Improving the robustness of motion modelling, control and localization for mobile robots in harsh conditions

Dominic Baril (111 103 819)
January 2024

Supervised by Prof. F. Pomerleau
and co-supervised by Prof. P. Giguère

1 Introduction

The field of mobile robotics has made significant advances in the last decade, leading to potentially disruptive innovations in automation for various industries. Autonomous systems are currently mature enough to be functional in controlled and structured operational environments, such as warehouses and urban areas under ideal weather. Uncrewed ground vehicles (UGVs) are proving to be effective solutions to current societal issues related to labor shortage, workplace security and operational efficiency. However, such issues are greater for industries such as agriculture, forestry, defense, mining and search and rescue, which require operation in outdoors, uncontrolled environments. In these cases, systems are subject to a higher spectrum of environmental hazards, such as harsh weather, traction variability and deployment in remote environments. However, as stated by Van Brummelen *et al.* [1], challenges inherent to these conditions remain an open question.

This thesis work aims to extend the proficiency and robustness of autonomous navigation systems to off-road environments and harsh weather. Autonomous navigation can be split into three key components: localization, path planning and path following. This work mainly focuses on path following, with some contributions to localization. For path following, the key problems related to navigating in such environments are the high variability of wheel-to-ground traction and complex vehicle dynamics [2]. For localization, the key problems are related to navigating in Global Navigation Satellite System (GNSS)-denied conditions, low geometrical constraints and dynamic environments [3]. In all, the research question for this work can be stated as follows:

How to increase the robustness of UGV path following and localization for off-road and winter conditions?

A UGV motion model is a key component to compute optimal commands with respect to motion predictions [4] and provide localization prior for localization systems [5]. Thus, this research project is focused on minimizing motion prediction error for models, which is directly correlated with the path following and localization errors. Current approaches for UGV modeling belong to two families: model-based, divided between kinematic and dynamic models, and learning-based, leveraging machine learning and driving data to predict motion. Both kinematic models and learning-based approaches share the advantage that they have a low expertise requirement for deployment and require a training dataset to reduce prediction error, leading to them being the most popular choice. To answer the aforementioned research question, three key issues were identified:

1. **How does UGV behavior differ between concrete and snow-covered terrain navigating? What kinematic model behaves best for both?**
2. **How can we standardize training dataset gathering and improve vehicle slip learning?**
3. **What are the impacts of the boreal forest environment and winter weather on lidar-based localization?**

These key issues guide the scientific contributions that were made through this work. The remainder of this document describes the current scientific production done through this project and the upcoming plan up to thesis submission. More specifically, Section 2 describes the currently submitted and published research work, summarizing contributions and lessons learned for each paper. Afterwards, Section 3 details the remaining research work and Section 3.1 provides a schedule leading to thesis submission. Lastly, Section 4 provides a brief conclusion.

2 Current scientific production

This section describes the current scientific done through this Ph.D. thesis work and collaborations done with other researchers. First, the three articles produced for which I acted as first author. All of these scientific contributions are related to the subproblems stated in Section 1. Then all of the work in which I have participated as co-author is described briefly. Since mobile robotics is a field requiring various expertise and human resources to conduct field deployments, all scientific production presented includes multiple co-authors.

2.1 Articles published and submitted as first author

D. Baril, V. Grondin, S.-P. Deschenes, J. Laconte, M. Vaidis, V. Kubelka, A. Gallant, P. Giguere, and F. Pomerleau, "Evaluation of Skid-Steering Kinematic Models for Subarctic Environments", in *2020 17th Conference on Computer and Robot Vision (CRV)*, IEEE, 2020, pp. 198–205

The first article that I published was submitted to the *Conference on Robots and Vision (CRV)*, in May 2020. This article aims to evaluate the performance of skid-steering mobile robots (SSMRs) kinematic motion models on dry concrete and snow-covered terrain. In this article, we collected a total of 2 km of human driving data to evaluate four kinematic models from the literature. We leverage lidar point cloud registration based on the iterative closest point (ICP) algorithm to generate ground truth localization. This work received the **Best Robot Paper Award** for the CRV conference this year. A seminar was conducted for this work.[1] The resulting contributions are as follows:

1. Validate SSMR kinematic motion models' fitness for a heavier platform on a relatively uniform concrete terrain;
2. Evaluate SSMR kinematic motion models' performance for snow-covered terrain using more than 2 km of trajectories traveled; and
3. Highlight the impact of angular motion on the accuracy of SSMRs kinematic modeling.

The four kinematic models evaluated are the extended differential-drive asymmetrical, the extended differential-drive symmetrical [6], the full linear [7] and the radius of curvature (ROC)-based [8]. We show that models with fewer parameters tend to perform better for angular prediction and models with more parameters perform better for translation prediction, due to their ability to predict non-zero lateral motion. However, once trained, the performance of all models is similar for both terrain types, suggesting that all kinematic models evaluated behave similarly. The largest prediction error occurs when the vehicle's angular velocity is at its maximum, which leads to the highest amount of vehicle slip.

Additionally, training kinematic models with empirical driving data leads to significant prediction error reduction, for both concrete and snow-covered terrain. The relation between training window and prediction error is also studied in this work, clearly showing that models perform best when predicting for the same horizon for which they were trained. We show that for the same commanded angular velocity, observed body angular velocity is higher on snow-covered terrain than on concrete. This phenomenon is due to the high friction caused by the tire deformation occurring during skidding on concrete, compared to soft terrain deformation on snow-covered terrain.

The take-home message for this published paper was that kinematic motion models are adequate for predicting SSMR motion, both on dry concrete and snow-covered terrain, however, they require a training dataset dependent on vehicle and terrain properties. During the experimental work conducted for this paper, we imitated similar work by having a human operator stimulate as many commands as possible, however, this process led to biased command stimulation and forward-only driving and proved to be time-consuming. Since deploying UGVs in off-road environments is a complex endeavor, reducing the time required to generate a motion model that is accurate enough for stable autonomous navigation is key.

[1] https://www.youtube.com/watch?v=FjrgZMmWTNI&t=25s

In this paper, my role was to lead the literature review and select the kinematic models to be evaluated. I also designed the experimental protocol for gathering the navigation dataset, but the dataset was gathered by co-authors. Once the data was collected, I processed it and produced the results that were used for the article. This analysis work was validated by a post-doctoral fellow, who is also a co-author. Lastly, I have led the redaction of the article, with significant support from both supervisors and all co-authors.

D. Baril, S.-P. Deschênes, L. Coupal, C. Goffin, J. Lépine, P. Giguère, and F. Pomerleau, "DRIVE: Data-driven Robot Input Vector Exploration", *ArXiv preprint, submitted to the International Conference on Robotics and Automation (ICRA)*, 2024

My second article as the first author was submitted to the *International Conference on Robotics and Automation (ICRA) 2024*. It is currently under review, with a result expected in January 2024. This article aims to solve the issue discovered in our previous work by standardizing and automating the UGV training dataset gathering task. Thus, we propose Data-driven Robot Input Vector Exploration (DRIVE), a protocol allowing to automatically generate a training dataset for commercial UGVs. We also propose a novel SSMR dynamics-based slip learning model, outperforming similar models. The experimental evaluation for this work totals over 7 km of driving data, conducted through three SSMRs with weights ranging from 75 kg to 470 kg. The deployment environments include indoor tile, snow-covered terrain, dry gravel and an indoor, leveled ice rink. Again, localization is provided by our local mapping framework, based on the ICP algorithm. The code and datasets are all available online and open-source.[2] For this work, the contributions are as follows:

1. DRIVE, a standardized UGV characterization and motion data generation protocol allowing to train motion models on the entire vehicle input space;

2. A novel slip-based UGV motion prediction model, leveraging the accuracy of model-based approaches and the minimal system characterization requirement of learning-based approaches.

As shown in our previously published work [2], a training dataset specific to vehicle and terrain properties is key to reducing motion prediction error. Most work on UGV motion modeling includes training dataset gathering. However, there exist little to no guidelines available to reproduce their work. DRIVE enables automatic UGV input-space characterization and training dataset gathering by sampling uniformly through the input-space. We show that this protocol leads to significantly increased prediction performance when compared to manual driving and radius of curvature (ROC) stimulation approaches. Another key finding is that for all experiments conducted, models reach convergence with a maximum of 46 sec of driving data. This result is of high importance for field robotics operations, which

[2] https://github.com/norlab-ulaval/DRIVE

are costly to deploy and for which UGV battery conservation is critical [3]. It also enables model re-training when driving conditions change drastically.

Furthermore, we propose a novel slip learning model that outperforms similar learning models for SSMRs navigation in off-road terrain. Contrarily to our previous work [2], we rely on additive slip, which facilitates slip learning since slip is computed by subtracting the commanded body velocity to the observed body velocity, similarly as proposed by Seegmiller *et al.* [10]. For slip learning, we rely on Bayesian linear regression (BLR), which enables fast prediction and training, which is ideal for real-time UGV deployment [11]. We show that by leveraging dynamics-aware basis functions for BLR, we have significant slip prediction performance improvement over the baseline BLR learning approach, which learns vehicle acceleration. The operational limit for our model is reached on the ice rink experiments, where extreme UGV slip severely impacts motion.

The biggest lesson learned is that DRIVE enables training dataset gathering for model convergence in 46 sec, for our slip-BLR model, which outperforms similar learning approaches. This protocol is interesting for field operations as it enables users to generate an accurate motion prediction model for any new UGV or environment configuration, effectively saving a lot of deployment resources and battery usage. In future work, we want to investigate dynamic modeling on the ice rink experiment to see if richer modeling approaches can improve prediction accuracy under dynamically complex driving conditions. This experimental dataset is also ideal for investigating the limit of adaptive modeling approaches by simulating a robot instantly changing terrain type.

In this paper, I designed the DRIVE protocol and iteratively improved over two years of experimental work. I have received significant support from colleagues at the lab for testing the protocol, including all co-authors. Once the protocol was ready, I led the dataset gathering work for all robots in all environments, always with support from co-authors. Afterwards, I was responsible for processing the dataset and implementing the models which we compared our contributions to, as well as completing the literature review, with high-level support from both supervisors. Lastly, I have completed most of the article writing and figures production autonomously as it is the third article I have completed as first author, with support from co-authors to provide reviews and finish figures.

D. Baril, S.-P. Deschênes, O. Gamache, M. Vaidis, D. LaRocque, J. Laconte, V. Kubelka, P. Giguère, and F. Pomerleau, "Kilometer-scale autonomous navigation in subarctic forests: challenges and lessons learned", *Field Robotics*, vol. 2, no. 1, pp. 1628–1660, 2022

The third article that concludes the list of articles submitted as first author was submitted and published in the *Field Robotics* journal. In this paper, we leverage the models evaluated in our previous work with our lidar-based localization and mapping system to create the Weather-Invariant Lidar-based Navigation (WILN) autonomous navigation system. With this system, we conducted 18.8 km of autonomous driving in a boreal forest, during wintertime, a first in the literature. For this work, we were awarded the **Relève Etoile Louis-Berlinguet Award** from the Fond de recherche du Québec en Natures et Technologies (FRQNT). We leverage the data acquired to produce a field report documenting the impact of the boreal forest biome and winter weather on lidar-based autonomous navigation. The dataset gathered in the field is available online and open-source.[3] A seminar was also given for this work.[4] The specific contributions for this article are as follows:

1. A comprehensive study of the impact of the boreal forest biome on lidar- and GNSS-based localization and autonomous navigation;
2. An overview of the impact of snow accumulation on the reliability of lidar-based localization over multiple days; and
3. A description of the WILN system, designed to enable wintertime autonomous navigation in a boreal forest.

While we have validated lidar-based localization and mapping are suitable for autonomous navigation in the test environment, we observed particular phenomenons that can lead to navigation failure, most of which are related to lidar localization. First, when navigating in a corridor-like forest trail, under dense vegetation, the low longitudinal geometric constraints lead to localization instability, which caused the UGV to crash with vegetation on some occasions. Secondly, the snow accumulation causes the environment to change significantly, which is critical in areas where few man-made structures are present to support lidar-based localization. For those issues, we have documented the impact on our localization system and highlighted the challenges to be solved to enable true year-long autonomy in boreal forests. An analysis of the quality of GNSS signal under the dense vegetation of boreal forests was also conducted, showing that it is unsuitable for navigation in tight forest trails, but functional on larger forest paths.

Since it was not the main focus of this work, an operator drove periodically with a snowmobile over the paths navigated by the UGV to tap the snow and enable large-scale navigation. We still investigated the impact of the environment on our selected kinematic path following controller [12]. Under these conditions, we show that the path following error is strongly correlated with reference path curvature, meaning that the most likely cause for path following failure is tight turning in forest trails. Qualitatively, we also show that deep snow navigation is complex for SSMRs since it significantly reduces the vehicle's ability to turn.

For this work, the main take-home message is that kinematic motion modeling and lidar-based localization and mapping are suitable for wintertime navigation in boreal forests,

[3] https://github.com/norlab-ulaval/Norlab_wiki/wiki/Kilometer-scale-autonomous-navigation-in-subarctic-forests:-challenges-and-lessons-learned

[4] https://www.youtube.com/watch?v=VWzKZvtnInA

but that challenges remain to be solved for true year-long autonomy. First, localization robustness to areas with corridor-like environments and dynamic environments should be addressed, by adapting the reference map and adding localization constraints. Secondly, kinematic path following controllers are suitable for condensed or shallow snow but fail in deep snow. Thirdly, energy consumption estimation and prediction are complex in cold weather, however, they are key to maximizing UGV usage and preventing breakdowns in remote areas. While those are the main challenges observed, multiple more lessons learned are highlighted in this work.

For this article, I have co-developed the WILN system with a co-author. I have then led the dataset gathering work, with significant support from all co-authors as this was the most labor-intensive task. Once the dataset was gathered, I coordinated the data processing, delegating the work to co-authors based on their expertise. Afterwards, I analyzed the data and produced initial results with significant support from both supervisors. For the paper production, I was responsible for writing most of the article, with support from co-authors for sections specific to their expertise. Final figures were produced by a post-doctoral co-author. Lastly, I have benefited from extensive reviews for all co-authors, significantly improving the quality of the article.

2.2 Articles published and submitted as co-author

S.-P. Deschenes, D. Baril, V. Kubelka, P. Giguere, and F. Pomerleau, "Lidar Scan Registration Robust to Extreme Motions", in *2021 18th Conference on Robots and Vision (CRV)*, IEEE, 2021, pp. 17–24: This article improves point cloud registration accuracy under aggressive motion by accounting for the error related to point cloud deskewing. Skewing is lidar scan error originating from the assumption that the sensor is static during the sweep. Under aggressive motion, characterized by high body velocities and acceleration, this assumption is broken, leading to a high scan error. For this work, I built the experimental rig and supported the experimental work. I have also participated in generating the results and for redaction.

T. Rouček, M. Pecka, P. Čížek, T. Petříček, J. Bayer, V. Šalanský, T. Azayev, D. Heřt, M. Petrlík, T. Báča, V. Spurný, V. Krátký, P. Petráček, D. Baril, M. Vaidis, V. Kubelka, F. Pomerleau, J. Faigl, K. Zimmermann, M. Saska, T. Svoboda, and T. Krajník, "System for multi-robotic exploration of underground environments CTU-CRAS-NORLAB in the DARPA subterranean challenge", *Field Robotics*, vol. 2, no. 1, pp. 1779–1818, Mar. 2022: This article describes the system that was built for the CTU-CRAS-Norlab team at the DARPA Subterranean Challenge Urban Circuit. This is the biggest robotics competition in the world and our team managed to class third overall and first among self-funded teams. The goal was to deploy a heterogeneous robot fleet in an unknown underground environment and to detect and localize artifacts within 5 cm of their true position. For this work, I co-built the UGV which was contributed by Norlab to the competition and participated in its integration to the complete fleet. I have also post-processed the navigation data after the competition and generated the 3D maps that were used as figures in the article.

C. Courcelle, D. Baril, F. Pomerleau, and J. Laconte, "On the Importance of Quantifying Visibility for Autonomous Vehicles Under Extreme Precipitation", in *Towards Human-Vehicle Harmonization*, De Gruyter, 2023, pp. 239–250: This article is a study of the impact of extreme precipitation on lidar-based localization, more precisely point cloud registration with the ICP algorithm. A novel lidar scan obstruction metric is proposed, enabling proper comparison and analysis of the impact of harsh weather. We rely on data collected with our own robot platforms and the Canadian Adverse Driving Conditions (CADC) Dataset [16] to study the relation between this metric and lidar-based localization error, with GNSS as ground truth. The results suggest that extreme precipitation has a significant impact on the UGV's ability to localize, however not enough data is available under such conditions. Gathering more data under extreme precipitation remains an open problem. In this work, I participated as an advisor on data analysis and experiments. I have also helped with writing the paper.

S.-P. Deschênes, D. Baril, M. Boxan, J. Laconte, P. Giguère, and F. Pomerleau, "Saturation-Aware Angular Velocity Estimation: Extending the Robustness of SLAM to Aggressive Motions", *ArXiv preprint, submitted to the International Conference on Robotics and Automation (ICRA)*, 2024: This article extends on the previous work on point cloud registration under aggressive motion [13]. This time, the use case is the sensor rig rolling down a steep hill, leading to gyroscope measurement saturation. Since such measurements are key to providing a prior for the ICP algorithm to converge, gyroscope saturation leads to localization failure. We leverage Gyro-free (GF)-Inertial Navigation System (INS) theory [18] to estimate body angular velocity under gyroscope saturation and show that lidar localization and mapping then becomes robust to aggressive motions. For this work, I was responsible for designing the sensor rig and co-responsible for the experimental work. I was responsible for finalizing most of the figures and participated in writing the text.

3 Work left before submitting the thesis

The three articles that I have written as the first author presented in Section 2 will represent the majority of the thesis by articles. A last chapter will be written in the thesis in which we will study the impact of the DRIVE UGV characterization protocol on path following performance under a realistic deployment setting. These results will be appended to the thesis to complete it. A schedule for the remaining work and a Gantt chart are presented in Section 3.1

We have developed a predictive control library in the context of industrial demonstrations for this project.[5] Such controllers are based on the Model Predictive Control (MPC) algorithm, enabling high-speed path following with low tracking error. In our case, we have implemented a simplified version of the slip-aware MPC controlled proposed by Hewing *et al.* [21]. Since the motion model acts as an interchangeable module for such controllers, the prediction accuracy improvements reached through our work on the DRIVE protocol will most likely improve the MPC path following performance. A last interesting result would be to quantify the path following error improvement reached through training the slip learning model with a DRIVE-gathered training dataset instead of manual driving. Typically, human driving data is limited in driving velocity since it is the first time for which the vehicle navigated the target trajectory. Our hypothesis is that by training the model through DRIVE, path following performance at high speed is improved as soon as the first time the UGV executes the desired trajectory.

3.1 Schedule for upcoming work

In this section, a plan for the upcoming steps for this thesis is presented. A Gantt chart is shown in Figure 1, providing visual support for the planning presented. The remaining work is split into three main categories, namely thesis writing, *ICRA* paper finalization and thesis review and defense. The thesis will be written via article insertion, where the articles selected are presented in Section 2.1. This task will require appending every article which will represent a distinct thesis chapter. Mathematical notation will be modified to be uniform for all articles and facilitate thesis reading and review. Then, I will write on writing the final chapter, which will present the results of the experiments described in Section 3. My plan is to invest half days on thesis writing and the other half days on final experiments to maintain motivation through the writing process. Final chapter writing will be done after those last experiments are conducted.

Furthermore, since our work on DRIVE is undergoing the peer review process, we will receive the results in January 2024. Paper improvements will need to be prioritized at this time for paper re-submission which will most likely be due at the end of January 2024. Then, the official paper presentation will be done during the three weeks leading to the conference, including a dry run in the laboratory to fetch initial feedback and maximize the presentation quality. Then, from May 14th to 17th 2024, I will attend the conference to present our paper on-site. In the case the paper is rejected at the conference, the reviews will still be into account and the paper will be improved and re-submitted to an alternative venue. In this case, slight changes to the planning presented in this section will be made to ensure thesis defense still happens in August 2024.

[5] https://github.com/norlab-ulaval/norlab_controllers

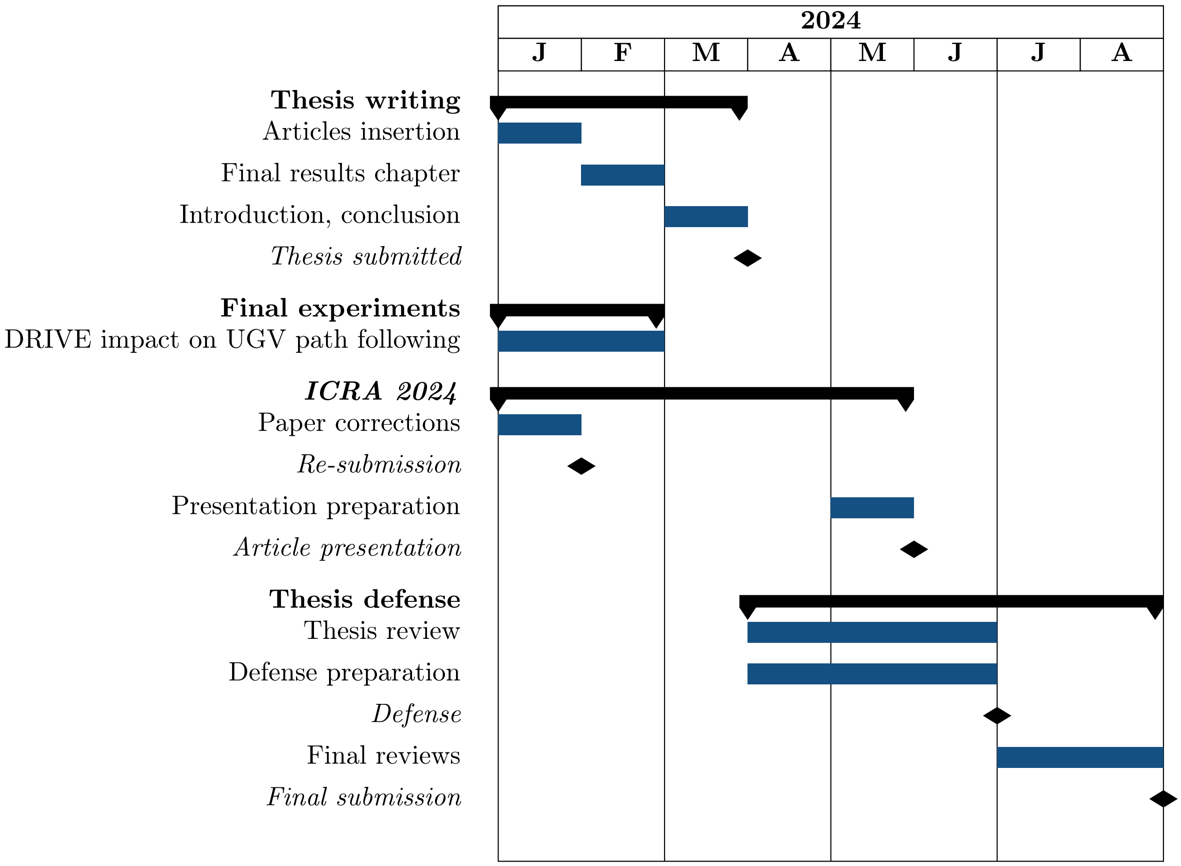


Figure 1: Gantt chart of the remaining tasks left to complete this thesis and planning for their completion leading to August 2024.

Lastly, once the thesis is submitted, we will follow standard academic delays defined by Université Laval for review, corrections and defense. A total of five months is scheduled between thesis submission and graduation, up to August 2024. This will allow time to validate the initial submission, have the committee review it, defend the thesis and produce the final version. With around a month of contingency, this planning will conclude the work done on this thesis.

4 Conclusion

In this document, we have described the current scientific work completed in the realization of this thesis. In total, I have published one conference paper and one journal paper and submitted another conference paper as the first author. Meanwhile, I have taken part in an additional four papers as a co-author. This scientific production will represent the majority of the content of my article insertion thesis.

For the last chapter of the thesis, I have identified three last experiments which would be relatively simple to complete considering the amount of work already completed. Providing these results are conclusive, they will be added to the final chapter of my thesis. The initial thesis submission is scheduled for March 2024, leading to the defense in August 2024 according to Université Laval standards.

References

[1] J. Van Brummelen, M. O'Brien, D. Gruyer, and H. Najjaran, "Autonomous vehicle perception: The technology of today and tomorrow", *Transportation Research Part C: Emerging Technologies*, vol. 89, no. July 2017, pp. 384–406, 2018.

[2] D. Baril, V. Grondin, S.-P. Deschenes, *et al.*, "Evaluation of Skid-Steering Kinematic Models for Subarctic Environments", in *2020 17th Conference on Computer and Robot Vision (CRV)*, IEEE, 2020, pp. 198–205.

[3] D. Baril, S.-P. Deschênes, O. Gamache, *et al.*, "Kilometer-scale autonomous navigation in subarctic forests: challenges and lessons learned", *Field Robotics*, vol. 2, no. 1, pp. 1628–1660, 2022.

[4] L. Brunke, M. Greeff, A. W. Hall, *et al.*, "Safe Learning in Robotics: From Learning-Based Control to Safe Reinforcement Learning", *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, no. 1, pp. 411–444, 2022.

[5] F. Dümbgen, C. Holmes, and T. D. Barfoot, "Safe and Smooth: Certified Continuous-Time Range-Only Localization", *IEEE Robotics and Automation Letters (RA-L)*, vol. 8, no. 2, pp. 1117–1124, 2023.

[6] A. Mandow, J. L. Martinez, J. Morales, J. L. Blanco, A. Garcia-Cerezo, and J. Gonzalez, "Experimental kinematics for wheeled skid-steer mobile robots", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007, pp. 1222–1227.

[7] G. Anousaki and K. Kyriakopoulos, "A dead-reckoning scheme for skid-steered vehicles in outdoor environments", in *IEEE International Conference on Robotics and Automation (ICRA)*, 2004, 580–585 Vol.1.

[8] T. Wang, Y. Wu, J. Liang, C. Han, J. Chen, and Q. Zhao, "Analysis and Experimental Kinematics of a Skid-Steering Wheeled Robot Based on a Laser Scanner Sensor", *Sensors*, vol. 15, no. 5, pp. 9681–9702, 2015.

[9] D. Baril, S.-P. Deschênes, L. Coupal, *et al.*, "DRIVE: Data-driven Robot Input Vector Exploration", *ArXiv preprint, submitted to the International Conference on Robotics and Automation (ICRA)*, 2024.

[10] N. Seegmiller and A. Kelly, "Enhanced 3D Kinematic Modeling of Wheeled Mobile Robots", in *Robotics: Science and Systems X (RSS)*, Robotics: Science and Systems Foundation, 2014.

[11] C. D. McKinnon and A. P. Schoellig, "Learn Fast, Forget Slow: Safe Predictive Learning Control for Systems With Unknown and Changing Dynamics Performing Repetitive Tasks", *IEEE Robotics and Automation Letters (RA-L)*, vol. 4, no. 2, pp. 2180–2187, 2019.

[12] G. Huskić, S. Buck, and A. Zell, "A Simple and Efficient Path Following Algorithm for Wheeled Mobile Robots", in *Advances in Intelligent Systems and Computing*, vol. 531, 2017, pp. 375–387.

[13] S.-P. Deschenes, D. Baril, V. Kubelka, P. Giguere, and F. Pomerleau, "Lidar Scan Registration Robust to Extreme Motions", in *2021 18th Conference on Robots and Vision (CRV)*, IEEE, 2021, pp. 17–24.

[14] T. Rouček, M. Pecka, P. Čížek, *et al.*, "System for multi-robotic exploration of underground environments CTU-CRAS-NORLAB in the DARPA subterranean challenge", *Field Robotics*, vol. 2, no. 1, pp. 1779–1818, Mar. 2022.

[15] C. Courcelle, D. Baril, F. Pomerleau, and J. Laconte, "On the Importance of Quantifying Visibility for Autonomous Vehicles Under Extreme Precipitation", in *Towards Human-Vehicle Harmonization*, De Gruyter, 2023, pp. 239–250.

[16] M. Pitropov, D. E. Garcia, J. Rebello, *et al.*, "Canadian Adverse Driving Conditions dataset", *International Journal of Robotics Research*, vol. 40, no. 4-5, pp. 681–690, 2021.

[17] S.-P. Deschênes, D. Baril, M. Boxan, J. Laconte, P. Giguère, and F. Pomerleau, "Saturation-Aware Angular Velocity Estimation: Extending the Robustness of SLAM to Aggressive Motions", *ArXiv preprint, submitted to the International Conference on Robotics and Automation (ICRA)*, 2024.

[18] M. Pachter, T. C. Welker, and R. E. Huffman, "Gyro-free INS Theory", *NAVIGATION*, vol. 60, no. 2, pp. 85–96, 2013.

[19] F. Djeumou, J. Y. Goh, U. Topcu, and A. Balachandran, "Autonomous Drifting with 3 Minutes of Data via Learned Tire Models", in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 968–974.

[20] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Information-Theoretic Model Predictive Control: Theory and Applications to Autonomous Driving", *IEEE Transactions on Robotics (T-RO)*, vol. 34, no. 6, pp. 1603–1622, 2018.

[21] L. Hewing, J. Kabzan, and M. N. Zeilinger, "Cautious Model Predictive Control Using Gaussian Process Regression", *IEEE Transactions on Control Systems Technology (T-CST)*, vol. 28, no. 6, pp. 2736–2743, 2020.

2020 17th Conference on Computer and Robot Vision (CRV)

Evaluation of Skid-Steering Kinematic Models for Subarctic Environments

Dominic Baril, Vincent Grondin, Simon-Pierre Deschênes, Johann Laconte, Maxime Vaidis,
Vladimír Kubelka, André Gallant, Philippe Giguère, François Pomerleau
Northern Robotics Laboratory, Université Laval, Québec City, Québec, Canada
{*dominic.baril, francois.pomerleau*}*@norlab.ulaval.ca*

***Abstract*—In subarctic and arctic areas, large and heavy skid-steered robots are preferred for their robustness and ability to operate on difficult terrain. State estimation, motion control and path planning for these robots rely on accurate odometry models based on wheel velocities. However, the state-of-the-art odometry models for skid-steer mobile robots (SSMRs) have usually been tested on relatively lightweight platforms. In this paper, we focus on how these models perform when deployed on a large and heavy (590 kg) SSMR. We collected more than 2 km of data on both snow and concrete. We compare the ideal differential-drive, extended differential-drive, radius-of-curvature-based, and full linear kinematic models commonly deployed for SSMRs. Each of the models is fine-tuned by searching their optimal parameters on both snow and concrete. We then discuss the relationship between the parameters, the model tuning, and the final accuracy of the models.**

***Keywords*-mobile robots, skid-steering vehicles, robot kinematics, winter**

Figure 1: The wheeled skid-steer platform, a *Warthog* from *Clearpath*, on a snow-covered testing site. Due to the steering type and the snow on the ground, the robot is very prone to skidding and slippage, making the control and state estimation of this system quite challenging.

