Copyright Notice

Probabilistic Localization Method for Trains

Oliver Heirich, Patrick Robertson, Adrián Cardalda García, Thomas Strang, Andreas Lehner

***Abstract*— The localization of trains in a railway network is necessary for train control or applications such as autonomous train driving or collision avoidance systems. Train localization is safety critical and therefore the approach requires a robust, precise and track selective localization. Satellite navigation systems (GNSS) might be a candidate for this task, but measurement errors and the lack of availability in parts of the railway environment do not fulfill the demands for a safety critical system. Therefore, additional onboard sensors, such as an inertial measurement unit (IMU), odometer and railway feature classification sensors (e.g. camera) are proposed. In this paper we present a top-down train localization approach from theory. We analyze causal dependencies and derive a general Bayesian filter. Furthermore we present a generic algorithm based on particle filter in order to process the multi-sensor data, the train motion and a known track map. The particle filter estimates a topological position directly in the track map without using map matching techniques. First simulations with simplified particular state and measurement models show encouraging results in critical railway scenarios.**

I. INTRODUCTION

Train localization systems are required for general train control, automatic train driving or collision avoidance. Current developments of a collision avoidance system for trains, named RCAS[1] [1], requires a robust and precise localization solution. Train localization can either be achieved by onboard sensors [2], [3], [4], or sensors with additional infrastructure in the railway environment, such as balises on tracks [5]. We focus on the onboard train localization method, where sensor data is combined with a track map and a train motion model. Onboard train sensors are global navigation satellite system receivers (GNSS), inertial measurement units (IMU), odometry and railway feature classification sensors [6], [7].

In contrast to map matching methods [2], [3], our approach is based on a particle filter, which spreads particles exclusively in the topological domain. We provide a top-down approach from theory, derive a general Bayesian filter as well as a generic particle filter for train localization. We propose the particle filter because railway topology requires a highly nonlinear transition model. We present encouraging results in situations with parallel tracks and switches from simulations of a particular implementation.

We discuss general map-based train localization and state of the art train localization sensors in Sec. II. We analyze the causal effects of physics for probabilistic train localization (Sec. IV) and derive a general Bayesian filter (Sec. V). A generic particle filter implementation is derived in Sec. VI and the algorithm is given in (Sec. VII). Sec. VIII contains a particular implementation with simplified models and simulation results.

O. Heirich, P. Robertson, A. Cardalda García, T. Strang and A. Lehner are with DLR (German Aerospace Center), Institute of Communications and Navigation, DLR e.V., Weßling, Germany {firstname.lastname}@dlr.de

[1]Railway Collision Avoidance System, http://www.collision-avoidance.org

II. TRAIN LOCALIZATION

The goal of train localization is to estimate the train position in the track network by topological coordinates. This topological position is defined by a unique track ID R and a track length variable s. The origin of that length has to be defined for direction dir of the train related to the track. A positive direction points away from the origin, a negative towards the origin. The topological pose is a triplet of track ID, length and direction and defines the train position and attitude in topological coordinates unambiguously:

$$\mathrm{P}^{\text{topo}} = \{R, s, dir\}. \quad (1)$$

The challenge of a map-based train localization is reliable track selective localization with availability in scenarios where parallel tracks and switches occur.

A. Train Localization Sensors

1) GNSS: Satellite navigation allows the absolute determination of positions in the geographic coordinate system without additional sensors and special prior knowledge about the actual position. The two main drawbacks of GNSS for railways are limited availability and relative low accuracy. An ideal condition for GNSS positioning is a direct line of sight from the antenna to the available satellites with no other objects in the vicinity. In the railways environment, this condition is not always given in tunnels, below bridges and station roofs and in dense forests where signals are blocked. In these scenarios, GNSS positioning may not be available or at least has a decreased accuracy due to the disadvantageous constellation of the visible satellites. The accuracy is also reduced by multipath effects which may be present over long distances. The railways environment contains a large amount of metallic structure such as tracks and power lines and the receiving antenna will in proximity to buildings, trees and other obstacles close to the track. A robust and precise localization can not be achieved by using GNSS only. A critical but frequent railways localization problem is the presence of parallel tracks. The distances of these parallel tracks are usually smaller than the GNSS accuracy. The limited availability of GNSS, the accuracy and the inability to measure in the topological domain do not cover the demands for a safety-of-life system such as collision avoidance.

2) IMU: An advantageous and complementary sensor is the inertial measurement unit (IMU). It combines the measurement of 6 different sensors by measuring three-dimensional translative accelerations and three-dimensional rotation turn rates of the train. The IMU provides continued data at a certain frequency. Inertial measurements suffer from errors such as sensor biases which are not constant over time and called drift. One conventional approach is the computation of attitude, speed and position [8] of a train. Therefore, the measurements are integrated once or twice. The drift causes high errors in the results because the errors sum up in every integration step. A different approach to apply an IMU is the measurement of the track's effects on the pose of the train. In previous work [9] we analyzed the position dependent geometric track characteristics measured by an IMU while the train is moving. The curvature of a track is stored as track geometry in a digital map. A train position with the known geometry from the map is compared with the measurements. As a result, the best matching positions in the track network can be derived. The big advantage of that method is the lack of integrations.

