# APPLICATION OF PARTICLE FILTERS TO MAP-MATCHING ALGORITHM

**Pavel Davidson[1], Jussi Collin[2], and Jarmo Takala[3]**
Department of Computer Systems, Tampere University of Technology, Finland
e-mail: pavel.davidson@tut.fi

## Abstract

**Key words:** land vehicle navigation, map-matching, sequential Monte Carlo method, particle filtering

*This paper presents the numerical probabilistic approach to map-matching problem within the framework of Bayesian theory. The proposed solution is based on sequential Monte Carlo method, so called particle filtering. This algorithm can be adapted for implementation on real-time portable car navigation systems equipped with GPS or dead reckoning sensors. The algorithm reliability and accuracy performance was investigated using simulated data and data from real-world driving tests in urban environment.*

## Introduction

Digital road network map is an important component of any car navigation system. Using map as a medium, complex information can be easily communicated to a driver. However, maps can be used not only for display purposes: map database is a source of valuable information that can be used to improve the accuracy of the position given by GPS or integrated GPS/DR navigation system. The process of improving navigation through more accurate positioning with the help of a map is called map-matching. The goal of map-matching is to exploit prior information contained in road networks. However, incorporating road map information within the conventional Kalman filtering framework is not an easy task. The reason is that road maps represent a constraint, which leads to highly non-Gaussian posterior densities that are difficult to represent accurately using the conventional techniques.

Existing map-matching algorithms can be classified as (a) Semi-deterministic approach including geometric and topological algorithms [1], (b) Probabilistic approach [2],[3], and (c) Fuzzy-logic and belief theory based approaches [8],[9]. The semi-deterministic algorithms are similar to the algorithm proposed by French [1]. The basic requirement of this algorithm is that vehicle is moving along predefined known route. Various conditional tests can be performed to determine whether the vehicle is travelling on the known road network. Existing semi-deterministic algorithms work well in situations in which the errors of position measurements do not exceed 10-15 m. This can be, e.g., the case of GPS receivers under open skies, or in environment with low multipath. However, if position errors become large, then improvements in existing map-matching algorithms may be necessary.

The probabilistic approach gives the most reliable solution to map-matching problem compared to other methods. It also overcomes the disadvantage of semi-deterministic methods: assumption that vehicle is moving on predefined route. The probabilistic algorithm calculates probabilities of vehicle traveling on different road segments to select a correct road segment and then estimates vehicle position on the selected road link. This approach differs from semi-deterministic approach as it does not perform any explicit map-matching step, and has advantage in both robustness and flexibility. Different versions of this algorithm were proposed in Dmitriev *et al*. [2] and Scott [3]. They both acknowledged that correct road segment identification is a key component of any map-aided estimator because the performance derived from the map-matching algorithm can be misleading if the vehicle location is projected to an incorrect road. Dmitriev *et al.* [2] proposed a mathematical framework for solving the map-matching problem based

[1]MSc, Researcher
[2]PhD, Senior Researcher
[3]Professor

on the recursive Bayesian estimation and non-linear filtering theory. They also acknowledged that during the turn a posteriori distribution of the vehicle position on the road is non-Gaussian and non-linear filtering methods are required to solve this problem.

Existing map-matching algorithms are not capable of satisfying the requirements of all intelligent transportation system (ITS) applications and services. For instance, bus priority service at road junctions requires a positioning accuracy of 5m (95%) [10]. None of the existing algorithms can meet this positioning requirement, especially, in dense urban areas. Car charging telematics applications require very reliable positioning. Tests made in London and Den Haag, using only GPS-based receivers, showed tolling errors varying from 1% to 10% [11]. The goal of this work is to improve position accuracy and reliability of the car navigation system by using digital road maps in addition to the main navigation system.

In this paper, we propose a numerical approach to a map-matching problem. The proposed solution is based on recursive implementation of Monte-Carlo based statistical signal processing known also as particle filtering. The basic principle is to use random samples (also referred to as particles) to represent the posterior density of the car position in a dynamic state estimation framework where road map information is used. Since particle filters have no restrictions on the type of models and noise distribution the velocity and heading measurement errors can be modeled accurately.

