Robust Vehicle Localization based on Multi-Level Sensor Fusion and Online Parameter Estimation

Maik Bevermeier, Sven Peschke, and Reinhold Haeb-Umbach
Department of Communications Engineering
University of Paderborn, Germany
Email: {bevermeier, peschke, haeb}@nt.uni-paderborn.de

***Abstract*—In this paper we present a novel vehicle tracking algorithm, which is based on multi-level sensor fusion of GPS (Global Positioning System) with Inertial Measurement Unit sensor data. It is shown that the robustness of the system to temporary dropouts of the GPS signal, which may occur due to limited visibility of satellites in narrow street canyons or tunnels, is greatly improved by sensor fusion. We further demonstrate how the observation and state noise covariances of the employed Kalman filters can be estimated alongside the filtering by an application of the Expectation-Maximization algorithm. The proposed time-variant multi-level Kalman filter is shown to outperform an Interacting Multiple Model approach while at the same time being computationally less demanding.**

I. Introduction

Current vehicle positioning approaches are often based on a combination of complex sensor units and external data sources like GPS. These sensor data are fused by employing one or more models of vehicle movements to obtain an improved location estimate [1]. Localization accuracy highly depends on the quality of the GPS position estimates. In areas, where shadowing effects caused by high buildings or mountains limit the visibility of satellites, even complete dropouts of the GPS signal can occur. However, accurate localization at all times is important for many applications, notably safety-related applications, such as those being discussed in the framework of Car-2-Car Communications.

In this paper we consider a method for sensor fusion, where sensor data delivered by an accelerometer and gyroscope, as part of an Inertial Measurement Unit (IMU), and by a GPS device are combined [2]. The proposed algorithm is based on a multi-level Kalman filtering (ML-KF), which is able to track the vehicle position, even in cases of a GPS signal dropout. We assume that these dropouts occur randomly and that the dropout interval is also random. These channel conditions are simulated by a Gilbert-Elliott model.

The ML-KF employs the Expectation-Maximization (EM-) algorithm to iteratively update system and measurement noise covariance matrices and, by doing so, adapts to changing driving conditions. An alternative would be to employ a set of predefined dynamical models, like 'constant velocity', 'constant acceleration' or a 'coordinated turn' and using an inference algorithm, which is able to select the appropriate state model at any given time, such as the IMM (Interacting Multiple Model) algorithm. In this paper we will compare our joint estimation and filtering approach with the multiple model approach.

The EM-algorithm as it is used in our setup consists of an iteration between two steps: The Expectation (E-)step involves the multi-level Kalman filtering of internal IMU signals and the external GPS signal based on a single kinematic model of vehicle movements and the current estimates of system and measurement covariances. Among other things, angle estimates of two sources are here optimally fused by taking into account their respective error covariances. The Maximization (M-)step incorporates the reestimation of the state and measurement covariance matrices of the corresponding models. We also show that the blockwise EM-algorithm is able to track the parameters and the position online by simple modifications in the procedure. The experimental results show that the multi-level Kalman filtering achieves higher localization performance and lower computational complexity compared to the multiple model approach.

The paper is organized as follows. In the next section, we give an overview of the equations for generating artificial GPS and IMU measurements based on different kinematic models. Then we describe the Gilbert-Elliott (GE) channel for modeling the GPS signal dropouts. We present the EM-algorithm with the multi-level Kalman filtering and a procedure for the online parameter estimation in section IV. In section V we give some performance results of our time-variant (TV) approach and compare it to an Interacting Multiple Model (IMM) approach [3]. The paper finishes with conclusions drawn in section VI.

II. Generation of Sensor Data

As a sufficient amount of field data was not available we artificially generated sensor data of GPS and IMU devices. We assumed the IMU consists of a gyroscope and an accelerometer, because these are the common components found in almost every car.

A realistic vehicle trajectory can only be described very coarsely with a single linear dynamical model. We therefore used a more sophisticated model here. The vehicle trajectory is generated employing a 3-state Markov chain, see fig.1. Each model state corresponds to a different linear dynamical system of vehicle movements: 'constant velocity' (CV), 'constant acceleration' (CA), and 'coordinated turn' (CT) [4]. The transition probabilities among these systems are given in tab. I;

they have been obtained by analysing field data of real test drives. At each time step k a 3-dimensional position vector

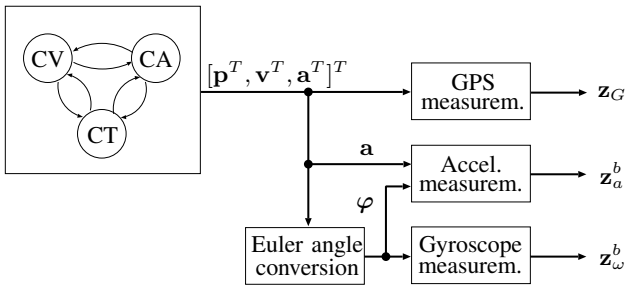

Fig. 1. Data generation chart.

$\mathbf{p}$, velocity vector $\mathbf{v}$ and acceleration vector $\mathbf{a}$, all in the navigation (NED)-frame, are drawn according to the chosen model, and a transition to the next model state is conducted with the transition probabilities of tab. I. This process can be described by the following state equation:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, r_k) + \mathbf{v}_k(r_k); \quad \mathbf{v}_k(r_k) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k(r_k)) \quad (1)$$

with $\mathbf{x} = [\mathbf{p}^T, \mathbf{v}^T, \mathbf{a}^T]^T$. r_k is a regime variable which indicates the currently chosen model, i.e. $r_k \in \{CV, CA, CT\}$. The sample spacing was set to $\Delta T_I = 1/100\ s$. $\mathbf{Q}_k(r_k)$ is the time-variant and model dependent covariance matrix of white Gaussian system noise $\mathbf{v}_k(r_k)$. A typical trajectory in the navigation frame is illustrated in fig. 2. These data form the ground truth from which artificial sensor measurements are generated as described next.

