UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Henri Poolakese

GNSS-free localization using milestone-board information

Bachelor's Thesis (9 ECTS)

Supervisor(s):
Naveed Muhammad, Ph.D.

Tartu 2025

GNSS-free localization using milestone-board information

Abstract:

This thesis explores the problem of autonomous vehicle localization without any reliance on GNSS (Global Navigation Satellite System). The proposed solution simulates a vision-based localization system that leverages destination signs embedded as virtual tags within OpenStreetMap (OSM) data. These semantic cues act as observational constraints for a particle filter-based localization framework.

The system architecture consists of a robot model, particle filter, sensor model, and visualization module. Experimentations were conducted using real road network data from OpenStreetMap. Experiments compared runs with different particle counts (200, 500, 1000 and 2000), showing that high localization accuracy — up to 99% — was achievable with help from road numbers. The application of destination tag constraints significantly improved accuracy at specific steps, in some cases by over 70%.

The results confirm that even with limited computational resources, robust localization can be achieved using semantic information from maps. This work contributes a lightweight and modular solution for the problem.

Keywords: Autonomous vehicles, localization, particle filter, OpenStreetMap
CERCS: P175 Informatics, systems theory, T125 Automation, robotics, control engineering

Satelliitnavigatsioonivaba positsioneerimine kaugusviitade abil

Lühikokkuvõte:

See lõputöö käsitleb autonoomsete sõidukite lokaliseerimist olukorras, kus GNSS-signaal ei ole (näiteks tunnelites, metsateedel või kaugemates maa piirkondades). Selline lahendus simuleerib lokaliseerimissüsteemi, mis kasutab OpenStreetMap-andmestikku ja kaugusviite semantiliste vihjetena aukohaleismisel.

Programm koosneb robotimudelist, osakestefiltrist, sensorist ja visualiseerimismoodulist. Eksperimendid viidi läbi Eesti teedevõrgustikus OpenStreetMap andmetel. Tehtud katsetused näitasid, et lokaliseerimise täpsus ulatus kuni 99%-ni. Kaugusviitade kasutamine aitas roboti asukohaleidmist teostada.

Võtmesõnad: Autonoomsed sõidukid, lokaliseerimine, osakestefilter, OpenStreetMap
CERCS: P175 Informaatika, süsteemiteooria, T125 Automatiseerimine, robootika, juhtimistehnika

Table of Contents

1. Introduction 1
    1.1 Structure of the Manuscript 2
2. Background 4
    2.1 Localization 4
    2.2 Destination Sign Information 4
    2.3 Particle Filter Localization 5
    2.4 Vision-Based Sensors 5
    2.5 Sensor Fusion 5
    2.6 OpenStreetMap and Road Network Representation 6
3. System Design 7
    3.1 Overview 7
    3.2 Architecture 7
4. Implementation 9
    4.1 Simulation Environment 9
    4.2 Robot Model 9
    4.3 Particle Filter 9
    4.4 Sensor Simulation 11
    4.5 Visualization and Logging 11
5. Experiments and Results 12
    5.1 Experiment Setup 12
    5.2 Experiments 13
        5.2.1 Localization without road numbers 13
        5.2.2 Localization with road numbers 23
6. Discussion 31
    6.1 Accuracy Trends 31
    6.2 Limitations of the Experiments 32
7. Conclusion 33
    7.1 Summary of Contributions 33
    7.2 Final Thoughts 33

References 34

License 36

1. Introduction

Accurate localization is critical for the effective operation of autonomous vehicles, particularly in environments where GNSS (Global Navigation Satellite System) signals are unreliable or unavailable, such as forest roads, tunnels, or rural areas [1]. Traditional GNSS-based methods may fail in these settings, presenting a challenge for autonomous navigation. In addition to that, for true autonomy, an autonomous vehicle should be capable of localizing itself even in the event of GNSS sensor failure. This thesis addresses these challenges by proposing and testing a novel approach to vehicle localization that leverages vision-based detection of destination sign information embedded within OpenStreetMap (OSM) data and tests whether it can enable accurate localization in GNSS-denied environment using a particle filter, under idealized conditions. This work explores the feasibility of utilizing virtual destination signs (also referred to as milestone boards) for localization in scenarios with limited sensor input.

Destination signs, often found along roads, provide valuable contextual information about nearby cities, distances and the road network. In Figure 1, red square indicates the

Figure 1. Example of Estonian destinaiton sign.

road number, the green rectangle is the European road number, white text is the destination name and distance to that destination. These signs can be interpreted as cues for autonomous vehicles to localize themselves—just like an experienced human driver would be able to estimate their coarse location on a map (a paper map, or even an abstract map in their mind) — bypassing the need for GNSS [2]. For true autonomy, level-5 vehicles of the future should be resilient against a GNSS sensor failure and thus should be able to localize without complete reliance on GNSS (just like a human driver can).

The objective of this research is to prototype and evaluate a vision-based localization system using virtual destination signs embedded into OSM data. The system employs a particle filter approach to estimate the vehicle's location based on synthetic destination sensing, leveraging contextual map information rather than relying on traditional GNSS signals [1, 3].

This study focuses on simulating vehicle localization in a controlled environment using pre-tagged destination data from OpenStreetMap. While the system mimics the behavior of a vision-based sensor, it does not involve real-world image processing, assuming perfect access to destination signs without accounting for real-world errors such as tag misinterpretation, visual occlusion, or sensor noise.

There exists substantial literature addressing localization where raw GNSS is available, and the aim is to refine it using perception. For example, Barsan et al. [4] and Wei et al. [5] propose systems that enhance GNSS localization using LiDAR intensity maps and compressed binary maps respectively. This thesis, in contrast, assumes no GNSS availability whatsoever and investigates localization based solely on destination sign cues, making it relevant for GNSS-denied environments.

1.1 Structure of the Manuscript

- **Chapter 2 – Background / Theory** introduces key localization concepts, destination sign information, particle filters, and relevant map data.

- **Chapter 3 – System Design** outlines the system components, architecture, and interaction between modules

- **Chapter 4 – Implementation** describes how the simulation and particle filter are realized in code.

- **Chapter 5 – Experiments** describes experiments

- **Chapter 6 – Results** evaluates the system under different conditions and reports metrics.

- **Chapter 7** – **Conclusion**

All of the text in this thesis was written by the author. No generative AI tools were used for content creation.

2. Background

2.1 Localization

Localization is the process of determining a vehicle's position and orientation within a given environment. A more formal definition of localization is: *"The problem of estimating a robot's position and orientation within a known map, using noisy sensor data and a probabilistic model of motion and sensing."* [6].

The challenge lies in integrating multiple data sources, such as vision-based sensors and OSM, to improve localization accuracy in the environment.

Autonomous vehicles typically rely on a combination of sensors to estimate their location [1]. GNSS is commonly used, but in forests or remote areas where GNSS signals may be unreliable or unavailable, alternative methods are needed. It is critical for maintaining reliable GNSS-free vehicle positioning to ensure smooth continuous operation. Techniques such as visual localization and sensor-based localization using odometry or LiDAR can be employed in these situations [2, 3]. Vision-based systems use cameras to identify features in the environment, while particle filters can estimate the vehicle's position relative to a pre-built map.

2.2 Destination Sign Information

Destination sign information, or destination tags, can be utilized for localization by providing contextual clues about nearby cities or roads [3]. These destination tags are not embedded in OpenStreetMap (OSM) data by default so manual editing of data is needed. By recognizing and interpreting these destination signs, an autonomous vehicle can estimate its position based on the distance to the nearest sign, city or road number.

Unlike traditional fiducial markers, destination tags are not placed intentionally for localization purposes, but their semantic information can serve as a powerful cue for vehicle localization [3]. For example, prior works using fiducial markers (e.g., ARTag, AprilTag) have demonstrated the utility of artificial visual tags in localization tasks [7].

2.3 Particle Filter Localization

Particle filters, or Monte Carlo Localization (MCL), provide a robust approach to vehicle localization in uncertain environments [1]. The core idea behind particle filtering is to represent a belief about the robot's position as a set of particles, each representing a possible state of the system. These particles are propagated according to the vehicle's motion model, and their weights are updated based on sensor observations [2].

In the context of this thesis, the robot utilizes virtual destination tags as sensory input. Each particle in the filter is weighted based on the information of the sensed destination tags, and particles with higher weights are considered more likely to represent the robot's actual position. The resampling step is probabilistic: particles are drawn according to their weights, so more likely hypotheses are reinforced while unlikely ones are discarded.

This approach is particularly useful in environments where sensor noise and uncertainty are prevalent, as the filter can effectively handle the ambiguities inherent in localization tasks [1].

2.4 Vision-Based Sensors

Vision-based sensors, such as cameras, provide valuable input for localization tasks, especially in environments where other sensor types, like GNSS, may be unreliable [2]. These sensors can identify objects, road signs, and other visual cues that are essential for localization. However, vision-based localization presents challenges, such as lighting conditions, occlusions, and varying environmental factors. In this thesis, vision-based sensors are simulated without these challenges.

2.5 Sensor Fusion

Sensor fusion combines data from multiple sensors to improve localization accuracy [2]. By integrating different sensor types, such as cameras and GNSS, a more reliable estimate of the vehicle's position can be obtained.

In this thesis GNSS won't be used and the virtual destination tags act as a key component of the sensor fusion process. The particle filter updates its belief about the robot's position based on the information derived from these tags, enhancing the overall accuracy of the localization system [1].

2.6 OpenStreetMap and Road Network Representation

OpenStreetMap (OSM) provides an open-source, detailed map of the world's road networks, including information about streets, intersections and cities. OSM data can be used to represent road networks for autonomous vehicle localization, offering a map-based framework for integrating sensor data [8].

3. System Design

3.1 Overview

The system designed in this thesis simulates an autonomous vehicle moving through a real-world road network in Estonia. It uses a pre-defined route between cities and attempts to localize itself using virtual destination tags embedded in OpenStreetMap (OSM) data. These tags simulate destination signs that might be visible in a real-world setting and are used to constrain the robot's estimated position.

A particle filter is employed to estimate the robot's position over time. The system also incorporates a sensor model that simulates the reading of destination signs as the robot progresses along its path. The simulated sensor detects destination tags at specific intervals and updates the robot's internal state.

3.2 Architecture

The system is composed of several core components:

- **Robot Model:** Simulates the robot's movement along the road network.

- **Particle Filter:** Maintains a set of particles representing possible robot positions and updates them based on motion and sensor inputs.

- **Sensor Model:** Simulates destination detection by reading virtual destination signs placed along the road.

- **Map Interface:** Provides access to the road network data from OpenStreetMap.

- **Visualization Module:** Uses Matplotlib to render the current position of the robot and particles on a dynamic map.

The architecture enables modular development and testing of each component, with particular emphasis on evaluating the utility of milestone-board information for localization.

Localization estimation is done with the particle filter's weighted mean position, it is continuously compared to the robot's ground truth.

4. Implementation

4.1 Simulation Environment

The simulation uses Python and the OSMnx library to download and manipulate the road network from OpenStreetMap. The road network graph is loaded for a specific geographic region in Estonia, and a shortest path is computed between two selected cities. This path forms the basis of the robot's route.

Destination tags are manually added to certain nodes and ways in the OSM data to simulate the presence of real-world destination signs. These tags represent signs that the robot can "see" as it moves through the environment. (Note: The reader can access the code here, https://github.com/HenriPoolakese/ParticleFilter).

4.2 Robot Model

The robot model represents an idealized vehicle that moves one step at a time along the precomputed shortest path. At each step, the robot checks whether the current node or the current road segment has an associated destination tag nearby. If such a tag exists, it is added to the robot's list of observed destinations.

4.3 Particle Filter

The particle filter maintains a set of particles, each representing a hypothesis of the robot's location. Particles move according to the robot's motion model, with random variation introduced to simulate uncertainty. When the robot detects a new destination tag, the filter updates each particle's weight based on the particle's location.

Resampling is done using a probabilistic process. The set of particles is resampled according to their weights, where higher-weight particles are more likely to be chosen. This allows the particle cloud to converge around the robot's true location over time.

Road network-based localization is done by maintaining digital map of Estonia's road network, represented as a graph `G = (V, E)`, where V is the nodes of the network and E is the roads between nodes with associated data like road length and geometry.

The particle filter consists of M particles, where each particle represents a hypothesis of the robot's state. The state of the filter at time t is represented as: $X_t = \{x_t^{(1)}, x_t^{(2)}, ..., x_t^{(M)}\}$. Each particle contains information like position, current road segment, movement, and weight.

The filter operates through the following update steps:

1. Initialization - Generates initial particle set X_0 by uniformly distributing particles across the road network with equal weight: $x_0^{(m)} \sim \text{Uniform}(V)$ and $w_0^{(m)} = 1/M$ for all $m \in \{1,...,M\}$ where V is the set of all road nodes in the graph G.

2. Motion Update - For each particle $x_{t-1}^{(m)}$, propagate according to the robot's motion model with noise: $x_t^{(m)} \sim p(x_t^{(m)} \mid x_{t-1}^{(m)}, u_t)$ where u_t is the robot's commanded motion, for example (step length = 50m). The motion model is implemented as:

    1. Select random neighbor edge
    2. Move along edge by distance $\Delta d = u_t + \mathcal{N}(0, \sigma^2)$
    3. If reaches a new node v, repeat process from new node

This constrained random walk ensures particles remain on valid road segments while simulating motion uncertainty.

