Navigation and Tracking of Road-Bound Vehicles

Fredrik Gustafsson, Umut Orguner, Thomas B. Schön, Per Skoglar and Rickard Karlsson

Abstract The performance of all navigation and tracking algorithms for road-bound vehicles can be improved by utilizing the trajectory constraint imposed from the road network. We refer to this approach as road-assisted navigation and tracking. Further, we refer to the process of incorporating the road constraint into the standard filter algorithms by dynamic map matching. Basically, dynamic map matching can be done in three different ways: (1) as a virtual measurement, (2) as a state noise constraint, or (3) as a manifold estimation problem where the state space is reduced. Besides this basic choice of approach, we survey the field from various perspectives: which filter that is applied, which dynamic model that is used to describe the motion of the vehicle, and which sensors that are used and their corresponding sensor models. Various applications using real data are presented as illustrations.

Fredrik Gustafsson, Umut Orguner, Thomas B. Schön, Per Skoglar, Rickard Karlsson
Division of Automatic Control,
Department of Electrical Engineering,
Linköping University, Sweden
e-mail: `{fredrik,umut,schon,skoglar,rickard}@isy.liu.se`

Contents

Navigation and Tracking of Road-Bound Vehicles 1
Fredrik Gustafsson, Umut Orguner, Thomas B. Schön, Per Skoglar and Rickard Karlsson

1 Introduction 4
2 State Estimation and Representation 7
2.1 Nonlinear Filtering 7
2.2 Introductory Illustrations 11
3 Basic Motion Models 13
3.1 Dead-Reckoning Model 13
3.2 A Complete Matlab Algorithm 15
3.3 Tracking Model 17
3.4 Manifold Model 18
4 Map Handling 18
4.1 The Shape Format 18
4.2 Computational Issues Related to the Map 20
5 Navigation Applications 21
5.1 Odometric Approach 21
5.2 Odometric Approach with Parameter Adaptation 23
5.3 Inertial Measurement Support 24
6 Tracking Approaches 28
6.1 Radar Support 28
6.2 Wireless Radio Network Support 29
6.3 Sensor Network Support 33
6.4 Vision Support 34
7 Conclusions 39
8 Acknowledgement 40
References 41

1 Introduction

There is a variety of localization, navigation and tracking applications that can be improved by restricting the location to be on roads marked on an available map. Essentially, the different applications are distinguished by what sensor combination that is used, and if the computations are done in the vehicle (navigation) or in the infrastructure (tracking). They all have in common that the on-road assumption greatly improves the accuracy, and that even quite poor signal to noise ratio of the sensors is sufficient to get a fairly good navigation performance.

The classical method to improve localization performance is *map matching*. Here, the position estimate computed from the sensors is mapped to the closest point in the road network. This is an appropriate method for presentation purposes, but it suffers from two problems. The first one is that it does not take the topography of the map into account, which implies that the localization can jump from one road to another due to quantization effects. The second one is that the motion dynamics of the vehicle is not combined with the map information in an optimal way. The purpose of this chapter is to survey different methods to what we will refer to as *dynamic map matching*.

Dynamic map matching combines a motion model, sensor models and the road network model in a nonlinear filter, taking uncertainties in all these three kind of models into account.

The problem boils down to fitting a distorted and noisy trajectory to the road network. Fig. 1 illustrates the principle and the basic problems considered one by one. In reality, several of these effects are combined. A typical example is navigation based on odometry, where the wheel speeds are integrated into a trajectory. The unknown absolute radius of the wheels implies a scaling error as in Fig. 1(c), the relative radii difference gives a bias in the yaw rate corresponding to Fig. 1(d), and the absolute course is not observed as illustrated in Fig. 1(b). Furthermore, the computed trajectory is uncertain due to noisy wheel speed measurements.

A generic nonlinear filter for navigation consists of the following main steps:

- *Time update or prediction:* Use a motion model to predict where the vehicle will be when the next measurement arrives.
- *Measurement update or correction:* Use the current measurement and a sensor model to update the information about the current location.

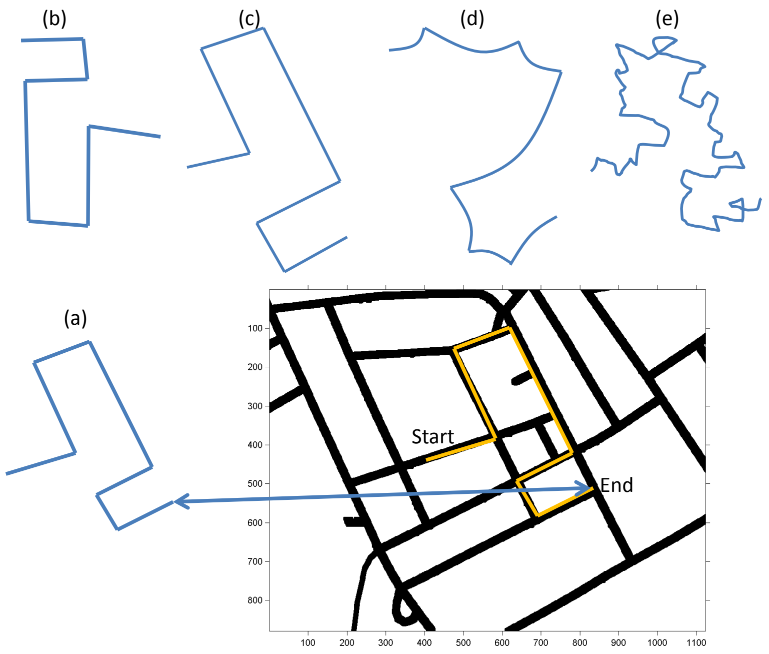


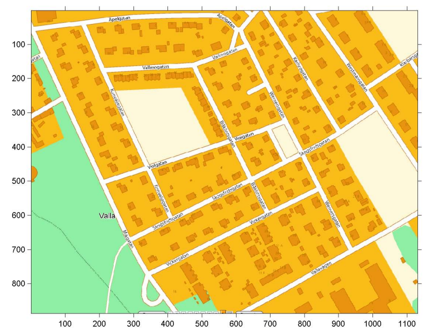
Fig. 1: The key idea in dynamic map matching is to fit an observed trajectory to the road network. (a) Undistorted trajectory. (b) Undistorted trajectory with random rotation. (c) Trajectory based on biased speed. (d) Trajectory based on biased yaw rate. (e) Trajectory with random noise.

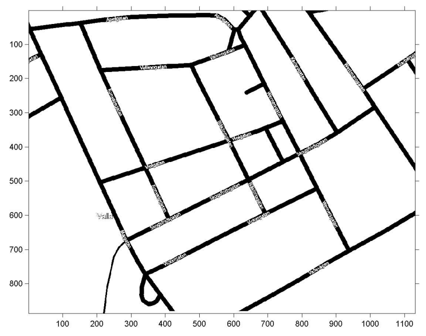
In a Bayesian framework, the information is represented by the posterior distribution given all available measurements. The process of computing the Bayesian posterior distribution is called filtering.

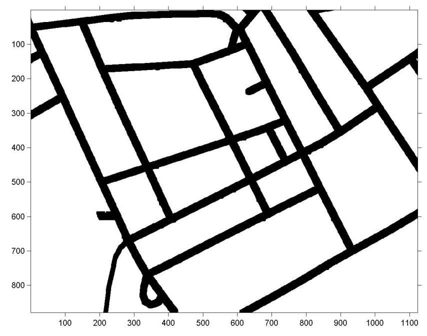
Fig. 2 illustrates how a standard map can be converted to a likelihood function for the position. Positions on roads get the highest likelihood, and the likelihood quickly decays as the distance to the closest road increases. A small offset can be added to the likelihood function to allow for off-road driving, for instance on un-mapped roads and parking areas.

There are three main principles for incorporating the on-road constraint:

1. As a *virtual measurement* using a road-tailored likelihood function, see Fig. 2(d), where predicted positions outside the road network are considered unlikely.


(a) Original map


(b) Binary map


(c) Filtered binary map

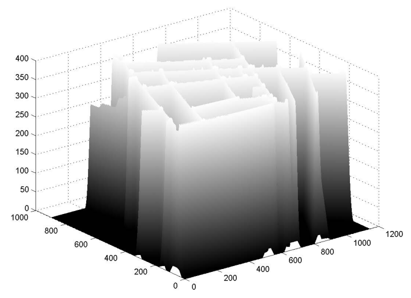

(d) On-road likelihood function

Fig. 2: (a) Original map. (b) Binary map, where the black areas corresponding to streets are mapped to one, and all other pixels are set to zero (white color). (c) The local maxima over a 4×4 region is computed to remove text information. (d) The resulting map is low-pass filtered to allow for small deviations from the road boarders, which yields a smooth likelihood function for on-road vehicles. See Listings 1 for complete Matlab code.

2. As a *state noise constraint* in the prediction step, so the predicted position is mostly on a road. Here, the likelihood in Fig. 2(d) is instrumental.
3. As *manifold filtering*, where the location is represented as the position along a road segment. This uses the topography of the map in a natural way.

As indicated explicitly in the first two cases, it is in practice necessary to allow the vehicle to temporary leave the road network to allow for off-road driving and un-mapped roads such as in parking areas and houses. This can be solved by having two modes in the filter, one on-road mode and one off-road mode, respectively.

There are two classes of problems with different support sensor options:

Navigation as driver information or for driver assistance systems as a GPS backup/support. Support sensors include wheel speed, inertial sensors and visual odometry using visual landmarks.
Tracking for surveillance or traffic monitoring. Support sensors include imagery sensors, radar, and sensor networks with microphones, geophones or magnetometers.

Tracking and navigation are in some sense dual problems, and the difference disappears in a cooperative setting where all equipment exchange information. Localization in cellular systems is one example, where network-centric (tracking) and user-centric (navigation) solutions exist, which are more or less similar. The main difference lies in where the algorithm is implemented, the algorithm itself can be the same. Here, we define tracking as all approaches that require infrastructure with communication ability (to exclude visual markers and passive radio beacons). In the sequel, we often use the term navigation for both classes of problems.

The outline is as follows. Section 2 surveys the different nonlinear filters that have been used for road-assisted navigation and tracking, with some illustrations from applications to illustrate the different concepts. Section 3 presents three fundamental and basic motion models, and provides concrete code. Section 4 discusses the data format used in the map, and explains the mathematical map matching operation. Section 5 summarizes some navigation applications while Section 6 overviews some tracking applications from our earlier research publications.

2 State Estimation and Representation

2.1 Nonlinear Filtering

We consider a general nonlinear motion model for the road-bound target with state x_k, position dependent measurement y_k, input signal u_k, process noise w_k, and measurement noise e_k:

$$x_{k+1} = f(x_k, u_k, v_k), \tag{1a}$$

$$y_k = h(x_k, u_k, e_k). \tag{1b}$$

The state includes at least position (X_k, Y_k) and heading (or course) ψ_k, and possibly derivatives of these and further parameters and states relevant in describing the motion.

Nonlinear filtering is the branch of statistical signal processing concerned with recursively estimating the state x_k in (1) based on the measurements up to time k, $y_{1:k} \triangleq \{y_1, \dots, y_k\}$ from sample 1 to k. The most general problem it solves is to compute the Bayesian conditional posterior density $p(x_k|y_{1:k})$.

There are several algorithms and representations for computing the posterior density:

- Kalman filter variants: the state is represented with a Gaussian distribution.
- Kalman filter banks based on multiple model approaches: the state is represented with a mixture of Gaussian distributions, each Gaussian mode having associated weights.
- Point mass and particle filters: the state is represented with a set of grid points or samples with an associated weight.
- Marginalized, or Rao-Blackwellized, particle filters: the state is represented with a number of trajectories over the road network, each one having an associated weight *and* Gaussian distribution for the other state variables than position.
- Finite state space models: the trajectory is represented by discrete probabilities for each combination of possible turns in the road network junctions.

