

High-Speed Localization for Autonomous Race Cars in GNSS-Denied Scenarios

Haoru Xue

Robotics Institute

Carnegie Mellon University

Pittsburgh, USA

haorux@andrew.cmu.edu

Abstract—High-speed autonomous racing requires accurate, frequent, real-time localization to robustly plan and control the vehicle. We propose an offline mapping and online localization pipeline for high-speed autonomous racing in GNSS-denied scenarios. We derive a variant to the Extended Kalman Filter to handle asynchronous observation inputs, and use the output state estimations to compensate for LiDAR distortions. This pipeline reduces the longitudinal localization error by up to 40% compared to the case when no state estimation is available to compensate for the distortions.

I. INTRODUCTION

Autonomous racing is a research area that has gained new ground in recent years. The Indy Autonomous Challenge (IAC) [1] empowers multi-agent autonomous racing near 200 MPH. These vehicles are equipped with RTK GNSS, cameras, LiDARs, and radars for perception and localization needs.

Localization is an important subject for vehicles moving at such speeds. Autonomous racing requires high-accuracy odometry estimation that can work reliably given noisy inputs under agile motion and vibration and still produce accurate results. The lateral velocity estimation, for example, gives critical information to model predictive controllers (MPC) to help stabilize the vehicle under high side-slip angle condition, even though it typically does not exceed 4-6 degrees on a formula race car. In addition to odometry, autonomous racing also requires a real-time map-based pose estimation system that couples with the odometry estimation and compensates for its drifting. Since the race track environment is static and known to the robots at run-time, it is often unnecessary to perform online simultaneous localization and mapping (SLAM). This can be done offline, and the robot can opt for localization-only on an existing map online. The normal distribution transform (NDT) [2] is an efficient example of such a method.

The challenges of using LiDAR localization for high-speed vehicles arise mainly from LiDAR distortion. Since the LiDAR sensor takes time to complete a scan of the environment, the vehicle would have moved significantly during the time the scan was taken. To put this into numbers, suppose that a race car traveling at 90 m/s takes a LiDAR scan within 30 ms, and then the car would have already moved 2.7 m during that time. This causes a significant downgrade in the ability of the localization system to localize on a pre-defined map, as visualized in figure 2.

In this project, we develop a high-speed localization system that is based solely on the map and does not rely on GNSS. We aim to derive a pipeline of mapping and localization methods for an IAC vehicle, producing high-quality localization data even in GNSS-denied regions. Our main contribution lies within state estimation using a high-fidelity vehicle dynamics model, whose output directly affects the quality of odometry estimation and LiDAR distortion compensation.¹



Fig. 1. The race car of Carnegie Mellon University, University of California Berkeley, and University of California San Diego in the Indy Autonomous Challenge. LiDARs and cameras are placed in the cockpit area.

II. BACKGROUND

Since high-speed autonomous racing is a relatively new research field, the history of existing work is shallow and comprises mostly implementation and modification of conventional approaches in robotics. [10] uses such an approach where a classic odometry model [12] is used for motion update, and Adaptive Monte Carlo Localization (AMCL) is used for localization. However, their work uses 2D SLAM and targets a specific planar race track, which is insufficient in recent autonomous racing challenges on banked oval race tracks. [8] proposes a combined LiDAR-camera localization. In their work, the longitudinal and lateral localizations along the track are treated independently. OpenVSLAM [11] is used for longitudinal localization, since for a race track, especially on a long straight, there is very little feature for a LiDAR to carry

¹The project code is available at <https://github.com/HaoruXue/cmu-16833-project>.