I. Introduction

Locomotion models are essential for many functionalities in mobile robots. For instance, the prediction step in Bayesian filters used in the state estimation for localization purposes heavily relies on such models. They are also a crucial component for path planners, which use them to identify the action sequence the vehicle needs to execute to reach a specific goal state. More importantly for our work, locomotion models can be employed to improve the accuracy of motion controllers. For instance, knowledge of the locomotion model enables high-speed path following for SSMRs [1] and the use of model-predictive control algorithms [2]. Consequently, having access to a precise and robust locomotion model is a key component of autonomy and safety in mobile robotics.

Different steering methods have been developed for a number of wheel geometries, a popular example being Ackerman [3]. However, these wheel geometries have inherent kinematic constraints, such as a minimum turn radius. Alternatively, the SSMR locomotion type is designed specifically to alleviate these constraints and offers high maneuverability including zero-radius turning. The SSMR model operates with a set of wheels or tracks on each side of the robot, typically mechanically linked such that they have the same rotational velocity. The difference in velocities between the left and right wheels translates into a rotational motion of the body of the robot, much like a differential-drive system. An example of a wheeled SSMR, used in this work, is shown in Figure 1. Wheeled skid-steering locomotion systems have proven to be suitable for driving at higher speeds on varying terrain [1]. The simplicity and robustness of the mechanical design, which includes no additional steering system, make them relatively cheap and dependable for outdoor deployments [1].

Kinematic models, which define the relationship between the wheel velocities and the robot velocity, are popular in the literature due to their simplicity and robustness to inaccurate parameter estimates [4]. However, the inherent slippage and skidding of SSMRs render the motion difficult to predict accurately through modeling.

In this work, we propose an experimental comparison of kinematic models applied to a heavy SSMR. A special emphasis has been placed on operating on snow-covered terrain. Because of several factors, such as uneven and unpredictable ground interaction forces, SSMR motion on snow-covered terrain can be particularly difficult to model. Modeling in these snowy conditions has also been little explored. Moreover, heavy platforms, such as ours at 590 kg, may perform differently than lighter platforms, even on uniform terrains such as pavement or concrete. Indeed, to the best of our knowledge, this is the first work to study the kinematic modeling of SSMRs above 120 kg.

978-1-7281-9891-0/20/$31.00 ©2020 IEEE
DOI 10.1109/CRV50864.2020.00034

In short, the main contribution of this work is an experimental investigation of five kinematic models on challenging terrains in order to:

1) validate their fitness for a heavier platform on a relatively uniform concrete terrain;
2) evaluate their performance for snow-covered terrain using more than $2\,\mathrm{km}$ of trajectories traveled; and
3) highlight the impact of angular motion on the accuracy of SSMRs kinematic modeling.

II. Related Work

As the motion of SSMRs has been heavily studied in the literature, various kinematic models have been proven accurate to describe their motion. However, the validation of these models has been made with rather light-weight robots, while larger and heavier robots make a more suitable choice for deployments in adverse conditions, such as snow-covered terrain because of their typically greater payload. The question of whether these models can or cannot be transferred to such heavy robots remains open.

To our knowledge, relatively few robotic deployments have been conducted in snowy environments. Apostolopoulos *et al.* [5] deployed the Nomad rover, a $725\,\mathrm{kg}$ four-wheel drive (4WD) gasoline-powered vehicle, which achieved the first autonomous discoveries of Antarctic meteorites. Ray *et al.* [6] deployed the Cool Robot, a $61\,\mathrm{kg}$ solar-powered robot that drove over $500\,\mathrm{km}$ in Antarctica. Gifford *et al.* [7] deployed MARVIN I and MARVIN II, $720\,\mathrm{kg}$ diesel-powered tracked rovers, which were used to conduct seismic and radar remote sensing of ice sheets in polar regions. Lever *et al.* [8] deployed the Yeti, an $81\,\mathrm{kg}$ 4WD electric-powered rover in Antarctica and Greenland. The platform was used to conduct autonomous ground-penetrating radar (GPR) surveys in polar regions. Paton *et al.* [9] used a Clearpath Robotics Grizzly unmanned ground vehicle (UGV) ($660\,\mathrm{kg}$) to perform autonomous route-following in unstructured and outdoor environments. They demonstrated the robustness of the algorithms through extensive field deployments spanning over $26\,\mathrm{km}$. However, autonomous route-following in deep snow provided unsatisfactory results.

It can be seen that most deployments on snow-covered terrains were performed with relatively heavy robots. None of the aforementioned work on autonomous rover deployment in snow have extensively studied SSMRs motion modeling in snow-covered terrain.

Because of the inherent slippage and skidding of SSMRs, straightforward models that assume pure rolling and no slippage are not accurate enough to describe their motion [10]. Mandow *et al.* [10] thus proposed an extension of the ideal differential-drive model, in which each set of wheels rotate around their own instantaneous center of rotation (ICR). Importantly, both ICRs are assumed to be constant for a given terrain. They also included additional parameters to take slippage into account. All parameters for this model are identified offline empirically. They validated their model with the Pioneer 3-AT robot, a $23.6\,\mathrm{kg}$ 4WD skid-steer platform, on asphalt using three different sets of tires.

To produce online estimates of the wheel and SSMRs ICRs, Pentzer *et al.* [11] tracked them individually using an extended Kalman filter (EKF), through the inclusion of position and heading measurements. They validated their algorithm on a $118\,\mathrm{kg}$ skid-steer robot. Moosavian *et al.* [12] and Wang *et al.* [13] proposed an experimentally-derived relationship between the radii of curvature and the amount of slippage for SSMRs motion. Wang *et al.* [13] validated their approach on a wheeled SSMR, which is a Pioneer 3-AT.

Alternatively, Anousaki *et al.* [14] have proposed a general linear model. This model does not take into account any physical parameters of the robot and all parameters are identified offline empirically. This model was validated on a Pioneer 2-AT robot, weighing $23.6\,\mathrm{kg}$.

Rabiee *et al.* [4] proposed a physically interpretable friction-based kinematic model, which accounts for slippage and skidding at the wheel level. This approach uses parameters of a dynamic friction model which are identified empirically and offline. The authors used a Clearpath Robotics Jackal platform ($16\,\mathrm{kg}$) for experimental validation.

As can be seen, all of the aforementioned models are tested on relatively light platforms. However, many winter field deployments of mobile robots, such as the ones mentioned above, use heavier platforms. Indeed, these larger and more powerful platforms are generally employed to allow for heavier payloads.

Thus, this work aims to experimentally validate the motion prediction accuracy of five kinematic models on snowy terrain, with a heavy skid-steer mobile robot.

III. Kinematic modeling of skid-steer motion

Kinematic models for SSMRs aim to describe the speed of the vehicle's local frame by using two inputs: the angular velocity of the left wheel ω_l and of the right wheel ω_r. Kinematic models do not take into account the acceleration. Direct kinematics for the vehicle $\dot{\boldsymbol{x}}$ on the plane (i.e., in 2D) can be stated as follows:

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{v} \\ \omega \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} = j(\omega_l, \omega_r), \tag{1}$$

where $j(\cdot)$ is the kinematic model linking the inputs to the vehicle's translational velocity $\boldsymbol{v}$ and angular velocity ω, as shown in Figure 2. The estimation of the kinematic states $\hat{\dot{\boldsymbol{x}}}$ can be computed from sensor measurements $\boldsymbol{y}$ and a Jacobian $\boldsymbol{J}$ expressed as a function of fixed parameters $\boldsymbol{k}$, such that

$$\hat{\dot{\boldsymbol{x}}} = \boldsymbol{J}(\boldsymbol{k})\boldsymbol{y}, \tag{2}$$

with $\boldsymbol{y} = [\omega_l, \omega_r]^T$. Based on this relation, we will define different models only by expanding the matrix $\boldsymbol{J}$ and its associated vector of parameters $\boldsymbol{k}$.

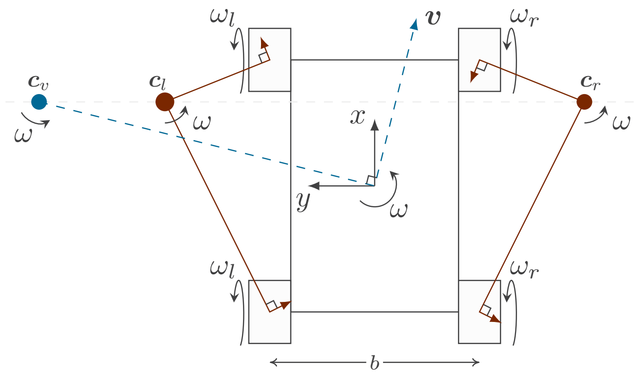


Figure 2: Diagram of a skid-steer vehicle. The instantaneous centers of rotation of the body (in blue) and tracks (in red) are shown as $\boldsymbol{c}_v$, $\boldsymbol{c}_l$ and $\boldsymbol{c}_r$.

A. *Ideal differential-drive*

The simplest model that could be used to predict the motion of an SSMR is an ideal differential-drive model expressed as

$$\boldsymbol{J} = r \begin{bmatrix} 1/2 & 1/2 \\ 0 & 0 \\ -1/b & 1/b \end{bmatrix}, \tag{3}$$

where r is the radius of the wheels and b is the width of the vehicle, as shown in Figure 2.

This model works well for two-wheeled mobile robots that have an ICR aligned with the center of each wheel. However, this model may be inefficient at predicting skid-steering behavior because it assumes that there is no lateral skidding or longitudinal slipping. Nevertheless, it is easy to implement and only depends on the measurable properties of the robot such as the wheel radii and the width of the robot.

B. *Extended differential-drive*

The lack of skid modeling from the ideal differential-drive model pushed Mandow *et al.* [10] to propose the extended differential-drive model for SSMRs. This model states that each set of wheels has a separate ICR, that lies on a line parallel to the local y-axis also containing the ICR of the vehicle body. This is shown via the red lines in Figure 2. This model assumes that the position of each ICR is terrain-dependant, but constant if the terrain remains constant. The two sets of wheels are also assumed to have the same angular velocity ω as the vehicle's body in the horizontal plane. If one has access to the true state $\dot{\boldsymbol{x}}$, this allows the relation between the ICRs positions and the translational and rotational velocities to be geometrically identified, as expressed by

$$\boldsymbol{c}_v(\dot{\boldsymbol{x}}) = \begin{bmatrix} x_v \\ y_v \end{bmatrix} = \frac{1}{\omega} \begin{bmatrix} -v_y \\ v_x \end{bmatrix} \tag{4}$$

$$\boldsymbol{c}_l(\dot{\boldsymbol{x}}, \omega_l) = \begin{bmatrix} x_v \\ y_l \end{bmatrix} = \frac{1}{\omega} \begin{bmatrix} -v_y \\ \alpha_l(r\omega_l - v_x) \end{bmatrix} \tag{5}$$

$$\boldsymbol{c}_r(\dot{\boldsymbol{x}}, \omega_r) = \begin{bmatrix} x_v \\ y_r \end{bmatrix} = \frac{1}{\omega} \begin{bmatrix} -v_y \\ \alpha_r(r\omega_r - v_x) \end{bmatrix}, \tag{6}$$

where α_l, $\alpha_r \in [0, 1]$ are slip parameters to take into account the mechanical characteristics of the wheels [10]. Since we do not have access to $\dot{\boldsymbol{x}}$, as we aim at estimating it, we can use (4)-(6) to express the Jacobian of the extended differential-drive kinematic model in terms of ICRs coordinates, such that

$$\boldsymbol{J}(\alpha_r, \alpha_l, x_v, y_r, y_l) = \frac{r}{y_l - y_r} \begin{bmatrix} -y_r & y_l \\ x_v & -x_v \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \alpha_l & 0 \\ 0 & \alpha_r \end{bmatrix}. \tag{7}$$

For a symmetric robot, we can simplify the model by making the assumptions that the ICRs are symmetric concerning the center of the robot (i.e., $y_0 = y_l = -y_r$ and $x_v = 0$) and that each set of wheels have the same slip parameter (i.e., $\alpha = \alpha_l = \alpha_r$). This symmetric extended differential-drive model will have a Jacobian in the form of

$$\boldsymbol{J}(\alpha, y_0) = \frac{r\alpha}{2y_0} \begin{bmatrix} y_0 & y_0 \\ 0 & 0 \\ -1 & 1 \end{bmatrix} = r\alpha \begin{bmatrix} 1/2 & 1/2 \\ 0 & 0 \\ -1/\hat{b} & 1/\hat{b} \end{bmatrix} \tag{8}$$

$$= \boldsymbol{J}(\alpha, \hat{b}) \quad \text{with } \hat{b} = 2y_0. \tag{9}$$

In this case, the model only has two parameters to train, a slip parameter α and an estimated virtual width of the vehicle $\hat{b}$. As with the ideal differential-drive model, the symmetric extended differential-drive model from Equation (9) still assumes that there is no lateral skidding (i.e., $v_y = 0$), but it is capable of modeling longitudinal slipping and loss of energy while steering. In this work, both the five-parameter and the symmetric two-parameter extended differential drive models are examined.

C. *ROC-based*

Wang *et al.* [13] experimented with SSMR to find a relation between slippage and the radius of curvature (ROC) of the motion. Looking at the previous model from Equation (8), since $v_y = 0$, it can be seen that the instantaneous radius of curvature of the robot is

$$R = \frac{v_x}{\omega} = \left| \frac{\omega_r + \omega_l}{\omega_r - \omega_l} \right| y_0 = \lambda y_0, \tag{10}$$

where λ is the time-varying path curvature variable. Through experiments, Wang *et al.* [13] identified the following relation between y_0 and λ with

$$y_0 = \frac{b}{2}\left(1 + \frac{\beta_1}{1 + \beta_2\sqrt{\lambda}}\right), \tag{11}$$

where β_1 and β_2 are parameters trained experimentally, giving the new Jacobian $\boldsymbol{J}(\alpha, \beta_1, \beta_2)$ following Equation (8) and Equation (11). Since λ is time-varying, the model adapts as a function of the ROC, in contrast with the other models. However, this model still does not address lateral skidding.

D. Full linear

Anousaki *et al.* [14] proposed a general linear model to account for some of the system's uncertainty and asymmetry inherent to SSMR expressed as

$$\boldsymbol{J}(\gamma_{11}, \cdots, \gamma_{32}) = \begin{bmatrix} \gamma_{11} & \gamma_{12} \\ \gamma_{21} & \gamma_{22} \\ \gamma_{31} & \gamma_{32} \end{bmatrix}, \tag{12}$$

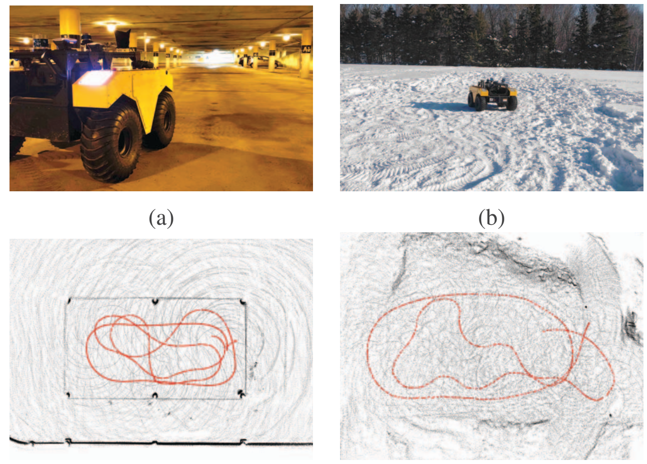
where γ_{ij} are the linear coefficients to be trained, leading to a model with six parameters. Unlike the previous models, this model requires no *a priori* knowledge of the system, and relies entirely on parameters estimated through training.

IV. Experimental setup

To compare the presented models, we collected data using a Clearpath Warthog UGV, as shown in Figure 1. Weighing $590\,\mathrm{kg}$, its dimensions are $1.52\,\mathrm{m} \times 1.38\,\mathrm{m} \times 0.83\,\mathrm{m}$ and can reach a top speed of $18\,\mathrm{km/h}$. As the robot has a skid-steering locomotion system, the wheels on each side of the robot are mechanically linked to a single motor. The robot is also equipped with a differential suspension, stabilizing the sensors and improving wheel-ground contact. A Robosense RS-32 lidar located at the front of the platform is used for localization. During the experiments, the lidar produced point clouds at $10\,\mathrm{Hz}$, the wheel velocity commands are sent at $20\,\mathrm{Hz}$ and the Inertial Measurement Unit (IMU) returns readings at $400\,\mathrm{Hz}$. The recorded data is then used for generating the ground truth trajectory using the Iterative-Closest-Point (ICP) algorithm. Importantly, a different trajectory was recorded for model training and validation.

We tested the models on two different types of surface: a flat concrete surface and a snow-covered terrain. As can be seen in Figure 3a and Figure 3b, these environments present radically different physical properties. In the first one, the robot drives on flat and dry concrete in an underground parking lot. Due to the high friction coefficient and the hardness of the ground, the skid-steering motion is induced by wheel deformation. Furthermore, the stick-slip phenomenon introduces additional unmodeled noise.

On the other hand, the snow-covered terrain is soft and exhibits low friction, meaning that the skid-steering motion is induced by terrain deformation. Additionally, the terrain



(a) (b) (c) (d)

Figure 3: Different road surface conditions used in experiments. (a) Underground parking lot with a dry concrete surface. (b) Snow-covered terrain. (c) The resulting map of the underground parking lot with one of the robot's trajectories plotted in red. The experimental area is surrounded by concrete pillars connected by a safety tape. (d) The resulting map of the snow-covered terrain with one of the robot's trajectories plotted in red.

unevenness adds unmodeled noise to the skid-steering motion when operating at high speeds. To better train and evaluate all of the the models, the trajectories were planned in a way to maximize the excitement range and coverage of the model input variables (i.e., left and right wheels commanded angular velocities). Some parts of these trajectories obtained by the ICP mapping algorithm are presented in Figure 3c and Figure 3d.

In order to obtain an accurate location of the platform to compare the different models, we used the ICP algorithm. This algorithm allows for accurate odometry measurements, by registering 3D point clouds together [15]. Because of its few centimeters margin of error, it was not necessary to resort to more precise measuring tools such as theodolites. Moreover, since each platform trajectory is estimated with identical ICP parameters, the magnitude of the localization error will be the same for each of the models tested. As a result, the ICP positioning and orientation errors will not change the behavior of the different models tested. The library `libpointmatcher` [16] was used to compute ICP offline, using a point-to-plane minimization.

To properly evaluate each model, it is necessary to train for model parameters and evaluate model performance on two separate trajectories. Our model training procedure is similar to the two-step method proposed in [10]. The first step is to obtain experimental data by driving the robot manually, while recording commanded wheel velocities and sensor measurements. The data is then processed offline to compute the ground truth localization, using the ICP algorithm.

The training path is then split into N distinct segments corresponding to a total traveled distance of h_t. The commanded wheel velocities corresponding to each of the N segments are then used to predict the robot's motion, starting from the initial position of the corresponding segment. The final model-predicted position of the robot is then compared with the ground truth position for each segment. Our loss function $l(\cdot)$ is the sum of N squared Mahalanobis distances using

$$l(\boldsymbol{k}) = \sum_{i=1}^{N} (\boldsymbol{x} - \hat{\boldsymbol{x}})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \hat{\boldsymbol{x}}), \tag{13}$$

where $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ are respectively the ground truth and model-predicted state of the robot defined as its 2D position and its orientation. The covariance matrix $\boldsymbol{\Sigma}$ is there to bring meters for the position and radians for the orientation in a common unitless value. In our case, we set this covariance to identity. The set of parameters for each model is represented by the vector $\boldsymbol{k}$. An optimization algorithm is then used to search for the set of parameters $\boldsymbol{k}$ minimizing the loss function $l(\cdot)$.

Previous work on model parameter identification for SSMR relied on temporal horizons for ground truth trajectory segmentation [4], [10], [17]. We investigated two different strategies when splitting the ground truth trajectory: by using a temporal and a spatial horizon. In particular, we found that using a spatial horizon for model parameter optimization allowed for the easy removal of outlier data generated when zero velocity commands were given to the robot. The trained models are then evaluated using two different metrics: the relative error of the translational prediction ε_t and the relative error of the angular prediction ε_θ, respectively computed as the translational error and angular error the predictor does per meter. These two metrics are computed on the evaluation horizon h_e over the entire evaluation trajectories, as in [18].

V. Results

In order to fine-tune and evaluate the different models, we first determine the impact of the training and evaluation horizon h_t and h_e on the model performance. Once the best horizons are chosen, the models are trained and compared in both linear and angular errors. We then deepen our study with the most promising model, the differential drive symmetric model, by looking at the relation between the errors and the commands sent to the robot.

In order to determine the impact of the training horizon h_t and the evaluation horizon h_e on the performance, we trained every model for various values for h_t and evaluated the translational error ε_t for various horizons h_e. Except for the ROC-based model, which did not react to a variation in training horizon h_t, the error ε_t of the different models had a similar behavior when the values for the horizons h_t and h_e were varying. As an example, the translational error ε_t values of the extended differential-drive asymmetric model prediction for varying training and evaluation horizons are shown in Figure 4.

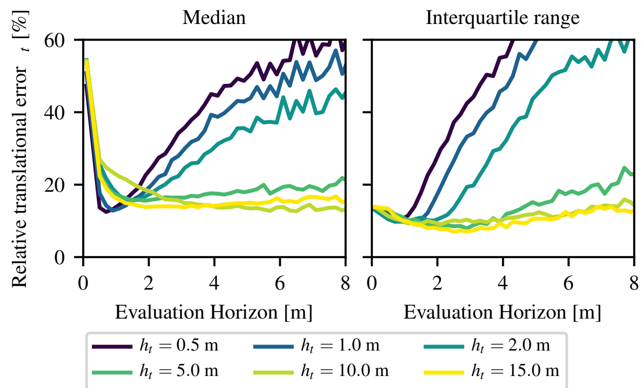


Figure 4: Relative translational error ε_t of the extended differential-drive asymmetric model in a snowy environment, as a function of the evaluation horizon window h_e for different training windows h_t. *Left:* Median of the relative linear error. *Right:* Interquatile range of the relative linear error.

We can first observe that the model error ε_t behaves differently depending on the training horizon h_t. Models trained on smaller horizons offer better performances at lower evaluation horizons while they suffer at higher evaluation horizons. This indicates that the model training horizon h_t should be chosen in accordance with the application of the model. In robotics applications, controllers mainly use small horizon. Thus, the training and the evaluation horizons should mainly use small horizons.

It can be seen in Figure 4 that training horizons of $2\,\mathrm{m}$ or $5\,\mathrm{m}$ quickly converge to a small error compared to longer training horizons, while not suffering of high error for long evaluation horizons as much as smaller training horizons. For this model, the $15\,\mathrm{m}$ h_t also shows quick convergence but this effect is only observed for the extended differential-drive asymmetric model on snow. Furthermore, all the median curves assume similar values at $h_e = 2\,\mathrm{m}$. The initial error for h_e approaching $0\,\mathrm{m}$ is caused by the combined measurement noise of the wheel velocities and our ground truth. Hence, taking an evaluation horizon of $2\,\mathrm{m}$ leads to a certain invariance of the error ε_t on the training horizons. Also, longer evaluation horizons h_e tend to introduce a high, interquartile range for predicting the errors ε_t, meaning that the evaluation is very dependant on the state of the robot. As we want to evaluate the models on complex trajectories, an evaluation horizon of $h_e = 2\,\mathrm{m}$ is chosen for the rest of this work. The evaluation trajectories differ from the training trajectory but meet the same goal of maximizing model input excitement range and coverage. We have used this h_t to train for all model parameters using the parameter identification method described in Section IV. The overview of the parameters can be found in Table I.

Table I: Summary of parameters used and trained for each model. For all models, the wheel radius r is equal to $0.3\,\mathrm{m}$.

Model	*Trained concrete*	*Trained snow*	*Bounds*
DD with $b = 1.2$ m	–	–	–
Extended	$\alpha = 0.94$	$\alpha = 0.86$	$\in [0, 1]$
DD Symmetric	$\hat{b} = 4.46\,\mathrm{m}$	$\hat{b} = 3.08\,\mathrm{m}$	$\in [0, \infty)$
Extended	$\alpha_l = 0.90$	$\alpha_l = 0.81$	$\in [0, 1]$
DD Asymmetric	$\alpha_r = 0.92$	$\alpha_r = 0.84$	$\in [0, 1]$
	$x_v = -2.57\,\mathrm{m}$	$x_v = -2.71\,\mathrm{m}$	$\in \mathbb{R}$
	$y_l = 4.66\,\mathrm{m}$	$y_l = 3.00\,\mathrm{m}$	$\in [0, \infty)$
	$y_r = -5.00\,\mathrm{m}$	$y_r = -3.85\,\mathrm{m}$	$\in (-\infty, 0]$
ROC	$\alpha = 0.91$	$\alpha_l = 0.80$	$\in [0, 1]$
with $b = 1.2$ m	$\beta_1 = 42.73$	$\beta_1 = 1.36$	$\in \mathbb{R}$
	$\beta_1 = 11.09$	$\beta_2 = -0.18$	$\in \mathbb{R}$
Full linear	$\gamma_{11} = 0.47$	$\gamma_{11} = 0.46$	$\in \mathbb{R}$
	$\gamma_{12} = 0.44$	$\gamma_{12} = 0.36$	$\in \mathbb{R}$
	$\gamma_{21} = -0.22$	$\gamma_{21} = -0.31$	$\in \mathbb{R}$
	$\gamma_{22} = 0.26$	$\gamma_{22} = 0.34$	$\in \mathbb{R}$
	$\gamma_{31} = -0.10$	$\gamma_{31} = -0.13$	$\in (-\infty, 0]$
	$\gamma_{32} = 0.08$	$\gamma_{32} = 0.12$	$\in [0, \infty)$

Legend: DD = Differential drive.

The translational errors ε_t and angular errors ε_θ for this experiment are shown in Figure 5 and Figure 6. In them, the median and quartiles at $25\,\%$ and $75\,\%$ are depicted with the gray box, while the underlying curves show the data distribution. In Figure 5, we can see that the residual errors for the ideal differential-drive model are much higher than for any of the trained models. Indeed, the differential-drive model does not take the slippage and skidding phenomenom, leading to high errors for SSMRs. It can be observed that the ideal differential drive performs better on snow that on concrete. This shows that SSMRs motion is closer to that of an ideal differentially driven robot when operating on snow-covered terrain than when operating on concrete.

Figure 6 shows in greater details the distribution of errors for the four trained models. There, it can be seen that except for the ROC-based one, all models tend to perform similarly. The model prediction error is similar for both snow (blue) and concrete (red). Furthermore, the extended differential-drive asymmetric and full linear models offer the best linear displacement predictions, which is due to the fact that they account for lateral motion. However, the extended differential-drive symmetric model offers more accurate angular displacement prediction while offering slightly inferior accuracy for translational displacement prediction than other kinematic models presented in this work. The fact that it only has two parameters to be trained makes it an interesting choice for skid-steering locomotion modeling. Indeed, fewer parameters lead to a smaller computational cost for training, and such a model is less prone to converge into a local minimum as the search space is smaller. Furthermore, richer

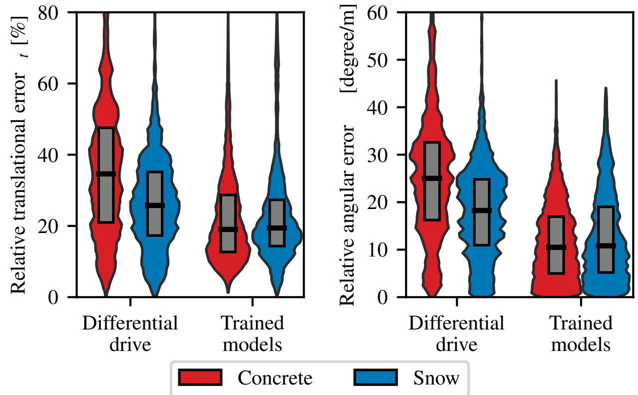


Figure 5: Overall errors of the differential-drive model and the trained models, combined in a single distribution. As expected, the trained models perform vastly better, as they take into account the wheel ground contact interactions.

models can also suffer from overfitting on the training data, which is less likely to happen with fewer parameter models.

It should be noted that the symmetry hypothesis underlying many models is very close to reality for our platform. This is understandable, as our vehicle is almost symmetric in design and the added components add negligible mass to the system. A SSMR not respecting this constraint would require the use of a model that allows for the asymmetry, such as the extended differential-drive asymmetric model or the full linear model, but at the cost of a higher model dimensionality, therefore bearing the aforementioned disadvantages.

In order to highlight the differences of wheel-ground interactions between the snow and the concrete, we measured the actual rotation of the robot for a series of given commands. Figure 7 shows the measured angular displacement given the commanded angular displacement for the evaluation trajectories on snow and concrete. As a lot of data was collected, the median and the quartiles at 25% and 75% were computed to ease the reading.

