper also presents some common slippage estimation approaches, such as Exteroceptive-based solutions, Proprioceptive-based solutions, Model-based solutions and Kalman filter-based solutions.

Source: proposal

2.0.5 InEKF library and its update modalities

write about the current measurement function and update methods used to update the state in the InEKF library

write abou tion step a data is use

2.0.6 PCV

PCV

2.0.7 Notations

These following notations follow the monogram notations introduced in MIT notes on manipulation by Russ Tedrake [12].

p^A Position of Point A
${}^W p^A$ Position of Point A measured from Point W
${}^W p^A_F$ Position of Point A measured from Point W represented in frame F
T^A_F Pose of frame or point A represented in frame F
R^A_F Rotation matrix of frame or point A represented in frame F

Here, A is target(frame or point) and W is measured from (frame or point) and F is expressed in (frame). T^{body}_{world} should be read as "The Pose of body frame represented in the world frame" or "The transformation matrix that converts a vector from the body frame to the world frame". $\boldsymbol{\xi}^{world}_{body}$ should be read as "The twist of the body frame represented in the world frame". The subscript for the twist is in motion and the superscript is the reference frame in which the twist is represented. Here, the twist of body frame is represented in the world frame.

2.0.8 Twists

Twists or Spatial velocity capture the Instantaneous motion of a rigid body in 3D space, and it merges the linear and angular velocities into a single entity [13]. The mathematical representation of a twist can vary depending on the source. Another interpretation of twist expresses the screw motion of the body, which is stated by the screw theory. [14] Pg: 46 Definition: 2.2. The book "Modern Robotics" by Lynch and Park [13] uses the twist convention that orders the angular velocity first and then the linear velocity in the vector 2.1

$$\boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{\omega} \\ \mathbf{v} \end{bmatrix} \in \mathbb{R}^6 \tag{2.1}$$

The "Rigid Body Dynamics" book [15] by Roy Featherstone uses the spatial velocity which orders the linear velocity first and then the angular velocity 2.2. This convention is also followed in the Kinematics and Dynamics Library (KDL) [16].

$$\boldsymbol{\xi} = \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} \in \mathbb{R}^6 \tag{2.2}$$

Where
$\boldsymbol{\omega}$ is Angular velocity
$\mathbf{v}$ is Linear velocity

Since, the proposed approach, we plan to use the KDL library to transformation of Twists, we will follow the spatial velocity 2.2 convention by ordering the linear velocity first and then the angular velocity in the Twist vector.

Transforming Twists

Twist can be represented in two ways based on the reference frame: Body twist and spatial twist. The body twist is represented in the body frame attached to the moving rigid body and the spatial twist is represented in the fixed world reference frame. Both twists represent the same physical quantity i.e., the Instantaneous motion of the rigid body.

Let's assume, we have a world frame and a body frame attached to an infinitely large rigid body. The body twist will the velocity experienced by a point of the rigid body at the origin of body frame by the motion of the rigid body. The spatial twist or twist of world frame will be the velocity experienced by a point of the rigid body at the origin of world frame by the motion of the rigid body. For more clear understanding, we take the example of a place flying above a person/observer. The plane is the Rigid body. The body twist answers "What does the pilot feel?" and the spatial twist answers "What does a stationary observer sees?" The twist transformation is done using Adjoint transformation. This Adjoint operation expressed as below, This adjoint representation of the Transformation matrix handles the rotation and the lever arm effect of the translation when transforming the twist from one frame to another frame.

