Rail-Constrained Train Localization Using Multilayer Map

Mochamad Ardiansyah Nugraha[1], Antoine Manzanera[1], Philippe Xu[1]

***Abstract*—Existing train self-localization suffers from inaccuracy due to the slip-prone wheel odometers and relies heavily on trackside beacons for absolute positioning. Leveraging the constrained trajectory of trains, recent advances in sensor fusion for autonomous systems provide new opportunities to improve localization precision. This work aims to exploit these characteristics to deliver a more reliable and precise train localization. We propose a multilayer map-based localization that combines a furnished map-based track geometry with a sensor-derived map to jointly constrain the state estimation. Preliminary results demonstrate that integrating map priors with sensor observations allows mutual compensation between inconsistent maps and measurements, removing the dependence on a single information source and improving robustness.**

***Index Terms*—Train localization, rail-constrained kinematics, multilayer map, sensor fusion, particle filter, LiDAR odometry.**

I. Introduction

Current localization technologies in autonomous trains heavily rely on beacons for absolute localization, paired with wheel odometers. These beacons are installed in certain spacing along the track, which requires high investment, proportional to the length of the track. Despite simple, the use of wheel odometers causes inaccurate readings due to slippage, especially in adverse weather or on steep slopes. While solely using Global Navigation Satellite System (GNSS) is not adapted for urban and underground environments like railways, recent advances in sensor fusion and computer vision tried to mitigate such challenges [1] - [3].

Unlike other ground vehicles, train movement is constrained to its track. It can only move in a longitudinal direction within a fixed trajectory. In addition, the track map is usually known. Previous works [4] - [6] addressed this by leveraging the track geometry as prior, applying a 1D motion model to simplify the state estimation. Zhang et al. [4] utilized estimated curvature from a dead-reckoning IMU and match it with a prior map. Similarly, Von Einem et al. [5] constrained the state space along 1D track control points. In contrast, Liu et al. [6] used heading angles and compare them to the 2D track map polylines to correct the state. However, [5] [6] still rely on GNSS in their fusion approaches, which cannot be generalized for all railway scenarios, notably in tunnels and dense urban environments. Moreover, all of them depend only on a single map, whose consistency might change over time.

[1]U2IS, Ecole nationale supérieur de techniques avancées (ENSTA), Institut Polytechnique de Paris, Palaiseau, France. Corresponding author: mochamad.nugraha@ensta.fr.

Research supported by Agence Nationale de la Recherche et de la Technologie (ANRT) France with fellowship funding Cifre n°2025/1181.

In this study, we propose a multilayer map-based train localization. It combines the theoretical track geometry map from engineering blueprints and a recent sensor map built offline for a more precise localization designed for autonomous trains. This approach balances the anchor between physics and sensor reality which can compensate if the track map is outdated, and vice versa.

II. Multilayer Map-based Localization

A. Multilayer Map

a) Track geometry: A track map $\mathcal{M}^G$ can be expressed as a set of j piecewise circular track segments m_i^G defined by its start curvilinear abscissa $\underline{s}$, end abscissa $\overline{s}$, curvature value κ, start altitude $\underline{h}$, and end altitude $\overline{h}$.

$$\mathcal{M}^G = \{m_i^G\}_{1:j} = \{([\underline{s}, \overline{s})_i^G, \kappa_i^G, [\underline{h}, \overline{h})_i^G)\}_{1:j} \tag{1}$$

with $\kappa < 0$ for right-turn curve, $\kappa = 0$ for straight segment, and $\kappa > 0$ for left-turn curve.

b) Keyposes (sensor map): A map from previously-acquired sensor-based mapping is also used on top of the track geometry to refine the localization. This map $\mathcal{M}^S$ is represented as a set of k keyposes p_i^S. Sensor map normally comprises 6 DoF pose (position and orientation) in global Euclidean frame $[x, y, z, \theta, \phi, \psi]^T$. In this work, the sensor map is adapted to rail representation:

$$\mathcal{M}^S = \{p_i^S\}_{1:k} = \{(x_i^S, y_i^S, z_i^S, \psi_i^S, s_i^S, \kappa_i^S)\}_{1:k} \tag{2}$$

with curvature value $\kappa = \Delta\psi/\Delta s$ and displacement $\Delta s = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$.