It can be seen that for low commanded angular displacement, the robot behaves similarly on snow and on concrete. However, for commanded angular displacements of over $30°$, the curve for resulting angular displacement grows faster when the robot is operated on snow than on concrete. A drop in measured angular displacement for high angular displacement commands can also be seen for snow-covered terrain. This result suggests a nonlinear effect in SSMRs motion on snow-covered terrain which could be hard to describe with linear kinematic models. Additionally, interquartile range is generally stable for both terrain types but higher on concrete than on snow, showing that for a specific angular displacement command, the range of possible angular displacement actually done by the SSMR is higher on concrete than on snow. Higher speeds were not reached for the concrete trajectory because of safety purposes. The lack of data for zero commanded angular velocity is due to a combi-

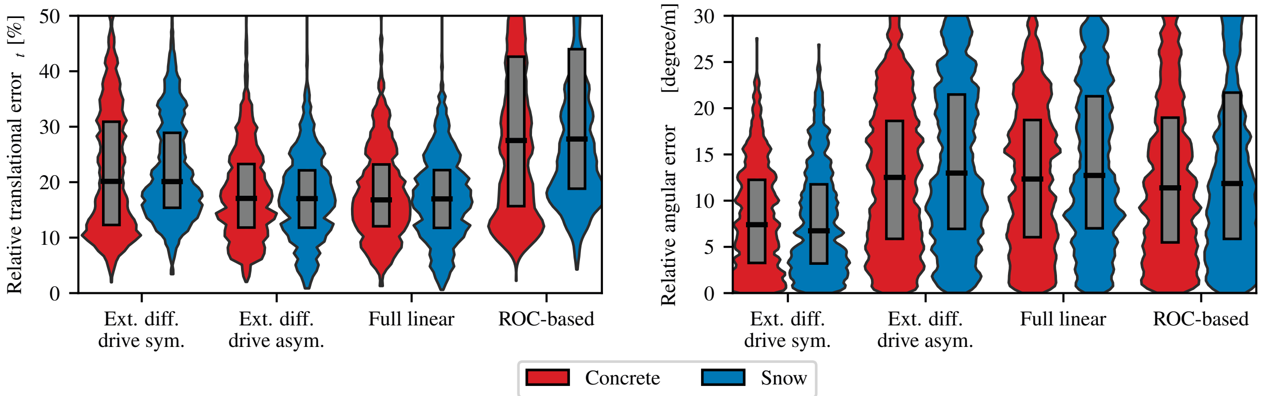


Figure 6: Overall error of the models depending on which environment the robot evolved in.

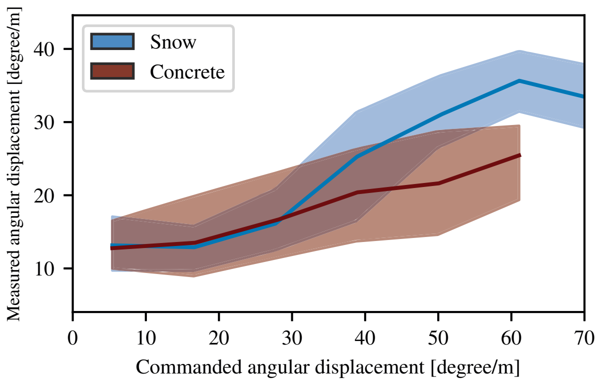


Figure 7: Actual rotation over one meter as a function of the difference of wheel velocity. The median is represented with a straight line, where the shaded zones represent the quartiles at $25\,\%$ and $75\,\%$. Because of the low friction coefficient of the snow, the robot tends to rotate more for the same command compared to concrete.

nation between the choice of a spatial evaluation window h_e = $2\,\mathrm{m}$ and the trajectory planning aiming to maximize commanded wheel velocities range and coverage, leading to no window with zero commanded angular displacement. It can be observed that for small angular displacement commands, the robot can have a higher angular displacement than the commanded one. This could be due to two phenomena: as the robot was driven at high velocities on uneven, snow-covered terrain, the suspension was not able to compensate for the vibrations leading to a loss of contact between the wheels and the ground. This additional noise in the command led the robot to slightly turn in straight lines. Also, the high momentum of the robot sometimes caused it to continue to turn in end of turns even if the actual command had a null angular displacement. In addition, the relative translational displacement is constant for every commanded linear displacement and is the same for snow and concrete. This shows that the angular displacement is the main factor of error while commanding SSMRs.

As demonstrated before, the best model in our case is the extended differential drive symmetric. The study of this model is then deepened to the study of the angular error. Indeed, the linear error was independent of the given commands in our study.

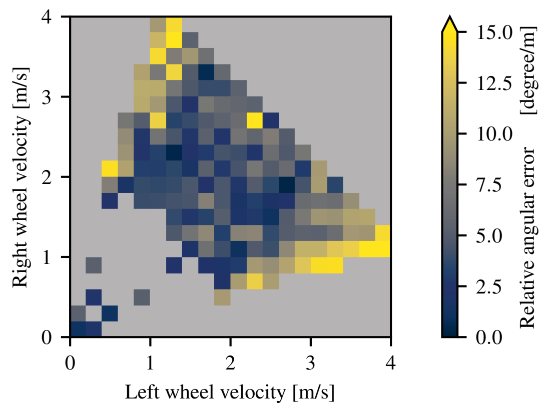


Figure 8: Relative angular error for the extended differential drive symetric model on snow. We observed a similiar behavior on the concrete.

An analysis of relative errors in prediction was conducted for the extended differential-drive symmetric model on both snow and concrete. The prediction error was computed for every evaluation trajectory segments, as well as the corresponding wheels angular velocities commands. This way, we were able to determine that the relative translational prediction errors ε_t were uncorrelated with mean angular velocity commands for each side. However, angular prediction errors ε_θ were correlated with mean angular velocity commands, as can be seen in Figure 8. It can be observed that the prediction error reaches its maximum in the areas when either side's commanded wheel velocity is about twice that of the other side. In particular, we see a steep increase in the prediction relative angular error in the high error areas. This suggests a non-linearity in the relationship, which cannot be captured by any of the linear models tested.

VI. Conclusion

In this paper, we compared five different kinematic models to describe the motion of a 590 kg SSMR platform both on concrete and on a snow-covered terrain. We have shown that the model training horizon should be selected based on the model's application. We have compared the prediction accuracy of five kinematic models for SSMRs. We have also highlighted differences in the behavior and model-prediction error of SSMRs when operating on the two different terrain types. We have also identified commanded angular wheel velocity sets that induce high prediction errors of angular displacement. As future work, we aim to implement more advanced models and to develop a path-following algorithm robust to multiple terrain types for a given trajectory.

Acknowledgements

This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) through the grant CRDPJ 511843-17 BRITE (bus rapid transit system), CRDPJ 527642-18 SNOW (Self-driving Navigation Optimized for Winter) and FORAC.

References

[1] G. Huskić, S. Buck, M. Herrb, S. Lacroix, and A. Zell, "High-speed path following control of skid-steered vehicles," *The International Journal of Robotics Research*, vol. 38, no. 9, pp. 1124–1148, 2019.

[2] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive driving with model predictive path integral control," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2016-June, IEEE, 2016, pp. 1433–1440.

[3] B. Shamah, M. D. Wagner, S. Moorehead, J. Teza, D. Wettergreen, and W. L. Whittaker, "Steering and control of a passively articulated robot," in *Proc. SPIE 4571, Sensor Fusion and Decentralized Control in Robotic Systems IV*, G. T. McKee and P. S. Schenker, Eds., 2001, pp. 96–107.

[4] S. Rabiee and J. Biswas, "A Friction-Based Kinematic Model for Skid-Steer Wheeled Mobile Robots," in *2019 International Conference on Robotics and Automation (ICRA)*, vol. 2019-May, IEEE, 2019, pp. 8563–8569.

[5] D. S. Apostolopoulos, M. D. Wagner, B. N. Shamah, L. Pedersen, K. Shillcutt, and W. L. Whittaker, "Technology and Field Demonstration of Robotic Search for Antarctic Meteorites," *The International Journal of Robotics Research*, vol. 19, no. 11, pp. 1015–1032, 2000.

[6] L. E. Ray, J. H. Lever, A. D. Streeter, and A. D. Price, "Design and power management of a solar-powered "Cool Robot" for polar instrument networks," *Journal of Field Robotics*, vol. 24, no. 7, pp. 581–599, 2007.

[7] C. M. Gifford, E. L. Akers, R. S. Stansbury, and A. Agah, "Mobile Robots for Polar Remote Sensing," in *The Path to Autonomous Robots*, Boston, MA: Springer US, 2009, pp. 1–22.

[8] J. H. Lever, A. J. Delaney, L. E. Ray, E. Trautmann, L. A. Barna, and A. M. Burzynski, "Autonomous GPR Surveys using the Polar Rover Yeti," *Journal of Field Robotics*, vol. 30, no. 2, pp. 194–215, 2013.

[9] M. Paton, F. Pomerleau, K. MacTavish, C. J. Ostafew, and T. D. Barfoot, "Expanding the Limits of Vision-based Localization for Long-term Route-following Autonomy," *Journal of Field Robotics*, vol. 34, no. 1, pp. 98–122, 2017.

[10] A. Mandow, J. L. Martinez, J. Morales, J. L. Blanco, A. Garcia-Cerezo, and J. Gonzalez, "Experimental kinematics for wheeled skid-steer mobile robots," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2007, pp. 1222–1227.

[11] J. Pentzer, S. Brennan, and K. Reichard, "Model-based Prediction of Skid-steer Robot Kinematics Using Online Estimation of Track Instantaneous Centers of Rotation," *Journal of Field Robotics*, vol. 31, no. 3, pp. 455–476, 2014.

[12] S. Moosavian and A. Kalantari, "Experimental slip estimation for exact kinematics modeling and control of a Tracked Mobile Robot," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2008, pp. 95–100.

[13] T. Wang, Y. Wu, J. Liang, C. Han, J. Chen, and Q. Zhao, "Analysis and Experimental Kinematics of a Skid-Steering Wheeled Robot Based on a Laser Scanner Sensor," *Sensors*, vol. 15, no. 5, pp. 9681–9702, 2015.

[14] G. Anousaki and K. Kyriakopoulos, "A dead-reckoning scheme for skid-steered vehicles in outdoor environments," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, IEEE, 2004, 580–585 Vol.1.

[15] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP variants on real-world data sets," *Autonomous Robots*, vol. 34, no. 3, pp. 133–148, 2013.

[16] F. Pomerleau, P. Krüsi, F. Colas, P. Furgale, and R. Siegwart, "Long-term 3D map maintenance in dynamic environments," *Proceedings - IEEE International Conference on Robotics and Automation*, no. 1, pp. 3712–3719, 2014.

[17] N. Seegmiller, "Dynamic model formulation and calibration for wheeled mobile robots," PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 2014.

[18] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2012, pp. 3354–3361.

DRIVE: Data-driven Robot Input Vector Exploration

Dominic Baril[1], Simon-Pierre Deschênes, Luc Coupal, Cyril Goffin,
Julien Lépine, Philippe Giguère, François Pomerleau[1]

***Abstract*— An accurate motion model is a fundamental component of most autonomous navigation systems. While much work has been done on improving model formulation, no standard protocol exists for gathering empirical data required to train models. In this work, we address this issue by proposing Data-driven Robot Input Vector Exploration (DRIVE), a protocol that enables characterizing uncrewed ground vehicles (UGVs) input limits and gathering empirical model training data. We also propose a novel learned slip approach outperforming similar acceleration learning approaches. Our contributions are validated through an extensive experimental evaluation, cumulating over 7 km and 1.8 h of driving data over three distinct UGVs and four terrain types. We show that our protocol offers increased predictive performance over common human-driven data-gathering protocols. Furthermore, our protocol converges with 46 s of training data, almost four times less than the shortest human dataset gathering protocol. We show that the operational limit for our model is reached in extreme slip conditions encountered on surfaced ice. DRIVE is an efficient way of characterizing UGV motion in its operational conditions. Our code and dataset are both available online at this link: `https://github.com/norlab-ulaval/DRIVE`.**

I. Introduction

The ability to model the motion of uncrewed ground vehicles (UGVs) is fundamental to enabling localization [1], path planning [2] and path following [3]. Poor vehicle-terrain characterization will lead to significant modeling errors, potentially causing system failure [4]. With limited available information and sensory measurements on vehicle and ground properties, generating a reliable UGV motion model remains challenging. For most models, training on empirical data is required to reduce modeling error [5]. This task requires deploying a UGV in its operational environment and manually drive it for an extended period [6]. Since energy consumption and deployment time are critical for various UGV applications, facilitating this task is of high importance. Additionally, standardizing this process could help engineers to ensure that their systems are satisfactory to norm ISO 34502:2022(E) on autonomous navigation.[2]

Most work on UGV motion modeling relies on manual driving to gather a training dataset, with little to no details on the driving protocol. Thus, we propose the *Data-driven Robot Input Vector Exploration (DRIVE)*, a protocol aiming to facilitate and standardize vehicle characterization with respect to the terrain, as illustrated in Figure 1. We start by identifying the true vehicle's input space, differing from the manufacturer's specifications. We then automatically send commands to the UGV to cover the entire true input space. This differs from the common manual driving approach, which tends to cover only forward driving, as shown by the red dots representing our previous work [7]. We show that broad input-space coverage offers significant modeling error reduction compared to narrow coverage. With this dataset, we train a learned vehicle slip model that maps UGV commands to resulting body velocities. The resulting trained parameters vary significantly depending on terrain, as highlighted by the green and blue diamond areas in Figure 1, representing navigation on gravel and snow respectively.

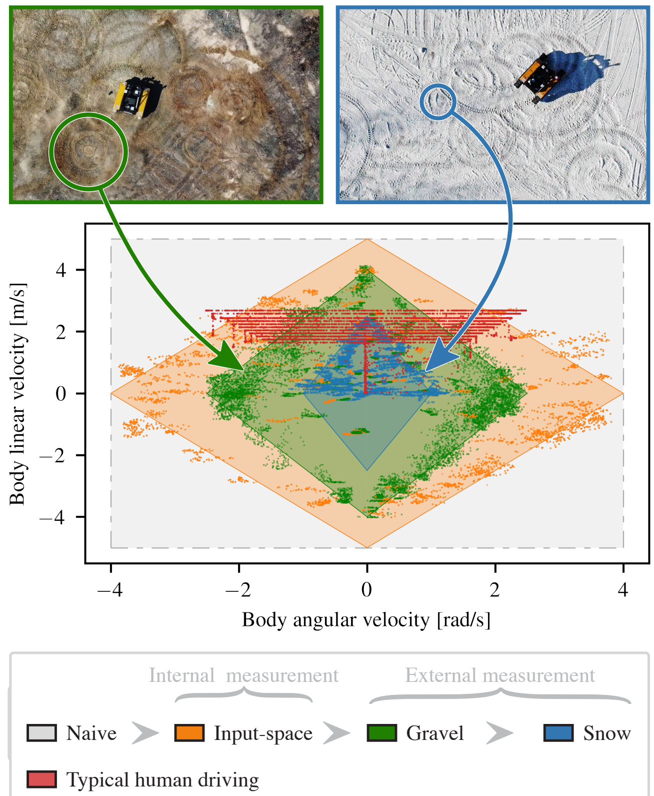


Fig. 1. Vehicle and terrain characterization done through DRIVE. The manufacturer-defined Naive input-space region is drawn in gray. The vehicle's true input-space, characterized through internal measurements, is shown in orange. Typical human driving is shown in red. The resulting body velocities are represented in green for gravel and blue for snow.

*This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) through the grant CRDPJ 527642-18 SNOW (Self-driving Navigation Optimized for Winter). FRQNT as well.

[1]Northern Robotics Laboratory, Université Laval, Quebec City, Quebec, Canada { dominic.baril, francois.pomerleau }@norlab.ulaval.ca

[2]"ISO 34502:2022(E): Road vehicles — Test scenarios for automated driving systems — Scenario-based safety evaluation framework", 2022

The specific contributions of this paper are (i) DRIVE, a standardized UGV characterization and motion data generation protocol allowing to train motion models on the entire vehicle input space; (ii) A novel slip-based UGV motion prediction model, leveraging the accuracy of model-based approaches and the minimal system characterization requirement of learning-based approaches. We validate our contributions with an extensive experimental evaluation featuring three distinct UGVs, with weights ranging from $75\,\mathrm{kg}$ to $470\,\mathrm{kg}$, two types of ground interaction (i.e., wheels and tracks) and four different terrain types. Our observations rely on driving data totaling $7\,\mathrm{km}$ and $1.8\,\mathrm{h}$.

II. Related Work

Most vehicle motion modeling approaches can be divided into two distinct categories: model-based and learning-based. Both categories share the requirement of using empirical driving data to train their parameters and reduce modeling errors. For both categories, there exists no standardized protocol for training dataset generation.

Model-based approaches can be split into two distinct categories: *kinematics* and *dynamics*. Kinematic models remain the most popular for UGVs due to their low computational complexity and number of parameters to train. For skid-steering mobile robots (SSMRs), Mandow *et al.* [8] reduced model prediction error by $15\,\%$ compared to the manufacturer's model using a kinematic model empirically identifying vehicle slip and skid. Seegmiller *et al.* [9] proposed a similar additive slip approach, computing slip based on kinematic quantities, yielding prediction error reduction between $70\,\%$ and $90\,\%$ depending on terrain type, again compared to the manufacturer's model. Bussmann *et al.* [10] extended the experimental validation for additive slip approaches and showed similar performance for a $900\,\mathrm{m}$ experiment on off-road terrain. On the other hand, dynamic models account for various forces acting on the vehicle's body. Seegmiller *et al.* [4] proposed a multi-body full dynamic motion model with a generic formulation based on vehicle geometry and properties. This work has been extended by Yang *et al.* [11], showing simulation errors of less than $3.4\,\%$ for vehicle slip ratio. While being more accurate than kinematic models, dynamic models require extensive vehicle characterization effort and expertise. For all the work mentioned above, empirical training data is acquired through a human driving the UGV with little to no guidelines, which motivates our standardized protocol.

Alternatively, **learning-based approaches** have been explored in the literature, yielding more accurate models for extreme UGV motion. In these approaches, part of the prediction is done through a nominal model, often represented by a unicycle model, with a module allowing to learning system dynamics. Gaussian processes (GPs) have become a popular approach to learn system dynamics, both for vehicle slip in off-road driving [12] and tire forces in high-speed road racing [13]. McKinnon *et al.* [14] have proposed a similar learning approach, however replacing GPs learning with Bayesian linear regression (BLR). The lower computational complexity of BLR, when compared to GPs, makes it a more suitable approach for real-time UGV motion prediction. Alternatively, Djeumou *et al.* [15] have proposed a tire-force learning framework allowing to perform autonomous drifting with $3\,\mathrm{min}$ of driving data. Deep learning has also been explored for motion prediction in off-road terrain. Williams *et al.* [6] have shown the ability to perform aggressive driving when relying on a $30\,\mathrm{min}$ training dataset. For increased resilience to sensor failure, Tremblay *et al.* [16] have proposed a multi-modal learned-dynamics model that leverages the various sensor measurements available for UGVs. Due to the importance of prediction uncertainty in enabling robust control for UGVs [3], this work focuses on BLR, which provides prediction uncertainty estimations [14]. Our novel *slip-based BLR* model allows us to leverage the minimal requirements of learning-based approaches in terms of system characterization, [14] as well as the better accuracy of model-based approaches [9]. In this work, the approach of McKinnon *et al.* [14] is used as a comparison point, as it is the closest to our model formulation.

Although both model-based and learning-based approaches require empirical training data, only a few **dataset-gathering protocols** have been published. Voser *et al.* [17] have proposed to maintain a steady forward velocity while slowly increasing angular velocity, enabling generation of a quasi-steady-state empirical dataset. Wang *et al.* [18] have proposed a similar approach with a varying commanded curvature radius to empirically identify the relation between angular velocity and SSMR skid. These approaches only cover a small subset of the vehicle's input space. One can also find large, multimodal datasets allowing to train and evaluate models for off-road and extreme driving [19]. However, such datasets overrepresent forward motion, are limited to a specific UGV and would require new training data for any new vehicle configuration. Manual training data gathering guidelines have been proposed by Williams *et al.* [6], asking the driver to vary his driving style. However, these remain time-consuming and subject to input space coverage bias. We demonstrate that training a motion model with the DRIVE protocol allows increased motion prediction performance and fast training dataset gathering.

III. Methodology and Theory

In this section, we provide details on DRIVE, our automated vehicle characterization and training dataset-gathering protocol. We then describe our proposed slip-based BLR motion model. Due to the limited number of UGVs accessible to us, we focus on SSMRs.

The involved model variables are depicted in Figure 2. We limit the states of the vehicle to planar motion, such that the robot's state ${}^{\mathcal{G}}\boldsymbol{q} = [x, y, \theta]^T$ represents the pose of the vehicle in the global coordinate frame $\mathcal{G}$. The robot's body frame $\mathcal{R}$, has its x and y axis aligned with the vehicle's longitudinal and lateral directions respectively. For most SSMRs, the input vector is defined as $\boldsymbol{u} = [\omega_l, \omega_r]^T$, representing the left and right wheel angular velocities. State propagation, allowing to compute the next state $\boldsymbol{q}_{t+dt}$ based on the current state $\boldsymbol{q}_t$ and input $\boldsymbol{u}$ is computed as follows:

$$^{\mathcal{G}}\boldsymbol{q}_{t+dt} = {}^{\mathcal{G}}\boldsymbol{q}_t + {}^{\mathcal{G}}_{\mathcal{R}}\boldsymbol{T}\left({}^{\mathcal{G}}\theta_t\right){}^{\mathcal{R}}\boldsymbol{v}_t\ dt, \tag{1}$$

$$^{\mathcal{R}}\boldsymbol{v}_t = {}^{\mathcal{R}}\boldsymbol{f}_t(\boldsymbol{u}_t) - {}^{\mathcal{R}}\boldsymbol{s}_t, \tag{2}$$

where ${}^{\mathcal{R}}\boldsymbol{v}_t$ is the vehicle's translational and rotational body velocity, oriented in the vehicle's inertial frame $\mathcal{R}$, and ${}^{\mathcal{G}}_{\mathcal{R}}\boldsymbol{T}\left({}^{\mathcal{G}}\theta_t\right)$ is a transformation matrix producing a rotation of the robot's angle in the world frame ${}^{\mathcal{G}}\theta$. The vehicle's body velocity ${}^{\mathcal{R}}\boldsymbol{v}_t \in \mathbb{R}^3$ is modeled as the commanded velocity ${}^{\mathcal{R}}\boldsymbol{f}_t(\boldsymbol{u}_t) \in \mathbb{R}^3$, to which we subtract the slip velocity ${}^{\mathcal{R}}\boldsymbol{s}_t \in \mathbb{R}^3$. The delay between UGV commands is represented by dt. The diamonds in the top-left inset of Figure 2 represent conceptually the set of possible commanded UGV body velocities $\mathcal{J}$ in orange and the set of actually possible body velocities $\mathcal{B}$. Thus, to characterize UGV motion, we require two things: a protocol to gather empirical data and a model to learn vehicle slip.

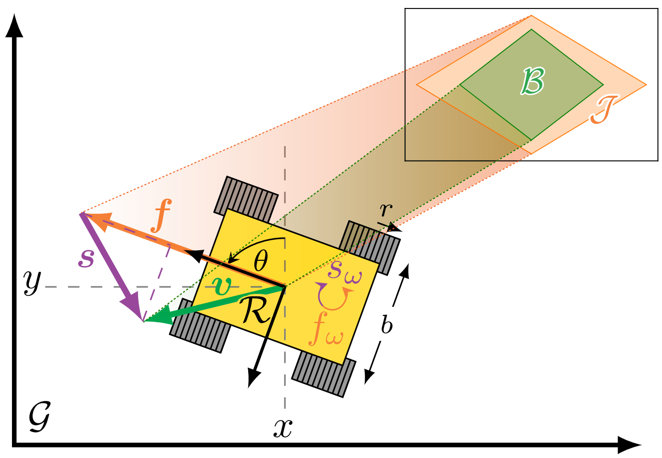


Fig. 2. Top view drawing of a SSMR. In orange is the commanded body velocity ${}^{\mathcal{R}}\boldsymbol{f}$ and green is the resulting body velocity ${}^{\mathcal{R}}\boldsymbol{v}$. The input-space $\mathcal{J}$ is shown in orange and the body velocity space $\mathcal{B}$ is shown in green. The difference between commanded and resulting body velocity is represented as the slip velocity ${}^{\mathcal{R}}\boldsymbol{s}$ in purple. All represented velocities have an angular component $(\cdot)_\omega$. Robot parameters are the wheel radius r and vehicle width b.

Our DRIVE protocol, described in Section III-A, automates the task of gathering a complete training dataset $\mathcal{D} = \{\boldsymbol{u}, \tilde{\boldsymbol{u}}, \boldsymbol{X}_x, \boldsymbol{s}_x, \boldsymbol{X}_y, \boldsymbol{s}_y, \boldsymbol{X}_\omega, \boldsymbol{s}_\omega\}$, where we concatenate n observations of slip input $\boldsymbol{X} \in \mathbb{R}^{n\times k}$ and slip velocities $\boldsymbol{s} \in \mathbb{R}^n$ for each vehicle dimension. The number of slip inputs per dimension is defined as k. This data is also used to train a powertrain model for each side of the SSMR by minimizing the error between commanded $\tilde{\boldsymbol{u}}$ and measured $\boldsymbol{u}$ vehicle input through the powertrain parameters described in Section III-B.1. We then use BLR to train our slip-based learning model, described in Section III-B.2.

A. Data-driven Robot Input Vector Exploration (DRIVE)

To learn any UGV slip model, we require a training dataset $\mathcal{D}$. Details on the composition of this dataset are given throughout Section III-B. It has been shown that numerous factors impact UGV dynamics, such as vehicle orientation [4], tire saturation [15] and ground properties [7]. However, coverage of the entire spectrum of dynamic features for UGVs would require a large-scale training dataset, and extracting insights from this dataset would be plagued by the curse of dimensionality [20]. We simplify our problem by enabling stimulation of the entire input space $\mathcal{J}$ for the UGV, represented as the orange diamond in Figure 2. We rely on random, uniform sampling of UGV inputs since it was previously shown that random search on multidimensional space is faster and offers similar performance as grid search [21]. Thus, our protocol coupled with an approximate UGV motion model allows fast training dataset gathering for faster deployment in novel operational conditions. The system requirements to perform our protocol are as follows: (i) a sub-servo system mapping body-level commands to wheel commands; (ii) vehicle acceleration limits to reduce strain on vehicle components; (iii) an accurate localization system, estimating the robot position and velocity in a global frame; (iv) a safety operator to prevent the vehicle from leaving a predefined safe perimeter during the random command sampling. Lastly, we release our protocol as an open-source package to facilitate replicability.

The first step of DRIVE is to send high body longitudinal velocity commands to the platform in both directions to determine the wheel velocity limits ω_{min} and ω_{max}. This way, we define the UGV's true input-space $\mathcal{J}$ as combinations of left and right commanded wheel velocities, such that $\omega_{min} \leq \omega_l, \omega_r \leq \omega_{max}$ for SSMRs. This input-space $\mathcal{J}$ is shown in terms of body velocities as the orange diamond in Figure 2. Once the input-space limits are defined, we sample input vectors $\boldsymbol{u}_s$ from our calibration distribution, defined as a two-dimensional uniform distribution parametrized by vehicle input limits $\boldsymbol{u}_s \sim \mathcal{U}_2\left(\boldsymbol{\omega}_{min}, \boldsymbol{\omega}_{max}\right)$. Our goal is then to maintain each sampled input for a duration sufficient to gather steady-state motion data. Thus, we define training windows of 2 s, the same duration as typically used in the seminal UGV path following work of Williams *et al.* [6]. We assume the UGV requires one training window to reach any desired wheel velocity. Since the majority of UGV motion is quasi-steady state [17], we define a training interval consisting of one transient training window and two steady training windows, lasting a total of 6 s. An example of two training intervals is shown in Figure 3. We use transient-state windows to train our powertrain model, described in Section III-B.1. We keep both steady-state and transient-state windows for our model training and evaluation to enable both steady-state and transient model training. This procedure is repeated until the user stops or the desired data generation time is reached, which varies depending on the user's goal.

B. Dynamics-aware slip-based model

1) Powertrain model: We start by defining a powertrain model to reduce the predicted wheel velocity $\hat{\omega}$ error, the result of which can be seen as both dashed lines in Figure 3. We use the same first-order plus dead time transient response

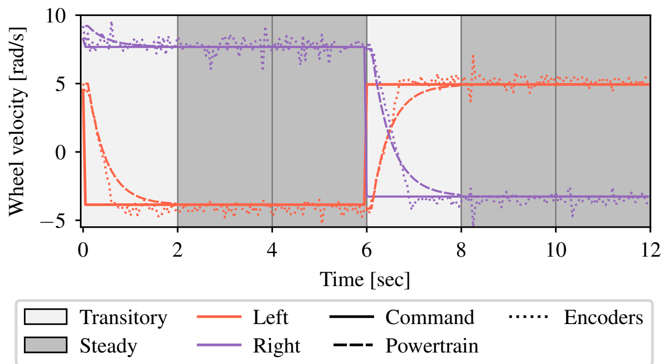


Fig. 3. Commanded, encoder-measured and modeled wheel velocities for both sides of a SSMR during two DRIVE training intervals. The powertrain model is described in Section III-B.1. Each training step consists of one transient-state window (in light gray) and two steady-state windows (in dark gray). Commands and measurements on the x-axis are acquired at a rate of 20 Hz.

model as used by Seegmiller *et al.* [5]:

$$\begin{aligned}\hat{\omega}_t &= \left(e^{\beta}\right)\omega_{t_0} + \left(1 - e^{\beta}\right)\tilde{\omega}_{t-\tau_d} \ , \\ \beta &= \frac{(t-\tau_d)}{\tau_c} \ ,\end{aligned} \tag{3}$$

where $\hat{\omega}$, $\tilde{\omega}$ and ω are the predicted, commanded and measured wheel velocities, respectively. We also define the initial time t_0 and prediction horizon at time t. Here, the parameters that require characterization are the time constant τ_c and the time delay τ_d. One should note that these parameters are not considered symmetrical in our protocol and are trained independently for both sides of SSMRs. Thus, our protocol can identify vehicle powertrain asymmetries.

