Online-Estimation of Road Map Elements using Spline Curves

Carsten Hasberg
Institut für Mess- und Regelungstechnik
Universität Karlsruhe
Engler Bunte Ring 21
D-76131 Karlsruhe, Germany
Email: hasberg@mrt.uka.de

Stefan Hensel
Institut für Mess- und Regelungstechnik
Universität Karlsruhe
Engler Bunte Ring 21
D-76131 Karlsruhe, Germany
Email: hensel@mrt.uka.de

Abstract—The basis for map assisted moving target tracking is a correct and up-to-date representation of the environment. In this contribution a method is proposed to model curved structures, e.g. roads or tracks, with cubic spline curves. The unknown model parameters are estimated based on corrupted measurements using a probabilistic approach. In particular, the method presented results in a linear formulation of the estimation problem, that offers itself to standard optimal estimation techniques. While the parameters are adapted to the measurements, the geometric progression of the curve is locally adjusted. The resulting representation provides information about the expected geometric run and the uncertainty of the estimated curve.

Keywords: Parameter Estimation, Cubic Spline Curve, Kalman Filter, Road Map, Mapping.

I. INTRODUCTION

Incorporating prior information about dynamic behaviour and measurement model both have a strong impact on the performance of object tracking algorithms [1]. For ground moving objects assumptions on road- or track-constrained motion may augment significant information to tracking processes. To obtain robust object tracking results a correct map of the road network must be given permanently [2][3].

Roads are often composed by a sequence of geometric primitives, e.g. straights, transition curves and circular arcs, to enable a comfortable and safe driving without any abrupt variations in lateral acceleration. Throughout this contribution an automated mapping approach is presented to extract precise geometric information about the road elements from position measurements. While the map update is carried out immediately, the actualised map can be integrated in the object tracking framework and support the object tracking algorithms with actual information.

In the following, a short overview of seminal contributions to the topic of automated mapping for curved structures is given. The novelty of this approach is compared to these.

In [4], the road is defined by a series of nodes that are adapted to a measured track. A map update is computed point-wise by averaging between the position on the smoothed track and the nearest node of the actual map estimation. There is no explicit model for the geometric uncertainty used in the map update process.

In [5] a polygon is used to approximate the road elements within a road network. The geometric model parameters are node positions and tangents at the nodes. Curvature information between two nodes is stored in an additional parameter. Based on velocity information at adjacent nodes, new nodes are added while the mapping error is adapted.

In [6] an approach is presented estimating road segments element-wise using a least squares method incorporating GPS traces. The nodes are interpolated by a spline curve and the parameters of the spline curve are stored in the map database. To obtain a precise geometric approximation of the real world, smooth piecewise defined polynomial curves with continuous curvature are used. Based on cubic spline curves an estimation framework is presented to model road constraints and extract the model parameters systematically.

As presented above, roads or tracks are build of clothoid elements with continuous curvature along the whole piecewise defined curve. While the handling of the resulting parameter space is quite difficult, it is interpolated with a spline curve, yielding to a compact parameter description. Cubic splines are used to archive a minimum upper bound for the resulting deviations along the run of the curve.

Compared to the approaches described above, the approximation error remains smaller especially in the case of strongly curved roads. Knowledge about curve structure, e.g. minimum arc radiuses, can systematically be integrated in the modelling process.

An update of the geometric model is carried out using disturbed position measurements. Each incoming measurement is associated with an initialised curve element and updates the model parameters locally. While the model parameters are estimated the statistical information about the progression of the curve and the uncertainty of the geometric information is adapted and stored in the generated spline map.

While the approaches [4] and [5] provide sharp regional limited map updates, our approach estimates a new geometry accumulating knowledge about the given curve, taking into account the precision of the measurement. In particular, the method proposed in this contribution results in a linear formulation of the estimation task, that offers itself to standard optimal estimation techniques.
While the method presented in [6] extracts a new map based on a global set of position measurements, our approach continuously provides an updated parameter set after each incoming position measurement.

By modelling interpolation and measurement errors separately, the optimal parameter update can be computed while the influence of both error sources is minimized. While measurement errors are caused by incorrect sensor systems, interpolation errors stem from the suboptimal choice of a geometric representation.