B. Rail-constrained Motion Model

Trains basically run in a one-dimensional manifold, tied to its track along the longitudinal distance or curvilinear abscissa s. We can express its heading or yaw ψ with respect to the global Euclidean frame. Thanks to its constrained nature, it is sufficient to localize a train by only using these two variables. Using the assumption of constant speed $\dot{s}$ and constant yaw rate $\dot{\psi}$, the state of a train can be represented as $\mathbf{x} = [s, \psi, \dot{s}, \dot{\psi}]^T$. Consequently, the state transition can be expressed as

$$\mathbf{x}_t = \begin{bmatrix} s_{t-1} + \dot{s}_{t-1} \cdot \Delta t \\ \psi_{t-1} + \dot{\psi}_{t-1} \cdot \Delta t \\ \dot{s}_{t-1} \\ \dot{\psi}_{t-1} \end{bmatrix} + \varepsilon_t \tag{3}$$

with ε_t is a process noise vector.

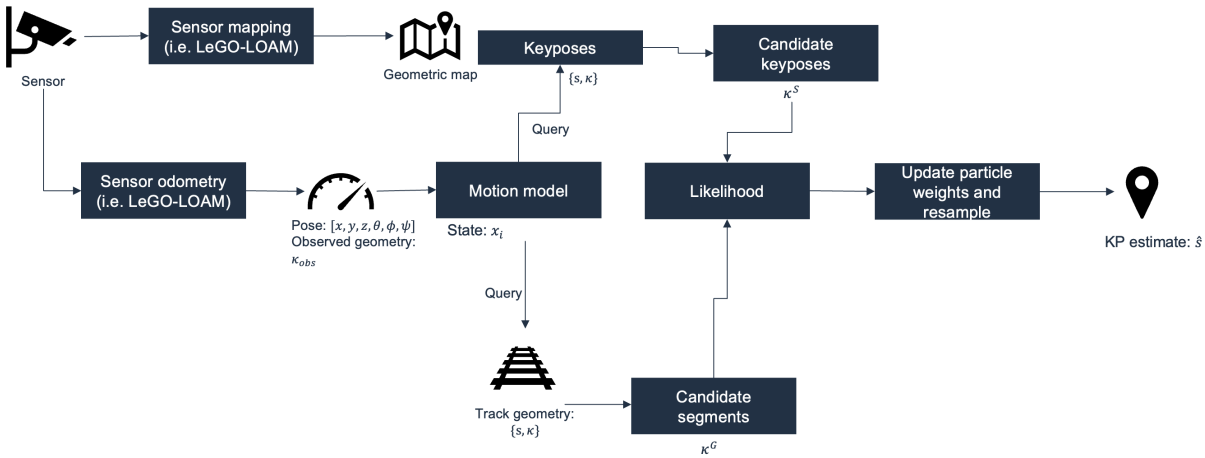

Fig. 1. Our proposed methodology of multilayer map-based train localization.

C. Measurement Model

Before updating the predicted state, the measurements are done three folds. An actual measurement of sensor odometry is used as observation $\mathbf{z_t}$, followed by the predicted measurements by two-stage map-matching $h(\mathbf{x}_t; \mathcal{M}^G, \mathcal{M}^S)$. Generally, sensor odometry yields 6-DoF pose $\mathbf{p}_{obs} = [x, y, z, \theta, \phi, \psi]^T$. For our case, we are interested only in the observed speed and geometry:

$$\mathbf{z}_t = \begin{bmatrix} \dot{s}_{obs} \\ \dot{\psi}_{obs} \\ \kappa_{obs} \end{bmatrix} \tag{4}$$

with $\dot{s}_{obs}$, $\dot{\psi}_{obs}$, and κ_{obs} are the train speed, yaw rate, and track curve value from sensor odometry, respectively.

Further, we do map matching by querying both track geometry map $\mathcal{M}^G$ and sensor map (keyposes) $\mathcal{M}^S$ using the predicted curvilinear abscissa s_t to get the candidate track segments represented by its curvature value:

$$h(\mathbf{x};_t \mathcal{M}^G, \mathcal{M}^S) = \begin{bmatrix} \dot{s}_t \\ \dot{\psi}_t \\ \kappa^G(s_t) \\ \kappa^S(s_t) \end{bmatrix} \tag{5}$$

with $\kappa^G(s_t) = \kappa_i^G$ such that $\underline{s}_i^G \leq s_t < \overline{s}_i^G$, and $\kappa^S(s_t) = \kappa_i^S$ such that $\underline{s}_i^S \leq s_t < \overline{s}_i^S$.