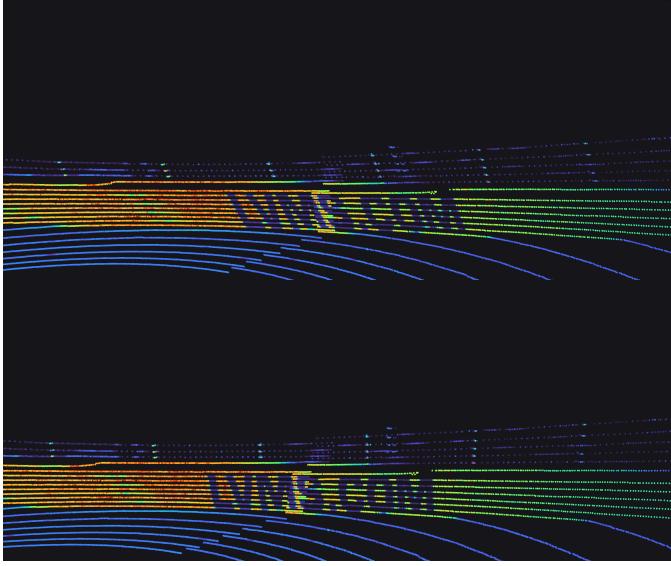


Fig. 2. Top: Visualization of the LiDAR scan distortion at 100 MPH, extracted from RACECAR [5]. The fence in the background is slanted due to longitudinal motion, creating a challenge for LiDAR localization. Bottom: The same LiDAR scan after distortion correction with our state estimation pipeline output.

out longitudinal localization, because the long stretch of walls around the track makes it quite difficult for distinguishable matches. OpenVSLAM proves to be very useful in this case, producing an accurate 3D track map. Lateral localization is solved with a simple wall-finding algorithm to compute and filter the vehicle’s distance from the race track’s outside wall. However, this decoupled approach takes advantage of the very specific race track they experimented with, which has walls parallel to the race track. This is an assumption that would not hold in many other race tracks. The LiDAR information is also not integrated in any sense to the SLAM process, but is simply used in the wall boundary detection. The system also has also only been evaluated in simulation, and no real-world experiment has since been followed.

The recently released RACECAR dataset [5] provides new open access to autonomous racing data. It includes camera and LiDAR data from the IAC race cars for localization, perception, and planning benchmarks. Multiple scenarios are included in the dataset, such as single-car slow-speed runs and multi-car high-speed runs, visualized in figure 3. The ground truth trajectory is also included, since the tracks in the dataset have good RTK GNSS signal reception. The authors also recognize that map-based localization in high-speed autonomous racing has been largely unsuccessful in the real world due to the various challenges we have discussed above.

It is worth noting that most current full-size autonomous racing applications do not carry any such localization method at all, but instead resort to high-accuracy real-time kinematics (RTK) global navigation satellite system (GNSS), providing centimeter-accurate pose and velocity information from a

combination of GNSS satellite systems (GPS, GLONASS, BeiDou, etc.). A Kalman filter such as [6] could be applied on top for the full localization solution. The combination of GNSS and map-based localization has also been investigated in [13] and shows promising results. However, as competition expands to race tracks with overhead occlusions (overpasses, trees, etc.) such as Monza, Italy, high-quality localization in GNSS-denied scenarios is becoming an important research topic in the field.

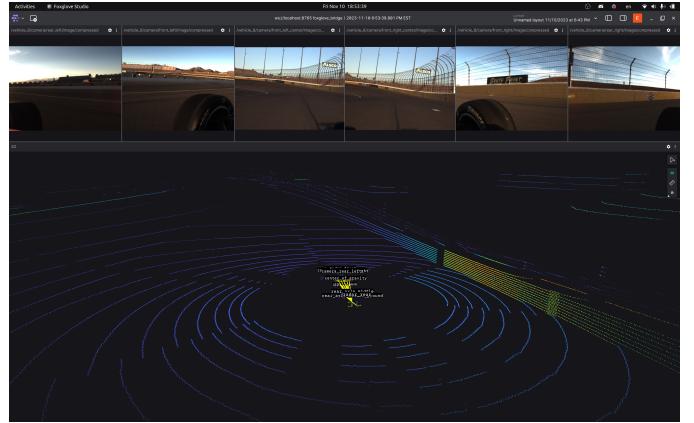


Fig. 3. Extracted LiDAR and camera data from the RACECAR dataset. This is a low-speed solo run contributed by Massachusetts Institute of Technology, University of Pittsburgh, and Rochester Institute of Technology.