The application of our spline model into an object tracking framework can be realized in a straight forward way. In the case of known model parameters, the curve can be evaluated in ground coordinates. Because of the chosen spline dimension the tangents and the curvature can also be calculated continuously along the curve.

For the sake of clarity, the whole concept is described for planar curves. However, it easily generalises to 3d curves.

The paper is structured as follows: Section II describes the basic concept of Kalman filtering and curve interpolation using cubic spline curves. Section III focuses on a linear formulation of the parameter tracking problem for the spline model parameters. In section IV, the data processing is described in detail. Experimental results for simulated data and for real GPS position measurements along a railway track will be presented in section V.

II. THEORETICAL BACKGROUND

A. Optimal State Estimation with the Kalman Filter

The Kalman filter [1] [7] is a tool to estimate a state vector that can be observed through indirect measurements which are subject to noise. The functional dependency between a state vector \( \mathbf{x}_k \) and the measurement vector \( \mathbf{z}_k \) is modelled by the linear matrix equation

\[
\mathbf{z}_k = \mathbf{H} \mathbf{x}_k + \mathbf{v}_k. \tag{1}
\]

The measurement noise \( \mathbf{v}_k \) is assumed white and zero-mean. Its covariance matrix \( \mathbf{Q} \) describes all deviations from the expected behaviour.

The Kalman filter allows incorporating knowledge of system dynamics into the estimation modelled by the linear matrix equation

\[
\mathbf{x}_{k+1} = \mathbf{F} \mathbf{x}_k + \mathbf{w}_k \tag{2}
\]

predicting the state vector from one discrete time instant \( k \) to the next. The system noise \( \mathbf{w}_k \) with the covariance matrix \( \mathbf{R} \), is again assumed to be white, zero-mean.

While both error sources can be assumed to be mutual statistically independent the resulting state estimation minimizes the mean square error. The state vector \( \mathbf{x} \) is estimated bias-free with mean vector \( \hat{\mathbf{x}} \) and with covariance matrix \( \mathbf{P} \), that has minimum trace among all linear estimators.

The Kalman filter state estimation is computed in two main steps [8]. Initially the estimate of the state vector is predicted with

\[
\hat{\mathbf{x}}_k = \mathbf{F} \hat{\mathbf{x}}_{k-1} \tag{3}
\]

\[
\mathbf{P}_k = \mathbf{F} \mathbf{P}_{k-1} \mathbf{F}^T + \mathbf{R}. \tag{4}
\]

The measurement update equations

\[
\mathbf{K}_k = \mathbf{P}_k \mathbf{H}^T (\mathbf{H} \mathbf{P}_k \mathbf{H}^T + \mathbf{R})^{-1} \tag{5}
\]

\[
\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H} \hat{\mathbf{x}}_k) \tag{6}
\]

\[
\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{P}_k \tag{7}
\]

adjust the predicted estimation with new incoming measurement information. The index \((\ldots)^-\) is used for the estimation of the state vector before the measurement update and the index \((\ldots)^+\) for the estimation after the update.

Further on the Kalman filter can easily be adapted to estimate time invariant parameters [9], if the preconditions presented above are fulfilled.

B. Interpolation with Cubic Spline Curves

Polynomial interpolation is the most fundamental of all interpolation concepts [10]. Nowadays, polynomial interpolation is mostly of theoretical value: Faster and more accurate methods have been developed. Those methods are piecewise polynomial. Thus they rely on the polynomial methods.

The curve \( C \) is to be approximated by a piecewise representation, a cubic spline curve. The original curve \( C \) is given by a set of samples by

\[
C = \{ \mathbf{P}_i | \mathbf{P}_i = (x_i, y_i), i = 0, \ldots, n \}. \tag{8}
\]

First step is to determine parameter values for each vertex \( P_i \). Chord length parameterization is chosen to achieve a parameterization proportional to the distances of the data points. It yields a curve, having the smallest curvature values [10]. The parameter values \( u_i \) for \( x_i = x_i(u_i) \) and \( y_i = y_i(u_i) \) at the supporting points \( P_i \) are calculated according to

\[
\tau_i = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \tag{9}
\]

\[
u_{i+1} = \nu_i + \tau_i \tag{10}
\]

as sketched out in Figure 1. The chord length is initialised as \( u_0 = 0 \) and \( i = 0, \ldots, n - 1 \).

