TWO-DIMENSIONAL TRACKING OF TRAJECTORIES OF VEHICLES AND OBSTACLES FOR A HEADWAY ALERT APPLICATION

Lars Hoehmann∗, Anton Kummert∗, and Su-Birm Park+

∗Department of Electrical, Information and Media Engineering
University of Wuppertal
42119 Wuppertal, Germany
{hoehmann | kummert}@uni-wuppertal.de

+ Delphi Electronics & Safety
42097 Wuppertal, Germany
su.birm.park@delphi.com

ABSTRACT

In this article we present a tracking algorithm for estimating and filtering two-dimensional trajectories of vehicles and various other obstacles that may occur in a traffic scenario. This is important for developing a robust headway alert application which need reliable position and velocity information of potential dangerous obstacles in the path of a car. The algorithm is based on range and angular data from a laser range finder which is mounted in front of the host car. This data are filtered by a Kalman filter and processed in order to generate reliable obstacle distance and movement information.

1. HEADWAY ALERT APPLICATION

The so called two second warning or headway alert is the main application for a laser range finder used in this paper. Two second warning is a particular type of headway alert systems. It is an essential step in collision avoidance system evolution. In this evolution the most challenging system is the collision avoidance. It is a high-end safety system that combines lane-keeping systems, 360° vehicle coverage to detect every obstacle and active interaction in the steering and braking. In contrast headway alert is just a forward looking object detecting system. The system’s intended operation field is on highway-like roadway environments. There is no active interaction like braking or accelerating - what an ACC1 system applies. Currently there are no headway alert systems available for series-production vehicles.

Figure 1 illustrates the basic processing steps of a

headway alert system. It starts with the detection of potential targets. Detections are provided by a forward looking sensor unit. For this application a laser range finder is used.

The laser range finder provides raw range data which is pre-processed in the detection unit and additionally rough angular information. Target tracking provides the capability of assigning detections to managed targets according to similar attributes. Also new targets are formed and coasted targets are deleted. Path estimation receives host car’s movement data (yaw rate and velocity) and estimates its path. The warning system is responsible for determining whether a warning is necessary and the decision process is based on tracking information and path estimation. The driver determines the function of the system by setting his favored “timed headway”.

The resulting task could be reduced to the above mentioned calculation of the required headway based on speed of the host car and the measured distance, which would lead to implementability in a cheap and cost effective way. In order to prevent the driver from fatiguing by false alarms that are caused by mis-detections and irrelevant obstacles, only clearly verified objects should cause warnings. Irrelevant obstacles can be non-moving objects like:

• guard-rails,
• traffic signs,
• signposts,
• pavement,
• trees and bushes,
• road work equipment,
• mailboxes.

Some further determinations are necessary for moving objects. Only objects driving in the host car’s estimated path are potential threats. To deal with these problems a tracking system is essential. This system tracks detected objects and delivers more reliable target information. If there is a true object in front of the car, it will be detected several times. So the system tracks its position and the warning system is fed with more feasible data. But if there are randomized detections over time and space, a tracking system does not track any objects.

2. SENSOR UNIT: LASER RANGE FINDER

The laser range finder is one key component of the warning system. It is used as a distance measuring unit to detect objects in front of the car. This sensor unit is a prototype for

©2007 EURASIP
330 EUSIPCO, Poznań 2007
automotive purposes. The system is designed to cover a horizontal 7° field of view. This is achieved by five adjacent detection areas which are also called beams. For each beam one laser detector assembly is responsible. The sensor beams are divided into
• a center beam,
• a left and a right inner beam,
• a left and a right outer beam.
The left and right inner beam are situated next to the center beam. These are enclosed by the left and right outer beams. Figure 2 part a) illustrates the alignment of the five beams. Part b) of figure 2 shows the dimension of the horizontal field of view. The left outer beam is controlled by detector #5, left inner beam by detector #4, center beam by detector #3, and so on.