III. METHOD

In this section, we will describe our offline mapping and online localization pipeline in detail. More attention is given to the latter, since it is our main contribution.

The first stage of the project is to produce a map of the race track based on various LiDAR SLAM methods [14], using LiDAR data from the RACECAR dataset. This will serve as our map for online localization. We purposely throw away the ground-truth trajectory in the mapping process because we want the mapping pipeline to work on GNSS-denied race tracks as well. We will select a low-speed solo scenario from the RACECAR dataset, in which the motion distortion from the LiDAR can be largely ignored to produce a higher quality map. For a readily available SLAM solution, we use LIO-SAM[9], which is a LiDAR odometry and mapping (LOAM)[14] method integrated with IMU and odometry updates. Motion prior is directly formulated into the optimization problem, which is different from previous approaches in LOAM where it is simply used as an initialization. It also enables efficient handling of large scenes through its selective keyframe approach and marginalization of dormant scans, which is critical for mapping a race track environment that could span several miles.

The second stage is to design the various components of the online localization pipeline, which is visualized on the right side in figure 4. We first use inertial measurements to estimate odometry with a high-fidelity vehicle dynamics model. Then we use its output to correct LiDAR point cloud

distortion, and initialize the NDT matching process. The combined localization result outputs both the global pose and the velocity of the race car. The details of all components are described in the following subsections.

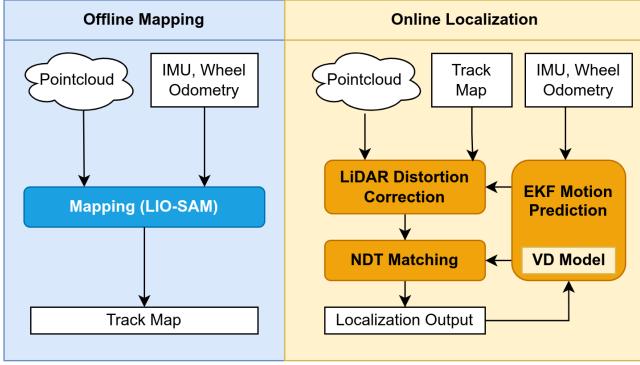


Fig. 4. Illustration of the offline mapping and online localization pipeline. Our contribution mainly lies within motion prediction and distortion correction step using highly optimized dynamics and observation functions for high-speed race car applications to improve the quality of the scans and NDT initializations.

A. Frenet Frame Representation

Frenet frame is commonly used in planning and controls for autonomous driving. Instead of expressing the vehicle pose in a local Cartesian coordinate, we do so w.r.t. a Frenet frame along the race track center-line. We use s to denote the longitudinal progress of the vehicle along the track, e_y to denote the lateral distance from the center line and e_ψ to denote the orientation deviation. This is visualized in figure 5.

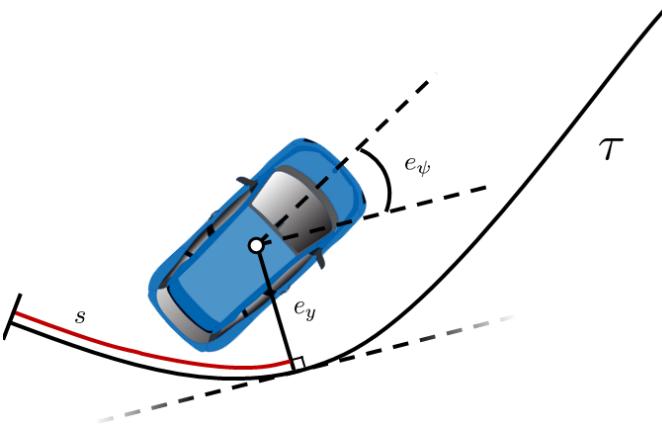


Fig. 5. Illustration of the race track Frenet frame.

There are two benefits with performing localization in the Frenet frame. First, it enables a convenient and intuitive representation of a vehicle's position on the race track. Second, it enables better linearization of the dynamics function, because e_ψ , the angular deviation, is always a small angle, since the vehicle would always travel roughly in the direction of the center line.

