Abstract—Ground target surveillance is getting a more and more important task, for civil as for military applications. Ground targets may be bound to infrastructural constraints, like vehicles moving on roads. Therefore, incorporating road map information into the tracking process is a current topic of research. This paper presents a tracking algorithm, which considers road map information. The algorithm supports junctions as well as curves by applying a Gaussian sum algorithm which leads to a variable structure multiple model approach. The update is done using Kalman filtering. Beside the filtering, some aspects of the whole tracking framework will be presented and some simulation results are given.

Keywords: GMTI, ground target, tracking, Kalman filtering, road map, multiple model.

I. INTRODUCTION

Ground surveillance is getting more and more important, since several applications exist, like civil applications [1], airport ground control [2]–[4], and military surveillance applications [5].

In contrast to air target tracking, ground targets move on the earth surface and more clutter exists. In order to reduce ground clutter, MTI sensors like GMTI are widely used. In addition, targets may be hidden by topographic structures, e.g. trees and hills.

Beside this, ground targets are often densely spaced (e.g. convoys) and infrastructural constraints may exist, like roads for example. Most of the vehicles move on roads and therefore it is a good choice to incorporate road maps into the tracking process.

There exist several filtering approaches in ground target tracking with road maps. Particle filters seem to be a good choice for this kind of application, since road map supported tracking leads to non-linear filter models [6]–[8]. In addition, velocity constraints can be applied easily. Disadvantage is the large computation time, which makes them demanding for real time applications.

Alternatively, Kalman filter based approaches are widely discussed in literature [5], [9]–[12]. Most of them deal with long straight roads only and use a Variable Structure Interacting Multiple Model (VS-IMM) approach to support junctions.

The work presented in this paper employs a Kalman filter based approach, which uses a one-dimensional on-road model. Since the update is done in the three-dimensional measurement space, equality constraints are applied to project the updated state back to the one-dimensional on-road state [13]. Crossings are supported by a Variable Structure Multiple Model (VS-MM) approach without interaction, since the interaction of the widely used VS-IMM leads to bad estimations at junctions. Beside this, the presented approach considers narrowly curved roads by further approximating the models with Gaussian sums similar to [12], but extends over junctions.

The paper is organized as follows: In section II, the filtering with road map support is described. This includes the model and propagation, the junction and curved road handling as well as the update of the filter. Section III will shortly describe the used framework including data association. Since only simulated data was used to create the results, section IV illustrates some aspects of the simulation. Several results will be presented in section V. Finally, a summary and conclusion will be given in section VI.

II. FILTERING WITH ROAD MAP SUPPORT

Roads are represented by a start and end position and a width. The start and end position is given in WGS84 coordinates. Roads are connected by their start or end positions and the connection defines a junction. Curves are approximated by small adjacent roads, as shown in figure 1. The roads are internally stored by a simple graph, which consists of nodes and edges. The edges are the roads and the nodes are the junctions of the roads. Edges and nodes can also have additional information, like speed limits, one-way flag, tunnel-flag. Road maps can be obtained from real maps or self created for testing and debugging purposes.
An overview of the filtering with road map information is shown in figure 2.

The process begins with the linear propagation of the on-road models along the road. This is followed by the junction handling, if the on-road position leaves the road segment at the start or end. Since the positional part of the propagated covariances may be much larger than the road, the covariances are subdivided using a Gaussian sum algorithm. The filter update with the measurement is done in the coordinate system of the sensor. Since the state may move out of the road during update, it is projected back to the road. After the projection, some maintenance is done.

In the next sections, the work flow will be described in more detail.

A. Model and propagation

The on-road filtering is done in the coordinate system of the road. For simplification, the targets move only on the center line of the road [7]. Therefore, a one-dimensional model satisfies the requirements and the state vector is given by

\[ \mathbf{x}_r = \begin{pmatrix} x_r \\ v_{x,r} \end{pmatrix}, \]

with \( x_r \) the distance to the start \( s_r \) of the road and \( v_{x,r} \) the velocity with reference to the road start \( s_r \) (see figure 3). The index \( r \) indicates the road, the state vector belongs to.