While data points \( P_i \) and parameter values \( u_i \) are available, both coordinate functions are interpolated separately with splines. For \( u \in [u_i, u_{i+1}] \) and \( i = 0, \ldots, n - 1 \) the coordinate functions are defined piecewise with

\[
s_x(u) = s_x^{(i)}(u) \tag{11}
\]

\[
s_y(u) = s_y^{(i)}(u). \tag{12}
\]

3rd order polynomials archive continuous curvature of the spline coordinate functions

\[
s_x^{(i)}(u) = a_{x,i} + b_{x,i} \Delta u_i + c_{x,i} \Delta u_i^2 + d_{x,i} \Delta u_i^3 \tag{13}
\]

\[
s_y^{(i)}(u) = a_{y,i} + b_{y,i} \Delta u_i + c_{y,i} \Delta u_i^2 + d_{y,i} \Delta u_i^3 \tag{14}
\]

with \( \Delta u_i = u - u_i \).
and points, are computed [11]. In a first step, the auxiliary variables result in the natural cubic spline. Amongst all twice continuous differentiable functions natural cubic splines yield the least oscillation about the function which is interpolated. To calculate the geometric curve parameters \(a_{x,i}, \ldots, d_{y,i}\), a set of smoothness conditions has to be fulfilled. For the \(x\)-component, these conditions are

\[
\begin{align*}
{s_x(i)(u_i)} & = x_i & \text{for } i = 0, \ldots, n \\
{s_x(i)(u_i)} & = s_x(i+1)(u_i) & \text{for } i = 1, \ldots, n \\
{s_x(i)(u_i)}' & = s_x(i+1)(u_i)' & \text{for } i = 1, \ldots, n \\
{s_x(i)(u_i)}'' & = s_x(i+1)(u_i)'' & \text{for } i = 1, \ldots, n.
\end{align*}
\]

Similar conditions apply for the \(y\)-component. For the \(n\) cubic polynomials it is necessary to determine \(4n\) conditions (since for one polynomial of degree three, there are four conditions on choosing the curve). Since, the interpolating property gives \(4n - 2\) conditions, it is additionally set

\[
\begin{align*}
{s_x(0)(u_0)}'' & = 0 \\
{s_x(n)(u_n)}'' & = 0
\end{align*}
\]

resulting in the natural cubic spline. Amongst all twice continuously differentiable functions natural cubic splines yield the least oscillation about the function which is interpolated.

Based on the expressed conditions (15) and (16), the second derivatives \(s_x(i)(u_i)''\), called the moments \(m_i\) at the supporting points, are computed [11]. In a first step, the auxiliary variables

\[
\begin{align*}
h_i & = u_{i+1} - u_i & \text{for } i = 0, \ldots, n - 1 \\
k_i & = h_{i-1} + h_i & \text{for } i = 1, \ldots, n - 1
\end{align*}
\]

and

\[
p_i = \frac{6}{h_i}(x_{i+1} - x_i) - \frac{6}{h_{i-1}}(x_i - x_{i-1})
\]

for \(i = 1, \ldots, n - 1\) are computed.

For natural boundary conditions are \(s_x(0)(u_0)'' = 0 = m_0\) and \(s_x(n)(u_n)'' = 0 = m_n\). By solving

\[
\begin{pmatrix}
p_1 \\
p_2 \\
\vdots \\
p_{n-1}
\end{pmatrix} =
\begin{pmatrix}
2k_1 & h_1 & 0 & \cdots & 0 \\
h_1 & 2k_2 & h_2 & \cdots \\
\vdots & \vdots & \ddots & \cdots \\
p_{n-1} & & & & h_n
\end{pmatrix}
\begin{pmatrix}
m_1 \\
m_2 \\
\vdots \\
m_{n-1}
\end{pmatrix}
\]