3. Measurement Update - When destination signs (constraints) are observed (z_t)at time t, update weights according to: $w_t^{(m)} = \eta \cdot p(z_t \mid x_t^{(m)}) \cdot w_{t-1}^{(m)}$

where η is a normalization factor and the observation model consider two factors:

    1. Distance constraints: Weight penalty increases based on distance.
    2. Road number constraints (if enabled): If on correct road weight remains unchanged

The final weight combines these factors multiplicatively.

4. Resampling - Resample particles with probability proportional to their weights using systematic resampling:

    1. Calculate normalized weights: $\acute{w}_t^{(m)} = w_t^{(m)} / \Sigma\, w_t^{(m)}$

2. Generate ordered thresholds: $u^{(k)} = (k + \tilde{u})/M$ where $\tilde{u} \sim \text{Uniform}(0,1)$

3. Select particles where cumulative weight crosses thresholds: $X_t = \{x_t^{(j)} \mid \sum_{i=1}^{j-1} \acute{w}_t^{(i)} \leq u^{(k)} < \sum_{i=1}^{j} \acute{w}_t^{(i)}\}$

This maintains particle diversity while focusing computation on high-probability regions.

Implementation Notes: Uses cached shortest-path distances for efficient weight calculation, each nodes distance from each destination is pre-computed Particles maintain only road-constrained movement through graph operations.

4.4 Sensor Simulation

The sensor is designed to detect destination signs. The robot senses based on the distance to the destination tag. These readings are used to update the particle filter [1].

The simulation assumes perfect vision (i.e., the robot correctly reads all available destination tags), which simplifies the implementation and allows for isolated evaluation of the destination-sign localization approach.

4.5 Visualization and Logging

A dynamic Matplotlib-based visualization shows the robot and particles moving along the road network. The visualization includes markers for detected destination tags and the robot's current path. A logging module tracks step-by-step information for analysis and debugging purposes.

5. Experiments and Results

5.1 Experiment Setup

To evaluate the effectiveness of the map-based localization, eight experiment scenarios were conducted. These experiments were designed to assess two key factors affecting localization performance:

1. Measurement capabilities:
   Standard mode where only destination name and distance information are used.
   Enhanced mode where destination, distance and the road number information are taken into consideration.
   (Note: The reader can refer to Figure 1)

2. Particle filter parameters:
   Four distinct particle counts of 200, 500, 1000 and 2000 (cf. Table 1)

3. Controlled Variables:
   Identical environment.
   Same platform used for all testing.

All experiments share these conditions: robot follows a fixed road network path from Tallinn to Tartu. Experiments with „Enhanced“ in their name indicate that the experiment had road number detection enabled meaning that if the destination sign had also road number information then both would be taken into consideration.

Table 1. Experimentation information

Experiment Name	Measurement Type	Particle count
Standard 1	Destination data	200
Enhanced 1	Destination data + road number	200
Standard 2	Destination data	500
Enhanced 2	Destination data + road number	500
Standard 3	Destination data	1000

Enhanced 3	Destination data + road number	1000
Standard 4	Destination data	2000
Enhanced 4	Destination data + road number	2000

5.2 Experiments

5.2.1 Localization without road numbers

On Figure 2 “Step: 40” indicates current progress of the experiment, where robot has moved 40 times with given movement distance, which is 50m in these experiments. The blue dot or circle represents the robot’s true position (i.e. the ground truth) and Active constraints indicate the latest distance sign read. As can be seen on Figure 3, where blue colored dot is outside of Tallinn on the main highway, last destination sign that the robot observed indicated that its 86km away from Paide and 179km away from Tartu, red text and red circle indicates cities. The +-6km show the cutoff range of distance measurements, if particle is 100km away from Paide at that point in time then its weight gets dramatically reduced.

Experiment - Standard 1, with 200 particles. As can be observed from Figures 2 - 4 the particles gather around the robot’s true position when constraint is observed.