3) Odometer: The odometer measures relative distance and velocity of the train, by counting wheel increments and measuring the period. Odometers suffer from errors of the velocity or relative distance measurements due to wheel slip in acceleration and deceleration phases and due to a changing wheel radius of worn tread [10]. Doppler radars for trains provide information about train velocity and displacement based on a different sensing method and errors occur due to different rail target conditions. By integrating the velocity for travel distance measurement, these errors sum up.

4) Feature classification sensors: The feature sensors measure and classify a specific high-level characteristic of the railway environment. A position can be derived in combination with feature information from the digital map. An important feature sensor is the switch detector, which is able to recognize switches. These detectors are often advanced with a switch way detector, which determines the travel direction towards the switch and the track way. Other classifiers may recognize parallel tracks, platforms, station roofs, railway signs and signals, power masts, bridges or tunnel entrances. These classifiers can be built up by camera based vision sensor [6] or magnetic sensor based on eddy current [4]. An eddy current sensor works as a metal detector and senses changes in the metallic structure of the tracks with its coils. Feature classification sensors suffer from errors like missed detection or false detection.

B. Track Topology

Tracks are connected by switches, crossings or diamond switch crossings. A track R is defined between connections, i.e. it contains no switch or crossing. This definition ensures, that a track R is always true one-dimensional with no other access than the track begin or track end. A switch connects three tracks, a crossing four tracks. According to the travel direction and switch way position, a track splits up into two other tracks when passing a switch facing and two tracks merge into one track by passing trailing.

C. Track Geometry

As railway tracks are fixed to the earth, any position on the tracks represent as well an absolute geographic position. The geometry of a track at a certain position is given by the attitude and the changes of the attitude over position. The track attitude contains heading ψ, bank ϕ for the lateral inclination and slope θ for the longitudinal inclination. The changes of the attitude over track position are curvature $\frac{d\psi}{ds}$, bank change $\frac{d\phi}{ds}$ and slope change $\frac{d\theta}{ds}$. A train will be forced to take the designated geographic positions and geometric track features while being on a certain track position. According to the train traveling direction, the signs of the attitude and its derivations are changing.

III. Train Localization Model

A. Train State

We define a train state P_k, which contains a discrete data set indexed by the discrete time step k. The data is topological pose $\mathrm{P}_k^{\text{topo}}$, geographic position $\mathrm{P}_k^{\text{geo}}$, the attitude $\mathrm{P}_k^{\text{att}}$ by the attitude angels (ϕ_k, θ_k, ψ_k) and $\mathrm{P}_k^{\text{turn}}$ is the turn velocity of the train ($\dot{\phi}_k$, $\dot{\theta}_k$, $\dot{\psi}_k$). The train state is:

$$\mathrm{P}_k = \{\mathrm{P}_k^{\text{topo}}, \mathrm{P}_k^{\text{geo}}, \mathrm{P}_k^{\text{att}}, \mathrm{P}_k^{\text{turn}}\} = \{R_k, s_k, dir_k, lat_k, long_k, alt_k, \phi_k, \theta_k, \psi_k, \dot{\phi}_k, \dot{\theta}_k, \dot{\psi}_k\}. \tag{2}$$

B. Train Motion

The train motion is one-dimensional due to constrains of the rail tracks and we define the motion state U_k by along track acceleration $\ddot{s}_k$, velocity $\dot{s}_k$ and a traveled distance Δs_k in the time between $k-1$ and k:

$$\mathrm{U}_k = (\Delta s_k, \dot{s}_k, \ddot{s}_k)^\top. \tag{3}$$

The train motion model computes the transition in the time between $k-1$ and k of Δs_k, $\dot{s}_k$, $\ddot{s}_k$ and the process noise $\nu_k^{\ddot{s}}$. The motion transition function f_{motion} is defined by:

$$\mathrm{U}_k = f_{\text{motion}}(\mathrm{U}_{k-1}, \nu_k^{\ddot{s}}) = \begin{pmatrix} \dot{s}_{k-1}\Delta t + \ddot{s}_{k-1}\frac{\Delta t^2}{2} \\ \dot{s}_{k-1} + \ddot{s}_{k-1}\Delta t \\ \ddot{s}_{k-1} + \nu_k^{\ddot{s}} \end{pmatrix}. \tag{4}$$

Further, the output of this train motion model is limited by train specific parameters, such as maximum velocity, acceleration and deceleration.

C. Digital Map

The digital map contains position relevant information for the localization procedure. The map contains information of topological, geographic and track geometric data and further information about the track net, track IDs and connections such as switches, crossings or dead endings. The position of a train standing on a certain track can be described either by topological or geographic position. A reasonable representation for railway navigation is the topological coordinate frame. As most sensors cannot directly measure in the topological system, the map contains additionally geographic and geometric track representation. The geographic and geometric information of a particular position is provided by the digital map and referenced by the topological pose:

$$\left.\begin{array}{l}\text{geographic track position}\\ \text{track geometry}\end{array}\right\} = f_{\text{map}}(\mathrm{P}_k^{\text{topo}}). \tag{5}$$

In practice, the map is organized by a list of tracks. Every track contains a unique track ID R, connections and track data parametrized to the one-dimensional position s of the track. The geographic and geometric track data is stored by supporting points of any kind of continuous graph representation. There are many methods of representing a continuous function by discrete points. The simplest might be the polygonal line approximation, which interconnects points with lines. More advanced methods for the geographic track representation use spline approximations [11].