the missing moments \(m_i\) for \(i = 1, \ldots, n - 1\) are calculated. Once the moments \(m_i\) are known for all the supporting points \(P_i\), the parameters of the piecewise polynomials are given by the following equations

\[
\begin{align*}
a_{x,i} & = x_i & \text{(21)} \\
b_{x,i} & = \frac{x_{i+1} - x_i}{h_i} - \frac{2m_i + m_{i+1}}{6} & \text{(22)} \\
c_{x,i} & = \frac{m_i}{2} & \text{(23)} \\
d_{x,i} & = \frac{m_{i+1} - m_i}{6h_i} & \text{(24)}
\end{align*}
\]

for \(i = 0, \ldots, n - 1\).

In an analogous manner to the described procedure for the \(x\)-component the values \(a_{y,i}, b_{y,i}, c_{y,i}\) and \(d_{y,i}\) can be extracted for the \(y\)-component.

### III. PARAMETER TRACKING

Based on the equations presented in Section II-B a linear relation between any point on the curve defined by its parameter value \(u\) and the model parameters is described.

The global curve parameter \(u\) is assumed to be known exactly. For a given \(u \in [u_i, u_{i+1}]\), the masking vector \(k^{(i)}\) is designed to select the corresponding component of the spline curve. For the \(x\)-component follows

\[
s_x(u) = k^{(i)T} \cdot s_x(u) = \begin{pmatrix} 0 & T & s_x^{(0)}(u) \\ \vdots & & \vdots \\ 0 & 1 & s_x^{(i-1)}(u) \\ 0 & 0 & s_x^{(i)}(u) \\ \vdots & \vdots & \vdots \\ 0 & 0 & s_x^{(n-1)}(u) \end{pmatrix}. \tag{25}
\]

To archive a compact matrix formulation, the supporting points of the spline curve are combined to column vectors \(x = (x_0, \ldots, x_n)^T\) and \(y = (y_0, \ldots, y_n)^T\). Next step is the combination of the local spline parameters of the piecewise defined polynomials to column vectors. The separation of the \(x\)- and the \(y\)-components retain unchanged. For the constant component of the polynomial follows, e.g.

\[
a_x = \begin{pmatrix} a_{x,0} \\ \vdots \\ a_{x,n-1} \end{pmatrix} \quad \text{and} \quad a_y = \begin{pmatrix} a_{y,0} \\ \vdots \\ a_{y,n-1} \end{pmatrix}. \tag{26}
\]

According to the calculations presented in Section II-B, the linear matrix equations

\[
\begin{pmatrix}
a_x \\
a_y \\
b_x \\
b_y \\
c_x \\
c_y \\
d_x \\
d_y
\end{pmatrix} =
\begin{pmatrix}
A_x \\
A_y \\
B_x \\
B_y \\
C_x \\
C_y \\
D_x \\
D_y
\end{pmatrix}
\]

are derived to evaluate the coefficients of the piecewise defined (20) polynomials. For a given set of supporting points \(P_i\), the spline
parameter vectors \( a_x, \ldots, d_y \), defining the continuous run of the curve, can be extracted straight forward using Eq. (27).

By using Eq. (25) and (27), the spline function for the \( x \)-component is

\[
s_x(u) = k^{(i)} \cdot s_x(u) = k^{(i)} T \left[ a_x + b_x \Delta u_i + c_x \Delta u_i^2 + d_x \Delta u_i^3 \right]
\]

\[
= k^{(i)} T \left[ A_x x + B_x \Delta u_i + C_x \Delta u_i^2 + D_x \Delta u_i^3 \right]
\]

\[
= \hat{H}(u) x.
\] (28)

Analogously, the calculations for the \( y \)-component result in

\[
s_y(u) = H_y(u) y.
\] (29)

In conclusion, the whole process of spline curve parameter formulation for given supporting points is given by a linear matrix equation

\[
s(u) = \left( \begin{array}{c} s_x(u) \\ s_y(u) \end{array} \right) = \left( \begin{array}{cc} H_x(u) & 0 \\ 0 & H_y(u) \end{array} \right) \left( \begin{array}{c} x \\ y \end{array} \right),
\] (30)