2) Body slip model: Next, we define a model enabling to compute both the commanded body velocity ${}^{\mathcal{R}}\boldsymbol{f}_t$ and resulting slip velocity ${}^{\mathcal{R}}\boldsymbol{s}_t$ with respect to predicted input $\tilde{\boldsymbol{u}}_t$. For SSMRs, the commanded body velocity ${}^{\mathcal{R}}\boldsymbol{f}_t(\tilde{\boldsymbol{u}}_t)$ can be modeled through the ideal differential-drive model [8] as

$${}^{\mathcal{R}}\boldsymbol{f}_t(\boldsymbol{u}_t) = \begin{bmatrix} f_x \\ f_y \\ f_\omega \end{bmatrix} = r \begin{bmatrix} \frac{1}{2}, \frac{1}{2} \\ 0, 0 \\ \frac{-1}{b}, \frac{1}{b} \end{bmatrix} \begin{bmatrix} \hat{\omega}_{lt} \\ \hat{\omega}_{rt} \end{bmatrix}, \tag{4}$$

where r and b are the SSMR's wheel or track sprocket radius and vehicle width, respectively, as shown in Figure 2. We use the estimated wheel velocities through Equation 3 as the input vector $\tilde{\boldsymbol{u}}_t$. We consider slip in each dimension of the vehicle separately ${}^{\mathcal{R}}\boldsymbol{s}_t = [s_x, s_y, s_\omega]^T$, with the form

$$s_t = \boldsymbol{\gamma}^T \, {}^{\mathcal{R}}\boldsymbol{x}_t + \eta, \tag{5}$$

where $\boldsymbol{\gamma} \in \mathbb{R}^k$ are the weights associated to each slip input and $\eta \sim \mathcal{N}(0, \sigma^2)$. We draw inspiration from off-road vehicles dynamics work in the literature to define dynamics-aware basis functions for vehicle slip [9]. As shown by Seegmiller *et al.* [4], the following set of basis functions to estimate vehicle slip shows similar performance as fully dynamic models in off-road terrain. Firstly, for longitudinal slip ${}^{\mathcal{R}}\boldsymbol{s}_x$, we use the vehicle's rolling resistance, proportional to commanded body longitudinal velocity ${}^{\mathcal{R}}\boldsymbol{x}_x = f_x$. Secondly, for lateral slip ${}^{\mathcal{R}}\boldsymbol{s}_y$, we use centrifugal force ${}^{\mathcal{R}}\boldsymbol{x}_y = \psi = (f_x f_\omega)$, proportional to commanded longitudinal and angular velocities. Thirdly, for angular slip ${}^{\mathcal{R}}\boldsymbol{s}_\omega$, we use three distinct slip learning inputs ${}^{\mathcal{R}}\boldsymbol{x}_\omega = [\psi, f_x, f_\omega]$. The first angular slip input is the vehicle's centrifugal force ψ. We then add UGV asymmetry, which can be caused by manufacturing imperfections and mechanical wear, causing angular velocity error proportional to commanded longitudinal velocity f_x. Finally, we account for the vehicle's skid, leading to an error between commanded angular velocity and actual angular velocity f_ω. It should be noted that the vehicle gravity-dependent parameters, used by Seegmiller *et al.* [9], are missing in this work. The reason is that we simplify our calibration protocol to be executed on planar terrain. The remainder of this section describes how we learn slip for a single dimension, but the process is the same for all dimensions of slip.

We use Bayesian linear regression (BLR) to estimate the values for $\boldsymbol{\gamma}$ and σ^2. For a more in-depth explanation of BLR, refer to the book written by Murphy [22]. It can be shown that the posterior for learned parameters $p(\boldsymbol{\gamma}, \sigma^2 | \mathcal{D}_\text{d})$ is distributed according to a Normal Inverse Gamma distribution $\text{NIG}(\boldsymbol{\gamma}, \sigma^2 | \boldsymbol{\gamma}, \boldsymbol{K}, a, b)$, where

$$\begin{aligned}\boldsymbol{\gamma} &= \boldsymbol{K}\left(\boldsymbol{K}_0^{-1}\gamma_0 + \boldsymbol{X}^T\boldsymbol{s}\right), \\ \boldsymbol{K} &= (\boldsymbol{K}_0^{-1} + \boldsymbol{X}^T\boldsymbol{X})^{-1}, \\ a &= a_0 + \frac{n}{2}, \\ b &= b_0 + \frac{1}{2}\left(\gamma_0^T\boldsymbol{K}_0^{-1}\gamma_0 + \boldsymbol{s}^T\boldsymbol{s} - \boldsymbol{\gamma}^T\boldsymbol{K}^{-1}\boldsymbol{\gamma}\right),\end{aligned} \tag{6}$$

where the estimated covariance of the distribution is represented by $\boldsymbol{K} \in \mathbb{R}^{k \times k}$. Priors for all parameters are defined by the $(\cdot)_0$ subscript. We define $\mathcal{D}_\text{d} = \{\boldsymbol{X}, \boldsymbol{s}\}$ as a training dataset consisting of vectors of n concatenated observed values for slip inputs $\boldsymbol{X}$ and observed slip velocities $\boldsymbol{s}$ for a specific dimension. The posterior equations can be used to train the BLR slip model for each dimension based on a training dataset $\mathcal{D}_\text{d}$. Once the model is trained, we can predict vehicle slip based on m test inputs $\tilde{\boldsymbol{X}} \in \mathbb{R}^{m \times k}$:

$$p\left(\hat{\boldsymbol{s}} | \tilde{\boldsymbol{X}}, \mathcal{D}_\text{d}\right) = \mathcal{T}\left(\boldsymbol{s} | \boldsymbol{X}\boldsymbol{\gamma}, \frac{b}{a}\left(\boldsymbol{I}_m + \boldsymbol{X}\boldsymbol{K}\boldsymbol{X}^T\right), 2a\right), \tag{7}$$

where $\mathcal{T}$ is a Student's t-distribution and $\hat{\boldsymbol{s}}$ represents a vector of m concatenated predicted slip velocities for a specific direction. In this work, we use an uninformative prior to ensure our protocol requires as little expertise as possible to execute. This consists of setting $a_0 = b_0 = 0$, $\boldsymbol{\gamma}_0 = \boldsymbol{0}$ and $\boldsymbol{K}_0 = \phi(\boldsymbol{X}^T\boldsymbol{X})^{-1}$ for any positive value ϕ. This allows to initialize our slip-based training model with little knowledge of the UGV except for wheel radius r and vehicle width b.

IV. Results

In this section, we evaluate the improvement of motion prediction accuracy when training models with the DRIVE protocol. We also analyze the number of training data required to reach convergence with our model. Finally, we demonstrate that for off-road navigation of SSMRs, learning vehicle slip based on dynamics-aware basis functions is more accurate than learning on vehicle acceleration.

A. Experimental Setup

We have conducted an extensive experimental evaluation of our calibration protocol and novel slip-based BLR model. Three distinct UGV platforms were used, as shown in Figure 4. First, we tested on a *Clearpath Robotics* Warthog on wheels, weighing $470\,\mathrm{kg}$, on gravel-covered terrain and an ice rink. The ice rink was leveled and recently resurfaced, leading to an extreme vehicle slip. Next, we tested on smaller platforms, namely a wheeled *Clearpath Robotics* Husky, weighing $75\,\mathrm{kg}$, and a tracked *Superdroid* HD2, weighing $80\,\mathrm{kg}$, both on indoor tile and snow-covered terrain. The Warthog has a top speed of $5\,\mathrm{m/s}$, which is around five times that of the HD2 at $1.2\,\mathrm{m/s}$ and of the Husky at $1\,\mathrm{m/s}$. These platforms and terrains were selected to maximize the difference in properties between experiments. For all platforms, localization ground truth is estimated through point-cloud registration with the iterative closest point (ICP) algorithm. This localization approach was selected to use a common, centimeter-accurate ground truth [23] across all indoor and outdoor experiments. The localization system for the Husky and HD2 robots is described in [24] and for the Warthog in [25]. For every experiment, the recorded data was split into two halves, the training dataset and the evaluation dataset, to enable extensive model evaluation. Our experimental dataset totals over $7\,\mathrm{km}$ and $1.8\,\mathrm{h}$ of driving data across all platforms and terrain types.

Fig. 4. Three different commercial platforms that were used for the experimental work: a *Superdroid* HD2 (1), a *Clearpath Robotics* Husky (2), and a *Clearpath Robotics* Warthog mounted on wheels (3). The platforms weigh $80\,\mathrm{kg}$, $75\,\mathrm{kg}$ and $470\,\mathrm{kg}$, respectively.

B. Protocol performance analysis

First, we define a function to evaluate model prediction performance. While learned models are trained on single-step vehicle slips or accelerations, our goal is to use them to predict vehicle motion over a specific horizon. One should note that we train our model on single-step slip velocities to simplify the learning problem. Williams *et al.* [6] showed that this simplification allows sufficient prediction performances for high-speed UGV path following. Thus, we use the multi-step root mean squared error (MRMSE) ϵ to evaluate prediction errors [14], with our localization as ground truth:

$$\epsilon = \frac{1}{h}\sum_{j=1}^{h}\sqrt{({}^{\mathcal{G}}\boldsymbol{q}_j - {}^{\mathcal{G}}\hat{\boldsymbol{q}}_j)^T\boldsymbol{\Sigma}({}^{\mathcal{G}}\boldsymbol{q}_j - {}^{\mathcal{G}}\hat{\boldsymbol{q}}_j)}\ , \qquad (8)$$

where h is the prediction window size. We define the measured state as ${}^{\mathcal{G}}\boldsymbol{q}$ and the model-predicted state as ${}^{\mathcal{G}}\hat{\boldsymbol{q}}$. All robots are commanded at $20\,\mathrm{Hz}$ and the prediction window length is set at $2\,\mathrm{s}$ reflecting the established path following work of Williams *et al.* [6]. We divide this error into translational MRMSE ϵ_T, for which $\boldsymbol{\Sigma} = \mathrm{diag}(1, 1, 0)$ and rotational MRMSE ϵ_R, for which $\boldsymbol{\Sigma} = \mathrm{diag}(0, 0, 1)$.

As a first evaluation, we evaluate the MRMSE for the HD2 on snow to highlight the improvement of using our input-space complete calibration protocol. Three distinct training datasets gathering approaches are shown, namely the linear-focused, similar to typical human driving [6], the angular-focused [18], and our DRIVE protocol. In the top subplot of Figure 5, we show the translational and angular error distributions for all three data-gathering methods. In the bottom subplot of Figure 5, it can be seen that the DRIVE protocol significantly outperforms other training data-gathering methods. Indeed, our data-gathering protocol offers a $31\,\%$ decrease in translation prediction error median when compared to the angular-focused approach and a $51\,\%$ decrease in rotation prediction error median when compared to the linear-focused approach. These values show that our input-space-aware DRIVE protocol allows gathering a dataset leading to decreased prediction error for both translation and rotation.

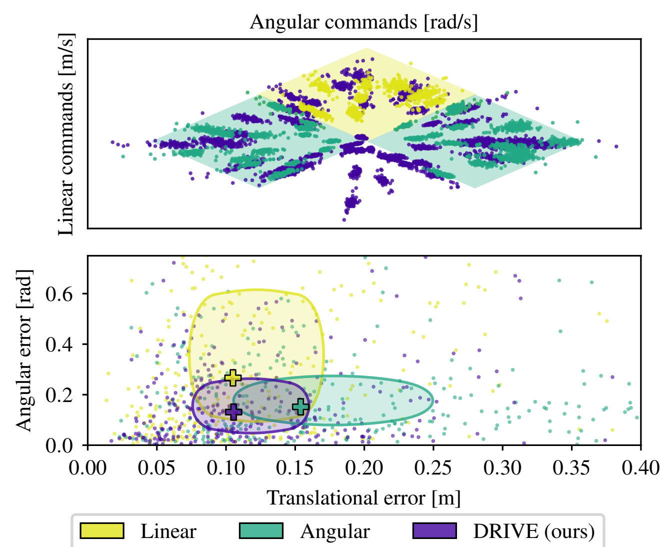


Fig. 5. Data-gathering protocol performance for the HD2 on snow experiment. The top subplot illustrates the three data-gathering methods compared in this work. In yellow, we have the linear-focused method. In teal, we have the angular-focused method. In blue-violet is our DRIVE approach. The crosses and regions on the bottom subplot show the medians and interquartile ranges for translational and angular prediction errors.

C. Slip-based learning predictive performance

In Section III-B, we propose a novel slip-based BLR model to predict UGV motion. Figure 6 shows this model's performance for both translational and rotational prediction, compared to another BLR learning approach based on vehicle acceleration [14]. We present three distinct datasets, namely HD2 on tile along with wheeled Warthog on gravel and ice. The rightmost results combine the prediction errors for all experiments conducted in this work. We also show the performance of the model provided by manufacturers (i.e., Naive) and the improvement done through powertrain modeling.

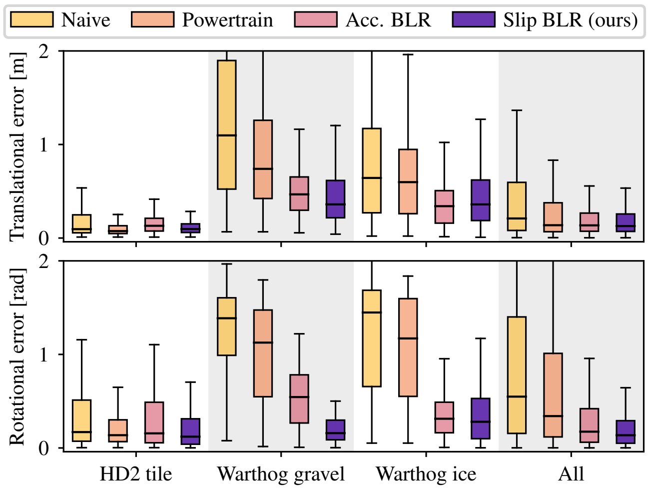


Fig. 6. Translational and rotational prediction errors for all models studied in this work. In yellow is the manufactured-defined naive model, in orange is the powertrain-aware model described in Section III-B.1, in red is the acceleration-based BLR model and in purple is our slip-based BLR model.

When accounting for all datasets, we observe a 34 % decrease in translation prediction error median and a 38 % decrease in rotation prediction error median when comparing the naive model with the powertrain-aware model. Also, our slip-based BLR approach leads to a 22 % decrease in rotation prediction error median and a 6 % decrease in translation prediction error median when compared to acceleration-based BLR. Looking at specific experiments, the Warthog in gravel shows the largest improvement between our slip BLR and acceleration BLR, with 71 % in rotation error median and 23 % in translation error median. In contrast, the HD2 on tile experiment shows a performance decrease for acceleration BLR and similar performance for slip BLR when compared to the powertrain model. Indeed, the indoor tile ground already had low prediction error for the powertrain-aware model. Lastly, the ice rink experiment shows similar performance between slip and acceleration BLR. This experiment corresponds to an extreme slip, similar to a UGV driving over black ice for an extended duration. This result shows the limit of our slip-based BLR model which still performs similarly or better than other models In this case, dynamic modeling could improve performance. Overall, we conclude that slip-based BLR offers improved performances for rotation prediction and similar performance in translation prediction over acceleration-based BLR, especially for driving at higher velocities on off-road terrains. For SSMRs in particular, rotation motion is the highest source of error due to the complexity of wheel-terrain skidding interactions [7], justifying the significance of our model.

Moreover, generating the training data is time and energy-consuming, which leads us to look for a trade-off between calibration duration and model prediction accuracy. Thus, we evaluated the relationship between training driving time and prediction accuracy. The results are shown in Figure 7. Three distinct experiments are presented, notably Husky on snow, HD2 on tile and Warthog on gravel. No other experiment is shown, to avoid cluttering, but similar results were observed. As specified in Section III-B.2, an uninformative prior is used for every platform, explaining the initially high errors.

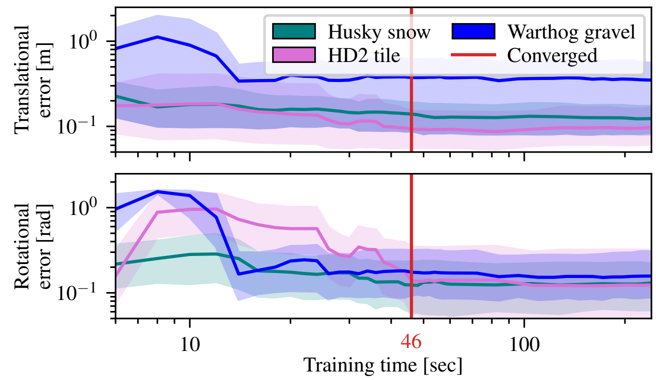


Fig. 7. The relation between training time and our slip-based BLR model prediction performance, for translation and rotation. Three datasets are shown, namely the Husky on snow in teal, the HD2 on tile in pink and the wheeled Warthog on gravel, in blue. We highlight at 46 s the converged value with the red line, for which our model converges for all UGVs tested. For all subplots, both axes are in log scale.

As shown by the red vertical line in Figure 7, the prediction accuracy stabilizes after 46 s of driving time for all robots. To compute this time, we evaluated the error gradient with respect to calibration time. We then evaluated the maximum time for which the translational and rotational error values for all shown experiments was under $0.01\,\mathrm{m/s}$ or $0.01\,\mathrm{rad/s}$, indicating all models have converged. Thus, users of the DRIVE protocol SSMRs can expect that the slip-based BLR motion model has converged after 46 s of training data, which is almost four times shorter than 180 s, the shortest training time observed in other work [15].

V. Conclusion

In this paper, we proposed *Data-driven Robot Input Vector Exploration (DRIVE)*, an automated vehicle characterization and training data generation protocol. We also propose a novel UGV prediction model called slip-based BLR. We show that training our model with our protocol offers improved prediction performances when comparing common training approaches and similar learning-based models. We also show that with our protocol, model convergence is reached with four times less driving time than the shortest similar protocol. We conclude that our protocol represents an efficient option for generating an initial motion model for UGVs. Future work would include generalizing our protocol to any vehicle geometry (e.g., Ackermann steering) and adapting our model formulation for complete dynamic models for extreme slip situations such as driving on surfaced ice. Adaptive modeling, relying on DRIVE to provide the initial training, should also be investigated.

References

[1] F. Dümbgen, C. Holmes, and T. D. Barfoot, "Safe and Smooth: Certified Continuous-Time Range-Only Localization," *IEEE Robotics and Automation Letters (RA-L)*, vol. 8, no. 2, pp. 1117–1124, 2023.

[2] R. Takemura and G. Ishigami, "Traversability-based Trajectory Planning with Quasi-Dynamic Vehicle Model in Loose Soil," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 8411–8417.

[3] L. Brunke, M. Greeff, A. W. Hall, *et al.*, "Safe Learning in Robotics: From Learning-Based Control to Safe Reinforcement Learning," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, no. 1, pp. 411–444, 2022.

[4] N. Seegmiller and A. Kelly, "High-Fidelity Yet Fast Dynamic Models of Wheeled Mobile Robots," *IEEE Transactions on Robotics (T-RO)*, vol. 32, no. 3, pp. 614–625, 2016.

[5] N. Seegmiller, F. Rogers-Marcovitz, G. Miller, and A. Kelly, "Vehicle model identification by integrated prediction error minimization," *The International Journal of Robotics Research (IJRR)*, vol. 32, no. 8, pp. 912–931, 2013.

[6] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Information-Theoretic Model Predictive Control: Theory and Applications to Autonomous Driving," *IEEE Transactions on Robotics (T-RO)*, vol. 34, no. 6, pp. 1603–1622, 2018.

[7] D. Baril, V. Grondin, S.-P. Deschenes, *et al.*, "Evaluation of Skid-Steering Kinematic Models for Subarctic Environments," in *17th Conference on Computer and Robot Vision (CRV)*, IEEE, 2020, pp. 198–205.

[8] A. Mandow, J. L. Martinez, J. Morales, J. L. Blanco, A. Garcia-Cerezo, and J. Gonzalez, "Experimental kinematics for wheeled skid-steer mobile robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007, pp. 1222–1227.

[9] N. Seegmiller and A. Kelly, "Enhanced 3D Kinematic Modeling of Wheeled Mobile Robots," in *Robotics: Science and Systems X (RSS)*, Robotics: Science and Systems Foundation, 2014.

[10] K. Bussmann, L. Meyer, F. Steidle, and A. Wedler, "Slip Modeling and Estimation for a Planetary Exploration Rover: Experimental Results from Mt. Etna," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 2449–2456.

[11] H. Yang, L. Ding, H. Gao, *et al.*, "High-Fidelity Dynamic Modeling and Simulation of Planetary Rovers Using Single-Input-Multi-Output Joints With Terrain Property Mapping," *IEEE Transactions on Robotics (T-RO)*, vol. 38, no. 5, pp. 3238–3258, 2022.

[12] J. Wang, M. T. H. Fader, and J. A. Marshall, "Learning-based model predictive control for improved mobile robot path following using Gaussian processes and feedback linearization," *Journal of Field Robotics (JFR)*, vol. 40, no. 5, pp. 1014–1033, 2023.

[13] L. Hewing, J. Kabzan, and M. N. Zeilinger, "Cautious Model Predictive Control Using Gaussian Process Regression," *IEEE Transactions on Control Systems Technology (T-CST)*, vol. 28, no. 6, pp. 2736–2743, 2020.

[14] C. D. McKinnon and A. P. Schoellig, "Learn Fast, Forget Slow: Safe Predictive Learning Control for Systems With Unknown and Changing Dynamics Performing Repetitive Tasks," *IEEE Robotics and Automation Letters (RA-L)*, vol. 4, no. 2, pp. 2180–2187, 2019.

[15] F. Djeumou, J. Y. Goh, U. Topcu, and A. Balachandran, "Autonomous Drifting with 3 Minutes of Data via Learned Tire Models," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 968–974.

[16] J.-F. Tremblay, T. Manderson, A. Noca, G. Dudek, and D. Meger, "Multimodal dynamics modeling for off-road autonomous vehicles," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 1796–1802.

[17] C. Voser, R. Y. Hindiyeh, and J. C. Gerdes, "Analysis and control of high sideslip manoeuvres," *Vehicle System Dynamics*, vol. 48, no. sup1, pp. 317–336, 2010.

[18] T. Wang, Y. Wu, J. Liang, C. Han, J. Chen, and Q. Zhao, "Analysis and Experimental Kinematics of a Skid-Steering Wheeled Robot Based on a Laser Scanner Sensor," *Sensors*, vol. 15, no. 5, pp. 9681–9702, 2015.

[19] S. Triest, M. Sivaprakasam, S. J. Wang, W. Wang, A. M. Johnson, and S. Scherer, "TartanDrive: A Large-Scale Dataset for Learning Off-Road Dynamics Models," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2022, pp. 2546–2552.

[20] R. Bellman, "Dynamic Programming," *Science*, vol. 153, no. 3731, pp. 34–37, 1966.

[21] J. Bergstra and Y. Bengio, "Random Search for Hyper-Parameter Optimization," *Journal of Machine Learning Research (JMLR)*, vol. 13, no. 10, pp. 281–305, 2012.

[22] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[23] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP variants on real-world data sets," *Autonomous Robots*, vol. 34, no. 3, pp. 133–148, 2013.

[24] K. Ebadi, L. Bernreiter, H. Biggie, *et al.*, "Present and Future of SLAM in Extreme Environments: The DARPA SubT Challenge," *arXiv preprint arXiv:2208.01787*, 2022.

[25] D. Baril, S.-P. Deschênes, O. Gamache, *et al.*, "Kilometer-scale autonomous navigation in subarctic forests: challenges and lessons learned," *Field Robotics*, vol. 2, no. 1, pp. 1628–1660, 2022.

Field Report

Kilometer-scale autonomous navigation in subarctic forests: challenges and lessons learned

Dominic Baril, Simon-Pierre Deschênes, Olivier Gamache, Maxime Vaidis, Damien LaRocque, Johann Laconte, Vladimír Kubelka, Philippe Giguère and François Pomerleau
Norlab, Université Laval, Québec, QC, Canada G1V 0A6

Abstract: Challenges inherent to autonomous wintertime navigation in forests include lack of a reliable Global Navigation Satellite System (GNSS) signal, low feature contrast, high illumination variations, and changing environment. This type of off-road environment is an extreme case of situations autonomous cars could encounter in northern regions. Thus, it is important to understand the impact of this harsh environment on autonomous navigation systems. To this end, we present a field report analyzing teach-and-repeat navigation in a subarctic forest while subject to fluctuating weather, including light and heavy snow, rain, and drizzle. First, we describe the system, which relies on point cloud registration to localize a mobile robot through a boreal forest, while simultaneously building a map. We experimentally evaluate this system in over 18.8 km of autonomous navigation in the teach-and-repeat mode. Over 14 repeat runs, only four manual interventions were required, three of which were due to localization failure and another one caused by battery power outage. We show that dense vegetation perturbs the GNSS signal, rendering it unsuitable for navigation in forest trails. Furthermore, we highlight the increased uncertainty related to localizing using point cloud registration in forest trails. We demonstrate that it is not snow precipitation, but snow accumulation, that affects our system's ability to localize within the environment. Finally, we expose some challenges and lessons learned from our field campaign to support better experimental work in winter conditions. Our dataset is available online.[1]

Keywords: SLAM, extreme environments, winter, navigation, GPS-denied operation

1. Introduction

Autonomous navigation has enabled mobile robots to be deployed in a wide variety of areas in order to support human operation. Specific examples include forestry (Oliveira et al., 2021),

[1] https://github.com/norlab-ulaval/Norlab_wiki/wiki/Kilometer-scale-autonomous-navigation-in-subarctic-forests:-challenges-and-lessons-learned

Received: 22 November 2021; revised: 22 May 2022; accepted: 7 June 2022; published: 15 July 2022.

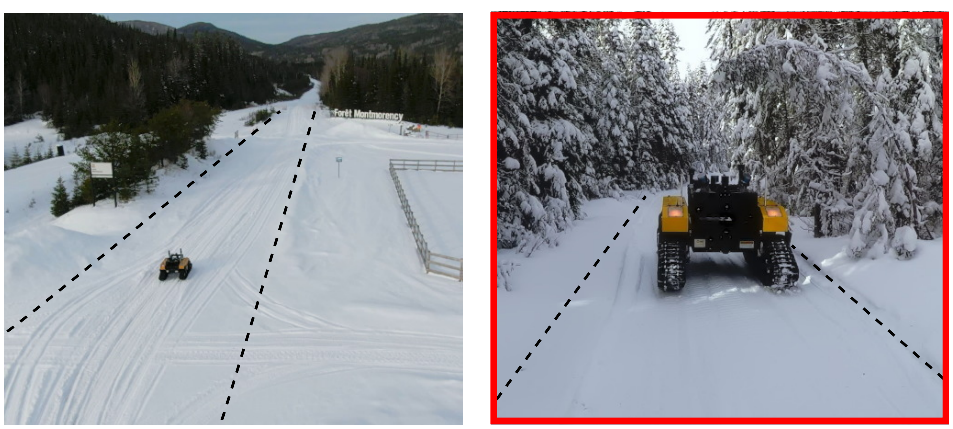
Correspondence: Dominic Baril, Norlab, Université Laval, Québec, QC, Canada G1V 0A6,
Email: dominic.baril@norlab.ulaval.ca

DOI: https://doi.org/10.55417/fr.2022050

mining (Marshall et al., 2016), disaster search and rescue (Kruijff et al., 2014), and military applications (Simon, 2015). Recently, Van Brummelen et al. (2018) have released a comprehensive review of the state-of-the-art perception technologies for autonomous vehicles. In this review, the authors identified key future challenges that need to be addressed for safer systems. One of these challenges is increasing the reliability of simultaneous localization and mapping (SLAM) algorithms to external factors such as dynamic environments, poor lighting, and weather conditions. In this regard, enabling true long-term autonomy for mobile robots will eventually require systems to be resilient to any road and weather condition.

To progress towards solving the challenges mentioned by Van Brummelen et al. (2018), this research study aims to examine the impact of subarctic environments and weather on the state-of-the-art autonomous navigation approaches. Access to the Montmorency boreal forest, located 70 km north of Québec City, Canada, during winter enabled us to deploy an autonomous system in difficult meteorological conditions, which allowed us to conduct this study. Thus, we present a field report on the deployment of an autonomous navigation framework in a boreal forest under harsh winter conditions. Subarctic regions are mostly covered by the boreal forest biome, and are thus ideal for this study. Boreal forests are characterized by dense, closed-crown conifer vegetation (Russell and Ritchie, 1988) and harsh winter weather. In the forest, we distinguish two path types: forest roads and forest trails, both shown in Figure 1. Forest roads are built to accommodate various vehicles, while forest trails are narrow and are not built to accommodate typical road vehicles. The dense vegetation surrounding forest trails is not traversable by most unmanned ground vehicles (UGVs), and thus autonomous navigation error tolerance is low when navigating on these trails. Moreover, dense vegetation is known to cause problems for autonomous navigation due to interference of the canopy with the global navigation satellite system (GNSS) signal (Kubelka et al., 2020). In addition to explaining the path types, Figure 1 also demonstrates the meteorological conditions of this field deployment. It was conducted over multiple days on a 0.7-m compacted snow cover, with varying precipitation and in subzero temperatures. These conditions complicate the logistics of the deployment, diminish the endurance both of the robotic system and the personnel, and require punctual planning. Yet, they reveal weaknesses of the contemporary robotic technology and point to new research problems.

To generate observations on the impact of subarctic environments on autonomous navigation technologies, we have built the Weather-Invariant Lidar-based Navigation (WILN) system. This



(a) Forest road. **(b)** Forest trail.

Figure 1. The main focus of this work is evaluating the impact of biome and weather on autonomous navigation. We distinguish two distinct path types, namely forest roads and forest trails. (a) The system navigating on a wide forest path, allowing greater system inaccuracies. (b) A forest trail, where the dense vegetation prevents the robot from navigating outside the trail and blocks the GNSS signal. The error tolerance for forest trail navigation is much lower than for forest roads. In this work, the majority of autonomous navigation is conducted in forest trails, as highlighted by the red mark on the right picture.

system is a minimal autonomous teach-and-repeat framework relying primarily on lidar sensor range measurements and point cloud registration for localization. Teach-and-repeat systems require a human operator to manually teach reference paths previously to repeating them autonomously. Typical 2D localization approaches eventually fail in outdoor, three-dimensional terrain such as boreal forest roads and trails (Krüsi et al., 2015). Sensor noise due to floating particles is also known to have a strong impact on 2D localization reliability (Ren et al., 2021). Thus, the WILN system relies on 3D lidar scans to localize. Kilometer-scale environments are inherently challenging for lidar-based localization, especially due to dynamically changing environments, lack of geometrical constraints, and high computation cost related to registering point clouds within large environments. The WILN system is specifically designed and tuned to overcome such challenges and enable us to gather observations on autonomous navigation in boreal forests.

The specific contributions of this paper are (i) a comprehensive study of the impact of the boreal forest biome on lidar- and GNSS-based localization and autonomous navigation; (ii) an overview of the impact of snow accumulation on the reliability of lidar-based localization over multiple days; and (iii) a description of the WILN system, designed to enable wintertime autonomous navigation in a boreal forest. The remainder of this paper is organized as follows: Section 2 overviews the related work and robotic deployments performed in winter weather conditions. It also compares related GNSS-denied long-range autonomous navigation algorithms. Section 3 describes the WILN framework in detail, including both our localization pipeline and the path-following controller. Section 4 describes the environment in which we conducted our deployments and presents the system hardware and the implementation parameters of the WILN framework. Section 5 provides the results of the field trials and Section 6 discusses them in the context of the forest and subarctic conditions. It also presents the lessons learned during this deployment. Finally, we conclude this paper in Section 7.