The advantages of a particle filter for this particular application is that it provides a natural way for road map information to be incorporated into vehicle position estimation and its ability to capture multi-modal distributions which tend to occur when there is uncertainty in which road the user is on. By considering multiple candidate roads, the particle filter is able to quickly adapt if an initial guess at the proper road is found to be incorrect.

This paper presents both simulation results and results from real-world data collection in a city environment. Both GPS and non-GPS position measurements were collected and simulated. These measurements were combined with OpenStreetMap (a freely-available map database) to calculate the position of the vehicle on the road as it drove through a city. A precision DGPS position solution was used as the reference in order to evaluate the accuracy of the particle-filter based solution. The results shown in this paper demonstrate that the proposed particle filter approach is reliable and accurate. It is able to correct large (about 50 m) errors in position by applying the map constraints. It is also demonstrated that the particle filter-based map-matching algorithm is robust to errors in the predetermined map.

**Digital maps**

A digital map is created by converting a paper map into a vector-encoded structure. Road network can be represented by its features expressed as vectors using Cartesian geometry. A feature is denoted as an existing item in the real world. The digitized road network typically represents the road data using line segments whose endpoints (nodes) and shapes are defined in terms of latitude and longitude. According to Zhao [4] nodes, segments, and shape points can be defined as:

- A node is a cross point or an end point of a street and is used to represent an intersection or a dead-end of a road.
- A segment is a piece of roadway between two nodes and is used to represent fragments of roadways and other features.
- Shape points are ordered collections of points, which map the curved portion of a given segment to a series of consecutive straight-line pieces. Road of any curvature can be approximated by a sequence of straight lines.

In this paper, we used a freely available map database OpenStreetMap [12]. The geographical information stored in navigable road maps (e.g., maps from OpenStreetMap) is usually expressed in geographic coordinates. As proposed in [7] only a limited area (called “road cache”) around the estimated location can be considered. After the road cache extraction, the points of the polylines that describe the roads are converted into a local tangent East-North-Up (ENU) frame. Then, by choosing a reference point, the transformation between ECEF (WGS84Earth-Centered, Earth-Fixed) and ENU is computed. Since the elevation is usually not available in a navigable map, we convert the map points in the working frame by supposing that they are all located at the ellipsoidal height. As the ENU frame is attached to a road cache, it should be noted that the working frame is temporary and valid only for small regions.

**Algorithm for solving the problem**

The objective of map-matching is to recursively estimate position of the vehicle from a set of measurements. The state vector consists of vehicle's North and East coordinates: $x_k = \left[P_k^N \quad P_k^E\right]^T$. Here the subscript $k$ corresponds to the time instant $t_k$. It is assumed that the vehicle is moving on the roads, which are known from the digital database. As proposed in [2] the roads can be described by an implicit nonlinear function $\rho^h(x)$ in the form of

$$R^h = \{x : \rho^h(x) = 0\},\ h = 1, M \cdot \tag{1}$$

For the purpose of map-aided estimation, the road network can be approximated by a set of road segments $R_{k,k+1}$, each of which is a straight line between the nodes $\xi_k, \xi_{k+1}$ that satisfy the equation (1).

It is assumed that the state can be described by partially observable discrete-time Markov chains. Furthermore, the state $x_k$ depends on the previous state $x_{k-1}$ according to the probabilistic law $p(x_k \mid x_{k-1})$. This problem can be stated as estimation of the sequence of states $x_{0:k} = \{x_0, \cdots, x_k\}$ given the series of observations $y_{1:k} = \{y_1, \cdots, y_k\}$ subject to the motion model $p(x_k \mid x_{k-1})$, measurement model $p(y_k \mid x_k)$ and constraints on the state vector given in form of the road network. The prior probability at $t_0$, $p(x_0)$ is assumed to be known. The goal is to find the "best" trajectory in terms of minimum mean-squire error (MMSE) criteria.