Because the system consists of just one transmitter and of 5 discrete detectors the five beams are processed one after the other. Processing is not done at once - by parallel read out but by serial read out. The transmitter assembly sends out one signal and detector device #1 is active and receives the signal. Then the next signal is sent out by the transmitter and detector device #2 is polled. This is done for all five detectors. These five definitions define one scan cycle and these scan cycles are repeated permanently. The scan direction is marked by an arrow in figure 2 part b). An FPGA and microcontroller based system beyond these sensor assembly controls the sensors’ workflow, calculates range data and provides them via an interface to external applications.

3. KALMAN FILTER

The Kalman Filter is a recursive filter which estimates the state for a dynamic system. This estimation is based on incomplete or noisy measurements of the system. The first paper about Kalman Filter has been published in 1960 by Rudolph E. Kalman and deals with linear and time-discrete systems [1]. It is used in many engineering applications, from radar to computer vision. Many examples deal with movement estimation for aircrafts and base on measurements from radar sensors. One advantage of Kalman Filters is its recursivity. It has just to store the last state parameters and this makes it interesting for real-time applications.

The Kalman Filter is based on linear algebra and the hidden Markov model (hmm). The hidden Markov model claims that the state parameters of a system are unknown. The objective of the hidden Markov model is to determine unknown state parameters from observable parameters. These observable parameters are measurements provided by a sensor that is looking to the system, for example. The assumption of Kalman Filters is that these measurements are perturbed by white noise. White noise is also an additional influence on the system. [3, 4] The Kalman Filter-algorithm operates with two kind of information. On the one hand the algorithm needs estimated state parameters from previous time steps. This is expressed by estimated state vector \( \hat{x}(k) \) and error covariance matrix \( P(k) \). On the other hand a current measurement \( z(k) \) is needed.

The algorithm is subdivided into two stages: Predict and Update. Within the predict stage the algorithm calculates an estimation for the current state based on estimation from the previous state. This is also called time-update. The second stage, the update stage uses predicted state parameters from the predict stage, includes the current measurement (if there is a measurement) and updates state parameters. If there is no actual true measurement available an upper-lever logic has to manage this problem (track manager). After update stage - also called measurement update - the loop starts again in the predict stage with an increased time parameter \( k \).

The term

\[
\hat{x}(k|k-1) = F(k)\hat{x}(k-1|k-1) + B(k)u(k) \tag{1}
\]

describes the prediction of the current state in predict stage, where \( \hat{x}(k-1) \) is the predicted state for the timestep \( k-1 \) under the assumption of previous timestep \( k-1 \). The state \( \hat{x}(k-1|k-1) \) is the estimated state from the previous timestep and \( F(k) \) is the state transition matrix. \( B(k) \) denotes the control input matrix and \( u(k) \) the input vector in a common system. The prediction error is included in the predicted error covariance matrix and calculated based on the previous estimated error covariance matrix by

\[
P(k|k-1) = F(k)P(k-1|k-1)F^T(k) + Q(k) \tag{2}
\]

where \( P(k-1|k-1) \) is previous error covariance matrix and \( Q(k) \) denotes the process noise covariance matrix. These two steps represent the predict stage.

Within the update stage the algorithm calculates different parameters. First measurement residual is calculated by

\[
\tilde{y}(k) = z(k) - H(k)\hat{x}(k|k-1)
\]

which is the difference of true measurement (\( z(k) \)) and measurement prediction. The measurement prediction is predicted state multiplied with measurement matrix \( H(k) \). In the next step residual covariance is determined by

\[
S(k) = H(k)P(k|k-1)H^T(k) + R(k)
\]

where \( P(k|k-1) \) is the error covariance matrix defined in the predict stage. Here the influence of the measurement noise occurs, defined by \( R(k) \). The next step is a fundamental step. The Kalman filter gain is calculated by

\[
K(k) = P(k|k-1)H^T(k)S^{-1}(k)
\]
This Kalman filter gain expresses how much the true measurement influences the current state. The influence can be seen in the next calculation step - the calculation of the updated state

\[ \hat{x}(k|k) = \hat{x}(k|k-1) + K(k)\gamma. \]

Here the final state is determined based on the predicted state and combined with the weighted - by the filter gain - residual. Finally error covariance matrix is updated in equation

\[ P(k|k) = (I - K(k)H(k))P(k|k-1). \]

If this equation is expanded to

\[ P(k|k) = P(k|k-1) - K(k)H(k)P(k|k-1) \]

it can be easily seen that the updated error covariance matrix is defined by the error covariance matrix from the predict stage and weighted by the Kalman filter gain.