For a transformation matrix T_{world}^{body} and a twist $\boldsymbol{\xi}_{body}$ represented in the body frame,

$$T_{world}^{body} = \begin{bmatrix} R_{world}^{body} & p_{world}^{body} \\ 0 & 1 \end{bmatrix} \tag{2.3}$$

The Adjoint transformation for the transformation matrix 2.3 is given by the equation 2.4

$$\mathrm{Ad}_{T_{world}^{body}} = \begin{bmatrix} R_{world}^{body} & 0 \\ [p_{world}^{body}]_{\times} R_{world}^{body} & R_{world}^{body} \end{bmatrix} \tag{2.4}$$

If we need to transform a twist of the body frame represented in the body frame $\boldsymbol{\xi}_{body}$ to the twist of body frame represented in the world frame $\boldsymbol{\xi}_{world}$, we can use the following equation 2.5

$$\boldsymbol{\xi}_{body}^{world} = \mathrm{Ad}_{T_{world}^{body}} \boldsymbol{\xi}_{body}^{body} \tag{2.5}$$

3

State of the Art

Mobile robots in general need to estimate their location relative to either their previous location or some landmarks in the environment for an effective autonomous navigation task. Proprioceptive sensor-based odometry estimation is effective in scenarios where external positioning systems are unavailable. However, this form of odometry suffers from sensor drift over time and wheel slippage. In order to combat these issues, sensor fusion techniques are commonly employed. This section discusses the previous works which attempted to solve this problem.

Source: proposal

3.1 Wheel Inertial Odometry Fusion

The estimation of a rover's pose relies on the fusion of wheel odometry and inertial odometry data, a technique explored across numerous studies with various approaches. One common approach involves Kalman Filter-based sensor fusion for integrating Inertial Measurement Unit (IMU) and wheel odometry data [6, 17]. This method aims to improve the accuracy of robot state estimation. For instance, combining filtered robot state information with GPS data through a weighted fusion technique has demonstrated enhanced accuracy in both indoor and outdoor scenarios [17]. This highlights how strong proprioceptive odometry contributes to overall robot state estimation.

To address challenges such as IMU drift and wheel slippage, robust Kalman filters and EKF have been proposed, leading to increased accuracy of rover localization even under wheel slip conditions [6]. In contrast, our approach actively estimates and compensates slip-related wheel odometry data to improve the accuracy.

An Error State Kalman filter has also been proposed in [8], which fuses the IMU data and wheel encoder data to improve the pose estimation accuracy. Furthermore, Unscented Kalman Filters (UKF) have been proposed for slip-aware motion estimation [7, 18]. One such approach integrates the innovation (the difference between actual and estimated state) over time during the update step to enhance estimation accuracy. In this method, the robot's state is initially estimated using encoder data processed through Instantaneous Center of Rotation (ICR) kinematics, which is then fused with inertial odometry for improved accuracy. However, a limitation of this specific method is its reliance on the ICR kinematics assumption that wheels on the same side have identical angular velocities, which may not hold true under varying slip conditions [7].

Another application of the UKF involves a novel track-to-track multi-sensor fusion algorithm that fuses IMU and wheel encoder data. This proposed fusion algorithm has been shown to generate consistent estimates compared to stand-alone odometry methods in simulated environments [18].

Source: proposal

3.2 Slippage estimation

Estimating and compensating for slip is very important for getting accurate odometry. Many methods try to handle this by modelling slip using terrain parameters, ground truth data from motion tracking systems or other approaches.

One of the methods discussed tracked vehicles, modelling slip on loose ground during straight and turning moves. This uses slip ratio for straight paths and slip angle for turns. The slip model details for this method are estimated using a regression model pre-trained on offline data from motion capture systems. [4]. This method cannot be used for an unknown environment without collecting offline data to train the regression model.

Collecting ground truth data in an unknown environment is not always possible; another approach proposed a slip-aware wheel odometry using a slip parameter method [19]. This method does not need actual position or speed data, which makes it good for places where external positioning systems are not available. However, this method assumes that a slip happens on all wheels at the same time on loose soil, and this might not be true in real traversal situations.

Another method to estimate slip relies on current measurements from the wheel motors. [3] This technique is based on terrain parameters, which are automatically tuned as the rover travels over a terrain. This eliminates the need for ground truth data collection. Additionally, the paper explains the unreliability of odometry in sandy and off-road conditions due to a higher frequency of slippage occurrences.

A unique idea of discarding wheel odometry data in the Error State Kalman Filter (ESKF) based fusion process when slip is detected is presented in [8]. This effectively eliminates the erroneous wheel odometry data and reduces the pose estimation error drastically. This idea of discarding erroneous wheel odometry data in the fusion process is closely related to the proposed approach of this project.