D. Transition Model

Trains move exclusively on railway tracks. The digital map contains information about these tracks and therefore the transition model includes the map. A train state transition happens at every time step k in the topological coordinate frame. The new topological pose $\mathrm{P}_k^{\text{topo}}$ is calculated from the previous pose $\mathrm{P}_{k-1}^{\text{topo}}$ and the traveled distance Δs_k:

$$\mathrm{P}_k^{\text{topo}} = f_{\text{map}}(\mathrm{P}_{k-1}^{\text{topo}}, \Delta s_k). \tag{6}$$

At any switch passing facing for example, the transition gets nonlinear and splits in two possibilities. Afterwards, the remaining train states are updated with information from the digital map:

$$\{\mathrm{P}_k^{\text{geo}}, \mathrm{P}_k^{\text{att}}, \mathrm{P}_k^{\text{turn}}\} = f_{\text{map}}(\mathrm{P}_k^{\text{topo}}, \dot{s}_k). \tag{7}$$

The train turn rates P^{turn} are calculated from track geometry and velocity by [9]:

$$\dot{\phi} = \frac{d\phi}{dt} = \frac{d\phi}{ds}\frac{ds}{dt} = \frac{d\phi}{ds}\dot{s}, \tag{8}$$

$$\dot{\theta} = \frac{d\theta}{ds}\dot{s}, \tag{9}$$

$$\dot{\psi} = \frac{d\psi}{ds}\dot{s}. \tag{10}$$

Finally, Eq. (6) and Eq. (7) are combined to one transition function:

$$\mathrm{P}_k = f_{\text{map}}(\mathrm{P}_{k-1}, \mathrm{U}_k). \tag{11}$$

IV. Probabilistic Train Localization

A probabilistic approach computes the estimates of the train state $\mathbf{P}_k$, which contains all states a train can take at the time step k. We denote $\mathbf{U}_k$ as the one dimensional train motion. The sensor measurements of time k are described by $\mathbf{Z}_k$ and $\mathbf{E}_k$ are measurement errors which are persistent over at least two time steps. Examples are multipath errors of the GNSS, drift of inertial sensors, calibration errors and wheel slip of the odometer or feature sensors. White Gaussian sensor noise is independent over time and not part of $\mathbf{E}_k$. Finally, the physical railway environment including the track network is time invariant and denoted by $\mathbf{M}$.

A. Dynamic Bayesian Network

A dynamic Bayesian network (DBN) visualizes causal dependencies of effects in a directed acyclic graph [12]. In Fig. 1 we draw the DBN for two time steps of a train, which is equipped with erroneous sensors and the causal dependencies are shown by pointers. From the DBN we

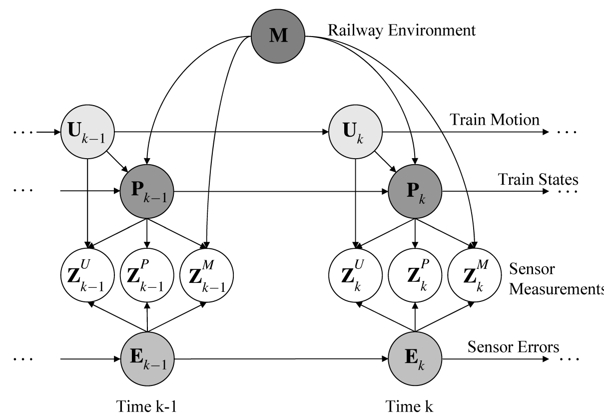

Fig. 1. Dynamic Bayesian network

can see, that the train state $\mathbf{P}_k$ is directly dependent to the last train state $\mathbf{P}_{k-1}$, the 1D train motion $\mathbf{U}_k$ and the railway environment $\mathbf{M}$. The railway environment is time invariant and has a strong influence on the train states caused by the tracks. The dependency of $\mathbf{P}_k$ to $\mathbf{P}_{k-1}$ and $\mathbf{U}_k$ to $\mathbf{U}_{k-1}$ can be explained by physical effects. A train can not change its position, speed or attitude randomly between two time steps because of the inertia of train mass and limited speed, acceleration and turn rates. The train has onboard sensors for localization purposes, which measure physical characteristics of the train states. The measurements $\mathbf{Z}_k$ are directly dependent to the train states $\mathbf{P}_k$ and independent from the last measurements.

B. Multiple Sensor Measurements

The measurements are split into three different groups $\mathbf{Z}_k^{\text{U}}$, $\mathbf{Z}_k^{\text{P}}$ and $\mathbf{Z}_k^{\text{M}}$. $\mathbf{Z}_k^{U}$ is dependent to train motion, the train state and to sensor errors and refers to acceleration and turn rate measurements of the IMU ($\mathbf{Z}^{\text{IMU}}$) and to velocity measurements by the odometer ($\mathbf{Z}^{\text{odo}}$):

$$\begin{aligned} p(\mathbf{Z}_k^{\text{U}}|\{\mathbf{PUE}\}_k) =& p(\mathbf{Z}_k^{\text{IMU}}|\mathbf{U}_k, \mathbf{P}_k, \mathbf{E}_k^{\text{IMU}})\cdot \\ & p(\mathbf{Z}_k^{\text{odo}}|\mathbf{U}_k, \mathbf{E}_k^{\text{odo}}). \end{aligned} \tag{12}$$