In reality many dynamic systems do not exactly fit this model. But one benefit of the Kalman filter is that it works with the existence of noise. So the uncertainty of system’s dynamics can be expressed by systems noise. Here is one exemplary system: A sensor measures the distance to a vehicle in front which is tracked via a Kalman filter based tracker. As long as the vehicle is moving with a constant acceleration the filter is able to provide a true estimation. If the acceleration is changing the filter is not able to work perfectly. But because the acceleration may not change significantly this uncertainty can be handled by the system noise.

4. TRACKING SYSTEM

4.1 Track File Manager

The track file manager is a software system which is implemented to manage detection, identified tracks and perform the Kalman filter algorithm. Tracks are identified obstacles which are managed by the system - in contrast to detections which are measurements of the sensor unit [5, 6, 7]. Figure 3 illustrates the whole tracking process. First all raw data is cleaned by filtering out invalid measurements (low figure of merit, invalid distance data, etc.) Then detections are merged to groups to reduce calculation and memory effort. This is performed by a special logic which combines proximate detections to one detection group. The track prediction process is defined by Kalman filter prediction step. Data association assigns detection groups to active and predicted tracks. There are many different algorithms for this process available (see [2] and [3]). One of these is the nearest neighbor algorithm, which uses distance information of detections and tracks for deciding whether a detection is assigned to a certain track or if not. The nearest neighbor algorithm is adapted for this tracking application by connecting the pair of detection group and managed track with the lowest normalized distance. The following equation describes the calculation of the normalized distance:

\[ d_f = \frac{1}{6} \left[ 3d_r + 2d_a + 1d_f \right] \]

The distance bases on weighted difference in range \( d_r \), weighted difference in angle \( d_a \) and weighted difference in figure of merit \( d_f \). Each of these differences is normalized to a range of 0 to 1 and weighted with a special factor in respect of their expected accuracy. This association process is followed by the track update or Kalman update step. Within this step the predicted track information is updated by real measurements from the previous step. Finally a track file maintenance process is performed to manage track initialization, track termination and track parameter updates.

4.2 2D-Tracking-Filter

Two basic parts of the track file manager are the kalman predict step and the kalman update step. This basic functionality is adapted to a two-dimensional tracking system. The sensor unit provides range data and angular data. There are two approaches to track detected objects:

1. Transform the range data and the angular data into 2D plane coordinates (x and y) and track within this system
2. Keep the range data and angular data and track based on these information

Figure 3: Track File Manager Processing

In this work the second approach is used.

The equation for the state calculation (see equation 1) is reduced to

\[ \hat{x}(k|k-1) = F(k)x(k-1|k-1). \]

This equation has no input vector \( u(k) \) because the system’s input is unknown (driver, traffic condition). The state vector for this tracking system is given by

\[ x(k) = \begin{pmatrix} r(k) \\ \dot{r}(k) \\ a(k) \\ \dot{a}(k) \end{pmatrix} \]
where $r(k)$ is the currently expected range, $a(k)$ the currently expected angle, $\dot{r}(k)$ the range rate and $\dot{a}(k)$ the angle rate. The state transition matrix is described by

$$F(k) = \begin{bmatrix} 1 & \Delta & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta \\ 0 & 0 & 0 & 1 \end{bmatrix} ,$$

where $\Delta$ denotes the time difference between the current measurement and the last measurement. The top left part of the matrix influences range state information and down right part influences angular state information. The system noise matrix $Q(k)$ is used to model the system’s dynamic. This influences the predicted state covariance (see equation 2)

$$P(k|k-1) = F(k)P(k-1|k-1)F^T(k) + Q(k).$$

$Q(k)$ contains the vehicle dynamic regarding range variation and angular variation. These are given by

$$Q(k) = \begin{bmatrix} \sigma^2_a \Delta^4 & \sigma^2_a \Delta^3 & 0 & 0 \\ \sigma^2_a \Delta^3 & \sigma^2_a \Delta^2 & 0 & 0 \\ 0 & 0 & \sigma^2_a \Delta^4 & \sigma_a^2 \Delta^3 \\ 0 & 0 & \sigma^2_a \Delta^3 & \sigma^2_a \Delta^2 \end{bmatrix} .$$