We point out that it is easy to transform from a Cartesian coordinate to Frenet coordinate, given a dynamics function that was originally in the former [3]:

$$\dot{e}_\psi = \omega_z - k \frac{v_x \cos e_\psi - v_y \sin e_\psi}{1 - e_y k} \quad (1a)$$

$$\dot{s} = \frac{v_x \cos e_\psi - v_y \sin e_\psi}{1 - e_y k} \quad (1b)$$

$$\dot{e}_y = v_x \sin e_\psi + v_y \cos e_\psi \quad (1c)$$

B. Extended Kalman Filter with Partial Update

Our motion estimation module is based on a variant of the Extended Kalman Filter [4] to specifically address the asynchronous and partial nature of inertial sensor measurements. This is because the linear velocity measurement, which mostly depend on wheel odometry, does not synchronize with linear acceleration and angular velocity measurements, which come from the IMU.

We consider the following 6 degree-of-freedom state space

$$\mathbf{x} = [s \ e_y \ e_\psi \ v_x \ v_y \ \omega_z] \quad (2)$$

where s , e_y , and e_ψ represents the pose of the vehicle in the Frenet coordinate frame of the race track surface, and v_x , v_y , and ω_z represents the body frame velocity. Note that we have omitted the vertical position, roll, and pitch to simplify the system. We argue that this is reasonable because the vehicle motion is constrained to the track surface, whose roll and pitch angle can be looked up. Additionally, the vehicle dynamics in the vertical direction is much slower than its lateral and longitudinal motion in high speed.

We use \mathbf{M} to denote a $m \times 6$ selection matrix that selects m states from the six state variables. For an observation function $h_i(\mathbf{Mx})$ that partially observes some state variables $\{x_1 \dots x_m\} \in \mathbf{x}$, we consider the following EKF partial filter update equation

$$\text{Motion Prediction} \quad \hat{\mathbf{x}}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) \quad (3a)$$

$$\text{Covariance Prediction} \quad \hat{\mathbf{P}}_k = \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^T + \mathbf{Q}_{k-1} \quad (3b)$$

$$\text{Partial Observation} \quad \mathbf{y} = \mathbf{z} - h(\mathbf{M}\hat{\mathbf{x}}_k) \quad (3c)$$

$$\text{Innovation Covariance} \quad \mathbf{S} = \mathbf{H}\hat{\mathbf{P}}_k\mathbf{H}^T + \mathbf{R} \quad (3d)$$

$$\text{Partial Kalman Gain Update} \quad \mathbf{K}\mathbf{M}^T = \hat{\mathbf{P}}_k\mathbf{H}^T\mathbf{S}^{-1} \quad (3e)$$

$$\text{State Update} \quad \mathbf{x}_k = \hat{\mathbf{x}}_k + \mathbf{K}\mathbf{M}^T\mathbf{y} \quad (3f)$$

$$\text{Partial Covariance Update} \quad \mathbf{P}_k = (\mathbf{I} - \mathbf{K}\mathbf{M}^T\mathbf{H})\hat{\mathbf{P}}_k \quad (3g)$$

where $f(\mathbf{x}, \mathbf{u})$ denotes the dynamics model; \mathbf{P} denotes covariance; \mathbf{Q} denotes process noise; \mathbf{R} denotes measurement covariance; \mathbf{K} denotes Kalman gain; \mathbf{F} denotes the Jacobian of the dynamics w.r.t. the state; and \mathbf{H} denotes the Jacobian of the observation w.r.t. the partial state variables. Note that equation (3) carries out a partial filter update with respect to the specific states it can observe. The covariance of the other states in this update cycle will be slightly inflated since

no measurement is available, but they would be subsequently handled by other measurements following the same procedure as in equation (3). We also note that the frequency of the EKF output is decoupled from the frequency of the incoming observations and runs on a constant timer. We simply perform a pure prediction from the last observation update and output the prediction. When another observation comes in, we roll back these predictions before carrying out a filter update.