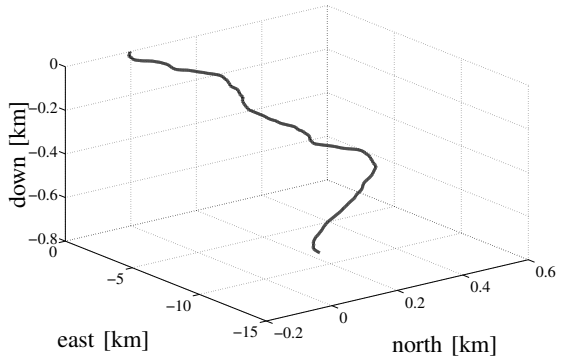

Fig. 2. Typical trajectory created by three models (CV, CA and CT).

We assume different sampling times for the GPS system and the Inertial Measurement Unit (IMU), where the GPS sampling time is assumed to be $\Delta T_G = 1\ s$. The GPS position data is obtained from the following linear measurement equation in the NED-frame

$$\mathbf{z}_{G,l} = \mathbf{H} \cdot \mathbf{x}_{G,l} + \mathbf{w}_{G,l}; \quad \mathbf{w}_{G,l} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{G,l}), \quad (2)$$

where index l counts the GPS sampling time and $\mathbf{w}_{G,l}$ is white Gaussian measurement noise with mean zero and the measurement covariance matrix $\mathbf{R}_{G,l}$, which accounts for the finite GPS positioning precision.

Model	CV	CA	CT
CV	0.6123	0.231	0.1567
CA	0.22	0.54	0.24
CT	0.154	0.166	0.68

TABLE I
TRANSITION PROBABILITIES ASSUMED FOR THE MARKOV MODEL TO GENERATE POSITION, VELOCITY AND ACCELERATION DATA.

For the generation of the IMU sensor data the position and velocity vector are transformed to Euler angles $\boldsymbol{\varphi} = [\alpha, \beta, \gamma]^T = [\text{roll}, \text{pitch}, \text{yaw}]^T$. From these the accelerometer measurements in vehicle body-frame $\mathbf{z}_a^b$ can be generated. Here, we make the assumption that the measurements are independent of corriolis terms or other earth-bounded influences like gravity. The simplified transformation equation is given by

$$\mathbf{a} = \boldsymbol{\Omega}_n^b(\boldsymbol{\varphi}) \cdot \mathbf{a}^b, \quad (3)$$

where $\boldsymbol{\Omega}_n^b$ is the matrix, transforming accelerations $\mathbf{a}^b$ from the body frame to the navigation frame [5]. The accelerometer measurements in body frame can now be written by a non-linear function $h(\boldsymbol{\Omega}_n^b(\boldsymbol{\varphi}_k), \mathbf{a}_k)$, which denotes the coupling between Euler angles and the accelerations:

$$\mathbf{z}_{a,k}^b = h(\boldsymbol{\Omega}_n^b(\boldsymbol{\varphi}_k), \mathbf{a}_k) + \boldsymbol{\delta e}_{a,k} + \mathbf{w}_{a,k}; \quad \mathbf{w}_{a,k} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{a,k}). \quad (4)$$

Again, $\mathbf{w}_{a,k}$ is assumed to be white Gaussian noise with zero mean and covariance matrix $\mathbf{R}_{a,k}$ which takes care of the limited accelerometer accuracy. The vector $\boldsymbol{\delta e}_{a,k}$ denotes an error term described later.

The gyroscope measurements $\mathbf{z}_{\omega,k}^b$ are the angular velocities in body-frame. They depend on the Euler angles and the Euler angle rates $\dot{\boldsymbol{\varphi}}$ with

$$\mathbf{z}_{\omega,k}^b = h(\boldsymbol{\varphi}_k, \dot{\boldsymbol{\varphi}}_k) + \boldsymbol{\delta e}_{\omega,k} + \mathbf{w}_{\omega,k}; \quad \mathbf{w}_{\omega,k} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{\omega,k}), \quad (5)$$

where $\dot{\boldsymbol{\varphi}} = \boldsymbol{\Xi}(\boldsymbol{\varphi}) \cdot \boldsymbol{\omega}^b$ with the well-known transformation matrix $\boldsymbol{\Xi}$ [3] and the true angular velocity vector $\boldsymbol{\omega}^b$. The additive vector $\mathbf{w}_{\omega,k}$ is white Gaussian noise with zero mean and covariance matrix $\mathbf{R}_{\omega,k}$, to model the finite accuracy of the gyroscope measurements.

The non-linear relation between the true values and the measurements of the gyroscope is further caused by scaling and quantization errors, where the scaling error is set to $2\ \%$ and the quantization error to $0.1\ deg./s$. (c.f. gyroscope measurement block in fig. 1). Further the variables $\boldsymbol{\delta e}_a$ in eq. (4) and $\boldsymbol{\delta e}_\omega$ in eq. (5) denote drift and bias errors of the accelerometer and the gyroscope sensor. These can be modeled by deterministic processes, where each of them can be described by [6]:

$$\delta e(t) = C_{\delta e,2} + C_{\delta e,1}(1 - e^{\frac{-t}{T_{\delta e}}}). \quad (6)$$

$C_{\delta e}$ and $T_{\delta e}$ are assumed to be time-variant parameters which are known.

III. Gilbert-Elliott Channel Model

It is well known that GPS availability may be impaired occasionally, e.g. when driving through narrow street canyons or tunnels. If GPS-based position information of neighboring cars is obtained via a Car-2-Car Communication (C2CC) link, it may occur that packets on the C2CC-link are lost, e.g. due to limited transmission bandwidth and heavy traffic.

Temporal unavailability of GPS is modeled by a so-called Gilbert-Elliot model [7], which is able to account for the bursty nature of signal dropouts and which found widespread use in communications research. The Gilbert-Elliot model is a 2-state Markov chain with a "good" (G) and "bad" (B) channel state, see fig. 3, indicating presence and loss of the GPS signal, respectively.