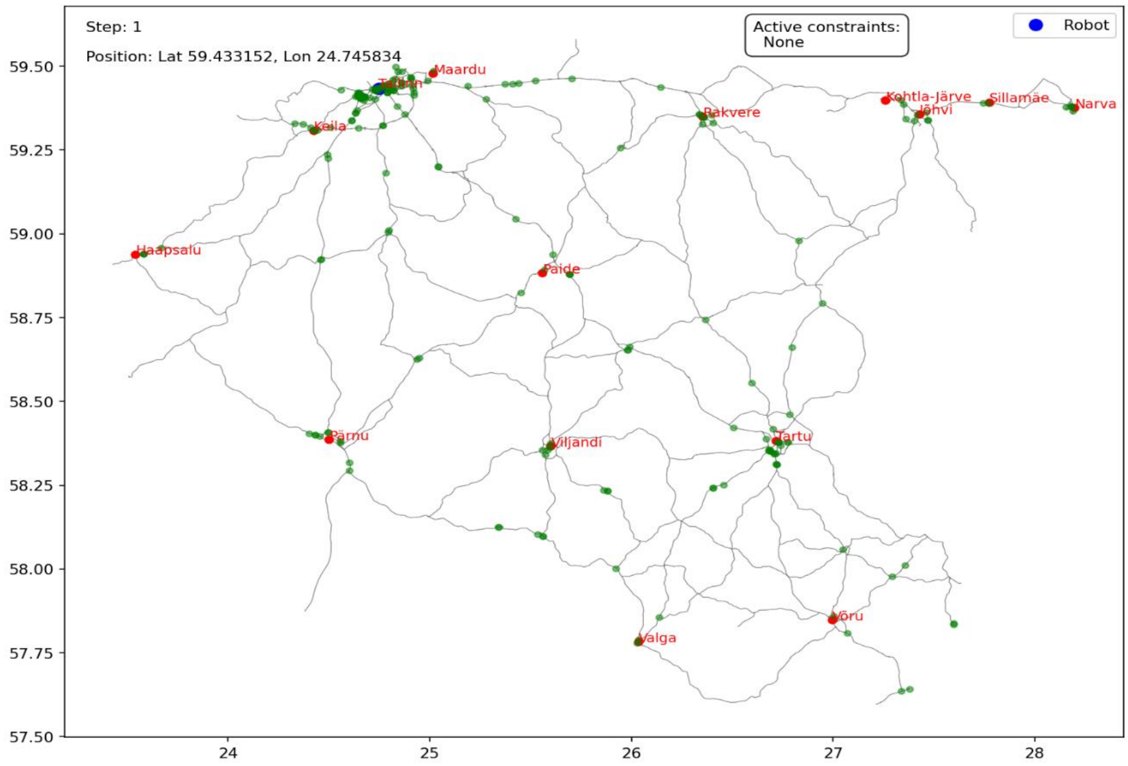

Figure 2. Standard 1, experiment with 200 particles, step 1

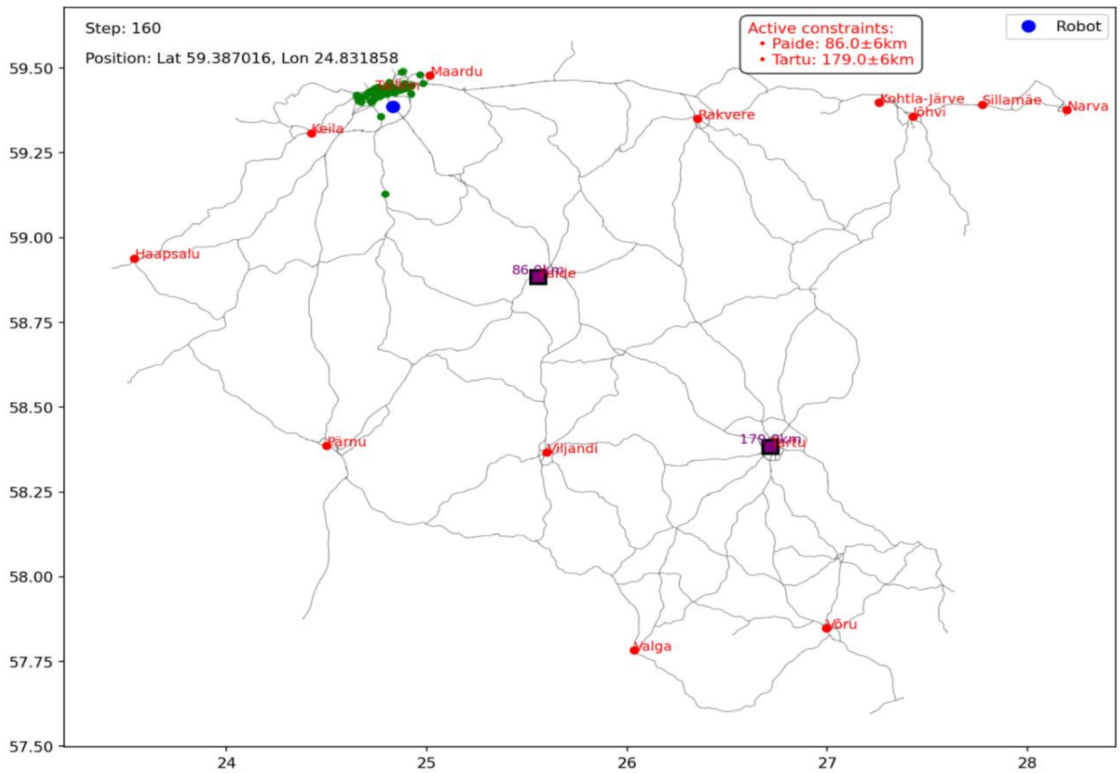

Figure 3. Standard 1, experiment with 200 particles, step 160

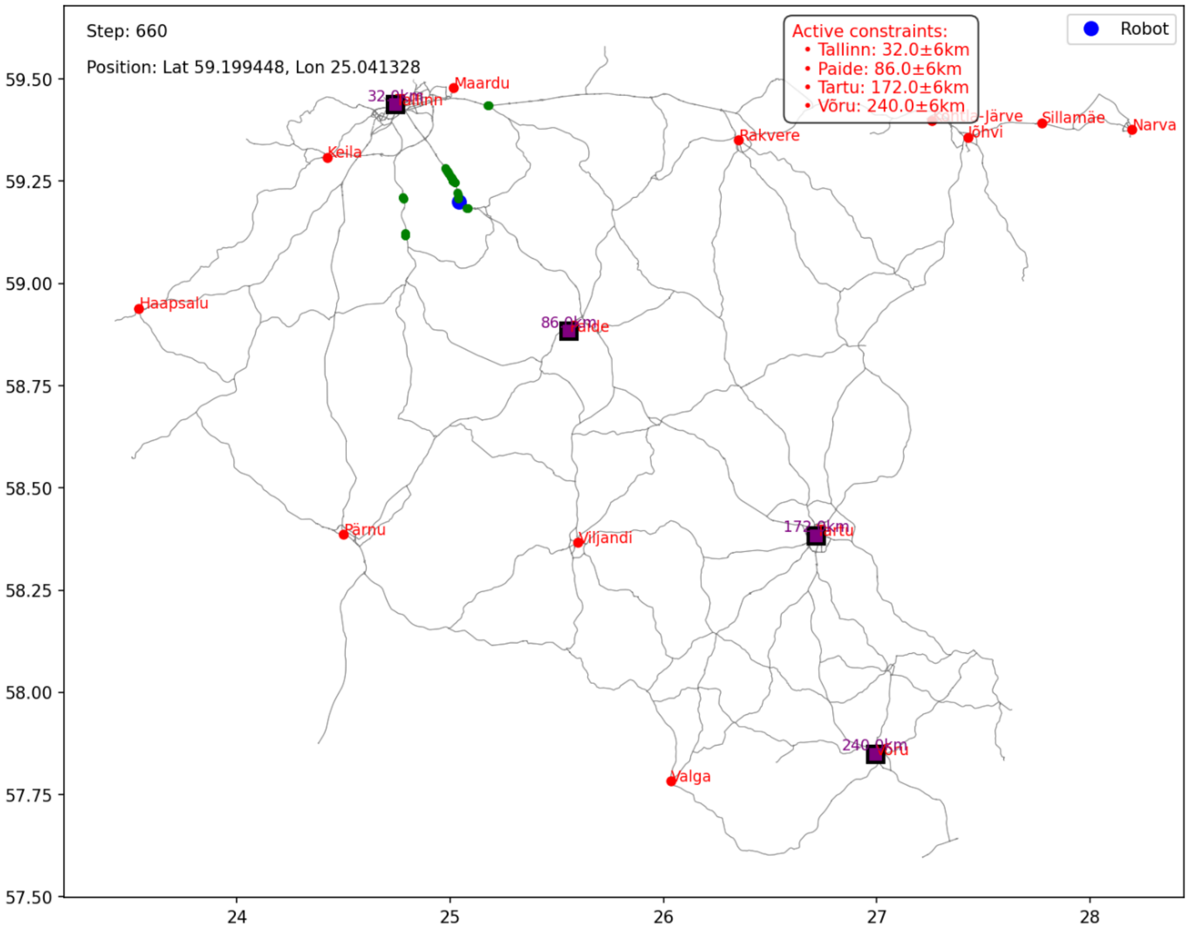

Figure 4. Standard 1, experiment with 200 particles, step 660

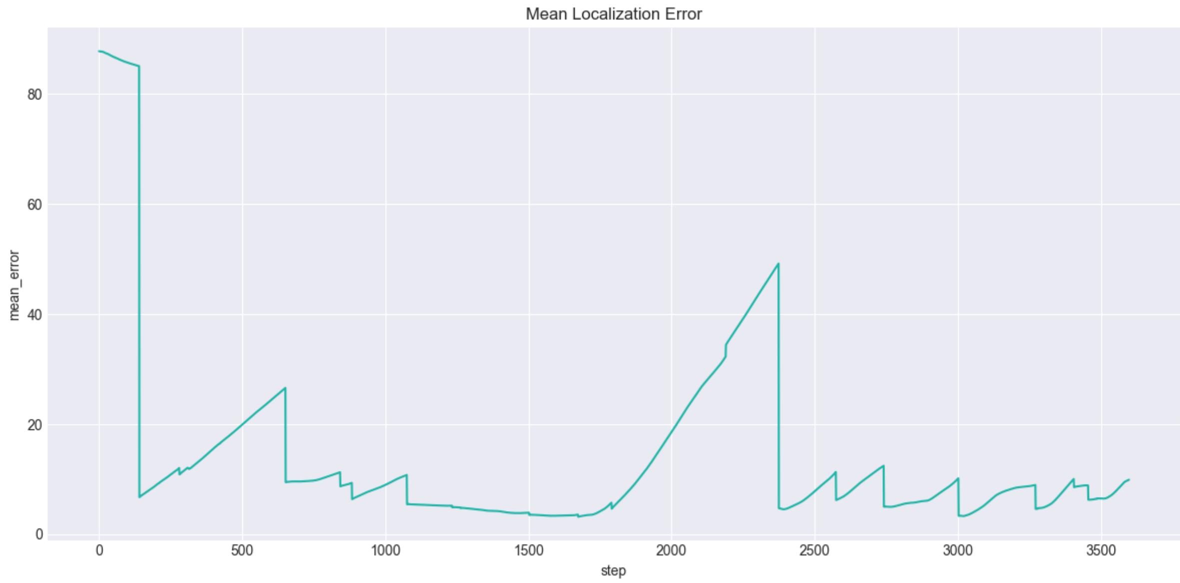

Figure 5. Standard 1, experiment with 200 particles, Localization error compared to step

On Figure 5, the mean error shows the estimated positioning error mean and step indicates the stage at which the estimation occurred. As can be observed on Figure 5, the error rate spike around areas where there are junctions or long distances without destination signs.

Experiment - Standard 2, with 500 particles.

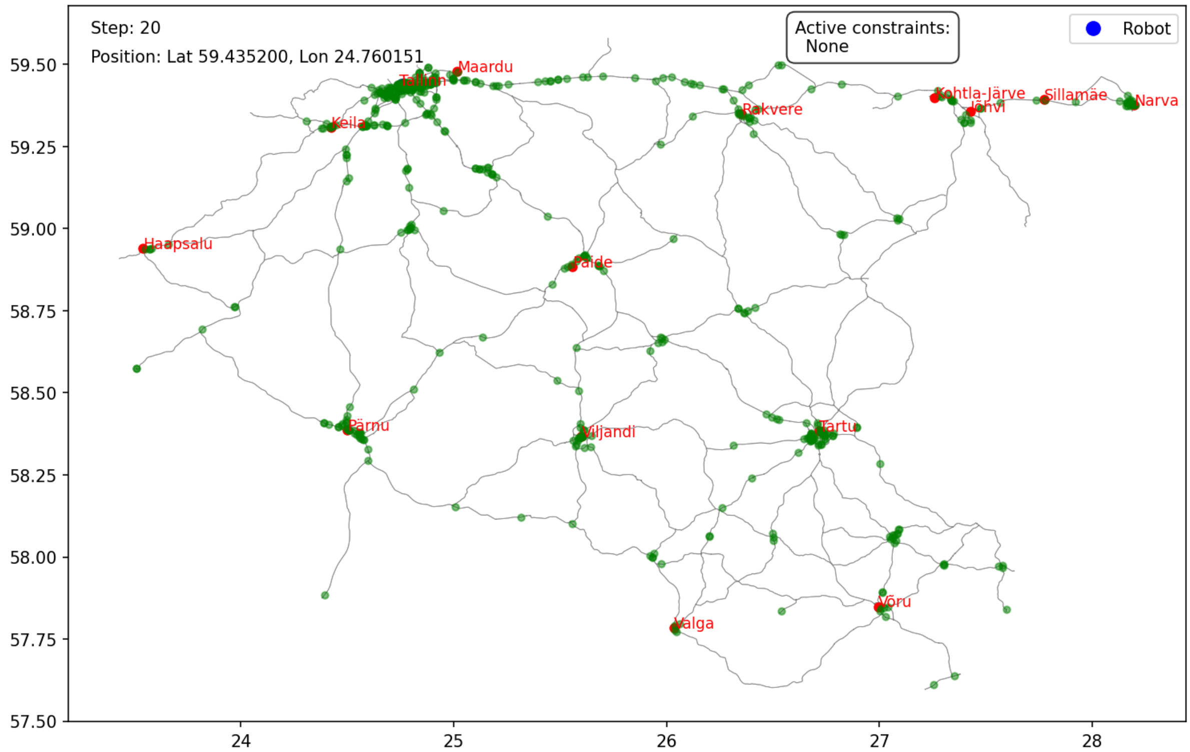

Figure 6. Standard 2, experiment with 500 particles, step 20

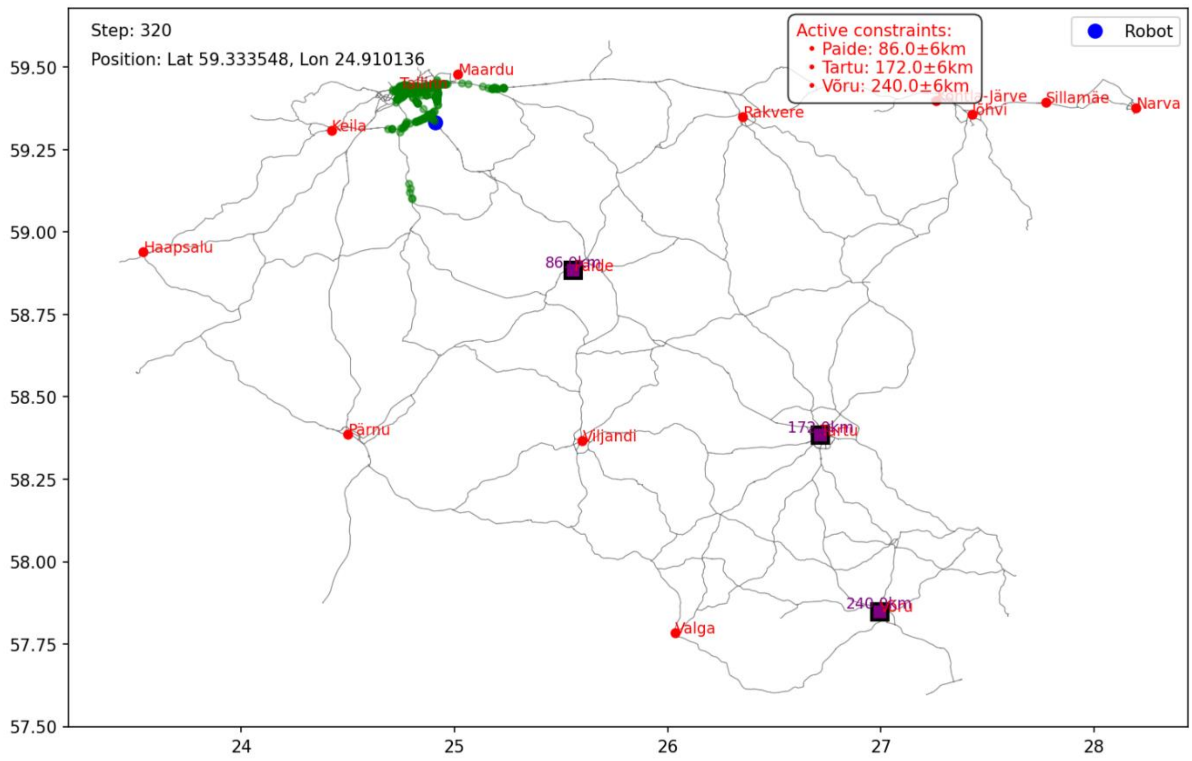

Figure 7. Standard 2, experiment with 500 particles, step 320

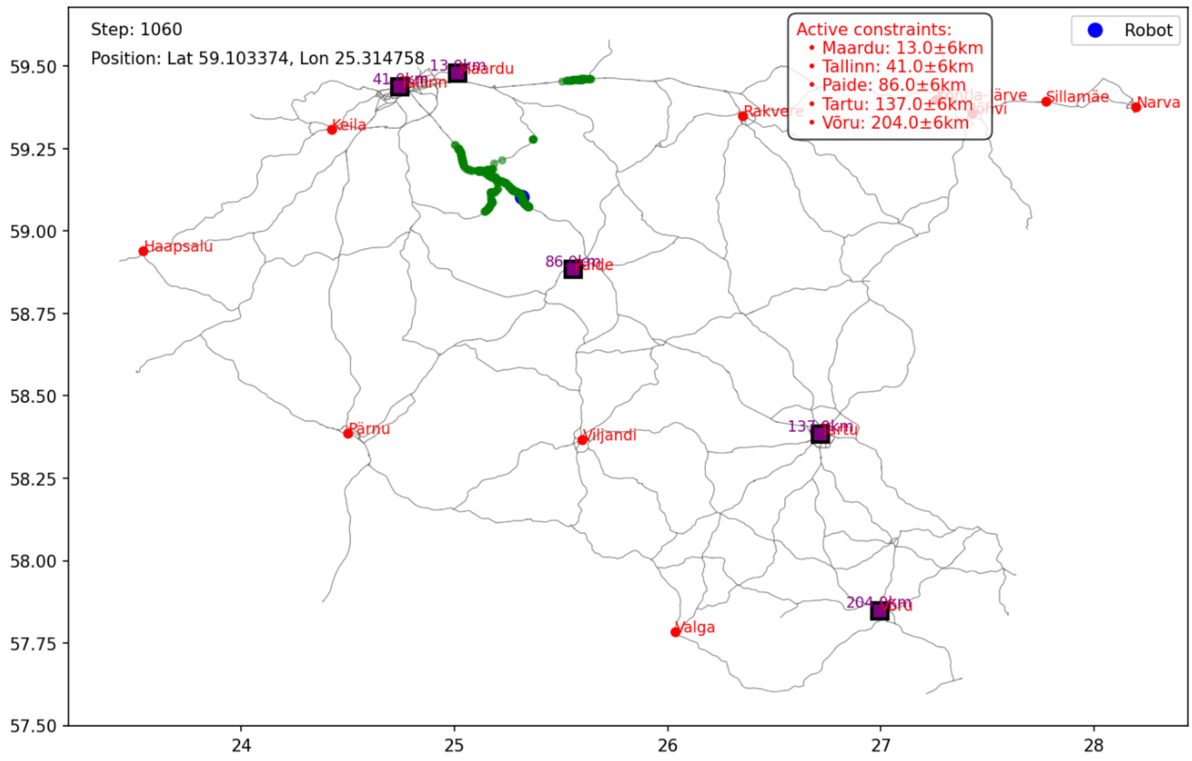

Figure 8. Standard 2, experiment with 500 particles, step 1060

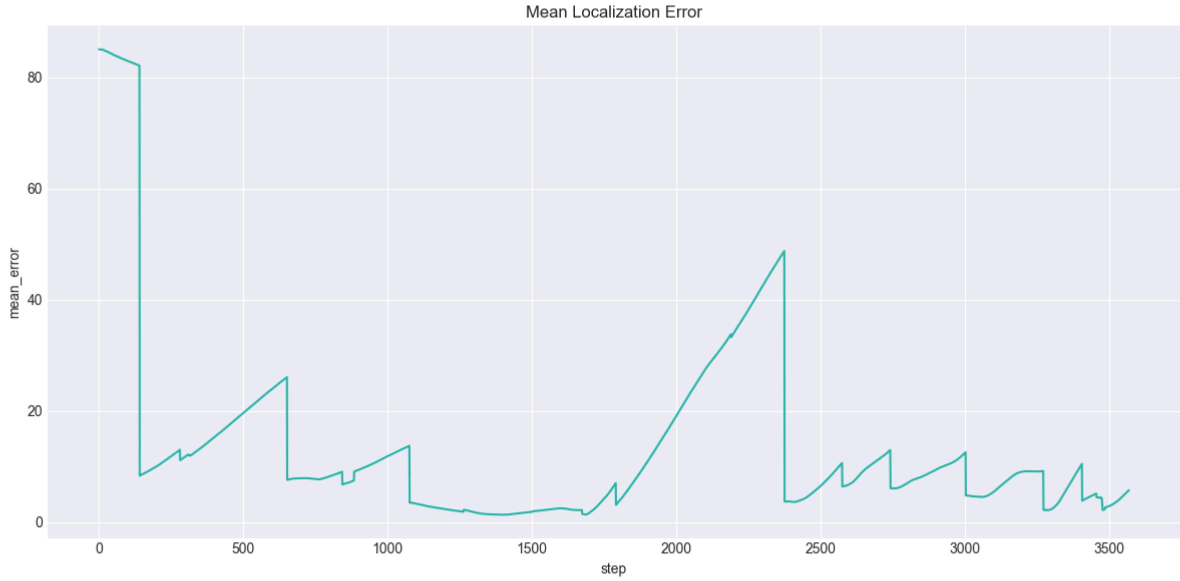

Figure 9. Standard 2, experiment with 500 particles, Localization error compared to step

Experiment - Standard 3, with 1000 particles.