To include the uncertainty of the geometric representation, the parameter vectors \( x \) and \( y \) are assumed to be random variables. Here, \( \hat{x} \) is the mean vector and \( Q_x \) the covariance matrix of the multivariate normal distribution of \( x \). The parameters of the random variable \( y \) are defined analogously. For each value of the curve parameter \( u \in [u_0, u_n] \), the matrices \( H_x(u) \) and \( H_y(u) \) can be computed. Because of the linear transformation in equation (30), the vector \( s(u) \) is also normal distributed [7]. The mean vector is

\[
\hat{s}(u) = \left( \begin{array}{c} \hat{s}_x(u) \\ \hat{s}_y(u) \end{array} \right) = \left( \begin{array}{cc} H_x(u) & 0 \\ 0 & H_y(u) \end{array} \right) \left( \begin{array}{c} \hat{x} \\ \hat{y} \end{array} \right)
\] (31)

and the covariance matrix is

\[
Q_u(u) = \left( \begin{array}{cc} \sigma_{x,x}^2(u) & 0 \\ 0 & \sigma_{y,y}^2(u) \end{array} \right)
\] (32)

with

\[
\sigma_{x,x}^2(u) = H_x(u) Q_x H_x^T(u)
\] (33)

\[
\sigma_{y,y}^2(u) = H_y(u) Q_y H_y^T(u).
\] (34)

Both the mean \( \hat{s}(u) \) and the covariance \( Q_u(u) \) of the derived stochastic geometric model depend on the global curve parameter \( u \). In Figure 2, the random variable is depicted for a given value \( u \) of the global curve parameter. Because both components of the parameter vector \( x \) and \( y \) are uncorrelated the principal axes are parallel to the \( x \)- and \( y \)-axes of the ground coordinate system.

Assuming static parameters, yield to a simple state space representation of the state vector \((x_k, y_k)^T\). The dynamic model is

\[
\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ y_k \end{pmatrix} + \omega_k
\] (35)

\[
\text{Figure 2. Curve model for a given } u = \tilde{u} \text{ including mean and covariance of the random variable}
\]

and the observation equation is given by

\[
\tilde{z}_k = \begin{pmatrix} H_x(u) & 0 \\ 0 & H_y(u) \end{pmatrix} \begin{pmatrix} x_k \\ y_k \end{pmatrix} + \nu_k.
\] (36)

The interpolation errors are modelled by the system noise \( \omega_k \) that is assumed white Gaussian distributed. The measurement noise \( \nu_k \) is caused by the measurement principle and corrupts the position measurement of the sensor. Altogether the resulting state space representation is linear and the model parameter estimation readily offers itself to a classical Kalman filter estimation approach as presented in Section II-A.

IV. PARAMETER ESTIMATION PROCESS

In this section the computation of the model parameter update is described step by step. To actuate the parameter estimation, an initial parameter set must be known to estimate the run of the curve. Here, it is assumed that this parameter set is given, e.g. from a coarse digital map. Based on these parameter values and the covariance matrices, that are initialized according to

\[
Q_x = diag(\sigma_{x,x}^2, \ldots, \sigma_{x,x}^2)
\] (37)

\[
Q_y = diag(\sigma_{y,y}^2, \ldots, \sigma_{y,y}^2),
\] (38)

the run of the curve \( s(u)^- \) is given, like sketched out in Figure 3.

To process the measurement update a data acquisition step between mean run of the curve \( \hat{s}(u)^- \) and measurement \( \tilde{z} = (\tilde{x}, \tilde{y})^T \) must be implemented. Minimization of the Euclidean distance between the curve and the actual measurement varying the global curve parameter \( u \) yields the closest point on the curve determined through \( \tilde{u} \) as visualised in Figure 4. While the data acquisition is performed the sensor platform is localized on the curve approximately. Throughout the concept this approximation is assumed to be correct and the value of the curve parameter is fixed to \( \tilde{u} \).