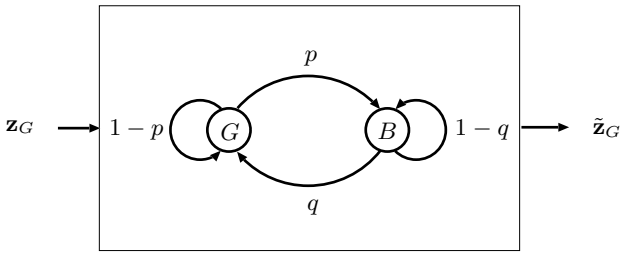

Fig. 3. 2-state Gilbert-Elliott model.

A transition from state G to state B occurs with the probability p and from B to G with probability q. This results in a mean dropout probability of $mdp = p/(p+q)$ and a conditional dropout probability of $cdp = 1-q$, where the latter indicates the probability that a GPS measurement is lost, given the previous one was lost. We defined five channel conditions, see tab. II, where channel "C0" denotes a lossless channel (GPS available all the time), while "C4" is a channel with the highest loss or dropout rate. The GPS receiver is assumed

Condition	C0	C1	C2	C3	C4
cdp	0	0.147	0.33	0.5	0.6
mdp	0	0.006	0.09	0.286	0.385

TABLE II
Channel parameters.

to be able to recognize a signal loss in our experiments, i.e. the value of the state variable $\tilde{r}_k \in \{G, B\}$ is assumed to be known at all times k. The lossy GPS-signal may thus be described by

$$\tilde{\mathbf{z}}_{G,k} = \begin{cases} \mathbf{z}_{G,k}, & \tilde{r}_k = G; \\ 0, & \tilde{r}_k = B. \end{cases} \tag{7}$$

IV. Sensor Fusion with Online Parameter Estimation

Although the ground truth vehicle trajectory has been generated with three interacting linear dynamical models we propose to track the vehicle from measurement data by employing a single dynamical model. This choice has been made on the grounds of saving computational effort compared to a multiple model tracking approach, see further below.

Our goal is to account for the changing vehicle dynamics by employing time-variant system and measurement covariance matrices $\mathbf{Q}$ and $\mathbf{R}$, which adapt to the current driving condition. These covariance matrices have thus to be estimated alongside the tracking task.

While tracking requires knowledge of the covariance matrices, estimation of them requires knowledge of the vehicle's track. The EM-algorithm is an effective way to obtain Maximum likelihood parameter estimates in the presence of incomplete data [8]. It iterates between Expectation (in our case tracking of the vehicle trajectory) and Maximization (estimation of the aforementioned covariance matrices).

The EM-algorithm is typically applied in batch mode as it requires availability of a complete block of data on which it iterates. It therefore does not seem to be applicable for online filtering and parameter estimation as is necessary in the application under investigation here. Nevertheless, we will show later on how this issue can be resolved.

In the following, $\mathbf{x}_k$ denotes the unknown state vector at time index $k = wN + n$, where N is the number of observations per block, w counts the data blocks (windows) and n the vectors within a window (see fig. 5).

Then, $\mathbf{y}^{(w)} = (\mathbf{x}^{(w)}, \mathbf{z}^{(w)})$ contains the unknown state vector sequence $\mathbf{x}^{(w)} = \mathbf{x}_{wN:wN+N-1}$ of window w and the corresponding observation vector sequence $\mathbf{z}^{(w)}$. They are considered as 'complete data' for the EM-algorithm [8]. Now, the EM-algorithm iteratively optimizes the expectation of the log-likelihood function $\tilde{p}(\mathbf{y}^{(w)}; \boldsymbol{\theta}^{(w)})|\hat{\boldsymbol{\theta}}^{(w),i}$ with respect to the unknown parameters. Here, $\boldsymbol{\theta}^{(w)} = (\mathbf{Q}^{(w)}, \mathbf{R}^{(w)})$ denote the parameters to be estimated. Let $\hat{\boldsymbol{\theta}}^{(w),i}$ denote the estimates from data block w at iteration step i. The objective function to be iteratively optimized is thus

$$J(\mathbf{y}^{(w)}; \boldsymbol{\theta}^{(w)}|\hat{\boldsymbol{\theta}}^{(w),i}) = E\left[\tilde{p}(\mathbf{y}^{(w)}; \boldsymbol{\theta}^{(w)})|\hat{\boldsymbol{\theta}}^{(w),i}\right]. \tag{8}$$

An iteration consists of the computation of the objective function (E-step) and its maximization w.r.t. the unknown parameters (M-step). In our case the E-step is carried out by multi-level Kalman filtering and the parameter estimation in the M-step is a maximization operation. Alltogether this results in a time-variant multi-level Kalman filtering approach to vehicle tracking (TV-ML-KF), see fig. 4 for an overview of the overall system. In the following we first consider the multi-level Kalman filtering (E-step) which delivers an estimate of the state vector to be used in the M-step, and subsequently a refinement of the state vector estimate by means of smoothing backward in time and finally the parameter estimation (M-step).

A. Multi-Level Kalman Filtering

As mentioned before the E-step is carried out by multi-level Kalman filtering (ML-KF) as shown in fig. 4.

The GPS position measurements $\tilde{\mathbf{z}}_G$ are available every $\Delta T_G = 1\ s$. Assuming absence of signal dropouts for the time being, the measurement equation is given by eq. (2).