Finally, the innovation is assessed by computing the residuals between the observed and predicted measurements, such that

$$\mathbf{r}_t = \begin{bmatrix} r_{\dot{s},t} \\ r_{\dot{\psi},t} \\ r_{G,t} \\ r_{S,t} \end{bmatrix} = \tilde{\mathbf{z}}_t - h(\mathbf{x}_t; \mathcal{M}^G, \mathcal{M}^S) = \begin{bmatrix} \dot{s}_{obs} - \dot{s}_t \\ \dot{\psi}_{obs} - \dot{\psi}_t \\ \kappa_{obs} - \kappa^G(s_t) \\ \kappa_{obs} - \kappa^S(s_t) \end{bmatrix} \tag{6}$$

III. Implementation

The experiment was carried out online using a real train set on a 1.6 km test track in the north of France. The train ran up to a maximum speed of 50 km/h and stopped at two different platforms. Under sunny weather conditions and a relatively flat track, the existing localization using beacons and wheel odometers is referred as ground truth, assuming no significant errors. The track geometry map was acquired from the infrastructure blueprint. In addition, we utilized a solid-state MEMS LiDAR Robosense M1 for building the sensor map, as well as during the runtime for LiDAR odometry. We considered using LeGO-LOAM [7] both for offline mapping and online odometry. Its lightweight runtime performance tailored for ground vehicles and the ability to capture edge and planar features are suitable for a structure-less environment like railways.

Due to lack of benchmark of rail localization, we employ Monte Carlo Localization (MCL) [8], or particle filter, as our proof-of-concept to build a solid baseline. As shown in Algorithm 1, the particle weights are updated based on the likelihood on how close the predicted state and measurements are to the actual observation from the LiDAR odometry. σ^2 represents the uncertainty of the sensor measurement. The Mahalanobis distances are weighted based on each channel i, parameterized by λ_i with $\Sigma_i \lambda_i = 1$.

We compare the performance using the available ground truth abscissa as shown in Fig 2, with average longitudinal error $e = \hat{s} - s_{GT}$, is 69.28 m and root mean square error (RMSE) of 69.40 m. This error is computed right after particles started to converge 14 s after the start. The evolution of error is reported as in Fig. 3, along with the respective speed.

The longitudinal error may be emphasized by perceptual aliasing. When particles are distributed along an extended track segment, they exhibit similar likelihood values, resulting ambiguity in the estimated abscissa. A similar effect occurs when conflicting likelihoods arise between two distinct track segments, including one located near the end of the track ($s > 4000$ m). After approximately 14 s, particles in this segment collapse, causing the filter to converge toward the

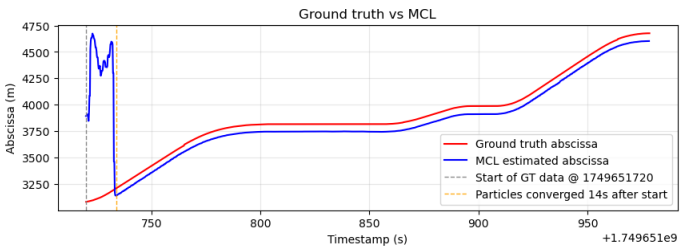

Fig. 2. Comparison of evolution of localization abscissa between ground truth (red) and map-based MCL (blue).

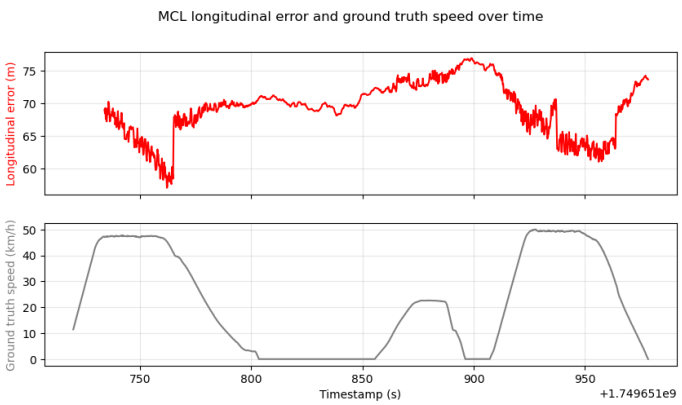

Fig. 3. Longitudinal error (red, top) and ground truth speed (bottom)

Algorithm 1 Map-based Monte Carlo Localization

Require: Maps $(\mathcal{M}^G, \mathcal{M}^S)$, particle set $\{x_i, w_i\}_{i=1}^N$, odometry input