2. Related Work

The aim of this paper is to present the impact of the boreal forest and winter conditions on autonomous navigation technologies with the goal of enabling true long-term robot autonomy. In this section, we show that while various off-road robotic deployments in winter conditions are documented in the literature, they mostly rely on the GNSS signal for localization (Lever et al., 2013). Vision-based localization approaches have enabled autonomous navigation in GNSS-denied environments. However, winter conditions have been shown to affect the performance of such approaches (Paton et al., 2017). Wintertime autonomous navigation in a boreal forest requires localization capabilities that are resilient to both winter conditions and GNSS-denied environments. Active sensors such as lidars are ideal for solving this problem since they are robust to lighting variation (Krüsi et al., 2015).

GNSS-based localization has been the standard for autonomous navigation in polar environments. The first rover to have been deployed in polar regions is *Nomad*, a gasoline-powered UGV weighing 725 kg. *Nomad* was stationed at Elephant Moraine, Antarctica, for a duration of four weeks with the goal of autonomously identifying meteorites (Apostolopoulos et al., 2000). The robot reached speeds upwards of 0.5 m s while using differential GNSS as the primary method of localization. The platform also used the stereo cameras and the lidar sensor for obstacle detection. However, stereo vision was found to be ineffective on blue ice and snow in Antarctica due to extreme lack of texture (Moorehead et al., 1999). Additionally, several operators were required in order to manually avoid undetected rocks. Expanding on this work, *MARVIN I* and *MARVIN II* were two diesel-powered skid-steering mobile robots (SSMRs) weighing 720 kg. *MARVIN I* and *MARVIN II* were deployed in Greenland (Stansbury et al., 2004) and Antarctica (Gifford et al., 2009), respectively. The goal of these robots was to increase survey safety in remote polar regions. The requirement of heavy sensor payloads led to the selection of large vehicles. *MARVIN I* used real-time kinematics (RTK) GNSS as the primary method of localization, achieving a centimeter-level accuracy in open environments. *MARVIN II* did not rely solely on RTK GNSS. When navigating out of the RTK reference receiver range, it transferred from RTK to differential GNSS. Transfer from RTK to differential GNSS increased rover oscillation with respect to the reference path. Through the *MARVIN I* and *II* deployments, Stansbury et al. (2004) highlighted the fact that turning maneuvers with SSMRs were linked to a high risk of immobilization in deep snow. Later, *Yeti*, a battery-powered 81 kg UGV, was deployed to conduct ground penetrating radar surveys in order to detect subsurface crevasses or other voids in 2010, 2011, and 2012 (Lever et al., 2013). Knowledge about such crevasses would enable human operators to plan safer paths for larger vehicles. *Yeti* did not include any obstacle detection system. During surveys, *Yeti* reached a top speed of 2.2 m /s and managed to acquire data on hundreds of crevasse encounters. It even located a previously undetected buried building in the South Pole. No obstacle detection and avoidance system was implemented on *Yeti* due to the open nature of ice sheets. Additionally, *Yeti* was deployed for further surveys in the McMurdo shear zone in 2014 (Arcone et al., 2016) and 2015 (Ray et al., 2020), while still relying on GNSS waypoint navigation. Based on the recorded data, the authors have presented a method to estimate ice sheet velocity fields by matching annual ground penetrating radar scans. The aforementioned work on field robotics in polar regions allowed the identification of various issues related to autonomous navigation on snow-covered terrain. Due to the complex nature of localizing in polar regions, GNSS is the most popular means of localization for such regions. However, subarctic navigation includes cluttered environments such as boreal forests. Autonomous navigation in those areas requires localization robust to GNSS-denied conditions, for which multiple approaches have been presented in the literature.

Since the GNSS signal is not always reliable due to the multipath effect or signal absorption due to tree canopy (Kubelka et al., 2020), approaches independent of these effects have been proposed. Furgale and Barfoot (2010) were the first to show that visual teach and repeat (VT&R) approaches are robust enough to perform large-scale autonomous navigation in GNSS-denied environments. The authors have deployed their system in the Canadian High Arctic. This environment was selected because of its similarity to lunar and Martian terrain. Most features consisted of rocks located within the reference trajectory. In this work, they successfully repeated reference paths up to 10 hours after they were manually driven. However, sensitivity to illumination change was identified as the main limit of the system. Thus, Churchill and Newman (2013) later introduced *experience-based navigation* to increase the robustness of VT&R to scene appearance change, caused by illumination variation or dynamic environment changes. This feature was added in VT&R through *multiexperience localization*, with the added ability to use landmarks from previous experiences in the same localization problem (Paton et al., 2016). In this work, the authors extend the allowable time between teach-and-repeat runs from a few hours to multiple days. Paton et al. (2015) also added color-constant image transformations to VT&R to mitigate the impact of illumination variations. Color-constant image transformations have been used by Clement et al. (2017) to perform autonomous route repeating by using the VT&R framework while relying solely on a monocular camera. While vision-based localization was demonstrated to be robust to illumination variation, Paton et al. (2015) have observed that localization frameworks relying only on passive cameras fail to localize in dark conditions.

To enable nighttime vision-based navigation, McManus et al. (2013) have proposed to use an intensity-based lidar to replace cameras for the VT&R framework. While this work demonstrates that this system is resilient to low illumination conditions, it suffers from motion distortion issues. Using headlights, MacTavish et al. (2017) proposed a bag-of-words approach to prioritize experiences most relevant to live operation. This in turn allows a growing number of robot experiences while limiting computation requirements. In this work, the authors have successfully repeated paths over a 31-h period, including day and night driving relying on headlights. Extending the experimental evaluation of VT&R, Paton et al. (2018) have logged over 140 km of autonomous navigation in an untended gravel pit, also including nighttime navigation. Clement et al. (2020) have used a deep neural network to learn a nonlinear color transform mapping that maximizes vision-based localization resiliency to appearance change. In this work, they successfully localized on routes over a 30-h period by relying only on a single experience. However, Congram and Barfoot (2021) still observed vision-based localization failures in low illumination, even when using headlights. In this work, the authors proposed to fuse vision-based localization with GNSS measurements to enable VT&R systems to function in areas where vision localization fails.

Vision-based localization was tested on snow-covered terrain numerous times. *Sno-mote Mk1* and *Mk2* were deployed on Alaskan glaciers and Wapekoneta, Ohio (Williams and Howard, 2009). *Sno-motes* are dual-drive 1:10 scale snowmobiles equipped with a single camera and GNSS. These robots were used to conduct manually driven traverses of about 100 m at a speed of $1\,\mathrm{m\,s^{-1}}$. The data gathered with the *Sno-motes* were then used to enhance visual simultaneous localization and mapping (SLAM) feature extraction methods in snow. Despite improving feature detection methods on snow, it was shown that snow is still feature sparse (Williams and Howard, 2009). Paton et al. (2017) then showed that vision-based localization is robust to intraseasonal, daily scene appearance change, by successfully repeating more than 26 km over multiple days. In this work, VT&R was also deployed on a 250-m path featuring a 0.3-cm snow cover. However, deep-snow path following leads to unstable UGV behavior due to features almost only being observed on the horizon, leading to inaccurate pose estimates, which led to path-following instability. MacTavish et al. (2018) have used multiexperience localization to successfully repeat trajectories over 100 days through day, night, winter, spring, and summer. In their work, the authors showed that vision-based localization is resilient to significant seasonal change as long as the UGV can repeat the path at a rate faster than scene appearance change. In our paper, we investigate the performance of a complementary approach relying on lidar measurements and observe its limitations under harsh winter weather and in boreal forest environments.

On the other hand, lidar-based localization is resilient to illumination variation, which can lead to localization failure for vision-based systems. Marshall et al. (2008) were the first to suggest a lidar teach and repeat (LT&R) using encoders and 2D lidars. In their work, a sequence of locally consistent and overlapping topometric maps (i.e., occupancy grids) are recorded along the path using 2D lidar measurements to allow the robot to localize during the repeat phase. The system was proven efficient for repeating paths in underground tunnels on a 10 t capacity hauler. Still relying only on 2D lidar scans for localization, Sprunk et al. (2013) have proposed to localize directly on 2D lidar scans, removing the requirement to build a topometric map offline. This system was deployed in an indoor, structured environment, resulting in a millimetric localization accuracy. Later, Mazuran et al. (2015) improved this framework by introducing a trajectory optimization step between the teach and the repeat phases, while still deploying the system in a similar indoor environment. Maddern et al. (2015) have studied the use of multiexperience localization, similarly to Churchill and Newman (2013), however, this time using scans measured by a push-broom 2D lidar. Local 3D swathes are produced by fusing 2D scans with vehicle odometry. These swathes are then matched to multiple prior experiences, allowing to deal with structural change in an urban environment. We show that our approach requires less memory as it relies on a single reference map and demonstrate its performance in an off-road, complex environment.

Related to off-road environments, Nieto et al. (2003) have proposed the FastSLAM algorithm relying on 2D lidar scans, which they have tested on the Victoria Park dataset. This dataset was recorded within an urban park, on uneven terrain and through sparse vegetation. In this work, the various trees in the park were used as landmarks to localize with the FastSLAM algorithm. Later, Jagbrant et al. (2015) deployed a similar lidar-based localization system in an almond orchard. However, an almond tree orchard contains significantly sparser vegetation than the boreal forest. Zhang and Singh (2018) have deployed a lidar and inertial measurement unit (IMU)-based framework in a sparse forest environment. In this work, they have shown resilience to multiseasonal change by merging a summer and winter map. However, due to the location where this work was conducted, no analysis of the impact of snowfall is discussed. Recently, Ren et al. (2021) deployed a lidar localization system in a desert biome. They identified the lack of features and geometrical constraints as an issue for point cloud registration. This issue is similar to the low feature contrast problem that affects vision-based approaches in snow-covered terrain (Paton et al., 2017).

As lidars are subject to noise created by precipitation in the environment, point cloud denoising has been studied in the literature. Schall et al. (2005) and Jenke et al. (2006) have proposed probabilistic approaches which tend to be computationally expensive. On the other hand, Schall et al. (2008) have proposed a neighborhood-based approach that is viable for real-time point cloud denoising. In a similar neighborhood-based approach, Charron et al. (2018) have proposed a dynamic radius outlier removal filter to denoise point clouds recorded by a self-driving vehicle during light snowfall in an urban setting. Duan et al. (2021) later proposed a principal-component analysis method to filter lidar scans, yielding increased performance. Utilizing the progress made on semantic segmentation, Heinzler et al. (2020) have proposed a learning-based approach to denoise point clouds, allowing to use information from the entire scene rather than the vicinity of specific points. While most point cloud denoising approaches work as input filters applied to lidar scans, we apply a post filter on the map after point cloud registration to remove dynamic points, as proposed by Pomerleau et al. (2014). We show through our results that our system is robust to real-time localization through moderate snowfall. We also show that it is not snowfall, but rather snow accumulation, that affects system performance, potentially leading to system failure.

Relying on the iterative closest point (ICP) algorithm, Krüsi et al. (2015) have deployed a LT&R system in off-road environments and busy city streets, successfully repeating paths of up to 1.3 km. These deployments have shown the resiliency of lidar-based perception to off-road and urban environments, under high illumination variations. However, this deployment does not cover the impact of dense vegetation and snowfall. In previous work, we have shown that, using the ICP algorithm to register 3D lidar scans, we can produce large-scale maps of a boreal forest offline (Babin et al., 2019). This paper aims to describe the field deployment of the WILN framework in a boreal forest during 5 days, effectively subjecting it to weak GNSS signal, high illumination variations, and low feature contrast. Such deployment conditions combine the challenges of navigation in snow-covered and GNSS-denied constrained environments. We report the impact of dense vegetation in forest trails and of snow accumulation on the performance of the WILN system.

3. Weather-Invariant Lidar-based Navigation (WILN) System Description

WILN is an autonomous teach-and-repeat system designed to be robust to kilometer-scale navigation, severe weather, and GNSS-denied conditions. As a localization prior, the system relies on IMU measurements and the wheel odometry, as described in Section 3.1. Registering 3D lidar scans through the ICP algorithm is the primary means of localization (Section 3.2). Map maintenance and tiling modules were added to enable kilometer-scale SLAM, both of which are described in Section 3.3. As for most teach-and-repeat frameworks, two modes are defined: the *teach phase* and the *repeat phase* (Section 3.4). A kinematic controller was selected to compute appropriate commands to solve the path-following problem (Section 3.5). This section ends with a system overview, including a visual representation of all WILN system components and their interactions (Section 3.6).

Four coordinate frames are defined in the WILN system, as illustrated in Figure 2. First, the global map frame $\mathcal{G}$ is defined, representing the world the robot navigates in. Second, the robot frame $\mathcal{R}$ is defined with its origin in the base of the robot chassis. The x axis of the robot frame $\mathcal{R}$ is parallel to the longitudinal direction and the y axis is parallel to the lateral direction of the robot. Third, the lidar frame $\mathcal{L}$ coincides with the lidar sensor origin. The rigid transform from the robot frame to the lidar frame ${}^{\mathcal{L}}_{\mathcal{R}}\boldsymbol{T}$ is assumed to be constant and found through system calibration. The reading point clouds (i.e., input points) $\mathcal{P}$ are originally observed in the lidar frame $\mathcal{L}$ and the reference point clouds (i.e., map points) are expressed in the map frame $\mathcal{G}$. The transform ${}^{\mathcal{G}}_{\mathcal{R}}\boldsymbol{T}$ between the frames $\mathcal{R}$ and $\mathcal{G}$ is updated by the ICP algorithm and constitutes the robot localization. For the path-following algorithm running in the repeat phase, the Frenet-Serret frame $\mathcal{S}$ is defined directly on the path. Lastly, the reference trajectory recorded in the teach phase is expressed as a vector of consecutive robot poses $\boldsymbol{x}_{\text{ref}} = \{{}^{\mathcal{G}}_{\mathcal{R}}\boldsymbol{T}_1, {}^{\mathcal{G}}_{\mathcal{R}}\boldsymbol{T}_2, \ldots, {}^{\mathcal{G}}_{\mathcal{R}}\boldsymbol{T}_n\}$, where n is the number of recorded poses in the trajectory.

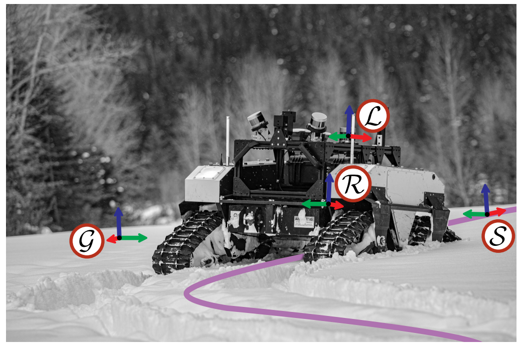


Figure 2. Coordinate frames used for WILN. In this instance, the robot is moving towards the path frame $\mathcal{S}$. The reference path that the robot aims to follow is drawn in light purple. The remaining relevant coordinate frames are the map frame $\mathcal{G}$, the lidar frame $\mathcal{L}$, and the robot frame $\mathcal{R}$.

3.1. Localization Prior: IMU and Wheel Odometry

The ICP algorithm requires high-frequency prior estimates of ${}^{\mathcal{G}}_{\mathcal{L}}\check{\boldsymbol{T}}$ that capture the motion between every two consecutive lidar scans. In WILN, the prior ${}^{\mathcal{G}}_{\mathcal{L}}\check{\boldsymbol{T}}$ is estimated through IMU measurements and wheel odometry. The robot orientation is estimated using the Madgwick filter[2] (Madgwick et al., 2011) based on gyroscope and accelerometer measurements. Linear displacement is based on wheel odometry, while taking into account the estimated robot orientation. The prior ${}^{\mathcal{G}}_{\mathcal{L}}\check{\boldsymbol{T}}$ is generated synchronously with the IMU at a frequency of 100 Hz. When subject to motion, it should be noted that lidar sensors generate skewed reading point clouds ${}^{\mathcal{L}}\mathcal{P}_s$. Hence, we deploy a point cloud deskewing filter in our system. First proposed by Bosse and Zlot (2009), such algorithms correct point cloud distortions by taking the lidar intrascan motion into account. Following the same idea, our high-frequency ${}^{\mathcal{G}}_{\mathcal{L}}\check{\boldsymbol{T}}$ prior is used to deskew the raw input point cloud. The resulting corrected ${}^{\mathcal{L}}\mathcal{P}$ is the one used in the ICP algorithm.

3.2. Iterative Closest Point (ICP)

To localize the robot, the incoming deskewed reading point clouds ${}^{\mathcal{L}}\mathcal{P}$ are registered to the reference point cloud ${}^{\mathcal{G}}\mathcal{Q}$ using the ICP algorithm. This allows a map of the environment to be built during the teach phase and enables the robot to be localized in this map during the repeat phase. An overview of the ICP pipeline is shown in Figure 3. The ICP algorithm iteratively matches points between two point clouds and looks for a rigid transform that minimizes the distance between each pair of the matched points. Pomerleau et al. (2015) presented a comprehensive review of the state of the art for the ICP algorithm. The WILN registration component is based on the modules presented in this review. To increase robustness of the algorithm, we apply the following three input filters to the reading ${}^{\mathcal{L}}\mathcal{P}$ before registering it into the reference ${}^{\mathcal{G}}\mathcal{Q}$:

1. **Random subsampling filter** parameterized by the ratio $\eta_s \in [0, 1]$ of the points kept after the subsampling. This subsampling is critical to reducing the computation time of the ICP algorithm to allow the SLAM problem to be solved in real time (Pomerleau et al., 2011).
2. **Bounding box filter** parameterized by the bounding box coordinates $\boldsymbol{b}_i = (x_{\min}, x_{\max}, y_{\min}, y_{\max}, z_{\min}, z_{\max}) \in \mathbb{R}^6$. It removes points originating from the robot body that would otherwise cause a trail of points in the reference point cloud $\mathcal{Q}$.
3. **Radius filter** parameterized by the maximum radius r around the lidar. Beyond this radius, the reading points are discarded. This allows only the relevant vicinity around the robot to be considered to further reduce the computation time.

[2] https://github.com/bjohnsonfl/Madgwick_Filter

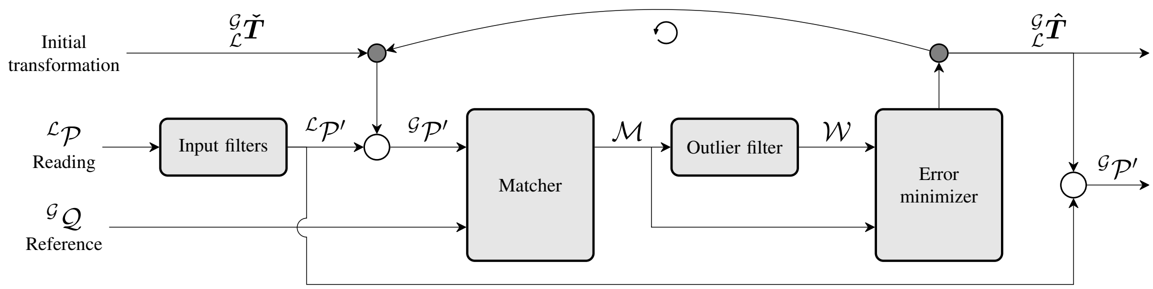


Figure 3. The ICP pipeline. The reading point cloud ${}^{\mathcal{L}}\mathcal{P}$ is filtered, and the initial transformation ${}^{\mathcal{G}}_{\mathcal{L}}\check{\boldsymbol{T}}$ is applied to it. The matcher finds sets of neighbors in ${}^{\mathcal{G}}\mathcal{Q}$ for each point of ${}^{\mathcal{G}}\mathcal{P}'$, which compose the set of matches $\mathcal{M}$. An outlier filter is then used to compute weights $\mathcal{W}$ associated with the matches $\mathcal{M}$. The weights $\mathcal{W}$ and the matches $\mathcal{M}$ are then used in the error minimizer to compute an output transformation ${}^{\mathcal{G}}_{\mathcal{L}}\hat{\boldsymbol{T}}$. The matching, outlier filtering, and error minimization steps are done iteratively until the error is beneath the settled threshold.

In our implementation, ${}^{\mathcal{L}}\mathcal{P}$ is observed in the lidar frame $\mathcal{L}$ and ${}^{\mathcal{G}}\mathcal{Q}$ is defined in the map frame $\mathcal{G}$. Thus, the ICP algorithm estimates the transform ${}^{\mathcal{G}}_{\mathcal{L}}\hat{\boldsymbol{T}}$ by minimizing an error function e using

$$ {}^{\mathcal{G}}_{\mathcal{L}}\hat{\boldsymbol{T}} = \arg\min_{\boldsymbol{T}} \left[\mathrm{e}\left({}^{\mathcal{G}}\mathcal{P}, {}^{\mathcal{G}}\mathcal{Q} \right) \right], \tag{1} $$

where ${}^{\mathcal{G}}\mathcal{P}$ is the reading point cloud $\mathcal{P}$ expressed in the map frame $\mathcal{G}$. For the first iteration, the prior ${}^{\mathcal{G}}_{\mathcal{L}}\check{\boldsymbol{T}}$ is used to compute this rigid transformation. The prior computation is explained in Section 3.1. To compute the error function, we associate points between the reading point cloud and the reference point cloud. Following this step, the ICP algorithm computes the optimal transform by iteratively minimizing the error between ${}^{\mathcal{G}}\mathcal{P}$ and ${}^{\mathcal{G}}\mathcal{Q}$, as specified in Equation 1. For better clarity, the remainder of this section concentrates on a single ICP iteration step that would be repeated until convergence.

Point matching is first done by finding the closest points in ${}^{\mathcal{G}}\mathcal{Q}$ for each point of ${}^{\mathcal{G}}\mathcal{P}'$. Thus, multiple points of ${}^{\mathcal{G}}\mathcal{Q}$ can be associated with each point of ${}^{\mathcal{G}}\mathcal{P}'$. Nearest-neighbor search is carried out via the use of a kd-tree[5] to decrease computation time. The matcher is parameterized by the number of nearest neighbors for each $\mathcal{P}$ point $n_m \in \mathbb{Z}_{>0}$ and the maximum allowable distance for a match, $d_{\max} \in \mathbb{R}_{>0}$. In order to speed up the nearest-neighbor searches, the $\varepsilon \in \mathbb{R}_{>0}$ parameter is set to 1 to allow approximations, as described in Arya and Mount (1993). Subsequently, we apply an outlier filter to add binary weights to each matched point to remove the outlier matches from the error function. The outlier filter adds a positive binary weight to the $\eta_d \in [0,1]$ proportion of nearest matches. Formally, let $\mathcal{M}_{\text{all}} = \text{match}(\mathcal{P}, \mathcal{Q}) = \{(\boldsymbol{p}, \boldsymbol{q}) \in \mathcal{P} \times \mathcal{Q}\}$ be the set of matches between $\mathcal{P}$ and $\mathcal{Q}$. We also define $\mathcal{M} \subseteq \mathcal{M}_{\text{all}}$, which contains only the n_m closest pairs of $\mathcal{M}$ for each point of $\mathcal{P}$ with a Euclidean distance below $d_{\max}$. Let $\mathcal{W} = \text{outlier}(\mathcal{P}, \mathcal{Q}) = \{\mathrm{w}(\boldsymbol{p}, \boldsymbol{q}) : (\boldsymbol{p}, \boldsymbol{q}) \in \mathcal{M}\}$ be the weights associated with these matches. Our system uses the point-to-plane error function, defined by

$$ \mathrm{e}(\mathcal{P}, \mathcal{Q}) = \sum_{k=1}^{K} \mathrm{w}(\boldsymbol{p}_k, \boldsymbol{q}_k) \left\| (\boldsymbol{p}_k - \boldsymbol{q}_k) \cdot \boldsymbol{n}_k \right\|_2, \tag{2} $$

where K is the number of matches in $\mathcal{M}$ and $\|\cdot\|_2$ is the L2 norm. The normal vector $\boldsymbol{n}_k$ around the 3D point $\boldsymbol{q}_k$ in $\mathcal{Q}$ is computed prior to the ICP algorithm. The error in Equation 2 can be iteratively minimized by recomputing the set of matched points $\mathcal{M}$ and their associated weights w at each iteration. Moreover, in the minimization process, we only optimize the translation and the yaw rotation, assuming that the prior roll and pitch angles are already optimal. This holds in the case of precise IMU calibration with respect to the robot frame $\mathcal{R}$. Transformation checkers are added to WILN to detect erroneous ICP solutions. An error is raised if the prior error exceeds predefined thresholds in translation error $\varepsilon_{t_{\min}} \in \mathbb{R}_{>0}$ or angular error $\varepsilon_{\theta_{\min}} \in \mathbb{R}_{>0}$. Additionally, the iterative process of the ICP algorithm is stopped after a maximum number of iterations, $i_{\max} \in \mathbb{Z}_{>0}$, is reached, returning the last transform. The resulting transform ${}^{\mathcal{G}}_{\mathcal{L}}\hat{\boldsymbol{T}}$ can be used to express the 3D

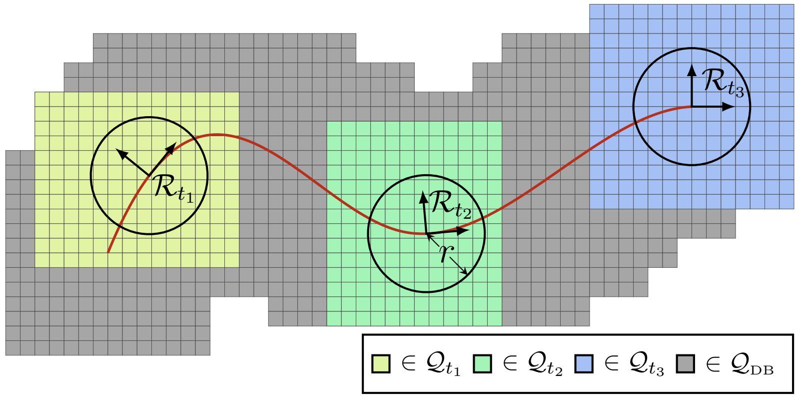


Figure 4. The management of the map by the *voxel manager* module. The robot trajectory is represented by the red line. Three distinct robot poses are represented by $\mathcal{R}_{t_1}$, $\mathcal{R}_{t_2}$, and $\mathcal{R}_{t_3}$. The local map ${}^{\mathcal{G}}\mathcal{Q}$ voxels are represented by the light yellow, green, and blue colors and the nonlocal map ${}^{\mathcal{G}}\mathcal{Q}_{\mathrm{DB}}$ voxels by the dark gray color.

robot pose in $\mathcal{G}$ by chaining ${}^{\mathcal{G}}_{\mathcal{L}}\hat{\boldsymbol{T}}\,{}^{\mathcal{L}}_{\mathcal{R}}\boldsymbol{T}$ since the latter was evaluated through system calibration. The ICP algorithm also returns the reading point cloud in the map frame ${}^{\mathcal{G}}\mathcal{P}$. Our implementation is strongly based on the `libpointmatcher` library (Pomerleau et al., 2013).[3] This library allows fast registration of large point clouds, which is essential to solving the SLAM problem in real time.

3.3. Large-Scale Mapping

The large-scale mapping subsystem uses the ICP implementation described in Section 3.2 to solve the SLAM problem in large-scale navigation. At each time step, a lidar scan ${}^{\mathcal{L}}\mathcal{P}$ is measured and registered to the reference map ${}^{\mathcal{G}}\mathcal{Q}$, yielding the transformed and filtered reading ${}^{\mathcal{G}}\mathcal{P}$ and the estimated transform between the lidar and map frames ${}^{\mathcal{G}}_{\mathcal{L}}\hat{\boldsymbol{T}}$. The registered scan ${}^{\mathcal{G}}\mathcal{P}$ is appended to the reference point cloud ${}^{\mathcal{G}}\mathcal{Q}$. To maintain a point density in the map, a point is only appended if its distance to the closest point in the map exceeds the user-defined minimal threshold $\rho \in \mathbb{R}_{>0}$. After merging the input point cloud into the map, two post-filters are applied to the map:

1. The **surface normal filter** computes the surface normals for all points required by the point-to-plane minimization described in Equation 2. For each map point, the positions of the $n_n \in \mathbb{Z}_{>0}$ closest points are taken into account to estimate the local surface normal.
2. The **dynamic point filter** removes points that are identified as dynamic, originating from, e.g., walking pedestrians, or falling snow. Its implementation is based on the one proposed by Pomerleau et al. (2014). The dynamic filter allows dynamic elements to be removed from the map, allowing the map to be maintained for extended periods of time. To filter out the dynamic points, ray tracing is used. If an incoming scan ${}^{\mathcal{L}}\mathcal{P}$ point is located behind a map point ${}^{\mathcal{L}}\mathcal{Q}$, the probability of this map point being dynamic is increased. If this probability surpasses the predefined threshold $\tau_d \in [0, 1]$, it is effectively removed from ${}^{\mathcal{G}}\mathcal{Q}$. To limit computation time for this filter, map points located further than the sensor range r do not enter the filtering process.

This new map ${}^{\mathcal{G}}\mathcal{Q}'$ is sent to the *voxel manager* module, illustrated in Figure 4. The purpose of the voxel manager is to limit the computational complexity of the map maintenance, instead of letting it grow with its size. Our implementation of the voxel manager module is similar to the one proposed by Ren et al. (2021). It involves the following steps: the new map ${}^{\mathcal{G}}\mathcal{Q}'$ is divided into voxels, then only the voxels that are close to the robot are kept in the random access memory (RAM). All points

[3] https://github.com/ethz-asl/libpointmatcher

located within those voxels are considered to be the local map ${}^{\mathcal{G}}\mathcal{Q}$. The remaining voxels represent the nonlocal map ${}^{\mathcal{G}}\mathcal{Q}_{\mathrm{DB}}$ and are stored in the database on the system hard drive.

The local map ${}^{\mathcal{G}}\mathcal{Q}$ voxels constitute a cube following the position of the $\mathcal{L}$ frame. The center of this cube is maintained close to the origin of the $\mathcal{L}$ frame. The initial length of its edge is defined as twice the lidar sensor range r plus a margin of two voxels on each side. This margin gives the manager enough time to load and unload voxels as the robot moves through the map. The loading and unloading operations are triggered each time the robot crosses two distinct voxel borders, preventing unnecessary computations if the robot oscillates on a single voxel border. The side effect is the varying number of voxels in the local map ${}^{\mathcal{G}}\mathcal{Q}$, as can be observed in the example for the second robot pose $\mathcal{R}_{t_2}$ in Figure 4. In this case, tile loading will be triggered once the sensor range crosses the next voxel frontier, which explains why only one voxel separates the sensor range from the local map edge. The voxel manager allows the maximum map maintenance computation time to be limited, allowing the system to solve the SLAM problem for kilometer-scale environments. This module is parametrized by the voxel size $v_s \in \mathbb{R}_{>0}$, in meters.