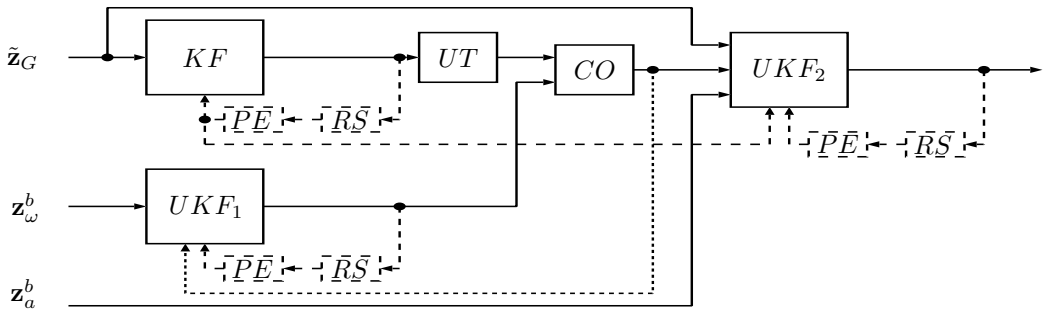

Fig. 4. ML-KF with sensor fusion (dotted line: possibility of feedback; dashed lines: M-step of EM-algorithm).

The state vector $\mathbf{x}_{G,l}$ of the Kalman Filter (KF) contains position, velocity, and acceleration in the navigation frame with $\mathbf{x}_G = [\mathbf{p}^T, \mathbf{v}^T, \mathbf{a}^T]^T$, where the state equation is approximated by

$$\mathbf{x}_{G,l+1} = \mathbf{F}_{CA}(\Delta T_G) \cdot \mathbf{x}_{G,l} + \mathbf{v}_{G,l}; \quad \mathbf{v}_{G,l} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{G,l}). \tag{9}$$

Note, that for $\mathbf{Q}_{G,l}$ and $\mathbf{R}_{G,l}$ the estimates of the system and measurement noise covariance matrices obtained from the last iteration of the EM-algorithm are used. The matrix $\mathbf{F}_{CA}(\Delta T_G)$ is the typical state transition matrix for a constant acceleration move which only depends on the GPS sampling time ΔT_G.

For later sensor fusion with the gyroscope data, the GPS estimates are transformed from NED coordinates to spherical coordinates by an Unscented Transform (UT). Due to the relation between spherical coordinates and the Euler angles, this is done for pitch β and yaw γ only ($\mathbf{x}_G \rightarrow \tilde{\boldsymbol{\varphi}}_G = [\beta, \gamma]^T_G$ and $\mathbf{P}_G \rightarrow \tilde{\mathbf{P}}_G$).

Parallel to this GPS Kalman Filter (denoted by KF in fig. 4) we use an Unscented Kalman Filter (UKF_1) for the gyroscope measurements $\mathbf{z}^b_\omega$, whose equations are given in the box. The state vector is $\mathbf{x}_{GY} = [\boldsymbol{\varphi}^T, \dot{\boldsymbol{\varphi}}^T, (\boldsymbol{\delta e}_\omega)^T, (\boldsymbol{\delta \dot{e}}_\omega)^T]^T$. Due to the linear state equation the prediction step of UKF_1 is the same as in a linear Kalman Filter, which saves computational complexity. Note, that only the covariance matrix $\mathbf{Q}_{GY,k}$ of the system noise is reestimated, while $\mathbf{R}_{GY,k}$ is assumed be known. Despite of the non-linear transformation (Euler angle conversion) we assume that the system noise is white and Gaussian, certainly a simplifying assumption.

The pitch and yaw angles estimated by UKF_1 and the output of UT, i.e. the filtered and transformed GPS estimates $\tilde{\boldsymbol{\varphi}}_G$ can now be combined (CO) in an optimal manner to $\tilde{\boldsymbol{\varphi}}_{CO} = [\beta, \gamma]^T_{CO}$ via

$$(\tilde{\mathbf{P}}_{CO,k})^{-1}\tilde{\boldsymbol{\varphi}}_{CO,k} = (\tilde{\mathbf{P}}_{GY,k})^{-1}\tilde{\boldsymbol{\varphi}}_{GY,k} + (\tilde{\mathbf{P}}_{G,k})^{-1}\tilde{\boldsymbol{\varphi}}_{G,k}, \tag{19}$$

$$(\tilde{\mathbf{P}}_{CO,k})^{-1} = (\tilde{\mathbf{P}}_{GY,k})^{-1} + (\tilde{\mathbf{P}}_{G,k})^{-1}. \tag{20}$$

As we can see, the error covariances of UKF_1 and UT are used as weights [9], [10].

A way to increase the robustness of the filtering towards sensor errors is to feed back the combiner (CO) output to the prediction step of the Unscented Kalman Filter (dotted line in fig. 4), where the relevant values in the state vector and the error covariance matrix are replaced by the more reliable combiner output.

UKF_1 equations:

$$\mathbf{x}_{GY,k|k-1} = \mathbf{F}_{GY}\mathbf{x}_{GY,k-1|k-1} \tag{10}$$

$$\mathbf{P}_{GY,k|k-1} = \mathbf{F}_{GY}\mathbf{P}_{GY,k-1|k-1}\mathbf{F}^T_{GY} + \mathbf{Q}_{GY,k}. \tag{11}$$

$$\mathbf{x}^\nu_{GY,k|k-1} = \begin{cases} \mathbf{x}_{GY,k|k-1}, & \nu = 0 \\ \mathbf{x}_{GY,k|k-1} + \sqrt{n+\lambda}(\mathbf{P}_{GY,k|k-1})^{1/2}_\nu, & \nu = 1,...,n \\ \mathbf{x}_{GY,k|k-1} - \sqrt{n+\lambda}(\mathbf{P}_{GY,k|k-1})^{1/2}_{\nu-n}, & \nu = n+1,...,2n \end{cases} \tag{12}$$

$$\mathbf{z}^b_{\omega,k|k-1} = \sum_{\nu=0}^{2n} w^\nu_m \mathbf{h}(\mathbf{x}^\nu_{GY,k|k-1}). \tag{13}$$

$$\mathbf{C_{xz}} = \sum_{\nu=0}^{2n} w^\nu_c (\mathbf{x}^\nu_{GY,k|k-1} - \mathbf{x}_{GY,k|k-1})(\mathbf{h}(\mathbf{x}^\nu_{GY,k|k-1}) - \mathbf{z}^b_{\omega,k|k-1})^T \tag{14}$$