C. Vehicle Dynamics Model

We will now describe the dynamics model used in equation (3a). We employ a single-track vehicle dynamics model, shown in figure 6, that is specifically designed to consider race car characteristics that have significantly more impact on vehicle localization than on passenger vehicles. This includes aerodynamic downforce and non-linear tire model.

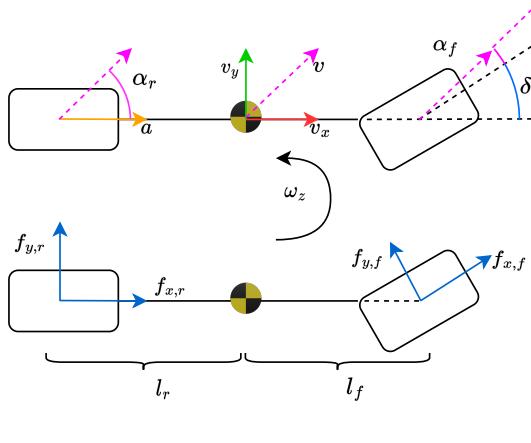


Fig. 6. Illustration of the single track dynamics model. The top figure shows the state and control variables. The bottom figure shows the tire force visualizations.

We start by describing the fundamental tire forces, considering a rear-wheel-drive, four-wheel-brake race car. The calculated force is per wheel and is decomposed into longitudinal and lateral tire forces. The front and rear longitudinal tire forces are

$$F_{x,f} = \frac{1}{2}(k_b m a_b - f_r m g \frac{l_r}{l}) \quad (4a)$$

$$F_{x,r} = \frac{1}{2}(m a_d + (1 - k_b) m a_b - f_r m g \frac{l_f}{l}) \quad (4b)$$

where k_b denotes the front brake bias (0-100%); f_r denotes rolling resistance; m denotes the vehicle mass; l denotes wheelbase; l_f, l_r denotes the distance between center-of-gravity and the two axles; a_d, a_b are derived from control variable a to enable the modeling of different longitudinal dynamics under driving and braking with an activation function:

$$a_d = a(0.5 \tanh(a) + 0.5) \quad (5a)$$

$$a_b = a(0.5 \tanh(-a) + 0.5) \quad (5b)$$

Then we describe normal tire loads considering longitudinal load transfer when accelerating and braking.

$$F_{z,f} = \frac{1}{2}(m g \frac{l_r}{l} - \frac{h_{cog}}{l} m a_x + f_D c_{l,f}) \quad (6a)$$

$$F_{z,r} = \frac{1}{2}(m g \frac{l_r}{l} + \frac{h_{cog}}{l} m a_x + f_D c_{l,r}) \quad (6b)$$

$$\text{where } f_D = \frac{1}{2} C_D A v_x^2 \quad (6c)$$

$$a_x = a - \frac{f_D}{m} - f_r g \quad (6d)$$

h_{cog} denotes the center-of-gravity height; $c_{l,f}, c_{l,r}$ denotes the lift coefficients at the axles; f_D denotes the air resistance; C_D denotes the drag coefficient; A denotes the frontal area of the car; a_x denotes the overall longitudinal acceleration.

We can now calculate the lateral tire forces F_y using normal tire forces F_z using a pure lateral slip Pacejka tire model [7].

$$\alpha_f = \delta - \arctan((l_f \omega_z + v_y)/v_x) \quad (7a)$$

$$\alpha_r = \arctan((l_r \omega_z - v_y)/v_x) \quad (7b)$$

$$F_{y,f} = \mu F_{z,f} \sin(C_f \arctan(B_f a_f)) \quad (7c)$$

$$F_{y,r} = \mu F_{z,r} \sin(C_r \arctan(B_r a_r)) \quad (7d)$$

where α_f, α_r denotes tire side slip angles; μ denotes tire-road friction coefficient; C, B are Pacejka tire parameters.