Ensure: Estimated pose on track $(\hat{s}, \hat{\psi})$

while odometry arrives **do**
  Read odometry and compute observations, $\mathbf{z}_t$
  Predict: propagate each particle with motion model + process noise, $\mathbf{x}_t$
  Map matching: evaluate map features at each particle position, $h(\mathbf{x}_t; \mathcal{M}^G, \mathcal{M}^S)$
  Update: compute residuals and weighted likelihood for each particle $\ell_i \propto \exp\left(-\frac{1}{2}\left[\lambda_{\dot{s}} \frac{r_{\dot{s},i}^2}{\sigma_{\dot{s}}^2} + \lambda_{\dot{\psi}} \frac{r_{\dot{\psi},i}^2}{\sigma_{\dot{\psi}}^2} + \lambda_G \frac{r_{G,i}^2}{\sigma_G^2} + \lambda_S \frac{r_{S,i}^2}{\sigma_S^2}\right]\right)$
  Update weights: $w_i \leftarrow w_i \ell_i$, then normalize $\sum_i w_i = 1$
  Compute $N_{eff} = 1/\sum_i w_i^2$
  if N_{eff} is below threshold **then**
    Resample particles and reset weights uniformly
  end if
  Estimate state by weighted mean: $\hat{s} = \sum_i w_i s_i$, $\hat{\psi} = \sum_i w_i \psi_i$
end while
return $(\hat{s}, \hat{\psi})$

beginning of the track ($s \approx 3200$ m).

IV. Conclusion

This paper shows a novel idea to leverage a GPS-free train localization using multilayer map. Using a simple methodology as baseline, this preliminary study demonstrates a promising result by combining the constrained-nature of railway and known priors. Future work will take into account slope likelihood from the track geometry as well as keyframe fitness score between the sensor map and observed point clouds to compensate perceptual aliasing in a degraded railway environment. Additionally, an infrastructure map describing the location of known signaling objects (e.g., signal posts, switches) along the track could be incorporated as a prior. Combined with object segmentation, these elements could serve as visual beacons for absolute localization.

References

[1] Y. Wang, W. Song, Y. Wang, X. Dai and Y. Lou, "MetroLoc: Metro Vehicle Mapping and Localization With LiDAR-Camera-Inertial Integration," in IEEE Transactions on Intelligent Transportation Systems, vol. 26, no. 2, pp. 1441-1453, Feb. 2025, doi: 10.1109/TITS.2024.3512000.

[2] J. Wortmann, F. Wulff, B. Schäufele, and I. Radusch, "LiDAR Based Landmark and Odometry Fusion for Precise Rail Localization Without Additional Infrastructure," in IEEE 101st Vehicular Technology Conference (VTC2025-Spring), Oslo, Norway: Jun. 2025, pp. 1–6. doi: 10.1109/VTC2025-Spring65109.2025.11174496.

[3] J. Xiong, C. Yuan, L. Xia, L. Xiao, Z. Sun, and G. Peng, "A LiDAR-IMU Tightly Coupled SLAM in Subway Tunnel Environments," in 5th International Conference on Computer, Control and Robotics (ICCCR), Hangzhou, China: May 2025, pp. 453–458. doi: 10.1109/ICCCR65461.2025.11072592.

[4] R. Zhang, C. Chen, H. Tang, Y. Qin, Z. Lai, and Y. Bai, "Onboard Metro Train Localization Based on the Train Motion and Track Geometry Feature Fusion," IEEE Trans. Intell. Transport. Syst., vol. 26, no. 10, pp. 16275–16290, Oct. 2025, doi: 10.1109/TITS.2025.3571433.

[5] C. Von Einem, A. Cramariuc, R. Siegwart, C. Cadena, and F. Tschopp, "Path-Constrained State Estimation for Rail Vehicles," in 2023 26th International Conference on Intelligent Transportation Systems (ITSC), Bilbao, Spain: Sep. 2023, pp. 4600–4607. doi: 10.1109/ITSC57777.2023.10422075.

[6] J. Liu, B. Cai, and J. Wang, "Track-constrained GNSS/odometer-based train localization using a particle filter," in IEEE Intelligent Vehicles Symposium (IV), Gothenburg: Jun. 2016, pp. 877–882. doi: 10.1109/IVS.2016.7535491.

[7] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid: Oct. 2018, pp. 4758–4765. doi: 10.1109/IROS.2018.8594299.

[8] F. Dellaert, D. Fox, W. Burgard and S. Thrun, "Monte Carlo localization for mobile robots," Proceedings IEEE International Conference on Robotics and Automation, Detroit, MI, USA, 1999, pp. 1322-1328 vol.2, doi: 10.1109/ROBOT.1999.772544.