This problem can be solved within the framework of Bayesian estimation theory [2]. According to the Bayesian view, the posterior probability density function (pdf) $p(x_{0:k}|y_{1:k})$ contains all the statistical information available about the state vector $x_k$, based on the information in the measurements $y_{1:k}$. The algorithm is derived from the recursive decomposition of $p(x_{0:k}|y_{1:k})$ based on "Bayes rule" and "Law of total probability" [13]

$$p(x_{0:k}|y_{1:k}) = \frac{p(y_k|x_{0:k}, y_{1:k-1})p(x_{0:k}|y_{1:k-1})}{p(y_k|y_{1:k-1})} = \frac{p(y_k|x_{0:k}, y_{1:k-1})p(x_k|x_{0:k-1}, y_{1:k-1})p(x_{0:k-1}|y_{1:k-1})}{p(y_k|y_{1:k-1})} \tag{2}$$

If the probabilistic model of the transitional density is described by a Markov process of the first order such as $p(x_k|x_{0:k-1}, y_{1:k-1}) = p(x_k|x_{k-1})$ then the calculation of $p(x_{0:k}|y_{1:k})$ can be simplified. It is calculated recursively as

$$p(x_{0:k}|y_{1:k}) = \frac{p(y_k|x_k)p(x_k|x_{k-1})p(x_{0:k-1}|y_{1:k-1})}{p(y_k|y_{1:k-1})} \propto p(y_k|x_k)p(x_k|x_{k-1})p(x_{0:k-1}|y_{1:k-1}) \tag{3}$$

The recursion (3) cannot be computed analytically but it can be calculated using the sequential Monte Carlo approximation. The key idea underlying the sequential Monte Carlo methods is to represent the probability density function by a finite set of sample trajectories (particles) and their associated weights $\{x_{0:k}^{(i)}, w_k^{(i)}\}$.

The generation of samples from $p(x_{0:k}|y_{1:k})$ is performed in two steps: prediction and update [5]. In the prediction step, each path $x_{0:k-1}^{(i)}$ is grown with one step to obtain $\tilde{x}_{0:k}^{(i)}$ by sampling from the proposal density function $p(x_k|x_{k-1}^{(i)})$. In the update step, each sample path is associated with a weight, which is proportional to the likelihood of the measurements

$$w_k^{(i)} \propto w_{k-1}^{(i)} p(y_k|x_k^{(i)}) . \tag{4}$$

The resulting set of weighted trajectories $\{x_{0:k}^{(i)}, w_k^{(i)}\}$, $i$=1, …, $N$, with normalized weights provides an approximation to the distribution $p(x_{0:k}|y_{1:k})$. Based on the discrete approximation of the posterior pdf, an estimate of the "best" trajectory at step $k$+1 can be obtained. The mean represents a Monte Carlo approximation of the posterior pdf expectation which gives the "best" trajectory in terms of MMSE.

**Transitional prior and likelihood**

The proposal transitional prior is based on the dead-reckoning equations.

$$x_{k+1} = \begin{bmatrix} P_{k+1}^N \\ P_{k+1}^E \end{bmatrix} = x_k + V_k T_k \begin{bmatrix} \cos\psi_k \\ \sin\psi_k \end{bmatrix} \tag{5}$$

In most of cases, an additive zero-mean Gaussian noise in the speed measurements can be a good approximation. Then the particles can be simply sampled from the transitional prior described by

$$p(x_k|x_{k-1}^{(i)}) = \mathrm{N}(x_k; a_k, \Sigma_k) . \tag{6}$$

In this formula, the mean for this normal distribution is calculated based on the dead-reckoning equations in (5) using the vehicle speed and heading measurements from onboard sensors. Additional zero-mean white Gaussian noise is added to the speed measurements to improve the diversity of particles. The standard deviation of this noise $\Sigma_k$ is one of the design parameters.

It should be mentioned that the proposed system model has two operational modes: (1) the vehicle is moving along straight parts of the road and (2) the vehicle is turning or moving along the curved parts of the road. Switching between these two modes is performed based on the analysis of the vehicle's heading rate data from the sensors.

During the first operation mode the particles are propagated using only speed information from the onboard speed sensors (odometer, GPS etc). The $i$-th particle heading is assumed to be the same as the heading of the road segment where this particle is located, and which is known from the map:

$$\psi_k^{(i)} = \psi_{seg}^{(i)} . \tag{7}$$

This propagation model can guarantee that the particles will always stay on the road. However, different particles can move on different road segments. The road segment with the highest probability (with more particles on it) is selected as the most likely road segment where vehicle is located. If the particles are moving on the correct road segment then estimated position cross-track error can be reduced substantially by applying a simple perpendicular projection of the position fixes onto the selected link. The estimated vehicle position can also be calculated as the weighted average of all the particle coordinates from this segment.