$$\mathbf{S}_k = \sum_{\nu=0}^{2n} w^\nu_c (\mathbf{h}(\mathbf{x}^\nu_{GY,k|k-1}) - \mathbf{z}^b_{\omega,k|k-1})(\mathbf{h}(\mathbf{x}^\nu_{GY,k|k-1}) - \mathbf{z}^b_{\omega,k|k-1})^T \tag{15}$$

$$+ \mathbf{R}_{GY,k} \tag{16}$$

$$\mathbf{x}_{GY,k|k} = \mathbf{x}_{GY,k|k-1} + \mathbf{K}_k(\mathbf{z}^b_{\omega,k} - \mathbf{z}^b_{\omega,k|k-1}) \tag{17}$$

$$\mathbf{P}_{GY,k|k} = \mathbf{P}_{GY,k|k-1} - \mathbf{K}_k\mathbf{S}_k\mathbf{K}^T_k, \tag{18}$$

with the Kalman gain $\mathbf{K}_k = \mathbf{C_{xz}}\mathbf{S}^{-1}_k$.
The weights are $w^\nu_m = \lambda/(n+\lambda), \nu = 0;\ w^\nu_m = 1/2(n+\lambda), \nu = 1,...,2n$ and $w^\nu_c = (\lambda/n+\lambda) + (1-\alpha^2+\beta), \nu = 0;\ w^\nu_c = 1/2(n+\lambda), \nu = 1,...,2n$ and α, β, λ are further scaling factors.

The last filtering step consists of a second Unscented Kalman Filter (UKF_2). Here, the accelerometer measurements are combined with the GPS measurements. The Euler angles, which are output by the combiner, serve as a control input $\mathbf{u}_k = [\alpha_{GY,k}, \tilde{\boldsymbol{\varphi}}^T_{CO,k}]^T$, where α_{GY} is the roll angle estimated by UKF_1 which is not available at combiner (CO) output. The state vector of UKF_2 is $\mathbf{x}_{AC} = [\mathbf{p}^T, \mathbf{v}^T, \mathbf{a}^T, (\boldsymbol{\delta e}_a)^T, (\boldsymbol{\delta \dot{e}}_a)^T]^T$. The state equation is comparable to eq. (9), where the CA state transition matrix depends here on ΔT_I. The corresponding system noise covariance matrix is $\mathbf{Q}_{AC,k}$. The measurement vector used for updating is

$$\mathbf{z}_{AC,k} = \begin{cases} [\tilde{\mathbf{z}}^T_{G,l}, (\mathbf{z}^b_{a,k})^T]^T, & l\Delta T_G = k\Delta T_I; \\ \mathbf{z}^b_{a,k}, & else, \end{cases} \tag{21}$$

and the corresponding non-linear measurement equation is

$$\mathbf{z}_{AC,k} = h(\mathbf{\Omega}_n^b(\mathbf{u}_k), \mathbf{x}_{AC,k}) + \mathbf{w}_{AC,k}; \\ \mathbf{w}_{AC,k} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{AC,k}), \quad (22)$$

where the estimates of noise covariances given by the last iteration of the EM-algorithm are used for $\mathbf{Q}_{AC,k}$ and $\mathbf{R}_{AC,k}$. Note, that

$$\mathbf{R}_{AC,k} = \begin{cases} blkdiag[\mathbf{R}_{G,l}, \mathbf{R}_{a,k}^b], & l\Delta T_G = k\Delta T_I; \\ \mathbf{R}_{a,k}^b, & else, \end{cases} \quad (23)$$

where we assume that $\mathbf{R}_{a,k}^b$ is the known covariance matrix describing the accuracy of the accelerometer measurements in the body-frame, which needs not be reestimated as it does not change over time.

B. Smoothing

In addition to the filtering step via ML-KF as described before, we recommend to further optimize the objective function in eq. (8) by conducting a smoothing step (reverse time filtering) on the previous data block, see fig. 5. To this end,

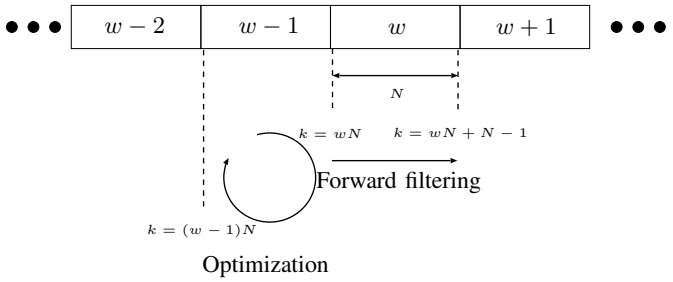

Fig. 5. Illustration of online filtering.

the estimated state vector trajectory of length N is further improved by backward smoothing within the data window $w-1$. The recursive smoother (RS) equations for each filter at times k with $k = (w-1)N, \ldots, (w-1)N + N - 1$ are [11]:

$$\mathbf{x}_{k|N} = \mathbf{x}_{k|k} + \mathbf{\Lambda}_k(\mathbf{x}_{k+1|N} - \mathbf{x}_{k+1|k}), \quad (24)$$

$$\mathbf{P}_{k|N} = \mathbf{P}_{k|k} + \mathbf{\Lambda}_k(\mathbf{P}_{k+1|N} - \mathbf{P}_{k+1|k})(\mathbf{\Lambda}_k)^T, \quad (25)$$

where $\mathbf{\Lambda}_k = \mathbf{P}_{k|k}(\mathbf{F}_k)^T(\mathbf{P}_{k+1|k})^{-1}$ is the smoother gain. $\mathbf{P}_{k|N}$ is the estimation error covariance matrix of the smoothed estimation vector $\mathbf{x}_{k|N}$. With this improved estimate of the unobservable state vector sequence, the Maximization step (M-step) of the EM-algorithm can be carried out.

C. Parameter Estimation

Estimates of the unknown system and measurement covariance matrices can now be obtained by maximizing the objective function (8).