Dependent sensor errors $\mathbf{E}^{\text{IMU}}$ are e.g. biases due to calibration errors or drift and $\mathbf{E}^{\text{odo}}$ are calibration errors or wheel slip. $\mathbf{Z}_k^{\text{P}}$ is dependent on the train state and the sensor errors and typical sensors are position sensors such as GNSS ($\mathbf{Z}^{\text{GNSS}}$) and the dependent errors ($\mathbf{E}^{\text{GNSS}}$) are multipath errors for instance:

$$p(\mathbf{Z}_k^{\text{P}}|\{\mathbf{PE}\}_k) = p(\mathbf{Z}_k^{\text{GNSS}}|\mathbf{P}_k^{\text{geo}}, \mathbf{E}_k^{\text{GNSS}}). \tag{13}$$

$\mathbf{Z}_k^{\text{P}}$ and $\mathbf{Z}_k^{\text{U}}$ are independent of $\mathbf{M}$ and measures only $\mathbf{P}_k$ directly. With the fact that the train is dependent on the railway environment or tracks respectively, the railway environment will have some influence on these measurements through the train states. $\mathbf{Z}_k^{\mathrm{M}}$ is dependent of the railway environment $\mathbf{M}$ represented for example by feature sensors ($\mathbf{Z}^{\text{feature}}$) such as camera vision or eddy current sensors. The dependent sensor errors ($\mathbf{E}^{\text{feature}}$) are calibration errors for instance:

$$p(\mathbf{Z}_k^{\mathrm{M}}|\{\mathbf{PE}\}_k, \mathbf{M}) = p(\mathbf{Z}_k^{\text{feature}}|\mathbf{P}_k^{\text{topo}}, \mathbf{E}_k^{\text{feature}}, \mathbf{M}). \tag{14}$$

These sensors measure features directly in the railway environment with different position than the train. The feature measurements are as well dependent to the train position in order to be near enough to the feature i.e. within the sensing distance.

C. Localization Posterior

The probabilistic localization posterior is the estimation of all train states over time $\mathbf{P}_{0:k}$, all train motions $\mathbf{U}_{0:k}$ and all measurement errors $\mathbf{E}_{0:k}$ given all measurements $\mathbf{Z}_{1:k}$ and the track network $\mathbf{M}$:

$$p(\mathbf{P}_{0:k}, \mathbf{U}_{0:k}, \mathbf{E}_{0:k}|\mathbf{Z}_{1:k}, \mathbf{M}). \tag{15}$$

This localization estimation problem is highly nonlinear because of the discrete distribution of the tracks R_k. In order to solve the localization with a Bayesian filter, this posterior is factorized in the next section.

V. Derivation of the Bayesian Filter

The Bayesian filter is derived from the dynamic Bayesian network (Fig. 1) and the localization posterior from Eq. (15). The purpose of the derivation is a factorized solution in a recursive form for the Bayesian filter implementation. The train states $\mathbf{P}_{0:k}$, motion states $\mathbf{U}_{0:k}$ and the sensor errors $\mathbf{E}_{0:k}$ are estimated, given all the measurements $\mathbf{Z}_{1:k}$ and the railway environment $\mathbf{M}$. We denote $\mathbf{P}_{0:k}, \mathbf{U}_{0:k}, \mathbf{E}_{0:k}$ as $\{\mathbf{PUE}\}_{0:k}$ to save space and rewrite the posterior:

$$p(\mathbf{P}_{0:k}, \mathbf{U}_{0:k}, \mathbf{E}_{0:k}|\mathbf{Z}_{1:k}, \mathbf{M}) = p(\{\mathbf{PUE}\}_{0:k}, |\mathbf{Z}_{1:k}, \mathbf{M}). \tag{16}$$

First, we write the posterior in a Bayesian formulation:

$$p(\{\mathbf{PUE}\}_{0:k}|\mathbf{Z}_{1:k}, \mathbf{M}) = \frac{p(\mathbf{Z}_k|\{\mathbf{PUE}\}_{0:k}, \mathbf{Z}_{1:k-1}, \mathbf{M}) \cdot p(\{\mathbf{PUE}\}_{0:k}|\mathbf{Z}_{1:k-1}, \mathbf{M})}{p(\mathbf{Z}_k|\mathbf{Z}_{1:k-1}, \mathbf{M})}. \tag{17}$$

The denominator is only a normalizing constant and can be neglected for a Bayesian filter. The posterior factorization is not exact anymore and the posterior is only proportional to the nominator. We denote the proportional formulation by $\propto$ and write for the posterior:

$$p(\{\mathbf{PUE}\}_{0:k}|\mathbf{Z}_{1:k}, \mathbf{M}) \propto p(\mathbf{Z}_k|\{\mathbf{PUE}\}_{0:k}, \mathbf{Z}_{1:k-1}, \mathbf{M}) \cdot p(\{\mathbf{PUE}\}_{0:k}|\mathbf{Z}_{1:k-1}, \mathbf{M}). \tag{18}$$