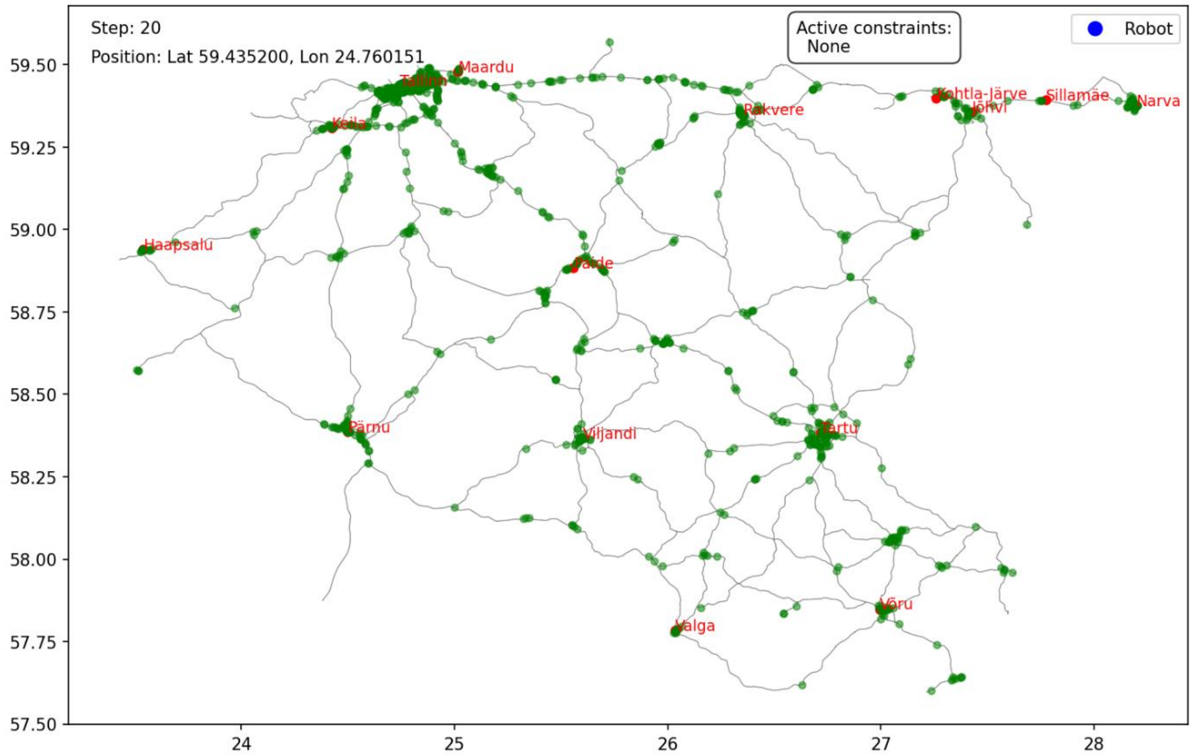

Figure 10. Standard 3, experiment with 1000, step 20

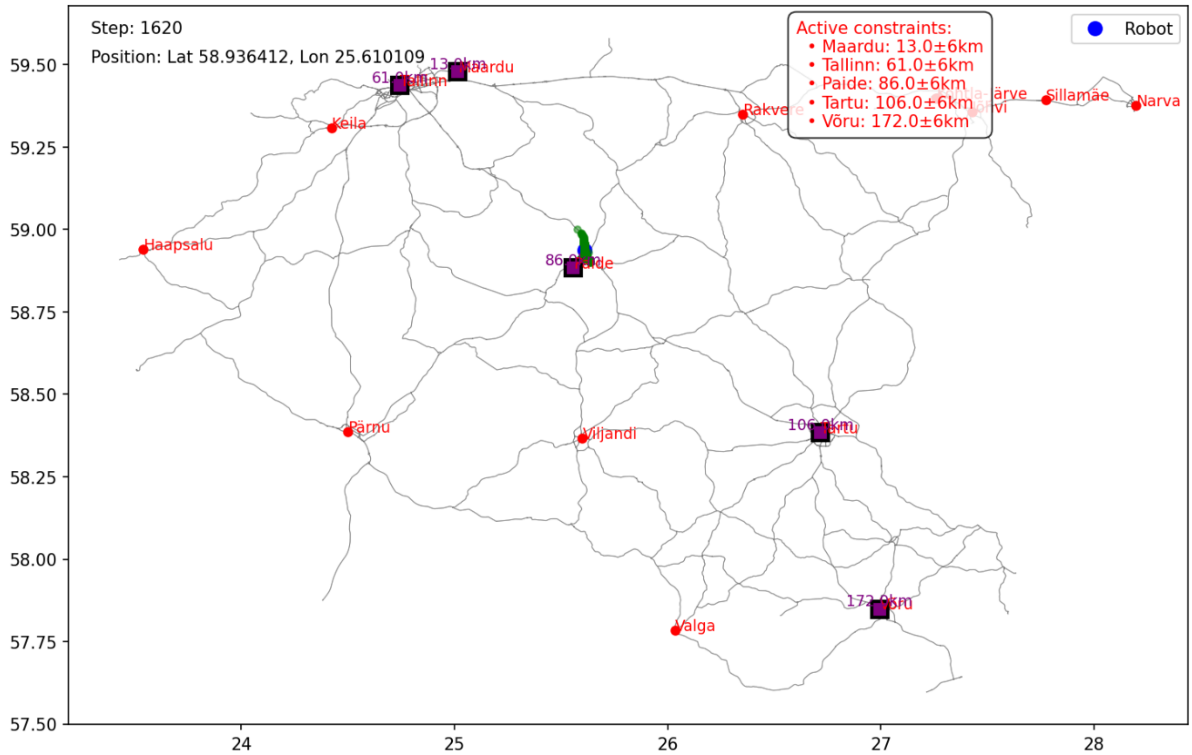

Figure 11. Standard 3, experiment with 1000, step 1620

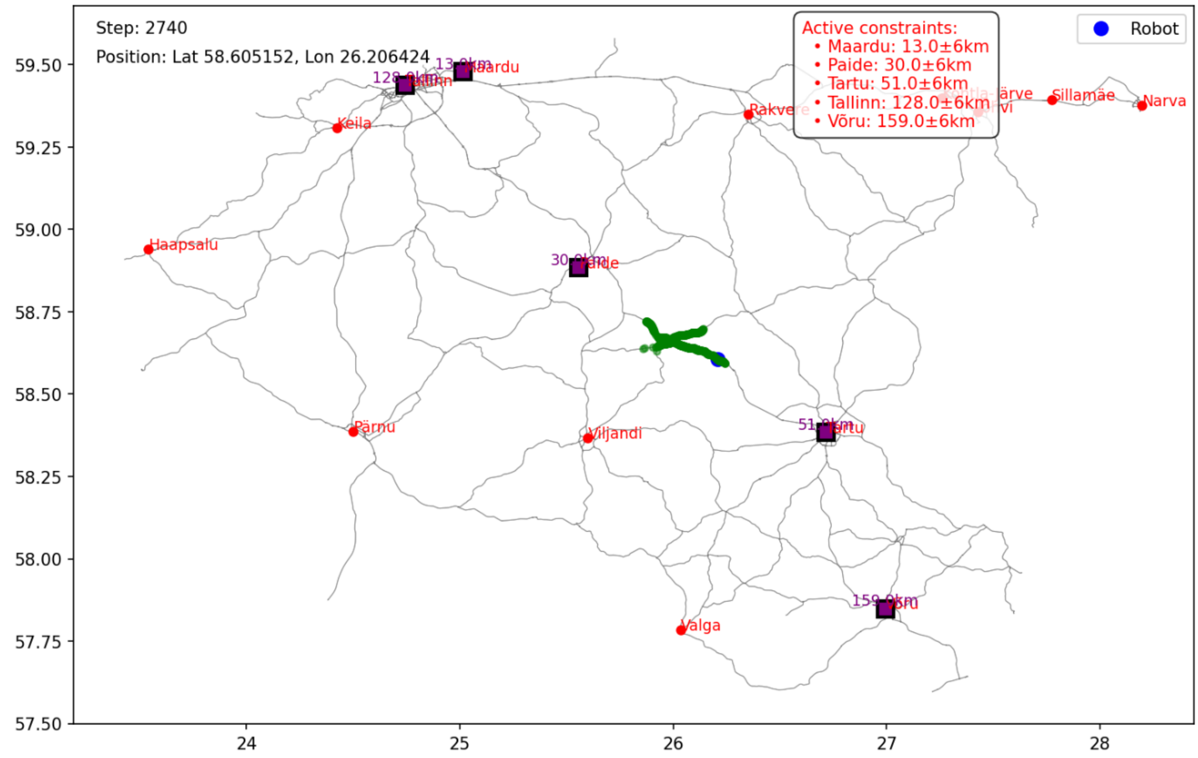

Figure 12. Standard 3, experiment with 1000, step 2740

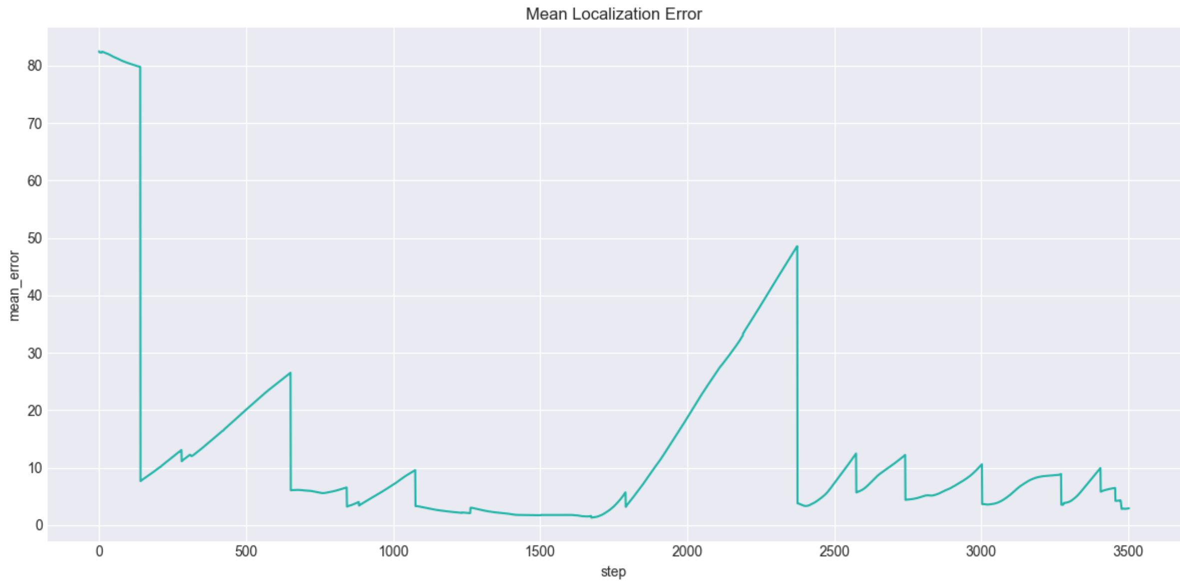

Figure 13. Standard 3, experiment with 1000 particles, Localization error compared to step

Experiment - Standard 4, with 2000 particles.