Source: proposal

3.3 Learning-based slip estimation

Learning-based methods offer robust solutions for slip detection and odometry improvement. One approach uses a Support Vector Machine (SVM) classifier to identify if a robot is immobilised due to wheel slipping [20]. This SVM leverages wheel speed and inertial measurements to categorise robot states like ”immobilized,” ”normal,” or ”unknown.” The classification relies on a feature vector that includes variances of roll rate, pitch rate, and z-axis acceleration, along with the mean of wheel angular acceleration. This SVM-based detection can be combined with a model-based slip detection using an Extended Kalman Filter (EKF). Fusing both methods can enhance accuracy by reducing false negatives.

Beyond classification, unsupervised learning methods, such as Self-Organising Maps, K-means clustering, and auto-encoding, can also categorise slip [21]. These techniques utilise data from IMUs, encoders, and motor currents, avoiding the computational intensity and speed constraints associated with visual data. One of the notable works in RWR odometry is presented in the thesis by Javier Hidalgo-Carrió [22]. This work uses Gaussian Processes (GP) to learn an odometry error model which predicts the slippage error. This error model is then used to correct the odometry and subsequently improve the SLAM performance.

Another data-driven method for slippage compensation integrates Visual-Inertial-Wheel-Odometry (VIWO) [23]. This involves using Gaussian process regression and Long Short-Term Memory (LSTM) networks to compensate for slips in wheel odometry. The compensated wheel odometry is then fused with inertial and visual odometry using a multi-state constraint Kalman filter (MSCKF). A feature confidence estimator further refines pose estimation by ensuring only reliable data is used.

Finally, Gaussian process-based learning can fuse wheel encoder data and IMU data [5]. An Extended Kalman Filter then integrates the learned model's output with IMU data. This approach has demonstrated improved performance over traditional EKF methods without learning methods. This method could be implemented as a continuation of this thesis.

Source: proposal

3.4 Limitations of previous work

Although several methods have been proposed for odometry fusion and slip estimation, they still exhibit certain limitations. The previous works related to RWR odometry are limited, and many of the previous works on sensor fusion for odometry assume continuous wheel-ground contact. This limits the use of such approaches in RWR. Many of the works rely on external systems like motion capture systems or offline data collection for terrain modelling and parameter tuning. These approaches are impractical for real-world applications in unknown or extraterrestrial terrains where such systems are not available [4].

Sensor fusion methods that combine wheel odometry with IMU data often do not handle wheel slippage explicitly. When slippage occurs, the fused estimate can become inaccurate and erroneous wheel odometry data is included without correction [17, 24]. Some approaches attempt to improve accuracy by discarding odometry samples during slippage [8], but this requires reliable slippage detection, which is still a challenging task under varying terrain conditions.

Several learning-based methods, such as those using SVM, CNN, or LSTM networks, have also been proposed for slip estimation and correction [20, 21, 23, 25]. While these approaches show promising results, they often require large amounts of labelled training data and vehicle- or terrain-specific models. This reduces their adaptability across different platforms or terrain types and increases the development time and effort.

In addition, some proposed methods rely on simplified assumptions such as equal angular velocities for wheels on the same side [7] or simultaneous slippage across all wheels [19]. These assumptions do not hold in practical scenarios where the terrain is uneven or when slip conditions vary from wheel to wheel.

Moreover, the increased complexity introduced by advanced filtering techniques like robust Kalman filters, Error State Kalman Filters, or multimodel architectures can hinder real-time performance, especially on resource-constrained robotic platforms [5, 6, 20, 21, 23]. Finally, several works validate their methods only in simulation or on publicly available datasets, without demonstrating performance in real-world conditions. This lack of validation under realistic operating environments raises concerns about their practical implementations [18, 25].

Based on the above-discussed limitations, a new approach to improve the odometry of RWR is discussed in the following section.