From the first factor of Eq. (18), unnecessary conditions such as previous measurements $\mathbf{Z}_{1:k-1}$, previous train states $\mathbf{P}_{0:k-1}$ and previous errors $\mathbf{E}_{0:k-1}$ are removed because of conditional independences as expressed in the DBN (Fig. 1):

$$\begin{aligned} p(\mathbf{Z}_k|\{\mathbf{PUE}\}_{0:k}, \mathbf{Z}_{1:k-1}, \mathbf{M}) = p(\mathbf{Z}_k|\{\mathbf{PUE}\}_k, \mathbf{M}) = \\ p(\mathbf{Z}_k^{\mathrm{M}}|\{\mathbf{PE}\}_k, \mathbf{M}) \cdot p(\mathbf{Z}_k^{\mathrm{P}}|\{\mathbf{PE}\}_k) \cdot p(\mathbf{Z}_k^{U}|\{\mathbf{PUE}\}_k). \end{aligned} \tag{19}$$

The second factor of Eq. (18) is factored by product rule to achieve a recursive formulation. Finally, unnecessary conditions on states are removed by making use of the Markov condition and conditional independences:

$$\begin{aligned} &p(\{\mathbf{PUE}\}_{0:k}|\mathbf{Z}_{1:k-1}, \mathbf{M}) = \\ &p(\{\mathbf{PUE}\}_k|\{\mathbf{PUE}\}_{k-1}, \mathbf{M}) \cdot p(\{\mathbf{PUE}\}_{0:k-1}|\mathbf{Z}_{1:k-1}, \mathbf{M}). \end{aligned} \tag{20}$$

The second factor of Eq. (20) is the posterior of the last time step and an implementation algorithm operates recursively by using the previous output as input of the new time step computation. The first factor of Eq. (20) is factorized again and independent conditions are removed:

$$\begin{aligned} &p(\{\mathbf{PUE}\}_k|\{\mathbf{PUE}\}_{k-1}, \mathbf{M}) = \\ &p(\mathbf{U}_k|\mathbf{U}_{k-1}) \cdot p(\mathbf{P}_k|\mathbf{P}_{k-1}, \mathbf{U}_k, \mathbf{M}) \cdot p(\mathbf{E}_k|\mathbf{E}_{k-1}). \end{aligned} \tag{21}$$

A. General factorized localization posterior

The factorized general posterior can now be completed by inserting Eq. (21) in Eq. (20) with Eq. (19) into Eq. (18):

$$\begin{aligned} &p(\{\mathbf{PUE}\}_{0:k}|\mathbf{Z}_{1:k}, \mathbf{M}) \propto \\ &\underbrace{p(\mathbf{Z}_k^{\mathrm{M}}|\{\mathbf{PE}\}_k, \mathbf{M}) \cdot p(\mathbf{Z}_k^{\mathrm{P}}|\{\mathbf{PE}\}_k) \cdot p(\mathbf{Z}_k^{\mathrm{U}}|\{\mathbf{PUE}\}_k)}_{\text{sensor likelihoods}} \cdot \\ &\underbrace{p(\mathbf{U}_k|\mathbf{U}_{k-1})}_{\text{1D motion model}} \cdot \underbrace{p(\mathbf{P}_k|\mathbf{P}_{k-1}, \mathbf{U}_k, \mathbf{M})}_{\text{train state transition}} \cdot \\ &\underbrace{p(\mathbf{E}_k|\mathbf{E}_{k-1})}_{\text{sensor errors transition}} \cdot \underbrace{p(\{\mathbf{PE}\}_{0:k-1}|\mathbf{Z}_{1:k-1}, \mathbf{M})}_{\text{recursion}}. \end{aligned} \tag{22}$$

The posterior is now in a general factorized form.

VI. Generic Particle Filter Implementation

For the Bayesian filter implementation we chose a particle filter for the estimation of the posterior Eq. (22), because of the discrete distribution of the tracks. A particle filter represents probability density functions by appropriate particle distributions with appropriate weights of N_p particles [13]. The posterior is represented by

$$p(\{\mathbf{PUE}\}_{0:k}|\mathbf{Z}_{1:k}, \mathbf{M}) \approx \{x_{0:k}^i, w_{0:k}^i\}_{i=1}^{N_p}, \tag{23}$$

where $x_{0:k}^i$ is the i-th particle with its weight w^i of N_p particles and $x_{0:k}^i$ is one sample of the posterior of all time steps until k. As described in [13], particles are generated from a function which is easy to calculate, called the proposal function:

$$x_{0:k}^i \sim q(\{\mathbf{PUE}\}_{0:k}|\mathbf{Z}_{1:k}, \mathbf{M}). \tag{24}$$

Afterwards these particles are weighted [13]. The weights are proportional to the fraction posterior over proposal function:

$$w_k^i \propto \frac{p(\{\mathbf{PUE}\}_{0:k}|\mathbf{Z}_{1:k}, \mathbf{M})}{q(\{\mathbf{PUE}\}_{0:k}|\mathbf{Z}_{1:k}, \mathbf{M})}. \tag{25}$$