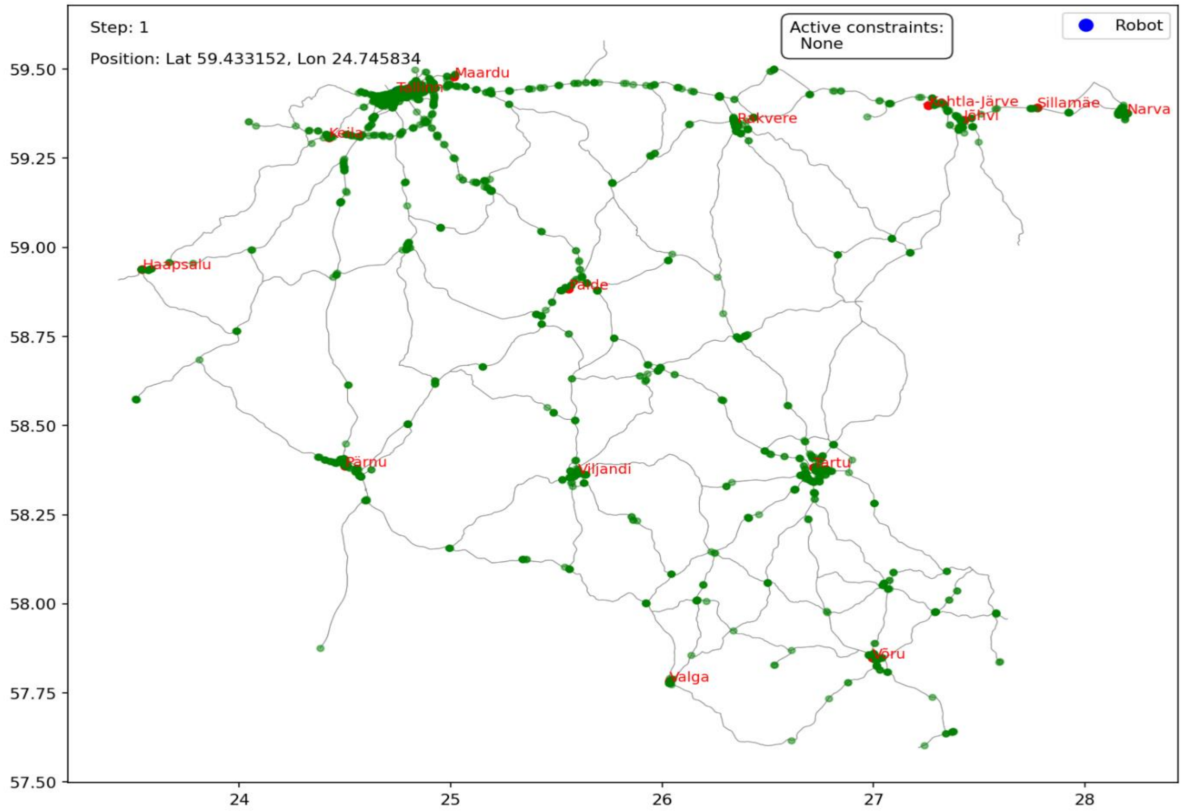

Figure 14. Standard 4, experiment with 2000, step 1

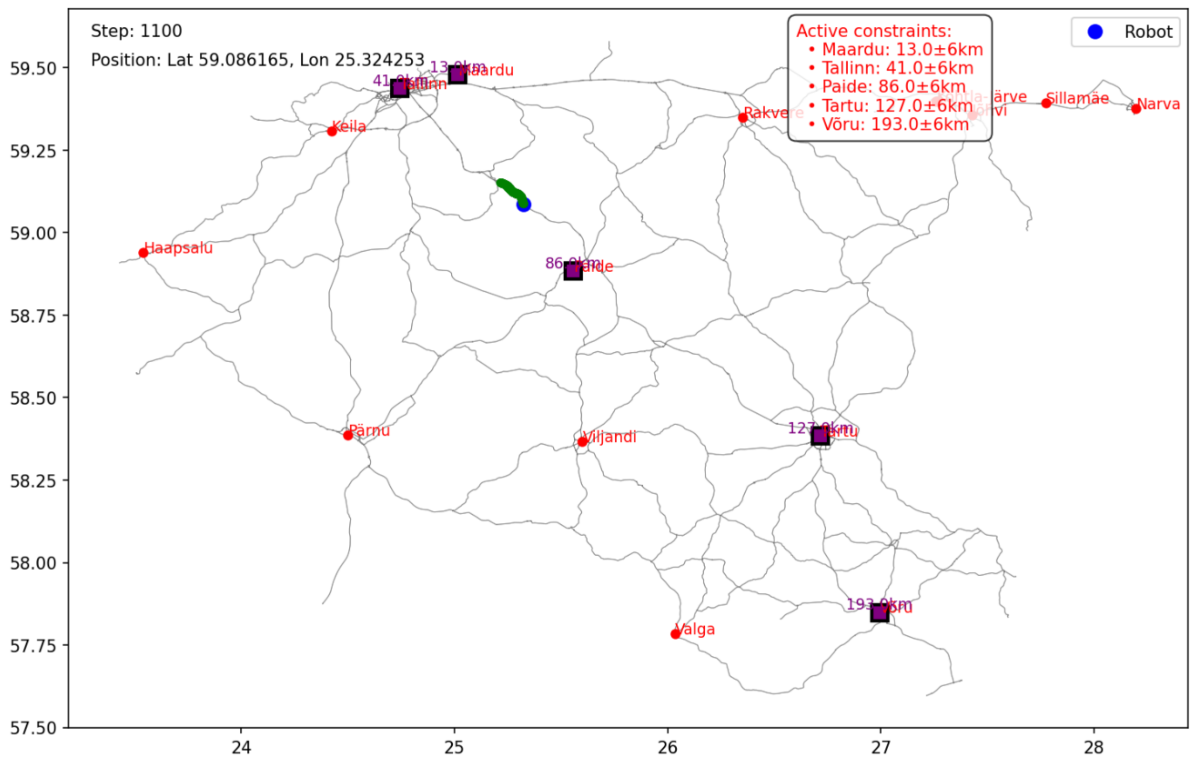

Figure 15. Standard 4, experiment with 2000, step 1100

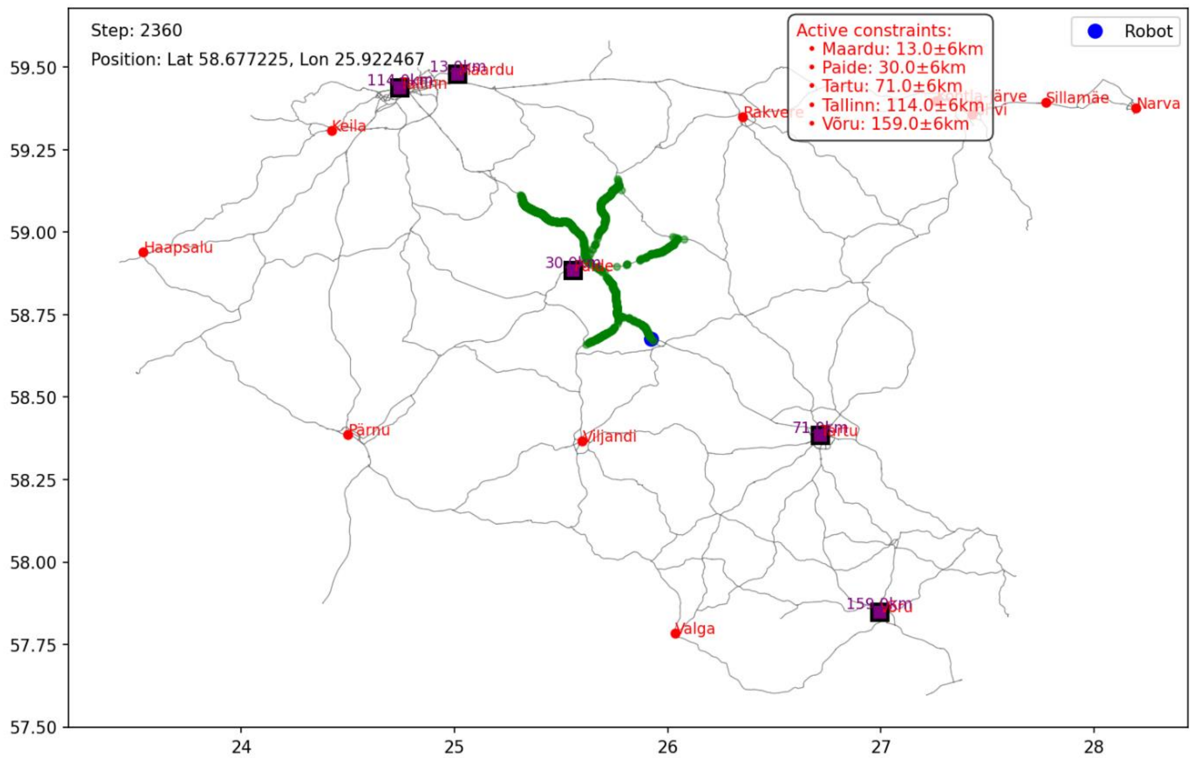

Figure 16. Standard 4, experiment with 2000, step 2360

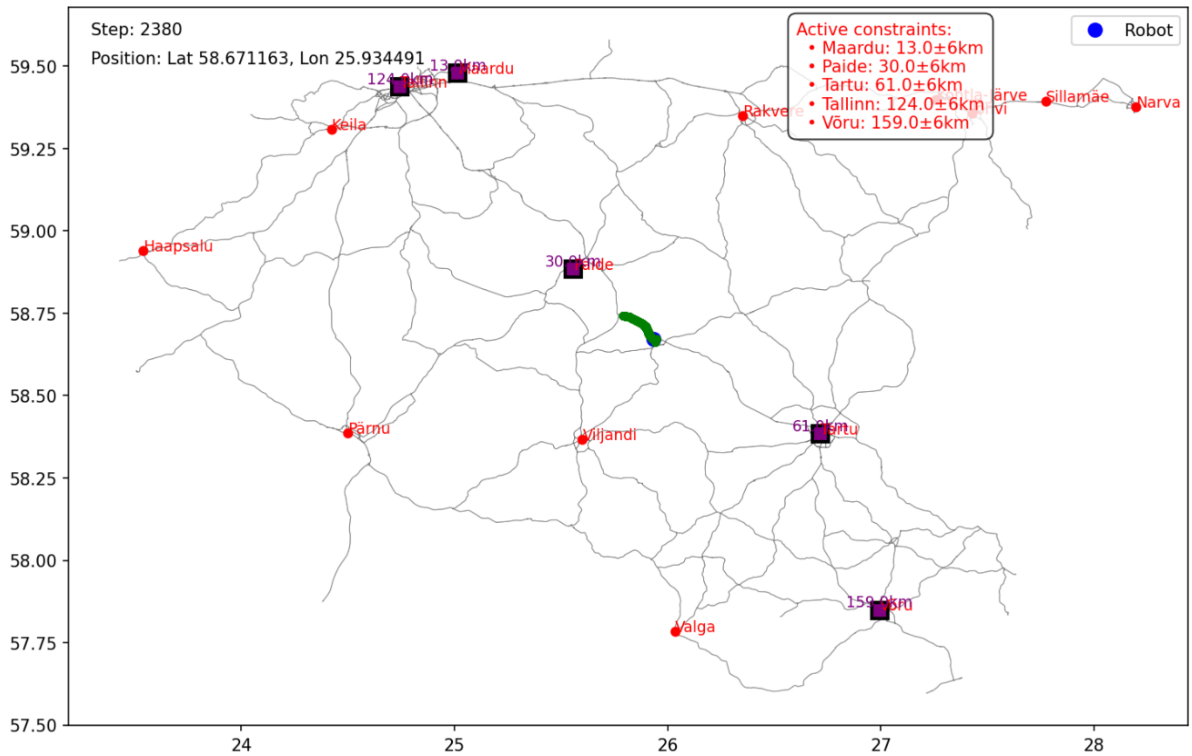

Figure 17. Standard 4, experiment with 2000, step 2380

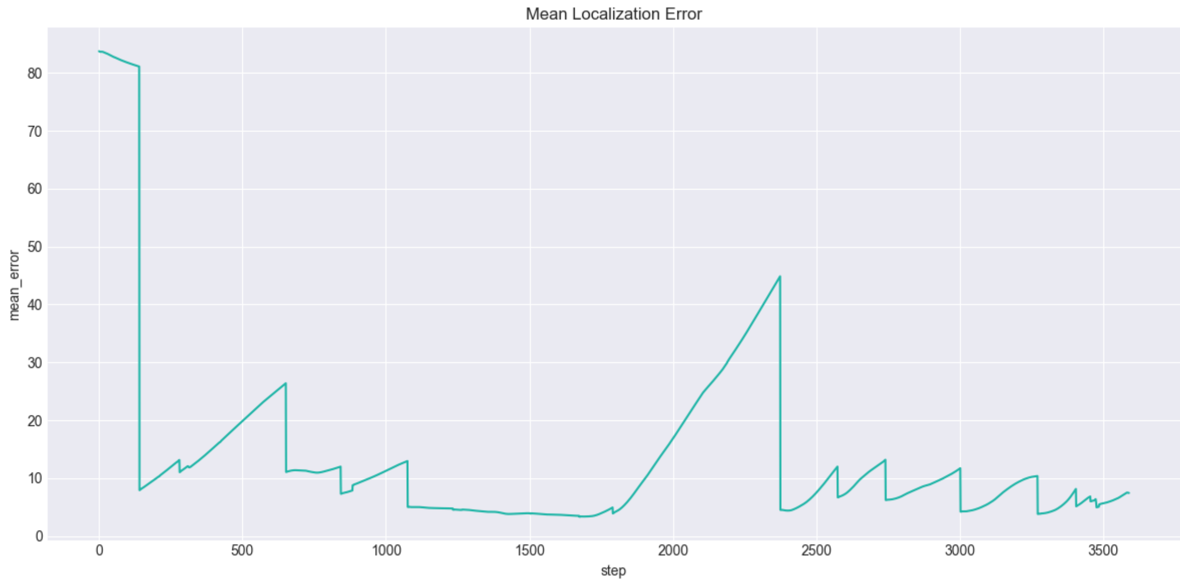

Figure 18. Standard 4, experiment with 2000 particles, Localization error compared to step

5.2.2 Localization with road numbers

These experiments were simulated with road numbers enabled. Green represents particles, blue robot's true position, red indicates cities. In the top left of particle simulation graph Active constraints can be seen, it shows the last seen constraints.