3.4. Teach and Repeat Phases

During the teach phase, the robot is driven along a specific path by a human operator. The large-scale mapping framework presented in Section 3.3 is used to build the reference map. The local ${}^{\mathcal{G}}\mathcal{Q}$ and nonlocal ${}^{\mathcal{G}}\mathcal{Q}_{\mathrm{DB}}$ map voxels are saved to a map database. The sequence of robot poses $\boldsymbol{x}_{\mathrm{ref}}$ estimated by the ICP algorithm is subsampled to keep only points that are at least d_{ref} apart from each other. This subsampled trajectory is also saved in the database and constitutes the reference path $\boldsymbol{x}_{\mathrm{ref}}$ for the repeat phase. The system can be taught various reference trajectories and maps, all of which are stored on the robot's hard drive.

During the repeat phase, the complete reference map and the reference path $\boldsymbol{x}_{\mathrm{ref}}$ are first loaded from the database. The voxel manager then rebuilds the local map ${}^{\mathcal{G}}\mathcal{Q}$ and the nonlocal map ${}^{\mathcal{G}}\mathcal{Q}_{\mathrm{DB}}$ voxels according to the robot's position. In this phase, no points are appended to the reference map and the ICP algorithm is only used to localize the robot in the local map. One should be cognizant that snow accumulation is heterogeneous in nature, meaning that it is significantly different on dense vegetation, buildings, and flat ground. Thus, dynamic map updates were not added to the repeat phase to ensure system reliability during this deployment. The estimated pose of the robot ${}^{\mathcal{G}}_{\mathcal{R}}\hat{\boldsymbol{T}}$ is then projected as a planar pose $\boldsymbol{x}_{\mathrm{2D}}$ and used by the path-following algorithm described in Section 3.5. The path-following algorithm computes the output system commands $\boldsymbol{u}$ to steer the UGV along the $\boldsymbol{x}_{\mathrm{ref}}$ poses. Note that we rely on a metrically consistent map, contrarily to VT&R, which relies on viewpoint-based localization (Furgale and Barfoot, 2010). This choice allows us to maintain a single 3D map, which is memory efficient comparatively to viewpoint-based localization. Indeed, the latter requires storing all experiences within a database, eventually leading to significant memory usage.

3.5. Path Following

Once the robot's pose ${}^{\mathcal{G}}_{\mathcal{R}}\hat{\boldsymbol{T}}$ has been estimated and the reference trajectory $\boldsymbol{x}_{\mathrm{ref}}$ loaded from the database, this information is used as the input to the path-following controller. The outputs of the path-following algorithm are the commanded longitudinal and angular velocities, respectively defined in the vector $\boldsymbol{u} = [v_x, \omega] \in \mathbb{R}^2$. The WILN framework uses the orthogonal-exponential (OE) controller to compute the command vector $\boldsymbol{u}$. This controller computes angular velocity through the orthogonal projection of the robot pose on the reference path and various heuristics to compute linear velocity (Huskić et al., 2017). For the WILN system implementation, we have used the Generic Robot Navigation (GeRoNa)[4] library proposed by Huskić et al. (2019) to allow fast computation of the command vector $\boldsymbol{u}$.

[4] https://github.com/cogsys-tuebingen/gerona

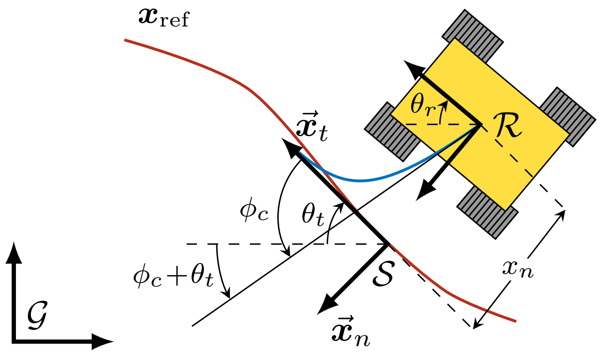


Figure 5. Main components of the orthogonal-exponential (OE) path-following algorithm used in this work. In red is the reference trajectory. In blue is the predicted exponential trajectory. The Frenet-Serret frame $\mathcal{S}$ is located on the referential trajectory.

At each time step, a control law is built to allow the robot to adjust its orientation to converge to the reference path $\boldsymbol{x}_{\text{ref}}$, as shown in Figure 5. A Frenet-Serret frame $\mathcal{S}$ is defined in the map frame. Its origin lies in the orthogonal projection of the robot frame $\mathcal{R}$ on the reference path. After, an exponential function is defined to allow the robot to converge to the reference path, drawn in blue in Figure 5. The angle of this exponential function is defined by

$$\phi_c = \arctan(-kx_n \exp(-kx_t)), \tag{3}$$

where $k \in \mathbb{R}_{>0}$ is a constant that allows the regulation of the convergence speed of the robot to the path. We also denote $x_t \in \mathbb{R}$ and $x_n \in \mathbb{R}$ as the current position of $\mathcal{R}$ in path frame $\mathcal{S}$, along the tangential $\vec{\boldsymbol{x}_t}$ and normal $\vec{\boldsymbol{x}_n}$ axes, respectively. Linear velocity is computed by modulating the target vehicle velocity v_n. Thus, the complete control law can be defined by

$$\boldsymbol{u} = \begin{bmatrix} v_x \\ \omega \end{bmatrix} = \begin{bmatrix} v_n \exp\left(-\left(\frac{K_g}{d_g}\right)\right) \\ K_h(\phi_c + \theta_e) \end{bmatrix}, \tag{4}$$

where $\theta_e \in \mathbb{R}_{>0}$ is the error between the robot angle θ_r and the path frame angle θ_t in the global frame $\mathcal{G}$. The $K_h \in \mathbb{R}_{>0}$ parameter is a gain on commanded angular velocity that was added in order to reduce controller overshoot when subject to high reference path curvature κ. In the original implementation, a parameter exists allowing to reduce commanded longitudinal velocity in areas with high path curvature κ. However, this parameter was omitted due to noise in the teach phase localization resulting in jerking motion for the UGV. The distance between the current robot pose and the end of the trajectory is represented by $d_g \in \mathbb{R}_{>0}$ and a goal proximity gain $K_g \in \mathbb{R}_{>0}$ is defined to gradually slow the robot as it reaches its goal. To respect the limitations of the robotic platform, the commanded angular velocity is limited to $\omega \in [-\omega_m, \omega_m]$. The commanded linear velocity is bounded to $v_x \in [v_{\min}, v_{\max}]$. A goal tolerance $\tau_g \in \mathbb{R}_{>0}$ parameter is used to allow the robot to finish the path repetition when within an acceptable distance from the $\boldsymbol{x}_{\text{ref}}$ end. Finally, a safety tolerance $\tau_w \in \mathbb{R}_{>0}$ parameter is defined to stop the robot if the distance between the origin of the robot's frame $\mathcal{R}$ and the closest trajectory point of $\boldsymbol{x}_{\text{ref}}$ exceeds the specified distance τ_w.

3.6. System Overview

An overview of the entire WILN pipeline is shown in Figure 6. During the teach phase, an operator drives the robot along the desired trajectory. Sensor measurements are used to solve the SLAM problem and build a database containing all reference maps and trajectories. A point cloud deskewing system is used to take intrascan lidar motion into account. This same system is used to produce a transformation prior ${}^{\mathcal{G}}_{\mathcal{L}}\check{\boldsymbol{T}}$ to allow the ICP algorithm to perform real-time point cloud registration. A map maintenance module is used to append the latest registered lidar scan ${}^{\mathcal{G}}\mathcal{P}$ to the local map

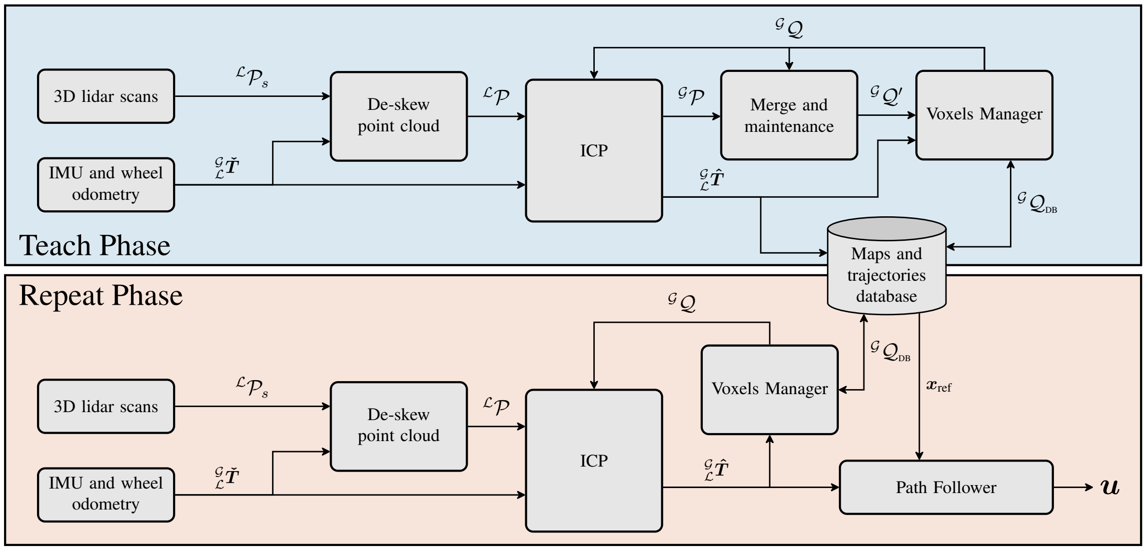


Figure 6. Overview of the WILN pipeline. The teach phase (in blue) takes as input the point clouds from the lidar and the odometry. The information is then used by the deskewing algorithm and sent into ICP, which is used to solve the SLAM problem. At the end of the teach phase, the local ${}^{\mathcal{G}}\mathcal{Q}$ and nonlocal ${}^{\mathcal{G}}\mathcal{Q}_{\text{DB}}$ maps are saved in the database. The inputs of the repeat phase (in red) are the same as for the teach phase. The transformation outputted by ICP is finally used by the path follower to compute the commands sent to the vehicle.

${}^{\mathcal{G}}\mathcal{Q}$. This module is also used to compute surface normals and remove dynamic points from the map, yielding the maintained map ${}^{\mathcal{G}}\mathcal{Q}'$. Then, the voxel manager module is used to split the map into the latest local map ${}^{\mathcal{G}}\mathcal{Q}$ and nonlocal map ${}^{\mathcal{G}}\mathcal{Q}_{\text{DB}}$. At the end of the teach phase, the entire map and reference trajectory $\boldsymbol{x}_{\text{ref}}$ are saved in the database for later use in the repeat phase. For the repeat phase, the ICP algorithm registers incoming lidar scans identically as in the teach phase. However, only the estimated robot pose ${}^{\mathcal{G}}_{\mathcal{L}}\hat{\boldsymbol{T}}$ is used. The reference trajectory $\boldsymbol{x}_{\text{ref}}$ and current robot pose ${}^{\mathcal{G}}_{\mathcal{L}}\hat{\boldsymbol{T}}$ are then sent to the path follower, which outputs the command vector $\boldsymbol{u}$. Since no point clouds are appended to the map, no map maintenance module is used during the repeat phase. The voxel manager module is used to build the local and nonlocal maps; only the former is taken into account by the ICP algorithm.

4. Experimental Setup

In this work, we conducted our deployment within the Montmorency boreal forest, located at a latitude of 47°19′15″N and a longitude of 70°9′0″W The deployment took place during winter, creating ideal conditions to evaluate the impact snowfall and dense vegetation can have on autonomous navigation. The hardware used to perform autonomous navigation is weatherproof and can navigate steep and soft terrain. Section 4.1 describes the winter-resilient hardware used to deploy the WILN system, including the UGV, sensing and computing hardware. Next, details about the various implementation parameters are presented in Section 4.2. Lastly, Section 4.3 lists the characteristics of the Montmorency boreal forest, the weather and conditions to which our system was subjected when navigating.

4.1. Hardware Description

Our system was deployed on a modified Clearpath Robotics Warthog UGV, as shown in Figure 7. The Warthog is a SSMR using two drive units located on each side of its chassis. For SSMRs, steering is done by rotating the wheels on each side of the vehicle at different velocities to create a

	Characteristics	Value
Physical	Mass	590 kg
	Footprint	2.13 m x 1.52 m
	Top speed	18 km/h
	Steering geometry	Skid-steering
	Locomotion	CAMSO ATV T4S Tracks
	Suspension	Geometric Passive Articulation
Sensors	Lidar	Robosense RS-32 (10 Hz)
	IMU	XSens MTi-10 (100 Hz)
	Wheel encoders	3 x Hall effect sensors (4 Hz)
	GNSS	Emlid Reach-RS+ (5 Hz)
Computing	Computer	Acrosser AIV-Q170V1FL
	CPU	i7-6700 TE (3.40 GHz)
	↳ Number of cores	4
	↳ Number of threads	8
	↳ RAM	16 Gb

Figure 7. Experimental setup for the WILN system on our Clearpath Robotics Warthog UGV. Left: Detailed specifications. Right: The numbers in red circles correspond to (1) Robosense RS-32 lidar, (2) XSens MTi-10 IMU, and (3) two Emlid Reach-RS+ GNSS antennas. The two GNSS antennas are not used for the localization of the WILN system.

skidding effect, effectively turning the vehicle. Vehicle motion control is achieved through a sub-servo system allowing each side's wheel velocity to controlled through Sevcon Gen4 drives and wheel encoders signal. A kinematic linear mapping between wheel velocities and body velocities allows body-velocity commands to be sent directly to the platform. Instead of wheels, the Warthog is mounted on four CAMSO ATV T4S tracks to maximize mobility, as depicted in Figure 7. The Warthog is also equipped with a differential suspension, increasing traction when navigating steep terrain. A Robosense RS-32 3D lidar is mounted in front of the robot, with no pitch or roll inclination. This lidar has a 200-m detection range and produces about 640 000 points per second. Three Hall effect sensors are added to each motor to provide wheel odometry for the robot. Completing the WILN sensor package, an XSens MTi-10 IMU provides angular velocity and body linear acceleration measurements. Additionally, two Emlid Reach-RS + RTK GNSS receivers were added to the robot chassis and a third was used as a fixed antenna to produce GNSS localization measurements. An Acrosser AIV-Q170V1FL onboard computer is used to record sensor data and perform all of the WILN system computations. All technical specifications for the platform are given in Figure 7.

4.2. Implementation Parameters

To allow the WILN framework to perform large-scale navigation, we hand-tuned the parameters of each subsystem. All parameters are enumerated in Section 3. Without conducting extensive calibration, we let the system repeat several short paths and manually adjusted parameters until observing stable performance. For this work, parameters were tuned to reach a working state for the WILN system, but no sensitivity analysis was conducted. Our goal was to evaluate the impact of the boreal forest and winter conditions on the system without extensive effort on system calibration. The mapping subsystem had three distinct goals: (i) enabling point cloud registration by computing surface normals, (ii) maintaining the map to remove dynamic points, and (iii) splitting the map into local and nonlocal through the voxel manager. This would prevent localization failures during repeat runs. Path-following overshoot in tight curves and controller oscillations were also minimized by adjusting the parameters. An overview of the tuned parameters is provided in Table 1.

Table 1. Parameter values related to each function used in our system. No extensive tuning was conducted, rather we have identified a minimal working system. Functions and parameters are split between the registration, mapping, and path-following subsystems. All functions and parameters are detailed in Section 3.

	Function	Parameters
Registration	Bounding box input filter 1	$\boldsymbol{b}_1 = [-1.5\,\text{m},\ 0.5\,\text{m},\ -1\,\text{m},\ 1\,\text{m},\ -1\,\text{m},\ 0.5\,\text{m}]$
	Bounding box input filter 2	$\boldsymbol{b}_2 = [-10\,\text{m},\ -1.5\,\text{m},\ -2.5\,\text{m},\ 2.5\,\text{m},\ -1\,\text{m},\ 1\,\text{m}]$
	Random sampling input filter	$\eta_s = 0.7$
	Maximum radius input filter	$r = 80\,\text{m}$
	KD tree matcher	$n_m = 7 \quad \varepsilon = 1 \quad d_{\max} = 2.0\,\text{m}$
	Trimmed distance outlier filter	$\eta_d = 0.7$
	Differential transformation checker	$\varepsilon_{\theta_{\min}} = 0.001 \quad \varepsilon_{t_{\min}} = 0.01$
	Counter transformation checker	$i_{\max} = 40$
Mapping	Large scale mapping	$\rho = 0.1\,\text{m} \quad v_s = 20\,\text{m}$
	Surface normal points filter	$n_n = 15$
	Dynamic points filter	$\tau_d = 0.8$
Path-following	Waypoint tolerance	$\tau_w = 1.0\,\text{m}$
	Goal tolerance	$\tau_g = 0.15\,\text{m}$
	Regulator path convergence	$k = 0.4$
	Compensation angular speed command	$K_h = 3.0$
	Max angular velocity	$\omega_m = 1.0\,\text{rad}\,\text{s}^{-1}$
	Goal position factor	$K_g = 0.5$
	Linear nominal speed	$v_{\text{nom}} = 1.5\,\text{m}\,\text{s}^{-1}$
	Linear minimal speed	$v_{\min} = 0.5\,\text{m}\,\text{s}^{-1}$
	Linear maximal speed	$v_{\max} = 1.5\,\text{m}\,\text{s}^{-1}$
	Reference trajectory pose distance	$d_{\text{ref}} = 5\,\text{cm}$

4.3. Environment

For this work, autonomous navigation was conducted in the Montmorency boreal forest. This environment is ideal to help highlight how dense vegetation and snow precipitation affects the performance of lidar and GNSS-based autonomous navigation, which are the main contributions of this paper. A digital terrain model of the deployment area, with a representation of the path network, is shown in Figure 8. It can be seen that three different paths were defined, all linking two points of interest (POIs). The goal of defining three different paths is to highlight the difference in localization performance between the two path types. Path A links the *Garage* and *Cabin* POIs through a cross-country ski trail and has the total distance of 1.4 km. According to guidelines, we expect this ski trail to have a minimal width of 4.5 m (Ministère des Forêts, de la Faune et des Parcs, 2017). Path B also links the *Garage* and *Cabin* POIs and has a total distance of 1.5 km. Path B is identical to path A for the first third of the path and then diverts to a foot trail, where the width is similar to the one observed on path A. We observed that multiple areas of path B have a lower width than the expected 4.5 m since this is a foot trail. To prevent UGV immobilization, the snow on paths A and B had been compacted by a snowmobile operator prior to the experiment. Path C connects the *Garage* and *Gazebo* POIs mostly through a network of wider roads and has a total distance of 0.5 km. Path C corresponds to a forest road, with a minimum width of 9.1 m according to official guidelines of the Montmorency boreal forest (Ministère des Forêts, de la Faune et des Parcs, 2021). It should be noted that path C was conducted as a round-trip path, consisting of a forward pass and a backward pass. This choice was made to increase autonomous navigation data and include data with the UGV driving in both directions. Also, it can be observed in the above mean sea level (AMSL) model in Figure 8 that the path network is located in a valley, surrounded by various mountains.

During the deployment week, the WILN system was subjected to fluctuating weather, including light and heavy snow, hail, and drizzle. We have access to extensive data gathered through a

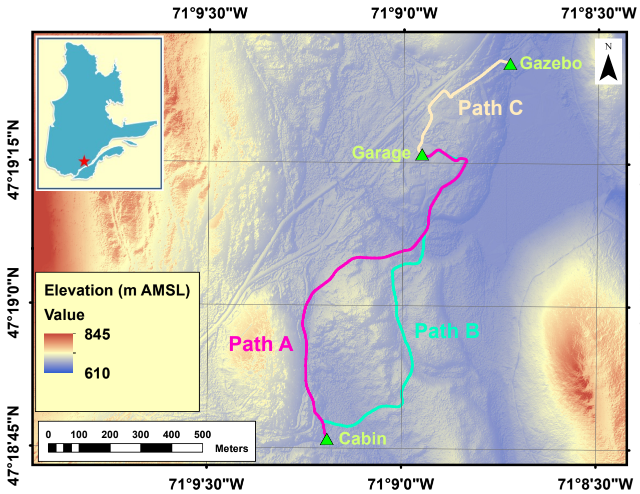


Figure 8. A digital terrain model of the Montmorency forest, where the WILN system was deployed, and the forest location in Québec, Canada (top left). We see the three different paths, both paths *A* (1.4 km) and *B* (1.5 km) going from the *Garage* points of interest (POI) to the *Cabin* POI, while path *C* (0.5 km) connects the *Garage* POI to the *Gazebo* POI. Global latitudinal and longitudinal coordinates are given in the margin. The coloring is defined by Above Mean Sea Level (AMSL) elevation. Image credit: Remote and Wood Sensing Laboratory, University of New Brunswick.

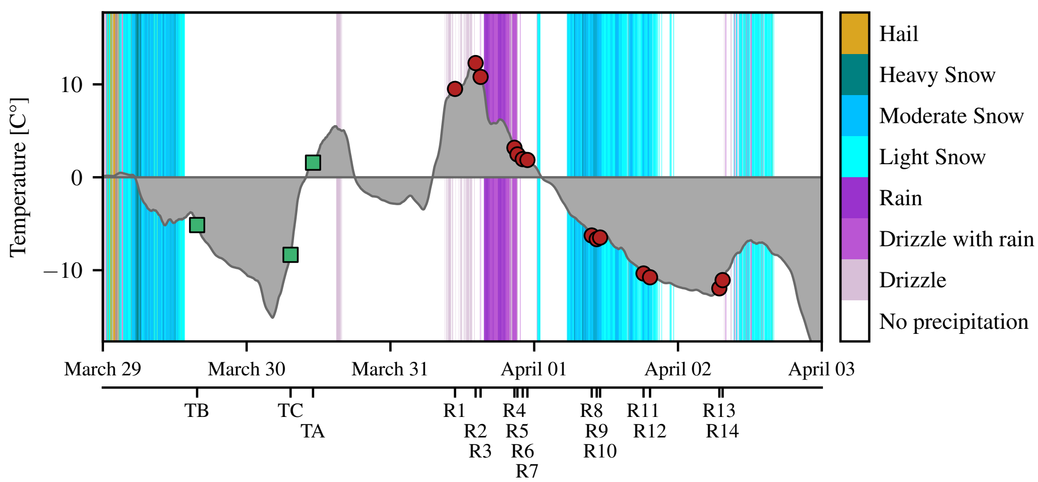


Figure 9. Temperature and weather during the deployment. Teach runs are depicted as green squares whereas repeat runs are depicted as red circles. Teach runs have been performed in relatively clear weather, while repeat runs suffered from rain and different levels of snowfall.

meteorological station located at the Montmorency forest. The area temperature was measured with a temperature probe of model 107 made by Campbell Scientific, which was covered by an antiradiation screen 2 m away from the ground. The precipitation type was measured with a Parsivel[2] disdrometer made by OTT. An overview of the meteorological conditions is illustrated in Figure 9. Outside temperatures ranged from −15.5 °C to 13.4 °C. The temperature fluctuated over and under the water freezing point (i.e., 0 °C), meaning that snow in the environment melted on some occasions and froze on others, effectively accumulating on the ground. Teach runs (i.e., TA, TB, and TC) were conducted when there was no precipitation. Runs R1 through R7 were conducted in rainy weather, while runs R8 through R12 were conducted in snowfall. The standard for weather codes shown in Figure 9 is code table 4680 for automatic weather stations (World Meteorological Organization, 2019). During runs R13 and R14, the system was not subject to any precipitation. While not shown in this figure, we performed autonomous repeats during both daytime and nighttime, with no impact on system performance. More details on each run are given in Section 5.

5. Results

The main goal of this work is to evaluate the impact of the boreal forest and winter weather on autonomous navigation performance. First, we document all three distinct reference maps and trajectories that were built through the teach phase. Then, we report on all repeat runs that were conducted over the deployment week. As a prior to further analysis, general information on all runs conducted during the deployment is presented. A general performance report for the system is shown in Section 5.1. Afterwards, we highlight the effects of the boreal forest biome, more specifically of the dense vegetation, on GNSS-based and ICP-based localization in Section 5.2. Then, Section 5.3 describes how snow precipitation and accumulation induces significant change in the environment landscape, resulting in ICP localization failures. Finally, Section 5.4 explains the impact of the environment on the OE controller path-following performance.

Teach runs were performed to record the reference maps ${}^{\mathcal{G}}\mathcal{Q}_{\mathrm{DB}}$ and the trajectories $\boldsymbol{x}_{\mathrm{ref}}$ for all three paths. During these runs, we manually drove the robot, while the map and trajectory database were built online. The resulting maps for each path are illustrated in Figure 10 with the coordinates defined in the map frame $\mathcal{G}$ specific to each path. In this visualization, the z component of the estimated surface normal vector is used to color each point. Thus, dark blue points represent walls, as can be seen on path C, green points typically represent vegetation, and yellow points represent ground surfaces. Paths A and B are mostly located within dense vegetation. On the other hand, path C is located on a much more open forest road. All reference paths in Figure 10 have been divided into areas to facilitate upcoming discussion. Table 2 presents relevant details about each

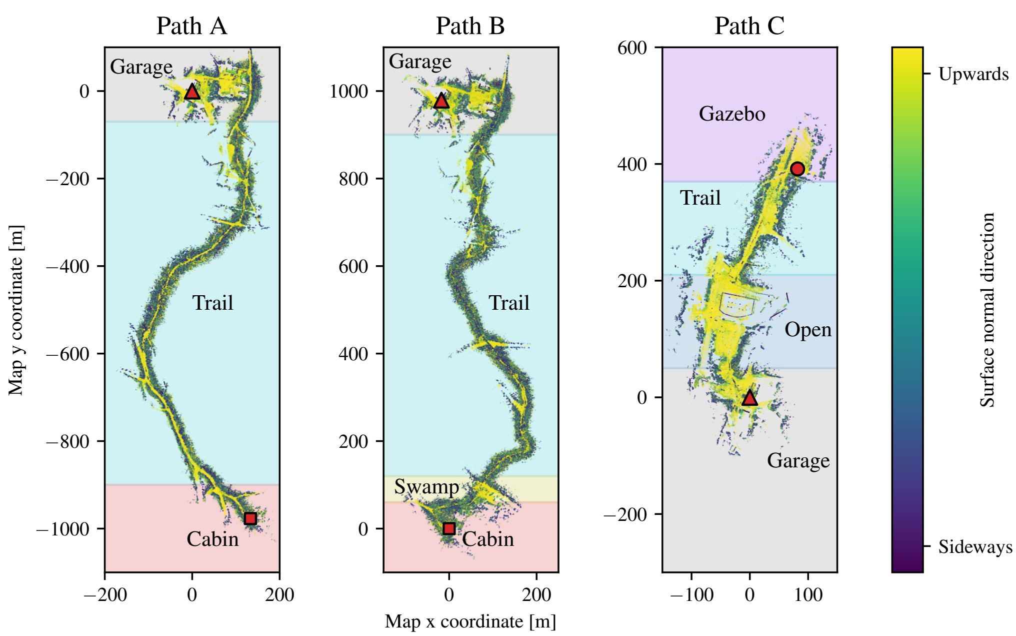


Figure 10. All reference maps recorded during the three teach phases for this work. The colormap represents the z component of the surface normal for all points in the map. Yellow points are the ones with the surface normal pointing upwards. Green and dark blue points have a surface normal pointing sideways. The red markers represent the reference trajectory POIs. All maps have been rotated in order to allow visualization in the same figure. The path areas identified here are used throughout the remainder of this section. The red markers represent the POIs associated to each path. The origin of path B is located at the *Cabin* POI.

Table 2. Overview of all runs conducted in this work. All times are defined in the local Eastern Standard Time. *A*' or *B*' means the run started at the *Cabin* POI. The column Δt defines the elapsed time since the teach run of the associated path. The column $\# \mathcal{P}$ shows the number of scans for each run. The illumination measured at the start of each run is given. Teach runs are not accounted for in the total autonomous distance traveled.

ID	Path	Start time (2021)	Duration (hh:mm)	Δt (hh:mm)	$\# \mathcal{P}$	Distance (km)	Illumination ($W\,m^{-2}$)	Interventions
TB	*B*'	March 29th 15:46	00:27	00:00	15 715	–	283.35	–
R2	*B*	March 31st 14:12	00:27	46:26	16 183	1.5	295.10	0
R5	*B*'	March 31st 21:14	00:27	53:28	16 056	1.5	0.00	0
R6	*B*	March 31st 22:05	00:27	54:19	16 163	1.5	0.00	0
R9	*B*'	April 1st 10:27	00:27	66:41	16 093	1.5	123.90	2
R11	*B*	April 1st 18:18	00:32	74:32	19 443	1.5	0.75	1
R14	*B*'	April 2nd 07:28	00:27	87:42	16 073	1.5	188.42	0
TC	*C*	March 30th 07:29	00:21	00:00	12 762	–	102.58	–
R1	*C*	March 31st 10:46	00:23	27:17	13 758	1.0	284.80	0
R10	*C*	April 1st 11:03	00:20	51:34	12 162	1.0	159.23	0
TA	*A*	March 30th 11:06	00:25	00:00	15 142	–	586.38	–
R3	*A*'	March 31st 15:04	00:26	27:58	15 535	1.4	101.55	0
R4	*A*	March 31st 20:44	00:26	33:38	15 641	1.4	0.00	0
R7	*A*'	March 31st 22:49	00:15	35:43	8862	**0.8**	0.00	1
R8	*A*	April 1st 09:25	00:28	46:19	15 623	1.4	84.68	0
R12	*A*'	April 1st 19:19	00:29	56:13	17 218	1.4	0.00	0
R13	*A*	April 2nd 06:55	00:26	67:49	15 583	1.4	99.63	0
Total	14 runs	–	07:13	–	258 012	18.8	–	4

of the teach and repeat runs conducted during the deployment. A total of 14 repeat runs were completed, summing up to 18.8 km and 7 h 13 min of autonomous repeating over 5 days. The last repeat run was started over 87 h after its respective teach run on path *B*. A battery power outage prevented us from completing run **R7**; it was therefore interrupted. The system also suffered from three localization failures in runs R9 and R11. For each run, the sun radiation measured by a CNR4 radiometer equipped with a CNF4 ventilation unit made by Kipp & Zonen located within the forest is given.