These different filters are briefly introduced below.

2.1.1 Kalman Filter Variants

The *Kalman filter* (KF) [22] solves the filtering problem in case the model (1) is linear and Gaussian. The solution involves propagating the mean $\hat{x}_{k|k}$ and the covariance $P_{k|k}$ for the posterior Gaussian distribution

$$p(x_k|y_{1:k}) = \mathcal{N}\big(x_k; \hat{x}_{k|k}, P_{k|k}\big). \tag{2}$$

The *extended Kalman filter* (EKF) [34] and the *unscented Kalman filter* (UKF) [21] approximate the posterior at each step with a Gaussian density according to (2).

The road constraints imply a kind of information that normally leads to a multi-modal posterior density (the target can be on either this road, or that road, etc). The approximation in using (2) inevitably destroys this information. A completely different approach to nonlinear filtering is based on approximating the posterior $p(x_k|y_{1:k})$ numerically. One straightforward extension is to assign one KF to each hypothesis in a *multiple model* (MM) framework, which leads to a *Kalman filter bank* (KFB) technique with a Gaussian mixture posterior approximation [39],

$$p(x_k|y_{1:k}) \approx \sum_{i=1}^{N} w_k^i \mathcal{N}\big(x_k; \hat{x}_{k|k}^i, P_{k|k}^i\big). \tag{3}$$

The mixture probabilities w_k^i are all positive and sum to one. Each Gaussian distribution can be interpreted as a conditional distribution given one specific hypothesis about how the driven path matches the road map topography, where the observed sequence of turns fit the map in different ways. The *Manhattan problem* is used to illustrate the combinatoric explosion of hypotheses that are possible in a regular road pattern. There are two conceptually different ways to limit the number of hypotheses: pruning where unlikely hypotheses are thrown away, and merging where similar hypotheses which end up at the same position are merged into one. The *interacting multiple model* (IMM) [2] algorithm is a popular choice for the latter approach. Since the number of modes varies depending on excitation, compare with Fig. 9, the concept of *variable structure* (VS) has been adopted in the literature, see for instance [27].

2.1.2 Point Mass and Particle Filter Variants

The class of *point mass filters* (PMF) [25] represents the state space using a regular grid of size N, where the grid points and the related weights (x^i, w_k^i) are used as a representation of the posterior. Different basis functions have been suggested, the simplest one being an impulse at each grid, when the posterior approximation can be written

$$p(x_k|y_{1:k}) \approx \sum_{i=1}^{N} w_k^i \delta(x_k - x_k^i), \tag{4}$$

where $\delta(x)$ denotes the Dirac-delta function. The *particle filter* (PF) [13] is the state of the art numerical solution today. It uses a stochastic grid $\{w_k^i, x_k^i\}_{i=1}^N$ that automatically changes at each iteration. Another difference is that it in its standard form approximates the trajectory $x_{1:k}$. Otherwise, the representation of the posterior approximation is very similar to (4),

$$p(x_{1:k}|y_{1:k}) \approx \sum_{i=1}^{N} w_k^i \delta(x_{1:k} - x_{1:k}^i). \tag{5}$$

One should here note that in many navigation applications the sensor model is only a function of position. With an assumption of additive noise, the sensor model in (1b) can in such cases be written

$$y_k = h(X_k, Y_k) + e_k. \tag{6}$$

If the state x_k only includes position and velocity $x_k = (X_k, Y_k, \dot{X}_k, \dot{Y}_k)^T$, which is the simplest possible standard model in target tracking, then the *marginalized particle filter* (MPF, also known as RBPF, the Rao-Blackwellized PF) applies. The basic idea in the MPF is to utilize the structure in the model, so the Kalman filter can be applied to a part of the state vector (the velocity $V_k = (\dot{X}_k, \dot{Y}_k)^T$ in this case) in an optimal way. The resulting posterior approximation is then

$$p(x_{1:k}|y_{1:k}) \approx \sum_{i=1}^{N} w_k^i \delta(X_{1:k} - X_{1:k}^i)\delta(Y_{1:k} - Y_{1:k}^i)\mathcal{N}\big(V_k; \hat{V}_{k|k}^i, P_{k|k}^i\big). \tag{7}$$

This means that each particle represents a trajectory in the map, which has an associated Gaussian distributed velocity vector attached to it. The MPF can be applied to many other cases where the motion model contains more states than just position and heading. One of our *key messages* is that the MPF is well suited for road-assisted navigation and tracking.

2.1.3 Finite State Space Models

All approaches so far have assumed a continuous state space based on the 2D position. A completely different approach is based on a discrete state representing road segments defined by its junctions with other road segments. Let m denote a certain road segment (possibly one-way to indicate the direction of travel). Its end is connected to a number of other road segments $n_1, n_2, \dots, n_m$. Let the transition probabilities from road m to another road n be defined as

$$\text{Prob}(n|m) = \begin{cases} \pi_{nm} & n = n_1, n_2, \dots, n_m, \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

Then the complete road topology and prior knowledge of driving behavior, is summarized in the matrix Π with elements π_{nm}. The discrete *hidden Markov model* (HMM) theory provides the optimal filter for estimating the road segment sequence. The basic sampling rate depends on an event process, triggered by an external detection mechanism for indicating when a junction is reached. The estimated sequence in the original sampling rate indexed by k can be obtained by repeating the road segment between the junctions,

$$p(m_{1:k}|y_{1:k}) = \sum_{i=1}^{N} w_k^i \delta(m_{1:k} - m_{1:k}^i). \tag{9}$$

In this case, δ is the discrete pulse function.

Assume that the length of road segment i is L^i, and let $l_k^i \in [0, L^i]$ be the driven distance at this road segment. Then, we can form a joint state vector x_k with the continuous state l_k^i, and possibly one or more derivatives of position, for each discrete mode. The two problems can be combined, and the resulting mixture of discrete and continuous states can be expressed as

$$p(m_{1:k}, x_{1:k}|y_{1:k}) = \sum_{i=1}^{N} w_k^i \delta(m_{1:k} - m_{1:k}^i) \mathcal{N}\big(x_k; \hat{x}_{k|k}^i, P_{k|k}^i\big). \tag{10}$$

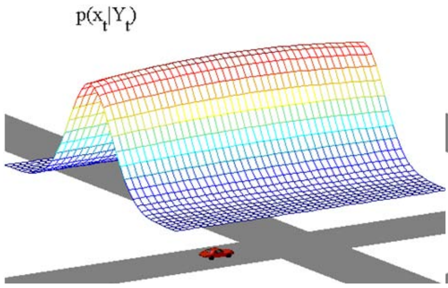
This can be seen as a version of the MPF, where the continuous state can be filtered analytically conditioned on a given sequence of discrete states. The resulting algorithm has a low-dimensional conditional state vector (*motion on a manifold*), and utilizes the map information in the most accurate way.

2.2 Introductory Illustrations

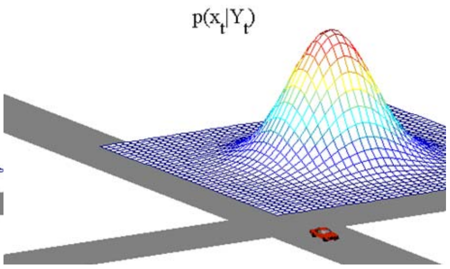
The Gaussian distribution is in many ways the most convenient representation in a range of applications, but as already mentioned a single Gaussian distribution has certain shortcomings for road-assisted navigation. We will here provide a couple of illustrations of posterior distributions, also showing the main principle in road-assisted navigation.

Consider the situation in Fig. 3, where a four-way intersection is approached. Suppose that the navigation algorithm has found the correct road segment and direction of travel, but that the information on the driven distance on this segment is uncertain. Then, we get the situation depicted in Fig. 3(a). Suppose now that the sensors detect a right turn. The posterior distribution in Fig. 3(b) is then quite informative about the position. This illustrates the information richness in the road map. Here, the Gaussian distribution is a feasible description of both the prior distribution after the prediction step and the posterior after a measurement update from an informative measurement.

The case in Fig. 3 assumed prior knowledge about the starting position or the direction of the vehicle. Suppose the prior distribution after the prediction step is instead rather uninformative. This is in Fig. 4(a) represented with two Gaussian distributions. Suppose now again that the sensors detect a right turn. Then, the posterior distribution will have four peaks, each one can be represented with a Gaussian distribution as shown in Fig. 4(b). If the intersections are regularly spaced (the "Manhattan problem"), it can be hard to resolve such ambiguities.



(a) PDF before turn.



(b) PDF after right turn.

Fig. 3: (a) The prior of the position close to a four-way intersection when the road segment of the vehicle is known, but its position along the segment is uncertain, can be modeled with a Gaussian distribution. (b) A few meters after a sensed right hand turn, the posterior distribution becomes very informative. Picture from [40].

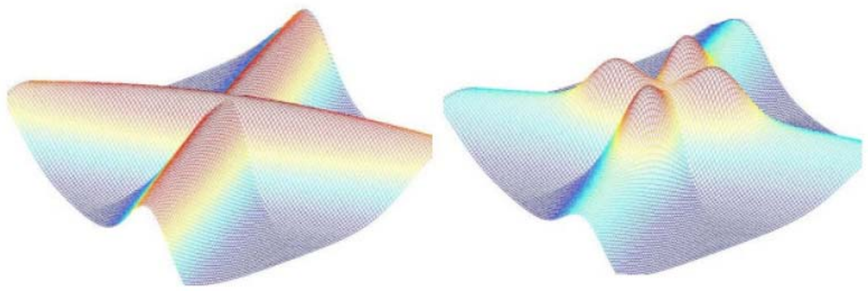


Fig. 4: A non-informative prior of the position close to a four-way intersection can be modeled using a Gaussian mixture with two modes, both centered at the intersection, and each one with a large eigenvalue spread in its covariance matrix along and transversal the road direction. A few meters after a sensed right hand turn, there are four different possibilities, leading to a Gaussian mixture with four modes. Picture from [40].

In manifold filtering, the posterior distribution is constrained to the road-network while the filter is operating in the on-road mode. A snapshot illustration is given in Fig. 5, where a Gaussian mixture summarizes the information from a sensor network.

Even though (mixtures of) Gaussian distributions are feasible representations of the posterior distribution, a sample based approximative representation is in many cases even more useful. Fig. 6 illustrates the key idea: to replace a parametric distribution with samples, or particles.