During the second operational mode when the vehicle is turning, its heading and speed are required; the propagation model can be described by the dead-reckoning equations in (5) where the vehicle heading $\psi_k$ is measured by the gyro, GPS, or differential heading odometry. The road segment identification (map-matching) is not performed at this step. During the turn the vehicle and the particles are moving along some trajectories as illustrated in Fig. 1.

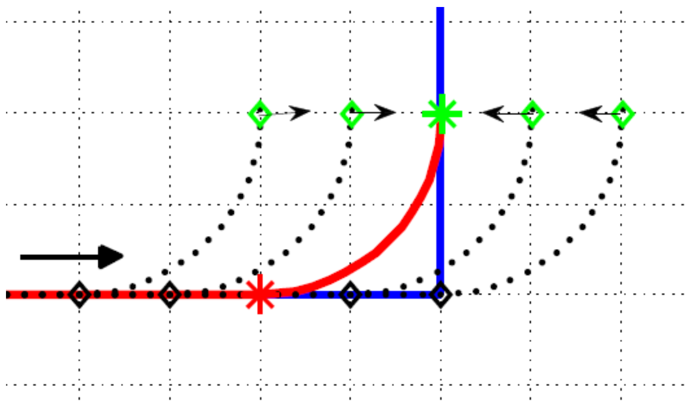

Figure 1. The true vehicle trajectory during the turn is shown by the solid line and the different particles trajectories are shown by the dotted lines. The asterisks designates the true vehicle location before and after the turn, the diamonds designate the particles locations before and after the turn. The direction of movement is shown by the large arrow.

There are some important features of these trajectories that help to reduce the along-track error of vehicle position estimation after the turn:

– The vehicle and the particles start turning at the same time since the turn is sensed by some heading-rate measuring device, e.g., gyroscope.
– The vehicle and the particles stop turning at the same time.
– If gyro and odometer are used as dead-reckoning sensors the accumulation of position errors during the turn is small. Therefore all these trajectories are nearly parallel and can be obtained by parallel translation of the true trajectory along the horizontal road link.
– In ideal case when propagation of particles during the turn is error free, the particles at the end of the turn will be on the same line parallel to the road link where they started the turn. Applying perpendicular projection of the particles position fixes onto the selected link will eliminate the along-track error of the estimated vehicle position accumulated before the turn.

Note that this approach of eliminating along-track error is valid for the case when the turn angle is different from ninety degrees. In other words it works for all curved roads. However, in reality, because of position errors accumulated during the turn, there will be some residual along-track error of the estimated vehicle position after the turn. The magnitude of this error depends on the quality of dead-reckoning sensors and curvature of the turn. For road links with small curvature, the reduction of along-track error is negligible.

The weights of the particles are updated using the recurrent formula (3). In this formula, $p\left(y_k \middle| x_{k-1}^{(i)}\right)$ is a likelihood calculated for each particle based on the proximity between the position fix and the particle, and the difference between the measured vehicle heading and the heading associated with this particle. The likelihood is calculated according to

$$p\left(y_k \middle| x_{k-1}^{(i)}\right) \propto \exp\left\{-\frac{1}{2}\left(\frac{\left(\psi_{seg}^{(i)} - \psi_{meas}\right)^2}{\sigma_{hdg}^2} + \frac{\left\|x_{k-1}^{(i)} - x_{meas}\right\|^2}{\sigma_{pos}^2}\right)\right\} \tag{8}$$

where $\psi_{seg}^{(i)}$ is the segments heading where the particle is currently located, $\psi_{meas}$ is a vehicle measured heading, $\sigma_{hdg}^2$ is heading measurement variance, $x_{k-1}^{(i)}$ is the $i$-th particle coordinate, $x_{meas}$ is the measured vehicle position, $\sigma_{pos}^2$ is the position measurement variance. If resampling is applied at each update step the relationship in (4) simplifies to:

$$w_k^{(i)} \propto p\left(y_k \middle| x_{k-1}^{(i)}\right). \tag{9}$$

**Simulations, field tests, and results**

This paper presents both simulation results and results from real-world data collection in a city environment. We start from the simulation results. To demonstrate the performance of the proposed algorithm the road network, car trajectory and position measurements were simulated (Fig. 2). It was assumed that the vehicle is traveling along the road (which is typically the case) and its heading matches the heading of the current road segment when the vehicle is travelling along straight stretches of road. We also assume that the terrain is flat and, therefore, the altitude will not be estimated. The road network consists of a set of parallel lines. However, this does not mean that the proposed algorithm can work only for this type of road network when the turns are always 90 degrees. The assumption of parallel roads and 90 degree turns is never used. The algorithm can work with roads of any shape and can be easily adapted for turns different from ninety degrees.

The triangles and circles denote the true vehicle position and the estimated position, respectively. The stars indicate the position measurements. For illustration purposes only, the corresponding measurement is connected with true position via a dotted line and with corresponding estimated position via a solid line.

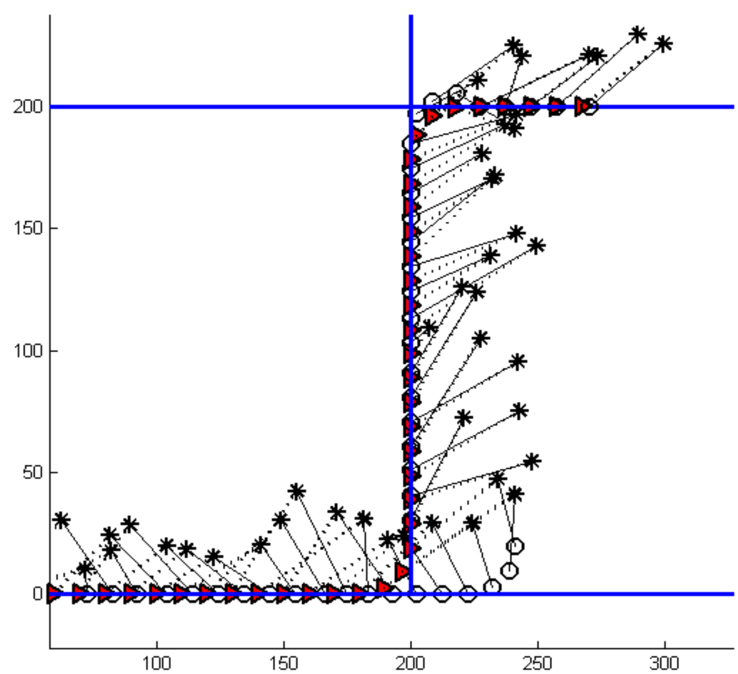


Figure 2. Simulation results. Triangles show the true vehicle position. Circles show the estimated position. Stars show the position measurements. For each time instant, the corresponding star, triangle, and circle are connected.

The speed over ground and heading (or ground track) is measured by some on-board sensors, for example, GPS. These measurements are corrupted by measurement noise with standard deviation of 1 m/sec and 1 deg, respectively. The position measurement errors include combination of constant offset and random noise with total RMS of approximately 40 m. Such large position errors correspond to performance of a GPS receiver in high-multipath urban environment.

The performance of the particle filter was evaluated when vehicle was moving along the trajectory that included several intersections with left and right turns. A part of this trajectory is shown in Fig. 2. The simulation results are based on 200 particles. The simulation results show that cross-track error is always reduced to a level of digital map error which is approximately 2-5 m in Western European countries. The along-track error is reduced after vehicle turns on intersections. For example before the second intersection the along-track error was approximately 10 m. After the turn the along-track error was reduced to approximately one meter. The distribution of particles along the road segment before and after the turn is shown in Fig. 3. Before the turn the deviation of estimated position from the true position is about 10 m and the standard deviation is about 4 m. In the histogram the horizontal axis represents the deviation of the particles from the true position in meters. After the turn, the deviation from the true position is less than 1 m and the standard deviation is about 2 m.