5.1. General Performance Report

This section aims to give a general overview of our system's performance and show that we achieved enough autonomy to generate observations on navigation in boreal forests. To characterize the reliability of our system, we computed the cross-track error, which is the distance between the robot frame $\mathcal{R}$'s origin and its orthogonal projection on the path, as defined by Mondoloni et al. (2005). The cross-track error for every run is shown in Figure 11. It can be seen that this error generally stays below 1.0 m for all runs, with some peaks corresponding to the various turns in each reference trajectory. This error is below the cross-country trail half-width of 2.25 m, as specified in Section 4.3. Note that UGV initialization was done using visual markers in the *Garage* area, but not in the *Cabin* area. The resulting high initialization error explains the high cross-track error at the end of paths *A* and *B*. Additionally, it can be seen in the last column of Table 2 that a total of four manual interventions were done through the deployment. The three interventions done through runs R9 and R11 were due to localization failure. The localization failure on run R11 occurred in the *Trail* area of path *B*. At this point, a large drift in localization caused the robot to divert from the trail and hit a tree, leading to a cross-track error peak, as can be seen in Figure 11. Two more localization failures occurred in the *Cabin* area of path *B* during run R9. For these failures, it can be seen in Figure 9 that run R9 occurred during a snowstorm, leading to significant snow accumulation on the ground. An in-depth analysis of this event is presented in Section 5.3. The intervention done

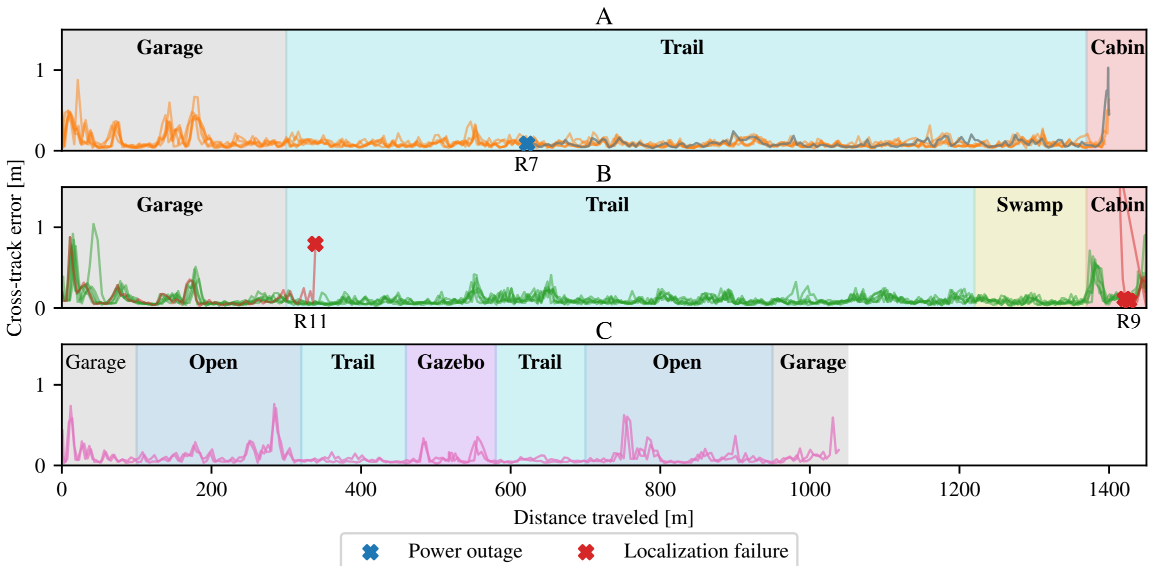


Figure 11. Cross-track error with respect to distance traveled for all runs. The results from paths *A*, *B*, and *C* are plotted in orange, green, and pink, respectively. The three localization failures from runs R9 and R11 are depicted as red crosses. Run R7, which ended prematurely due to a battery outage, is depicted as a blue cross. The shaded areas for each path correspond to those shown in Figure 10.

in run R7 was due to battery power outage. Excluding the intervention done due to battery power outage, there has been an average of one intervention per 4.7 km traveled during this deployment. Thus, we conclude that the WILN system showed a satisfactory degree of autonomy to generate observations on autonomous navigation in boreal forests.

5.2. Impact of the Boreal Forest Biome

The first main goal of this work is to evaluate the impact of dense vegetation inherent to boreal forests on both GNSS and lidar-based localization for autonomous navigation. During all repeat runs, the GNSS signal statistics were logged in both receivers mounted on the Warthog UGV. Additionally, measurements were taken from a fixed reference receiver during each run to enable RTK-GNSS positioning. The data were post-processed offline using `RTKLIB`[5] (Takasu and Yasuda, 2009) to produce the most accurate measurements possible. To estimate the GNSS localization error, we compared the manually measured distance between the GNSS receivers mounted on the UGV with the distance indicated by the GNSS localization, similarly as in Vaidis et al. (2021). This metric does not take into account localization biases to which both receivers might be subjected; thus we consider the error metric an optimistic one. The results of the GNSS error with respect to the mean number of satellites between the two mobile receivers are presented in Figure 12. Our dataset does not contain a sufficient number of data points with less than seven satellites; we therefore exclude them from our analysis. This shows that for reliable GNSS localization on forest trails, the minimum number of observed satellites, n_{cs}, is 18. If a lower number of satellites is observed, the system is at risk of collision with the vegetation at the edges of the path. Since the off-road paths are wider, they allow slightly lower n_{cs} of 16.5 for safe navigation.

Based on the critical number of satellites determined in Figure 12, we have conducted an analysis of all reference trajectories to determine the areas considered to be GNSS denied. Figure 13 shows the mean number of observed satellites in a georeferenced satellite image. On paths *A* and *B*, the

[5] http://www.rtklib.com/

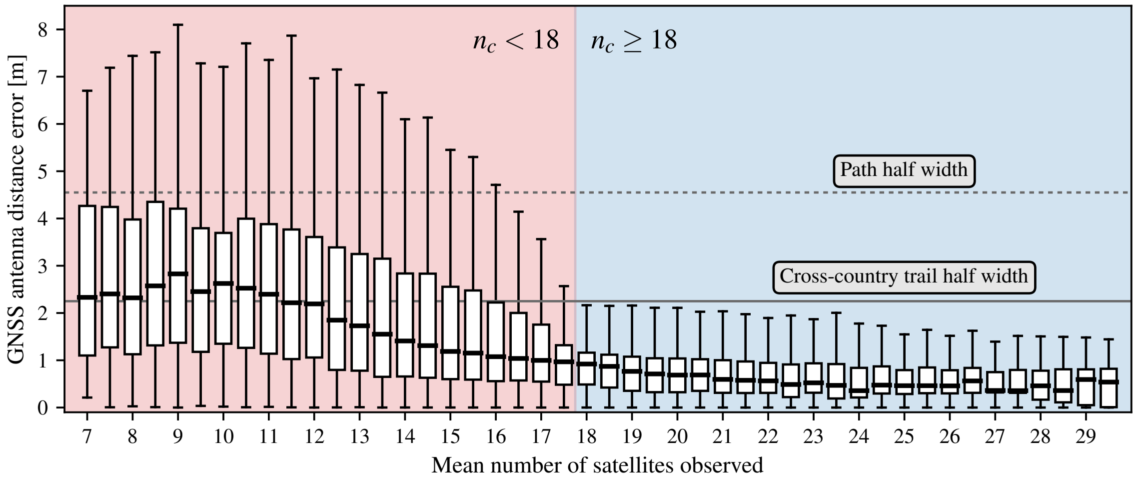


Figure 12. GNSS error with respect to the mean number of observed satellites between the two receivers mounted on the UGV over all runs. The thick black lines represent medians, the boxes represent the first and third quartiles, and the whiskers represent the 10th and 90th percentiles. The expected forest road and forest trail half-widths are 4.55 m and 2.25 m, respectively. The shaded blue area denotes the acceptable number of observed satellites for reliable navigation in wood trails. The area in red is unacceptable for forest trail navigation, based on the critical threshold n_c.

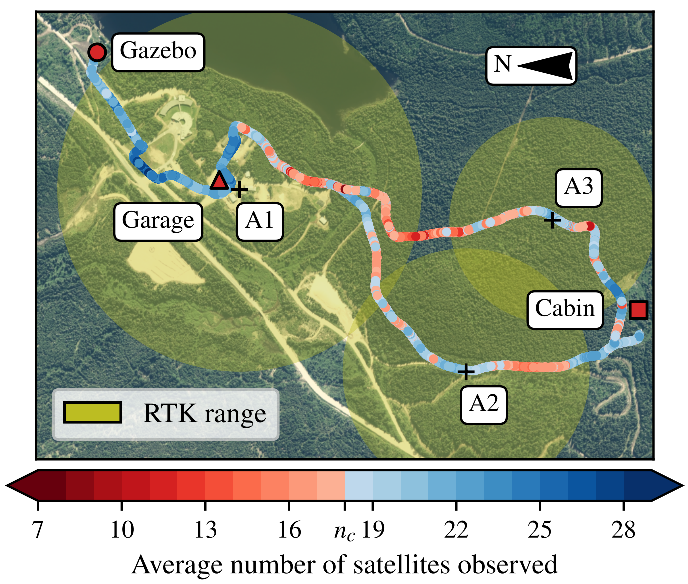


Figure 13. GNSS satellites coverage along the reference trajectories. The blue palette is used for the sections where the GNSS accuracy would be sufficient for the width of the trail. The red sections would not allow pure GNSS navigation. For RTK-GNSS localization, the distinct positions of the reference receiver are the A1, A2, and A3 points. The radio signal range of the reference receivers is shown in olive. Image credit: Forêt ouverte.

mean number of satellites drops below the critical number n_c at several locations, mostly in areas located within dense vegetation. The mean number of observed satellites increases over n_c in the open areas of the trail network. Path C is the only path that could be repeated reliably using GNSS localization. This is due to the fact that its environment is considerably more open compared to A and B, as shown in Figure 10. Additionally, it is necessary to stress that to achieve the presented accuracy in real time, a stable data link between the mobile GNSS receivers and the reference

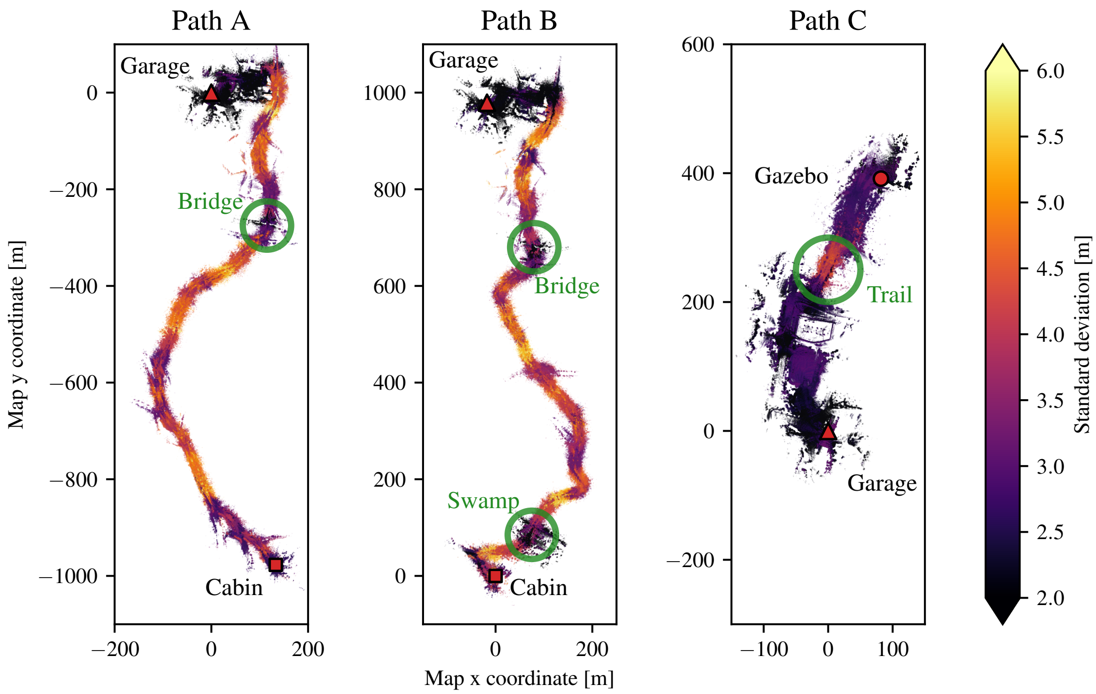


Figure 14. Top view of a set of 500 lidar scans ${}^{\mathcal{G}}\mathcal{P}$ from runs R4, R5, and R1, conducted on paths *A*, *B*, and *C*, respectively. All of the scans are distributed evenly time-wise in their respective runs. The standard deviation of the point-to-plane error e after applying perturbations is used to color each scan. A higher standard deviation is linked to a higher localization uncertainty. Distinctive areas are highlighted by the green circles. The starting and ending POIs for each trajectory are denoted by the red markers.

receiver must be maintained. In practice, this data link is usually implemented by a radio link or by a mobile internet connection, which can be complex to set up in remote locations. In our case, the reference receiver radio signal could not cover the entirety of the A and B paths. Yet, we confirm that the reference receiver positioned on an elevated spot which is not occluded by trees (i.e., A1) provides better range compared to the locations A2 and A3 in the forest.

Furthermore, the ICP algorithm provided localization accurate enough to allow the robot to complete each repeat run without colliding with obstacles. Yet, we observed that the corridor-like nature of the boreal forest trails leads to a high localization uncertainty. To investigate this phenomenon, we extracted 500 lidar scans $\mathcal{P}$ from runs R4, R5, and R1, uniformly distributed time-wise, all of which are shown in Figure 14. These runs are the first repeat runs conducted in the same direction for each reference path. For each lidar scan $\mathcal{P}$, we expressed the map $\mathcal{Q}$ in the corresponding lidar frame $\mathcal{L}$. The initially well-registered scans were perturbed along the robot's longitudinal axis by translations from -6 m to 6 m with increments of 0.05 m. The point-to-plane error e, defined in Equation 2, was then evaluated for each perturbed scan with respect to the reference $\mathcal{Q}$. It should be noted that the ICP algorithm was not executed to convergence for the analysis. The unregistered error was computed for each perturbed point cloud. The standard deviation of the error was used to color each scan.

The areas located within the cross-country skiing trails are related to a considerably higher standard deviation. Indeed, the ICP error function, presented in Equation 2, is much flatter in such areas. A pattern emerges: the standard deviation increases when the robot traverses long and narrow forest corridors, and then decreases when driving through trail intersections. Moreover, the standard deviation in the *Bridge* area, located in paths A and B, decreases notably compared to its surroundings. The river breaks the line of trees in this area and provides additional geometrical constraints, effectively improving the localization accuracy. A similar effect can be observed in the *Swamp* area in path B. A considerable amount of vegetation is perpendicular to the longitudinal direction of the robot, better constraining the registration error function. Analogously, the results from path C demonstrate that the standard deviation is much lower in the wide forest paths. The only place where the localization uncertainty increases again is the narrow trail area marked by the green circle.

5.3. Impact of Snow Precipitation and Accumulation

Another important goal of the deployment is to study the impact of winter weather on autonomous navigation. Paton et al. (2017) already mentioned illumination variation, low feature contrast, and a changing environment as issues for vision-based navigation. Thus, in this work, we focus on the impact of snow precipitation on lidar-based localization. While the system was deployed under various weather conditions, as shown in Figure 9, we did not observe any correlation between the precipitation and the immediate accuracy of localization. However, the snow accumulation led to significant changes in the environment, which affected the ability of our system to localize between teach and repeat runs. More specifically, the WILN system failed to initialize its localization in the reference map during two attempts to start run R9. The third attempt was successful, and the system repeated the entire path without fail subsequently. As documented by Figure 9, run R9 was conducted right after moderate snowfall, and there was thus a significant amount of snow accumulation in the environment.

To analyze the impact of snow accumulation on the reliability of the lidar-based localization, we extracted 500 lidar scans ${}^{\mathcal{G}}\mathcal{P}$ from runs R5, R9, and R14. These scans were already registered with the map and are distributed evenly with respect to time. All three runs were conducted on path B from the *Cabin* POI to the *Garage* POI. Then, we evaluated the overlap percentage for each ${}^{\mathcal{G}}\mathcal{P}$ scan, presented in Figure 15. A scan point was considered as overlapping the map if it was located closer than $0.5\,\mathrm{m}$ to any $\mathcal{Q}$ point.

It is apparent that the scan overlap in run R9 is considerably lower than the scan overlap in R5 and R14, except for the *Garage* area, where R14 has the lowest scan overlap. For the *Cabin*, *Swamp*, and *Trails* areas, the lower overlap ratio can be attributed to snow that accumulated in the environment, which was significantly higher in R9 due to snowfall. Specifically for the *Cabin* area, the large drop in the scan overlap explains the multiple failed initialization attempts mentioned earlier. Furthermore, looking at the *Garage* area, it can be seen that scan overlap decreases as

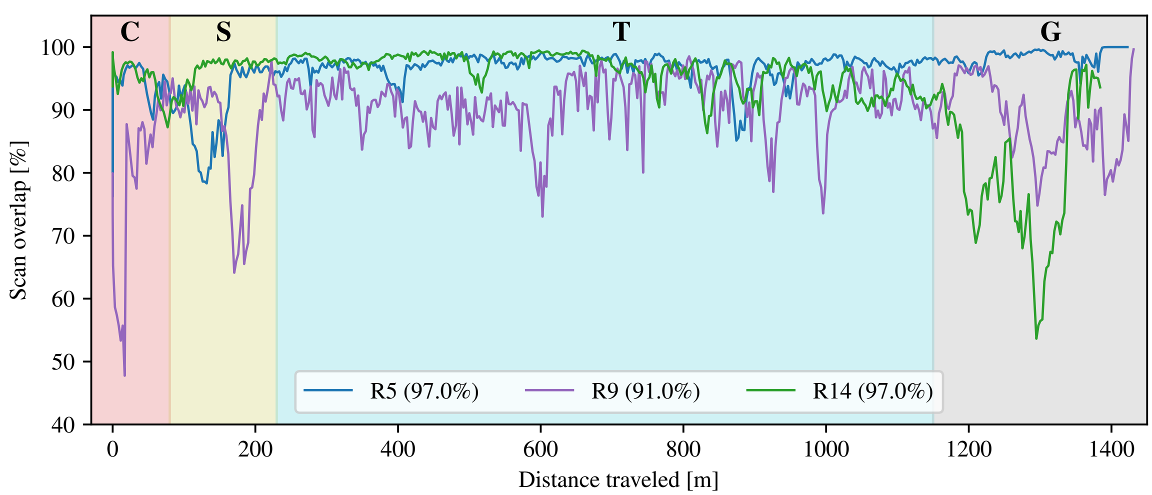


Figure 15. Overlapping percentage of lidar scans ${}^{\mathcal{G}}\mathcal{P}$ on the reference map ${}^{\mathcal{G}}\mathcal{Q}$. The results from runs R5, R9, and R14 are plotted in blue, purple, and green, respectively. The median scan coverage for each run is written between parentheses in the legend. The shaded **C** area represents the *Cabin* area, **S** represents the *Swamp* area, **T** represents the *Trails* area, and **G** represents the *Garage* area.

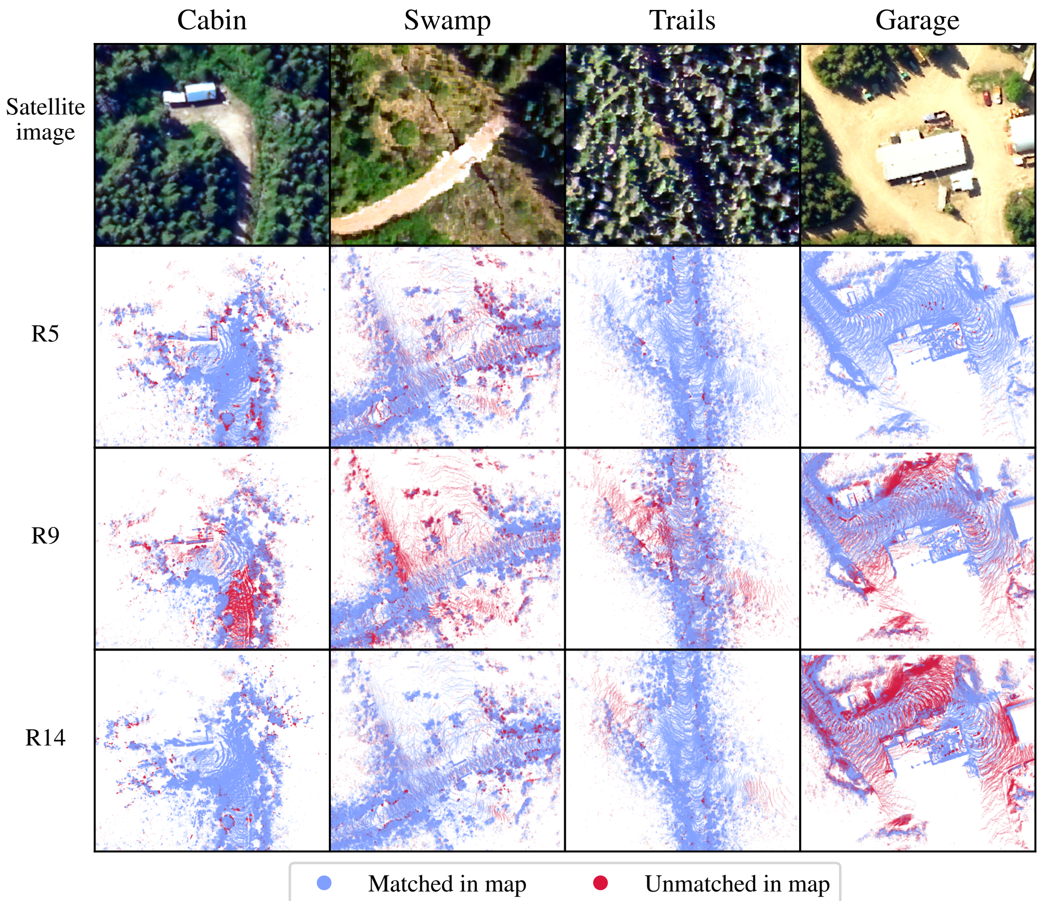


Figure 16. A detail of various parts of path *B*. The satellite images in the top row illustrate the locations. In the registered lidar scans from runs R5, R9, and R14, the points are colored blue if they have a counterpart in the reference map. Contrarily, red points miss their map counterpart and mark a new feature or a change in the environment.

time progresses. Indeed, intermittent snow plowing operations were conducted in this area. Growing snowbanks at the edge of the road led to dynamic changes in the area landscape and consequently to a progressive decrease in lidar scan overlap with the reference map, as shown in Figure 15.

As specific examples, we extracted four problematic areas related to snow accumulation, all of which are shown in Figure 16. A red color mask was applied to the lidar scans $\mathcal{P}$ to highlight the nonoverlapping points. In the *Cabin* area during run R9, it is apparent that snow accumulation led to a variation in the terrain steepness, which affected the ability of the ICP algorithm to initialize its localization within the reference map. The difference in the scan overlap between run R9 and other runs is also noticeable in the *Swamp* and *Trail* areas, although to a lesser extent. Following what is shown in Figure 15, the overlap percentage in the *Garage* area was gradually decreasing as time passed. This phenomenon was related to the aforementioned snow plowing operation taking place in that area, effectively changing the snow landscape. During run R14, there was also a large truck parked near the *Garage*, which contributed to the unmatched points. It should be noted that while scan overlap reaches low values in the *Garage* area, the WILN system did not suffer from localization failure during any run in this area. The location contains multiple buildings, resulting in a higher number of geometrical constraints, allowing the ICP algorithm to localize the robot despite variations in landscape. We assume that the scan overlap would continue to decrease as more time is elapsed between a repeat run and its respective teach run.

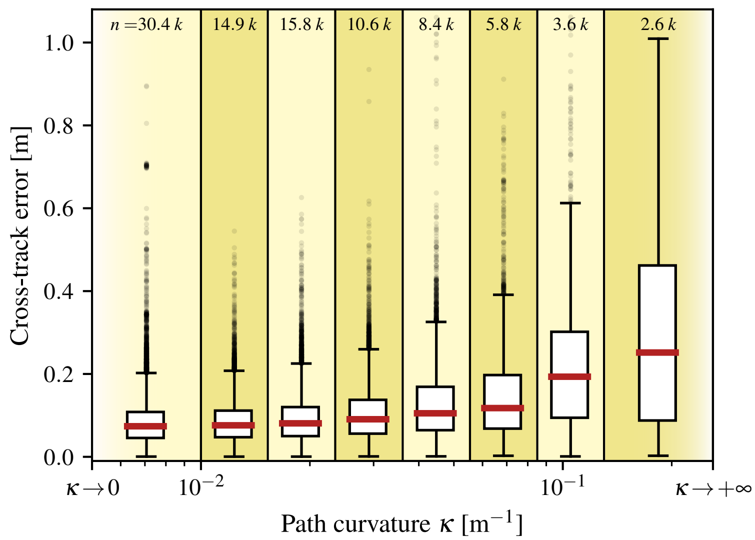


Figure 17. Cross-track error as a function of path curvature for all repeat runs. All errors were split into bins representing different path curvatures. The thick red lines represent medians, the boxes represent the first and third quartiles, respectively, and the whiskers represent the 10th and 90th percentiles. The path curvatures on abscissa are defined in a logarithmic scale. All data related to path curvatures smaller than 0.01 m^{-1} are grouped in the first bin. Similarly, all data related to path curvatures greater than or equal to 0.13 m^{-1} are grouped in the last bin.

5.4. Path-Following Performance

In this section, we have concatenated the cross-track error for all runs; the results can be seen in Figure 17. The median cross-track error across all A, B, and C path runs is 0.083, 0.088, and 0.075 m, respectively. The maximum observed cross-track errors across all runs is 1.19 m for path A, 1.17 m for path B, and 0.92 m for path C. These results suggest that the path-following performance slightly decreases when navigating on a forest trail (i.e., paths A and B), compared to the forest road (i.e., path C). When comparing the results, we did not observe a correlation between weather conditions and cross-track error.

Nonetheless, we did observe significant correlation between path curvature and cross-track error. To analyze this phenomenon, we estimated the path curvature κ of each reference trajectory point. To remove localization noise from the curvature computation, we took ten nearest reference trajectory points into account. We then subsampled all repeat trajectories to keep points at a distance of 0.1 m from each other. For each point of the subsampled trajectories, we found the closest reference trajectory point, computed the cross-track error ϵ_{CT}, and associated it with the reference point's curvature κ. This computation allows us to correlate path curvature with cross-track error, as shown in Figure 17.

We detect that cross-track error increases proportionally to the path curvature. For low curves (i.e., the first six bins), the 90th percentile of the cross-track errors remains below 0.6 m. For straight lines (i.e., the first bin), we attribute the excessive errors over 0.6 m to the initialization error in the beginning of the repeat runs. When navigating in moderate turns (i.e., the last two bins), the 90th percentile of cross-track errors reaches 1.0 m. Due to limitations of the OE controller, the system did not navigate in tight curves (i.e., turning on the spot). The high motion prediction inaccuracy of the kinematic differential-drive model when turning at high angular velocities is the cause for this high increase in cross-track error, as highlighted by Baril et al. (2020). While this error is below the cross-country trail half-width of 2.25 m, which we consider the limit for safe navigation in forest trails, it remains high. It should be noted that the repeat runs were conducted at a conservative nominal velocity v_{nom} of 1.5 m/s. Therefore, there is a significant risk of hitting vegetation for autonomous systems relying on kinematic controllers when navigating in a boreal forest at higher nominal velocities.

6. Challenges and Lessons Learned

In this section, we use the results presented in Section 5 to elaborate how autonomous navigation algorithms should be improved to enable multiyear autonomy. We first explain the challenges related to navigating tight forest trails with dense vegetation in Section 6.1. We also discuss how localization algorithms should adapt to dynamic environments to be deployed all year long in northern environments in Section 6.2. Moreover, we discuss the limitations of kinematic and time-invariant path-following controllers and the risk associated with robot immobilization when navigating in deep snow or through multiple seasons in Section 6.3. Finally, Section 6.5 presents the lessons learned through this field report.

6.1. The *Forest Corridor* Effect

We have highlighted the impact of the dense vegetation on GNSS signal reception in Section 5.2. We observed that GNSS-based localization enables autonomous navigation on forest roads (i.e., path C), but the signal is not reliable when navigating boreal forest trails (i.e., paths A and B). Indeed, the GNSS error can reach upwards to 8 m in forest trails, as can be seen in Figure 12. This effect is due to the signal being absorbed by the dense vegetation in forest trails, an example of which is shown in Figure 1b. Additionally, fusing GNSS measurements with lidar-based localization was investigated by Babin et al. (2019). The authors mentioned that a major issue for real-time RTK-GNSS is that the range of the reference receiver is severely affected by the dense vegetation. In this field report, we have observed the same radio signal range issue, as shown in Figure 13. Enabling real-time RTK-GNSS within boreal forest trails would thus require to set multiple reference antennas throughout the environment. Lastly, it is known that GNSS signal can be jammed, making systems relying on this signal easier to disrupt (Ren et al., 2021). Thus, completely autonomous navigation in a boreal forest requires a localization framework that is resilient to GNSS-denied conditions.

Moreover, we showed in Section 5.2 that featureless corridors are challenging for lidar-based localization algorithms. As can be observed in Figure 14, the uniform nature of boreal forest vegetation leads to low geometrical constraints in the robot's longitudinal direction. Low geometrical constraints lead to high registration uncertainty for the ICP algorithm, as highlighted by Gelfand et al. (2003). This lack of features makes the ICP registration cost function flat in the corresponding direction and sensitive to noise in the lidar measurements. We observe a similar phenomenon in the forest trails surrounded by dense vegetation. Since the lidar scans are not dense enough to distinguish between single branches, the trees resemble large blobs of randomly distributed points in the scans and ultimately in the map. Therefore, point cloud degradation is not limited to the large open areas that were mentioned by Ren et al. (2021). In particular, we have observed that point cloud degradation led to the ICP localization to jumps forward of up to 0.5 m. Such jumps in ICP localization lead to instability and eventually to system failure. Also, they disrupt the function of motion controllers, leading to jerky motion. A system aware of this phenomenon could potentially reduce its velocity in the problematic areas to prevent system failure. A more accurate and adaptive motion prediction model, similar to the one discussed in Section 6.3, could also allow prior error to be reduced and improve localization resiliency in forest corridors. We also argue that future work should investigate multimodal localization approaches. System reliability could be greatly improved by relying on a heterogeneous sensor suite.