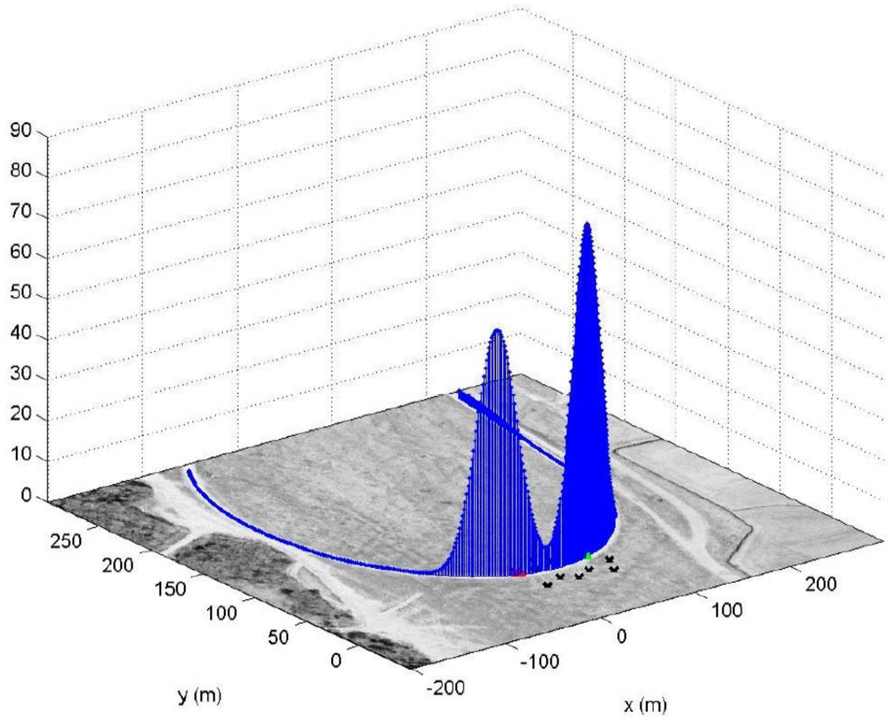


Fig. 5: A Gaussian mixture distribution for modeling the posterior along a road segment, which is marked with a solid line. A microphone network provides the measurements, and each sensor is marked with a cross.

3 Basic Motion Models

We here describe three specific and simple two-dimensional motion models that are typical for the road-assisted applications.

3.1 Dead-Reckoning Model

A very instructive and also quite useful motion model is based on a state vector consisting of position (X, Y) and course (yaw angle) ψ. This assumes that there are measurements of yaw rate (derivative of course) $\dot{\psi}$ and speed ϑ on-board the platform, in which case the principle of *dead-reckoning* can be applied.

The dead-reckoning model can be formulated in continuous time using the following equations:

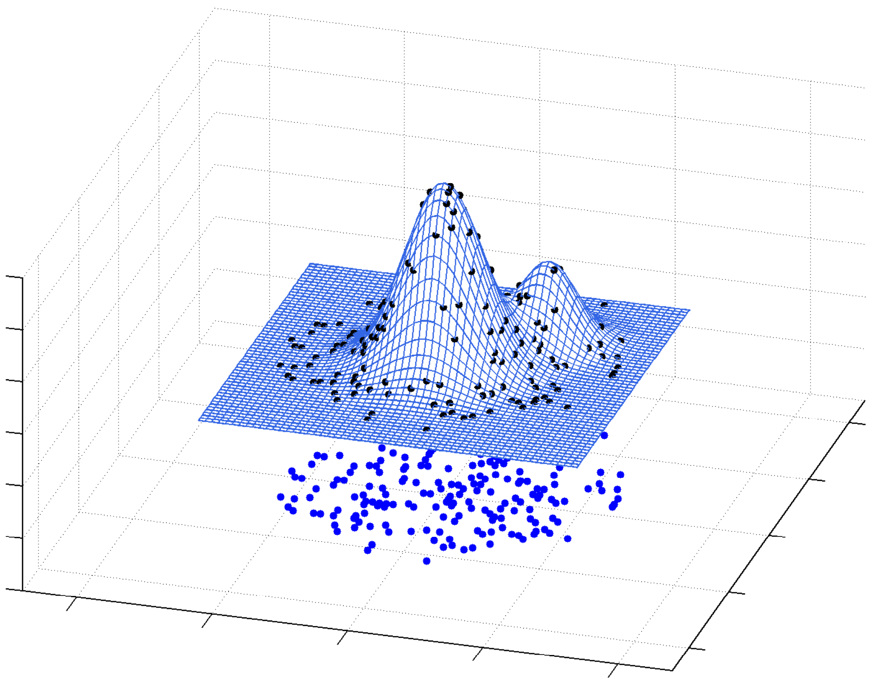


Fig. 6: A bi-modal Gaussian distribution can as any other distribution be approximated by a set of random samples. This is the idea of the particle filter, and the particle representation of the posterior distribution.

$$x(t) = \begin{pmatrix} X(t) \\ Y(t) \\ \psi(t) \end{pmatrix}, \quad \dot{x}(t) = \begin{pmatrix} \vartheta(t)\cos(\psi(t)) \\ \vartheta(t)\sin(\psi(t)) \\ \dot{\psi}(t) \end{pmatrix} \tag{11}$$

A discrete time model for the nonlinear dynamics is given by

$$\begin{aligned} X(t+T) &= X(t) + \frac{2\vartheta(t)}{\dot{\psi}(t)} \sin(\frac{\dot{\psi}(t)T}{2}) \cos(\psi(t) + \frac{\dot{\psi}(t)T}{2}) \\ &\approx X(t) + \vartheta(t)T\cos(\psi(t)), && \text{(12a)} \\ Y(t+T) &= Y(t) + \frac{2\vartheta(t)}{\dot{\psi}(t)} \sin(\frac{\dot{\psi}(t)T}{2}) \sin(\psi(t) + \frac{\dot{\psi}(t)T}{2}) \\ &\approx Y(t) + \vartheta(t)T\sin(\psi(t)), && \text{(12b)} \\ \psi(t+T) &= \psi(t) + T\dot{\psi}(t). && \text{(12c)} \end{aligned}$$

Finally, plugging in the observed speed $\vartheta^m(t)$ and angular velocity $\dot{\psi}^m(t)$ gives the following dynamic model with process noise $w(t)$

$$X(t+T) = X(t) + \vartheta^m(t)T\cos(\psi(t)) + T\cos(\psi(t))w_\vartheta(t), \tag{13a}$$
$$Y(t+T) = Y(t) + \vartheta^m(t)T\sin(\psi(t)) + T\sin(\psi(t))w_\vartheta(t), \tag{13b}$$
$$\psi(t+T) = \psi(t) + T\dot{\psi}^m(t) + T\sin(\psi(t))w_{\dot{\psi}}(t). \tag{13c}$$

This model has the following structure

$$x_{k+1} = f(x_k, u_k) + g(x_k, u_k)v_k, \qquad u_k = \begin{pmatrix} \vartheta^m_k \\ \dot{\psi}^m_k \end{pmatrix} \tag{13d}$$

that fits the particle filter perfectly. Normally, additional support sensors are needed to get observability of the absolute position. This is for instance the case in many robotics applications. However, the road map contains sufficiently rich information in itself.

Note that the speed and the angular velocity measurements are modeled as inputs, rather than measurements. This is in accordance to many navigation systems, where inertial measurements are dead-reckoned in similar ways. The alternative is to extend the state vector with speed and angular velocity, but this increased state dimension would make the particle filter less efficient unless some Rao-Blackwellization is used.

3.2 A Complete Matlab Algorithm

Suppose that an input sequence $u_{1:N}$ or speed and angular velocity (yaw rate), and a likelihood map similar to the one in Fig. 4(d) are given. The likelihood function $L(i, j)$ is assumed to be represented with a matrix where each row i corresponds to the corresponding element $X(j)$ in the vector X, and similarly for the column $Y(j)$ for a vector Y. A likelihood function such as the one in Figure 2 can be generated from an arbitrary map using the code in Listings 1.

Listing 1: Matlab code for generating likelihood from a bitmapped map

```
y = imread('valla.png');      % Snapshot of map
ys=sum(y,3);                  % r+g+b
ind=find(ys~=761);            % White rgb value in map
yr=zeros(size(ys));           % 0 for non street areas
yr(ind)=1;                    % 1 for street areas
ym=locmin2(yr,4);             % Special: local min over 9x9 square
L=conv2(1-ym,ones(20,20));    % LP-smoothing gives likelihood
surf(L);
shading interp;
campos=[-1000 -4000 10000];
```

The function `locmin2` is non-standard but simple. It computes the local minimum over a square of nine times nine pixels (within a distance of 4 pixels). Using this likelihood and some additional parameters for the coordinate transformation from pixels to world coordinates, the particle filter can be implemented in Matlab as given in Listings 2.

Listing 2: Complete Matlab listing for positioning.

```
function Xhat = MapAidedPositioning(y,u,L,pe,vrand,f,h,p0,dp)
Tf          = size(u,2);                      % Number of data
N           = 1000;                           % Particles
[IndX,IndY] = find(L>0.5*max(L(:)));          % Thresholding
IndR        = ceil(length(IndX)*rand(N,1));   % Random road points
Psi         = 2*pi*rand(N,1);                 % Random heading

% Coordinate transformation
X = [0:size(L,2)-1] * dp(1) + p0(1);
Y = [0:size(L,1)-1] * dp(2) + p0(2);

% Initialization
Xp(1,:) = IndX(IndR) * dp(1) + p0(1);
Xp(2,:) = IndY(IndR) * dp(2) + p0(2);
Xp(3,:) = Psi;

for k = 1:Tf
   % Road likelihood
   w = interp2(X,Y,L,Xp(1,:),Xp(2,:),'nearest',0);

   w = w.*pe(y(:,k),Xp);
% Measurement likelihood
   w = w/sum(w);                             % Normalization
   Xhat(:,:) = w(:)'*Xp';                    % Mean estimate
   Xp = resample(Xp,w);                    % Resampling
   vk = vrand(N);                            % Process noise
   Xp = f(Xp,u(:,k),vk);                     % State prediction
end
```

The code in Listings 2 is complete, except for the basic resampling function `resample`. Implementations and a discussion on this function are found in [15]. Initialization is performed over the whole road network. The PF is the simplest possible bootstrap (SIR) one originally proposed in [13], and there are more advanced ones that can be more efficient for this application, see [15] for a more thorough treatment of implementation and code aspects.

Finally, if there are more sensor information relating to position (such as temporary GPS positions) or course (such as a compass), these are easily incorporated in the filter as additional likelihood function multiplications.

3.3 Tracking Model

The simplest possible tracking model, yet one of the most common ones in applications, is given by a two-dimensional version of Newton's force law:

$$x(t) = \begin{pmatrix} X(t) \\ Y(t) \\ \dot{X}(t) \\ \dot{Y}(t) \end{pmatrix}, \quad \dot{x}(t) = \begin{pmatrix} \dot{X}(t) \\ \dot{Y}(t) \\ w^X(t) \\ w^Y(t) \end{pmatrix} \tag{14a}$$

The corresponding discrete time model is given by

$$x_{k+1} = \begin{pmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} x_k + \begin{pmatrix} T^2/2 & 0 \\ T & 0 \\ 0 & T^2/2 \\ 0 & T \end{pmatrix} \begin{pmatrix} w_k^X \\ w_k^Y \end{pmatrix}. \tag{14b}$$

Suppose the sensor model depends on the position only, similarly to (6),

$$y_k = h(X_k, Y_k) + e_k. \tag{15}$$

Since the motion model is linear in the state and noise, the MPF applies, so the velocity component can be handled in a numerically very efficient way.

The code in Listings 3 gives a fundamental example that fits GPS measurements to a road map using a constant velocity model.

Listing 3: Example of how to use the function in Listing 2 for tracking using GPS measurements.

```
% Measurement model.
h=inline('sum( (repmat(y,size(x,2))-[x(1,:);x(2,:)]).^2 ),1)');
pe=inline('exp(-0.5*sum((repmat(y,1,size(x,2))-x(1:2,:)).^2)/10^2)','y','x');

% Dynamic model.
f=inline(['[x(1,:)+u(1,:)+0.5*v(1,:).*cos(x(3,:));'...
          ' x(2,:)+u(2,:)+0.5*v(1,:).*sin(x(3,:));'...
          ' x(3,:)+v(2,:)];'],'x','u','v');
vrand=inline('randn(2,N)');

% PF
Xhat=MapAidedPositioning(GPS,[0;0],L,pe,vrand,f,h,[0 0],[0.5 0.5]);
```

3.4 Manifold Model

We here return to the filter framework discussed in Section 2.1.3. The manifold model is essentially a one-dimensional version of the tracking model (14),

$$x(t) = \begin{pmatrix} l(t) \\ \dot{l}(t) \end{pmatrix}, \quad \dot{x}(t) = \begin{pmatrix} \dot{l}(t) \\ w(t) \end{pmatrix}, \tag{16a}$$

with a discrete time counterpart

$$x_{k+1} = \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix} x_k + \begin{pmatrix} T^2/2 \\ T \end{pmatrix} v_k. \tag{16b}$$

With a particle representation of the position l_k along the current road segment, the event of passing a junction is easily detected ($l_k^i < 0$ or $l_k^i > L^i$), and the new road segment can be initialized according to the prior probability π_{ji}. Note that the velocity component $\dot{l}_k$ can be represented with a KF according to the MPF algorithm, since the velocity is linear and Gaussian in the model. The posterior then resembles the expression in (7).