This example shows that the map-matching based on particle filter can improve substantially the positioning by reducing both along-track and cross-track errors. The cross-track error can be eliminated when the vehicle is not turning. In this case if the road segment is correctly identified, the vehicle position calculated by GPS or another navigation system can be corrected by projecting it onto chosen road link since we know that the vehicle is located on this segment. But this does not eliminate the along-tack error, which can be removed only when the vehicle is turning on intersections or moving on curved roads. In this case, the particle filter is switched to the second

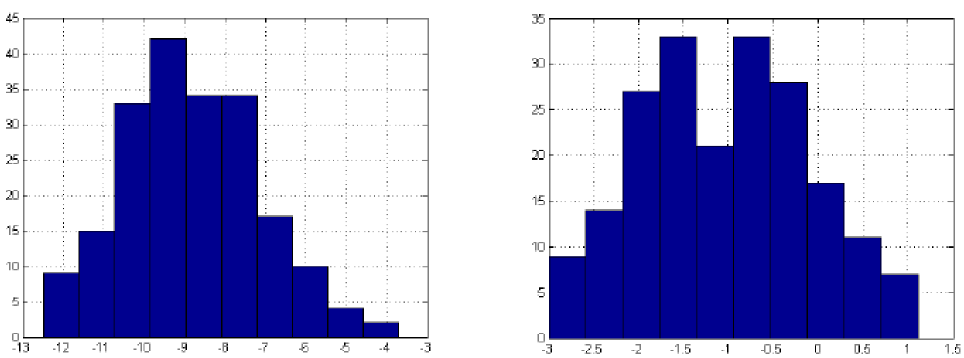


Figure 3. Distribution of particles along the road segment before and after the turn.

operation mode in which the vehicle heading is measured by an on-board sensor. After the turn the vehicle position estimate was substantially improved by reducing the along-track error to sub-meter level.

The proposed map-matching algorithm was also tested with actual digital maps and real-world heading rate and ground speed data. The heading rate was measured by the VTI Technologies' SCC1300 low-cost MEMS gyro [15]. The speed data was collected from the standard speed sensor installed in the car using the standardized digital communication port and on-board diagnostics (OBD) interface. The algorithm position accuracy was analyzed against the vehicle position as determined from the high accuracy Novatel differential GPS receiver with carrier phase capability. It should be mentioned that GPS was not used for navigation, although GPS signals were available most of the time. The purpose of these tests was to show that map aided low-cost dead reckoning navigation system can provide accurate navigation for long period of time without using any GPS data.

An example of map-matching algorithm performance during the turn is shown in Fig. 4. The progression of the original uncorrected dead reckoning solution is shown by the squares. The output rate of this solution is 1 Hz, and the last square corresponds to the currently estimated vehicle position. The black asterisk designates the corrected dead reckoning position estimate. Black points are the particles with their weighted mean location shown by the magenta asterisk. The diamond shows the true vehicle location as it determined by the high accuracy DGPS receiver. During the turn the particles' coordinates are not projected onto the road network and the particles are propagated using speed and heading from the on-board sensors according to the following formula:

$$x_{k+1}^{(i)} = x_k^{(i)} + V_k^{(i)} T_k \begin{bmatrix} \cos\psi_k^{(i)} \\ \sin\psi_k^{(i)} \end{bmatrix} \tag{10}$$

Here index $i$ corresponds to $i$-th sampled trajectory (particle), $i$=1,…, N (number of particles). $V_k^{(i)}$ is sampled from the normal distribution $V_k^{(i)} = N\left(V_k;\ \ \hat{V}_k,\ \ \Sigma_k\right)$ where $\hat{V}_k$ is the measured or estimated speed, $\Sigma_k$ is the standard deviation of this white Gaussian noise. This noise is added to improve diversity of the particles. The standard deviation of this noise $\Sigma_k$ is one of the design parameters. The last plot shows the moment when the turn is completed and map-matching is performed again.

This example shows that before the turn the along track error of the map-matching algorithm was approximately 25 m. This error was reduced after the turn to less than 5 m. The corrected dead reckoning solution was updated after the turn so its error becomes less than 10 m. The calculated position offset will be now added to the original dead reckoning solution until next position offset. The corrected dead reckoning position will be used as a position measurement when performing position update of the particles.