Finally, these equations describe the system evolution. The state variables $v_x, v_y, \omega_z, e_\psi, s, e_y$ describe the position and velocity of a vehicle in the Frenet frame of the race track. The control variables a, δ describe the acceleration and steering input of the vehicle.

$$\dot{v}_x = \frac{1}{m}(2F_{x,r} + 2F_{x,f} \cos \delta - 2F_{y,f} \sin \delta - F_D) + \omega_z v_y \quad (8a)$$

$$\dot{v}_y = \frac{1}{m}(2F_{y,r} + 2F_{y,f} \cos \delta + 2F_{x,f} \sin \delta) - \omega_z v_y \quad (8b)$$

$$\dot{\omega}_z = \frac{1}{J_{zz}}((2F_{y,f} \cos \delta + 2F_{x,f} \sin \delta)l_f - 2F_{y,r}l_r) \quad (8c)$$

$$\dot{e}_\psi = \omega_z - k \frac{v_x \cos e_\psi - v_y \sin e_\psi}{1 - e_y k} \quad (8d)$$

$$\dot{s} = \frac{v_x \cos e_\psi - v_y \sin e_\psi}{1 - e_y k} \quad (8e)$$

$$\dot{e}_y = v_x \sin e_\psi + v_y \cos e_\psi \quad (8f)$$

where J_{zz} denotes the vehicle's yaw moment of inertia.

We note that despite the level of complication, the vehicle dynamics parameters involved in these equations are readily available in racing applications.

D. Observation Functions

Of the six state variables, the angular velocity ω_z is directly measurable by the IMU, and the longitudinal velocity v_x is directly measurable by the wheel speed sensor. The lateral velocity v_y is calculated based on the average wheel speed v_w and current steering angle δ :

$$v_y = v_w \tan(\delta)$$

Since we are handling GNSS-denied situations, the Frenet pose of the vehicle is not measured but instead integrated in the EKF.

E. Point Cloud Distortion Correction

After a LiDAR starts to register a new point cloud, it takes time to rotate to a certain angle and perform the measurement. Similar to camera motion distortion, this creates a jelly effect on the point cloud under high speed. The IAC car is equipped with 3 Luminar Hydra LiDARs with individual timestamps for every laser scan, so we can correct the motion distortion using the EKF-estimated velocity.

Assuming a point is taken at angle θ with distance d after time Δt has passed since the beginning of the scan, the corrected measurement is

$$d_{\text{corr}} = d - \mathbf{v} \cdot \Delta t \cdot \omega \quad (9a)$$

$$\theta_{\text{corr}} = \theta - \omega \cdot \Delta t \quad (9b)$$

where \mathbf{v}, ω are the velocities.

IV. EXPERIMENT

We evaluated our online high-speed localization pipeline on the RACECAR dataset, specifically on the high-speed multi-car data between Technical University of Munich (TUM) and Polytechnic University of Milan (PoliMove) at Las Vegas Motor Speedway in 2023. The vehicles travel at a speed of up to 140 MPH.

To compare the effect of distortion correction, we set up an alternative pipeline that uses the raw uncorrected point cloud for NDT matching. We then compare the lateral and longitudinal error of the final localization output compared to the ground-truth GNSS trajectory.

V. RESULT

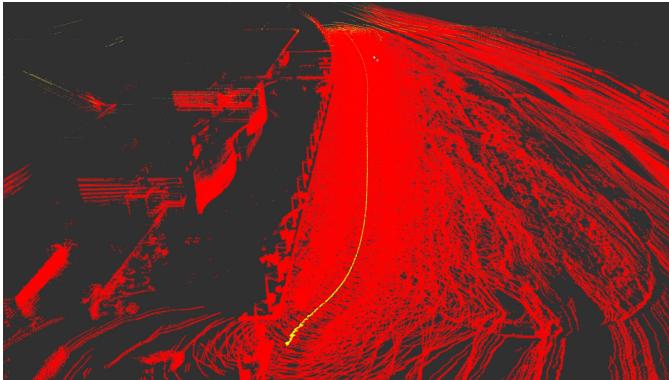


Fig. 7. Point cloud map of the Las Vegas Motor Speedway.