The same sensor model as in (6) and (15) now also includes a transformation

$$y_k = h(X^i(l_k), Y^i(l_k)) + e_k, \tag{17}$$

where $X^i(l)$ is the mapping from the driven distance l on road segment i to the Cartesian position X, and similarly for $Y^i(l)$.

4 Map Handling

This section describes the fundamentals of vectorized road maps, and some key calculations needed in (dynamic) map matching.

4.1 The Shape Format

In a geographic information system (GIS) different forms of geographically referenced information can be analyzed and displayed. There are two classical methods to store GIS data: raster data (images) and vector data. Different geometrical types can be described by vector data and basically there are three broad type categories; zero-dimensional points are used to represent points-of-interest, lines are used to represent linear features such as roads and topological lines, and polygons are used to represent particular areas such as lakes. There exist many approaches to store geospatial vector data and one popular representation is the ESRI shape file. There are 14 different shape types, for example, a road network is represented as a number of PolyLines. A PolyLine is an ordered set of vertices that consists of one or more parts. A part is a connected sequence of two or more points. Parts may or may not be connected to one another. Parts may or may not intersect one another. See [10] for more details. Apart from the vector information, each data item may also have attributes that describe properties of the item. Examples of attributes in the road case are road type, street name, speed limit, driving direction, etc. Here, only road network information is considered, but of course there are other types of information such as terrain type and topological data that can facilitate the target tracking and sensor fusion performance.

For target tracking purposes it is convenient to have a slightly different representation with redundant information to facilitate and speed up the data processing. One data structure represents the roads and this structure contains the road stretch and the corresponding attributes. This structure is more or less the raw shape data plus an ID number for each road and an intersection ID for each road end. An additional structure is used for the intersections and it contains the location and all connected roads (IDs) of each intersection. The exact data structure depends on what type of additional information is included, such as driving direction and prior probabilities for roads in an intersection. The described road structure contains the following fields:

- ID – unique road ID
- N – number of parts
- x – $(1 \times N)$ vector with x coordinates
- y – $(1 \times N)$ vector with y coordinates
- z – $(1 \times N)$ vector with z coordinates
- i_1 – intersection ID of the road start intersection
- i_2 – intersection ID of the road end intersection

and the intersection structure contains

- ID – unique intersection ID
- M – number of connecting roads
- r – $(1 \times M)$ vector with IDs of the connecting roads
- x – x coordinate
- y – y coordinate
- z – z coordinate.

4.2 Computational Issues Related to the Map

Even though the shape format allows for curved road segments, all available maps use the straight line representation. This means that for instance roundabouts are approximated using a number of straight lines. This section will describe a couple of computations needed in road-assisted navigation.

First, consider the virtual measurement approach. If the likelihood is represented as the grid in Fig. 2(d), then the measurement update is based on interpolation in this grid. This approach requires certain pre-computations and a large memory. A more efficient approach is based on defining a likelihood function based on the distance d to the closest road point. The likelihood can for instance be defined as $l(d) = e^{-\frac{d^2}{2\sigma^2}} + l_0$, where l_0 is optionally added to allow for off-road driving.

Denote the shortest distance to road segment i with d_i, see Fig. 7(a). Using the scalar product, it is given by

$$d_i = x_k^p - p_i - \frac{\left(x_k^p - p_i\right)^T \left(p_{i+1} - p_i\right)}{\left(p_{i+1} - p_i\right)^T \left(p_{i+1} - p_i\right)} (p_{i+1} - p_i). \tag{18}$$

Note that this value must satisfy $d_i \in [0, L_i]$ to be feasible. This calculation has to be performed for all road segments to find the minimum $d = \min_i d_i$. Clearly, an efficient data base handling is required. Further, there is a need for an efficient pre-scan of a suitable candidate set of road segments. Here, the absolute norm to each end point can be used, $\|x_k^p - p_i\| = |x_k^X - p_i^X| + |x_k^Y - p_i^Y|$.

Note that the above operation is also needed in standard map matching using vectorized maps.

The second approach to road-assisted navigation is based on a state constraint in the prediction step, and this can be rather tricky. Consider a linear motion model, for instance the one in (14b),

$$x_{k+1} = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix} x_k + \begin{pmatrix} B_1 \\ B_2 \end{pmatrix} v_k, \tag{19}$$

where the upper blocks A_1 and B_1 correspond to position. Now, if $v_k \in \mathcal{N}(0, \sigma^2 I_2)$, we need to generate constrained samples from this distribution that assures that $A_1 x_k + B_1 v_k$ corresponds to a point on the road network. Fig. 7(b) illustrates the geometry for this directional process noise. In this case with a linear road segment, the conditional distribution can be computed analytically. The result is a one-dimensional Gaussian distribution where the variance is smaller than σ^2, the larger d the smaller variance.

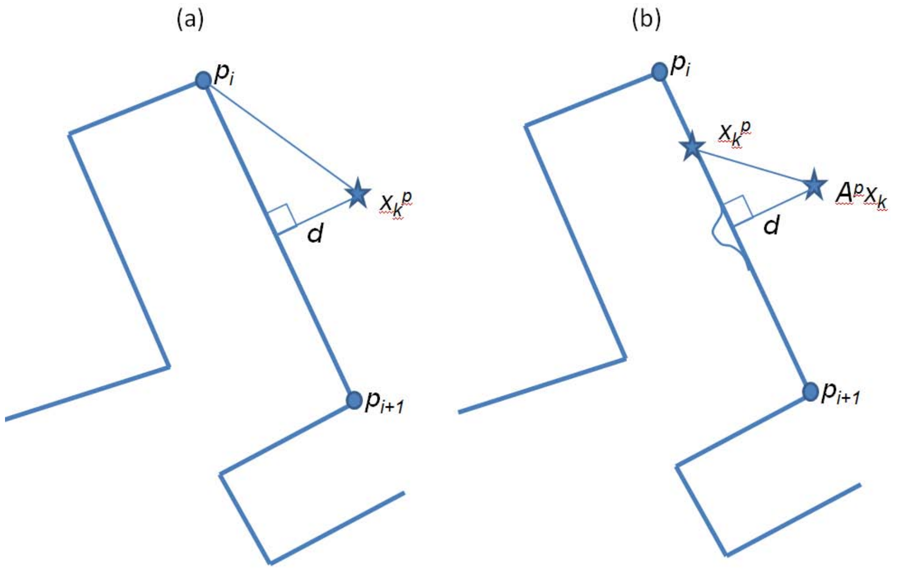


Fig. 7: Computational issues related to the trajectory in Fig. 1. (a) The closest distance d to the road network needs to be computed in the virtual measurement approach. (b) A random noise that takes any prediction back to the road network needs to be generated in the state constraint approach.

5 Navigation Applications

This section surveys some approaches to on-board navigation systems based on dead-reckoning. It explains the basic sensor models, and provides some illustrative examples from field tests. Such systems can be used as a support or back-up to satellite based navigation.

5.1 Odometric Approach

Odometry is the term used for dead-reckoning the rotational speeds of two wheels on the same axle of a vehicle. It is used in a large range of robotics applications, as well as in some vehicle navigation systems. Odometric navigation or positioning based on inertial sensors or dead reckoning sensors is challenging due to drift and bias in the measurements. Particularly for relatively cheap sensors that are available in ordinary passenger vehicles. Map aided positioning based on vectorized road charts in combination with information from internal automotive sensors such as individual wheel speed and

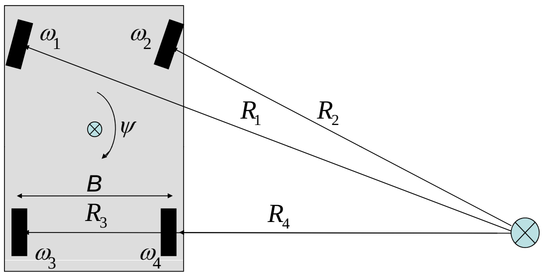


Fig. 8: Notation for the lateral dynamics and curve radius relations for a four-wheeled vehicle.

yaw rate available from the CAN-bus has been studied in several Master's Theses [18,20,26,40] and made commercially available at Nira Dynamics AB.

In [18] the basic theory and implementation for map aided vehicle positioning, was studied, where the particle filter was demonstrated to yield sufficiently good navigation performance when incorporating information from yaw rate and wheel speed sensors. The unknown wheel radius parameter estimation problem was also addressed. The raw signals are the angular velocities of the wheels which can be measured by the ABS sensors in cars. The angular velocities can be converted to virtual measurements of the absolute longitudinal velocity and yaw rate as (see [14] or Chapter 13 and 14 in [16] for details), assuming a front wheel driven vehicle with slip-free motion of the rear wheels,

$$\vartheta^m = \frac{\omega_3 r_3 + \omega_4 r_4}{2} \approx \vartheta + w_\vartheta, \tag{20a}$$

$$\dot{\psi}^m = \vartheta^m \frac{2}{B} \frac{\frac{\omega_3 r_3}{\omega_4 r_4} - 1}{\frac{\omega_3 r_3}{\omega_4 r_4} + 1} \approx \dot{\psi} + w_{\dot{\psi}}. \tag{20b}$$

See Fig. 8 for the notation. The noise terms can be assumed to be Gaussian,

$$w_\vartheta \sim \mathcal{N}(\delta_\vartheta, \sigma_\vartheta^2), \tag{21a}$$

$$w_{\dot{\psi}} \sim \mathcal{N}(\delta_{\dot{\psi}}, \sigma_{\dot{\psi}}^2). \tag{21b}$$

We assume in this section that both the mean and the variance are known. Fig. 9 shows an example.

During this development stage different hardware platforms were investigated. In [40] varies particle filter variants were studied and the MPF/RBPF was used as an efficient method to incorporate more states in real-time. The ability to enhance positioning while driving slightly off-road was demonstrated in [26]. This is important for handeling of inaccurate maps or when the map information is not available. In [20] the positioning aspect was shifted from the use of internal data such as wheel speed information to the case when inertial measurements are available from an external IMU sensor. Typically the problem studied used accelerometers and gyros as a stand alone sensor for positioning in combination with the road map. Several different models were studied also in combination with the available internal automotive sensors.

In Fig 9 the map aided positioning using wheel speed information and road map information is demonstrated, where GPS information is used as a ground truth reference only. In Fig 9 (a), the particle filter is initialized in the vicinity of the GPS position (blue circle) at a traffic light. The initial distribution is uniform on road segments in a region around the GPS fix. As seen the GPS position is located slightly off-road, which could be due to a measurement error, multi-path phenomenon, or that the road segment width does not match the actual road width. To ensure a robust algorithm against these issues, particles are allowed slightly off-road. The expected mean from the particle filter (red circle) is far away from the true position. In Fig 9 (b) the algorithm has been active for some time, and the vehicle has turned left at the crossing. As seen, the PDF is now highly multi-modal. The PF algorithm uses only wheel speeds from the CAN-bus. The GPS is only used to evaluate the ground truth. In Fig 9 (c), after yet some turns, the filter has converged to a uni-modal distribution and the mean estimate is close to the GPS position.