The objective function can be decomposed in two components by noting that

$$\tilde{p}(\mathbf{x}, \mathbf{z}; \boldsymbol{\theta}) = \tilde{p}(\mathbf{x}; \boldsymbol{\theta}) + \tilde{p}(\mathbf{z}|\mathbf{x}; \boldsymbol{\theta}). \quad (26)$$

Of these two components the first is independent of the measurement noise while the second is independent of the system noise covariance matrix. We thus obtain

$$\hat{\mathbf{Q}}^{i+1} = \underset{\mathbf{Q}}{\operatorname{argmax}}\, E[\tilde{p}(\mathbf{x}^{(w-1)}; \boldsymbol{\theta}^{(w-1)})|\hat{\boldsymbol{\theta}}^{(w-1),i}], \quad (27)$$

$$\hat{\mathbf{R}}^{i+1} = \underset{\mathbf{R}}{\operatorname{argmax}}\, E[\tilde{p}(\mathbf{z}^{(w-1)}|\mathbf{x}^{(w-1)}; \boldsymbol{\theta}^{(w-1)})|\hat{\boldsymbol{\theta}}^{(w-1),i}]. \quad (28)$$

This is the operation to be carried out in the blocks denoted PE (Parameter estimation) in fig. 4.

D. Iterative Refinement

The computations on block $w-1$ can usually be carried out at a much higher speed than the forward filtering on block w described in section IV-A, because the forward filter has to wait for the next measurement to become available, while at the time that the samples of data block w arrive, the samples in window $w-1$ are already cached and available for processing. This allows for carrying out several iterations of the EM-algorithm on block $w-1$ without a compromise on the processing latency, as long as the last sample of block w has not yet arrived.

The whole algorithm is summarized as follows:

1) Forward filtering of data in window w employing the currently available parameter estimates $\hat{\boldsymbol{\theta}}^{(w-2)}$ to compute $J(\mathbf{y}^{(w)}; \boldsymbol{\theta}^{(w)}|\hat{\boldsymbol{\theta}}^{(w-2)})$.
2) Concurrently: Smoothing of the filter outputs of block $w-1$ (eqs. (24), (25)).
3) Parameter estimation using the smoothed data of block $w-1$: $\operatorname{argmax}_{\mathbf{Q},\mathbf{R}} J(\mathbf{y}^{(w-1)}; \boldsymbol{\theta}^{(w-1)}|\hat{\boldsymbol{\theta}}^{(w-1)})$ (eqs. (27), (28)).
4) If one of the following conditions is met:
   - Forward filtering of block w is finished.
   - Maximum no. of iterations i is reached.

   then abort parameter estimation and goto 6. Else goto the next step.
5) Forward filtering of the signals in block $w-1$: $J(\mathbf{x}^{(w-1)}; \boldsymbol{\theta}^{(w-1)}|\hat{\boldsymbol{\theta}}^{(w-1),i})$. Goto step 2.
6) Advance block index ($w := w+1$) and goto step 1, using the parameter estimates of block $w-1$.

We propose to estimate $\mathbf{R}_{G,l}$ over the last 40-60 sec. and the other time-variant covariance matrices every 2-8 sec.

V. Experimental Results

We compared the performance of the proposed system with the multi-model filtering approach, both with respect to computationaly complexity and achievable tracking accuracy. Inference for a switching dynamical model system can be accomplished with the Interacting Multiple Model (IMM) algorithm. Due to the similarity of the CA and the CT model, we decided to use the IMM approach only with two dynamical models, CV and the CA, which reduced its complexity somewhat.

As a further comparison we implemented a time-variant GPS-based tracking approach, where, similar to our proposed

time-variant multi-level Kalman filtering approach, the system and measurement noise covariance matrices are estimated iteratively with the EM algorithm. Between two successive GPS measurements the positions were calculated by dead reckoning using the accelerometer and gyroscope data [12]. This approach is denoted by TV-GPS-DR in the following. The dead reckoning approach using only IMU measurements is denoted TV-DR.

The artificially generated data set was of length of $800\ s.$, where the time-variant standard deviation of the GPS measurements was set to $20\ m.$ at the beginning, increases to $1600\ m^2$ at $t = 200\ s.$ and decreases to $100\ m^2$ at time $t = 600\ s.$ The subsequent values of the system noise covariances were drawn from a normal distribution whose standard deviation was 0.45 times of the mean of $0.2\ m/s^2$ for CV ($0.8\ m/s^3$ for CA and CT), however limited to positive values.

A. Complexity

The Multi-Level Kalman Filter uses three state estimators for filtering the INS and GPS measurements. Although this seems to be a substantial computational complexity, tab. III shows that the computational effort is significantly lower compared to the IMM approach. The table shows the mean elapsed times of the different processing stages of the three approaches. Processing was carried out on a 1.6 GHz Dual Core Processor with 1 GB RAM. The length of the data block we considered here is $8\ s.$ Note, that the IMM state vector is

Block-length: $8\ s.$	TV-GPS-DR	TV-IMM	TV-ML-KF
Forward filtering	$2.441\ s.$	$9.9504\ s.$	$2.798\ s.$
Backward filtering/ Smooth.	$0.2996\ s.$	$15.957\ s.$	$0.3448\ s.$
Parameter estimation	$0.0926\ s.$	$0.0694\ s.$	$0.0926\ s.$

TABLE III
PROCESSING DURATIONS OF FILTER STEPS.

of size (27×1), while the TV-ML-KF consists of three filters with state vector dimensions 9×1 (KF), 12×1 (UKF_1) and 15×1 (UKF_2). The dimension of the filters in TV-GPS-DR are the same as those of TV-ML-KF.

The higher elapsed time for the IMM is mostly due to the higher dimension of the state vector. Compared to the other filter operations the calculations of the matrix inversions inside the Kalman filters are the computationally most expensive steps, as the complexity of an inversion is of order $\mathcal{O}(D^3)$, where $(D \times D)$ is the matrix dimension. Due to the smaller state dimensions of the multi-level method, these calculations can be done much faster than one inversion of a matrix of higher dimension (IMM).