Figure 7 visualizes the point cloud map constructed using LIO-SAM [9]. Significant preprocessing was necessary on top of the RACECAR dataset to improve the quality of the map, including IMU filtering, timestamp alignment, and frame alignment.

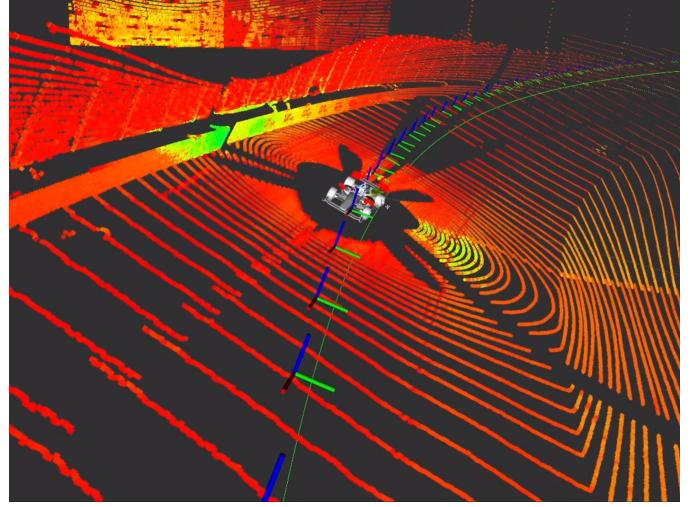


Fig. 8. Odometry visualization showing consistent, overlapping trajectories on the banked turn during two consecutive laps.

The direct result of the distortion correction pipeline can be visualized in Figure 2, which shows the correction working at 100 MPH.

We then proceed to discuss the important results of our experiment, which is the online localization pipeline in figures 9 and 10.

Figure 9 shows the error of the longitudinal position of the vehicle on the race track. The x-axis also labels the track sections. It shows that our distortion correction pipeline can improve the longitudinal localization quality by around 1 meter in the turns and around 0.5 meter on the straights, compared to not using any corrections. We are able to achieve an average longitudinal error of around 1.2 m in the turns and 0.5 on the straights. It also suggests that longitudinal localization is more challenging in turns, as both methods incur a higher localization error in those sections.

Figure 10 shows the error of the lateral position of the vehicle. This is calculated in the normal direction of the track center line. Once again, our distortion correction pipeline shows improvements, although not as apparent as in the longitudinal localization case. This is presumably because the lateral motion of the vehicle is relatively small compared to the longitudinal traveling speed, thus suffering less from any motion distortion problem, especially on the straights. However, during turns, the lateral error also has an increasing trend using both methods, suggesting a more complex lateral dynamics in these areas.

VI. CONCLUSION

We proposed an offline mapping and online localization pipeline for autonomous racing in GNSS-denied scenarios. We designed a variant of EKF that can process asynchronous observations and takes advantage of a high-fidelity vehicle dynamics model in the Frenet frame to improve the filter predictions. The resulting odometry estimation is used to carry

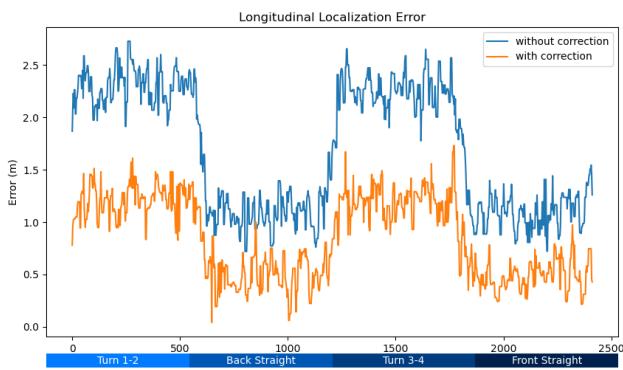


Fig. 9. Longitudinal localization error: with and without point cloud motion correction.