Experiment - Enhanced 1, with 200 particles

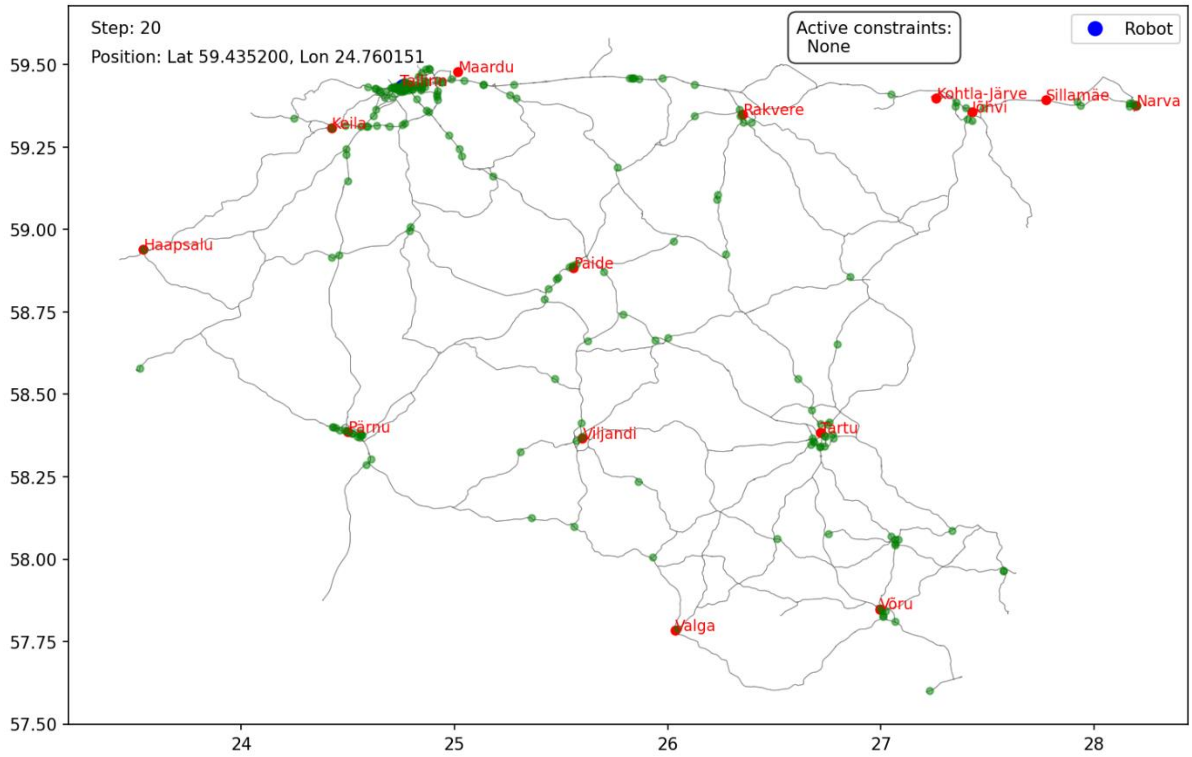

Figure 19. Enhanced 1, experiment with 200, step 20

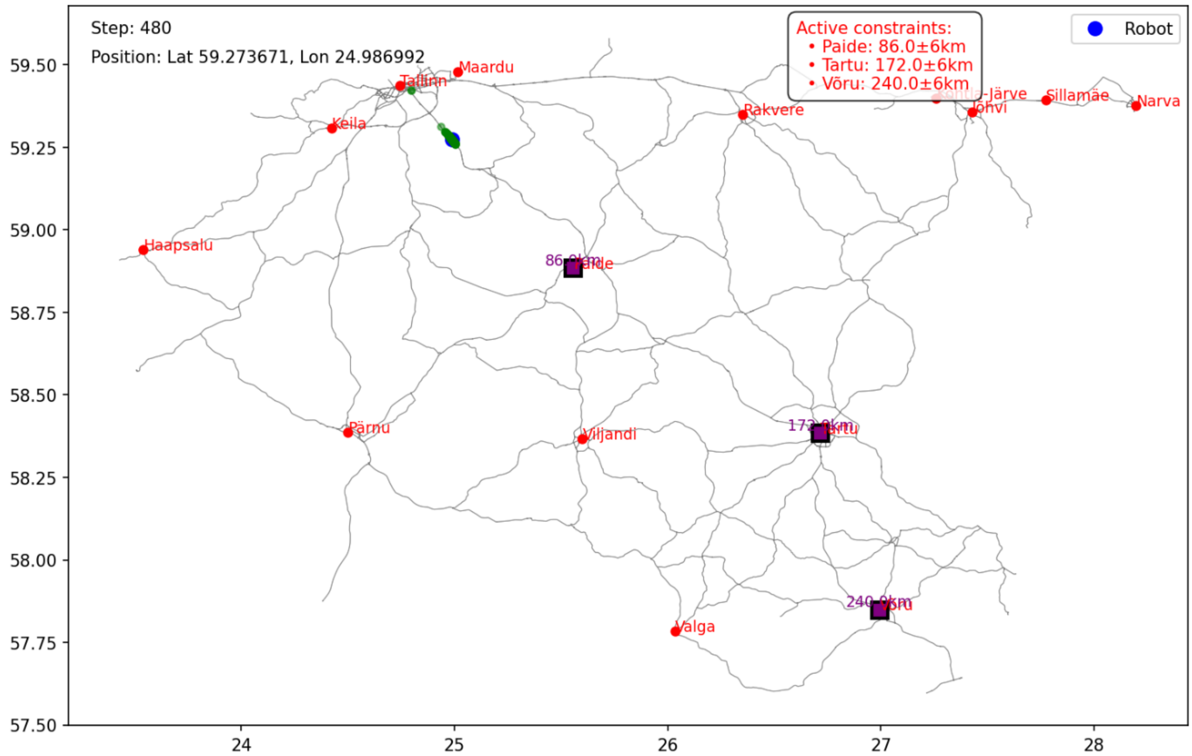

Figure 20. Enhanced 1, experiment with 200, step 480

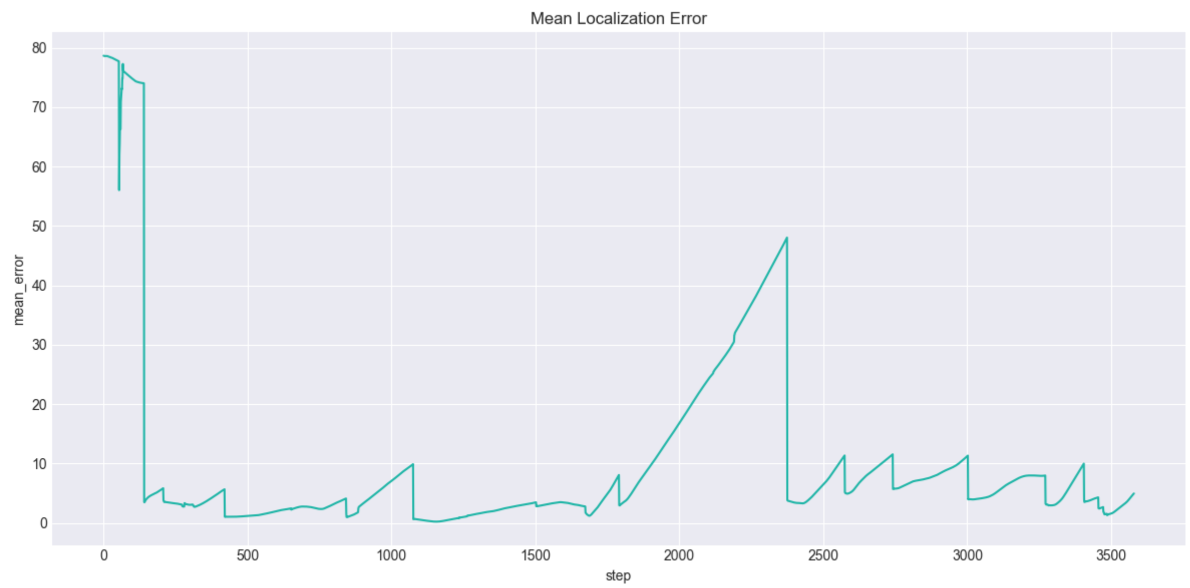

Figure 21. Enhanced 1, experiment with 200 particles, Localization error compared to step