Note, that the time-variant IMM approach needs block-lengths of $30 - 40\ s.$, a latency which is not acceptable in a real driving scenario. The reason is that in each block, the algorithm has to calculate the covariances for all models, which depend on the estimated model probabilities. The model probabilities have to be quite accurate to be able to obtain reliable covariance estimates.

B. Positioning Performance

Fig. 6 displays typical trajectories for the different tracking approaches. The estimated trajectories of TV-ML-KF and the IMM approach are close to the true trajectory. The trajectory obtained from GPS-only positioning also follows the true one, however its larger estimation error can be clearly seen. On the other hand the pure dead reckoning approach TV-DR (without GPS measurements as reference points) exhibits strong drifts as expected due to the accumulation of the sensor errors.

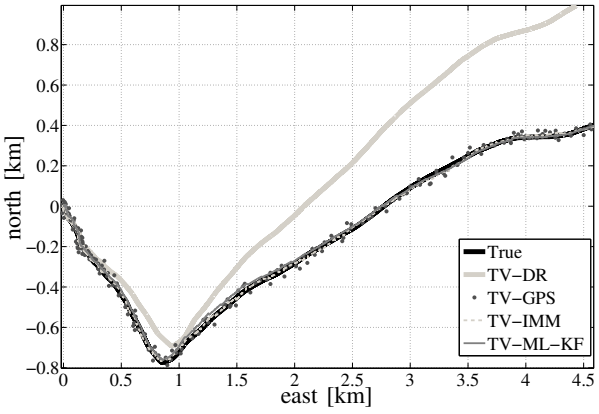

Fig. 6. Estimated trajectory for time-variant covariances.

Fig. 7 shows for a part of the simulated trajectory the distance $\Delta\mathbf{p} = \sqrt{|\mathbf{p}_{\text{true}} - \hat{\mathbf{p}}|^2}$ between the true and the estimated vehicle position over time. As can be seen, the IMM method and the multi-level Kalman Filter are comparable in performance, while a Kalman filtering of GPS measurements alone performs poorly.

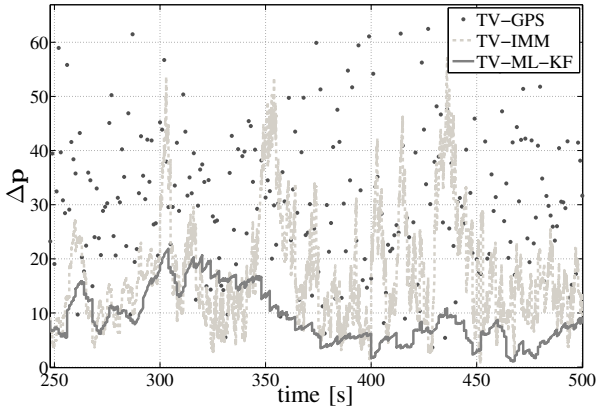

Fig. 7. Positioning error over time.

Fig. 8 displays the cumulative density function (CDF) of the positioning error for the different approaches. It can be seen, that the time-variant GPS and DR approaches show an inferior performance compared to TV-IMM and TV-ML-KF. The reason for the slightly worse positioning accuracy

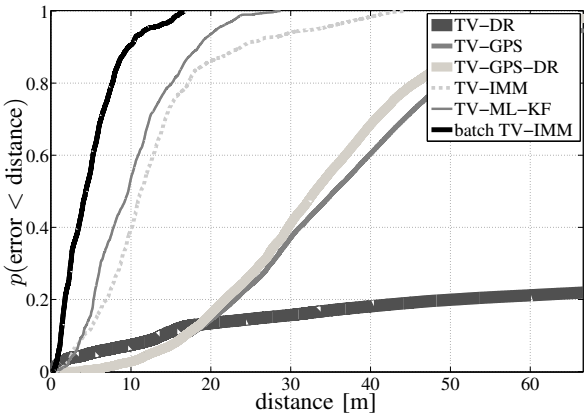

Fig. 8. Cumulative density functions for different approaches.

of TV-IMM compared to TV-ML-KF is probably the higher dimension of the state vector, which results in less reliable parameter reestimation. As a kind of performance upper bound we included the performance of a batch TV-IMM filter, which incorporates a forward and backward filtering on the whole data set of $800\ s.$, a setup certainly beyond any practical use in a real driving scenario.

Sofar we have assumed a GPS signal without dropouts. Next we are going to consider the impact of a partial loss of GPS on the localization performance. Fig. 9 shows the position accuracy of TV-ML-KF for the channel conditions defined in tab. II. The TV-ML-KF method seems to be very robust against GPS signal errors. When there are only short intervals (channel condition "C1" or "C2") without GPS reference positions, these can be compensated by the multi-level filter structure. Even for increasing dropout intervals like channel "C3" the probability for an error less than $16\ m$ is still 0.5. The figure also contains the results obtained with TV-GPS with no dropouts for comparison purposes.

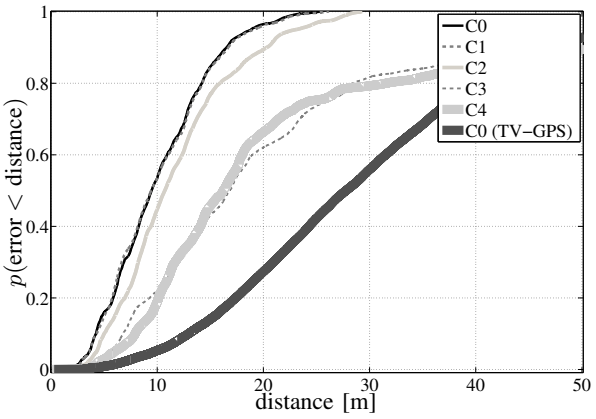

Fig. 9. Cumulative density functions for different channel conditions (TV-ML-KF).