Source: proposal

4

Methodology

This chapter contains information on the how part of the problem statement. What techniques are used to estimate the odometry using InEKF-pcv

4.1 Setup

- We have a rimless wheeled rover named coyote which has four wheel joints and a bogie joint between body and the rear axle.

  add the make and model of wheel encoder

- A Mti680(G) [?] series IMU is used to get the linear acceleration and angular rotation of the robot body
- The Motion capture for ground truth data is done by VICON system
- The ROCK framework by DFKI [?] is used to run the Invariant extended Kalman filter library [?] by RossHartley
- The MARS simulator is used for running simulated experiments and the development of the proposed approach
- Multi function hall and the crater hall at DFKI is used to run real world experiments
-

  add the cpu specs of the coyote rover

4.2 Experimental Design

5

Solution

5.1 Proposed algorithm

The proposed approach modifies the existing InEKF library by adding a new update method that uses Potential Contact Velocity vector as a measurement to correct the filter state. As discussed in the background chapter, the PCV vector is calculated using the wheel encoder measurements and compared against the PCV calculated from the filter state. The difference between these two sources of information is used to get the innovation/residual for the update/correction step of the filter.

5.2 Implementation details

6

Evaluation and Results

6.1 Experiment Description

Describe the experiments/evaluation you are performing to analyse your method.

6.2 Experimental Setup

Describe your experimental setup in detail.

6.3 Results

Describe the results of your experiments in detail.

7 Conclusions

7.1 Contributions

7.2 Lessons learned

7.3 Future work

A

Design Details

Your first appendix

B

Parameters

Your second chapter appendix

C

[in the case of using AI-assistants] Description of AI-Generated Content

Describe in detail what content was generated using an AI assistant. Name the AI assistant and how it was used (e.g. which prompts were used, and for which parts of the project).

References

[1] Lucas Agostinho, Nuno Ricardo, Maria Inês Pereira, Antoine Hiolle, and Andry Pinto. A practical survey on visual odometry for autonomous driving in challenging scenarios and conditions. *IEEE Access*, 10, 08 2022.

[2] Ross Hartley, Maani Ghaffari Jadidi, Jessy W. Grizzle, and Ryan M. Eustice. Contact-aided invariant extended kalman filtering for legged robot state estimation. In *Proceedings of Robotics: Science and Systems (RSS)*, Pittsburgh, Pennsylvania, June 2018.

[3] Lauro Ojeda, Daniel Cruz, Giulio Reina, and Johann Borenstein. Current-based slippage detection and odometry correction for mobile robots and planetary rovers. *IEEE Transactions on Robotics*, pages 366–378, Apr 2006.

[4] Genki Yamauchi, Keiji Nagatani, Takeshi Hashimoto, and Kenichi Fujino. Slip-compensated odometry for tracked vehicle on loose and weak slope. *Robomech Journal*, pages 1–11, Dec 2017.

[5] Martin Brossard and Silvère Bonnabel. Learning Wheel Odometry and IMU Errors for Localization. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 291–297, May 2019.

[6] Shounak Das, Cagri Kilic, Ryan Watson, and Jason Gross. A Comparison of Robust Kalman Filters for Improving Wheel-Inertial Odometry in Planetary Rovers. *Proceedings of the 34th International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GNSS+*, pages 2621–2632, Dec 2021.

[7] Fangxu Liu, Xueyuan Li, Shihua Yuan, and Wei Lan. Slip-Aware Motion Estimation for Off-Road Mobile Robots via Multi-Innovation Unscented Kalman Filter. *IEEE Access*, pages 43482–43496, Mar 2020.

[8] Song Xiaobo, Chen Ziming, Du Luyao, and Xiang Kui. Multi-sensor Fusion SLAM Algorithm Considering Wheel Slip. In *2024 30th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*, pages 1–6, Oct 2024.

[9] Jorge De León, Raúl Cebolla, and Antonio Barrientos. A Sensor Fusion Method for Pose Estimation of C-Legged Robots. *Sensors 2020, Vol. 20, Page 6741*, page 6741, Nov 2020.