Knowledge of \( \tilde{u} \) allows to formulate the update step. First, the matrices \( H_x(\tilde{u}) \) and \( H_y(\tilde{u}) \) are computed as described in
Section III. The innovations for both components

\[ \tilde{x} - H_x(\hat{u}) \hat{x} \]  
\[ \tilde{y} - H_y(\hat{u}) \hat{y} \]

are calculated and the parameter vectors are estimated with a Kalman filter as described in Section II-A. In Figure 5, both the run of the mean before and after the measurement update step are shown. The uncertainty of the model parameters is adapted and decreases locally as shown in Figure 6.

While the system noise is added to the parameter estimations in the prediction step the influence of actual measurements can be guaranteed. In our implementation the system noise is fixed to a constant value to remain the dynamic properties of the model. Another possibility is to assume time invariant states and set the system noise to zero.

V. EXPERIMENTAL RESULTS

The parameter tracking framework has been tested with simulated data and real GPS traces of a train localization system.

A. Simulation Results

In this simulation the system noise is assumed to be zero. In Figure 7 the progression of the curve before the update is presented. The resulting geometric run after the computation of a couple of position measurements is shown in the Figures 8 and 9.

The figures illustrate the smooth adaption of the curve to the measurements. If there is no additional sensor information available the geometric properties of the curve remain, while in zones with many new measurement information the curve is adapted.

As described in Section IV the spline parameter \( u \) is chosen while the Euclidean distance between the actual position measurement and the curve is minimized.

B. Estimation of Railway Track Geometry with GPS Measurements

To verify the approach it is tested with GPS position measurements that were taken along a light railway with a
simple GPS receiver mounted on a train. Based on this data the run of a defined section of the rail track of about 1000 m is estimated.

The uncertainty of the GPS position measurements is influenced by ionospheric and atmospheric delays, multipath deviations and satellite shadowing [12]. Additionally the chosen track segment is located within a narrow valley, where the receiver is not able to track the satellites permanently, while the train passes the segment.

Assuming the real position, located exactly on the rail track, to be disturbed with normal distributed white noise, yields to the following observation equation

\[ \tilde{z}_{GPS} = \tilde{s}(u) + v. \] (41)

To obtain a precise geometric estimation of the railroad segment the presented method is used to combine the position measurements with a-priori knowledge about the progression of the curve. In simulations it was found, that an interpolation error of approximately 0.3 m results for an Euclidean distance between two supporting points of 30 m.

The highly accurate map, described in [13], was used as ground truth and the Euclidian distance between the estimated curve and the true map was calculated. The distance is assumed to be normally distributed. Mean and standard deviation are computed to evaluate the quality of the solution. After a first run the mean was \( \mu_0 = 4.6 \) m and the standard deviation \( \sigma_0 = 6 \) m. The spline parameters were updated with position measurements of 10 runs. Each includes around 40 position measurements. The mean geometric error of the resulting curve decreases to \( \mu_{10} = 0.8 \) m while the standard deviation is \( \sigma_{10} = 0.7 \) m after all update steps were calculated. The results are depicted in Figure 10.

VI. SUMMARY AND OUTLOOK

This contribution presents an approach combining geometrical a-priori knowledge about the shape of curved structures and
corrupted curve position measurements to estimate the exact geometric progression. Therefore a framework is presented to model track elements with cubic spline curves and approximate the geometric probabilities of the real word precisely.

The section-wise description of the spline enables a clear modelling and the computation of the parameter tracking problem is carried out straightforward, even if the number of supporting points increases. The resulting representation provides information about the expected geometric progression and the uncertainty of the estimated curve.

The method proposed in this contribution results in a linear formulation of the estimation task that offers itself to standard optimal estimation techniques. The estimation is realized with a classical Kalman filter. An orthogonal measurement residuum is minimized while the curve parameter vector is updated. The update step is processed immediately, when new measurement information is available. Results for simulated data and real GPS position measurements are presented.

In future work a focus will be on object tracking algorithms that combine the map information source with other sensor information to achieve precise fusion results. Moreover the linear formulation of the estimation task will be included in a general linear regression framework to compare the solution archived with the Kalman filter versus other estimation approaches.

REFERENCES