5.2 Odometric Approach with Parameter Adaptation

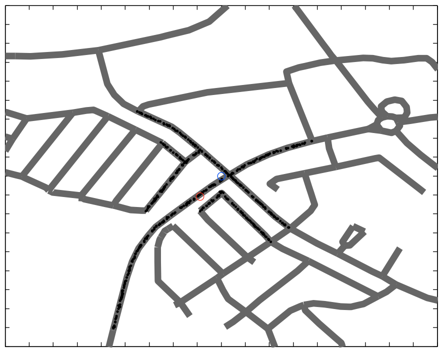
The offsets in (21) depend on the wheel radii as

$$\delta_{\vartheta} \propto r_3 + r_4, \tag{22a}$$

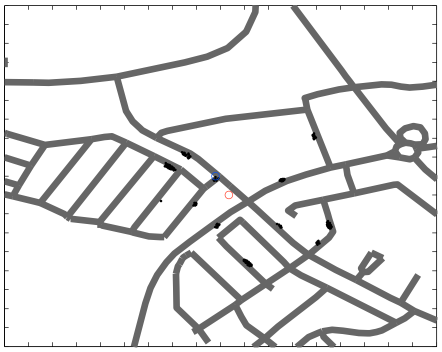
$$\delta_{\dot{\psi}} \propto r_3 - r_4. \tag{22b}$$

Further, the noise variances depend on the surface. Both surface and wheel radii change over time, but with different rates, and the offsets are crucial for dead-reckoning performance.

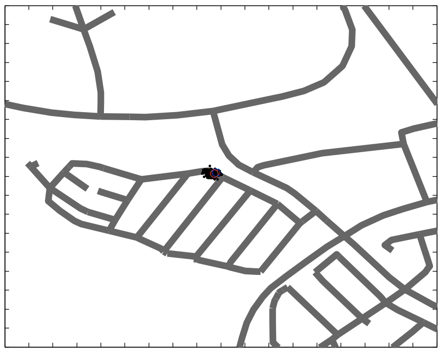
The standard approach to deal with this problem is to augment the state vector with these two offsets (or the more physical wheel radius offsets). These parameters are then estimated adaptively in the filter. Fig. 10 illustrates one



(a) The PF is depicted shortly after initialized using prior information.



(b) Multi-modal PDF representing the position.



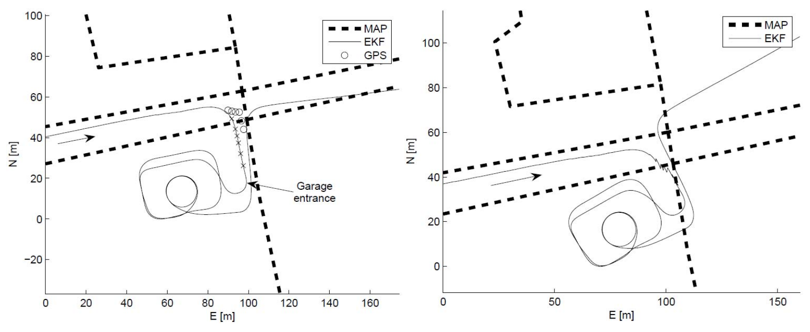
(c) The PF has converged to a uni-modal PDF

Fig. 9: Map aided positioning using wheel speed sensor information in combination with road map information.

advantage with this approach: dead-reckoning improves to enable accurate GPS integrity monitoring, so that small GPS errors are detected.

5.3 Inertial Measurement Support

The drawback with the approaches in the preceding sections is that they require wheel speed signals. This is not easy for portable and after-market solutions. An appealing approach is to base the dead-reckoning on an *inertial measurement unit* (IMU). This can either be three-dimensional (two horizontally mounted accelerometers and a yaw-rate gyro) or a full six-dimensional



(a) GPS integrity check

(b) No GPS integrity check

Fig. 10: GPS supported odometry. Multi-path propagation close to the parking house gives unreliable GPS positions (marked in (a)). (a) Bias adaptation gives the predictive performance required to exclude GPS outliers. (b) Illustration of what happens if GPS outliers are used in the filter.

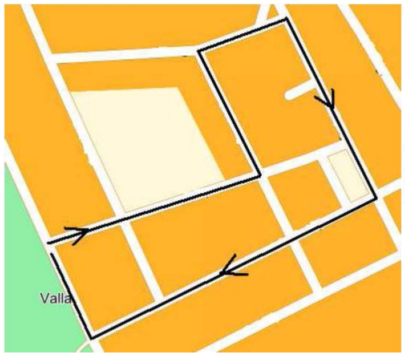
unit with three accelerometers and three gyros. The former assumes a flat world and no roll and pitch dynamics of the vehicle, while the latter allows for a more flexible and versatile full state estimation framework. Further, just one single lateral accelerometer can be used to detect cornering, an important event in dynamic map matching. We here summarize the results in [20].

Fig. 11 and Fig. 12 (from [20]) illustrate some different combinations, providing the following conclusions:

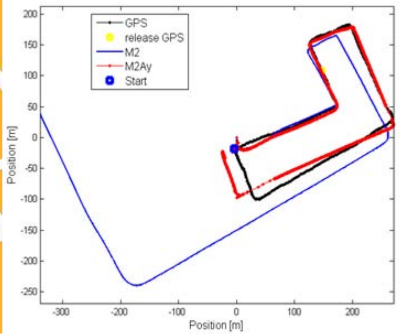
- The wheel speed based dead-reckoning gives superior performance.
- Pure dead-reckoning of an IMU cannot be used as a backup solution to GPS, unless the initial alignment and offset estimation is improved, see Fig. 11(b).
- Dynamic map matching based on dead-reckoning of an IMU and cornering detection from lateral acceleromater is a feasible backup solution to GPS, see Fig. 11(b). However, the robustness is not convincing, see the first plot in Fig. 12.
- When wheel speed signals are available, also including IMU does not improve the result of dynamic map matching significantly, see Fig. 11(c-d) and the last two plots in Fig. 12. However, for some driving conditions this might be the case.

In [20] several different models were studied to find a feasible real-time implementation with sufficient flexibility and performance. The studied models are summarized below:

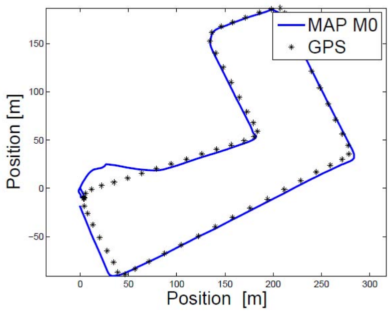
- Model M0: a pure odometric model (see Section 3).



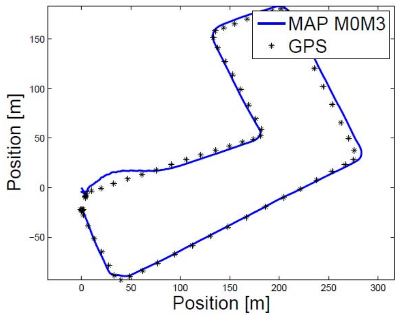
(a) Driven path on map



(b) Comparison of pure 2D dead-reckoning (M2) and 2D dead-reckoning with corner detection and map matching (M2Ay)



(c) Wheel speed based dead-reckoning and dynamic map matching (M0)



(d) Wheel speed and 6D IMU based dead-reckoning and dynamic map matching (M0M3)

Fig. 11: IMU supported navigation. Pictures from [20].

- Model M1: a general 3D constant acceleration model with quaternions for orientation representation and accelerometer and rate gyro biases (22 states).
- Model M2: utilized the 2D property of vehicle positioning and it used a coordinated turn model (11 states)
- Model M2ay: the same as M2 but also using the lateral accelerometer to improve positioning.
- Model M3: a simplified coordinated turn model without biases (6 states).

In Model M1, the following state vector is used

$$x = \left(p\ v\ a\ q\ w\ a_{\text{bias}}\ \ w_{\text{bias}}\right)^T, \tag{23}$$

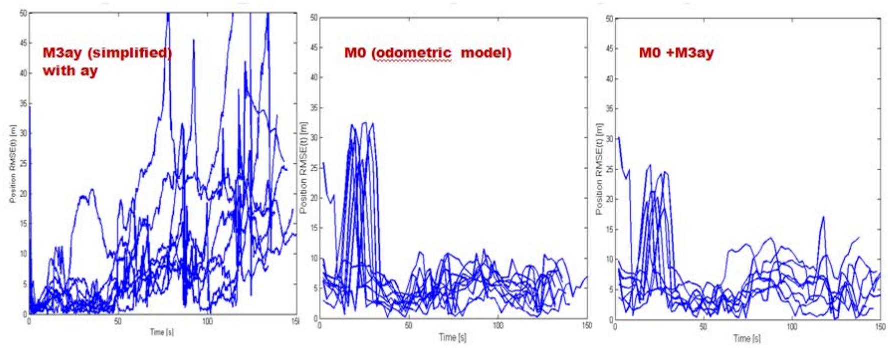


Fig. 12: The test drive in Fig. 11(a) is repeated many times, so the average performance and robustness of the alternatives can be compared. Picture from [20].

where p is the 3D position vector, v the 3D velocity vector, a the 3D acceleration vector, q the four components of the quaternion vector representing orientation, and ω is the angular rate. The continuous time dynamics for the model is given by

$$\dot{x}(t) = \left(v \ a \ w_a \ -0.5S(w)q \ w_w \ w_{a,\text{bias}} \ w_{w,\text{bias}}\right)^T, \tag{24}$$

where w_a is the noise for the acceleration, w_ω is the noise for the rotation, and $S(\omega)$ a rotation matrix.

In order to simplify the dynamics and adapt the model to a more common 2D vehicle scenario, Model M1 is simplified. Basically a coordinated turn model is used, where only the longitudinal speed and acceleration are considered as states together with 2D position and a full orientation description with bias terms. This leads to 11 states in Model M2 which describes the vehicle positioning problem very well. The model can be improved by supporting it with information from the lateral accelerometer a_y (Model M2ay). If further reduction of the state vector is need biases terms can be considered as fixed parameters, which will only be updated at the initial alignment. Furthermore, IMU signals can be considered as input signals, yielding only 6 states. For details we refer to [20].

Fig 11 shows some comparative results from field trials in the same area as in the map of Fig 2. Fig 12 compares the performance over 10 different experiments on the same route. As seen the pure IMU model has worse performance than the odometric model. This is because it was hard to estimate the biases and perform a sufficiently good initial alignment, compensating for the gravitational vector. This can probably be improved, however since noisy signals are double integrated to yield position, it is a much tougher problem than integrating wheel speed signals once. As seen a sensor fusion between automotive sensors and IMU yields basically the same result as the odometric one.

From Fig. 11 and Fig. 12 we conclude the following:

- The wheel speed based dead-reckoning gives superior performance.
- Pure dead-reckoning of IMU cannot be used as a backup solution to GPS, unless the initial alignment and offset estimation is improved, see Fig. 11(b).
- Dynamic map matching based on dead-reckoning of IMU and cornering detection from lateral acceleromater is a feasible backup solution to GPS, see Fig. 11(b). However, the robustness is not convincing, see the first plot in Fig. 12.
- When wheel speed signals are available, also including IMU does not improve the result of dynamic map matching significantly, see Fig. 11(c-d) and the last two plots in Fig. 12. However, for some driving conditions this might be the case.

6 Tracking Approaches

Tracking road-bound vehicles is important in surveillance and certain applications in intelligent transportation systems. This section illustrate how this can be done using wireless radio measurements, microphone sensor networks, radars and airborne video cameras.