Experiment - Enhanced 2, with 500 particles

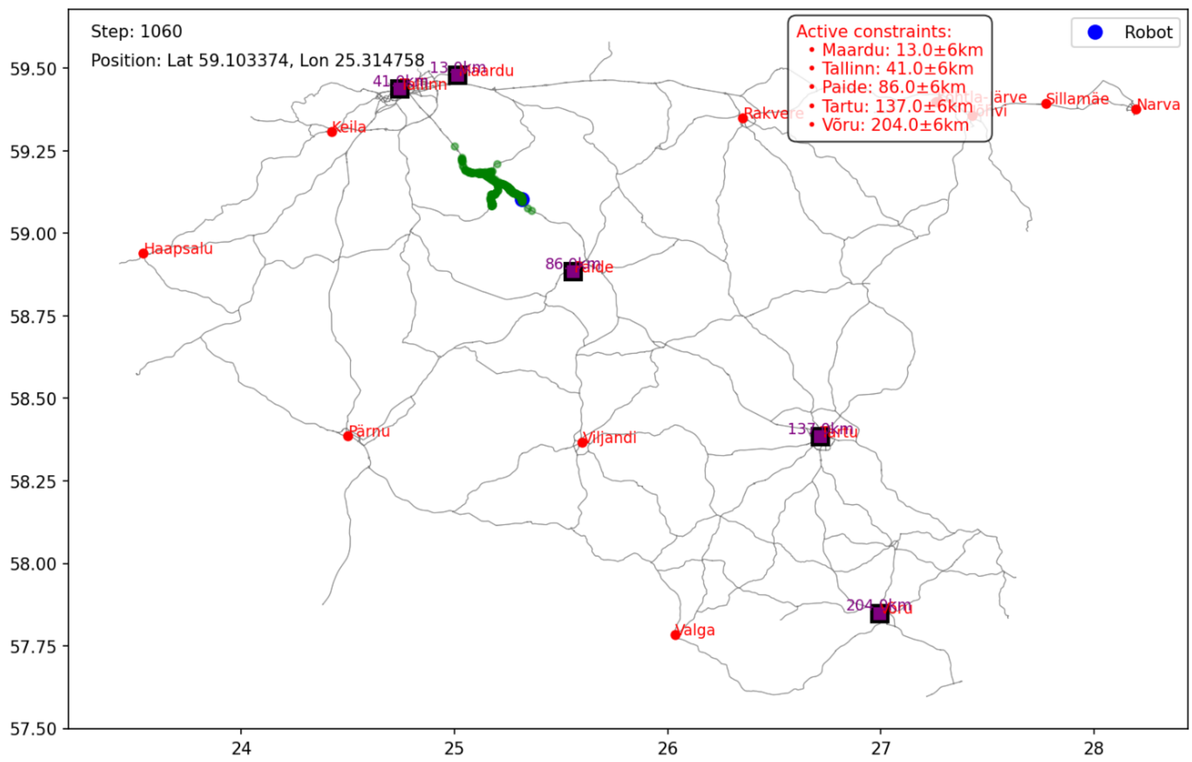

Figure 22. Enhanced 2, experiment with 500, step 1060

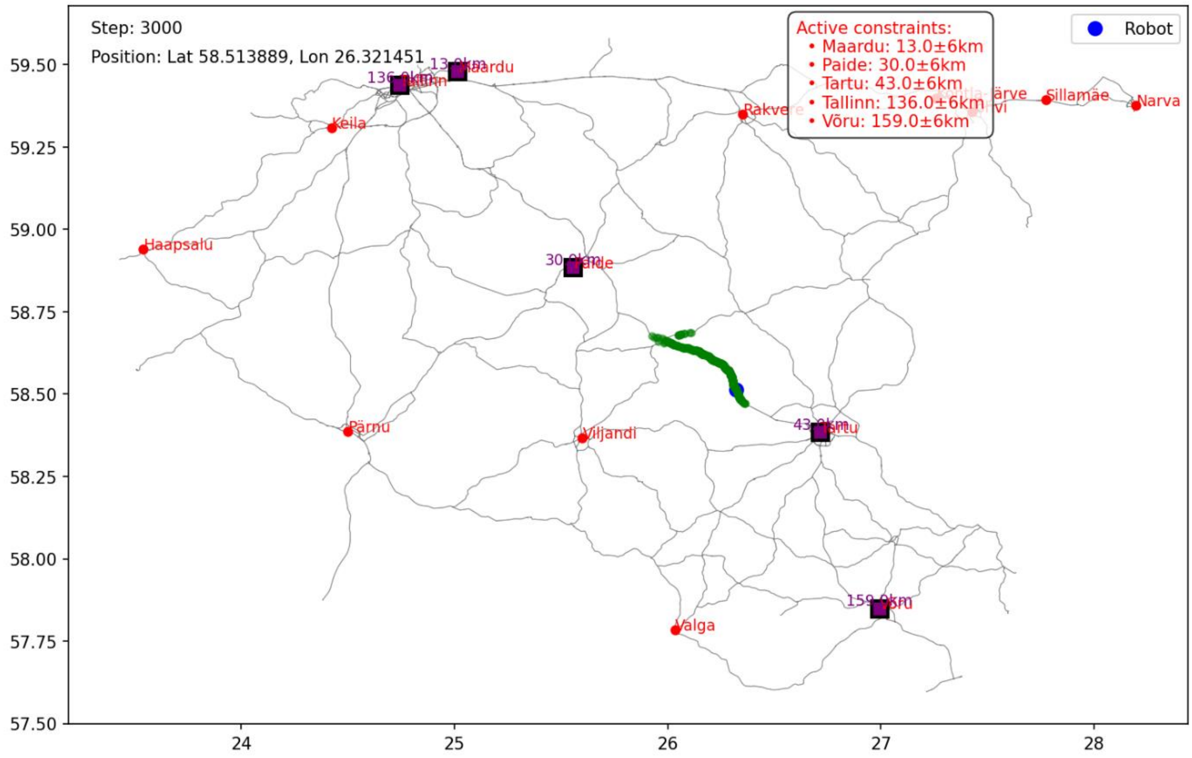

Figure 23. Enhanced 2, experiment with 500, step 3000

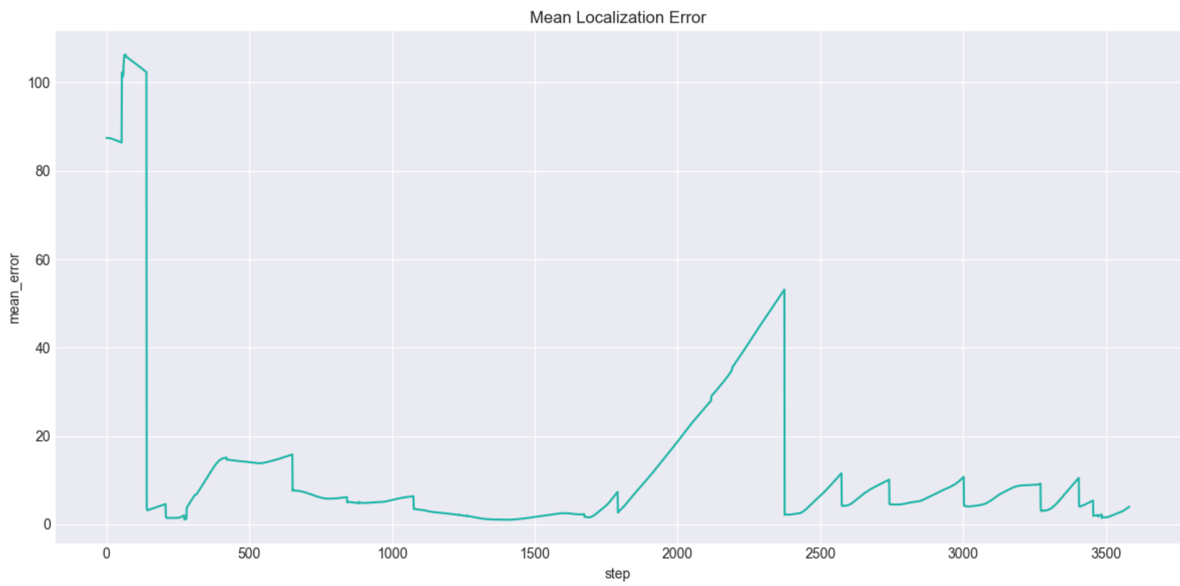

Figure 24. Enhanced 2, experiment with 500 particles, Localization error compared to step

Experiment - Enhanced 3, with 1000 particles

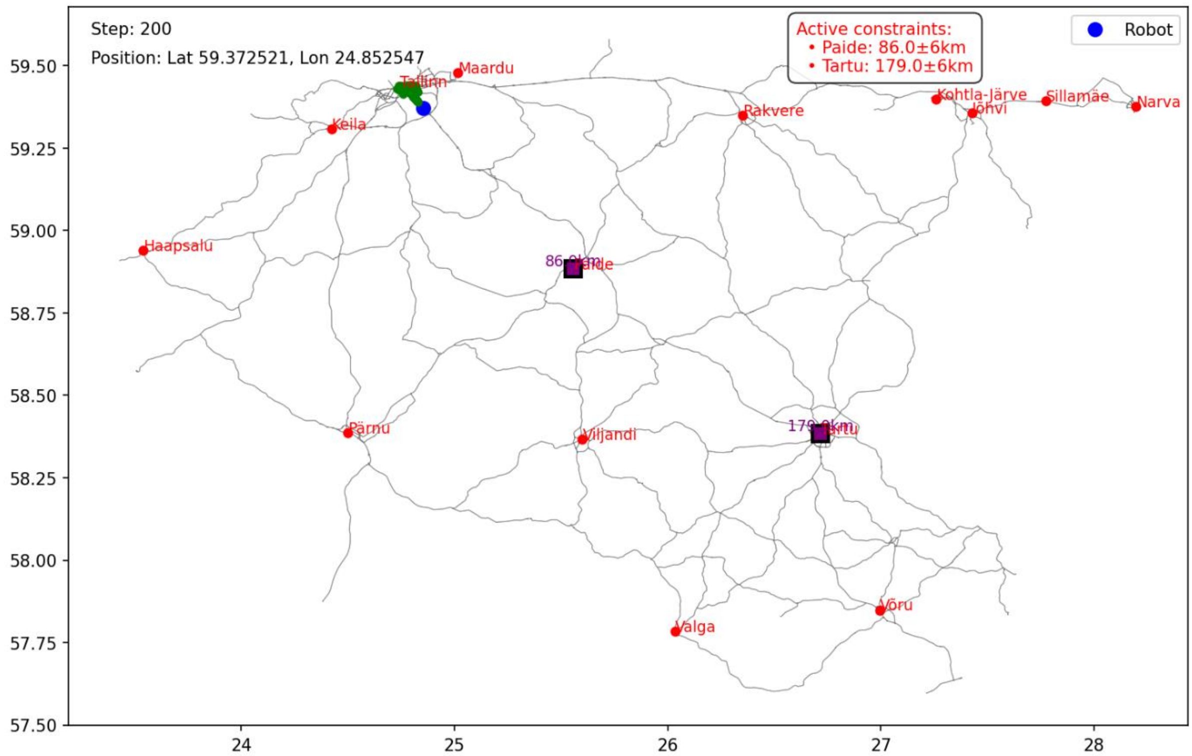

Figure 25. Enhanced 3, experiment with 1000, step 200

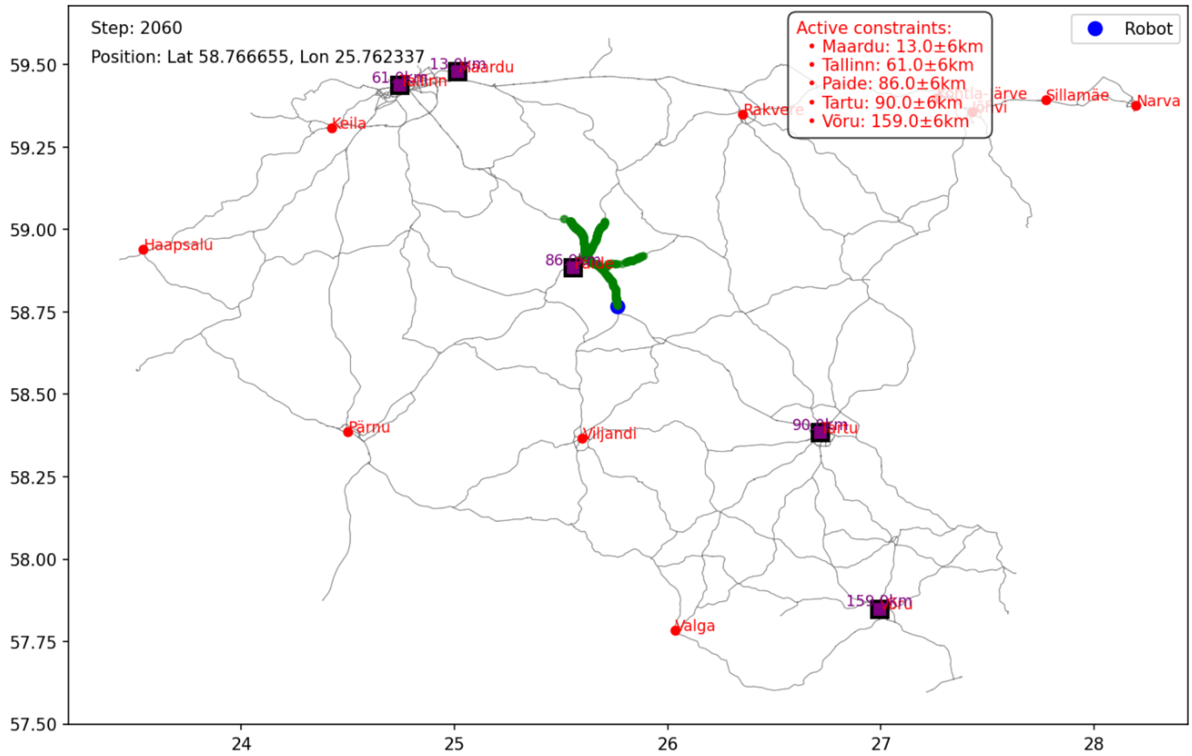

Figure 26. Enhanced 3, experiment with 1000, step 2060

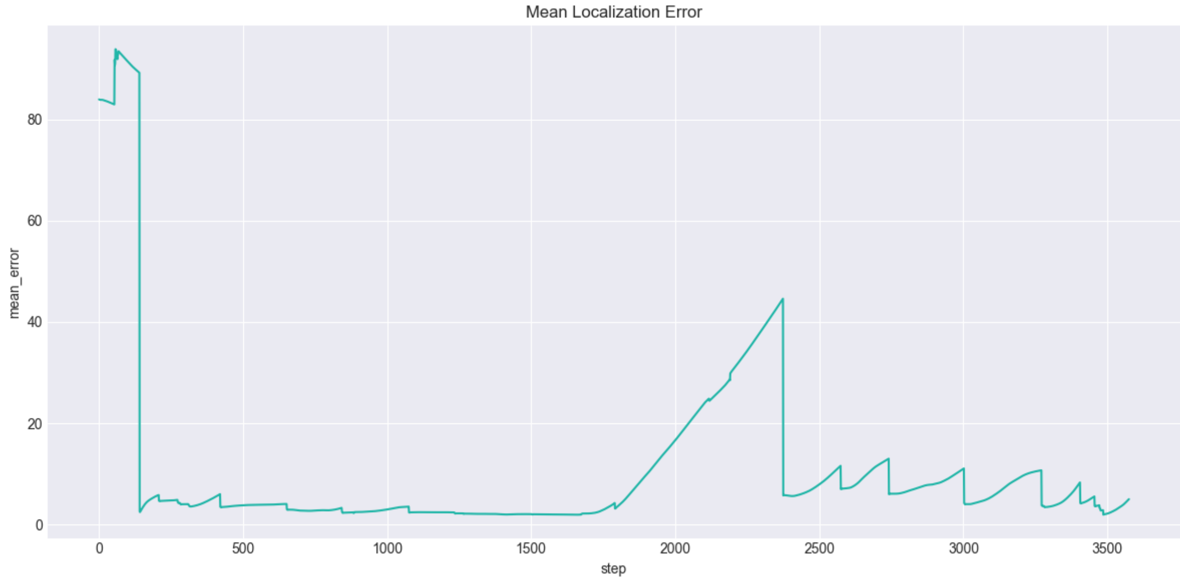

Figure 27. Enhanced 3, experiment with 1000 particles, Localization error compared to step