6.2. Impact of Snow Accumulation

Our initial hypothesis was that during snowstorms, the precipitation would affect the localization accuracy of the ICP algorithm. Yet, as highlighted in Section 5.3, we found no correlation between

Figure 18. The result of a failed localization initialization, during run R9. The high initial error in robot pose resulted in the Warthog diverging from the reference path and heading towards vegetation. The robot reached a 2.25 m deviation before being manually stopped and driven back to the initialization position.

precipitation intensity and localization accuracy. One should note that while our system was subject to moderate snowfall, no similar work documents the impact of whiteout conditions, which would be caused by heavy snowfall and/or blizzard. We argue that a metric to quantify noise caused by precipitation will be useful to compare various field reports and datasets in the literature. Such a metric would allow to compare various localization approaches under extreme conditions. Rather, we discovered that it was the snow accumulation that severely affected the lidar-based localization. On certain surface types, such as the ground or rooftops, the snow keeps accumulating during the snowstorms. Yet on other surfaces (e.g., trees or running water), only a limited amount accumulates before it falls off or melts. Our data show that lidar-based localization in areas located within deep woods suffer from changes in terrain topology, while areas located near buildings are resilient to this phenomenon. Various examples of the structural change in the environment can be observed in Figure 16. Focusing on the *Cabin* area shown in the first column, it is apparent that terrain steepness changes significantly when comparing run R9 with other similar runs. MacTavish et al. (2018) have mentioned that multiexperience vision-based localization suffered from fast scene appearance change, such as what would be caused by a snowstorm. As mentioned in Section 3.4, the reference map is not updated during the repeat phase for the WILN system. Thus, WILN is also subject to localization failure when attempting to localize in deep woods after snowfall; an example of the result is shown in Figure 18. In this case, the significant structural change in the environment leads to erroneous UGV localization and eventually to crashing with vegetation. However, future work should enable dynamic map maintenance using either multiple maps, similarly to what is proposed by Zhang and Singh (2018) and Maddern et al. (2015) or dynamic maintenance of a single map, similarly to Pomerleau et al. (2014). We argue that using a metric similar to the one presented in Section 5.3 would allow to identify significant structural change in the environment, thus requiring map maintenance. Nevertheless, autonomous systems should adapt their reference maps when observing a mismatch between the current sensor measurements and the reference data to ensure a complete, year-long autonomy.

6.3. Path Following in Snow-Covered Terrain

In Section 5.4, we described the performance of the OE controller when used on the Warthog SSMR. We showed that when navigating at the target velocity v_{nom} of 1.5 m /s, this controller is able to repeat kilometer-scale paths, with an error that remains below the half-width of the forest trails. This result is interesting considering that the OE requires little knowledge of the UGV properties and has low computation time. We have observed that path curvature increases cross-track error,

(a) *Trimble* S7 total stations. **(b)** Fixed *Emlid* GNSS antenna. **(c)** Deep snow navigation. **(d)** Run R7 power outage.

Figure 19. Various lessons learned during this deployment. (a) Total stations that were intended to measure the ground truth localization of the UGV in the *Garage* area. Despite considerable effort to prevent sinkage, those devices shifted significantly, resulting in unusable data. (b) The fixed GNSS antenna to enable RTK-GNSS. This device was placed in open areas to ensure good GNSS signal reception. (c) The Warthog UGV navigating in deep snow. A $1.8\,\mathrm{m}$ tall human operator stands next to the robot for scale. The snow level is estimated at $0.7\,\mathrm{m}$ in this area. The bottom of the robot chassis is resting directly on snow, affecting traction and eventually leading to UGV immobilization. (d) Warthog battery failure during run R7. A generator (in red) was used to recharge the robot batteries and later drive it manually back to the *Garage* POI.

as depicted in Figure 17. This suggests that SSMR motion is difficult to predict when rotating at higher velocities. Thus, we can assume that navigation at higher nominal velocities would potentially lead to the robot crashing into the vegetation. As mentioned in Section 3.5, the original controller implementation presented in Huskić et al. (2017) as well as predictive controllers, such as the one presented in Ostafew et al. (2016), include a parameter reducing vehicle longitudinal velocity when crossing high path curvature. We initially planned to add a similar feature to WILN; however, localization noise in the reference trajectory led to jerking motion in the repeat phase. Reference trajectory smoothing should be investigated in the future to enable speed reduction relative to reference path curvature. Additionally, the OE controller does not react to UGV immobilization. This poses a high risk in off-road navigation, as Figure 19c and Figure 21a demonstrate. A controller based on a more accurate dynamic model that adapts to the various conditions would enable path repeating at higher speeds without failing. A model that can adapt to the differences in UGV behavior across the entire reference trajectory would also improve the path-following resiliency to WILN navigation in steep and soft terrain. Furthermore, WILN is not a completely autonomous system as it does not include obstacle avoidance and planning, as the one described in Krüsi et al. (2015). As future work, it would be interesting to add precipitation-resilient obstacle avoidance. We also argue that obstacle avoidance should include soft ground that prevents the UGV from reaching the next part of the reference path.

6.4. Effect of High-Speed Navigation on Localization

During our preliminary tests, we noticed that the WILN system was not robust to quick motions of the robot. When the robot rotated too quickly, the localization would fail because of motion distortions in lidar scans. To solve this issue, we added a point cloud deskewing step, as described in Section 3.1, to correct motion distortions. This solution was sufficient in our case because the robot was driven at the relatively low speed of $1.5\,\mathrm{m\,s^{-1}}$. However, at higher speeds, motion distortion in lidar scans is more important and odometry accuracy is decreased because of slipping, skidding, and potential robot immobilization. To solve this issue, we introduced an algorithm that computes the uncertainty of each point of a lidar scan to know which points can be trusted (Deschênes et al., 2021). Points which are likely to be more affected by motion distortion have a higher uncertainty than others, and thus are given less influence during the registration process. Also, points which are too uncertain can be completely removed from the map to make sure that it stays crisp. This uncertainty-based algorithm, which can be used in combination with deskewing algorithms, would help to preserve localization and mapping precision when navigating at high speeds with the WILN system.

6.5. Lessons Learned from Field Deployments

Through the multiple deployments conducted in the completion of this field report, we have learned various lessons that generally apply to deploying mobile robots in northern environments. We have observed multiple issues related to robot navigation in deep snow, to localization uncertainty in forest corridor, and to the winter weather affecting robot autonomy. We argue that autonomous systems need to be resilient to these challenges to enable true long-term autonomy. Additionally, we present various logistical difficulties that we encountered in the context of our deployments in the Montmorency forest. The rest of this section will summarize the major lessons learned throughout the realization of this field report.

Any static equipment slowly sinks in snow. In the beginning of our deployment, we installed various static equipment to record position reference. A fixed GNSS reference receiver was installed to enable the RTK-GNSS and three *Trimble* S7 robotic total stations were positioned to measure ground truth robot poses in the *Garage* area, as shown in Figure 19a and Figure 19b. Since the precision of these systems relies on the fact that the equipment remains static, we invested a considerable effort in shoveling and stabilizing snow in the areas where this equipment was installed to prevent the sinkage (see Figure 19a). However, despite our efforts, we observed that the tripods carrying the equipment had sunk in the snow by several centimeters. The movement caused by this sinking effect made the affected ground truth measurements unusable. The outside temperature oscillating around the water freezing point and the snow and rain precipitations shown in Figure 9 led to snow melting, further increasing the tripod sinkage. Thus, significant effort and potentially using fixed structures is required to use such static equipment in winter conditions.

Deep snow significantly affects UGV mobility. As highlighted by Stansbury et al. (2004) and Lever et al. (2013), vehicle immobilization is a significant hazard when navigating on snow-covered terrain. We have attempted to manually drive the Warthog on a deep snow cover (i.e., 0.7 m), as shown in Figure 19c. During these runs, we noticed that robot mobility was significantly affected, especially during turning maneuvers. Also, the UGV chassis would float in sufficiently deep snow, resulting in reduced traction. At this point, turning maneuvers would make the robot sink even deeper, eventually requiring a human intervention to recover the robot. To prevent immobilization, backtracking maneuvers were necessary to compact the snow under the robot to maintain a minimal level of traction. This experience shows that immobilization prevention and recovery are required for mobile robot operation spanning multiple seasons in forest environments.

Winter weather significantly affected the battery capacity of the Warthog UGV. Outside temperatures during this deployment reached a minimum of $-15.5\,°\mathrm{C}$, as shown in Figure 9. In our configuration, the Warthog is equipped with lead-acid batteries, whose capacity is known to be affected by low temperatures. Additionally, the CAMSO ATV T4S tracks that the robot used to drive on snow-covered terrain significantly increased energy consumption compared to using wheels on solid ground. These factors contributed to a UGV battery depletion that occurred during run R7, as captured by the photo in Figure 19d. The robot recovery required a gas generator to charge the batteries, and the operators were required to drive back to the *Garage* POI in the middle of the night. Thus, assuming that the vehicle battery capacity is stable for a period spanning multiple seasons is false and will eventually lead to a system power outage. Autonomous vehicle traversal planning should be conducted conservatively because the system recovery in remote environments is a costly operation. Future planners should include safety margins to take autonomy loss into account for mobile robots deployed in subzero weather.

Resiliency to illumination variance is key to deploying mobile robots in northern environments. Day length is subject to high variation in higher latitudes. To highlight this phenomenon, Figure 20 compares the sun radiation measured in the Montmorency forest during the deployment week and during the summer and winter solstice weeks. The days of the winter solstice week are short and

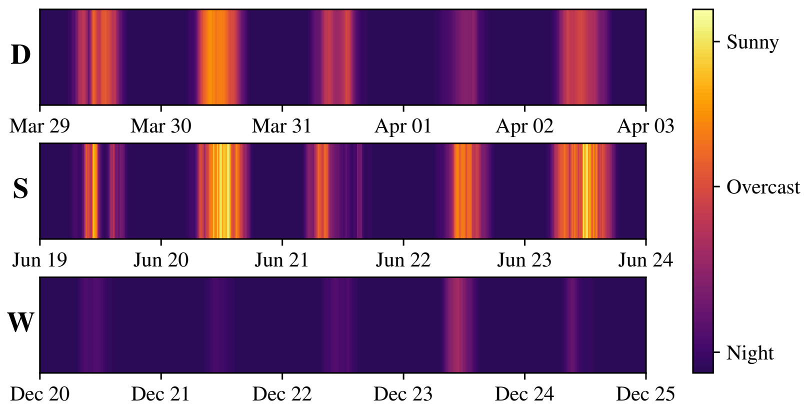


Figure 20. Sun radiation measured at the Montmorency forest for three distinct weeks. On top is the week during which this deployment (**D**) was taking place. In the middle is the week containing the 2021 summer (**S**) solstice. At the bottom is the week containing the 2020 winter (**W**) solstice. The coloring is proportional to the measured sun radiation every hour.

receive minimum daylight. We argue that systems should be resilient to low- or no-illumination conditions to enable year-long autonomy in the northern environments. In remote areas, no artificial light sources such as streetlights or buildings are present to provide illumination for the robot. As discussed by Congram and Barfoot (2021), vision-based localization is affected by low illumination conditions, even when using headlights. However, making robots resilient to navigate in dark environments is key to enable autonomy in the discussed environment. As highlighted by Krüsi et al. (2015), lidar-based localization is resilient to high illumination variations, whereas vision-based approaches may fail (Paton et al., 2017). This is due to the fact that lidars are active sensors that do not rely on an external energy source to produce measurements. Moreover, the lack of daylight has another practical impact. Deploying mobile robots in night conditions is difficult due to low visibility, low temperatures, and operator fatigue, as can be seen in the system recovery snapshot shown in Figure 19d.

Resilience to multiseasonal change and unexpected events. In a later deployment during the fall of 2021, we attempted to repeat all paths by using the maps shown in Figure 10 as reference. We successfully repeated paths A and C by localizing in a map that was recorded 209 days before. However, we had similar initialization problems as the ones discussed in Section 6.2 when attempting path B. After multiple trials, we managed to start a repeat run of B, but the robot became immobilized in a mud pit, as shown in Figure 21a. The repeat phase was canceled at this point, and the robot was manually driven to the *Garage* POI. This experience supports our insight that immobilization prevention, recovery, and dynamic map adjustment are key to enable multiseasonal navigation. Later, when attempting to repeat path C, we noticed that concrete blocks had been added to the path by the forest management (see Figure 21b). The robot was manually driven to avoid collision with these blocks and left to autonomously repeat the rest of the path. Once again, we argue that, to deploy a robotic system in a boreal forest for a period spanning multiple years, autonomous navigation should be adaptive to the significant changes in the environment encountered during those deployments. An interesting avenue for future work would be to use a global planning algorithm, similar to the one presented by Guo and Barfoot (2019) that takes multiseasonal change into account. Such a system could replan paths based on major dynamic events happening on a global path network.

7. Conclusion

This paper is a field report presenting over 18 km of autonomous path repeating in boreal forest trails and paths, including runs under harsh weather and high illumination variations. We have

(a) The Warthog stuck in a mud pit. **(b)** Previously unseen obstacles in path *C*.

Figure 21. Issues related to multiseason path repeating. (a) The Warthog stuck in a mud pit that was undetected during the teach phase of path *B*. Traversable terrain varies between seasons. (b) Obstacles that were moved in path *C* by human operators during summer. The presence of these obstacles required a human to take control of the UGV to avoid a collision.

described the WILN system, designed to enable wintertime lidar-based navigation, and deployed it in the Montmorency boreal forest to evaluate the performance of lidar-based and GNSS localization. We have highlighted the impact of the boreal forest biome and winter conditions on autonomous navigation technologies. Based on the data recorded during these runs, we have shown that forest trails are GNSS-denied environments and that localization uncertainty is high in such trails due to low geometrical constraints in the vehicle's longitudinal direction. Moreover, we have highlighted the impact of snow accumulation and dynamic changes in the environment on lidar-based localization over multiple days in wintertime. We have discussed the requirement to improve the adaptiveness of autonomous navigation to enable multiyear robotic deployments in boreal forests.

While we have shown that our WILN system is able to autonomously repeat paths through a boreal forest in harsh winter conditions, more work remains to enable true long-term autonomy. Lidar-based localization adaptive to changes in the environment is key to allowing mobile robot deployment in boreal forests throughout an entire year. The ability to detect large variations in traction conditions will be key to preventing system failures from vehicle immobilization. Improving interaction between the localization and control systems would also be beneficial to the system adaptability. For example, modulating controller commands based on the localization uncertainty could prevent localization and mapping failures. Furthermore, we have observed through data post-processing that the reference maps built through the teach phase were close to global consistency. In the future, we would like to improve the mapping system to create globally consistent maps and enable localization initialization at any place in the map. This would in turn allow repeating kilometer-scale loops without requiring the UGV to stop for re-initialization at the end of each loop. Another interesting feature would be to enable global localization within the map. Such a feature would enable to solve the robot kidnapping problem and reset the robot localization at any point in the reference path.

Acknowledgments

This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) through the grant CRDPJ 527642-18 SNOW (Self-driving Navigation Optimized for Winter) and FORAC. Support from the Montmorency forest staff was crucial to allow safe UGV deployment in a closed environment. Meteorological data were a courtesy of the Hydrometeorology laboratory at Université Laval. Geographic data processing was provided by the Remote and Wood Sensing Laboratory of the University of New Brunswick.

ORCID

Dominic Baril https://orcid.org/0000-0002-7283-8406
Simon-Pierre Deschênes https://orcid.org/0000-0003-0594-5487
Olivier Gamache https://orcid.org/0000-0003-3278-5650
Maxime Vaidis https://orcid.org/0000-0002-1749-7207
Damien LaRocque https://orcid.org/0000-0003-3256-046X
Johann Laconte https://orcid.org/0000-0002-2274-9835
Vladimír Kubelka https://orcid.org/0000-0001-8393-9969
Philippe Giguère https://orcid.org/0000-0002-7520-8290
François Pomerleau https://orcid.org/0000-0003-1288-2744

References

Apostolopoulos, D. S., Wagner, M. D., Shamah, B. N., Pedersen, L., Shillcutt, K., and Whittaker, W. L. (2000). Technology and field demonstration of robotic search for antarctic meteorites. *International Journal of Robotics Research*, 19(11):1015–1032.

Arcone, S. A., Lever, J. H., Ray, L. E., Walker, B. S., Hamilton, G., and Kaluzienski, L. (2016). Ground-penetrating radar profiles of the McMurdo Shear Zone, Antarctica, acquired with an unmanned rover: Interpretation of crevasses, fractures, and folds within firn and marine ice. *Geophysics*, 81(1):WA21–WA34.

Arya, S. and Mount, D. M. (1993). Approximate nearest neighbor queries in fixed dimensions. *Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 271–280.

Babin, P., Dandurand, P., Kubelka, V., Giguère, P., and Pomerleau, F. (2019). Large-scale 3D mapping of subarctic forests. In *Proceedings of the Conference on Field and Service Robotics (FSR). Springer Tracts in Advanced Robotics*.

Baril, D., Grondin, V., Deschenes, S., Laconte, J., Vaidis, M., Kubelka, V., Gallant, A., Giguere, P., and Pomerleau, F. (2020). Evaluation of skid-steering kinematic models for subarctic environments. In *2020 17th Conference on Computer and Robot Vision (CRV)*, pages 198–205.

Bosse, M. and Zlot, R. (2009). Continuous 3D scan-matching with a spinning 2D laser. In *2009 IEEE International Conference on Robotics and Automation*, pages 4312–4319. IEEE.

Charron, N., Phillips, S., and Waslander, S. L. (2018). De-noising of lidar point clouds corrupted by snowfall. In *2018 15th Conference on Computer and Robot Vision (CRV)*, pages 254–261. IEEE.

Churchill, W. and Newman, P. (2013). Experience-based navigation for long-term localisation. *International Journal of Robotics Research*, 32(14):1645–1661.

Clement, L., Kelly, J., and Barfoot, T. D. (2017). Robust monocular visual teach and repeat aided by local ground planarity and color-constant imagery. *Journal of Field Robotics*, 34(1):74–97.

Clement, L., Gridseth, M., Tomasi, J., and Kelly, J. (2020). Learning matchable image transformations for long-term metric visual localization. *IEEE Robotics and Automation Letters*, 5(2):1492–1499.

Congram, B. and Barfoot, T. D. (2021). Relatively lazy: Indoor-outdoor navigation using vision and GNSS. In *2021 18th Conference on Robots and Vision (CRV)*, pages 25–32. IEEE.

Deschênes, S.-P., Baril, D., Kubelka, V., Giguère, P., and Pomerleau, F. (2021). Lidar scan registration robust to extreme motions. In *2021 18th Conference on Robots and Vision (CRV)*, pages 17–24.

Duan, Y., Yang, C., Chen, H., Yan, W., and Li, H. (2021). Low-complexity point cloud denoising for LiDAR by PCA-based dimension reduction. *Optics Communications*, 482(September 2020):126567.

Furgale, P. and Barfoot, T. D. (2010). Visual teach and repeat for long-range rover autonomy. *Journal of Field Robotics*, 27(5):534–560.

Gelfand, N., Ikemoto, L., Rusinkiewicz, S., and Levoy, M. (2003). Geometrically stable sampling for the ICP algorithm. In *Fourth International Conference on 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings*, volume 2003-January, pages 260–267. IEEE.

Gifford, C. M., Akers, E. L., Stansbury, R. S., and Agah, A. (2009). Mobile robots for polar remote sensing. In *The Path to Autonomous Robots*, pages 1–22. Springer US, Boston, MA.

Guo, H. and Barfoot, T. D. (2019). The robust Canadian traveler problem applied to robot routing. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 5523–5529.

Heinzler, R., Piewak, F., Schindler, P., and Stork, W. (2020). CNN-based lidar point cloud de-noising in adverse weather. *IEEE Robotics and Automation Letters*, 5(2):2514–2521.

Huskić, G., Buck, S., and Zell, A. (2017). A simple and efficient path following algorithm for wheeled mobile robots. In *Advances in Intelligent Systems and Computing*, volume 531, pages 375–387.

Huskić, G., Buck, S., and Zell, A. (2019). GeRoNa: Generic Robot Navigation. *Journal of Intelligent & Robotic Systems*, 95(2):419–442.

Jagbrant, G., Underwood, J. P., Nieto, J., and Sukkarieh, S. (2015). LiDAR based tree and platform localisation in almond orchards. In Mejias, L., Corke, P., and Roberts, J., editors, *Field and Service Robotics: Results of the 9th International Conference*, pages 469–483, Springer International.

Jenke, P., Wand, M., Bokeloh, M., Schilling, A., and Straßer, W. (2006). Bayesian point cloud reconstruction. *Computer Graphics Forum*, 25(3):379–388.

Kruijff, G. J. M., Janíček, M., Keshavdas, S., Larochelle, B., Zender, H., Smets, N. J. J. M., Mioch, T., Neerincx, M. A., Diggelen, J. V., Colas, F., Liu, M., Pomerleau, F., Siegwart, R., Hlaváč, V., Svoboda, T., Petříček, T., Reinstein, M., Zimmermann, K., Pirri, F., Gianni, M., Papadakis, P., Sinha, A., Balmer, P., Tomatis, N., Worst, R., Linder, T., Surmann, H., Tretyakov, V., Corrao, S., Pratzler-Wanczura, S., and Sulk, M. (2014). Experience in system design for human-robot teaming in urban search and rescue. In Yoshida, K., and Tadokoro, S., editors, *Field and Service Robotics: Results of the 8th International Conference*, pages 111–125. Springer, Berlin.

Krüsi, P., Bücheler, B., Pomerleau, F., Schwesinger, U., Siegwart, R., and Furgale, P. (2015). Lighting-invariant adaptive route following using iterative closest point matching. *Journal of Field Robotics*, 32(4):534–564.

Kubelka, V., Dandurand, P., Babin, P., Giguère, P., and Pomerleau, F. (2020). Radio propagation models for differential GNSS based on dense point clouds. *Journal of Field Robotics*, 37(8):1347–1362.

Lever, J. H., Delaney, A. J., Ray, L. E., Trautmann, E., Barna, L. A., and Burzynski, A. M. (2013). Autonomous GPR surveys using the polar rover Yeti. *Journal of Field Robotics*, 30(2):194–215.

MacTavish, K., Paton, M., and Barfoot, T. D. (2017). Visual triage: A bag-of-words experience selector for long-term visual route following. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2065–2072. IEEE.

MacTavish, K., Paton, M., and Barfoot, T. D. (2018). Selective memory: Recalling relevant experience for long-term visual localization. *Journal of Field Robotics*, 35(8):1265–1292.

Maddern, W., Pascoe, G., and Newman, P. (2015). Leveraging experience for large-scale lidar localisation in changing cities. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, volume 2015-June, pages 1684–1691. IEEE.

Madgwick, S. O. H., Harrison, A. J. L., and Vaidyanathan, R. (2011). Estimation of IMU and MARG orientation using a gradient descent algorithm. In *2011 IEEE International Conference on Rehabilitation Robotics*, pages 1–7. IEEE.

Marshall, J., Barfoot, T., and Larsson, J. (2008). Autonomous underground tramming for center-articulated vehicles. *Journal of Field Robotics*, 25(6–7):400–421.

Marshall, J., Bonchis, A., Nebot, E., and Scheding, S. (2016). Robotics in mining, In Siciliano, B., and Khatib, O., editors, *Springer Handbook of Robotics*, pages 1549–1576, Springer International.

Mazuran, M., Sprunk, C., Burgard, W., and Tipaldi, G. D. (2015). LexTOR: Lexicographic teach optimize and repeat based on user preferences. *Proceedings - IEEE International Conference on Robotics and Automation*, 2015-June(6):2780–2786.

McManus, C., Furgale, P., Stenning, B., and Barfoot, T. D. (2013). Lighting-invariant visual teach and repeat using appearance-based lidar. *Journal of Field Robotics*, 30(2):254–287.

Ministère des Forêts, de la Faune et des Parcs (2017). Ski de fond; guide des principes d'aménagement, d'entretient et de signalisation des sentiers. *Collection de BAnQ, Québec, Canada*.

Ministère des Forêts, de la Faune et des Parcs (2021). Guide d'application du règlement sur l'aménagement durable des forêts du domaine de l'état. *Gouvernement du Québec*. Retrieved from https://mffp.gouv.qc.ca/RADF/guide/annexes/annexe-4/.

Mondoloni, S., Swierstra, S., and Paglione, M. (2005). Assessing trajectory prediction performance & metrics definition. In *24th Digital Avionics Systems Conference*, volume 1. IEEE.

Moorehead, S., Simmons, R., Apostolopoulos, D., and Whittaker, W. L. (1999). Autonomous navigation field results of a planetary analog robot in antarctica.

Nieto, J., Guivant, J., Nebot, E., and Thrun, S. (2003). Real time data association for FastSLAM. In *2003 IEEE International Conference on Robotics and Automation*, volume 1, pages 412–418. IEEE.

Oliveira, L. F. P., Moreira, A. P., and Silva, M. F. (2021). Advances in forest robotics: A state-of-the-art survey. *Robotics*, 10(2).

Ostafew, C. J., Schoellig, A. P., and Barfoot, T. D. (2016). Robust constrained learning-based NMPC enabling reliable mobile robot path tracking. *International Journal of Robotics Research*, 35(13):1547–1563.

Paton, M., MacTavish, K., Ostafew, C. J., and Barfoot, T. D. (2015). It's not easy seeing green: Lighting-resistant stereo visual teach & repeat using color-constant images. *Proceedings - IEEE International Conference on Robotics and Automation*, 2015-June(June):1519–1526.

Paton, M., MacTavish, K., Warren, M., and Barfoot, T. D. (2016). Bridging the appearance gap: Multi-experience localization for long-term visual teach and repeat. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 2016-November, pages 1918–1925. IEEE.

Paton, M., Pomerleau, F., MacTavish, K., Ostafew, C. J., and Barfoot, T. D. (2017). Expanding the limits of vision-based localization for long-term route-following autonomy. *Journal of Field Robotics*, 34(1):98–122.

Paton, M., MacTavish, K., Berczi, L.-P., van Es, S. K., and Barfoot, T. D. (2018). I can see for miles and miles: An extended field test of visual teach and repeat 2.0. In Hutter, M. and Siegwart, R., editors, *Field and Service Robotics*, pages 415–431, Springer International.

Pomerleau, F., Magnenat, S., Colas, F., Ming Liu, and Siegwart, R. (2011). Tracking a depth camera: Parameter exploration for fast ICP. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3824–3829. IEEE.

Pomerleau, F., Colas, F., Siegwart, R., and Magnenat, S. (2013). Comparing ICP variants on real-world data sets. *Autonomous Robots*, 34(3):133–148.

Pomerleau, F., Krusi, P., Colas, F., Furgale, P., and Siegwart, R. (2014). Long-term 3D map maintenance in dynamic environments. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, number 1, pages 3712–3719. IEEE.

Pomerleau, F., Colas, F., and Siegwart, R. (2015). *A Review of Point Cloud Registration Algorithms for Mobile Robotics*. Now Publishers.

Ray, L., Jordan, M., Arcone, S. A., Kaluzienski, L. M., Walker, B., Koons, P. O., Lever, J., and Hamilton, G. (2020). Velocity field in the McMurdo shear zone from annual ground penetrating radar imaging and crevasse matching. *Cold Regions Science and Technology*, 173(February):103023.

Ren, R., Fu, H., Xue, H., Li, X., Hu, X., and Wu, M. (2021). LiDAR-based robust localization for field autonomous vehicles in off-road environments. *Journal of Field Robotics*, 38(8):1059–1077.

Russell, E. and Ritchie, J. C. (1988). Postglacial vegetation of Canada. *Bulletin of the Torrey Botanical Club*, 115:231.

Schall, O., Belyaev, A., and Seidel, H. P. (2005). Robust filtering of noisy scattered point data. *Point-Based Graphics, 2005 - Eurographics/IEEE VGTC Symposium Proceedings*, pages 71–77.

Schall, O., Belyaev, A., and Seidel, H. P. (2008). Adaptive feature-preserving non-local denoising of static and time-varying range data. *CAD Computer Aided Design*, 40(6):701–707.

Simon, P. (2015). Military robotics: Latest trends and spatial grasp solutions. *International Journal of Advanced Research in Artificial Intelligence*, 4.

Sprunk, C., Tipaldi, G. D., Cherubini, A., and Burgard, W. (2013). Lidar-based teach-and-repeat of mobile robot trajectories. *IEEE International Conference on Intelligent Robots and Systems*, pages 3144–3149.

Stansbury, R. S., Akers, E. L., Harmon, H. P., and Agah, A. (2004). Survivability, mobility, and functionality of a rover for radars in polar regions. *International Journal of Control, Automation and Systems*, 2(3):343–353.

Takasu, T. and Yasuda, A. (2009). Development of the low-cost RTK-GPS receiver with an open source program package RTKLIB. *International Symposium on GPS/GNSS*, (October):4–6.

Vaidis, M., Giguere, P., Pomerleau, F., and Kubelka, V. (2021). Accurate outdoor ground truth based on total stations. *Proceedings - 2021 18th Conference on Robots and Vision, CRV 2021*, pages 1–8.

Van Brummelen, J., O'Brien, M., Gruyer, D., and Najjaran, H. (2018). Autonomous vehicle perception: The technology of today and tomorrow. *Transportation Research Part C: Emerging Technologies*, 89(July 2017):384–406.

Williams, S. and Howard, A. M. (2009). Developing monocular visual pose estimation for arctic environments. *Journal of Field Robotics*, 27(2):145–157.

World Meteorological Organization (2019). *Manual on Codes.* World Meteorological Organization.
Zhang, J. and Singh, S. (2018). Laser-visual-inertial odometry and mapping with high robustness and low drift. *Journal of Field Robotics*, 35(8):1242–1264.

How to cite this article: Baril, D., Deschênes, S.-P., Gamache, O., Vaidis, M., LaRocque, D., Laconte, J., Kubelka, V., Giguère, P., & Pomerleau, F. (2022). Kilometer-scale autonomous navigation in subarctic forests: challenges and lessons learned. *Field Robotics, 2*, 1628–1660.