[10] Khuram Naveed. A Tutorial on Modeling and Control of Slippage in Wheeled Mobile Robots. *arXiv preprint arXiv:2306.14074*, Jun 2023.

[11] Ramon Gonzalez and Karl Iagnemma. Slippage estimation and compensation for planetary exploration rovers. State of the art and future challenges. *Journal of Field Robotics*, pages 564–577, Jun 2018.

[12] Russ Tedrake. *Robotic Manipulation.* Massachusetts Institute of Technology, 2024.

[13] Kevin M. Lynch and Frank C. Park. *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, USA, 1st edition, 2017.

[14] Richard M. Murray, S. Shankar Sastry, and Li Zexiang. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Inc., USA, 1st edition, 1994.

[15] Roy Featherstone. *Rigid Body Dynamics Algorithms*. Springer Publishing Company, Incorporated, 2016.

[16] Orocos. GitHub - orocos orocos_kinematics_dynamics: Orocos Kinematics and Dynamics C++ library. `https://github.com/orocos/orocos_kinematics_dynamics`, 2026. Accessed 08-02-2026.

[17] Sofia Yousuf and Muhammad Bilal Kadri. Sensor fusion of ins, odometer and gps for robot localization. In *2016 IEEE Conference on Systems, Process and Control (ICSPC)*, pages 118–123, Dec 2016.

[18] Mahboubeh Zarei and Robin Chhabra. Sequential Sensor Fusion for Slip Estimation in Mobile Robots. *American Control Conference (ACC) 2024*, pages 2697–2702, Sep 2024.

[19] Kristin Bussmann, Lukas Meyer, Florian Steidle, and Armin Wedler. Slip Modeling and Estimation for a Planetary Exploration Rover: Experimental Results from Mt. Etna. *IEEE International Conference on Intelligent Robots and Systems*, pages 2449–2456, Dec 2018.

[20] Karl Iagnemma and Chris C. Ward. Classification-based wheel slip detection and detector fusion for mobile robots on outdoor terrain. *Autonomous Robots*, pages 33–46, Jan 2009.

[21] Justin Kruger, Arno Rogg, and Ramon Gonzalez. Estimating Wheel Slip of a Planetary Exploration Rover via Unsupervised Machine Learning. *IEEE Aerospace Conference Proceedings*, Mar 2019.

[22] Javier Hidalgo-Carrió. Adaptive localization and mapping for planetary rovers, Jun 2018.

[23] Niraj Reginald, Omar Al-Buraiki, Thanacha Choopojcharoen, Baris Fidan, and Ehsan Hashemi. Visual-Inertial-Wheel Odometry with Slip Compensation and Dynamic Feature Elimination. *Sensors (Basel, Switzerland)*, page 1537, Mar 2025.

[24] Aleksandr Mikov, Alexey Panyov, Vasily Kosyanchuk, and Igor Prikhodko. Sensor Fusion For Land Vehicle Localization Using Inertial MEMS and Odometry. In *2019 IEEE International Symposium on Inertial Sensors and Systems (INERTIAL)*, pages 1–2, Apr 2019.

[25] Michelle Valente, Cyril Joly, and Atnaud De La Fortelle. Deep Sensor Fusion for Real-Time Odometry Estimation. *IEEE International Conference on Intelligent Robots and Systems*, pages 6679–6685, Nov 2019.

[26] Haoyang Ye, Yuying Chen, and Ming Liu. Tightly Coupled 3D Lidar Inertial Odometry and Mapping. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3144–3150, May 2019.

[27] Jo Yung Wong. *Theory of ground vehicles*. John Wiley & Sons, 2022.

[28] Movella. General information — mti-680g rtk gnss/ins. https://mtidocs.movella.com/general-information-3$mti-680-g-rtk-gnss-ins, 2026. Accessed 01-02-2026.

[29] Rock Core Team. rock-robotics.org. https://www.rock-robotics.org/, 2026. Accessed 01-02-2026.

[30] Hartley, Ross. Invariant ekf: C++ library for aided inertial navigation. https://github.com/RossHartley/invariant-ekf, 2026. Accessed 01-02-2026.