## Conclusions

This paper has developed and demonstrated a numerical probabilistic map-matching algorithm to accurately display vehicle location on a digital road map. However, maps can be used for more than simply display purposes. This paper has also shown how map-matching algorithm can improve car navigation system position accuracy. This becomes very important in urban environment when GPS position accuracy deteriorates because of multipath or during GPS outages when position calculation is based on DR sensors. In the case of map aided dead reckoning navigation system the position errors can be kept bounded as opposed to the unbounded error growth of the conventional dead reckoning. The examples show that the accuracy of estimated position using map-matching algorithm is about 2-5 m compared to tens of meters position errors without using the map-matching algorithm.

The performance of the proposed particle filter based map-matching algorithm is limited by the following factors:

- Frequency at which the turns occur. When the vehicle turns more often the position accuracy is better
- Position errors in digital road network. In Western European countries this error usually does not exceed 5 m.
- Uncertainty of vehicle location on the road. The road link is described by its centerline but the actual vehicle location can be slightly different.
- Errors in velocity and heading-rate sensors limit the along-track error estimation during turns.

The particle filter performance can be adjusted by changing the following parameters: ground speed noise, position and heading variances. Increasing ground speed noise improves the particles diversity. Position and heading variances have to match position and heading measurement errors of onboard sensors. If GPS is used to measure heading, the heading errors can be quite large during the turns and when the vehicle speed is low. This can also limit the accuracy of along-track error estimation.

The proposed map-matching algorithm is implemented now in Matlab for post-processing actual test data or simulated data. It should be noted that although the algorithm was tested off-line, it can easily be adapted for real-time implementation on portable navigation devices since the amount of computations is not large for modern signal processors. The future work may include the detection of cases when the car leaves the road network. The algorithm can be also extended for use in pedestrian navigation systems.