The model used is a simple linear constant velocity, white noise acceleration model [14], i.e.

\[ \mathbf{x}_r(k + 1) = A \cdot \mathbf{x}_r(k) + \nu(k) \]

with \( \nu(k) \) the process noise, the transition matrix

\[ A = \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix}, \]

and \( T \) the sampling time. The propagation equations for the estimated state \( \hat{\mathbf{x}}_r \) and error covariance \( \hat{\mathbf{P}}_r \) are given by

\[ \mathbf{x}_r(k + 1 | k) = A \cdot \mathbf{x}_r(k | k) \]

\[ \mathbf{P}_r(k + 1 | k) = A \cdot \mathbf{P}_r(k | k) \cdot A^T + \mathbf{Q}, \]

with the covariance matrix \( \mathbf{Q} \), which is given by

\[ \mathbf{Q} = 2\tau \sigma_a^2 \begin{pmatrix} T^3/3 & T^2/2 \\ T^2/2 & T \end{pmatrix}, \]

with \( \tau \) the target maneuver correlation time and \( \sigma_a \) the expected acceleration level. In the following, the time step notation will be omitted.

B. Junction handling

Roads only have a finite length \( l_r \). Therefore, the position \( x_r \) of the state may be outside of the road (only along the center line), i.e. \( x_r < 0 \) or \( x_r > l_r \). In this case, the state must be adapted to the adjacent road, that the position is inside that road as shown in figure 4.

If the position \( x_r \) passes a junction with more than one road connected to it, a new model for each adjacent road \( i \) is created as shown in figure 5 (for better illustration, two-dimensional covariances are drawn in this and the following figures). The original road is ignored. The new model consists of the adjusted state \( \hat{x}_r \) and the estimation error covariance \( \hat{\mathbf{P}}_i \). The covariance needs no modification during the adjustment, i.e. \( \hat{\mathbf{P}}_i = \mathbf{P}_r \).

The created models are held in a variable structure, multiple model manner (VS-MM) [15]. The weights for the new models
be written as
\[ \sum \]

one of the on-road models is larger than the road itself (one sigma value). Figure 6 illustrates this case. This leads to a bad association and bad estimation results.

To overcome this problem, a large covariance of a model is further approximated by a sum of weighted Gaussian distributions [12] (see figure 7).

The derivation of the Gaussian sum is done by representing road segments \( r \) by \( \text{rect} \left( \frac{x - m_i}{l_i} \right) \), with center \( m_i \) and length \( l_i \). All road segments stringed together ignoring junctions can be written as
\[
\sum_i \text{rect} \left( \frac{x - m_i}{l_i} \right) = 1. \tag{7}
\]

Rewriting the propagated distribution \( N(x; x_r, P_r) \) and considering, that roads far away from the estimated position can be ignored, yields to
\[
N(x; x_r, P_r) \approx N(x; x_r, P_r) \sum_i \text{rect} \left( \frac{x - m_i}{l_i} \right) \tag{8}
\]

A road segment can be approximated by
\[
\text{rect} \left( \frac{x - m_i}{l_i} \right) \approx \exp \left\{ \frac{1}{2} \left( \frac{x - m_i}{\sigma_i^2} \right)^2 \right\} = \sqrt{2\pi \sigma_i} N(x; m_i, \sigma_i) \tag{9}
\]

with \( \sigma_i = l_i/2 \) the uncertainty set to half of the road length \( l_i \). This finally leads to
\[
N(x; x_r, P_r) \approx \sum_i N(x; x_r, P_r) \sqrt{2\pi \sigma_i} N(x; m_i, \sigma_i). \tag{10}
\]

Applying the Kalman update equation results in the Gaussian sum approximation
\[
N(x; x_r, P_r) \approx \sum_i \alpha_i N(x; x'_i, P'_i), \tag{11}
\]

with weights \( \alpha_i \). The resulting covariances fit the run of the roads much better, as shown in figure 7.

\[
\begin{align*}
&x'_1, P'_1, \alpha_5 \\
&x'_4, P'_4, \alpha_4 \\
&x'_1, P'_2, \alpha_2 \\
&x'_3, P'_3, \alpha_3
\end{align*}
\]

Figure 7. Gaussian sum approximation of a "long" covariance.

The described approach can be extended to support junctions as well. In this case, for each adjacent road of the junction a new state and subdivided covariance is set up and the weighing factor is divided by the number of adjacent roads of the junction.