In Fig. 10 we look at the pitch angle estimates of the different filter approaches. The GPS based pitch estimates alone are of worse quality and not presented here. The IMM approach estimates the pitch angle quite well. The best estimate is delivered by the combination of both, Kalman filtered GPS measurements and the gyroscope estimates (CO). The reason for the performance gain is the combination via their error covariance matrices. In this way, the quality of the individual estimators is incorporated, which results in an optimal angle estimation in the Bayesian sense. Further, for comparison

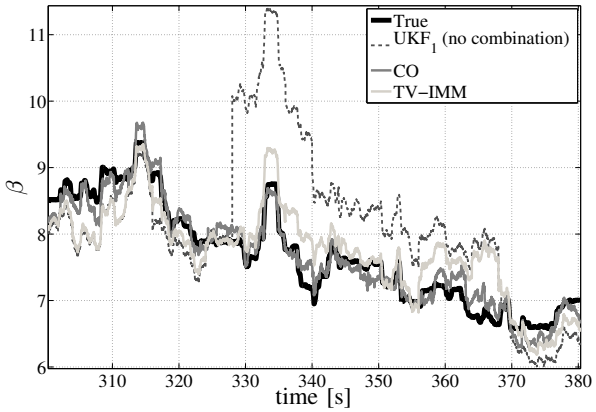

Fig. 10. Pitch angle estimates of different filtering approaches.

the gyroscope measurements of UKF_1 are given, when no feedback and combination is done. Note, that the results of the acceleration estimation depend on the angle estimation (eq. 3). So, a reliable direction estimation is important.

C. Parameter Tracking

Fig. 11 illustrates the true and estimated measurement variances for the GPS positions in north-, east-, and down-direction ($\mathbf{R}_G = blkdiag(\sigma_n^2, \sigma_e^2, \sigma_d^2)$), when there are GPS signal dropouts (channel "C2"). The true variances $\sigma_{n,e,d}^2$ are set to $400\ m^2$ at the beginning of the testset. Then, at $t = 200\ s$. the true variances increase to $1600\ m^2$ and decrease to $100\ m^2$ at time $t = 600\ s$. As we can see, the changes in the measurement variances of the GPS signal are tracked quite well by the reestimation of the covariance matrix with the EM-algorithm. Note, that only the last 60 GPS ($60\ s.$) observations here are used for the tracking, when available. Therefore, the tracking needs some time to reach the desired values. Further, we use false initialization values at the beginning of the data set.

VI. Conclusions

In this paper, a sensor fusion algorithm combined with online parameter estimation is proposed. The fusion algorithm uses a Multi-Level Kalman Filter architecture for position estimation. The time-variant covariance matrices of the corresponding state estimators are reestimated at least every $20\ s$. The proposed scheme is compared to other common filter approaches (GPS alone or dead reckoning) and to an Interacting Multiple Model algorithm. Despite the higher computational

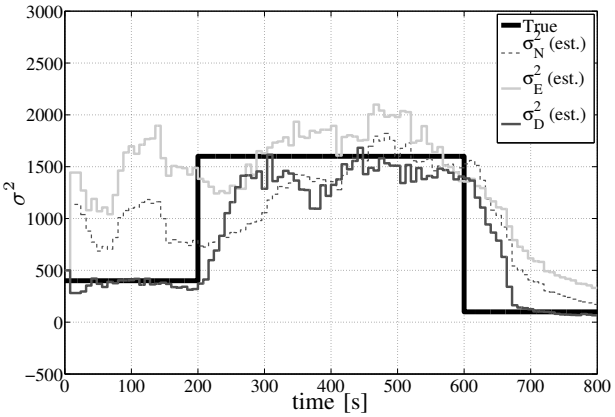

Fig. 11. Tracking of the GPS measurement variances (TV-ML-KF).

complexity of the IMM compared to the time-variant Multi-Level Kalman Filter (TV-ML-KF), the position estimates are less accurate. Further it is shown that TV-ML-KF has only a low latency, which is important when it is to be employed in a real driving scenario.

REFERENCES

[1] J. Farrell and M. Barth, *The Global Positioning System and Inertial Navigation*, McGraw-Hill, 1998.

[2] M. Bevermeier, S. Peschke, and R. Haeb-Umbach, *Joint Parameter Estimation and Tracking in a Multi-Stage Kalman Filter for Vehicle Positioning*, *for publishing in proc. of IEEE Vehicular Technology Conference VTC 2009*, April 2009.

[3] D. Huang and H. Leung, An Expectation-Maximization Based Interacting Multiple Model Approach for Cooperative Driving Systems, *IEEE Trans. on Intelligent Transportation Systems*, June 2005.

[4] X. Rong Li and V. P. Jilkov, A Survey of Maneuvering Target Tracking: Dynamic Models, *Proc. of SPIE Conference on Signal and Data Processing of Small Targets*, Orlando, Florida, USA, April 2000.

[5] C. Jekeli, *Inertial Navigation Systems with Geodetic Applications*, *de Gryter*, 2001.

[6] B. Barshan and H. F. Durrant-Whyte, Inertial Navigation Systems for Mobile Robots, *IEEE Trans. on Rob. and Autom.*, Vol. 11, No. 3, June 1995.

[7] E. N. Gilbert, *Capacity of a burst-noise channel*, *Bell System Technical Journal*, Bd. 39, p. 1253-1265, 1960.

[8] T. K. Moon, *The expectation-maximization algorithm*, *IEEE Signal Processing Magazine*, Vol. 13, No. 6, Nov. 1996.

[9] R. E. Helmick, W. D. Blair, and S. A. Hoffman, Fixed-Interval Smoothing for Markovian Switching Systems, *IEEE Trans. on Information Theory*, Vol. 41, No. 6, November 1995.

[10] T. Kailath, A.H. Sayed, and B. Hassibi, *Linear Estimation*, Prentice Hall, 2001.

[11] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*, John Wiley & Sons, Inc., 2001.

[12] D. Obradovic, H. Lenz, and M. Schupfner, *Fusion of Sensor Data in Siemens Car Navigation System*, *IEEE Trans. on Vehicular Techn.*, Vol. 56, No. 1, Jan. 2007.