6.1 Radar Support

A radar system emits electromagnetic waves and analyzes the reflected waves to determine the range, direction, and radial speed of both moving and fixed objects. Both the initial Kalman filter based studies [23,35,36] and subsequent particle filter based approaches [1,31] on target tracking with road network information were motivated by radar applications. In this direction of research, extensive effort was spent for improving the methods [9, 24, 30, 33, 38, 41] especially for ground moving target indication (GMTI) which is a mode of operation of a radar system where the range rate (Doppler) is used to discriminate moving targets against stationary clutter. The probability of detection in GMTI systems depends not only on the environment topography, but also on the relative radial velocity of the target.

Let $x_k = (X_k, Y_k)^T$ be the position of the target relative a global Cartesian reference system. For simplicity, assume that the sensor is located at the origin. The GMTI observation model can be expressed as

$$y_k = h(x_k, u_k, e_k) = \begin{pmatrix} r_k \\ \theta_k \\ \dot{r}_k \end{pmatrix} + e_k = \begin{pmatrix} \sqrt{X_k^2 + Y_k^2} \\ \text{arctan}_2(Y_k, X_k) \\ \frac{X_k \dot{X}_k + Y_k \dot{Y}_k}{\sqrt{X_k^2 + Y_k^2}} \end{pmatrix} + e_k \tag{25}$$

where e_k is the measurement noise modeled as

$$e_k \sim \mathcal{N}\left(0, \text{diag}(\sigma_r^2, \sigma_\theta^2, \sigma_{\dot{r}}^2)\right). \tag{26}$$

This is just a 2D model, but it is straightforward to include the elevation angle to get a 3D description.

A basic simulation example is here given to show the advantages of using road network information when tracking a moving on-road target. The target is detected if the radial speed is above the minimum detectable velocity (MDV). False detections are assumed to be uniformly and independently distributed and the number of false detections is assumed to be Poisson distributed. A global nearest neighbor (GNN) association algorithm is used with standard gating and initiator logic. Snapshots from two GMTI road target tracking examples are shown in Fig. 13 where an off-road target model (left) and an on-road target model (right) are used, respectively. A particle filter is used in both cases and it is possible to see that an on-road model is advantageous since the resulting particle cloud is significantly denser, i.e., the variance is smaller. The advantage of using road information is even more obvious when the target is not detected, e.g. due to Doppler blindness, and the filter have to predict the target motion. More extensive analysis of the target tracking problem with GMTI can be found in [1], [41] and references therein. A radar system emits electromagnetic waves and analyzes the reflected waves to determine the range, direction, and radial speed of both moving and fixed objects. Ground moving target indication (GMTI) is a mode of operation of a radar system where the range rate (Doppler) is used to discriminate moving targets against stationary clutter. The probability of detection depends on the environment topography, but also on the relative radial velocity of the target.

6.2 Wireless Radio Network Support

There are a number of different measurement types that can be used to position a wireless network user. Most of the work focuses on the range measurements depending on time of arrival (TOA), time difference of arrival (TDOA) observations and received signal strength (RSS) observations, see [17] and the references therein. Among such alternatives, the RSS measurements, which do not require any additional hardware, are the most easily available. The RSS measurement might, on the other hand, be much more noisy than other type of measurements and prior information like the road maps proves to be

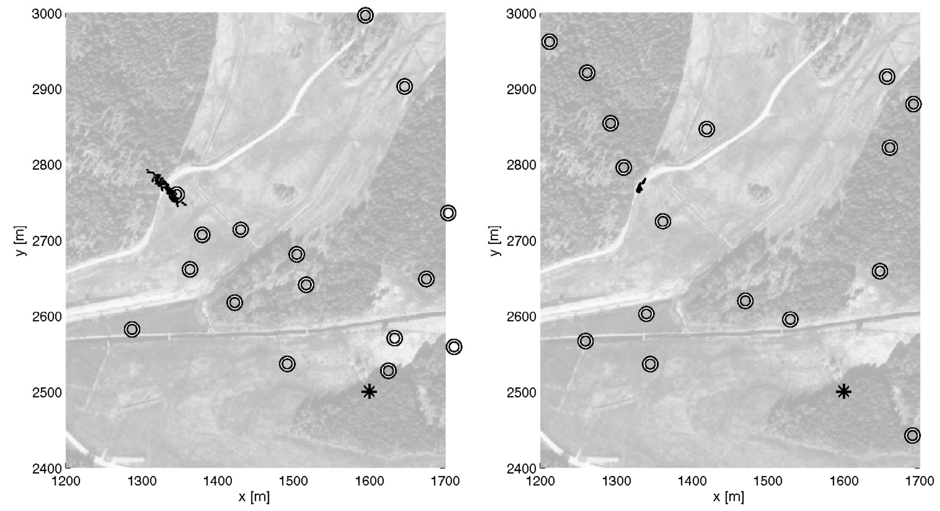


Fig. 13: Snapshots from two GMTI road target tracking examples with an off-road target model (left) and an on-road target model (right). The stationary radar sensor is located near the lower right corner and the circles indicate all detections, both false and true. A particle filter is used and the particles are represented by dots. An on-road model is advantageous since the resulting particle cloud is significantly denser, i.e., the variance is smaller. In particular, when the target is not detected due to Doppler blindness, the prediction of the target motion is better when using the road information.

crucial in obtaining a reasonable performance. The two most common models connecting the target range to the RSS measurements are the general exponential path loss model, which is known as Okumura-Hata model [19,42], and a dedicated power map constructed off-line for the region of interest.

The Okumura-Hata model says that the RSS value in a log power scale decreases linearly with the log-distance to the antenna i.e.,

$$z_k = P_{BS} - 10\alpha \log_{10}(\|p_{BS} - x_k^p\|_2) + e_k, \tag{27}$$

where z_k is the RSS measurement; x_k^p is the target position; P_{BS} is transmitted signal power (in dB); α is the path loss exponent; e_k is the measurement noise and p_{BS} is the position of the antenna (base-station (BS)) and the notation $\|\cdot\|_2$ denotes the standard ℓ_2-norm. This is quite a crude approximation, where the noise level is high and further depends on multi-path and non-line of sight (NLOS) conditions.

The second alternative is to determine the RSS values at discrete points in the area of surveillance and save this in a database. This can be done using off-line measurement campaigns, adaptively by contribution from users or using cell planning tools. The advantage of this effort is a large gain in signal to noise ratio and less sensitivity to multi-path and NLOS conditions. The set of RSS values that are collected for each position from various BSs is called the *fingerprint* for that location. The idea of matching observations of RSS to the map of the previously measured RSS values is known as *fingerprinting.* The set of all fingerprints forms an unconventional but informative measurement model and this can be used in localization.

With both alternatives, there are two possible approaches: static and dynamic localization. In the static approach, no assumptions about the target motion is made and for each measurement one generates a position estimate based on the corresponding measurement. In this case, the estimator is a static function of the input measurement. In the second approach, one can use a motion model for the target behavior along with an estimator (which is a dynamic function of the measurements) sequentially updating the estimates with the incoming measurements. Since the information in different measurements are fused by means of the motion model with such an approach, much better accuracy is achievable.

In this section, we present the results obtained in the example study presented in [5] where WiMAX RSS data from three sites and seven BSs was collected in the urban area (in Brussels) shown in Fig. 14. The collected data was saved in a database and used to locate the target based on a separate test data. The following five approaches were used to localize the target.

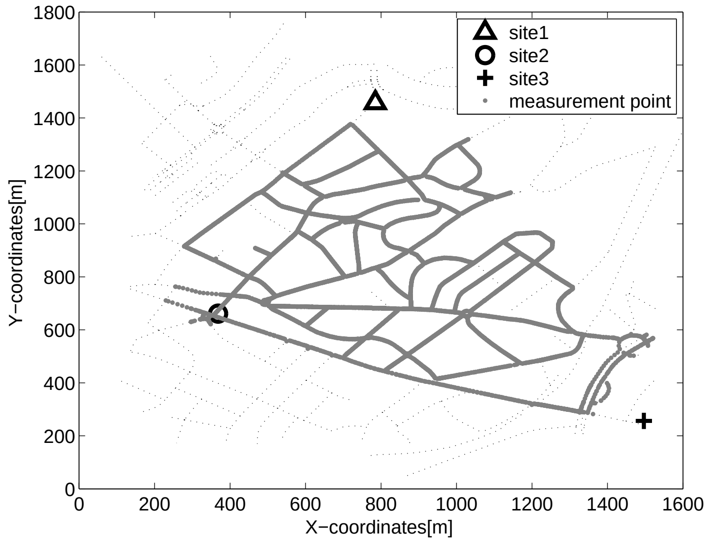
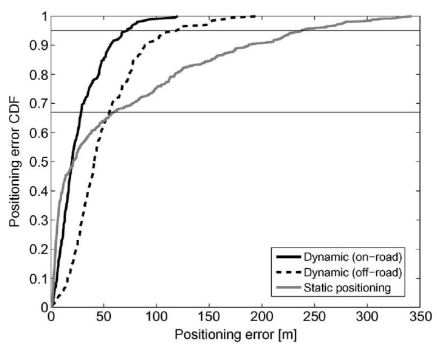
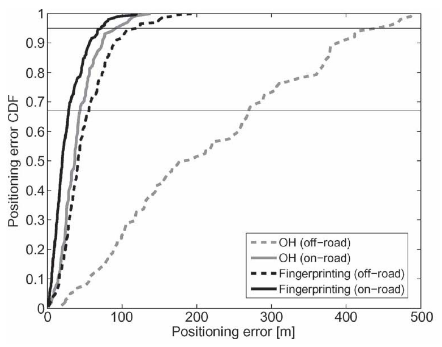


Fig. 14: MAP of the area under study, sites with base stations and the measurement locations.

- **Static Positioning:** The test data is used in a static manner as described above to locate the target. Basically, the position of the closest fingerprint to the collected measurement becomes the estimate.
- **Dynamic–OH–off-road:** A particle filtering based approach with an off-road (i.e., no road-map information) target motion model and the OH-model of (27) as the measurement model.
- **Dynamic–OH–on-road:** A particle filtering based approach with a road-constrained target motion model and the OH-model of (27) as the measurement model.
- **Dynamic–Fingerprinting–off-road:** A particle filtering based approach with an off-road (i.e., no road-map information) target motion model and the fingerprint database as the measurement model. See the details on how to utilize the fingerprints in the measurement likelihood calculation in [4].
- **Dynamic–Fingerprinting–on-road:** A particle filtering based approach with a road-constrained target motion model and the fingerprint database as the measurement model. See the details on how to utilize the fingerprints in the measurement likelihood calculation in [4].



(a) Dynamic vs. Static Estimation



(b) OH-model vs. Fingerprinting

Fig. 15: Wireless network supported tracking

The results in the form of the cumulative distribution functions of the position estimation errors are shown in Fig. 15 in two plots. The dynamic and static estimation results are compared in Fig. 15a. A definite advantage of the dynamic approaches seen even without the road-map information with 95% probability though in rare events (below 0.67%) static estimation can sometimes get better results. On average, the road information in the dynamic case seems to result in about 25m's better accuracy. Fig. 15b compares the OH-based and the fingerprinting based methods. Without the road constraints, the simple OH-model behaves much worse than the fingerprinting based methods with positioning errors above 400m's (95% line). However, when the on-road model is utilized, the accuracy can reach up to 100m's (95% line) which is even better than the off-road fingerprinting case. Hence, the road map information is capable of making even the crudest measurement models behave as good as sophisticated ones.