Covariances subdivided by this means are also added to the VS-MM, whereas the original model is removed. The subdivision is done, until the weight \( \alpha_i \) for a subdivided model is below a given threshold. The subdivided covariances are not linked any more to the original distribution after added to the VS-MM, i.e. they are handled as independent on-road models.

D. Filter update

The update of the propagated state and covariance is done in the coordinate system of the sensor. For this, the one-dimensional on-road state and covariance must be enlarged to three dimensions and afterwards converted to the coordinate system of the sensor.
The state is enlarged to three dimensions by just adding components with the value zero:

\[ \mathbf{x}_{r,3D} = \begin{pmatrix} x_r & 0 & 0 & v_{x,r} & 0 & 0 \end{pmatrix}^T , \]

where 3D is the local coordinate system of the road in three dimensions. The original covariance

\[ \mathbf{P}_r = \begin{pmatrix} \sigma_x^2 & \sigma_{x,v_y}^2 & \sigma_{x,v_z}^2 \\ \sigma_{x,v_y}^2 & \sigma_{v_y}^2 & 0 \\ \sigma_{x,v_z}^2 & 0 & \sigma_{v_z}^2 \end{pmatrix} \]

must be extended and finally leads to

\[ \mathbf{P}_{r,3D} = \begin{pmatrix} \text{POS} & \text{MIX} & \text{VEL} \end{pmatrix} , \]

with

\[ \text{POS} = \begin{pmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_{v_y}^2 & 0 \\ 0 & 0 & \sigma_{v_z}^2 \end{pmatrix} \]

\[ \text{MIX} = \begin{pmatrix} \sigma_{x,v_y}^2 & 0 & 0 \\ 0 & \sigma_{v_y}^2 & 0 \\ 0 & 0 & \sigma_{v_z}^2 \end{pmatrix} \]

\[ \text{VEL} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \]

with \( \sigma_w = w_r / 2 \), i.e. the road width is assumed as uncertainty. \( \sigma_h \) is the uncertainty in the height, \( \sigma_{v_y} \), the velocity uncertainty across the road and \( \sigma_{v_z} \), the velocity uncertainty in height.

The enlarged three-dimensional state and covariance are then transformed to the coordinate system of the sensor S, i.e. the state and covariance are given by \( \mathbf{x}_s \) and \( \mathbf{P}_s \).

The measurements of the sensor are two-dimensional polar measurements with a range \( R \) and azimuth \( \varphi_{AZ} \) which can be converted to three-dimensional Cartesian measurements knowing the terrain elevation. With these converted measurements \( \mathbf{z}(k) = (x \ y \ z)^T \), the update of the filter is performed.

All models are handled using the Variable Structure Multiple Model (VS-MM) approach without an interaction. The weight \( w_i \) for each model \( i \) is therefore calculated by

\[ w_i(k+1) = \frac{w_i(k) \cdot \Lambda_i(k+1)}{\sum_j w_j(k) \cdot \Lambda_j(k+1)} \]

with innovation likelihood \( \Lambda_i(k) = p_i(\mathbf{z}(k) | \mathbf{Z}(k-1)) \), with \( \mathbf{Z}(k-1) = \{ z(0), \ldots, z(k-1) \} \) all available sensor measurements up to time step \( k-1 \).

E. State projection

The updated state \( \mathbf{x}_s(k+1 | k+1) \) and updated covariance \( \mathbf{P}_s(k+1 | k+1) \) are in the coordinate system of the sensor and must be transformed back to the one-dimensional on-road coordinate system. This comprises the transformation to the 3D road coordinate system and a projection of the state and covariance from three dimensions to one dimension. For this, state equality constraints are used \([13],[16]\). The state equality constraints are defined by

\[ \mathbf{D} \cdot \mathbf{x} = \mathbf{d}, \]

with

\[ \mathbf{D} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \]

and \( \mathbf{d} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \)

and project out the y and z component of the position and velocity of the state vector \( \mathbf{x}_{r,3D} \) and covariance \( \mathbf{P}_{r,3D} \) respectively and leads to the one-dimensional on-road state \( \mathbf{x}_r \) and covariance \( \mathbf{P}_r \). For simplification, the road is assumed to be of infinite length during the projection. Therefore, the projected position may lie outside of the road. Thus, the adjustment of the state is done again, as described in section II-B. This is only one possible solution, since especially at junctions, several possibilities for the projection exist, i.e. the state and covariance could be projected to several roads.