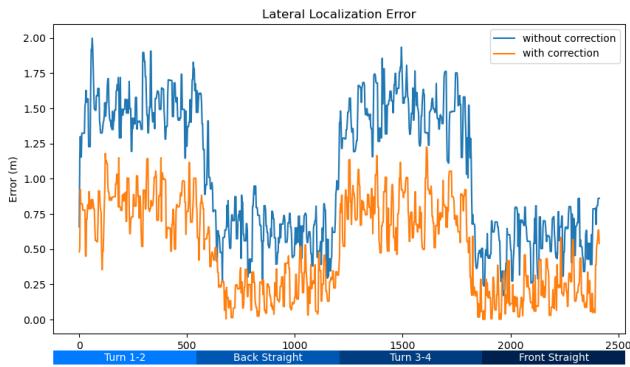


Fig. 10. Lateral localization error: with and without point cloud motion correction.

out LiDAR distortion correction, which proves to improve the localization result, especially in the longitudinal direction.

Future work can be done to improve this pipeline. First, the observation function used in the EKF is highly simplified. A more sophisticated model can be used to accurately calculate longitudinal and lateral velocities based on individual wheel speeds and steering angle. Second, to better handle a featureless environment like a race track, we can take advantage of visual-lidar odometry methods [15] and combine the current state estimation system with visual odometry.

REFERENCES

- [1] Indy Autonomous Challenge - Official Website, February 2023.
- [2] P. Biber and W. Strasser. The normal distributions transform: a new approach to laser scan matching. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, volume 3, pages 2743–2748 vol.3, 2003.
- [3] Alexander Heilmeier Fabian Christ, Alexander Wischnewski and Boris Lohmann. Time-optimal trajectory planning for a race car considering variable tyre-road friction coefficients. *Vehicle System Dynamics*, 59(4):588–612, 2021.
- [4] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45, 03 1960.
- [5] Amar Kulkarni, John Chrosniak, Emory Ducote, Florian Sauerbeck, Andrew Saba, Utkarsh Chirimar, John Link, Marcello Cellina, and Madhur Behl. RACECAR – The Dataset for High-Speed Autonomous Racing, June 2023. arXiv:2306.03252 [cs].
- [6] Thomas Moore and Daniel Stouch. A generalized extended kalman filter implementation for the robot operating system. In Emanuele Menegatti, Nathan Michael, Karsten Berns, and Hiroaki Yamaguchi, editors, *Intelligent Autonomous Systems 13*, pages 335–348, Cham, 2016. Springer International Publishing.
- [7] Hans B. Pacejka and Igo Besselink. *Tire and Vehicle Dynamics*. Elsevier, 2012.
- [8] Florian Sauerbeck, Lucas Baierlein, Johannes Betz, and Markus Lienkamp. A Combined LiDAR-Camera Localization for Autonomous Race Cars. *SAE International Journal of Connected and Automated Vehicles*, 5(1):61–71, January 2022. Number: 12-05-01-0006.
- [9] Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Rus Daniela. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5135–5142. IEEE, 2020.
- [10] Stahl, Tim, Wischnewski, Alexander, Betz, Johannes, and Lienkamp, Markus. Ros-based localization of a race vehicle at high-speed using lidar. *E3S Web Conf.*, 95:04002, 2019.
- [11] Shinya Sumikura, Mikiya Shibuya, and Ken Sakurada. OpenVSLAM: A Versatile Visual SLAM Framework. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM ’19, pages 2292–2295, New York, NY, USA, 2019. ACM.
- [12] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT Press, 2010.
- [13] Alexander Wischnewski, Tim Stahl, Johannes Betz, and Boris Lohmann. Vehicle dynamics state estimation and localization for high performance race cars**research was supported by the basic research fund of the institute of automotive technology of the technical university of munich. *IFAC-PapersOnLine*, 52(8):154–161, 2019. 10th IFAC Symposium on Intelligent Autonomous Vehicles IAV 2019.
- [14] Ji Zhang and Sanjiv Singh. Loam: Lidar odometry and mapping in real-time. 07 2014.
- [15] Ji Zhang and Sanjiv Singh. Visual-lidar odometry and mapping: Low-drift, robust, and fast. volume 2015, 05 2015.