A. Recursive Proposal Density

The proposal density is practically generated and sampled to obtain the particle distribution. Further, a proposal density in recursive factorization is desired:

$$q(\{\mathbf{PUE}\}_{0:k}|\mathbf{Z}_{1:k},\mathbf{M}) = q(\{\mathbf{PUE}\}_k|\{\mathbf{PUE}\}_{0:k-1},\mathbf{Z}_{1:k},\mathbf{M}) \cdot \underbrace{q(\{\mathbf{PUE}\}_{0:k-1}|\mathbf{Z}_{1:k-1},\mathbf{M})}_{\text{recursion}} \tag{26}$$

Now, we design the proposal function. We are free to choose any method of spreading particles in our domain. If we choose badly, there would be too less or even no particles near the actual truth. We define ($\hat{=}$) our proposal function:

$$q(\{\mathbf{PUE}\}_k|\{\mathbf{PUE}\}_{0:k-1},\mathbf{Z}_{1:k},\mathbf{M}) \hat{=} p(\mathbf{E}_k|\mathbf{E}_{k-1}) \cdot p(\mathbf{U}_k|\mathbf{U}_{k-1}) \cdot p(\mathbf{P}_k|\mathbf{P}_{k-1},\mathbf{U}_k,\mathbf{M}). \tag{27}$$

$p(\mathbf{E}_k|\mathbf{E}_{k-1})$ is the sensor error process model and calculates the transition of an error of one time step. The transition $p(\mathbf{P}_k|\mathbf{P}_{k-1},\mathbf{U}_k,\mathbf{M})$ is calculated from the map, by the old topological position, velocity and a distance traveled between the time steps. The distance and velocity are computed with Eq. (4) by the train motion transition $p(\mathbf{U}_k|\mathbf{U}_{k-1})$ and a sampled process noise.

B. Particle Weights Update

The particle weights are derived from Eq. (25). The posterior Eq. (22) is divided by the proposal Eq. (27) in Eq. (26):

$$\begin{aligned} w_k^i \propto & \frac{p(\mathbf{Z}_k^{\mathrm{M}}|\{\mathbf{PE}\}_k,\mathbf{M}) \cdot p(\mathbf{Z}_k^{\mathrm{P}}|\{\mathbf{PE}\}_k) \cdot p(\mathbf{Z}_k^{\mathrm{U}}|\{\mathbf{PUE}\}_k)}{1} \\ & \cdot \frac{p(\mathbf{U}_k|\mathbf{U}_{k-1}) \cdot p(\mathbf{E}_k|\mathbf{E}_{k-1}) \cdot p(\mathbf{P}_k|\mathbf{P}_{k-1},\mathbf{U}_k,\mathbf{M})}{p(\mathbf{U}_k|\mathbf{U}_{k-1}) \cdot p(\mathbf{E}_k|\mathbf{E}_{k-1}) \cdot p(\mathbf{P}_k|\mathbf{P}_{k-1},\mathbf{U}_k,\mathbf{M})} \\ & \cdot \underbrace{\frac{p(\{\mathbf{PUE}\}_{0:k-1}|\mathbf{Z}_{1:k-1},\mathbf{M})}{q(\{\mathbf{PUE}\}_{0:k-1}|\mathbf{Z}_{1:k-1},\mathbf{M})}}_{\text{recursion } = w_{k-1}^i} \\ \propto & \; p(\mathbf{Z}_k^{\mathrm{M}}|\{\mathbf{PE}\}_k,\mathbf{M}) \cdot p(\mathbf{Z}_k^{\mathrm{P}}|\{\mathbf{PE}\}_k) \cdot p(\mathbf{Z}_k^{\mathrm{U}}|\{\mathbf{PUE}\}_k) \\ & \cdot w_{k-1}^i. \end{aligned} \tag{28}$$

Finally the weight is computed recursively of the old weight and the sensor likelihoods Eq. (12)-Eq. (14):

$$\begin{aligned} w_k^i \propto & \; w_{k-1}^i \cdot p(\mathbf{Z}_k^{\mathrm{IMU}}|\mathbf{U}_k,\mathbf{P}_k,\mathbf{E}_k^{\mathrm{IMU}}) \cdot p(\mathbf{Z}_k^{\mathrm{odo}}|\mathbf{U}_k,\mathbf{E}_k^{\mathrm{odo}}) \\ & \cdot p(\mathbf{Z}_k^{\mathrm{GNSS}}|\mathbf{P}_k^{\mathrm{geo}},\mathbf{E}_k^{\mathrm{GNSS}}) \cdot p(\mathbf{Z}_k^{\mathrm{feature}}|\mathbf{P}_k^{\mathrm{topo}},\mathbf{E}_k^{\mathrm{feature}},\mathbf{M}). \end{aligned} \tag{29}$$

VII. Summary of the Generic Algorithm

1) Initialize all N_p particles to $U_0 = (0,0,0)$, $\mathbf{E}_0$ and $\mathbf{P}_0^{i,\mathrm{topo}}$ to suitable initial distribution and get the remaining pose from the map $\mathrm{P}_k^i = f_{map}(\mathrm{P}_0^{i,\mathrm{topo}}, 0)$.
2) For every time step k:
   a) Draw samples from the train acceleration, assign it to N_p particles and compute train motion U_k^i by Eq. (4).
   b) Compute train state $\mathrm{P}_k^i = f_{\mathrm{map}}(\mathrm{P}_{k-1}^{i,\mathrm{topo}}, \mathrm{U}_k^i)$
   c) Compute sensor error transitions $p(\mathrm{E}_k^i|\mathrm{E}_{k-1}^i)$.
   d) Weight all N_p particles according to eq. (29).
   e) Perform resampling if necessary

A suitable initialization distribution of 1) is either a uniform distribution of topological positions over the hole track network or the particles are initialized on tracks in a certain vicinity of the first GNSS position measurement.