F. Maintenance

At each junction, new on-road models are created. Also, additional models are created during the subdivision. This leads to an increasing number of models. Therefore, the number of models must be limited. There are two approaches used to limit the model number: Merging and pruning.

Two models \( i \) and \( j \) are merged, if the statistical distance

\[ d = \sqrt{\sum (\mathbf{x}_i(k+1) - \mathbf{x}_j(k+1))^2} \]

is below a threshold. In the implementation, all models of one road are clustered using the minimum spanning tree algorithm. Models of different roads are not merged so far.

Models are pruned or removed, if the weight is below a threshold. In the implementation, the threshold \( th \) is chosen according to the largest weight \( w_{\text{max}}(k) \): \( th = \frac{w_{\text{max}}(k)}{\alpha} \), with \( \alpha \) chosen empirically.

III. FRAMEWORK

The filtering is embedded in a large framework, which does the whole track handling. This mainly includes track initiation, track dropping, data association and track output, and of course the filtering.

The next sections will only describe the parts of the framework, which must be adapted to use road map supported filters.

A. Output

The presentation of results of a road map supported filtering is often difficult: The on-road filter may contain several models on different roads. The question is, how to combine all the models for the output. There are two simple approaches:

The first is to combine all states and covariances to one mean and covariance using moment matching \([14]\). The disadvantage of this approach is, that the position of the calculated state \( \mathbf{x}_O \) may lie outside any road (see 8). This appears especially at curves and junctions.

Another possibility is to provide the best model, i.e. the model with the largest weight \( w_i(k+1) \). The disadvantage is, that this doesn’t represent the real estimation, but, on he other hand, the position of the state is always on a road.
B. Data Association

The data association consists of gating and scoring, whereby the gating is used to ignore tracks, which are unlikely associated to the plot. The score value is used to do a more sophisticated decision, which plot belongs to which track. The framework uses a SD-assignment [17] algorithm to solve the gating is used to ignore tracks, which are unlikely associated to the plot. The score value can be calculated in several ways: The simplest solution is to use the combined state and covariance (moment matching), propagate according to a linear free mass model to the plot using the statistical distance 

\[
\sum_{i} w_i(k) 
\cdot N(\mathbf{z}(k); \mathbf{Hx}_{i}(k|k-1), \mathbf{HP}_{i}(k|k-1)\mathbf{H}^T + \mathbf{R}) \ .
\]  

(22)

The advantage of this approach is, that the road structure is considered. Disadvantage is the much more needed computation time.

C. Initiation

A track is only initiated, if there are any roads available. For this, a initial state and covariance is projected to each road. If the projected state is within the road and the statistical distance of the projected state and the original state and covariance is below a threshold, an on-road kinematic is created for this road. If there is at least one kinematic, a new track is created.

IV. SIMULATION

For the creation of the results, simulated data was used. The simulation emulates a GMTI sensor. This sensor is mounted on a flying platform and sweeps over a small area on the ground.

The simulation of this sensor is plot based, i.e. the targets are treated as points. Wave propagation effects are not considered, but some other aspects are regarded:

- GMTI sensors only detect targets, which have some radial velocity to the sensor. The decision, if a plot is created by the simulation depends on the probability of detection \( P_d \). This probability is dependent on the radial velocity \( v_R \) (range rate) of the target. This dependency is shown in figure 9. In the figure, \( v_{\text{min}} \) is the minimal detectable velocity, i.e. the sensor cannot detect any target with a radial velocity \( v_R \) below this value. \( v_{\text{full}} \) describes the velocity, above which a target will be detected for sure. \( v_{\text{blind}} \) is the blind velocity of the blind Doppler. The probability of detection based on the radial velocity is also multiplied by an overall probability to account for other missed detections and closely spaced targets, which may be fused to a single plot by the sensor.
- Ground targets may be hidden by any other topographic structure, like hills. Therefore, elevation data is used, to detect such occlusions. As data source DTED (Digital Terrain Elevation Data) or SRTM (Shuttle Radar Topography Mission) can be used.