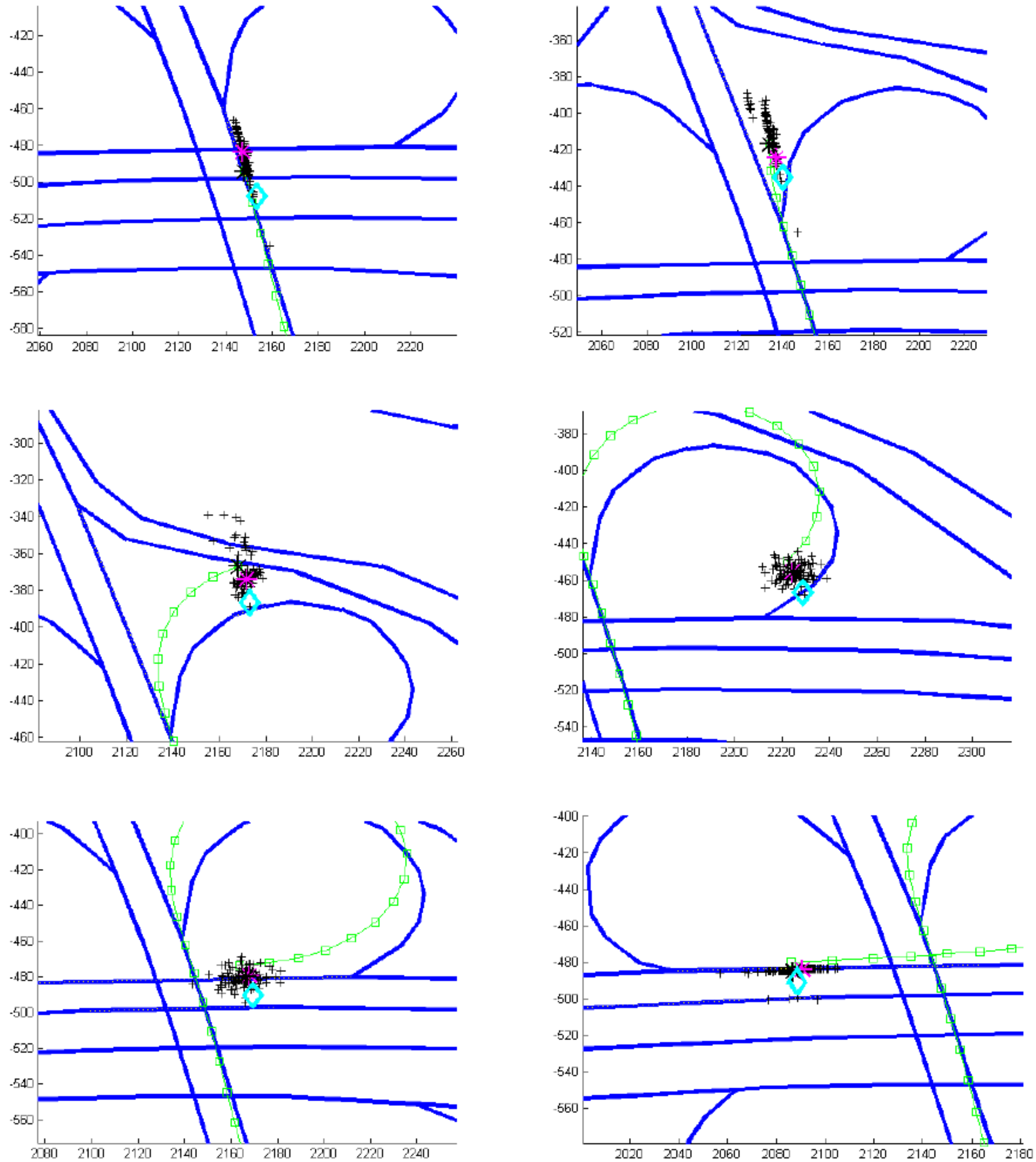


Figure 4. The proposed map-matching algorithm performance in car test where only speed data from the odometer and heading data from the gyro were used. GPS was not used in this test. The true vehicle position is shown by the diamond. The progression of DR trajectory is shown by the squares. The estimated position is shown by the asterisk.

## References


[1] French, R.L., “Land vehicle navigation and tracking,” Global Positioning System: Theory and Applications, vol. 2, pp. 275-301, 1996.

[2] Dmitriev, S., Stepanov, O., Rivkin, B., Koshaev, D., “Optimal map-matching for car navigation system,” in Proc. of 6th International Conference on Integrated Navigation Systems, St. Petersburg 1999.

[3] Scott, C., “Improved GPS positioning for motor vehicles through map matching,” in Proc. of ION GPS-94, Salt Lake City, UT, 1994, pp. 1019-1028.

[4] Zhao, Y., Vehicle Location and Navigation System, Artech House, pp. 83-103, 1997.

[5] Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T., “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking,” IEEE Transactions on Signal Processing, vol. 50, no. 2, Feb. 2002.

[6] Gustafsson, F., Gunnarsson, F., Bergman, N., Forssell, U., Jansson, J., Karlsson, R., and Nordlund, P.-J., ”Particle filters for positioning, navigation, and tracking,” IEEE Transactions on Signal Processing, vol. 50, no. 2, Feb. 2002.

[7] Fouque, C., Bonnifait, P., and Bétaille, D., ”Enhancement of global vehicle localization using navigable road maps and dead-reckoning,” in Proc. of ION Position, Location and Navigation Symposium, 2008.

[8] Syed, S. and Cannon, M.E., “Fuzzy logic-based map matching algorithm for vehicle navigation system in urban canyons,” in Proc. ION National Technical Meeting, San Diego, CA, Jan. 26-28, 2004.

[9] Kim, S. and Kim, J.-H., ”Adaptive fuzzy-network-based C-measure map-matching algorithm for car navigation system,” IEEE Transactions on Industrial Electronics, vol. 48, no. 2, Apr. 2001.

[10] Quddus, M.A., Ochieng, W.Y., Zhao, L., Noland, R.B., ”A general map matching algorithm for transport telematics applications,” GPS Solutions, vol. 7, no. 3, pp. 157-167, 2003.

[11] Grush, B., ”Road tolling isn’t navigation,” European Journal of Navigation, vol. 6, no 1, Feb. 2008.

[12] OpenStreetMap, http://www.openstreetmap.org/

[13] Doucet A., Godsill S., and Andrieu C., “On sequential Monte Carlo sampling methods for Bayesian filtering”, *Statistics and Computing* (2000) **10,** 197–208

[14] P. Davidson, J. Collin, J. Raquet and J. Takala, "Application of Particle Filters for Vehicle Positioning Using Road Maps," in proceedings of ION GNSS conference, Portland OR, September 22-24, 2010

[15] http://www.vti.fi/en/products/gyroscopes/