VIII. Proof Of Concept

A. Particular Implementation

As a proof of concept of the proposed generic algorithm, we implement a simplified 2D train model because trains experience only small bank and slope angles. The IMU is ideally aligned with the train axes and Z^{a_x} measures the longitudinal train acceleration:

$$\mathrm{Z}_k^{a_x} = h^{a_x}(\mathrm{U}_k) + \nu_{a_x} = \ddot{s} + \nu_{a_x}. \tag{30}$$

The measurement function h^{a_x} links the train state with the measurements. In our planar world, the measured train body turn rate is approximately the yaw rate: $\omega_k^z \approx \dot{\psi}_k$. A moving train turns in curves and is affected by a turn rate and centrifugal acceleration in lateral direction while moving with a velocity $\dot{s}$:

$$\mathrm{Z}_k^{a_y} = h^{a_y}(\mathrm{P}_k,\mathrm{U}_k) + \nu_k^{a_y} = \frac{d\psi}{ds} \cdot \dot{s}_k^2 + \nu_k^{a_y}, \tag{31}$$

$$\mathrm{Z}_k^{\omega_z} = h^{\omega_z}(\mathrm{P}_k,\mathrm{U}_k) + \nu_k^{\omega_z} = \frac{d\psi}{ds} \cdot \dot{s}_k + \nu_k^{\omega_z}. \tag{32}$$

The curvature $\frac{d\psi}{ds}$ is stored in the digital map and dependent on the actual topological position: $f_{map}(\mathrm{P}_k^{\mathrm{topo}})$. For the particle filter implementation we use Gaussian models $\mathcal{N}(x|m,\Sigma)$ for the measurement models, where x is the argument, m the mean and Σ the covariance. Instead of the process noise, we sample directly from the train acceleration measurement model Eq. (30). The computation of the proposal function of step 2a) (Sec. VII) samples accelerations from a Gaussian measurement model:

$$\ddot{s}_k^i \sim \mathcal{N}(h^{a_x}(\mathrm{U}_k^i)|\mathrm{Z}_k^{a_x}, \sigma_{a_x}^2). \tag{33}$$

The train motion U_k^i is calculated by Eq. (4), followed by step 2b). For the measurement model we implement a likelihood function by a Gaussian distribution. The sensor likelihood function calculates how likely a measurement fits to the estimation. We define for the GNSS measurement:

$$p(\mathbf{Z}_k^{\mathrm{GNSS}}|\{\mathbf{PE}\}_k) \hat{=} \mathcal{N}(\mathrm{P}_k^{i,\mathrm{geo}}|\mathrm{Z}_k^{\mathrm{GNSS}}, \Sigma_{\mathrm{GNSS}}). \tag{34}$$

For the IMU measurements a_y and ω_z we define:

$$p(\mathbf{Z}_k^{a_y}|\{\mathbf{PUE}\}_k) \hat{=} \mathcal{N}(h^{a_y}(\mathrm{P}_k^i,\mathrm{U}_k^i)|\mathrm{Z}_k^{a_y}, \sigma_{a_y}^2), \tag{35}$$

$$p(\mathbf{Z}_k^{\omega_z}|\{\mathbf{PUE}\}_k) \hat{=} \mathcal{N}(h^{\omega_z}(\mathrm{P}_k^i,\mathrm{U}_k^i)|\mathrm{Z}_k^{\omega_z}, \sigma_{\omega_z}^2). \tag{36}$$

B. Simulation Results

In the first example we demonstrate how the particles converge to the true position. Therefore a train is moving on a single track in Fig. 2. The train motion and the sensors (GNSS and IMU) are simulated. The localization algorithm gets noisy measurements of position with 1Hz and accelerations and turn rates with 10Hz. The particle

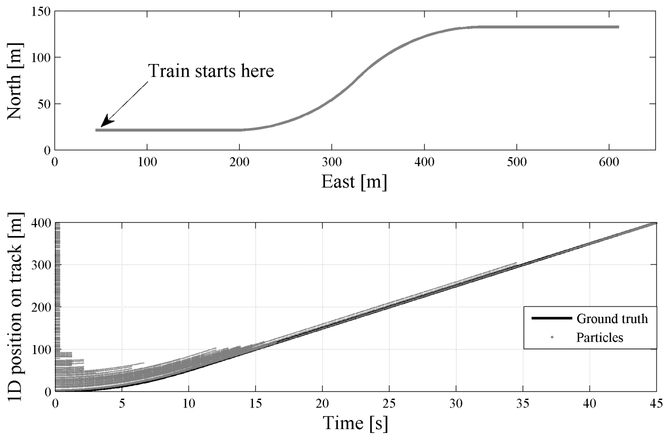

Fig. 2. [Top] Geometric track constructed from straights and curves. [Bottom] Visualized particle distribution of track position estimate over time.