Experiment - Enhanced 4, with 2000 particles

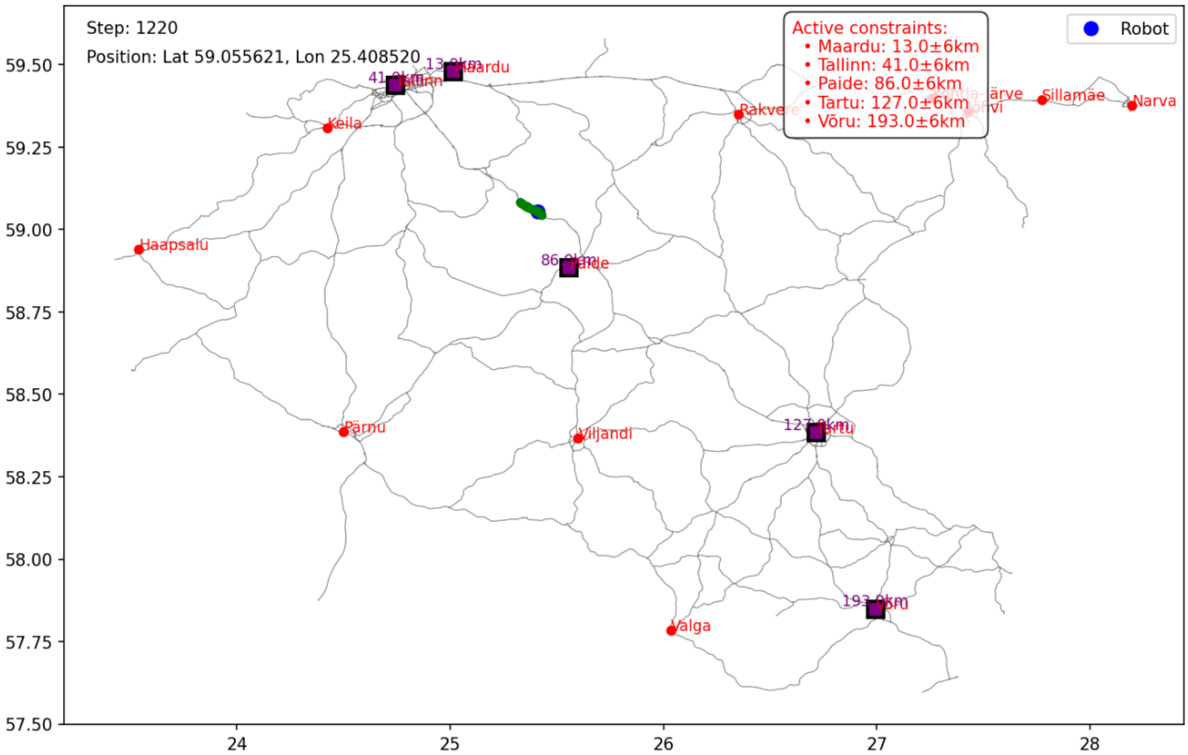

Figure 28. Enhanced 4, experiment with 2000, step 1220

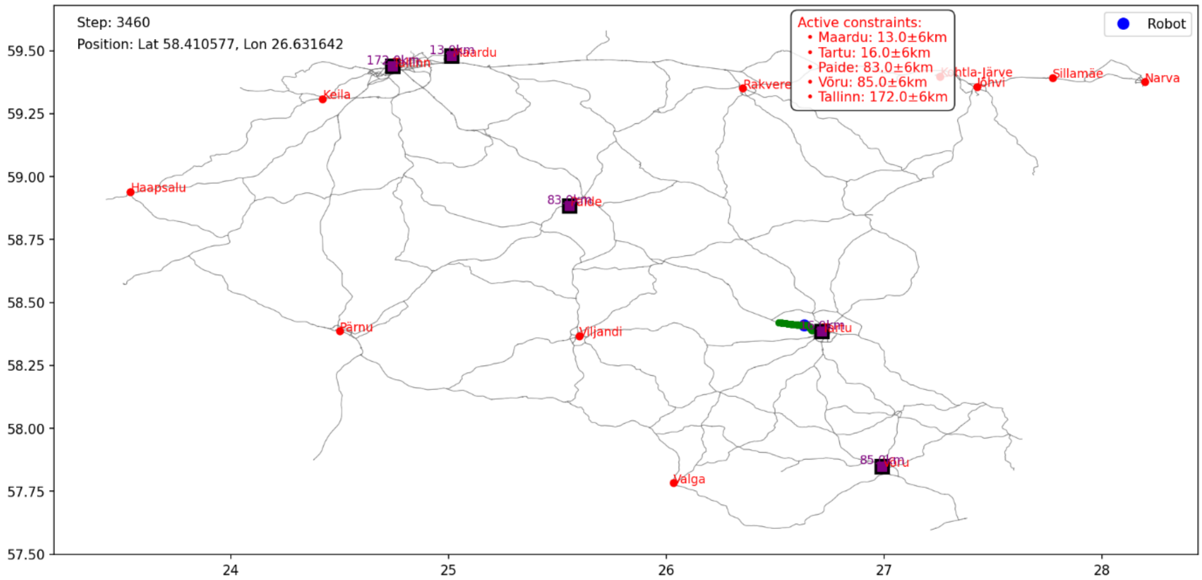

Figure 29. Enhanced 4, experiment with 2000, step 3460

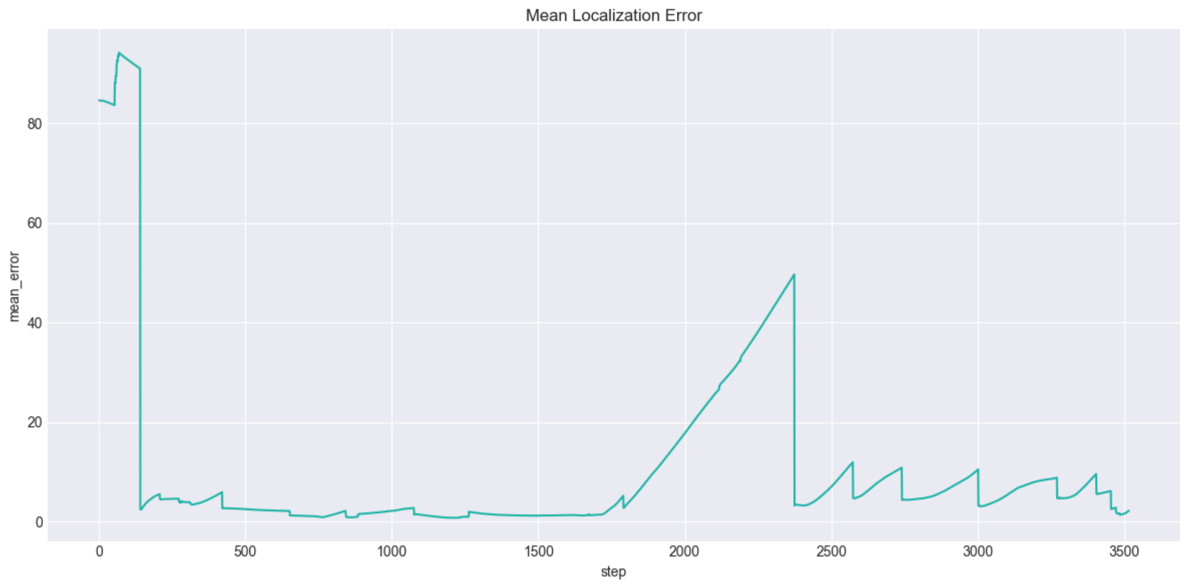

Figure 30. Enhanced 4, experiment with 2000 particles, Localization error compared to step

Table 2. Average and Peak Accuracy precetage for each Standard experiment

Standard	200 particles	500 particles	1000 particles	2000 particles
Peak	96.40%	98.50%	98.40%	96.00%
Average	83.80%	83.90%	84.00%	83.40%

Table 3. Average and Peak Accuracy precetage for each Enhanced experiment

Enhanced	200 particles	500 particles	1000 particles	2000 particles
Peak	99.70%	99.00%	97.90%	99.20%
Average	86.90%	88.10%	88.40%	88.60%

The accuracy metrics in Table 2 and Table 3 quantify the particle filter's performance by measuring the proximity of particles to the robot's ground truth position at each simulation step. Peak accuracy represents the highest achieved localization precision, reflecting moments where the particle cloud converged tightly around the true robot position (e.g., 99.70% in the enhanced method, Table 3). In contrast, average accuracy indicates the filter's consistency over time, with lower values (e.g., 83.80% for standard experiments, Table 2) revealing periods of particle dispersion or tracking drift.

6. Discussion

All simulations successfully achieved high Peak accuracy (Table 2, Table 3) if there were sufficient amount of destination signs, despite the variation in particle count or constraint type. However, higher particle counts showed better accuracy (Table 2, Table 3) and smoother error curve (Figure 18, Figure 30). Also, using road numbers to further corroborate localization accuracy bringing down the error curve about 1 to 5-points (Table 2, Table 3). The further apart the destination signs were, the worse the mean error became (Figure 30), in one section, between the steps 2000 and 2400 of all Localization mean error graphs (Figure 5, Figure 9, Figure 13, Figure 18, Figure 21, Figure 24, Figure 27, Figure 30), large increase in mean error can be observed, in that section of the road there were no destination signs. Also, if there were junctions in the road after the destination sign then the mean error increased as can be seen from all of the mean error graphs in steps 2500-3500, where mean error raises after every encounter with a destination sign.

6.1 Accuracy Trends

The simulation results reveal clear patterns in localization performance: Low particle counts (e.g., 200) produced more erratic behaviour, particularly in constraint-sparse regions, due to insufficient sampling diversity (Figure 8). Higher particle counts led to improved results, allowing the system to maintain higher accuracy (Figure 5, Figure 18) and (Table 2, Table 3), but as can be observed from mean error graph graphs, accuracy lowers during empty road segments with junctions like steps 1900 - 2400. The addition of road number information increased the semantic richness of destination tags, providing stronger observational constraints and boosting overall reliability but suffered same problems when put into a region with no semantic clues. Destination tags proved highly effective as localization constraints. Their impact was especially visible at close intervals, near junctions or city limits, where the semantic context from signs reduced the search space for possible locations. The inclusion of road numbers acted as an additional disambiguation layer, near junctions where robot could have been at multible possible locaitons, enabling the filter to eliminate more improbable hypotheses earlier.

6.2 Limitations of the Experiments

This program has several notable limitations. Performance issues arise with large graphs, distance calculations, while resampling process becomes slow when particle count of the experiment reaches into the thousands, one experiment may take more than several hours to complete. It relies heavily on the data quality and has been modified with hardcoded destinaiton signs. Implementation issues include bad error handling, inefficient memory management and slow visualization process. Functionality is also rather limited, the robot navigates a predetermind route and distance calculation are constantly wrong due to limited road network.

7. Conclusion

This thesis presented a prototype of vision-based localization framework that utilizes simulated destination tags and road-data from OpenStreetMap. The experiments confirm that such semantic cues do provide reliable localization without any reliance on GNSS if there are enough destination signs, performance suffered in sign-sparce regions.

7.1 Summary of Contributions

This research demonstrates a proof-of-concept for using vision-based destination sign sensing for vehicle localization. A particle filter was adapted to use virtual destination signs, and the system's performance was evaluated with real-time visualizations.

7.2 Final Thoughts

This thesis presents an approach to localization using semantic cues from the road network. By leveraging structured map data and semantic information from common roadside elements, autonomous systems can localize with human-like reasoning. This approach has potential to enhance navigation safety and autonomy in complex, signal-challenged environments.

Future improvements could involve integrating real image detection for road signs, utilizing other map features such as buildings or speed limits, and extending the system to real robot simulators. Other improvements include optimization for larger and more accurate models.

References

[1] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, 2010, pp. 4372-4378, doi: 10.1109/ROBOT.2010.5509700.

[2] Jesse Levinson, Michael Montemerlo, Sebastian Thrun, “Map-Based Precision Vehicle Localization in Urban Environments”, Proceedings of Robotics: Science and Systems, 2007.

[3] Philipp Ruchti, Bastian Steder, Michael Ruhnke, Wolfram Burgard, “Localization on OpenStreetMap Data using a 3D Laser Scanner”, Proceedings of International Conference on Robotics and Automation (ICRA), 2015.

[4] I. A. Barsan, S. Wang, A. Pokrovsky, and R. Urtasun, “Learning to localize using a lidar intensity map,” arXiv preprint arXiv:2012.10902, 2020.

[5] X. Wei, I. A. Barsan, S. Wang, J. Martinez, and R. Urtasun, “Learning to localize through compressed binary maps,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 10316–10324.

[6] M. Juurmaa, K. Nurmela, and A. Kaseorg, “Particle filter-based self-localization using map constraints and sparse perceptual cues,” 2023 IEEE International Conference on Robot & Human Interactive Communication (RO-MAN), Busan, South Korea. DOI: 10.1109/RO-MAN56745.2023.10287946.

[7] Y.-H. Huang and C.-T. Lin, "Indoor localization method for a mobile robot using LiDAR and a dual AprilTag," Electronics, vol. 12, no. 4, pp. 1023, 2023, doi: 10.3390/electronics12041023.

[8] OpenStreetMap contributors, ”Map of Estonia,” OpenStreetMap, 2025. [Online]. Available: **https://www.openstreetmap.org/#map=8/58.613/25.024, Map of Estonia, 2025**

License

Non-exclusive licence to reproduce the thesis and make the thesis public

I, Henri Poolakese ,
(author´s name)

grant the University of Tartu a free permit (non-exclusive licence) to

reproduce, for the purpose of preservation, including for adding to the digital archives of the University of Tartu until the expiry of the term of copyright, my thesis

GNSS-free localization using milestone-board information ,
(title of thesis)

supervised by Naveed Muhammad ;
(supervisor´s name)

2. grant the University of Tartu a permit to make the thesis specified in point 1 available to the public via the web environment of the University of Tartu, including via the digital archives, under the Creative Commons licence CC BY NC ND 4.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright;

3. am aware of the fact that the author retains the rights specified in points 1 and 2;

4. confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Henri Poolakese

12/08/2025