The top left part of equation (3) describes a constant acceleration $Q(k)$, which is the error covariance matrix of $q(k)$ with

$$q(k) = \begin{bmatrix} \Delta^3 \\ \Delta^2 \\ \Delta \\ 1 \end{bmatrix}a(k) ,$$

where $a(k)$ is normally distributed, with mean 0 and standard deviation $\sigma_a$ and describes constant acceleration. The top right part of equation (3) characterizes angular deviation. It has the same dynamic model like the range model, but uses its own standard deviation for the normally distributed system: $\sigma_q$, $\sigma_a$ and $\sigma_\theta$ influence the performance of the Kalman Filter. They have been set after several test runs to an acceptable value for this tracking application.

For the Kalman Update the following definitions are important

$$z(k) = \begin{bmatrix} r_c \\ \alpha_c \end{bmatrix} ,$$

$z(k)$ represents the vector of raw measurements for range ($r_c$) and angle($\alpha_c$). The measurement matrix $H(k)$ transforms the state vector to a simplified vector - containing range and angle

$$H(k) = \begin{bmatrix} 1000 \\ 0010 \end{bmatrix} .$$

The measurement noise vector $R(k)$ is given by

$$R(k) = \begin{bmatrix} \sigma_r^2 & \sigma_r \sigma_a \\ \sigma_r \sigma_a & \sigma_a^2 \end{bmatrix} ,$$

where the $\sigma$-terms describe measurement errors and express the order of inaccuracy of the sensor unit. All other matrices are derived from these illustrated matrices. So the equations can be summarized to:

5. RESULTS

This chapter provides an overview of the performance and test results of the implemented tracking system. It demonstrates the life cycle of detection data from raw detection to tracked objects.

Figure 4 shows a raw data test set which has not been filtered yet. The x-axis represent the index and the y-axis the measured ranges. Different markers (cross, x, dot) mark different detection beams. This figure already suggests potential target’s range trends.

In contrast figure 5 demonstrates the result of the cleaning process. This figure shows less detections in the near field. Especially in the index range of 0 to 1500 most of the detections in the near field (up to 40 meters) are filtered out. This figure already shows the contour of the ranges of a potential target.

The figures represent a long time frame. So the distances of potential targets are steeply rising and falling. Figure 6 demonstrates the result of the tracking process.

It is based on the filtered data from the raw data cleaning process. The figure shows a reduced number of crosses. This is achieved by grouping single detection which has fulfilled the grouping criterion and by performed tracking. The first target is marked by crosses. It already represents the tracked range of a detected true object.

X marks the second recognized target. In this figure they represent sign posts along the motorway. This is indicated by
the high steeply falling ranges, which is caused by their high negative speeds.

6. CONCLUSIONS

A new application for a Kalman filter in an automotive scenario is presented in this paper. First the application headway alert is presented, which is a type of active safety system for automotive applications. Then a special sensor unit - a laser range finder - is described and the basics of Kalman filters are explained. The implementation of a track file manager, which basically includes a Kalman filter, is shown in the next chapters - followed by achieved results.

This work shows that the Kalman filter based track file manager fulfilled the qualifications for a 2D-tracking application - a headway alert system. The Kalman filter is an algorithm, which has low requirements in memory effort and complexity. This system, which is presented in this paper, is running offline and without direct driver feedback. So there is still future work to do:

- optimization of the filter performance and of the sensor unit
- extension of the driver feedback devices

Due to the prototype status of the laser range finder, further improvements of this hardware unit are necessary.

REFERENCES