filter does not have a knowledge of the start position and initializes the first particle distribution with 400 particles by an uniform distribution over the hole track. As seen in Fig. 2, the particles condense over time to the correct topological pose. At the beginning, the particles are spread over the hole length of 400 meters. The GNSS is weighting at every second and the particles are resampled. Over time, the particles converge to the truth. Two converging steps are at 16 and at 34 seconds, when the train passes changes of the track curvature and the inertial sensor likelihoods weight according to the track geometry. In order to show these effects, the GNSS deviation is set to 20 meter in this example. The particles are able to convert much quicker with lower GNSS deviations as in Fig. 2. The second example shows a track network with four tracks in Fig. 3. The train starts at track

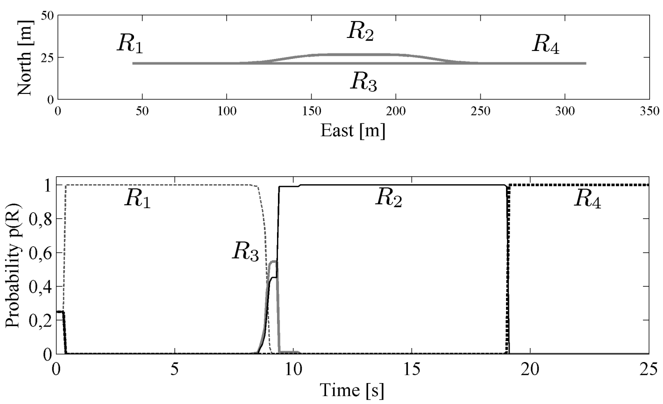

Fig. 3. [Top] Track network with four different tracks. [Bottom] Track probabilities of the train position.

R_1, moves over R_2 and finally to R_4. The probabilities of the estimated train positions on one track are shown in Fig. 3. At the start, particles are uniformly distributed over the hole track network and every track shows the probability of $\frac{1}{4}$th. The particles converge quickly to R_1 and the train moves towards the left switch. The probabilities of R_2 and R_3 rise up when the first particles are split at the switch. At the switch, $p(R_3)$ is slightly higher because some particles are ahead the train and better weighted when the train is still on the straight track R_1. As soon as the train turns on R_2, lateral acceleration and turn rate measurements weight $p(R_3)$ low and $p(R_2)$ high.

IX. Conclusions

We have proposed a top-down train localization approach from theory and provided a causal analysis of effects for the train localization and sensor measurements. We have presented a derivation the Bayesian recursive filter as well as a generic particle filter implementation. In order to proof our concept, a simplified particular implementation is given. First simulations shows a convergence of the particle estimates as well as a track selective train localization in scenarios where switches and parallel tracks occur.

Advantageous extensions are an implementation of 3 dimensional measurements model of an inertial measurement unit (IMU) and dependent sensor error models as well as sensor likelihood models for feature sensors. The digital map is here ideal, i.e. the data is complete and exact. Future work can focus on localization with map uncertainties.

References

[1] T. Strang, M. Meyer zu Hoerste, and X. Gu, "A Railway Collision Avoidance System Exploiting Ad-hoc Inter-vehicle Communications and Galileo," in *13th World Congress on Intelligent Transportation Systems, London, UK*, 2006.

[2] O. Plan, "GIS-gestützte Verfolgung von Lokomotiven im Werkbahnverkehr," Ph.D. dissertation, University of German Federal Armed Forces Munich, Germany, 2003.

[3] F. Böhringer, "Gleisselektive Ortung von Schienenfahrzeugen mit bordautonomer Sensorik," Ph.D. dissertation, Universität Karlsruhe (TH), Germany, 2008.

[4] S. Hensel, C. Hasberg, and C. Stiller, "Probabilistic Rail Vehicle Localization With Eddy Current Sensors in Topological Maps," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, no. 4, pp. 1525 –1536, dec. 2011.

[5] International Union of Railways, "RMTS - European Rail Management Traffic System," http://www.uic.org/spip.php?rubrique847.

[6] J. Wohlfeil, "Vision based rail track and switch recognition for self-localization of trains in a rail network," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, june 2011, pp. 1025 –1030.

[7] A. Geistler, "Bordautonome Ortung von Schienenfahrzeugen mit Wirbelstrom-Sensoren," Ph.D. dissertation, Universität Karlsruhe (TH), Germany, 2007.

[8] D. Titterton and J. Weston, *Strapdown Inertial Navigation Technology, 2nd Edition*. The Institution of Electrical Engineers (IEE), 2004.

[9] O. Heirich, A. Lehner, P. Robertson, and T. Strang, "Measurement and analysis of train motion and railway track characteristics with inertial sensors," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, oct. 2011, pp. 1995 –2000.

[10] M. Malvezzi, B. Allotta, and M. Rinchi, "Odometric estimation for automatic train protection and control systems," *Vehicle System Dynamics*, vol. 49, no. 5, pp. 723–739, 2011.

[11] C. Hasberg and S. Hensel, "Bayesian mapping with probabilistic cubic splines," in *Machine Learning for Signal Processing (MLSP), 2010 IEEE International Workshop on*, 29 2010-sept. 1 2010, pp. 367 – 372.

[12] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1995, iSBN: 978-0131038059.

[13] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *Signal Processing, IEEE Transactions on*, vol. 50, no. 2, pp. 174 –188, feb 2002.