6.3 Sensor Network Support

This section considers the problem of localizing an unknown number of targets around an acoustic sensor network. Since acoustic power is additive at each sensor, the RSS from different sources cannot be resolved and the framework is difficult to extend to multiple target tracking. In practice, the exponential signal decay rate implies that the closest target will dominate each sensor observation. One applicable approach is to consider each sensor as a binary proximity sensor as studied in for instance [3]. However, this requires an excessive amount of sensors to get accurate multi-target tracking (MTT). Another problem associated with the RSS measurements is that the emitted acoustic powers from the targets are unknown and must be estimated along with the target states. For these reasons, MTT ideas with acoustic sensors appeared in the literature with either direction of arrival (DOA) [7,8,11,12] or time difference of arrival (TDOA) [28] measurements. The power (and/or energy) based measurements case was also examined with few examples in [6,37] which assume that either the number of targets or the emitted powers are known.

When the road information is supplied, the problem can be tackled much more easily. We here summarize results in [29]. The map of the area under study is shown in Fig. 16 along with the road information and microphone positions. The onroad position coordinates p_η are marked in the figure at each 50 meters. We have 10 microphones collecting data at 4kHz placed around the road. Each microphone position is illustrated with a cross sign in Fig. 16.

The synchronized recordings of a motorcycle and a car are used while the correct positions are measured with GPS sensors. The correct positions of the targets projected onto the road coordinates are shown in Fig. 17a. The microphone network is also illustrated in Fig. 17a with cross signs at $t = 0$ denoting the closest onroad point to each microphone. The recordings for the motorcycle and the car were obtained separately and we obtain our two-target data by adding the sound waveforms for the two cases. The onroad position coordinates are discretized uniformly with 5m distance between adjacent points and the point mass filter based Emitted Power Density (EPD) filter of [29] is run

1. For car's sound data only;
2. For motorcycle's sound data only;
3. For the superposed sound data of the car and the motorcycle.

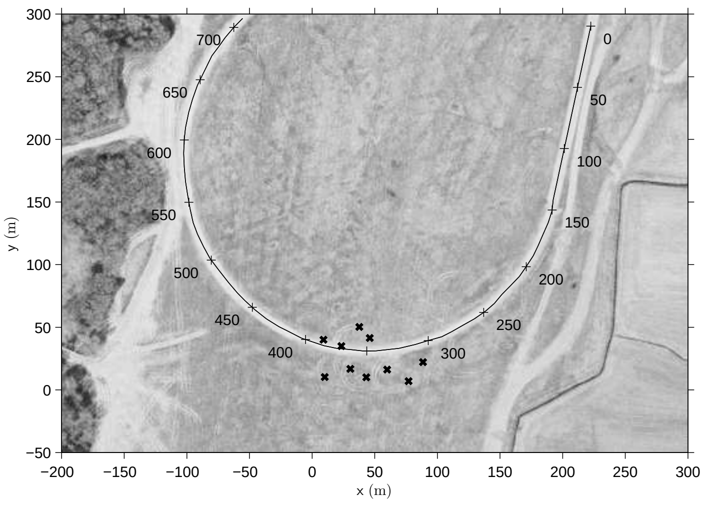


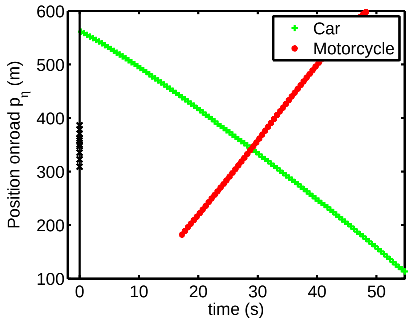
Fig. 16: The map of the area, road segment, microphones and coordinates used in the example. The distance markings on the road segment denote the onroad position coordinates. Microphone positions are illustrated with cross signs.

The resulting position estimates obtained are illustrated in Fig.s 17b, 17c and 17d respectively. Single target detection and tracking seem to be good except for some occasional missing detections in the motorcycle only case. In the two target case, the target initiation delays a little and target loss happens a little earlier. However, both targets can be tracked quite similarly to the single target cases.

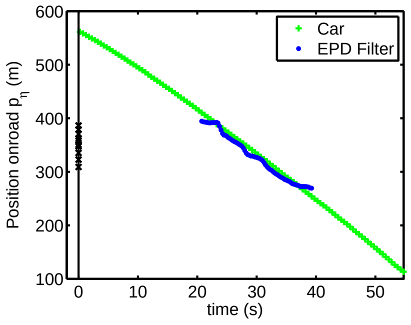
6.4 Vision Support

Vision sensors are bearings-only sensors providing the azimuth and inclination to the target relative the sensor platform. A vision sensor is here defined as a staring-array electro-optical/infrared sensor (EO/IR) with limited field-of-view (FOV). Let $p = (X, Y, Z)^T$ be the 3D position of the target relative a global Cartesian reference system. For simplicity, assume that the sensor is located at the origin. An observation at time t is the relative angles between the sensor and the target, i.e.,

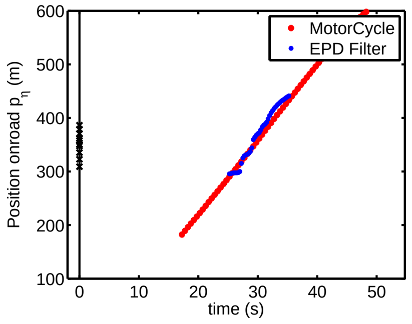
$$y_k = h(x_k, u_k, e_k) = \begin{pmatrix} \phi_k \\ \theta_k \end{pmatrix} + e_k = \begin{pmatrix} \arctan_2(Y_k, X_k) \\ \arctan_2(Z_k, \sqrt{X_k^2 + Y_k^2}) \end{pmatrix} + e_k, \quad (28)$$



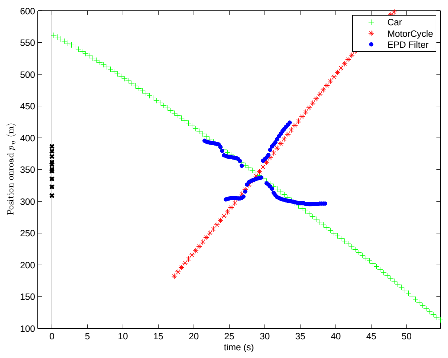
(a) GPS data



(b) Car



(c) Motorcycle



(d) Both Car and Motorcycle

Fig. 17: The correct onroad positions of the two targets and the estimation results. The closest onroad point to each microphone is also illustrated with cross signs at $t = 0$.

where e_k is the measurement noise modeled as

$$e_k \sim \mathcal{N}\left(0, \sigma^2 I_{2\times 2}\right). \tag{29}$$

A measurement y_k is obtained by transforming a detection at an image point $(u \;\; v)^T$ to azimuth and inclination angles given the knowledge of the sensor orientation. For an ideal vision sensor, a point $p^b = (X^b \;\; Y^b \;\; Z^b)^T$, expressed in Cartesian coordinates relative the camera fixed reference system, is projected in a virtual image plane onto the image point $(u \;\; v)^T$ according to the ideal perspective projection formula

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{f}{Z^b} \begin{pmatrix} X^b \\ Y^b \end{pmatrix} \tag{30}$$

where f is the focal length of the camera. However, in practice the intrinsic parameters of the vision sensor model must be estimated to handle lens distortion etc.

Prior information about the environment, like the road network, will improve the target tracking performance significantly. In particular, in the vision sensor case, the problem is not fully observable if the sensor is stationary. However, as the example below indicates, there is more prior information apart from the road network that can support the estimation process. In this simulation example a single car is tracked by a UAV equipped with a camera. The simulation environment and the path of the car are shown in Fig. 18. The sensor platform is flying in a circle with a radius of 100 meters and approximately 100 meters above the ground, the approximate sensor view is shown Fig. 19. An observation is the azimuth and inclination angles obtained from a detection.

The results from three different target tracking filters are shown in Fig. 19 and Fig. 20. The first filter is a standard bootstrap PF that assumes that the car will always be on the known road network manifold (this filter is called "on-road PF"). The second filter is a bootstrap PF based on a coordinated-turn like model that is not using the road network information (this filter is called "off-road PF"). The third filter is a multiple model PF (MMPF) with two sub-filters, one sub-filter identical to the on-road PF and one sub-filter identical to the off-road PF (this filter is called "on/off-road MMPF"). All filters have 1000 particles in total. The root mean square position errors for 100 simulation runs are shown in Fig. 20. The car is occluded behind a building between 12-17 [s] and the errors grow due to that, especially the off-road PF have serious problem here. The car is rediscovered, but after about 18 [s] the car enters a parking lot that is not part of the road network model. Hence the on-road PF diverge, but the MMPF and off-road PF can handle that mode change. In Fig. 20 the on-road mode probability of the MMPF is shown. When the car is on the parking lot the on-road probability is very small.

Knowledge about the buildings and vegetation is also very useful to be able to draw conclusions from non-detection [38]. In the current example the car moves through an intersection just before it is occluded by a building, see Fig. 19. A target tracking filter that is not using this so called "negative information" will spread its particles on both roads since no detections are received. However, a filter that utilizes the negative information would discard the particles on the visible road segment, since if the car would have been on that road segment it should have been detected.

Even though this example is a simulation where some problems, such as navigation error and multiple targets are neglected, it is still possible to draw some general conclusions. Using road network data as prior information will improve the target tracking performance for sensor vision applications. Especially when the motion of the sensor platform is rather limited since bearings-only tracking performance depends very much on the movement of the sensor platform. The road information is also very useful to predict the target motion in the case of non-detection, for instance due to occlusion. However, algorithms that rely much on prior information should always be used with a fail-safe algorithm that can take over when the prior information is wrong. If the navigation error is slowly varying the measurements will be biased and may cause major problems for the filter that uses the road network as state space manifold, especially when the targets are close to intersections. Using "negative information" is a conceptually simple thing to do to increase the performance in environments and situations where the probability of detection varies. The gain of negative information is more obvious when using it in the road network context.

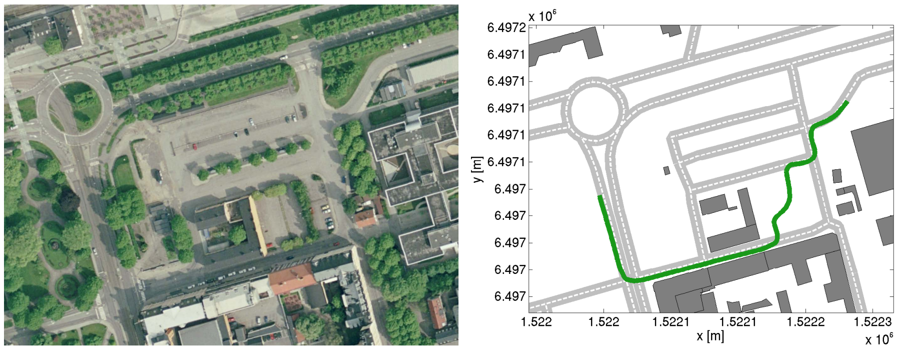


Fig. 18: Left: The simulation environment [32]. Right: The path of the car, driving from left to right. The sensor platform is flying north of this area.