The simulation allows the definition of paths for the sensor, the position the sensor is looking to and of the targets. The result are plots with azimuth, elevation, range and range rate and can be directly used by the tracking framework.

V. RESULTS

This section will present some results of the road map assisted filtering. Results are obtained from the GMTI simulation. The platform is flying to the south-east at an altitude of \(10000\) m starting in the north of the observation area at a distance of \(100\) km. The uncertainties of the measurements are assumed to be \( \sigma_R = 20\) m for the range, \( \sigma_{AZ} = 0.005 \) for the azimuth and \( \sigma_{EL} = 0.005 \) for the elevation angle. The target velocity is \(10\) m/s. The acceleration uncertainty is set to \( \sigma_a = 1\) m/s\(^2\) and the correlation time to \( \tau = 10\) s. The minimal detectable velocity of the GMTI simulation is set to \( v_{\text{min}} = 1\) m/s, and the full detectable velocity to \( v_{\text{full}} = 3\) m/s, i.e. especially in the curves, some missed detections occur. The used road map includes some very curvy parts and a simple junction (see figure 10). This ensures that multiple on-road models will exist.

Figure 10 shows the result from a run of the tracker. In figure 11, a closer view of figure 10 is shown.

In figure 11, the on-road models can be seen. Due to the large measurement uncertainty, the plot is quite far from the road. Since several possibilities for an on-road targets exist in the given sampling time, several on-road models exist.

\[
\mathbf{x}_1, \mathbf{P}_1, \alpha_1
\]

\[
\mathbf{x}_2, \mathbf{P}_2, \alpha_2
\]

\[
\mathbf{x}_3, \mathbf{P}_3, \alpha_3
\]

Figure 8. State output: Combination of all on-road models may lead to a state \( \mathbf{x}_O \), which position is outside of any road.

\[
P_d
\]

\[
0 \quad v_{\text{min}} \quad v_{\text{full}} \quad v_{\text{blind}} \quad v_R
\]

Figure 9. Probability of detection \( P_d \) of the simulated GMTI sensor.
Figure 10. Results of a run: The crosses are the measurements, which are connected to the estimated state.

Figure 11. Single track output with on-road models shown as two-dimensional ellipses.

Figure 12. One target in clutter: On-road filter used.

Figure 13. Close up view of one target in clutter

Figure 12 shows a run with clutter. The density of the clutter was set to $2.0 \cdot 10^{-7}$ clutter plots per scan per square meter. Figure 13 shows a closer view of figure 12. As seen in the figures, the target is tracked without a loss and no other tracks are established.

The next results are Monte Carlo runs, which show the estimation accuracy of the filter. In this runs, the filtering quality, i.e. position and velocity errors are of interest only. 10000 Monte-Carlo runs were executed for the scenario shown in figure 10. The results are shown in figure 14. For comparison, an IMM filter with a three-dimensional free mass, white noise acceleration model without road map support is also included in figure 14. The IMM filter contains a low and high noise model with the uncertainties $\sigma_{a,low} = 1 \text{ m/s}^2$ and $\sigma_{a,high} = 10 \text{ m/s}^2$. As seen in figure 14, the road map supported tracker leads to much better estimation results in position and velocity as the free mass IMM filter.

VI. SUMMARY

In this paper, an algorithm for tracking ground targets which move on roads was presented. In difference to many other publications, narrow curves are also supported by applying a Gaussian sum approximation to the on-road models, which expands over junctions as well. All models are kept in a VS-MM approach. The on-road filter is embedded in a tracking framework, which also allows the tracking in scenarios with clutter.

The results of the tracking show that the filter provides good estimations in scenario with and without clutter. The Monte Carlo runs also show that the filter has much better estimation accuracies than a regular IMM filter without road map support.

The next steps are the handling of closely spaced targets. These targets cannot be separated by a sensor all the time, and missed detections occur. There exists some ideas, how to solve this problem [18].
Another important topic are missed detections due to the blind Doppler, small radial velocities or occlusions through the topography. If the framework knows about these effects, they can be used as negative information [19].

FIGURE 14. Simulation results: Position (a) and velocity errors (b), each with and without (WO) road map support.

REFERENCES