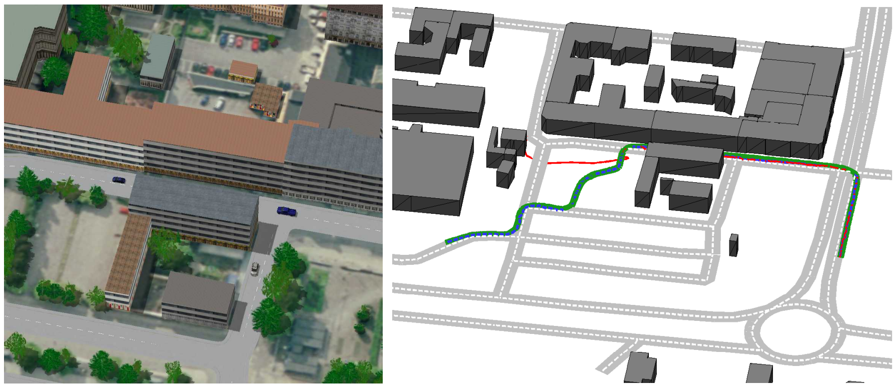


Fig. 19: Left: The car to the right of the building will soon be occluded. Right: The filter results. The MMPF (blue) and the off-road PF (black) are hard to discriminate from the ground truth (green). The PF (red) is diverging when the car is off-road.

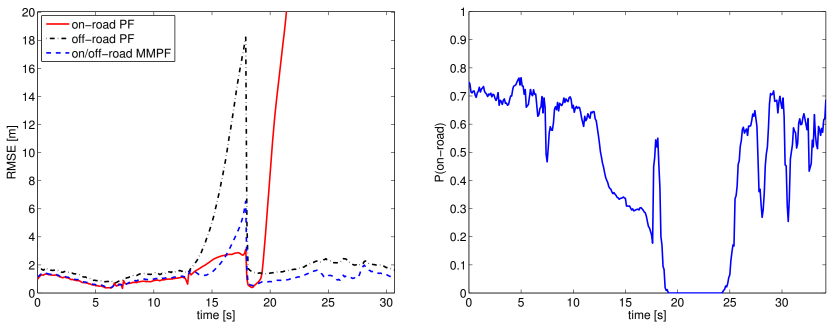


Fig. 20: Left: The RMSE results. Right: on-road mode probability in the MMPF.

7 Conclusions

Map matching is an appealing approach to road-assisted navigation when accurate position information is available, for instance from GNSS (global navigation satellite systems), see Chapter *In-Car Navigation Basics* for a survey. Without a reliable position sensor, the navigation problem is more challenging, and a kind of dynamic map matching is needed that takes both a motion model of the vehicle and the topology of the map into account. We have surveyed state of the art algorithms for road-assisted navigation and tracking using the following main principles for how to incorporate the road-constraint into a filtering framework:

- The road constraint is included as a virtual measurement. This fits the particle filter algorithm well, where the measurement update corresponds to multiplying each weight with a scalar that depends on the distance to the closest road point. The advantage is its simplicity. The disadvantage is the potentially poor particle efficiency, where a large number of positions end up outside the road network.
- The road constraint is converted into a direction process noise that projects the state back to the road network. This is a commonly used approach in tracking applications such as GMTI (ground moving target indicator). The advantages are that it fits a Kalman filter framework, and that it provides better particle efficiency in a particle filter than the virtual measurement approach. The disadvantage is that the mathematical operation to generate such noise is quite complex, and that ad-hoc approximations may be needed.
- The road network is interpreted as a manifold, where a discrete state is used to represent the road network between junctions, and a continuous state variable represents the one-dimensional position between the junctions. The advantage is that this approach utilizes all information in an efficient way. The disadvantage is a more complex algorithm.

A wide range of sensor combination and performance indicators were presented.

The position estimate from dynamic map matching can never be more accurate than the road map itself, and commercial maps are always subject to small deviations from reality. For navigation purposes, this does not pose any problems. For advanced driver assistance systems (ADAS), the position relative to the road and both stationary and dynamic obstacles are needed, and this is the subject of Chapter *Situational awareness and road prediction for trajectory control applications.*

8 Acknowledgement

This work has been supported by the Swedish Research Council (VR) under the Linnaeus Center CADICS, the VR project grant *Extended Target Tracking*, the SSF excellence center MOVIII, the Vinnova excellence center FOCUS and the strategic research environments ELLITT in the ICT area and Security Link in the security area.

The authors would also like to thank NIRA Dynamics, SAAB EDS and FOI for initiating a series of stimulating master thesis in this area. Special thanks to Mussa Bshara for collecting the WiMAX data in Section 6.2.

References

1. M. S. Arulampalam, N. Gordon, M. Orton, and B. Ristic. A variable structure multiple model particle filter for GMTI tracking. In *Proceedings of International Conference on Information Fusion*, volume 2, pages 927–934, July 2002.
2. H.A.P. Blom and Y. Bar-Shalom. Interacting multiple model algorithm for systems with Markovian switching coefficients. *IEEE Transactions on Automatic Control*, 33(8):780–783, 1988.
3. Y. Boers, H. Driessen, and L. Schipper. Particle filter based sensor selection in binary sensor networks. In *Prooceedings of the 11th International Conference on Information Fusion*, 2008.
4. M. Bshara, U. Orguner, F. Gustafsson, and L. VanBiesen. Fingerprinting localization in wireless networks based on received signal strength measurements: A case study on wimax networks. 2010. www.control.isy.liu.se/ fredrik/reports/09tvtmussa.pdf.
5. M. Bshara, U. Orguner, F. Gustafsson, and L. VanBiesen. Robust tracking in cellular networks using hmm filters and cell-id measurements. 2010.
6. M.F. Bugallo and P.M. Djuric. Tracking of time-varying number of moving targets in wireless sensor fields by particle filtering. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 4, pages 865–868, May 2006.
7. V. Cevher, A.C. Sankaranarayanan, J.H. McClellan, and Rama Chellappa. Target tracking using a joint acoustic video system. *IEEE Trans. Multimedia*, 9(4):715–727, June 2007.
8. V. Cevher, R. Velmurugan, and J.H. McClellan. Acoustic multitarget tracking using direction-of-arrival batches. *IEEE Trans. Signal Process.*, 55(6):2810–2825, June 2007.
9. Yang Cheng and T. Singh. Efficient particle filtering for road-constrained target tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 43(4):1454–1469, October 2007.
10. ESRI. ESRI shapefile technical description – an ESRI white paper. In *http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf*, July 1998.
11. Maurice Fallon and Simon Godsill. Multi target acoustic source tracking using track before detect. In *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 102–105, October 2007.
12. Maurice Fallon and Simon Godsill. Multi target acoustic source tracking with an unknown and time varying number of targets. In *Prooceeding of the Conference on Hands-Free Speech Communication and Microphone Arrays (HSCMA)*, pages 77–80, May 2008.
13. N.J. Gordon, D.J. Salmond, and A.F.M. Smith. A novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings on Radar and Signal Processing*, volume 140, pages 107–113, 1993.
14. F. Gustafsson. Rotational speed sensors: Limitations, pre-processing and automotive applications. *IEEE Instrumentation and Measurement Magazine*, 13(2):16–23.
15. F. Gustafsson. Particle filter theory and practice with positioning applications. *IEEE Transactions on Aerospace and Electronics Magazine Part II: Tutorials*, 7(July):53–82, 2010.
16. F. Gustafsson. *Statistical Sensor Fusion*. Studentlitteratur, 2010.
17. F. Gustafsson and F. Gunnarsson. Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements. *IEEE Signal Processing Magazine*, 22:41–53, 2005.
18. P. Hall. A Bayesian approach to map-aided vehicle positioning. Master's Thesis LiTH-ISY-EX-3102, Department of Electrical Engineering. Linköping University, Linköping, Sweden, 2001.
19. M. Hata. Empirical formula for propagation loss in land mobile radio services. *IEEE Transactions on Vehicular Technology*, 29(3):317–325, 1980.

20. G. Hedlund. Map aided positioning using an inertial measurement unit. Master's Thesis LiTH-ISY-EX-4196, Department of Electrical Engineering. Linköping University, Linköping, Sweden, 2008.
21. S.J. Julier, J.K. Uhlmann, and Hugh F. Durrant-Whyte. A new approach for filtering nonlinear systems. In *IEEE American Control Conference*, pages 1628–1632, 1995.
22. R.E. Kalman. A new approach to linear filtering and prediction problems. *J Basic Engr. Trans. ASME Series D*, 82:35–45, 1960.
23. T. Kirubarajan, Y. Bar-Shalom, K. R. Pattipati, and I. Kadar. Ground target tracking with variable structure IMM estimator. *IEEE Trans. Aerosp. Electron. Syst.*, 36(1):26–46, January 2000.
24. J. Koller and M. Ulmke. Data fusion for ground moving target indicator. *Aerospace Science and Technology*, 11:261–270, 2007.
25. S.C. Kramer and H.W. Sorenson. Recursive Bayesian estimation using piece-wise constant approximations. *Automatica*, 24:789–801, 1988.
26. J. Kronander. Map aided positioning using an inertial measurement unit. Master's Thesis LiTH-ISY-EX-3578, Department of Electrical Engineering. Linköping University, Linköping, Sweden, 2004.
27. X. R. Li and Y. Bar-Shalom. Multiple-model estimation with variable structure. *IEEE Trans. Autom. Control*, 41(4):478–493, April 1996.
28. Wing-Kin Ma, Ba-Ngu Vo, S.S. Singh, and A. Baddeley. Tracking an unknown time-varying number of speakers using TDOA measurements: A random finite set approach. *IEEE Trans. Signal Process.*, 54(9):3291–3304, September 2006.
29. U. Orguner and F. Gustafsson. Multi target tracking with acoustic power measurements using emitted power density. In *Proceedings of 13th International Conference on Information Fusion (FUSION '10)*, July 2010.
30. O. Payne and A. Marrs. An unscented particle filter for GMTI tracking. In *Proceedings of Aerospace Conference*, volume 3, pages 1869–1875, March 2004.
31. B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, chapter 10. Artech House, London, 2004.
32. J. Rydell, G. Haapalahti, J. Karlholm, F. Näsström, P. Skoglar, K.-G. Stenborg, and M. Ulvklo. Autonomous functions for UAV surveillance. In *International Conference on Intelligent Unmanned Systems (ICIUS)*, November 2010.
33. David Salmond, Martin Clark, Richard Vinter, and Simon Godsill. Ground target modelling, tracking and prediction with road networks. In *International Conference on Information Fusion*, July 2007.
34. S.F. Schmidt. Application of state-space methods to navigation problems. *Advances in Control Systems*, pages 293–340, 1966.
35. P. J. Shea, T. Zadra, D. Klamer, E. Frangione, and R. Brouillard. Improved state estimation through use of roads in ground tracking. In *Proceedings of Signal and Data Processing of Small Targets*, volume 4048, pages 312–332. SPIE, 2000.
36. P. J. Shea, T. Zadra, D. Klamer, E. Frangione, and R. Brouillard. Precision tracking of ground targets. In *Proceedings of Aerospace Conference*, volume 3, pages 473–482. IEEE, 2000.
37. X. Sheng and Y.-H. Hu. Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks. *IEEE Transactions on Signal Processing*, 53(1):44–53, 2005.
38. P. Skoglar, U. Orguner, D. Törnqvist, and F. Gustafsson. Road target tracking with an approximative Rao-Blackwellized particle filter. In *12th International Conference on Information Fusion*, 2009.
39. H.W. Sorenson and D.L. Alspach. Recursive Bayesian estimation using Gaussian sum. *IEEE Transactions on Automatic Control*, 17:439–448, 1972.
40. N. Svenzén. Real time map-aided positioning using a Bayesian approach. Master's Thesis LiTH-ISY-EX-3297, Department of Electrical Engineering. Linköping University, Linköping, Sweden, 2003.

41. M. Ulmke and W. Koch. Road-map assisted ground moving target tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 42(4):1264–1274, October 2006.
42. Y.Okumura, E. Ohmori, T. Kawano, and K. Fukuda. Field strength and its variability in VHF and UHF land-mobile radio service. *Review of the Electrical Communication Laboratory*, 16(9-10), 1968.